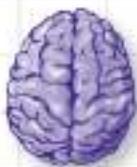


A Brain-Friendly Guide

Head First Python



Load important
programming concepts
directly into your brain

Preserve
your data
in a pickle



Move your custom
app to the Web



Model data
with lists,
sets, and
dictionaries

Hook up with
JSON, Android,
and App Engine



Share your code with
the world on PyPI

O'REILLY®

Paul Barry

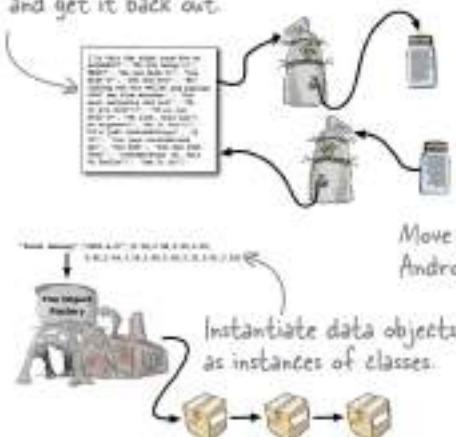
Head First Python

Python/Programming Languages

What will you learn from this book?

Ever wished you could learn Python from a book? *Head First Python* helps you learn the language through a unique method that goes beyond syntax and how-to manuals. You'll quickly grasp Python's fundamentals, then move on to persistence, exception handling, web development, SQLite, data wrangling, and Google App Engine. You'll also learn how to write mobile apps for Android, all thanks to the power that Python gives you. *Head First Python* is a complete learning experience that will help you become a bona fide Python programmer.

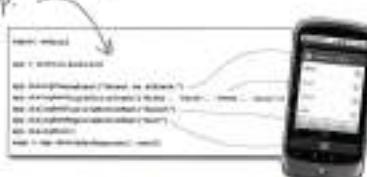
Get your data into a pickle...
and get it back out.



Port your scripts from the Python shell to the Web.



Move your webapp to phones as an
Android app.



Why does this book look so different?

We think your time is too valuable to waste struggling with new concepts. Using the latest research in cognitive science and learning theory to craft a multi-sensory learning experience, *Head First Python* uses a visually rich format designed for the way your brain works, not a text-heavy approach that puts you to sleep.

US \$49.99

CAN \$57.99

ISBN: 978-1-449-38267-4



Safari
Books Online

Free online edition
for 45 days with
purchase of this book.
Details on last page.

"*Head First Python* is a great introduction to not just the Python language, but Python as it's used in the real world. The book goes beyond the syntax to teach you how to create applications for Android phones, Google's App Engine, and more."

—David Griffiths,
author and Agile coach

"Where other books start with theory and progress to examples, *Head First Python* jumps right in with code and explains the theory as you read along. The breadth of examples and explanation cover the majority of what you'll use in your job every day."

—Jeremy Jones, coauthor of
Python for Unix and Linux
System Administrators

O'REILLY®
oreilly.com
headfirstlabs.com

Advance Praise for *Head First Python*

“*Head First Python* is a great introduction to not just the Python language, but Python as it’s used in the real world. The book goes beyond the syntax to teach you how to create applications for Android phones, Google’s App Engine, and more.”

— **David Griffiths, author and Agile coach**

“Where other books start with theory and progress to examples, *Head First Python* jumps right in with code and explains the theory as you read along. This is a much more effective learning environment, because it engages the reader to *do* from the very beginning. It was also just a joy to read. It was fun without being flippant and informative without being condescending. The breadth of examples and explanation covered the majority of what you’ll use in your job every day. I’ll recommend this book to anyone starting out on Python.”

— **Jeremy Jones, coauthor of *Python for Unix and Linux System Administration***

“*Head First Python* is a terrific book for getting a grounding in a language that is increasing in relevance day by day.”

— **Phil Hartley, University of Advancing Technology**

Praise for other *Head First* books

“Kathy and Bert’s *Head First Java* transforms the printed page into the closest thing to a GUI you’ve ever seen. In a wry, hip manner, the authors make learning Java an engaging ‘what’re they gonna do next?’ experience.”

— **Warren Keuffel, Software Development Magazine**

“Beyond the engaging style that drags you forward from know-nothing into exalted Java warrior status, *Head First Java* covers a huge amount of practical matters that other texts leave as the dreaded ‘exercise for the reader....’ It’s clever, wry, hip and practical—there aren’t a lot of textbooks that can make that claim and live up to it while also teaching you about object serialization and network launch protocols.”

— **Dr. Dan Russell, Director of User Sciences and Experience Research
IBM Almaden Research Center (and teaches Artificial Intelligence at
Stanford University)**

“It’s fast, irreverent, fun, and engaging. Be careful—you might actually learn something!”

— **Ken Arnold, former Senior Engineer at Sun Microsystems
Coauthor (with James Gosling, creator of Java), *The Java Programming
Language***

“I feel like a thousand pounds of books have just been lifted off of my head.”

— **Ward Cunningham, inventor of the Wiki and founder of the Hillside Group**

“Just the right tone for the geeked-out, casual-cool guru coder in all of us. The right reference for practical development strategies—gets my brain going without having to slog through a bunch of tired, stale professor-speak.”

— **Travis Kalanick, founder of Scour and Red Swoosh
Member of the MIT TR100**

“There are books you buy, books you keep, books you keep on your desk, and thanks to O’Reilly and the Head First crew, there is the penultimate category, Head First books. They’re the ones that are dog-eared, mangled, and carried everywhere. *Head First SQL* is at the top of my stack. Heck, even the PDF I have for review is tattered and torn.”

— **Bill Sawyer, ATG Curriculum Manager, Oracle**

“This book’s admirable clarity, humor and substantial doses of clever make it the sort of book that helps even non-programmers think well about problem-solving.”

— **Cory Doctorow, co-editor of Boing Boing
Author, *Down and Out in the Magic Kingdom*
and *Someone Comes to Town, Someone Leaves Town***

Praise for other *Head First* books

“I received the book yesterday and started to read it...and I couldn’t stop. This is definitely très ‘cool.’ It is fun, but they cover a lot of ground and they are right to the point. I’m really impressed.”

— **Erich Gamma, IBM Distinguished Engineer, and coauthor of *Design Patterns***

“One of the funniest and smartest books on software design I’ve ever read.”

— **Aaron LaBerge, VP Technology, ESPN.com**

“What used to be a long trial and error learning process has now been reduced neatly into an engaging paperback.”

— **Mike Davidson, CEO, Newsvine, Inc.**

“Elegant design is at the core of every chapter here, each concept conveyed with equal doses of pragmatism and wit.”

— **Ken Goldstein, Executive Vice President, Disney Online**

“I ♥ *Head First HTML with CSS & XHTML*—it teaches you everything you need to learn in a ‘fun-coated’ format.”

— **Sally Applin, UI Designer and Artist**

“Usually when reading through a book or article on design patterns, I’d have to occasionally stick myself in the eye with something just to make sure I was paying attention. Not with this book. Odd as it may sound, this book makes learning about design patterns fun.

“While other books on design patterns are saying ‘Bueller...Bueller...Bueller...’ this book is on the float belting out ‘Shake it up, baby!’”

— **Eric Wuehler**

“I literally love this book. In fact, I kissed this book in front of my wife.”

— **Satish Kumar**

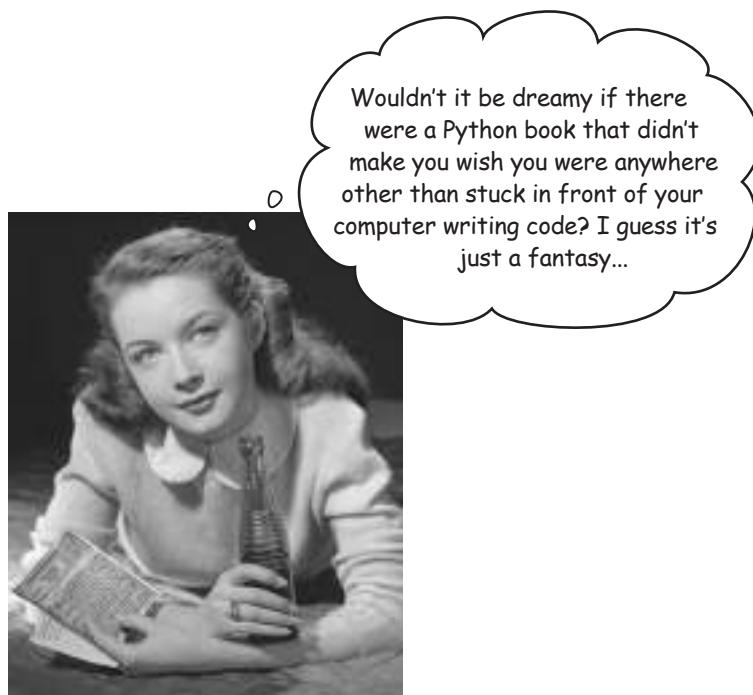
Other related books from O'Reilly

Learning Python
Programming Python
Python in a Nutshell
Python Cookbook
Python for Unix and Linux System Administration

Other books in O'Reilly's *Head First* series

Head First Algebra
Head First Ajax
Head First C#, Second Edition
Head First Design Patterns
Head First EJB
Head First Excel
Head First 2D Geometry
Head First HTML with CSS & XHTML
Head First iPhone Development
Head First Java
Head First JavaScript
Head First Object-Oriented Analysis & Design (OOA&D)
Head First PHP & MySQL
Head First Physics
Head First PMP, Second Edition
Head First Programming
Head First Rails
Head First Servlets & JSP, Second Edition
Head First Software Development
Head First SQL
Head First Statistics
Head First Web Design
Head First WordPress

Head First Python



Wouldn't it be dreamy if there
were a Python book that didn't
make you wish you were anywhere
other than stuck in front of your
computer writing code? I guess it's
just a fantasy...

Paul Barry

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Head First Python

by Paul Barry

Copyright © 2011 Paul Barry. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly Media books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: (800) 998-9938 or corporate@oreilly.com.

Series Creators:

Kathy Sierra, Bert Bates

Editor:

Brian Sawyer

Cover Designer:

Karen Montgomery

Production Editor:

Rachel Monaghan

Proofreader:

Nancy Reinhardt

Indexer:

Angela Howard

Page Viewers:

Deirdre, Joseph, Aaron, and Aideen



Deirdre
↗



Joseph
↗



Aaron
↑



Aideen
↓

Printing History:

November 2010: First Edition.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. The *Head First* series designations, *Head First Python*, and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

No athletes were pushed too hard in the making of this book.



RepKover™ This book uses RepKover™, a durable and flexible lay-flat binding.

ISBN: 978-1-449-38267-4

[M]

I dedicate this book to all those generous people in the Python community who have helped to make this great little language the *first-rate* programming technology it is.

And to those that made learning Python and its technologies just complex enough that people need a book like this to learn it.

Author of Head First Python



Paul Barry recently worked out that he has been programming for close to a quarter century, a fact that came as a bit of a shock. In that time, Paul has programmed in lots of different programming languages, lived and worked in two countries on two continents, got married, had three kids (well...his wife Deirdre actually *had them*, but Paul was there), completed a B.Sc. and M.Sc. in Computing, written or cowritten three other books, as well as a bunch of technical articles for *Linux Journal* (where he's a Contributing Editor).

When Paul first saw *Head First HTML with CSS & XHTML*, he loved it so much he knew immediately that the Head First approach would be a great way to teach programming. He was only too delighted then, together with David Griffiths, to create *Head First Programming* in an attempt to prove his hunch correct.

Paul's day job is working as a lecturer at The Institute of Technology, Carlow, in Ireland. As part of the Department of Computing and Networking, Paul gets to spend his day exploring, learning, and teaching programming technologies to his students, including Python.

Paul recently completed a post-graduate certificate in Learning and Teaching and was more than a bit relieved to discover that most of what he does conforms to current third-level best practice.

Table of Contents (Summary)

	Intro	xxiii
1	Meet Python: <i>Everyone Loves Lists</i>	1
2	Sharing Your Code: <i>Modules of Functions</i>	33
3	Files and Exceptions: <i>Dealing with Errors</i>	73
4	Persistence: <i>Saving Data to Files</i>	105
5	Comprehending Data: <i>Work That Data!</i>	139
6	Custom Data Objects: <i>Bundling Code with Data</i>	173
7	Web Development: <i>Putting It All Together</i>	213
8	Mobile App Development: <i>Small Devices</i>	255
9	Manage Your Data: <i>Handling Input</i>	293
10	Scaling Your Webapp: <i>Getting Real</i>	351
11	Dealing with Complexity: <i>Data Wrangling</i>	397
i	Leftovers: <i>The Top Ten Things (We Didn't Cover)</i>	435

Table of Contents (the real thing)

Intro

Your brain on Python. Here you are trying to learn something, while here your brain is doing you a favor by making sure the learning doesn't stick. Your brain's thinking, "Better leave room for more important things, like which wild animals to avoid and whether naked snowboarding is a bad idea." So how do you trick your brain into thinking that your life depends on knowing Python?

Who is this book for?	xxiv
We know what you're thinking	xxv
Metacognition	xxvii
Bend your brain into submission	xxix
Read me	xxx
The technical review team	xxxii
Acknowledgments	xxxiii

meet python

Everyone loves lists

1

You're asking one question: "What makes Python different?"

The short answer is: *lots of things*. The longer answers starts by stating that there's lots that's familiar, too. Python is a lot like any other *general-purpose* programming language, with **statements**, **expressions**, **operators**, **functions**, **modules**, **methods**, and **classes**. All the *usual stuff*, really. And then there's the other stuff Python provides that makes the programmer's life—*your life*—that little bit easier. You'll start your tour of Python by learning about **lists**. But, before getting to that, there's another important question that needs answering...

What's to like about Python?	2
Install Python 3	3
Use IDLE to help learn Python	4
Work effectively with IDLE	5
Deal with complex data	6
Create simple Python lists	7
Lists are like arrays	9
Add more data to your list	11
Work with your list data	15
For loops work with lists of any size	16
Store lists within lists	18
Check a list for a list	20
Complex data is hard to process	23
Handle many levels of nested lists	24
Don't repeat code; create a function	28
Create a function in Python	29
Recursion to the rescue!	31
Your Python Toolbox	32

The Holy Grail, 1975, Terry Jones & Terry Gilliam, 91 mins

Graham Chapman

Michael Palin, John Cleese, Terry Gilliam, Eric Idle & Terry Jones

2

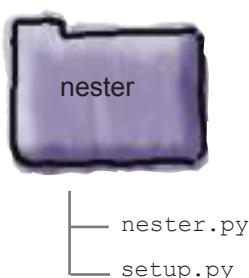
sharing your code

Modules of functions

Reusable code is great, but a shareable module is better.

By sharing your code as a Python module, you open up your code to the entire Python community...and it's always good to share, isn't it? In this chapter, you'll learn how to create, install, and distribute your own shareable modules. You'll then load your module onto Python's software sharing site on the Web, so that *everyone* can benefit from your work. Along the way, you'll pick up a few new tricks relating to Python's functions, too.

It's too good not to share	34
Turn your function into a module	35
Modules are everywhere	36
Comment your code	37
Prepare your distribution	40
Build your distribution	41
A quick review of your distribution	42
Import a module to use it	43
Python's modules implement namespaces	45
Register with the PyPI website	47
Upload your code to PyPI	48
Welcome to the PyPI community	49
Control behavior with an extra argument	52
Before you write new code, think BIF	53
Python tries its best to run your code	57
Trace your code	58
Work out what's wrong	59
Update PyPI with your new code	60
You've changed your API	62
Use optional arguments	63
Your module supports both APIs	65
Your API is still not right	66
Your module's reputation is restored	70
Your Python Toolbox	71



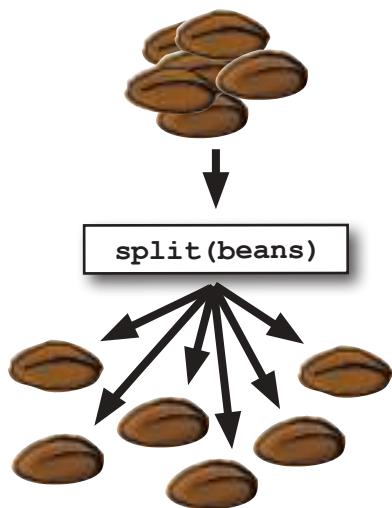
3

files and exceptions

Dealing with errors

It's simply not enough to process your list data in your code. You need to be able to get your data *into* your programs with ease, too. It's no surprise then that Python makes reading data from **files** easy. Which is great, until you consider what can go *wrong* when interacting with data *external* to your programs... and there are lots of things waiting to trip you up! When bad stuff happens, you need a strategy for getting out of trouble, and one such strategy is to deal with any exceptional situations using Python's **exception handling** mechanism showcased in this chapter.

Data is external to your program	74
It's all lines of text	75
Take a closer look at the data	77
Know your data	79
Know your methods and ask for help	80
Know your data (better)	82
Two very different approaches	83
Add extra logic	84
Handle exceptions	88
Try first, then recover	89
Identify the code to protect	91
Take a pass on the error	93
What about other errors?	96
Add more error-checking code...	97
...Or add another level of exception handling	98
So, which approach is best?	99
You're done...except for one small thing	101
Be specific with your exceptions	102
Your Python Toolbox	103



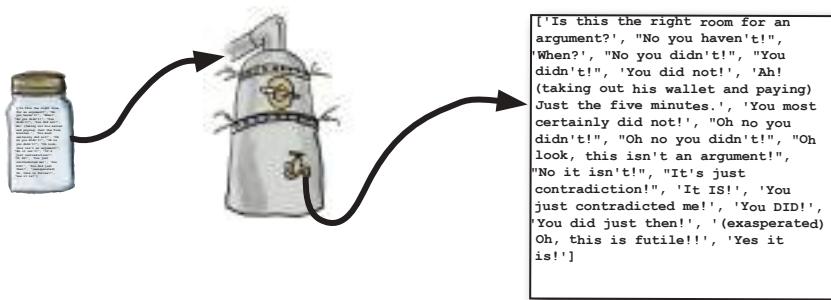
4 persistence

Saving data to files

It is truly great to be able to process your file-based data.

But what happens to your data when you're done? Of course, it's best to save your data to a disk file, which allows you to use it again at some later date and time. Taking your memory-based data and storing it to disk is what **persistence** is all about. Python supports all the usual tools for writing to files and also provides some cool facilities for *efficiently* storing Python data.

Programs produce data	106
Open your file in write mode	110
Files are left open after an exception!	114
Extend try with finally	115
Knowing the type of error is not enough	117
Use with to work with files	120
Default formats are unsuitable for files	124
Why not modify print_lol()?	126
Pickle your data	132
Save with dump and restore with load	133
Generic file I/O with pickle is the way to go!	137
Your Python Toolbox	138



5

comprehending data

Work that data!

Data comes in all shapes and sizes, formats and encodings.

To work effectively with your data, you often have to manipulate and transform it into a common format to allow for efficient processing, sorting, and storage. In this chapter, you'll explore Python goodies that help you work your data up into a sweat, allowing you to achieve data-munging greatness.

Coach Kelly needs your help	140
Sort in one of two ways	144
The trouble with time	148
Comprehending lists	155
Iterate to remove duplicates	161
Remove duplicates with sets	166
Your Python Toolbox	172



This chapter's
guaranteed to give you
a workout!

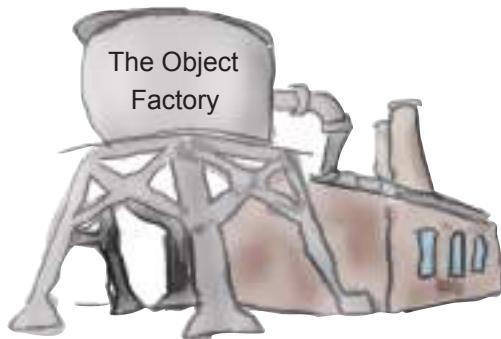
custom data objects

6

Bundling code with data

It's important to match your data structure choice to your data. And that choice can make a big difference to the complexity of your code. In Python, although really useful, lists and sets aren't the only game in town. The Python **dictionary** lets you organize your data for speedy lookup by *associating your data with names*, not numbers. And when Python's built-in data structures don't quite cut it, the Python **class** statement lets you define your own. This chapter shows you how.

Coach Kelly is back (with a new file format)	174
Use a dictionary to associate data	178
Bundle your code and its data in a class	189
Define a class	190
Use class to define classes	191
The importance of self	192
Every method's first argument is self	193
Inherit from Python's built-in list	204
Coach Kelly is impressed	211
Your Python Toolbox	212



7

web development

Putting it all together

Sooner or later, you'll want to share your app with lots of people. You have many options for doing this. Pop your code on PyPI, send out lots of emails, put your code on a CD or USB, or simply install your app manually on the computers of those people who need it. Sounds like a lot of work...not to mention boring. Also, what happens when you produce the next best version of your code? What happens then? How do you manage the update? Let's face it: it's such a pain that you'll think up really creative excuses not to. Luckily, you don't have to do any of this: just create a webapp instead. And, as this chapter demonstrates, using Python for web development is a breeze.

It's good to share	214
You can put your program on the Web	215
What does your webapp need to do?	218
Design your webapp with MVC	221
Model your data	222
View your interface	226
Control your code	234
CGI lets your web server run programs	235
Display the list of athletes	236
The dreaded 404 error!	242
Create another CGI script	244
Enable CGI tracking to help with errors	248
A small change can make all the difference	250
Your webapp's a hit!	252
Your Python Toolbox	253



8

mobile app development

Small devices

Putting your data on the Web opens up all types of possibilities.

Not only can anyone from anywhere interact with your webapp, but they are increasingly doing so from a collection of diverse computing devices: PCs, laptops, tablets, palmtops, and even mobile phones. And it's not just humans interacting with your webapp that you have to support and worry about: *bots* are small programs that can automate web interactions and typically want your data, not your human-friendly HTML. In this chapter, you exploit Python on Coach Kelly's mobile phone to write an app that interacts with your webapp's data.

The world is getting smaller	256
Coach Kelly is on Android	257
Don't worry about Python 2	259
Set up your development environment	260
Configure the SDK and emulator	261
Install and configure Android Scripting	262
Add Python to your SL4A installation	263
Test Python on Android	264
Define your app's requirements	266
The SL4A Android API	274
Select from a list on Android	278
The athlete's data CGI script	281
The data appears to have changed type	284
JSON can't handle your custom datatypes	285
Run your app on a real phone	288
Configure AndFTP	289
The coach is thrilled with his app	290
Your Python Toolbox	291



manage your data

Handling input

9

The Web and your phone are not just great ways to display data.

They are also great tools to for accepting input from your users. Of course, once your webapp accepts data, it needs to put it somewhere, and the choices you make when deciding what and where this “somewhere” is are often the difference between a webapp that’s easy to grow and extend and one that isn’t. In this chapter, you’ll extend your webapp to accept data from the Web (via a browser or from an Android phone), as well as look at and enhance your back-end data-management services.

Your athlete times app has gone national	294
Use a form or dialog to accept input	295
Create an HTML form template	296
The data is delivered to your CGI script	300
Ask for input on your Android phone	304
It’s time to update your server data	308
Avoid race conditions	309
You need a better data storage mechanism	310
Use a database management system	312
Python includes SQLite	313
Exploit Python’s database API	314
The database API as Python code	315
A little database design goes a long way	316
Define your database schema	317
What does the data look like?	318
Transfer the data from your pickle to SQLite	321
What ID is assigned to which athlete?	322
Insert your timing data	323
SQLite data management tools	326
Integrate SQLite with your existing webapp	327
You still need the list of names	332
Get an athlete’s details based on ID	333
You need to amend your Android app, too	342
Update your SQLite-based athlete data	348
The NUAC is over the moon!	349
Your Python Toolbox	350



10

scaling your webapp

Getting real

The Web is a great place to host your app...until things get real.

Sooner or later, you'll hit the jackpot and your webapp will be *wildly successful*. When that happens, your webapp goes from a handful of hits a day to thousands, possibly ten of thousands, *or even more*. Will you be ready? Will your web server handle the **load**? How will you know? What will it **cost**? **Who will pay**? Can your data model scale to millions upon millions of data items without *slowing to a crawl*? Getting a webapp up and running is easy with Python and now, thanks to Google App Engine, **scaling** a Python webapp is achievable, too.

There are whale sightings everywhere	352
The HFWWG needs to automate	353
Build your webapp with Google App Engine	354
Download and install App Engine	355
Make sure App Engine is working	356
App Engine uses the MVC pattern	359
Model your data with App Engine	360
What good is a model without a view?	363
Use templates in App Engine	364
Django's form validation framework	368
Check your form	369
Controlling your App Engine webapp	370
Restrict input by providing options	376
Meet the “blank screen of death”	378
Process the POST within your webapp	379
Put your data in the datastore	380
Don't break the “robustness principle”	384
Accept almost any date and time	385
It looks like you're not quite done yet	388
Sometimes, the tiniest change can make all the difference...	389
Capture your user's Google ID, too	390
Deploy your webapp to Google's cloud	391
Your HFWWG webapp is deployed!	394
Your Python Toolbox	395



11

dealing with complexity

Data wrangling

It's great when you can apply Python to a specific domain area. Whether it's *web development*, *database management*, or *mobile apps*, Python helps you **get the job done** by *not* getting in the way of you coding your solution. And then there's the other types of problems: the ones you can't categorize or attach to a domain. Problems that are in themselves so *unique* you have to look at them in a different, highly specific way. Creating **bespoke** software solutions to these type of problems is an area where Python *excels*. In this, your final chapter, you'll *stretch* your Python skills to the limit and solve problems along the way.

What's a good time goal for the next race?	398
So...what's the problem?	400
Start with the data	401
Store each time as a dictionary	407
Dissect the prediction code	409
Get input from your user	413
Getting input raises an issue...	414
Search for the closest match	416
The trouble is with time	418
The time-to-seconds-to-time module	419
The trouble is still with time...	422
Port to Android	424
Your Android app is a bunch of dialogs	425
Put your app together...	429
Your app's a wrap!	431
Your Python Toolbox	432

A screenshot of Microsoft Excel showing a table titled "Formula: V02". The table contains data for running times across different distances. The columns represent distances: 100m, 200m, 400m, 800m, 1500m, 10km, 20km, and Marathon. The rows represent times in minutes and seconds, such as 54.8 for 100m and 2:29:44 for Marathon. The data is presented in a grid format with rows and columns labeled.

	A	B	C	D	E	F	G	H	I	J
1	100m	54.8	52.9	51.1	50.3	57.5	75.8	74.2	72.5	
2	200m	8.00	8.10	8.21	8.33	8.44	8.55	8.66	8.77	8.88
3	400m	12.49	13.08	13.24	13.42	14.99	14.16	14.33	14.50	14.67
4	800m	21.19	21.80	22.17	22.67	23.18	23.50	24.22	24.55	
5	1500m	28.94	27.30	28.08	28.95	29.24	30.04	30.49	31.25	
6	10km	41.91	42.27	43.24	44.23	45.23	46.26	47.27	48.31	
7	20km	44.46	45.46	46.46	47.51	48.56	50.62	51.69	52.16	
8	20km	58.29	57.45	59.03	1:00.25	1:01.66	1:03.08	1:04.53	1:06.00	
9	50.1mi	59.40	1:01.00	1:02.12	1:03.56	1:05.20	1:06.51	1:08.21	1:09.62	
10	25km	1:11.43	1:19.26	1:19.59	1:18.40	1:16.24	1:20.10	1:21.66	1:23.46	
11	30km	1:27.16	1:19.06	1:35.11	1:35.17	1:37.26	1:39.37	1:41.52		
12	Marathon	2:04.34	2:04.24	2:11.17	2:14.16	2:17.18	2:20.21	2:23.21	2:24.44	

leftovers



The Top Ten Things (we didn't cover)

You've come a long way.

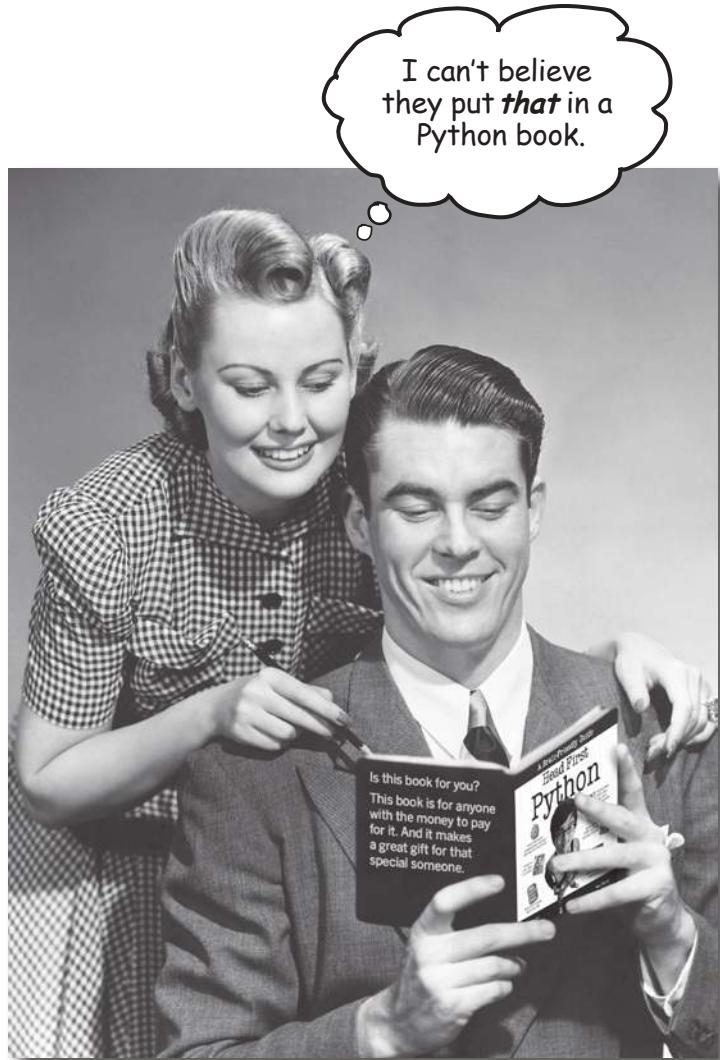
But learning about Python is an activity that never stops. The more Python you code, the more you'll need to learn new ways to do certain things. You'll need to master new tools and new techniques, too. There's just not enough room in this book to show you everything you might possibly need to know about Python. So, here's our list of the top ten things we didn't cover that you might want to learn more about next.

#1: Using a “professional” IDE	436
#2: Coping with scoping	437
#3: Testing	438
#4: Advanced language features	439
#5: Regular expressions	440
#6: More on web frameworks	441
#7: Object relational mappers and NoSQL	442
#8: Programming GUIs	443
#9: Stuff to avoid	444
#10: Other books	445



how to use this book

Intro



In this section, we answer the burning question:
"So why DID they put that in a Python book?"

Who is this book for?

If you can answer “yes” to all of these:

- ➊ Do you already know how to program in another programming language?
- ➋ Do you wish you had the know-how to program Python, add it to your list of tools, and make it do new things?
- ➌ Do you prefer actually doing things and applying the stuff you learn over listening to someone in a lecture rattle on for hours on end?

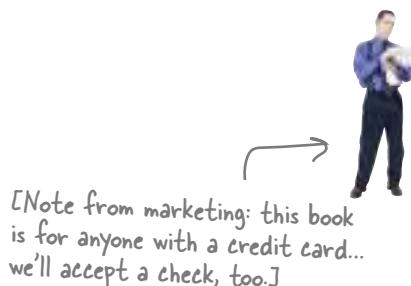
this book is for you.

Who should probably back away from this book?

If you can answer “yes” to any of these:

- ➊ Do you already know most of what you need to know to program with Python?
- ➋ Are you looking for a reference book to Python, one that covers all the details in excruciating detail?
- ➌ Would you rather have your toenails pulled out by 15 screaming monkeys than learn something new? Do you believe a Python book should cover *everything* and if it bores the reader to tears in the process then so much the better?

this book is **not** for you.



We know what you're thinking

“How can *this* be a serious Python book?”

“What’s with all the graphics?”

“Can I actually *learn* it this way?”

We know what your brain is thinking

Your brain craves novelty. It’s always searching, scanning, *waiting* for something unusual. It was built that way, and it helps you stay alive.

So what does your brain do with all the routine, ordinary, normal things you encounter? Everything it *can* to stop them from interfering with the brain’s *real* job—recording things that *matter*. It doesn’t bother saving the boring things; they never make it past the “this is obviously not important” filter.

How does your brain *know* what’s important? Suppose you’re out for a day hike and a tiger jumps in front of you, what happens inside your head and body?

Neurons fire. Emotions crank up. *Chemicals surge*.

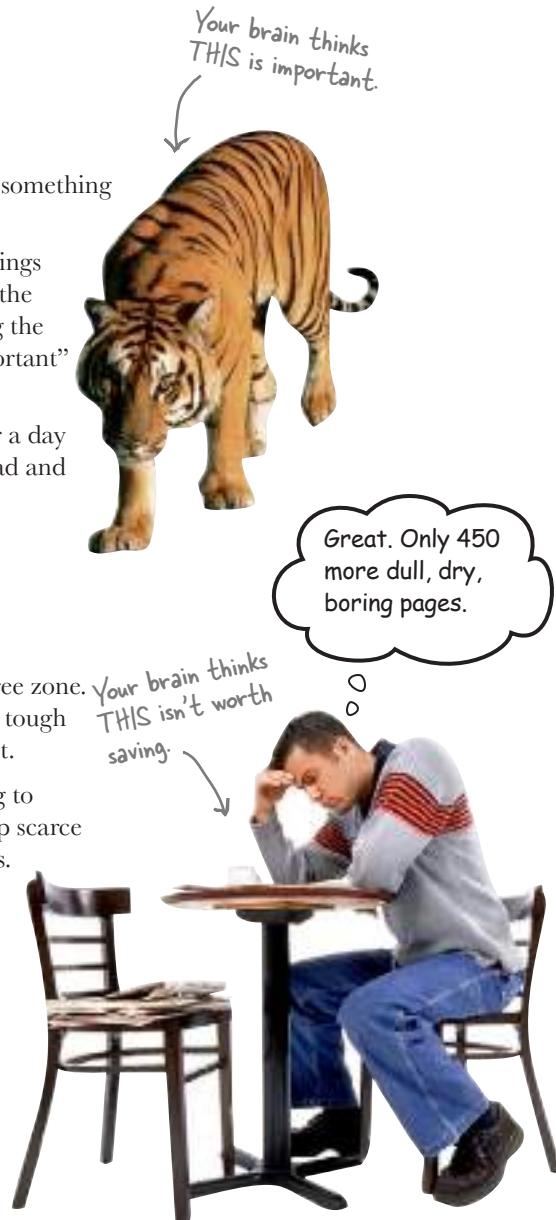
And that’s how your brain knows...

This must be important! Don’t forget it!

But imagine you’re at home, or in a library. It’s a safe, warm, tiger-free zone. You’re studying. Getting ready for an exam. Or trying to learn some tough technical topic your boss thinks will take a week, ten days at the most.

Just one problem. Your brain’s trying to do you a big favor. It’s trying to make sure that this *obviously* non-important content doesn’t clutter up scarce resources. Resources that are better spent storing the really *big* things.

Like tigers. Like the danger of fire. Like how you should never have posted those “party” photos on your Facebook page. And there’s no simple way to tell your brain, “Hey brain, thank you very much, but no matter how dull this book is, and how little I’m registering on the emotional Richter scale right now, I really *do* want you to keep this stuff around.”



We think of a “Head First” reader as a learner.

So what does it take to *learn* something? First, you have to *get it*, then make sure you don’t *forget it*. It’s not about pushing facts into your head. Based on the latest research in cognitive science, neurobiology, and educational psychology, *learning* takes a lot more than text on a page. We know what turns your brain on.

Some of the Head First learning principles:

Make it visual. Images are far more memorable than words alone, and make learning much more effective (up to 89% improvement in recall and transfer studies). It also makes things more understandable. **Put the words within or near the graphics** they relate to, rather than on the bottom or on another page, and learners will be up to twice as likely to solve problems related to the content.

Use a conversational and personalized style. In recent studies, students performed up to 40% better on post-learning tests if the content spoke directly to the reader, using a first-person, conversational style rather than taking a formal tone. Tell stories instead of lecturing. Use casual language. Don’t take yourself too seriously. Which would you pay more attention to: a stimulating dinner party companion or a lecture?

Get the learner to think more deeply. In other words, unless you actively flex your neurons, nothing much happens in your head. A reader has to be motivated, engaged, curious, and inspired to solve problems, draw conclusions, and generate new knowledge. And for that, you need challenges, exercises, and thought-provoking questions, and activities that involve both sides of the brain and multiple senses.

Get—and keep—the reader’s attention. We’ve all had the “I really want to learn this but I can’t stay awake past page one” experience. Your brain pays attention to things that are out of the ordinary, interesting, strange, eye-catching, unexpected. Learning a new, tough, technical topic doesn’t have to be boring. Your brain will learn much more quickly if it’s not.

Touch their emotions. We now know that your ability to remember something is largely dependent on its emotional content. You remember what you care about. You remember when you *feel* something. No, we’re not talking heart-wrenching stories about a boy and his dog. We’re talking emotions like surprise, curiosity, fun, “what the...?”, and the feeling of “I Rule!” that comes when you solve a puzzle, learn something everybody else thinks is hard, or realize you know something that “I’m more technical than thou” Bob from engineering *doesn’t*.

Metacognition: thinking about thinking

If you really want to learn, and you want to learn more quickly and more deeply, pay attention to how you pay attention. Think about how you think. Learn how you learn.

Most of us did not take courses on metacognition or learning theory when we were growing up. We were *expected* to learn, but rarely *taught* to learn.

But we assume that if you're holding this book, you really want to learn how to design user-friendly websites. And you probably don't want to spend a lot of time. If you want to use what you read in this book, you need to *remember* what you read. And for that, you've got to *understand* it. To get the most from this book, or *any* book or learning experience, take responsibility for your brain. Your brain on *this* content.

The trick is to get your brain to see the new material you're learning as Really Important. Crucial to your well-being. As important as a tiger. Otherwise, you're in for a constant battle, with your brain doing its best to keep the new content from sticking.

So just how **DO** you get your brain to treat programming like it was a hungry tiger?

There's the slow, tedious way, or the faster, more effective way. The slow way is about sheer repetition. You obviously know that you *are* able to learn and remember even the dullest of topics if you keep pounding the same thing into your brain. With enough repetition, your brain says, "This doesn't *feel* important to him, but he keeps looking at the same thing *over* and *over* and *over*, so I suppose it must be."

The faster way is to do **anything that increases brain activity**, especially different types of brain activity. The things on the previous page are a big part of the solution, and they're all things that have been proven to help your brain work in your favor. For example, studies show that putting words *within* the pictures they describe (as opposed to somewhere else in the page, like a caption or in the body text) causes your brain to try to make sense of how the words and picture relate, and this causes more neurons to fire. More neurons firing = more chances for your brain to *get* that this is something worth paying attention to, and possibly recording.

A conversational style helps because people tend to pay more attention when they perceive that they're in a conversation, since they're expected to follow along and hold up their end. The amazing thing is, your brain doesn't necessarily *care* that the "conversation" is between you and a book! On the other hand, if the writing style is formal and dry, your brain perceives it the same way you experience being lectured to while sitting in a roomful of passive attendees. No need to stay awake.

But pictures and conversational style are just the beginning...



Here's what WE did:

We used **pictures**, because your brain is tuned for visuals, not text. As far as your brain's concerned, a picture really *is* worth a thousand words. And when text and pictures work together, we embedded the text *in* the pictures because your brain works more effectively when the text is *within* the thing the text refers to, as opposed to in a caption or buried in the text somewhere.

We used **redundancy**, saying the same thing in *different* ways and with different media types, and *multiple senses*, to increase the chance that the content gets coded into more than one area of your brain.

We used concepts and pictures in **unexpected** ways because your brain is tuned for novelty, and we used pictures and ideas with at least *some emotional content*, because your brain is tuned to pay attention to the biochemistry of emotions. That which causes you to *feel* something is more likely to be remembered, even if that feeling is nothing more than a little **humor, surprise, or interest**.

We used a personalized, **conversational style**, because your brain is tuned to pay more attention when it believes you're in a conversation than if it thinks you're passively listening to a presentation. Your brain does this even when you're *reading*.

We included more than 80 **activities**, because your brain is tuned to learn and remember more when you **do** things than when you *read* about things. And we made the exercises challenging-yet-do-able, because that's what most people prefer.

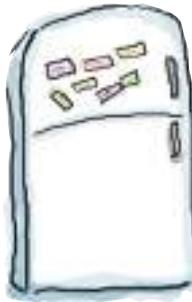
We used **multiple learning styles**, because *you* might prefer step-by-step procedures, while someone else wants to understand the big picture first, and someone else just wants to see an example. But regardless of your own learning preference, *everyone* benefits from seeing the same content represented in multiple ways.

We include content for **both sides of your brain**, because the more of your brain you engage, the more likely you are to learn and remember, and the longer you can stay focused. Since working one side of the brain often means giving the other side a chance to rest, you can be more productive at learning for a longer period of time.

And we included **stories** and exercises that present **more than one point of view**, because your brain is tuned to learn more deeply when it's forced to make evaluations and judgments.

We included **challenges**, with exercises, and by asking **questions** that don't always have a straight answer, because your brain is tuned to learn and remember when it has to *work* at something. Think about it—you can't get your *body* in shape just by *watching* people at the gym. But we did our best to make sure that when you're working hard, it's on the *right* things. That **you're not spending one extra dendrite** processing a hard-to-understand example, or parsing difficult, jargon-laden, or overly terse text.

We used **people**. In stories, examples, pictures, etc., because, well, because *you're* a person. And your brain pays more attention to *people* than it does to *things*.



Cut this out and stick it
on your refrigerator.

Here's what YOU can do to bend your brain into submission

So, we did our part. The rest is up to you. These tips are a starting point; listen to your brain and figure out what works for you and what doesn't. Try new things.

1 Slow down. The more you understand, the less you have to memorize.

Don't just *read*. Stop and think. When the book asks you a question, don't just skip to the answer. Imagine that someone really *is* asking the question. The more deeply you force your brain to think, the better chance you have of learning and remembering.

2 Do the exercises. Write your own notes.

We put them in, but if we did them for you, that would be like having someone else do your workouts for you. And don't just *look* at the exercises. **Use a pencil.** There's plenty of evidence that physical activity *while* learning can increase the learning.

3 Read the “There are No Dumb Questions.”

That means all of them. They're not optional sidebars, **they're part of the core content!** Don't skip them.

4 Make this the last thing you read before bed. Or at least the last challenging thing.

Part of the learning (especially the transfer to long-term memory) happens *after* you put the book down. Your brain needs time on its own, to do more processing. If you put in something new during that processing time, some of what you just learned will be lost.

5 Talk about it. Out loud.

Speaking activates a different part of the brain. If you're trying to understand something, or increase your chance of remembering it later, say it out loud. Better still, try to explain it out loud to someone else. You'll learn more quickly, and you might uncover ideas you hadn't known were there when you were reading about it.

6 Drink water. Lots of it.

Your brain works best in a nice bath of fluid. Dehydration (which can happen before you ever feel thirsty) decreases cognitive function.

7 Listen to your brain.

Pay attention to whether your brain is getting overloaded. If you find yourself starting to skim the surface or forget what you just read, it's time for a break. Once you go past a certain point, you won't learn faster by trying to shove more in, and you might even hurt the process.

8 Feel something.

Your brain needs to know that this *matters*. Get involved with the stories. Make up your own captions for the photos. Groaning over a bad joke is *still* better than feeling nothing at all.

9 Write a lot of code!

There's only one way to learn to program: **writing a lot of code.** And that's what you're going to do throughout this book. Coding is a skill, and the only way to get good at it is to practice. We're going to give you a lot of practice: every chapter has exercises that pose a problem for you to solve. Don't just skip over them—a lot of the learning happens when you solve the exercises. We included a solution to each exercise—don't be afraid to **peek at the solution** if you get stuck! (It's easy to get snagged on something small.) But try to solve the problem before you look at the solution. And definitely get it working before you move on to the next part of the book.

Read Me

This is a learning experience, not a reference book. We deliberately stripped out everything that might get in the way of learning whatever it is we're working on at that point in the book. And the first time through, you need to begin at the beginning, because the book makes assumptions about what you've already seen and learned.

This book is designed to get you up to speed with Python as quickly as possible.

As you need to know stuff, we teach it. So you won't find long lists of technical material, no tables of Python's operators, nor its operator precedence rules. We don't cover *everything*, but we've worked really hard to cover the essential material as well as we can, so that you can get Python into your brain *quickly* and have it stay there. The only assumption we make is that you already know how to program in some other programming language.

This book targets Python 3

We use Release 3 of the Python programming language in this book, and we cover how to get and install Python 3 in the first chapter. That said, we don't completely ignore Release 2, as you'll discover in Chapters 8 through 11. But trust us, by then you'll be so happy using Python, you won't notice that the technologies you're programming are running Python 2.

We put Python to work for you right away.

We get you doing useful stuff in Chapter 1 and build from there. There's no hanging around, because we want you to be *productive* with Python right away.

The activities are NOT optional.

The exercises and activities are not add-ons; they're part of the core content of the book. Some of them are to help with memory, some are for understanding, and some will help you apply what you've learned. ***Don't skip the exercises.***

The redundancy is intentional and important.

One distinct difference in a Head First book is that we want you to *really* get it. And we want you to finish the book remembering what you've learned. Most reference books don't have retention and recall as a goal, but this book is about *learning*, so you'll see some of the same concepts come up more than once.

The examples are as lean as possible.

Our readers tell us that it's frustrating to wade through 200 lines of an example looking for the two lines they need to understand. Most examples in this book are shown within the smallest possible context, so that the part you're trying to learn is clear and simple. Don't expect all of the examples to be robust, or even complete—they are written specifically for learning, and aren't always fully functional.

We've placed a lot of the code examples on the Web so you can copy and paste them as needed. You'll find them at two locations:

<http://www.headfirstlabs.com/books/hfpython/>

<http://python.itcarlow.ie>

The Brain Power exercises don't have answers.

For some of them, there is no right answer, and for others, part of the learning experience of the Brain Power activities is for you to decide if and when your answers are right. In some of the Brain Power exercises, you will find hints to point you in the right direction.

The technical review team

David Griffiths



Phil Hartley



Jeremy Jones



Technical Reviewers:

David Griffiths is the author of *Head First Rails* and the coauthor of *Head First Programming*. He began programming at age 12, when he saw a documentary on the work of Seymour Papert. At age 15, he wrote an implementation of Papert's computer language LOGO. After studying Pure Mathematics at University, he began writing code for computers and magazine articles for humans. He's worked as an agile coach, a developer, and a garage attendant, but not in that order. He can write code in over 10 languages and prose in just one, and when not writing, coding, or coaching, he spends much of his spare time traveling with his lovely wife—and fellow *Head First* author—Dawn.

Phil Hartley has a degree in Computer Science from Edinburgh, Scotland. Having spent more than 30 years in the IT industry with specific expertise in OOP, he is now teaching full time at the University of Advancing Technology in Tempe, AZ. In his spare time, Phil is a raving NFL fanatic.

Jeremy Jones is coauthor of *Python for Unix and Linux System Administration*. He has been actively using Python since 2001. He has been a developer, system administrator, quality assurance engineer, and tech support analyst. They all have their rewards and challenges, but his most challenging and rewarding job has been husband and father.

Acknowledgments

My editor:

Brian Sawyer was *Head First Python*'s editor. When not editing books, Brian likes to run marathons in his spare time. This turns out to be the perfect training for working on another book with me (our second together). O'Reilly and Head First are lucky to have someone of Brian's caliber working to make this and other books the best they can be.



The O'Reilly team:

Karen Shaner provided administrative support and very capably coordinated the technical review process, responding quickly to my many queries and requests for help. There's also the back-room gang to thank—the O'Reilly Production Team—who guided this book through its final stages and turned my InDesign files into the beautiful thing you're holding in your hands right now (or maybe you're on an iPad, Android tablet, or reading on your PC—that's cool, too).

And thanks to the other Head First authors who, via Twitter, offered cheers, suggestions, and encouragement throughout the entire writing process. You might not think 140 characters make a big difference, but they really do.

I am also grateful to **Bert Bates** who, together with **Kathy Sierra**, created this series of books with their wonderful *Head First Java*. At the start of this book, **Bert** took the time to set the tone with a marathon 90-minute phone call, which stretched my thinking on what I wanted to do to the limit and pushed me to write a better book. Now, some nine months after the phone call, I'm pretty sure I've recovered from the mind-bending **Bert** put me through.

Friends and colleagues:

My thanks again to **Nigel Whyte**, Head of Department, Computing and Networking at The Institute of Technology, Carlow, for supporting my involvement in yet another book (especially so soon after the last one).

My students (those enrolled on 3rd Year Games Development and 4th Year Software Engineering) have been exposed to this material in various forms over the last 18 months. Their positive reaction to Python and the approach I take with my classes helped inform the structure and eventual content of this book. (And yes, folks, some of this is on your final).

Family:

My family, **Deirdre, Joseph, Aaron**, and **Aideen** had to, once more, bear the grunts and groans, huffs and puffs, and more than a few roars on more than one occasion (although, to be honest, not as often they did with *Head First Programming*). After the last book, I promised I wouldn't start another one "for a while." It turned out "a while" was no more than a few weeks, and I'll be forever grateful that they didn't gang up and throw me out of the house for breaking my promise. Without their support, and especially the ongoing love and support of my wife, Deirdre, this book would not have seen the light of day.

The without-whom list:

My technical review team did an excellent job of keeping me straight and making sure what I covered was spot on. They confirmed when my material was working, challenged me when it wasn't and not only pointed out when stuff was wrong, but provided suggestions on how to fix it. This is especially true of **David Griffiths**, my co-conspirator on *Head First Programming*, whose technical review comments went above and beyond the call of duty. **David's** name might not be on the cover of this book, but a lot of his ideas and suggestions grace its pages, and I was thrilled and will forever remain grateful that he approached his role as tech reviewer on *Head First Python* with such gusto.

Safari® Books Online



Safari Books Online is an on-demand digital library that lets you easily search over 7,500 technology and creative reference books and videos to find the answers you need quickly.

With a subscription, you can read any page and watch any video from our library online. Read books on your cell phone and mobile devices. Access new titles before they are available for print, and get exclusive access to manuscripts in development and post feedback for the authors. Copy and paste code samples, organize your favorites, download chapters, bookmark key sections, create notes, print out pages, and benefit from tons of other time-saving features.

O'Reilly Media has uploaded this book to the Safari Books Online service. To have full digital access to this book and others on similar topics from O'Reilly and other publishers, sign up for free at <http://my.safaribooksonline.com/?portal=oreilly>.

1 meet python



Everyone loves lists



You're asking one question: “What makes Python different?”

The short answer is: *lots of things*. The longer answers starts by stating that there's lots that's familiar, too. Python is a lot like any other *general-purpose* programming language, with **statements**, **expressions**, **operators**, **functions**, **modules**, **methods**, and **classes**. All the *usual stuff*, really. And then there's the other stuff Python provides that makes the programmer's life—*your life*—that little bit easier. You'll start your tour of Python by learning about **lists**. But, before getting to that, there's another important question that needs answering...

What's to like about Python?

Lots. Rather than tell you, this book's goal is to *show* you the greatness that is Python.



Before diving head first into Python, let's get a bit of housekeeping out of the way.

To work with and execute the Python code in this book, you need a copy of the **Python 3 interpreter** on your computer. Like a lot of things to do with Python, it's not difficult to install the interpreter. Assuming, of course, it's not already there...

Install Python 3

Before you *write* and *run* Python code, you need to make sure the Python interpreter is *on* your computer. In this book, you'll start out with Release 3 of Python, the very latest (and best) version of the language.

A release of Python might already be on your computer. Mac OS X comes with Python 2 preinstalled, as do most versions of Linux (which can also ship with Release 3). Windows, in contrast, doesn't include any release of Python. Let's check your computer for Python 3. Open up a command-line prompt and, if you are using Mac OS X or Linux, type:

python3 -v

That's an UPPERCASE "v", by the way.

On Windows, use this command:

c:\Python31\python.exe -v

Using the
UPPERCASE "v"
results in the Python
version appearing on
screen.

Without the
UPPERCASE "v",
you are taken
into the Python
interpreter.

Use the `quit()`
command to exit
the interpreter and
return to your OS
prompt.



If Python 3 is missing from
your computer, download
a copy for your favorite OS
from the www.python.org
website.

```
File Edit Window Help WhichPython?
$ python3 -v
Python 3.1.2
$
$ python3
Python 3.1.2 (r312:79360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "help", "copyright", "credits" or "license" for more info.
>>>
>>> quit()
$
```

When you install Python 3, you also get **IDLE**, Python's simple—yet surprisingly *useful*—integrated development environment. IDLE includes a color syntax-highlighting editor, a debugger, the Python Shell, and a complete copy of Python 3's online documentation set.

Let's take a quick look at IDLE.

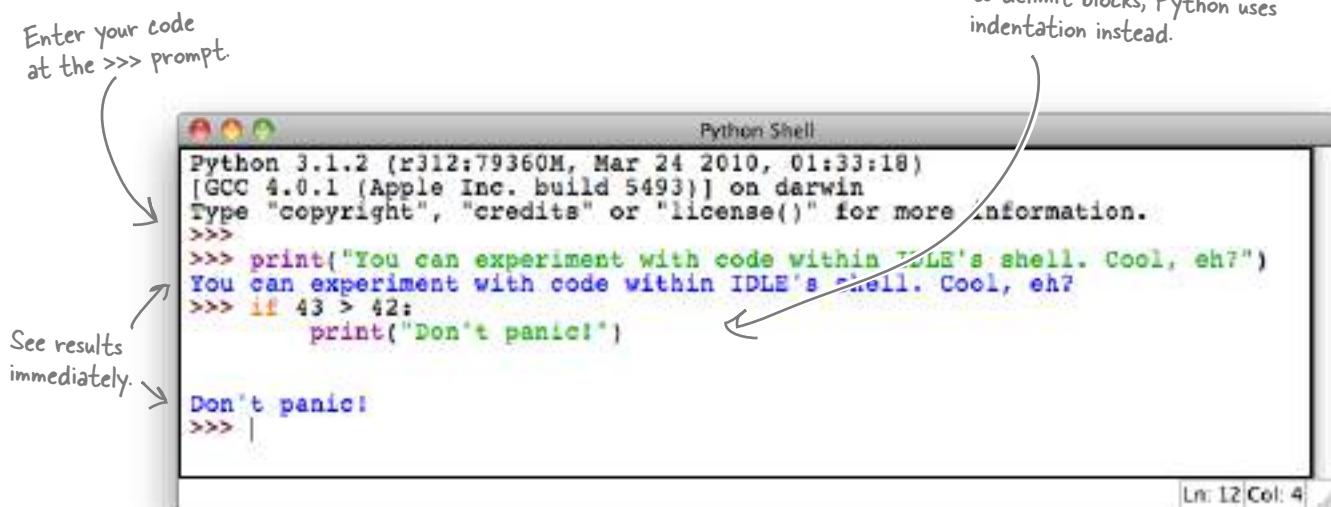
Use IDLE to help learn Python

IDLE lets you *write* code in its full-featured code editor as well as *experiment* with code at the **Python Shell**. You'll use the code editor later in this book but, when learning Python, IDLE's shell really rocks, because it lets you try out new Python code *as you go*.

When you first start IDLE, you are presented with the “triple chevron” prompt (`>>>`) at which you enter code. The shell takes your code statement and *immediately* executes it for you, displaying any results produced on screen.

IDLE knows all about Python syntax and offers “completion hints” that pop up when you use a **built-in function** like `print()`. Python programmers generally refer to built-in functions as **BIFs**. The `print()` BIF displays messages to *standard output* (usually the screen).

Unlike other C-based languages, which use { and } to delimit blocks, Python uses indentation instead.



IDLE uses colored syntax to highlight your code. By default, built-in functions are purple, strings are green, and language keywords (like `if`) are orange. Any results produced are in blue. If you hate these color choices, don't worry; you can easily change them by adjusting IDLE's preferences.

IDLE also knows all about Python's indentation syntax, which requires code blocks be **indented**. When you start with Python, this can be hard to get used to, but IDLE keeps you straight by automatically indenting as needed.

**IDLE knows
Python's syntax
and helps you
conform to
the Python
indentation rules.**

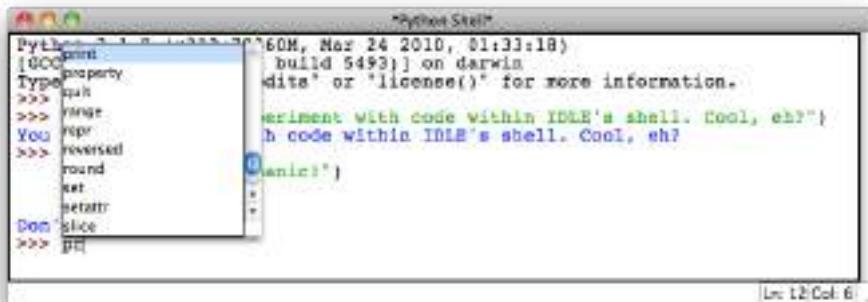
Work effectively with IDLE

IDLE has lots of features, but you need to know about only a few of them to get going.

TAB completion

Start to type in some code, and then press the **TAB** key. IDLE will offer suggestions to help you complete your statement.

Type "pr" and then TAB at the >>> prompt to see IDLE's list of command completion suggestions.



Recall code statements

Press **Alt-P** to recall the *previous* code statement entered into IDLE or press **Alt-N** to move to the *next* code statement (assuming there is one). Both key combinations can be used to cycle rapidly through all of the code you've entered into IDLE, re-executing any code statements as needed.

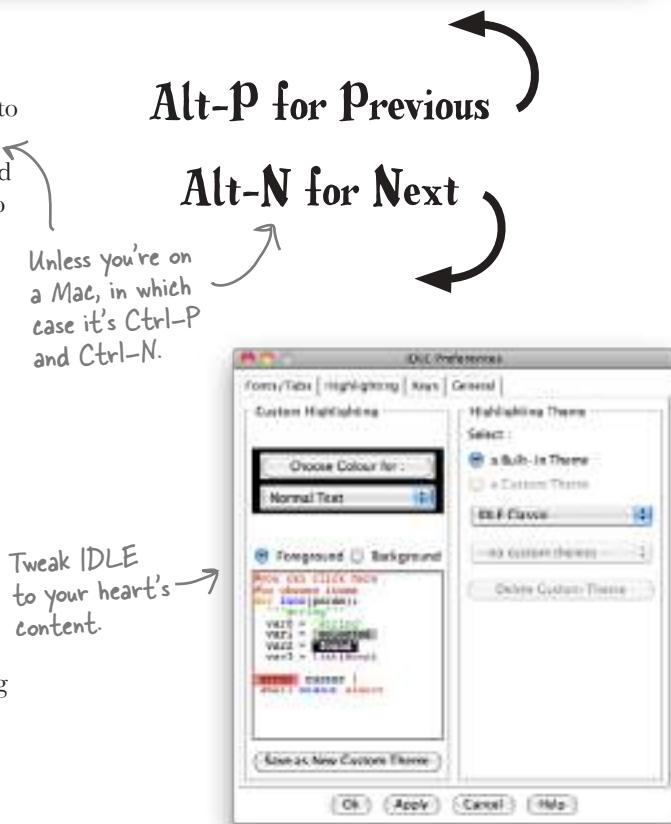
Edit recalled code

Once you recall your code statement, you can *edit* it and move around the statement using the arrow keys. It's possible to edit any statement that you've previously entered, even code statements that span multiple lines.

Adjust IDLE's preferences

IDLE's preferences dialog lets you adjust its default behavior to your tastes. There are four tabs of settings to tweak. You can control font and tab behavior, the colors used to syntax highlight, the behavior of certain key-combinations, and IDLE's start-up settings. So, if shocking pink strings is really your thing, IDLE gives you the power to change how your code looks on screen.

This is how IDLE looks on my computer. It might look a little different on yours, but not by much. (And, yes, it's meant to look this ugly.)



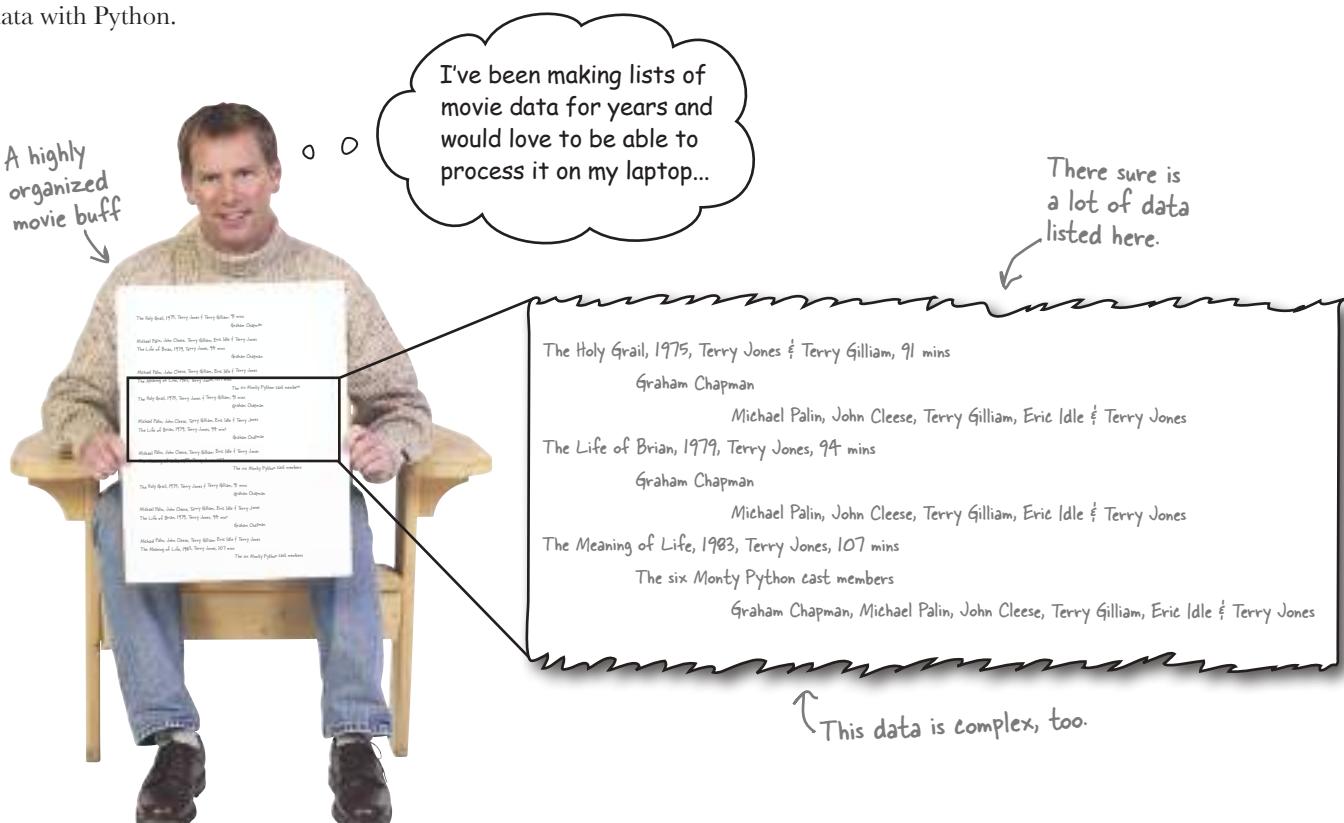
Tweak IDLE to your heart's content.

Deal with complex data

Any program of any worth that you create has to work with **data**. Sometimes, the data is simple and straightforward—*easy to work with*. Other times, the data you have to work with is *complex* in its structure and meaning, forcing you to work hard to make sense of it all, let alone write code to process it.

To tame complexity, you can often arrange your data as a **list**: there's the list of customers, your friend's list, the shopping list, and your to-do list (to name a few). Arranging data in lists is so common that Python makes it easy for you to create and process lists in code.

Let's look at some complex data before learning how to create and process list data with Python.



On first glance, this collection of data does indeed look quite complex. However, the data appears to conform to some sort of structure: there's a line for a list of basic movie facts, then another line for the lead actor(s), followed by a third line listing the movie's supporting actors.

This looks like a structure you can work with...

Create simple Python lists

Let's start with the following simple list of movie titles and work up from there:

The Holy Grail

The Life of Brian

The Meaning of Life

A short list of some
Monty Python movies

Here's the same list written in a way that Python understands:

```
movies = ["The Holy Grail",
          "The Life of Brian",
          "The Meaning of Life"]
```

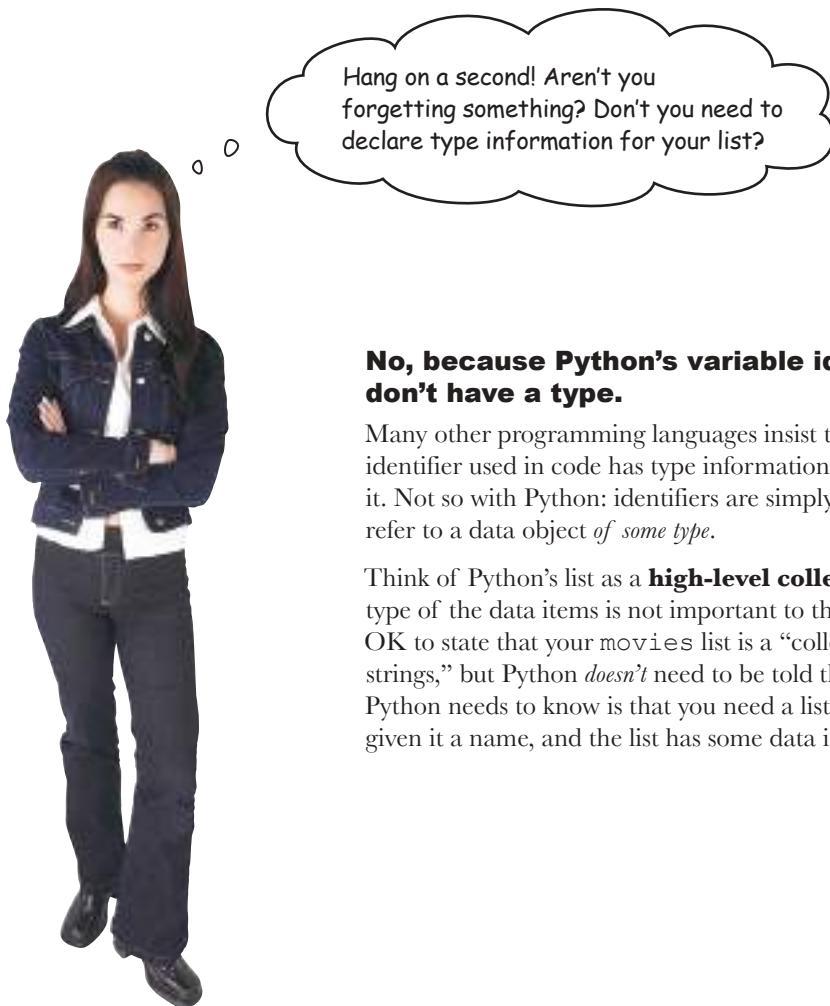
To turn the human-friendly list into a Python-friendly one, follow this four-step process:

- 1** **Convert** each of the names into strings by surrounding the data with quotes.
- 2** **Separate** each of the list items from the next with a comma.
- 3** **Surround** the list of items with opening and closing square brackets.
- 4** **Assign** the list to an identifier (`movies` in the preceding code) using the assignment operator (`=`).

It's perfectly OK to put your list creation code *all on one line*, assuming, of course, that you have room:

```
movies = ["The Holy Grail", "The Life of Brian", "The Meaning of Life"]
```

This works, too.



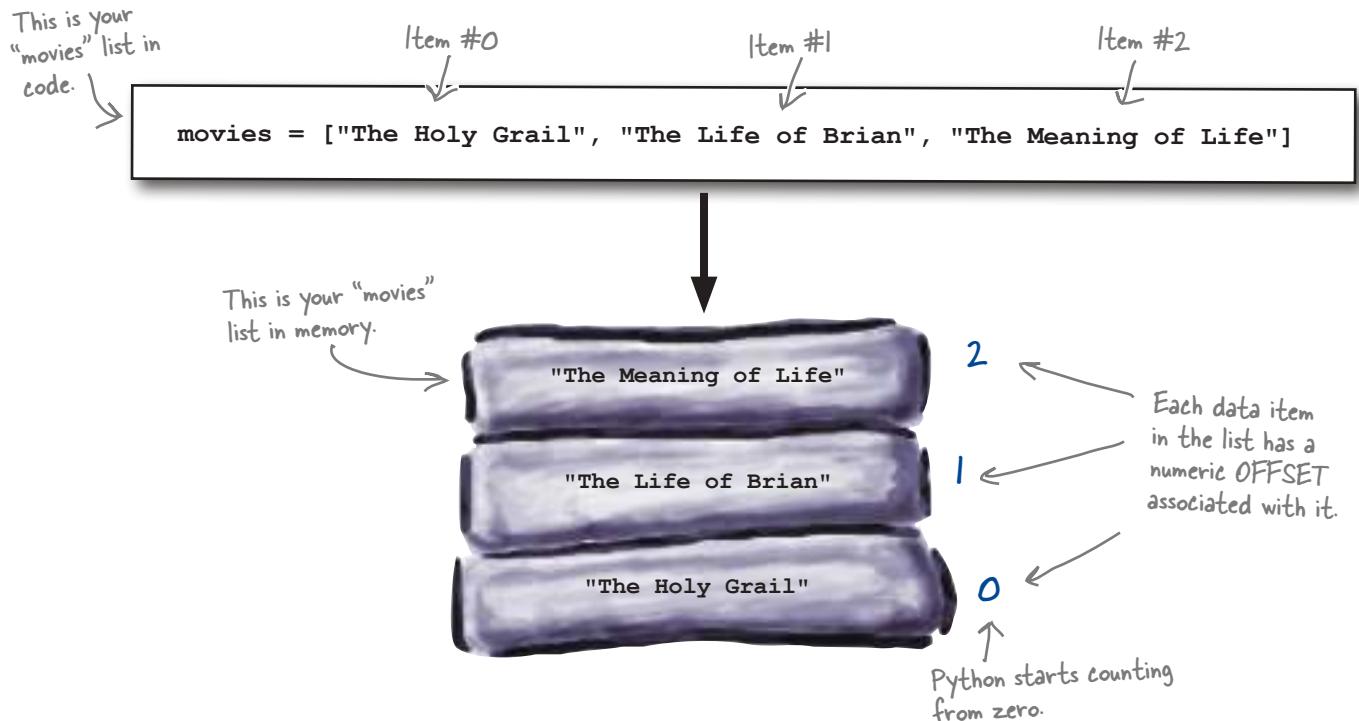
No, because Python's variable identifiers don't have a type.

Many other programming languages insist that every identifier used in code has type information declared for it. Not so with Python: identifiers are simply names that refer to a data object of *some type*.

Think of Python's list as a **high-level collection**. The type of the data items is not important to the list. It's OK to state that your `movies` list is a “collection of strings,” but Python *doesn't* need to be told this. All Python needs to know is that you need a list, you've given it a name, and the list has some data items in it.

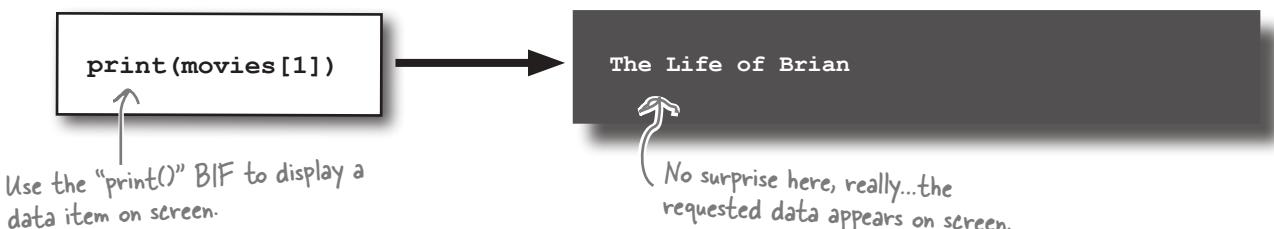
Lists are like arrays

When you create a list in Python, the interpreter creates an array-like data structure in memory to hold your data, with your data items stacked from the bottom up. Like array technology in other programming languages, the first slot in the stack is numbered 0, the second is numbered 1, the third is numbered 2, and so on:



Access list data using the square bracket notation

As with arrays, you can access the data item in a list slot using the *standard square bracket offset notation*:



Let's use IDLE to learn a bit about how lists work.



An IDLE Session

Lists in Python might look like arrays, but they are much more than that: they are full-blown Python *collection objects*. This means that lists come with ready-to-use functionality in the form of **list methods**.

Let's get to know some of Python's list methods. Open up IDLE and follow along with the code entered at the `>>>` prompt. You should see exactly the same output as shown here.

Start by defining a list of names, which you then display on screen using the `print()` BIF. Then, use the `len()` BIF to work out how many data items are in the list, before accessing and displaying the value of the second data item:

```
>>> cast = ["Cleese", 'Palin', 'Jones', "Idle"]
>>> print(cast)
['Cleese', 'Palin', 'Jones', 'Idle']
>>> print(len(cast))
4
It's OK to invoke a BIF on
the results of another BIF.
>>> print(cast[1])
Palin
```

With your list created, you can use list methods to add a single data item to the end of your list (using the `append()` method), remove data from the end of your list (with the `pop()` method), and add a collection of data items to the end of your list (thanks to the `extend()` method):

```
>>> cast.append("Gilliam")
>>> print(cast)
['Cleese', 'Palin', 'Jones', 'Idle', 'Gilliam']
Methods are invoked using the
common "." dot notation.

>>> cast.pop()
'Gilliam'
>>> print(cast)
['Cleese', 'Palin', 'Jones', 'Idle']
It's another list: items separated by commas,
surrounded by square brackets.
>>> cast.extend(["Gilliam", "Chapman"])
>>> print(cast)
['Cleese', 'Palin', 'Jones', 'Idle', 'Gilliam', 'Chapman']
```

Finally, find and remove a *specific* data item from your list (with the `remove()` method) and then add a data item before a *specific* slot location (using the `insert()` method):

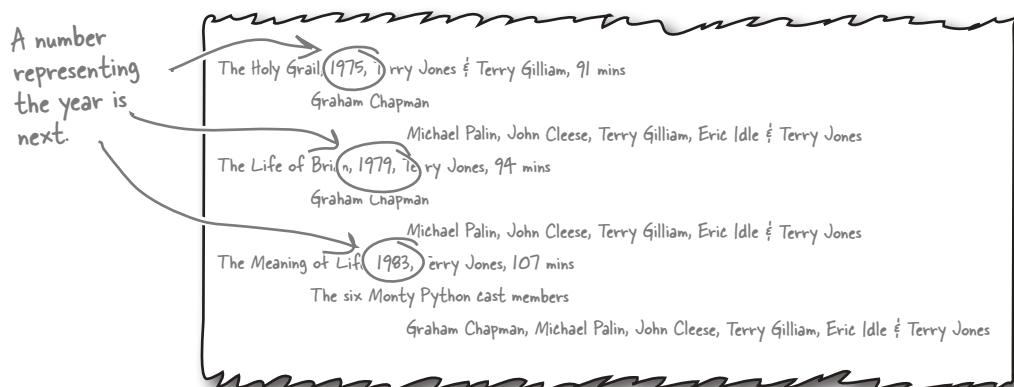
```
>>> cast.remove("Chapman")
>>> print(cast)
['Cleese', 'Palin', 'Jones', 'Idle', 'Gilliam']
>>> cast.insert(0, "Chapman")
>>> print(cast)
['Chapman', 'Cleese', 'Palin', 'Jones', 'Idle', 'Gilliam']
After all that, we end up with
the cast of Monty Python's
Flying Circus!
```

Add more data to your list

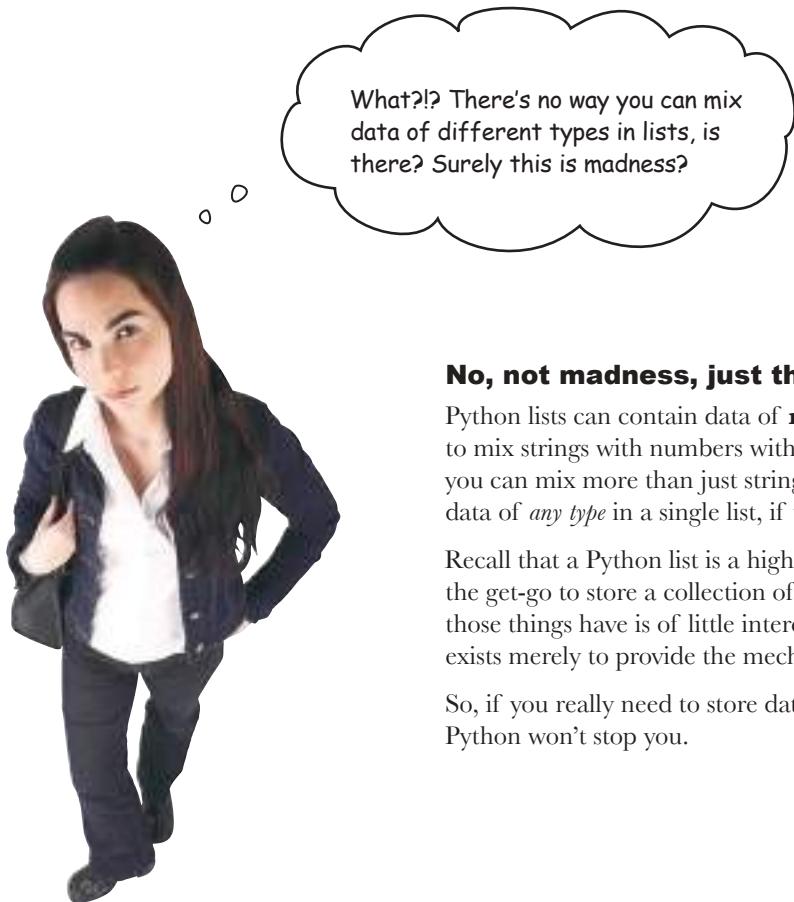
With your list of movie names created, now you need to add more of the movie buff's complex data to it. You have a choice here:



Either strategy works. Which works best for *you* depends on what you are trying to do. Let's recall what the movie buff's data looks like:



The next piece of data you need to add to your list is a **number** (which represents the year the movie was released), and it must be inserted *after* each movie name. Let's do that and see what happens.



No, not madness, just the way Python works.

Python lists can contain data of **mixed type**. It's perfectly OK to mix strings with numbers within the *same* Python list. In fact, you can mix more than just strings and numbers; you can store data of *any type* in a single list, if you like.

Recall that a Python list is a high-level collection, designed from the get-go to store a collection of "related things." What *type* those things have is of little interest to the list, because the list exists merely to provide the mechanism to store data in list form.

So, if you really need to store data of mixed type in a list, Python won't stop you.



Exercise

Let's take a bit of time to try to work out which strategy to use when adding data to your list *in this case*.

Given the following list-creation code:

```
movies = ["The Holy Grail", "The Life of Brian", "The Meaning of Life"]
```

- 1 Work out the Python code required to insert the numeric year data into the preceding list, changing the list so that it ends up looking like this:

```
["The Holy Grail", 1975, "The Life of Brian", 1979, "The Meaning of Life", 1983]
```

Write your
insertion →
code here.

.....
.....
.....
.....
.....
.....
.....
.....

- 2 Now write the Python code required to re-create the list with the data you need all in one go:

Write your
re-creation →
code here.

.....
.....
.....
.....

In this case, which of these two methods do you think is best? (Circle your choice).

1 or 2



Exercise Solution

Let's take a bit of time to try and work out which strategy to use when adding data to your list in this case.

Given the following list-creation code:

```
movies = ["The Holy Grail", "The Life of Brian", "The Meaning of Life"]
```

- 1 You were to work out the Python code required to insert the numeric year data into the preceding list:

Insert the first year
BEFORE the second list
item.

movies.insert(1, 1975)

Insert the second year
BEFORE the fourth list item.

movies.insert(3, 1979)

Did you get the math right?
After the first insertion, the
list grows, so you have to take
that into consideration when
working out where to do the
second insert.

movies.append(1983)

Then append the last year to
the end of the list.

- 2 You were also to write the Python code required to recreate the list with the data you need all in one go:

movies = ["The Holy Grail", 1975,

"The Life of Brian", 1979,

"The Meaning of Life", 1983]

Assign all your data to the "movies"
identifier. What was previously there is
replaced.

In this case, which of these two methods do you think is best? (You were to circle your choice.)

1

or

2

Yes, method 2 seems the better
option here...that is, for a small
list like this. Also, there's no
tricky counting to do.

Work with your list data

You often need to *iterate* over your list and perform some action on each item as you go along. Of course, it is always possible to do something like this, which works but *does not scale*:

Define a list and populate its items with the names of two movies.

Display the value of each individual list item on the screen.

```
fav_movies = ["The Holy Grail", "The Life of Brian"]

print(fav_movies[0])
print(fav_movies[1])
```

This is the list-processing code.

This code works as expected, making the data from the list appear on screen. However, if the code is later amended to add *another* favorite movie to the list, the list-processing code stops working as expected, because the *list-processing code does not mention the third item*.

Big deal: all you need to do is add another `print()` statement, right?

Yes, adding one extra `print()` statement works for one extra movie, but what if you need to add another hundred favorite movies? The *scale* of the problem defeats you, because adding all those extra `print()` statements becomes such a chore that you would rather find an excuse not to have to do.

It's time to iterate

Processing every list item is such a common requirement that Python makes it especially convenient, with the built-in `for` loop. Consider this code, which is a rewrite of the previous code to use a `for` loop:

Define a list and populate it just as you did before.

Use "for" to iterate over the list, displaying the value of each individual item on screen as you go.

```
fav_movies = ["The Holy Grail", "The Life of Brian"]

for each_flick in fav_movies:
    print(each_flick)
```

This is the list-processing code, using a `for` loop.

Using a `for` loop scales and works with any size list.

For loops work with lists of any size

Python's **for** loop exists to process lists and other *iterations* in Python. Lists are the most common iterated data structure in Python, and when you need to iterate a list, it's best to use **for**:

The keyword "for" indicates the start of the loop and comes before the target identifier.

→ **for**

target identifier

The keyword "in" separates the target identifier from your list.

in

list

A colon ":" follows your list name and indicates the start of your list-processing code.

: →

list-processing code

The list-processing code MUST be indented under the for loop.

The *list-processing code* is referred to by Python programmers as the **suite**.

The **target identifier** is like any other name in your code. As your list is iterated over, the target identifier is assigned *each* of the data values in your list, in turn. This means that each time the loop code executes, the target identifier refers to a *different* data value. The loop keeps iterating until it exhausts all of your list's data, no matter how big or small your list is.

An alternative to using **for** is to code the iteration with a **while** loop.

Consider these two snippets of Python code, which perform the *same action*:

When you use "while", you have to worry about "state information," which requires you to employ a counting identifier.

```
count = 0
while count < len(movies):
    print(movies[count])
    count = count+1
```

```
for each_item in movies:
    print(each_item)
```

When you use "for", the Python interpreter worries about the "state information" for you.

These while and for statements do the same thing.

there are no
Dumb Questions

Q: So...when iterating over a list, I should always use `for` instead of `while`?

A: Yes, unless you have a really good reason to use (or need the extra control of) a `while` loop. The `for` loop takes care of working from the start of your list and continuing to the end. It's next to impossible to get stung by an off-by-one error when you use `for`. This is not the case with `while`.

Q: So, lists aren't really like arrays then, because they do so much more?

A: Well...they are in that you can access individual data items in your list with the standard square bracket notation, but—as you've seen—Python's lists can do so much more. At Head First Labs, we like to think of lists as "arrays on steroids."

Q: And they work this way only in Python 3, right?

A: No. There are certain enhancements to lists that were added in Python 3, but release 2 of Python has lists, too. All of what you've learned about lists so far will work with lists in Releases 2 and 3 of Python.

Q: Why are we using Python 3? What's wrong with Python 2, anyway? Lots of programmers seem to be using it.

A: Lots of programmers are using Python 2, but the future of Python development lies with Release 3. Of course, moving the entire Python community to Python 3 won't happen overnight, so there's an awful lot of projects that will continue to run on Release 2 for the foreseeable future. Despite 2's dominance at the moment, at Head First Labs we think the new bits in 3 are well worth the added investment in learning about them now. Don't worry: if you know 2, Python 3 is easy.

Q: Seeing as Python's lists shrink and grow as needed, they must not support bounds-checking, right?

A: Well, lists are *dynamic*, in that they shrink and grow, but they are *not magic*, in that they cannot access a data item that does not exist. If you try to access a nonexistent data item, Python responds with an `IndexError`, which means "out of bounds."

Q: What's with all the strange references to Monty Python?

A: Ah, you spotted that, eh? It turns out that the creator of Python, *Guido van Rossum*, was reading the scripts of the Monty Python TV shows while designing his new programming language. When Guido needed a name for his new language, he chose "Python" as a bit of a joke (or so the legend goes).

Q: Do I need to know Monty Python in order to understand the examples?

A: No, but as they say in the official Python documentation: "it helps if you do." But don't worry: you'll survive, even if you've never heard of *Monty Python*.

Q: I notice that some of your strings are surrounded with double quotes and others with single quotes. What's the difference?

A: There isn't any. Python lets you use either to create a string. The only rule is that if you start a string with one of the quotes, then you have to end it with the same quote; you can't mix'n'match. As you may have seen, IDLE uses single quotes when displaying strings within the shell.

Q: What if I need to embed a double quote in a string?

A: You have two choices: either escape the double quote like this: \ ", or surround your string with single quotes.

Q: Can I use any characters to name my identifiers?

A: No. Like most other programming languages, Python has some rules that must be adhered to when creating names. Names can start with a letter character or an underscore, then include any number of letter characters, numbers, and/or underscores in the rest of the name. Strange characters (such as %\$£) are not allowed and you'll obviously want to use names that have meaning *within the context of your code*. Names like `members`, `the_time`, and `people` are much better than `m`, `t`, and `p`, aren't they?

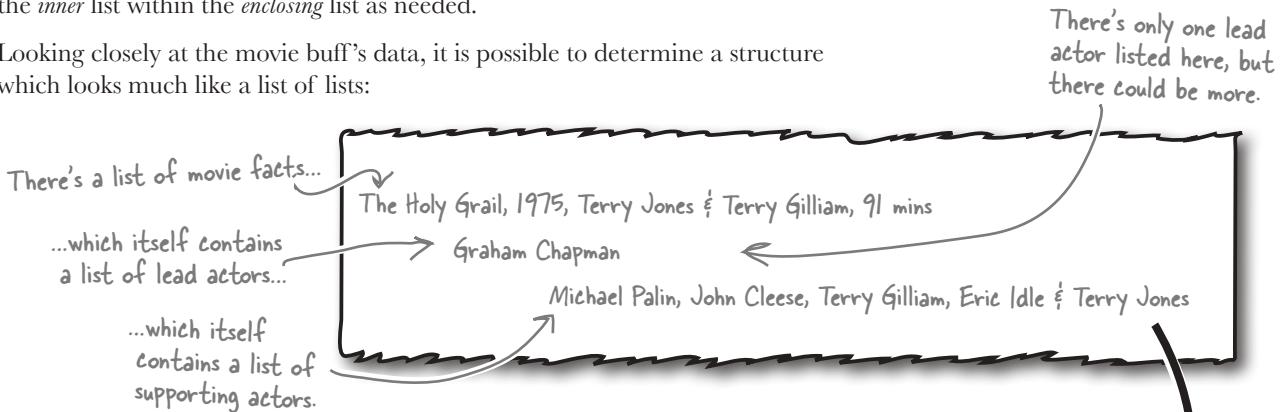
Q: Yes, good naming practice is always important. But what about case sensitivity?

A: Yes, Python is the "sensitive type," in that Python code *is* case sensitive. This means that `msg` and `MSG` are two different names, so be careful. Python (and IDLE) will help with the problems that can occur as a result of this. For instance, you can use an identifier in your code only if it has been given a value; unassigned identifiers cause a runtime error. This means that if you type `mgs` when you meant `msg`, you'll find out pretty quickly when Python complains about your code having a `NameError`.

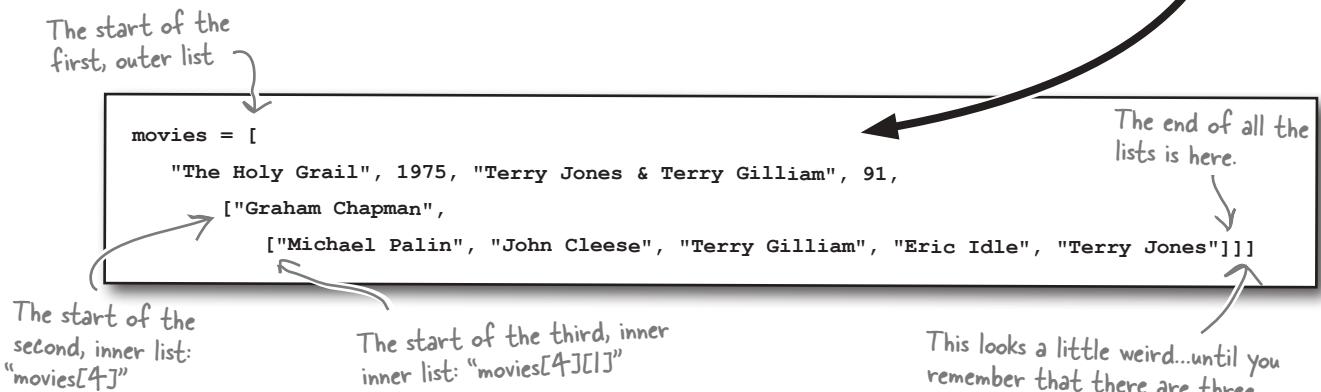
Store lists within lists

As you've seen, lists can hold data of mixed type. But it gets even better than that: lists can hold collections of anything, *including other lists*. Simply **embed** the *inner* list within the *enclosing* list as needed.

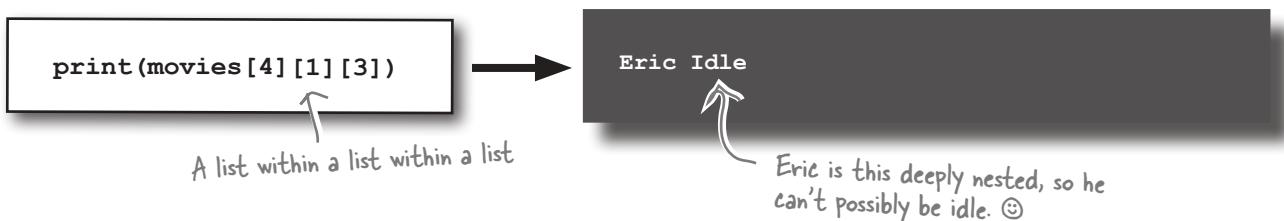
Looking closely at the movie buff's data, it is possible to determine a structure which looks much like a list of lists:



In Python, you can turn this real list of data into code with little or no effort. All you need to remember is that every list is a collection of items separated from each other with commas and surrounded with square brackets. And, of course, any list item can itself be another list:



So, a list within a list is possible, as is a list within a list within a list (as this example code demonstrates). In fact, it's possible to nest lists within lists to most any level with Python. And you can *manipulate* every list with its own list methods and *access* it with the square bracket notation:





An IDLE Session

Creating a list that contains another list is straightforward. But what happens when you try to process a list that contains another list (or lists) using the **for** loop from earlier in this chapter?

Let's use IDLE to work out what happens. Begin by creating the list of the movie data for "The Holy Grail" in memory, display it on screen, and then process the list with your **for** loop:

```
>>> movies = ["The Holy Grail", 1975, "Terry Jones & Terry Gilliam", 91,
              ["Graham Chapman", ["Michael Palin", "John Cleese",
              "Terry Gilliam", "Eric Idle", "Terry Jones"]]]
>>> print(movies)
['The Holy Grail', 1975, 'Terry Jones & Terry Gilliam', 91, ['Graham Chapman', ['Michael Palin',
'John Cleese', 'Terry Gilliam', 'Eric Idle', 'Terry Jones']]]
>>> for each_item in movies:
    print(each_item)
```

The list within a list within a list has been created in memory.

The "for" loop prints each item of the outer loop ONLY.

The inner list within the inner list is printed "as-is."

The Holy Grail
1975
Terry Jones & Terry Gilliam
91
['Graham Chapman', ['Michael Palin', 'John Cleese', 'Terry Gilliam', 'Eric Idle', 'Terry Jones']]



Your for loop is working OK. I think the trouble is that you haven't told it what to do with any inner lists that it finds, so it just prints everything, right?

Yes, that's correct: the loop code isn't complete.

At the moment, the code within the loop simply prints each list item, and when it finds a list at a slot, it simply displays the entire list on screen. After all, the inner list is just another list item as far as the outer enclosing list is concerned. What's we need here is some mechanism to spot that an item in a list is in fact *another* list and take the appropriate action.

That sounds a little tricky. But can Python help?

Check a list for a list

Each time you process an item in your list, you need to check to see if the item is another list. If the item is a list, you need to process the **nested list** before processing the next item in your outer list. Deciding *what to do when* in Python follows the familiar **if... else...** pattern:

The keyword "if"
indicates the start
of the decision code.

if

some condition holds

A colon (:) follows your
condition test.

:

This code executes if the
condition holds (i.e., it's TRUE).

the "true" suite

else: ← Look! Another colon.

This code executes if the condition
does NOT hold (i.e., it's FALSE).

the "false" suite

Note: both suites
are indented.

No surprises here, as the **if** statement in Python works pretty much as expected. But what *condition* do you need to check? You need a way to determine if the item currently being processed is a list. Luckily, Python ships with a BIF that can help here: `isinstance()`.

What's cool about the `isinstance()` BIF is that it lets you check if a *specific* identifier holds data of a *specific* type:

Create a short list and
assign it to an identifier.

Ask if "names" is a list (it is).

Assign a number to an
identifier.

Ask if "num_names" is a
list (it isn't).

An IDLE Session

Let's use the IDLE shell to learn a little about how `isinstance()` works:

```
>>> names = ['Michael', 'Terry']
>>> isinstance(names, list)
True
>>> num_names = len(names)
>>> isinstance(num_names, list)
False
```

Refer to a Python type
here. In this case, the type
is "list".



Here's a copy of the current list-processing code. Your task is to rewrite this code using an `if` statement and the `isinstance()` BIF to process a list that displays another list.

```
for each_item in movies:  
    print(each_item)
```

Write your
new code →
here.

.....

.....

.....

.....

.....

.....

.....

.....

there are no Dumb Questions

Q: Are there many of these BIFs in Python?

A: Yes. At the last count, there were over 70 BIEs in Python 3.

Q: Over 70! How am I to remember that many, let alone find out what they all are?

A: You don't have to worry about remembering. Let Python do it for you.

Q: How?

A: At the Python or IDLE shell, type `dir(__builtins__)` to see a list of the built-in stuff that comes with Python (that's two leading and trailing underscore characters, by the way). The shell spits out a big list. Try it. All those lowercase words are BIFs. To find out what any BIF does—like `input()`, for example—type `help(input)` at the shell for a description of the BIFs function.

Q: Why so many BIFs?

A: Why not? Because Python comes with lots of built-in functionality, it can mean less code for you to write. This Python philosophy is known as “batteries included”: there’s enough included with Python to let you do most things well, without having to rely on code from third parties to get going. As well as lots of BIFs, you’ll find that Python’s standard library is rich and packed with features waiting to be exploited by you.



Exercise Solution

Here's a copy of the current list-processing code. Your task was to rewrite this code using an *if* statement and the `isinstance()` BIF to process a list that displays another list.

```
for each_item in movies:  
    print(each_item)
```

Process the "movies" list as before.

You need to check if the current item is a list.

If it is a list, use another "for" loop to process the nested list.

```
for each_item in movies:
```

```
    if isinstance(each_item, list):
```

```
        for nested_item in each_item:
```

```
            print(nested_item)
```

```
    else:
```

Did you manage to get your indentation right?

```
        print(each_item)
```

The inner loop needs a new target identifier.

If the current item of the enclosing list isn't a list, display it on screen.



An IDLE Session

Let's use IDLE to see if this code makes a difference to the output displayed on screen:

```
>>> for each_item in movies:  
    if isinstance(each_item, list):  
        for nested_item in each_item:  
            print(nested_item)  
    else:  
        print(each_item)
```

The Holy Grail

1975

Terry Jones & Terry Gilliam

91

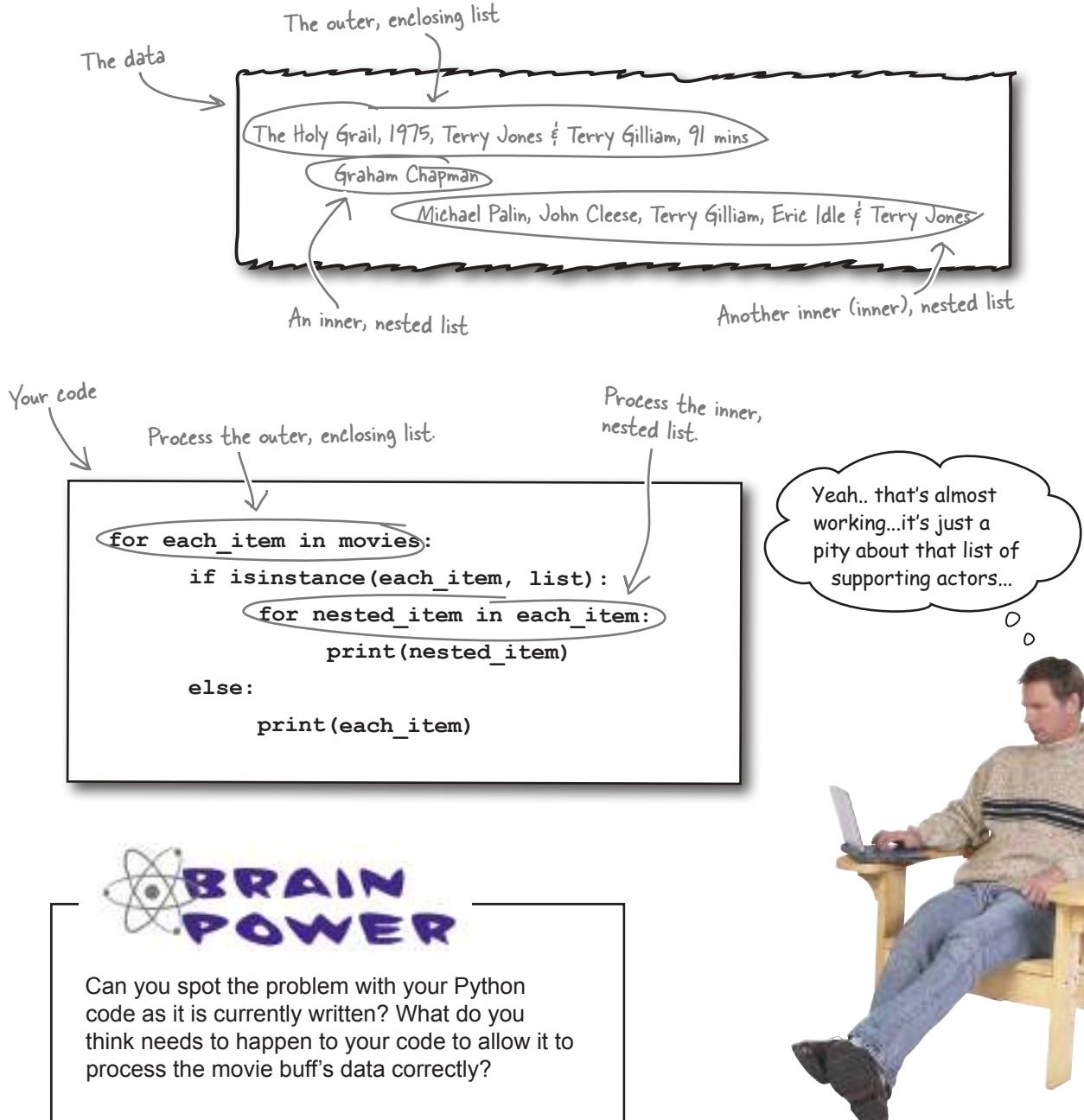
Graham Chapman

['Michael Palin', 'John Cleese', 'Terry Gilliam', 'Eric Idle', 'Terry Jones']

This is a little better, but not by much...there's another nested list here that's not being processed properly.

Complex data is hard to process

The movie buff's data is *complex*. Let's take another look at a subset of the data and your Python code that processes it.

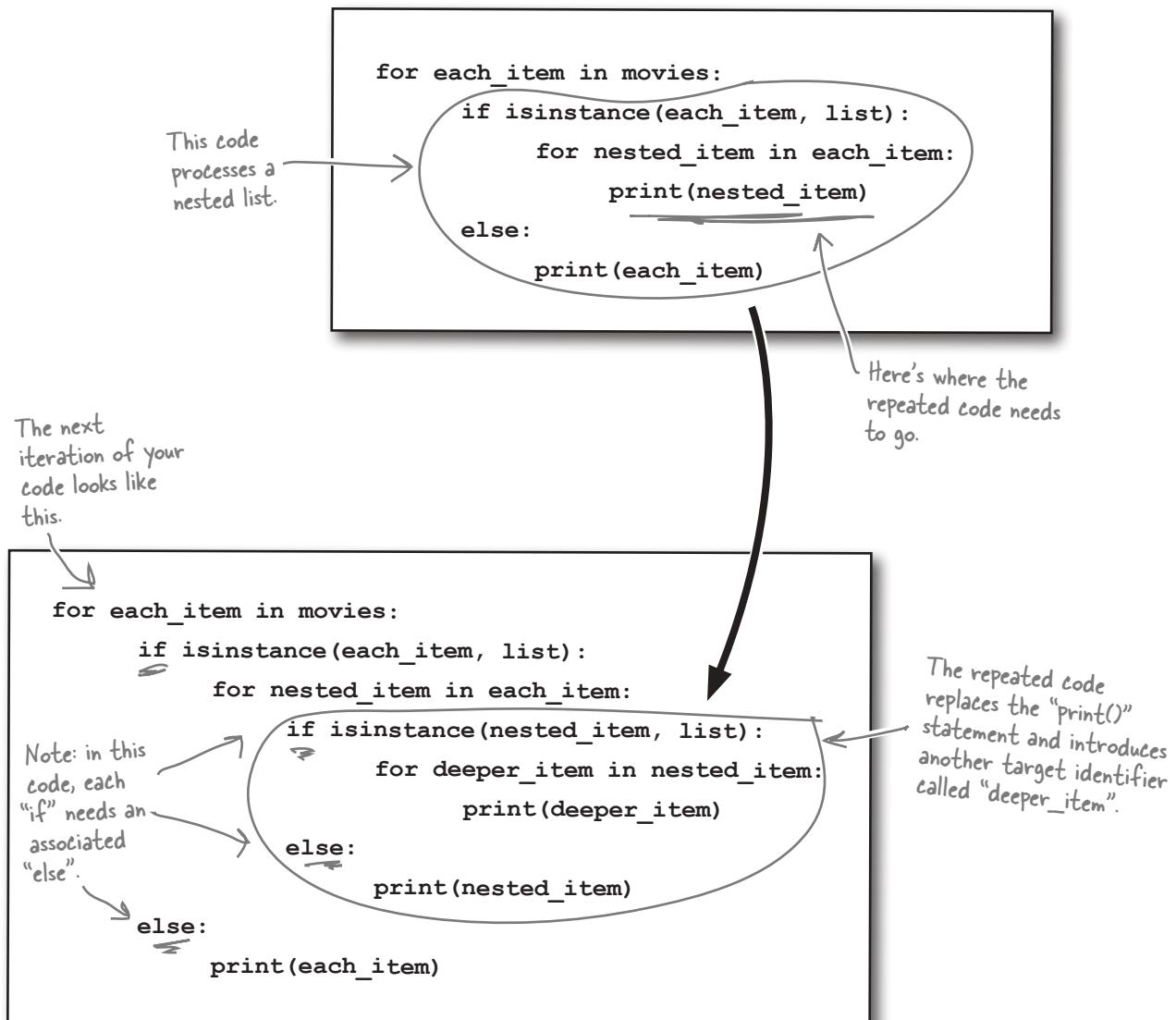


Handle many levels of nested lists

The data and your code are not in sync.

The movie buff's data is a list that contains a nested list that itself contains a nested list. The trouble is that your code knows only how to process a list nested inside an enclosing list.

The solution, of course, is to *add more code to handle the additionally nested list*. By looking at the existing code, it's easy to spot the code you need to repeat:





An IDLE Session

Let's use IDLE once more to test this latest iteration of your code:

```
>>> for each_item in movies:
    if isinstance(each_item, list):
        for nested_item in each_item:
            if isinstance(nested_item, list):
                for deeper_item in nested_item:
                    print(deeper_item)
            else:
                print(nested_item)
    else:
        print(each_item)
```

The Holy Grail

1975

Terry Jones & Terry Gilliam

91

Graham Chapman

Michael Palin

John Cleese

Terry Gilliam

Eric Idle

Terry Jones

Process a deeply nested
list inside a nested list
inside an enclosing list.

It works! This time, you
see all of your list data
on screen.

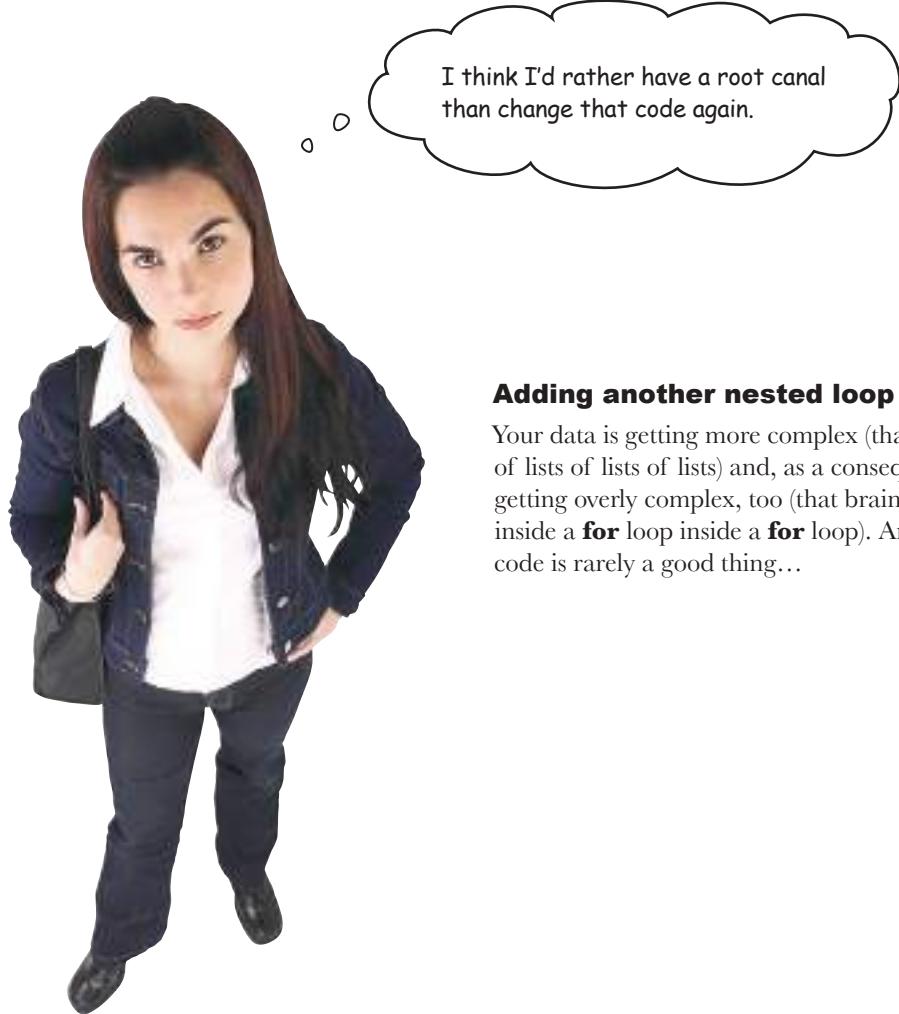


I just love that...in fact, I love it so much I've decided to add another list to my data. I want to include the other movies each supporting actor has starred in. If I add the data, can you change your code to print this data, too?

That's more list data and more Python code.

The data has to be embedded as *another* nested list within the already deeply nested list of supporting actors. That's possible to do, even though it makes your head hurt just to think about a list of lists of lists of lists! Amending your code is just a matter of adding another **for** loop and an **if** statement.

That doesn't sound like too much trouble, does it?



Adding another nested loop is a huge pain.

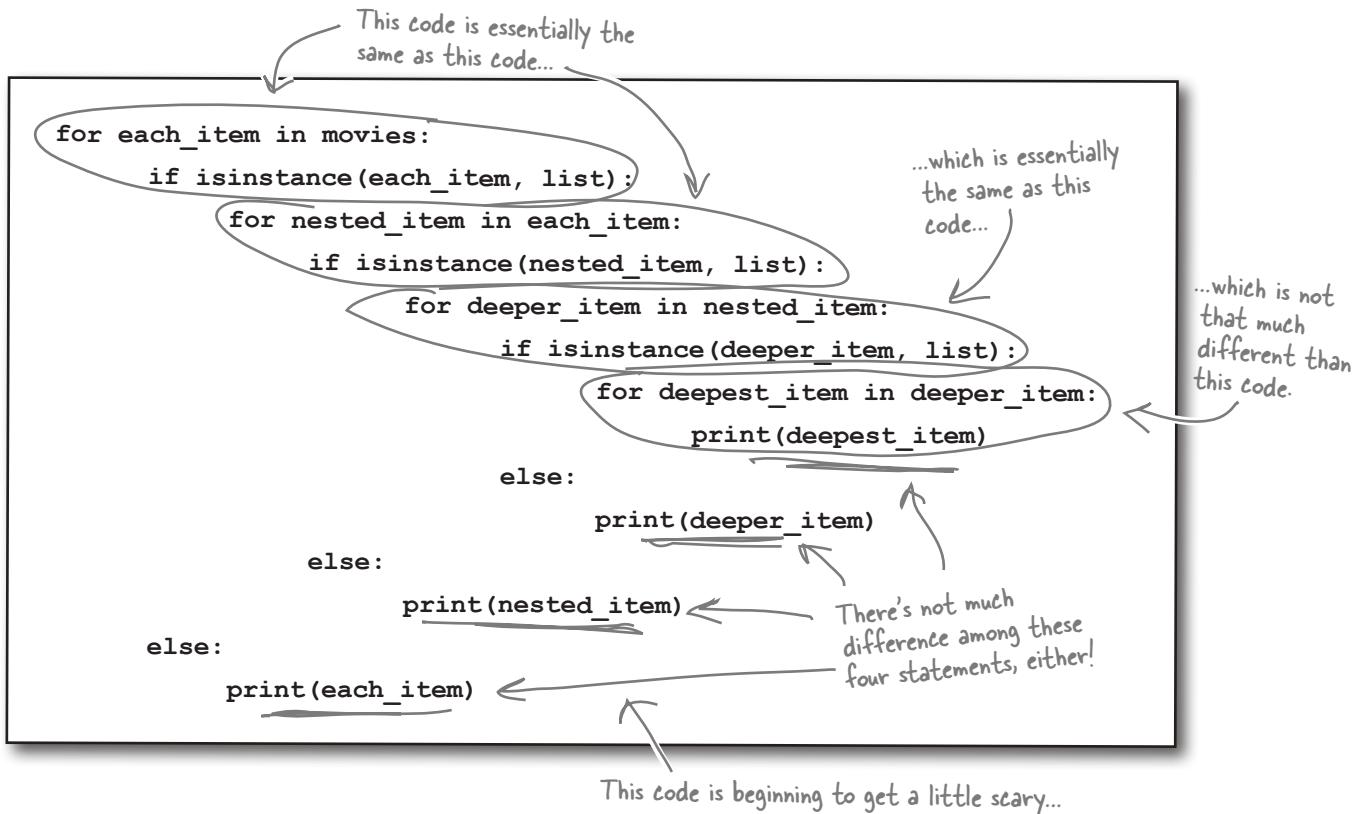
Your data is getting more complex (that mind-bending list of lists of lists of lists) and, as a consequence, your code is getting overly complex, too (that brain-exploding **for** loop inside a **for** loop inside a **for** loop). And overly complex code is rarely a good thing...



Wouldn't it be dreamy if there were an efficient way to process lists, preferably using a technique that resulted in less code, not more? But I know it's just a fantasy...

Don't repeat code; create a function

Take a look at the code that you've created so far, which (in an effort to save *you* from having *your* brain explode) has already been amended to process *yet another* nested list. Notice anything?



Your code now contains a lot of *repeated code*. It's also a mess to look at, even though it works with the movie buff's amended data. All that nesting of **for** loops is hard to read, and it's even harder to ensure that the **else** suites are associated with the correct **if** statement.

There has to be a better way...but what to do?

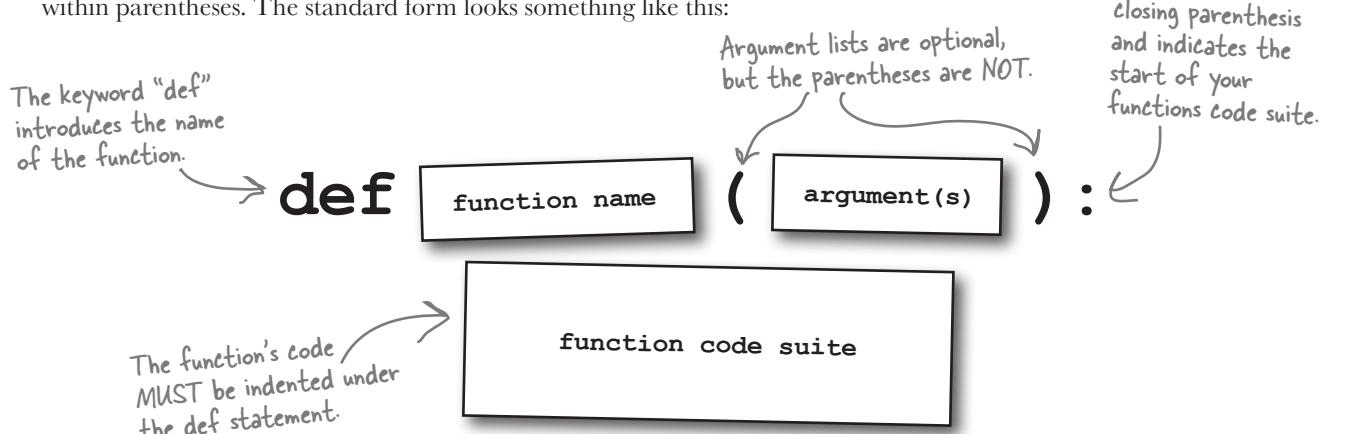
When code repeats in this way, most programmers look for a way to take the general pattern of the code and turn it into a **reusable function**. And Python programmers think this way, too. Creating a reusable function lets you invoke the function as needed, as opposed to cutting and pasting existing code.

So, let's turn the repeating code into a function.

Create a function in Python

A function in Python is a *named suite of code*, which can also take an **optional** list of arguments if required.

You define a Python function using the **def** statement, providing a name for your function and specifying either an empty or populated argument list within parentheses. The standard form looks something like this:



What does your function need to do?

Your function needs to take a list and process each item in the list. If it finds a nested list within the first list, the function needs to **repeat**. It can do this by *invoking itself* on the nested list. In other words, the function needs to **recur**—that is, *invoke itself* from within the function code suite.



Let's call the function that you'll create `print_lol()`. It takes one argument: a list to display on screen. Grab your pencil and complete the code below to provide the required functionality:

```
def print_lol(the_list):
```

```
    for .....
```

```
        if .....
```

```
    else:
```

recursive function



You were to call the function that you'll create `print_lol()`. It takes one argument: a list to display on screen. You were to grab your pencil and complete the code to provide the required functionality:

```
def print_lol(the_list):  
    Process the provided list with a "for" loop.    for each_item in the_list:  
        if isinstance(each_item, list):  
            print_lol(each_item)    If the item being processed  
        else:                                is itself a list, invoke the  
            print(each_item)    function.  
    If the item being processed ISN'T  
    a list, display the item on screen.
```



An IDLE Session

Let's use IDLE one final time to test your new function. Will it work as well as your earlier code?

```
>>> def print_lol(the_list):  
    for each_item in the_list:  
        if isinstance(each_item, list):  
            print_lol(each_item)  
        else:  
            print(each_item)  
  
>>> print_lol(movies)    Define the function.
```

```
The Holy Grail  
1975  
Terry Jones & Terry Gilliam  
91  
Graham Chapman  
Michael Palin  
John Cleese  
Terry Gilliam  
Eric Idle  
Terry Jones
```

It works, too! The recursive function produces EXACTLY the same results as the earlier code.

Recursion to the rescue!

The use of a *recursive function* has allowed you to **reduce** 14 lines of messy, hard-to-understand, brain-hurting code into a *six-line function*. Unlike the earlier code that needs to be amended to support additional nested lists (should the movie buff require them), the recursive function *does not need to change* to process any depth of nested lists properly.

Python 3 defaults its *recursion limit* to 1,000, which is a lot of lists of lists of lists...and this limit can be changed should you ever need even more depth than that.



What a great start!

By taking advantage of functions and recursion, you've solved the code complexity problems that had crept into your earlier list-processing code.

By creating `print_lol()`, you've produced a reusable chunk of code that can be put to use in many places in your (and others) programs.

You're well on your way to putting Python to work!



Your Python Toolbox

You've got Chapter 1 under your belt and you've added some key Python goodies to your toolbox.

Python Lingo

- “BIF” – a built-in function.
- “Suite” – a block of Python code, which is indented to indicate grouping.
- “Batteries included” – a way of referring to the fact that Python comes with most everything you’ll need to get going quickly and productively.

IDLE Notes

- The IDLE shell lets you experiment with your code as you write it.
- Adjust IDLE’s preferences to suit the way you work.
- Remember: when working with the shell, use Alt-P for Previous and use Alt-N for Next (but use Ctrl if you’re on a Mac).



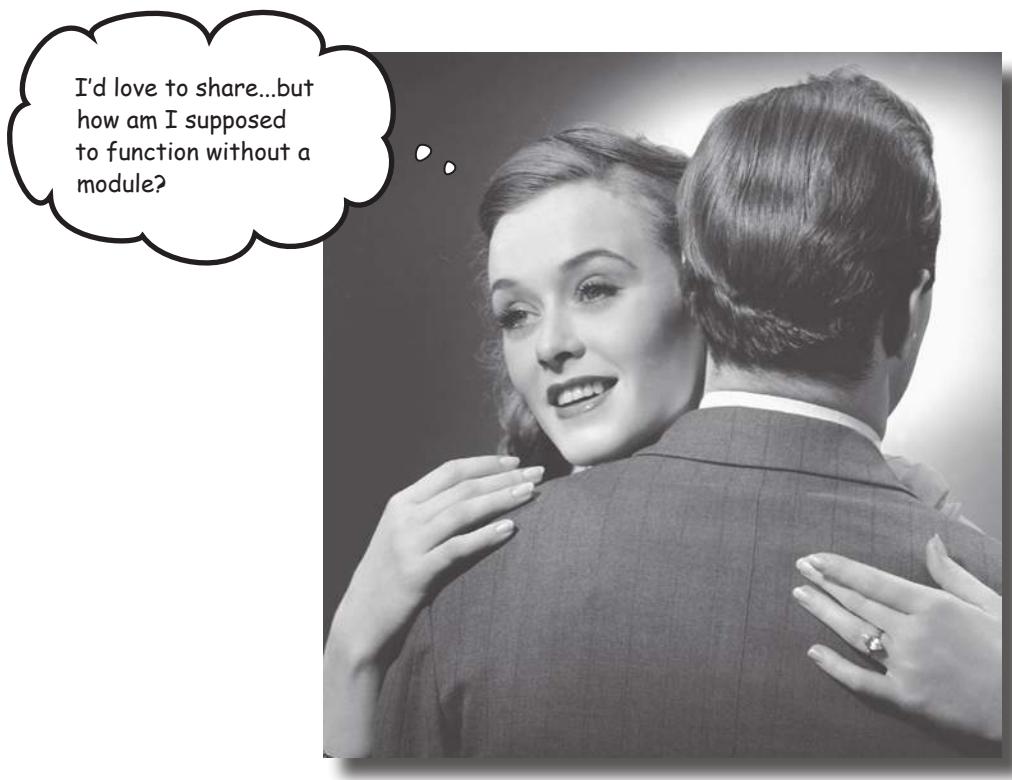
BULLET POINTS

- Run Python 3 from the command line or from within IDLE.
- Identifiers are names that refer to data objects. The identifiers have no “type,” but the data objects that they refer to do.
- `print()` BIF displays a message on screen.
- A **list** is a collection of data, separated by commas and surrounded by square brackets.
- Lists are like arrays on steroids.
- Lists can be used with BIFs, but also support a bunch of list methods.
- Lists can hold any data, and the data can be of mixed type. Lists can also hold other lists.
- Lists shrink and grow as needed. All of the memory used by your data is managed by Python for you.
- Python uses indentation to group statements together.
- `len()` BIF provides a length of some data object or count the number of items in a collection, such as a list.
- The **for** loop lets you iterate a list and is often more convenient to use than an equivalent **while** loop.
- The **if... else...** statement lets you make decisions in your code.
- `isinstance()` BIF checks whether an identifier refers to a data object of some specified type.
- Use **def** to define a custom function.

2 sharing your code



Modules of functions



Reusable code is great, but a shareable module is better.

By sharing your code as a Python module, you open up your code to the entire Python community...and it's always good to share, isn't it? In this chapter, you'll learn how to create, install, and distribute your own shareable modules. You'll then load your module onto Python's software sharing site on the Web, so that *everyone* can benefit from your work. Along the way, you'll pick up a few new tricks relating to Python's functions, too.

It's too good not to share

You've been showing your function to other programmers, and they like what they see.



Yes, a function this good should be shared with the world.

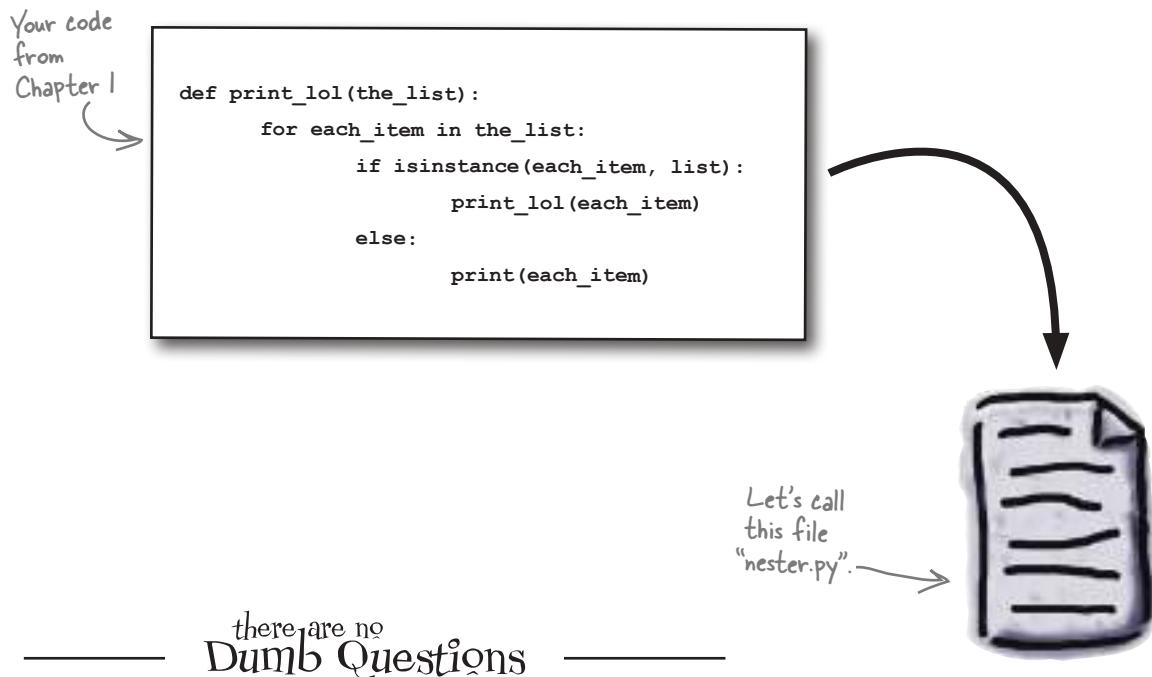
Python provides a set of technologies that make this easy for you, which includes *modules* and the *distribution utilities*:

- **Modules let you organize your code for optimal sharing.**
- **The distribution utilities let you share your modules with the world.**

Let's turn your function into a module, then use the distribution utilities to share your module with the wider Python programming community.

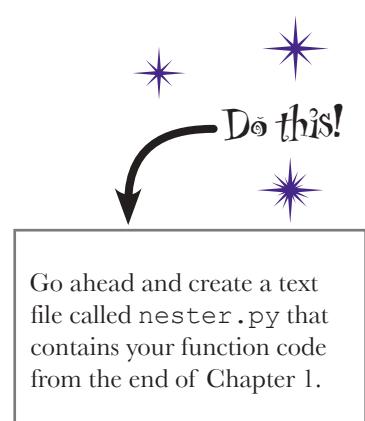
Turn your function into a module

A **module** is simply a text file that contains Python code. The main requirement is that the name of the file needs to end in `.py`: the Python extension. To turn your function into a module, save your code into an appropriately named file:



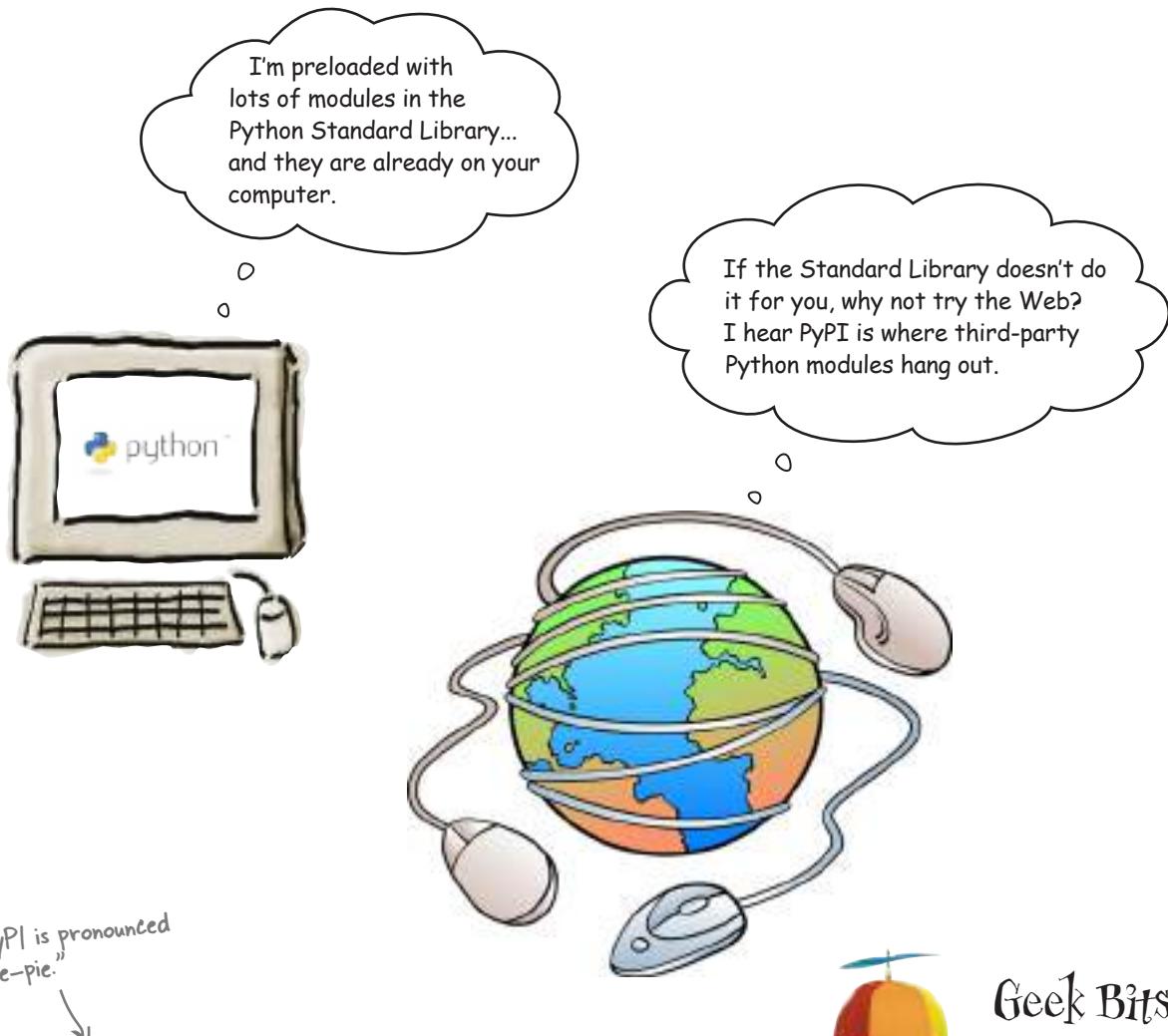
Q: What's the best Python editor?

A: The answer to that question really depends on who you ask. However, you can, of course, use *any* text editor to create and save your function's code in a text file. Something as simple as NotePad on Windows works fine for this, as does a full-featured editor such as TextMate on Mac OS X. And there's also full-fledged IDEs such as Eclipse on Linux, as well as the classic `vi` and `emacs` editors. And, as you already know, Python comes with IDLE, which also includes a built-in code editor. It might not be as capable as those other "real" editors, but IDLE is installed with Python and is essentially guaranteed to be available. For lots of jobs, IDLE's edit window is all the editor you'll ever need when working with your Python code. Of course, there are other IDEs for Python, too. Check out WingIDE for one that specifically targets Python developers.



Modules are everywhere

As might be expected, you'll find Python modules in lots of places.



The **Python Package Index** (or *PyPI* for short) provides a centralized repository for third-party Python modules on the Internet. When you are ready, you'll use PyPI to publish your module and make your code available for use by others. And your module *is* ready, *but* for one important addition.

What do you think is missing from your module?



Geek Bits

If you are already familiar with Perl's CPAN repository, you can think of PyPI as the Python equivalent.

Comment your code

It's always a good idea to include comments with your code. As your plan to share your module with the world, well-written comments help to document your work.

In Python, a common commenting technique is to use a **triple quote** for multiple-line comments. When you use a triple quote without assigning it to a variable, everything *between* the triple quotes is considered a comment:

Start with a triple quote...

```
"""This is the standard way to
include a multiple-line comment in
your code."""
```

...and end with a triple quote.

Hello! I'm a big string who just happens to be a Python comment, too. Nice, eh?



Sharpen your pencil

Put your module comment here.

Here is your module code (which is saved in the file `nester.py`). In the spaces provided, use your pencil to compose two comments: the first to describe the module and the second to describe the function.

```
def print_lol(the_list):
    ...
        for each_item in the_list:
            if isinstance(each_item, list)
                print_lol(each_item)
            else:
                print(each_item)
```

Add a comment for your function →



Here is your module code (which is saved in the file `nester.py`). In the spaces provided, you were asked to use your pencil to compose two comments: the first to describe the module and the second to describe the function.

“““This is the “`nester.py`” module, and it provides one function called `print_lol()` which prints lists that may or may not include nested lists.”““

Did you remember to include the triple quotes?

```
def print_lol(the_list):
```

“““This function takes a positional argument called “`the_list`”, which is any Python list (of, possibly, nested lists). Each data item in the provided list is (recursively) printed to the screen on its own line.”““

```
for each_item in the_list:  
    if isinstance(each_item, list):  
        print_lol(each_item)  
    else:  
        print(each_item)
```

There are no changes to the actual code here; you’re just adding some comments.

there are no Dumb Questions

Q: How do I know where the Python modules are on my computer?

A: Ask IDLE. Type `import sys; sys.path` (all on one line) into the IDLE prompt to see the list of locations that your Python interpreter searches for modules.

Q: Hang on a second. I can use “,” to put more than one line of code on the same line in my Python programs?

A: Yes, you can. However, I don’t recommend that you do so. Better to give each Python statement its own line; it makes your code much easier for you (and others) to read.

Q: Does it matter where I put my `nester.py` module?

A: For now, no. Just be sure to put it somewhere where you can find it later. In a while, you’ll install your module into your local copy of Python, so that the interpreter can find it without you having to remember when you actually put it.

Q: So comments are like a funny-looking string surrounded by quotes?

A: Yes. When a triple-quoted string is not assigned to a variable, it’s treated like a comment. The comments in your code are surrounded by three double quotes, but you could have used single quotes, too.

Q: Is there any other way to add a comment to Python code?

A: Yes. If you put a “#” symbol anywhere on a line, everything from that point to the end of the current line is a comment (unless the “#” appears within a triple quote, in which case it’s part of *that* comment). A lot of Python programmers use the “#” symbol to quickly switch on and off a single line of code when testing new functionality.



An IDLE Session

Now that you've added your comments and created a module, let's test that your code is still working properly. Rather than typing your function's code into IDLE's prompt, bring the `nester.py` file into IDLE's edit window, and then press F5 to run the module's code:

Note that the
comments are
color coded.

```

nester.py - /Users/benmp/HeadFirstPython/chapter2/nester.py

"""This is the "nester.py" module and it provides one function called print_lol()
which prints lists that may or may not include nested lists."""

def print_lol(the_list):
    """This function takes one positional argument called "the list", which
    is any Python list (of - possibly - nested lists). Each data item in the
    provided list is (recursively) printed to the screen on its own line."""

    for each_item in the_list:
        if isinstance(each_item, list):
            print_lol(each_item)
        else:
            print(each_item)

Ln 13 Col 0

```

Nothing appears to happen, other than the Python shell "restarting" and an empty prompt appearing:

```

>>> ===== RESTART =====
>>>
>>>

```

What's happened is that the Python interpreter has reset and the code in your module has executed. The code defines the function but, other than that, does little else. The interpreter is patiently waiting for you to do something with your newly defined function, so let's create a list of lists and invoke the function on it:

```

>>> movies = [
    "The Holy Grail", 1975, "Terry Jones & Terry Gilliam", 91, ← Define the list of movies facts
    ["Graham Chapman",
     ["Michael Palin", "John Cleese", "Terry Gilliam", "Eric Idle", "Terry Jones"]]


```

```

>>> print_lol(movies) ← Invoke the function on the list.
The Holy Grail

```

1975

Terry Jones & Terry Gilliam

91

Graham Chapman

Michael Palin

John Cleese

Terry Gilliam

Eric Idle

Terry Jones

Cool. Your code continues to
function as expected. The data
in the list of lists is displayed
on screen.

Prepare your distribution

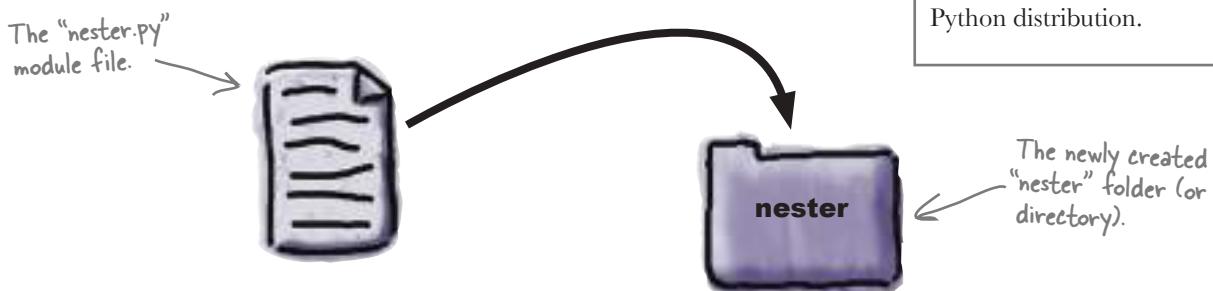
In order to share your newly created module, you need to prepare a **distribution**. This is the Python name given to the collection of files that together allow you to build, package, and distribute your module.

Once a distribution exists, you can install your module into your local copy of Python, as well as upload your module to PyPI to share with the world. Follow along with the process described on these two pages to create a distribution for your module.



1 Begin by creating a folder for your module.

With the folder created, copy your `nester.py` module file into the folder. To keep things simple, let's call the folder `nester`:



Follow along with each of the steps described on these pages. By the time you reach the end, your module will have transformed into a Python distribution.

2 Create a file called "setup.py" in your new folder.

This file contains metadata about your distribution. Edit this file by adding the following code:

```
Import the
"setup" function
from Python's
distribution utilities.

from distutils.core import setup

setup(
    name          = 'nester',
    version       = '1.0.0',
    py_modules    = ['nester'],
    author        = 'hfpython',
    author_email = 'hfpython@headfirstlabs.com',
    url          = 'http://www.headfirstlabs.com',
    description   = 'A simple printer of nested lists',
)
```

These are the setup function's argument names.

Associate your module's metadata with the setup function's arguments.

These are the values Head First Labs use with their modules; your metadata will be different.

Build your distribution

You now have a folder with two files in it: your module's code in `nester.py` and metadata about your module in `setup.py`. Now, it's time to build your distribution.

3 Build a distribution file.

The distribution utilities include all of the smarts required to build a distribution. Open a terminal window within your `nester` folder and type a single command: `python3 setup.py sdist`.

Enter the command at the prompt.

A collection of status messages appears on screen, confirming the creation of your distribution.

```
File Edit Window Help Build
$ python3 setup.py sdist
running sdist
running check
warning: sdist: manifest template 'MANIFEST.in' does not exist
warning: sdist: standard file not found: should have README

writing manifest file 'MANIFEST'
creating nester-1.0.0
making hard links in nester-1.0.0...
hard linking nester.py -> nester-1.0.0
hard linking setup.py -> nester-1.0.0
creating dist
Creating tar archive
removing 'nester-1.0.0' (and everything under it)
$
```

Note: if you are using Windows, replace "python3" in these commands with "c:\Python31\python.exe".

4 Install your distribution into your local copy of Python.

Staying in the terminal, type this command: `sudo python3 setup.py install`.

Another bunch of status messages appear on screen, confirming the installation of your distribution.

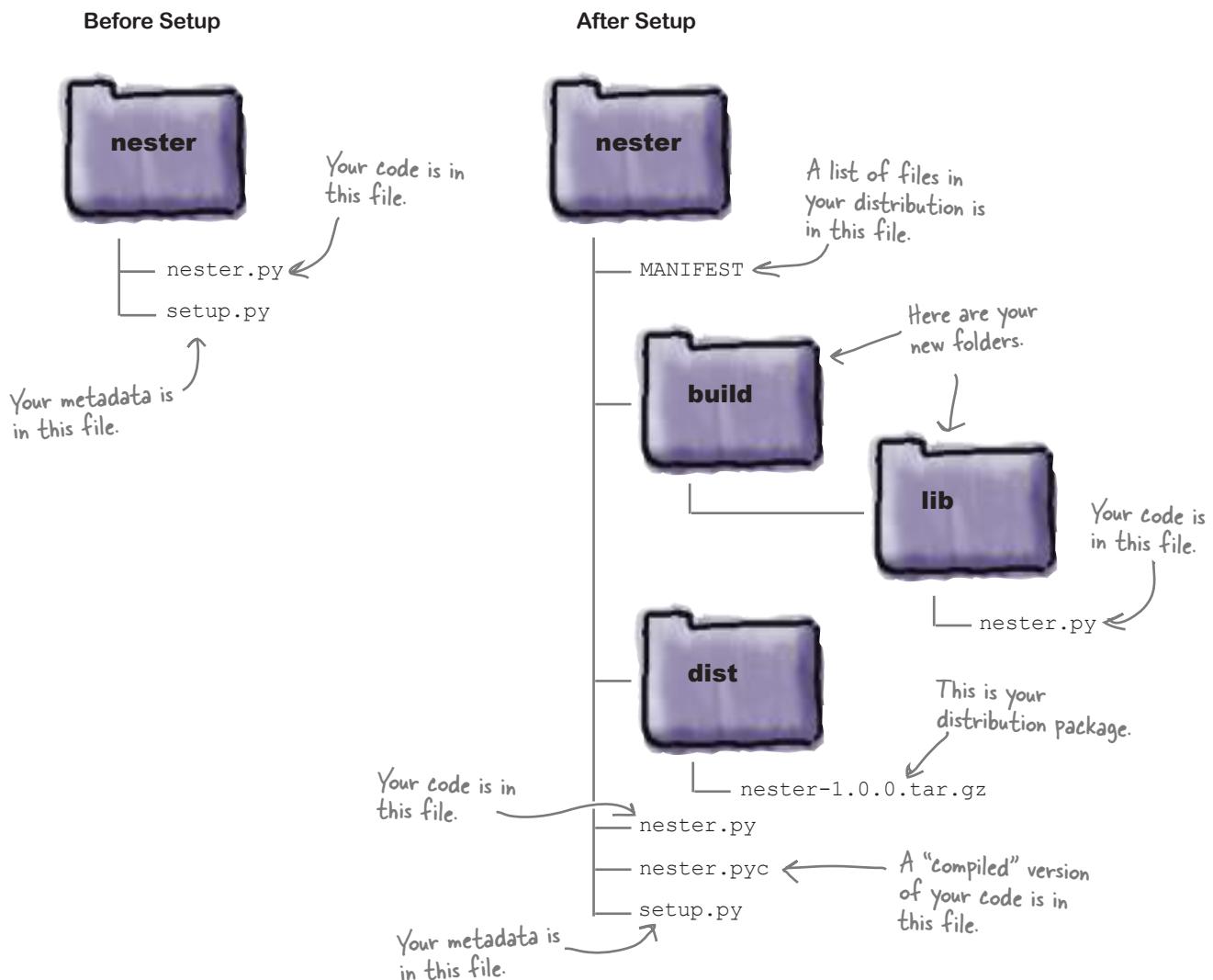
```
File Edit Window Help Install
$ python3 setup.py install
running install
running build
running build_py
creating build
creating build/lib
copying nester.py -> build/lib
running install_lib
copying build/lib/nester.py -> /Library/Frameworks/Python.framework/Versions/3.1/lib/python3.1/site-packages
byte-compiling /Library/Frameworks/Python.framework/Versions/3.1/lib/python3.1/site-packages/nester.py to nester.pyc
running install_egg_info
Writing /Library/Frameworks/Python.framework/Versions/3.1/lib/python3.1/site-packages/nester-1.0.0-py3.1.egg-info
```

Your distribution is ready.

A quick review of your distribution

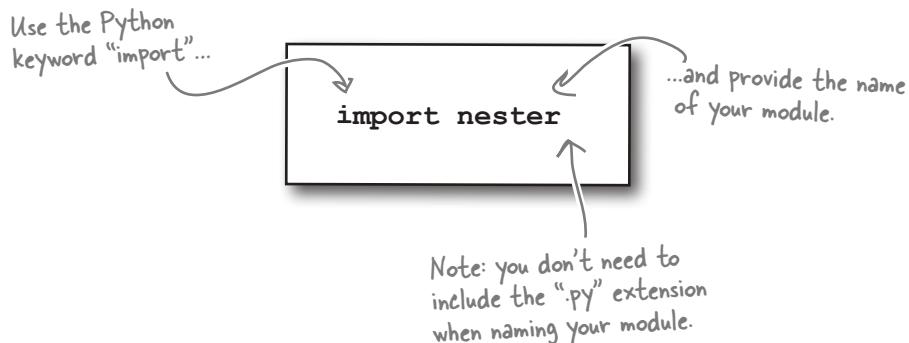
Thanks to Python's distribution utilities, your module has been *transformed* into a distribution and *installed* into your local copy of Python.

You started with a single function, which you entered into a file called `nester.py`, creating a module. You then created a folder called `nester` to house your module. The addition of a file called `setup.py` to your folder allowed you to build and install your distribution, which has resulted in a number of additional files and two new folders appearing within your `nester` folder. These files and folders are all created for you by the distribution utilities.



Import a module to use it

Now that your module is built, packaged as a distribution, and installed, let's see what's involved in using it. To use a module, simply `import` it into your programs or import it into the IDLE shell:



The `import` statement tells Python to include the `nester.py` module in your program. From that point on, you can use the module's functions as if they were entered directly into your program, right? Well...that's what you might expect. Let's check out the validity of your assumption.



Write a small program that imports your newly created module, defines a small list called "cast," and then uses the function provided by your module to display the contents of the list on screen. Use the following list data (all strings): Palin, Cleese, Idle, Jones, Gilliam, and Chapman.

Open your program in IDLE's edit window, and then press F5 to execute your code. Describe what happens in the space below:



Exercise Solution

You were to write a small program that imports your newly created module, defines a small list called “cast,” and then uses the function provided by your module to display the contents of the list on screen. You were to use the following list data (all strings): Palin, Cleese, Idle, Jones, Gilliam, and Chapman.

`import nester`

It's a simple three-line program. There's nothing too difficult here.

`cast = ['Palin', 'Cleese', 'Idle', 'Jones', 'Gilliam', 'Chapman']`

`print_lol(cast)`

Open your program in IDLE’s edit window, and then press F5 to execute your code. Describe what happens in the space below:

But it didn't work! → IDLE gives an error, and the program does not run!



An IDLE Session

With your program in the IDLE edit window, pressing F5 (or choosing Run Module from the Run menu) does indeed cause problems:

```
try_nester.py - /Users/barryp/HeadFirstPython/chapter2/try_nester.py
import nester
cast = ['Palin', 'Cleese', 'Idle', 'Jones', 'Gilliam', 'Chapman']
print_lol(cast)
Ln: 6 Col: 0
```

Your program does not appear to have executed and an error message is reported:

```
>>> ===== RESTART =====
>>>
>>> Traceback (most recent call last):
  File "/Users/barryp/HeadFirstPython/chapter2/try_nester.py", line 4, in <module>
    print_lol(cast)
NameError: name 'print_lol' is not defined
```

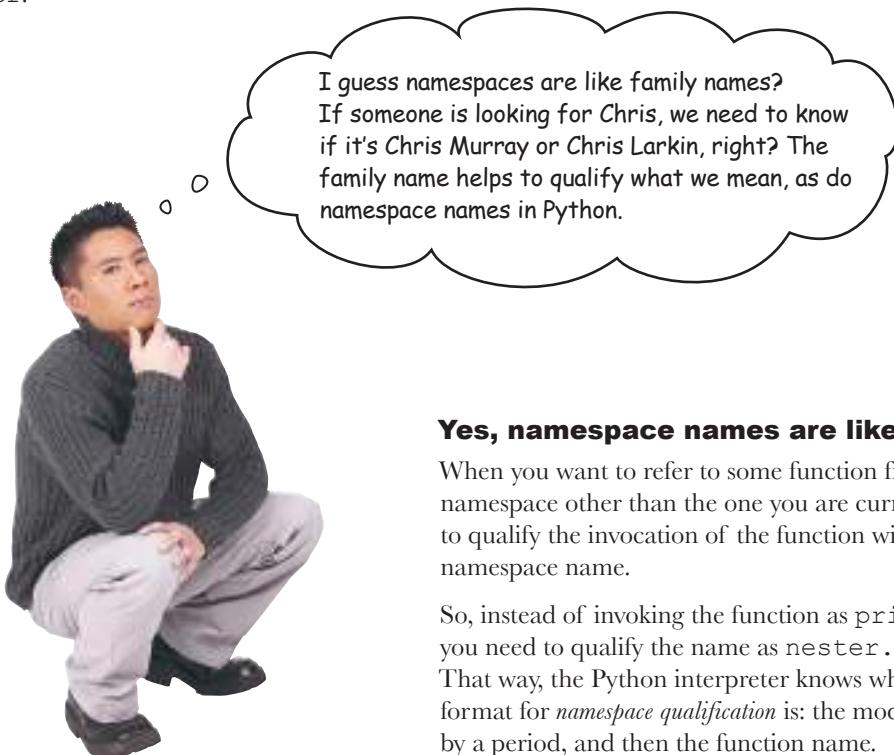
With your program in IDLE, pressing F5 causes a NameError...it looks like your function can't be found!!!

Python's modules implement namespaces

All code in Python is associated with a *namespace*.

Code in your main Python program (and within IDLE's shell) is associated with a namespace called `__main__`. When you put your code into its own module, Python automatically creates a namespace with the same name as your module. So, the code in your module is associated with a namespace called `nester`.

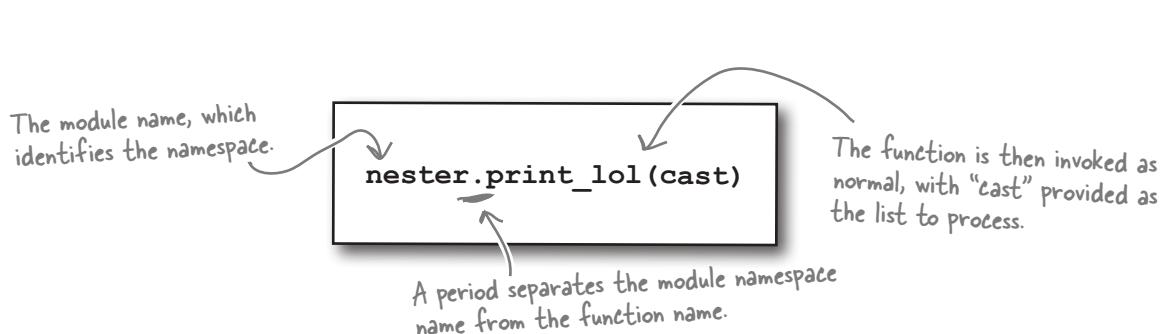
That's a double underscore in front of the word "main" and after it.



Yes, namespace names are like family names.

When you want to refer to some function from a module namespace other than the one you are currently in, you need to qualify the invocation of the function with the module's namespace name.

So, instead of invoking the function as `print_lol(cast)`, you need to qualify the name as `nester.print_lol(cast)`. That way, the Python interpreter knows where to look. The format for *namespace qualification* is: the module's name, followed by a period, and then the function name.





An IDLE Session

Let's test this. Staying at the IDLE shell, import your module, create the list, and then try to invoke the function without a qualifying name. You're expecting to see an error message:

```
>>> import nester
>>> cast = ['Palin', 'Cleese', 'Idle', 'Jones', 'Gilliam', 'Chapman']
>>> print_lol(cast)
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    print_lol(cast)
NameError: name 'print_lol' is not defined
```

As expected, your code has caused a NameError, because you didn't qualify the name.

When you qualify the name of the function with the namespace, things improve dramatically:

```
>>> nester.print_lol(cast)
Palin
Cleese
Idle
Jones
Gilliam
Chapman
```

This time, things work as expected...the data items in the list are displayed on screen.



Geek Bits

When you use a plain import statement, such as `import nester`, the Python interpreter is instructed to allow you to access `nester`'s functions using namespace qualification. However, it is possible to be *more specific*. If you use `from nester import print_lol`, the specified function (`print_lol` in this case) is *added* to the current namespace, effectively *removing* the requirement for you to use namespace qualification. But you need to be **careful**. If you already have a function called `print_lol` defined in your current namespace, the specific import statement overwrites your function with the imported one, which might not be the behavior you want.

Your module is now ready for upload to PyPI.

Register with the PyPI website

In order to upload your distribution to PyPI, you need to register with the PyPI website. This is a relatively straightforward process.

Begin by surfing over to the PyPI website at <http://pypi.python.org/> and requesting a PyPI ID:

The screenshot shows the "Manual user registration" form. Handwritten annotations provide guidance for each field:

- Username:** Provide the Username you'd like to use. Example: hfpthon.
- Password:** Enter your chosen password twice for confirmation purposes.
- Email Address:** Provide a valid email address. Example: richard@python.org
- PGP Key ID (optional):** Don't worry about providing a PGP key unless you actually have one.
- Usage Agreement:** By registering to upload content to PyPI, I agree and affirmatively acknowledge the following:
 - Content is restricted to Python packages and related information only.
 - Any content uploaded to PyPI is provided on a non-confidential basis.
 - The PSF is free to use or disseminate any content that I upload on an unrestricted basis. The rights granted to the web site are granted an irrevocable, worldwide, royalty-free, nonexclusive license to the content, including in digital form.
 - I represent and warrant that I have complied with all government regulations concerning my content, in particular, if I am subject to United States law, I represent and warrant that I have obtained all necessary approvals to upload the content I upload. I further affirm that any content I provide is not intended for use by any party in violation of the United States Export Administration Regulations.
- I agree:** Don't forget to click the "I agree" checkbox before clicking on the Register button.

If all of your registration details are in order, a confirmation message is sent to the email address submitted on the registration form. The email message contains a link you can click to confirm your PyPI registration:



You are now registered with PyPI.

Click the confirmation link to complete your PyPI registration.

Upload your code to PyPI

You're ready to rock! The code in your function has been placed in a module, used to create a distribution and installed into your local copy of Python. To upload your distribution to PyPI, complete these two steps: command-line registration with PyPI and command-line *uploading*.

It might seem strange to have to register with PyPI *again*, seeing as you just did this with their website. However, the command-line uploading tool needs to be made aware of your PyPI Username and Password, and that's what this registration does. Don't worry: you have to do this only once.

Instruct setup to →
register your details.

Confirm that you
want to use your
just-created PyPI
credentials. →

Use your PyPI
settings and save
them for future
use. →

```
File Edit Window Help Register
$ python3 setup.py register
running register
running check
We need to know who you are, so please choose either:
1. use your existing login,
2. register as a new user,
3. have the server generate a new password for you (and email it to you), or
4. quit
Your selection [default 1]:
1
Username: hpython
Password:
Registering nester to http://pypi.python.org/pypi
Server response (200): OK
I can store your PyPI login so future submissions will be faster.
(the login will be stored in /Users/barryp/.pypirc)
Save your login (y/N)?y
```

With your registration details entered and saved, you are now ready to upload your distribution to PyPI. Another command line does the trick:

Note: If you try to upload a module called "nester", you'll get an error as that name's already taken. ☺

Instruct setup to →
upload your software
distribution to PyPI. →

Setup confirms
that the upload →
is successful. Your
distribution is now
part of PyPI. →

```
File Edit Window Help Upload
$ python3 setup.py sdist upload
running sdist
running check
reading manifest file 'MANIFEST'
creating nester-1.0.0
making hard links in nester-1.0.0...
hard linking nester.py -> nester-1.0.0
hard linking setup.py -> nester-1.0.0
Creating tar archive
removing 'nester-1.0.0' (and everything under it)
running upload
Submitting dist/nester-1.0.0.tar.gz to http://pypi.python.org/pypi
Server response (200): OK
$
```

Welcome to the PyPI community

Congratulations! You are now a full-fledged, card-carrying member of the PyPI community. Your distribution has joined the over 10,000 other uploads on PyPI. Feel free to surf on over to the PyPI website to confirm the upload.

Programmers from all over the globe are now able to download, unpack, and install your module into their local copy of Python, which is pretty cool when you think about it.

Sit back, put your feet up, and wait for the plaudits to begin...

You've now written and published your code...
how cool is that?

there are no
Dumb Questions

Q: Which is best: plain imports or specific imports?

A: Neither, really. Most programmers mix and match based on their own personal preference and taste (although there are plenty of programmers willing to argue that their preferred way is the “one true way”).

Note that the `from module import function` form pollutes your current namespace: names already defined in your current namespace are *overwritten* by the imported names.

Q: And when I press F5 in IDLE’s edit window, it’s as if the module’s code is imported with an `import` statement, right?

A: Yes, that is essentially what happens. The code in your edit window is compiled and executed by Python, and any names in the edit window are imported into the namespace being used by IDLE’s shell. This is handy, because it makes it easy to test functionality with IDLE. But bear in mind that outside of IDLE, you still need to import your module *before* you can use its functionality.

Q: Is it really necessary for me to install my modules into my local copy of Python? Can’t I just put them in any old folder and import them from there?

A: Yes, it is possible. Just bear in mind that Python looks for modules in a very specific list of places (recall the `import sys; sys.path` trick from earlier in this chapter). If you put your modules in a folder not listed in Python’s path list, chances are the interpreter won’t find them, resulting in `ImportErrors`. Using the distribution utilities to build and install your module into your local copy of Python avoids these types of errors.

Q: I noticed the distribution utilities created a file called `nester.pyc`. What’s up with that?

A: That’s a very good question. When the interpreter executes your module code for the first time, it reads in the code and translates it into an internal bytecode format which is ultimately executed. (This idea is very similar to the way the Java JVM works: your Java code is turned into a class file as

a result of your Java technologies compiling your code.) The Python interpreter is smart enough to skip the translation phase the next time your module is used, because it can determine when you’ve made changes to the original module code file. If your module code hasn’t changed, no translation occurs and the “compiled” code is executed. If your code has changed, the translation occurs (creating a new `pyc` file) as needed. The upshot of all this is that when Python sees a `pyc` file, it tries to use it because doing so makes everything go much faster.

Q: Cool. So I can just provide my users with the `pyc` file?

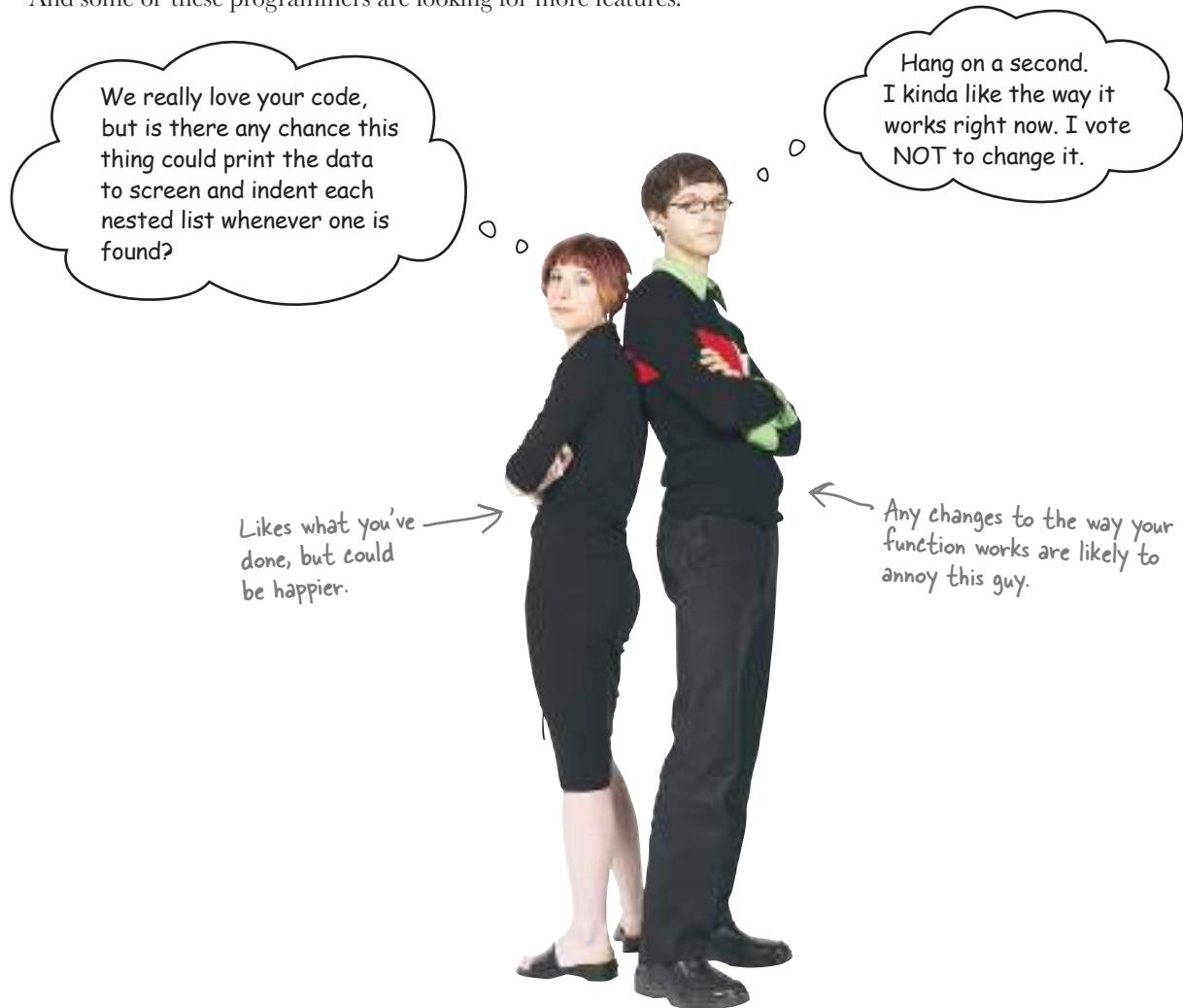
A: No, don’t do that, because the use of the `pyc` file (if found) is primarily a runtime optimization performed by the interpreter.

Q: So, can I delete the `pyc` file if I don’t need it?

A: Sure, if you really want to. Just be aware that you lose any potential runtime optimization.

With success comes responsibility

Lots of programmers from many different locations are using your module.
And some of these programmers are looking for more features.



Requests for change are inevitable

You need to keep your current users happy by maintaining the existing functionality, while at the same time providing enhanced functionality to those users that require it. This could be tricky.

What are your options here?

Life's full of choices

When it comes to deciding what to do here, there's no shortage of suggestions.

That's soooo easy. Simply create another function called "print_lol2", right? You could then import the function you want using the specific form of the import statement. It's not that hard, really...

Yeah, that might just work.

You could edit your module's code and define a new function called `print_lol2`, then code up the function to perform the nested printing. When you want to use the original function, use this specific form of the import statement: `from nester import print_lol`. When you want to use the new, improved version of the function, use this import statement: `from nester import print_lol2`.

Which would work, but...

But that suggestion is twice the work...which might be OK sometimes...but the creation of a second, almost identical, function seems wasteful to me.



Right. A second function is wasteful.

Not only are you introducing an almost identical function to your module, which might create a potential maintenance nightmare, but you're also making things much more difficult for the users of your module, who must decide *ahead of time* which version of the function they need. Adding a second function makes your module's *application programming interface* (API) more complex than it needs to be.

There has to be a better strategy, doesn't there?

Control behavior with an extra argument

If you add an extra argument to your function, you can handle indentation within your current code without too much trouble.



Take your function to the next level

At the moment, your function has a single argument: `the_list`. If you add a second argument called `level`, you can use it to control indentation. A positive value for `level` indicates the number of tab-stops to include when displaying a line of data on screen. If `level` is 0, no indentation is used; if it's 1, use a single tab-stop; if it's 2, use two tab-stops; and so on.

It's clear you are looking at some sort of looping mechanism here, right? You already know how to iterate over a variably sized list, but how do you iterate a fixed number of times in Python?

Does Python provide any functionality that can help?

Before you write new code, think BIF

When you come across a need that you think is *generic*, ask yourself if there's a built-in function (BIF) that can help. After all, iterating a fixed number of times is something you'll need to do *all the time*.

And remember: Python 3 includes over 70 BIFs, so there's a lot of functionality waiting to be discovered.



Use your pencil to draw a line matching each BIF to the correct description. The first one is done for you. Once you have all your lines drawn, circle the BIF you think you need to use in the next version of your function.

BIF

What the BIF does

`list()`

Creates a numbered list of paired-data, starting from 0.

`range()`

Returns the unique identification for a Python data object.

`enumerate()`

A factory function that creates a new, empty list.

`int()`

Returns the next item from a iterable data structure such as a list.

`id()`

Returns an iterator that generates numbers in a specified range on demand and as needed.

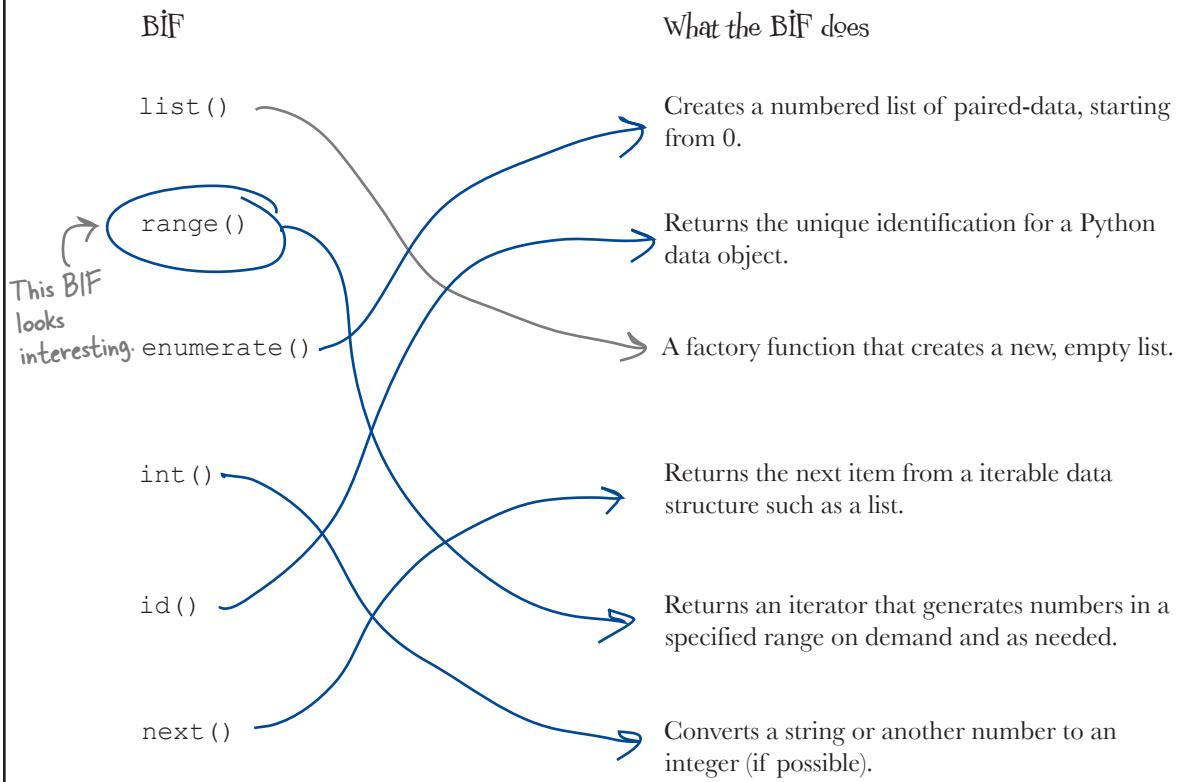
`next()`

Converts a string or another number to an integer (if possible).

WHO DOES WHAT?

SOLUTION

You were to use your pencil to draw a line matching each BIF to the correct description. Once you had all your lines drawn, you were to circle the BIF you think you need to use in the next version of your function.



The `range()` BIF iterates a fixed number of times

The `range()` BIF gives you the control you need to iterate a specific number of times and can be used to generate a list of numbers from zero up-to-but-not-including some number. Here's how to use it:

"num" is the target identifier and is assigned each of the numbers generated by "range()" in turn.

```
for num in range(4):
    print(num)
```

Generate numbers up-to-but-not-including 4.

The numbers 0, 1, 2, and 3 will appear on screen.

there are no
Dumb Questions

Q: Don't I need to import the BIFs in order to use them in my program?

A: No. For all intents and purposes, the BIFs are specifically imported into every Python program as well as IDLE.

Q: So the BIFs must belong to the `__main__` namespace, right?

A: No. They are automatically imported into the `__main__` namespace, but the BIFs have their very own namespace called (wait for it) `__builtins__`.

Q: I get how `range()` works, but surely I could just as easily use a `while` loop to do the same thing?

A: Yes, you can, but it's not as elegant as using `range()`. Seriously, though, the `while` equivalent not only requires you to write more code, but it also makes it your responsibility to worry about loop state, whereas `range()` worries about this for you. As a general rule, Python programmers look for ways to reduce the amount of code they need to write and worry about, which leads to better code robustness, fewer errors, and a good night's sleep.

Q: So BIFs are actually good for me?

A: BIFs exist to make your programming experience as straightforward as possible by providing a collection of functions that provide common solutions to common problems. Since they are included with Python, you are pretty much assured that they have been tested to destruction and do "exactly what it says on the tin." You can depend on the BIFs. Using them gives your program a leg up and makes you look good. So, yes, the BIFs are good for you!



Exercise

Now that you know a bit about the `range()` BIF, amend your function to use `range()` to indent any nested lists a specific number of tab-stops.

Hint: To display a TAB character on screen using the `print()` BIF yet avoid taking a new-line (which is `print()`'s default behavior), use this Python code: `print("\t", end='')`.

```
"""This is the "nester.py" module and it provides one function called print_lol()
which prints lists that may or may not include nested lists."""
```

```
def print_lol(the_list, .....):
```

Include the name of the extra argument.

```
    """This function takes a positional argument called "the_list", which
is any Python list (of - possibly - nested lists). Each data item in the
provided list is (recursively) printed to the screen on it's own line."""
```

```
for each_item in the_list:
```

```
    if isinstance(each_item, list):
```

```
        print_lol(each_item)
```

```
    else:
```

*Don't forget to edit
the comment.*

```
        .....  
        print(each_item)
```

*Add code here to take the required
number of tab-stops.*



Exercise Solution

Now that you know a bit about the `range()` BIF, you were to amend your function to use `range()` to indent any nested lists a specific number of tab-stops.

Hint: To display a TAB character on screen using the `print()` BIF yet avoid taking a new-line (which is `print()`'s default behavior), use this Python code: `print("\t", end='')`.

```
"""This is the "nester.py" module and it provides one function called print_lol()
which prints lists that may or may not include nested lists."""
```

```
def print_lol(the_list, ..... level .....):
    """This function takes a positional argument called "the_list", which
       is any Python list (of - possibly - nested lists). Each data item in the
       provided list is (recursively) printed to the screen on its own line.
    A second argument called "level" is used to insert tab-stops when a nested list is encountered."""
    for each_item in the_list:
        if isinstance(each_item, list):
            print_lol(each_item)
        else:
            for tab_stop in range(level):
                .....  

                print("\t", end="") ← Display a TAB character for each level
            print(each_item)
```

Use the value of "level" to control
how many tab-stops are used.



An IDLE Session

It's time to test the new version of your function. Load your module file into IDLE, press F5 to import the function into IDLE's namespace, and then invoke the function on your `movies` list with a second argument:

```
>>> print_lol(movies, 0) ← Invoke your function, being sure to provide
                           a second argument.  

The Holy Grail
1975
Terry Jones & Terry Gilliam } ← The data in "movies" starts to appear on screen...
91
Traceback (most recent call last): ← ...then all hell breaks loose! Something is not right here.
  File "<pyshell#2>", line 1, in <module>
    print_lol(movies,0)
  File "/Users/barryp/HeadFirstPython/chapter2/nester/nester.py", line 14, in print_lol
    print_lol(each_item)
TypeError: print_lol() takes exactly 2 positional arguments (1 given) ← Here's your clue as to
                                                               what's gone wrong.
```

Your code has a `TypeError`, which caused it to crash.

Python tries its best to run your code

Unlike compiled languages (such as C, Java, C#, and others), Python doesn't *completely* check the validity of your code *until it runs*. This allows Python to do some rather cool things that just aren't possible in those other languages, such as dynamically defining functions at runtime. This, of course, can be very flexible and powerful.

The cost to you, however, is that you need to be very careful when writing your code, because something that typically would be caught and flagged as an "error" by a traditional, statically typed, compiled language often goes unnoticed in Python.



Take another look at the error on the opposite page. Why do you think Python is giving you this particular error? What do you think is wrong?

Trace your code

When you're trying to work out what went wrong with a program that looks like it should be OK, a useful technique is to trace what happens as each line of code executes. Here's the code that you are currently working with. At only three lines long (remember: the creation of the list is one line of code), it doesn't look like it should cause any trouble:

```
import nester ← These two lines look OK.

movies = [ "The Holy Grail", 1975, "Terry Jones & Terry Gilliam",
           91, ["Graham Chapman", ["Michael Palin",
           "John Cleese", "Terry Gilliam", "Eric Idle",
           "Terry Jones"]]]]

nester.print_lol(movies, 0) ← You are invoking the function with two arguments,
                           so that's OK, too.
```

With the data assigned to the function's arguments, the function's code starts to execute on each data item contained within the passed-in list:

The "movies" list is assigned to "the_list", and the value 0 is assigned to "level".

```
To save space, the entire comment is not shown.

Process each item in the list...
...then decide what to do next based on whether or not the data item is a list.

def print_lol(the_list, level):
    """This function ..."""

    for each_item in the_list:
        if isinstance(each_item, list):
            print_lol(each_item) ← If the data item is a list, recursively invoke
                                   the function...hang on a second, that doesn't
                                   look right!?
        else:
            for tab_stop in range(level):
                print("\t", end=' ')
            print(each_item)
```

Work out what's wrong

There's your problem: the recursive invocation of your function is using the old *function signature* that required only one argument. The new version of your function requires two arguments.

The fix is easy: provide the correct number of arguments when calling the new version of your function. So, this snippet of code from your function:

```
if isinstance(each_item, list):
    print_lol(each_item)
```

needs to be rewritten to specify the correct number of arguments:

```
if isinstance(each_item, list):
    print_lol(each_item, level)
```



Not so fast. Surely the nested list needs to be printed after a specific number of tab-stops? At the moment, your code sets "level" to 0 but never changes the value, so "level" never has any impact on your displayed output...

Right. Your use of “level” needs one final tweak.

The whole point of having `level` as an argument is to allow you to control the nested output. Each time you process a nested list, you need to increase the value of `level` by 1. Your code snippet needs to look like this:

```
if isinstance(each_item, list):
    print_lol(each_item, level+1)
```

It's time to perform that update.

Simply increment the value of `level` by 1 each time you recursively invoke your function.

Update PyPI with your new code

Go ahead and edit your `nester.py` module (in the `nester` folder) to invoke your function properly. Now that you have a new version of your module, it's a good idea to update the distribution that you uploaded to PyPI.

With your code amended, there's also a small change needed to your distribution's `setup.py` program. You've changed your API, so adjust the value associated with `version` in `setup.py`. Let's move from version 1.0.0 to 1.1.0:

```
from distutils.core import setup

setup(
    name      = 'nester',
    version   = '1.1.0', Change the value  
associated with "version"  
to indicate to PyPI that  
this is indeed a new  
version.
    py_modules = ['nester'],
    author    = 'hfpython',
    author_email = 'hfpython@headfirstlabs.com',
    url       = 'http://www.headfirstlabs.com',
    description = 'A simple printer of nested lists',
)
```

Just as you did when you created and uploaded your distribution, invoke the `setup.py` program within your distribution folder to perform the upload:

```
File Edit Window Help UploadAgain
$ python3 setup.py sdist upload
running sdist
running check
reading manifest file 'MANIFEST'
creating nester-1.1.0
making hard links in nester-1.1.0...
hard linking nester.py -> nester-1.1.0
hard linking setup.py -> nester-1.1.0
Creating tar archive
removing 'nester-1.1.0' (and everything under it)
running upload
Submitting dist/nester-1.1.0.tar.gz to http://pypi.python.org/pypi
Server response (200): OK
$
```

Don't you just
love those
"200 OK"
messages?



Your new distribution is now available on PyPI.



Mark: Take a look at this, guys...the `nester` module has been updated on PyPI.

Bob: Version 1.1.0...

Laura: I wonder what's changed?

Mark: It still works with nested lists of lists, but now you can see the nested structure on screen, which I think is pretty cool.

Laura: And useful. I've been waiting for that feature.

Bob: Eh...OK...but how do I upgrade my existing local copy?

Mark: Just follow the same steps as when you downloaded and installed `nester` from PyPI the first time.

Bob: So I download the package file, unpack it, and ask `setup.py` to install it into my Python for me?

Mark: Yes. It couldn't be any easier.

Laura: And what about my existing version of `nester`; what happens to that "old" version?

Bob: Yeah...do I have two `nester` modules now?

Mark: No. When you use `setup.py` to install the latest version it becomes the current version and effectively replaces the previous module, which was the 1.0.0 release.

Bob: And PyPI knows to give you the latest version of the module, too, right?

Mark: Yes, when you surf the PyPI website and search for `nester`, you are always provided with the latest version of the module.

Laura: Well, I use this module all the time and I've been waiting for this feature. I think I'll update right away.

Mark: I've already upgraded mine, and it works a treat.

Bob: Yeah, I use it a lot, too, so I guess I'll keep my system up to date and install the latest version. It's probably not a good idea to rely on out-of-date software, right?

Mark: I'd say. And, there's nothing quite like progress.

Laura: Catch you later, guys, I've got work to do.

Bob: Me, too. I'm off to PyPI to grab the latest `nester` and install it into my local copy of Python. I'll give it a quick test to confirm all is OK.

Mark: Later, dudes...

You've changed your API

Your new version of `nester` is indeed better, but not for all your users.



In your rush to release the latest and greatest version of your module, you forgot about *some* of your existing users. Recall that not all of your users want the new nested printing feature. However, by adding the second argument to `print_lol()`, you've changed your function's signature, which means your module has a different API. Anyone using the old API is going to have problems.

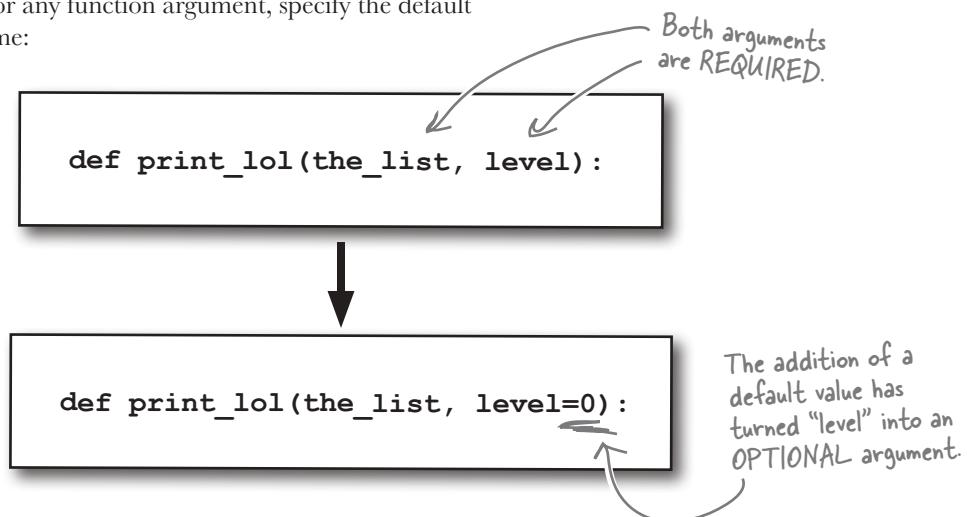
The ideal solution would be to provide **both** APIs, one which switches on the new feature and another that doesn't. Maybe the feature could be optional?

But how would that work?

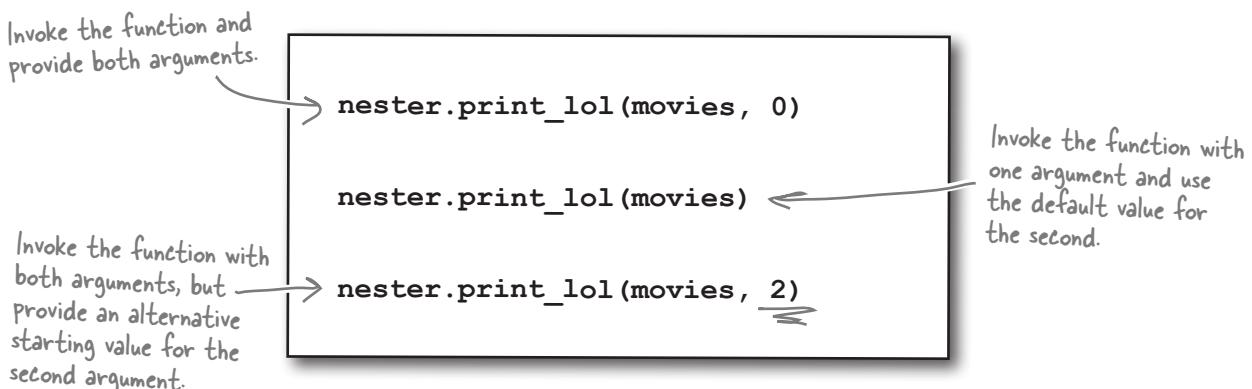
Use optional arguments

To turn a required argument to a function into an *optional* argument, provide the argument with a default value. When no argument value is provided, the default value is used. When an argument value is provided, it is used *instead* of the default. The key point is, of course, that the default value for the argument effectively makes the argument optional.

To provide a default value for any function argument, specify the default value *after* the argument name:



With the default value for the argument defined, you can now invoke the function in a number of different ways:



Your function now supports different signatures, but the functionality remains as it was.



An IDLE Session

Amend your code to give the `level` argument a default value of 0 and then load your code into the IDLE editor. Press F5 to load the code into the shell and then follow along to confirm that the latest version of your function works as expected. Start by defining a short list of lists and use the function to display the list on screen:

```
>>> names = ['John', 'Eric', ['Cleese', 'Idle'], 'Michael', ['Palin']]  
>>> print_lol(names, 0)
```

```
John  
Eric  
Cleese  
Idle  
Michael  
Palin
```

The standard behavior works as expected, with nested lists indented.

Now try to do the same thing without specifying the second argument. Let's rely on the default value kicking in:

```
>>> print_lol(names)
```

```
John  
Eric  
Cleese  
Idle  
Michael  
Palin
```

Without specifying the second argument, the default is used and works, too.

Now specify a value for the second argument and note the change in the function's behavior:

```
>>> print_lol(names, 2)
```

```
John  
Eric  
Cleese  
Idle  
Michael  
Palin
```

Specify an alternative value for the second argument and the indenting starts from that level.

One final example provides what looks like a silly value for the second argument. Look what happens:

```
>>> print_lol(names, -9)
```

```
John  
Eric  
Cleese  
Idle  
Michael  
Palin
```

Using a negative value effectively switches OFF the indenting, as the count for "level" is unlikely to become a positive integer. This looks exactly like the original output from version 1.0.0, right?

Your module supports both APIs

Well done! It looks like your module is working well, as both APIs, the original 1.0.0 API and the newer 1.1.0 API, can now be used.

Let's take a moment to create and upload a new distribution for PyPI. As before, let's amend the `version` setting in the `setup.py` program:

```
name      = 'nester',
version   = '1.2.0',
py_modules = ['nester'],
```

Once again, be sure to change the value associated with "version" in "setup.py".

And with the code changes applied, upload this new version of your distribution to PyPI:

```
File Edit Window Help UploadThree
$ python3 setup.py sdist upload
running sdist
running check
reading manifest file 'MANIFEST'
creating nester-1.2.0
making hard links in nester-1.2.0...
hard linking nester.py -> nester-1.2.0
hard linking setup.py -> nester-1.2.0
Creating tar archive
removing 'nester-1.2.0' (and everything under it)
running upload
Submitting dist/nester-1.2.0.tar.gz to http://pypi.python.org/pypi
Server response (200): OK
$
```

This all looks fine and dandy.



Success! The messages from `setup.py` confirm that your latest version of `nester` is up on PyPI. Let's hope this one satisfies all of your users.



Consider your code carefully. How might some of your users still have a problem with this version of your code?

Your API is still not right

Although the API lets your users invoke the function in its original form, *the nesting is switched on by default*. This behavior is not required by everyone and some people aren't at all happy.



Of course, if you have some functionality that really ought to be optional (that is, *not* the default), you should adjust your code to make it so. But how?

One solution is to **add a third argument** which is set to `True` when the indenting is required and `False` otherwise. If you ensure that this argument is `False` by default, the original functionality becomes the default behavior and users of your code have to request the new indenting feature explicitly.

Let's look at adding this final revision.

**1**

Amend your module one last time to add a third argument to your function. Call your argument `indent` and set it initially to the value `False`—that is, do not switch on indentation by default. In the body of your function, use the value of `indent` to control your indentation code.

Note: to save a bit of space, the comments from the module are not shown here. Of course, you need to make the necessary adjustments to your comments to keep them in sync with your code.

```

Put the extra argument
here.
↓
def print_lol(the_list, ..... , level=0):
    for each_item in the_list:
        if isinstance(each_item, list):
            print_lol(each_item, ..... , level+1)
        else:
            Add a line of code → ..... for tab_stop in range(level):
                print("\t", end='')
            print(each_item)
    What needs to
    go in here?
    ↓

```

2

With your new code additions in place, provide the edit you would recommend making to the `setup.py` program prior to uploading this latest version of your module to PyPI:

.....
.....

3

Provide the command you would use to upload your new distribution to PyPI:

.....
.....



Exercise Solution

1

You were to amend your module one last time to add a third argument to your function. You were to call your argument `indent` and set it initially to the value `False`—that is, do not switch on indentation by default. In the body of your function, you were to use the value of `indent` to control your indentation code.

Did you include the
default value?

```
def print_lol(the_list, ..... , level=0):
```

```
    for each_item in the_list:
```

```
        if isinstance(each_item, list):
```

```
            print_lol(each_item, ..... , indent=....., level+1)
```

```
        else:
```

A simple "if" statement does the trick.

```
            if indent: ← Don't forget the colon at the end of the "if" line.
```

```
                for tab_stop in range(level):
```

```
                    print("\t", end='')
```

```
            print(each_item)
```

Your signature has changed,
so be sure to update this
invocation.

A sweet alternative to this "for" loop
is this code: `print("\t" * level, end="")`.

2

With your new code additions in place, you were to provide the edit you would recommend making to the `setup.py` program prior to uploading this latest version of your module to PyPI:

Edit "setup.py" so that it reads: `version = '1.3.0'`



It's a new version of your module, so be sure to change the value associated with "version" in your "setup.py" file.

3

You were to provide the command you would use to upload your new distribution to PyPI:

`python3 setup.py sdist upload`

Remember: if you are on Windows use "C:\Python31\python.exe"
instead of "python3".



An IDLE Session

A final test of the functionality should convince you that your module is now working exactly the way you and your users want it to. Let's start with the original, default behavior:

```
>>> names = ['John', 'Eric', ['Cleese', 'Idle'], 'Michael', ['Palin']]
```

```
>>> print_lol(names)
```

John

Eric

Cleese

Idle

Michael

Palin

The original, default functionality
is restored (that should please
Bob).

Next, turn on indentation by providing `True` as the second argument:

```
>>> names = ['John', 'Eric', ['Cleese', 'Idle'], 'Michael', ['Palin']]
```

```
>>> print_lol(names, True)
```

John

Eric

Cleese

Idle

Michael

Palin

By providing a second argument,
it's possible to switch on indented
output (keeping Laura happy).

And, finally, control where indentation begins by providing a third argument value:

```
>>> names = ['John', 'Eric', ['Cleese', 'Idle'], 'Michael', ['Palin']]
```

```
>>> print_lol(names, True, 4)
```

John

Eric

Cleese

Idle

Michael

Palin

Indenting from a specific
tab-stop is also possible.



Go ahead and edit your
`setup.py` file; then upload
your distribution to PyPI.

Your module's reputation is restored

Congratulations! Word of your new and improved module is spreading fast.



Your Python skills are starting to build

You've created a useful module, made it shareable, and uploaded it to the PyPI website. Programmers all over the world are downloading and using your code in their projects.

Keep up the good work.



Your Python Toolbox

You've got Chapter 2 under your belt and you've added some key Python goodies to your toolbox.

Python Lingo

- Use a “triple-quoted string” to include a multiple-line comment in your code.
- “PyPI” is the Python Package Index and is well worth a visit.
- A “namespace” is a place in Python’s memory where names exist.
- Python’s main namespace is known as `__main__`.

IDLE Notes

- Press F5 to “run” the code in the IDLE edit window.
- When you press F5 to “load” a module’s code into the IDLE shell, the module’s names are specifically imported into IDLE’s namespace. This is a convenience when using IDLE. Within your code, you need to use the `import` statement explicitly.



BULLET POINTS

- A **module** is a text file that contains Python code.
- The **distribution utilities** let you turn your module into a shareable package.
- The `setup.py` program provides *metadata* about your module and is used to build, install, and upload your packaged distribution.
- Import your module into other programs using the `import` statement.
- Each module in Python provides its own **namespace**, and the namespace name is used to qualify the module’s functions when invoking them using the `module.function()` form.
- Specifically import a function from a module into the current namespace using the `from module import function` form of the `import` statement.
- Use `#` to comment-out a line of code or to add a short, one-line comment to your program.
- The built-in functions (BIFs) have their own namespace called `__builtins__`, which is automatically included in every Python program.
- The `range()` BIF can be used with `for` to iterate a fixed number of times.
- Including `end=''` as a argument to the `print()` BIF switches off its automatic inclusion of a new-line on output.
- Arguments to your functions are **optional** if you provide them with a **default value**.

3 files and exceptions



* Dealing with errors *

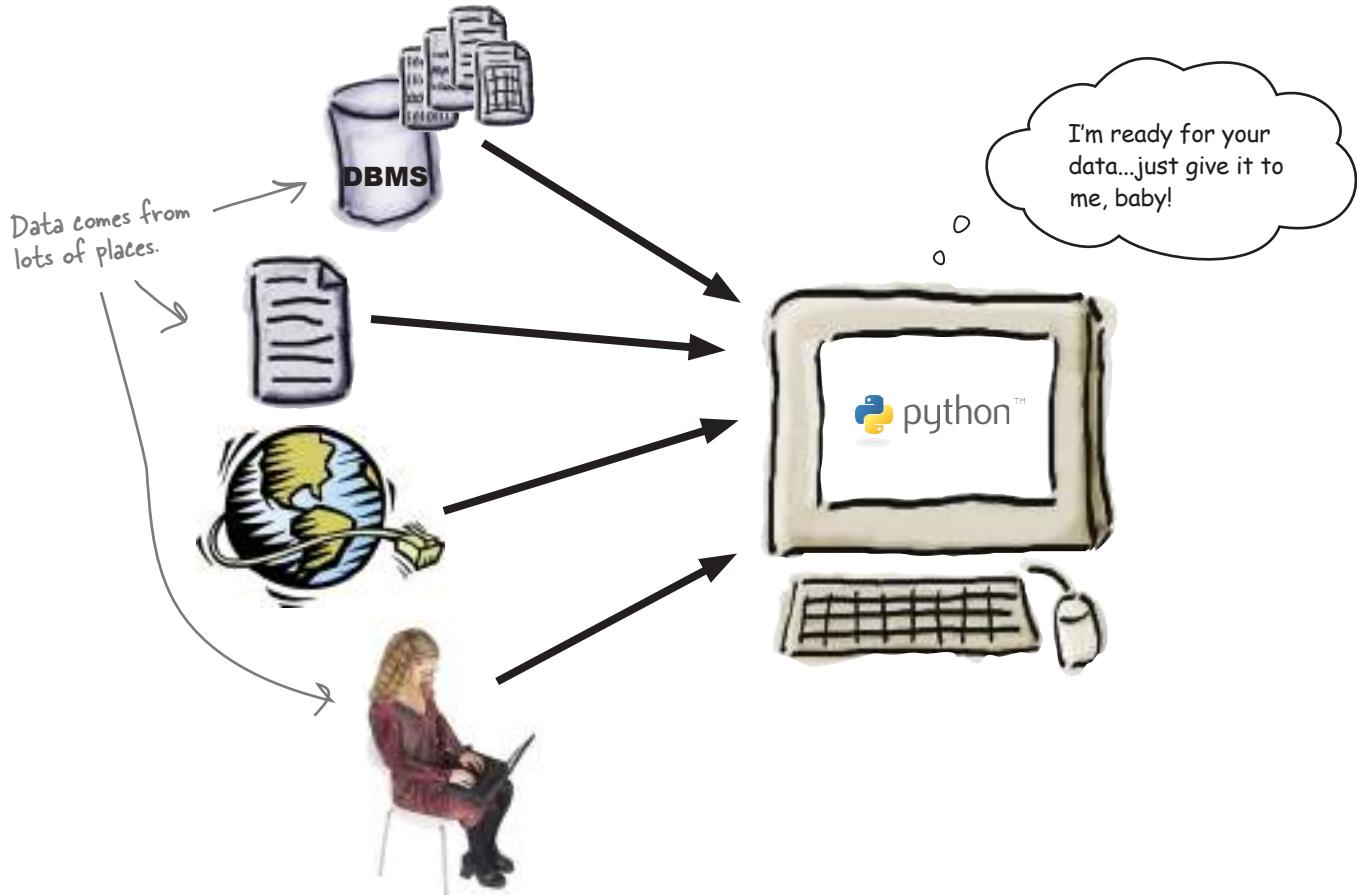


It's simply not enough to process your list data in your code.

You need to be able to get your data *into* your programs with ease, too. It's no surprise then that Python makes reading data from **files** easy. Which is great, until you consider what can go *wrong* when interacting with data *external* to your programs...and there are lots of things waiting to trip you up! When bad stuff happens, you need a strategy for getting out of trouble, and one such strategy is to deal with any exceptional situations using Python's **exception handling** mechanism showcased in this chapter.

Data is external to your program

Most of your programs conform to the *input-process-output model*: data comes in, gets manipulated, and then is stored, displayed, printed, or transferred.



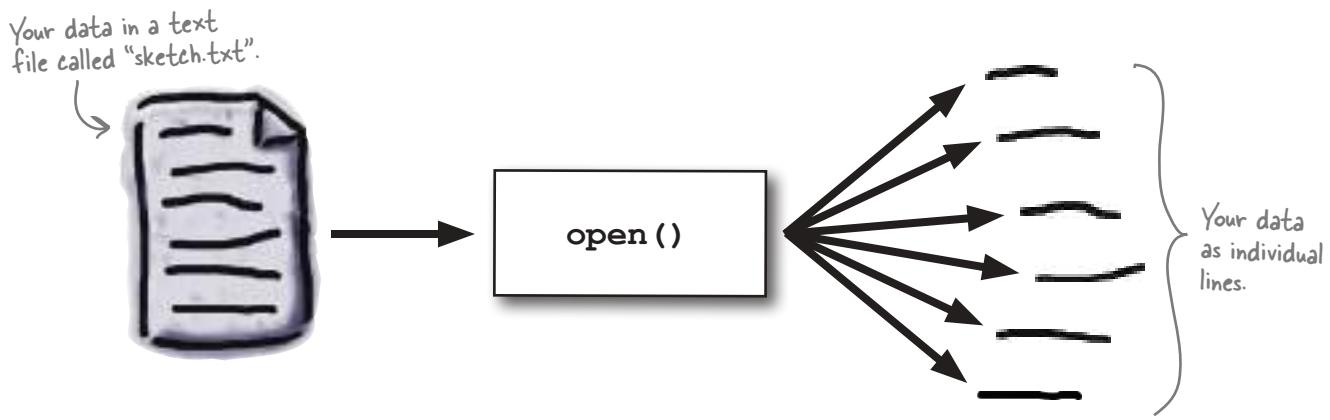
So far, you've learned how to **process** data as well as **display** it on screen. But what's involved in getting data into your programs? Specifically, what's involved in reading data from a file?

How does Python read data from a file?

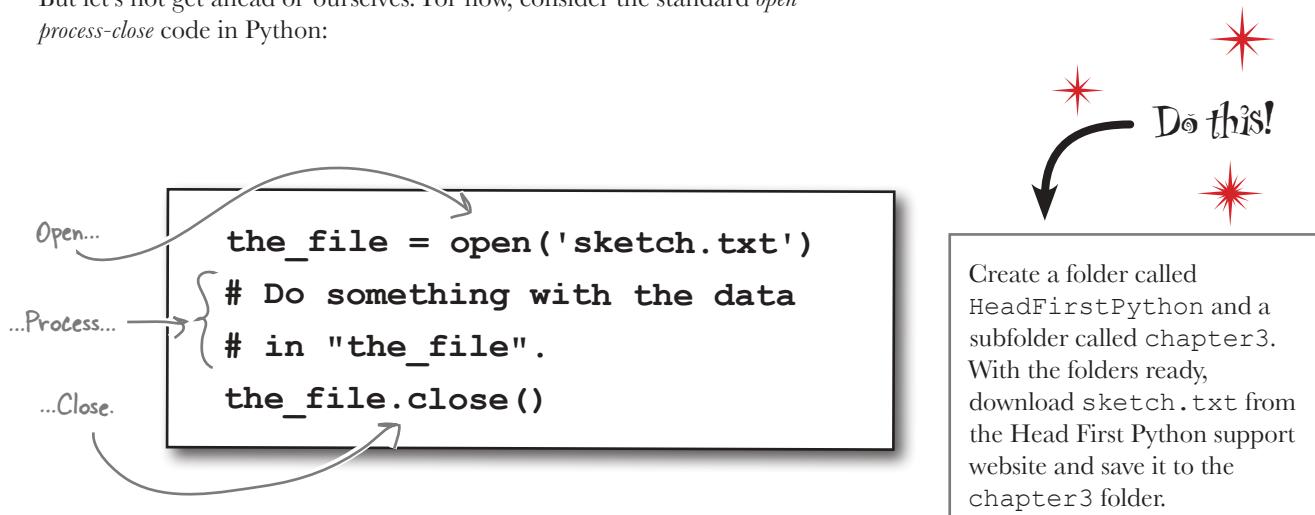
It's all lines of text

The basic input mechanism in Python is **line based**: when read into your program from a text file, data arrives one line at a time.

Python's `open()` BIF lives to interact with files. When combined with a **for** statement, reading files is straightforward.



When you use the `open()` BIF to access your data in a file, an **iterator** is created to feed the lines of data from your file to your code one line at a time. But let's not get ahead of ourselves. For now, consider the standard *open-process-close* code in Python:



Let's use IDLE to get a feel for Python's file-input mechanisms.



An IDLE Session

Start a new IDLE session and import the `os` module to change the current working directory to the folder that contains your just-downloaded data file:

```
>>> import os ← Import "os" from the Standard Library.  
>>> os.getcwd() } ← What's the current working directory?  
'/Users/barryp/Documents' } ← Change to the folder that contains your data file.  
>>> os.chdir('../HeadFirstPython/chapter3') } ← Confirm you are now in the right place.  
>>> os.getcwd()  
'/Users/barryp/HeadFirstPython/chapter3'
```

Now, open your data file and read the first two lines from the file, displaying them on screen:

```
>>> data = open('sketch.txt') ← Open a named file and assign the file to a file object called "data".  
>>> print(data.readline(), end='')  
Man: Is this the right room for an argument? } Use the "readline()" method to grab a  
>>> print(data.readline(), end='') } line from the file, then use the "print()" Other Man: I've told you once.  
BIF to display it on screen.
```

Let's "rewind" the file back to the start, then use a `for` statement to process every line in the file:

```
>>> data.seek(0) ← Use the "seek()" method to return to the start of the file.  
0 And yes, you can use "tell()" with Python's files, too.  
>>> for each_line in data:  
    print(each_line, end='') } ← This code should look familiar: it's a standard iteration using the file's data as input.
```

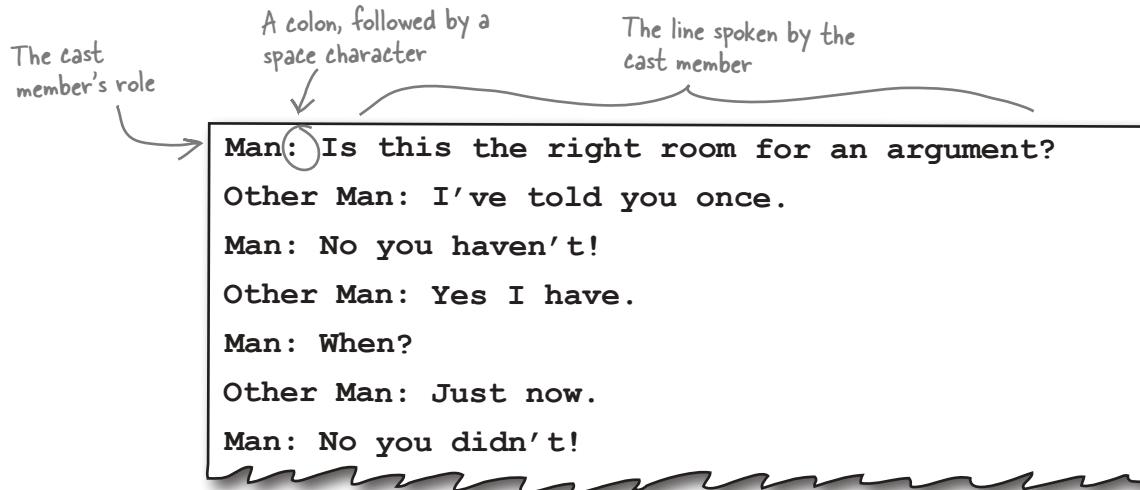
```
Man: Is this the right room for an argument?  
Other Man: I've told you once.  
Man: No you haven't!  
Other Man: Yes I have.  
Man: When?  
Other Man: Just now.  
Man: No you didn't!  
...  
Man: (exasperated) Oh, this is futile!!  
(pause)
```

Every line of the data is displayed on screen (although for space reasons, it is abridged here).

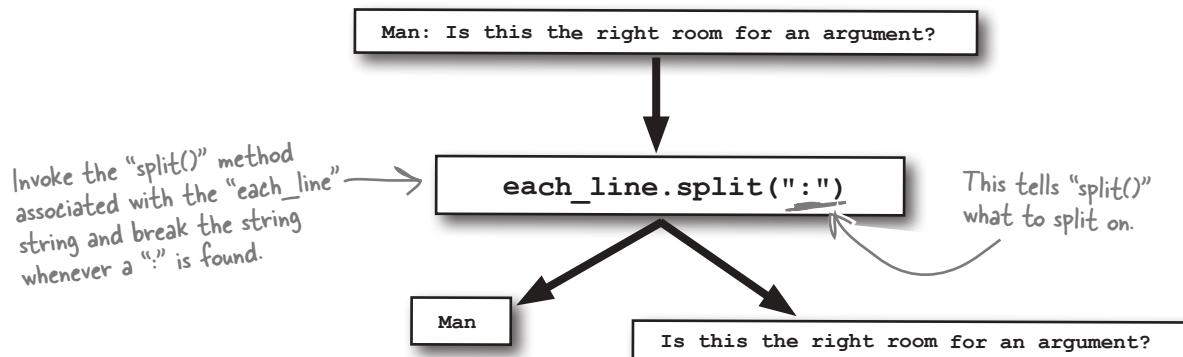
```
Other Man: No it isn't!  
Man: Yes it is!  
>>> data.close() ← Since you are now done with the file, be sure to close it.
```

Take a closer look at the data

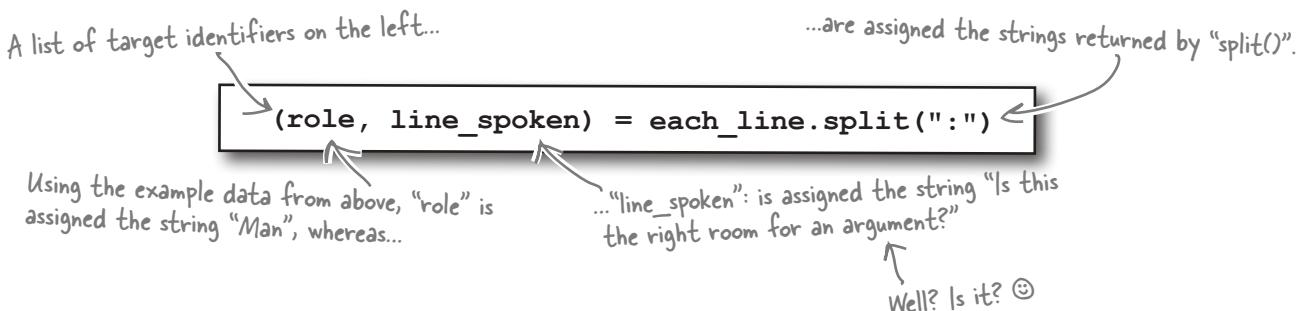
Look closely at the data. It appears to conform to a specific format:



With this format in mind, you can process each line to *extract* parts of the line as required. The `split()` method can help here:



The `split()` method returns a list of strings, which are assigned to a list of target identifiers. This is known as *multiple assignment*:





An IDLE Session

Let's confirm that you can still process your file while splitting each line. Type the following code into IDLE's shell:

```
>>> data = open('sketch.txt') <-- Open the data file.
```

```
>>> for each_line in data:
```

```
    (role, line_spoken) = each_line.split(':')
    print(role, end='')
    print(' said: ', end='')
    print(line_spoken, end='')
```

Process the data, extracting each part from each line and displaying each part on screen.

Man said: Is this the right room for an argument?

Other Man said: I've told you once.

Man said: No you haven't!

Other Man said: Yes I have.

Man said: When?

Other Man said: Just now.

Man said: No you didn't!

Other Man said: Yes I did!

Man said: You didn't!

Other Man said: I'm telling you, I did!

Man said: You did not!

Other Man said: Oh I'm sorry, is this a five minute argument, or the full half hour?

Man said: Ah! (taking out his wallet and paying) Just the five minutes.

Other Man said: Just the five minutes. Thank you.

Other Man said: Anyway, I did.

Man said: You most certainly did not!

Traceback (most recent call last):

```
  File "<pyshell#10>", line 2, in <module>
    (role, line_spoken) = each_line.split(':')
```

```
ValueError: too many values to unpack
```

This all looks OK.

Whoops! There's something seriously wrong here.



It's a ValueError, so that must mean there's something wrong with the data in your file, right?

Know your data

Your code worked fine for a while, then crashed with a *runtime error*. The problem occurred right after the line of data that had the *Man* saying, “You most certainly did not!”

Let’s look at the data file and see what comes after this successfully processed line:

```

Man: You didn't!
Other Man: I'm telling you, I did!
Man: You did not!
Other Man: Oh I'm sorry, is this a five minute argument, or the full half hour?
Man: Ah! (taking out his wallet and paying) Just the five minutes.
Other Man: Just the five minutes. Thank you.
Other Man: Anyway, I did.
Man: You most certainly did not! ← The error occurs AFTER this line
      of data.
Other Man: Now let's get one thing quite clear: I most definitely told you!
Man: Oh no you didn't!
Other Man: Oh yes I did!

```

Notice anything?

Notice anything about the *next* line of data?

The next line of data has two colons, not one. This is enough extra data to upset the `split()` method due to the fact that, as your code currently stands, `split()` expects to break the line into two parts, assigning each to `role` and `line_spoken`, respectively.

When an extra colon appears in the data, the `split()` method breaks the line into *three parts*. Your code hasn’t told `split()` what to do with the third part, so the Python interpreter *raises* a `ValueError`, complains that you have “too many values,” and terminates. A **runtime error** has occurred.



What approach might you take to solve this data-processing problem?

Do this!

To help diagnose this problem, let’s put your code into its own file called `sketch.py`. You can copy and paste your code from the IDLE shell into a new IDLE edit window.

Know your methods and ask for help

It might be useful to see if the `split()` method includes any functionality that might help here. You can ask the IDLE shell to tell you more about the `split()` method by using the `help()` BIF.

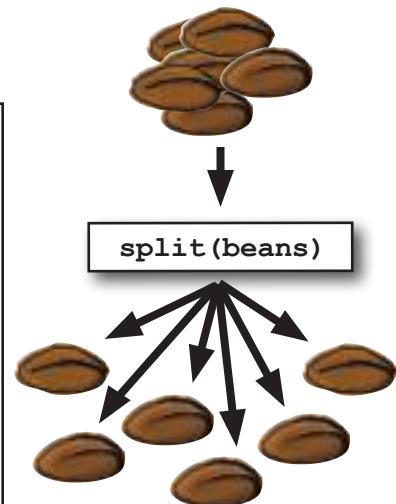
 An IDLE Session

```
>>> help(each_line.split)
Help on built-in function split:

split(...)
    S.split([sep[, maxsplit]]) -> list of strings

    Return a list of the words in S, using sep as the
    delimiter string. If maxsplit is given, at most maxsplit
    splits are done. If sep is not specified or is None, any
    whitespace string is a separator and empty strings are
    removed from the result.
```

Looks like "split()" takes an optional argument.



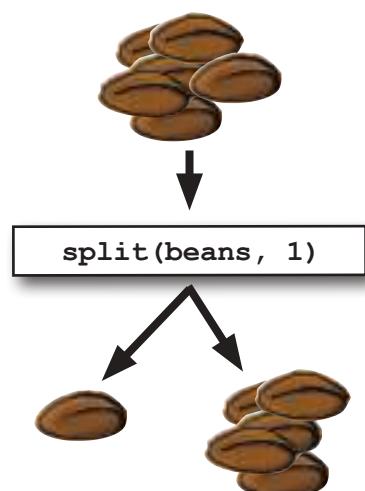
The optional argument to `split()` controls how many breaks occur within your line of data. By default, the data is broken into as many parts as is possible. But you need only two parts: the name of the character and the line he spoke.

If you set this optional argument to 1, your line of data is only ever broken into two pieces, effectively negating the effect of any extra colon on any line.

Let's try this and see what happens.

 Geek Bits

IDLE gives you searchable access to the entire Python documentation set via its Help → Python Docs menu option (which will open the docs in your web browser). If all you need to see is the documentation associated with a single method or function, use the `help()` BIF within IDLE's shell.





An IDLE Session

Here's the code in the IDLE edit window. Note the extra argument to the split() method.

```
sketch.py - /Users/barryp/HeadFirstPython/chapter4/sketch.py
data = open('sketch.txt')

for each_line in data:
    (role, line_spoken) = each_line.split(':', 1)
    print(role, end='')
    print(' said: ', end='')
    print(line_spoken, end='')

data.close()

Ln: 11 Col: 0
```

With the edit applied and saved, press F5 (or select **Run Module** from IDLE's **Run** menu) to try out this version of your code:

```
>>> ===== RESTART =====
>>>
Man said: Is this the right room for an argument?
Other Man said: I've told you once.
Man said: No you haven't!
Other Man said: Yes I have.
Man said: When?
Other Man said: Just now.
...
Other Man said: Anyway, I did.
Man said: You most certainly did not!
Other Man said: Now let's get one thing quite clear: I most definitely told you!
Man said: Oh no you didn't!
Other Man said: Oh yes I did!
Man said: Oh no you didn't!
Other Man said: Oh yes I did!
Man said: Oh look, this isn't an argument!
```

The displayed output is abridged to allow the important stuff to fit on this page.

Cool. You made it past the line with two colons...

...but your joy is short lived. There's ANOTHER ValueError!!

```
Traceback (most recent call last):
  File "/Users/barryp/HeadFirstPython/chapter4/sketch.py", line 5, in <module>
    (role, line_spoken) = each_line.split(':', 1)
ValueError: need more than 1 value to unpack
```

That's enough to ruin your day. What could be wrong now?

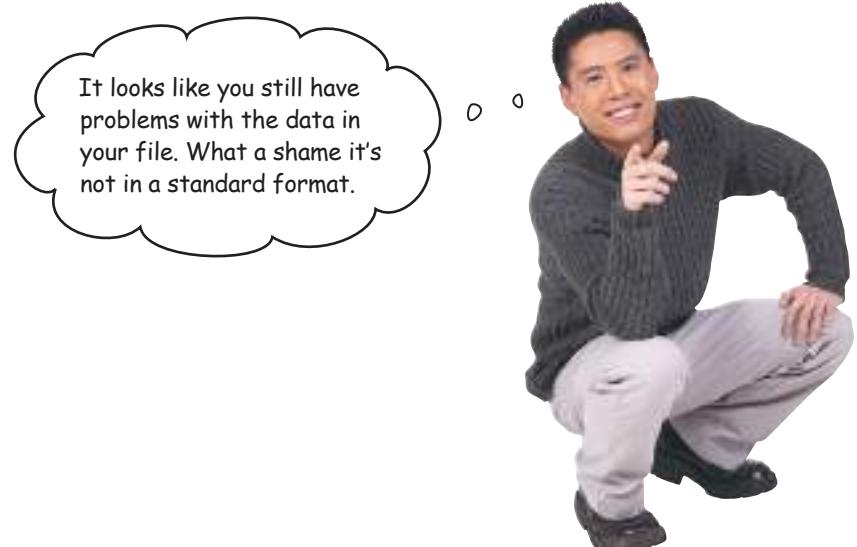
Know your data (better)

Your code has raised another `ValueError`, but this time, instead of complaining that there are “too many values,” the Python interpreter is complaining that it doesn’t have enough data to work with: “need more than 1 value to unpack.” Hopefully, another quick look at the data will clear up the mystery of the missing data.

```
Other Man: Now let's get one thing quite clear: I most definitely told you!
Man: Oh no you didn't!
Other Man: Oh yes I did!
Man: Oh no you didn't!
Other Man: Oh yes I did!
Man: Oh look, this isn't an argument!
(pause) ←
Other Man: Yes it is!
Man: No it isn't! → What's this?!? Some of the data doesn't conform to
(pause) ← the expected format...which can't be good.
Man: It's just contradiction!
Other Man: No it isn't!
```

The case of the missing colon

Some of the lines of data contain no colon, which causes a problem when the `split()` method goes looking for it. The lack of a colon prevents `split()` from doing its job, causes the runtime error, which then results in the complaint that the interpreter needs “more than 1 value.”



Two very different approaches



Jill's suggested approach certainly works: add the extra logic required to work out whether it's worth invoking `split()` on the line of data. All you need to do is work out how to check the line of data.

Joe's approach works, too: let the error occur, spot that it has happened, and then recover from the runtime error...somehow.

Which approach works best here?

Add extra logic

Let's try *each* approach, then decide which works best here.

In addition to the `split()` method, every Python string has the `find()` method, too. You can ask `find()` to try and locate a substring in another string, and if it can't be found, the `find()` method returns the value `-1`. If the method locates the substring, `find()` returns the index position of the substring in the string.



An IDLE Session

Assign a string to the `each_line` variable that does not contain a colon, and then use the `find()` method to try and locate a colon:

```
>>> each_line = "I tell you, there's no such thing as a flying circus."  
>>> each_line.find(':')  
-1 ← The string does NOT contain a colon, so "find()" returns -1 for NOT FOUND.
```

Press **Alt-P** twice to recall the line of code that assigns the string to the variable, but this time edit the string to include a colon, then use the `find()` method to try to locate the colon:

```
>>> each_line = "I tell you: there's no such thing as a flying circus."  
>>> each_line.find(':')  
10 ← The string DOES contain a colon, so "find()" returns a positive index value.
```

And you thought this approach wouldn't work? Based on this IDLE session, I think this could do the trick.





Exercise

Adjust your code to use the extra logic technique demonstrated on the previous page to deal with lines that don't contain a colon character.

```
data = open('sketch.txt')  
  
for each_line in data:  
    if .....  
        (role, line_spoken) = each_line.split(':', 1)  
        print(role, end='')  
        print(' said: ', end='')  
        print(line_spoken, end='')  
  
data.close()
```

What condition needs to go here?



Sharpen your pencil

Can you think of any potential problems with this technique?
Grab your pencil and write down any issues you might have with
this approach in the space provided below:



Exercise Solution

You were to adjust your code to use the extra logic technique to deal with lines that don't contain a colon character:

```
data = open('sketch.txt')

for each_line in data:
    if not each_line.find(':') == -1:
        (role, line_spoken) = each_line.split(':', 1)
        print(role, end='')
        print(' said: ', end='')
        print(line_spoken, end='')

data.close()
```

Note the use of the "not" keyword, which negates the value of the condition.

It takes a few seconds to get your head around this condition, but it does work.



It's OK if your issues are different. Just so long as they are similar to these.

You were to think of any potential problems with this technique, grabbing your pencil to write down any issues you might have with this approach.

There might be a problem with this code if the format of the data file changes, which will require changes to the condition.

The condition used by the if statement is somewhat hard to read and understand.

This code is a little "fragile"...it will break if another exceptional situation arises.



Test DRIVE

Amend your code within IDLE's edit window, and press F5 to see if it works.

```

Python Shell
>>> ===== RESTART =====
>>>
Man said: Is this the right room for an argument?
Other Man said: I've told you once.
Man said: No you haven't!
Other Man said: Yes I have.
Man said: When?
Other Man said: Just now.
Man said: No you didn't!
Other Man said: Yes I did!
Man said: You didn't!
Other Man said: I'm telling you, I did!
Man said: You did not!
Other Man said: Oh I'm sorry, is this a five minute argument, or the full half hour?
Man said: Ah! (taking out his wallet and paying) Just the five minutes.
Other Man said: Just the five minutes. Thank you.
Other Man said: Anyway, I did.
Man said: You most certainly did not!
Other Man said: Now let's get one thing quite clear: I most definitely told you!
Man said: Oh no you didn't!
Other Man said: Oh yes I did!
Man said: Oh no you didn't!
Other Man said: Oh yes I did!
Man said: Oh look, this isn't an argument!
Other Man said: Yes it is!
Man said: No it isn't!
Man said: It's just contradiction!
Other Man said: No it isn't!
Man said: It IS!
Other Man said: It is NOT!
Man said: You just contradicted me!
Other Man said: No I didn't!
Man said: You DID!
Other Man said: No no no!
Man said: You did just then!
Other Man said: Nonsense!
Man said: (exasperated) Oh, this is futile!!
Other Man said: No it isn't!
Man said: Yes it is!
>>> |
```

Ln: 149 Col: 4

No errors
this time.

Your program works...although it is **fragile**.

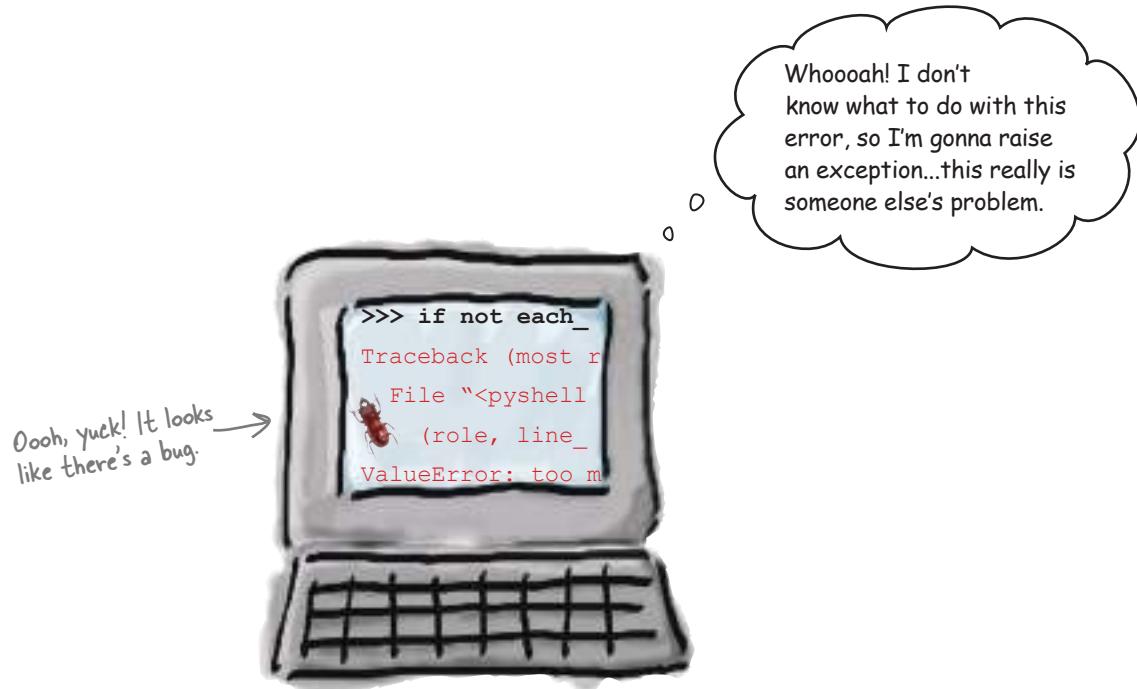
If the format of the file changes, your code will need to change, too, and *more code* generally means *more complexity*. Adding extra logic to handle exceptional situations works, but it might cost you in the long run.



Handle exceptions

Have you noticed that when something goes wrong with your code, the Python interpreter displays a *traceback* followed by an error message?

The **traceback** is Python's way of telling you that something *unexpected* has occurred during runtime. In the Python world, runtime errors are called **exceptions**.



Of course, if you decide to *ignore* an exception when it occurs, your program crashes and burns.

But here's the skinny: Python lets you *catch* exceptions as they occur, which gives you with a chance to possibly recover from the error and, critically, **not** crash.

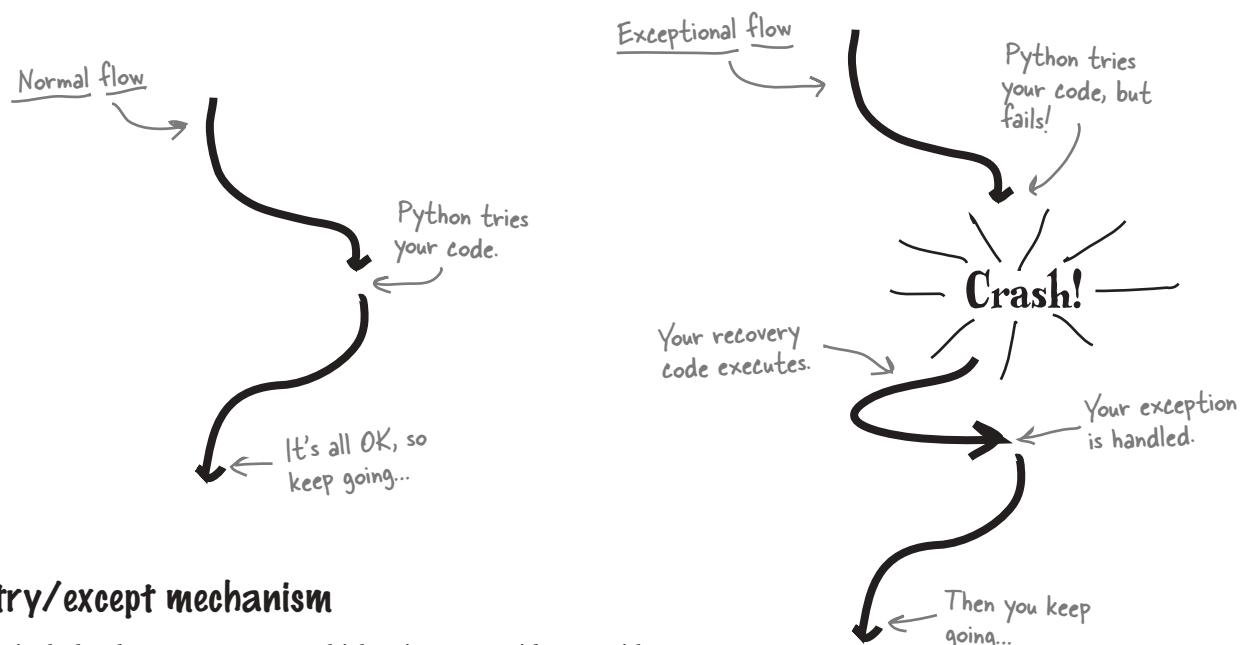
By controlling the runtime behavior of your program, you can ensure (as much as possible) that your Python programs are robust in the face of *most* runtime errors.

Try the code first. Then deal with errors as they happen.

Try first, then recover

Rather than adding extra code and logic to guard against bad things happening, Python's **exception handling** mechanism lets the error occur, spots that it has happened, and then gives you an opportunity to recover.

During the *normal flow of control*, Python tries your code and, if nothing goes wrong, your code continues as normal. During the *exceptional flow of control*, Python tries your code only to have something go wrong, your recovery code executes, and then your code continues as normal.



The try/except mechanism

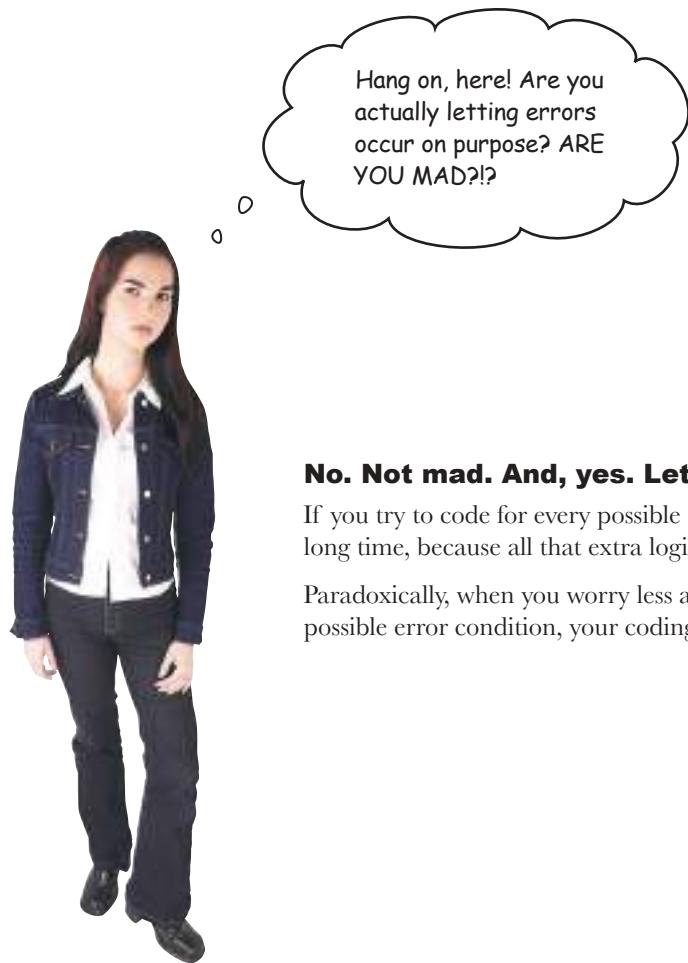
Python includes the **try** statement, which exists to provide you with a way to systematically handle exceptions and errors at runtime. The general form of the **try** statement looks like this:

try:
Both "try" and "except" are Python keywords.

your code (which might cause a runtime error)

except:

your error-recovery code



No. Not mad. And, yes. Letting errors occur.

If you try to code for every possible error, you'll be at it for a long time, because all that extra logic takes a while to work out.

Paradoxically, when you worry less about covering *every* possible error condition, your coding task actually gets *easier*.

Identify the code to protect

In order to plug into the Python exception handling mechanism, take a moment to identify the code that you need to *protect*.



Study your program and circle the line or lines of code that you think you need to protect. Then, in the space provided, state why.

```
data = open('sketch.txt')

for each_line in data:
    (role, line_spoken) = each_line.split(':', 1)
    print(role, end='')
    print(' said: ', end='')
    print(line_spoken, end='')

data.close()
```

State your reason
why here. ↗

.....
.....
.....
.....
.....

there are no
Dumb Questions

Q: Something has been bugging me for a while. When the `split()` method executes, it passes back a list, but the target identifiers are enclosed in regular brackets, not square brackets, so how is this a list?

A: Well spotted. It turns out that there are **two** types of list in Python: those that can change (enclosed in square brackets) and those that cannot be changed once they have been created (enclosed in regular brackets). The latter is an *immutable* list, more commonly referred to as a *tuple*. Think of tuples as the same as a list, except for one thing: once created, the data they hold **cannot** be changed under any circumstances. Another way to think about tuples is to consider them to be a *constant list*. At Head First, we pronounce “tuple” to rhyme with “couple.” Others pronounce “tuple” to rhyme with “rupal.” There is no clear consensus as to which is correct, so pick one and stick to it.



You were to study your program and circle the line or lines of code that you think you need to protect. Then, in the space provided, you were to state why.

```
data = open('sketch.txt')

for each_line in data:
    (role, line_spoken) = each_line.split(':', 1)
    print(role, end='')
    print(' said: ', end='')
    print(line_spoken, end='')

data.close()
```

These four
lines of code
all need to be
protected.

If the call to "split()" fails, you don't want the three "print()" statements executing, so it's best to protect all four lines of the "if" suite, not just the line of code that calls "split()".

OK. I get that the code can be protected from an error. But what do I do when an error actually occurs?

Yeah...good point. It's probably best to ignore it, right? I wonder how...

Take a pass on the error

With this data (and this program), it is best if you ignore lines that don't conform to the expected format. If the call to the `split()` method causes an exception, let's simply **pass** on reporting it as an error.

When you have a situation where you might be expected to provide code, but don't need to, use Python's `pass` statement (which you can think of as the *empty* or *null* statement.)

Here's the `pass` statement combined with `try`:

```
data = open('sketch.txt')

for each_line in data:
    try:
        (role, line_spoken) = each_line.split(':', 1)
        print(role, end='')
        print(' said: ', end='')
        print(line_spoken, end='')

    except:
        pass

data.close()
```

This code is protected from runtime errors.

If a runtime error occurs, this code is executed.

Now, no matter what happens when the `split()` method is invoked, the `try` statement *catches* any and all exceptions and *handles* them by ignoring the error with `pass`.

Let's see this code in action.



Make the required changes to your code in the IDLE edit window.



An IDLE Session

With your code in the IDLE edit window, press F5 to run it.

```
sketch-try.py - /Users/bairyp/HeadFirstPython/chapter4/sketch-try.py
data = open('sketch.txt')
for each_line in data:
    try:
        (role, line_spoken) = each_line.split(':', 1)
        print(role, end=' ')
        print(' said: ', end=' ')
        print(line_spoken, end=' ')
    except:
        pass
data.close()

Ln: 14 Col: 0
```

```
>>> ===== RESTART =====
>>>
Man said: Is this the right room for an argument?
Other Man said: I've told you once.
Man said: No you haven't!
Other Man said: Yes I have.
Man said: When?
...
OOther Man said: Nonsense!
Man said: (exasperated) Oh, this is futile!!
Other Man said: No it isn't!
Man said: Yes it is!
```

This code works, and there are no runtime errors, either.



So...both approaches work.
But which is better?

Fireside Chats



Tonight's talk: **Approaching runtime errors with extra code and exception handlers**

Extra Code:

By making sure runtime errors never happen, I keep my code safe from tracebacks.

Complexity never hurt anyone.

I just don't get it. You're more than happy for your code to explode in your face...then you decide it's probably a good idea to put out the fire?!?

But the bad things *still* happen to you. They never happen with me, because I don't let them.

Well...that depends. If you're smart enough—and, believe me, I am—you can think up all the possible runtime problems and code around them.

Hard work never hurt anyone.

Of course all my code is needed! How else can you code around all the runtime errors that are going to happen?

Um, uh...most of them, I guess.

Look: just cut it out. OK?

Exception Handler:

At the cost of added complexity....

I'll be sure to remind you of that the next time you're debugging a complex piece of code at 4 o'clock in the morning.

Yes. I concentrate on getting my work done first and foremost. If bad things happen, I'm ready for them.

Until something else happens that you weren't expecting. Then you're toast.

Sounds like a whole heap of extra work to me.

You did hear me earlier about debugging at 4 AM, right? Sometimes I think you actually enjoy writing code that you don't need...

Yeah...how many?

You don't know, do you? You've no idea what will happen when an unknown or unexpected runtime error occurs, do you?

What about other errors?

It is true that both approaches work, but let's consider what happens when other errors surface.



Handling missing files

Frank's posed an interesting question and, sure enough, the problem caused by the removal of the data file makes life more complex for Jill and Joe. When the data file is missing, *both* versions of the program crash with an `IOError`.

★ ★ ★
Do this! →

Rename the data file, then run both versions of your program again to confirm that they do indeed raise an `IOError` and generate a traceback.

Add more error-checking code...

If you're a fan of the “let's not let errors happen” school of thinking, your first reaction will be to *add extra code* to check to see if the data file exists before you try to open it, right?

Let's implement this strategy. Python's `os` module has some facilities that can help determine whether a data file exists, so we need to import it from the Standard Library, then add the required check to the code:

```
import os

Check whether
the file exists. → if os.path.exists('sketch.txt'):
    data = open('sketch.txt')

    for each_line in data:
        if not each_line.find(':') == -1:
            (role, line_spoken) = each_line.split(':', 1)
            print(role, end='')
            print(' said: ', end='')
            print(line_spoken, end='')

    data.close()

All of this
code remains
unchanged.   else:
    print('The data file is missing!')
```

Inform the user of the bad news. →



An IDLE Session

A quick test of the code confirms that this new problem is dealt with properly. With this new version of your code in IDLE's edit window, press F5 to confirm all is OK.

```
>>> ----- RESTART -----
>>>
The data file is missing! ← Exactly what was expected. Cool.
>>>
```

...Or add another level of exception handling

If you are a fan of the “handle exceptions as they occur” school of thinking, you’ll simply wrap your code within another **try** statement.

Add another “try” statement.

```
try:
    data = open('sketch.txt')

    for each_line in data:
        try:
            (role, line_spoken) = each_line.split(':', 1)
            print(role, end='')
            print(' said: ', end='')
            print(line_spoken, end='')

        except:
            pass

    data.close()

except:
    print('The data file is missing!')
```

As with the other program, all of this code remains unchanged.

Give the user the bad news.



An IDLE Session

Another quick test is required, this time with the version of your program that uses exception handling. Press F5 to give it a spin.

```
>>> ===== RESTART =====
>>>
The data file is missing!
```

As expected, this version of the program handles the missing file, too.

So, which approach is best?

Well...it depends on who you ask! Here are both versions of your code:

```
import os
if os.path.exists('sketch.txt'):
    data = open('sketch.txt')
    for each_line in data:
        if not each_line.find(':') == -1:
            (role, line_spoken) = each_line.split(':', 1)
            print(role, end=' ')
            print(' said ', end='')
            print(line_spoken, end=' ')
    data.close()
else:
    print('The datafile is missing!')
```

This version uses extra logic
to handle File I/O errors.

```
try:
    data = open('sketch.txt')
    for each_line in data:
        try:
            (role, line_spoken) = each_line.split(':', 1)
            print(role, end=' ')
            print(' said ', end='')
            print(line_spoken, end=' ')
        except:
            pass
    data.close()
except:
    print('The datafile is missing!')
```

This version uses another "try"
statement to handle File I/O errors.

Let's ask a simple question about these two versions of your program: *What do each of these programs do?*



Sharpen your pencil

Grab your pencil. In box 1, write down what you think the program on the left does. In box 2, write down what you think the program on the right does.

1

.....
.....
.....

2

.....
.....
.....



There's a lot to write, so you actually need more space for your description than was provided on the previous page.

You were to grab your pencil, then in box 1, write down what you thought the program on the left does. In box 2, write down what you thought the program on the right does.

1

The code on the right starts by importing the “os” library, and then it uses “path.exists” to make sure the data file exists, before it attempts to open the data file. Each line from the file is then processed, but only after it has determined that the line conforms to the required format by checking first for a single “?” character in the line. If the “?” is found, the line is processed; otherwise, it’s ignored. When we’re all done, the data file is closed. And you get a friendly message at the end if the file is not found.

Now...that's more like it.

2

The code on the right opens a data file, processes each line in that file, extracts the data of interest and displays it on screen. The file is closed when done. If any exceptions occur, this code handles them.

Complexity is rarely a good thing

Do you see what's happening here?

As the list of errors that you have to worry about grows, the **complexity** of the “add extra code and logic” solution increases to the point where it starts to *obscure the actual purpose of the program*.

This is not the case with the exceptions handling solution, in which it's *obvious* what the main purpose of the program is.

By using Python's exception-handling mechanism, you get to concentrate on what your code *needs to do*, as opposed to worrying about what can go wrong and writing extra code to avoid runtime errors.

Prudent use of the `try` statement leads to code that is easier to read, easier to write, and—perhaps most important—easier to fix when something goes wrong.

Concentrate on what your code needs to do.

You're done...except for one small thing

Your exception-handling code is good. In fact, your code might be *too good* in that it is *too general*.

At the moment, no matter what error occurs at runtime, it is handled by your code because it's *ignored* or a error message is *displayed*. But you really need to worry only about `IOErrors` and `ValueErrors`, because those are the types of exceptions that occurred earlier when you were developing your program.

Although it is great to be able to handle all runtime errors, it's probably unwise to be too generic...you will want to know if something other than an `IOError` or `ValueError` occurs as a result of your code executing at runtime. If something else does happen, your code might be handling it in an *inappropriate* way.

```
try:
    data = open('sketch.txt')

    for each_line in data:
        try:
            (role, line_spoken) = each_line.split(':', 1)
            print(role, end='')
            print(' said: ', end='')
            print(line_spoken, end='')

        except:
            pass

    data.close()

except:
    print('The data file is missing!')
```

The diagram shows a block of Python code with several handwritten annotations:

- A large bracket underlines the first `try:` block, and another bracket underlines the inner `try:` block within it.
- An arrow points from the `pass` statement in the inner `except:` block to a handwritten note: "This code and this code runs when ANY runtime error occurs within the code that is being tried."
- Another arrow points from the outer `except:` block to the same handwritten note.

As your code is currently written, it is too generic. Any runtime error that occurs is handled by one of the `except` suites. This is unlikely to be what you want, because this code has the potential to *silently ignore runtime errors*.

You need to somehow use `except` in a less generic way.

Be specific with your exceptions

If your exception-handling code is designed to deal with a specific type of error, be sure to specify the error type on the `except` line. In doing so, you'll take your exception handling code from *generic* to *specific*.

```
try:  
    data = open('sketch.txt')  
  
    for each_line in data:  
        try:  
            (role, line_spoken) = each_line.split(':', 1)  
            print(role, end='')  
            print(' said: ', end='')  
            print(line_spoken, end='')  
        except ValueError:  
            pass  
        finally:  
            data.close()  
    except IOError:  
        print('The data file is missing!')
```

Specify the type of runtime error you are handling.

Of course, if an *different* type of runtime error occurs, it is no longer handled by your code, but at least now you'll get to hear about it. When you are specific about the runtime errors your code handles, your programs no longer silently ignore some runtime errors.





Your Python Toolbox

You've got Chapter 3 under your belt and you've added some key Python techniques to your toolbox.

Python Lingo

- An “exception” occurs as a result of a runtime error, producing a traceback.
- A “traceback” is a detailed description of the runtime error that has occurred.

IDLE Notes

- Access Python's documentation by choosing Python Docs from IDLE's Help menu. The Python 3 documentation set should open in your favorite web browser.



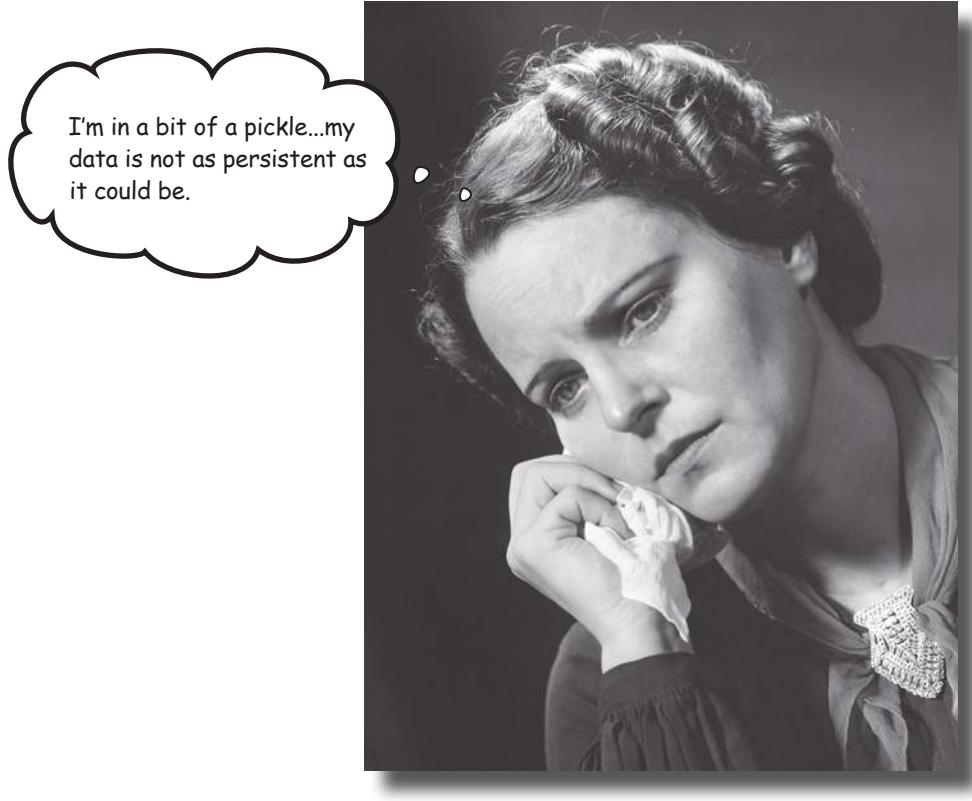
BULLET POINTS

- Use the `open()` BIF to open a disk file, creating an iterator that reads data from the file one line at a time.
- The `readline()` method reads a single line from an opened file.
- The `seek()` method can be used to “rewind” a file to the beginning.
- The `close()` method closes a previously opened file.
- The `split()` method can break a string into a list of parts.
- An unchangeable, constant list in Python is called a **tuple**. Once list data is assigned to a tuple, it cannot be changed. Tuples are *immutable*.
- A `ValueError` occurs when your data does not conform to an expected format.
- An `IOError` occurs when your data cannot be accessed properly (e.g., perhaps your data file has been moved or renamed).
- The `help()` BIF provides access to Python's documentation within the IDLE shell.
- The `find()` method locates a specific substring within another string.
- The `not` keyword negates a condition.
- The `try/except` statement provides an exception-handling mechanism, allowing you to protect lines of code that might result in a runtime error.
- The `pass` statement is Python's empty or null statement; it does nothing.

4 persistence



* Saving data to files *

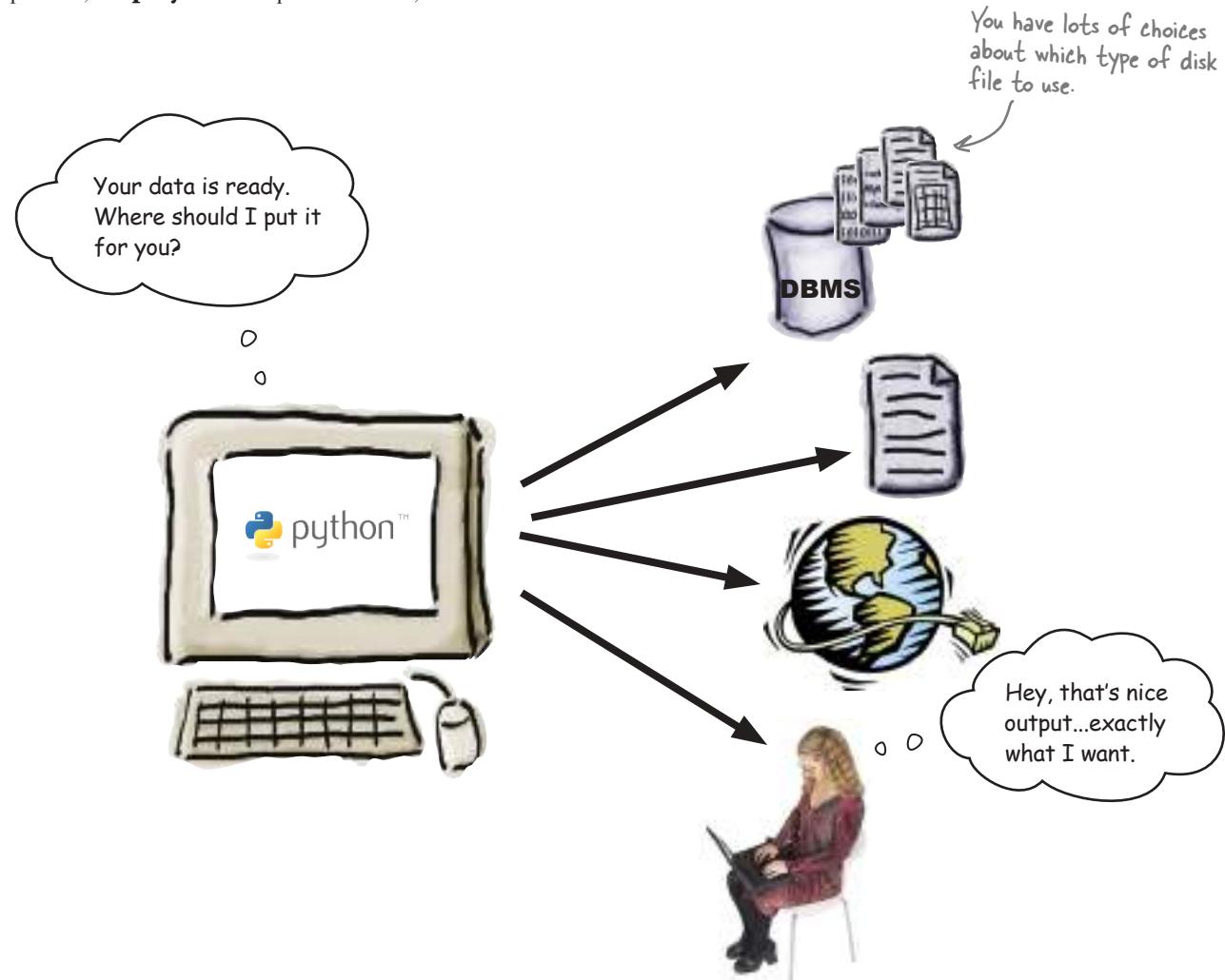


It is truly great to be able to process your file-based data.

But what happens to your data when you're done? Of course, it's best to save your data to a disk file, which allows you to use it again at some later date and time. Taking your memory-based data and storing it to disk is what **persistence** is all about. Python supports all the usual tools for writing to files and also provides some cool facilities for *efficiently* storing Python data. So...flip the page and let's get started learning them.

Programs produce data

It's a rare program that reads data from a disk file, processes the data, and then throws away the processed data. Typically, programs **save** the data they process, **display** their output on screen, or **transfer** data over a network.



Before you learn what's involved in writing data to disk, let's process the data from the previous chapter to work out who said what to whom.

When that's done, you'll have something worth saving.



Code Magnets

Add the code magnets at the bottom of this page to your existing code to satisfy the following requirements:

1. Create an empty list called `man`.
 2. Create an empty list called `other`.
 3. Add a line of code to remove unwanted whitespace from the `line_spoken` variable.
 4. Provide the conditions and code to add `line_spoken` to the correct list based on the value of `role`.
 5. Print each of the lists (`man` and `other`) to the screen.

Here are your magnets.

```
if role == 'Man':  
    line_spoken = line_spoken.strip()  
    print(other)  
    man.append(line_spoken)  
else:  
    man = []  
    other = []  
    print(man)  
    if role == 'Other Man':  
        other.append(line_spoken)
```



Code Magnets Solution

You were to add the code magnets to your existing code to satisfy the following requirements:

1. Create an empty list called `man`.
2. Create an empty list called `other`.
3. Add a line of code to remove unwanted whitespace from the `line_spoken` variable.
4. Provide the conditions and code to add `line_spoken` to the correct list based on the value of `role`.
5. Print each of the lists (`man` and `other`) to the screen.

```

man = []
other = []

try:
    data = open('sketch.txt')
    for each_line in data:
        try:
            (role, line_spoken) = each_line.split(':', 1)
            line_spoken = line_spoken.strip()
            if role == 'Man':
                man.append(line_spoken)
            elif role == 'Other Man':
                other.append(line_spoken)
        except ValueError:
            pass
    data.close()
except IOError:
    print('The datafile is missing!')

print(man)
print(other)

```

Assign an empty list to "man" and "other".

Assign the stripped string back onto itself.

"elif" means "else if."

The "strip()" method removes unwanted whitespace from a string.

Update one of the lists based on who said what.

Conclude by displaying the processed data on screen.



Test DRIVE

Load your code into IDLE's edit window and take it for a spin by pressing F5. Be sure to save your program into the same folder that contains `sketch.txt`.

The code in IDLE's edit window

```

man = []
other = []

try:
    data = open('sketch.txt')
    for each_line in data:
        try:
            (role, line_spoken) = each_line.split(':', 1)
            line_spoken = line_spoken.strip()
            if role == 'Man':
                man.append(line_spoken)
            elif role == 'Other Man':
                other.append(line_spoken)
            except ValueError:
                pass
    data.close()
except IOError:
    print('The datafile is missing!')
print(man)
print(other)

```

Python Shell

```

Python 3.1.2 (r312:79360M, Mar 26 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> =====
RESTART =====
>>>
['Is this the right room for an argument?', "No you haven't!", 'When?', "No yo
u didn't!", "You didn't!", 'You did not!', 'Ah! (taking out his wallet and pay
ing) Just the five minutes.', "You most certainly did not!", "Oh no you didn't
!","Oh no you didn't!", "Oh look, this isn't an argument!", "So it isn't!", "It's
just contradiction!", "It is!", 'You just contradicted me!', 'You DID!', 'You
did just then!', '{exasperated} Oh, this is futile!', 'Tee it is!']
["I've told you once.", 'Tee I have!', 'Just now.', 'Tee I did!', "I'm telling
you, I did!", "Oh I'm sorry, is this a five minute argument, or the full half
hour?", 'Just the five minutes. Thank you.', 'Anyway, I did.', 'Now let's get
one thing quite clear: I most definitely told you!', 'Oh yes I did!', 'Oh yes
I did!', 'Tee it is!', 'No it isn't!', 'It is NOT!', 'No I didn't!', 'No no no
!', 'Nonsense!', 'No it isn't!']
>>>

```

Ln 8 Col. 4

And here's what appears on screen: the contents of the two lists.

It worked, as expected.



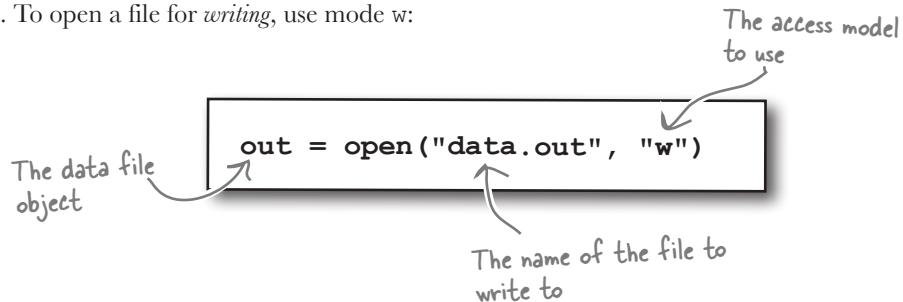
Surely Python's `open()` BIF can open files for writing as well as reading, eh?

Yes, it can.

When you need to **save** data to a file, the `open()` BIF is all you need.

Open your file in write mode

When you use the `open()` BIF to work with a disk file, you can specify an *access mode* to use. By default, `open()` uses mode `r` for *reading*, so you don't need to specify it. To open a file for *writing*, use mode `w`:



By default, the `print()` BIF uses standard output (usually the screen) when displaying data. To write data to a file instead, use the `file` argument to specify the data file object to use:

```
print("Norwegian Blues stun easily.", file=out)
```

What gets written to the file

The name of the data file object to write to

When you're done, be sure to close the file to ensure all of your data is written to disk. This is known as **flushing** and is very important:

```
out.close()
```

This is VERY important when writing to files.



Geek Bits

When you use access mode `w`, Python opens your named file for **writing**. If the file already exists, it is *cleared of its contents*, or *clobbered*. To **append** to a file, use access mode `a`, and to open a file for writing and reading (without clobbering), use `w+`. If you try to open a file for writing that does not already exist, it is first created for you, and then opened for writing.



At the bottom of your program, two calls to the `print()` BIF display your processed data on screen. Let's amend this code to save the data to two disk files instead.

Call your disk files `man_data.txt` (for what the man said) and `other_data.txt` (for what the other man said). Be sure to both **open** and **close** your data files, as well as protect your code against an `IOError` using **try/except**.

```
man = []
other = []

try:
    data = open('sketch.txt')
    for each_line in data:
        try:
            (role, line_spoken) = each_line.split(':', 1)
            line_spoken = line_spoken.strip()
            if role == 'Man':
                man.append(line_spoken)
            elif role == 'Other Man':
                other.append(line_spoken)
        except ValueError:
            pass
    data.close()
except IOError:
    print('The datafile is missing!')
```

Go on, try.

Be sure to
close your
files.

```
print(man, ..... )
print(other, ..... ) }
```

Specify the files to
write to when you invoke
"print()".

Open your two
data files here.

Handle any
exceptions here.



At the bottom of your program, two calls to the `print()` BIF display your processed data on screen. You were to amend this code to save the data to two disk files instead.

You were to call your disk files `man_data.txt` (for what the man said) and `other_data.txt` (for what the other man said). You were to make sure to both **open** and **close** your data files, as well as protect your code against an `IOError` using **try/except**.

```

man = []
other = []

try:
    data = open('sketch.txt')
    for each_line in data:
        try:
            (role, line_spoken) = each_line.split(':', 1)
            line_spoken = line_spoken.strip()
            if role == 'Man':
                man.append(line_spoken)
            elif role == 'Other Man':
                other.append(line_spoken)
        except ValueError:
            pass
    data.close()
except IOError:
    print('The datafile is missing!')

try:
    man_file = open('man_data.txt', 'w')
    other_file = open('other_data.txt', 'w')
    print(man, file=man_file)
    print(other, file=other_file)
    man_file.close()
    other_file.close()
except IOError:
    print('File error.')

```

All of this code is unchanged.

Did you remember to open your files in WRITE mode?

Open your two files, and assign each to file objects.

Use the "print()" BIF to save the named lists to named disk files.

Don't forget to close BOTH files.

Handle an I/O exception, should one occur.



Test DRIVE

Perform the edits to your code to replace your two `print()` calls with your new file I/O code. Then, run your program to confirm that the data files are created:

The top screenshot shows a desktop environment with a file browser window titled "Chapter4". Inside the browser, under "Devices" (MacBook HD) and "Places" (Applications, Desktop, Boards, Computer, Devintosh), there are two files: "sketch.txt" and "sketch4.py". Below the browser is a "Python Shell" window with the following text:

```
Python 3.1.2 (r312:79360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>>
```

A handwritten note on the left side of this screenshot says: "Before your program runs, there are no data files in your folder, just your code."

The bottom screenshot shows the same desktop environment after the program has run. The "sketch.txt" and "sketch4.py" files remain, but now there are two additional files: "man_data.txt" and "other_data.txt". A handwritten note on the left side of this screenshot says: "When you run your code, this is all you see, what looks like an 'empty' IDLE shell." Another handwritten note on the left side of the desktop area says: "After your program runs, two new files are CREATED in your folder."

That code worked, too. You've created two data files, each holding the data from each of your lists. Go ahead and open these files in your favorite editor to confirm that they contain the data you expect.



Consider the following carefully: what happens to your data files if the **second** call to `print()` in your code causes an `IOError`?

Files are left open after an exception!

When all you ever do is read data from files, getting an `IOError` is annoying, but rarely dangerous, because your data is still in your file, even though you might be having trouble getting at it.

It's a different story when writing data to files: if you need to handle an `IOError` *before* a file is closed, your written data might become corrupted and there's no way of telling until *after* it has happened.

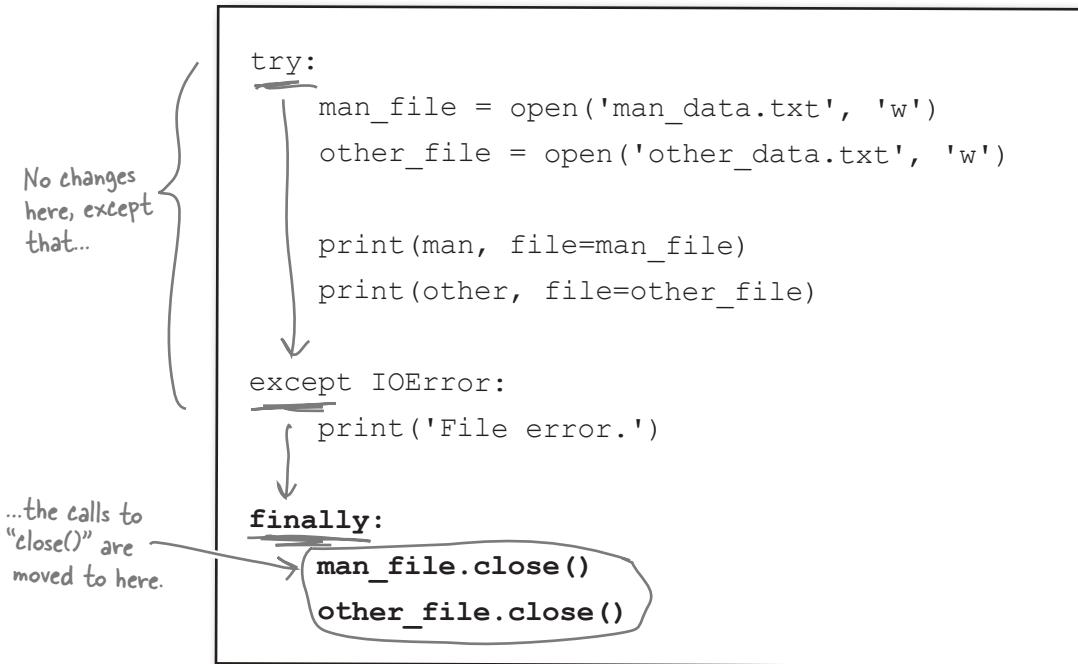
```
try: ✓ Ok.  
    man_file = open('man_data.txt', 'w') ✓ OK  
    other_file = open('other_data.txt', 'w') ✓ OK  
  
    print(man, file=man_file) ✓ OK  
    print(other, file=other_file) ✗ Not OK!!  
  
    Crash!  
  
    man_file.close()  
    other_file.close() } These two lines of code  
except IOError: ✓ OK  
    print('File error.') ✓ OK
```

Your exception-handling code is doing its job, but you now have a situation where your data could *potentially* be corrupted, which can't be good.

What's needed here is something that lets you run some code regardless of whether an `IOError` has occurred. In the context of your code, you'll want to make sure the files are closed *no matter what*.

Extend try with finally

When you have a situation where code must *always* run no matter what errors occur, add that code to your `try` statement's `finally` suite:



If no runtime errors occur, any code in the `finally` suite executes. Equally, if an `IOError` occurs, the `except` suite executes and *then* the `finally` suite runs.

No matter what, the code in the `finally` suite *always* runs.

By moving your file closing code into your `finally` suite, you are reducing the possibility of data corruption errors.

This is a big improvement, because you're now ensuring that files are closed properly (even when write errors occur).

But what about *those* errors?

How do you find out the specifics of the error?

Q: I'm intrigued. When you stripped the `line_spoken` data of unwanted whitespace, you assigned the result back to the `line_spoken` variable. Surely invoking the `strip()` method on `line_spoken` changed the string it refers to?

A: No, that's not what happens. Strings in Python are *immutable*, which means that once a string is created, it **cannot** be changed.

Q: But you did change the `line_spoken` string by removing any unwanted whitespace, right?

A: Yes and no. What actually happens is that invoking the `strip()` method on the `line_spoken` string creates a new string with leading and trailing whitespace removed. The new string is then assigned to `line_spoken`, replacing the data that was referred to before. In effect, it is as if you changed `line_spoken`, when you've actually completely replaced the data it refers to.

Q: So what happens to the replaced data?

A: Python's built-in memory management technology reclaims the RAM it was using and makes it available to your program. That is, unless some other Python data object is also referring to the string.

Q: What? I don't get it.

A: It is conceivable that another data object is referring to the string referred to by `line_spoken`. For example, let's assume you have some code that contains two variables that refer to the same string, namely "Flying Circus." You then decide that one of the variables needs to be in all UPPERCASE, so you invoke the `upper()` method on it. The Python interpreter takes a copy of the string, converts it to uppercase, and returns it to you. You can then assign the uppercase data back to the variable that used to refer to the original data.

Q: And the original data cannot change, because there's another variable referring to it?

A: Precisely. That's why strings are immutable, because you never know what other variables are referring to any particular string.

Q: But surely Python can work out how many variables are referring to any one particular string?

A: It does, but only for the purposes of garbage collection. If you have a line of code like `print('Flying Circus')`, the string is not referred to by a variable (so any variable reference counting that's going on isn't going to count it) but is still a valid string object (which might be referred to by a variable) and it cannot have its data changed under any circumstances.

Q: So Python variables don't actually contain the data assigned to them?

A: That's correct. Python variables contain a reference to a data object. The data object contains the data and, because you can conceivably have a string object used in many different places throughout your code, it is safest to make all strings immutable so that no nasty side effects occur.

Q: Isn't it a huge pain not being able to adjust strings "in place"?

A: No, not really. Once you get used to how strings work, it becomes less of an issue. In practice, you'll find that this issue rarely trips you up.

Q: Are any other Python data types immutable?

A: Yes, a few. There's the tuple, which is an immutable list. Also, all of the number types are immutable.

Q: Other than learning which is which, how will I know when something is immutable?

A: Don't worry: you'll know. If you try to change an immutable value, Python raises a `TypeError` exception.

Q: Of course: an exception occurs. They're everywhere in Python, aren't they?

A: Yes. Exceptions make the world go 'round.

Knowing the type of error is not enough

When a file I/O error occurs, your code displays a generic “File Error” message. This is too generic. How do you know what actually happened?



Who knows?

It turns out that the Python interpreter knows...and it will give up the details if only you'd ask.

When an error occurs at runtime, Python raises an exception of the specific type (such as `IOError`, `ValueError`, and so on). Additionally, Python creates an **exception object** that is passed *as an argument* to your **except** suite.

Let's use IDLE to see how this works.



An IDLE Session

Let's see what happens when you try to open a file that doesn't exist, such as a disk file called `missing.txt`. Enter the following code at IDLE's shell:

```
>>> try:
    data = open('missing.txt')
    print(data.readline(), end='')
except IOError:
    print('File error')
finally:
    data.close()
```

File error ← There's your error message, but...

Traceback (most recent call last):
 File "<pyshell#8>", line 7, in <module>
 data.close()
NameError: name 'data' is not defined

...what's this?!? Another exception was raised and it killed your code.

As the file doesn't exist, the `data` file object **wasn't** created, which subsequently makes it impossible to call the `close()` method on it, so you end up with a `NameError`. A quick fix is to add a small test to the `finally` suite to see if the `data` name exists *before* you try to call `close()`. The `locals()` BIF returns a collection of names defined in the current scope. Let's exploit this BIF to only invoke `close()` when it is safe to do so:

finally: ← The "in" operator tests for membership.

if 'data' in locals(): ← This is just the bit of code that needs to change. Press Alt-P to edit your code at IDLE's shell.

 data.close()

File error ← No extra exceptions this time.
 Just your error message.

Here you're searching the collection returned by the `locals()` BIF for the string `data`. If you find it, you can assume the file was opened successfully and safely call the `close()` method.

If some other error occurs (perhaps something awful happens when your code calls the `print()` BIF), your exception-handling code **catches** the error, **displays** your "File error" message and, finally, **closes** any opened file.

But you still are none the wiser as to *what actually caused the error*.

When an exception is raised and handled by your `except` suite, the Python interpreter passes an exception object into the suite. A small change makes this exception object available to your code as an identifier:

```
except IOError as err:           Give your exception object
    print('File error: ' + err)   a name...
                                ...then use it as part of
                                your error message.
```

But when you try to run your code with this change made, another exception is raised:

```
Traceback (most recent call last):
  File "<pyshell#18>", line 5, in <module>
    print('File error:' + err)
TypeError: Can't convert 'IOError' object to str implicitly
```

Whoops! Yet another exception; this time it's a "TypeError".

This time your error message didn't appear *at all*. It turns out exception objects and strings are not *compatible* types, so trying to concatenate one with the other leads to problems. You can convert (or *cast*) one to the other using the `str()` BIF:

```
except IOError as err:
    print('File error: ' + str(err))           Use the "str()" BIF to force the
                                                exception object to behave like a string.
```

Now, with this final change, your code is behaving exactly as expected:

```
File error: [Errno 2] No such file or directory: 'missing.txt'
```

And you now get a specific error message that tells you exactly what went wrong.

Of course, all this extra logic is starting to obscure the real meaning of your code.



Wouldn't it be dreamy if there were a way to take advantage of these mechanisms without the code bloat? I guess it's just a fantasy...

Use with to work with files

Because the use of the **try/except/finally** pattern is so common when it comes to working with files, Python includes a statement that abstracts away some of the details. The `with` statement, when used with files, can dramatically reduce the amount of code you have to write, because it negates the need to include a `finally` suite to handle the closing of a *potentially* opened data file. Take a look:

The diagram illustrates the equivalence between two pieces of Python code. On the left, a box contains the standard `try/except/finally` pattern:

```
try:  
    data = open('its.txt', "w")  
    print("It's...", file=data)  
except IOError as err:  
    print('File error: ' + str(err))  
finally:  
    if 'data' in locals():  
        data.close()
```

An annotation points to the `try/except/finally` block with the text: "This is the usual 'try/except/finally' pattern."

On the right, a box contains the code using the `with` statement:

```
try:  
    with open('its.txt', "w") as data:  
        print("It's...", file=data)  
    except IOError as err:  
        print('File error: ' + str(err))
```

An annotation points to the `with` block with the text: "The use of 'with' negates the need for the 'finally' suite."

A curved arrow points from the `finally` block on the left to the `with` block on the right, indicating that the `with` statement replaces the need for the explicit `finally` block.

When you use `with`, you no longer have to worry about closing any opened files, as the Python interpreter automatically takes care of this for you. The `with` code on the right is *identical* in function to that on the left. At Head First Labs, we know which approach we prefer.



Geek Bits

The `with` statement takes advantage of a Python technology called the **context management protocol**.



Grab your pencil and rewrite this **try/except/finally** code to use `with` instead. Here's your code with the appropriate `finally` suite added:

```
try:
    man_file = open('man_data.txt', 'w')
    other_file = open('other_data.txt', 'w')

    print(man, file=man_file)
    print(other, file=other_file)
except IOError as err:
    print('File error: ' + str(err))
finally:
    if 'man_file' in locals():
        man_file.close()
    if 'other_file' in locals():
        other_file.close()
```

Write your
“with” code
here.



You were to grab your pencil and rewrite this **try/except/finally** code to use with instead. Here's your code with the appropriate finally suite added:

```
try:
    man_file = open('man_data.txt', 'w')
    other_file = open('other_data.txt', 'w')

    print(man, file=man_file)
    print(other, file=other_file)
except IOError as err:
    print('File error: ' + str(err))
finally:
    if 'man_file' in locals():
        man_file.close()
    if 'other_file' in locals():
        other_file.close()
```

try:

with open('man_data.txt', 'w') as man_file:

print(man, file=man_file)

with open('other_data.txt', 'w') as other_file:

print(other, file=other_file)

except IOError as err:

print('File error: ' + str(err))

Using two "with" statements to rewrite the code without the "finally" suite.

Note the use of the comma.

Or combine the two "open()" calls into one "with" statement:

with open('man_data.txt', 'w') as man_file, open('other_data.txt', 'w') as other_file:

print(man, file=man_file)

print(other, file=other_file)



Test DRIVE

Add your `with` code to your program, and let's confirm that it continues to function as expected. Delete the two data files you created with the previous version of your program and then load your newest code into IDLE and give it a spin.

No errors in the IDLE shell appears to indicate that the program ran successfully.

```

Python 3.1.2 (r312:79360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> RESTART
>>>
>>>
Ln: 6 Col: 4

```

If you check your folder, your two data files should've reappeared. Let's take a closer look at the data file's contents by opening them in your favorite text editor (or use IDLE).

The image shows two Vim windows side-by-side. The top window is titled 'man_data.txt + (~/HeadFirstPython/chapter4) - VIM3' and contains the following text:

```

['Is this the right room for an argument?', "No you haven't!", 'When?', "No you didn't!", "You didn't!", 'You did not!', 'Ah! (taking out his wallet and paying) Just the five minutes.', 'You most certainly did not!', 'Oh no you didn't!', 'Oh no you didn't!', 'Oh look, this isn't an argument!', 'No it isn't!', "It's just contradiction!", 'It IS!', 'You just contradicted me!', 'You DID!', 'You did just then!', '(exasperated) Oh, this is futile!!', 'Yes it is!']

```

The bottom window is titled 'other_data.txt + (~/HeadFirstPython/chapter4) - VIM3' and contains the following text:

```

["I've told you once.", 'Yes I have.', 'Just now.', 'Yes I did!', 'I'm telling you, I did!', 'Oh I'm sorry, is this a five minute argument, or the full half hour?', 'Just the five minutes. Thank you.', 'Anyway, I did.', 'Oh yes I did!', 'Oh yes I did!', 'Yes it is!', 'No it isn't!', 'It is NOT!', 'No I didn't!', 'No no not!', 'Nonsense!', 'No it isn't!']

```

Annotations with arrows point from the text in each window to descriptive labels: 'Here's what the man said.' points to the first window, and 'Here's what the other man said.' points to the second window.

You've saved the lists in two files containing what the *Man* said and what the *Other man* said. Your code is smart enough to handle any exceptions that Python or your operating system might throw at it.

Well done. This is really coming along.

Default formats are unsuitable for files

Although your data is now stored in a file, it's not really in a useful format.
Let's experiment in the IDLE shell to see what impact this can have.



An IDLE Session

Use a `with` statement to open your data file and display a single line from it:

```
>>> with open('man_data.txt') as mdf:  
    print(mdf.readline())
```

Note: no need to close your file,
because "with" does that for you.

```
['Is this the right room for an argument?', "No you haven't!", 'When?', "No you didn't!", "You  
didn't!", 'You did not!', 'Ah! (taking out his wallet and paying) Just the five minutes.',  
'You most certainly did not!', "Oh no you didn't!", "Oh no you didn't!", "Oh look, this isn't  
an argument!", "No it isn't!", "It's just contradiction!", 'It IS!', 'You just contradicted  
me!', 'You DID!', 'You did just then!', '(exasperated) Oh, this is futile!!!', 'Yes it is!']
```

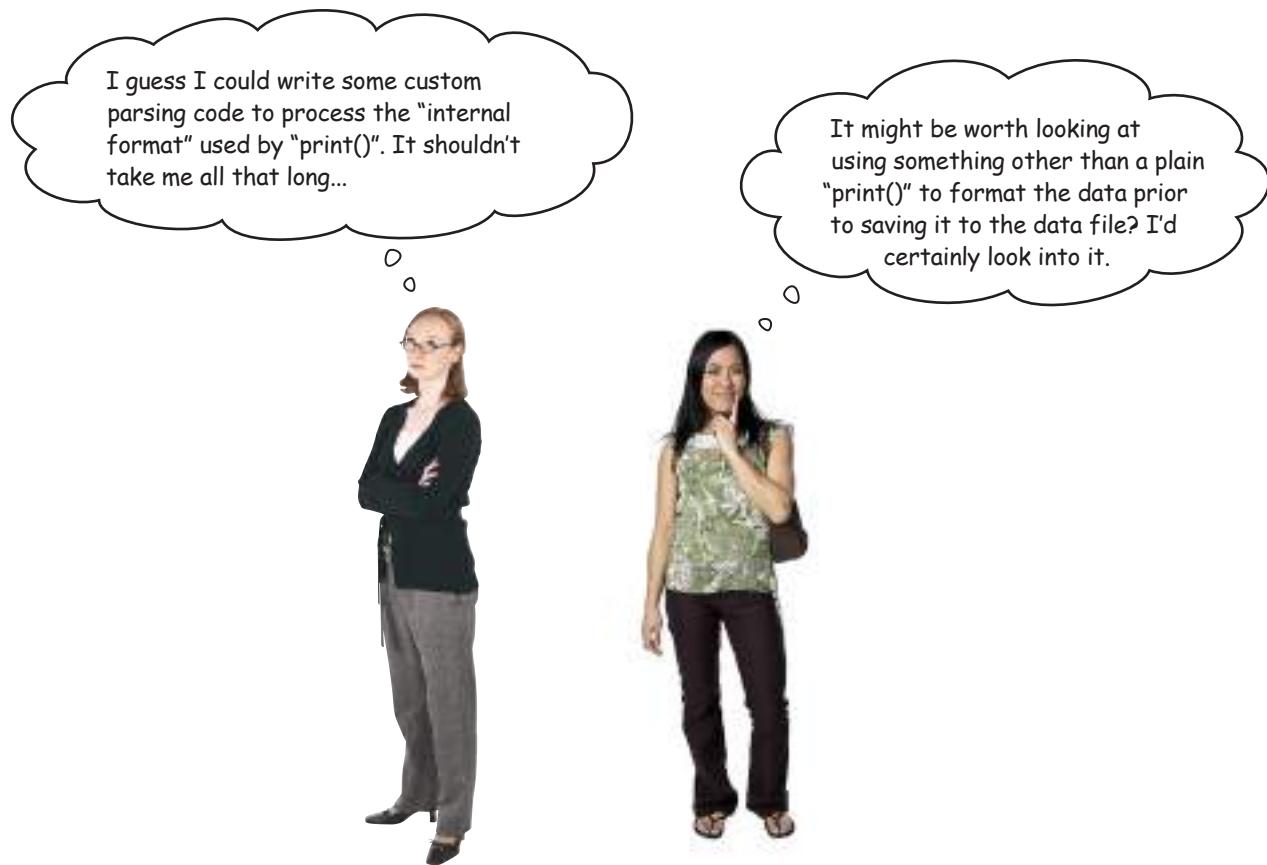
Yikes! It would appear your list is converted to a large string by `print()` when it is saved. Your experimental code reads a single line of data from the file and gets *all* of the data as one large chunk of text...so much for your code saving your *list* data.

What are your options for dealing with this problem?



Geek Bits

By default, `print()` displays your data in a format that mimics how your list data is actually stored by the Python interpreter. The resulting output is not really meant to be processed further... its primary purpose is to show you, the Python programmer, what your list data "looks like" in memory.



Parsing the data in the file is a possibility...although it's *complicated* by all those square brackets, quotes, and commas. Writing the required code is doable, but it is a lot of code just to read back in your saved data.

Of course, if the data is in a *more easily parseable format*, the task would likely be easier, so maybe the second option is worth considering, too?



Can you think of a function you created from earlier in this book that might help here?

Why not modify print_lol()?

Recall your print_lol() function from Chapter 2, which takes any list (or list of lists) and displays it on screen, one line at a time. And nested lists can be indented, if necessary.

This functionality sounds perfect! Here's your code from the nester.py module (last seen at the end of Chapter 2):

```
nester.py - /Users/barryp/Downloads/nester-1.3.0/nester.py

"""This is the "nester.py" module and it provides one function called print_lol()
which prints lists that may or may not include nested lists."""

def print_lol(the_list, indent=False, level=0):
    """ Prints a list of (possibly) nested lists.

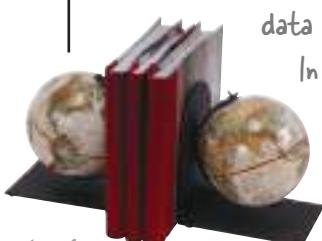
    This function takes a positional argument called "the_list", which
    is any Python list (of - possibly - nested lists). Each data item in the
    provided list is (recursively) printed to the screen on its own line.
    A second argument called "indent" controls whether or not indentation is
    shown on the display. This defaults to False: set it to True to switch on.
    A third argument called "level" (which defaults to 0) is used to insert
    tab-stops when a nested list is encountered."""

    for each_item in the_list:
        if isinstance(each_item, list):
            print_lol(each_item, indent, level+1)
        else:
            if indent:
                for tab_stop in range(level):
                    print("\t", end=' ')
            print(each_item)
```

This code currently displays your data on the screen.

Amending this code to print to a disk file instead of the screen (known as *standard output*) should be relatively straightforward. You can then save your data in a more usable format.

the Scholar's Corner



Standard Output The default place where your code writes its data when the “print()” BIF is used. This is typically the screen.

In Python, standard output is referred to as “sys.stdout” and is importable from the Standard Library’s “sys” module.



**1**

Let's add a fourth argument to your `print_lol()` function to identify a place to write your data to. Be sure to give your argument a default value of `sys.stdout`, so that it continues to write to the screen if no file object is specified when the function is invoked.

Fill in the blanks with the details of your new argument. (**Note:** to save on space, the comments have been removed from this code, but be sure to update your comments in your `nester.py` module after you've amended your code.)

```
def print_lol(the_list, indent=False, level=0, .....):
    for each_item in the_list:
        if isinstance(each_item, list):
            print_lol(each_item, indent, level+1, .....)
        else:
            if indent:
                for tab_stop in range(level):
                    print("\t", end=' ', .....)
            print(each_item, .....)
```

2

What needs to happen to the code in your `with` statement now that your amended `print_lol()` function is available to you?

.....
.....
.....

3

List the name of the module(s) that you now need to import into your program in order to support your amendments to `print_lol()`.

.....
.....



Exercise Solution

- 1** You were to add a fourth argument to your `print_lol()` function to identify a place to write your data to, being sure to give your argument a default value of `sys.stdout` so that it continues to write to the screen if no file object is specified when the function is invoked.

You were to fill in the blanks with the details of your new argument. (**Note:** to save on space, the comments have been removed from this code, but be sure to update those in your `nester.py` module after you've amended your code).

```

def print_lol(the_list, indent=False, level=0, ..... fh=sys.stdout):
    for each_item in the_list:
        if isinstance(each_item, list):
            print_lol(each_item, indent, level+1, ..... fh)
        else:
            if indent:
                for tab_stop in range(level):
                    print("\t", end='', ..... file=fh)
            print(each_item, ..... file=fh)

```

Add the fourth argument and give it a
default value.

Note: the
signature has
changed.

Adjust the two
calls to "print()"
to use the new
argument.

- 2** What needs to happen to the code in your `with` statement now that your amended `print_lol()` function is available to you?

The code needs to be adjusted so that instead of using the
"print()" BIF, the code needs to invoke "print_lol()" instead.

- 3** List the name of the module(s) that you now need to import into your program in order to support your amendments to `print_lol()`.

The program needs to import the amended "nester" module.



Test Drive

Before taking your code for a test drive, you need to do the following:

1. Make the necessary changes to `nester` and install the amended module into your Python environment (see Chapter 2 for a refresher on this). You might want to upload to PyPI, too.
2. Amend your program so that it imports `nester` and uses `print_lol()` instead of `print()` within your `with` statement. **Note:** your `print_lol()` invocation should look something like this:
`print_lol(man, fh=man_file).`

When you are ready, take your latest program for a test drive and let's see what happens:

As before, there's no output on screen.

>>>") followed by a blank line. Handwritten text 'As before, there's no output on screen.' has an arrow pointing to the terminal window."/>

```
Python 3.1.2 (r312i2179360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>>
```

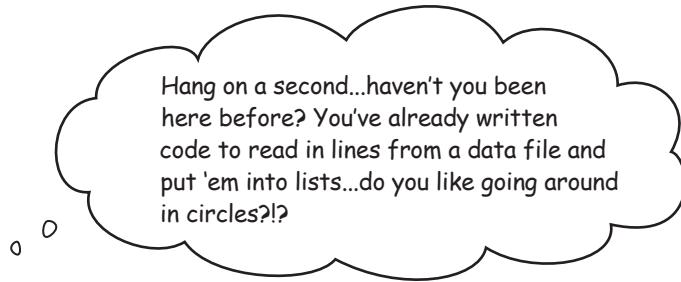
Let's check the contents of the files to see what they look like now.

```
Is this the right room for an argument?
No you taxes'n't!
Whent?
No you didn't!
You didn't!
You did'nt!
An' (taking out his wallet and paying) just the time minutes.
You most certainly did not!
Oh no you didn't!
Oh no you didn't!
Oh look, this isn't an argument!
No it isn't!
It's just contradiction!
It IS!
You just contradicted me!
You DID!
You did just then!
Desperated! Oh, this is terrible!
Yes it is!
```

```
I've told you once.
You I have.
Just now.
You I did.
I'm telling you, I did!
Oh I'm sorry, is this a five minute argument, or the full half hour?
JUST THE FIVE MINUTES. Thank you.
Anyway, I did.
Oh yes I did!
Oh yes I did!
Yes it is!
No it isn't!
It is NOT!
No I didn't!
No no no!
Nonsense!
No it isn't!
```

This is looking good. By amending your `nester` module, you've provided a facility to save your list data in a legible format. It's now way easier on the eye.

But does this make it any easier to read the data back in?



That's a good point.

This problem is not unlike the problem from the beginning of the chapter, in that you've got lines of text in a disk file that you need to process, only now you have *two* files instead of one.

You know how to write the code to process your new files, but writing custom code like this is specific to the format that you've created for this problem. This is *brittle*: if the data format changes, your custom code will have to change, too.

Ask yourself: *is it worth it?*



Custom Code Exposed

This week's interview:
When is custom code appropriate?

Head First: Hello, CC, how are you today?

Custom Code: Hi, I'm great! And when I'm not great, there's always something I can do to fix things. Nothing's too much trouble for me. Here: have a seat.

Head First: Why, thanks.

Custom Code: Let me get that for you. It's my new custom *SlideBack&Groove™*, the 2011 model, with added cushions and lumbar support...and it automatically adjusts to your body shape, too. How does that feel?

Head First: Actually [relaxes], that feels kinda groovy.

Custom Code: See? Nothing's too much trouble for me. I'm your "go-to guy." Just ask; absolutely anything's possible when it's a custom job.

Head First: Which brings me to why I'm here. I have a "delicate" question to ask you.

Custom Code: Go ahead, shoot. I can take it.

Head First: When is custom code appropriate?

Custom Code: Isn't it obvious? It's *always* appropriate.

Head First: Even when it leads to problems down the road?

Custom Code: Problems?!? But I've already told you: nothing's too much trouble for me. I live to customize. If it's broken, I fix it.

Head First: Even when a readymade solution might be a better fit?

Custom Code: Readymade? You mean (I hate to say it): *off the shelf*?

Head First: Yes. Especially when it comes to writing complex programs, right?

Custom Code: What?!? That's where I excel: creating beautifully crafted custom solutions for folks with complex computing problems.

Head First: But if something's been done before, why reinvent the wheel?

Custom Code: But everything I do is custom-made; that's why people come to me...

Head First: Yes, but if you take advantage of other coders' work, you can build your own stuff in half the time *with less code*. You can't beat that, can you?

Custom Code: "Take advantage"...isn't that like *exploitation*?

Head First: More like collaboration, sharing, participation, *and* working together.

Custom Code: [shocked] You want me to give my code...*away*?

Head First: Well...more like share and share alike. I'll scratch your back if you scratch mine. How does that sound?

Custom Code: That sounds disgusting.

Head First: Very droll [laughs]. All I'm saying is that it is not always a good idea to create everything from scratch with custom code when a good enough solution to the problem might already exist.

Custom Code: I guess so...although it won't be as perfect a fit as that chair.

Head First: But I will be able to sit on it!

Custom Code: [laughs] You should talk to my buddy **Pickle**...he's forever going on about stuff like this. And to make matters worse, he lives in a library.

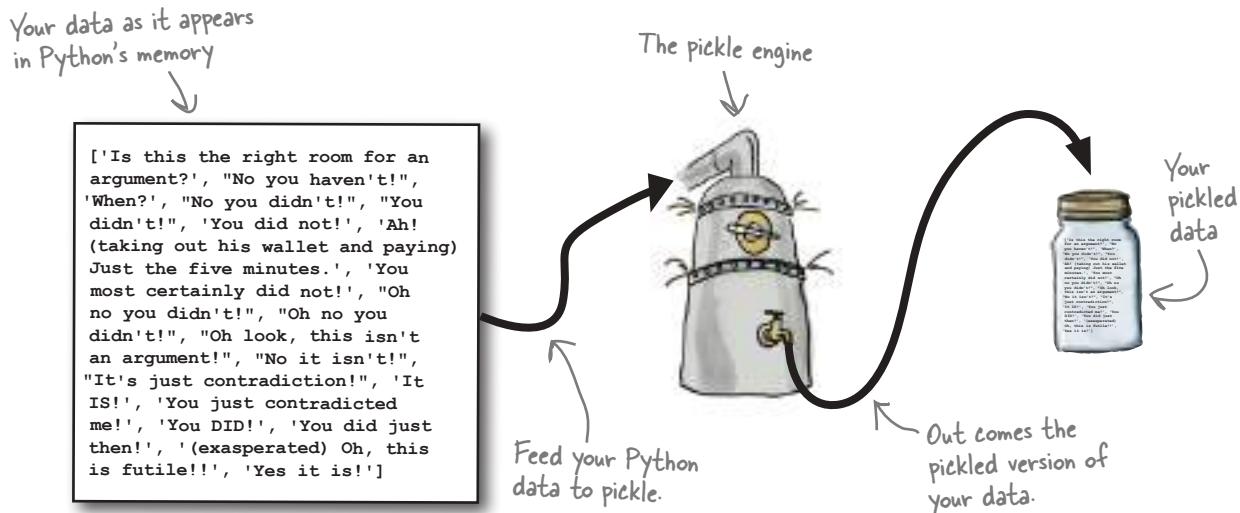
Head First: I think I'll give him a shout. Thanks!

Custom Code: Just remember: you know where to find me if you need any custom work done.

Pickle your data

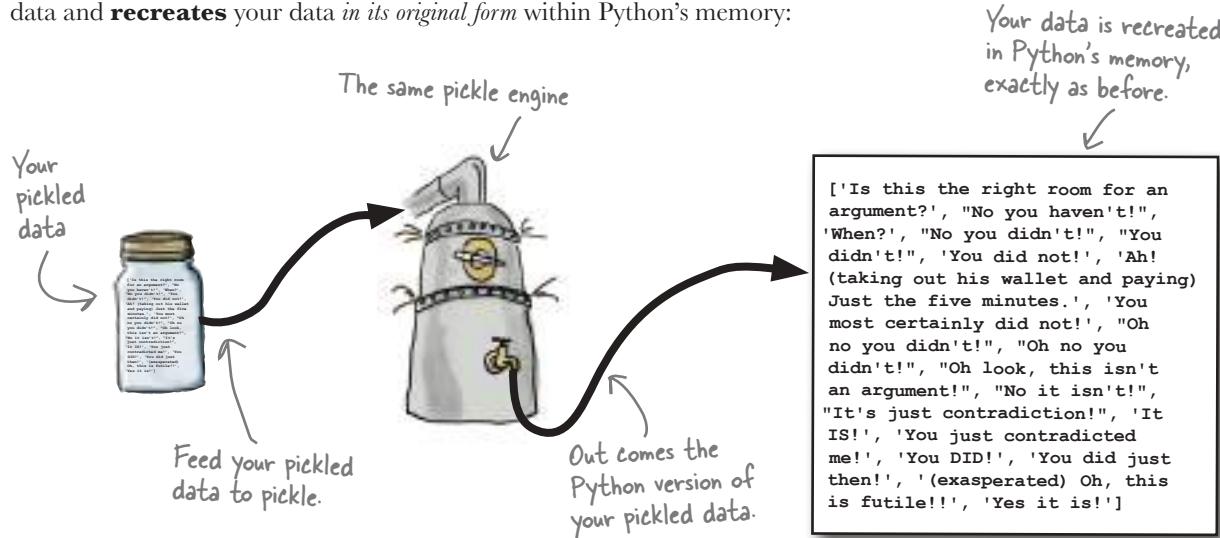
Python ships with a standard library called `pickle`, which can save and load almost any Python data object, including lists.

Once you *pickle* your data to a file, it is **persistent** and ready to be read into another program at some later date/time:



You can, for example, **store** your pickled data on disk, **put** it in a database, or **transfer** it over a network to another computer.

When you are ready, reversing this process unpickles your persistent pickled data and **recreates** your data *in its original form* within Python's memory:



Save with dump and restore with load

Using pickle is straightforward: import the required module, then use `dump()` to save your data and, some time later, `load()` to restore it. The only requirement when working with pickled files is that they have to be opened in *binary access mode*:

```

import pickle
...
with open('mydata.pickle', 'wb') as mysavedata:
    pickle.dump([1, 2, 'three'], mysavedata)
...
with open('mydata.pickle', 'rb') as myrestoreddata:
    a_list = pickle.load(myrestoreddata)
print(a_list)

```

Always remember to import the "pickle" module.

To save your data, use "dump()".

Assign your restored data to an identifier.

The "b" tells Python to open your data files in **BINARY** mode.

Once your data is back in your program, you can treat it like any other data object.

What if something goes wrong?

If something goes wrong when pickling or unpickling your data, the `pickle` module raises an exception of type `PickleError`.



Here's a snippet of your code as it currently stands. Grab your pencil and strike out the code you no longer need, and then replace it with code that uses the facilities of `pickle` instead. Add any additional code that you think you might need, too.

```

try:
    with open('man_data.txt', 'w') as man_file, open('other_data.txt', 'w') as other_file:
        nester.print_lol(man, fh=man_file)
        nester.print_lol(other, fh=other_file)
except IOError as err:
    print('File error: ' + str(err))

```

Sharpen your pencil Solution

Import "pickle" near
the top of your
program.

```
import pickle
```

try:

```
    with open('man_data.txt', 'wb') as man_file, open('other_data.txt', 'wb') as other_file:  
        nester.print_lol(man, f=man_file)  
        nester.print_lol(other, f=other_file)  
    pickle.dump(man, man_file)  
    pickle.dump(other, other_file)
```

except IOError as err:

```
    print('File error: ' + str(err))
```

except pickle.PickleError as perr:

```
    print('Pickling error: ' + str(perr))
```

Here's a snippet of your code as it currently stands. You were to grab your pencil and strike out the code you no longer need, and then replace it with code that uses the facilities `pickle` instead. You were also to add any additional code that you think you might need.

Change the access mode to
be "writeable, binary".

'wb'

'wb'

Replace the two calls to "nester.print_lol()
with calls to "pickle.dump()".

Don't forget to handle any exceptions
that can occur.

there are no Dumb Questions

Q: When you invoked `print_lol()` earlier, you provided only two arguments, even though the function signature requires you to provide four. How is this possible?

A: When you invoke a Python function in your code, you have options, especially when the function provides default values for some arguments. If you use positional arguments, the position of the argument in your function invocation dictates what data is assigned to which argument. When the function has arguments that also provide default values, you do not need to always worry about positional arguments being assigned values.

Q: OK, you've completely lost me. Can you explain?

A: Consider `print()`, which has this signature: `print(value, sep=' ', end='\n', file=sys.stdout)`. By default, this BIF displays to standard output (the screen), because it has an argument called `file` with a default value of `sys.stdout`. The `file` argument is the fourth positional argument. However, when you want to send data to something other than the screen, you do not need to (nor want to have to) include values for the second and third positional arguments. They have default values anyway, so you need to provide values for them only if the defaults are not what you want. If all you want to do is to send data to a file, you invoke the `print()` BIF like this: `print("Dead Parrot Sketch", file='myfavmonty.txt')` and the fourth positional argument uses the value you specify, while the other positional arguments use their defaults. In Python, not only do the BIFs work this way, but your custom functions support this mechanism, too.



Test DRIVE

Let's see what happens now that your code has been amended to use the standard `pickle` module instead of your custom `nester` module. Load your amended code into IDLE and press F5 to run it.

Once again, you
get no visual clue
that something has
happened.

```
Python 3.1.2 (r312i79360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
>>>
>>>
```

Ln: 6 Col: 4

So, once again, let's check the contents of the files to see what they look like now:

The left Vim window shows the file `man_data.txt` with the following content:

```
<80>^C]q;qX;rshere is this the right room for an argument?@^MX@rshere@No
you haven't q;qX;rshere@when?@^CX;rshere@No you didn't!@q;qX^K;rshere@You al
dn't!@q;qX^K;rshere@You did not!@q;qX^K;rshere@Ah! (taking out his wallet and pa
ying) Just the five minutes.@^CX^K;rshere@You most certainly did not!@q;qX^K
;rshere@Oh no you didn't!@q;qX^K;rshere@Oh no you didn't!@q;qX^K;rshere@Oh
look, this isn't an argument!@^CX^K;rshere@No it isn't!@q;qLXX^K;rshere@It's just
contradiction!@^MX^K;rshere@It IS!@q;qLXX^K;rshere@I predicted we'd@^CX^K;rshere@You DID!@q;qX^K;rshere@You c
asperated! Oh, this is futile!!@q;qX^K;rshere@Yes it is!@q;qSe.
```

The right Vim window shows the file `other_data.txt` with the following content:

```
<80>^C]q;qX;rshere@I've told you once,@^MX@rshere@Yes I have.q^BX @rshere
@Just now.q^CX
@rshere@Yes I did!@q;qX^K;rshere@Oh I'm sorry, is this a five minute argument, or the full half hour?@^CX^K;rshere@Just
the five minutes. Thank you.q^CX^K;rshere@Anyway, I did.q^MX@rshere@Now let's
got one thing quite clear: I most definitely told you q X;rshere@Oh yes I did!q
X;rshere@Oh yes I did!q^MX
@rshere@Yes it is!@q;qX^K;rshere@No it isn't!@q;qMX
@rshere@It is NOT!@q;qX^K;rshere@No I didn't!@q;qX
@rshere@No sense!@q;qX^K;rshere@No it isn't!@q;qRe.
```

Note: The text in the Vim windows is heavily encoded with various escape sequences and non-printing characters, appearing as a mix of random symbols and occasional readable words like 'MX' and 'MX^K'. This is the result of Python's `pickle` module serializing objects into a binary format that is not meant to be human-readable.

It appears to have worked...but these files look like *gobbledygook!* What gives?

Recall that Python, not you, is pickling your data. To do so efficiently, Python's `pickle` module uses a custom binary format (known as its **protocol**). As you can see, viewing this format in your editor looks decidedly *weird*.

Don't worry: it is supposed to look like this.



An IDLE Session

pickle really shines when you load some previously pickled data into another program. And, of course, there's nothing to stop you from using pickle with nester. After all, each module is designed to serve different purposes. Let's demonstrate with a handful of lines of code within IDLE's shell. Start by importing any required modules:

```
>>> import pickle
>>> import nester
```

No surprises there, eh?

Next up: create a new identifier to hold the data that you plan to unpickle. Create an empty list called new_man:

```
>>> new_man = []
```

Yes, almost too exciting for words, isn't it? With your list created, let's load your pickled data into it. As you are working with external data files, it's best if you enclose your code with **try/except**:

```
>>> try:
    with open('man_data.txt', 'rb') as man_file:
        new_man = pickle.load(man_file)
except IOError as err:
    print('File error: ' + str(err))
except pickle.PickleError as perr:
    print('Pickling error: ' + str(perr))
```

This code is not news to you either. However, at this point, your data has been unpickled and assigned to the new_man list. It's time for nester to do its stuff:

```
>>> nester.print_lol(new_man)
Is this the right room for an argument?
No you haven't!
When?
No you didn't!
...
You did just then!
(exasperated) Oh, this is futile!!
Yes it is!
```

Not all the data is shown here, but trust us: it's all there.

And to finish off, let's display the first line spoken as well as the last:

```
>>> print(new_man[0])
Is this the right room for an argument?
>>> print(new_man[-1])
Yes it is!
```

See: after all that, it is the right room! ☺

Generic file I/O with pickle is the way to go!



Python takes care of your file I/O details, so you can concentrate on what your code actually does or needs to do.

As you've seen, being able to work with, save, and restore data in lists is a breeze, thanks to Python. But what other **data structures** does Python support *out of the box*?

Let's dive into Chapter 5 to find out.



Your Python Toolbox

You've got Chapter 4 under your belt and you've added some key Python techniques to your toolbox.

Python Lingo

- “Immutable types” – data types in Python that, once assigned a value, cannot have that value changed.
- “Pickling” – the process of saving a data object to persistence storage.
- “Unpickling” – the process of restoring a saved data object from persistence storage.



BULLET POINTS

- The `strip()` method removes unwanted whitespace from strings.
- The `file` argument to the `print()` BIF controls where data is sent/saved.
- The `finally` suite is always executed no matter what exceptions occur within a `try/except` statement.
- An exception object is passed into the `except` suite and can be assigned to an identifier using the `as` keyword.
- The `str()` BIF can be used to access the stringed representation of any data object that supports the conversion.
- The `locals()` BIF returns a collection of variables within the current scope.
- The `in` operator tests for membership.
- The “+” operator concatenates two strings when used with strings but adds two numbers together when used with numbers.
- The `with` statement automatically arranges to close all opened files, even when exceptions occur. The `with` statement uses the `as` keyword, too.
- `sys.stdout` is what Python calls “standard output” and is available from the standard library’s `sys` module.
- The standard library’s `pickle` module lets you easily and efficiently save and restore Python data objects to disk.
- The `pickle.dump()` function saves data to disk.
- The `pickle.load()` function restores data from disk.

5 comprehending data



Work that data!



Data comes in all shapes and sizes, formats and encodings.

To work effectively with your data, you often have to manipulate and transform it into a common format to allow for efficient processing, sorting, and storage. In this chapter, you'll explore Python goodies that help you work your data up into a sweat, allowing you to achieve data-munging greatness. So, flip the page, and let's not keep the coach waiting...

Coach Kelly needs your help



The coach is an old friend, and you'd love to help. His crack squad of U10 athletes has been training hard. With each 600m run they do, Coach Kelly has recorded their time in a text file on his computer. There are four files in all, one each for James, Sarah, Julie, and Mikey.

julie.txt

2.59, 2.11, 2:11, 2:23, 3-10, 2-23, 3:10, 3.21, 3-21

2-34, 3:21, 2.34, 2.45, 3.01, 2:01, 2:01, 3:10, 2-22

James.txt

2:58, 2.58, 2:39, 2-25, 2-55, 2:54, 2.18, 2:55, 2:55

2:22, 3.01, 3:01, 3.02, 3:02, 3.02, 3:22, 2.49, 2:38

sarah.txt

Mikey.txt

Initially, the coach needs a quick way to know the *top three fastest times* for each athlete.

Can you help?



Before proceeding with this chapter, take a few moments to download the four data files from the *Head First Python* support website.



Let's begin by reading the data from each of the files into its own list. Write a short program to process each file, creating a list for each athlete's data, and display the lists on screen.

Hint: Try splitting the data on the commas, and don't forget to strip any unwanted whitespace.

Write your code here.



Exercise Solution

Open each of the data files in turn, read the line of data from the file, and create a list from the line of data.

Let's begin by reading the data from each of the files into its own list. You were to write a short program to process each file, creating a list for each athlete's data. You were then to display the lists on screen.

Open the file.

`with open('james.txt') as jaf:`

`data = jaf.readline()` ← Read the line of data.

`james = data.strip().split(',') ← Convert the data to a list.`

`with open('julie.txt') as juf:`

`data = juf.readline()`

`julie = data.strip().split(',') ←`

`with open('mikey.txt') as mif:`

`data = mif.readline()`

`mikey = data.strip().split(',') ←`

`with open('sarah.txt') as saf:`

`data = saf.readline()`

`sarah = data.strip().split(',') ←`

`print(james)`

`print(julie)`

`print(mikey)`

`print(sarah)`

Display the four lists on screen.

there are no
Dumb Questions

Q: That `data.strip().split(',')` line looks a little weird. Can you explain what's going on?

A: That's called *method chaining*. The first method, `strip()`, is applied to the line in `data`, which removes any unwanted whitespace from the string. Then, the results of the stripping are processed by the second method, `split(',')`, creating a list. The resulting list is then applied to the target identifier in the previous code. In this way, the methods are **chained together** to produce the required result. It helps if you read method chains from left to right.



Test DRIVE

Load your code into IDLE and run it to confirm that it's all OK for now:

Here's your
program as
displayed in
IDLE.

And here's the output
produced by running
your code.

The screenshot shows the Python IDLE interface. The top window is titled "coach.py - /Users/barnyp/HeadFirstPython/chapter5/coach.py" and contains the following Python code:

```

with open('james.txt') as jaf:
    data = jaf.readline()
    james = data.strip().split(',')
with open('julie.txt') as juf:
    data = juf.readline()
    julie = data.strip().split(',')
with open('mikey.txt') as mif:
    data = mif.readline()
    mikey = data.strip().split(',')
with open('sarah.txt') as saf:
    data = saf.readline()
    sarah = data.strip().split(',')

print(james)
print(julie)
print(mikey)
print(sarah)

```

The bottom window is titled "Python Shell" and shows the output of running the code:

```

Python 3.1.2 (r312:79360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> -----
RESTART
>>>
['2:34', '3:21', '2.34', '2.45', '3.01', '2:01', '2:01', '3:10', '2-22']
[2.59, 2.11, '2:11', '2:23', '3-10', '2-23', '3:10', '3.21', '3-21']
['2:22', '3.01', '3:01', '3.02', '3:02', '3.02', '3:22', '2.49', '2:30']
['2:58', '2.58', '2:39', '2-25', '2-55', '2:54', '2.18', '2:55', '2:55']
>>> |

```

So far, so good. Coach Kelly's data is now represented by four lists in Python's memory. Other than the use of method chaining, there's nothing much new here, because you've pretty much mastered reading data from files and using it to populate lists.

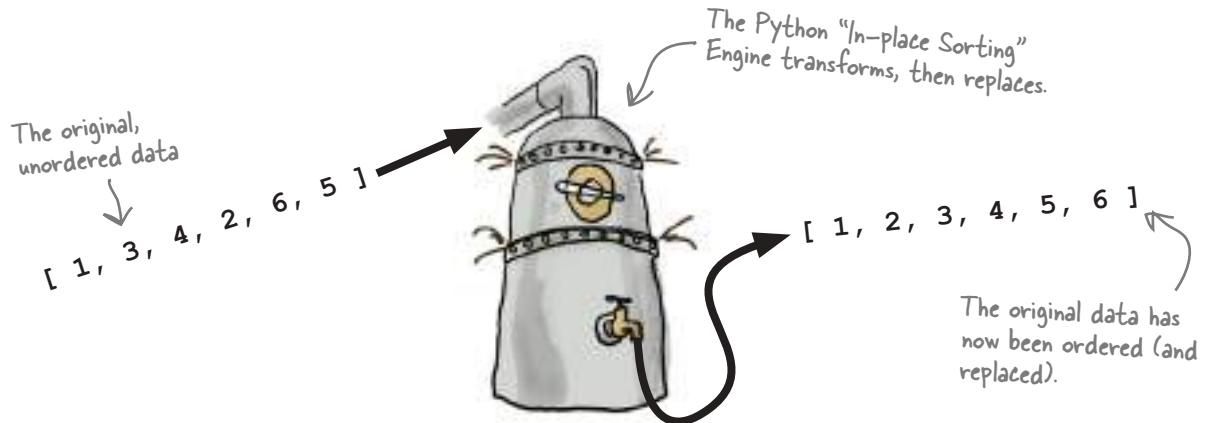
There's nothing to show the coach yet, so no point in disturbing him until his data is arranged in ascending order, which requires you to *sort* it.

Let's look at your sorting options in Python.

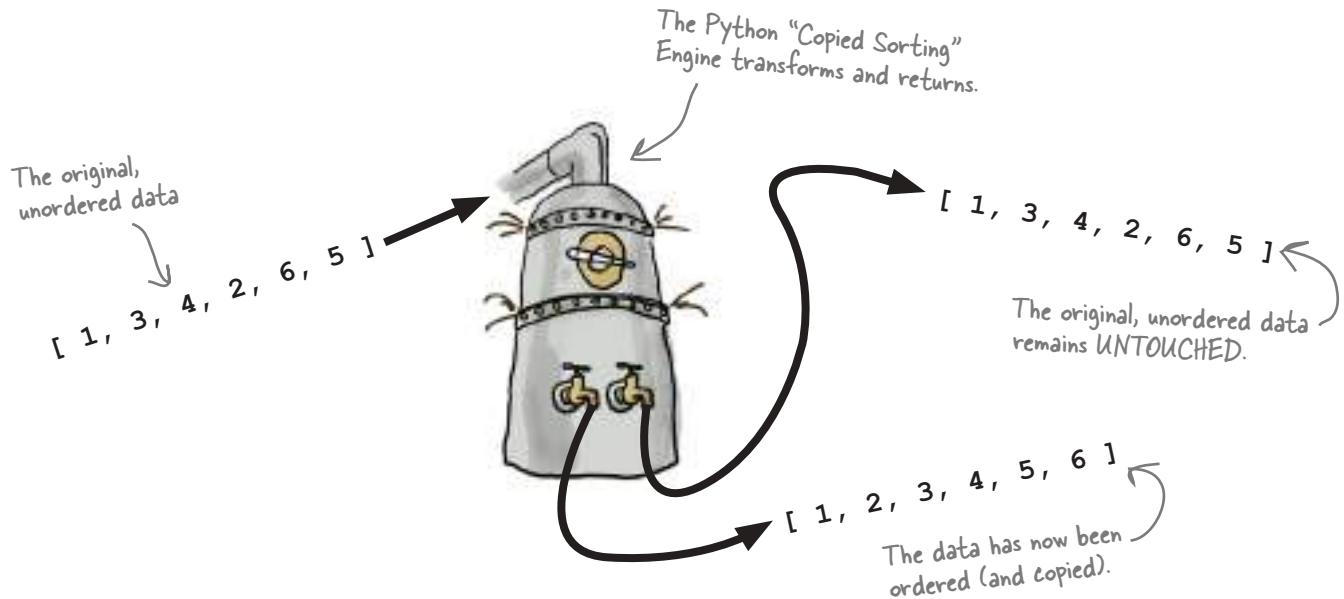
Sort in one of two ways

When it comes to sorting your data using Python, you have two options.

In-place sorting takes your data, *arranges* it in the order you specify, and then *replaces* your original data with the sorted version. The original ordering is lost. With lists, the `sort()` method provides in-place sorting:



Copied sorting takes your data, arranges it in the order you specify, and then *returns a sorted copy* of your original data. Your original data's ordering is maintained and only the copy is sorted. In Python, the `sorted()` BIF supports copied sorting.





An IDLE Session

Let's see what happens to your data when each of Python's sorting options is used. Start by creating an unordered list at the IDLE shell:

```
>>> data = [6, 3, 1, 2, 4, 5]
```

} Create a list of
unordered data and
assign to a variable.

```
>>> data
```

```
[6, 3, 1, 2, 4, 5]
```

Perform an in-place sort using the `sort()` method that is built in as standard to every Python list:

```
>>> data.sort() ← Perform IN-PLACE sorting on the data.
```

```
>>> data
```

```
[1, 2, 3, 4, 5, 6] ← The data's ordering has changed.
```

Reset data to its original unordered state, and then perform a copied sort using the `sorted()` BIF:

```
>>> data = [6, 3, 1, 2, 4, 5]
```

```
>>> data
```

```
[6, 3, 1, 2, 4, 5] ← The data's ordering has been reset.
```

```
>>> data2 = sorted(data) ← Perform COPIED sorting on the data.
```

```
>>> data
```

```
[6, 3, 1, 2, 4, 5] ← Same as it ever was.
```

```
>>> data2
```

```
[1, 2, 3, 4, 5, 6] ← The copied data is ordered  
from lowest to highest.
```



Sharpen your pencil

Either sorting option works with the coach's data, but let's use a copied sort for now to arrange to sort the data on output. In the space below, provide four amended `print()` statements to replace those at the bottom of your program.

.....
.....
.....
.....
.....



Either sorting option works with the coach's data, but let's use a copied sort for now to arrange to sort the data on output. You were to provide four amended `print()` statements to replace those at the bottom of your program.

Simply call
“sorted()” on the
data BEFORE
you display it on
screen.

```
print(sorted(james))  
print(sorted(julie))  
print(sorted(mikey))  
print(sorted(sarah))
```

there are no
Dumb Questions

Q: What happens to the unsorted data when I use `sort()`?

A: For all intents and purposes, it disappears. Python takes a copy, sorts it, and then replaces your original data with the sorted version.

Q: And there's no way to get the original data back?

A: No. If the ordering of the original data is important to you, use the `sorted()` BIF to transform your data into a sorted copy.



Geek Bits

You've already seen *method chaining*, and now it's time to say "hello" to *function chaining*. Function chaining allows you to apply a series of functions to your data. Each function takes your data, performs some operation on it, and then passes the transformed data on to the next function. Unlike method chains, which read from left to right, function chains read from **right to left** (just to keep things interesting).



Test DRIVE

Let's see if this improves your output in any way. Make the necessary amendments to your code and run it.

```

coach2.py - /Users/barryp/HeadFirstPython/chapter5/coach2.py
with open('james.txt') as jaf:
    data = jaf.readline()
james = data.strip().split(',')
with open('julie.txt') as juf:
    data = juf.readline()
julie = data.strip().split(',')
with open('mikey.txt') as mif:
    data = mif.readline()
mikey = data.strip().split(',')
with open('sarah.txt') as saf:
    data = saf.readline()
sarah = data.strip().split(',')
print(sorted(james))
print(sorted(julie))
print(sorted(mikey))
print(sorted(sarah))

```

Here's the updates to the code.

But look at THIS! The data is not at all sorted...which is, like, weird.

Python Shell

```

>>> -----
RESTART -----
[ '2-22', '2-34', '2.34', '2.45', '2:01', '2:01', '3.01', '3:10', '3:21'
[ '2-23', '2.11', '2.59', '2:11', '2:23', '3-10', '3-21', '3.21', '3:10'
[ '2.49', '2:22', '2:38', '3.01', '3.02', '3.02', '3:01', '3:02', '3:22'
[ '2-25', '2-55', '2.18', '2.58', '2:39', '2:54', '2:55', '2:55', '2:58'
>>> |

```

Look at this: 2-55 is coming BEFORE 2.18...now that is weird.



Hey, it looks like your data values are not uniform. Is the problem with all those periods, dashes, and colons?

Yes. The minute and seconds separators are confusing Python's sorting technology.

When recording his athletes' times in each of their files, Coach Kelly sometimes used a different character to separate minutes from seconds. It looks like you need to fix your data.

The trouble with time

Well...there's never enough of it, is there?

Let's look closely at the coach's data to see what the problem is. Here's Sarah raw data again:

```
2:58,2.58,2:39,2-25,2-55,2:54,2.18,2:55,2:55
```

sarah.txt

Oh, look: what a lovely bunch of strings...



Recall that data read from a file comes into your program *as text*, so Sarah's data looks like this once you turn it into a list of "times":

```
['2:58', '2.58', '2:39', '2-25', '2-55', '2:54', '2.18', '2:55', '2:55']
```

These are all strings, even though the coach thinks they're times.

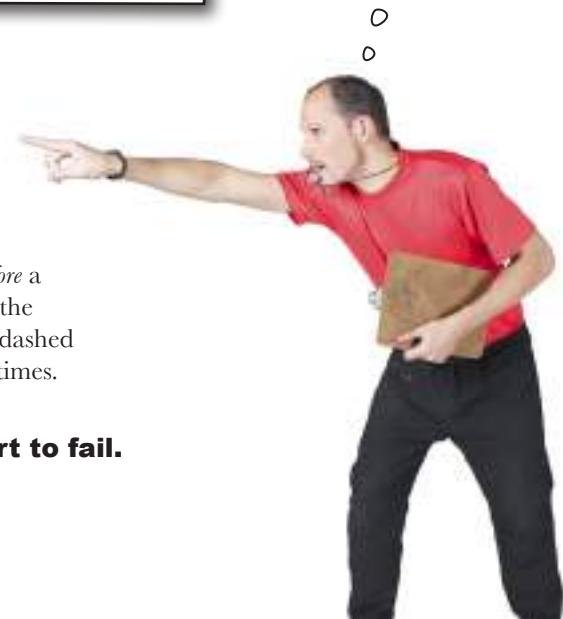
And when you sort Sarah's data, it ends up in this order (which isn't quite what you were expecting):

```
['2-25', '2-55', '2.18', '2.58', '2:39', '2:54', '2:55', '2:55', '2:58']
```

Whoops! That's not right. How can 2.18 come after 2-55?

Whoops again. 2:39 can't come between 2.58 and 2:54, can it?

I don't get what the problem is...they're all times to me.



Python sorts the strings, and when it comes to strings, a dash comes *before* a period, which itself comes *before* a colon. As all the strings start with 2, the next character in each string acts like a grouping mechanism, with the dashed times grouped and sorted, then the period times, and finally the colon times.

Nonuniformity in the coach's data is causing the sort to fail.



Code Magnets

Let's create a function called `sanitize()`, which takes as input a string from each of the athlete's lists. The function then processes the string to replace any dashes or colons found with a period and returns the sanitized string. **Note:** if the string already contains a period, there's no need to sanitize it.

Rearrange the code magnets at the bottom of the page to provide the required functionality.

```
def sanitize(time_string):
```

.....

.....

.....

.....

.....

.....

.....

.....

```
return(mins + '.' + secs)
```

- Return the sanitized time string to the caller of this function.

Your magnets are waiting.

```
time string.split(splitter)
```

else:

```
if '-' in time_string:
```

```
elif ':' in time_string:
```

```
return(time_string)
```

$$(\min_{\theta} \sigma_{\text{loss}}) =$$

```
splitter = ...
```

```
splitter = '--'
```



Code Magnets Solution

You were to create a function called `sanitize()`, which takes as input a string from each of the athlete's lists. The function then processes the string to replace any dashes or colons found with a period and returns the sanitized string. **Note:** if the string already contains a period, there's no need to sanitize it.

You were to rearrange the code magnets at the bottom of the previous page to provide the required functionality.

```
def sanitize(time_string):  
    .....  
    if '-' in time_string:  
        splitter = '-'  
    elif ':' in time_string:  
        splitter = ':'  
    else:  
        return(time_string)  
    .....  
    (mins, secs) = time_string.split(splitter)  
    .....  
    return(mins + '.' + secs)
```

Use the "in" operator to check if the string contains a dash or a colon.

Do nothing if the string does NOT need to be sanitized.

Split the string to extract the minutes and seconds parts.

Of course, on its own, the `sanitize()` function is not enough. You need to iterate over each of your lists of data and use your new function to convert each of the athlete's times into the correct format.

Let's put your new function to work right away.



Exercise

Let's write the code to convert your existing data into a sanitized version of itself. Create four new lists to hold the sanitized data. Iterate over each athlete's list data and append each sanitized string from each list to the appropriate new list. Conclude your program by printing a sorted copy of each new list to the screen.

The code that reads the data from the data files remains unchanged (and has been compressed to fit on this page). 

```
✓with open('james.txt') as jaf: data = jaf.readline()
james = data.strip().split(',')
with open('julie.txt') as juf: data = juf.readline()
julie = data.strip().split(',')
with open('mikey.txt') as mif: data = mif.readline()
mikey = data.strip().split(',')
with open('sarah.txt') as saf: data = saf.readline()
sarah = data.strip().split(',')  
✓
```

Add your new code here. ↴

What happens
to the four
“print()”
statements?

```
print(.....)
print(.....)
print(.....)
print(.....)
```



Exercise Solution

Let's write the code to convert your existing data into a sanitized version of itself. You were to create four new lists to hold the sanitized data. You were then to iterate over each athlete's data and append each sanitized string from each list to an appropriate new list. You were to conclude your program by printing a sorted copy of each new list to the screen.

```
with open('james.txt') as jaf: data = jaf.readline()
james = data.strip().split(',')
with open('julie.txt') as juf: data = juf.readline()
julie = data.strip().split(',')
with open('mikey.txt') as mif: data = mif.readline()
mikey = data.strip().split(',')
with open('sarah.txt') as saf: data = saf.readline()
sarah = data.strip().split(',')
```

Create four new, initially empty lists.

```
clean_james = []
clean_julie = []
clean_mikey = []
clean_sarah = []
```

```
for each_t in james:
    clean_james.append(sanitize(each_t))
for each_t in julie:
    clean_julie.append(sanitize(each_t))
for each_t in mikey:
    clean_mikey.append(sanitize(each_t))
for each_t in sarah:
    clean_sarah.append(sanitize(each_t))
```

Take each of the data items in the original lists, sanitize them, and then append the sanitized data to the appropriate new list.

The four "print()" statements now display the new lists, which are sorted, too.

```
print(sorted(clean_james))
print(sorted(clean_julie))
print(sorted(clean_mikey))
print(sorted(clean_sarah))
```



Test DRIVE

Combine your `sanitize()` function with your amended code from the previous page, and then press F5 in IDLE to confirm the sorting is now working as expected.

```

Python Shell
RESTART
>>> -----
>>> -----
[ '2.01', '2.01', '2.22', '2.34', '2.34', '2.45', '3.01', '3.10', '3.21' ]
[ '2.11', '2.11', '2.23', '2.23', '2.59', '3.10', '3.10', '3.21', '3.21' ]
[ '2.22', '2.38', '2.49', '3.01', '3.01', '3.02', '3.02', '3.02', '3.22' ]
[ '2.18', '2.25', '2.39', '2.54', '2.55', '2.55', '2.55', '2.58', '2.58' ]
>>> |-----|-----|-----|-----|
Ln: 10 Col: 4

```

The sanitized data only contains '.' as a separator.

The ordering works, because all the times are now comparable.

Four sorted lists

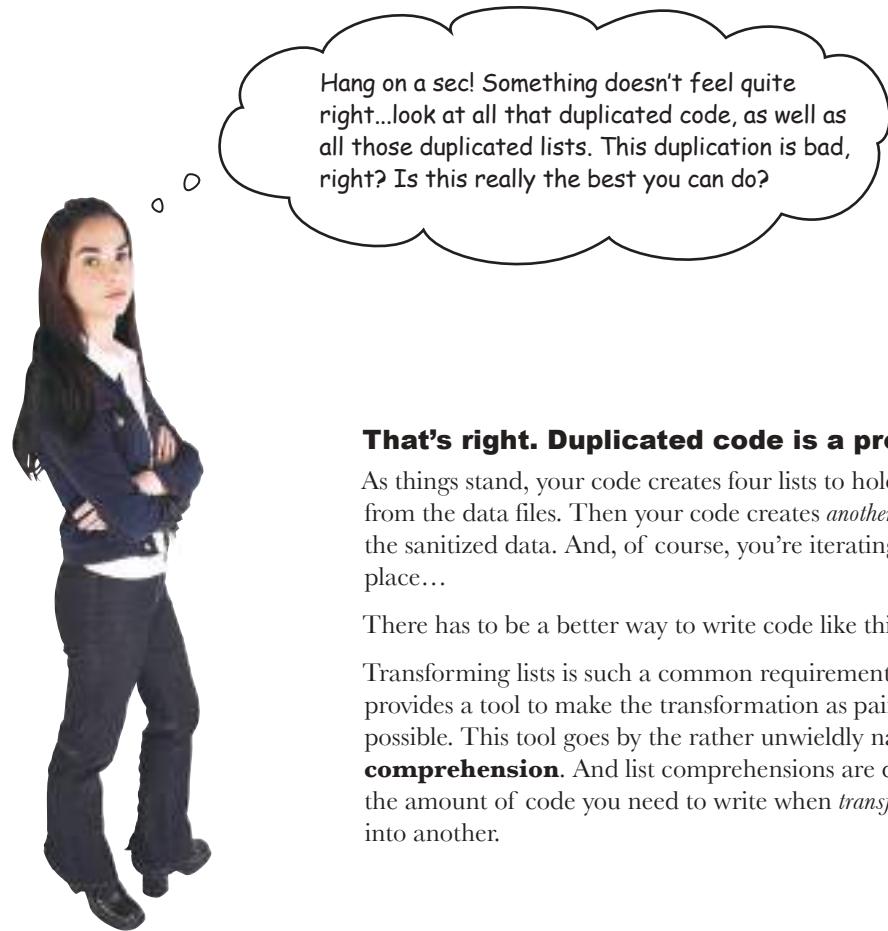
This output looks *much better*.

It's taken a bit of work, but now the data from each of the four files is both **sorted** and **uniformly formatted**. By preprocessing your data *before* you sort it, you've helped ensure Python's sorting technology performs correctly.



Geek Bits

By default, both the `sort()` method and the `sorted()` BIF order your data in **ascending** order. To order your data in **descending** order, pass the `reverse=True` argument to either `sort()` or `sorted()` and Python will take care of things for you.



That's right. Duplicated code is a problem.

As things stand, your code creates four lists to hold the data as read from the data files. Then your code creates *another* four lists to hold the sanitized data. And, of course, you're iterating all over the place...

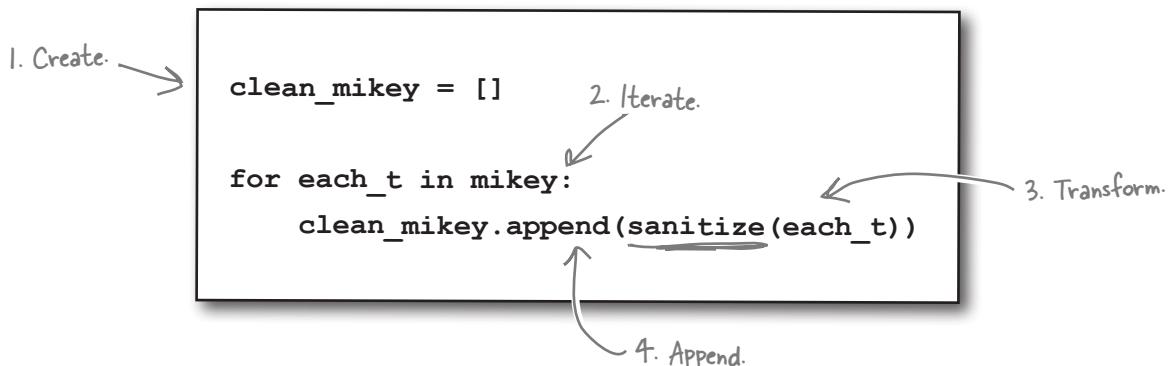
There has to be a better way to write code like this.

Transforming lists is such a common requirement that Python provides a tool to make the transformation as painless as possible. This tool goes by the rather unwieldy name of **list comprehension**. And list comprehensions are designed to reduce the amount of code you need to write when *transforming* one list into another.

Comprehending lists

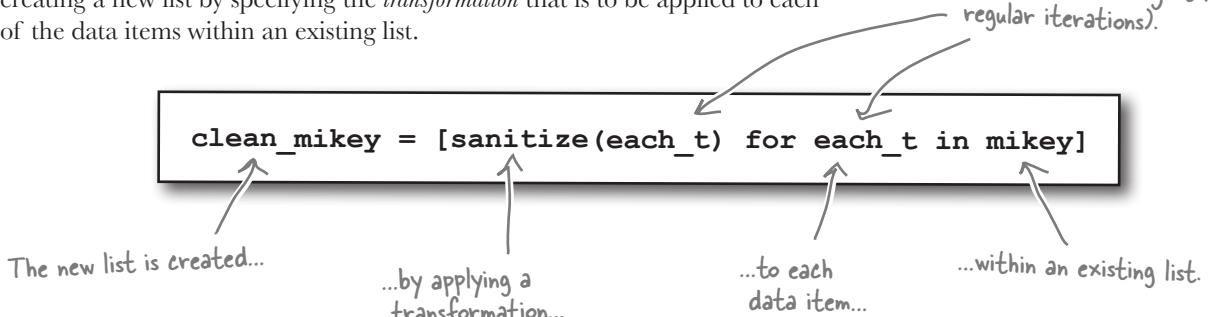
Consider what you need to do when you transform one list into another. Four things have to happen. You need to:

- 1 Create a new list to hold the transformed data.
- 2 Iterate each data item in the original list.
- 3 With each iteration, perform the transformation.
- 4 Append the transformed data to the new list.



Here's the same functionality as a **list comprehension**, which involves creating a new list by specifying the *transformation* that is to be applied to each of the data items within an existing list.

You get to pick the target identifier to use (just like with regular iterations).



What's interesting is that the transformation has been reduced to a *single line of code*. Additionally, there's no need to specify the use of the `append()` method as this action is implied within the list comprehension. Neat, eh?



An IDLE Session

Let's see some other list comprehension examples. Open up your IDLE shell and follow along with these one-liner transformations.

Start by transforming a list of minutes into a list of seconds:

```
>>> mins = [1, 2, 3]
>>> secs = [m * 60 for m in mins]
>>> secs
[60, 120, 180]
```

Simply multiply the minute values by 60.

How about meters into feet?

```
>>> meters = [1, 10, 3]
>>> feet = [m * 3.281 for m in meters]
>>> feet
[3.281, 32.81, 9.843]
```

Yes, there are 3.281 feet in a meter.

Given a list of strings in mixed and lowercase, it's a breeze to transform the strings to UPPERCASE:

```
>>> lower = ["I", "don't", "like", "spam"]
>>> upper = [s.upper() for s in lower]
>>> upper
['I', 'DON\'T', 'LIKE', 'SPAM']
```

Every string comes with the "upper()" method.

Let's use your `sanitize()` function to transform some list data into correctly formatted times:

```
>>> dirty = ['2-22', '2:22', '2.22']
>>> clean = [sanitize(t) for t in dirty]
>>> clean
['2.22', '2.22', '2.22']
```

It's never been so easy to turn something dirty into something clean. ☺

It's also possible to assign the results of the list transformation back onto the original target identifier. This example transforms a list of strings into floating point numbers, and then replaces the original list data:

```
>>> clean = [float(s) for s in clean]
>>> clean
[2.22, 2.22, 2.22]
```

The "float()" BIF converts to floating point.

And, of course, the transformation can be a function chain, if that's what you need:

```
>>> clean = [float(sanitize(t)) for t in ['2-22', '3:33', '4.44']]
>>> clean
[2.22, 3.33, 4.44]
```

Combining transformations on the data items is supported, too!



Now that you know about list comprehensions, let's write four of them to process the coach's four lists of timing values. Transform each of your lists into sorted, sanitized version of themselves. Grab your pencil and in the space provided, scribble the list comprehensions you plan to use.

.....
.....
.....
.....
.....
.....
.....
.....

^{there are no}
Dumb Questions

Q: So...let me get this straight: list comprehensions are good and list iterations are bad, right?

A: No, that's not the best way to look at it. If you have to perform a transformation on every item in a list, using a list comprehension is the way to go, especially when the transformation is easily specified on one line (or as a function chain). List iterations can do everything that list comprehensions can, they just take more code, but iterations do provide more flexibility should you need it.



Geek Bits

Python's list comprehension is an example of the language's support for *functional programming concepts*. There's plenty of debate about the **best way** to develop program code: either procedurally, using functional programming techniques, or using object orientation. At Head First Labs, we try not to get involved in this debate, other than to rejoice in the fact that Python supports, in one way or another, all three of these programming practices.



The list comprehension performs the transformation, and the new list is then ordered by the "sorted()" BIF.

Now that you know about list comprehensions, you were to write four of them to process the coach's four lists of timing values. You were to transform each of your lists into sorted, sanitized version of themselves. You were to grab your pencil and in the space provided, scribble the list comprehensions you plan to use.

`sorted([sanitize(t) for t in james])`

`sorted([sanitize(t) for t in julie])`

`sorted([sanitize(t) for t in mikey])`

`sorted([sanitize(t) for t in sarah])`

Rinse and repeat for the other lists.



Watch it!

Be careful about where you use the `sorted()` BIF when defining your list comprehensions.

You may have been tempted to use the function chain `sorted(sanitize(t))` within your list comprehension. Don't be. Recall that the transformation works on one list item at a time, **not the entire list**. In this example, the `sorted()` BIF expects to sort a list, not an individual data item.

The beauty of list comprehensions

The use of list comprehensions with the coach's athlete data has resulted in a lot less code for you to maintain. Additionally, as you get used to list comprehension syntax and usage, you'll find that their use is natural and matches the way your brain thinks about your data and the transformations that you might want to apply.

Let's confirm that your new code is working as expected.



Test DRIVE

Replace your list iteration code from earlier with your four new (beautiful) list comprehensions. Run your program to confirm that the results have not changed.

```

coach4.py - /Users/barryp/HeadFirstPython/chapter5/coach4.py

def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)

with open('james.txt') as jaf:
    data = jaf.readline()
james = data.strip()

with open('julie.txt') as julie:
    data = julie.readline()
julie = data.strip()

with open('mikey.txt') as mick:
    data = mick.readline()
mikey = data.strip()

with open('sarah.txt') as saf:
    data = saf.readline()
sarah = data.strip()

print(sorted([sanitize(t) for t in james]))
print(sorted([sanitize(t) for t in julie]))
print(sorted([sanitize(t) for t in mikey]))
print(sorted([sanitize(t) for t in sarah]))

```

Your new list comprehensions produce EXACTLY the same output as your earlier list iterations.

As expected, the output matches that from earlier.

You've written a program that reads Coach Kelly's data from his data files, stores his raw data in lists, sanitizes the data to a uniform format, and then sorts and displays the coach's data on screen. And all in 25 lines of code.

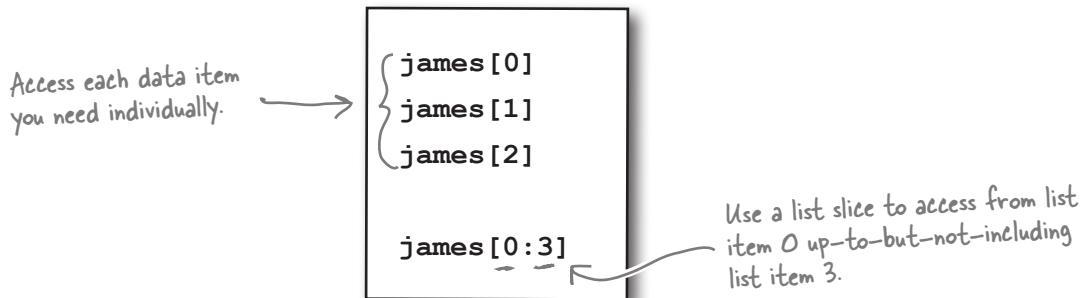
It's probably safe to let the coach take a look at your output now.

What will the coach think?



In your haste to sanitize and sort your data, you forgot to worry about what you were actually supposed to be doing: *producing the three fastest times for each athlete*. And, of course, there's no place for any duplicated times in your output.

Accessing the first three data items from any list is easy. Either specify each list item *individually* using the standard notation or use a **list slice**:



But...what about removing duplicates from your list?

Iterate to remove duplicates

Processing a list to remove duplicates is one area where a list comprehension can't help you, because duplicate removal is *not* a transformation; it's more of a *filter*. And a duplicate removal filter needs to examine the list being created as *it is being created*, which is not possible with a list comprehension.

To meet this new requirement, you'll need to revert to regular list iteration code.



Assume that the fourth from last line of code from your current program is changed to this:

```
james = sorted([sanitize(t) for t in james])
```

That is, instead of printing the sanitized and sorted data for James to the screen, this line of code replaces James's unordered and nonuniform data with the sorted, sanitized copy.

Your next task is to write some code to remove any duplicates from the `james` list produced by the preceding line of code. Start by creating a new list called `unique_james`, and then populate it with the unique data items found in `james`. Additionally, provide code to display only the top three fastest times for James.

Hint: you might want to consider using the `not in` operator.



Exercise Solution

Assume that the fourth from last line of code from your current program is changed to this:

```
james = sorted([sanitize(t) for t in james])
```

That is, instead of printing the sanitized and sorted data for James to the screen, this line of code replaces James's unordered and non-uniform data with the sorted, sanitized copy.

Your next task was to write some code to remove any duplicates from the `james` list produced by the preceding line of code. You were to start by creating a new list called `unique_james` and then populate it with the unique data items found in `james`. Additionally, you were to provide code to only display the top three fastest times for James.

Create the empty list to hold the unique data items.

→ `unique_james = []`

Iterate over the existing data...

`for each_t in james:`

...and if the data item ISN'T already in the new list...

`if each_t not in unique_james:`

...append the unique data item to the new list.

`unique_james.append(each_t)`

Slice the first three data items from the list and display them on screen.

→ `print(unique_james[0:3])`



Do this!



Repeat the code on this page for the rest of the coach's lists: julie, mikey & sarah. Add all of your new code to your existing program.



Test DRIVE

Take all of the recent amendments and apply them to your program. Run this latest code within IDLE when you are ready.

```

coach5.py - /Users/barryp/HeadFirstPython/chapter5/coach5.py

james = sorted([sanitize(t) for t in james])
julie = sorted([sanitize(t) for t in julie])
mikey = sorted([sanitize(t) for t in mikey])
sarah = sorted([sanitize(t) for t in sarah])

unique_james = []
for each_t in james:
    if each_t not in unique_james:
        unique_james.append(each_t)

print(unique_james[0:3])

unique_julie = []
for each_t in julie:
    if each_t not in unique_julie:
        unique_julie.append(each_t)

print(unique_julie[0:3])

unique_mikey = []
for each_t in mikey:
    if each_t not in unique_mikey:
        unique_mikey.append(each_t)

print(unique_mikey[0:3])

unique_sarah = []
for each_t in sarah:
    if each_t not in unique_sarah:
        unique_sarah.append(each_t)

print(unique_sarah[0:3])

```

The code above is annotated with arrows pointing to specific parts of the code:

- An arrow points to the first four lines (defining sorted lists) with the text "Sort and sanitize each list."
- An arrow points to the two nested loops (for each_t in ...: if each_t not in ...: unique_... .append(each_t)) with the text "Remove the duplicates."
- An arrow points to the final print statement for the sarah list with the text "Looking good!"

Python Shell

```

>>> -----
RESTART
>>>
pri
[['2.01', '2.22', '2.34'],
 ['2.11', '2.23', '2.59'],
 ['2.22', '2.38', '2.49'],
 ['2.18', '2.25', '2.39']]
>>> |

```

Ln: 69 Col: 4

It worked!

You are now displaying only the top three times for each athlete, and the duplicates have been successfully removed.

The list iteration code is what you need in this instance. There's a little bit of duplication in your code, but it's not too bad, is it?



"Not too bad"...you're kidding, right?!?
Surely there's something that can be done
with all that duplicated duplicate code?

The irony is hard to avoid, isn't it?

The code that removes duplicates from your lists is itself duplicated.

Sometimes such a situation is unavoidable, and sometimes creating a small function to factor out the duplicated code can help. But something still doesn't feel quite right here...



Remove duplicates with sets

In addition to lists, Python also comes with the **set** data structure, which behaves like the sets you learned all about in math class.

The overriding characteristics of sets in Python are that the data items in a set are *unordered* and *duplicates are not allowed*. If you try to add a data item to a set that already contains the data item, Python simply ignores it.

Create an empty set using the `set()` BIF, which is an example of a *factory function*:

Create a new, empty set, and assign it to a variable.

```
distances = set()
```

It is also possible to *create* and *populate* a set in one step. You can provide a list of data values between curly braces or specify an existing list as an argument to the `set()` BIF, which is the *factory function*:

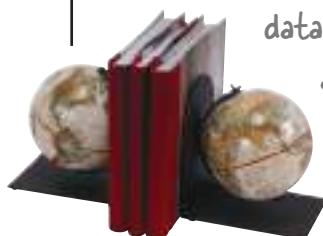
```
distances = {10.6, 11, 8, 10.6, "two", 7}
```

Any duplicates in the supplied list of data values are ignored.

Any duplicates in the "james" list are ignored. Cool.

```
distances = set(james)
```

the Scholar's Corner



Factory Function: A factory function is used to make new data items of a particular type. For instance, "`set()`" is a factory function because it makes a new set. In the real world, factories make things, hence the name.



Fireside Chats



Tonight's talk: Does list suffer from set envy?

List:

[sings] “Anything you can do, I can do better. I can do anything better than you.”

Can you spell “d-a-t-a l-o-s-s”? Getting rid of data automatically sounds kinda *dangerous* to me.

Seriously?

And that’s *all* you do?

And they pay you for that?!?

Have you ever considered that I like my duplicate values. I’m very fond of them, you know.

Which isn’t very often. And, anyway, I can always rely on the kindness of others to help me out with any duplicates that I don’t need.

Set:

I’m resisting the urge to say, “No, you can’t.” Instead, let me ask you: what about handling duplicates? When I see them, I throw them away *automatically*.

But that’s what I’m supposed to do. Sets aren’t allowed duplicate values.

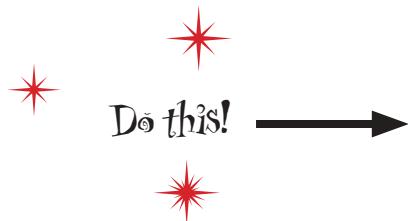
Yes. That’s why I exist...to store sets of values. Which, when it’s needed, is a real lifesaver.

That’s all I need to do.

Very funny. You’re just being smug in an effort to hide from the fact that you can’t get rid of duplicates on your own.

Yeah, right. Except when you *don’t* need them.

I think you meant to say, “the kindness of `set()`”, didn’t you?



To extract the data you need, replace all of that list iteration code in your current program with four calls to `sorted(set(...)) [0:3]`.



Head First Code Review

The Head First Code Review Team has taken your code and annotated it in the only way they know how: *they've scribbled all over it*. Some of their comments are confirmations of what you might already know. Others are suggestions that might make your code better. Like all code reviews, these comments are an attempt to improve the quality of your code.

I think we
can make a few
improvements here.

```
def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)
```

A comment would
be nice to have
here.



Meet the Head
First Code Review
Team.

What happens
if one of these
files is missing?!

Where's your
exception handling
code?

There's a lot
going on here,
but we find it's
not too hard to
understand if you
read it from the
inside out.

```
with open('james.txt') as jaf:
    data = jaf.readline()
james = data.strip().split(',')
with open('julie.txt') as juf:
    data = juf.readline()
julie = data.strip().split(',')
with open('mikey.txt') as mif:
    data = mif.readline()
mikey = data.strip().split(',')
with open('sarah.txt') as saf:
    data = saf.readline()
sarah = data.strip().split(',')
```

There's a bit of duplication here. You
could factor out the code into a small
function; then, all you need to do is call
the function for each of your athlete
data files, assigning the result to an
athlete list.

```
print(sorted(set([sanitize(t) for t in james]))[0:3])
print(sorted(set([sanitize(t) for t in julie]))[0:3])
print(sorted(set([sanitize(t) for t in mikey]))[0:3])
print(sorted(set([sanitize(t) for t in sarah]))[0:3])
```

Ah, OK. We get it.
The slice is applied to
the list produced by
"sorted()", right?



Let's take a few moments to implement the review team's suggestion to turn those four `with` statements into a function. Here's the code again. In the space provided, create a function to abstract the required functionality, and then provide one example of how you would call your new function in your code:

```
with open('james.txt') as jaf:
    data = jaf.readline()
james = data.strip().split(',')

with open('julie.txt') as juf:
    data = juf.readline()
julie = data.strip().split(',')

with open('mikey.txt') as mif:
    data = mif.readline()
mikey = data.strip().split(',')

with open('sarah.txt') as saf:
    data = saf.readline()
sarah = data.strip().split(',')
```

Write your new function here.



.....
.....
.....
.....
.....
.....

Provide one example call.





Exercise Solution

You were to take a few moments to implement the review team's suggestion to turn those four with statements into a function. In the space provided, you were to create a function to abstract the required functionality, then provide one example of how you would call your new function in your code:

```
with open('james.txt') as jaf:
    data = jaf.readline()
james = data.strip().split(',')

with open('julie.txt') as juf:
    data = juf.readline()
julie = data.strip().split(',')

with open('mikey.txt') as mif:
    data = mif.readline()
mikey = data.strip().split(',')

with open('sarah.txt') as saf:
    data = saf.readline()
sarah = data.strip().split(',')
```

Create a new function.

`def get_coach_data(filename):`

Accept a filename as the sole argument.

`try:`

Open the file, and read the data.

`with open(filename) as f:`

`data = f.readline()`

Perform the split/strip trick on the data prior to returning it to the calling code.

`return(data.strip().split(','))`

Add the suggested exception-handling code.

`except IOError as ioerr:`

Tell your user about the error (if it occurs) and return "None" to indicate failure.

`print('File error: ' + str(ioerr))`

`return(None)`

Calling the function is straightforward.

`sarah = get_coach_data('sarah.txt')`

Provide the name of the file to process.



Test DRIVE

It's time for one last run of your program to confirm that your use of sets produces the same results as your list-iteration code. Take your code for a spin in IDLE and see what happens.

```

coach5.py - /Users/barryp/HeadFirstPython/chapter5/coach5.py

def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)

with open('james.txt') as jaf:
    data = jaf.readline()
james = data.strip().split(',')

```

Python Shell

```

>>> -----
>>>
>>> ['2.01', '2.22', '2.34']
['2.11', '2.23', '2.59']
['2.22', '2.38', '2.49']
['2.18', '2.25', '2.39']
>>> |

```

```

sarah = data.strip().split(',')

print(sorted(set([sanitize(t) for t in james]))[0:3])
print(sorted(set([sanitize(t) for t in julie]))[0:3])
print(sorted(set([sanitize(t) for t in mikey]))[0:3])
print(sorted(set([sanitize(t) for t in sarah]))[0:3])

```

As expected, your latest code does the business. Looking good!

Excellent!

You've processed the coach's data perfectly, while taking advantage of the `sorted()` BIF, *sets*, and *list comprehensions*. As you can imagine, you can apply these techniques to many different situations. You're well on your way to becoming a Python *data-munging master*!



That's great work, and just what I need. Thanks! I'm looking forward to seeing you on the track soon...



Your Python Toolbox

You've got Chapter 5 under your belt and you've added some more Python techniques to your toolbox.

Python Lingo

- “In-place” sorting – transforms and then replaces.
- “Copied” sorting – transforms and then returns.
- “Method Chaining” – reading from left to right, applies a collection of methods to data.
- “Function Chaining” – reading from right to left, applies a collection of functions to data.

More Python Lingo

- “List Comprehension” – specify a transformation on one line (as opposed to using an iteration).
- A “slice” – access more than one item from a list.
- A “set” – a collection of unordered data items that contains no duplicates.



BULLET POINTS

- The `sort()` method changes the ordering of lists *in-place*.
- The `sorted()` BIF sorts most any data structure by providing *copied sorting*.
- Pass `reverse=True` to either `sort()` or `sorted()` to arrange your data in **descending** order.
- When you have code like this:

```
new_l = []
for t in old_l:
    new_l.append(len(t))
```

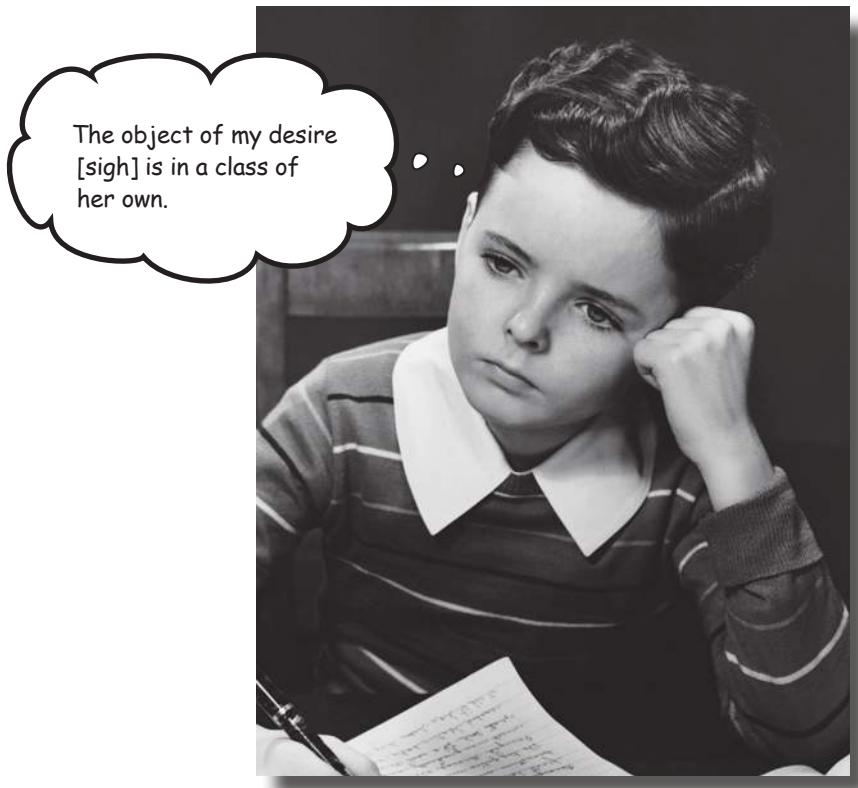
rewrite it to use a **list comprehension**, like this:

```
new_l = [len(t) for t
in old_l]
```
- To access more than one data item from a list, use a slice. For example:
`my_list[3:6]` accesses the items from index location 3 up-to-but-not-including index location 6.
- Create a **set** using the `set()` factory function.

6 custom data objects



Bundling code with data



It's important to match your data structure choice to your data.

And that choice can make a big difference to the complexity of your code. In Python, although really useful, lists and sets aren't the only game in town. The Python **dictionary** lets you organize your data for speedy lookup by *associating your data with names*, not numbers. And when Python's built-in data structures don't quite cut it, the Python **class** statement lets you define your own. This chapter shows you how.

Coach Kelly is back (with a new file format)



The output from your last program in Chapter 5 was exactly what the coach was looking for, but for the fact that no one can tell which athlete belongs to which data. Coach Kelly thinks he has the solution: he's added identification data to each of his data files:

Sarah Sweeney, 2002-6-17, 2:58, 2.58, 2:39, 2-25, 2-55, 2:54, 2.18, 2:55, 2:55, 2:22, 2-21, 2.22

Sarah's full name

Sarah's date of birth

Sarah's timing data

This is "sarah2.txt", with extra data added.

If you use the `split()` BIF to extract Sarah's data into a list, the first data item is Sarah's name, the second is her date of birth, and the rest is Sarah's timing data.

Let's exploit this format and see how well things work.

Grab the updated files from the *Head First Python* website.



Code Magnets

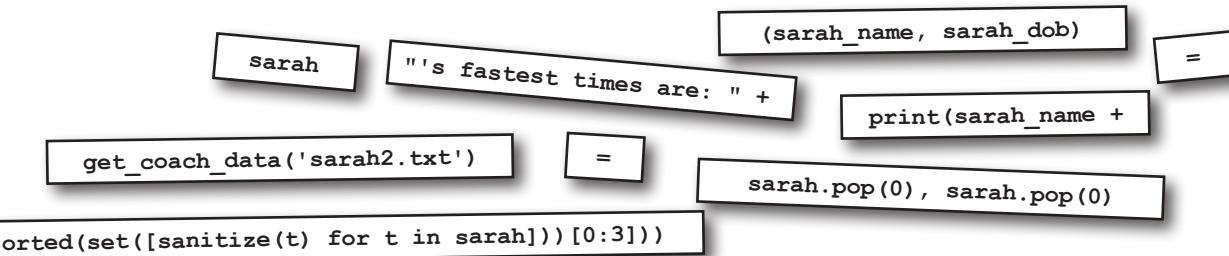
Let's look at the code to implement the strategy outlined at the bottom of the previous page. For now, let's concentrate on Sarah's data. Rearrange the code magnets at the bottom of this page to implement the list processing required to extract and process Sarah's three fastest times from Coach Kelly's raw data.

Hint: the `pop()` method removes and returns a data item from the specified list location.

```
def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)

def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readline()
        return(data.strip().split(','))
    except IOError as ioerr:
        print('File error: ' + str(ioerr))
        return(None)
```

Rearrange the
magnets here.





Code Magnets Solution

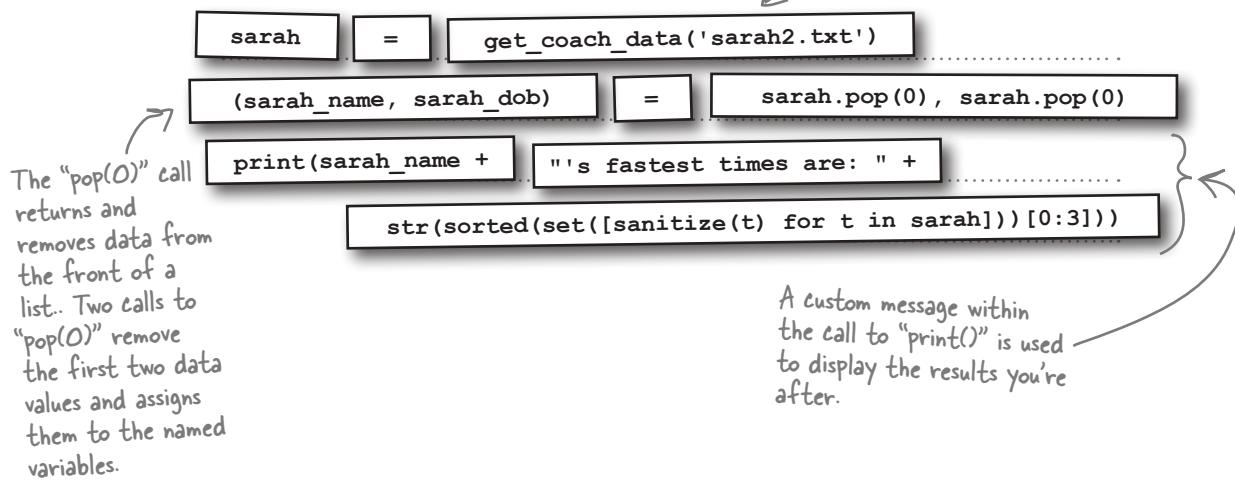
Let's look at the code to implement the strategy outlined earlier. For now, let's concentrate on Sarah's data.

You were to rearrange the code magnets at the bottom of the previous page to implement the list processing required to extract and process Sarah's three fastest times from Coach Kelly's raw data.

```
def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)
```

```
def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readline()
        return(data.strip().split(','))
    except IOError as ioerr:
        print('File error: ' + str(ioerr))
        return(None)
```

Use the function to turn
Sarah's data file into a list,
and then assign it to the
"sarah" variable.





Test DRIVE

Let's run this code in IDLE and see what happens.

```

coach2.py - /Users/barnyp/HeadFirstPython/chapter6/coach2.py
def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)

def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readlines()
        return(data.strip().split(','))
    except IOError as ioerr:
        print('File error: ' + str(ioerr))
        return(None)

sarah = get_coach_data('sarah2.txt')
(sarah_name, sarah_dob) = sarah.pop(0), sarah.pop(0)
print(sarah_name + "'s fastest times are: " +
      str(sorted(set([sanitize(t) for t in sarah]))[0:3]))
|
```

Your latest code ↗

This output ↘ is much more understandable.

```

Python Shell
Python 3.1.2 (r312:79360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
Sarah Sweeney's fastest times are: ['2.18', '2.21', '2.22']
>>> |
```

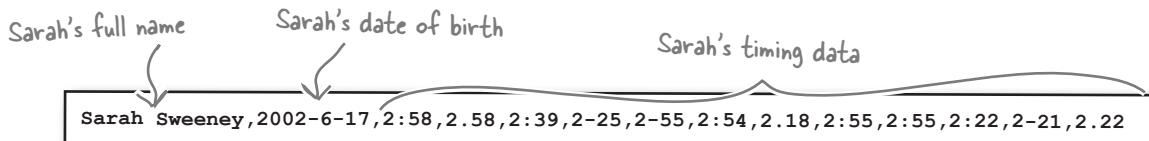
Ln: 7 Col: 4

This program works as expected, and is *fine*...except that you have to name and create Sarah's three variables in such a way that it's possible to identify *which* name, date of birth, and timing data relate to Sarah. And if you add code to process the data for James, Julie, and Mikey, you'll be up to 12 variables that need juggling. This just about works for now with four athletes. But what if there are 40, 400, or 4,000 athletes to process?

Although the data is related in “real life,” within your code things are **disjointed**, because the three related pieces of data representing Sarah are stored in *three separate variables*.

Use a dictionary to associate data

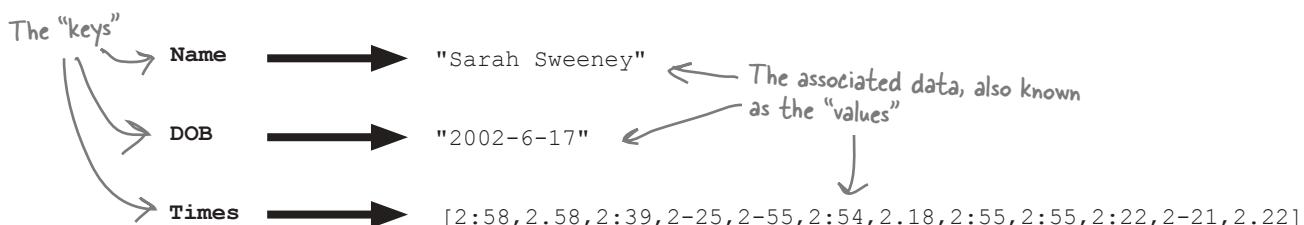
Lists are *great*, but they are not always the best data structure for every situation. Let's take another look at Sarah's data:



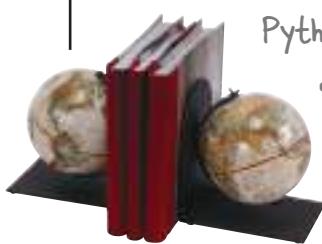
There's a definite **structure** here: the athlete's name, the date of birth, and then the list of times.

Let's continue to use a list for the timing data, because that still makes sense. But let's make the timing data part of *another* data structure, which associates all the data for an athlete with a single variable.

We'll use a Python **dictionary**, which *associates data values with keys*:



the Scholar's Corner



Dictionary A built-in data structure (included with Python) that allows you to associate data with keys, as opposed to numbers. This lets your in-memory data closely match the structure of your actual data.



Fireside Chats



Tonight's talk: **To use a list or not to use a list?**

Dictionary:

Hi there, List. I hear you're great, but not always the best option for complex data. That's where I come in.

True. But when you do, you lose any structure associated with the data you are processing.

Isn't it always?

You *guess* so? When it comes to modeling your data in code, it's best not to guess. Be firm. Be strong. Be assertive. Use a dictionary.

[laughs] Oh, I do love your humor, List, even when you know you're on thin ice. Look, the rule is simple: *if your data has structure, use a dictionary, not a list.* How hard is that?

Which *rarely* makes sense. Knowing when to use a list and when to use a dictionary is what separates the good programmers from the great ones, right?

List:

What?!? Haven't you heard? You can put *anything* into a list, anything at all.

Well...assuming, of course, that structure is important to you.

Ummmm, uh...I guess so.

That sounds like a slogan from one of those awful self-help conferences. Is that where you heard it?

Not that hard, really. Unless, of course, you are a list, and you miss being used for every piece of data in a program...

I guess so. Man, I do hate it when you're right!



Geek Bits

The Python **dictionary** is known by different names in other programming languages. If you hear other programmers talking about a "mapping," a "hash," or an "associative array," they are talking about a "dictionary."



An IDLE Session

Let's see the Python dictionary in action. Follow along with this IDLE session on your computer, ensuring that you get the same results as shown.

Start by creating two empty dictionaries, one using curly braces and the other using a factory function:

```
>>> cleese = {}
>>> palin = dict()
>>> type(cleese)
<class 'dict'>
>>> type(palin)
<class 'dict'>
```

Both techniques create an empty dictionary, as confirmed.

Add some data to both of these dictionaries by *associating values with keys*. Note the actual structure of the data is presenting itself here, as each dictionary has a Name and a list of Occupations. Note also that the palin dictionary is being created at the same time:

```
>>> cleese['Name'] = 'John Cleese'
>>> cleese['Occupations'] = ['actor', 'comedian', 'writer', 'film producer']
>>> palin = {'Name': 'Michael Palin', 'Occupations': ['comedian', 'actor', 'writer', 'tv']}
```

With your data associated with keys (which are strings, in this case), it is possible to access an individual data item using a notation similar to that used with lists:

```
>>> palin['Name'] ← Use square brackets to index into the dictionary to access data items, but instead of numbers, index with keys.
'Michael Palin'
>>> cleese['Occupations'][-1] ← Use numbers to access a list item stored at a particular dictionary key. Think of this as "index-chaining" and read from right to left: "...the last item of the list associated with Occupations..."
```

As with lists, a Python dictionary can grow dynamically to store additional key/value pairings. Let's add some data about birthplace to each dictionary:

```
>>> palin['Birthplace'] = "Broomhill, Sheffield, England" ← Provide the data associated with the new key.
>>> cleese['Birthplace'] = "Weston-super-Mare, North Somerset, England"
```

Unlike lists, a Python dictionary **does not maintain insertion order**, which can result in some unexpected behavior. The key point is that the dictionary *maintains the associations*, not the ordering:

```
>>> palin
{'Birthplace': 'Broomhill, Sheffield, England', 'Name': 'Michael Palin', 'Occupations':
['comedian', 'actor', 'writer', 'tv']}
>>> cleese
{'Birthplace': 'Weston-super-Mare, North Somerset, England', 'Name': 'John Cleese',
'Occupations': ['actor', 'comedian', 'writer', 'film producer']}
```

The ordering maintained by Python is different from how the data was inserted. Don't worry about it; this is OK.



It's time to apply what you now know about Python's dictionary to your code. Let's continue to concentrate on Sarah's data for now. Strike out the code that you no longer need and replace it with new code that uses a dictionary to hold and process Sarah's data.

```
def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)

def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readline()
        return(data.strip().split(','))
    except IOError as ioerr:
        print('File error: ' + str(ioerr))
        return(None)

sarah = get_coach_data('sarah2.txt')
(sarah_name, sarah_dob) = sarah.pop(0), sarah.pop(0)
print(sarah_name + "'s fastest times are: " +
      str(sorted(set([sanitize(t) for t in sarah]))[0:3]))
```

*Strike out the code
you no longer need.*

*Add your dictionary
using and processing
code here.*



It's time to apply what you now know about Python's dictionary to your code. Let's continue to concentrate on Sarah's data for now. You were to strike out the code that you no longer needed and replace it with new code that uses a dictionary to hold and process Sarah's data.

```
def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)
```

```
def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readline()
        return(data.strip().split(','))
    except IOError as ioerr:
        print('File error: ' + str(ioerr))
        return(None)
```

You don't need this code anymore.

```
sarah = get_coach_data('sarah2.txt')
(sarah_name, sarah_dob) = sarah.pop(0), sarah.pop(0)
print(sarah_name + "'s fastest times are: " +
      str(sorted(set([sanitize(t) for t in sarah])))[:3]))
```

Create an empty dictionary.

```
sarah_data = {}
```

```
sarah_data['Name'] = sarah.pop(0)
```

```
sarah_data['DOB'] = sarah.pop(0)
```

```
sarah_data['Times'] = sarah
```

Populate the dictionary with the data by associating the data from the file with the dictionary keys..

```
print(sarah_data['Name'] + "'s fastest times are: " +
```

```
      str(sorted(set([sanitize(t) for t in sarah_data['Times']]))[:3]))
```

Refer to the dictionary when processing the data.



Test DRIVE

Let's confirm that this new version of your code works exactly as before by testing your code within the IDLE environment.

```

coach3.py - /Users/barryp/HeadFirstPython/chapter6/coach3.py

def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)

def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readline()
        return(data.strip().split(','))
    except IOError as ioerr:
        print('File error: ' + str(ioerr))
        return(None)

sarah = get_coach_data('sarah2.txt')
sarah_data = {}
sarah_data['Name'] = sarah.pop(0)
sarah_data['DOB'] = sarah.pop(0)
sarah_data['Times'] = sarah

print(sarah_data['Name'] + "'s fastest times are: " +
      str(sorted(set([sanitize(t) for t in sarah_data['Times']]))[0:3]))

```

Your dictionary code produces the same results as earlier.

Python Shell

```

>>> -----
RESTART
>>>
Sarah Sweeney's faatest times are: ['2.18', '2.21', '2.22']
>>> |

```

Ln: 43 Col: 4

Which, again, works as expected...the difference being that you can now more easily determine and control which identification data **associates** with which timing data, because they are *stored in a single dictionary*.

Although, to be honest, it does take *more* code, which is a bit of a bummer. Sometimes the extra code is worth it, and sometimes it isn't. In this case, it most likely is.

Let's review your code to see if we can improve anything.



Head First Code Review

The Head First Code Review Team has been at it again: *they've scribbled all over your code*. Some of their comments are confirmations; others are suggestions. Like all code reviews, these comments are an attempt to improve the quality of your code.

```
def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)
```

```
def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readlines()
        return(data.strip().split(','))
```

except IOError as ioerr:

```
    print('File error: ' + str(ioerr))
    return(None)
```

```
sarah = get_coach_data('sarah2.txt')
```

```
sarah_data = {}
sarah_data['Name'] = sarah.pop(0)
sarah_data['DOB'] = sarah.pop(0)
sarah_data['Times'] = sarah
```

```
print(sarah_data['Name'] + "'s fastest times are: " +
      str(sorted(set([sanitize(t) for t in sarah_data['Times']]))[0:3]))
```

You might want to consider moving this code into the `get_coach_data()` function, too, because doing so would rather nicely abstract away these processing details. But whether you do or not is up to you. It's your code, after all!

It's great to see
you taking some of our
suggestions on board.
Here are a few more...



Rather than building the dictionary as you go along, why not do it all in one go? In fact, in this situation, it might even make sense to do this processing within the `get_coach_data()` function and have the function return a populated dictionary as opposed to a list.

Then, all you need to do is create the dictionary from the data file using an appropriate function call, right?



Actually, those review comments are really useful. Let's take the time to apply them to your code. There are four suggestions that you need to adjust your code to support:

1. Create the dictionary all in one go.
 2. Move the dictionary creation code into the `get_coach_data()` function, returning a dictionary as opposed to a list.
 3. Move the code that determines the top three times for each athlete into the `get_coach_data()` function.
 4. Adjust the invocations within the main code to the new version of the `get_coach_data()` function to support its new mode of operation.

Grab your pencil and write your new `get_coach_data()` function in the space provided below. Provide the four calls that you'd make to process the data for each of the athletes and provide four amended `print()` statements:



You were to take the time to apply the code review comments to your code. There were four suggestions that you needed to adjust your code to support:

1. Create the dictionary all in one go.
 2. Move the dictionary creation code into the `get_coach_data()` function, returning a dictionary as opposed to a list.
 3. Move the code that determines the top three times for each athlete into the `get_coach_data()` function.
 4. Adjust the invocations within the main code to the new version of the `get_coach_data()` function to support its new mode of operation.

You were to grab your pencil and write your new `get_coach_data()` function in the space provided below, as well as provide the four calls that you'd make to process the data for each of the athletes and provide four amended `print()` statements:

```
def get_coach_data(filename):
```

try:

with open(filename) as f:

```
data = f.readline()
```

1. Create a temporary list to hold the data BEFORE creating the dictionary all in one go.

→ templ = data.strip().split(',')

```
return({'Name': templ.pop(0),
```

'DOB' : templ.pop(0),

```
'Times': str(sorted(set([sanitize(t) for t in templ]))[0:3]))})
```

```
except IOError as ioerr:
```

```
print('File error: ' + str(ioerr))
```

```
return(None)
```

2. The dictionary creation code is now part of the function.

3. The code that determines the top three scores is part of the function, too.

4. Call the function for an athlete and adjust the "print()" statement as needed.

```
james = get_coach_data('james2.txt')
```

We are showing only these two lines of code for one athlete (because repeating it for the other three is a trivial exercise).

```
> print(james['Name'] + "'s fastest times are: " + james['Times'])
```



Test DRIVE

Let's confirm that *all* of the re-factoring suggestions from the Head First Code Review Team are working as expected. Load your code into IDLE and take it for a spin.

```

def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)

def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readline()
            templ = data.strip().split(',')
        return({'Name' : templ.pop(0),
               'DOB' : templ.pop(0),
               'Times' : str(sorted(set([sanitize(t) for t in templ]))[0:3])})
    except IOError as ioerr:
        print('File error: ' + str(ioerr))
        return(None)

james = get_coach_data('james2.txt')
julie = get_coach_data('julie2.txt')
mikey = get_coach_data('mikey2.txt')
sarah = get_coach_data('sarah2.txt')

print(james['Name'] + "'s fastest times are: " + james['Times'])
print(julie['Name'] + "'s fastest times are: " + julie['Times'])
print(mikey['Name'] + "'s fastest times are: " + mikey['Times'])
print(sarah['Name'] + "'s fastest times are: " + sarah['Times'])

```

All of the data processing is moved into the function.

This code has been considerably tidied up and now displays the name of the athlete associated with their times.

Python Shell

```

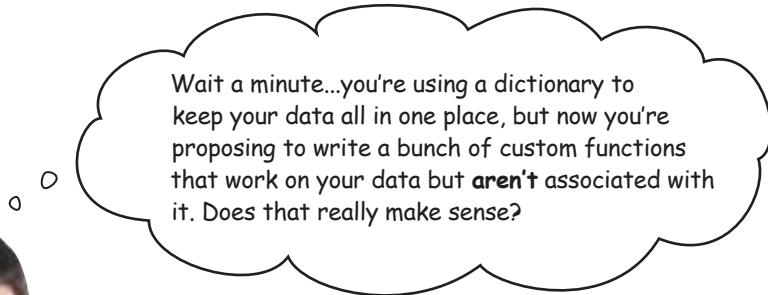
>>> -----
>>> RESTART -----
>>>
James Lee's fastest times are: ['2.01', '2.16', '2.22']
Julie Jones's fastest times are: ['2.11', '2.23', '2.59']
Mikey McManus's fastest times are: ['2.22', '2.31', '2.38']
Sarah Sweeney's fastest times are: ['2.18', '2.21', '2.22']
>>> |

```

Looking good!

To process additional athletes, all you need is two lines of code: the first invokes the `get_coach_data()` function and the second invokes `print()`.

And if you require additional functionality, it's no big deal to write more functions to provide the required functionality, is it?



Keeping your code and its data together is good.

It does indeed make sense to try and associate the functions with the data they are meant to work on, doesn't it? After all, the functions are only going to make sense when *related* to the data—that is, the functions will be *specific* to the data, not general purpose. Because this is the case, it's a great idea to try and bundle the code with its data.

But how? Is there an easy way to associate custom code, in the form of functions, with your custom data?

Bundle your code and its data in a class

Like the majority of other modern programming languages, Python lets you create and define an object-oriented **class** that can be used to *associate code with the data that it operates on*.



Why would anyone want to do this?

Using a class helps reduce complexity.

By associating your code with the data it works on, you reduce complexity as your code base grows.



So what's the big deal with that?



Reduced complexity means fewer bugs.

Reducing complexity results in fewer bugs in your code. However, it's a fact of life that your programs will have functionality added over time, which will result in additional complexity. Using classes to **manage** this complexity is a *very good thing*.



Yeah? But...who really cares?

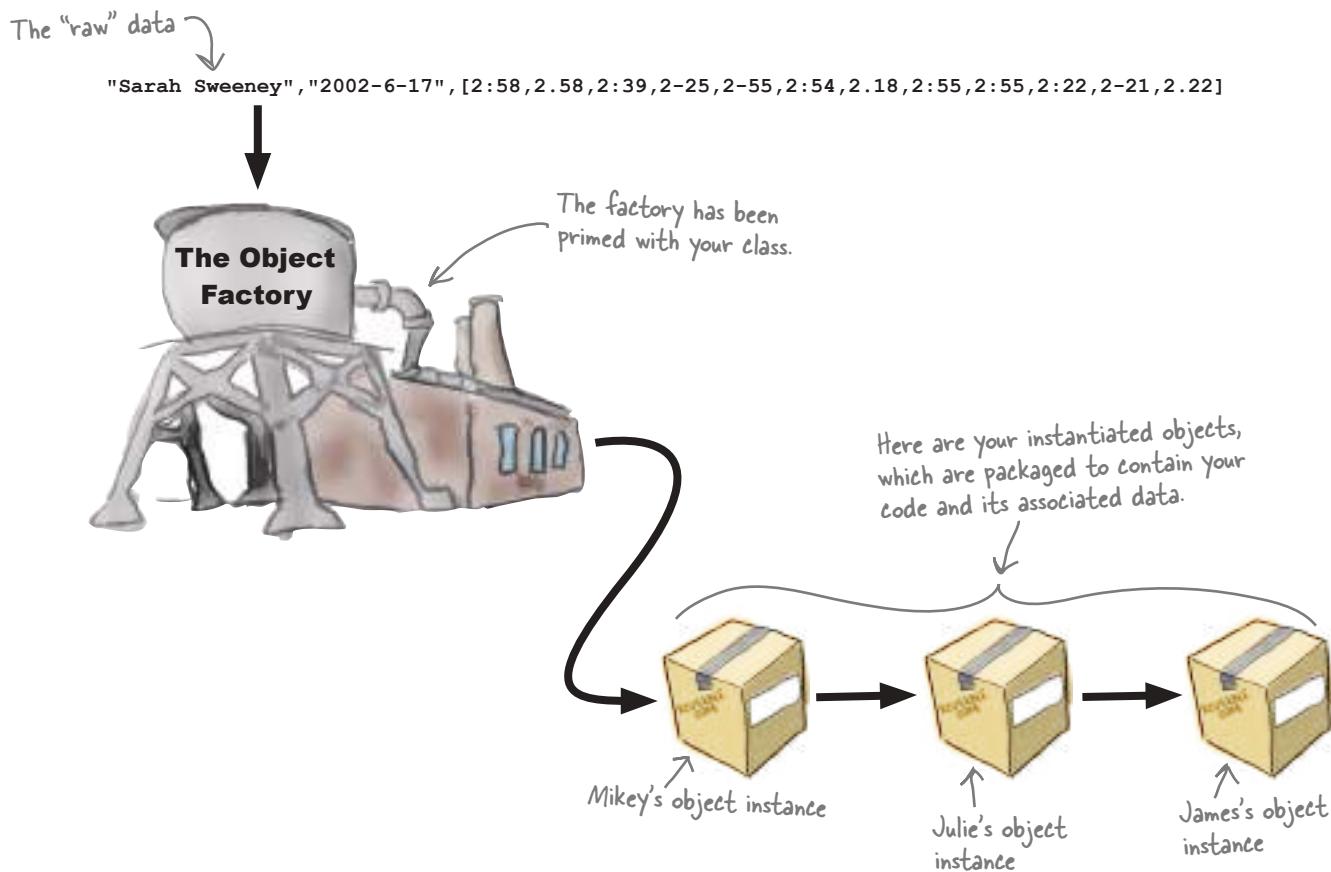
Fewer bugs means more maintainable code.

Using classes lets you keep your code and your data together in one place, and as your code base grows, this really can make quite a difference. Especially when it's 4 AM and you're under a deadline...

Define a class

Python follows the standard object-oriented programming model of providing a means for you to define the code and the data it works on as a *class*. Once this definition is in place, you can use it to create (or *instantiate*) **data objects**, which inherit their characteristics from your class.

Within the object-oriented world, your code is often referred to as the class's **methods**, and your data is often referred to as its **attributes**. Instantiated data objects are often referred to as **instances**.



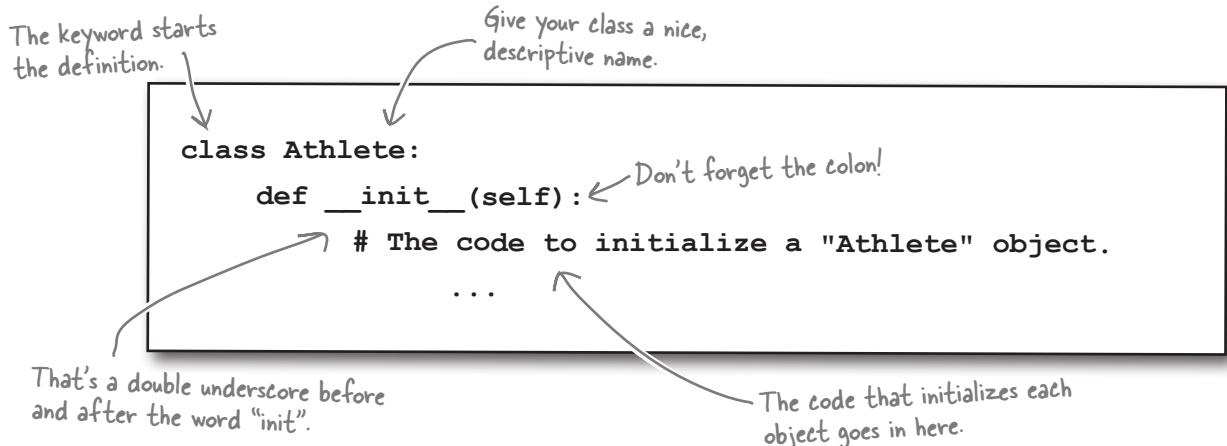
Each object is **created from** the class and **shares** a similar set of characteristics. The methods (your code) are the *same* in each instance, but each object's attributes (your data) *differ* because they were created from your raw data.

Let's look at how classes are defined in Python.

Use class to define classes

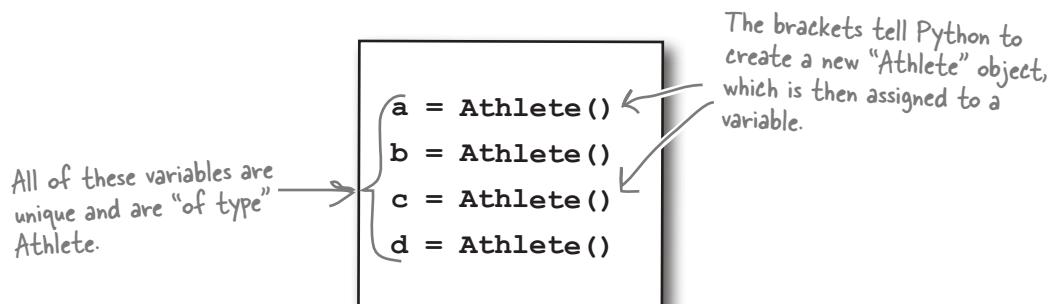
Python uses **class** to create objects. Every defined class has a *special method* called `__init__()`, which allows you to control how objects are initialized.

Methods within your class are defined in much the same way as functions, that is, using `def`. Here's the basic form:



Creating object instances

With the class in place, it's easy to create object instances. Simply assign a *call* to the class name to each of your variables. In this way, the class (together with the `__init__()` method) provides a mechanism that lets you create a **custom factory function** that you can use to create as many object instances as you require:



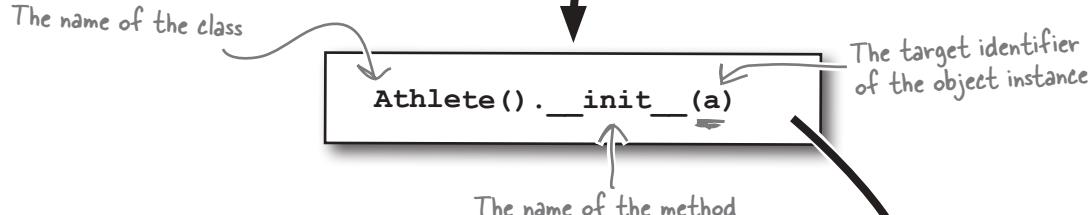
Unlike in C++-inspired languages, Python has no notion of defining a constructor called "new." Python does object construction for you, and then lets you customize your object's initial state using the `__init__()` method.

The importance of self

To confirm: when you define a class you are, in effect, defining a *custom factory function* that you can then use in your code to create instances:



When Python processes this line of code, it turns the factory function call into the following call, which identifies the *class*, the *method* (which is automatically set to `__init__()`), and the *object instance* being operated on:



Now take another look at how the `__init__()` method was defined in the class:

```
def __init__(self):  
    # The code to initialize an "Athlete" object.  
    ...
```

Check out what Python turns your object creation invocation into. Notice anything?

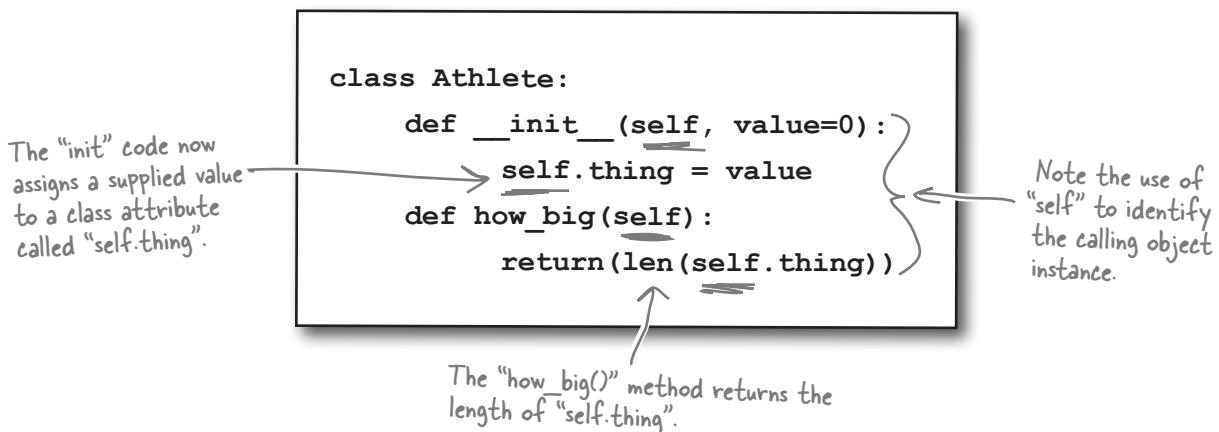
The target identifier is assigned to the `self` argument.

This is a very important argument assignment. Without it, the Python interpreter can't work out which object instance to apply the method invocation to. Note that the class code is designed to be *shared* among all of the object instances: the methods are shared, the attributes are **not**. The `self` argument helps identify which object instance's data to work on.

Every method's first argument is self

In fact, not only does the `__init__()` method require `self` as its first argument, but *so does every other method defined within your class*.

Python arranges for the first argument of every method to be the invoking (or *calling*) object instance. Let's extend the sample class to store a value in a object attribute called `thing` with the value set during initialization. Another method, called `how_big()`, returns the length of `thing` due to the use of the `len()` BIF:



When you invoke a class method on an object instance, Python arranges for the first argument to be the invoking object instance, which is *always* assigned to each method's `self` argument. This fact alone explains why `self` is so important and also why `self` needs to be the *first argument to every object method you write*:

What you write:

```
d = Athlete("Holy Grail")
```

What Python executes:

```
Athlete.__init__(d, "Holy Grail")
```

```
d.how_big()
```

```
Athlete.how_big(d)
```

The class

The method

The target identifier (or instance)



An IDLE Session

Let's use IDLE to create some object instances from a new class that you'll define. Start by creating a small class called Athlete:

```
>>> class Athlete:
    def __init__(self, a_name, a_dob=None, a_times=[]):
        self.name = a_name
        self.dob = a_dob
        self.times = a_times
```

Note the default values for two of the arguments.

Three attributes are initialized and assigned to three class attributes using the supplied argument data.

With the class defined, create two unique object instances which derive their characteristics from the Athlete class:

```
>>> sarah = Athlete('Sarah Sweeney', '2002-6-17', ['2:58', '2.58', '1.56'])
>>> james = Athlete('James Jones')
>>> type(sarah)
<class '__main__.Athlete'>
>>> type(james)
<class '__main__.Athlete'>
```

Create two unique athletes (with "james" using the default argument values).

Confirm that both "sarah" and "james" are athletes.

Even though sarah and james are both athletes and were created by the Athlete class's factory function, they are stored at different memory addresses:

```
>>> sarah
<__main__.Athlete object at 0x14d23f0>
>>> james
<__main__.Athlete object at 0x14cb7d0>
```

These are the memory addresses on our computer, which will differ from the values reported on yours. The key point is the memory address for "sarah" and "james" differ.

Now that sarah and james exist as object instances, you can use the familiar dot notation to access the attributes associated with each:

```
>>> sarah.name
'Sarah Sweeney'
>>> james.name
'James Jones'
>>> sarah.dob
'2002-6-17'
>>> james.dob
>>> sarah.times
['2:58', '2.58', '1.56']
>>> james.times
[]
```

The "james" object instance has no value for "dob", so nothing appears on screen.



Here's your code (except for the `sanitize()` function, which doesn't need to change). With your pencil, write code to define the `Athlete` class. In addition to the `__init__()` method, define a new method called `top3()` that, when invoked, returns the top three times.

Be sure to adjust the `get_coach_data()` function to return an `Athlete` object as opposed to a dictionary, and don't forget to amend your `print()` statements, too.

Write your Athlete class code here.

```
def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readline()
            templ = data.strip().split(',')
        return({'Name' : templ.pop(0),
               'DOB' : templ.pop(0),
               'Times': str(sorted(set([sanitize(t) for t in templ]))[0:3]))}
    except IOError as ioerr:
        print('File error: ' + str(ioerr))
        return(None)

james = get_coach_data('james2.txt')
print(james['Name'] + "'s fastest times are: " + james['Times'])
```

What needs to change here to ensure this function returns an Athlete object as opposed to a dictionary?

This line of code needs to change, too.



Here's your code (except for the `sanitize()` function, which doesn't need to change). With your pencil, you were to write code to define the `Athlete` class. In addition to the `__init__()` method, you were to define a new method called `top3()` that, when invoked, returns the top three times. You were to be sure to adjust the `get_coach_data()` function to return an `Athlete` object as opposed to a dictionary, and you weren't to forget to amend `print()`, too.

`class Athlete:`

```
def __init__(self, a_name, a_dob=None, a_times=[]):
    self.name = a_name
    self.dob = a_dob
    self.times = a_times
```

There's nothing new here as this code is taken straight from the most recent IDLE session.

Did you remember to use "self"?

```
def top3(self):
    return(sorted(set([sanitize(t) for t in self.times]))[0:3])
```

`def get_coach_data(filename):`

```
try:
    with open(filename) as f:
        data = f.readline()
    templ = data.strip().split(',')
    return({'Name': templ.pop(0),
            'DOB': templ.pop(0), Athlete(templ.pop(0), templ.pop(0), templ),
            'Times': str(sorted(set([sanitize(t) for t in templ]))[0:3])})
```

Remove the dictionary creation code and replace it with Athlete object creation code instead.

`except IOError as ioerr:`

```
    print('File error: ' + str(ioerr))
    return(None)
```

`james = get_coach_data('james2.txt')`

Use the dot notation to get at your data.

```
james.name
print(james['Name'] + "'s fastest times are: " + james['Times'])
```

`str(james.top3())`

Invoke the "top3()" method and convert its results to a string prior to its display on screen.



Test DRIVE

With these changes applied to your program, let's ensure you continue to get the same results as earlier. Load your code into IDLE and run it.

```
coach4.py - /Users/barryp/HeadFirstPython/chapter6/coach4.py

class Athlete:
    def __init__(self, a_name, a_dob=None, a_times=[]):
        self.name = a_name
        self.dob = a_dob
        self.times = a_times

    def top3(self):
        return(sorted(set([sanitize(t) for t in self.times]))[0:3])

def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readlines()
        templ = data.strip().split(',')
        return(Athlete(templ.pop(0), templ.pop(0), templ))
    except IOError as ioerr:
        print('File error: ' + str(ioerr))
        return(None)

james = get_coach_data('james2.txt')
julie = get_coach_data('julie2.txt')
mikey = get_coach_data('mikey2.txt')
sarah = get_coach_data('sarah2.txt')

print(james.name + "'s fastest times are: " + str(james.top3()))
print(julie.name + "'s fastest times are: " + str(julie.top3()))
print(mikey.name + "'s fastest times are: " + str(mikey.top3()))
print(sarah.name + "'s fastest times are: " + str(sarah.top3()))
```

The code to the "sanitize()" function is not shown here, but it is still part of this program.

```
Python Shell
>>> ===== RESTART =====
>>>
James Lee's fastest times are: ['2.01', '2.16', '2.22']
Julie Jones's fastest times are: ['2.11', '2.23', '2.59']
Mikey McManus's fastest times are: ['2.22', '2.31', '2.38']
Sarah Sweeney's fastest times are: ['2.18', '2.21', '2.22'] }
```

Cool! There's no change here.

And to make objects do more, I just add more methods, right?



Yes, that's correct: more functionality = more methods.

Simply add methods to **encapsulate** the new functionality you need within your class. There's no limit to how many methods a class can have, so feel free to knock yourself out!

Q: I'm not sure I see why the `top3()` method is coded to return a three-item list, as opposed to a string? Surely a string would make the `print()` statement in the main program easier to write?

A: It would, but it wouldn't be as flexible. By returning a list (albeit a small one), the `top3()` method lets the calling code decide what happens next, as opposed to forcing the caller to work with a string. Granted, the current program needs to treat the list like a string, but not all programs will want or need to.

Q: Why does the class even need the `top3()` method? Why not store the top three times as an attribute within the class and create it as part of the object's creation?

A: Again, better not to, because doing so is *less flexible*. If you compute and store the top three times at object creation, you make it harder to extend the list of timing data associated with the object.

For instance, if you add more timing data **after** the object is created, you'll need to arrange to recompute the top three (because the new times might be fast) and update the attribute. However, when you compute the top three times "on the fly" using a call to the `top3()` method, you always ensure you're using the most up-to-date data.

Q: OK, I get that. But, with a little extra work, I could do it during object creation, right?

A: Well, yes...but we really don't advise that. By preserving the original data in each object's attributes, you are supporting the extension of the class to support additional requirements in the future (whatever they might be). If you process the data as part of the object initialization code, the assumptions you make about how programmers will use your class might just come back to bite you.

Q: But what if I'm the only programmer that'll ever use a custom class that I write?

A: Trust us: you'll thank yourself for coding your class to be as flexible as possible when you come to use it for some other purpose in a future project. When you are creating a class, you have no idea how it will be used by other programmers in their projects. And, if you think about, you have no idea how you might use it in the future, too.

Q: OK, I think I'm convinced. But tell me: how do I go about adding more times to my existing Athlete objects?

A: To do more, add more methods. With your `Athlete` class created, it's a breeze to extend it to do more work for you: simply add more methods.

So, if you want to add a single new timing value to your `times` attribute, define a method called `add_time()` to do it for you. Additionally, you can add a list of times by defining a method called `add_times()`. Then all you need to do in your code is say something like this:

```
sarah.add_time('1.31')
to add a single time to Sarah's timing data, or say this:
james.add_times(['1.21', '2.22'])
to add a bunch of times to James's data.
```

Q: But surely, knowing that `times` is a list, I could write code like this to do the same thing?

```
sarah.times.append('1.31')
james.times.append(['1.21', '2.22'])
```

A: You could, but that would be a *disaster*.

Q: What?!? Why do you say that? There's nothing wrong with my suggestion, is there?

A: Well...it does indeed work. However, the **problem** with writing code like that is that it exposes (and exploits) that fact that the timing data is stored in a list within the `Athlete` class. If you later change your class implementation to use (for instance) a string *instead of* a list, you may well break all of the existing code that uses your class and that exploits the fact that the timing data is a list.

By defining your own API with `add_time()` and `add_times()`, you leave open the possibility that the way the data is stored within your class can change in the future (obviously, only if such a change makes sense). It is worth noting that one of the reasons for using object orientation is to hide away the details of a class's implementation from the users of that class. Defining your own API *directly* supports this design ideal. Exposing the internals of your class's implementation and expecting programmers to exploit it **breaks** this fundamental ideal in a very big way.



Let's add two methods to your class. The first, called `add_time()`, appends a single additional timing value to an athlete's timing data. The second, `add_times()`, extends an athlete's timing data with one or more timing values supplied as a list.

Here's your current class: add the code to implement these two new methods.

```
class Athlete:
    def __init__(self, a_name, a_dob=None, a_times=[]):
        self.name = a_name
        self.dob = a_dob
        self.times = a_times

    def top3(self):
        return(sorted(set([sanitize(t) for t in self.times]))[0:3])
```

Add your new methods here.



.....
.....
.....
.....
.....
.....
.....
.....
.....



Sharpen your pencil

Don't put down the pencil just yet! Provide a few lines of code to test your new functionality:

.....
.....
.....
.....
.....

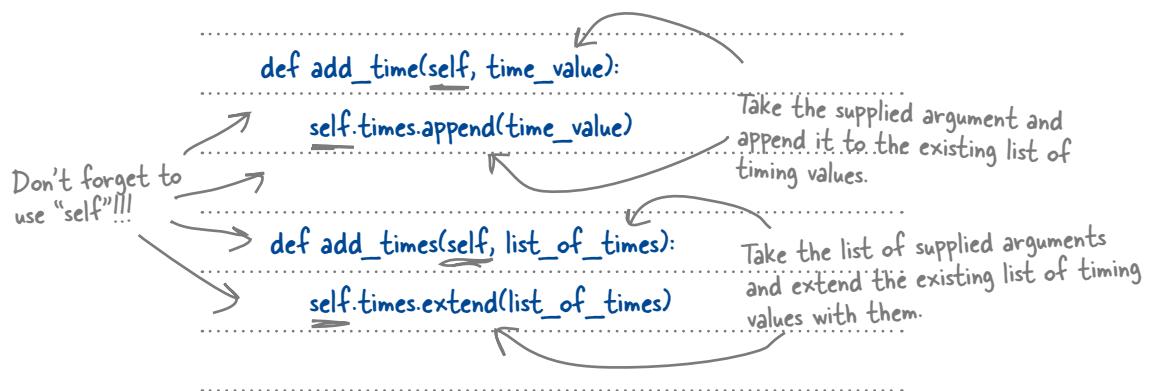


Let's add two methods to your class. The first, called `add_time()`, appends a single additional timing value to an athlete's timing data. The second, `add_times()`, extends an athlete's timing data with one or more timing values supplied as a list.

Here's your current class: you were to add the code to implement these two new methods.

```
class Athlete:
    def __init__(self, a_name, a_dob=None, a_times=[]):
        self.name = a_name
        self.dob = a_dob
        self.times = a_times

    def top3(self):
        return(sorted(set([sanitize(t) for t in self.times]))[0:3])
```



Create a new object instance for Vera.

→ `vera = Athlete('Vera Vi')`

`vera.add_time('1.31')` ← Add a single timing value.

`print(vera.top3())` ← This will display a list with only one value in it: 1.31.

`vera.add_times(['2.22', '1-21', '2:22'])` ← Add three more timing values.

`print(vera.top3())` ← The top 3 timing scores are now: 1.21, 1.31 and 2.22.

While still holding on firmly to your pencil, you were to provide a few lines of code to test your new functionality:



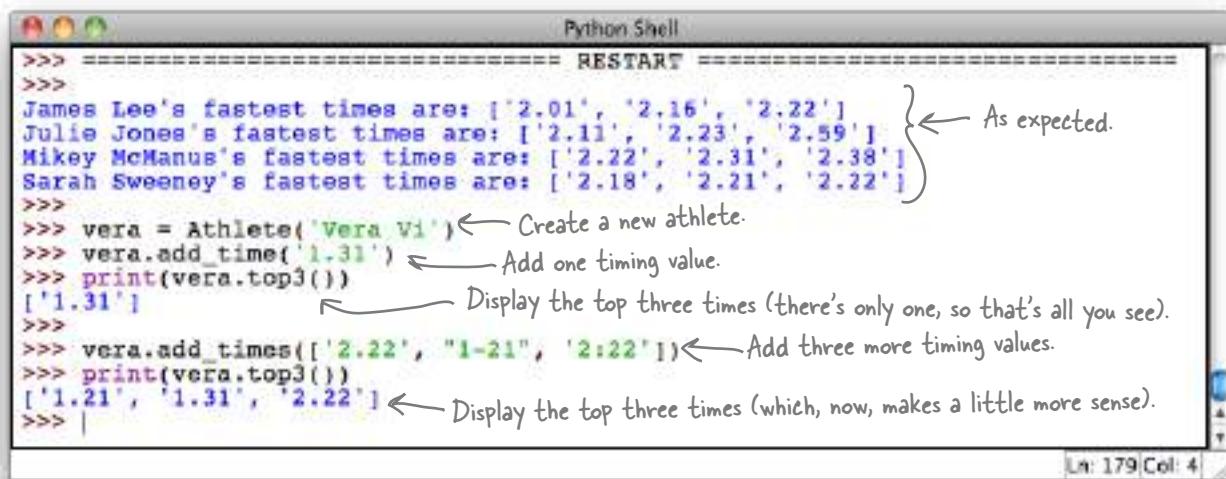
Do this!

Amend your code with the updated version of your Athlete class *before* proceeding with this Test Drive.



Test Drive

After running your existing program, try out your test code in the IDLE shell to confirm that everything is working as expected.



```

Python Shell
>>> ===== RESTART =====
>>>
James Lee's fastest times are: ['2.01', '2.16', '2.22']
Julie Jones's fastest times are: ['2.11', '2.23', '2.59'] } As expected.
Mikey McManus's fastest times are: ['2.22', '2.31', '2.38']
Sarah Sweeney's fastest times are: ['2.18', '2.21', '2.22']
>>>
>>> vera = Athlete('Vera Vi') ← Create a new athlete.
>>> vera.add_time('1.31') ← Add one timing value.
>>> print(vera.top3())
['1.31'] ← Display the top three times (there's only one, so that's all you see).
>>>
>>> vera.add_times(['2.22', '1.21', '2.22']) ← Add three more timing values.
>>> print(vera.top3())
['1.21', '1.31', '2.22'] ← Display the top three times (which, now, makes a little more sense).
>>>

```

Great: it worked.

You've **packaged** your code with your data and created a custom class from which you can create objects that share behaviors. And when extra functionality is required, **add more methods** to implement the required functionality.

By **encapsulating** your athlete code and data within a custom class, you've created a much more **maintainable** piece of software. You will thank yourself for doing this when, in six months, you need to amend your code.

Well done. This is really coming along!



Yes, your Athlete class is much like a list.

Your Athlete class does indeed behave like a **list** *most of the time*, and you've added methods to expose some list functionality to the users of your class. But it's true: *you are* reinventing the wheel here. Your add_time() method is a thin wrapper around the list append() method and your add_times() method is list's extend() method in disguise.

In fact, your Athlete class only differs from Python's list due to the inclusion of the name and dob object attributes.



Inherit from Python's built-in list

Python's **class** lets you create a custom class *from scratch*, just like you did with your Athlete class. However, **class** also lets you create a class by *inheriting* from any other existing class, including Python's built-in data structure classes that provide you with list, set, and dict. Such classes are referred to as *subclasses*.

What's really nice is that when you inherit from an existing class (such as list), you are given all of the existing functionality for free.

As your existing class is really nothing more than a **list** with added attributes, perhaps a better design is to kill off your Athlete class and replace it with a class that inherits from the built-in list class? It's certainly worth considering, isn't it?



Fireside Chats



Tonight's talk: **Inheritance, a.k.a. He looks just like his father.**

Custom Class:

Programmers like me because they get to control *everything* in their code...and you know programmers: *they love to code.*

Design! Phooey! Real programmers eat, sleep, dream, snore, and exhale code. All that design talk is for people who *can't* code!

No, no, no: you're not listening. It's all done with control. When you build everything from the ground up, you're in control, as *it's all your code.*

Of course, especially when there are custom requirements to be taken into consideration. In that case, a brand-spanking new custom class is the only way to go.

Yeah, right...it's a win-win for you, not me.

I guess so, although I'm still a fan of custom code...

Inherited Class:

Yes, they do. But sometimes writing everything from scratch is not the best design decision.

Is it really? So, you're saying it's much better to do everything from scratch and repeat the work of others, because your way is the best way. Are you serious?!?

And you're happy to reinvent the wheel, even though someone else solved that problem eons ago?

Not if you can extend someone else's class to handle your custom requirements. That way, you get the best of both worlds: inherited functionality (so you're not reinventing the wheel) together with the custom bits. It's a win-win situation.

But it's not about us: it's to do with making the life of the programmer easier, even the ones that live to code, right?



An IDLE Session

Let's see what's involved in inheriting from Python's built-in `list` class. Working in IDLE's shell, start by creating a custom list derived from the built-in `list` class that also has an attribute called `name`:

```
>>> class NamedList(list):
    def __init__(self, a_name):
        list.__init__([])
        self.name = a_name
```

Provide the name of the class that this new class derives from.

Initialize the derived from class, and then assign the argument to the attribute.

With your `NamedList` class defined, use it to create an object instance, check the object's type (using the `type()` BIF), and see what it provides (using the `dir()` BIF):

```
>>> johnny = NamedList("John Paul Jones") ← Create a new "NamedList" object instance.
>>> type(johnny)
<class '__main__.NamedList'> ← Yes, "johnny" is a "NamedList".
>>> dir(johnny)
['__add__', '__class__', '__contains__', '__delattr__', '__delitem__', '__dict__', '__doc__',
 '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__',
 '__iadd__', '__imul__', '__init__', '__iter__', '__le__', '__len__', '__lt__', '__module__',
 '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__',
 '__rmul__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__',
 '__weakref__', 'append', 'count', 'extend', 'index', 'insert', 'name', 'pop', 'remove',
 'reverse', 'sort']
```

"johnny" can do everything a list can, as well as store data in the "name" attribute.

Use some of the functionality supplied by the `list` class to add to the data stored in `johnny`:

```
>>> johnny.append("Bass Player")
>>> johnny.extend(['Composer', 'Arranger', 'Musician']) } ← Add data to the "NamedList" using the methods provided by the list built in.
>>> johnny
['Bass Player', 'Composer', 'Arranger', 'Musician']
>>> johnny.name
'John Paul Jones' ← Access the list data, as well as the attribute data.
```

Because `johnny` is a `list`, it's quite OK to do list-type things to it:

```
>>> for attr in johnny:
    print(johnny.name + " is a " + attr + ".") } ← "johnny" is like any other list, so feel free to use it wherever you'd use a list.
```

```
John Paul Jones is a Bass Player.
John Paul Jones is a Composer.
John Paul Jones is a Arranger.
John Paul Jones is a Musician.
```

Confirmation: John's a busy boy. ☺



Exercise

Here is the code for the now defunct `Athlete` class. In the space provided below, rewrite this class to inherit from the built-in `list` class. Call your new class `AthleteList`. Provide a few lines of code to exercise your new class, too:

```
class Athlete:

    def __init__(self, a_name, a_dob=None, a_times=[]):
        self.name = a_name
        self.dob = a_dob
        self.times = a_times

    def top3(self):
        return(sorted(set([sanitize(t) for t in self.times]))[0:3])

    def add_time(self, time_value):
        self.times.append(time_value)

    def add_times(self, list_of_times):
        self.times.extend(list_of_times)
```

Write your new class code here.

Exercise your
code here



Here is the code for the now defunct Athlete class. In the space provided below, you were to rewrite this class to inherit from the built-in list class. You were to call your new class AthleteList, as well as provide a few lines of code to exercise your new class:

```
class Athlete:
    def __init__(self, a_name, a_dob=None, a_times=[]):
        self.name = a_name
        self.dob = a_dob
        self.times = a_times

    def top3(self):
        return(sorted(set([sanitize(t) for t in self.times]))[0:3])
```

~~def add_time(self, time_value):
 self.times.append(time_value)~~

~~def add_times(self, list_of_times):
 self.times.extend(list_of_times)~~

The class name
has changed.

Inherit from the built-in list class.

These
methods
aren't needed
anymore.

```
class AthleteList(list):
    def __init__(self, a_name, a_dob=None, a_times=[]):
```

Nothing new here...
this code is very
similar to the
"NamedList" init code.

list.__init__([])

self.name = a_name

self.dob = a_dob

self.extend(a_times)

def top3(self):

```
return(sorted(set([sanitize(t) for t in self]))[0:3])
```

Use the new
class's name.

Now that you're
inheriting from
the built-in list,
you can use its
methods to get
your work on.

vera = AthleteList('Vera Vi')

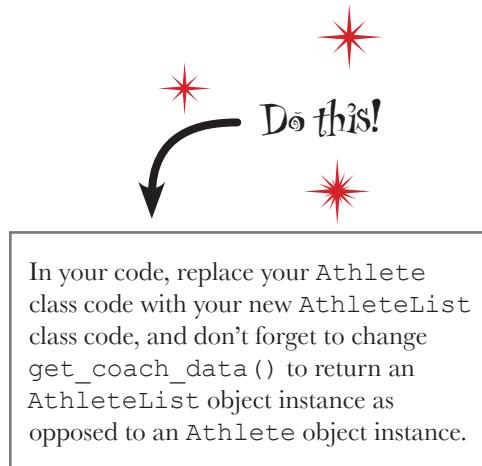
vera.append('1.31')

print(vera.top3())

vera.extend(['2.22', '1-21', '2.22'])

print(vera.top3())

This code does a good job of
exercising your new class.



^{there are no} Dumb Questions

Q: Sorry...but not three minutes ago you were telling me not to expose the inner workings of my class to its users, because that was fundamentally a bad idea. Now you're doing the exact opposite! What gives?

A: Well spotted. In this particular case, it's OK to expose the fact that the class is built on top of `list`. This is due to the fact that the class is deliberately called `AthleteList` to distinguish it from the more generic `Athlete` class. When programmers see the word "list" in a class name, they are likely to expect the class to work like a `list` and then some. This is the case with `AthleteList`.

Q: And I can inherit from any of the built-in types?

A: Yes.

Q: What about inheriting from more than one class...does Python support multiple inheritance?

A: Yes, but it's kind of scary. Refer to a good Python reference text for all the gory details.

Q: Can I inherit from my own custom classes?

A: Of course, that's the whole idea. You create a generic class that can then be "subclassed" to provide more specific, targeted functionality.

Q: Can I put my class in a module file?

A: Yes, that's a really good idea, because it lets you share your class with many of your own programs and with other programmers. For instance, if you save your `AthleteList` class to a file called `athletelist.py`, you can import it into your code using this line of code:

```
from athletelist import AthleteList
```

then use the class as if it was defined in your current program. And, of course, if you create a really useful class, pop it into its own module and upload it to PyPI for the whole world to share.



Test DRIVE

One last run of your program should confirm that it's working to specification now. Give it a go in IDLE to confirm.

```
coach6.py - /Users/barryp/HeadFirstPython/chapter6/coach6.py

def sanitize(time_string):
    if '-' in time_string:
        splitter = '-'
    elif ':' in time_string:
        splitter = ':'
    else:
        return(time_string)
    (mins, secs) = time_string.split(splitter)
    return(mins + '.' + secs)

class AthleteList(list):

    def __init__(self, a_name, a_dob=None, a_times=[]):
        list.__init__([])
        self.name = a_name
        self.dob = a_dob
        self.extend(a_times)

    def top3(self):
        return(sorted(set([sanitize(t) for t in self]))[0:3])

    def get_coach_data(filename):
        try:
            with open(filename) as f:
                data = f.readlines()
            templ = data.strip().split(',')
            return(AthleteList(templ.pop(0), templ.pop(0), templ))
        except IOError as ioerr:
            print('File error: ' + str(ioerr))
            return(None)

james = get_coach_data('james2.txt')
julie = get_coach_data('julie2.txt')
mikey = get_coach_data('mikey2.txt')
sarah = get_coach_data('sarah2.txt')

print(james.name + "'s fastest times are: " + str(james.top3()))
print(julie.name + "'s fastest times are: " + str(julie.top3()))
print(mikey.name + "'s fastest times are: " + str(mikey.top3()))
print(sarah.name + "'s fastest times are: " + str(sarah.top3()))
```

Your entire
program now
produces the
output the
coach wants.

```
Python Shell
>>> ===== RESTART =====
>>>
James Lee's fastest times are: ['2.01', '2.16', '2.22']
Julie Jones's fastest times are: ['2.11', '2.23', '2.59']
Mikey McManus's fastest times are: ['2.22', '2.31', '2.38']
Sarah Sweeney's fastest times are: ['2.18', '2.21', '2.22']
>>> |
Ln: 306 Col: 4
```

Coach Kelly is impressed



By basing your class on built-in functionality, you've leveraged the power of Python's data structures while providing the custom solution your application needs.

You've **engineered** a much more *Maintainable solution* to Coach Kelly's data processing needs.

Good job!



Your Python Toolbox

You've got Chapter 6 under your belt and you've added some key Python techniques to your toolbox.

Python Lingo

- "Dictionary" – a built-in data structure that allows you to associate data values with keys.
- "Key" – the look-up part of the dictionary.
- "Value" – the data part of the dictionary (which can be any value, including another data structure).

More Python Lingo

- "self" – a method argument that always refers to the current object instance.



BULLET POINTS

- Create an empty dictionary using the `dict()` factory function or using `{}`.
- To access the value associated with the key `Name` in a dictionary called `person`, use the familiar square bracket notation: `person['Name']`.
- Like `list` and `set`, a Python's dictionary dynamically grows as new data is added to the data structure.
- Populate a dictionary as you go:
`new_d = {}` or `new_d = dict()` and then
`d['Name'] = 'Eric Idle'`
or do the same thing all in the one go:
`new_d = {'Name': 'Eric Idle'}`
- The `class` keyword lets you define a class.
- Class **methods** (your code) are defined in much the same way as functions, that is, with the `def` keyword.
- Class **attributes** (your data) are just like variables that exist within object instances.
- The `__init__()` method can be defined within a class to initialize object instances.
- Every method defined in a class must provide `self` as its first argument.
- Every attribute in a class must be prefixed with `self`. in order to associate it data with its instance.
- Classes can be built from scratch or can **inherit** from Python's built-in classes or from other custom classes.
- Classes can be put into a Python module and **uploaded** to PyPI.

7 web development

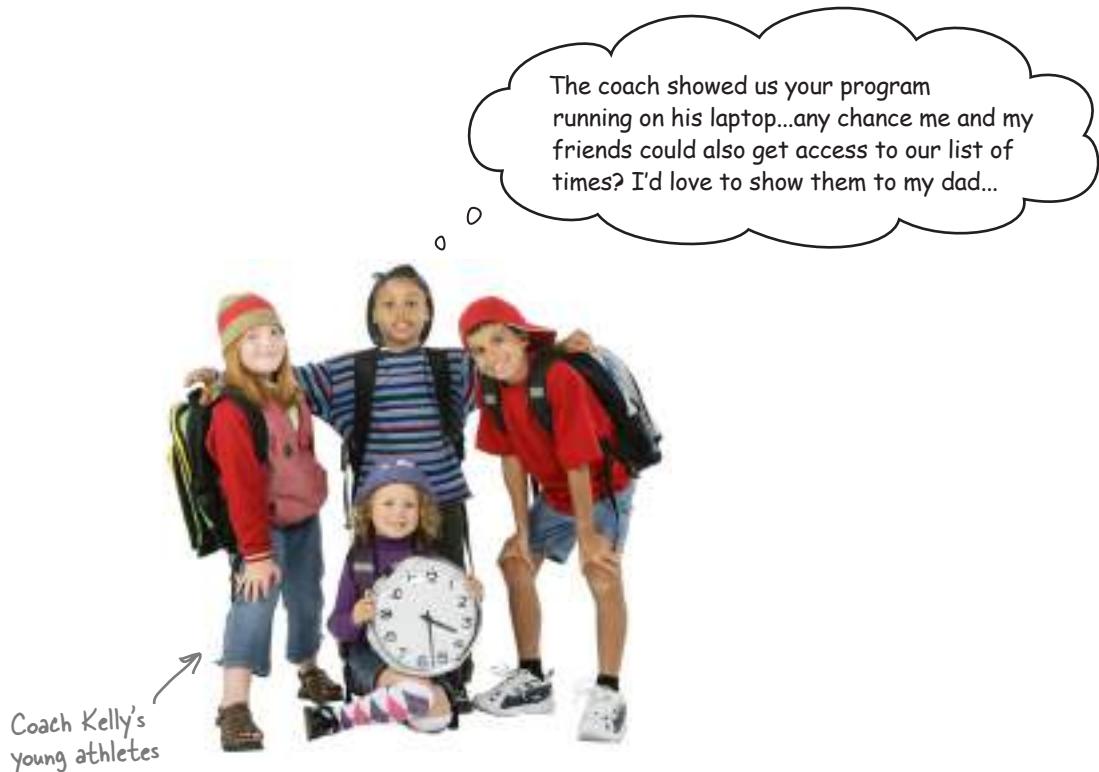
Putting it all together



Sooner or later, you'll want to share your app with lots of people.

You have many options for doing this. Pop your code on PyPI, send out lots of emails, put your code on a CD or USB, or simply install your app manually on the computers of those people who need it. Sounds like a lot of work...not to mention boring. Also, what happens when you produce the next best version of your code? What happens then? How do you manage the update? Let's face it: it's such a pain that you'll think up really creative excuses not to. Luckily, you don't have to do any of this: just create a webapp instead. And, as this chapter demonstrates, using Python for web development is a breeze.

It's good to share



You're a victim of your own success.

The new requests come flooding in right after Coach Kelly starts showing off your latest program. It appears that everyone wants access to the coach's data!

The thing is: what's the “best way” to do this?

You can put your program on the Web



A “webapp” is what you want.

If you develop your program as a *Web-based application* (or *webapp*, for short), your program is:

- Available to everyone who can get to your website
- In one place on your web server
- Easy to update as new functionality is needed

But...how do webapps actually work?



Webapps Up Close

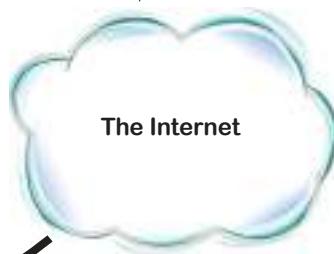
No matter what you do on the Web, it's all about *requests* and *responses*. A **web request** is sent from a web browser to a web server as the result of some user interaction. On the web server, a **web response** (or *reply*) is formulated and *sent back* to the web browser. The entire process can be summarized in five steps.

Step 1: Your user enters a web address, selects a hyperlink, or clicks a button in her chosen web browser.



I just type the web address into my browser's location bar and press Enter...

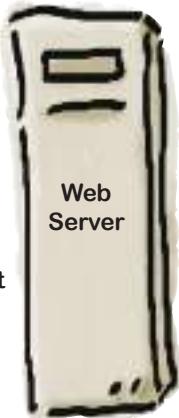
Step 2: The web browser converts the user's action into a web request and sends it to a server over the Internet.



Here comes a web request.

Hey, hello there...what's this?
A web request just for me? How nice...

Step 3: The web server receives the web request and has to decide what to do next.



Deciding what to do next

One of two things happen at this point. If the web request is for **static content**—such as an HTML file, image, or anything else stored on the web server's hard disk—the web server locates the *resource* and returns it to the web browser as a **web response**.

If the request is for **dynamic content**—that is, content that must be *generated*—the web server runs a program to produce the **web response**.

Step 4: The web server processes the web request, creating a web response, which is sent back over the Internet to the waiting web browser.



Here you go...a web response generated just for you. Enjoy!

Here comes a web response.



The Internet

That's exactly what I need. Thanks!



Step 5: The web browser receives the web response and displays it on your user's screen.

The (potentially) many substeps of step 4

In practice, step 4 can involve multiple *substeps*, depending on what the web server has to do to produce the response. Obviously, if all the server has to do is locate static content and send it back to the browser, the substeps aren't too taxing, because it's all just file I/O.

However, when dynamic content must be *generated*, the substeps involve the web server *locating* the program to execute, *executing* the located program, and then *capturing* the output from the program as the web response...which is then sent back to the waiting web browser.

This dynamic content generation process has been standardized since the early days of the Web and is known as the **Common Gateway Interface** (CGI). Programs that conform to the standard are often referred to as *CGI scripts*.

What does your webapp need to do?

Let's take a moment to consider what you want your webapp to look like and how it should behave on your user's web browser. You can then use this information to help you specify what your webapp needs to do.



Sharpen your pencil



There's nothing like grabbing your pencil and a few blank paper napkins to quickly sketch a simple web design. You probably need three web pages: a "welcome" page, a "select an athlete" page, and a "display times" page. Go ahead and draw out a rough design on the napkins on this page, and don't forget to draw any linkages between the pages (where it makes sense).

back-of-the-napkin sketch

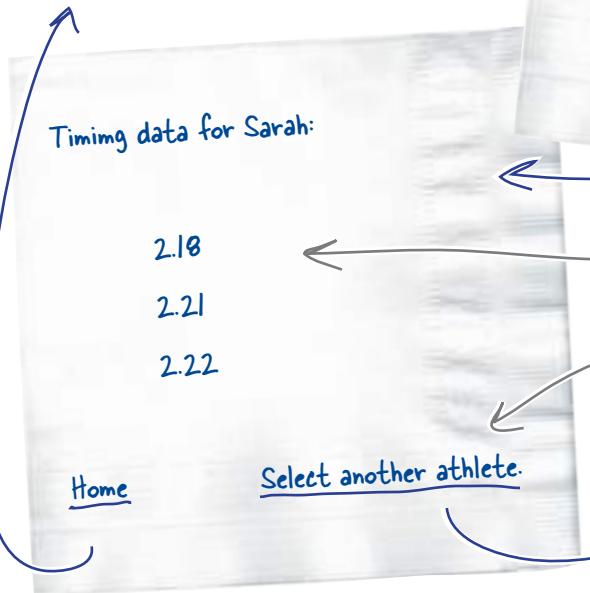
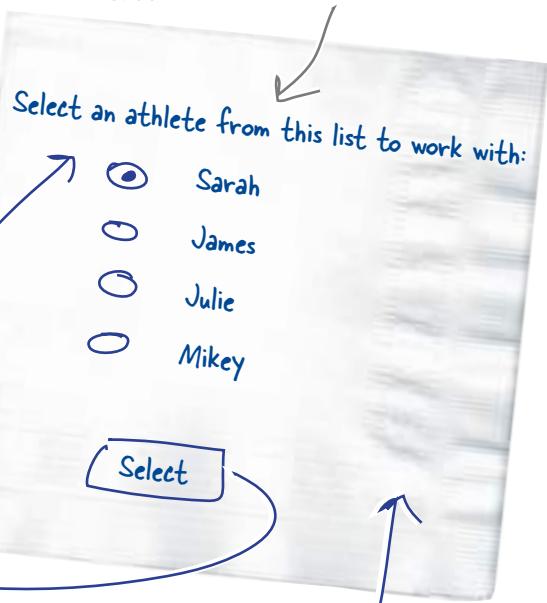


The home page displays a friendly graphic and a link to start the web app.



There's nothing like grabbing your pencil and a few blank paper napkins to quickly sketch a simple web design. You probably need three web pages: a "welcome" page, a "select an athlete" page, and a "display times" page. You were to draw out a rough design on the napkins. You were to draw any linkages between the pages (where it made sense).

Click on the home page's link to go to a page that displays a list of all the coach's athletes. Click on an athlete's radio button and then the "Select" button to see the data.



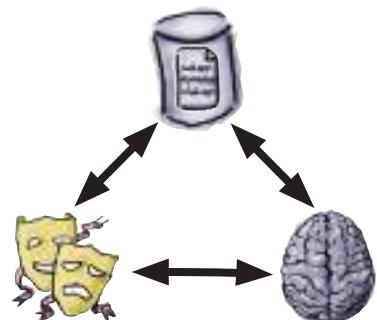
The third web page displays the selected athlete's data and provides links back to the other two pages.

Design your webapp with MVC

Now that you have an idea of the pages your webapp needs to provide, your next question should be: *what's the best way to build this thing?*

Ask 10 web developers that question and you'll get 10 different answers; the answer often depends on whom you ask.

Despite this, the general consensus is that great webapps conform to the *Model-View-Controller* pattern, which helps you segment your webapp's code into easily manageable functional chunks (or *components*):



The Model

The code to store (and sometimes process) your webapp's data



The View

The code to format and display your webapp's user interface(s)



The Controller

The code to glue your webapp together and provide its business logic

By following the MVC pattern, you build your webapp in such a way as to enable your webapp to grow as new requirements dictate. You also open up the possibility of splitting the workload among a number of people, one for each component.

Let's build each of the MVC components for your webapp.

Model your data

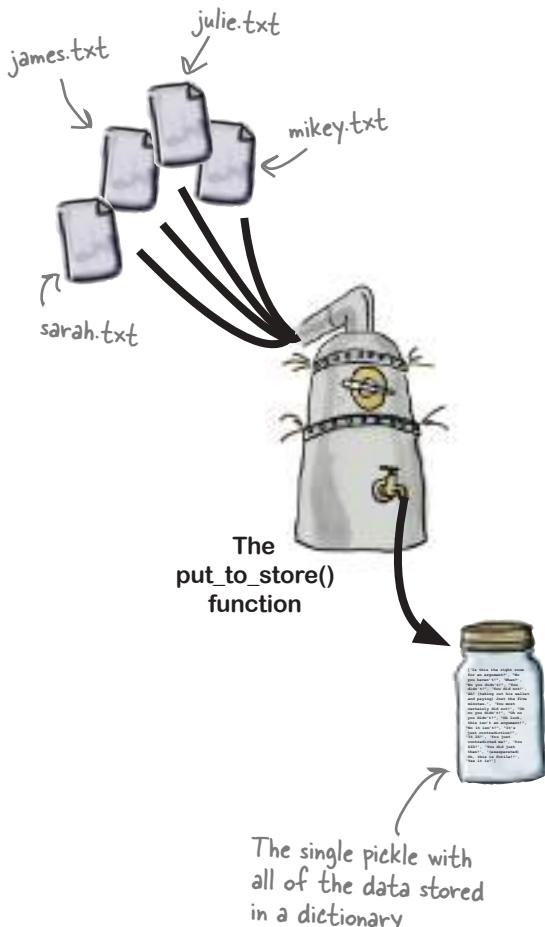
Your web server needs to store a single copy of your data, which in this case is Coach Kelly's timing values (which start out in his text files).

When your webapp starts, the data in the text files needs to be converted to `AthleteList` object instances, stored within a dictionary (indexed by athlete name), and then saved as a pickle file. Let's put this functionality in a new function called `put_to_store()`.

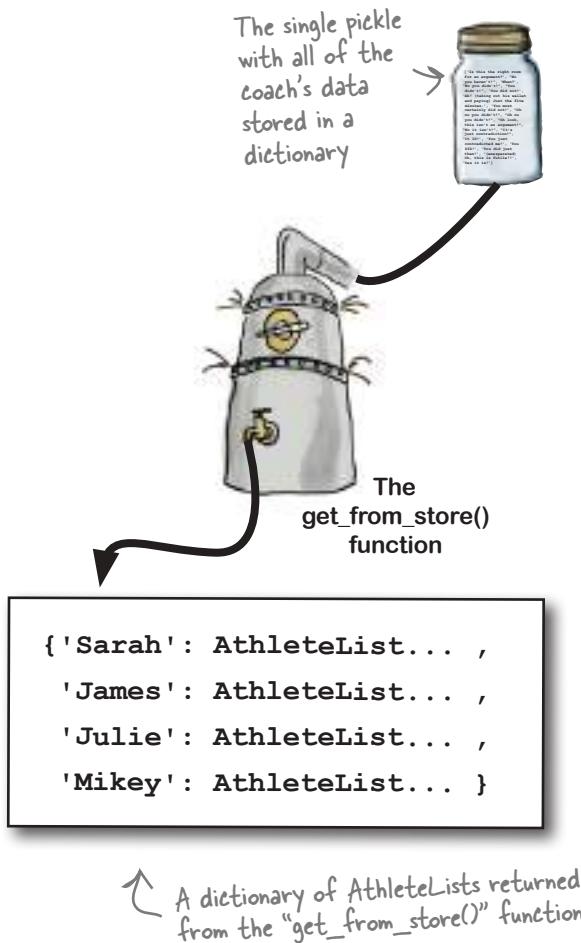
While your webapp runs, the data in the pickle needs to be available to your webapp as a dictionary. Let's put this functionality in another new function called `get_from_store()`.



When your webapp starts:



While your webapp runs:





Exercise

Here is the outline for a new module called `athletemodel.py`, which provides the functionality described on the previous page. Some of the code is already provided for you. Your job is to provide the rest of the code to the `put_to_store()` and `get_from_store()` functions. Don't forget to protect any file I/O calls.

```
import pickle
from athletelist import AthleteList

def get_coach_data(filename):
    # Not shown here as it has not changed since the last chapter.
```

```
def put_to_store(files_list):
    all_athletes = {}
```

This function is called with a list of filenames as its sole argument.

You need code in here to populate the dictionary with the data from the files.

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

And don't forget to save the dictionary to a pickle (and check for file I/O errors).

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

```
.....
```

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



Here is the outline for a new module called `athletemodel.py`, which provides the functionality described on the previous page. Some of the code is already provided for you. Your job was to provide the rest of the code to the `put_to_store()` and `get_from_store()` functions. You were not to forget to protect any file I/O calls.

```
import pickle
from athletelist import AthleteList

def get_coach_data(filename):
    # Not shown here as it has not changed since the last chapter.
```

```
def put_to_store(files_list):
    all_athletes = {}
```

Take each file, turn it
into an `AthleteList`
object instance, and
add the athlete's data
to the dictionary.

for each_file in files_list:

 → `ath = get_coach_data(each_file)`

`all_athletes[ath.name] = ath` ←

Each athlete's name is
used as the "key" in the
dictionary. The "value" is
the `AthleteList` object
instance.

try:

 with open('athletes.pickle', 'wb') as athf:

 → `pickle.dump(all_athletes, athf)`

 except IOError as ioerr:

 print('File error (put_and_store): ' + str(ioerr))

Save the entire
dictionary of
`AthleteLists`
to a pickle.

return(all_athletes)

And don't forget
a try/except to
protect your file
I/O code.

```
def get_from_store():
```

```
    all_athletes = {}
```

try:

 with open('athletes.pickle', 'rb') as athf:

 → `all_athletes = pickle.load(athf)`

 except IOError as ioerr:

 print('File error (get_from_store): ' + str(ioerr))

Simply read the
entire pickle into
the dictionary.
What could be
easier?

return(all_athletes)

Again...don't
forget your
try/except.



An IDLE Session

Let's test your code to ensure that it is working to specification. Type your code into an IDLE edit window and save your code into a folder that also includes the coach's text files. Press F5 to import your code to the IDLE shell, and then use the `dir()` command to confirm that the import has been successful:

```
>>> dir()
['AthleteList', '__builtins__', '__doc__', '__name__', '__package__', 'get_coach_data',
'get_from_store', 'pickle', 'put_to_store']
```

Create a list of files to work with, and then call the `put_to_store()` function to take the data in the list of files and turn them into a dictionary stored in a pickle:

```
>>> the_files = ['sarah.txt', 'james.txt', 'mikey.txt', 'julie.txt']
>>> data = put_to_store(the_files)
>>> data
```

```
{'James Lee': ['2-34', '3:21', '2.34', '2.45', '3.01', '2:01', '2:01', '3:10', '2-22', '2-
01', '2.01', '2:16'], 'Sarah Sweeney': ['2:58', '2.58', '2:39', '2-25', '2-55', '2:54', '2.18',
'2:55', '2:22', '2-21', '2.22'], 'Julie Jones': ['2.59', '2.11', '2:11', '2:23', '3-
10', '2-23', '3:10', '3.21', '3-21', '3.01', '3.02', '2:59'], 'Mikey McManus': ['2:22', '3.01',
'3:01', '3.02', '3:02', '3:22', '2.49', '2:38', '2:40', '2.22', '2-31']}
```

Here's all of the AthleteLists.



At this point, the `athletes.pickle` file should appear in the same folder as your code and text files. Recall that this file is a *binary* file, so trying to view it in IDLE or in your editor is not going to make much sense. To access the data, use the dictionary returned by the `put_to_store()` or `get_from_store()` functions.

Use the existing data in the `data` dictionary to display each athlete's name and date of birth:

```
>>> for each_athlete in data:
    print(data[each_athlete].name + ' ' + data[each_athlete].dob)
```

```
James Lee 2002-3-14
Sarah Sweeney 2002-6-17
Julie Jones 2002-8-17
Mikey McManus 2002-2-24
```

By accessing the "name" and "dob" attributes, you can get at the rest of the AthleteList data.

Use the `get_from_store()` function to load the pickled data into another dictionary, then confirm that the results are as expected by repeating the code to display each athlete's name and date of birth:

```
>>> data_copy = get_from_store()
>>> for each_athlete in data_copy:
    print(data_copy[each_athlete].name + ' ' + data_copy[each_athlete].dob)
```

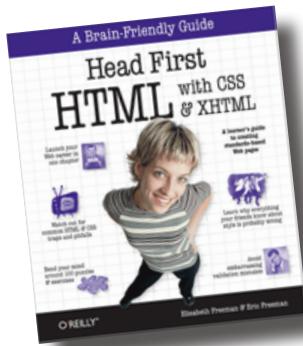
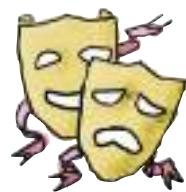
```
James Lee 2002-3-14
Sarah Sweeney 2002-6-17
Julie Jones 2002-8-17
Mikey McManus 2002-2-24
```

The data in the returned dictionary is as expected, exactly the same as that produced by `put_to_store()`.

View your interface

With your model code written and working, it's time to look at your view code, which creates your webapp's user interface (UI).

On the Web, UIs are created with HTML, the Web's markup technology. If you are new to HTML, it is worth taking some time to become familiar with this *critical* web development technology. There's lots of material on the Web and more than a few good books out there.



[Note from Marketing:
This is the book that we
recommend for quickly
getting up to speed with
HTML...not that we're
biased or anything. ☺].

Hey, we hear you are getting into
web development? We have a small
module that we put together that might help
you generate HTML. It's a little rough, but it
works. You're more than welcome to use it for
your projects, if you like.

(Most of) the Head First
Code Review Team



YATE: Yet Another Template Engine

Your friends over at the Head First Code Review Team heard you're planning to write some code to generate HTML for your webapp's UI. They've sent over some code that they swear will make your life easier. It's a small library of HTML-generating helper functions called `yate`. The code was produced quickly and was originally designed to be "throw away," so the team has provided it *as is*. It's somewhat raw, but it should be OK.

```

from string import Template

def start_response(resp="text/html"):
    return('Content-type: ' + resp + '\n\n')

def include_header(the_title):
    with open('templates/header.html') as headf:
        head_text = headf.read()
    header = Template(head_text)
    return(header.substitute(title=the_title))

def include_footer(the_links):
    with open('templates/footer.html') as footf:
        foot_text = footf.read()
    link_string = ''
    for key in the_links:
        link_string += '<a href="' + the_links[key] + '">' + key + '</a>&nbsp;&nbsp;&nbsp;'
    footer = Template(foot_text)
    return(footer.substitute(links=link_string))

def start_form(the_url, form_type="POST"):
    return('<form action="' + the_url + '" method="' + form_type + '">')

def end_form(submit_msg="Submit"):
    return('<p></p><input type="submit value="' + submit_msg + '">')

def radio_button(rb_name, rb_value):
    return('<input type="radio" name="' + rb_name +
          '" value="' + rb_value + '"> ' + rb_value + '<br />')

def u_list(items):
    u_string = '<ul>'
    for item in items:
        u_string += '<li>' + item + '</li>'
    u_string += '</ul>'
    return(u_string)

def header(header_text, header_level=2):
    return('<h' + str(header_level) + '>' + header_text +
          '</h' + str(header_level) + '>')

def para(para_text):
    return('<p>' + para_text + '</p>')

```

There's not much help here, just the code. No comments, explanations, documentation, or anything!



LONG Exercise

Let's get to know the `yate` code before proceeding with the rest of this chapter. For each chunk of code presented, provide a written description of what you think it does in the spaces provided:

Take a moment to look up the "Template" module in Python's documentation set.

```
from string import Template
```

Write your explanations in the spaces.

One has already been done for you.

```
def start_response(resp="text/html"):
    return('Content-type: ' + resp + '\n\n')
```

→ This function takes a single (optional) string as its argument and uses it to create a CGI "Content-type:" line, with "text/html" as the default.

```
def include_header(the_title):
    with open('templates/header.html') as headf:
        head_text = headf.read()
    header = Template(head_text)
    return(header.substitute(title=the_title))
```

```
def include_footer(the_links):
    with open('templates/footer.html') as footf:
        foot_text = footf.read()
    link_string = ''
    for key in the_links:
        link_string += '<a href="' + the_links[key] + '">' + key +
                      '</a>&nbsp;&nbsp;&nbsp;&nbsp;''
    footer = Template(foot_text)
    return(footer.substitute(links=link_string))
```

```
def start_form(the_url, form_type="POST"):
    return('<form action="' + the_url + '" method="' + form_type + '">'')
```

```
.....
```

```
.....
```

```
def end_form(submit_msg="Submit"):
    return('<p></p><input type=submit value="' + submit_msg + '">''</form>')
```

```
.....
```

```
.....
```

```
def radio_button(rb_name, rb_value):
    return('<input type="radio" name="' + rb_name +
           '" value="' + rb_value + '">' + rb_value + '<br />')
```

```
.....
```

```
.....
```

```
def u_list(items):
    u_string = '<ul>'
    for item in items:
        u_string += '<li>' + item + '</li>'
    u_string += '</ul>'
    return(u_string)
```

```
.....
```

```
.....
```

```
.....
```

```
def header(header_text, header_level=2):
    return('<h' + str(header_level) + '>' + header_text +
          '</h' + str(header_level) + '>')
```

```
.....
```

```
.....
```

```
def para(para_text):
    return('<p>' + para_text + '</p>')
```

```
.....
```

```
.....
```



LONG Exercise SOLUTION

Let's get to know the `yate` code before proceeding with the rest of this chapter. For each chunk of code presented, you were to provide a written description of what you think it does:

```
from string import Template
```

Import the "Template" class from the standard library's "string" module. This allows for simple string-substitution templates.

Note the default for "resp".

```
def start_response(resp="text/html"):
    return('Content-type: ' + resp + '\n\n')
```

This function takes a single (optional) string as its argument and uses it to create a CGI "Content-type:" line, with "text/html" as the default.

Open the template file (which is HTML), read it in, and substitute in the provided "title".

```
def include_header(the_title):
    with open('templates/header.html') as headf:
        head_text = headf.read()
        header = Template(head_text)
    return(header.substitute(title=the_title))
```

This function takes a single string as its argument and uses at the title for the start of a HTML page. The page itself is stored within a separate file in "templates/header.html", and the title is substituted in as needed.

Open the template file (which is HTML), read it in, and substitute in the provided dictionary of HTML links in "the_links".

```
def include_footer(the_links):
    with open('templates/footer.html') as footf:
        foot_text = footf.read()
        link_string = ''
        for key in the_links:
            link_string += '<a href="' + the_links[key] + '">' + key +
                           '</a>&nbsp;&nbsp;&nbsp;&nbsp;'
        footer = Template(foot_text)
    return(footer.substitute(links=link_string))
```

Turn the dictionary of links into a string, which is then substituted into the template.

Similar to the "include_header" function, this one uses its single string as its argument to create the end of a HTML page. The page itself is stored within a separate file in "templates/footer.html", and the argument is used to dynamically create a set of HTML link tags. Based on how they are used, it looks like the argument needs to be a dictionary.

This looks a little weird, but it's an HTML hack for forcing spaces into a string.

```
def start_form(the_url, form_type="POST"):
    return('<form action="' + the_url + '" method="' + form_type + '">'')
```

This is typically either "POST" or "GET".

This function returns the HTML for the start of a form and lets the caller specify the URL to send the form's data to, as well as the method to use.

```
def end_form(submit_msg="Submit"):
    return('<p></p><input type=submit value="' + submit_msg + '">''></form>')
```

This function returns the HTML markup, which terminates the form while allowing the caller to customize the text of the form's "submit" button.

```
def radio_button(rb_name, rb_value):
    return('<input type="radio" name="' + rb_name +
          '" value="' + rb_value + '">''> ' + rb_value + '<br />')
```

Given a radio-button name and value, create a HTML radio button (which is typically included within a HTML form). Note: both arguments are required.

```
def u_list(items):
    u_string = '<ul>'
    for item in items:
        u_string += '<li>' + item + '</li>'
    u_string += '</ul>'
    return(u_string)
```

A simple "for" loop does the trick.

Given a list of items, this function turns the list into a HTML unnumbered list. A simple "for" loop does all the work, adding a LI to the UL element with each iteration.

```
def header(header_text, header_level=2):
    return('<h' + str(header_level) + '>' + header_text +
          '</h' + str(header_level) + '>')
```

Create and return a HTML header tag (H1, H2, H2, and so on) with level 2 as the default.. The "header_text" argument is required.

```
def para(para_text):
    return('<p>' + para_text + '</p>')
```

Enclose a paragraph of text (a string) in HTML paragraph tags. Almost not worth the effort, is it?

Q: Where are the HTML templates used in the `include_header()` and `include_footer()` functions?

A: They are included with the `yate` module's download. Go ahead and grab them from the *Head First Python* support website, and put them into a folder of your choice.

Q: Why do I need `yate` at all? Why not include the HTML that I need right in the code and generate it with `print()` as needed?

A: You could, but it's not as flexible as the approach shown here. And (speaking from bitter experience) using a collection of `print()` statements to generate HTML works, but it turns your code into an *unholy mess*.

Q: And you did this because you are using MVC?

A: Partly, yes. The reason the MVC pattern is being followed is to ensure that the model code is separate from the view code, which are both separate from the controller code. No matter the size of the project, following MVC can make your life easier.

Q: But surely MVC is overkill for something this small?

A: We don't think so, because you can bet that your webapp will grow, and when you need to add more features, the MVC "separation of duties" really shines.



An IDLE Session

Let's get to know the `yate` module even more. With the code downloaded and tucked away in an easy-to-find folder, load the module into IDLE and press F5 to take it for a spin. Let's start by testing the `start_response()` function. The CGI standard states that every web response must start with a header line that indicates the type of the data included in the request, which `start_response()` lets you control:

```
>>> start_response()
'Content-type: text/html\n\n'           ← The default CGI response header,
                                         plus variations on a theme.
>>> start_response("text/plain")
'Content-type: text/plain\n\n'           ←
>>> start_response("application/json")
'Content-type: application/json\n\n'
```

The `include_header()` function generates the start of a web page and lets you customize its title:

```
>>> include_header("Welcome to my home on the web!")
<html>\n<head>\n<title>Welcome to my home on the web!</title>\n<link type="text/css"
rel="stylesheet" href="/coach.css" />\n</head>\n<body>\n<h1>Welcome to my home on the web!<
h1>\n'
```



This all looks a little bit messy, but don't worry; it's meant to be processed by your web browser, NOT by you. Your web browser will have no difficulty working with this HTML. Note the inclusion of a link to a CSS file (more on this in a bit).

The `include_footer()` function produces HTML that terminates a web page, providing links (if provided as a dictionary). An empty dictionary switches off the inclusion of the linking HTML:

```
>>> include_footer({'Home': '/index.html', 'Select': '/cgi-bin/select.py'})
'<p>\n<a href="/index.html">Home</a>&nbsp;&nbsp;&nbsp;&nbsp;<a href="/cgi-bin/select.py">Select</a>&nbsp;&nbsp;&nbsp;&nbsp;\n</p>\n</body>\n</html>\n'
>>> include_footer({})
'<p>\n\n</p>\n</body>\n</html>\n'
```

With links included, and without.

The `start_form()` and `end_form()` functions bookend a HTML form, with the parameter (if supplied) adjusting the contents of the generated HTML:

```
>>> start_form("/cgi-bin/process-athlete.py")
'<form action="/cgi-bin/process-athlete.py" method="POST">'
>>> end_form()
'<p></p><input type=submit value="Submit"></form>'
>>> end_form("Click to Confirm Your Order")
'<p></p><input type=submit value="Click to Confirm Your Order"></form>'
```

The argument allows you to specify the name of the program on the server to send the form's data to.

HTML radio buttons are easy to create with the `radio_button()` function:

```
>>> for fab in ['John', 'Paul', 'George', 'Ringo']:
    radio_button(fab, fab)

'<input type="radio" name="John" value="John"> John<br />'
'<input type="radio" name="Paul" value="Paul"> Paul<br />'
'<input type="radio" name="George" value="George"> George<br />'
'<input type="radio" name="Ringo" value="Ringo"> Ringo<br />'
```

Which one is your favorite?
Select from the list of radio buttons.

Unordered list are a breeze with the `u_list()` function:

```
u_list(['Life of Brian', 'Holy Grail'])
'<ul><li>Life of Brian</li><li>Holy Grail</li></ul>'
```

Again, not too easy on your eye, but fine as far as your web browser is concerned.

The `header()` function lets you quickly format HTML headings at a selected level (with 2 as the default):

```
>>> header("Welcome to my home on the web")
'<h2>Welcome to my home on the web</h2>'
>>> header("This is a sub-sub-sub-sub heading", 5)
'<h5>This is a sub-sub-sub-sub heading</h5>'
```

Nothing too exciting here, but it works as expected. Same goes for here.

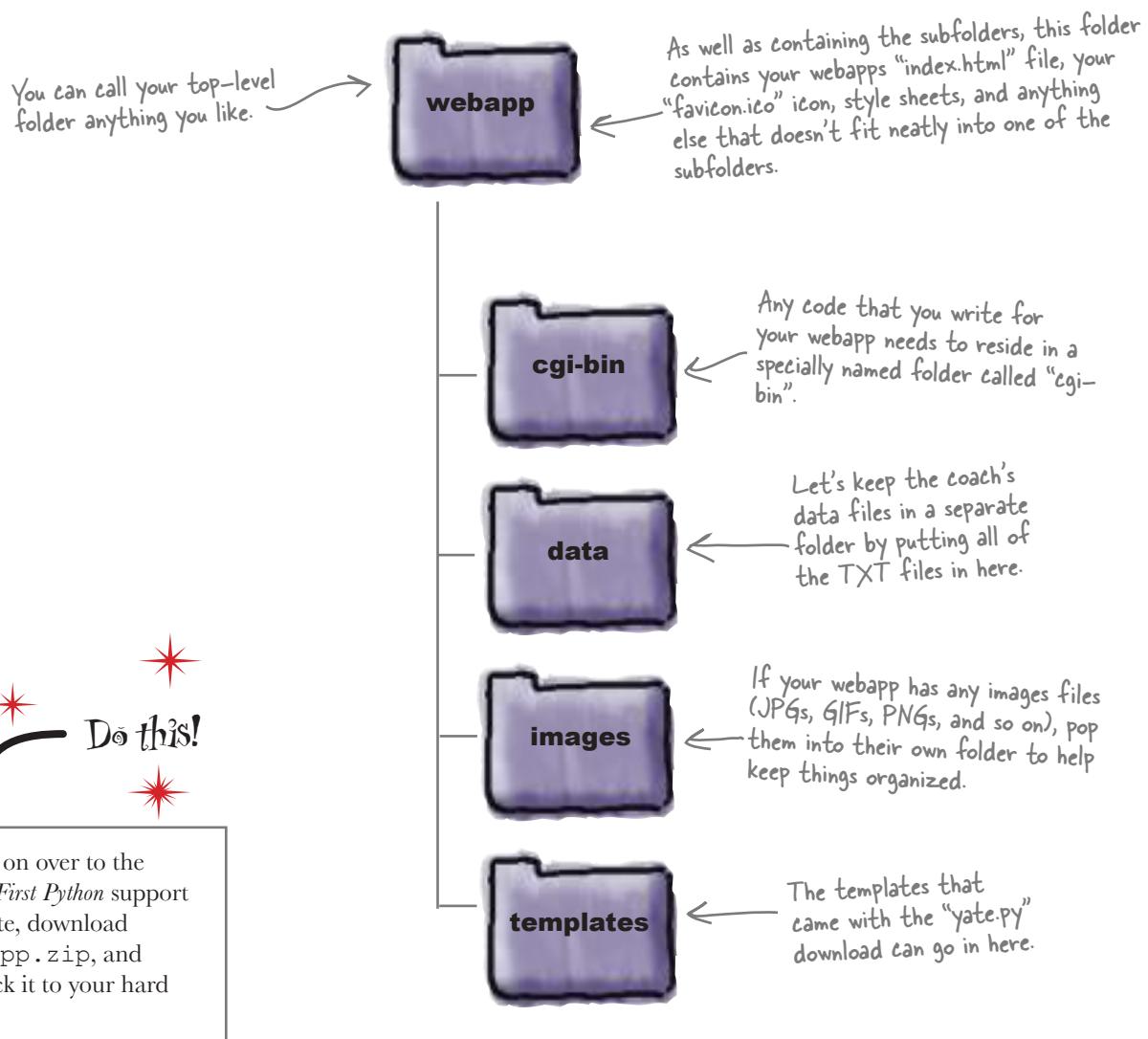
Last, but not least, the `para()` function encloses a chunk of text within HTML paragraph tags:

```
>>> para("Was it worth the wait? We hope it was...")
'<p>Was it worth the wait? We hope it was...</p>'
```

Control your code

Your **model** code is ready, and you have a good idea of how the `yate` module can help you with your **view** code. It's time to glue it all together with some **controller** code.

First things first: you need to arrange your wedapp's directory structure to help keep things organized. To be honest, anything goes here, although by giving it a little thought, you can enhance your ability to extend your webapp over time. Here's one folder structure that *Head First Labs* recommends.



CGI lets your web server run programs

The Common Gateway Interface (CGI) is an Internet standard that allows for a web server to run a **server-side program**, known as a *CGI script*.

Typically, CGI scripts are placed inside a special folder called `cgi-bin`, so that the web server knows where to find them. On some operating systems (most notably UNIX-styled systems), CGI scripts must be set to *executable* before the web server can execute them when responding to a web request.



Import the HTTP
server and CGI
modules.

Specify a port.

Create a
HTTP server.

Display a friendly
message and start
your server.

So...to run my webapp,
I need a web server with
CGI enabled.



All webapps need to run on web servers.

Practically every web server on the planet supports CGI. Whether you're running *Apache*, *IIS*, *nginx*, *Lighttpd*, or any of the others, they all support running CGI scripts written in Python.

But using one of these tools here is *overkill*. There's no way the coach is going to agree to download, unpack, install, configure, and manage one of these industry heavyweights.

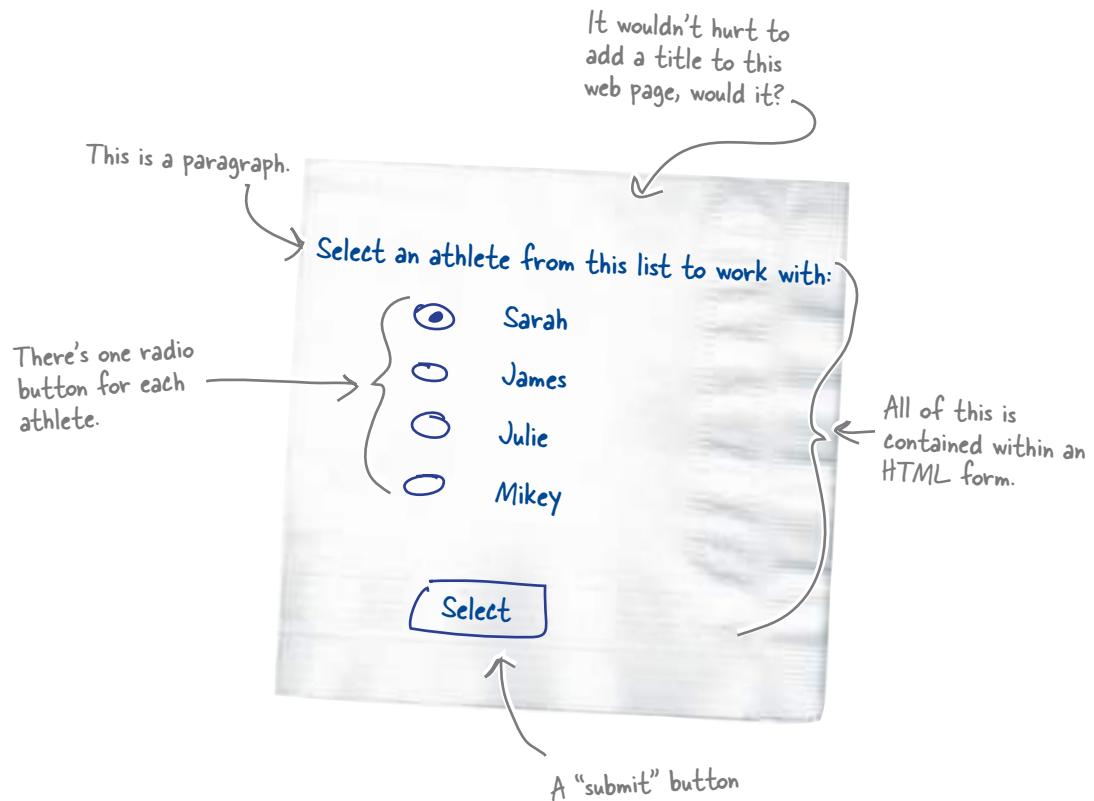
As luck would have it, Python comes with its very own web server, included in the `http.server` library module. Check the contents of the `webapp.zip` download: it comes with a CGI-enabled web server called `simplehttpd.py`.

Here are the five lines of code needed to build a web server in Python.

```
from http.server import HTTPServer, CGIHTTPRequestHandler
port = 8080
httpd = HTTPServer(('', port), CGIHTTPRequestHandler)
print("Starting simple_httpd on port: " + str(httpd.server_port))
httpd.serve_forever()
```

Display the list of athletes

Let's create a program called `generate_list.py` which, when executed by the web server, dynamically generates a HTML web page that looks something like this:



When your user selects an athlete by clicking on her radio button and clicking Select, a *new* web request is sent to the web server. This new web request contains data about which radio button was pressed, *as well as the name of a CGI script to send the form's data to*.

Recall that all of your CGI scripts need to reside in the `cgi-bin` folder on your web server. With this in mind, let's make sure your `generate_list.py` CGI script sends its data to another program called:

`cgi-bin/generate_timing_data.py`

Pool Puzzle



Your **job** is to take the code from the pool and place them into the blank lines in the CGI script. You may **not** use the same line of code more than once. Your **goal** is to make a CGI script that will generate a HTML page that matches the hand-drawn design from the previous page.

I've started things off for you.

```
import athletemodel
import yate
import glob
```

Import the modules that you need. You've already met "athletemodel" and "yate". The "glob" module lets you query your operating system for a list of file names.

```
data_files = glob.glob("data/*.txt")
athletes = athletemodel.put_to_store(data_files)
```

Use your "put_to_store()" function to create a dictionary of athletes from the list of data files.

Let's add a link to the bottom of the generated HTML page that takes your user home.

```
print(yate.include_footer({ "Home": "/index.html" }))
```

Note: each thing from the pool can be used once!

```
print(yate.start_form("generate_timing_data.py"))
print(yate.para("Select an athlete from the list to work with:"))
print(yate.include_header("Coach Kelly's List of Athletes"))
print(yate.radio_button("which_athlete", athletes[each_athlete].name))
for each_athlete in athletes:
    print(yate.start_response())
    print(yate.end_form("Select"))
```

Pool Puzzle Solution



Your **job** was to take the code from the pool and place them into the blank lines in the CGI script. You were **not** to use the same line of code more than once. Your **goal** was to make a CGI script that generates a HTML page that matches the hand-drawn design.

Always start with a Content-type line.

Start generating the form, providing the name of the server-side program to link to.

Generate a radio-button for each of your athletes.

```
import athletemodel
import yate
import glob

data_files = glob.glob("data/*.txt")
athletes = athletemodel.put_to_store(data_files)

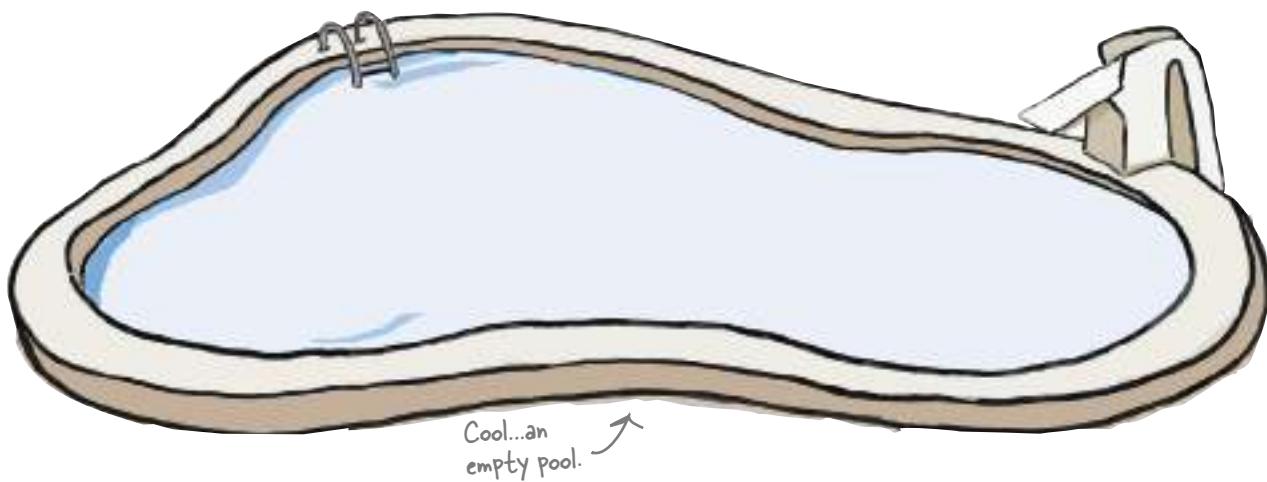
print(yate.start_response())
print(yate.include_header("Coach Kelly's List of Athletes"))
print(yate.start_form("generate_timing_data.py"))
print(yate.para("Select an athlete from the list to work with:"))

for each_athlete in athletes:
    print(yate.radio_button("which_athlete", athletes[each_athlete].name))

print(yate.end_form("Select")) ← End the form generation with a custom "Submit" button.
print(yate.include_footer({"Home": "/index.html"}))
```

Start generating the web page, providing an appropriate title.

A paragraph telling your user what to do





What you need to do next depends on the operating system you're running your web server on.

Watch it!

If you are running on Windows, stop reading right now and proceed to the Test Drive. However, if you are running a Unix-based system (such as Linux, Mac OS X, or BSD) you need to do two things to prepare your CGI script for execution:

1. Set the **executable bit** for your CGI using the **chmod +x** command.
2. Add the following line of code to the very top of your program:

```
#!/usr/local/bin/python3
```

From your terminal window, type
chmod +x generate_list.py to set the executable bit. You
need do this only once.



Test Drive

To test drive your CGI script, you need to have a web server up and running. The code to `simplehttpd.py` is included as part of the `webapp.zip` download. After you unpack the ZIP file, open a terminal window in the `webapp` folder and start your web server:

```
File Edit Window Help WebServerOnWindows
c:\Python31\python.exe simplehttpd.py
Starting simple_httpd on port: 8080
```

Use this command
on Windows-based
systems.

```
File Edit Window Help WebServerOnUNIX
$ python3 simple_httpd.py
Starting simple_httpd on port: 8080
```

Use this command
on Unix-based
systems.



—Test DRIVE, continued—

With your web server running, let's load up Coach Kelly's home page and get things going. You've started your web server running on port 8080 on your computer, so you need to use the following web address in your web browser: `http://localhost:8080`.

The screenshot shows a web browser window titled "Welcome to Coach Kelly's Website". The URL bar contains "`http://localhost:8080/`". The main content area features a large photo of a smiling man in a red t-shirt and a "Welcome to Coach Kelly's Website." heading. Below it, text reads: "For now, all that you'll find here is my athlete's **timing data**. Enjoy! See you on the track!" A cursor arrow points from the text "See you on the track!" towards the "timing data" link.

The "timing data" hyperlink is waiting for you to click it.

The coach's home page appears in your browser. It's called "index.html" and it is included in the "webapp.zip" download.

...and your web server springs into life, logging (to the screen) any and all web requests that it processes.

The screenshot shows a terminal window with the title "DisplayingHomePage". The command entered was `$ python3 simple_httpd.py`. The output shows the server starting and then logging several requests for files like index.html, coach.css, and favicon.ico.

```
File Edit View Help DisplayingHomePage
$ python3 simple_httpd.py
Starting simple_httpd on port: 8080
localhost - - [12/Sep/2010 14:30:03] "GET / HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:30:03] "GET /coach.css HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:30:03] "GET /images/coach-head.jpg HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:30:03] "GET /favicon.ico HTTP/1.1" 200 -
```

Sure enough, clicking on the home page's link runs the `generate_list.py` program on the web server, which displays Coach Kelly's athletes as a list of radio buttons.

The screenshot shows a Mac OS X desktop environment. In the foreground, a web browser window titled "Coach Kelly's List of Athletes" is open at the URL `http://localhost:8080/cgi-bin/generate_list.py`. The page content includes a heading "Coach Kelly's List of Athletes", a list of athletes with radio buttons, and a "Select" button. A "Home" link is also present. In the background, a terminal window displays the command `$ python3 simple_httpd.py` followed by a log of HTTP requests from the local host. Annotations with arrows point from the text to the browser and terminal windows:

- An annotation points from the text "Looking good...the web page has been generated correctly..." to the browser window.
- An annotation points from the text "...and your web server logs the web request to run the 'generate_list.py' CGI script." to the terminal window.

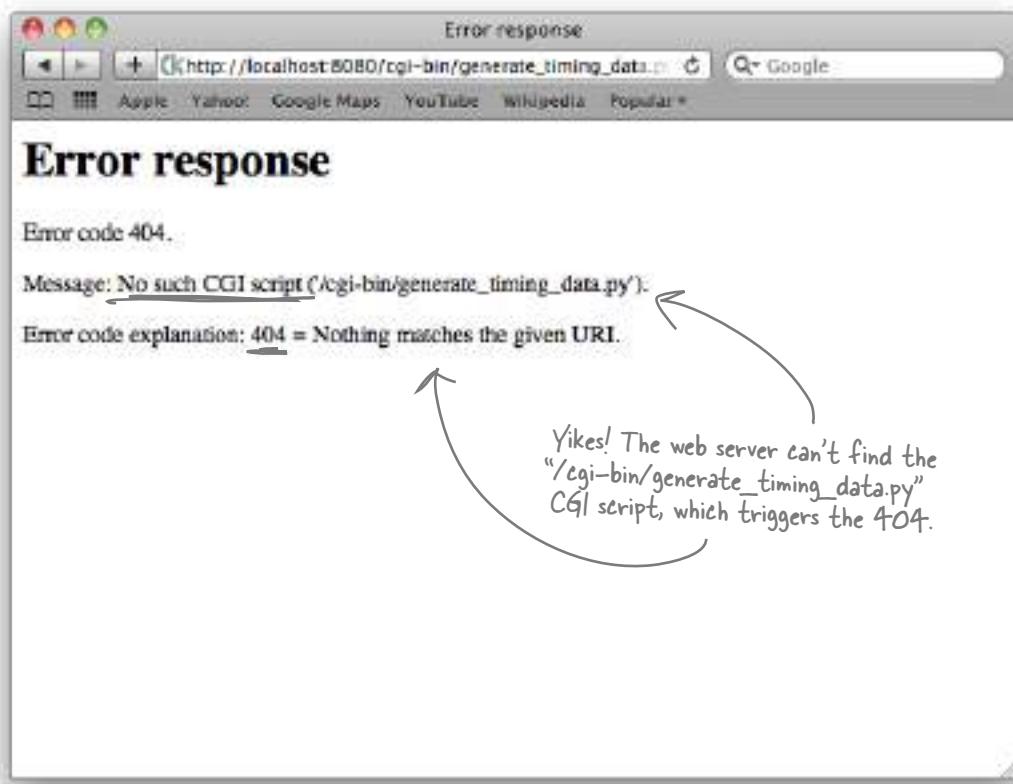
```
File Edit Window Help DisplayingHomePage
$ python3 simple_httpd.py
Starting simple_httpd on port: 8080
localhost - - [12/Sep/2010 14:30:03] "GET / HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:30:03] "GET /coach.css HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:30:03] "GET /images/coach-head.jpg HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:30:03] "GET /favicon.ico HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:45:16] "GET /cgi-bin/generate_list.py HTTP/1.1" 200 -
```

You can click the Home hyperlink to return to the coach's home page, or select an athlete from the list (by clicking on their radio-button), before pressing the Select button to continue.

Select an athlete and press Select. What happens?

The dreaded 404 error!

Whoops! Your web server has responded with a “404” error code, which is its way of telling you that something was wrong with *your* request. The web server is in fact telling you that it can’t locate the resource that your web browser requested, so it’s telling you that *you* made a mistake:



Check the web server’s console window to confirm that your attempt to post your form’s data to `generate_timing_data.py` resulted in failure.

Which isn’t really that surprising seeing as *you have yet to write that code!* So...things aren’t as bad as they first appear. The “404” error is *exactly* what you would expect to be displayed in this situation, so your `generate_list.py` CGI is working fine. What’s needed is the code to the *other* CGI script.

If you create the required CGI script, you’ll be back on track.

Fireside Chats



Tonight's talk: **To be CGI or not to be CGI, that is the question.**

A Python Program:

Listen: you're really not all that different than me; you just work on a web server, whereas I can work *anywhere*.

Special?!? But you only work on the Web, nowhere else. How's that "special"?

Nonsense! The truth is that you work *only* on the Web and break pretty quickly when used elsewhere. You don't even have control over your own I/O.

Like [sniggers] generating text in the form of HTML? That's really taxing...

Oh, get over yourself! You're a regular program, just like me. I can generate HTML, too, I just choose not to.

I guess so...

Ummmm...I guess so.

A Python CGI Script:

Yes. I like to think of myself as *special*.

Because all the cool stuff works on the Web these days and I'm designed, optimized, tailored, and engineered for the Web. Because the Web's a cool place, it follows that I must be cool, too. See: *special*.

I don't need control over my input and output. I have a friendly web server to take care of that for me. My input *comes from* the web server and my output *goes to* the web server. This arrangement allows me to concentrate on the important stuff.

Smirk all you want; HTML makes the World Wide Web go around and I'm a master at generating it *dynamically, on demand, and as needed*. Without me, the Web would be a pretty static place.

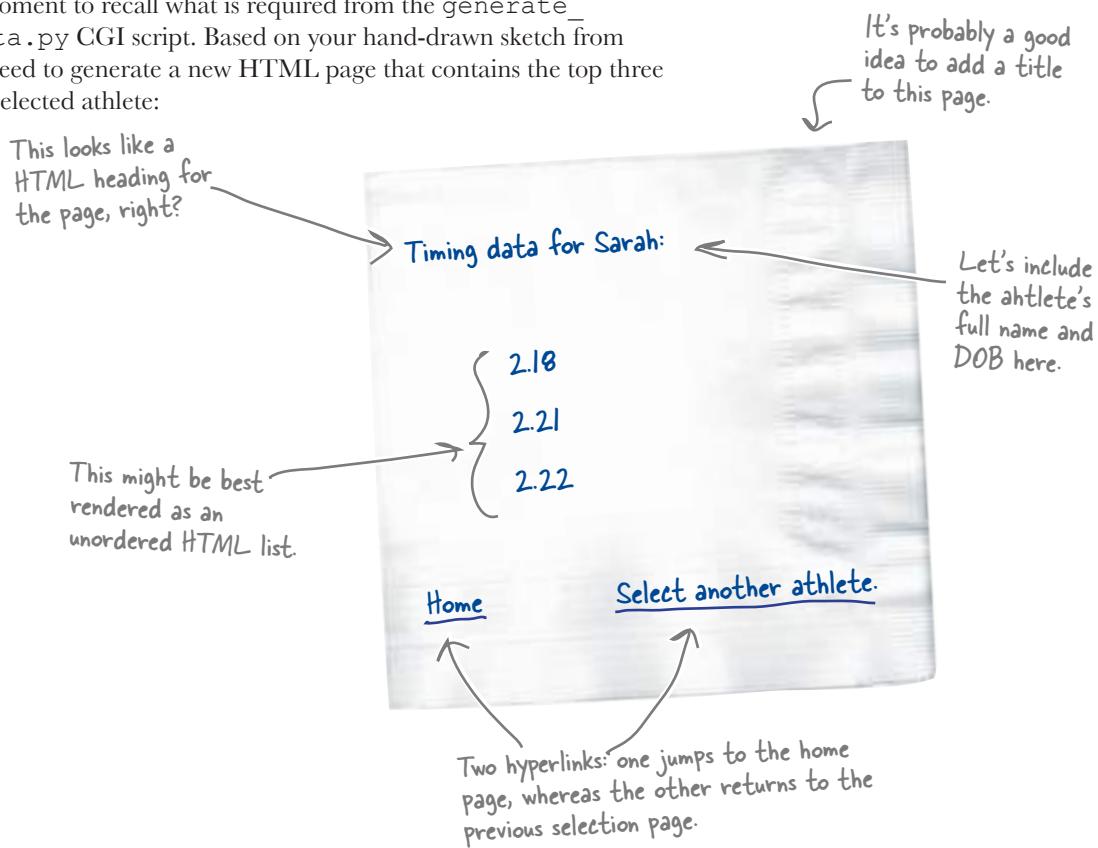
And if you did generate HTML, you'd want it displayed somewhere...like in a browser?

And to do that you'd need to rely on the services of a friendly web server, right?

Which would make you a CGI script. So, you'd be *special*, too. Q.E.D.

Create another CGI script

Let's take a moment to recall what is required from the `generate_timing_data.py` CGI script. Based on your hand-drawn sketch from earlier, your need to generate a new HTML page that contains the top three times for the selected athlete:



But how do you know which athlete is selected?

When you click on a radio-button and then press the Select button, a new web request is sent to the server. The web request identifies the CGI script to execute (in this case, that's `generate_timing_data.py`), together with the form's data. The web server arranges to send the form's data to your CGI script *as its input*. Within your code, you can access the form data using Python's `cgi` module, which is part of the standard library:

```
import cgi  
form_data = cgi.FieldStorage()  
athlete_name = form_data['which_athlete'].value
```

Import the "cgi" library.

Grab all of the form data and put it in a dictionary.

Access a named piece of data from the form's data.



Write the code
to your new CGI
script here. ✓

It's time to exercise your newly acquired web-coding chops. Grab your pencil and write the code for the `generate_timing_data.py` CGI script. It's not too different from the `generate_list.py` code, so you should be able to reuse a lot of your existing code.



It's time to exercise your newly acquired web-coding chops. You were to grab your pencil and write the code for the generate_timing_data.py CGI script. It's not too different from the generate_list.py code, so you should be able to reuse a lot of your existing code.

```
#!/usr/local/bin/python3
```

← This line is needed on Unix-based systems only.

```
import cgi
import athletemodel
import yate
```

Import the libraries and modules you intend to use.

Get the data from the model. → `athletes = athletemodel.get_from_store()`

Which athlete's data are you working with?

```
form_data = cgi.FieldStorage()
athlete_name = form_data['which_athlete'].value
```

Nothing new here or here.

```
print(yate.start_response())
print(yate.include_header("Coach Kelly's Timing Data"))
```

```
print(yate.header("Athlete: " + athlete_name + ", DOB: " +
    athletes[athlete_name].dob + ".))
```

Grab the athlete's name and DOB.

```
print(yate.para("The top times for this athlete are:"))
```

```
print(yate.u_list(athletes[athlete_name].top3()))
```

Turn the top three list into an unordered HTML list.

The bottom of this web page has two links.

```
print(yate.include_footer({"Home": "/index.html",
    "Select another athlete": "generate_list.py"}))
```

↑ A link back to the previous CGI script.



Test DRIVE

Note: If you are on a Unix-based system, don't forget to add "chmod +x generate_timing_data.py" to set the executable bit.

Your web server should still be running from earlier. If it isn't, start it again. In your web browser, return to the coach's home page, then select the hyperlink to display the list of athletes, select Sarah, and then press the button.

This all looks OK.

Ah, phooey!
Something's not
quite right here.
Where's Sarah's
top three times?

```

File Edit Window Help HoustonWeHaveAProblem
$ python3 simple_httpd.py
Starting simple_httpd on port: 8080
localhost - - [12/Sep/2010 14:30:03] "GET / HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:30:03] "GET /coach.css HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:30:03] "GET /images/coach-head.jpg HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:30:03] "GET /favicon.ico HTTP/1.1" 200 -
localhost - - [12/Sep/2010 14:45:16] "GET /cgi-bin/generate_list.py HTTP/1.1" 200 -
localhost - - [12/Sep/2010 16:12:27] "GET /cgi-bin/generate_list.py HTTP/1.1" 200 -
localhost - - [12/Sep/2010 16:12:29] "POST /cgi-bin/generate_timing_data.py HTTP/1.1" 200 -
Traceback (most recent call last):
  File "/Users/barryp/HeadFirstPython/chapter7/cgi-bin/generate_timing_data.py", line 21, in <module>
    print(yate.u_list(athletes[athlete_name].top3()))
TypeError: 'list' object is not callable
localhost - - [12/Sep/2010 16:12:29] CGI script exit status 0x100

```

Does the
web server's
logging
information
tell you
anything?

[View Solution](#)

Your CGI has suffered from a `TypeError` exception, but other than looking at the web server's logging screen, it's not clear *on the web browser screen* that anything has gone wrong.



BRAIN BARBELL

What do you think is the problem here? Take a moment to study the error message before flipping the page.

Enable CGI tracking to help with errors

The CGI standard dictates that any output generated by a server-side program (your CGI script) should be captured by the web server and sent to the waiting web browser. Specifically, anything sent to STDOUT (standard output) is captured.

When your CGI script raises an exception, Python arranges for the error message to display on STDERR (standard error). The CGI mechanism is programmed to ignore this output because all it wants is the CGI script's standard output.



This behavior is fine when the webapp is deployed, but *not* when it's being developed. Wouldn't it be useful to see the details of the exception in the browser window, as opposed to constantly having to jump to the web server's logging screen?

Well...guess what? Python's standard library comes with a CGI tracking module (called `cgitb`) that, when enabled, arranges for detailed error messages to appear in your web browser. These messages can help you work out where your CGI has gone wrong. When you've fixed the error and your CGI is working well, simply switch off CGI tracking:

```
import cgitb  
cgitb.enable()
```

Add these two lines near the start of your CGI scripts to enable Python's CGI tracking technology.



Test DRIVE

Add the two CGI tracking lines of code near the top of your `generate_timing_data.py` CGI script. Press the Back button on your web browser and press the Select button again. Let's see what happens this time.

Wow! Look at all of this detail.

Coach Kelly's Timing Data

Athlete: Sarah Sweeney, DOB: 2002-6-17.

The top times for this athlete are:

TypeError

/Library/Frameworks/Python.framework/Versions/3.1/Resources/Python.app/Contents/MacOS
Sun Sep 12 20:13:21

A problem occurred in a Python script. Here is the sequence of function calls leading up to the error, in the order they occurred:

```

/Users/barryp/HeadFirstPython/chapter7/cgi-bin/generate_timing_data.py in ()
 9         athletes[athlete_name].dob + "."))
10     print(yate.para("The top times for this athlete are:"))
-> 21     print(yate.u_list(athletes[athlete_name].top3()))
22     print(yate.include_footer({'None': '/index.html',
23                               "Select another athlete": "generate_list.py"}))

built-in print = <built-in function print>, yate = <module 'yate' from '/Users/barryp/HeadFirstPython/chapter7/cgi-bin/yate.py'>, yate.u_list = <function u_list>, athletes = {'James Lee': ['2:38', '1:21', '2:34', '2:45', '3:01', '2:02', '2:01', '3:10', '2:22', '2:03', '2:01', '2:16'], 'Julia Jones': ['2:59', '2:11', '2:11', '2:23', '3:10', '2:23', '3:10', '3:10', '3:21', '3:21', '3:01', '3:02', '2:59'], 'Mickey McManus': ['2:22', '3:01', '3:01', '3:02', '3:02', '3:22', '2:49', '2:38', '2:40', '2:22', '2:31'], 'Sarah Sweeney': ['2:58', '2:58', '2:39', '2:25', '2:55', '2:54', '2:18', '2:55', '2:25', '2:22', '2:22', '2:21', '2:22']}, athlete_name = 'Sarah Sweeney', top3 undefined

```

TypeError: 'list' object is not callable
args = ("list' object is not callable",)
with_traceback = <built-in method with_traceback of TypeError object>

Notice that the CGI tracking module tries to identify exactly where the problem with your code lies.

What's this? `top3()` is undefined!?

A small change can make all the difference

The CGI tracking output indicates an error with the use of the `top3()` method from the `AthleteList` class.

A quick review of the code to the `AthleteList` class uncovers the source of the error: the `top3()` method has been redesignated as a **class property**.

```
@property  
def top3(self):  
    return(sorted(set([self.sanitize(t) for t in self]))[0:3])
```

This decorator allows you to access the data returned by "top3()" as if it were a class attribute.

The use of the `@property` decorator allows the `top3()` method to appear like an attribute to users of the class. So, instead of calling the `top3()` method like this:

```
print(yate.u_list(athletes[athlete_name].top3()))
```

A method call always needs the parentheses...

Treat the `top3()` method as if it was another class attribute, and call it like this:

```
print(yate.u_list(athletes[athlete_name].top3))
```

...unless the method is declared to be an "`@property`", in which case parentheses are NOT required.

It's a small change, but it's an important one

When a change is made to the way a class is used, you need to be careful to consider what impact the change has on existing programs, both yours and those written by others.

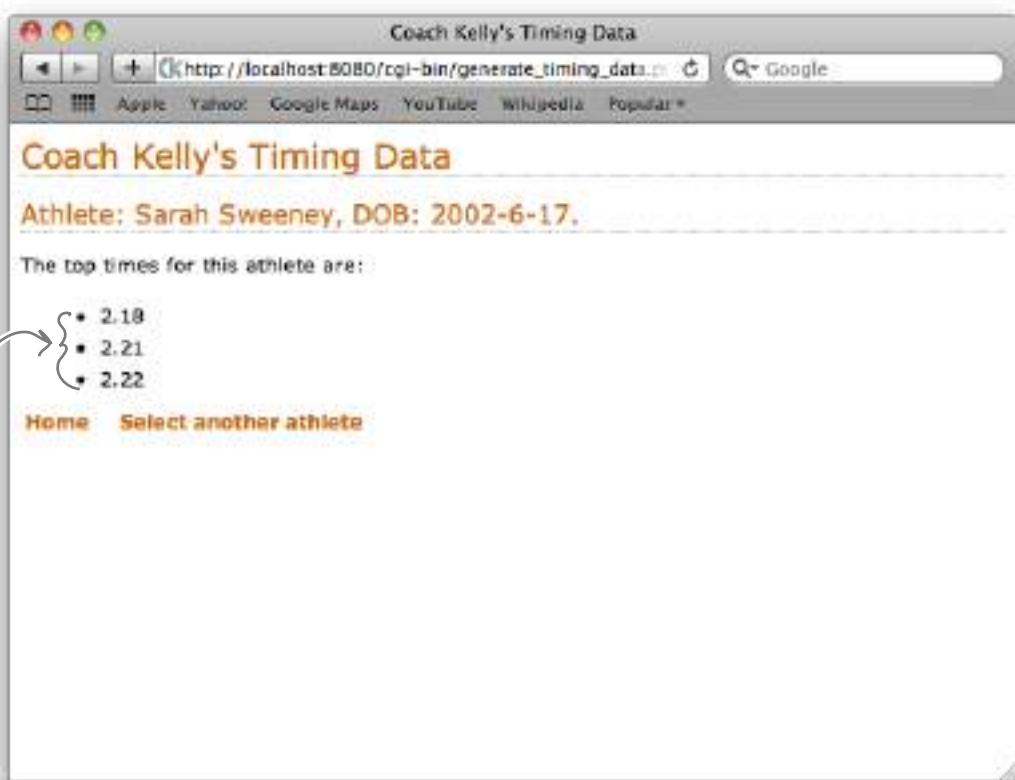
At the moment, you are the only one using the `AthleteList` class, so it's not a big deal to fix this. But imagine if thousands of programmers were using and relying on your code...

Let's fix your CGI script and try again.



Test DRIVE

Make the small edit to your code to remove the brackets from the call to the `top3()` method, press your web browser's Back button, and press the Select button one last time.



there are no
Dumb Questions

Q: What happens if the coach recruits new athletes?

A: All Coach Kelly needs do is create a new text file similar to the others, and your webapp handles the rest by dynamically including the new athlete the next time your webapp runs, which occurs when someone clicks on the home page's "timing data" hyperlink.

Q: Shouldn't the server's data be in a database as opposed to a pickle? Surely that would be better, right?

A: In this case, it's probably overkill to use a database, but it might be worth considering sometime in the future.

Your webapp's a hit!



By moving your program to the Web, you've made it a no-brainer for Coach Kelly to share his data with not only his athletes, but with *anyone* else that needs to access his data.

By conforming to the **MVC** pattern and using **CGI**, you've built a webapp in such a way that it's easy to extend as new requirements are identified.

Congratulations! You're a web developer.



Your Python Toolbox

You've got Chapter 7 under your belt and you've added some key Python techniques to your toolbox.

Python Lingo

- “@property” – a decorator that lets you arrange for a class method to appear as if it is a class attribute.

Web Lingo

- “webapp” – a program that runs on the Web.
- “web request” – sent from the web browser to the web server.
- “web response” – sent from the web server to the web browser in response to a web request.
- “CGI” – the Common Gateway Interface, which allows a web server to run a server-side program.
- “CGI script” – another name for a server-side program.



BULLET POINTS

- The **Model-View-Controller** pattern lets you design and build a webapp in a maintainable way.
- The **model** stores your webapp's data.
- The **view** displays your webapp's user interface.
- The **controller** glues everything together with programmed logic.
- The standard library `string` module includes a class called `Template`, which supports simple string substitutions.
- The standard library `http.server` module can be used to build a simple web server in Python.
- The standard library `cgi` module provides support for writing CGI scripts.
- The standard library `glob` module is great for working with lists of filenames.
- Set the **executable** bit with the `chmod +x` command on Linux and Mac OS X.
- The standard library `cgitb` module, when enabled, lets you see CGI coding errors within your browser.
- Use `cgitb.enable()` to switch on CGI tracking in your CGI code.
- Use `cgi.FieldStorage()` to access data sent to a web server as part of a web request; the data arrives as a Python dictionary.

8 mobile app development



Small devices



Putting your data on the Web opens up all types of possibilities.

Not only can anyone from anywhere interact with your webapp, but they are increasingly doing so from a collection of diverse computing devices: PCs, laptops, tablets, palmtops, and even mobile phones. And it's not just humans interacting with your webapp that you have to support and worry about: *bots* are small programs that can automate web interactions and typically want your data, not your human-friendly HTML. In this chapter, you exploit Python on Coach Kelly's mobile phone to write an app that interacts with your webapp's data.

The world is getting smaller

Coach Kelly is continuing to use his webapp every day, but he's having a problem with his new smartphone.



There's more than just desktop computers out there.

Who knew that your users would try to interact with your webapp using something other than a desktop computer or laptop?

It's a diverse computing environment out there.

Coach Kelly is on Android

The coach has a lovely new smartphone that's running Google's Android operating system. Sure enough, when you check it out, the webapp is way too small and not much use on the coach's three-inch screen:



And don't go telling
me to do all that two-
fingered zoom and double-
tapping thing. That just
drives me crazy!



Obviously, the coach needs to access his data and run his webapp on his phone...but what's the *best way* to do this if not through the phone's browser?



Sharpen your pencil

Open your web browser on your desktop computer (or phone) and enter "Python for Android" into your favorite search engine. Make a note in the space below of the most promising site from your search results:



Is this the one
you found?

You were to open your web browser on your desktop computer (or phone) and enter "Python for Android" into your favorite search engine. You were then to make a note in the space below of the most promising site from your search results:

<http://code.google.com/p/android-scripting> (the home of the SL4A project.)

Run Python on the coach's smartphone

A quick search of the Web uncovers a pleasant surprise: *Python runs on Android*.

At least a *version* of Python runs on Android. A project called *Scripting Layer for Android (SL4A)* provides technology to let you run Python on any Android device. But there's a catch.



Yes. SL4A ships with Python 2, not 3.

Python 3, this book's preference, is the best version of Python yet, but it achieves its greatness at the cost of a lack of backward compatibility. There's some stuff in 3 that will never work in 2 and vice versa.

Is this fact alone a *show-stopper*?

Don't worry about Python 2

The fact that Python 2 is available for Android and you've learned Python 3 in this book is nothing to lose sleep over. *Python 2 is still Python*, and the differences between Python 2 and Python 3 are easy to manage.

Think about your webapp for a minute. Right now, the model, view, and controller code resides on the web server, which is running Python 3.

Your web browser
runs here.



The Internet

I'm quite happy
to run Python 3
all day long...

Web
Server

All of your webapp
code runs here.

Python 3 is still
going strong...

If you move the user interaction to the smartphone, the model and some of the controller code *stay* on the server (and continue to run on Python 3), whereas the view code and the rest of the controller code *move* to the smartphone, where they need to be rewritten to run on Python 2.

Run Python code?
No problem...just so
long as it's Python 2.

Half your webapp's
code runs here...



The Internet

Web
Server

...and the other
half runs here.

Set up your development environment

Understandably, the coach won't let you have his phone to work on until you have something that works. Thankfully, *Google* provides a cross-platform **Android emulator** that lets you develop for the phone as needed, even though you don't own any hardware.

Download the Software Development Kit (SDK)

Let's get started developing for Android. Visit this website and download the SDK for your computer and operating system:

<http://developer.android.com/sdk/index.html>



Follow along with these instructions to ensure you have your Android development environment correctly set up on your computer.

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r17-windows.zip	23699014 bytes	934d0260900f02315894f19e57410
Mac OS X (Intel)	android-sdk_r17-mac_x86.zip	19229546 bytes	0330ex0eb006790bf6dc72c5e0f919
Linux (x86)	android-sdk_r17-linux_x86.zip	17114817 bytes	e10cf5da3dfeaf147dd4edc58bb004d

The Android
SDK website

Despite what this website might look like it's telling you, you do **not** need to install *Eclipse* to run the Android emulator. However, you do need to have a *Java Runtime Environment* installed. If you are unsure about this, don't worry: the Android emulator will advise your best course of action if it spots that Java is missing.

Note: This is how the Android SDK download page looks at the time of this writing. It might look a little different for you. No worries: just download the latest version of the SDK.

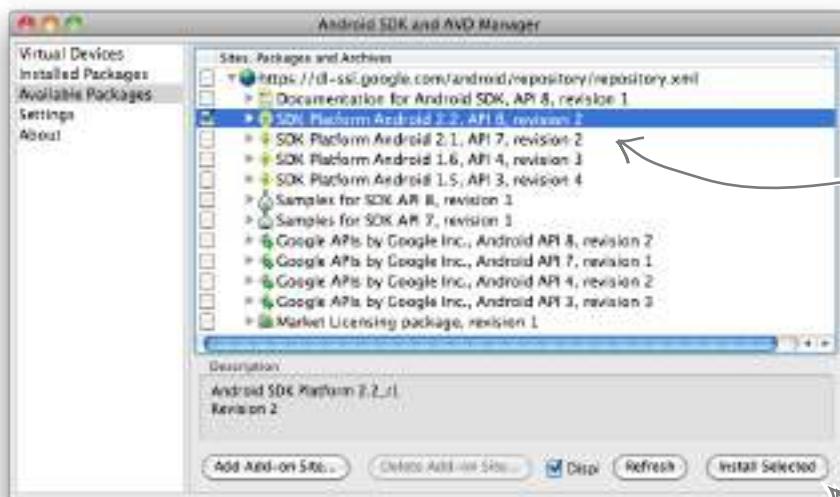
Configure the SDK and emulator

You need to do *two things* to configure the SDK and emulator: add an *Android Platform* and create an *Android Virtual Device* (known as an *AVD*).

Add an Android platform

The coach is running **Android 2.2** on his phone, so let's add a 2.2 platform to mimic this setup. Open up the Android SDK and AVD Manager tool, select Available Packages, and pick 2.2 for installation.

The Android download contains a folder called "tools". Run the "android" program within this folder.



This is the only version of the SDK that you need.

This might take a minute or two, depending on the speed of your network connection.

Create a new Android Virtual Device (AVD)

With the 2.2 platform downloaded and installed, create a new Android Virtual Device.



Give your AVD a name, and select a target.

Set the size of the virtual SDcard: 512 is more than enough.

Click on "Create AVD".

Your AVD is a simulated Android phone.

Install and configure Android Scripting

With the emulator ready, use the AVD Manager to start your 2.2 device. Click on the emulator's browser (the little globe), surf to this web address:

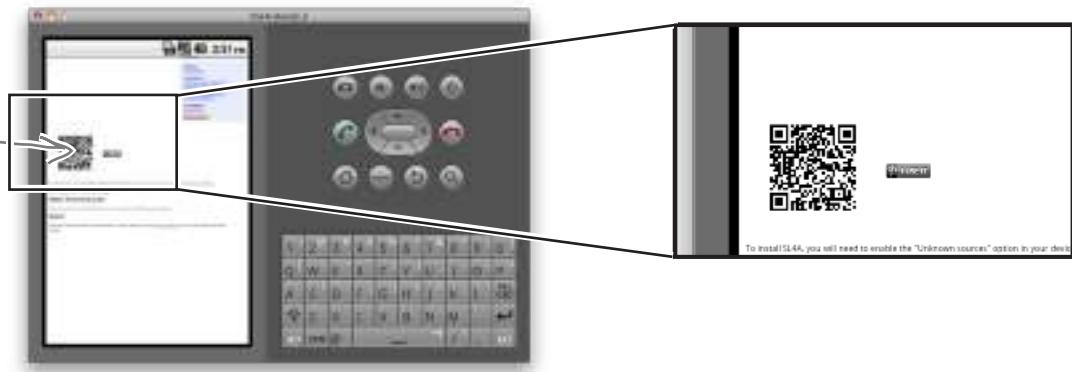
<http://code.google.com/p/android-scripting>

and tap on the “boxed” bar code near the bottom of the page:

These instructions work on a “real” phone, too. Just be sure to enable “Unknown sources” to allow for non-Market application downloads.

Don't worry if it takes your emulator a minute or two to start. The emulator is slower than the actual phone...

On the emulator, tap on the “boxed” bar code to start the SL4A download.



When the download completes, select the emulator's Menu button → More → Downloads, and then tap on the sl4a_r2.apk file to install the SL4A package on the emulator. When the install completes, tap Done.

The version available to you might be different, but don't worry: download the latest release.

The version you see might be different than this. Don't worry; yours is the most recent.



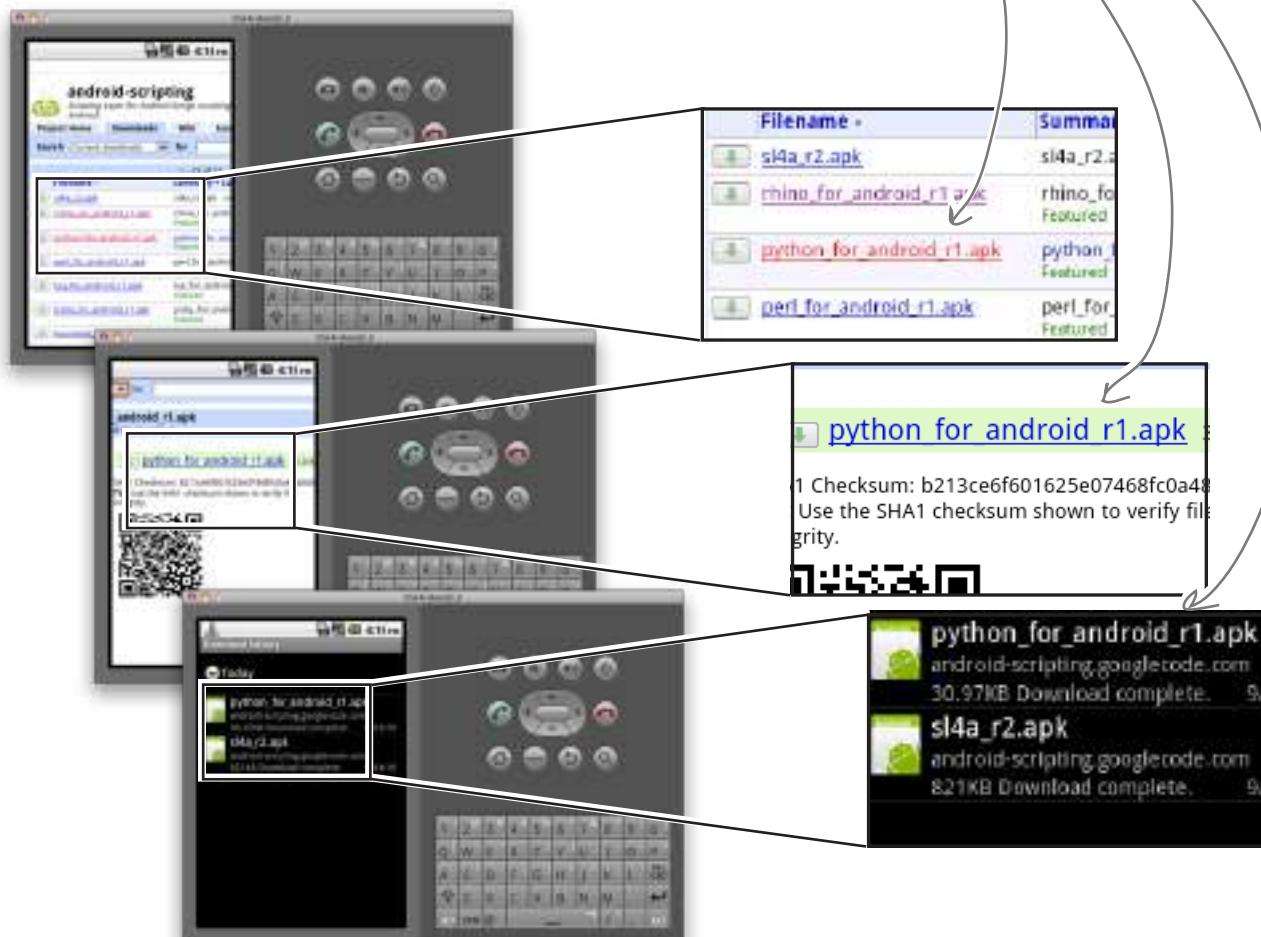
Add Python to your SL4A installation

Return to the emulator's web browser, double-tap on the screen to zoom in, and select the Downloads tab. Double-tap again and tap the following link:

[python_for_android_r1.apk](#)

Again, the version you see might be different than this. Select the most recent file.

Tap the download link, and tap on the package name to download it. Select Menu → More → Downloads, and tap on the newly downloaded package.



The Python for Android app runs. When you are ready, tap Open -> Install to complete the installation. This downloads, extracts, and installs the Python support files for Android, which can take a few minutes to complete. When it does, Python 2.6.2 and Python for Android are installed on your emulator and ready for action.

Let's confirm everything is working with a quick test.

Test Python on Android

Return to your emulator's main screen and find an app called *SL4A* added to your list of app icons. Tap this app to display the list of Python scripts preinstalled with Python for Android. Simply tap on any script name to execute it:



The “menu” button.



Watch it!

Be sure to set the SL4A rotation mode to automatic.

Your screen might switch to landscape by default the first time you run a script. To fix this, choose *Menu* → *Preferences*, scroll down to **Rotation mode**, and set its value to **Automatic**.

Take your Android emulator for a spin

Here's a four-line Python script that you can create to test your installation. Let's call this script *mydroidtest.py*:

```
Import the "android" library and create a new app object instance. →
import android
app = android.Android()

Create an appropriate message and display it on screen. →
msg = "Hello from Head First Python on Android"
app.makeTextToast(msg)
```

To transfer your script to the emulator, you need to copy it to the emulator's virtual SD card. Another program within the *tools* folder called *adb* helps with this:

Issue this command at your terminal window to transfer your script to the emulator.

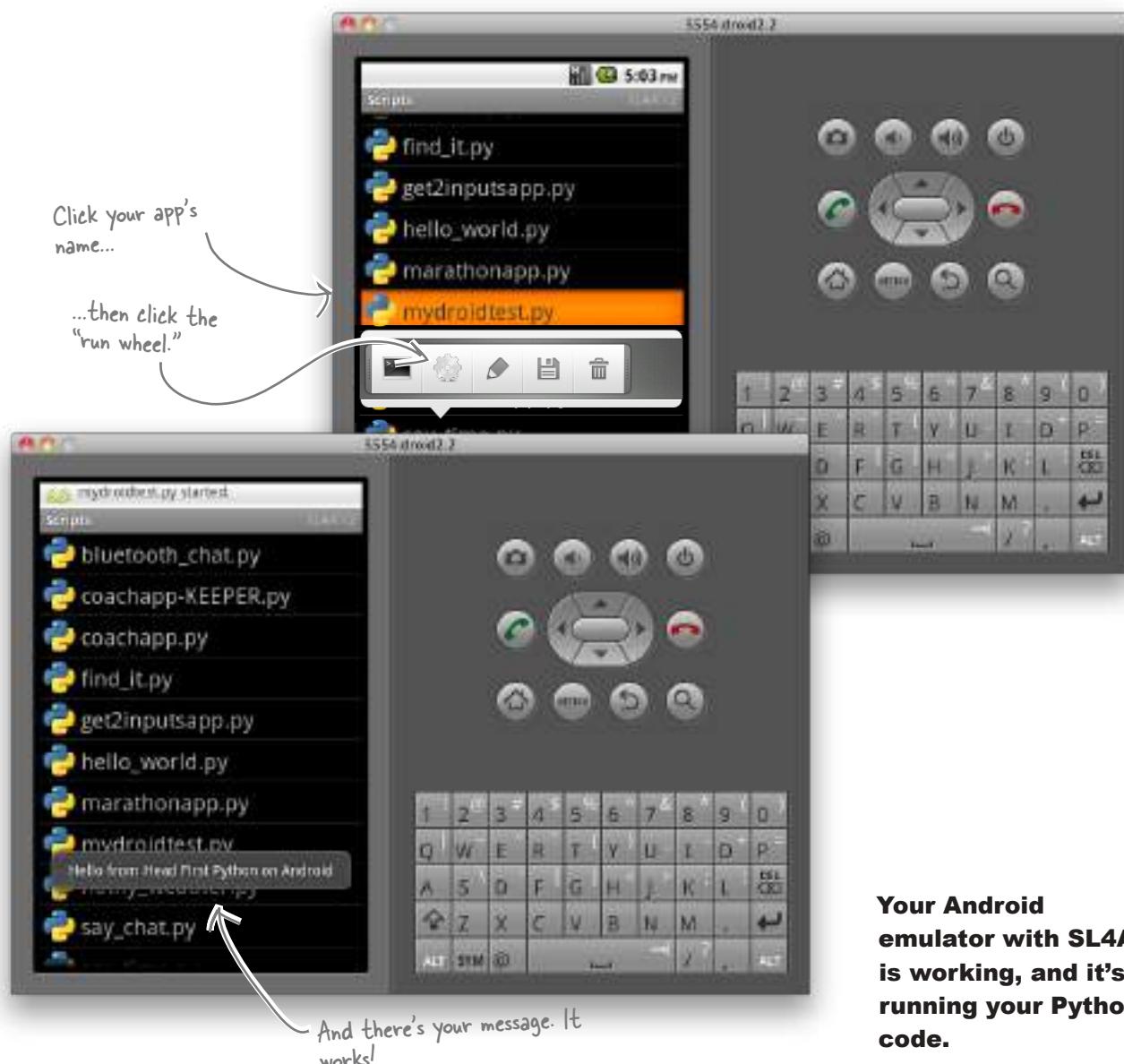
```
tools/adb push mydroidtest.py /sdcard/sl4a/scripts
```

Your script should now appear on the list of scripts available to SL4A.



Test Drive

Let's confirm that your Android setup is working. With the SL4A app open, simply tap on your script's name to run it, and then click the *run wheel* from the menu.



Your Android emulator with SL4A is working, and it's running your Python code.

Define your app's requirements

Let's think a little bit about what your Android app needs to do.



Frank: Well...first off, the view code no longer has to generate HTML, so that makes things interesting.

Jill: In fact, you need the web server only to supply your data on request, not all that generated HTML.

Joe: Ah ha! I've solved it. Just send the pickle with all the data from the server to the Android phone. It can't be all that hard, can it?

Jill: Sorry, guys, that'll cause problems. The pickle format used by Python 3 is *incompatible* with Python 2. You'll certainly be able to send the pickle to the phone, but the phone's Python won't be able to work with the data in the pickle.

Frank: Darn...what are our options, then? Plain data?

Joe: Hey, good idea: just send the data as one big string and parse it on the phone. Sounds like a workable solution, right?

Jill: No, that's a potential disaster, because you never know in what format that stringed data will arrive. You need an *data interchange format*, something like XML or JSON.

Frank: Hmm...I've heard XML is a hound to work with...and it's probably overkill for this simple app. What's the deal with JSON?

Joe: Yes, of course, I keep hearing about JSON. I think they use it in lots of different places on the Web, especially with AJAX.

Frank: Oh, dear...pickle, XML, JSON, and now AJAX...I think my brain might just explode here.

Jill: Never worry, you only need to know JSON. In fact, you don't even need to worry about understanding JSON at all; you just need to know how to use it. And, guess what? JSON comes standard with Python 2 and with Python 3...and the format is compatible. So, we can use JSON on the web server and on the phone.

Frank & Joe: Bonus! That's the type of technology we like!



JSON Exposed

This week's interview:
The Data Interchange Lowdown

Head First: Hello, JSON. Thanks for agreeing to talk to us today.

JSON: No problem. Always willing to play my part in whatever way I can.

Head First: And what *is* that, exactly?

JSON: Oh, I'm just one of the most widely used data interchange formats on the Web. When you need to transfer data over the Internet, you can rely on me. And, of course, you'll find me *everywhere*.

Head First: Why's that?

JSON: Well...it's really to do with my name. The "JS" in JSON stands for "JavaScript" and the "ON" stands for "Object Notation." See?

Head First: Uh...I'm not quite with you.

JSON: I'm JavaScript's object notation, which means I'm *everywhere*.

Head First: Sorry, but you've completely lost me.

JSON: The first two letters are the key ones: I'm a JavaScript standard, which means you'll find me everywhere JavaScript is...which means I'm in every major web browser on the planet.

Head First: What's that got to do with Python?

JSON: That's where the other two letters come into play. Because I was initially designed to allow JavaScript data objects to be transferred from one JavaScript program to another, I've been extended to allow objects to be transferred regardless of what programming language is used to create the data. By using the JSON library provided by your favorite programming language, you can create data that is interchangeable. If you can read a JSON data stream, you can recreate data as you see fit.

Head First: So I could take an object in, say,

Python, use JSON to convert it to JSON's object notation, and then send the converted data to another computer running a program written in C#?

JSON: And as long as C# has a JSON library, you can recreate the Python data as C# data. Neat, eh?

Head First: Yes, that sounds interesting...only [winks] why would anyone in their right mind want to program in C#?

JSON: [laughs] Oh, come on now: *be nice*. There's plenty of reasons to use different programming languages for different reasons.

Head First: Which goes some of the way to explain why we have so many great programming titles, like *Head First C#*, *Head First Java*, *Head First PHP and MySQL*, *Head First Rails*, and *Head First JavaScript*.

JSON: Was that a shameless, self-serving plug?

Head First: You know something...I think it might well have been! [laughs].

JSON: [laughs] Yes, it pays to advertise.

Head First: And to share data, right?

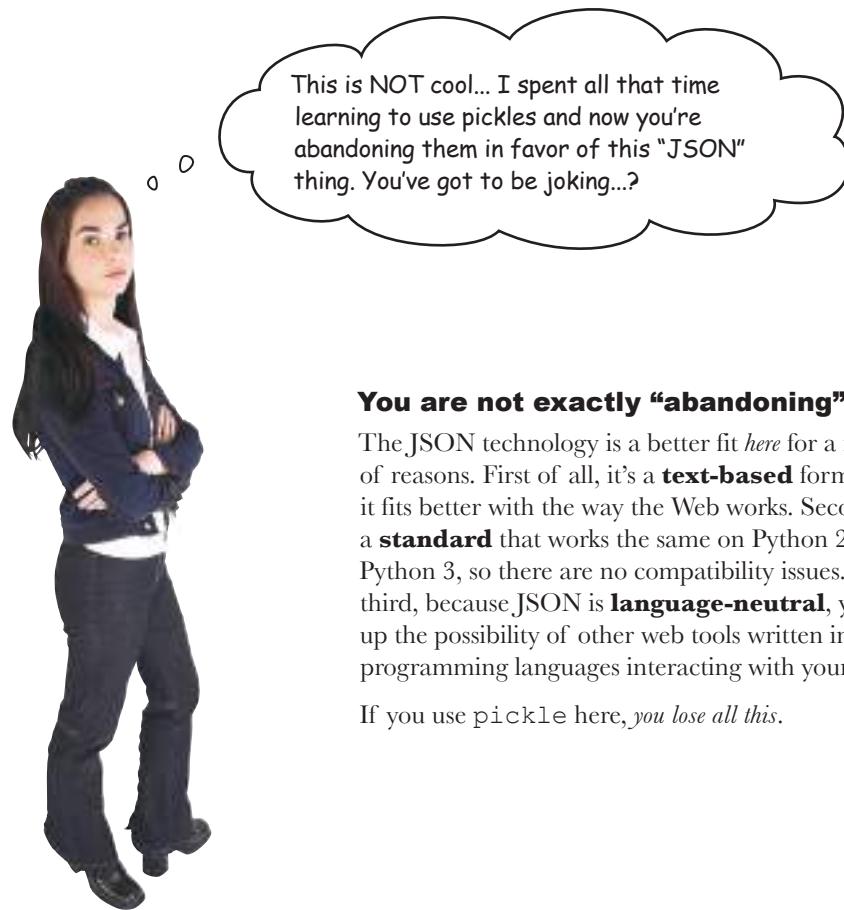
JSON: Yes! And that's exactly my point: when you need a *language-neutral data interchange format* that is easy to work with, it's hard to pass me by.

Head First: But how can you be "language neutral" when you have *JavaScript* in your name?

JSON: Oh, that's just my name. It's what they called me when the only language I supported was JavaScript, and it kinda stuck.

Head First: So they should really call you something else, then?

JSON: Yes, but "WorksWithEveryProgramming LanguageUnderTheSunIncludingPythonObject Notation" doesn't have quite the same ring to it!



This is NOT cool... I spent all that time learning to use pickles and now you're abandoning them in favor of this "JSON" thing. You've got to be joking...?

You are not exactly “abandoning” pickle.

The JSON technology is a better fit *here* for a number of reasons. First of all, it's a **text-based** format, so it fits better with the way the Web works. Second, it's a **standard** that works the same on Python 2 and Python 3, so there are no compatibility issues. And third, because JSON is **language-neutral**, you open up the possibility of other web tools written in other programming languages interacting with your server.

If you use `pickle` here, *you lose all this*.



An IDLE Session

JSON is an established web standard that comes preinstalled with Python 2 and Python 3. The JSON API is not that much different to the one used by pickle:

```

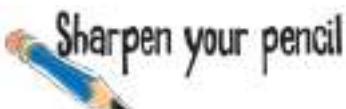
Import the JSON library.
>>> import json

Create a list of lists.
>>> names = ['John', ['Johnny', 'Jack'], 'Michael', ['Mike', 'Mikey', 'Mick']]
>>> names
['John', ['Johnny', 'Jack'], 'Michael', ['Mike', 'Mikey', 'Mick']]

Transform the Python list-of-lists into a JSON list of lists.
>>> to_transfer = json.dumps(names)
The format is similar,
>>> to_transfer
                    but different.
'["John", ["Johnny", "Jack"], "Michael", ["Mike", "Mikey", "Mick"]]

Transform the JSON list of lists back
into one that Python understands.
>>> from_transfer = json.loads(to_transfer)
The new data is exactly the same
>>> from_transfer
                    as the original list of lists.
['John', ['Johnny', 'Jack'], 'Michael', ['Mike', 'Mikey', 'Mick']]

>>> names
['John', ['Johnny', 'Jack'], 'Michael', ['Mike', 'Mikey', 'Mick']]
```



Add a new function to the `athletemodel` module that, when called, returns the list of athlete names as a string.

Call the new function `get_names_from_store()`.



You were to add a new function to the *athletemodel* module that, when called, returns the list of athlete names as a string.

You were to add all the new function `get_names_from_store()`.

```
def get_names_from_store():
    athletes = get_from_store() ← Get all the data from the pickle.
    response = [athletes[each_ath].name for each_ath in athletes]
    return(response) ← Return the list to the caller.
```

Extract a list
of athlete names →
from the data.

So...rather than running
a CGI script to create a HTML
web page, you want me to deliver
just the data, right? That's OK. Not
a problem—just be sure to tell me
which script to run...

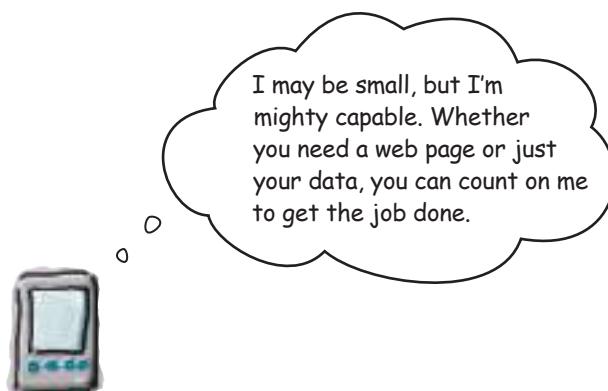




With your new function written and added to the `athletemodel` module, create a new CGI script that, when called, returns the data from the `get_names_from_store()` function to the web requester as a JSON data stream.

Call your new script cgi-bin/generate_names.py.

Hint: Use application/json as your Content-type.





Exercise Solution

Don't forget this
"magic" first line
if you're running
on Linux or Mac
OS X.

With your new function written and added to the `athletemodel` module, you were to create a new CGI script that, when called, returns the data from the `get_names_from_store()` function to the web requester as a JSON data stream.

You were to call your new script `cgi-bin/generate_names.py`.

→ `#!/usr/local/bin/python3`

`import json`

`import athletemodel`

`import yate`

Do your imports.

`names = athletemodel.get_names_from_store()`

Get the data from
the model.

`print(yate.start_response('application/json'))`

Sort "names", then convert
to JSON and send to
STDOUT.

`print(json.dumps(sorted(names)))`

Start with the
appropriate
"Content-type":
line.



Watch it!

**Take care testing
your JSON-generating
CGI code.**

The behavior you see
when testing your JSON-
generating CGI script will differ
depending on the web browser you
are using. For instance, Firefox might
attempt to download the generated
data as opposed to display it on screen.



Test DRIVE

If it is not already running, start your web server and be sure to set the executable bit with the `chmod +x cgi-bin/generate_names.py` command (if on Linux or Mac OS X). When you're ready, grab your favorite web browser and take your new CGI for a spin.

Enter the web address of the CGI in your browser's location bar.



The web server's logging information confirms that the CGI executed.



```
File Edit Window Help GeneratingJSON
$ python3 simple_httpd.py
Starting simple_httpd on port: 8080
localhost - - [18/Sep/2010 06:31:29] "GET /cgi-bin/generate_names.py HTTP/1.1" 200 -
localhost - - [18/Sep/2010 06:35:29] "GET /cgi-bin/generate_list.py HTTP/1.1" 200 -
localhost - - [18/Sep/2010 06:35:35] "POST /cgi-bin/generate_timing_data.py HTTP/1.1" 200 -
localhost - - [18/Sep/2010 06:35:38] "GET /cgi-bin/generate_list.py HTTP/1.1" 200 -
localhost - - [18/Sep/2010 06:35:40] "GET /index.html HTTP/1.1" 200 -
localhost - - [18/Sep/2010 06:35:49] "GET /cgi-bin/generate_names.py HTTP/1.1" 200 -
```

That worked!

Now all you have to do is arrange for the Android emulator to request the data within a Python script and display the list of names on the smartphone's screen. *How hard can that be?*

The SL4A Android API

The SL4A technology provides a high-level API to the low-level Android API, and SL4A's API is documented in the online API reference:

<http://code.google.com/p/android-scripting/wiki/ApiReference>

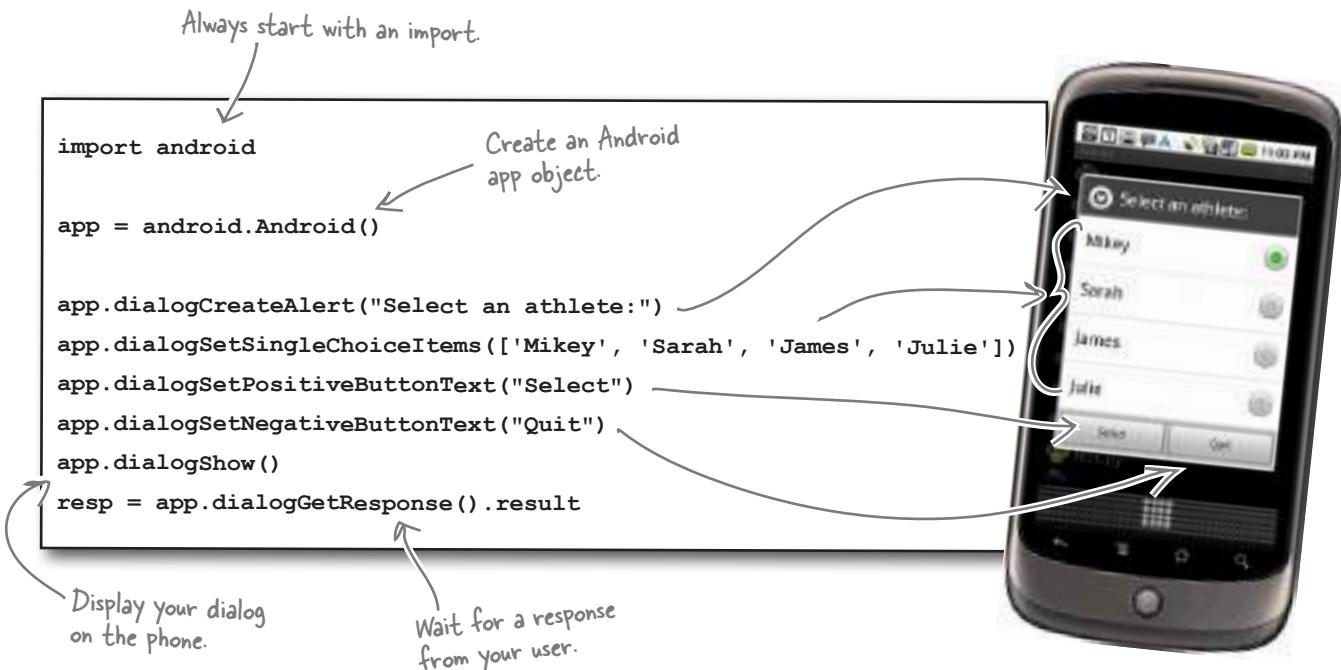
Recall the code from earlier, which demonstrated a minimal Android SL4A app:

Import the "android" library and create a new app object instance.

Create an appropriate message and display it on screen.

```
import android
app = android.Android()
msg = "Hello from Head First Python on Android"
app.makeText(msg)
```

Six calls to the Android API let you create a list of selectable items in a dialog, together with *positive* and *negative* buttons, which are used to indicate the selection your user made. Note how each of the calls to the Android "dialog" API results in something appearing on screen.





Android Code Magnets

Here is the code to a program that queries your web server for the list of names as a JSON array and then displays the list on the smartphone. The only trouble is, the second half of the program is a bunch of mixed-up code magnets at the bottom of the screen. Your job is to rearrange the magnets to complete the program.

```

import android
import json
import time
from urllib import urlencode
from urllib2 import urlopen
Do the usual imports...these
ones pull in web client
functionality.

All of this
program's
messages are in
one place.
The name
of the CGI
script to run
on the web
server
This code's a
mess...can you
fix it?
    } hello_msg      = "Welcome to Coach Kelly's Timing App"
    list_title     = 'Here is your list of athletes:'
    quit_msg       = "Quitting Coach Kelly's App."
    web_server     = 'http://192.168.1.33:8080' ←
    get_names_cgi  = '/cgi-bin/generate_names.py'

def send_to_server(url, post_data=None):
    if post_data:
        page = urlopen(url, urlencode(post_data))
    else:
        page = urlopen(url)
    return(page.read().decode("utf8"))

Change this to the
web address that's
running your web
server.

This function takes both a
web address (url) and some
optional data (post_data)
and sends a web request to
your web server. The web
response is returned to the
caller.

athlete_names = sorted(json.loads(send_to_server(web_server + get_names_cgi)))
status_update(quit_msg)
app.dialogShow()
app.dialogCreateAlert(list_title)
def status_update(msg, how_long=2):
    app.makeText(msg)
    time.sleep(how_long)
app.dialogSetNegativeButton('Quit')
resp = app.dialogGetResponse().result
status_update(hello_msg)
app.dialogSetPositiveButton('Select')
app = android.Android()
app.dialogSetSingleChoiceItems(athlete_names)
app.dialogSetSingleChoiceItems(athlete_names)

```



Android Code Magnets Solution

Here is the code to a program that queries your web server for the list of names as a JSON array and then displays the list on the smartphone. The only trouble is, the second half of the program is a bunch of mixed-up code magnets at the bottom of the screen. Your job was to rearrange the magnets to complete the program.

```
import android
import json
import time
from urllib import urlencode
from urllib2 import urlopen

hello_msg      = "Welcome to Coach Kelly's Timing App"
list_title     = 'Here is your list of athletes:'
quit_msg       = "Quitting Coach Kelly's App."
web_server     = 'http://192.168.1.33:8080'
get_names_cgi  = '/cgi-bin/generate_names.py'
```

```
def send_to_server(url, post_data=None):
    if post_data:
        page = urlopen(url, urlencode(post_data))
    else:
        page = urlopen(url)
    return(page.read().decode("utf8"))
```

Create an Android app object.

```
app = android.Android()
```

```
def status_update(msg, how_long=2):
    app.makeText(msg)
    time.sleep(how_long)
```

Say "hello".

```
status_update(hello_msg)
```

```
athlete_names = sorted(json.loads(send_to_server(web_server + get_names_cgi)))
```

```
app.dialogCreateAlert(list_title)
```

```
app.dialogSetSingleChoiceItems(athlete_names)
```

```
app.dialogSetPositiveButtonText('Select')
```

```
app.dialogSetNegativeButton('Quit')
```

```
app.dialogShow()
```

```
resp = app.dialogGetResponse().result
```

```
status_update(quit_msg)
```

This is a little function for displaying short messages on the phone.

Send the web request to your server, then turn the JSON response into a sorted list.

Create a two-buttoned dialog from the list of athlete names.

Wait for the user to tap a button, then assign the result to "resp".

Say "bye bye."



Test DRIVE

Recall that (for now) your Android Python scripts run within the emulator, not within IDLE. So use the tools/adb program to copy your program to the emulator. Call your program `coachapp.py`. When the code is copied over, start SL4A on your emulator, and then tap your script's name.



This is looking really good! Your app has communicated with your web server, requested and received the list of athlete names, and displayed the list on your emulator.



If your app doesn't run, don't panic. Check your code for typos.

Run your app again in the Python terminal by tapping on the little terminal icon to the left of the “run wheel” within SL4A. If your code raises an error, you’ll see any messages on the emulator’s screen, which should give you a good idea of what went wrong.

positive or negative

Select from a list on Android

When your user taps on a button, the “result” of the call to `dialogGetResponse()` is set to `positive` if the first button is tapped or `negative` if the second button is tapped. In your code, you can check the value of `resp`, which is a dictionary, and the `which` key is set to either `positive` or `negative`.

A subsequent call to `dialogGetSelectedItems()` returns the index value of the selected list item.



So...if the positive button is tapped, you can index into the list of athlete names to see which athlete was selected from the displayed list. The selected name can then be sent to the web server to request the rest of the athlete's data using the `send_to_server()` function.

You can use this behavior in the next version of your code.



- 1** Assume that you have a CGI script called `cgi-bin/generate_data.py`, which, when called, requests the data for a named athlete from the server.

Provide the code (which includes a call to `thensend_to_server()` function) to implement this functionality:

- 2** Additionally, write the code required to display the list of times returned from the server within an Android dialog.

Hints: Use the `dialogSetItems()` method from the Android API to add a list of items to a dialog. Also, remember that the data arriving over the Internet will be formatted using JSON.



Sharpen your pencil Solution

- 1 You were to assume that you have a CGI script called cgi-bin/generate_data.py, which, when called requests the data for a named athlete from the server.

Provide the name of the CGI to run.

You were to provide the code (which includes a call to the send_to_server() function) to implement this functionality:

```
get_data_cgi = '/cgi-bin/generate_data.py'
```

Send the request to the web server, together with the athlete name.

```
send_to_server(web_server + get_data_cgi, {'which_athlete': which_athlete})
```

Include the data.

- 2 Additionally, you were to write the code required to display the list of times returned from the server within an Android dialog:

When your user taps the "positive" button...work out the index value chosen.

```
if resp['which'] in ('positive'):
```

```
    selected_athlete = app.dialogGetSelectedItems().result[0]
```

The index value is in the first element of the list of results returned from the dialog.

Look up the athlete's name using the index value.

```
which_athlete = athlete_names[selected_athlete]
```

Dynamically create the dialog's title.

```
athlete_title = which_athlete + ' top 3 times:'
```

Send a new web request to the server to fetch the athlete's data.

The user needs to see only the data this time, so you need to use "dialogSetItems()".

```
app.dialogCreateAlert(athlete_title)
```

```
app.dialogSetItems(athlete['Top3'])
```

```
app.dialogSetPositiveButtonText('OK')
```

```
app.dialogShow()
```

Set the single button's text.

Wait for a tap from the user.

```
resp = app.dialogGetResponse().result
```

The athlete's data CGI script

Here's the code for the `cgi-bin/generate_data.py` CGI script, which takes a web request and returns the indicated athlete's data from the model:

```
#!/usr/local/bin/python3

import cgi
import json
import athletemodel
import yate

athletes = athletemodel.get_from_store()
form_data = cgi.FieldStorage()
athlete_name = form_data['which_athlete'].value
print(yate.start_response('application/json'))
print(json.dumps(athletes[athlete_name]))
```

Annotations on the code listing:

- A callout points to the line `athletes = athletemodel.get_from_store()` with the text "Get all the data from the model."
- A callout points to the line `form_data = cgi.FieldStorage()` with the text "Process the data sent with the request and extract the athlete's name."
- A callout points to the line `print(json.dumps(athletes[athlete_name]))` with the text "Start a web response, with JSON as the data type."
- A callout points to the line `print(json.dumps(athletes[athlete_name]))` with the text "Include the indicated athlete's data in the web response, formatted by JSON."

The complete Android app, so far

You've made quite a few changes to your program at this stage. Before you test it on the Android emulator, take a moment to look at your code *in its entirety*:

```
import android
import json
import time

from urllib import urlencode
from urllib2 import urlopen

hello_msg      = "Welcome to Coach Kelly's Timing App"
list_title     = 'Here is your list of athletes:'
quit_msg       = "Quitting Coach Kelly's App."

web_server     = 'http://192.168.1.34:8080'

get_names_cgi = '/cgi-bin/generate_names.py'
get_data_cgi  = '/cgi-bin/generate_data.py'
```

The rest of your code is on the following page.

```
def send_to_server(url, post_data=None):
    if post_data:
        page = urlopen(url, urlencode(post_data))
    else:
        page = urlopen(url)
    return(page.read().decode("utf8"))

app = android.Android()

def status_update(msg, how_long=2):
    app.makeText(msg)
    time.sleep(how_long)

status_update(hello_msg)

athlete_names = sorted(json.loads(send_to_server(web_server + get_names_cgi)))

app.dialogCreateAlert(list_title)
app.dialogSetSingleChoiceItems(athlete_names)
app.dialogSetPositiveButtonText('Select')
app.dialogSetNegativeButtonText('Quit')
app.dialogShow()
resp = app.dialogGetResponse().result

if resp['which'] in ('positive'):
    selected_athlete = app.dialogGetSelectedItems().result[0]
    which_athlete = athlete_names[selected_athlete]
    athlete = json.loads(send_to_server(web_server + get_data_cgi,
                                         {'which_athlete': which_athlete}))

    athlete_title = athlete['Name'] + ' (' + athlete['DOB'] + ')', top 3 times:'
    app.dialogCreateAlert(athlete_title)
    app.dialogSetItems(athlete['Top3'])
    app.dialogSetPositiveButtonText('OK')
    app.dialogShow()
    resp = app.dialogGetResponse().result

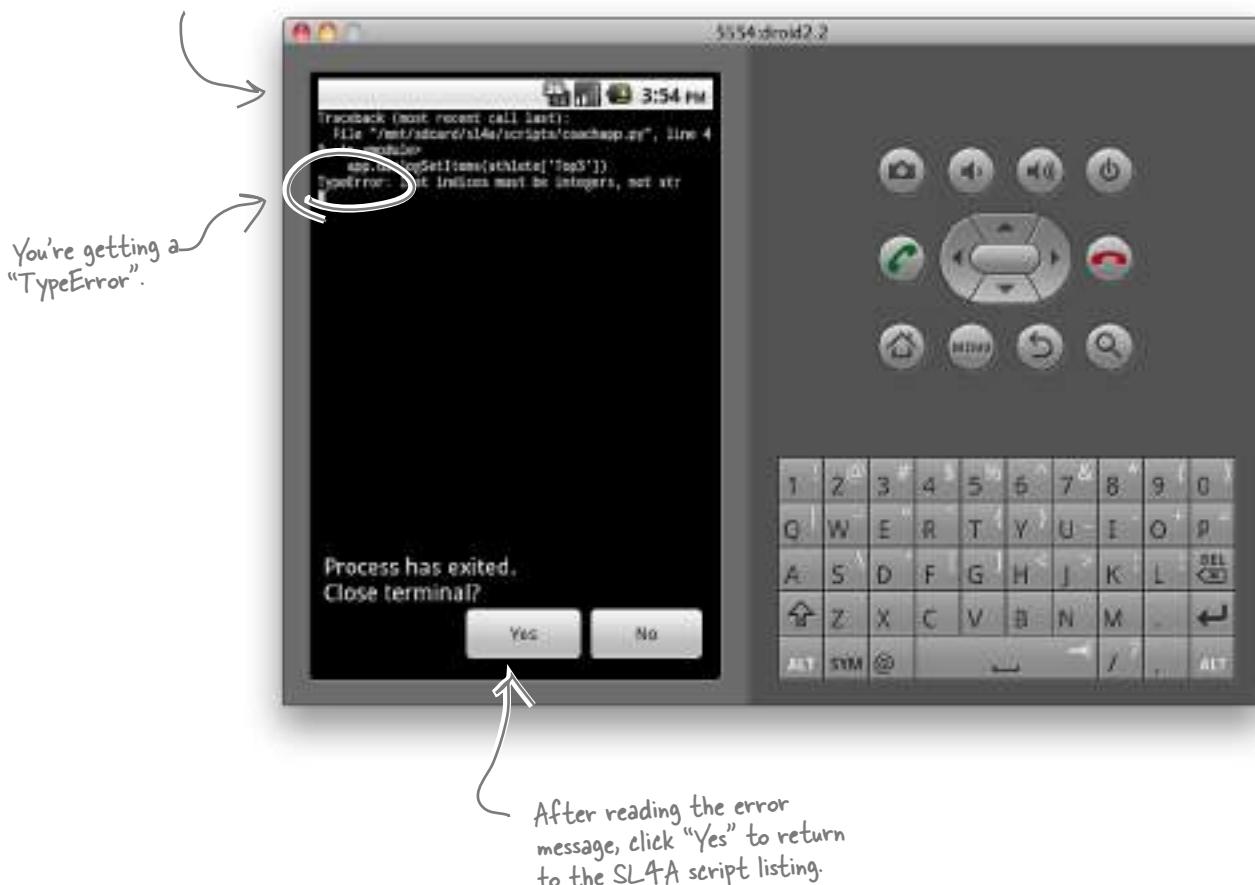
status_update(quit_msg)
```



Test DRIVE

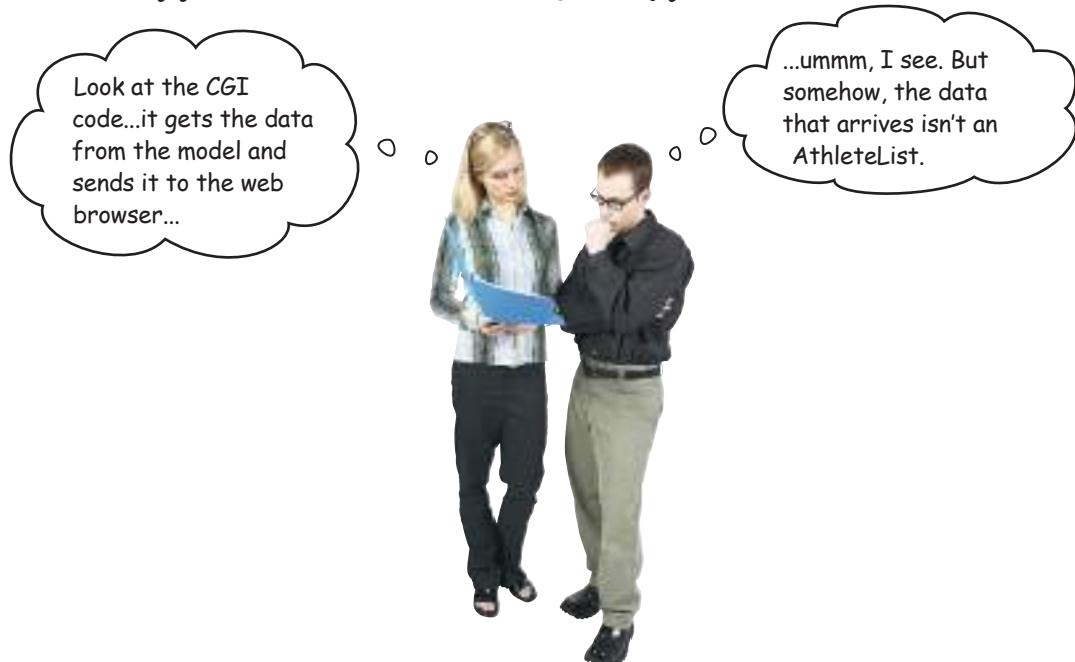
Let's give the latest version of your app a go. Copy the app to your emulator, and put the new CGI script in your `cgi-bin` folder on your web server (remember to set the executable bit, if needed). What happens when you run your latest app using the emulator's Python shell as opposed to the "run wheel"?

You are dumped into the Python shell with a rather nasty error message.



Yikes! Your code has a `TypeError`, which is crashing your app when you try to display the selected athlete's timing data. *Why do you think this is happening?*

The data appears to have changed type



Let's add a debugging line of code to your CGI script to try and determine what's going on. Recall that the CGI mechanism captures any output your script sends to *standard output* by default, so let's use code like this to send your debugging message to the web server's console, which is displaying on *standard error*:

Import "sys"
from the
standard library.

```
import sys

print(json.dumps(athletes[athlete_name]), file=sys.stderr)
```

Redirect the output
from "print()" to
"stderr", rather
than the default,
which is "stdout".

Run your app again and, of course, it's still crashes with a `TypeError`. However, if you check your web server's console screen, you'll see that the data being sent as the JSON web response is clearly visible. *Notice anything?*

This is a list of
athlete timing
values...but where's
the name and
DOB values?

```
File Edit Window Help JustWhatsInTheData
$ python3 simple_httppd.py
Starting simple_httppd on port: 8080
192.168.1.33 - - [18/Sep/2010 17:40:04] "GET /cgi-bin/generate_names.py HTTP/1.1" 200 -
192.168.1.33 - - [18/Sep/2010 17:40:08] "POST /cgi-bin/generate_data.py HTTP/1.1" 200 -
["2:44", "3:01", "2.44", "2.55", "2.51", "2:41", "2:41", "3:00", "2:32", "2.11", "2:26"]
```

JSON can't handle your custom datatypes

Unlike `pickle`, which is smart enough to pickle your custom classes, the JSON library that comes with Python isn't. This means that the standard library's JSON library can work with Python's built-in types, but not with your `AthleteList` objects.

The solution to this problem is straightforward: add a method to your `AthleteList` class to convert your data into a dictionary, and send *that* back to the app. Because JSON supports Python's dictionary, this should work.



Let's create a new method in your `AthleteList` class. Called `to_dict()`, your new method needs to convert the class's attribute data (`name`, `DOB`, and `top3`) into a dictionary. Be sure to decorate your new method with `@property`, so that it appears to be a new attribute to users of your class.

.....
.....
.....
.....
.....
.....
.....
.....

there are no
Dumb Questions

Q: What's the purpose of this `@property` thing again?

A: The `@property` decorator lets you specify that a method is to be presented to users of your class as *if it were an attribute*. If you think about things, your `to_dict()` method doesn't change the state of your object's data in any way: it merely exists to return the object's attribute data as a dictionary. So, although `to_dict()` is a method, it behaves more like an attribute, and using the `@property` decorator lets you indicate this. Users of your class (that is, other programmers) don't need to know that when they access the `to_dict` attribute they are in fact running a method. All they see is a **unified interface**: attributes access your class's data, while methods manipulate it.



Exercise Solution

Let's create a new method in your `AthleteList` class. Called `to_dict()`, your new method needs to convert the class's attribute data (name, DOB, and `top3`) into a dictionary. Be sure to decorate your new method with `@property`, so that it appears to be a new attribute to users of your class.

Decorate your
new method with
“`@property`”.

`@property`

`def as_dict(self):` Create a new method.

`return({'Name': self.name,`

`'DOB': self.dob,`

`'Top3': self.top3})`

Return a dictionary of the object's
data attributes.

Did you remember to use “self”?

Do this!

As well as updating your `AthleteList` class code, be sure to change `cgi-bin/generate-data.py` to return a dictionary, rather than the object instance, when servicing its web request.

While you're making changes, adjust the `coachapp.py` app code to include the athlete's name and DOB values in the second dialog's title.



Test DRIVE

With your changes applied to AthleteList.py, cgi-bin/generate_data.py and coachapp.py, use the adb tool to copy the latest version of your app to the emulator. Let's see how things work now.

Here's the code
that your app uses →
in response to an
athlete selection.

```

coachapp.py - /Users/barryw/HeadFirstPython/chapter8/coachapp.py
if resp['which'] in ('positive'):
    selected_athlete = app.dialogGetSelectedItems().result[0]
    which_athlete = athlete_names[selected_athlete]
    athlete = json.loads(send_to_server(web_server + get_data.cgi,
                                         {'which_athlete': which_athlete}))

    athlete_title = athlete['name'] + " (" + athlete['DOB'] + ")", top 3 times"
    app.dialogCreateAlert(athlete_title)
    app.dialogSetItems(athlete['Top'])
    app.dialogSetPositiveButtonText('OK')
    app.dialogShow()
    resp = app.dialogGetResponse().result

status_update(quit_msg)

```

SSS4.droid2.2

SSS4.droid2.2

SSS4.droid2.2

1 2 3 4 5 6 7 8 9 0
Q W E R T Y U I O P
A S D F G H J K L ;
Z X C V B N M , .
ALT FN @

Success.

Your app displays the selected athlete's top three times on screen. *How cool is that?*

Run your app on a real phone

Now that your app is running successfully on your emulator, it's time to try it on a real phone. This is where things get *interesting*.

There are many options when it comes to copying your code to a real device:

- Use file transfer over Bluetooth.
- Use file transfer with a USB connection.
- Use the Android SDK's adb tool with USB.
- Use a file transfer tool over WiFi.

Unfortunately, which technique to use (and which work) depends very much on your phone.

At Head First Labs, we've had the greatest and most consistent success with the last option: *use a file transfer tool over WiFi*.



Watch it!

These instructions do not work on the emulator.

*The Android emulator does not currently support Google's **Android Market**, which you'll need access to use when following along with the instructions on these pages.*

Step 1: Prepare your computer

To transfer files securely between your Android phone and your computer, enable SSH file transfers by running an **SSH server** on your computer. How you do this depends on the operating system you are running:

- **Windows**: download one of the many free SSH servers.
- **Mac OS X**: enable remote logins.
- **Linux**: install and enable OpenSSH Server.

Step 2: Install AndFTP on your Android phone

Use the Android Market on your phone to find and install the **AndFTP** app. This excellent tool lets you transfer files to and from your Android phone over FTP, SFTP, and FTPS.

To use it with the SSH server running on your computer, you'll want to select **SFTP** as the file transfer protocol within the app, because **AndFTP** defaults to using the FTP protocol.

Let's take a look at what's involved.



The AndFTP app is one of our faves.

Configure AndFTP

With AndFTP running on your phone, configure it to connect to your computer (*Hostname*) using SFTP as the transfer protocol (*Type*). Leave the *Port*, *Username*, *Password*, and *Remote dir* entries as they are, but change the *Local dir* entry to `/sdcard/sl4a/scripts`.



With the connection set up, tap AndFTP's Connect button to establish a connection to your SSH server, entering your Username and Password when prompted.

With the connection to the server established, navigate to the server folder containing the file(s) you want to transfer to the phone, mark the files for download, and tap the Download button.

When the download completes, click Disconnect to terminate the connection between the phone and your computer. If you transferred a Python program, it should now be added to the list of scripts within SL4A.

It's time to let Coach Kelly take a look.



The coach is thrilled with his app



Welcome to the future!

You've delivered a solution that automates interaction with your website while providing a modern interface on an Android phone. Your app allows your users to access web data *directly* on their mobile device.

The fact that your server code runs on Python 3 and your Android client code runs on Python 2 makes very little difference: *it's all just Python code, after all.*

All that's left to do is write some code to satisfy Coach Kelly's latest request, and you'll get to that in the next chapter.

This is great work.



Your Python Toolbox

You've got Chapter 8 under your belt and you've added some key Python techniques to your toolbox.

Python Lingo

- "Python 2" – the previous release of Python, which has compatibility "issues" with Python 3 (and are not worth getting worked up over).

Android Lingo

- "SL4A" – the Scripting Layer for Android lets you run Python on your Android device.
- "AVD" – an Android Virtual Device which lets you emulate your Android device on your computer.



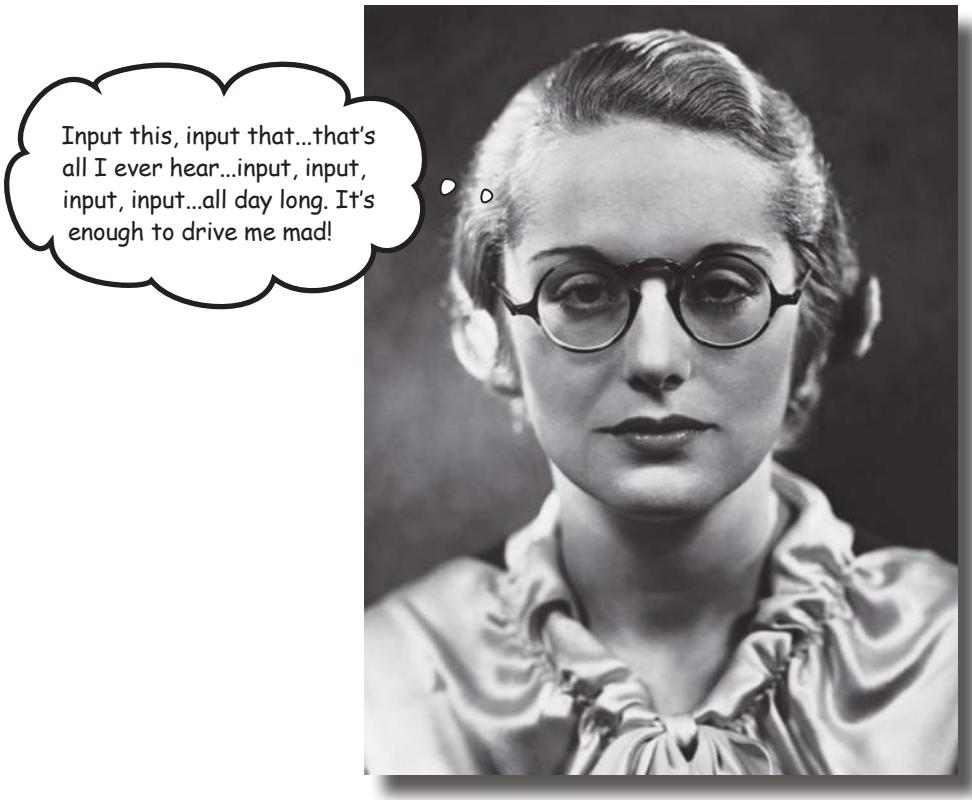
BULLET POINTS

- The `json` library module lets you convert Python's built-in types to the text-based JSON data interchange format.
- Use `json.dumps()` to create a stringed version of a Python type.
- Use `json.loads()` to create a Python type from a JSON string.
- Data sent using JSON needs to have its `Content-Type`: set to `application/json`.
- The `urllib` and `urllib2` library modules (both available in Python 2) can be used to send encoded data from a program to a web server (using the `urlencode()` and `urlopen()` functions).
- The `sys` module provides the `sys.stdin`, `sys.stdout` and `sys.stderr` input streams.

9 manage your data



Handling input



The Web and your phone are not just great ways to display data.

They are also great tools to for accepting input from your users. Of course, once your webapp accepts data, it needs to put it somewhere, and the choices you make when deciding what and where this “somewhere” is are often the difference between a webapp that’s easy to grow and extend and one that isn’t. In this chapter, you’ll extend your webapp to accept data from the Web (via a browser or from an Android phone), as well as look at and enhance your back-end data-management services.

Your athlete times app has gone national



The *National Underage Athletics Committee (NUAC)* took one look at your Android app and realized it's just what they need...almost.

There are many ways to improve your webapp, but for now, let's concentrate on the committee's most pressing need: *adding a new time value to an existing athlete's data set*.

Adding new data to text files isn't going to work: there are just too many coaches around the country adding data. The committee wants something that's user friendly from any web browser or Android phone.

Can you help?

Use a form or dialog to accept input



On the Web, your user interacts with your web form and enters data. When she presses the **submit** button, the web browser gathers up all of the form's data and sends it to the web server *as part of the web request*.

On your Android phone, you can use the `dialogGetInput()` method to get input from the user, then mimic the behavior of the web form's **submit** button in code.

In fact, you've done this already: check out this line of code from your `coachapp.py` app, which sends the selected athlete name to your web server:

```
athlete = json.loads(send_to_server(web_server + get_data_cgi, {'which_athlete': which_athlete}))
```

Here's where the
data is included
with the web
request.

Create an HTML form template

Let's extend `yate.py` to support the creation of a HTML form. Take a look at this simple form, together with the HTML markup used to produce it.

The screenshot shows a web browser window titled "form.html" with the URL "http://localhost:8080/form.html". The page contains a form with a text input field labeled "Enter a timing value:" and a submit button labeled "Send". A callout bubble points to the "Send" button with the text: "Click the "Send" button to submit the form's data to your web server." Another callout bubble points to the "action" attribute in the source code with the text: "The name of the CGI script to send the form's data to."

```
<form action="cgi-bin/process-time.py" method="POST">
  Enter a timing value:
  <input type="Text" name="TimeValue" size=40>
  <br />
  <input type="Submit" value="Send">
</form>
```

When your user clicks on the **Send** button, any data in the input area is sent to the web server as part of the web request.

On your web server, you can access the CGI data using the facilities provided by the standard library's `cgi` module:

A callout bubble points to the first line of the script with the text: "Get the data sent from the form as part of the web request." Another callout bubble points to the line `timing_value = form["TimeValue"].value` with the text: "Access the value associated with the "TimeValue" key from the form's data."

```
import cgi
form = cgi.FieldStorage()
timing_value = form["TimeValue"].value
```

The `cgi` module converts the data associated with the web request into a dictionary-like object that you can then query to extract what you need.



Let's turn the HTML form from the previous page into a template within the `yate.py` module.

- 1 Start by creating a new template called `templates/form.html` that allows you to parameterize the form's CGI script name, method, input tags, and submit button text:

With the template ready, write the code for two functions you intend to add to `yate.py`.

- 2 The first, called `create_inputs()`, takes a list of one or more strings and creates HTML `<INPUT>` tags for each string, similar to the one that accepts `TimeValue` on the previous page.

The second, called `do_form()`, uses the template from Part 1 of this exercise together with the `create_inputs()` function to generate a HTML form.

```
def create_inputs(inputs_list):
```

Given a list of `<INPUT>` tag names.

Return the generated tags to the caller.

```
return(html_inputs)
```

The name of the CGI script and a list of `<INPUT>` tag names are required arguments.

```
def do_form(name, the_inputs, method="POST", text="Submit"):
```

The HTTP method and text to the "Submit" button have sensible default values.

Substitute the arguments and generated `<INPUT>` tags into the template to create the form.

```
return(form.substitute(cgi_name=name, http_method=method,
list_of_inputs=inputs, submit_text=text))
```



Exercise Solution

You were to turn the HTML form into a template within the `yate.py` module.

- 1 You were to start by creating a new template called `templates/form.html` that allows you to parameterize the form's CGI script name, method, input tags, and submit button text.

The CGI script's name and associated HTTP method are parameterized.

```
<form action=${cgi_name} method=${http_method}>
  Enter a timing value: ${list_of_inputs}<br />
  <input type="Submit" value=${submit_text}></form>
```

The list of `<INPUT>` tags and the submit button's text is also parameterized.

- 2

With the template ready, you were to write the code for two functions to add to `yate.py`.

The first, called `create_inputs()`, takes a list of one or more strings and creates HTML `<INPUT>` tags for each string, similar to the one that accepts `TimeValue`.

The second, called `do_form()`, uses the template from Part 1 of this exercise together with the `create_inputs()` function to generate a HTML form:

```
def create_inputs(inputs_list):
    html_inputs = ""
    for each_input in inputs_list:
        html_inputs = html_inputs + '<input type="Text" name=' + each_input + '" size=40>'
```

This "continuation" character lets you split a long line of code over multiple lines.

Take each name and create an `<INPUT>` tag.

```
return(html_inputs)
```

```
def do_form(name, the_inputs, method="POST", text="Submit"):
```

Grab the template from your disk.

```
with open('templates/form.html') as formf:
```

```
    form_text = formf.read()
```

```
    inputs = create_inputs(the_inputs)
```

Create a template form.

```
    form = Template(form_text)
```

Create the list of `<INPUT>` tags.

```
    return(form.substitute(cgi_name=name, http_method=method,
                           list_of_inputs=inputs, submit_text=text))
```



Test DRIVE

Here's the code to a CGI script called `cgi-bin/test-form.py`, which generates the HTML form from earlier. As you can see, there's nothing to it.

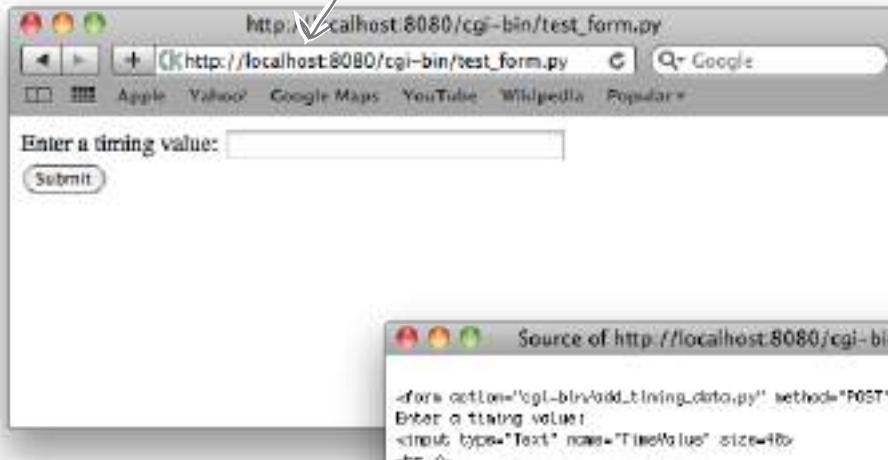
```
#!/usr/local/bin/python3
import yate
print(yate.start_response('text/html'))
print(yate.do_form('add_timing_data.py', ['TimeValue'], text='Send'))
```

Always start with a CGI response.

Dynamically create the form, supplying any arguments as required.

Set the executable bit (if required on your OS) using `chmod + x test_form.py`, and then use your browser to confirm that your HTML form-generating code is working.

Enter the URL for the CGI script into
your web browser's location bar.



The generated HTML form appears within the browser's window.

Use your browser's "View Source" menu option to confirm that the generated form is exactly what you need.



Great. You've extended `yate.py` to support the creation of a simple data entry form. Now all you need to do is to decide what happens once the data arrives on your server.

The data is delivered to your CGI script

In addition to running your webapp, the web server also arranges to deliver any submitted form data to your waiting CGI script. Python's `cgi` library converts the data into a dictionary and, as you already know, provides you with convenient access to the submitted data:

```
import cgi  
  
form = cgi.FieldStorage()
```

All of your form's data has been added to the "form" dictionary.

Additional information about the web request is also available to you via the web server's *environment*. Typically, you won't need to access or use this data directly. However, occasionally, it can be useful to report on some of it.

Here is some code that takes advantage of Python's built-in support for querying your CGI script's environment using the `os` library, assuming the environment values have been set by a friendly web server. Note that the data in the environment is available to your code as a dictionary.

```
import os ← Be sure to include the  
import time "os" library in your list of  
import sys imports.  
  
addr = os.environ['REMOTE_ADDR'] ← Query three environment variables  
host = os.environ['REMOTE_HOST'] and assign their values to variables.  
method = os.environ['REQUEST_METHOD']  
  
cur_time = time.asctime(time.localtime()) ← Get the current time.  
  
print(host + ", " + addr + ", " + cur_time + ": " + method, file=sys.stderr)
```

Display the queried data on standard error.

Let's exploit both code snippets on this page to **log** the data sent from a form to your web server's console. When you are convinced that the data is arriving at your web server *intact*, you can extend your code to store the received data in your model.

Let's write a CGI to display your form's data.



CGI Magnets

You need a new CGI script called `add_timing_data.py`, which processes the data from a form and displays the data on your web server's console screen. The CGI needs to query the environment, arranging to display the logged data on one line. The code exists, but most of it is all over the floor. Rearrange the magnets to produce a working program.

Don't forget
this line if you
are running on
Max OS X or
Linux.

```
#!/usr/local/bin/python3
```

```
import cgi
import os
import time
import sys
import yate
```

There's really
nothing new
here.

```
print(yate.start_response('text/plain'))
addr = os.environ['REMOTE_ADDR']
host = os.environ['REMOTE_HOST']
method = os.environ['REQUEST_METHOD']
cur_time = time.asctime(time.localtime())
print(host + ", " + addr + ", " + cur_time + ": " + method + ": ",
      end='', file=sys.stderr)
```

There's not much of a
response for now...so
just send back plain
text to the waiting
web browser.

```
print('OK.')
form = cgi.FieldStorage()
print(each_form_item + '->' + form[each_form_item].value,
      file=sys.stderr)
for each_form_item in form.keys():
      print(file=sys.stderr)
```



CGI Magnets Solution

You need a new CGI script called `add_timing_data.py`, which processes the data from a form and displays the data on your web server's console screen. The CGI needs to query the environment, arranging to display the logged data on one line. The code exists, but most of it is all over the floor. You were to rearrange the magnets to produce a working program.

```
#! /usr/local/bin/python3

import cgi
import os
import time
import sys
import yate

print(yate.start_response('text/plain'))
addr = os.environ['REMOTE_ADDR']
host = os.environ['REMOTE_HOST']
method = os.environ['REQUEST_METHOD']
cur_time = time.asctime(time.localtime())
print(host + ", " + addr + ", " + cur_time + ": " + method + ": ",
      end='', file=sys.stderr)
form = cgi.FieldStorage()
for each_form_item in form.keys():
    print(each_form_item + '->' + form[each_form_item].value,
          end=' ', file=sys.stderr)
print(file=sys.stderr)
print('OK.')
```

Ensure that this "print()" function does NOT take a newline.

Take a newline on standard error.



Test DRIVE

Let's use your form-generating CGI script from earlier to try out `add_timing_data.py`. As you enter data in the form and press the **Send** button, watch what happens on the web server's console.

Enter some data into your web form.

The web browser displays a very basic response. All is "OK".

The web server's logging screen displays the data that arrived, as well as the name associated with it.

```

File Edit Window Help DisplayingPOSTs
$ python3 simple_httpd.py
Starting simple_httpd on port: 8080
localhost - - [21/Sep/2010 17:34:54] "GET /cgi-bin/test_form.py HTTP/1.1" 200 -
localhost - - [21/Sep/2010 17:34:54] "GET /favicon.ico HTTP/1.1" 200 -
localhost - - [21/Sep/2010 17:35:24] "POST /cgi-bin/add_timing_data.py HTTP/1.1" 200 -
localhost, 127.0.0.1, Tue Sep 21 17:35:24 2010: POST: TimeValue->2.22
  
```

That worked perfectly. The data entered into the form is delivered to your CGI script on the your server. Your next challenge is to provide the same user input experience on an Android phone.

Ask for input on your Android phone

When you ask for user input on Android, the dialog that your user sees looks something like this example, which asks your user to confirm or change the web address and port for your server.



A single Android call creates this interface for you using the `dialogGetInput()` method:

```
title = 'Which server should I use?'
message = "Please confirm the server address/name to use for your athlete's timing data:"
data = 'http://192.168.1.33:8080'
resp = app.dialogGetInput(title, message, data).result
```

Pressing the **Ok** button sets `resp` to the data entered into the input area.

Pressing the **Cancel** button sets `resp` to `None`, which is Python's internal null-value.

The result of your user's interaction with the dialog is assigned to "resp".

Let's create some Android data-entry dialogs.



Exercise

Let's create a small Android app that interacts with your user twice. The first dialog asks the user to confirm the web address and port to use for the web server. Assuming your user taps the OK button on your dialog, a second dialog pops up to request the timing value to send to the server. As with the first dialog, tapping the OK button continues execution by sending the newly acquired timing value to the web server. Tapping Cancel at any time causes your app to exit.

Some of the code is provided for you. Your job is to complete the program. Write the code you think you need under this code, and call your program `get2inputsapp.py`:

```
import android
from urllib import urlencode
from urllib2 import urlopen

server_title = 'Which server should I use?'
server_msg = "Please confirm the server address/name to use for your athlete's timing data."
timing_title = 'Enter data'
timing_msg = 'Provide a new timing value:'
web_server = 'http://192.168.1.33:8080'
add_time_cgi = '/cgi-bin/add_timing_data.py'

app = android.Android()

def send_to_server(url, post_data=None):
    if post_data:
        page = urlopen(url, urlencode(post_data))
    else:
        page = urlopen(url)
    return(page.read().decode("utf8"))
```

There's
nothing new
here...you've
seen all of
this code
before.



You were to create a small Android app that interacts with your user *twice*. The first dialog asks the user to confirm the web address and port to use for the web server. Assuming your user taps the OK button on your dialog, a second dialog pops up to request the timing value to send to the server. As with the first dialog, tapping the OK button continues execution by sending the newly acquired timing value to the web server. Tapping Cancel at any time causes your app to exit.

Some of the code was provided for you. Your job was to complete the program by writing the code you think you need under this code and call your program `get2inputsapp.py`.

```

import android
from urllib import urlencode
from urllib2 import urlopen

server_title = 'Which server should I use?'
server_msg = "Please confirm the server address/name to use for your athlete's timing data:"
timing_title = 'Enter data'
timing_msg = 'Provide a new timing value:'
web_server = 'http://192.168.1.33:8080'
add_time_cgi = '/cgi-bin/add_timing_data.py'

app = android.Android()

def send_to_server(url, post_data=None):
    if post_data:
        page = urlopen(url, urlencode(post_data))
    else:
        page = urlopen(url)
    return(page.read().decode("utf8"))

resp = app.dialogGetInput(server_title, server_msg, web_server).result
    
```

The first dialog asks your user to confirm the web address and port to use.

.....

if resp is not None:

web_server = resp

resp = app.dialogGetInput(timing_title, timing_msg).result

if resp is not None:

new_time = resp

send_to_server(web_server + add_time_cgi, {'TimingValue': new_time})

If your user did NOT tap on the Cancel button...
...the second dialog asks for a new timing value.

Again, if your user did NOT tap on the Cancel button...
...the app sends the data to the web server.



Test DRIVE

Let's copy `get2inputsapp.py` to the emulator using the `adb` tool:

```
tools/adb push get2inputsapp.py /sdcard/sl4a/scripts
```

The `get2inputsapp.py` app appears on the list of scripts within SL4A. Go ahead and give it a tap:

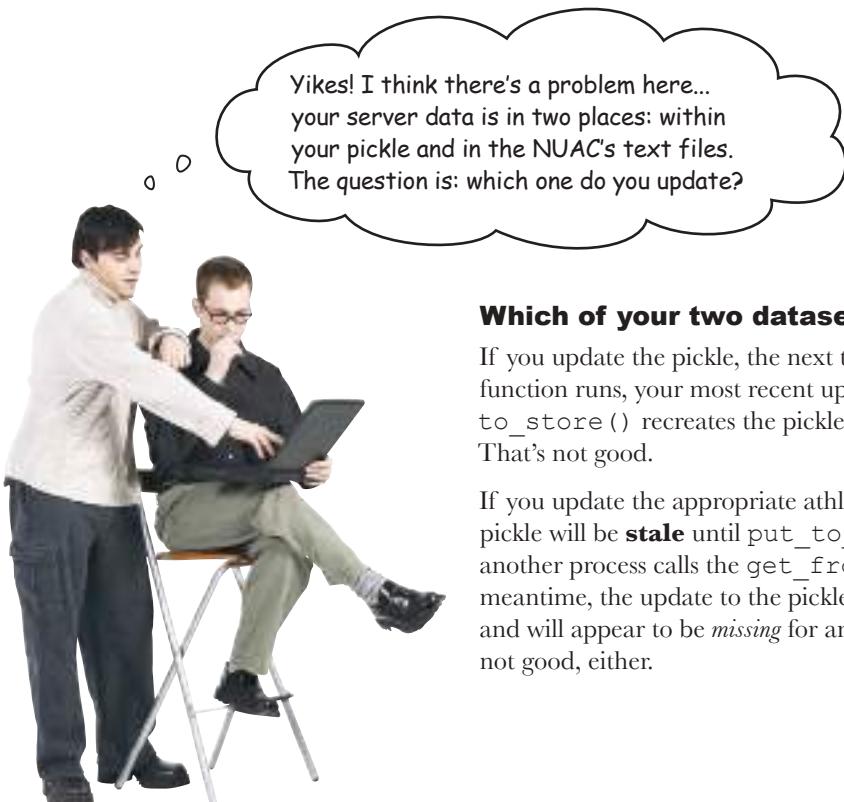


```
File Edit Window Help InputsFromAndroid
$ python3 simple_httpp.py
Starting simple_httpp on port: 8080
localhost - - [21/Sep/2010 17:34:54] "GET /cgi-bin/test_form.py HTTP/1.1" 200 -
localhost - - [21/Sep/2010 17:34:54] "GET /favicon.ico HTTP/1.1" 200 -
localhost - - [21/Sep/2010 17:35:24] "POST /cgi-bin/add_timing_data.py HTTP/1.1" 200 -
localhost, 127.0.0.1, Tue Sep 21 17:35:24 2010: POST: TimeValue->2.22
192.168.1.33 - - [21/Sep/2010 20:50:30] "POST /cgi-bin/add_timing_data.py HTTP/1.1" 200 -
localhost, 192.168.1.33, Tue Sep 21 20:50:30 2010: POST: TimingValue->2.56
```

The web server's log confirms the data was sent from your emulator.

Perfect. That's working, too. Regardless of where your data originates—on the Web or a phone—your app can send it to your web server.

It's time to update your server data



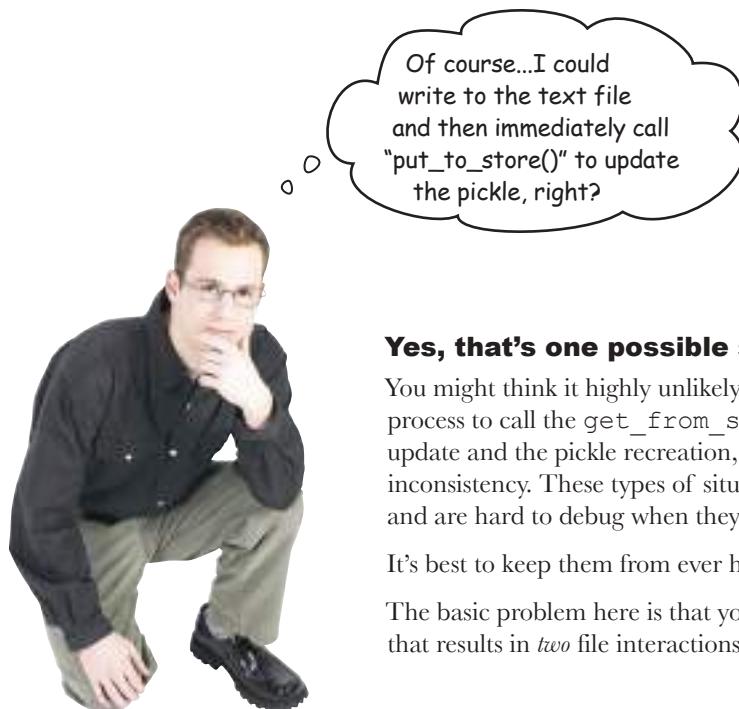
Which of your two datasets should you update?

If you update the pickle, the next time the `put_to_store()` function runs, your most recent update will vanish as `put_to_store()` recreates the pickle from the data in the text files. That's not good.

If you update the appropriate athlete's text file, the data in the pickle will be **stale** until `put_to_store()` runs again. If another process calls the `get_from_store()` function in the meantime, the update to the pickle might not have been applied and will appear to be *missing* for anyone reading your data. That's not good, either.



Avoid race conditions



Yes, that's one possible solution, but it's a poor one.

You might think it highly unlikely...but it is possible for another process to call the `get_from_store()` function *between* the text file update and the pickle recreation, resulting in a short period of data inconsistency. These types of situations are known as **race conditions** and are hard to debug when they occur.

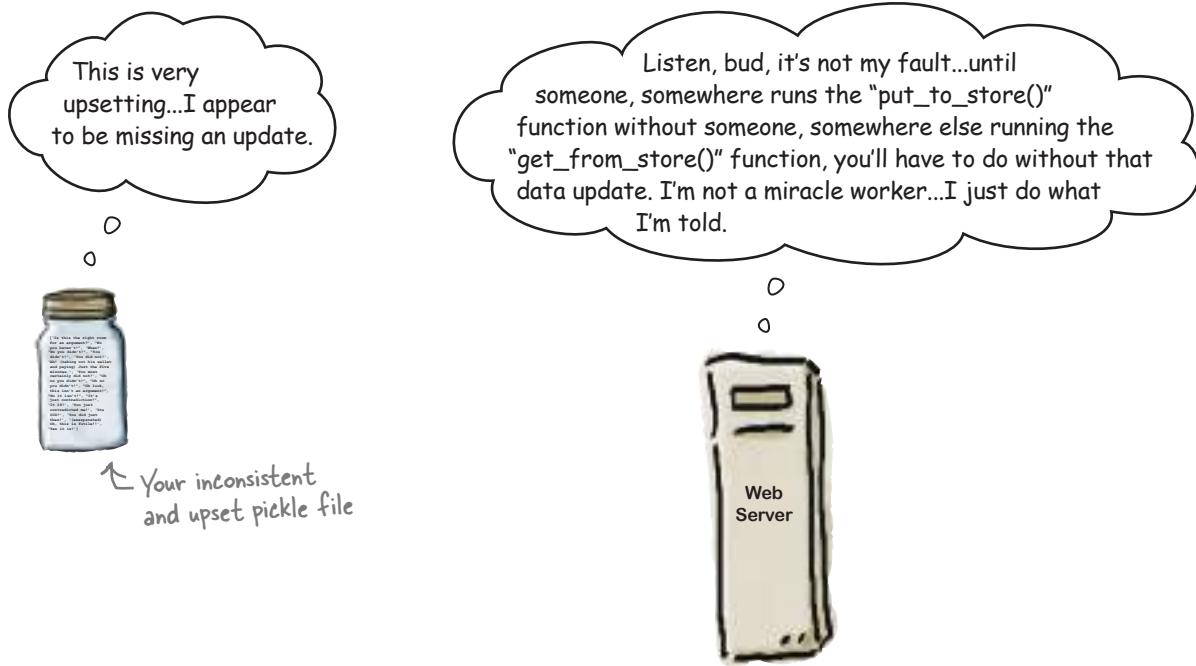
It's best to keep them from ever happening if you can.

The basic problem here is that you have *one* update with *one* piece of data that results in *two* file interactions. If nothing else, that's just wasteful.



You need a better data storage mechanism

Your initial text files and pickle design is fine when only one user is accessing the data. However, now that more than one person can access the data at any time, and from anywhere, your design is in need of improvement. Above all, you need to *avoid that race condition*.



there are no
Dumb Questions

Q: Surely you should have thought about this problem long ago and designed this “properly” from the start?

A: That’s certainly one way to look at things, and hindsight is always a wonderful thing! However, programs have a tendency to start out small, then grow to provide more features, which can introduce complexity. Recall that the coach’s app started life as a simple “standalone” text-based program, which was then moved to the Web to support multiple users. Part of the app was then redeveloped for use on an Android phone. And yes, if we’d known all of this ahead of time, we might have been designed it differently.

Q: So I’m facing a rewrite of large chunks of my code?

A: Let’s see. You did build your program using the MVC pattern, and you are using Python, so those two facts should take the sting out of any potential rewrite, assuming a rewrite is what’s required here.



Wouldn't it be dreamy if I could put my data in only one place and support all my app's requirements? But I know it's just a fantasy...

which database management system?

Use a database management system

You need to move away from your text file and pickle combination and use a *real* database management system. You have plenty of choices here...



All of these fine technologies will work, but they are overkill for your app's data requirements. And besides some of these are way beyond the NUAC's budget, let alone their ability to set up, run, and maintain such a system.

What you need is something that's effectively hidden from the NUAC yet lets you take advantage of what a database management system has to offer.

If only such a technology existed...

Python includes SQLite

Python 3 comes preinstalled with Release 3 of **SQLite**, a full-featured, zero-config, SQL-based data management system.

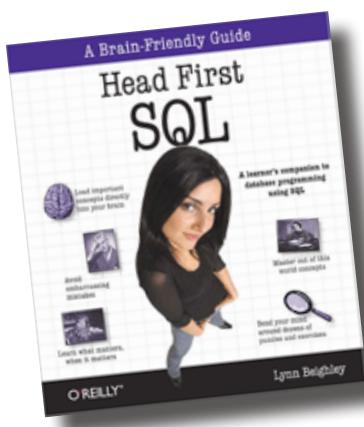
To use SQLite, simply import the `sqlite3` library and use Python's standardized database API to program it. There's really nothing to it: no database setup, no config, and no ongoing maintenance.

With your data stored in SQLite, rewrite your webapp's model code to use SQL to access, manipulate, and query your data. You can plan to move to one of the bigger database systems if and when your application needs dictate such a move.

SQLite sounds perfect for the NUAC's data, doesn't it?



Geek Bits —



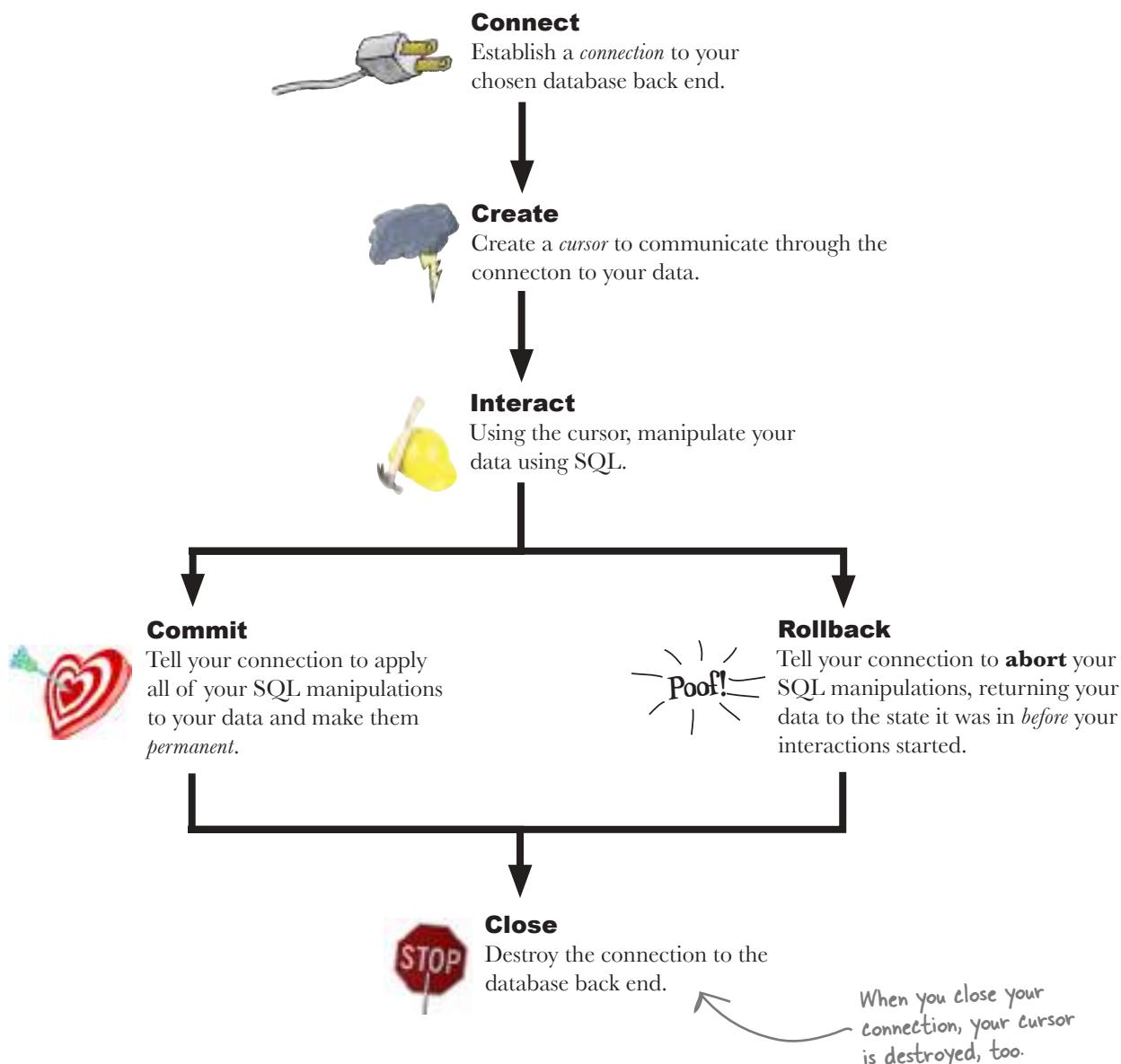
The material in this chapter assumes you are comfortable with SQL database technology. If you are new to SQL (or just need a quick refresher), take a look at *Head First SQL*, which comes **highly recommended**.



[Note from Marketing: Available anywhere good books are sold and to anyone with a valid credit card.]

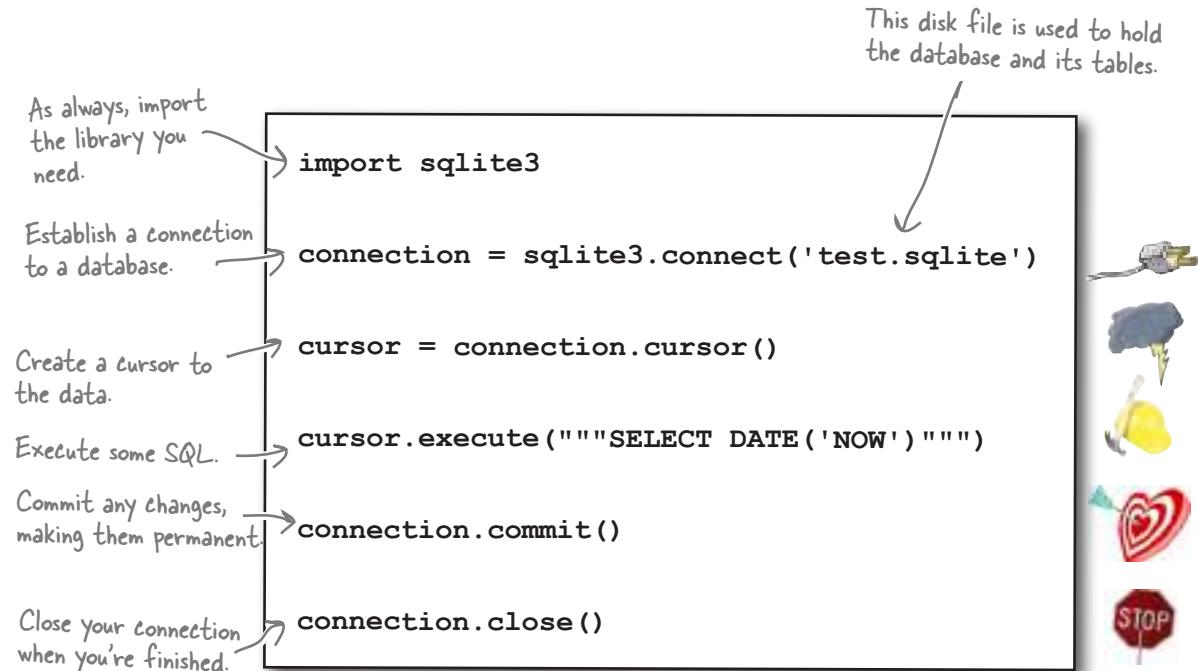
Exploit Python's database API

The **Python Database API** provides a standard mechanism for programming a wide variety of database management systems, including SQLite. The process you follow in your code is the same regardless of which back-end database you're using.



The database API as Python code

Here's how to implement an interaction with a database using the `sqlite3` module:



Depending on what happens during the **Interact** phase of the process, you either make any changes to your data permanent (`commit`) or decide to abort your changes (`rollback`).

You can include code like this in your program. It is also possible to interact with your SQLite data from within IDLE's shell. Whichever option you choose, you are interacting with your database using Python.

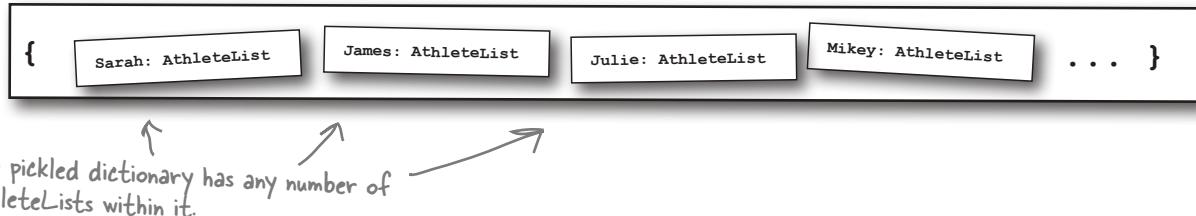
It's great that you can use a database to hold your data. But what schema should you use? Should you use one table, or do you need more? What data items go where? How will you design your database?

Let's start working on the answers to these questions.

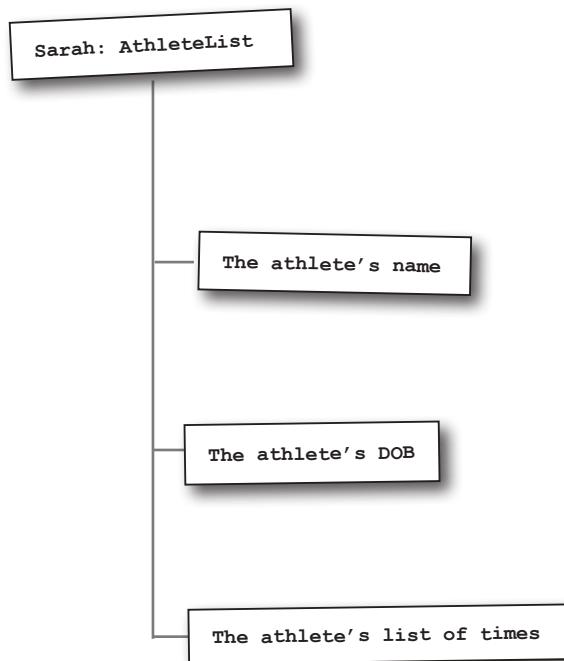
A little database design goes a long way

Let's consider how the NUAC's data is currently stored within your pickle.

Each athlete's data is an `AthleteList` object instance, which is associated with the athlete's name in a dictionary. The entire dictionary is pickled.



Each `AthleteList` has the following attributes:



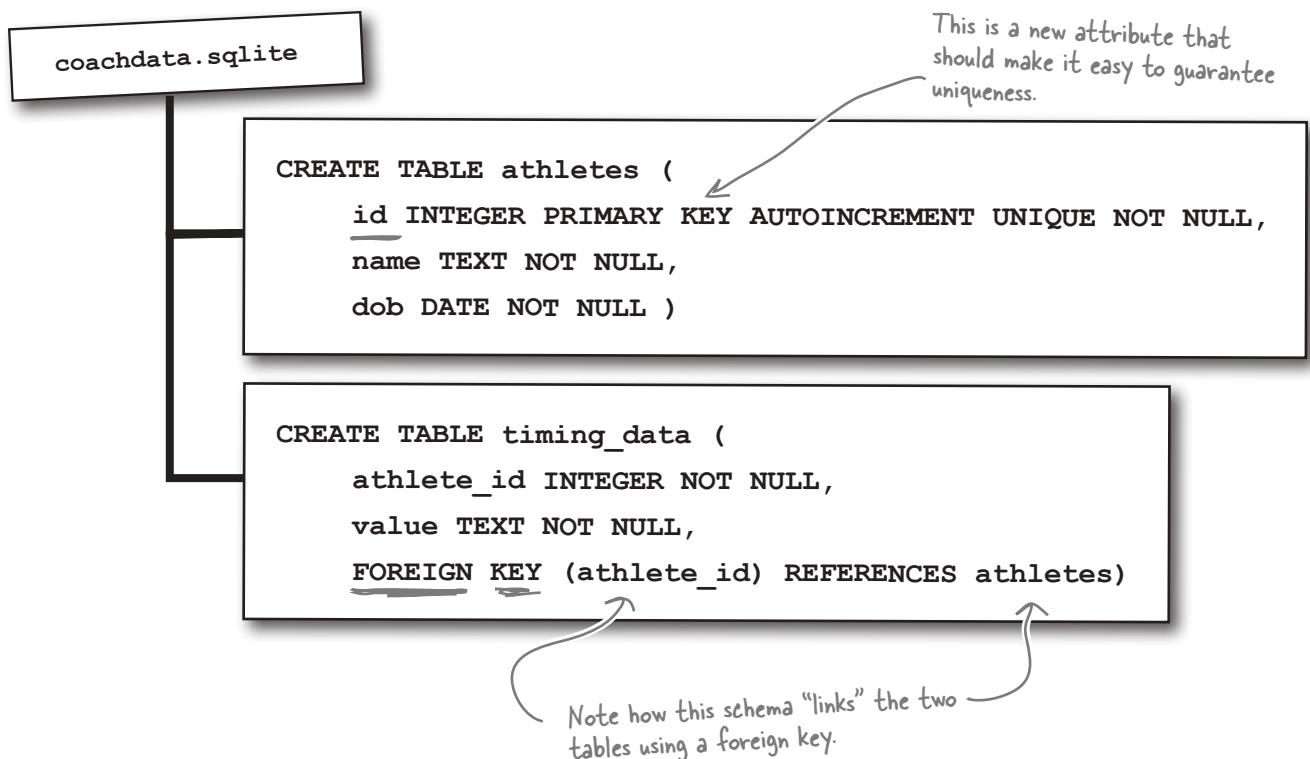
With this arrangement, it is pretty obvious which name, date of birth, and list of times is associated with which individual athlete. But how do you model these relationships within a SQL-compliant database system like *SQLite*?

You need to define your schema and create some tables.

Define your database schema

Here is a suggested SQL schema for the NUAC's data. The database is called `coachdata.sqlite`, and it has two related tables.

The first table, called `athletes`, contains rows of data with a unique ID value, the athlete's name, and a date-of-birth. The second table, called `timing_data`, contains rows of data with an athlete's unique ID and the actual time value.



There can be one *and only one* row of data for each athlete in the `athletes` table. For each athlete, the value of `id` is guaranteed to be *unique*, which ensures that two (or more) athletes with the same name are kept separate within the system, because they have different ID values.

Within the `timing_data` table, each athlete can have any number of time values associated with their unique `athlete_id`, with an individual row of data for each recorded time.

Let's look at some sample data.

What does the data look like?

If the two tables were created and then populated with the data from the NUAC's text files, the data in the tables might look something like this.

id	name	dob
1	James Lee	2002-03-14
2	Sarah Sweeney	2002-06-17
3	Vera Vi	2002-12-25
4	Julie Jones	2002-08-17
5	Sally Sanchez	2002-11-24
6	Mikey McManus	2002-02-24

This is what the data in the "athletes" table might look like, with one row of data for each athlete.

If you create these two tables then arrange for your data to be *inserted* into them, the NUAC's data would be in a format that should make it easier to work with.

Looking at the tables, it is easy to see how to add a new timing value for an athlete. Simply add another row of data to the `timing_data` table.

Need to add an athlete? Add a row of data to the `athletes` table.

Want to know the fastest time? Extract the smallest value from the `timing_data` table's `value` column?

Let's create and populate these database tables.

This is what the data in the "timing_data" table might look like, with multiple rows of data for each athlete and one row for each timing value.

athlete_id	value
1	2.01
1	2.16
1	2.22
1	2.34
1	2.45
1	3.01
1	3.1
1	3.21
2	2.18
2	2.21
2	2.22
2	2.25
2	2.39
2	2.54
2	2.55
2	2.58
3	2.41
3	2.49
3	3.01
3	3.02
3	3.11
3	3.23
4	2.11
4	2.23

There's more data in this table than shown here.



SQLite Magnets

Let's create a small Python program that creates the coachdata.sqlite database with the empty athletes and timing_data tables. Call your program `createDBtables.py`. The code you need is almost ready. Rearrange the magnets at the bottom of the page to complete it.

```
import sqlite3
```

```
cursor.execute("""CREATE TABLE athletes (
```

```
    athlete_id INTEGER NOT NULL,  
    value TEXT NOT NULL,  
FOREIGN KEY (athlete_id) REFERENCES athletes)""")
```

```
connection.commit()  
connection.close()
```

dob DATE NOT NULL) """")

```
cursor = connection.cursor()
```

```
cursor.execute("""CREATE TABLE timing_data (
```

```
connection = sqlite3.connect('coachdata.sqlite')
```

name TEXT NOT NULL,

`id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY, name VARCHAR(255) NOT NULL, email VARCHAR(255) NOT NULL, password VARCHAR(255) NOT NULL, created_at DATETIME NOT NULL, updated_at DATETIME NOT NULL`



SQLite Magnets Solution

Your job was to create a small Python program that creates the coachdata.sqlite database with the empty athletes and timing_data tables. You were to call your program createDBtables.py. The code you needed was almost ready, and you were to rearrange the magnets at the bottom of the page to complete it.

```
import sqlite3

connection = sqlite3.connect('coachdata.sqlite')
cursor = connection.cursor()

cursor.execute("""CREATE TABLE athletes (
    id INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE NOT NULL,
    name TEXT NOT NULL,
    dob DATE NOT NULL )""")

cursor.execute("""CREATE TABLE timing_data (
    athlete_id INTEGER NOT NULL,
    value TEXT NOT NULL,
    FOREIGN KEY (athlete_id) REFERENCES athletes)""")

connection.commit()
connection.close()
```

The commit isn't always required with most other database systems, but it is with SQLite.

Transfer the data from your pickle to SQLite

As well as writing the code to create the tables that you need, you also need to arrange to transfer the data from your existing model (your text files and pickle combination) to your new database model. Let's write some code to do that, too.

You can add data to an existing table with the SQL `INSERT` statement. Assuming you have data in variables called `name` and `dob`, use code like this to add a new row of data to the `athletes` table:

```
cursor.execute("INSERT INTO athletes (name, dob) VALUES (?, ?)", (name, dob))
```

You don't need to worry about supplying a value for the "id" column, because SQLite provides one for you automatically.



Ready Bake Python Code

Here's a program, called `initDBathletes.py`, which takes your athlete data from your existing model and loads it into your newly created `SQLite` database.

```
import sqlite3
connection = sqlite3.connect('coachdata.sqlite')
cursor = connection.cursor()

import glob
import athletemodel
data_files = glob.glob("../data/*.txt")
athletes = athletemodel.put_to_store(data_files)

for each_ath in athletes:
    name = athletes[each_ath].name
    dob = athletes[each_ath].dob

    cursor.execute("INSERT INTO athletes (name, dob) VALUES (?, ?)", (name, dob))
    connection.commit() ← Make the change(s) permanent.

connection.close()
```

What ID is assigned to which athlete?

You need to query the data in your database table to work out which ID value is automatically assigned to an athlete.

With SQL, the SELECT statement is the *query king*. Here's a small snippet of code to show you how to use it with Python, assuming the name and dob variables have values:

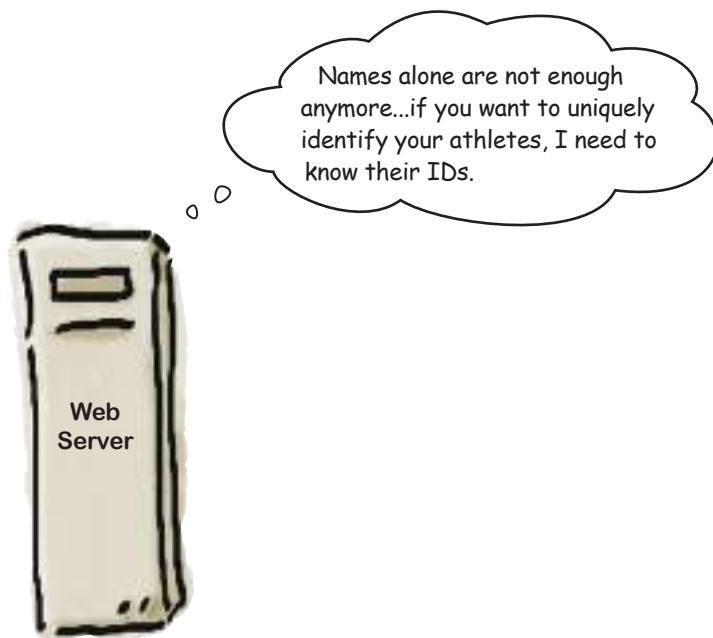
```
cursor.execute("SELECT id from athletes WHERE name=? AND dob=?", (name, dob))
```

Again, the placeholders indicate where the data values are substituted into the query.

If the query succeeds and returns data, it gets added to your cursor. You can call a number of methods on your cursor to access the results:

- `cursor.fetchone()` returns the next row of data.
- `cursor.fetchmany()` returns multiple rows of data.
- `cursor.fetchall()` returns all of the data.

Each of these cursor methods return a list of rows.



Insert your timing data

You're on a roll, so let's *keep coding* for now and produce the code to take an athlete's timing values out of the pickle and add them to your database. Specifically, you'll want to arrange to add a new row of data to the `timing_data` table for each time value that is associated with each athlete in your pickle.

Those friendly coders over at the Head First Code Review Team have just announced they've added a `clean_data` attribute to your `AthleteList` class. When you access `clean_data`, you get back a list of timing values that are sanitized, sorted, and free from duplicates. The Head First Code Review Team has excellent timing; that attribute should come in handy with your current coding efforts.



Sharpen your pencil

Again, it's OK to assume in your code that the "name" and "dob" variables exist and have values assigned to them.

Grab your pencil and write the lines of code needed to query the athletes table for an athlete's name and DOB, assigning the result to a variable called `the_current_id`. Write another query to extract the athlete's times from the pickle and add them to the timing data table.



Sharpen your pencil Solution

Query the "athletes" table for the ID.

You were to grab your pencil and write the lines of code needed to query the `athletes` table for an athlete's name and DOB, assigning the result to a variable called `the_current_id`. You were then to write another query to extract the athlete's times from the pickle and add them to the `timing_data` table.

Remember:
`"fetchone()"`
returns a list.

Take each of
the "clean"
times and use
it, together
with the ID,
within the
SQL "INSERT"
statement.

```
cursor.execute("SELECT id from athletes WHERE name=? AND dob=?",
              (name, dob))
```

```
the_current_id = cursor.fetchone()[0]
```

```
for each_time in athletes[each_ath].clean_data:
```

```
    cursor.execute("INSERT INTO timing_data (athlete_id, value) VALUES (?, ?)",
                  (the_current_id, each_time))
```

`connection.commit()` ← As always, make the
change(s) permanent.

It often makes sense
to split your execute
statement over multiple
lines.

Add the ID and the time
value to the "timing_
data" table.



Add the code to your `initDBathletes.py` code from earlier, just after the `connection.commit()` call. Rename your program `initDBtables.py`, now that both the `athletes` and `timing_data` tables are populated with data by a single program.

That's enough coding (for now). Let's transfer your pickled data.



Test DRIVE

You've got two programs to run now: `createDBtables.py` creates an empty database, defining the two tables, and `initDBtables.py` extracts the data from your pickle and populates the tables. Rather than running these programs within IDLE, let's use the Python command-line tool instead.

If you are running Windows,
replace "python3" with this:
"C:\Python31\python.exe".

Be careful
to run both
programs ONLY
once.

```
File Edit Window Help PopulateTheTables
$ python3 createDBtables.py
$ python3 initDBtables.py
$
```



Hello? Something happened there,
didn't it? I ran the programs but nothing
appeared on screen...how do I know if
anything worked?

SQLite data management tools

When it comes to checking if your manipulations of the data in your database worked, you have a number of options:

Life really is
too short.

- a Write more code to check that the database is in the state that you expect it.

Which can certainly work, but is error-prone, tedious, and way too much work.



- b Use the supplied "sqlite3" command-line tool.

Simply type `sqlite3` within a terminal window to enter the SQLite "shell." To find out which commands are available to you, type `.help` and start reading. The tool is a little basic (and cryptic), but it works.

That's a period,
followed by the
word "help".



- c Use a graphical database browser.

There are lots of these; just Google "sqlite database browser" for more choices than you have time to review. Our favorite is the SQLite Manager, which installs into the Firefox web browser as an extension.

Works great,
but only on
Firefox.



This is what
SQLite Manager
looks like.



ID	name	dob
1	James Lee	2002-3-14
2	Sarah Sweeney	2002-6-17
3	Vera Vi	2002-12-25
4	Julie Jones	2002-8-17
5	Sally Sanchez	2002-11-24
6	Mickey McManus	2002-2-24

Great, all of the athletes are in the "athletes" table.

But how do you integrate your new database into your webapp?

Integrate SQLite with your existing webapp



Joe: This should be easy. We just have to rewrite the code in `athletemodel.py` to use the database, while keeping the API the same.

Frank: What do you mean by *keeping the API the same*?

Joe: Well...take the `get_from_store()` function, for instance. It returns an `AthleteList` dictionary, so we need to make sure that when we update `get_from_store()` to use our database that it *continues* to return a dictionary, just as it's always done.

Frank: Ah, now I get it: we can query the database, grab all the data, turn it into a big dictionary containing all of our `AthleteList` objects and then return that to the caller, right?

Joe: Yes, exactly! And the best of it is that the calling code doesn't need to change at all. Don't you just love the beauty of MVC?

Frank: Umm...I guess so.

Jim: [cough, cough]

Frank: What's up, Jim?

Jim: Are you guys crazy?

Joe & Frank: What?!!

Jim: You are bending over backward to maintain compatibility with an API that exists only because of the way your data model was initially designed. Now that you've reimplemented how your data is stored in your model, you need to consider if you need to change your API, too.

Joe & Frank: Change our API? *Are you crazy?!*

Jim: No, not crazy, just pragmatic. If we can simplify the API by redesigning it to better fit with our database, then we should.

Joe: OK, but we haven't got all day, y'know.

Jim: Don't worry: it'll be worth the effort.



Long Exercise

Let's spend some time amending your model code to use your SQLite database as opposed to your pickle. Start with the code to your `athletemodel.py` module. Take a pencil and strike out the lines of code you no longer need.

```
import pickle

from athletelist import AthleteList

def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readline()
        templ = data.strip().split(',')
        return(AthleteList(templ.pop(0), templ.pop(0), templ))
    except IOError as ioerr:
        print('File error (get_coach_data): ' + str(ioerr))
        return(None)

def put_to_store(files_list):
    all_athletes = {}
    for each_file in files_list:
        ath = get_coach_data(each_file)
        all_athletes[ath.name] = ath
    try:
        with open('athletes.pickle', 'wb') as athf:
            pickle.dump(all_athletes, athf)
    except IOError as ioerr:
        print('File error (put_and_store): ' + str(ioerr))
    return(all_athletes)
```

```
def get_from_store():
    all_athletes = {}
    try:
        with open('athletes.pickle', 'rb') as athf:
            all_athletes = pickle.load(athf)
    except IOError as ioerr:
        print('File error (get_from_store): ' + str(ioerr))
    return(all_athletes)

def get_names_from_store():
    athletes = get_from_store()
    response = [athletes[each_ath].name for each_ath in athletes]
    return(response)
```

Remember: there's no requirement to maintain the existing API.





LONG Exercise SOLUTION

Let's spend some time amending your model code to use your SQLite database as opposed to your pickle. Start with the code to your `athletemodel.py` module. You were to take a pencil and strike out the lines of code you no longer need.

```

import pickle

from athletelist import AthleteList

def get_coach_data(filename):
    try:
        with open(filename) as f:
            data = f.readlines()
        templ = data.strip().split(',')
        return AthleteList(templ.pop(0), templ.pop(0), templ)
    except IOError as ioerr:
        print('File error (get_coach_data): ' + str(ioerr))
    return None

def put_to_store(files_list):
    all_athletes = {}
    for each_file in files_list:
        ath = get_coach_data(each_file)
        all_athletes[ath.name] = ath
    try:
        with open('athletes.pickle', 'wb') as athf:
            pickle.dump(all_athletes, athf)
    except IOError as ioerr:
        print('File error (put_and_store): ' + str(ioerr))
    return all_athletes

```

None of this code is needed anymore, because SQLite provides the data model for you.

```
def get_from_store():
    all_athletes = []
    try:
        with open('athletes.pickle', 'rb') as athf:
            all_athletes = pickle.load(athf)
    except IOError as ioerr:
        print('File error (get_from_store): ' + str(ioerr))
    return(all_athletes)
```

```
def get_names_from_store():
    athletes = get_from_store()
    response = [athletes[each_ath].name for each_ath in athletes]
    return(response)
```



This might seem a little
drastic...but sometimes a
redesign requires you to throw
away obsolete code.

You still need the list of names

Throwing away all of your “old” model code makes sense, but you still need to generate a list of names from the model. Your decision to use SQLite is about to pay off: all you need is a simple SQL SELECT statement.



Ready Bake Python Code

Here's the code for your new `get_names_from_store()` function:

```
import sqlite3

db_name = 'coachdata.sqlite'

def get_names_from_store():
    connection = sqlite3.connect(db_name)
    cursor = connection.cursor()
    results = cursor.execute("""SELECT name FROM athletes""")
    response = [row[0] for row in results.fetchall()]
    connection.close()
    return(response)
```

Connect to the database.

Extract the data you need.

Formulate a response.

Return the list of names to the caller.

I guess in this case it actually makes perfect sense to maintain the API for this call.



Get an athlete's details based on ID

In addition to the list of names, you need to be able to extract an athlete's details from the `athletes` table based on ID.



Ready Bake

Python Code

Here's the code for another new function called `get_athlete_from_id()`:

A new function gets the data associated with a specific ID.

Get the "name" and "DOB" values from the `athletes` table.

Get the list of times from the "timing_data" table.

Return the athlete's data to the caller.

```
def get_athlete_from_id(athlete_id):
    connection = sqlite3.connect(db_name)
    cursor = connection.cursor()

    results = cursor.execute("""SELECT name, dob FROM athletes WHERE id=?""",
                           (athlete_id,))
    (name, dob) = results.fetchone()

    results = cursor.execute("""SELECT value FROM timing_data WHERE athlete_id=?""",
                           (athlete_id,))
    data = [row[0] for row in results.fetchall()]

    response = {
        'Name': name,
        'DOB': dob,
        'data': data,
        'top3': data[0:3]}
    connection.close()
    return(response)
```

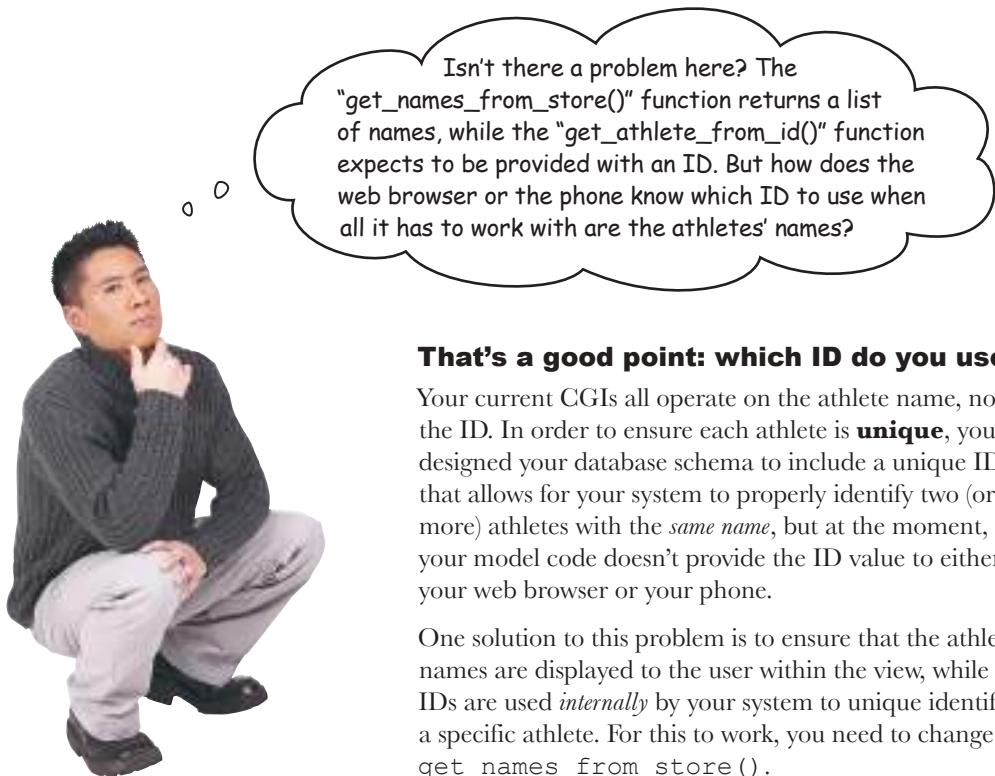
Note the use of the placeholder to indicate where the "athlete_id" argument is inserted into the SQL SELECT query.

Take the data from both query results and turn it into a dictionary.

This function is a more involved than `get_names_from_store()`, but not by much. It still follows the API used with working with data stored in SQLite. This is coming along nicely.

With the model code converted, you can revisit your CGI scripts to use your new model API.

Let's see what's involved with converting the CGIs.



That's a good point: which ID do you use?

Your current CGIs all operate on the athlete name, not the ID. In order to ensure each athlete is **unique**, you designed your database schema to include a unique ID that allows for your system to properly identify two (or more) athletes with the *same name*, but at the moment, your model code doesn't provide the ID value to either your web browser or your phone.

One solution to this problem is to ensure that the athlete names are displayed to the user within the view, while the IDs are used *internally* by your system to uniquely identify a specific athlete. For this to work, you need to change `get_names_from_store()`.



Here is the current code for your `get_names_from_store()` function. Rather than amending this code, create a new function, called `get_namesID_from_store()`, based on this code but including the ID values as well as the athlete names in its response. Write your new function in the space provided.

```
import sqlite3

db_name = 'coachdata.sqlite'

def get_names_from_store():
    connection = sqlite3.connect(db_name)
    cursor = connection.cursor()
    results = cursor.execute('""SELECT name FROM athletes""')
    response = [row[0] for row in results.fetchall()]
    connection.close()
    return(response)
```



Here is your current code for your `get_names_from_store()` function. Rather than amending this code, you were to create a new function, called `get_namesID_from_store()`, based on this code but including the ID values as well as the athlete names in its response. You were to write your new function in the space provided.

```
import sqlite3

db_name = 'coachdata.sqlite'

def get_names_from_store():
    connection = sqlite3.connect(db_name)
    cursor = connection.cursor()
    results = cursor.execute("""SELECT name FROM athletes""")
    response = [row[0] for row in results.fetchall()]
    connection.close()
    return(response)
```

```
def get_namesID_from_store():
    connection = sqlite3.connect(db_name)
    cursor = connection.cursor()
    results = cursor.execute("""SELECT name, id FROM athletes""")
    response = results.fetchall()
    connection.close()
    return(response)
```

Arrange to include the value of "id" in the SQL "SELECT" query.

There's no need to process "results" in any way...assign everything returned from the query to "response".

Remember: when you close your connection, your cursor is also destroyed, so you'll generate an exception if you try and use `return(results.fetchall())`.



This is the "generate_list.py" CGI script.

```
#! /usr/local/bin/python3

import glob
import athletemodel
import yate

data_files = glob.glob("data/*.txt")
athletes = athletemodel.put_to_store(data_files)

print(yate.start_response())
print(yate.include_header("NUAC's List of Athletes"))
print(yate.start_form("generate_timing_data.py"))
print(yate.para("Select an athlete from the list to work with:"))
for each_athlete in sorted(athletes):
    print(yate.radio_button("which_athlete", athletes[each_athlete].name))
print(yate.end_form("Select"))
print(yate.include_footer({"Home": "/index.html"}))
```

Part 1: With your model code ready, let's revisit each of your CGI scripts to change them to support your new model. At the moment, all of your code assumes that a list of athlete names or an AthleteList is returned from your model. Grab your pencil and amend each CGI to work with athlete IDs where necessary.

This is "generate_timing_data.py".

```
#! /usr/local/bin/python3

import cgi
import athletemodel
import yate

athletes = athletemodel.get_from_store()
form_data = cgi.FieldStorage()
athlete_name = form_data['which_athlete'].value

print(yate.start_response())
print(yate.include_header("NUAC's Timing Data"))
print(yate.header("Athlete: " + athlete_name + ", DOB: " + athletes[athlete_name].dob + "."))
print(yate.para("The top times for this athlete are:"))
print(yate.u_list(athletes[athlete_name].top3))
print(yate.para("The entire set of timing data is: " + str(athletes[athlete_name].clean_data) +
               " (duplicates removed)."))
print(yate.include_footer({"Home": "/index.html", "Select another athlete": "generate_list.py"}))
```

Note the change to the title.

This "Sharpen" is continued on the next page, but no peeking! Don't flip over until you've amended the code on this page.

Another title change.



This is the
“generate_names.py”
CGI.

```
#! /usr/local/bin/python3
import json

import athletemodel
import yate

names = athletemodel.get_names_from_store()

print(yate.start_response('application/json'))
print(json.dumps(sorted(names)))
```

And here is the
“generate_data.py”
CGI.

```
#! /usr/local/bin/python3
import cgi
import json
import sys

import athletemodel
import yate

athletes = athletemodel.get_from_store()

form_data = cgi.FieldStorage()
athlete_name = form_data['which_athlete'].value

print(yate.start_response('application/json'))
print(json.dumps(athletes[athlete_name].as_dict))
```

Part 2: You’re not done with that pencil just yet! In addition to amending the code to the CGIs that support your web browser’s UI, you also need to change the CGIs that provide your webapp data to your Android app. Amend these CGIs, too.

Sharpen your pencil Solution

This is the "generate_list.py" CGI script.

```
#! /usr/local/bin/python3
import cgi
import athletemodel
import yate

data_files = glob.glob("data/*")
athletes = athletemodel.get_namesID_from_store(data_files)

print(yate.start_response())
print(yate.include_header("NUAC's List of Athletes"))
print(yate.start_form("generate_timing_data.py"))
print(yate.para("Select an athlete from the list to work with:"))
for each_athlete in sorted(athletes):
    print(yate.radio_button('which_athlete', each_athlete[0], each_athlete[1]))
print(yate.end_form("Select"))
print(yate.include_footer({"Home": "/index.html"}))
```

This is "generate_timing_data.py".

```
#! /usr/local/bin/python3
import cgi
import athletemodel
import yate

athletes = athletemodel.get_namesID_from_store()
form_data = cgi.FieldStorage()
athlete_name = form_data['which_athlete'].value
athlete = athletemodel.get_athlete_from_id(athlete_name)
print(yate.start_response())
print(yate.include_header("NUAC's Timing Data"))
print(yate.header("Athlete: " + athlete['Name'] + ", DOB: " + athlete['DOB']))
print(yate.para("The top times for this athlete are:"))
print(yate.ul_list(athlete['times'], athlete['top3']))
print(yate.para("The entire set of timing data is: " + str(athlete['data']) +
               " (duplicates removed.)"))
print(yate.include_footer({"Home": "/index.html", "Select another athlete": "generate_list.py"}))
```

Part 1: With your model code ready, you were to revisit each of your CGI scripts to change them to support your new model. At the moment, all of your code assumes that a list of athlete names or an AthleteList is returned from your model. You were to grab your pencil and amend each CGI to work with athlete IDs where necessary.

You no longer need the "glob" module, as "get_namesID_from_store()" does all this work for you.

The "athletes" are now a list of lists, so amend the code to get at the data you need.

each_athlete[0], each_athlete[1]

radio_button_id() ?!

It looks like you might need a slightly different "radio_button()" function!?

Get the athlete's data from the model, which returns a dictionary.

Use the returned data as needed, accessing each of the dictionary key/values to get at the athlete's data.

athlete['Name'] + " DOB: " + athlete['DOB']

str(athlete['data'])



This is the "generate_names.py" CGI.

```
#! /usr/local/bin/python3
import json

import athletemodel
import yate

names = athletemodel.get_names_from_store()

print(yate.start_response('application/json'))
print(json.dumps(sorted(names)))
```

And here is the "generate_data.py" CGI.

```
#! /usr/local/bin/python3
import cgi
import json
import sys

import athletemodel
import yate

athletes = athletemodel.get_from_store()

form_data = cgi.FieldStorage()
athlete_name = form_data['which_athlete'].value
athlete = athletemodel.get_athlete_from_id(athlete_id)
print(yate.start_response('application/json'))
print(json.dumps(athletes[athlete_name].as_dict))
```

Part 2: You weren't done with that pencil just yet! In addition to amending the code to the CGIs that support your web browser's UI, you also needed to change the CGIs that provide your webapp data to your Android app. You were to amend these CGIs, too.

The tiniest of changes need to be made to these CGIs, because your Android app is only interested in your webapp's data, NOT all of that generated HTML.

Add this code to "yate.py" to support the creation of radio buttons that provide a value for the button that differs from the button text.

```
def radio_button_id(rb_name, rb_value, rb_id):
    return('<input type="radio" name="' + rb_name +
          '" value="' + str(rb_id) + '">' + rb_value + '<br />')
```

A third argument lets you specify an ID to go with the radio button.



Test DRIVE

Before you run your amended webapp, be sure to move your SQLite database into the top-level directory of your webapp (that is, into the same folder your `index.html` file). That way, your model code can find it, so move it into your webapp's root folder *now*. When you are ready, take your SQL-powered webapp for a spin.

Start (or
restart) your
web server.

```
File Edit Window Help StartYourWebEngine
$ python3 simple_httpd.py
Starting simple_httpd on port: 8080
```

Welcome to the National Underage Athletics Committee's Website

Welcome to the NUAC's Website.

Here is our athlete's timing data. Enjoy!

See you on the track!

Click on the link on the home page.

And there's Sally's timing data.

The NUAC's List of Athletes

Select an athlete from the list to work with:

- James Lee
- Julie Jones
- Mickey McManus
- Sally Sanchez
- Sarah Sweeney
- Vera Vi

Select

NUAC's Timing Data

Athlete: Sally Sanchez, DOB: 2002-11-24,

The top times for this athlete are:

- * 2.11
- * 2.26
- * 2.31

The entire set of timing data is: [2.11, 2.26, 2.31, 2.32, 2.41, 2.44, 2.51, 2.59, 3.00, 3.01] (duplicates removed).

[Home](#) [Select another athlete](#)

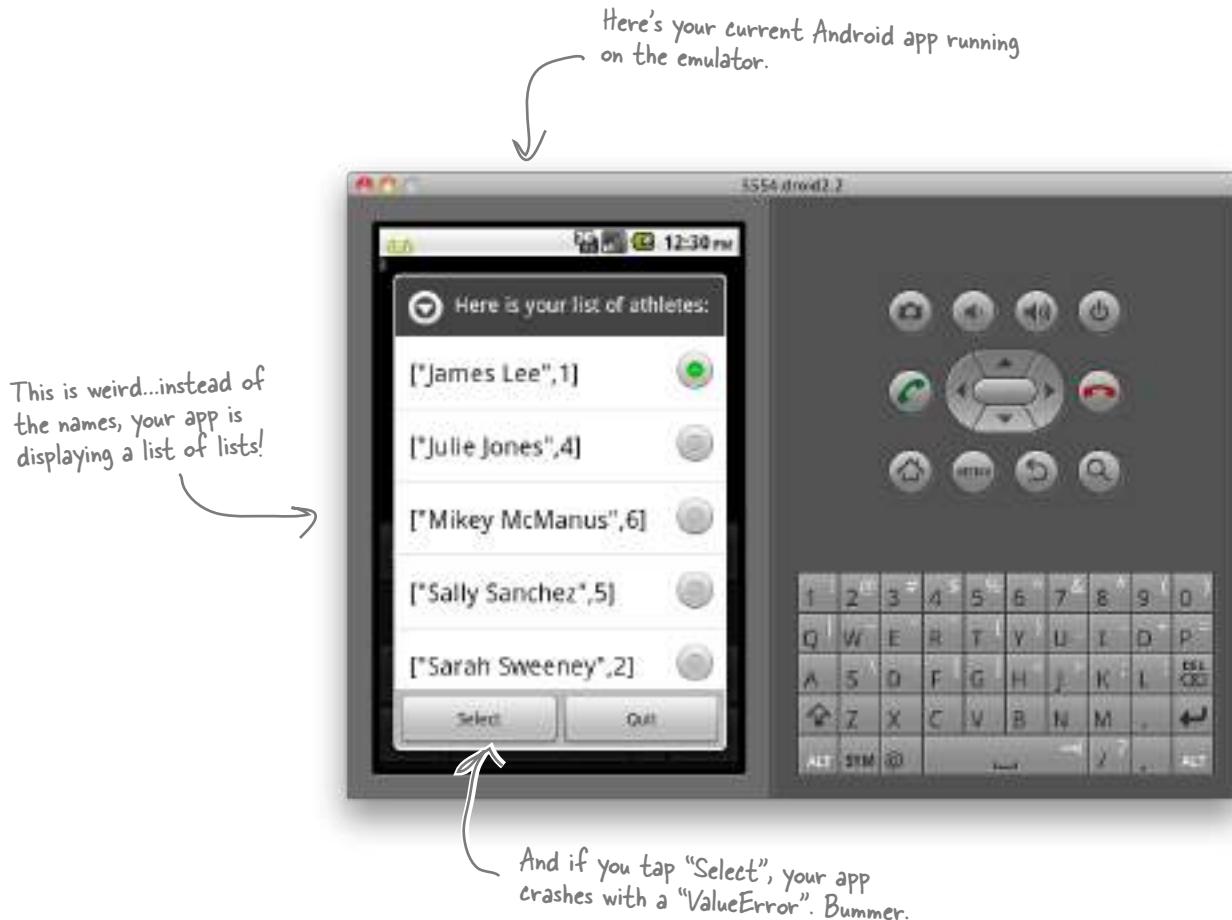
Display the list of athlete names as radio buttons.

That worked well. But what about your Android app?

You need to amend your Android app, too

Unlike your HTML-based webapp, where all of your code resides and is executed on your web server, your Android app runs on your phone and it is programmed to work with a list of names, not a list of names *and* athlete IDs.

When you run `coachapp.py` on your emulator, weirdness ensues...



Just like with the CGI scripts, you need to amend your Android app to work with the data that's now arriving from your web server—that is, a list of lists as opposed to a list.

That shouldn't take too long, should it?



Here is your current `coachapp.py` code, which you need to amend to support the way your webapp's model now works. Grab a pencil and make the necessary changes to this code.

```

import android, json, time

from urllib import urlencode
from urllib2 import urlopen

hello_msg      = "Welcome to NUAC's Timing App"
list_title     = 'Here is your list of athletes:'
quit_msg       = "Quitting NUAC's App."
web_server     = 'http://192.168.1.34:8080'
get_names_cgi = 'cgi-bin/generate_names.py'
get_data_cgi   = '/cgi-bin/generate_data.py'

def send_to_server(url, post_data=None):
    # There is no change to this code from the previous chapter.

app = android.Android()

def status_update(msg, how_long=2):
    # There is no change to this code from the previous chapter.

status_update(hello_msg)
athlete_names = sorted(json.loads(send_to_server(web_server + get_names_cgi)))
app.dialogCreateAlert(list_title)
app.dialogSetSingleChoiceItems(athlete_names)
app.dialogSetPositiveButtonText('Select')
app.dialogSetNegativeButtonText('Quit')
app.dialogShow()
resp = app.dialogGetResponse().result

if resp['which'] in ('positive'):
    selected_athlete = app.dialogGetSelectedItems().result[0]
    which_athlete = athlete_names[selected_athlete]
    athlete = json.loads(send_to_server(web_server + get_data_cgi, {'which_athlete': which_athlete}))
    athlete_title = athlete['Name'] + ' (' + athlete['DOB'] + ')', top 3 times:'
    app.dialogCreateAlert(athlete_title)
    app.dialogSetItems(athlete['Top3'])
    app.dialogSetPositiveButtonText('OK')
    app.dialogShow()
    resp = app.dialogGetResponse().result
status_update(quit_msg)

```



Exercise Solution

Here is your current `coachapp.py` code, which you need to amend to support the way your webapp's model now works. You were to grab a pencil and make the necessary changes to this code.

```

import android, json, time
from urllib import urlencode
from urllib2 import urlopen

hello_msg      = "Welcome to NUAC's Timing App"
list_title     = 'Here is your list of athletes:'
quit_msg       = "Quitting NUAC's App."
web_server     = 'http://192.168.1.34:8080'
get_names_cgi  = 'cgi-bin/generate_names.py'
get_data_cgi   = '/cgi-bin/generate_data.py'

def send_to_server(url, post_data=None):
    # There is no change to this code from the previous chapter.

app = android.Android()

def status_update(msg, how_long=2):
    # There is no change to this code from the previous chapter.

status_update(hello_msg) athletes =
athletes_names = sorted(json.loads(send_to_server(web_server + get_names_cgi)))
app.dialogCreateAlert(list_title) athlete_names = [ath[0] for ath in athletes]

app.dialogSetSingleChoiceItems(athlete_names)
app.dialogSetPositiveButtonText('Select')
app.dialogSetNegativeButtonText('Quit')
app.dialogShow()
resp = app.dialogGetResponse().result

if resp['which'] in ('positive'):
    selected_athlete = app.dialogGetSelectedItems().result[0]
    which_athlete = selected_athlete['id'] athletes[selected_athlete][1]
    athlete = json.loads(send_to_server(web_server + get_data_cgi, {'which_athlete': which_athlete}))
    athlete_title = athlete['Name'] + ' (' + athlete['DOB'] + ')', top 3 times:
    app.dialogCreateAlert(athlete_title)
    app.dialogSetItems(athlete['top3']) athlete['top3']
    app.dialogSetPositiveButtonText('OK')
    app.dialogShow()
    resp = app.dialogGetResponse().result
status_update(quit_msg)

```

Extract the athlete names ONLY from the list of lists.

This is a cool use of a comprehension.

Determine the ID associated with the selected athlete.

A small adjustment to next line is needed to access the "top3" attribute.

Android Pool Puzzle



Your **job** is to take the code from the pool and place it into the blank lines in the program. Your **goal** is to write the code to have your app provide the user with a mechanism to add a timing value to the server for the currently selected athlete. For now, send your data to the cgi-bin/add_timing_data.py CGI script.

Hint: the code from get2inputsapp.py (from earlier in this chapter) should come in handy here.

```
app.dialogSetNegativeButtonText('Add Time')
```

Add another button to the existing dialog in the current version of your app.

...

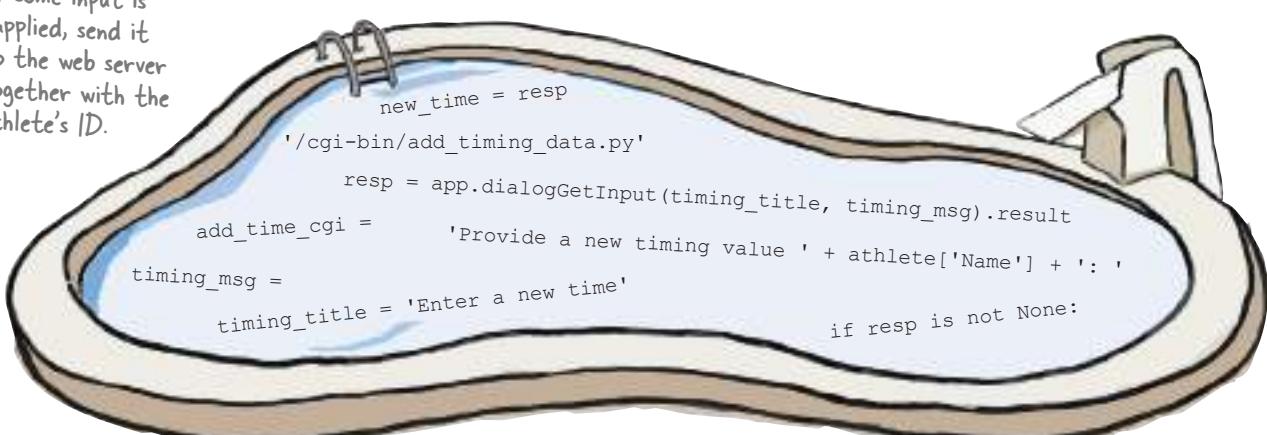
```
if resp['which'] in ('positive'):
    pass
elif resp['which'] in ('negative'):
```

Based on the button that's tapped, either do nothing ("pass") or start a new dialog with the user.

```
send_to_server(web_server + add_time_cgi, {'Time': new_time, 'Athlete': which_athlete})
```

If some input is supplied, send it to the web server together with the athlete's ID.

```
new_time = resp
'/cgi-bin/add_timing_data.py'
resp = app.dialogGetInput(timing_title, timing_msg).result
add_time_cgi =      'Provide a new timing value ' + athlete['Name'] + ': '
timing_msg =        'Enter a new time'
timing_title =      'Enter a new time'
if resp is not None:
```



Android Pool Puzzle Solution



Your **job** was to take the code from the pool and place it into the blank lines in the program. Your **goal** was to write the code to have your app provide the user with a mechanism to add a timing value to the server for the currently selected athlete. For now, you were to send your data to the cgi-bin/add_timing_data.py CGI script.

Hint: the code from get2inputsapp.py (from earlier in this chapter) should come in handy here.

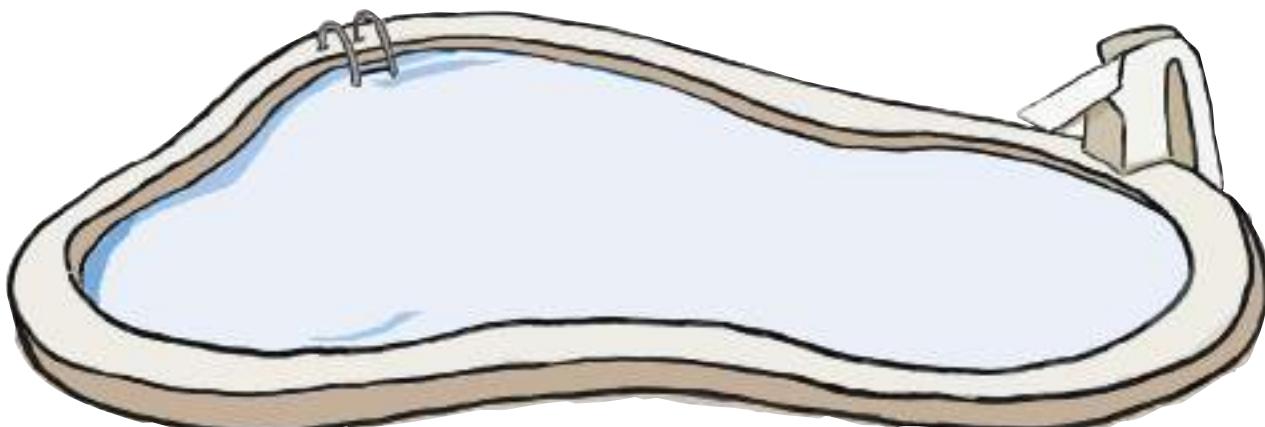
```
app.dialogSetNegativeButtonText('Add Time')
```

```
...
```

```
if resp['which'] in ('positive'):
    pass
elif resp['which'] in ('negative'):
    timing_title = 'Enter a new time'
    timing_msg =  'Provide a new timing value ' + athlete['Name'] + ': '
    add_time_cgi = '/cgi-bin/add_timing_data.py'
.....
resp = app.dialogGetInput(timing_title, timing_msg).result
if resp is not None:
    new_time = resp
    send_to_server(web_server + add_time_cgi, {'Time': new_time, 'Athlete': which_athlete})
```

Define the dialog's titles and specify the CGI to send the data to.

Display the dialog and wait for some user input.





Test DRIVE

Use the tools/adb command to copy your latest app to the emulator, and give your app a go.

Select "Vera" from the list of athletes...

...to see Vera's top 3 times, then tap the "Add Time" button...

...to enter a new time, which is then sent to your web server.

```

File Edit Window Help DataFromAndroid
$ python3 simple_httpd.py
Starting simple_httpd on port: 8080
198.162.1.34 - - [27/Sep/2010 14:51:47] "GET /cgi-bin/generate_names.py HTTP/1.1" 200 -
198.162.1.34 - - [27/Sep/2010 14:52:01] "POST /cgi-bin/generate_data.py HTTP/1.1" 200 -
198.162.1.34 - - [27/Sep/2010 14:52:19] "POST /cgi-bin/add_timing_data.py HTTP/1.1" 200 -
localhost, 198.162.1.34, Mon Sep 27 14:52:19 2010: POST: Athlete->3 Time->1.33

```

Great: your data is sent from your Android app to your web server.

Update your SQLite-based athlete data

All that's left is to change the `cgi-bin/add_timing_data.py` CGI script to write your submitted data to your database, as opposed to the web server's console screen.

At this point, it's a trivial exercise, because a single SQL `INSERT` statement will do the heavy lifting.

The screenshot shows a terminal window titled "add_timing_data.py - /Users/barryp/HeadFirstPython/chapter9/cgi-bin/add_timing_data.py". The code is as follows:

```
#! /usr/local/bin/python3
import cgi
import sqlite3
import yate

print(yate.start_response('text/plain'))
form = cgi.FieldStorage()
the_id = form_data['Athlete'].value
the_time = form_data['Time'].value

connection = sqlite3.connect('coachdata.sqlite')
cursor = connection.cursor()
cursor.execute("INSERT INTO timing_data (athlete_id, value) VALUES (?, ?)",
               (the_id, the_time))
connection.commit()
connection.close()
print('OK.')
```

Annotations with arrows explain the code:

- An arrow points from the line `the_id = form_data['Athlete'].value` to the text "Get the data sent to your web browser from your Android app."
- A curly brace annotation covers the `cursor.execute` line and the line below it, with the text "INSERT the data into your 'timing_data' table."

With this version of your CGI script running on your web server, any new times entered by anyone on an Android phone are added to the data in the database.

The NUAC no longer has to worry about adding data to text files, because the files are effectively obsoleted by the use of SQLite.

You've produced a robust solution that is more manageable, scalable, programmable, and extendable. And it's all thanks to the power of Python, its database API and the inclusion of `sqlite3` in the standard library.

All that's left to do is sit back, relax and bask in the glory of your latest programming creation...

The NUAC is over the moon!

Of course, your use of SQLite gives you more than just easy insertions of data. With the NUAC's data in tables, it's easy to answer some of the questions that have been on their mind.



To answer these and other queries on the data in the NUAC's database, you'll have to bone up on your SQL. Then it's up to you to take it from there.

You've converted your webapp to use an SQL database. As your data management needs increase, you can consider alternative *heavy-duty* data management technologies as needed.

This is great work. Your webapp is ready for the big time.



Your Python Toolbox

You've got Chapter 9 under your belt and you've added some key Python tools to your every expanding Python toolbox.

Python Lingo

- "Database API" – a standardized mechanism for accessing an SQL-based database system from within a Python program.

Database Lingo

- "Database" – a collection of one or more tables.
- "Table" – a collection of one or more rows or data, arranged as one or more columns.
- "SQL" – the "Structured Query Language" is the language of the database world and it lets you work with your data in your database using statements such as CREATE, INSERT, and SELECT.



BULLET POINTS

- The `fieldStorage()` method from the standard library's `cgi` module lets you access data sent to your web server from within your CGI script.
- The standard `os` library includes the `environ` dictionary providing convenient access to your program's environment settings.
- The SQLite database system is included within Python as the `sqlite3` standard library.
- The `connect()` method establishes a connection to your database file.
- The `cursor()` method lets you communicate with your database via an existing connection.
- The `execute()` method lets you send an SQL query to your database via an existing cursor.
- The `commit()` method makes changes to your database permanent.
- The `rollback()` method cancels any pending changes to your data.
- The `close()` method closes an existing connection to your database.
- The "?" placeholder lets you parameterize SQL statements within your Python code.

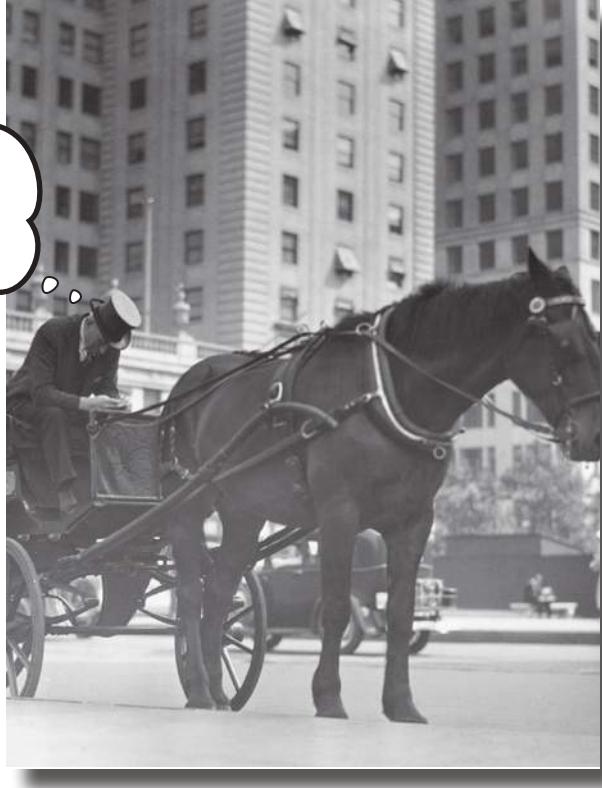
10 scaling your webapp



Getting real



It all started with the internal combustion engine, then it was the electric engine, and now there's App Engine. Will this torture never end?



The Web is a great place to host your app...until things get real.

Sooner or later, you'll hit the jackpot and your webapp will be *wildly successful*. When that happens, your webapp goes from a handful of hits a day to thousands, possibly ten of thousands, *or even more*. Will you be ready? Will your web server handle the **load**? How will you know? What will it **cost**? *Who will pay*? Can your data model scale to millions upon millions of data items without *slowing to a crawl*? Getting a webapp up and running is easy with Python and now, thanks to Google App Engine, **scaling** a Python webapp is achievable, too. So...flip the page and find out how.

There are whale sightings everywhere

The Head First Whale Watching Group (HFWWG) coordinates the live cetacean sightings for the entire country. To date, they've provided a PDF form on their website that members of the public can download, fill in, and mail to the HFWWG central office.

The form contains the essential data needed to record the sighting:

**Head First Whale Watching Group
Casual Sighting Form**



Name: _____

Email: _____

Date: _____

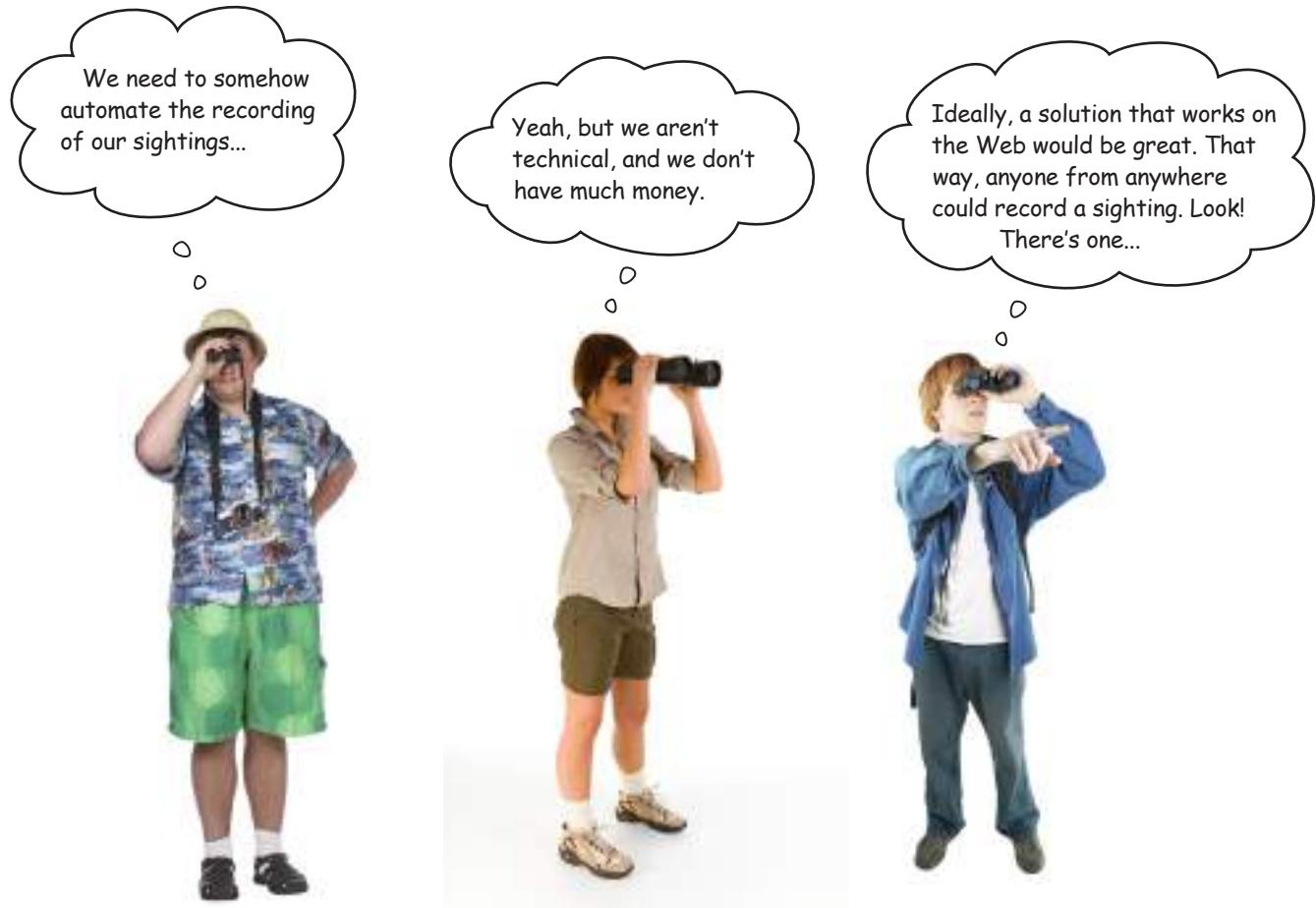
Time: _____

Location: _____

Fin Type:	Falcate	Triangular	Rounded	
Whale Type:	Humpback	Orca	Blue	Killer
	Beluga	Fin	Gray	Sperm
Blow Type:	Tall	Bushy	Dense	
Wave Type:	Flat	Small	Moderate	
	Large	Breaking	High	

After a busy sightings weekend, the central office is swamped with completed forms for **thousands** of sightings...which is a *data-entry nightmare* as all those forms can take an age to process manually. There's nothing worse than being stuck in front of your computer entering data when all you want to do is be out on the water looking for humpbacks...

The HFWWG needs to automate



Suggesting to the HFWWG that they invest in an expensive web hosting solution isn't going to make you any friends. It's way too expensive to buy the capacity they'll need for the busy weekends and a total waste of capacity when sightings are infrequent.

Suggesting that they invest in a large, state-of-the-art web server that can be hosted in the central office is also a nonstarter: there's no one to look after a setup like that, and the broadband link required to handle the anticipated traffic would blow their budget right out of the water.

Is there another option?

Build your webapp with Google App Engine

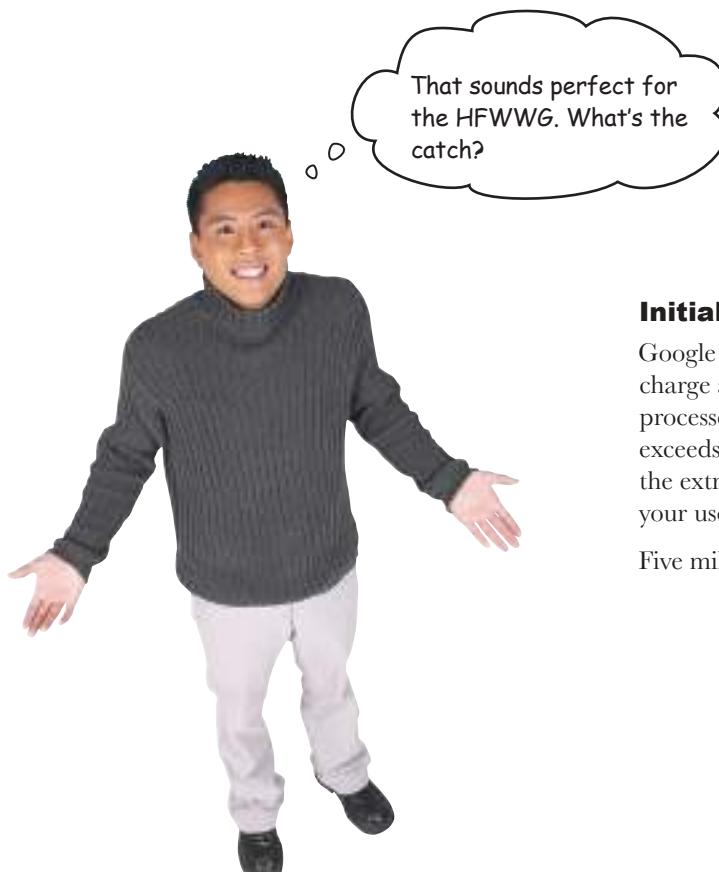
Google App Engine (GAE) is a set of technologies that lets you host your webapp on Google's cloud computing infrastructure.

GAE constantly monitors your running webapp and, based on your webapp's current activity, adjusts the resources needed to serve up your webapp's pages. When things are busy, GAE increases the resources available to your webapp, and when things are quiet, GAE reduces the resources until such time as extra activity warrants increasing them again.

On top of this, GAE provides access to Google's *BigTable* technology: a set of database technologies that make storing your webapp's data a breeze. Google also backs up your webapp's data on a regular basis, replicates your webapp over multiple, geographically dispersed web servers, and keeps App Engine running smoothly 24/7.

And the *best part*? GAE can be programmed with Python.

And the *even better part*? You can start running your webapp on GAE *for free*.



Initially, there isn't one.

Google provides this webapp hosting service at no charge and will continue to do so until your webapp processes five million page views per month. Once it exceeds this threshold, you'll need to pay Google for the extra capacity used. If you never reach the limit, your use of GAE is not charged.

Five million page views? That's a lot of sightings...

Download and install App Engine

When your webapp is ready for deployment, you'll upload it to the Google cloud and run it from there. However, during development, you can run a test version of your webapp locally on your computer. All you need is a copy of the GAE SDK, which is available from here:



<http://code.google.com/appengine/>

Download the GAE Python SKD for your operating system. Windows, Mac OS X, and Linux are all supported, and installation is straightforward.

After installation, Windows and Mac OS X users will find a nice, graphical front end added to their system.

On Linux, a new folder called "google_appengine" is created after a successful install.

GAE uses Python 2.5

The version of Python built into GAE is a modified version of the Python 2.5 release. As when you worked with Python for Android, the fact that you aren't running Python 3 isn't such a big deal with GAE, although you do need to ensure Python 2.5 is installed on your computer. Open up a terminal window and type:

```
python2.5 -v
```

If this command gives an error, pop on over to the Python website and grab the 2.5 release for your operating system.

*there are no
Dumb Questions*

Q: Aren't things going backward here? First, there was Python 3, then it was Python 2.6 for Android, and now we are dropping down to 2.5 for App Engine? What gives?

A: That's a great question. It's important to remember to always code to the restrictions placed on you. You might think that it sucks that GAE runs on Python 2.5, but you shouldn't. Think of it as just another restriction placed on the code you write—that is, it must target Release 2.5 of Python. As with the Android code you created in the previous chapters, the GAE code you are about to write is not all that different than the Python code for 3. In fact, you will be hard pressed to spot the difference.

Make sure App Engine is working

The environment supported by GAE within the Google cloud supports standard CGI or Python's WSGI. To build a GAE-compatible webapp, you need three things: a **folder** to hold your webapp's files, some **code** to execute, and a **configuration file**.

To test your setup, create a folder called `mygaetest`. Within the folder, create a small CGI you can use to test GAE. Call this CGI `sayhello.py`. Use this code:

```
print('Content-type: text/plain\n')  
  
print('Hello from Head First Python on GAE!')
```

It doesn't get much easier than this...a plain-text message is displayed within your browser whenever this CGI runs.

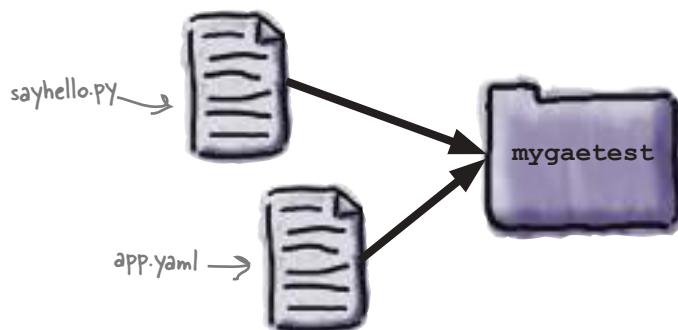
The configuration file *must* be called `app.yaml` and it, too, must be in your webapp's folder. This file tells the Google cloud a little bit about your webapp's runtime environment. Here's a basic configuration file:

```
application: mygaetest  
version: 1  
runtime: python  
api_version: 1  
  
handlers:  
- url: /.*  
  script: sayhello.py
```

The "application" line identifies your webapp and is the same name as your folder.
"runtime" tells GAE that your webapp is written in and will run on Python.
Think of the "handlers" section of the configuration file as a top-level webapp routing mechanism.
The "version" line identifies the current version of your webapp (and usually starts at 1).
The "api_version" indicates the release of GAE you are targeting.
This entry tells GAE to route all requests to your webapp to your "sayhello.py" program.

Do this!

Go ahead and create the folder called `mygaetest` and the two files shown here.





Test DRIVE

Click this button to start your webapp.

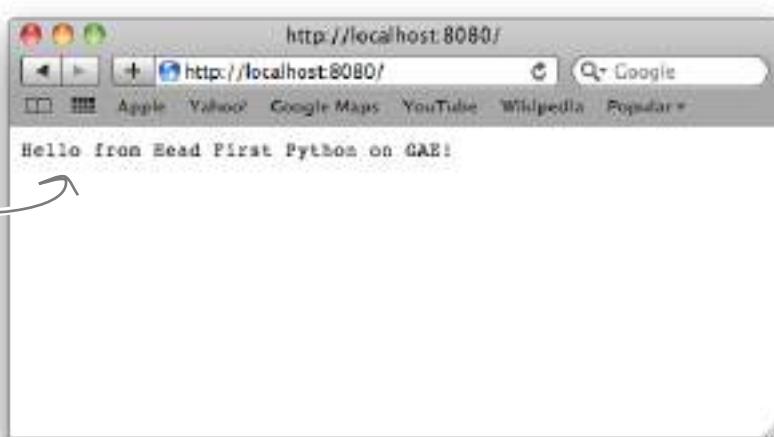
The GAE SDK includes a test web server, so let's use it to take your test GAE webapp for a spin. If you are running on Windows or Mac OS X, fire up the Google App Engine Launcher front end. This tool makes it easy to start, stop, and monitor your webapp. On Linux, you'll need to invoke a command to kick things off. If you are using the GAE Launcher, choose File -> Add Existing Application from the menu system to browse and select your webapp's folder. Also: be sure to edit the Launcher's Preferences to select Python 2.5 as your preferred Python Path.

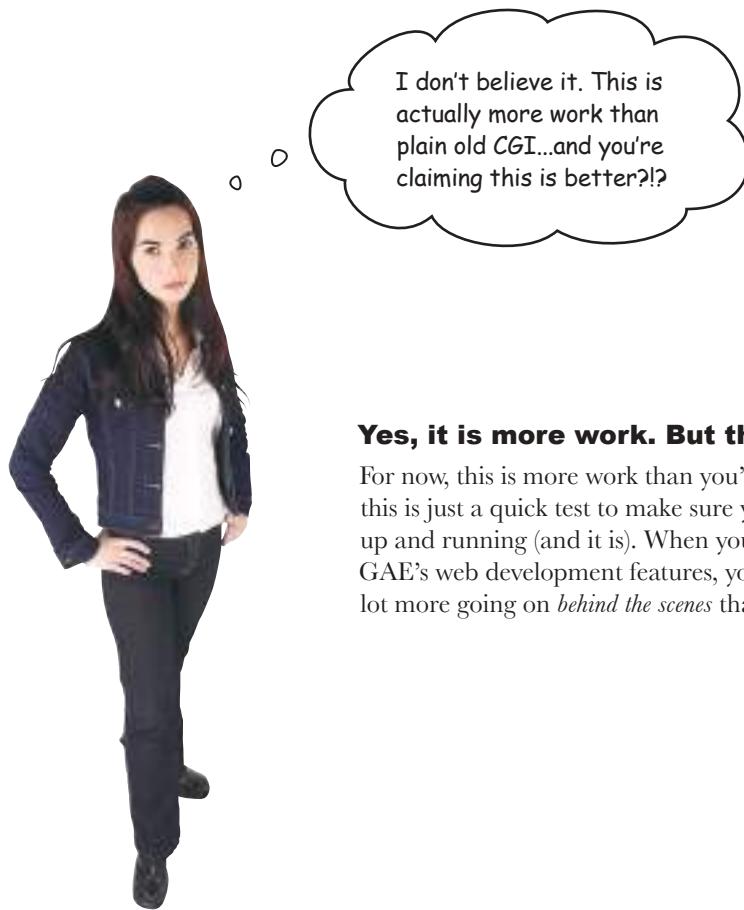
This is how the GAE Launcher looks on Mac OS X...it looks similar on Windows.

There is no graphical front end for Linux, so start your GAE webapp from the command line.

With your webapp running and waiting on port 8080, open your favorite web browser and surf on over to the <http://localhost:8080> web address.

And there it is...
the message from
your test webapp!





Yes, it is more work. But that's about to change.

For now, this is more work than you're used to, but remember that this is just a quick test to make sure your GAE test environment is up and running (and it is). When you start to work with some of GAE's web development features, you'll initially see that there's a lot more going on *behind the scenes* than meets the eye.

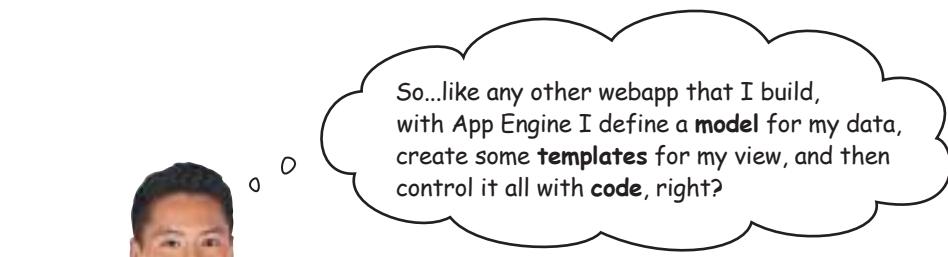
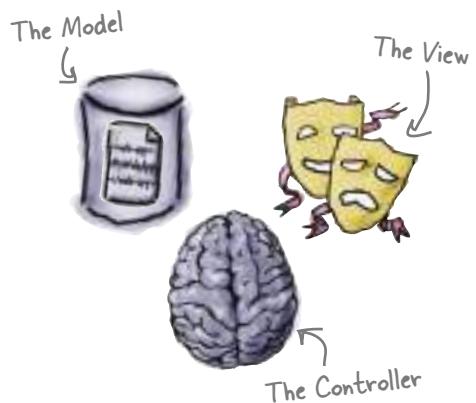
App Engine uses the MVC pattern

Google has built GAE to conform to the familiar Model-View-Controller (MVC) pattern.

Like your webapp from the previous chapter, the model component of a GAE-enabled webapp uses a back-end data storage facility that's known as the **datastore**. This is based on Google's *BigTable* technology, which provides a "NoSQL" API to your data, as well as a SQL-like API using Google's Query Language (GQL).

GAE's views use templates, but unlike the simple string templates from the previous chapter, GAE uses the **templating system** from the Django Project, which is one of Python's leading web framework technologies. In addition to templates, GAE includes Django's forms-building technology.

And, of course, any controller **code** is written in Python and can use the CGI or WSGI standards. Unfortunately, you can't use your `yate` module with GAE, because it is a Python 3 library (and would need to be extensively rewritten to support Python 2). Not to worry: the facilities provided by GAE "out of the box" are more than enough to build *great* webapps.



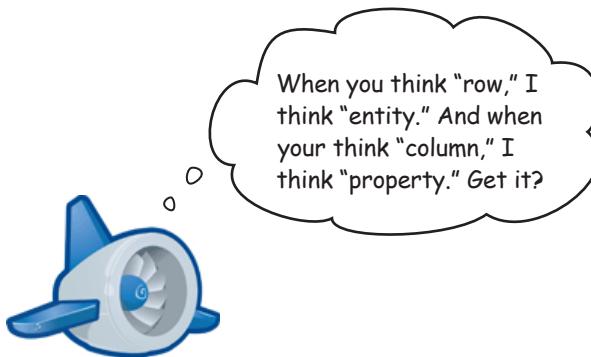
Yes, it's the same process as any other webapp.

Google has worked hard to ensure that the move to App Engine is as painless as possible. If you understand MVC (as you now do), you are well on your way to creating with GAE. It's just a matter of working out how GAE implements each of the MVC components.

Model your data with App Engine

App Engine refers to data items stored within its datastore as *properties*, which are defined within your model code.

Think of properties as a way to define the name and types of data within your database schema: each property is like the column type associated piece of data stored in a row, which App Engine refers to as an *entity*.



As with traditional SQL-based databases, your GAE datastore properties are of a specific, predeclared type. There are *lots* to choose from, for instance:

- **db.StringProperty**: a string of up to 500 characters
- **db.Blob**: a byte string (binary data)
- **db.DateProperty**: a date
- **db.TimeProperty**: a time,
- **db.IntegerProperty**: a 64-bit integer
- **db.UserProperty**: a Google account

For the full list of property types supported, pop on over to <http://code.google.com/appengine/docs/python/datastore/typesandpropertyclasses.html> and take a look.

Here's some sample data from the previous chapter.

This data is stored as a "db.IntegerProperty".

Store this data as "db.StringProperty".

This data is stored as a "db.DateProperty".

id	name	dob
1	James Lee	2002-03-14
2	Sarah Sweeney	2002-06-17
3	Vera Vi	2002-12-25
4	Julie Jones	2002-08-17
5	Sally Sanchez	2002-11-24
6	Mikey McManus	2002-02-24



Pool Puzzle

Your **job** is to take the properties from the pool and place them in the correct place in the class code, which is in a file called `hfwwgDB.py`. Your **goal** is to assign the correct property type to each of the attributes within your `Sighting` class.

Import the "db" module from the GAE extensions.

```
from google.appengine.ext import db
```

Hawaiian Whales Watching Group Kauai Sighting Form				
Name:				
From:				
Date:				
Time:				
Location:				
Fin Type:	Finless	Humpback	Orca	Blue
Whale Type:	Bright	Fin	Orca	Blue
Whale Type:	Nah	Bratty	Horror	
Wave Type:	Fish	Small	Molasses	
	Large	Wormy	Huge	

```
class Sighting(db.Model):
    name = .....  

    email = .....  

    date = .....  

    time = .....  

    location = .....  

    fin_type = .....  

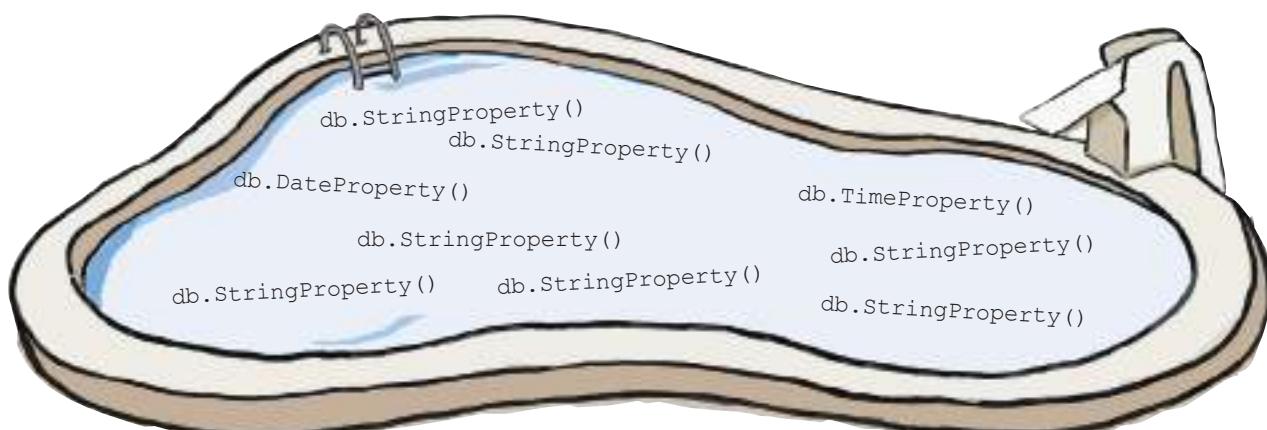
    whale_type = .....  

    blow_type = .....  

    wave_type = .....
```

Each property is assigned to a name.

Create a class called "Sighting" that inherits from the GAE "db.Model" class.





Pool Puzzle Solution

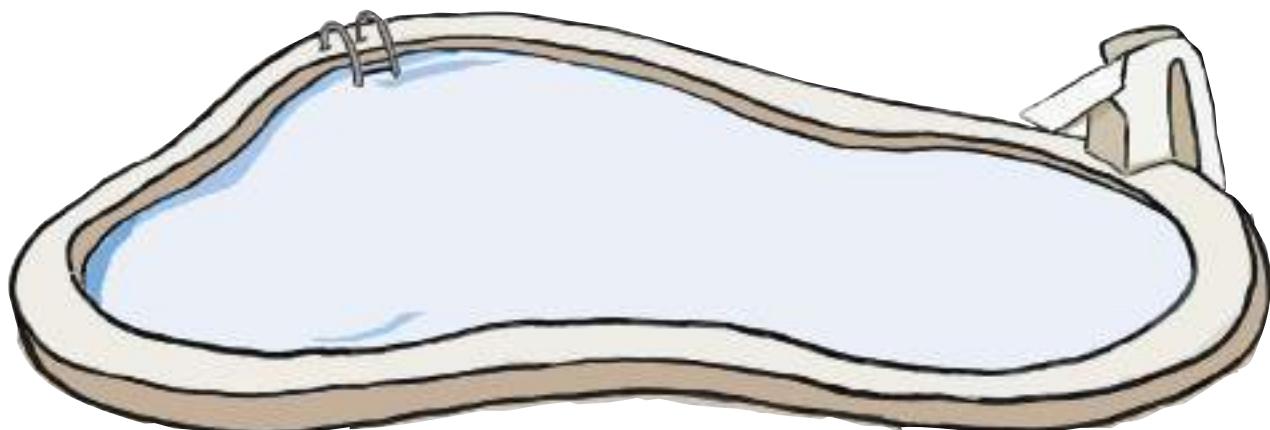
Your **job** was to take the properties from the pool and place them in the correct place in the class code, which is in a file called `hfwwgDB.py`. Your **goal** was to assign the correct property type to each of the attributes within your `Sighting` class.

```
from google.appengine.ext import db
```

```
class Sighting(db.Model):
    name =           db.StringProperty()
    email =          db.StringProperty()
    date =           db.DateProperty()
    time =           db.TimeProperty()
    location =       db.StringProperty()
    fin_type =       db.StringProperty()
    whale_type =     db.StringProperty()
    blow_type =      db.StringProperty()
    wave_type =      db.StringProperty()
```

Hawaiian Islands Whales				
Name	Fin Type	Blow Type	Whale Type	Waved
Humpback	Finned	Orca	Blue	Yes
Minke	Bluntnose	Fin	Orca	No
Sei	Notched	Blow	Minke	No
Fin	Large	Blowing	Blue	No

Everything is a "StringProperty", except the "date" and "time" fields.



What good is a model without a view?

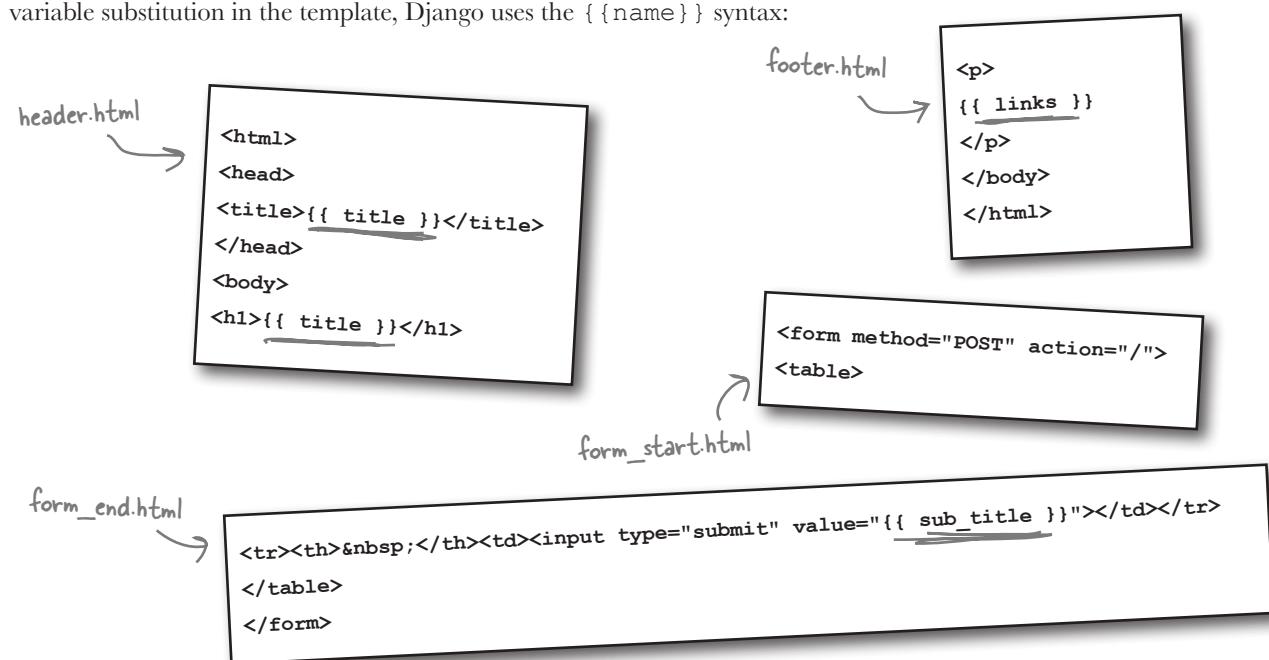
GAE not only lets you *define* the schema for your data, but it also *creates* the entities in the datastore. The first time you go to **put** your data in the datastore, GAE springs to life and makes room for your data. There's no extra work required by you, other than defining your model in code. It's useful to think of GAE as executing something similar to a SQL CREATE command *on the fly* and *as needed*. But how do you get data into the GAE datastore?

The short answer is that *you put it there*, but you first need to get some data from your webapp's user...and to do that, you need a *view*. And views are easy when you use *templates*.

App Engine templates in an instant

Recall that the templating technology built into GAE is based on technology from the Django Project. Django's templating system is more sophisticated than the simple string-based templates used in the previous chapter. Like your templates, Django's templates can substitute data into HTML, but they can also execute *conditional* and *looping* code.

Here are four templates you'll need for your HTWWG webapp. Two of them should be familiar to you: they are adaptions of those used in the previous chapter. The other two are new. Go ahead and grab them from this book's support website. As you can see, rather than using the \$name syntax for variable substitution in the template, Django uses the {{ name }} syntax:



Use templates in App Engine

To use a template, import the template module from `google.appengine.ext.webapp` and call the `template.render()` function. It is useful to assign the output from `template.render()` to a variable, which is called `html` in this code snippet:

```
from google.appengine.ext.webapp import template  
  
html = template.render('templates/header.html', {'title': 'Report a Possible Sighting'})
```

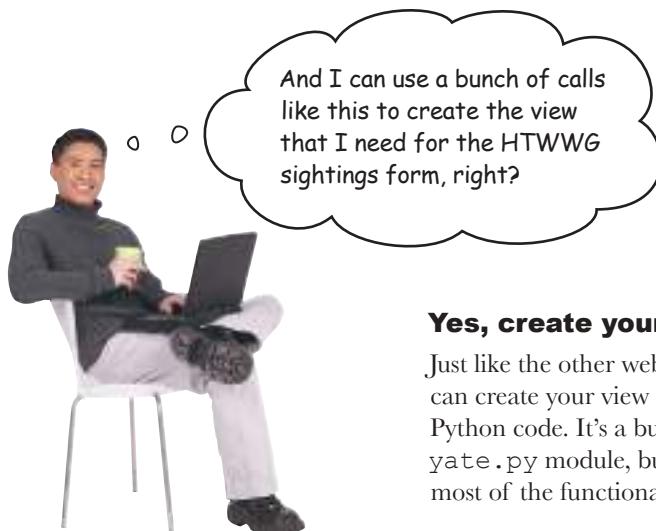
As usual, start with your import.

Call "template.render()..."

...supplying the template name...

...as well as a dictionary that maps values to the named template variable.

This is similar to the mechanism your `yate.py` module uses to parameterize the data displayed within your HTML pages.



Yes, create your view with templates.

Just like the other webapps that you've built, you can create your view in much the same way using Python code. It's a bummer that you can't use your `yate.py` module, but Django's templates provide most of the functionality you need here.

there are no
Dumb Questions

Q: Should I create one big template for my entire web page?

A: You could, if you want. However, if you build up your view from snippets of HTML in templates, you open up the possibility of reusing those HTML snippets in lots of places. For instance, to maintain a consistent look and feel, you can use the same header and footer template on all of your web pages, assuming of course that your header and footer aren't already embedded in an entire web page (which can't be reused).



- 1** Let's write the rest of the code needed to create a view that displays a data entry form for your HFWWG webapp.

In addition to your web page header code (which already exists and is provided for you), you need to write code that starts a new form, displays the form fields, terminates the form with a submit button, and then finishes off the web page. Make use of the templates you've been given and (here's the rub) do it all in *no more than four additional lines of code*.

This code goes into a new program called "hfwwg.py".

```
from google.appengine.ext.webapp import template

html = template.render('templates/header.html', {'title': 'Report a Possible Sighting'})
```

```
html = html +
```

Extend the contents of "html" with
the rest of the HTML you need.

Remember: no
more than 4
lines of code!

- 2** Now that you have attempted to write the code required in *no more than four lines of code*, what problem(s) have you encountered. In the space below, note down any issue(s) you are having.



- 1 You were to write the rest of the code needed to create a view that displays a data entry form for your HFWWG webapp.

In addition to your webpage header code (which already exists and is provided for you), you were to write code with starts a new form, displays the form fields, terminates the form which a submit button, then finishes off the webpage. You were to make use of the templates you've been given and (here's the rub) you had to do it all in *no more than four more lines of code*.

```
from google.appengine.ext.webapp import template

html = template.render('templates/header.html', {'title': 'Report a Possible Sighting'})
```

The "render()" function always expects two arguments. If you don't need the second one, be sure to pass an empty dictionary.

```
html = html + template.render('templates/form_start.html', {})
```

This is an issue,
isn't it?

We need to generate the FORM fields in here...but how?!

```
html = html + template.render('templates/form_end.html', {'sub_title': 'Submit Sighting'})
```

```
html = html + template.render('templates/footer.html', {'links': ''})
```

- 2 Having attempted to write the code required in *no more than four lines of code*, you were to make a note of any issue(s) you encountered.

This is IMPOSSIBLE to do in just four lines of code, because there's no way...

to generate the FORM fields that I need. I can't even use the "do_form()"

function from "yate.py", because that code is not compatible with Python 2.5...

this just sucks!

You may have written something like this...assuming, of course, you haven't thrown your copy of this book out the nearest window in frustration. ☺



Wouldn't it be dreamy if I could avoid hand-coding a <FORM> and generate the HTML markup I need from an existing data model? But I know it's just a fantasy...

Django's form validation framework

Templates aren't the only things that App Engine "borrows" from Django. It also uses its form-generating technology known as the *Form Validation Framework*. Given a data model, GAE can use the framework to generate the HTML needed to display the form's fields within a HTML table. Here's an example GAE model that records a person's essential birth details:

This code is in a file called "birthDB.py".

```
from google.appengine.ext import db

class BirthDetails(db.Model):
    name = db.StringProperty()
    date_of_birth = db.DateProperty()
    time_of_birth = db.TimeProperty()
```

This model is used with Django's framework to generate the HTML markup needed to render the data-entry form. All you need to do is inherit from a GAE-included class called `djangoforms.ModelForm`:

```
from google.appengine.ext.webapp import template
from google.appengine.ext.db import djangoforms
import birthDB

class BirthDetailsForm(djangoforms.ModelForm):
    class Meta:
        model = birthDB.BirthDetails
        ...
html = template.render('templates/header.html', {'title': 'Provide your birth details'})
html = html + template.render('templates/form_start.html', {})
html = html + str(BirthDetailsForm(auto_id=False))
html = html + template.render('templates/form_end.html', {'sub_title': 'Submit Details'})
html = html + template.render('templates/footer.html', {'links': ''})
```

Import the forms library in addition to your GAE data model.

Create a new class by inheriting from the "djangoforms.ModelForm" class, and then link your new class to your data model.

Use your new class to generate your form.

There is some code missing from here...but don't worry: you'll get to it in just a moment. For now, just concentrate on understanding the links between the model, the view code, and the Django form validation framework.

Check your form

The framework generates the HTML you need and produces the following output within your browser.

It's not the prettiest web page ever made, but it works.

Use the View Source menu option within your web browser to inspect the HTML markup generated.

```

<html>
<head>
<title>Provide your birth details</title>
</head>
<body>
<h1>Provide your birth details</h1>
<form method="POST" action="/">
<table>
<tr><th>Name:</th><td><input type="text" name="name" /></td></tr>
<tr><th>Date of birth:</th><td><input type="text" name="date_of_birth" /></td></tr>
<tr><th>Time of birth:</th><td><input type="text" name="time_of_birth" /></td></tr>
<tr><th><br/></th><td><input type="submit" value="Submit Details"></td></tr>
</table>
</form>
<p></p>
</body>
</html>

```

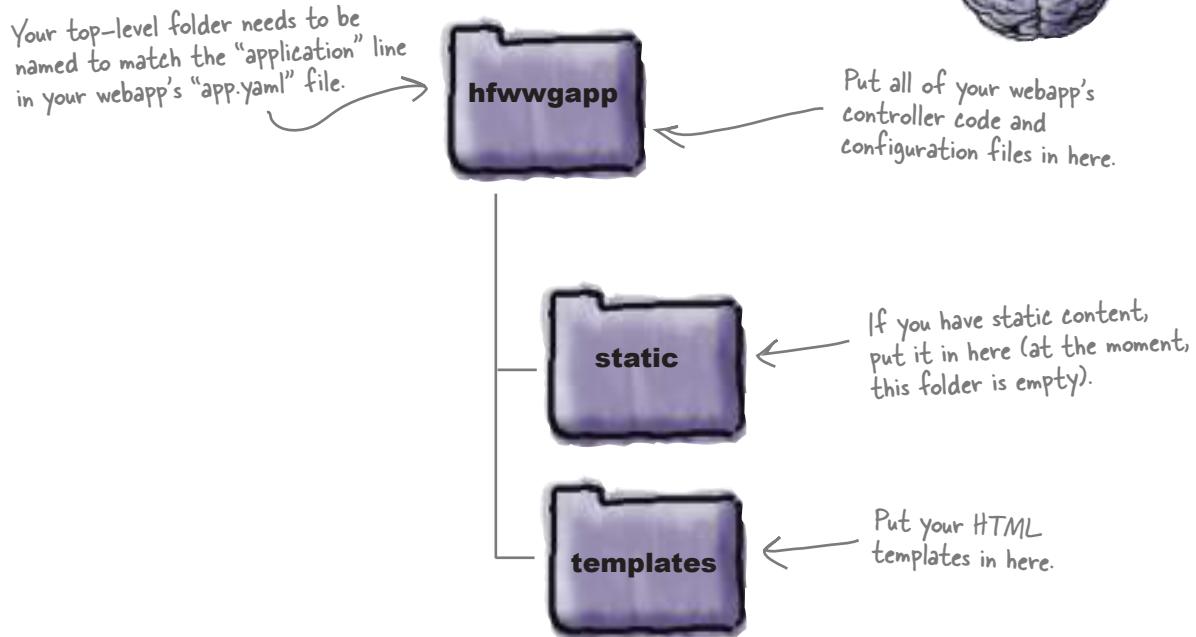
By setting "auto_id" to "False" in your code, the form generator uses your model property names to identify your form's fields.

The Django framework is smart enough to create sensible labels for each of your input fields (based on the names used in your model).

It's time to tie things all together with your controller code.

Controlling your App Engine webapp

Like your other webapps, it makes sense to arrange your webapp **controller** code within a specific folder structure. Here's one suggestion:



As you've seen, any CGI can run on GAE, but to get the most out of Google's technology, you need to code to the WSGI standard. Here's some *boilerplate* code that every WSGI-compatible GAE webapp starts with:

```
Import APP Engine's "webapp" class.           Import a utility that runs your webapp.
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app

class IndexPage(webapp.RequestHandler):
    def get(self):
        pass

This method runs when a GET web request is received by your webapp.

app = webapp.WSGIApplication([(('/.*', IndexPage)], debug=True)

def main():
    run_wsgi_app(app)

Start your webapp.

if __name__ == '__main__':
    main()
```

Create a new "webapp" object for your application.

This class responds to a web request from your web browser.

This is not unlike switching on CGI tracking.

Just use these two lines of code as-is.



App Engine Code Magnets

Let's put everything together. Your model code is already in your `hfwwgDB.py` file. All you need to do is move that file into your webapp's top-level folder. Copy your templates folder in there, too. Your webapp's controller code, in a file called `hfwwg.py`, also needs to exist in your top-level folder. The only problem is that some of the code's all over the floor. Rearrange the magnets to fix things.

```
from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app
from google.appengine.ext import db
from google.appengine.ext.webapp import template
from google.appengine.ext.db import djangoforms
```

All of the imports have survived...so there's no need to rearrange them.

Let's test how well you've been paying attention. There's no guiding lines on the fridge door.

What's missing from in here?

```
html = template.render('templates/header.html', {'title': 'Report a Possible Sighting'})
html = html + template.render('templates/form_start.html', {})

html = html + template.render('templates/form_end.html', {'sub_title': 'Submit Sighting'})
html = html + template.render('templates/footer.html', {'links': ''})

app = webapp.WSGIApplication([(__.'/.*', SightingInputPage)], debug=True)
```

There's only one small change from the boilerplate code in that "IndexPage" is not being linked to.

```
class SightingForm(djangoforms.ModelForm):
```

```
class SightingInputPage(webapp.RequestHandler):
    def get(self):
```

```
    html = html + str(SightingForm())
```

```
    self.response.out.write(html)
```

```
    model = hfwwgDB.Sighting
```

```
    class Meta:
```

```
        import hfwwgDB
```



App Engine Code Magnets Solution

Let's put everything together. Your model code is already in your `hfwwgDB.py` file. You were to move that file into your webapp's top-level folder, as well as copy your templates folder in there, too. Your webapp's controller code, in a file called `hfwwg.py`, also needs to exist in your top-level folder. The only problem is that some of the code's all over the floor. You were to rearrange the magnets to fix things:

```

from google.appengine.ext import webapp
from google.appengine.ext.webapp.util import run_wsgi_app
from google.appengine.ext import db
from google.appengine.ext.webapp import template
from google.appengine.ext.db import djangoforms

import hfwwgDB

```

Import your GAE data model code.

```

class SightingForm(djangoforms.ModelForm):
    class Meta:
        model = hfwwgDB.Sighting

```

Use your model to create a sighting form that inherits from the "django.ModelForm" class.

```

class SightingInputPage(webapp.RequestHandler):
    def get(self):

```

The connected handler class is called "SightingInputPage" and it provides a method called "get" which responds to a GET web request.

```

        html = template.render('templates/header.html', {'title': 'Report a Possible Sighting'})
        html = html + template.render('templates/form_start.html', {})
        html = html + str(SightingForm())

```

Include the generated form in the HTML response.

```

        html = html + template.render('templates/form_end.html', {'sub_title': 'Submit Sighting'})
        html = html + template.render('templates/footer.html', {'links': ''})
        self.response.out.write(html)

```

Did you guess this correctly? You need to send a response back to the waiting web browser and this line of code does just that.

```

app = webapp.WSGIApplication([('/', SightingInputPage)], debug=True)

def main():
    run_wsgi_app(app)

if __name__ == '__main__':
    main()

```



Test DRIVE

It's been a long time coming, but you are now ready to test the first version of your sightings form. If you haven't done so already, create an `app.yaml` file, too. Set the application line to `hfwwg` and the script line to `hfwwg.py`. One final step is to use the Add Existing Application menu option within the GAE Launcher to select your top-level folder as the location of your webapp.

The launcher adds your webapp into its list and assigns it the next available protocol port—in this case, 8081.

And here's your generated HTML form in all its glory.

This is looking good. Let's get a quick opinion from the folks over at the HFWWG.



OK, we get it. Web design is not your thing.

Not to worry, you know all about code reuse, right? So, let's reuse *someone else's* **cascading style sheets** (CSS) to help with the "look" of your generated HTML form.

But who can you "borrow" from and not lose sleep feeling guilty over it?

As luck would have it, the authors of *Head First HTML with CSS & XHTML* created a bunch of stylesheets for their web pages and have made them available to you. Grab a slightly amended copy of some of their great stylesheets from this book's support website. When you unzip the archive, a folder called static appears: pop this entire folder into your webapp's top-level folder.

There's a file in static called favicon.ico. Move it into your top-level folder.

Improve the look of your form

To integrate the stylesheets into your webapp, add two link tags to your header.html template within your templates folder. Here's what the tags need to look like:

```
<link type="text/css" rel="stylesheet" href="/static/hfwwg.css" />
<link type="text/css" rel="stylesheet" href="/static/styledform.css" />
```

Add these two lines to the top of your "header.html" template.

GAE is smart enough to *optimize* the delivery of static content—that is, content that does *not* need to be generated by code. Your CSS files are static and are in your static folder. All you need to do is tell GAE about them to enable optimization. Do this by adding the following lines to the handlers section of your app.yaml file:

Provide the URL location for your static content.

```
- url: /static
  static_dir: static
```

Switch on the optimization.



Test DRIVE

With your stylesheets in place and your app.yaml file amended, ask your browser to **reload** your form.

Looking good. →

Report a Possible Sighting

Name:

Email:

Date:

Time:

Location:

Fin type:

Whale type:

Blow type:

Wave type:

Submit Sighting

A little style goes
a long way...that's
looking great!



Restrict input by providing options

At the moment, your form accepts anything in the Fin, Whale, Blow, and Wave input areas. The paper form restricts the data that can be provided for each of these values. Your HTML form should, too.



Anything you can do to cut down on input errors is a good thing. As the youngest member of the group, I was "volunteered" to work on data clean-up duties...

Finless	Humpback	Sperm	Sperm
Blow Type: Humpback	Orca	Blue	Gray
Blow Type: Fin			
Blow Type: Bushy			
Blow Type: Dense			

Wave Type:	Flat	Small	Moderate
Ledge			
Breaking			
High			

Providing a list of choices restricts what users can input.

Instead of using HTML's INPUT tag for all of your form fields, you can use the SELECT/OPTION tag pairing to *restrict* what's accepted as valid data for any of the fields on your form. To do this, you'll need more HTML markup. That's the *bad news*.

The *good news* is that the form validation framework can generate the HTML markup you need for you. All you have to provide is the list of data items to use as an argument called `choices` when defining your property in your model code. You can also indicate when multiple lines of input are acceptable using the `multiline` argument to a property.

Apply these changes to your model code in the `hfwwgDB.py` file.

This naming convention helps identify these lists as containing constant values.

Define your lists of values near the top of your code.

```
_FINS = ['Falcate', 'Triangular', 'Rounded']
_WHALES = ['Humpback', 'Orca', 'Blue', 'Killer', 'Beluga', 'Fin', 'Gray', 'Sperm']
_BLOWS = ['Tall', 'Bushy', 'Dense']
_WAVES = ['Flat', 'Small', 'Moderate', 'Large', 'Breaking', 'High']

location = db.StringProperty(multiline=True)
fin_type = db.StringProperty(choices=_FINS)
whale_type = db.StringProperty(choices=_WHALES)
blow_type = db.StringProperty(choices=_BLOWS)
wave_type = db.StringProperty(choices=_WAVES)
```

Switch on multiple-line input.

Use your lists of values when defining your properties.



Test DRIVE

With these changes applied to your model code, refresh your web browser once more.

Your form is not only looking good, but it's more functional, too.

The "location" field is now displayed over multiple lines.

Name: Eagle-eyed Jack Jones

Email: eej@hfwwg.org

Date: Jan 6, 2011

Time: 8:02am

Location: Just off the coast – really close to shore.

Fin type: Triangular

Whale type: Fin

Blow type: Bushy

Wave type: Breaking

Submit Sighting

Each of the "type" fields now have drop-down selection menus associated with them.

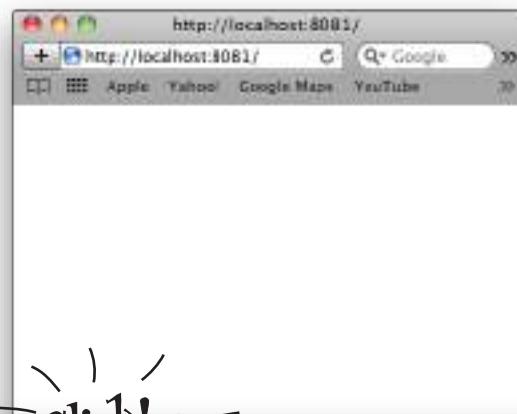
Your form now looks great! Go ahead and enter some test data, and then press the Submit Sighting button.

What happens?

Meet the “blank screen of death”

Submitting your form’s data to the GAE web server produces a *blank screen*.

Whoops...that's not
exactly user-friendly.



To work out what happened (or what didn’t happen), you need to look at the logging information for your GAE webapp.

If you are running GAE on Linux, your logging messages are displayed on screen. If you are on Windows or Mac OS X, click the **Logs** button within the Launcher to open up the **Log Console** for your webapp.



Your request resulted in a 405 status code from the web server. According to the official HTTP RFC standards document, 405 stands for:

“Method Not Allowed. The method specified in the Request-Line is not allowed for the resource identified by the Request-URI. The response MUST include an Allow header containing a list of valid methods for the requested resource”.

Your last web request
resulted in a 405.

Ummm...that's as clear
as mud, isn't it?

Process the POST within your webapp

What the 405 status code actually tells you is that posted data arrived at your webapp intact, but that your webapp *does not* have any way of processing it. There's a method missing.

Take a quick look back at your code: the only method currently defined is called `get()`. This method is invoked whenever a GET web request arrives at your webapp and, as you know, it displays your sightings form.

In order to process posted data, you need to define *another* method. Specifically, you need to add a new method called `post()` to your `SightingInputPage` class.

App Engine handles requests as well as responses

Your `get()` method produces your HTML form and returns a web response to the waiting web browser using the `self.response` object and by invoking the `out.write()` method on it.

In addition to helping you with your web responses, GAE also helps you process your web requests using the `self.request` object. Here are a few lines of code that displays all of the data posted to your web server:



```
Define a new
method called
"post".
```

```
Don't forget to use "self" with
all your methods.
```

```
The "arguments()" method returns a list
of the field names used on your form.
```

```
def post(self):
    for field in self.request.arguments():
        self.response.out.write(field)
        self.response.out.write(': ')
        self.response.out.write(self.request.get(field))
        self.response.out.write('<br />')
```

```
The "get()" method returns the value associated
with the provided form field name.
```

So...if you know the *name* of your form field, you can access its value from within your webapp using the `self.request.get()` method.

But what do you do with the data once you have it?

Put your data in the datastore

Your data is sent to your webapp by GAE and you can use the `self.request.get()` method to access each input field value by name. Recall the `BirthDetails` model from earlier in this chapter:

This code is in a file called "birthDB.py".

```
from google.appengine.ext import db

class BirthDetails(db.Model):
    name = db.StringProperty()
    date_of_birth = db.DateProperty()
    time_of_birth = db.TimeProperty()
```

Assume that an HTML form has sent data to your webapp. The data is destined to be stored in the GAE datastore. Here's some code to do the heavy lifting:

Generate a HTML response to say "thanks."

Send your response to the waiting web browser.

```
def post(self):
    new_birth = birthDB.BirthDetails() Create a new "BirthDetails" object to hold your data.

    new_birth.name = self.request.get('name')
    new_birth.date = self.request.get('date_of_birth')
    new_birth.time = self.request.get('time_of_birth') Get each of the form's data values and assign them to your new object's attributes.

    new_birth.put() Put (save) your data to the GAE datastore.

    html = template.render('templates/header.html', {'title': 'Thank you!'})
    html = html + "<p>Thank you for providing your birth details.</p>"
    html = html + template.render('templates/footer.html',
                                  {'links': 'Enter <a href="/">another birth</a>. '})
    self.response.out.write(html)
```

There's nothing to it: **create** a new object from your data model, **get** the data from your HTML form, **assign** it to the object's attributes, and then use the `put()` method to **save** your data in the datastore.



Based on what you know about how to put your HTML form's data into the GAE datastore, create the code for the `post()` method that your webapp now needs. Some of the code has been done for you already. You are to provide the rest.

```
def post(self):
```

Put your code
here. —

```
html = template.render('templates/header.html',
                      {'title': 'Thank you!'})

html = html + "<p>Thank you for providing your sighting data.</p>"

html = html + template.render('templates/footer.html',
                             {'links': 'Enter <a href="/">another sighting</a>.'})

self.response.out.write(html)
```



Based on what you know about how to put your HTML form's data into the GAE datastore, you were to create the code for the `post()` method that your webapp now needs. Some of the code has been done for you already. You were to provide the rest.

```

def post(self):
    new_sighting = hfwwgDB.Sighting()           Create a new "Sighting" object.

    new_sighting.name = self.request.get('name')
    new_sighting.email = self.request.get('email')
    new_sighting.date = self.request.get('date')
    new_sighting.time = self.request.get('time')
    new_sighting.location = self.request.get('location')
    new_sighting.fin_type = self.request.get('fin_type')
    new_sighting.whale_type = self.request.get('whale_type')
    new_sighting.blow_type = self.request.get('blow_type')
    new_sighting.wave_type = self.request.get('wave_type')           For each of the data
                                                                    values received from
                                                                    the HTML form,
                                                                    assign them to the
                                                                    attributes of the
                                                                    newly created object.

    new_sighting.put()           Store your populated object in the GAE
                                datastore.

html = template.render('templates/header.html',
                      {'title': 'Thank you!'})
html = html + "<p>Thank you for providing your sighting data.</p>"
html = html + template.render('templates/footer.html',
                           {'links': 'Enter <a href="/">another sighting</a>.'})

self.response.out.write(html)

```



Test DRIVE

Add your `post()` code to your webapp (within the `hfwwg.py` file) and press the **Back** button on your web browser. Click the Submit Sighting button once more and see what happens this time.

Here's your form with the data waiting to be submitted.



Report a Possible Sighting

Name:	Eagle-eyed Jack Jones
Email:	ej@hfwwg.org
Date:	Jan 6, 2011
Time:	8:02am

But when you click the button, something bad has happened...your webapp has crashed.



```
Loc http://localhost:8081/
File Apple Yahoo! Google Maps YouTube Wikipedia Popularity
Whale
Blowhole
Wave

Traceback (most recent call last):
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/GoogleAppEngine-d
    handler.post(*groups)
  File "/Users/berryp/HeadFirstPython/chapter10/hfwwg/hfwwg.py", line 27, in post
    new_sighting.date = self.request.get('date')
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/GoogleAppEngine-d
    value = self.validate(value)
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/GoogleAppEngine-d
    value = super(DateProperty, self).validate(value)
  File "/Applications/GoogleAppEngineLauncher.app/Contents/Resources/GoogleAppEngine-d
    (self.name, self.data_type._name_))
BadValueError: Property date must be a date
```

It looks like you might have a problem with the format of your date property, doesn't it?

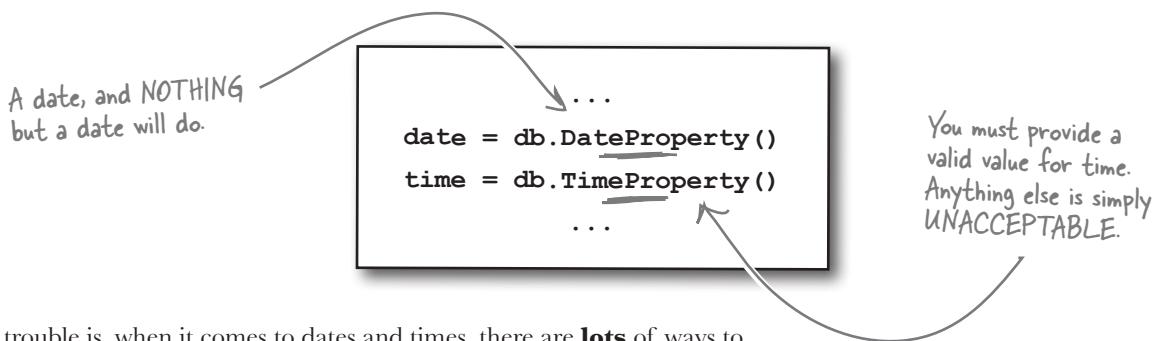
Phooey...that's disappointing, isn't it?

At the very least, you were expecting the data from the form to make it into the datastore...but something has stopped this from happening. What do you think is the problem?

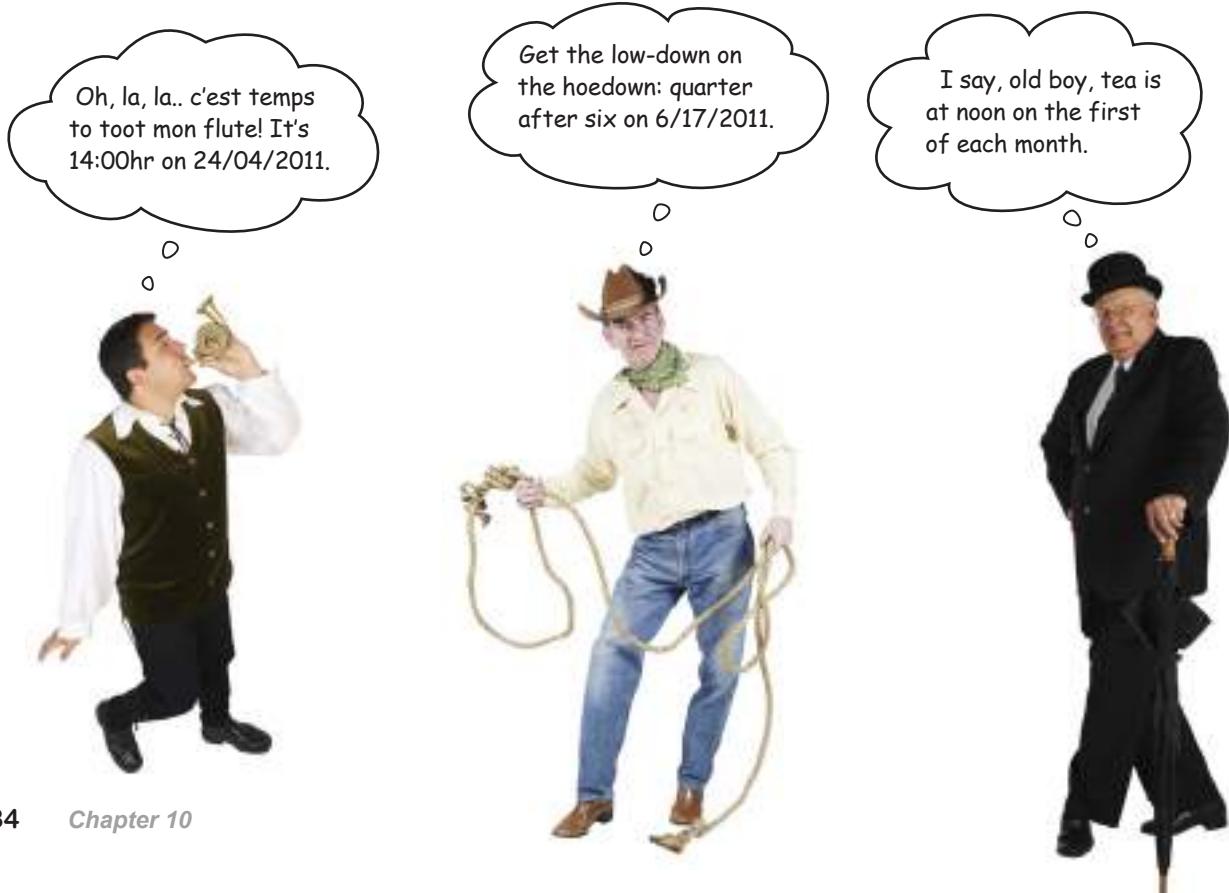
Don't break the "robustness principle"

The **Robustness Principle** states: “Be conservative in what you send; be liberal in what you accept.” In other words, don’t be too picky when *requesting* data of a certain type from your users, but when *providing* data, give ‘em exactly what they need.

If you make it *too hard* for your users to enter data into your system, things will likely break. For instance, within your model code, consider how date and time are defined:



The trouble is, when it comes to dates and times, there are **lots** of ways to specify values.



Accept almost any date and time

If you are going to insist on asking your users to provide a properly formatted date and time, you'll need to do one of two things:

- Specify in detail the **format** in which you expect the data.
- **Convert** the entered data into a format with which you can work.

Both approaches have *problems*.

For example, if you are too picky in requesting a date in a particular format, you'll slow down your user and might end up picking a date format that is foreign to them, resulting in confusion.

If you try to convert *any* date or time entered into a common format that the datastore understands, you'll be biting off more than you can chew. As an example of the complexity that can occur, how do you know if your user entered a date in mm/dd/yyyy or dd/mm/yyyy format? (You don't.)

There is a third option

If your application doesn't require *exact* dates and times, *don't require them of your user*.

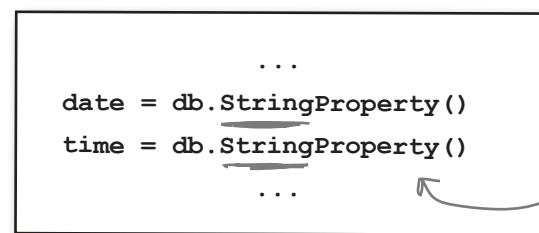
With your sightings webapp, the date and time can be *free-format fields* that accept any value (in any format). What's important is *the recording of the sighting*, not the exact date/time it occurred.

Of course, other webapps might not be as fast and loose with dates and times. When that's the case, you'll need to revert one of the options discussed earlier on this page and do the best you can.

Use "db.StringProperty()" for dates and times

If you relax the datatype restrictions on the date and time fields, not only do you make it easier on your user, but you also make it easier on you.

For the sightings webapp, the solution is to change the property type for date and time within the hfwwgDB.py file from what they currently are to db.StringProperty().



It's a small change,
but it'll make all the
difference.

Let's see what difference this change makes.



Test DRIVE

Change the types of date and time within `htwwgDB.py` to `db.StringProperty()`, being sure to save the file once you've made your edit. Click Back in your web browser and submit your sightings data once more.

OK, folks...

let's try
this again.

Success! It appears to have worked this time.

Let's enter another sighting, just to be sure.

By relaxing the restrictions you placed on the types of data you'll accept, your webapp now appears to be working fine. Go ahead and enter a few sightings by clicking on the link on your thank-you page and entering more data.

With a few sightings entered, let's use App Engine's included **developer console** to confirm that the sightings are in the datastore.

To access the console, enter `http://localhost:8081/_ah/admin` into your web browser's location bar and click on the **List Entities** button to see your data.

In addition to viewing your existing data in the datastore, you can use the console to enter new test data.

Key	ID	Key Name	Name type	Date	email	Pin type	Location	name	time	name type	while type
ap0v2ndl	0	Batty	Bat	Jan 5, 2011	batty@hwwwg.org	Triangular	Just off the coast - really close to shore	Eagle-eyed	0:00m	Breaking	Fish
ap0v2ndl	1	Dennis	Diver	Wed, Jan 12, 2011	danny@hwwwg.org	Rounded	Off a lighthouse Rock, looking north/west	Mary Kitey	0:30m	Moderate	Diving
ap0v2ndl	2	Jim	Fisherman	February 20, 2011	jimmy@ocean.com	Circular	Looking west at Smuggler's Cove	James Page	8:00m	Moderate	Diving
ap0v2ndl	3	Muffy	Flounder	20 March 2011	muffy@ocean.com	Rounded	Mary Mountain Bay	Goliath	4:00p	Flat	Humpback

There's all the data you entered, which is in a slightly different order than what you might expect. But it's all in there. (App Engine stores your properties in alphabetical order, by name.)

App Engine has assigned a "Key" and an "ID" to each of your entities, which comes in handy when you need to uniquely identify a sighting.

Your GAE webapp is now ready for prime time.

Before you deploy it to Google's cloud infrastructure, let's run it by the folk at HFWWG to see if they are happy for their webapp to "go live."

It looks like you're not quite done yet



Is this a **big** change?

You would imagine that it would be. You'll have to create a new entity to hold your registered user login information, and you'll also need *another* form to ask users to provide their registration data (which you'll need to store in the datastore). With that in place, you'll need *yet another* form to ask your users to log in, and then you'll have to come up with a mechanism to *restrict* only registered and logged-in users to view your webapp's pages, assuming you can come up with something robust that will work...?

Or...as this is GAE, you could just switch on authorization.

Sometimes, the tiniest change can make all the difference...

The engineers at Google designed App Engine to deploy on Google's cloud infrastructure. As such, they decided to allow webapps running on GAE to access the *Google Accounts* system.

By switching on **authorization**, you can require users of your webapp to log into their Google account *before* they see your webapp's pages. If a user tries to access your webapp and he isn't not logged in, GAE redirects to the Google Accounts login and registration page. Then, after a successful login, GAE returns the user to your waiting webapp. *How cool is that?*

To switch on authorization, make one small change to your app.yaml file:

```
application: hfwwgapp
version: 1
runtime: python
api_version: 1

handlers:
- url: /static
  static_dir: static

- url: /.*
  script: hfwwg.py
  login: required
```

That's all there is to it.

Now, when you try to access your webapp, you are asked to log in *before* proceeding.



Capture your user's Google ID, too

Now that your webapp requires your users to log in, let's arrange to capture the user login information as part of the sighting.

Start by adding the following property to your entity's list of attributes in your `hfwwgDB.py` file. Add it right after the `wave_type` property.

Create a new
attribute in your
"Sighting" class...

```
which_user = db.UserProperty()
```

...and set its
property type.

Let's ensure that Django's form validation framework *excludes* this new attribute when generating your HTML form. Within your `hfwwg.py` file, change your `SightingForm` class to look like this:

Make sure Django
doesn't include
the new attribute
in your generated
form.

```
class SightingForm(djangoforms.ModelForm):  
    class Meta:  
        model = hfwwgDB.Sighting  
        exclude = ['which_user']
```

Staying within your `hfwwg.py` file, add another import statement near the top of your program:

```
from google.appengine.api import users
```

Import GAE's
Google Accounts
API.

In your `post()` method, right before you `put` your new sighting to the datastore, add this line of code:

```
new_sighting.which_user = users.get_current_user()
```

When you put your
data to the datastore,
this code includes
the Google ID of the
currently logged-in user.

Every time a user adds a sighting to the datastore, GAE ensures that the user's Google Account ID is saved, too. This extra identification information allows the HFWWG to track exactly who reported which sighting, and should (hopefully) cut down on the amount of spam your webapp might attract.

All that's left to do is to deploy your webapp to Google's cloud.

Deploy your webapp to Google's cloud

With your webapp developed and tested locally, you are now ready to deploy to the Google cloud. This is a two-step process: *register* and *upload*.

To register your webapp on the Google cloud, click the **Dashboard** button on the GAE Launcher.

The “Dashboard” button opens your web browser and takes you to the GAE “My Applications” page (after you sign in with your Google ID).

On Linux, use the
“appcfg.py” script to
deploy.

The screenshot shows the Google App Engine Launcher window at the top, displaying a list of registered applications: 'mygreetest' and 'helloworld'. The 'mygreetest' application is selected. Below the launcher is a web browser window titled 'Create an Application' at the URL <http://appengine.google.com/start/craeapp>. The browser form has fields for 'Application Identifier' (containing '.appspot.com') and 'Check Availability', and an 'Application Title' field (containing 'helloworld'). A note below the title says 'Don't use "helloworld" as that's already taken.' To the right of the browser window, handwritten-style annotations provide instructions: 'Enter your webapp's name in the box, then click on the "Check Availability" button.' with an arrow pointing to the 'Check Availability' button; 'Optional, enter a title for your webapp, and then click the "Create Application" button.' with an arrow pointing to the 'Create Application' button; and 'Don't use "helloworld" as that's already taken.' with an arrow pointing to the note about the taken name.

Assuming all went according to plan and GAE confirmed that your application has been created, all that's left to do is to *deploy*. Return to the GAE Launcher and click on the **Deploy** button. The console displays a bunch of status message while the deployment progresses. If all is well, you'll be told that “appcfg.py has finished with exit code 0”.

Your GAE webapp is now ready to run on Google's cloud.



Test Drive, on Google

Let's take your webapp for a spin on Google's cloud. Open your web browser and surf to a web address that starts with your webapp's name and ends in `.appspot.com`. For the HFWWG webapp, the web address is `http://hfwwgapp.appspot.com`. When you first attempt to go to their webapp, App Engine redirects you to the Google login page.

The screenshot shows a web browser window with the following details:

- Title Bar:** Google Accounts
- Address Bar:** google.com https://www.google.com
- Toolbar:** Back, Forward, Stop, Home, Refresh, Search, etc.
- Page Content:**
 - Section Header:** Google accounts
 - Text:** HeadFirst Whale Watching Group uses Google Accounts for Sign In.
 - Note:** Google is not associated with the contents of HeadFirst Whale Watching Group or its owners. If you sign in, Google will share your email address with HeadFirst Whale Watching Group, but not your password or any other personal information.
 - Note:** HeadFirst Whale Watching Group may use your email address to personalise your experience on their website.
 - Text:** Provide your Google ID and password, or sign up for a new Google account (if you don't have one already).
- Sign-in Form:** A box on the right contains fields for Email and Password, a Stay signed in checkbox, and a Sign in button. It also includes links for "Can't access your account?" and "Forgot your password?".
- Create Account:** A box below the sign-in form contains the text "Don't have a Google Account? Create an account now".
- Page Bottom:** ©2010 Google - [Google Home](#) - [Terms of Service](#) - [Privacy Policy](#) - [Help](#)
- Bottom Left:** Done

After a successful login, your sighting form appears. Go ahead and enter some test data:

The sighting form served → from the Google cloud is EXACTLY the same as the form served by your test server.

Return to the <http://appengine.google.com> site to log into the console. The UI is a little different than the test console, but you can use the Datastore Viewer to confirm that your data has been stored correctly.

Click on this link to see your data as stored in the Google cloud.

ID	Name	Blow_type	Date	Email	Fin_type	Location	name	time	wave_type	whale_type	which_user
1	jonathan.baldwin	Tall	Mar 3, 2011	j@jasp.com	Triangular	At Bluffie's Beach, about 50m from shore.	jonathan.baldwin	11:59am	Moderate	Gray	http://python

Your HFWWG webapp is deployed!



This is professional work.

You've built a *great data-entry webapp* and *deployed* it on Google's cloud. No matter how busy things get, whether there's a handful of sightings per day or tens of thousands, *your webapp can handle the load*, thanks to Google's App Engine. And, best of all, the cash-strapped HFWWG doesn't pay a penny until their sightings activity reaches the level of *millions* of sightings per month!

Did you even notice that you wrote all of your code using Python 2.5?



Your Python Toolbox

You've got Chapter 10 under your belt and you've added more great Python technology to your ever-expanding Python toolbox.

App Engine Lingo

- “Datastore” – the data repository used by Google App Engine to permanently store your data.
- “Entity” – the name used for a “row of data”.
- “Property” – the name used for a “data value”.

- Every App Engine webapp must have a configuration file called `app.yaml`.
- Use the *GAE Launcher* to start, stop, monitor, test, upload, and deploy your webapps.
- App Engine's templating technology is based on the one used in the Django Project.
- App Engine can also use Django's Form Validation Framework.
- Use the `self.response` object to construct a GAE web response.
- Use the `self.request` object to access form data within a GAE webapp.
- When responding to a GET request, implement the required functionality in a `get()` method.
- When responding to a POST request, implement the required functionality in a `post()` method.
- Store data in the App Engine datastore using the `put()` method.

11 dealing with complexity



Data wrangling



Once I build up a head of steam, it's not all that hard to keep on running, and running, and running...



It's great when you can apply Python to a specific domain area.

Whether it's *web development*, *database management*, or *mobile apps*, Python helps you **get the job done** by *not* getting in the way of you coding your solution. And then there's the other types of problems: the ones you can't categorize or attach to a domain. Problems that are in themselves so *unique* you have to look at them in a different, highly specific way. Creating **bespoke** software solutions to these type of problems is an area where Python *excels*. In this, your final chapter, you'll *stretch* your Python skills to the limit and solve problems along the way.

What's a good time goal for the next race?

The Head First Marathon Club has spent years collecting and collating data on their distance runners. Over time, this data has helped the club produce a large spreadsheet of *pace data* that helps their runners predict their performance over various distances. The spreadsheet is *huge* and runs to 50 columns of tightly packed data.

Let's take a look at the club's data, as well as the way the runners and their coach use it.

The runner selects a distance, say 15K, and we time her over the length of her run.



Here's a portion of the Marathon Club's spreadsheet data.

	A	B	C	D	E	F	G
1	V02	84.8	82.9	81.1	79.3	77.5	75.8
2	2mi	8:00	8:10	8:21	8:33	8:44	8:56
3	5k	12:49	13:06	13:24	13:42	14:00	14:19
4	5mi	21:10	21:48	22:17	22:47	23:18	23:50
5	10K	26:54	27:30	28:08	28:45	29:24	30:04
6	15K	41:31	42:27	43:24	44:23	45:23	46:24
7	10mi	44:46	45:46	46:48	47:51	48:59	50:02
8	20K	56:29	57:45	59:03	1:00:23	1:01:45	1:03:08
9	13.1mi	59:49	1:01:09	1:02:32	1:03:56	1:05:21	1:06:51
10	25K	1:11:43	1:13:20	1:14:59	1:16:40	1:18:24	1:20:10
11	30K	1:27:10	1:29:08	1:31:08	1:33:11	1:35:17	1:37:26
12	Marathon	2:05:34	2:08:24	2:11:17	2:14:15	2:17:16	2:20:21
13							

The timed distance is 15km.

The predicted marathon goal.



G	H	I	J	K	L	M	N	O
74.2	72.5	70.9	69.4	67.9	66.4	64.9	63.5	62.1
9:08	9:20	9:33	9:46	9:59	10:13	10:26	10:41	10:56
14:38	14:58	15:18	15:39	16:00	16:22	16:44	17:06	17:28
24:22	24:55	25:28	26:03	26:38	27:14	27:51	28:28	29:05
30:45	31:26	32:09	32:52	33:36	34:22	35:08	35:56	36:43
47:27	48:31	49:36	50:43	51:52	53:02	54:14	55:27	56:40
51:09	52:18	53:29	54:41	55:55	57:11	58:28	59:47	60:54
1:04:33	1:06:00	1:07:29	1:09:01	1:10:34	1:12:09	1:13:46	1:15:26	1:17:05
1:08:21	1:09:53	1:11:28	1:13:04	1:14:43	1:16:24	1:18:07	1:19:52	1:21:37
1:21:58	1:23:49	1:25:49	1:27:37	1:29:36	1:31:37	1:33:40	1:35:47	1:37:54
1:39:37	1:41:52	1:44:09	1:46:30	1:48:54	1:51:21	1:53:51	1:56:25	1:58:55
2:23:31	2:26:44	2:30:02	2:33:25	2:36:52	2:40:24	2:44:00	2:47:42	2:51:15

So...what's the problem?



Not to mention: forgetting the sheets, keeping the sheets up to date, and having to flip back and forth through the sheets looking for a closest match.

Of course, word of your newly acquired Python programming skills is getting around, especially among the running crowd. Ideally, the Marathon Club needs an Android app that can be loaded onto a bunch of phones and carried in each coach's pocket. The app needs to automate the lookup and distance predictions.

Are you up to the challenge? Do you think you can help?

Start with the data

For now, let's not worry about creating the Android app; you'll get to that soon enough. Instead, let's solve the central **data wrangling problem** and then, when you have a working solution, we'll worry about *porting* your solution to Android. We'll begin by getting the data into a format you can easily process with Python.

Most spreadsheet programs can export data to the widely used CSV format. The club has done this for you and created a file called `PaceData.csv`, which includes the data for each of the rows from the original the spreadsheet.

Here's a sample of the raw data from the start of the CSV:

The first line of data is the column headings from the spreadsheet. They look like numbers but are actually headings that represent estimated maximal oxygen consumption (or V02 Max, in ml/kg-min) for the race times in each column. Because they have no affect on the timing data, we'll just treat them as headings.

```
v02,84.8,82.9,81.1,79.3,77.5,75.8,74.2,72.5,70.9,69.4,67.9,66.4,64.9,63.5,62.1,60.7,59.4,58.1,56.8,55.  
2mi,8:00,8:10,8:21,8:33,8:44,8:56,9:08,9:20,9:33,9:46,9:59,10:13,10:26,10:41,10:55,11:10,11:25,11:40,1  
5k,12:49,13:06,13:24,13:42,14:00,14:19,14:38,14:58,15:18,15:39,16:00,16:22,16:44,17:06,17:30,17:53,18:
```

The first value on each of the rest of the lines is the timed distance or row label.

The rest of each line is a list of recorded run times.



Sharpen your pencil

You somehow have to model the data from the CSV file in your Python program. Can you think of a data structure that might help here? Justify your selection.

.....
.....
.....
.....

Do this!

Grab a copy of `PaceData.csv` from this book's support website.



You somehow have to model the data from the CSV file in your Python program. You were to think of a data structure that might help here? You were also to justify your selection.

The list of headings can be stored in a LIST.

Ummmm...

there's lots
to think
about here.

The list of times from each row can also be stored in a LIST, but they
also need to be associated with the headings in the very first row of
data, so maybe a DICTIONARY is what's needed here?

Maybe it's some COMBINATION of the two?!?

Take another look at the data

The first row of data in the CSV file is the **column headings**, with the very first value on the line, the v02 string, being redundant (it won't ever be used in this version of the app). The rest of the first line's data are headings associated with the time values in each of the columns.

Of course, the data in the columns is *also* associated with each row, which is identified by a **row label** in the first column, such as 2mi, 5k, and so on.

Let's look at the data in the CSV file again, which has been reformatted to help highlight the associations.

v02,	84.8,	82.9,	81.1,	79.3,	77.5,	75.8,	74.2
2mi,	8:00,	8:10,	8:20,	8:30,	8:40,	8:50,	9:00
5k,	12:49,	13:06,	13:24,	13:42,	14:00,	14:19,	14:38

This string is redundant.

The column headings are on the first line.

The row labels are unique to each row of times.

The times on each row are associated with their row label, but ALSO with a column heading.

But can we capture all these associations in code?



Marathon Magnets

Here's some code that reads the raw data from the CSV data file. The column headings from the first line are loaded into a list called `column_headings`. The rest of the data (all the rows of times) are loaded into a dictionary called `row_data`, with each row of data keyed with the **row label string** from the start of each line. Of course, as luck would have it, someone was cleaning the fridge door, and they've left a bunch of magnets on the floor. See if you can arrange the magnets into their correct order.

What needs
to go here?



```
with open('PaceData.csv') as paces:
```

Process the
"column_headings"
list here.

Process
"row_data"
here.

```
num_cols = len(column_headings)
print(num_cols, end=' -> ')
print(column_headings)

num_2mi = len(row_data['2mi'])
print(num_2mi, end=' -> ')
print(row_data['2mi'])

num_Marathon = len(row_data['Marathon'])
print(num_Marathon, end=' -> ')
print(row_data['Marathon'])
```

With the data
loaded, this code
lets you check if it's
all OK.

`column_headings =`
`row_data[row_label] = row`

`row_label = row.pop(0)`

`row = each_line.strip().split(',')`

`paces.readline().strip().split(',')`

`column_headings.pop(0)`

`row_data = {}`



Marathon Magnets Solution

Here's some code that reads the raw data from the CSV data file. The column headings from the first line are loaded into a list called `column_headings`. The rest of the data (all the rows of times) are loaded into a dictionary called `row_data`, with each row of data keyed with the **row label string** from the start of each line. Of course, as luck would have it, someone was cleaning the fridge door, and they've left a bunch of magnets on the floor. You were to see if you could arrange the magnets into their correct order.

Create the
column headings
from the first
line of data.

```
row_data = {}
```

You need to be sure to create an empty
dictionary for the row times.

with open('PaceData.csv') as paces:

```
column_headings =
```

```
paces.readline().strip().split(',')
```

```
column_headings.pop(0)
```

Delete the first
heading, the
"VO2" string—you don't
need it.

for each_line in paces:

← Process the rest of the file.

```
row = each_line.strip().split(',') ←
```

↑ Read a line from the file, strip it
of unwanted whitespace, and then
split the line on comma.

Extract the
row label.

```
row_label = row.pop(0)
```

It's the same deal here:
take the line, strip it, and
then split on comma.

```
row_data[row_label] = row
```

```
num_cols = len(column_headings)
print(num_cols, end=' -> ')
print(column_headings)
```

Use the row label
together with the rest
of the line's data to
update the dictionary.

```
num_2mi = len(row_data['2mi'])
print(num_2mi, end=' -> ')
print(row_data['2mi'])
```

```
num_Marathon = len(row_data['Marathon'])
print(num_Marathon, end=' -> ')
print(row_data['Marathon'])
```



Test Drive

Load your code into IDLE and, with the CSV in the same folder as your code, run it to see what you get on screen.

Your code in IDLE →

```

row_data = {}

with open('PaceData.csv') as paces:

    column_headings = paces.readline().strip().split(',')
    column_headings.pop(0)

    for each_line in paces:
        row = each_line.strip().split(',')
        row_label = row.pop(0)
        row_data[row_label] = row

    num_cols = len(column_headings)
    print(num_cols, end='-> ')
    print(column_headings)

    num_2mi = len(row_data['2mi'])
    print(num_2mi, end='-> ')
    print(row_data['2mi'])

```

The output confirms that each row of data has 50 data items.

The column headings

The "2mi" row of data

The "Marathon" row of data

Python Shell

```

Python 3.1.2 (r312:79360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
RESTART

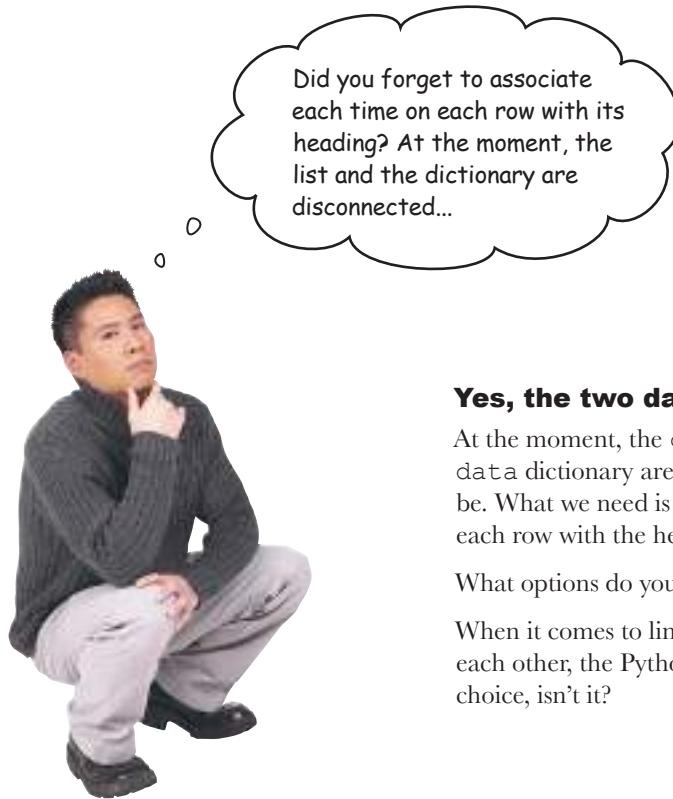
50-> ['84.8', '82.9', '81.1', '79.3', '77.5', '75.8', '74.2', '72.5', '70.9', '69.4', '67.9', '66.4', '64.9', '63.5', '62.1', '60.7', '59.4', '58.1', '56.8', '55.5', '54.3', '53.1', '52', '50.8', '49.7', '48.6', '47.5', '46.5', '45.5', '44.5', '43.5', '42.5', '41.6', '40.7', '39.8', '38.9', '38', '37.2', '36.4', '35.6', '34.4', '34', '33.3', '32.6', '31.8', '31.1', '30.5', '29.8', '29.1', '28.5']
50-> ['8:00', '8:10', '8:21', '8:33', '8:44', '8:56', '9:08', '9:20', '9:33', '9:46', '9:59', '10:13', '10:26', '10:41', '10:55', '11:10', '11:25', '11:40', '11:56', '12:12', '12:29', '12:45', '13:03', '13:20', '13:38', '13:57', '14:16', '14:35', '14:54', '15:15', '15:35', '15:56', '16:18', '16:40', '17:02', '17:25', '17:49', '18:13', '18:38', '19:03', '19:28', '19:55', '20:22', '20:49', '21:17', '21:40', '22:15', '22:45', '23:16', '23:48']
50-> ['2:05:34', '2:08:24', '2:11:17', '2:14:15', '2:17:16', '2:20:21', '2:23:31', '2:26:44', '2:30:02', '2:33:25', '2:36:52', '2:40:24', '2:44:00', '2:47:42', '2:51:28', '2:55:20', '2:59:16', '3:03:18', '3:07:26', '3:11:39', '3:15:58', '3:20:22', '3:24:53', '3:29:29', '3:34:12', '3:39:01', '3:43:57', '3:48:59', '3:54:09', '3:59:25', '4:04:48', '4:10:18', '4:15:56', '4:21:42', '4:27:35', '4:33:36', '4:39:46', '4:46:04', '4:52:30', '4:59:05', '5:05:48', '5:12:41', '5:19:43', '5:26:55', '5:34:16', '5:41:48', '5:49:29', '5:57:21', '6:05:23', '6:13:37']
>>>

```

Ln. 9 Col. 4

That's a great start: you've managed to read the data from the CSV and put the headings into a list and the data into a dictionary.

What's next?



Did you forget to associate each time on each row with its heading? At the moment, the list and the dictionary are disconnected...

Yes, the two data structures should be linked.

At the moment, the `column_headings` list and the `row_data` dictionary are not linked in any way, and they need to be. What we need is some way to *connect* each of the times in each row with the heading that tops their column of data.

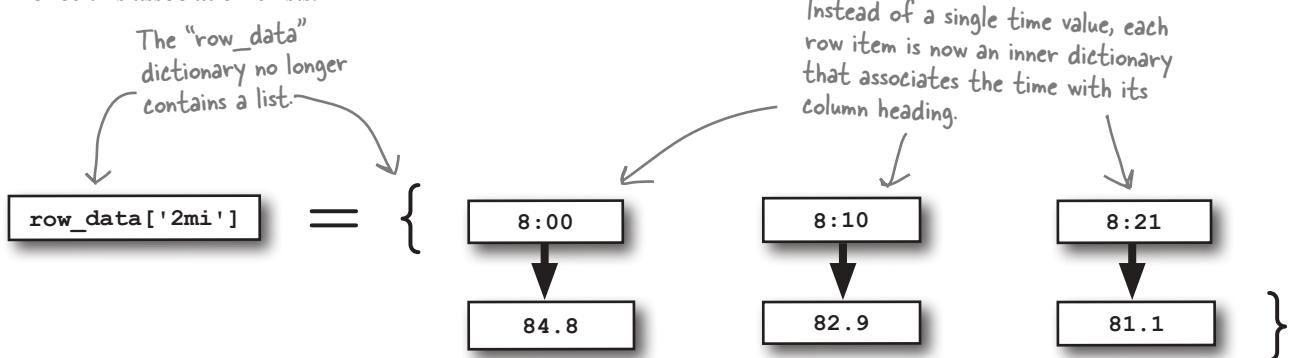
What options do you have here?

When it comes to linking (or associating) two data items with each other, the Python dictionary is the data structure of choice, isn't it?

Store each time as a dictionary

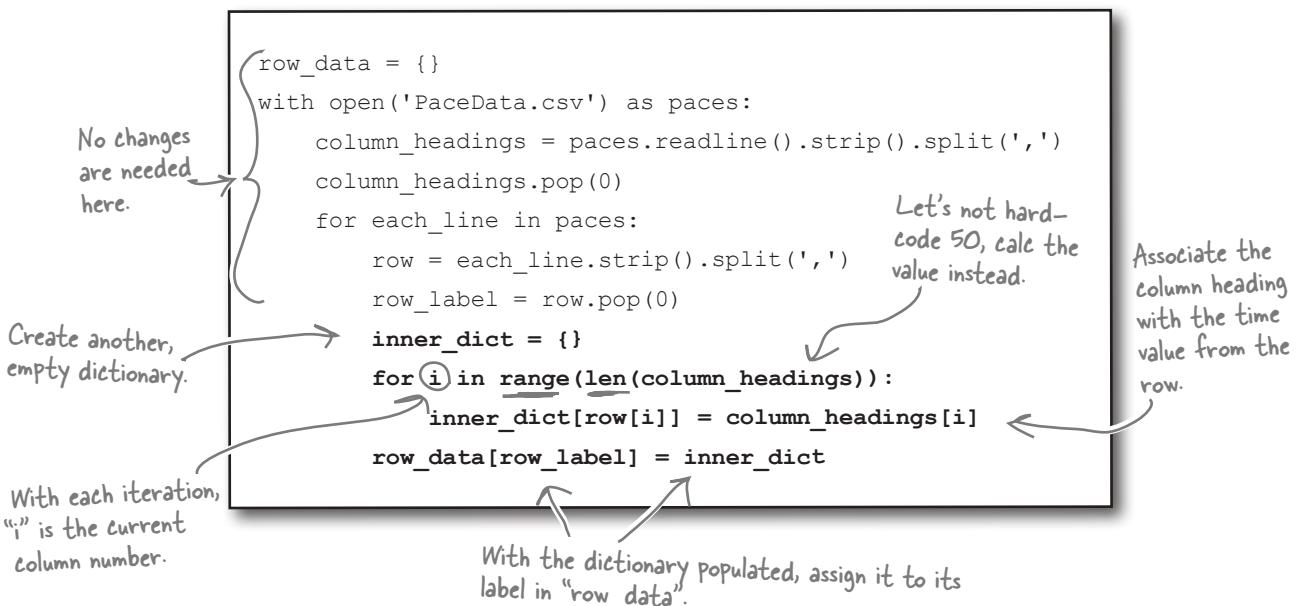
Rather than simply storing each time in the `row_data` dictionary as a number, let's store the data as *as a dictionary*, with the *key* set to the time and the *value* set to the column heading. That way, you can quickly and easily determine for any time which column it is associated with, right?

Here's a portion of what the data structure looks like in Python's memory once this association exists:



All you need to do is work out how to populate the inner dictionary with the row data and the associated columns headings...and you'll have all the data you need.

The trick in creating the data structure is to realize that each row, *including the column headings*, are of a **fixed size**: 50 items. Knowing this, it's not much work to create the dictionary you need:





An IDLE Session

Go ahead and add the extra dictionary populating code to your program. Let's remove all of those `print()` statements from the end of your program, because you'll use the IDLE shell to test your code. Run the code by pressing F5 or by selecting the Run Module option from the Run menu. Use the `dir()` BIF to confirm that your program code executed and that a collection of variables have been created in Python's namespace:

```
>>> dir()
['__builtins__', '__doc__', '__name__', '__package__', 'column_headings', 'each_line', 'i',
'inner_dict', 'paces', 'row', 'row_data', 'row_label']
```

All of your code's variables exist.

V02	84.8	82.9	81.1	79.3	77.5
2mi	8:00	8:10	8:21	8:33	8:44
5k	12:49	13:06	13:24	13:42	14:00
5mi	21:19	24:48	22:17	22:47	23:18
10k	26:54	27:30	28:08	28:45	29:24
15k	41:31	42:27	43:24	44:23	45:23
run	44:46	45:46	46:00	47:51	48:56
20k	56:29	57:45	59:03	1:00:23	1:01:45
13.1mi	59:49	1:01:09	1:02:32	1:03:56	1:05:23
25k	1:11:43	1:13:20	1:14:59	1:16:40	1:18:24
30k	1:27:10	1:19:08	1:31:08	1:33:11	1:35:17
Marathon	2:05:34	2:08:24	2:11:17	2:14:15	2:17:16

Take another look at (part of) the spreadsheet data file above, and let's try and find the column heading associated with the **43:24** time on the **15k** row. Let's then use the column heading to find the predicted time for a **20k** run:

```
>>> column_heading = row_data['15k']['43:24']
>>> column_heading
'81.1'
```

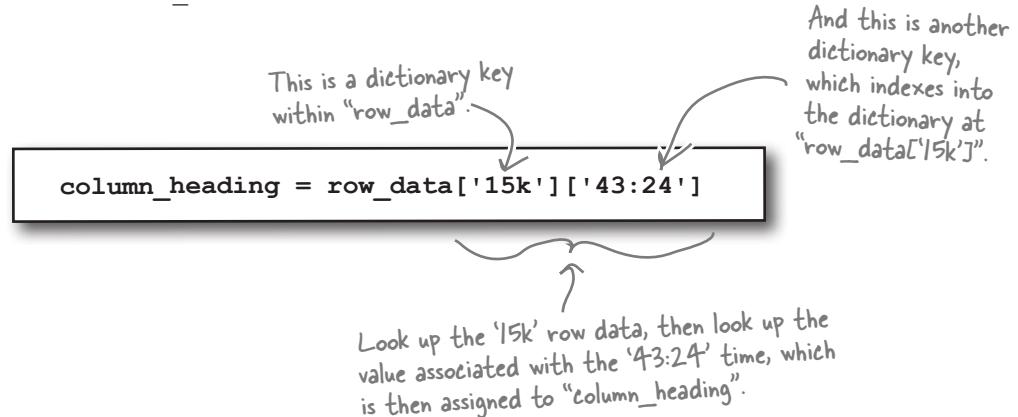
The associated column heading is correctly identified as "81.1".

```
>>> prediction = [k for k in row_data['20k'].keys() if row_data['20k'][k] == column_heading]
>>> prediction
['59:03']
```

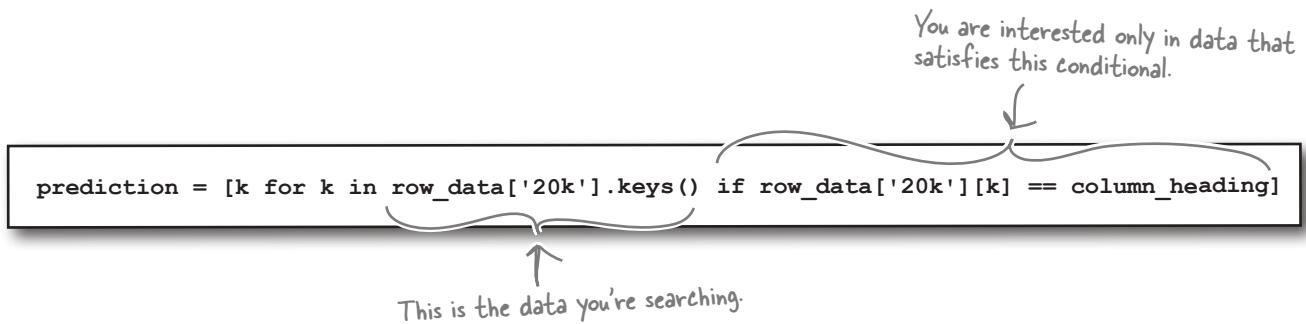
A time of "59:03" is correctly predicted, too.

Dissect the prediction code

Let's take a moment to review what just happened at the bottom of the IDLE Session from the last page. This line of code is a *double-dictionary lookup* on the dictionary-of-dictionaries stored in `row_data`:

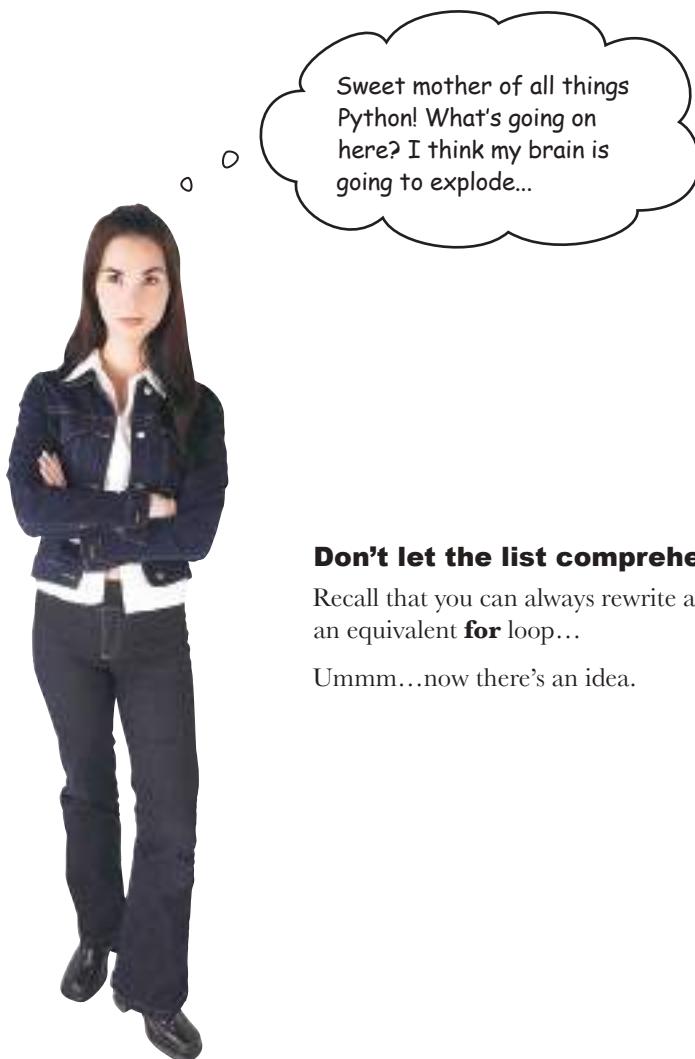


Working out the predicted time in the **20k** row of data involves **finding the key** in the row's dictionary whose *value* is set to the just-discovered value stored in `column_heading`.



A *conditional list comprehension* is put to good use here. Recall that the list comprehension syntax is a shorthand notation for a **for** loop. The loop searches through the data in the list of keys associated with the dictionary stored at `row_data['20k']`. If the value associated with the key (in `k`) is the same as `column_heading`, the value of `k` is added to the comprehensions results, which are then assigned to a new list call `predicton`.

There's really an awful lot going on in that comprehension.



Don't let the list comprehension put you off.

Recall that you can always rewrite a list comprehension using an equivalent **for** loop...

Umm...now there's an idea.



Rewrite each of the list comprehensions on this page to use a **for** loop.

```
times = [t for t in row_data['Marathon'].keys()]
```

.....
.....
.....
.....
.....
.....

```
headings = [h for h in sorted(row_data['10mi'].values(), reverse=True)]
```

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

```
time = [t for t in row_data['20k'].keys() if row_data['20k'][t] == '79.3']
```

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

for loop



You were to rewrite each of the list comprehensions to use a **for** loop.

```
times = [t for t in row_data['Marathon'].keys()]
```

Start with an
empty list.

→ times = []

for each_t in row_data['Marathon'].keys():

 times.append(each_t)

With each iteration, append the key (which is a time value) onto the "times" list.

Turn the dictionary's keys into a list.

```
headings = [h for h in sorted(row_data['10mi'].values(), reverse=True)]
```

Start with an
empty list.

→ headings = []

for each_h in sorted(row_data['10mi'].values(), reverse=True):

 headings.append(each_h)

...being sure to sort the values in
reverse order (biggest first).

With each iteration, append the value (which is a column heading) onto the "times" list.

```
time = [t for t in row_data['20k'].keys() if row_data['20k'][t] == '79.3']
```

Start with an
empty list.

→ time = []

for each_t in row_data['20k'].keys():

 if row_data['20k'][each_t] == '79.3':

 time.append(each_t)

With each iteration, check to see if the column heading (the value part of the dictionary) equals "79.3" and if it does, append the time to the list.

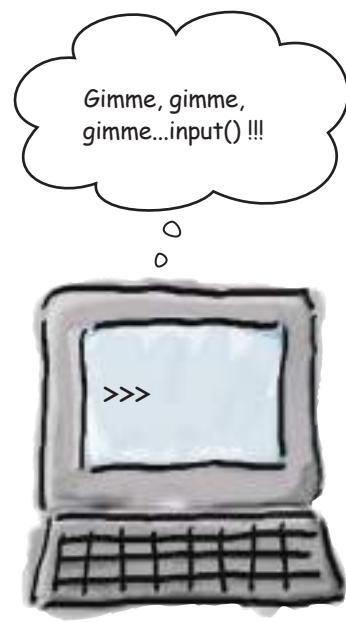
There's a
definite pattern
emerging here. ☺

Get input from your user

Now that you have your data within a Python data structure, it's time to ask your user what it is they are looking for.

Specifically, you need to know three things: the *distance* run, the *time* recorded, and the distance a *prediction* is required for.

When you get to move your app onto Android, you can use a nice graphical dialog to ask your user for input, but for now, let's quickly create a text-based user interface, which will allow you to develop and test the rest of the functionality required from your application. When you're done, you'll create the Android app.



Use `input()` for input

Python has the `input()` BIF that can help here, which is used to display a prompt on screen, and then accept keyboard input, returning what was entered as a string to your code.



An IDLE Session

Using the `input()` BIF is best demonstrated with some examples:

```
>>> res = input('What is your favorite programming language: ')
What is your favorite programming language: Python
>>> res
'Python'
```

The entered data is assigned to "res" and it's a STRING.

Provide the prompt to display to your user.

The `input()` BIF returns a string, which has been stripped of any trailing newline character, which would typically be included at the end of any input string. It is important to note that any input is returned as a string, regardless of what type of data you think you might be entering:

The entered data is assigned to "age" and it's a string, even though you might want to treat it like it's a number.

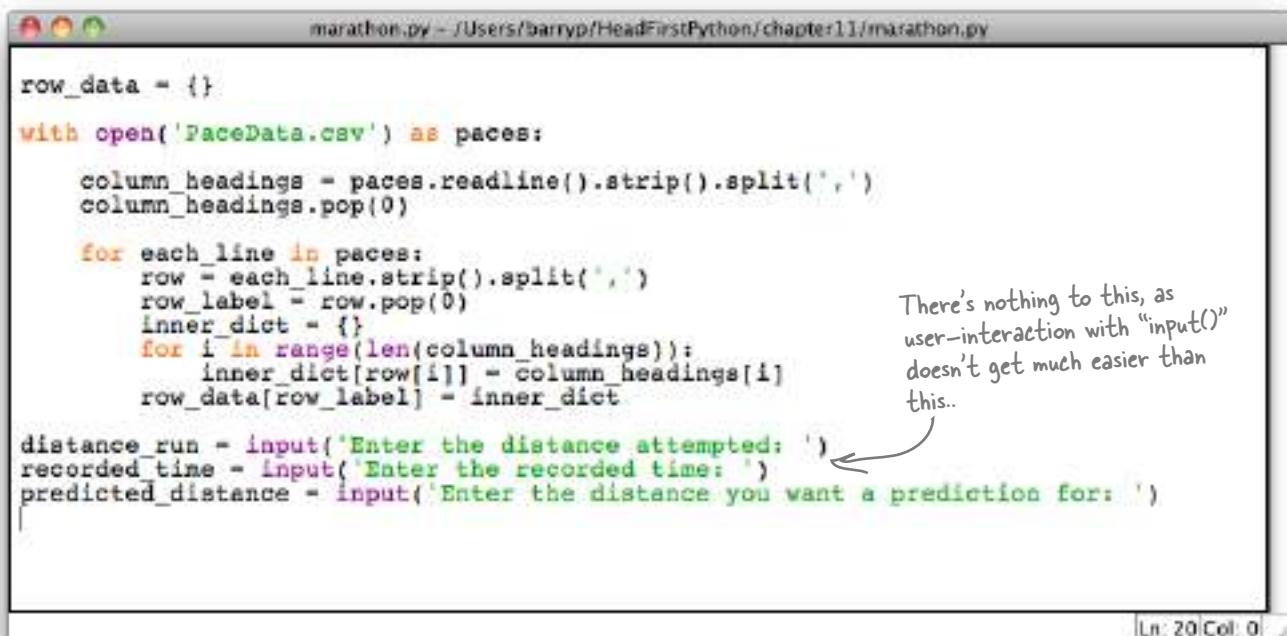
Convert the input to the type you need BEFORE using the data.

```
>>> age = input('What is your age: ')
What is your age: 21
>>> age
'21'
>>> int(age)
21
```

[Editor's note: Yeah... dream on, Paul. ☺]

Getting input raises an issue...

It's not hard to use `input()` to get the, um, er...input you need. Here's your code from earlier with three calls to `input()` added to interact with your user.



```
marathon.py - /Users/barryp/HeadFirstPython/chapter11/marathon.py

row_data = {}

with open('PaceData.csv') as paces:

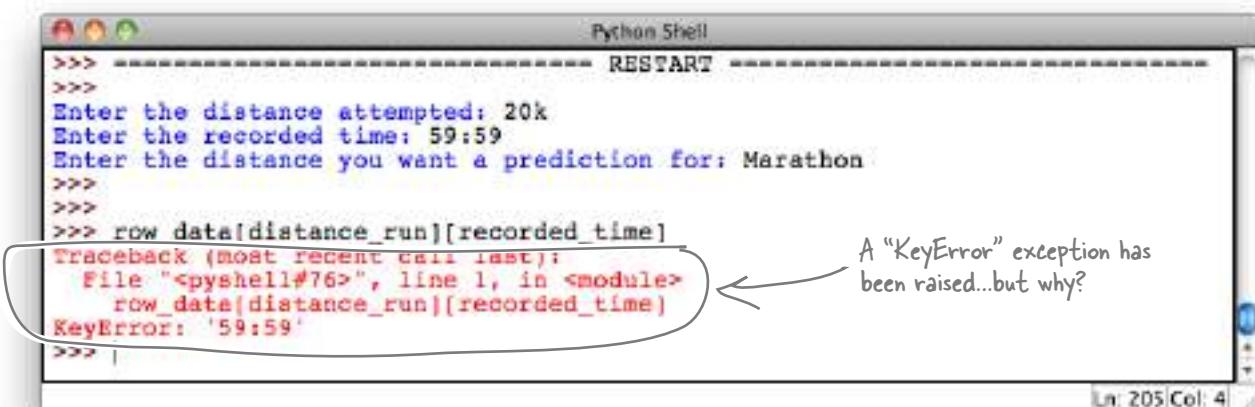
    column_headings = paces.readline().strip().split(',')
    column_headings.pop(0)

    for each_line in paces:
        row = each_line.strip().split(',')
        row_label = row.pop(0)
        inner_dict = {}
        for i in range(len(column_headings)):
            inner_dict[row[i]] = column_headings[i]
        row_data[row_label] = inner_dict

distance_run = input('Enter the distance attempted: ')
recorded_time = input('Enter the recorded time: ')
predicted_distance = input('Enter the distance you want a prediction for: ')
```

There's nothing to this, as user-interaction with "input()" doesn't get much easier than this..

When your program runs, your user enters some data, and *look what happens*:



```
>>> -----
>>> ----- RESTART -----
>>>
Enter the distance attempted: 20k
Enter the recorded time: 59:59
Enter the distance you want a prediction for: Marathon
>>>
>>>
>>> row_data[distance_run][recorded_time]
Traceback (most recent call last):
  File "<pyshell#76>", line 1, in <module>
    row_data[distance_run][recorded_time]
KeyError: '59:59'
```

A "KeyError" exception has been raised...but why?

If it's not in the dictionary, it can't be found.

The data in the `row_data` dictionary originally comes from the spreadsheet and is read into your program from the CSV file.

If the data value entered into the `recorded_time` variable is in the dictionary, things are going to be fine, because there's a *match*. However, if the data entered into the `recorded_time` variable *doesn't* match anything in the dictionary, you'll get a `KeyError`.

But how is this “problem” handled during training?



The entered time for a 20k run (59:59) falls between these two values on the pace sheet.

10k	26:54	27:30	28:08	28:45	29:24
15k	41:31	42:27	43:24	44:23	45:23
10mi	44:46	45:46	46:46	47:51	48:56
20k	56:29	57:45	59:03	1:00:23	1:01:45
13.1mi	59:49	1:01:09	1:02:32	1:03:56	1:05:23
25k	1:11:43	1:13:20	1:14:59	1:16:40	1:18:24
30k	1:27:10	1:19:08	1:31:08	1:33:11	1:35:17
Marathon	2:05:34	2:08:24	2:11:17	2:14:15	2:17:16

Search for the closest match

All you need to do is search the row of data for the closest match, right? And guess what? The Head First Code Review Team think they have some functions that might help here.

There's nothing better than sharing our code with our fellow Python programmers. Check out our "find_it" module.



This code is in a file called "find_it.py" and you can download a copy from this book's support website.

```
Find_it.py - /Users/barryp/HeadFirstPython/chapter11/Find_it.py
def find_closest(look_for, target_data):
    def what_s_the_difference(first, second):
        if first == second:
            return(0)
        elif first > second:
            return(first - second)
        else:
            return(second - first)

    max_diff = 9999999
    for each_thing in target_data:
        diff = what_s_the_difference(each_thing, look_for)
        if diff == 0:
            found_it = each_thing
            break
        elif diff < max_diff:
            max_diff = diff
            found_it = each_thing
    return(found_it)
```

The "find_closest" function does a simple linear search, returning the value in "target_data" that most closely matches the "look_for" argument.

Here's an example of a nested function, which is allowed in Python. Given two values, this function returns the difference between them.

This may not be the most efficient search code ever written, but it works.



An IDLE Session

Let's test the `find_it.py` module to try and determine if it meets the requirements of your application. Load the module into IDLE and then press **F5** or choose Run Module from the Run menu:

```
>>> find_closest(3.3, [1.5, 2.5, 4.5, 5.2, 6])
2.5
>>> find_closest(3, [1, 5, 6])
1
>>> find_closest(3, [1, 3, 4, 6])
3
>>> find_closest(3.6, [1.5, 2.5, 4.5, 5.2, 6])
4.5
>>> find_closest(3, [1, 4, 6])
4
>>> find_closest(2.6, [1.5, 2.5, 4.5, 5.2, 6])
2.5
```

Given a value to look for and some target data, the "find_closest" function seems to be doing the trick.

Let's try it with some of data that more closely resembles your CSV data:

```
>>> find_closest('59:59', ['56:29', '57:45', '59:03', '1:00:23', '1:01:45'])
Traceback (most recent call last):
  File "<pyshell#23>", line 1, in <module>
    find_closest('59:59', ['56:29', '57:45', '59:03', '1:00:23', '1:01:45'])
  File "/Users/barryp/HeadFirstPython/chapter11/find_it.py", line 15, in find_closest
    if diff == 0:
  File "/Users/barryp/HeadFirstPython/chapter11/find_it.py", line 11, in whats_the_difference
    TypeError: unsupported operand type(s) for -: 'str' and 'str'
```

yikes! Something's seriously broken here.



What do you think has gone wrong here? Why does the `find_closest()` function crash when asked to work with data from your CSV file?

The trouble is with time

The data in your CSV file is a *representation* of timing values. Rather than actual numbers, the values in the CSV are **strings**. This is great for you, because *you* understand what the representation means. Python, on the other hand, sees the data only as strings.

When you send your data to the `find_closest()` function, Python attempts to treat your strings as numbers and chaos ensues. What might work would be to convert the time-strings into numbers. But how?



The time-to-seconds-to-time module

The Head First Code Review Team's generosity knows no bounds. Sure enough, their rather strangely named `tm2secs2tm.py` module looks like it might help.

Here's the guy's
"tm2secs2tm.py" module. ↴

Grab a copy of this code from
this book's support website.



```
tm2secs2tm.py - /Users/barryp/HeadFirstPython/chapter11/tm2secs2tm.py
import time

def format_time(time_string):
    tlen = len(time_string)
    if tlen < 3:
        original_format = '%S'
    elif tlen < 6:
        original_format = '%M:%S'
    else:
        original_format = '%H:%M:%S'
    time_string = time.strptime('%H:%M:%S', time.strptime(time_string, original_format))
    return time_string

def time2secs(time_string):
    time_string = format_time(time_string)
    hours, mins, secs = time_string.split(':')
    seconds = int(secs) + (int(mins)*60) + (int(hours)*60*60)
    return seconds

def secs2time(seconds):
    return time.strftime('%H:%M:%S', time.gmtime(seconds))
```

This function ensures that all times are formatted in "HH:MM:SS" format. This helps keep things simple when doing conversions to seconds.

Given a "time string", convert it to a value in seconds.

Convert a value in seconds to a "time string".

Ln: 24 Col: 0



Exercise

Now that you have the `tm2secs2tm.py` and `find_it.py` modules, let's create a function that uses the facilities provided by these modules to solve your searching problem. Your new function, called `find_nearest_time()`, takes two arguments: the time to look for and a list of times to search. The function returns the closest time found as a string:

The code you
need has been
started for
you.

```
from find_it import find_closest
from tm2secs2tm import time2secs, secs2time

def find_nearest_time(look_for, target_data):
```

Unlike in the previous
chapter, it is possible
to do what you need
to do here in only
four lines of code.



Exercise Solution

Now that you have the `tm2secs2tm.py` and `find_it.py` modules, you were to create a function that uses the facilities provided by these modules to solve your searching problem. Your new function, called `find_nearest_time()`, takes two arguments: the time to look for and a list of times to search. The function returns the closest time found as a string:

Import the team's code. ↗ { from `find_it` import `find_closest`
from `tm2secs2tm` import `time2secs`, `secs2time`

The function takes two arguments, a time string and a list of time strings.

def `find_nearest_time(look_for, target_data):`

Convert the time string you are looking for into its equivalent value in seconds. ↗ what = `time2secs(look_for)`

where = [`time2secs(t)` for t in `target_data`] ↗ Convert the lines of time strings into seconds.

Call "find_closest()", supplying the converted data. ↗ res = `find_closest(what, where)`

return(`secs2time(res)`) ↗ Return the closest match to the calling code, after converting it back to a time string.



An IDLE Session

Let's try out your code at the IDLE shell to see if your time "problems" have been resolved:

Here's some of your race data. Let's work with data from the "20k" row.

10k	26:54	27:30	28:08	28:45	29:24
15k	41:31	42:27	43:24	44:23	45:23
40mi	44:46	45:46	46:48	47:51	48:56
20k	56:29	57:45	59:03	1:00:23	1:01:45
13.1mi	59:49	1:01:09	1:02:32	1:03:56	1:05:23
25k	1:11:43	1:13:20	1:14:59	1:16:40	1:18:24
30k	1:27:10	1:19:08	1:31:08	1:33:11	1:35:17
Marathon	2:05:34	2:08:24	2:11:17	2:14:15	2:17:16

```
>>> find_nearest_time('59:59', ['56:29', '57:45', '59:03', '1:00:23', '1:01:45'])
'01:00:23'
>>> find_nearest_time('1:01:01', ['56:29', '57:45', '59:03', '1:00:23', '1:01:45'])
'01:00:23'
>>> find_nearest_time('1:02:01', ['56:29', '57:45', '59:03', '1:00:23', '1:01:45'])
'01:01:45'
>>> find_nearest_time('57:06', ['56:29', '57:45', '59:03', '1:00:23', '1:01:45'])
'00:56:29'
```

Great! This appears to be working fine.



Test DRIVE

With all this code available to you, it's an easy exercise to put it all together in your program and produce a complete solution to the Marathon Club's prediction problem. Let's take it for a test run.

```
marathon.py - /Users/barryp/HeadFirstPython/chapter11/marathon.py

from find_it import find_closest
from tm2secs2tm import time2secs, secs2time

def find_nearest_time(look_for, target_data):
    what = time2secs(look_for)
    where = [time2secs(t) for t in target_data]
    res = find_closest(what, where)
    return(secs2time(res))

row_data = {}

with open('PaceData.csv') as paces:
    column_headings = paces.readline().strip().split(',')
    column_headings.pop(0)

    for each_line in paces:
        row = each_line.strip().split(',')
        row_label = row.pop(0)
        inner_dict = {}
        for i in range(len(column_headings)):
            inner_dict[row[i]] = column_headings[i]
        row_data[row_label] = inner_dict

distance_run = input('Enter the distance attempted: ')
recorded_time = input('Enter the recorded time: ')
predicted_distance = input('Enter the distance you want a prediction for: ')
closest_time = find_nearest_time(recorded_time, row_data[distance_run])
closest_column_heading = row_data[distance_run][closest_time]

prediction = [k for k in row_data[predicted_distance].keys()
              if row_data[predicted_distance][k] == closest_column_heading]

print('The predicted time running ' + predicted_distance + ' is: ' + prediction[0] + '.')


```

This code is used "as is".

Find the nearest time within the data.

Extract the column heading.

Search for a predicted time at the desired distance and display it on screen.

Ln: 37 Col: 0

```
Python Shell
>>> ===== RESTART =====
>>>
Enter the distance attempted: 20k
Enter the recorded time: 59:59
Enter the distance you want a prediction for: Marathon
Traceback (most recent call last):
  File "/Users/barryp/HeadFirstPython/chapter11/marathon.py", line 31, in <module>
    closest_column_heading = row_data[distance_run][closest_time]
KeyError: '01:00:23'


```

Try out your program with the same data input from earlier.

Another "KeyError"...

Ln: 13 Col: 4

After all that, you're getting the same error as before. Bummer.

The trouble is still with time...

Or, to be more precise, with how the `tm2secs2tm.py` module formats time strings. Take another look at the results from the previous IDLE Session. Do you notice anything strange about the results returned by the call to the `find_nearest_time()` function?

```
>>> find_nearest_time('59:59', ['56:29', '57:45', '59:03', '1:00:23', '1:01:45'])
'01 00:23'
>>> find_nearest_time('1:01:01', ['56:29', '57:45', '59:03', '1:00:23', '1:01:45'])
'01.00:23'
>>> find_nearest_time('1:02:01', ['56:29', '57:45', '59:03', '1:00:23', '1:01:45'])
'01:01:45'
>>> find_nearest_time('57:06', ['56:29', '57:45', '59:03', '1:00:23', '1:01:45'])
'00 56:29'
```

All the returned times use the "HH:MM:SS" format.

When your code takes one of these returned values and tries to index into your dictionary, there's no match found, because your dictionary's keys do **not** confirm to the HH:MM:SS format. The solution to this problem is to ensure that *every time* you use a time-string in your code, make sure it's in HH:MM:SS format:

```
from find_it import find_closest
from tm2secs2tm import time2secs, secs2time, format_time

def find_nearest_time(look_for, target_data):
    what = time2secs(look_for)
    where = [time2secs(t) for t in target_data]
    res = find_closest(what, where)
    return(secs2time(res))

row_data = {}
with open('FacebookData.csv') as pages:
    column_headings = pages.readline().strip().split(',')
    column_headings.pop(0)
    for each_line in pages:
        row = each_line.strip().split(',')
        row_label = row.pop(0)
        inner_dict = {}
        for i in range(len(column_headings)):
            inner_dict[format_time(row[i])] = column_headings[i]
        row_data[row_label] = inner_dict

distance_run = input('Enter the distance attempted: ')
recorded_time = input('Enter the recorded time: ')
predicted_distance = input('Enter the distance you want a prediction for: ')

closest_time = find_nearest_time(format_time(recorded_time), row_data[distance_run])
closest_column_heading = row_data[predicted_distance][closest_time]

prediction = [k for k in row_data[predicted_distance].keys()
              if row_data[predicted_distance][k] == closest_column_heading]

print('The predicted time running ' + predicted_distance + ' is: ' + prediction[0] + '.)
```

Import the "format_time()" function from the "tm2secs2tm.py" module.

Use the function to ensure the times used internally by your code are formatted in "HH:MM:SS" format.



Test DRIVE

Let's try your code one more time. Hopefully, now that all of the time strings within the system conform to HH:MM:SS format, your code will behave itself.

```
Python 3.1.2 (r312:79360M, Mar 24 2010, 01:33:18)
[GCC 4.0.1 (Apple Inc. build 5493)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> -----
RESTART

>>>
Enter the distance attempted: 20k
Enter the recorded time: 59:59
Enter the distance you want a prediction for: Marathon
Traceback (most recent call last):
  File "/Users/barryp/HeadFirstPython/chapter11/marathon.py", line 31, in <module>
    closest_column_heading = row_data[distance_run][closest_time]
KeyError: '01:00:23'
>>> -----
RESTART

>>>
Enter the distance attempted: 20k
Enter the recorded time: 59:59
Enter the distance you want a prediction for: Marathon
The predicted time running Marathon is: 02:14:15.
>>> -----
RESTART

>>>
Enter the distance attempted: 5mi
Enter the recorded time: 23:45
Enter the distance you want a prediction for: 10mi
The predicted time running 10mi is: 00:50:02.
>>> -----
RESTART

>>>
Enter the distance attempted: 10k
Enter the recorded time: 32:15
Enter the distance you want a prediction for: 25k
The predicted time running 25k is: 01:25:49.
>>> -----
```

This is the previous test, which crashed with a "KeyError".

This time around, your program behaves itself and works fine.

Another test confirms that things are working well.

And one final test makes sure.

This is working well. You've solved your application's central problem: your program **reads** in the spreadsheet data from the CSV file, **turns** it into a dictionary of dictionaries, and lets you **interact** with your user to **acquire** the recorded time at a particular distance before **predicting** a time for another distance.

Not counting the code provided by the Head First Code Review Team, you've written fewer than 40 lines of code to solve this problem. *That's quite an achievement.* All that's left to do is to port your program to the club's Android's phones.

And porting to Android won't take too long, will it?

Port to Android

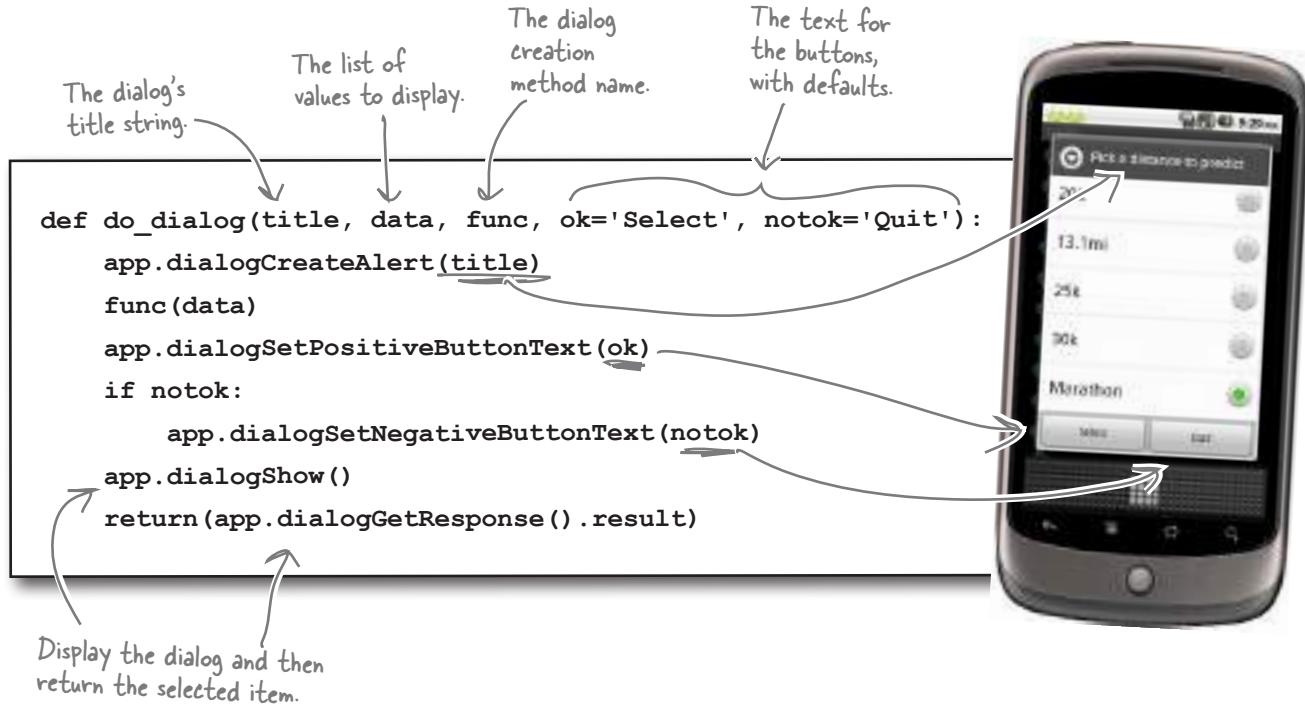
Your code is working great. Now it's time to **port** your text-based Python program to Android. Most of your code doesn't need to change, only the parts that interact with your user.

Obviously, you'll want to make things as easy to use as possible for users of your latest Android app, providing an interface not unlike this one.



Your Android app is a bunch of dialogs

Your Android app interacts with your users through a series of dialogs. Other than the single dialog that requests data *from* your user, the other three share certain similarities. You can take advantage of these shared features by creating a utility function which abstracts the dialog creation details:



Sharpen your pencil

Assume the existence of a list called `distances`, which contains the row distance labels (2mi, 5k, 5mi, and so on). In the space below, provide the two calls to the `do_dialog()` function needed to create the two `dialogSetSingleChoiceItems` shown on the left of the previous page.

.....
.....
.....
.....



You were to assume the existence of a list called `distances`, which contains the row distance labels. In the space below, you were to provide the two calls to the `do_dialog()` function needed to create the two `dialogSetSingleChoiceItems`.

```

do_dialog("Pick a distance", distances, ← Provide the list of
          app.dialogSetSingleChoiceItems) items to display.

Provide the dialog title. ← Ditto: do it all again for the other dialog.

do_dialog("Pick a distance to predict", distances, ← Provide the type of dialog to use.
          app.dialogSetSingleChoiceItems)

do_dialog('The predicted time running ' + predicted_distance +
          ' is: ', prediction, app.dialogSetItems, "OK", None)

```

Here's another example. ← This last one's a little trickier, because you have to build up the dialog title from some variables (that you'll need to have created first). ← Use a different dialog creating method this time. ← Override the default values for the dialog's buttons.

Get your Android app code ready

To use your dialog creating code, **import** the necessary libraries, **define** some constants, **create** an Android object, and **reuse** some code from earlier in this book:

```

import time ← Do your imports.
import android
...
distances = [ '2mi', '5k', '5mi', '10k', '15k', '10mi', '20k',
              '13.1mi', '25k', '30k', 'Marathon' ] ← Create a list of row labels.

...
hello_msg = "Welcome to the Marathon Club's App"
quit_msg = "Quitting the Marathon Club's App." ← Define two friendly messages.

...
app = android.Android() ← Create an Android app object.

def status_update(msg, how_long=2):
    app.makeText(msg)
    time.sleep(how_long) ← This function is taken "as-is" from earlier in this book.

```

Android Pool Puzzle



Your **job** is to take the code from the pool and place it into the blank lines in your Android app code.

You can assume that the `row_data` dictionary exists and has been populated. The variables shown at the bottom of the last page have also been created, and the `status_update()` and `do_dialog()` functions are available to you. Your **goal** is to arrange the code so that it implements the UI interactions you need.

```

status_update(hello_msg)
resp = do_dialog("Pick a distance", distances, .....)

.....  

distance_run = .....  

distance_run = distances[distance_run]  

..... = app.dialogGetInput("Enter a " + distance_run + " time:",  

                           "Use HH:MM:SS format:").result  

closest_time = find_nearest_time(format_time( ..... ), row_data[distance_run])  

closest_column_heading = row_data[distance_run][closest_time]  

resp = do_dialog("Pick a distance to predict", distances, .....)  

.....  

predicted_distance = .....  

predicted_distance = distances[predicted_distance]  

prediction = [k for k in row_data[predicted_distance].keys()  

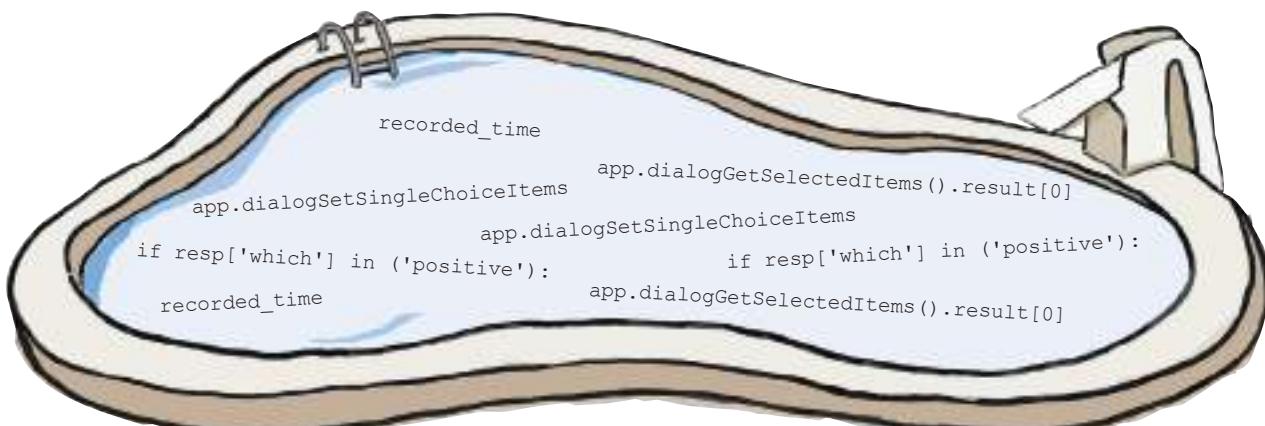
              if row_data[predicted_distance][k] == closest_column_heading]  

do_dialog('The predicted time running ' + predicted_distance + ' is: ',  

          prediction, app.dialogSetItems, "OK", None)
status_update(quit_msg)

```

The `dialogGetInput()` method displays the input dialog box.



Android Pool Puzzle Solution



Your **job** was to take the code from the pool and place it into the blank lines in your Android app code. You were to assume that the `row_data` dictionary exists and has been populated. The variables you need also have been created, and the `status_update()` and `do_dialog()` functions were available to you. Your **goal** was to arrange the code so that it implements the UI interactions you need.

Ask your user to pick a distance from the list of labels.

```

status_update(hello_msg)
resp = do_dialog("Pick a distance", distances, app.dialogSetSingleChoiceItems)
if resp['which'] in ('positive'):
    distance_run = app.dialogGetSelectedItems().result[0]
    distance_run = distances[distance_run]
    recorded_time = app.dialogGetInput("Enter a " + distance_run + " time:",
                                       "Use HH:MM:SS format:").result
    closest_time = find_nearest_time(format_time(recorded_time), row_data[distance_run])
    closest_column_heading = row_data[distance_run][closest_time]
    resp = do_dialog("Pick a distance to predict", distances, app.dialogSetSingleChoiceItems)
    if resp['which'] in ('positive'):
        predicted_distance = app.dialogGetSelectedItems().result[0]
        predicted_distance = distances[predicted_distance]
        prediction = [k for k in row_data[predicted_distance].keys()
                      if row_data[predicted_distance][k] == closest_column_heading]
        do_dialog('The predicted time running ' + predicted_distance + ' is: ',
                  prediction, app.dialogSetItems, "OK", None)
status_update(quit_msg)

```

Work out what column heading to use.

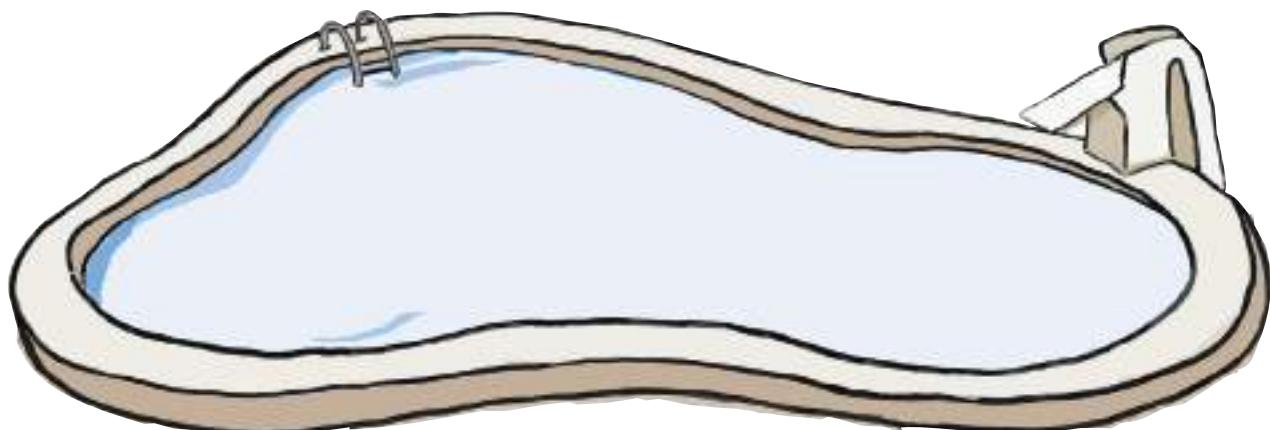
Look up the prediction.

Assign the selected distance label to "distance_run".

Ask your user enter the recorded time.

Ask your user to pick a distance from the list of labels to predict to.

Display the predicted time at the selected distance to your user.



Put your app together...

You now have all the code you need to create your app:

```

marathonapp.py - /Users/barryp/HeadFirstPython/chapter11/marathonapp.py

import time
import android
from find_it import find_nearest
from tm2secs2tm import time2secs, secs2time, format_time

def find_nearest_time(look_for, target_data):
    what = time2secs(look_for)
    where = [time2secs(t) for t in target_data]
    res = find_nearest(what, where)
    return(secs2time(res))

distances = ['5mi', '8k', '8mi', '10k', '10mi', '100k',
             '13.1mi', '26k', '30k', 'Marathon']
hello_msg = "Welcome to the Marathon Club's App"
quit_msg = "Quitting the Marathon Club's App."
row_data = {}

with open('/sdcard/clts/scripts/PaceData.csv') as paces:
    column_headings = paces.readline().strip().split(',')
    column_headings.pop(0)
    for each_line in paces:
        row = each_line.strip().split(',')
        row_label = row.pop(0)
        inner_dict = {}
        for i in range(len(column_headings)):
            inner_dict[format_time(row[i])] = column_headings[i]
        row_data[row_label] = inner_dict

app = android.Android()
def status_update(msg, how_long=2):
    app.makeToast(msg)
    time.sleep(how_long)

def do_dialog(title, data, func, ok='Select', notok='Quit'):
    app.dialogCreateAlert(title)
    func(data)
    app.dialogSetPositiveButtonText(ok)
    if notok:
        app.dialogSetNegativeButtonText(notok)
    app.dialogShow()
    return(app.dialogGetResponse().result)

status_update(hello_msg)

resp = do_dialog("Pick a distance", distances, app.dialogSetSingleChoiceItems)
if resp['which'] in ('positive'):
    distance_run = app.dialogGetSelectedItems().result[0]
    distance_run = distances[int(distance_run)]
    recorded_time = app.dialogGetInput("Enter a " + distance_run + " time:",
                                       "Use HH:MM:SS format").result
    closest_time = find_nearest_time(format_time(recorded_time), row_data[distance_run])
    closest_column_heading = row_data[distance_run][closest_time]
    resp = do_dialog("Pick a distance to predict", distances, app.dialogSetSingleChoiceItems)
    if resp['which'] in ('positive'):
        predicted_distance = app.dialogGetSelectedItems().result[0]
        predicted_distance = distances[int(predicted_distance)]
        prediction = {k for k in row_data[predicted_distance].keys()
                      if row_data[predicted_distance][k] == closest_column_heading}
        do_dialog('The predicted time running ' + predicted_distance + ' is: ' +
                  prediction, app.dialogSetItems, "OK", None)

status_update(quit_msg)
|
```

Do your imports.

Include your "find_nearest()" function.

Declare your constants.

NOTE: the location of the data file on the SDCARD is specific to Android.

Grab and preprocess your CSV data.

Create your Android app object and include your helper functions.

Display your UI to your user and process the resulting interaction.

Ln: 67 Col: 0



Test DRIVE

It's time to test your Android app on the Android Emulator before loading a working application onto a "real" phone. Start your Android emulator and begin by transferring your code and the files it needs onto the emulator's *SDCARD*. Use the `adb` command in the tools folder to copy `marathonapp.py`, `find_it.py`, `tm2sec2tm.py` and `PaceData.csv` to the emulator, and then take your app for a spin.

```
File Edit Window Help CopyToEmulator
$ tools/adb push marathonapp.py /mnt/sdcard/s14a/scripts
43 KB/s (2525 bytes in 0.056s)
$ tools/adb push find_it.py /mnt/sdcard/s14a/scripts
7 KB/s (555 bytes in 0.069s)
$ tools/adb push tm2secs2tm.py /mnt/sdcard/s14a/scripts
12 KB/s (628 bytes in 0.050s)
$ tools/adb push PaceData.csv /mnt/sdcard/s14a/scripts
59 KB/s (4250 bytes in 0.069s)
```

Copy your code
and its support
files to the
emulator with
these commands.



And there it is...
waiting for you
to test it.

Go on. You know you want to: tap that app!

Your app's a wrap!

All that's left to do is transfer your working Android app to the Marathon Club's phones...and that's easy when you use **AndFTP**. When you show off your latest work, the club's members can't believe their eyes.



And there's no stopping you!

You've put your Python skills and techniques to great use here.

Whether you're building an app for the smallest handheld device or the biggest web server, your Python skills help you get the job done.

Congratulations!



Your Python Toolbox

You've got Chapter 11 under your belt and you've demonstrated a mastery of your Python toolbox. Congratulations and well done!

Python Lingo

- A “conditional” list comprehension is one that includes a trailing “if” statement, allowing you to control which items are added to the new list as the comprehension runs.
- List comprehensions can be rewritten as an equivalent “for” loop.



BULLET POINTS

- The `input()` BIF lets you prompt and receive input from your users.
- If you find yourself using **Python 2** and in need of the `input()` function, use the `raw_input()` function instead.
- Build complex data structures by combining Python’s built-in lists, sets, and dictionaries.
- The `time` module, which is part of the standard library, has a number of functions that make converting between time formats possible.

It's time to go...



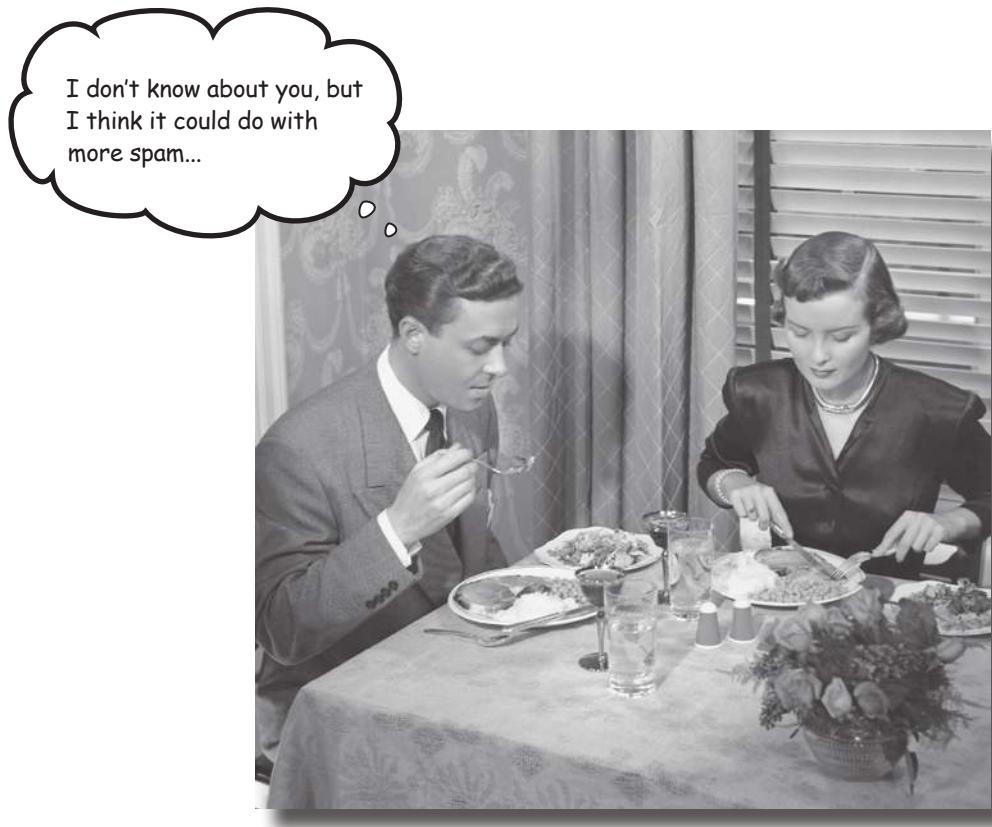
It's been a blast having you with us here on Lake Python. Come back soon and often. We love it when you drop by.

This is just the beginning

We're sad to see you leave, but there's nothing like taking what you've learned and putting it to use. You're just beginning your Python journey and you're in the driver's seat. We're dying to hear how things go, so drop us a line at the Head First Labs website, www.headfirstlabs.com, and let us know how Python is paying off for YOU!



The Top Ten Things (we didn't cover)



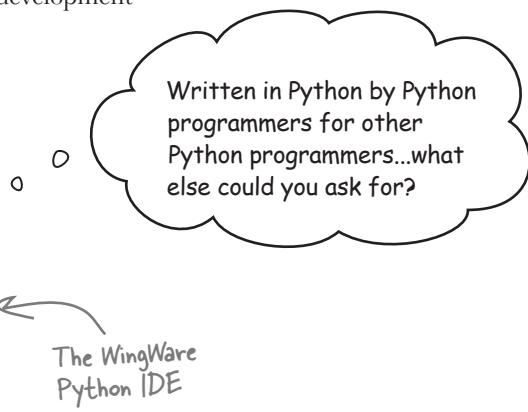
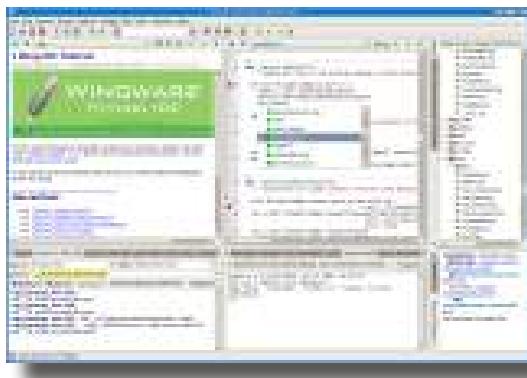
You've come a long way.

But learning about Python is an activity that never stops. The more Python you code, the more you'll need to learn new ways to do certain things. You'll need to master new tools and new techniques, too. There's just not enough room in this book to show you everything you might possibly need to know about Python. So, here's our list of the top ten things we didn't cover that you might want to learn more about next.

#1: Using a “professional” IDE

Throughout this book, you’ve used Python’s **IDLE**, which is great to use when first learning about Python and, although it’s a little quirky, can handle most programming tasks. It even comes with a built-in **debugger** (check out the Debug menu), which is surprisingly well equipped. Chances are, however, sooner or later, you’ll probably need a more full-featured integrated development environment.

One such tool worth looking into is the **WingWare Python IDE**. This professional-level development tool is specifically geared toward the Python programmer, is written by and maintained by Python programmers, and is *itself* written in Python. **WingWare Python IDE** comes in various licencing flavor: it’s free if you’re a student or working on an open source project, but you’ll need to pay for it if you are working within a for-profit development environment.



More general tools also exist. If you are running Linux, the **KDevelop IDE** integrates well with Python.

And, of course, there are all those *programmer editors* which are often all you’ll ever need. Many Mac OS X programmers swear by the **TextMate** programmer’s editor. There’s more than a few Python programmers using **emacs** and **vi** (or its more common variant, **vim**). Your author is a huge fan of **vim**, but also spends large portions of his day using **IDLE** and the Python shell.

#2: Coping with scoping

Consider the following short program:

```

scope.py - /Users/barryp/HeadFirstPython/Top Ten Things/scope.py
name = "Head First Python"
def what_happens_here():
    print(name)
    name = name + " is a great book!"
    print(name)
what_happens_here() ← Call the function.
print(name)

```

A global variable called "name".

A function which attempts to read from and write to the global variable called "name".

See what "name" is set to after the function runs.

Ln: 12 Col: 0

If you try to run this program, Python complains with this error message:

UnboundLocalError: local variable 'name' referenced before assignment...whatever that means!

When it comes to scope, Python is quite happy to let you access and read the value of a global variable within a function, **but you cannot change it**. When Python sees the assignment, it looks for a local variable called name, doesn't find it, and throws a hissy fit and an UnboundLocalError exception. To access *and* change a global variable, you must explicitly declare that's your intention, as follows:

```

scope.py - /Users/barryp/HeadFirstPython/Top Ten Things/scope.py
name = "Head First Python"
def what_happens_here():
    print(name)
    global name
    name = name + " is a great book!"
    print(name)
what_happens_here()
print(name)

```

Ln: 13 Col: 0

Some programmers find this quite ugly. Others think it's what comes to pass when you watch Monty Python reruns while designing your programming language. No matter what everyone thinks: this is what we're stuck with! ☺

#3: Testing

Writing code is one thing, but **testing** it is quite another. The combination of the Python shell and IDLE is great for testing and experimenting with small snippets of code, but for anything substantial, a testing framework is a must.

Python comes with *two* testing frameworks out of the box.

The first is familiar to programmers coming from another modern language, because it's based on the popular *xUnit* testing framework. Python's `unittest` module (which is part of the standard library) lets you create test code, test data, and a test suite for your modules. These exist in separate files from your code and allow you to exercise your code in various ways. If you already use a similar framework with your current language, rest assured that Python's implementation is essentially the same.

The other testing framework, called `doctest`, is also part of the standard library. This framework allows you to take the output from a Python shell or IDLE session and use it as a test. All you need to do is copy the content from the shell and add it to your modules *documentation strings*. If you add code like this to the end of your modules, they'll be ready for "doctesting":

```
if __name__ == "__main__":
    import doctest
    doctest.testmod()
```

If your code is imported as a module, this code does **NOT** run. If you run your module from the Command line, your tests run.

What do you mean:
you can't hear me...I
guess I should've
tested this first, eh?



If you then run your module at your operating system's command line, your tests run. If all you want to do is import your module's code and not run your tests, the previous `if` statement supports doing just that.

For more on `unittest` and `doctest`, search the online Python documentation on the Web or via IDLE's *Help* menu.

#4: Advanced language features

With a book like this, we knew we'd never get to cover the entire Python language unless we tripled the page count.

And let's face it, no one would thank us for that!

There's a lot more to Python, and as your confidence grows, you can take the time to check out these advanced language features:

Anonymous functions: the `lambda` expression lets you create small, one-line, non-named functions that can be incredibly useful once you understand what's going on.

Generators: like iterators, generators let you process sequences of data. Unlike iterators, generators, through the use of the `yield` expression, let you minimize the amount of RAM your program consumes while providing iterator-like functionality on large datasets.

Custom exceptions: create your own exception object based on those provided as standard by Python.

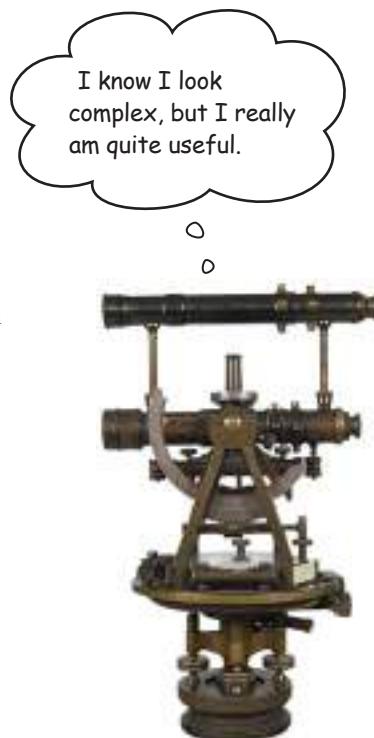
Function decorators: adjust the behavior of a preexisting function by hooking into its start-up and teardown mechanisms.

Metaclasses: custom classes that themselves can create custom classes. These are really only for the truly brave, although you did use a metaclass when you created your Sightings form using the *Django form validation framework* in Chapter 10.

Most (but not all) of these language features are primarily of interest to the Python programmer building tools or language extensions for use by other Python programmers.

You might never need to use some of these language features in your code, but they are all worth knowing about. Take the time to understand *when* and *where* to use them.

See #10 of this appendix for a list of my favorite Python books (other than this one), which are all great starting points for learning more about these language features.



#5: Regular expressions

When it comes to working with textual data, Python is a bit of a natural. The built-in `string` type comes with so many methods that most of the standard string operations such as finding and splitting are covered. However, what if you need to extract a specific part of a string or what if you need to search and replace within a string based on a specific specification? It is possible to use the built-in string methods to implement solutions to these types of problems, but—more times than most people would probably like to admit to—using a **regular expression** works better.

Consider this example, which requires you to extract the area code from the `phone_number` string and which uses the built-in string methods:

```
area_code.py - /Users/barryp/HeadFirstPython/Top Ten Things/area_code.py

phone_number = "Home: (555) 265-2901"
start = phone_number.find('(') ← Find the opening "(".
start = start+1
end = start+3 ← Calculate where the area code is
                in the string.
area_code = phone_number[start:end] ← Extract the area code.
print('The area code is: ' + area_code)

Ln: 12 Col: 0
```

This code works fine, but it *breaks* when presented with the following value for `phone_number`:

```
phone_number = "Cell (mobile): (555)-918-8271"
```

Why does this phone number cause the program to fail? Try it and see what happens...

When you use a **regular expression**, you can specify *exactly* what it is you are looking for and *improve the robustness* of your code:

```
area_code_re.py - /Users/barryp/HeadFirstPython/Top Ten Things/area_code_re.py

import re
phone_number = "Home: (555) 265-2901"
results = re.search('\((\d{3})\)', phone_number)
area_code = results.group(1)
print('The area code is: ' + area_code)

Ln: 10 Col: 0
```



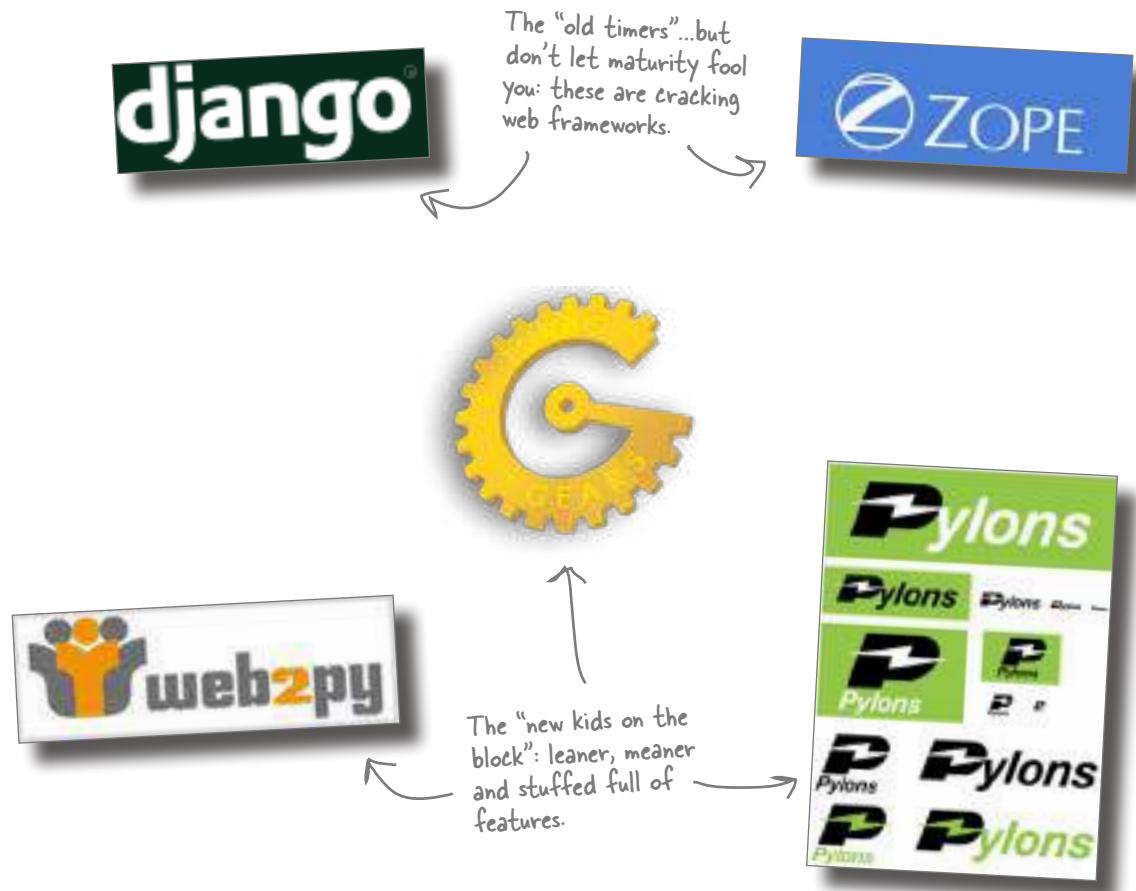
Jeff Friedl's regular expression "bible", which is well worth a look if you want to learn more. Look up the "re" module in Python's docs, too.

This looks a little strange, but this regular expression is looking for an opening "(" followed by three digits and then a closing ")". This specification is much more likely to find the area code and won't break as quickly as the other version of this program.

#6: More on web frameworks

When it comes to building web applications, CGI works, but it's a little old-fashioned. As you saw in Chapter 10, Google's App Engine technology supports CGI, but also WSGI and a number of web framework technologies. If you aren't deploying to the cloud and prefer to roll your own, you have plenty of choices. What follows is a representative sample. My advice: try a few on for size and see which one works best for you.

Search for the following terms in your favorite search engine: **Django**, **Zope**, **TurboGears**, **Web2py**, and **Pylons**.



#7: Object relational mappers and NoSQL

Working with SQL-based databases in Python is well supported, with the inclusion of SQLite in the standard library a huge boon. Of course, the assumption is you are familiar with SQL and happy to use SQL to work with your data.

But what if you aren't? What if you *detest* SQL?

An **object relational mapper** (ORM) is a software technology that lets you use an underlying SQL-based database *without* having to know anything about SQL. Rather than the procedural interface based on the Python database API, ORMs provide an object-oriented interface to your data, exposing it via *method calls* and *attribute lookups* as opposed to columns and rows.

Many programmers find ORMs a much more natural mechanism for working with stored datasets and the Python community creates and supports a number of them.

One of the most interesting is **SQL Alchemy**, which is popular and included in a number of the web framework technologies discussed in #6. Despite being hugely popular anyway, **SQL Alchemy** is also interesting because it supports *both* Python 2 and Python 3, which makes it a standout technology (for now).

If you find yourself becoming increasingly frustrated by SQL, check out an ORM. Of course, you have already experienced a similar technology: *Google App Engine's* **datastore** API is very similar in style to those APIs provided by the major Python ORMs.



There's NoSQL, too.

In addition to database technologies that let you avoid working with the underlying SQL-based database, a new breed of technologies have emerged that let you *drop* your SQL database in its entirety. Known collectively as **NoSQL**, these data tools provide an alternative non-SQL API to your data and do not use an SQL-based database management system at all. As these technologies are relatively new, there's been more activity around Python 2 than Python 3, but they are still worth checking out. **CouchDB** and **MongoDB** are the two most closely associated with robust Python implementations. If you like working with your data in a Python dictionary and wished your database technology let you store your data in much the same way, then you need to take a look at **NoSQL**: *it's a perfect fit*.



#8: Programming GUIs

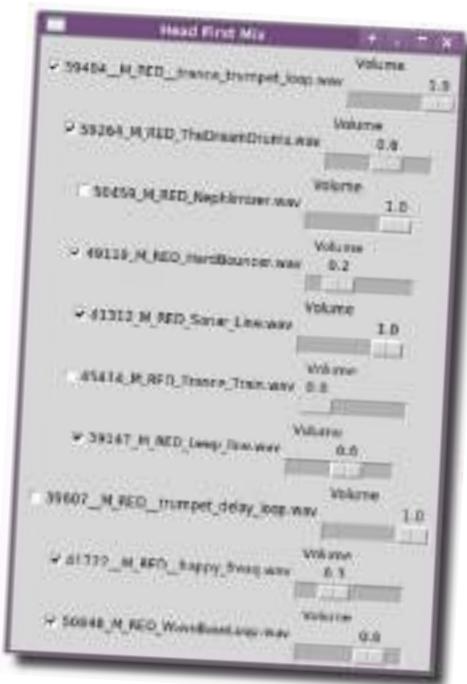
In this book, you've created text-based interfaces, web-based interfaces and interfaces that ran on Android devices. But what if you want to create a desktop application that runs on your or your user's desktop computer? Are you out of luck, or can Python help here, too?

Well...as luck would have it, Python comes preinstalled with a GUI-building toolkit called **tkinter** (shorthand for *Tk Interface*). It's possible to create a usable and useful graphical user interface (GUI) with **tkinter** and deploy it on Mac OS X, Windows, and Linux. With the latest version of **Tk**, your developed app takes on the characteristics of the underlying operating system, so when you run on Windows, your app looks like a Windows desktop app, when it runs on Linux, it looks like a Linux desktop app, and so on.

You write your Python and **tkinter** code *once*, then run it anywhere and it just works. There are lots of great resources for learning to program with **tkinter**, with one of the best being the last few chapters of *Head First Programming*, but since plugging *that* book would be totally shameless, I won't mention it again.

Other GUI-building technologies do exist, with the **PyGTK**, **PyKDE**, **wxPython**, and **PyQT** toolkits coming up in conversation more than most. Be warned, however, that most of these toolkits target Python 2, although support for Python 3 is on its way. Search the Web for any of the project names to learn more.

Oh, look: it's one of the
GUIs created in "Head First
Programming"...and yes, I said
I wouldn't mention THAT
book again, but isn't this GUI
beautiful? ☺



#9: Stuff to avoid

When it comes to stuff to avoid when using Python, there's a very short list. A recent tweet on *Twitter* went something like this:



Threads do indeed exist in Python but should be **avoided** where possible.

This has *nothing* to do with the quality of Python's threading library and *everything* to do with Python's implementation, especially the implementation known as **CPython** (which is more than likely the one you're running now). Python is implemented using a technology known as the **Global Interpreter Lock (GIL)**, which enforces a restriction that Python can only ever run on a single interpreter process, even in the presence of multiple processors.

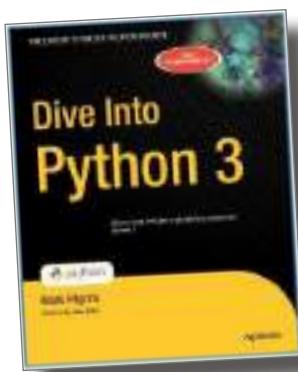
What all this means to you is that your beautifully designed and implemented program that uses threads will never run faster on multiple processors even if they exist, because *it can't use them*. Your threaded application will run **serially** and, in many cases, run *considerably slower* than if you had developed the same functionality without resorting to threads.

Main message: **don't use threads** with Python until the GIL restriction is removed...*if it ever is*.

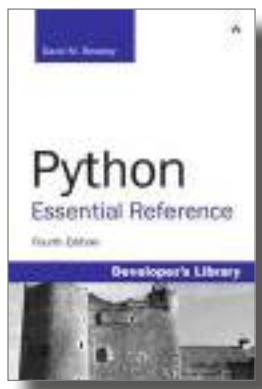


#10: Other books

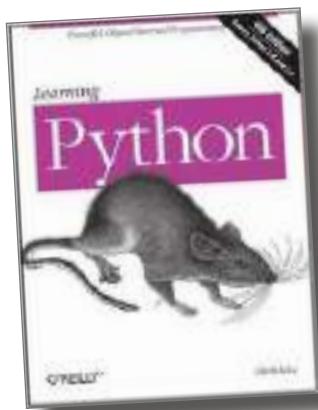
There are lots of great books that cover Python in general, as well as specifically within a particular problem domain. Here is a collection of my favorite Python books, which we have no hesitation in recommending to you.



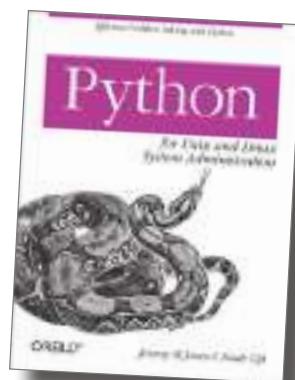
Includes a great case study involving the porting of a complex Python 2 module to Python 3.



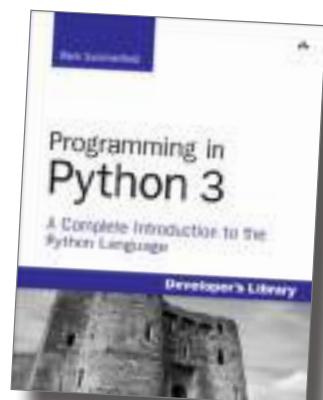
The best desktop reference on the market



At 1,200 pages, this is the definitive language reference for Python: it's got everything in it!



If you are a sysadmin, then this is the Python book for you.



Includes some big examples with big technology: XML, parsing and advanced language features.



Index

Symbols and Numbers

404 Error, from web server 242
405 Error, from web server 378
>>> (chevron, triple) IDLE prompt 4
:(colon)
 in for loop 16
 in function definition 29
 in if statement 20
,(comma) separating list items 7
{(curly braces)
 creating dictionaries 180
 creating sets 166
=(equal sign)
 assignment operator 7
 in function argument definition 63
(...) (parentheses)
 enclosing function arguments 29
 enclosing immutable sets 91
+(plus sign) addition or concatenation operator 138
#(pound sign) preceding one-line comments 38
@property decorator 250, 253, 285
?(question mark) parameter placeholder 321, 350
“...” or ‘...’ (quotes) enclosing each list item 7
“””...””” or “...” (quotes, triple) enclosing comments 37
;(semicolon) separating statements on one line 38
[...] (square brackets)

A

“a” access mode 110
access modes for files 110, 133

addition operator (+) 138
Alt-N keystroke, IDLE 5
Alt-P keystroke, IDLE 5
AndFTP app 288–289
Android apps
 accepting input from 278–282, 295, 304–307
 converting from Python code 424–430
 creating 274–277, 281–282
 data for. *See* JSON data interchange format
 integrating with SQLite 342–348
 running on phone 288–289
 scripting layer for. *See* SL4A
 troubleshooting 277
Android emulator
 installing and configuring 260–262
 running scripts on 264–265, 272–273, 283
Android Market 288
Android Virtual Device. *See* AVD
anonymous functions 439
append() method, lists 10, 14
apps. *See* Android apps; webapps
app.yaml file 356, 395
arguments for functions
 adding 52, 66–68
 optional 63–64, 134
arrays
 associative. *See* dictionaries
 similarity to lists 9–10, 17
as keyword 119, 138
assignment operator (=) 7
associative arrays. *See* dictionaries
attributes, class 190, 194, 212
authorization, user 389–393
AVD (Android Virtual Device) 261, 291

B

“batteries included” 32
BIFs. *See* built-in functions
BigTable technology 354, 359
blue text in IDLE 4
books
 Dive Into Python 3 (CreateSpace) 445
 Head First HTML with CSS & XHTML (O’Reilly)
 374
 Head First Programming (O’Reilly) 443
 Head First SQL (O’Reilly) 313
 Learning Python (O’Reilly) 445
 Mastering Regular Expressions (O’Reilly) 440
 Programming in Python 3 (Addison-Wesley Professional) 445
 Python Essential Reference (Addison-Wesley Professional) 445
 Python for Unix and Linux System Administration (O’Reilly) 445
braces. *See* curly braces
brackets, regular. *See* parentheses
brackets, square. *See* square brackets
BSD, running CGI scripts on 239
build folder 42
built-in functions (BIFs). *See also* specific functions
 displayed as purple text in IDLE 4
 help on 21
 importing of, not needed 55
 namespace for 55
 number of, in Python 21
 __builtins__ namespace 55, 71

C

cascading style sheet (CSS) 374–375
case sensitivity of identifiers 17
cgi-bin folder 234, 235
CGI (Common Gateway Interface) scripts 217, 235, 243, 253. *See also* WSGI
 location of 234, 235
 running 239
 running from Android 264–265, 272–273, 283

sending data to 300–303
tracking module for 248–249
troubleshooting 242, 247–250
writing 236–238, 244–246
writing for Android. *See* SL4A
cgi library 300
cgitb module 248–249, 253
chaining
 functions 146, 172
 methods 142, 172
chevron, triple (>>>) IDLE prompt 4
chmod command 239, 253
classes 189–191
 attributes of 190, 194, 212
 benefits of 189
 converting data to dictionary 285–286
 defining 190–193, 194, 195–196
 inherited 204–209, 212
 instances of 190, 191, 194, 195–196
 metaclasses 439
 methods of 190, 195–196, 198–200
 in modules 209, 212
class keyword 191, 212
close() method, database connection 315, 350
close() method, files 75, 103
code editors 35, 436. *See also* IDLE
colon (:)
 in for loop 16
 in function definition 29
 in if statement 20
comma (,) separating list items 7
comments 37–38
commit() method, database connection 315, 350
Common Gateway Interface scripts. *See* CGI scripts
comprehension, list 154–159, 172, 409–411, 432
concatenation operator (+) 138
conditional list comprehension 409–411, 432
conditions. *See* if/else statement
connection, database
 closing 314, 315
 creating 314, 315

`connect()` method, `sqlite3` 315, 350

context management protocol 120

Controller, in MVC pattern 221

- for GAE webapps 359, 370–373

- for webapps 234–238, 244–246

“copied” sorting 144, 145–146, 172

`CREATE TABLE` statement, SQL 317, 319–320

CSS (cascading style sheet) 374–375

CSV format, converting to Python data types 401–405

curly braces (`{}`)

- creating dictionaries 180

- creating sets 166

`cursor()` method, database connection 315, 350

custom code 131

custom exceptions 439

D

data

- for Android apps. *See* JSON data interchange format

- bundling with code. *See* classes

- duplicates in, removing 161–163, 166–167

- external. *See* database management system; files

- for GAE webapps. *See* datastore, for GAE

- nonuniform, cleaning 148–153

- race conditions with 309–310

- Robustness Principle for 384–387

- searching for closest match 416–417

- sending to web server 275, 291

- sorting 144–147, 172

- storing. *See* persistence

- transforming, list comprehensions for 154–159

database API 314–315, 350

database management system 312

- closing connection to 314, 315

- commit changes to data 314, 315, 350

- connecting to 314, 315

- cursor for, manipulating data with 314, 315

- designing 316–318

- inserting data into 321, 324, 348

- integrating with Android apps 342–348

- integrating with webapps 327–341

- managing and viewing data in 326

- process for interacting with 314–315

- querying 322, 332–333

rollback changes to data 314, 315, 350

schema for 317

SQLite for. *See* SQLite

tables in 317, 319–320, 350

data folder 234

data interchange format. *See* JSON data interchange format

data objects. *See also* specific data objects

- getting next item from 54

- ID for 54

- length of, determining 32

- names of. *See* identifiers

datastore, for GAE 359–360, 380–383, 384–387, 395

data types

- converting CSV data into 401–405

- converting strings to integers 54

- in datastore 360

- immutable 91, 103, 116, 138

- for JSON 285

- for list items 8, 12

date and time data

- format compatibility issues 418–423

- property type for 362, 384–385

`db.Blob()` type 360

`db.DateProperty()` type 360

`db.IntegerProperty()` type 360

`db.StringProperty()` type 360, 385

`db.TimeProperty()` type 360

`db.UserProperty()` type 360, 390

decision statement. *See* if/else statement

decorators, function 439

`def` keyword 29, 191, 212

`dialogCreateAlert()` method, Android API 274, 276, 280

`dialogGetInput()` method, Android API 295, 304–306

`dialogGetResponse()` method, Android API 274, 276, 278, 280

`dialogGetSelectedItems()` method, Android API 278, 280

`dialogSetItems()` method, Android API 279, 280

`dialogSetNegativeButtonText()` method, Android API 274, 276

`dialogSetPositiveButtonText()` method, Android API 274, 276, 280

- dialogSetSingleChoiceItems() method, Android API 274, 276
dialogShow() method, Android API 276, 280
dict() factory function 180, 212
dictionaries 178–182, 212
 accessing items in 180, 212
 compared to lists 179
 converting class data to 285–286
 creating 180, 182, 186
 dictionaries within 407–409
 keys for 178, 180, 212
 populating 180, 212
 reading CSV data into 403–404
 values of 178, 180, 212
dir() command 225
directory structure. *See* folder structure
dist folder 42
distribution
 creating 40–42
 updating 60–61, 65
 uploading to PyPI 48
Dive Into Python 3 (CreateSpace) 445
djangoforms.ModelForm class 368
Django Project
 Form Validation 368–369, 395
 templates 363–366, 395
doctest framework 438
documentation for Python 3 3, 80, 103
dot notation 10, 194, 196
double quotes. *See* quotes
dump() function, pickle 133–134, 138
dumps() function, json 269, 272, 281, 291
dynamic content 216, 217
- ## E
- Eclipse editor 35
editors 35, 436. *See also* IDLE
elif keyword 108. *See also* if/else statement
else keyword. *See* if/else statement
emacs editor 35, 436
- enable() function, cgitb 248, 253
end_form() function, yate 231, 233
entities, in datastore 360, 395
enumerate() built-in function 54
environ dictionary 300, 350
equal sign (=)
 assignment operator 7
 in function argument definition 63
errors. *See* exception handling; troubleshooting
exception handling 88–95, 103. *See also* troubleshooting
 benefits of 95, 100
 closing files after 114–115, 120–123
 custom exceptions 439
 defining with try/except statement 89, 91–94
 ignoring found errors 93–94
 IndexError exception 17
 IOError exception 103, 112–114, 117–119
 for missing files 96–98
 NameError exception 44, 118
 PickleError exception 133–134
 specific errors, checking for 101–102
 specific errors, details about 117–119
 TypeError exception 56–57, 116, 247–249, 283–285
 ValueError exception 78–79, 81–82, 103
exception objects 119, 138
except keyword. *See* try/except statement
execute() method, cursor 315, 322, 324, 350
extend() method, lists 10
- ## F
- F5 key, IDLE 39, 44, 49, 71
factory functions 166
favicon.ico file, for webapp 234
fetchall() method, cursor 322
fetchmany() method, cursor 322
fetchone() method, cursor 322
FieldStorage() method, cgi 244, 253, 296, 300, 350
files. *See also* persistence
 access modes for 110, 133
 appending data to 110
 checking for existence of 118

closing 75, 110
 closing after exception 114–115, 120–123
 CSV format, converting to Python data types 401–405
 exceptions involving, determining type of 117–119
 flushing 110
 missing, exception handling for 96–98
 opening 75, 109–110
 opening in binary access mode 133
 reading data from 75–78, 142–143
 rewinding 76
 splitting lines in 77–78
 writing 110–113
 writing, custom formats for 126–130
 writing, default format for 124–125
 writing, pickle library for. *See* pickle library
 finally keyword 115, 138
 find() method, strings 84–86, 103
 Firefox, SQLite Manager for 326
 folder structure
 for distribution 42
 for GAE 356, 370
 for webapps 234
 for loop 15–17, 32
 compared to list comprehension 432
 nesting 19–22
 forms, HTML 295
 creating from template 296–299
 Form Validation Framework for 368–369
 input restrictions for 376–377, 384–387
 sending data to CGI scripts 300–303
 stylesheets for 374–375
 Form Validation Framework 368–369, 395
 405 Error, from web server 378
 404 Error, from web server 242
 Friedl, Jeff (author, *Mastering Regular Expressions*) 440
 from statement 46, 49
 functional programming concepts 157
 function decorators 439
 functions
 adding arguments to 52, 66–68
 anonymous 439
 built-in. *See* built-in functions (BIFs)

chaining 146, 172
 creating 28–30, 170–171
 optional arguments for 63–64, 134
 recursive 31
 sharing. *See* modules

G

GAE (Google App Engine) 354
 configuration and setup for 356–357
 controller code for 370–373
 data modeling with 360–362
 datastore for 359, 380–383, 384–387, 395
 deploying webapps to Google cloud 391
 folder structure for 356, 370
 form generation for 368–369
 form input restrictions for 376–377, 384–387
 form stylesheets for 374–375
 MVC pattern used by 359
 SDK for, downloading 355
 troubleshooting 378
 user authorization for 389–393
 view for, designing 363–369
 GAE Launcher 357, 391, 395
 garbage collection 116
 generators 439
 get() method, GAE 370, 379, 395
 GET web request 370
 GIL (Global Interpreter Lock) 444
 glob module 237, 253
 Google App Engine. *See* GAE
 Google BigTable technology 354, 359
 GQL (Google Query Language) API 359
 green text in IDLE 4
 GUI (graphical user interface), building 443

H

hashes. *See* dictionaries
 header() function, yate 231, 233
 Head First HTML with CSS & XHTML (O'Reilly) 374
 Head First Programming (O'Reilly) 443

Head First SQL (O'Reilly) 313

help() built-in function 80, 103

HTML

generating for webapp interface 230–231

learning 226

templates for, with Django 363–366

HTML forms. *See* forms, HTML

HTTP server 235

http.server module 235, 253

I

id() built-in function 54

IDE 436. *See also* IDLE

identifiers 7, 17, 32

IDLE 3–5, 32

colored syntax used in 4

indenting enforced in 4

preferences, setting 5

prompt in (>>>) 4

recalling and editing code statements 5

running or loading code in 39, 44, 49, 71

TAB completion 5

if/else statement 20, 32

elif keyword 108

in list comprehension 432

negating conditions in 86, 103

images folder 234

immutable data types 138

lists 91, 103, 116

numbers 116

strings 116

import statement 43, 46, 49, 71

include_footer() function, *yate* 230, 232, 233

include_header() function, *yate* 230, 232

indentation rules

enforced in IDLE 4

for loops 16

for function definitions 29

for if statement 20

IndexError exception 17

index.html file, for webapp 234

inherited classes 204–209, 212

__init__() method 191, 212

in operator 16, 118, 138

“in-place” sorting 144, 145, 172

input

from Android apps 278–282, 295, 304–307

HTML forms for. *See* forms, HTML

from keyboard after screen prompt 413–414, 432

input() built-in function 413–414, 432

insert() method, lists 10, 14

INSERT statement, SQL 321, 324, 348

instances of classes 190, 191, 194, 195–196

int() built-in function 54

integers, converting strings to 54

interface. *See* View, in MVC pattern

IOError exception 103, 112–114, 117–119

I/O (input/output), handling. *See* files

isinstance() built-in function 20–22, 32

iterations

for loop 15–17, 19–22, 32

generating with range() function 54–56

while loop 16–17

J

JSON data interchange format 266–267, 291

API for, using 269–272

browser differences with 272

data types supported by 285

incompatibility with pickle data objects 284–285

K

KDevelop IDE 436

keys, in dictionary 178, 180, 212

keywords, displayed as orange text in IDLE 4

L

lambda expression 439
 Learning Python (O'Reilly) 445
`len()` built-in function 10, 32
`lib` folder 42
 Linux
 code editors for 35
 GAE log messages on 378
 IDEs for 436
 installing Python 3 on 3
 running CGI scripts on 239, 272
 running GAE Launcher on 357
 transferring files to Android device 288
`list()` built-in function 54
 list comprehension 154–159, 172, 409–411, 432
 lists 32. *See also* data objects
 adding items to 10–14
 bounds checking for 17
 classes inherited from 204–208
 compared to dictionaries 179
 compared to sets 167
 creating 6–7, 54
 data types in 8, 12
 duplicates in, removing 161–163
 extracting specific item from 175–176
 getting next item from 54
 identifiers for 7
 immutable 91, 103, 116
 iterating 15–17, 157
 length of, determining 10, 32
 methods for 10
 nested, checking for 20–22
 nested, creating 18–19
 nested, handling 23–25, 28–31
 numbered, creating 54
 reading CSV data into 403–404
 removing items from 10
 similarity to arrays 9–10, 17
 slice of 160, 172
`load()` function, pickle 133, 138
`loads()` function, json 269, 276, 280, 291

`locals()` built-in function 118, 138
 loops. *See* iterations

M

Mac OS X
 code editors for 35
 GAE log messages on 378
 IDEs for 436
 installing Python 3 on 3
 running CGI scripts on 239, 272
 running GAE Launcher on 357
 transferring files to Android device 288
`__main__` namespace 45
 MANIFEST file 42
 mappings. *See* dictionaries
 Mastering Regular Expressions (O'Reilly) 440
 metaclasses 439
 methods 190. *See also* specific methods
 chaining 142, 172
 for classes 195–196, 198–200
 creating 212
 results of, as attributes 250, 253
 self argument of 212
 ModelForm class, djangoforms 368
 Model, in MVC pattern 221
 for GAE webapps 359, 360–362
 for webapps 222–225
 Model-View-Controller pattern. *See* MVC pattern
 modules 34–36, 71
 adding functionality to 50–52
 classes in 209, 212
 creating 35
 distribution for, creating 40–42
 distribution for, updating 60–61, 65
 distribution for, uploading to PyPI 48
 importing 43–44, 46
 loading in IDLE 39, 49, 71
 locations for 38, 49
 namespaces for 45–46, 71
 in Python Standard Library 36
 third-party 36

Monty Python 17
multiple inheritance 209
MVC (Model-View-Controller) pattern 221, 232, 253, 359
 Controller 234–238, 244–246, 370–373
 Model 222–225, 360–362
 View 226–233, 363–369

N

NameError exception 44, 118
names. *See* identifiers
namespaces 45–46, 71
next() built-in function 54
NoSQL 359, 442
NotePad editor 35
not in operator 161–162
not keyword 86, 103
numbered lists 54

O

object relational mapper. *See* ORM (object relational mapper)
objects. *See* data objects
open() built-in function 75, 103, 109–110
orange text in IDLE 4
ORM (object relational mapper) 442
os module 76, 300, 350

P

para() function, yate 231, 233
parentheses (...)
 enclosing function arguments 29
 enclosing immutable lists 91
pass statement 93, 103
persistence 105
 pickle library for 132–137
 reading data from files 222–224
 writing data to files 110–113, 222–224
PickleError exception 133–134

pickle library 132–137, 138
 data modeling using 222–224
 incompatibility with JSON data types 284–285
 transferring data to a database 321–325
plus sign (+) addition or concatenation operator 138
pop() method, lists 10, 175–176
post() method, GAE 379–383, 395
POST web request 379
pound sign (#) preceding one-line comments 38
print() built-in function 10, 32, 124–125
 disabling automatic new-line for 56, 71
 displaying TAB character with 56
 writing to a file 110, 128, 138
Programming in Python 3 (Addison-Wesley Professional) 445
properties, in datastore 360, 395
@property decorator 250, 253, 285
purple text in IDLE 4
put() method, GAE 395
.pyc file extension 42, 49
.py file extension 35
PyPI (Python Package Index) 36
 registering on website 47
 uploading distributions to 48
 uploading modules to 209
Python 2
 compared to Python 3 17
 raw_input() built-in function 432
 running on Android smartphones 258–259, 291
 using with Google App Engine 355
Python 3
 compared to Python 2 17
 documentation for 3, 80, 103
 editors for 35, 436
 installing 3
 interpreter for. *See* IDLE
 learning 445
python3 command
 building a distribution 41
 checking for Python version 3
 installing a distribution 41
 uploading a new distribution 68

Python Essential Reference (Addison-Wesley Professional)
445
Python for Unix and Linux System Administration
(O'Reilly) 445
Python, Monty 17
Python Package Index. *See* PyPI
Python Standard Library 36

Q

querying a database 322, 332–333
question mark (?) parameter placeholder 321, 350
quotes ("..." or '...') enclosing each list item 7
quotes, triple (""""..."""" or """...""") enclosing comments 37

R

"r" access mode 110
race conditions 309–310
radio_button() function, yate 231, 233
range() built-in function 54–56, 71
raw_input() built-in function 432
readline() method, files 76, 103, 142
recursion 31
regular brackets. *See* parentheses
regular expressions 440
re module 440
remove() method, lists 10
render() function, template 364, 366
Robustness Principle 384–387
rollback() method, database connection 315, 350
runtime errors 88. *See also* exception handling; troubleshooting

S

schema, database 317
scoping of variables 437
Scripting Layer for Android. *See* SL4A
scripts. *See* CGI scripts; SL4A

sdist command 41
seek() method, files 76, 103
SELECT/OPTION, HTML tag 376
SELECT statement, SQL 322, 332–333
self argument 192–193, 212
self.request object 379, 395
self.response object 372, 379, 395
semicolon (;) separating statements on one line 38
set() built-in function 166, 172
sets 166, 167, 172
setup() built-in function 40
setup.py file 40, 71
single quotes. *See* quotes
SL4A (Scripting Layer for Android) 258, 291
 adding Python to 263
 Android apps, creating 274–277
 automatic rotation mode, setting 264
 documentation for 274
 installing 262
 Python versions supported 258–259
slice of a list 160, 172
smartphones, apps on. *See* Android apps
sorted() built-in function 144–147, 153, 158, 172
sort() method, lists 144–145, 153, 172
split() method, strings 77–78, 80–81, 103, 142
SQL Alchemy 442
SQLite 313, 350
 closing connection to 314, 315
 committing data to 314, 315
 connecting to 314, 315
 cursor for, manipulating data with 314, 315
 designing database 316–318
 inserting data into 321, 324, 348
 integrating with Android apps 342–348
 integrating with webapps 327–341
 managing data in 326
 process for interacting with 314–315
 querying 322, 332–333
 rollback changes to data 314
 schema for database 317
 tables in, creating 319–320

sqlite3 command 326
sqlite3 library 313, 315, 350
SQLite Manager, for Firefox 326
SQL (Structured Query Language) 313, 350. *See also* NoSQL; SQLite; ORM
square brackets ([...])
 accessing dictionary items 180, 212
 accessing specific list items 9, 18
 enclosing all list items 7, 18
standard error (sys.stderr) 248, 291
standard input (sys.stdin) 291
Standard Library, Python 36
standard output (sys.stdout) 126–128, 291
start_form() function, yate 231, 233
start_response() function, yate 230, 232
static content 216, 217
static folder 370
str() built-in function 119, 138
strings
 concatenating 138
 converting other objects to 119
 converting to integers 54
 displayed as green text in IDLE 4
 finding substrings in 84–86
 immutable 116
 sorting 148
 splitting 77–78, 80–81
 substitution templates for 230, 253
strip() method, strings 108, 138, 142
Structured Query Language. *See* SQL
stylesheets for HTML forms 374–375
suite 16, 29, 32
sys module 291
sys.stdout file 126–128, 138

T

TAB character, printing 56
TAB completion, IDLE 5
tables, database 317, 319–320, 350
target identifiers, from split() method 77, 91
.tar.gz file extension 42
Template class 230, 253
template module 364
templates folder 234, 370
templates for GAE 363–366, 395
testing code 438
TextMate editor 35, 436
third-party modules 36
threads 444
time data
 format compatibility issues 418–423
 property type for 362, 384–385
time module 419, 432
Tk Interface (tkinter) 443
traceback 88, 103. *See also* exception handling; troubleshooting
tracing code 58–59
triple chevron (>>>) IDLE prompt 4
triple quotes ("””...””” or “”...””) enclosing comments 37
troubleshooting. *See also* exception handling
 404 Error, from web server 242
 405 Error, from web server 378
 Android apps 277
 GAE webapps 378
 testing code 438
 tracing code 58–59
try/except statement 89, 93–94, 101–102, 103
 finally keyword for 115, 138
 with statement and 120–123
tuples 91, 103, 116
TypeError exception 56–57, 116, 247–249, 283–285

U

u_list() function, yate 231, 233
 unittest module 438
 urlencode() function, urllib 291
 urllib2 module 291
 urllib module 291
 urlopen() function, urllib2 291
 user authorization 389–393
 user input. *See* forms, HTML; input
 UserProperty() type, db 390

V

ValueError exception 78–79, 81–82, 103
 values, part of dictionary 178, 180, 212
 variables, scope of 437
 vi editor 35, 436
 View, in MVC pattern 221
 for GAE webapps 359, 363–369
 for webapps 226–233
 vim editor 436

W

“w” access mode 110
 “w+” access mode 110
 “wb” access mode 133

webapps 215–217, 253
 controlling code for 221, 234–238, 244–246
 data modeling for 221, 222–225
 designing with MVC 221
 design requirements for 218–220
 directory structure for 234
 Google App Engine for. *See* GAE
 input data, sending to CGI scripts 300–303
 input forms for. *See* forms, HTML
 SQLite used with 327–341
 view for 221, 226–233
 Web-based applications. *See* webapps
 web frameworks 441. *See also* CGI; WSGI
 web request 216, 253, 395
 web response 216–217, 253, 395
 web server 216–217, 235
 Web Server Gateway Interface (WSGI) 356, 370. *See also* CGI scripts
 while loop 16–17, 55
 WingIDE editor 35
 WingWare Python IDE 436
 with statement 120–123, 138
 WSGI (Web Server Gateway Interface) 356, 370. *See also* CGI scripts

Y

yate (Yet Another Template Engine) library 226–233
 yield expression 439

PERCEPTIONS OF A RENEGADE MIND

DAVID DICKIE

PERCEPTIONS OF A RENEGADE MIND



DAVID ICKE

**PERCEPTIONS
OF A
RENEGADE
MIND**

ickonic
publishing

First published in July 2021.



**New Enterprise House
St Helens Street
Derby
DE1 3GY
UK**

email: gareth.icke@davidicke.com

Copyright © 2021 David Icke

No part of this book may be reproduced in any form without permission from the Publisher, except for the quotation of brief passages in criticism

Cover Design: Gareth Icke
Book Design: Neil Hague

**British Library Cataloguing-in
Publication Data**
A catalogue record for this book is
available from the British Library

eISBN 978-18384153-1-0

PERCEPTIONS
OF A
RENEGADE
MIND



DAVID ICKE

Dedication:

To *Freeeeeedom!*

ICKONIC



THE ALTERNATIVE

NEW. DIFFERENT. REVOLUTIONARY

HUNDREDS OF CUTTING EDGE DOCUMENTARIES,
FEATURE FILMS, SERIES & PODCASTS.

SIGN UP NOW AT ICKONIC.COM

THE LIFE STORY OF DAVID ICKE
RENEGADE
THE FEATURE LENGTH FILM



AVAILABLE NOW AT DAVIDICKE.COM

Renegade:

Adjective

'Having rejected tradition: Unconventional.'

Merriam-Webster Dictionary

Acquiescence to tyranny is the death of the spirit

You may be 38 years old, as I happen to be. And one day, some great opportunity stands before you and calls you to stand up for some great principle, some great issue, some great cause. And you refuse to do it because you are afraid

... You refuse to do it because you want to live longer ...

You're afraid that you will lose your job, or you are afraid that you will be criticised or that you will lose your popularity, or you're afraid that somebody will stab you, or shoot at you or bomb your house; so you refuse to take the stand.

Well, you may go on and live until you are 90, but you're just as dead at 38 as you would be at 90. And the cessation of breathing in your life is but the belated announcement of an earlier death of the spirit.

Martin Luther King

**How the few control the many and always have – the many do
whatever they're told**

'Forward, the Light Brigade!'
Was there a man dismayed?
Not though the soldier knew
 Someone had blundered.
Theirs not to make reply,
Theirs not to reason why,
Theirs but to do and die.
 Into the valley of Death
 Rode the six hundred.

Cannon to right of them,
Cannon to left of them,
Cannon in front of them
 Volleyed and thundered;
Stormed at with shot and shell,
 Boldly they rode and well,
 Into the jaws of Death,
 Into the mouth of hell
 Rode the six hundred

Alfred Lord Tennyson (1809-1892)

The mist is lifting slowly
I can see the way ahead
And I've left behind the empty streets
That once inspired my life
And the strength of the emotion
Is like thunder in the air
'Cos the promise that we made each other
Haunts me to the end

The secret of your beauty
And the mystery of your soul
I've been searching for in everyone I meet
And the times I've been mistaken
It's impossible to say
And the grass is growing
Underneath our feet

The words that I remember
From my childhood still are true
That there's none so blind
As those who will not see
And to those who lack the courage
And say it's dangerous to try
Well they just don't know
That love eternal will not be denied

I know you're out there somewhere
Somewhere, somewhere
I know you're out there somewhere

Somewhere you can hear my voice
I know I'll find you somehow
Somehow, somehow
I know I'll find you somehow
And somehow I'll return again to you

The Moody Blues

Are you a gutless wonder - or a Renegade Mind?

Monuments put from pen to paper,
Turns me into a gutless wonder,
And if you tolerate this,
Then your children will be next.
Gravity keeps my head down,
Or is it maybe shame ...

Manic Street Preachers

Rise like lions after slumber
In unvanquishable number.
Shake your chains to earth like dew
Which in sleep have fallen on you.
Ye are many – they are few.

Percy Shelley

Contents

CHAPTER 1	'I'm thinking' – Oh, but <i>are you?</i>
CHAPTER 2	Renegade perception
CHAPTER 3	The Pushbacker sting
CHAPTER 4	'Covid': The calculated catastrophe
CHAPTER 5	There <i>is no</i> 'virus'
CHAPTER 6	Sequence of deceit
CHAPTER 7	War on your mind
CHAPTER 8	'Reframing' insanity
CHAPTER 9	We must have it? So what is it?
CHAPTER 10	Human 2.0
CHAPTER 11	Who controls the Cult?
CHAPTER 12	Escaping Wetiko
POSTSCRIPT	
APPENDIX	Cowan-Kaufman-Morell Statement on Virus Isolation
BIBLIOGRAPHY	
INDEX	

CHAPTER ONE

I'm thinking' – Oh, but *are* you?

Think for yourself and let others enjoy the privilege of doing so too
Voltaire

French-born philosopher, mathematician and scientist René Descartes became famous for his statement in Latin in the 17th century which translates into English as: 'I think, therefore I am.'

On the face of it that is true. Thought reflects perception and perception leads to both behaviour and self-identity. In that sense 'we' are what we think. But who or what is doing the thinking and is thinking the only route to perception? Clearly, as we shall see, 'we' are not always the source of 'our' perception, indeed with regard to humanity as a whole this is rarely the case; and thinking is far from the only means of perception. Thought is the village idiot compared with other expressions of consciousness that we all have the potential to access and tap into. This has to be true when we *are* those other expressions of consciousness which are infinite in nature. We have forgotten this, or, more to the point, been manipulated to forget.

These are not just the esoteric musings of the navel. The whole foundation of human control and oppression is control of perception. Once perception is hijacked then so is behaviour which is dictated by perception. Collective perception becomes collective behaviour and collective behaviour is what we call human society. Perception is all and those behind human control know that which is

why perception is the target 24/7 of the psychopathic manipulators that I call the Global Cult. They know that if they dictate perception they will dictate behaviour and collectively dictate the nature of human society. They are further aware that perception is formed from information received and if they control the circulation of information they will to a vast extent direct human behaviour.

Censorship of information and opinion has become globally Nazi-like in recent years and never more blatantly than since the illusory ‘virus pandemic’ was triggered out of China in 2019 and across the world in 2020. Why have billions submitted to house arrest and accepted fascistic societies in a way they would have never believed possible? Those controlling the information spewing from government, mainstream media and Silicon Valley (all controlled by the same Global Cult networks) told them they were in danger from a ‘deadly virus’ and only by submitting to house arrest and conceding their most basic of freedoms could they and their families be protected. This monumental and provable lie became the *perception* of the billions and therefore the *behaviour* of the billions. In those few words you have the whole structure and modus operandi of human control. Fear is a perception – False Emotion Appearing Real – and fear is the currency of control. In short ... get them by the balls (or give them the impression that you have) and their hearts and minds will follow. Nothing grips the dangly bits and freezes the rear-end more comprehensively than fear.

World number 1

There are two ‘worlds’ in what appears to be one ‘world’ and the prime difference between them is knowledge. First we have the mass of human society in which the population is maintained in coldly-calculated ignorance through control of information and the ‘education’ (indoctrination) system. That’s all you really need to control to enslave billions in a perceptual delusion in which what are perceived to be *their* thoughts and opinions are ever-repeated mantras that the system has been downloading all their lives through ‘education’, media, science, medicine, politics and academia

in which the personnel and advocates are themselves overwhelmingly the perceptual products of the same repetition. Teachers and academics in general are processed by the same programming machine as everyone else, but unlike the great majority they never leave the ‘education’ program. It gripped them as students and continues to grip them as programmers of subsequent generations of students. The programmed become the programmers – the programmed programmers. The same can largely be said for scientists, doctors and politicians and not least because as the American writer Upton Sinclair said: ‘It is difficult to get a man to understand something when his salary depends upon his not understanding it.’ If your career and income depend on thinking the way the system demands then you will – bar a few free-minded exceptions – concede your mind to the Perceptual Mainframe that I call the Postage Stamp Consensus. This is a tiny band of perceived knowledge and possibility ‘taught’ (downloaded) in the schools and universities, pounded out by the mainstream media and on which all government policy is founded. Try thinking, and especially speaking and acting, outside of the ‘box’ of consensus and see what that does for your career in the Mainstream Everything which bullies, harasses, intimidates and ridicules the population into compliance. Here we have the simple structure which enslaves most of humanity in a perceptual prison cell for an entire lifetime and I’ll go deeper into this process shortly. Most of what humanity is taught as fact is nothing more than programmed belief. American science fiction author Frank Herbert was right when he said: ‘Belief can be manipulated. Only knowledge is dangerous.’ In the ‘Covid’ age belief is promoted and knowledge is censored. It was always so, but never to the extreme of today.

World number 2

A ‘number 2’ is slang for ‘doing a poo’ and how appropriate that is when this other ‘world’ is doing just that on humanity every minute of every day. World number 2 is a global network of secret societies and semi-secret groups dictating the direction of society via

governments, corporations and authorities of every kind. I have spent more than 30 years uncovering and exposing this network that I call the Global Cult and knowing its agenda is what has made my books so accurate in predicting current and past events. Secret societies are secret for a reason. They want to keep their hoarded knowledge to themselves and their chosen initiates and to hide it from the population which they seek through ignorance to control and subdue. The whole foundation of the division between World 1 and World 2 is *knowledge*. What number 1 knows number 2 must not. Knowledge they have worked so hard to keep secret includes (a) the agenda to enslave humanity in a centrally-controlled global dictatorship, and (b) the nature of reality and life itself. The latter (b) must be suppressed to allow the former (a) to prevail as I shall be explaining. The way the Cult manipulates and interacts with the population can be likened to a spider's web. The 'spider' sits at the centre in the shadows and imposes its will through the web with each strand represented in World number 2 by a secret society, satanic or semi-secret group, and in World number 1 – the world of the seen – by governments, agencies of government, law enforcement, corporations, the banking system, media conglomerates and Silicon Valley ([Fig 1](#) overleaf). The spider and the web connect and coordinate all these organisations to pursue the same global outcome while the population sees them as individual entities working randomly and independently. At the level of the web governments *are* the banking system *are* the corporations *are* the media *are* Silicon Valley *are* the World Health Organization working from their inner cores as one unit. Apparently unconnected countries, corporations, institutions, organisations and people are on the *same team* pursuing the same global outcome. Strands in the web immediately around the spider are the most secretive and exclusive secret societies and their membership is emphatically restricted to the Cult inner-circle emerging through the generations from particular bloodlines for reasons I will come to. At the core of the core you would get them in a single room. That's how many people are dictating the direction of human society and its transformation

through the ‘Covid’ hoax and other means. As the web expands out from the spider we meet the secret societies that many people will be aware of – the Freemasons, Knights Templar, Knights of Malta, Opus Dei, the inner sanctum of the Jesuit Order, and such like. Note how many are connected to the Church of Rome and there is a reason for that. The Roman Church was established as a revamp, a rebranding, of the relocated ‘Church’ of Babylon and the Cult imposing global tyranny today can be tracked back to Babylon and Sumer in what is now Iraq.



Figure 1: The global web through which the few control the many. (Image Neil Hague.)

Inner levels of the web operate in the unseen away from the public eye and then we have what I call the cusp organisations located at the point where the hidden meets the seen. They include a series of satellite organisations answering to a secret society founded in London in the late 19th century called the Round Table and among them are the Royal Institute of International Affairs (UK, founded in 1920); Council on Foreign Relations (US, 1921); Bilderberg Group (worldwide, 1954); Trilateral Commission (US/worldwide, 1972); and the Club of Rome (worldwide, 1968) which was created to exploit environmental concerns to justify the centralisation of global power to ‘save the planet’. The Club of Rome instigated with others the human-caused climate change hoax which has led to all the ‘green

new deals' demanding that very centralisation of control. Cusp organisations, which include endless 'think tanks' all over the world, are designed to coordinate a single global policy between political and business leaders, intelligence personnel, media organisations and anyone who can influence the direction of policy in their own sphere of operation. Major players and regular attenders will know what is happening – or some of it – while others come and go and are kept overwhelmingly in the dark about the big picture. I refer to these cusp groupings as semi-secret in that they can be publicly identified, but what goes on at the inner-core is kept very much 'in house' even from most of their members and participants through a fiercely-imposed system of compartmentalisation. Only let them know what they need to know to serve your interests and no more. The structure of secret societies serves as a perfect example of this principle. Most Freemasons never get higher than the bottom three levels of 'degree' (degree of knowledge) when there are 33 official degrees of the Scottish Rite. Initiates only qualify for the next higher 'compartment' or degree if those at that level choose to allow them. Knowledge can be carefully assigned only to those considered 'safe'. I went to my local Freemason's lodge a few years ago when they were having an 'open day' to show how cuddly they were and when I chatted to some of them I was astonished at how little the rank and file knew even about the most ubiquitous symbols they use. The mushroom technique – keep them in the dark and feed them bullshit – applies to most people in the web as well as the population as a whole. Sub-divisions of the web mirror in theme and structure transnational corporations which have a headquarters somewhere in the world dictating to all their subsidiaries in different countries. Subsidiaries operate in their methodology and branding to the same centrally-dictated plan and policy in pursuit of particular ends. The Cult web functions in the same way. Each country has its own web as a subsidiary of the global one. They consist of networks of secret societies, semi-secret groups and bloodline families and their job is to impose the will of the spider and the global web in their particular country. Subsidiary networks control and manipulate the national political system, finance, corporations, media, medicine, etc. to

ensure that they follow the globally-dictated Cult agenda. These networks were the means through which the ‘Covid’ hoax could be played out with almost every country responding in the same way.

The ‘Yessir’ pyramid

Compartmentalisation is the key to understanding how a tiny few can dictate the lives of billions when combined with a top-down sequence of imposition and acquiescence. The inner core of the Cult sits at the peak of the pyramidal hierarchy of human society ([Fig 2](#) overleaf). It imposes its will – its agenda for the world – on the level immediately below which acquiesces to that imposition. This level then imposes the Cult will on the level below them which acquiesces and imposes on the next level. Very quickly we meet levels in the hierarchy that have no idea there even is a Cult, but the sequence of imposition and acquiescence continues down the pyramid in just the same way. ‘I don’t know why we are doing this but the order came from “on-high” and so we better just do it.’ Alfred Lord Tennyson said of the cannon fodder levels in his poem *The Charge of the Light Brigade*: ‘Theirs not to reason why; theirs but to do and die.’ The next line says that ‘into the valley of death rode the six hundred’ and they died because they obeyed without question what their perceived ‘superiors’ told them to do. In the same way the population capitulated to ‘Covid’. The whole hierarchical pyramid functions like this to allow the very few to direct the enormous many.

Eventually imposition-acquiescence-imposition-acquiescence comes down to the mass of the population at the foot of the pyramid. If they acquiesce to those levels of the hierarchy imposing on them (governments/law enforcement/doctors/media) a circuit is completed between the population and the handful of super-psychopaths in the Cult inner core at the top of the pyramid. Without a circuit-breaking refusal to obey, the sequence of imposition and acquiescence allows a staggeringly few people to impose their will upon the entirety of humankind. We are looking at the very sequence that has subjugated billions since the start of 2020. Our freedom has not been taken from us. Humanity has given it

away. Fascists do not impose fascism because there are not enough of them. Fascism is imposed by the population acquiescing to fascism. Put another way allowing their perceptions to be programmed to the extent that leads to the population giving their freedom away by giving their perceptions – their mind – away. If this circuit is not broken by humanity ceasing to cooperate with their own enslavement then nothing can change. For that to happen people have to critically think and see through the lies and window dressing and then summon the backbone to act upon what they see. The Cult spends its days working to stop either happening and its methodology is systematic and highly detailed, but it can be overcome and that is what this book is all about.

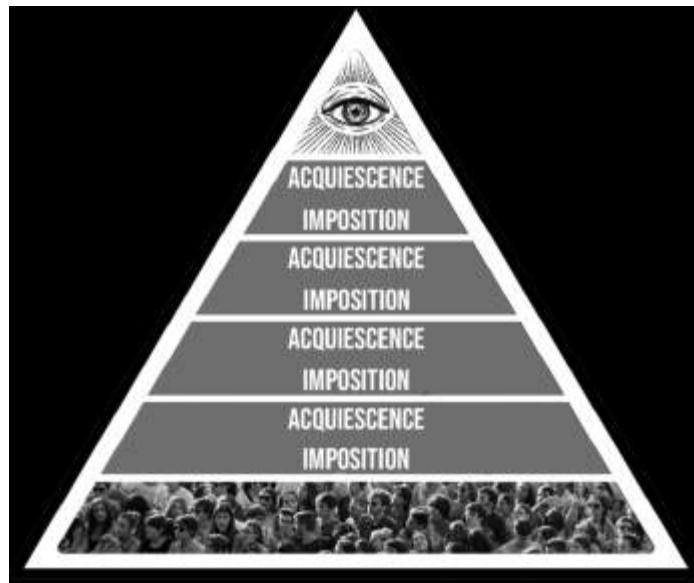


Figure 2: The simple sequence of imposition and compliance that allows a handful of people at the peak of the pyramid to dictate the lives of billions.

The Life Program

Okay, back to world number 1 or the world of the ‘masses’. Observe the process of what we call ‘life’ and it is a perceptual download from cradle to grave. The Cult has created a global structure in which perception can be programmed and the program continually topped-up with what appears to be constant confirmation that the program is indeed true reality. The important word here is ‘appears’.

This is the structure, the fly-trap, the Postage Stamp Consensus or Perceptual Mainframe, which represents that incredibly narrow band of perceived possibility delivered by the ‘education’ system, mainstream media, science and medicine. From the earliest age the download begins with parents who have themselves succumbed to the very programming their children are about to go through. Most parents don’t do this out of malevolence and mostly it is quite the opposite. They do what they believe is best for their children and that is what the program has told them is best. Within three or four years comes the major transition from parental programming to full-blown state (Cult) programming in school, college and university where perceptually-programmed teachers and academics pass on their programming to the next generations. Teachers who resist are soon marginalised and their careers ended while children who resist are called a problem child for whom Ritalin may need to be prescribed. A few years after entering the ‘world’ children are under the control of authority figures representing the state telling them when they have to be there, when they can leave and when they can speak, eat, even go to the toilet. This is calculated preparation for a lifetime of obeying authority in all its forms. Reflex-action fear of authority is instilled by authority from the start. Children soon learn the carrot and stick consequences of obeying or defying authority which is underpinned daily for the rest of their life. Fortunately I daydreamed through this crap and never obeyed authority simply because it told me to. This approach to my alleged ‘bettters’ continues to this day. There can be consequences of pursuing open-minded freedom in a world of closed-minded conformity. I spent a lot of time in school corridors after being ejected from the classroom for not taking some of it seriously and now I spend a lot of time being ejected from Facebook, YouTube and Twitter. But I can tell you that being true to yourself and not compromising your self-respect is far more exhilarating than bowing to authority for authority’s sake. You don’t have to be a sheep to the shepherd (authority) and the sheep dog (fear of not obeying authority).

The perceptual download continues throughout the formative years in school, college and university while script-reading ‘teachers’, ‘academics’ ‘scientists’, ‘doctors’ and ‘journalists’ insist that ongoing generations must be as programmed as they are. Accept the program or you will not pass your ‘exams’ which confirm your ‘degree’ of programming. It is tragic to think that many parents pressure their offspring to work hard at school to download the program and qualify for the next stage at college and university. The late, great, American comedian George Carlin said: ‘Here’s a bumper sticker I’d like to see: We are proud parents of a child who has resisted his teachers’ attempts to break his spirit and bend him to the will of his corporate masters.’ Well, the best of luck finding many of those, George. Then comes the moment to leave the formal programming years in academia and enter the ‘adult’ world of work. There you meet others in your chosen or prescribed arena who went through the same Postage Stamp Consensus program before you did. There is therefore overwhelming agreement between almost everyone on the basic foundations of Postage Stamp reality and the rejection, even contempt, of the few who have a mind of their own and are prepared to use it. This has two major effects. Firstly, the consensus confirms to the programmed that their download is really how things are. I mean, everyone knows that, right? Secondly, the arrogance and ignorance of Postage Stamp adherents ensure that anyone questioning the program will have unpleasant consequences for seeking their own truth and not picking their perceptions from the shelf marked: ‘Things you must believe without question and if you don’t you’re a dangerous lunatic conspiracy theorist and a harebrained nutter’.

Every government, agency and corporation is founded on the same Postage Stamp prison cell and you can see why so many people believe the same thing while calling it their own ‘opinion’. Fusion of governments and corporations in pursuit of the same agenda was the definition of fascism described by Italian dictator Benito Mussolini. The pressure to conform to perceptual norms downloaded for a lifetime is incessant and infiltrates society right

down to family groups that become censors and condemners of their own ‘black sheep’ for not, ironically, being sheep. We have seen an explosion of that in the ‘Covid’ era. Cult-owned global media unleashes its propaganda all day every day in support of the Postage Stamp and targets with abuse and ridicule anyone in the public eye who won’t bend their mind to the will of the tyranny. Any response to this is denied (certainly in my case). They don’t want to give a platform to expose official lies. Cult-owned-and-created Internet giants like Facebook, Google, YouTube and Twitter delete you for having an unapproved opinion. Facebook boasts that its AI censors delete 97-percent of ‘hate speech’ before anyone even reports it. Much of that ‘hate speech’ will simply be an opinion that Facebook and its masters don’t want people to see. Such perceptual oppression is widely known as fascism. Even Facebook executive Benny Thomas, a ‘CEO Global Planning Lead’, said in comments secretly recorded by investigative journalism operation Project Veritas that Facebook is ‘too powerful’ and should be broken up:

I mean, no king in history has been the ruler of two billion people, but Mark Zuckerberg is ... And he's 36. That's too much for a 36-year-old ... You should not have power over two billion people. I just think that's wrong.

Thomas said Facebook-owned platforms like Instagram, Oculus, and WhatsApp needed to be separate companies. ‘It’s too much power when they’re all one together’. That’s the way the Cult likes it, however. We have an executive of a Cult organisation in Benny Thomas that doesn’t know there is a Cult such is the compartmentalisation. Thomas said that Facebook and Google ‘are no longer companies, they’re countries’. Actually they are more powerful than countries on the basis that if you control information you control perception and control human society.

I love my oppressor

Another expression of this psychological trickery is for those who realise they are being pressured into compliance to eventually

convince themselves to believe the official narratives to protect their self-respect from accepting the truth that they have succumbed to meek and subservient compliance. Such people become some of the most vehement defenders of the system. You can see them everywhere screaming abuse at those who prefer to think for themselves and by doing so reminding the compliers of their own capitulation to conformity. ‘You are talking dangerous nonsense you Covidiot!!’ Are you trying to convince me or yourself? It is a potent form of Stockholm syndrome which is defined as: ‘A psychological condition that occurs when a victim of abuse identifies and attaches, or bonds, positively with their abuser.’ An example is hostages bonding and even ‘falling in love’ with their kidnappers. The syndrome has been observed in domestic violence, abused children, concentration camp inmates, prisoners of war and many and various Satanic cults. These are some traits of Stockholm syndrome listed at goodtherapy.org:

- Positive regard towards perpetrators of abuse or captor [see ‘Covid’].
- Failure to cooperate with police and other government authorities when it comes to holding perpetrators of abuse or kidnapping accountable [or in the case of ‘Covid’ cooperating with the police to enforce and defend their captors’ demands].
- Little or no effort to escape [see ‘Covid’].
- Belief in the goodness of the perpetrators or kidnappers [see ‘Covid’].
- Appeasement of captors. This is a manipulative strategy for maintaining one’s safety. As victims get rewarded – perhaps with less abuse or even with life itself – their appeasing behaviours are reinforced [see ‘Covid’].
- Learned helplessness. This can be akin to ‘if you can’t beat ‘em, join ‘em’. As the victims fail to escape the abuse or captivity, they may start giving up and soon realize it’s just easier for everyone if they acquiesce all their power to their captors [see ‘Covid’].

- Feelings of pity toward the abusers, believing they are actually victims themselves. Because of this, victims may go on a crusade or mission to 'save' [protect] their abuser [see the venom unleashed on those challenging the official 'Covid' narrative].
- Unwillingness to learn to detach from their perpetrators and heal. In essence, victims may tend to be less loyal to themselves than to their abuser [*definitely* see 'Covid'].

Ponder on those traits and compare them with the behaviour of great swathes of the global population who have defended governments and authorities which have spent every minute destroying their lives and livelihoods and those of their children and grandchildren since early 2020 with fascistic lockdowns, house arrest and employment deletion to 'protect' them from a 'deadly virus' that their abusers' perceptually created to bring about this very outcome. We are looking at mass Stockholm syndrome. All those that agree to concede their freedom will believe those perceptions are originating in their own independent 'mind' when in fact by conceding their reality to Stockholm syndrome they have by definition conceded any independence of mind. Listen to the 'opinions' of the acquiescing masses in this 'Covid' era and what gushes forth is the repetition of the official version of everything delivered unprocessed, unfiltered and unquestioned. The whole programming dynamic works this way. I must be free because I'm told that I am and so I think that I am.

You can see what I mean with the chapter theme of 'I'm thinking – Oh, but *are you?*' The great majority are not thinking, let alone for themselves. They are repeating what authority has told them to believe which allows them to be controlled. Weaving through this mentality is the fear that the 'conspiracy theorists' are right and this again explains the often hysterical abuse that ensues when you dare to contest the official narrative of anything. Denial is the mechanism of hiding from yourself what you don't want to be true. Telling people what they want to hear is easy, but it's an infinitely greater challenge to tell them what they would rather not be happening.

One is akin to pushing against an open door while the other is met with vehement resistance no matter what the scale of evidence. I don't want it to be true so I'll convince myself that it's not. Examples are everywhere from the denial that a partner is cheating despite all the signs to the reflex-action rejection of any idea that world events in which country after country act in exactly the same way are centrally coordinated. To accept the latter is to accept that a force of unspeakable evil is working to destroy your life and the lives of your children with nothing too horrific to achieve that end. Who the heck wants that to be true? But if we don't face reality the end is duly achieved and the consequences are far worse and ongoing than breaking through the walls of denial today with the courage to make a stand against tyranny.

Connect the dots – but how?

A crucial aspect of perceptual programming is to portray a world in which everything is random and almost nothing is connected to anything else. Randomness cannot be coordinated by its very nature and once you perceive events as random the idea they could be connected is waved away as the rantings of the tinfoil-hat brigade. You can't plan and coordinate random you idiot! No, you can't, but you can hide the coldly-calculated and long-planned behind the *illusion* of randomness. A foundation manifestation of the Renegade Mind is to scan reality for patterns that connect the apparently random and turn pixels and dots into pictures. This is the way I work and have done so for more than 30 years. You look for similarities in people, modus operandi and desired outcomes and slowly, then ever quicker, the picture forms. For instance: There would seem to be no connection between the 'Covid pandemic' hoax and the human-caused global-warming hoax and yet they are masks (appropriately) on the same face seeking the same outcome. Those pushing the global warming myth through the Club of Rome and other Cult agencies are driving the lies about 'Covid' – Bill Gates is an obvious one, but they are endless. Why would the same people be involved in both when they are clearly not connected? Oh, but they

are. Common themes with personnel are matched by common goals. The ‘solutions’ to both ‘problems’ are centralisation of global power to impose the will of the few on the many to ‘save’ humanity from ‘Covid’ and save the planet from an ‘existential threat’ (we need ‘zero Covid’ and ‘zero carbon emissions’). These, in turn, connect with the ‘dot’ of globalisation which was coined to describe the centralisation of global power in every area of life through incessant political and corporate expansion, trading blocks and superstates like the European Union. If you are the few and you want to control the many you have to centralise power and decision-making. The more you centralise power the more power the few at the centre will have over the many; and the more that power is centralised the more power those at the centre have to centralise even quicker. The momentum of centralisation gets faster and faster which is exactly the process we have witnessed. In this way the hoaxed ‘pandemic’ and the fakery of human-caused global warming serve the interests of globalisation and the seizure of global power in the hands of the Cult inner-circle which is behind ‘Covid’, ‘climate change’ and globalisation. At this point random ‘dots’ become a clear and obvious picture or pattern.

Klaus Schwab, the classic Bond villain who founded the Cult’s Gates-funded World Economic Forum, published a book in 2020, *The Great Reset*, in which he used the ‘problem’ of ‘Covid’ to justify a total transformation of human society to ‘save’ humanity from ‘climate change’. Schwab said: ‘The pandemic represents a rare but narrow window of opportunity to reflect, reimagine, and reset our world.’ What he didn’t mention is that the Cult he serves is behind both hoaxes as I show in my book *The Answer*. He and the Cult don’t have to reimagine the world. They know precisely what they want and that’s why they destroyed human society with ‘Covid’ to ‘build back better’ in their grand design. Their job is not to imagine, but to get humanity to imagine and agree with their plans while believing it’s all random. It must be pure coincidence that ‘The Great Reset’ has long been the Cult’s code name for the global imposition of fascism and replaced previous code-names of the ‘New World

'Order' used by Cult frontmen like Father George Bush and the 'New Order of the Ages' which emerged from Freemasonry and much older secret societies. New Order of the Ages appears on the reverse of the Great Seal of the United States as 'Novus ordo seclorum' underneath the Cult symbol used since way back of the pyramid and all seeing-eye ([Fig 3](#)). The pyramid is the hierarchy of human control headed by the illuminated eye that symbolises the force behind the Cult which I will expose in later chapters. The term 'Annuit Coeptis' translates as 'He favours our undertaking'. We are told the 'He' is the Christian god, but 'He' is not as I will be explaining.



Figure 3: The all-seeing eye of the Cult 'god' on the Freemason-designed Great Seal of the United States and also on the dollar bill.

Having you on

Two major Cult techniques of perceptual manipulation that relate to all this are what I have called since the 1990s Problem-Reaction-Solution (PRS) and the Totalitarian Tiptoe (TT). They can be uncovered by the inquiring mind with a simple question: Who benefits? The answer usually identifies the perpetrators of a given action or happening through the concept of 'he who most benefits from a crime is the one most likely to have committed it'. The Latin 'Cue bono?' – Who benefits? – is widely attributed to the Roman orator and statesman Marcus Tullius Cicero. No wonder it goes back so far when the concept has been relevant to human behaviour since

history was recorded. Problem-Reaction-Solution is the technique used to manipulate us every day by covertly creating a problem (or the illusion of one) and offering the solution to the problem (or the illusion of one). In the first phase you create the problem and blame someone or something else for why it has happened. This may relate to a financial collapse, terrorist attack, war, global warming or pandemic, anything in fact that will allow you to impose the ‘solution’ to change society in the way you desire at that time. The ‘problem’ doesn’t have to be real. PRS is manipulation of perception and all you need is the population to believe the problem is real. Human-caused global warming and the ‘Covid pandemic’ only have to be *perceived* to be real for the population to accept the ‘solutions’ of authority. I refer to this technique as NO-Problem-Reaction-Solution. Billions did not meekly accept house arrest from early 2020 because there was a real deadly ‘Covid pandemic’ but because they perceived – believed – that to be the case. The antidote to Problem-Reaction-Solution is to ask who benefits from the proposed solution. Invariably it will be anyone who wants to justify more control through deletion of freedom and centralisation of power and decision-making.

The two world wars were Problem-Reaction-Solutions that transformed and realigned global society. Both were manipulated into being by the Cult as I have detailed in books since the mid-1990s. They dramatically centralised global power, especially World War Two, which led to the United Nations and other global bodies thanks to the overt and covert manipulations of the Rockefeller family and other Cult bloodlines like the Rothschilds. The UN is a stalking horse for full-blown world government that I will come to shortly. The land on which the UN building stands in New York was donated by the Rockefellers and the same Cult family was behind Big Pharma scalpel and drug ‘medicine’ and the creation of the World Health Organization as part of the UN. They have been stalwarts of the eugenics movement and funded Hitler’s race-purity expert Ernst Rudin. The human-caused global warming hoax has been orchestrated by the Club of Rome through the UN which is

manufacturing both the ‘problem’ through its Intergovernmental Panel on Climate Change and imposing the ‘solution’ through its Agenda 21 and Agenda 2030 which demand the total centralisation of global power to ‘save the world’ from a climate hoax the United Nations is itself perpetrating. What a small world the Cult can be seen to be particularly among the inner circles. The bedfellow of Problem-Reaction-Solution is the Totalitarian Tiptoe which became the Totalitarian Sprint in 2020. The technique is fashioned to hide the carefully-coordinated behind the cover of apparently random events. You start the sequence at ‘A’ and you know you are heading for ‘Z’. You don’t want people to know that and each step on the journey is presented as a random happening while all the steps strung together lead in the same direction. The speed may have quickened dramatically in recent times, but you can still see the incremental approach of the Tiptoe in the case of ‘Covid’ as each new imposition takes us deeper into fascism. Tell people they have to do this or that to get back to ‘normal’, then this and this and this. With each new demand adding to the ones that went before the population’s freedom is deleted until it disappears. The spider wraps its web around the flies more comprehensively with each new diktat. I’ll highlight this in more detail when I get to the ‘Covid’ hoax and how it has been pulled off. Another prime example of the Totalitarian Tiptoe is how the Cult-created European Union went from a ‘free-trade zone’ to a centralised bureaucratic dictatorship through the Tiptoe of incremental centralisation of power until nations became mere administrative units for Cult-owned dark suits in Brussels.

The antidote to ignorance is knowledge which the Cult seeks vehemently to deny us, but despite the systematic censorship to that end the Renegade Mind can overcome this by vociferously seeking out the facts no matter the impediments put in the way. There is also a method of thinking and perceiving – *knowing* – that doesn’t even need names, dates, place-type facts to identify the patterns that reveal the story. I’ll get to that in the final chapter. All you need to know about the manipulation of human society and to what end is still out there – *at the time of writing* – in the form of books, videos

and websites for those that really want to breach the walls of programmed perception. To access this knowledge requires the abandonment of the mainstream media as a source of information in the awareness that this is owned and controlled by the Cult and therefore promotes mass perceptions that suit the Cult. Mainstream media lies all day, every day. That is its function and very reason for being. Where it does tell the truth, here and there, is only because the truth and the Cult agenda very occasionally coincide. If you look for fact and insight to the BBC, CNN and virtually all the rest of them you are asking to be conned and perceptually programmed.

Know the outcome and you'll see the journey

Events seem random when you have no idea where the world is being taken. Once you do the random becomes the carefully planned. Know the outcome and you'll see the journey is a phrase I have been using for a long time to give context to daily happenings that appear unconnected. Does a problem, or illusion of a problem, trigger a proposed 'solution' that further drives society in the direction of the outcome? Invariably the answer will be yes and the random – *abracadabra* – becomes the clearly coordinated. So what is this outcome that unlocks the door to a massively expanded understanding of daily events? I will summarise its major aspects – the fine detail is in my other books – and those new to this information will see that the world they thought they were living in is a very different place. The foundation of the Cult agenda is the incessant centralisation of power and all such centralisation is ultimately in pursuit of Cult control on a global level. I have described for a long time the planned world structure of top-down dictatorship as the Hunger Games Society. The term obviously comes from the movie series which portrayed a world in which a few living in military-protected hi-tech luxury were the overlords of a population condemned to abject poverty in isolated 'sectors' that were not allowed to interact. 'Covid' lockdowns and travel bans anyone? The 'Hunger Games' pyramid of structural control has the inner circle of the Cult at the top with pretty much the entire

population at the bottom under their control through dependency for survival on the Cult. The whole structure is planned to be protected and enforced by a military-police state ([Fig 4](#)).

Here you have the reason for the global lockdowns of the fake pandemic to coldly destroy independent incomes and livelihoods and make everyone dependent on the ‘state’ (the Cult that controls the ‘states’). I have warned in my books for many years about the plan to introduce a ‘guaranteed income’ – a barely survivable pittance – designed to impose dependency when employment was destroyed by AI technology and now even more comprehensively at great speed by the ‘Covid’ scam. Once the pandemic was played and lockdown consequences began to delete independent income the authorities began to talk right on cue about the need for a guaranteed income and a ‘Great Reset’. Guaranteed income will be presented as benevolent governments seeking to help a desperate people – desperate as a direct result of actions of the same governments. The truth is that such payments are a trap. You will only get them if you do exactly what the authorities demand including mass vaccination (genetic manipulation). We have seen this theme already in Australia where those dependent on government benefits have them reduced if parents don’t agree to have their children vaccinated according to an insane health-destroying government-dictated schedule. Calculated economic collapse applies to governments as well as people. The Cult wants rid of countries through the creation of a world state with countries broken up into regions ruled by a world government and super states like the European Union. Countries must be bankrupted, too, to this end and it’s being achieved by the trillions in ‘rescue packages’ and furlough payments, trillions in lost taxation, and money-no-object spending on ‘Covid’ including constant all-medium advertising (programming) which has made the media dependent on government for much of its income. The day of reckoning is coming – as planned – for government spending and given that it has been made possible by printing money and not by production/taxation there is inflation on the way that has the

potential to wipe out monetary value. In that case there will be no need for the Cult to steal your money. It just won't be worth anything (see the German Weimar Republic before the Nazis took over). Many have been okay with lockdowns while getting a percentage of their income from so-called furlough payments without having to work. Those payments are dependent, however, on people having at least a theoretical job with a business considered non-essential and ordered to close. As these business go under because they are closed by lockdown after lockdown the furlough stops and it will for everyone eventually. Then what? The 'then what?' is precisely the idea.



Figure 4: The Hunger Games Society structure I have long warned was planned and now the 'Covid' hoax has made it possible. This is the real reason for lockdowns.

Hired hands

Between the Hunger Games Cult elite and the dependent population is planned to be a vicious military-police state (a fusion of the two into one force). This has been in the making for a long time with police looking ever more like the military and carrying weapons to match. The pandemic scam has seen this process accelerate so fast as

lockdown house arrest is brutally enforced by carefully recruited fascist minds and gormless system-servers. The police and military are planned to merge into a centrally-directed world army in a global structure headed by a world government which wouldn't be elected even by the election fixes now in place. The world army is not planned even to be human and instead wars would be fought, primarily against the population, using robot technology controlled by artificial intelligence. I have been warning about this for decades and now militaries around the world are being transformed by this very AI technology. The global regime that I describe is a particular form of fascism known as a technocracy in which decisions are not made by clueless and co-opted politicians but by unelected technocrats – scientists, engineers, technologists and bureaucrats. Cult-owned-and-controlled Silicon Valley giants are examples of technocracy and they already have far more power to direct world events than governments. They are with their censorship *selecting* governments. I know that some are calling the 'Great Reset' a Marxist communist takeover, but fascism and Marxism are different labels for the same tyranny. Tell those who lived in fascist Germany and Stalinist Russia that there was a difference in the way their freedom was deleted and their lives controlled. I could call it a fascist technocracy or a Marxist technocracy and they would be equally accurate. The Hunger Games society with its world government structure would oversee a world army, world central bank and single world cashless currency imposing its will on a microchipped population ([Fig 5](#)). Scan its different elements and see how the illusory pandemic is forcing society in this very direction at great speed. Leaders of 23 countries and the World Health Organization (WHO) backed the idea in March, 2021, of a global treaty for 'international cooperation' in 'health emergencies' and nations should 'come together as a global community for peaceful cooperation that extends beyond this crisis'. Cut the Orwellian bullshit and this means another step towards global government. The plan includes a cashless digital money system that I first warned about in 1993. Right at the start of 'Covid' the deeply corrupt Tedros

Adhanom Ghebreyesus, the crooked and merely gofer ‘head’ of the World Health Organization, said it was possible to catch the ‘virus’ by touching cash and it was better to use cashless means. The claim was ridiculous nonsense and like the whole ‘Covid’ mind-trick it was nothing to do with ‘health’ and everything to do with pushing every aspect of the Cult agenda. As a result of the Tedros lie the use of cash has plummeted. The Cult script involves a single world digital currency that would eventually be technologically embedded in the body. China is a massive global centre for the Cult and if you watch what is happening there you will know what is planned for everywhere. The Chinese government is developing a digital currency which would allow fines to be deducted immediately via AI for anyone caught on camera breaking its fantastic list of laws and the money is going to be programmable with an expiry date to ensure that no one can accrue wealth except the Cult and its operatives.



Figure 5: The structure of global control the Cult has been working towards for so long and this has been enormously advanced by the ‘Covid’ illusion.

Serfdom is so smart

The Cult plan is far wider, extreme, and more comprehensive than even most conspiracy researchers appreciate and I will come to the true depths of deceit and control in the chapters ‘Who controls the

Cult?' and 'Escaping Wetiko'. Even the world that we know is crazy enough. We are being deluged with ever more sophisticated and controlling technology under the heading of 'smart'. We have smart televisions, smart meters, smart cards, smart cars, smart driving, smart roads, smart pills, smart patches, smart watches, smart skin, smart borders, smart pavements, smart streets, smart cities, smart communities, smart environments, smart growth, smart planet ... smart *everything* around us. Smart technologies and methods of operation are designed to interlock to create a global Smart Grid connecting the entirety of human society including human minds to create a centrally-dictated 'hive' mind. 'Smart cities' is code for densely-occupied megacities of total surveillance and control through AI. Ever more destructive frequency communication systems like 5G have been rolled out without any official testing for health and psychological effects (colossal). 5G/6G/7G systems are needed to run the Smart Grid and each one becomes more destructive of body and mind. Deleting independent income is crucial to forcing people into these AI-policed prisons by ending private property ownership (except for the Cult elite). The Cult's Great Reset now openly foresees a global society in which no one will own any possessions and everything will be rented while the Cult would own literally everything under the guise of government and corporations. The aim has been to use the lockdowns to destroy sources of income on a mass scale and when the people are destitute and in unrepayable amounts of debt (problem) Cult assets come forward with the pledge to write-off debt in return for handing over all property and possessions (solution). Everything – literally everything including people – would be connected to the Internet via AI. I was warning years ago about the coming Internet of Things (IoT) in which all devices and technology from your car to your fridge would be plugged into the Internet and controlled by AI. Now we are already there with much more to come. The next stage is the Internet of Everything (IoE) which is planned to include the connection of AI to the human brain and body to replace the human mind with a centrally-controlled AI mind. Instead of perceptions

being manipulated through control of information and censorship those perceptions would come direct from the Cult through AI. What do you think? You think whatever AI decides that you think. In human terms there would be no individual 'think' any longer. Too incredible? The ravings of a lunatic? Not at all. Cult-owned crazies in Silicon Valley have been telling us the plan for years without explaining the real motivation and calculated implications. These include Google executive and 'futurist' Ray Kurzweil who highlights the year 2030 for when this would be underway. He said:

Our thinking ... will be a hybrid of biological and non-biological thinking ... humans will be able to extend their limitations and 'think in the cloud' ... We're going to put gateways to the cloud in our brains ... We're going to gradually merge and enhance ourselves ... In my view, that's the nature of being human – we transcend our limitations.

As the technology becomes vastly superior to what we are then the small proportion that is still human gets smaller and smaller and smaller until it's just utterly negligible.

The sales-pitch of Kurzweil and Cult-owned Silicon Valley is that this would make us 'super-human' when the real aim is to make us post-human and no longer 'human' in the sense that we have come to know. The entire global population would be connected to AI and become the centrally-controlled 'hive-mind' of externally-delivered perceptions. The Smart Grid being installed to impose the Cult's will on the world is being constructed to allow particular locations – even one location – to control the whole global system. From these prime control centres, which absolutely include China and Israel, anything connected to the Internet would be switched on or off and manipulated at will. Energy systems could be cut, communication via the Internet taken down, computer-controlled driverless autonomous vehicles driven off the road, medical devices switched off, the potential is limitless given how much AI and Internet connections now run human society. We have seen nothing yet if we allow this to continue. Autonomous vehicle makers are working with law enforcement to produce cars designed to automatically pull over if they detect a police or emergency vehicle flashing from up to 100 feet away. At a police stop the car would be unlocked and the

window rolled down automatically. Vehicles would only take you where the computer (the state) allowed. The end of petrol vehicles and speed limiters on all new cars in the UK and EU from 2022 are steps leading to electric computerised transport over which ultimately you have no control. The picture is far bigger even than the Cult global network or web and that will become clear when I get to the nature of the ‘spider’. There is a connection between all these happenings and the instigation of DNA-manipulating ‘vaccines’ (which aren’t ‘vaccines’) justified by the ‘Covid’ hoax. That connection is the unfolding plan to transform the human body from a biological to a synthetic biological state and this is why synthetic biology is such a fast-emerging discipline of mainstream science. ‘Covid vaccines’ are infusing self-replicating synthetic genetic material into the cells to cumulatively take us on the Totalitarian Tiptoe from Human 1.0 to the synthetic biological Human 2.0 which will be physically and perceptually attached to the Smart Grid to one hundred percent control every thought, perception and deed.

Humanity needs to wake up and *fast*.

This is the barest explanation of where the ‘outcome’ is planned to go but it’s enough to see the journey happening all around us. Those new to this information will already see ‘Covid’ in a whole new context. I will add much more detail as we go along, but for the minutiae evidence see my mega-works, *The Answer*, *The Trigger* and *Everything You Need to Know But Have Never Been Told*.

Now – how does a Renegade Mind see the ‘world’?

CHAPTER TWO

Renegade Perception

It is one thing to be clever and another to be wise

George R.R. Martin

A simple definition of the difference between a programmed mind and a Renegade Mind would be that one sees only dots while the other connects them to see the picture. Reading reality with accuracy requires the observer to (a) know the planned outcome and (b) realise that everything, but *everything*, is connected.

The entirety of infinite reality is connected – that's its very nature – and with human society an expression of infinite reality the same must apply. Simple cause and effect is a connection. The effect is triggered by the cause and the effect then becomes the cause of another effect. Nothing happens in isolation because it *can't*. Life in whatever reality is simple choice and consequence. We make choices and these lead to consequences. If we don't like the consequences we can make different choices and get different consequences which lead to other choices and consequences. The choice and the consequence are not only connected they are indivisible. You can't have one without the other as an old song goes. A few cannot control the world unless those being controlled allow that to happen – cause and effect, choice and consequence. Control – who has it and who doesn't – is a two-way process, a symbiotic relationship, involving the controller and controlled. 'They took my freedom away!!' Well, yes, but you also gave it to them. Humanity is

subjected to mass control because humanity has acquiesced to that control. This is all cause and effect and literally a case of give and take. In the same way world events of every kind are connected and the Cult works incessantly to sell the illusion of the random and coincidental to maintain the essential (to them) perception of dots that hide the picture. Renegade Minds know this and constantly scan the world for patterns of connection. This is absolutely pivotal in understanding the happenings in the world and without that perspective clarity is impossible. First you know the planned outcome and then you identify the steps on the journey – the day-by-day apparently random which, when connected in relation to the outcome, no longer appear as individual events, but as the proverbial *chain* of events leading in the same direction. I'll give you some examples:

Political puppet show

We are told to believe that politics is 'adversarial' in that different parties with different beliefs engage in an endless tussle for power. There may have been some truth in that up to a point – and only a point – but today divisions between 'different' parties are rhetorical not ideological. Even the rhetorical is fusing into one-speak as the parties eject any remaining free thinkers while others succumb to the ever-gathering intimidation of anyone with the 'wrong' opinion. The Cult is not a new phenomenon and can be traced back thousands of years as my books have documented. Its intergenerational initiates have been manipulating events with increasing effect the more that global power has been centralised. In ancient times the Cult secured control through the system of monarchy in which 'special' bloodlines (of which more later) demanded the right to rule as kings and queens simply by birthright and by vanquishing others who claimed the same birthright. There came a time, however, when people had matured enough to see the unfairness of such tyranny and demanded a say in who governed them. Note the word – *governed* them. Not served them – *governed* them, hence government defined as 'the political direction and control exercised over the

actions of the members, citizens, or inhabitants of communities, societies, and states; direction of the affairs of a state, community, etc.' Governments exercise control over rather than serve just like the monarchies before them. Bizarrely there are still countries like the United Kingdom which are ruled by a monarch *and* a government that officially answers to the monarch. The UK head of state and that of Commonwealth countries such as Canada, Australia and New Zealand is 'selected' by who in a *single family* had unprotected sex with whom and in what order. Pinch me it can't be true. Ouch! Shit, it is. The demise of monarchies in most countries offered a potential vacuum in which some form of free and fair society could arise and the Cult had that base covered. Monarchies had served its interests but they couldn't continue in the face of such widespread opposition and, anyway, replacing a 'royal' dictatorship that people could see with a dictatorship 'of the people' hiding behind the concept of 'democracy' presented far greater manipulative possibilities and ways of hiding coordinated tyranny behind the illusion of 'freedom'.

Democracy is quite wrongly defined as government selected by the population. This is not the case at all. It is government selected by *some* of the population (and then only in theory). This 'some' doesn't even have to be the majority as we have seen so often in first-past-the-post elections in which the so-called majority party wins fewer votes than the 'losing' parties combined. Democracy can give total power to a party in government from a minority of the votes cast. It's a sleight of hand to sell tyranny as freedom. Seventy-four million Trump-supporting Americans didn't vote for the 'Democratic' Party of Joe Biden in the distinctly dodgy election in 2020 and yet far from acknowledging the wishes and feelings of that great percentage of American society the Cult-owned Biden government set out from day one to destroy them and their right to a voice and opinion. Empty shell Biden and his Cult handlers said they were doing this to 'protect democracy'. Such is the level of lunacy and sickness to which politics has descended. Connect the dots and relate them to the desired outcome – a world government run by self-appointed technocrats and no longer even elected

politicians. While operating through its political agents in government the Cult is at the same time encouraging public distain for politicians by putting idiots and incompetents in theoretical power on the road to deleting them. The idea is to instil a public reaction that says of the technocrats: 'Well, they couldn't do any worse than the pathetic politicians.' It's all about controlling perception and Renegade Minds can see through that while programmed minds cannot when they are ignorant of both the planned outcome and the manipulation techniques employed to secure that end. This knowledge can be learned, however, and fast if people choose to get informed.

Politics may at first sight appear very difficult to control from a central point. I mean look at the 'different' parties and how would you be able to oversee them all and their constituent parts? In truth, it's very straightforward because of their structure. We are back to the pyramid of imposition and acquiescence. Organisations are structured in the same way as the system as a whole. Political parties are not open forums of free expression. They are hierarchies. I was a national spokesman for the British Green Party which claimed to be a different kind of politics in which influence and power was devolved; but I can tell you from direct experience – and it's far worse now – that Green parties are run as hierarchies like all the others however much they may try to hide that fact or kid themselves that it's not true. A very few at the top of all political parties are directing policy and personnel. They decide if you are elevated in the party or serve as a government minister and to do that you have to be a yes man or woman. Look at all the maverick political thinkers who never ascended the greasy pole. If you want to progress within the party or reach 'high-office' you need to fall into line and conform. Exceptions to this are rare indeed. Should you want to run for parliament or Congress you have to persuade the local or state level of the party to select you and for that you need to play the game as dictated by the hierarchy. If you secure election and wish to progress within the greater structure you need to go on conforming to what is acceptable to those running the hierarchy

from the peak of the pyramid. Political parties are perceptual gulags and the very fact that there are party 'Whips' appointed to 'whip' politicians into voting the way the hierarchy demands exposes the ridiculous idea that politicians are elected to serve the people they are supposed to represent. Cult operatives and manipulation has long seized control of major parties that have any chance of forming a government and at least most of those that haven't. A new party forms and the Cult goes to work to infiltrate and direct. This has reached such a level today that you see video compilations of 'leaders' of all parties whether Democrats, Republicans, Conservative, Labour and Green parroting the same Cult mantra of 'Build Back Better' and the 'Great Reset' which are straight off the Cult song-sheet to describe the transformation of global society in response to the Cult-instigated hoaxes of the 'Covid pandemic' and human-caused 'climate change'. To see Caroline Lucas, the Green Party MP that I knew when I was in the party in the 1980s, speaking in support of plans proposed by Cult operative Klaus Schwab representing the billionaire global elite is a real head-shaker.

Many parties – one master

The party system is another mind-trick and was instigated to change the nature of the dictatorship by swapping 'royalty' for dark suits that people believed – though now ever less so – represented their interests. Understanding this trick is to realise that a single force (the Cult) controls all parties either directly in terms of the major ones or through manipulation of perception and ideology with others. You don't need to manipulate Green parties to demand your transformation of society in the name of 'climate change' when they are obsessed with the lie that this is essential to 'save the planet'. You just give them a platform and away they go serving your interests while believing they are being environmentally virtuous. America's political structure is a perfect blueprint for how the two or multi-party system is really a one-party state. The Republican Party is controlled from one step back in the shadows by a group made up of billionaires and their gofers known as neoconservatives or Neocons.

I have exposed them in fine detail in my books and they were the driving force behind the policies of the imbecilic presidency of Boy George Bush which included 9/11 (see *The Trigger* for a comprehensive demolition of the official story), the subsequent ‘war on terror’ (war of terror) and the invasions of Afghanistan and Iraq. The latter was a No-Problem-Reaction-Solution based on claims by Cult operatives, including Bush and British Prime Minister Tony Blair, about Saddam Hussein’s ‘weapons of mass destruction’ which did not exist as war criminals Bush and Blair well knew.

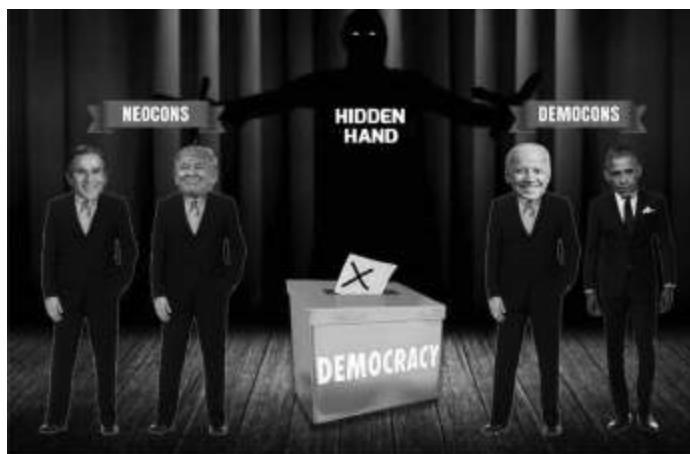


Figure 6: Different front people, different parties – same control system.

The Democratic Party has its own ‘Neocon’ group controlling from the background which I call the ‘Democons’ and here’s the penny-drop – the Neocons and Democons answer to the same masters one step further back into the shadows (Fig 6). At that level of the Cult the Republican and Democrat parties are controlled by the same people and no matter which is in power the Cult is in power. This is how it works in almost every country and certainly in Britain with Conservative, Labour, Liberal Democrat and Green parties now all on the same page whatever the rhetoric may be in their feeble attempts to appear different. Neocons operated at the time of Bush through a think tank called The Project for the New American Century which in September, 2000, published a document entitled *Rebuilding America’s Defenses: Strategies, Forces, and Resources*

For a New Century demanding that America fight ‘multiple, simultaneous major theatre wars’ as a ‘core mission’ to force regime-change in countries including Iraq, Libya and Syria. Neocons arranged for Bush (‘Republican’) and Blair (‘Labour Party’) to front-up the invasion of Iraq and when they departed the Democons orchestrated the targeting of Libya and Syria through Barack Obama (‘Democrat’) and British Prime Minister David Cameron (‘Conservative Party’). We have ‘different’ parties and ‘different’ people, but the same unfolding script. The more the Cult has seized the reigns of parties and personnel the more their policies have transparently pursued the same agenda to the point where the fascist ‘Covid’ impositions of the Conservative junta of Jackboot Johnson in Britain were opposed by the Labour Party because they were not fascist enough. The Labour Party is likened to the US Democrats while the Conservative Party is akin to a British version of the Republicans and on both sides of the Atlantic they all speak the same language and support the direction demanded by the Cult although some more enthusiastically than others. It’s a similar story in country after country because it’s all centrally controlled. Oh, but what about Trump? I’ll come to him shortly. Political ‘choice’ in the ‘party’ system goes like this: You vote for Party A and they get into government. You don’t like what they do so next time you vote for Party B and they get into government. You don’t like what they do when it’s pretty much the same as Party A and why wouldn’t that be with both controlled by the same force? Given that only two, sometimes three, parties have any chance of forming a government to get rid of Party B that you don’t like you have to vote again for Party A which ... you don’t like. This, ladies and gentlemen, is what they call ‘democracy’ which we are told – wrongly – is a term interchangeable with ‘freedom’.

The cult of cults

At this point I need to introduce a major expression of the Global Cult known as Sabbatian-Frankism. Sabbatian is also spelt as Sabbatean. I will summarise here. I have published major exposés

and detailed background in other works. Sabbatian-Frankism combines the names of two frauds posing as 'Jewish' men, Sabbatai Zevi (1626-1676), a rabbi, black magician and occultist who proclaimed he was the Jewish messiah; and Jacob Frank (1726-1791), the Polish 'Jew', black magician and occultist who said he was the reincarnation of 'messiah' Zevi and biblical patriarch Jacob. They worked across two centuries to establish the Sabbatian-Frankist cult that plays a major, indeed central, role in the manipulation of human society by the Global Cult which has its origins much further back in history than Sabbatai Zevi. I should emphasise two points here in response to the shrill voices that will scream 'anti-Semitism': (1) Sabbatian-Frankists are NOT Jewish and only pose as such to hide their cult behind a Jewish façade; and (2) my information about this cult has come from Jewish sources who have long realised that their society and community has been infiltrated and taken over by interloper Sabbatian-Frankists. Infiltration has been the foundation technique of Sabbatian-Frankism from its official origin in the 17th century. Zevi's Sabbatian sect attracted a massive following described as the biggest messianic movement in Jewish history, spreading as far as Africa and Asia, and he promised a return for the Jews to the 'Promised Land' of Israel. Sabbatianism was not Judaism but an inversion of everything that mainstream Judaism stood for. So much so that this sinister cult would have a feast day when Judaism had a fast day and whatever was forbidden in Judaism the Sabbatians were encouraged and even commanded to do. This included incest and what would be today called Satanism. Members were forbidden to marry outside the sect and there was a system of keeping their children ignorant of what they were part of until they were old enough to be trusted not to unknowingly reveal anything to outsiders. The same system is employed to this day by the Global Cult in general which Sabbatian-Frankism has enormously influenced and now largely controls.

Zevi and his Sabbatians suffered a setback with the intervention by the Sultan of the Islamic Ottoman Empire in the Middle East and what is now the Republic of Turkey where Zevi was located. The

Sultan gave him the choice of proving his ‘divinity’, converting to Islam or facing torture and death. Funnily enough Zevi chose to convert or at least appear to. Some of his supporters were disillusioned and drifted away, but many did not with 300 families also converting – only in theory – to Islam. They continued behind this Islamic smokescreen to follow the goals, rules and rituals of Sabbatianism and became known as ‘crypto-Jews’ or the ‘Dönmeh’ which means ‘to turn’. This is rather ironic because they didn’t ‘turn’ and instead hid behind a fake Islamic persona. The process of appearing to be one thing while being very much another would become the calling card of Sabbatianism especially after Zevi’s death and the arrival of the Satanist Jacob Frank in the 18th century when the cult became Sabbatian-Frankism and plumbed still new depths of depravity and infiltration which included – still includes – human sacrifice and sex with children. Wherever Sabbatians go paedophilia and Satanism follow and is it really a surprise that Hollywood is so infested with child abuse and Satanism when it was established by Sabbatian-Frankists and is still controlled by them? Hollywood has been one of the prime vehicles for global perceptual programming and manipulation. How many believe the version of ‘history’ portrayed in movies when it is a travesty and inversion (again) of the truth? Rabbi Marvin Antelman describes Frankism in his book, *To Eliminate the Opiate*, as ‘a movement of complete evil’ while Jewish professor Gershom Scholem said of Frank in *The Messianic Idea in Judaism*: ‘In all his actions [he was] a truly corrupt and degenerate individual ... one of the most frightening phenomena in the whole of Jewish history.’ Frank was excommunicated by traditional rabbis, as was Zevi, but Frank was undeterred and enjoyed vital support from the House of Rothschild, the infamous banking dynasty whose inner-core are Sabbatian-Frankists and not Jews. Infiltration of the Roman Church and Vatican was instigated by Frank with many Dönmeh ‘turning’ again to convert to Roman Catholicism with a view to hijacking the reins of power. This was the ever-repeating modus operandi and continues to be so. Pose as an advocate of the religion, culture or country that you want to control and then

manipulate your people into the positions of authority and influence largely as advisers, administrators and Svengalis for those that appear to be in power. They did this with Judaism, Christianity (Christian Zionism is part of this), Islam and other religions and nations until Sabbatian-Frankism spanned the world as it does today.

Sabbatian Saudis and the terror network

One expression of the Sabbatian-Frankist Dönme within Islam is the ruling family of Saudi Arabia, the House of Saud, through which came the vile distortion of Islam known as Wahhabism. This is the violent creed followed by terrorist groups like Al-Qaeda and ISIS or Islamic State. Wahhabism is the hand-chopping, head-chopping ‘religion’ of Saudi Arabia which is used to keep the people in a constant state of fear so the interloper House of Saud can continue to rule. Al-Qaeda and Islamic State were lavishly funded by the House of Saud while being created and directed by the Sabbatian-Frankist network in the United States that operates through the Pentagon, CIA and the government in general of whichever ‘party’. The front man for the establishment of Wahhabism in the middle of the 18th century was a Sabbatian-Frankist ‘crypto-Jew’ posing as Islamic called Muhammad ibn Abd al-Wahhab. His daughter would marry the son of Muhammad bin Saud who established the first Saudi state before his death in 1765 with support from the British Empire. Bin Saud’s successors would establish modern Saudi Arabia in league with the British and Americans in 1932 which allowed them to seize control of Islam’s major shrines in Mecca and Medina. They have dictated the direction of Sunni Islam ever since while Iran is the major centre of the Shiite version and here we have the source of at least the public conflict between them. The Sabbatian network has used its Wahhabi extremists to carry out Problem-Reaction-Solution terrorist attacks in the name of ‘Al-Qaeda’ and ‘Islamic State’ to justify a devastating ‘war on terror’, ever-increasing surveillance of the population and to terrify people into compliance. Another insight of the Renegade Mind is the streetwise understanding that

just because a country, location or people are attacked doesn't mean that those apparently representing that country, location or people are not behind the attackers. Often they are *orchestrating* the attacks because of the societal changes that can be then justified in the name of 'saving the population from terrorists'.

I show in great detail in *The Trigger* how Sabbatian-Frankists were the real perpetrators of 9/11 and not '19 Arab hijackers' who were blamed for what happened. Observe what was justified in the name of 9/11 alone in terms of Middle East invasions, mass surveillance and control that fulfilled the demands of the Project for the New American Century document published by the Sabbatian Neocons. What appear to be enemies are on the deep inside players on the same Sabbatian team. Israel and Arab 'royal' dictatorships are all ruled by Sabbatians and the recent peace agreements between Israel and Saudi Arabia, the United Arab Emirates (UAE) and others are only making formal what has always been the case behind the scenes. Palestinians who have been subjected to grotesque tyranny since Israel was bombed and terrorised into existence in 1948 have never stood a chance. Sabbatian-Frankists have controlled Israel (so the constant theme of violence and war which Sabbatians love) and they have controlled the Arab countries that Palestinians have looked to for real support that never comes. 'Royal families' of the Arab world in Saudi Arabia, Bahrain, UAE, etc., are all Sabbatians with allegiance to the aims of the cult and not what is best for their Arabic populations. They have stolen the oil and financial resources from their people by false claims to be 'royal dynasties' with a genetic right to rule and by employing vicious militaries to impose their will.

Satanic 'illumination'

The Satanist Jacob Frank formed an alliance in 1773 with two other Sabbatians, Mayer Amschel Rothschild (1744-1812), founder of the Rothschild banking dynasty, and Jesuit-educated fraudulent Jew, Adam Weishaupt, and this led to the formation of the Bavarian Illuminati, firstly under another name, in 1776. The Illuminati would

be the manipulating force behind the French Revolution (1789-1799) and was also involved in the American Revolution (1775-1783) before and after the Illuminati's official creation. Weishaupt would later become (in public) a Protestant Christian in archetypal Sabbatian style. I read that his name can be decoded as Adam-Weishaupt or 'the first man to lead those who know'. He wasn't a leader in the sense that he was a subordinate, but he did lead those below him in a crusade of transforming human society that still continues today. The theme was confirmed as early as 1785 when a horseman courier called Lanz was reported to be struck by lighting and extensive Illuminati documents were found in his saddlebags. They made the link to Weishaupt and detailed the plan for world takeover. Current events with 'Covid' fascism have been in the making for a very long time. Jacob Frank was jailed for 13 years by the Catholic Inquisition after his arrest in 1760 and on his release he headed for Frankfurt, Germany, home city and headquarters of the House of Rothschild where the alliance was struck with Mayer Amschel Rothschild and Weishaupt. Rothschild arranged for Frank to be given the title of Baron and he became a wealthy nobleman with a big following of Jews in Germany, the Austro-Hungarian Empire and other European countries. Most of them would have believed he was on their side.

The name 'Illuminati' came from the Zohar which is a body of works in the Jewish mystical 'bible' called the Kabbalah. 'Zohar' is the foundation of Sabbatian-Frankist belief and in Hebrew 'Zohar' means 'splendour', 'radiance', 'illuminated', and so we have 'Illuminati'. They claim to be the 'Illuminated Ones' from their knowledge systematically hidden from the human population and passed on through generations of carefully-chosen initiates in the global secret society network or Cult. Hidden knowledge includes an awareness of the Cult agenda for the world and the nature of our collective reality that I will explore later. Cult 'illumination' is symbolised by the torch held by the Statue of Liberty which was gifted to New York by French Freemasons in Paris who knew exactly what it represents. 'Liberty' symbolises the goddess worshipped in

Babylon as Queen Semiramis or Ishtar. The significance of this will become clear. Notice again the ubiquitous theme of inversion with the Statue of 'Liberty' really symbolising mass control ([Fig 7](#)). A mirror-image statute stands on an island in the River Seine in Paris from where New York Liberty originated ([Fig 8](#)). A large replica of the Liberty flame stands on top of the Pont de l'Alma tunnel in Paris where Princess Diana died in a Cult ritual described in *The Biggest Secret*. Lucifer 'the light bringer' is related to all this (and much more as we'll see) and 'Lucifer' is a central figure in Sabbatian-Frankism and its associated Satanism. Sabbatians reject the Jewish Torah, or Pentateuch, the 'five books of Moses' in the Old Testament known as Genesis, Exodus, Leviticus, Numbers, and Deuteronomy which are claimed by Judaism and Christianity to have been dictated by 'God' to Moses on Mount Sinai. Sabbatians say these do not apply to them and they seek to replace them with the Zohar to absorb Judaism and its followers into their inversion which is an expression of a much greater global inversion. They want to delete all religions and force humanity to worship a one-world religion – Sabbatian Satanism that also includes worship of the Earth goddess. Satanic themes are being more and more introduced into mainstream society and while Christianity is currently the foremost target for destruction the others are planned to follow.



Figure 7: The Cult goddess of Babylon disguised as the Statue of Liberty holding the flame of Lucifer the 'light bringer'.



Figure 8: Liberty's mirror image in Paris where the New York version originated.

Marx brothers

Rabbi Marvin Antelman connects the Illuminati to the Jacobins in *To Eliminate the Opiate* and Jacobins were the force behind the French Revolution. He links both to the Bund der Gerechten, or League of the Just, which was the network that inflicted communism/Marxism on the world. Antelman wrote:

The original inner circle of the Bund der Gerechten consisted of born Catholics, Protestants and Jews [Sabbatian-Frankist infiltrators], and those representatives of respective subdivisions formulated schemes for the ultimate destruction of their faiths. The heretical Catholics laid plans which they felt would take a century or more for the ultimate destruction of the church; the apostate Jews for the ultimate destruction of the Jewish religion.

Sabbatian-created communism connects into this anti-religion agenda in that communism does not allow for the free practice of religion. The Sabbatian 'Bund' became the International Communist Party and Communist League and in 1848 'Marxism' was born with the Communist Manifesto of Sabbatian assets Karl Marx and Friedrich Engels. It is absolutely no coincidence that Marxism, just a different name for fascist and other centrally-controlled tyrannies, is being imposed worldwide as a result of the 'Covid' hoax and nor that Marxist/fascist China was the place where the hoax originated. The reason for this will become very clear in the chapter 'Covid: The calculated catastrophe'. The so-called 'Woke' mentality has hijacked

traditional beliefs of the political left and replaced them with far-right make-believe ‘social justice’ better known as Marxism. Woke will, however, be swallowed by its own perceived ‘revolution’ which is really the work of billionaires and billionaire corporations feigning being ‘Woke’. Marxism is being touted by Wokers as a replacement for ‘capitalism’ when we don’t have ‘capitalism’. We have cartelism in which the market is stitched up by the very Cult billionaires and corporations bankrolling Woke. Billionaires love Marxism which keeps the people in servitude while they control from the top.

Terminally naïve Wokers think they are ‘changing the world’ when it’s the Cult that is doing the changing and when they have played their vital part and become surplus to requirements they, too, will be targeted. The Illuminati-Jacobins were behind the period known as ‘The Terror’ in the French Revolution in 1793 and 1794 when Jacobin Maximillian de Robespierre and his Orwellian ‘Committee of Public Safety’ killed 17,000 ‘enemies of the Revolution’ who had once been ‘friends of the Revolution’. Karl Marx (1818-1883), whose Sabbatian creed of Marxism has cost the lives of at least 100 million people, is a hero once again to Wokers who have been systematically kept ignorant of real history by their ‘education’ programming. As a result they now promote a Sabbatian ‘Marxist’ abomination destined at some point to consume them. Rabbi Antelman, who spent decades researching the Sabbatian plot, said of the League of the Just and Karl Marx:

Contrary to popular opinion Karl Marx did not originate the Communist Manifesto. He was paid for his services by the League of the Just, which was known in its country of origin, Germany, as the Bund der Gaeachteten.

Antelman said the text attributed to Marx was the work of other people and Marx ‘was only repeating what others already said’. Marx was ‘a hired hack – lackey of the wealthy Illuminists’. Marx famously said that religion was the ‘opium of the people’ (part of the Sabbatian plan to demonise religion) and Antelman called his books, *To Eliminate the Opiate*. Marx was born Jewish, but his family converted to Christianity (Sabbatian modus operandi) and he

attacked Jews, not least in his book, *A World Without Jews*. In doing so he supported the Sabbatian plan to destroy traditional Jewishness and Judaism which we are clearly seeing today with the vindictive targeting of orthodox Jews by the Sabbatian government of Israel over 'Covid' laws. I don't follow any religion and it has done much damage to the world over centuries and acted as a perceptual straightjacket. Renegade Minds, however, are always asking *why* something is being done. It doesn't matter if they agree or disagree with what is happening – *why* is it happening is the question. The 'why?' can be answered with regard to religion in that religions create interacting communities of believers when the Cult wants to dismantle all discourse, unity and interaction (see 'Covid' lockdowns) and the ultimate goal is to delete all religions for a one-world religion of Cult Satanism worshipping their 'god' of which more later. We see the same 'why?' with gun control in America. I don't have guns and don't want them, but why is the Cult seeking to disarm the population at the same time that law enforcement agencies are armed to their molars and why has every tyrant in history sought to disarm people before launching the final takeover? They include Hitler, Stalin, Pol Pot and Mao who followed confiscation with violent seizing of power. You know it's a Cult agenda by the people who immediately race to the microphones to exploit dead people in multiple shootings. Ultra-Zionist Cult lackey Senator Chuck Schumer was straight on the case after ten people were killed in Boulder, Colorado in March, 2021. Simple rule ... if Schumer wants it the Cult wants it and the same with his ultra-Zionist mate the wild-eyed Senator Adam Schiff. At the same time they were calling for the disarmament of Americans, many of whom live a long way from a police response, Schumer, Schiff and the rest of these pampered clowns were sitting on Capitol Hill behind a razor-wired security fence protected by thousands of armed troops in addition to their own armed bodyguards. Mom and pop in an isolated home? They're just potential mass shooters.

Zion Mainframe

Sabbatian-Frankists and most importantly the Rothschilds were behind the creation of 'Zionism', a political movement that demanded a Jewish homeland in Israel as promised by Sabbatai Zevi. The very symbol of Israel comes from the German meaning of the name Rothschild. Dynasty founder Mayer Amschel Rothschild changed the family name from Bauer to Rothschild, or 'Red-Shield' in German, in deference to the six-pointed 'Star of David' hexagram displayed on the family's home in Frankfurt. The symbol later appeared on the flag of Israel after the Rothschilds were centrally involved in its creation. Hexagrams are not a uniquely Jewish symbol and are widely used in occult ('hidden') networks often as a symbol for Saturn (see my other books for why). Neither are Zionism and Jewishness interchangeable. Zionism is a political movement and philosophy and not a 'race' or a people. Many Jews oppose Zionism and many non-Jews, including US President Joe Biden, call themselves Zionists as does Israel-centric Donald Trump. America's support for the Israel government is pretty much a gimme with ultra-Zionist billionaires and corporations providing fantastic and dominant funding for both political parties. Former Congresswoman Cynthia McKinney has told how she was approached immediately she ran for office to 'sign the pledge' to Israel and confirm that she would always vote in that country's best interests. All American politicians are approached in this way. Anyone who refuses will get no support or funding from the enormous and all-powerful Zionist lobby that includes organisations like mega-lobby group AIPAC, the American Israel Public Affairs Committee. Trump's biggest funder was ultra-Zionist casino and media billionaire Sheldon Adelson while major funders of the Democratic Party include ultra-Zionist George Soros and ultra-Zionist financial and media mogul, Haim Saban. Some may reel back at the suggestion that Soros is an Israel-firster (Sabbatian-controlled Israel-firster), but Renegade Minds watch the actions not the words and everywhere Soros donates his billions the Sabbatian agenda benefits. In the spirit of Sabbatian inversion Soros pledged \$1 billion for a new university network to promote 'liberal values and tackle intolerance'. He made the announcement during his annual speech

at the Cult-owned World Economic Forum in Davos, Switzerland, in January, 2020, after his ‘harsh criticism’ of ‘authoritarian rulers’ around the world. You can only laugh at such brazen mendacity. How *he* doesn’t laugh is the mystery. Translated from the Orwellian ‘liberal values and tackle intolerance’ means teaching non-white people to hate white people and for white people to loathe themselves for being born white. The reason for that will become clear.

The ‘Anti-Semitism’ fraud

Zionists support the Jewish homeland in the land of Palestine which has been the Sabbatian-Rothschild goal for so long, but not for the benefit of Jews. Sabbatians and their global Anti-Semitism Industry have skewed public and political opinion to equate opposing the violent extremes of Zionism to be a blanket attack and condemnation of all Jewish people. Sabbatians and their global Anti-Semitism Industry have skewed public and political opinion to equate opposing the violent extremes of Zionism to be a blanket attack and condemnation of all Jewish people. This is nothing more than a Sabbatian protection racket to stop legitimate investigation and exposure of their agendas and activities. The official definition of ‘anti-Semitism’ has more recently been expanded to include criticism of Zionism – a *political movement* – and this was done to further stop exposure of Sabbatian infiltrators who created Zionism as we know it today in the 19th century. Renegade Minds will talk about these subjects when they know the shit that will come their way. People must decide if they want to know the truth or just cower in the corner in fear of what others will say. Sabbatians have been trying to label me as ‘anti-Semitic’ since the 1990s as I have uncovered more and more about their background and agendas. Useless, gutless, fraudulent ‘journalists’ then just repeat the smears without question and on the day I was writing this section a pair of unquestioning repeaters called Ben Quinn and Archie Bland (how appropriate) outright called me an ‘anti-Semite’ in the establishment propaganda sheet, the London *Guardian*, with no supporting evidence. The

Sabbatian Anti-Semitism Industry said so and who are they to question that? They wouldn't dare. Ironically 'Semitic' refers to a group of languages in the Middle East that are almost entirely Arabic. 'Anti-Semitism' becomes 'anti-Arab' which if the consequences of this misunderstanding were not so grave would be hilarious. Don't bother telling Quinn and Bland. I don't want to confuse them, bless 'em. One reason I am dubbed 'anti-Semitic' is that I wrote in the 1990s that Jewish operatives (Sabbatians) were heavily involved in the Russian Revolution when Sabbatians overthrew the Romanov dynasty. This apparently made me 'anti-Semitic'. Oh, really? Here is a section from *The Trigger*:

British journalist Robert Wilton confirmed these themes in his 1920 book *The Last Days of the Romanovs* when he studied official documents from the Russian government to identify the members of the Bolshevik ruling elite between 1917 and 1919. The Central Committee included 41 Jews among 62 members; the Council of the People's Commissars had 17 Jews out of 22 members; and 458 of the 556 most important Bolshevik positions between 1918 and 1919 were occupied by Jewish people. Only 17 were Russian. Then there were the 23 Jews among the 36 members of the vicious Cheka Soviet secret police established in 1917 who would soon appear all across the country.

Professor Robert Service of Oxford University, an expert on 20th century Russian history, found evidence that ['Jewish'] Leon Trotsky had sought to make sure that Jews were enrolled in the Red Army and were disproportionately represented in the Soviet civil bureaucracy that included the Cheka which performed mass arrests, imprisonment and executions of 'enemies of the people'. A US State Department Decimal File (861.00/5339) dated November 13th, 1918, names [Rothschild banking agent in America] Jacob Schiff and a list of ultra-Zionists as funders of the Russian Revolution leading to claims of a 'Jewish plot', but the key point missed by all is they were not 'Jews' – they were Sabbatian-Frankists.

Britain's Winston Churchill made the same error by mistake or otherwise. He wrote in a 1920 edition of the *Illustrated Sunday Herald* that those behind the Russian revolution were part of a 'worldwide conspiracy for the overthrow of civilisation and for the reconstitution of society on the basis of arrested development, of envious malevolence, and impossible equality' (see 'Woke' today because that has been created by the same network). Churchill said there was no need to exaggerate the part played in the creation of Bolshevism and in the actual bringing about of the Russian

Revolution 'by these international and for the most part atheistical Jews' ['atheistical Jews' = Sabbatians]. Churchill said it is certainly a very great one and probably outweighs all others: 'With the notable exception of Lenin, the majority of the leading figures are Jews.' He went on to describe, knowingly or not, the Sabbatian modus operandi of placing puppet leaders nominally in power while they control from the background:

Moreover, the principal inspiration and driving power comes from the Jewish leaders. Thus Tchitcherin, a pure Russian, is eclipsed by his nominal subordinate, Litvinoff, and the influence of Russians like Bukharin or Lunacharski cannot be compared with the power of Trotsky, or of Zinovieff, the Dictator of the Red Citadel (Petrograd), or of Krassin or Radek – all Jews. In the Soviet institutions the predominance of Jews is even more astonishing. And the prominent, if not indeed the principal, part in the system of terrorism applied by the Extraordinary Commissions for Combatting Counter-Revolution has been taken by Jews, and in some notable cases by Jewesses.

What I said about seriously disproportionate involvement in the Russian Revolution by Jewish 'revolutionaries' (Sabbatians) is provable fact, but truth is no defence against the Sabbatian Anti-Semitism Industry, its repeater parrots like Quinn and Bland, and the now breathtaking network of so-called 'Woke' 'anti-hate' groups with interlocking leaderships and funding which have the role of discrediting and silencing anyone who gets too close to exposing the Sabbatians. We have seen 'truth is no defence' confirmed in legal judgements with the Saskatchewan Human Rights Commission in Canada decreeing this: 'Truthful statements can be presented in a manner that would meet the definition of hate speech, and not all truthful statements must be free from restriction.' Most 'anti-hate' activists, who are themselves consumed by hatred, are too stupid and ignorant of the world to know how they are being used. They are far too far up their own virtue-signalling arses and it's far too dark for them to see anything.

The 'revolution' game

The background and methods of the 'Russian' Revolution are straight from the Sabbatian playbook seen in the French Revolution

and endless others around the world that appear to start as a revolution of the people against tyrannical rule and end up with a regime change to more tyrannical rule overtly or covertly. Wars, terror attacks and regime overthrows follow the Sabbatian cult through history with its agents creating them as Problem-Reaction-Solutions to remove opposition on the road to world domination. Sabbatian dots connect the Rothschilds with the Illuminati, Jacobins of the French Revolution, the 'Bund' or League of the Just, the International Communist Party, Communist League and the Communist Manifesto of Karl Marx and Friedrich Engels that would lead to the Rothschild-funded Russian Revolution. The sequence comes under the heading of 'creative destruction' when you advance to your global goal by continually destroying the status quo to install a new status quo which you then also destroy. The two world wars come to mind. With each new status quo you move closer to your planned outcome. Wars and mass murder are to Sabbatians a collective blood sacrifice ritual. They are obsessed with death for many reasons and one is that death is an inversion of life. Satanists and Sabbatians are obsessed with death and often target churches and churchyards for their rituals. Inversion-obsessed Sabbatians explain the use of inverted symbolism including the *inverted* pentagram and *inverted* cross. The inversion of the cross has been related to targeting Christianity, but the cross was a religious symbol long before Christianity and its inversion is a statement about the Sabbatian mentality and goals more than any single religion.

Sabbatians operating in Germany were behind the rise of the occult-obsessed Nazis and the subsequent Jewish exodus from Germany and Europe to Palestine and the United States after World War Two. The Rothschild dynasty was at the forefront of this both as political manipulators and by funding the operation. Why would Sabbatians help to orchestrate the horrors inflicted on Jews by the Nazis and by Stalin after they organised the Russian Revolution? Sabbatians hate Jews and their religion, that's why. They pose as Jews and secure positions of control within Jewish society and play the 'anti-Semitism' card to protect themselves from exposure

through a global network of organisations answering to the Sabbatian-created-and-controlled globe-spanning intelligence network that involves a stunning web of military-intelligence operatives and operations for a tiny country of just nine million. Among them are Jewish assets who are not Sabbatians but have been convinced by them that what they are doing is for the good of Israel and the Jewish community to protect them from what they have been programmed since childhood to believe is a Jew-hating hostile world. The Jewish community is just a highly convenient cover to hide the true nature of Sabbatians. Anyone getting close to exposing their game is accused by Sabbatian place-people and gofers of 'anti-Semitism' and claiming that all Jews are part of a plot to take over the world. I am not saying that. I am saying that Sabbatians – the *real* Jew-haters – have infiltrated the Jewish community to use them both as a cover and an 'anti-Semitic' defence against exposure. Thus we have the Anti-Semitism Industry targeted researchers in this way and most Jewish people think this is justified and genuine. They don't know that their 'Jewish' leaders and institutions of state, intelligence and military are not controlled by Jews at all, but cultists and stooges of Sabbatian-Frankism. I once added my name to a pro-Jewish freedom petition online and the next time I looked my name was gone and text had been added to the petition blurb to attack me as an 'anti-Semite' such is the scale of perceptual programming.

Moving on America

I tell the story in *The Trigger* and a chapter called 'Atlantic Crossing' how particularly after Israel was established the Sabbatians moved in on the United States and eventually grasped control of government administration, the political system via both Democrats and Republicans, the intelligence community like the CIA and National Security Agency (NSA), the Pentagon and mass media. Through this seriously compartmentalised network Sabbatians and their operatives in Mossad, Israeli Defense Forces (IDF) and US agencies pulled off 9/11 and blamed it on 19 'Al-Qaeda hijackers' dominated by men from, or connected to, Sabbatian-ruled Saudi

Arabia. The '19' were not even on the planes let alone flew those big passenger jets into buildings while being largely incompetent at piloting one-engine light aircraft. 'Hijacker' Hani Hanjour who is said to have flown American Airlines Flight 77 into the Pentagon with a turn and manoeuvre most professional pilots said they would have struggled to do was banned from renting a small plane by instructors at the Freeway Airport in Bowie, Maryland, just *six weeks* earlier on the grounds that he was an incompetent pilot. The Jewish population of the world is just 0.2 percent with even that almost entirely concentrated in Israel (75 percent Jewish) and the United States (around two percent). This two percent and globally 0.2 percent refers to *Jewish* people and not Sabbatian interlopers who are a fraction of that fraction. What a sobering thought when you think of the fantastic influence on world affairs of tiny Israel and that the Project for the New America Century (PNAC) which laid out the blueprint in September, 2000, for America's war on terror and regime change wars in Iraq, Libya and Syria was founded and dominated by Sabbatians known as 'Neocons'. The document conceded that this plan would not be supported politically or publicly without a major attack on American soil and a Problem-Reaction-Solution excuse to send troops to war across the Middle East. Sabbatian Neocons said:

... [The] process of transformation ... [war and regime change] ... is likely to be a long one, absent some catastrophic and catalysing event – like a new Pearl Harbor.

Four months later many of those who produced that document came to power with their inane puppet George Bush from the long-time Sabbatian Bush family. They included Sabbatian Dick Cheney who was officially vice-president, but really de-facto president for the entirety of the 'Bush' government. Nine months after the 'Bush' inauguration came what Bush called at the time 'the Pearl Harbor of the 21st century' and with typical Sabbatian timing and symbolism 2001 was the 60th anniversary of the attack in 1941 by the Japanese Air Force on Pearl Harbor, Hawaii, which allowed President Franklin Delano Roosevelt to take the United States into a Sabbatian-

instigated Second World War that he said in his election campaign that he never would. The evidence is overwhelming that Roosevelt and his military and intelligence networks knew the attack was coming and did nothing to stop it, but they did make sure that America's most essential naval ships were not in Hawaii at the time. Three thousand Americans died in the Pearl Harbor attacks as they did on September 11th. By the 9/11 year of 2001 Sabbatians had widely infiltrated the US government, military and intelligence operations and used their compartmentalised assets to pull off the 'Al-Qaeda' attacks. If you read *The Trigger* it will blow your mind to see the utterly staggering concentration of 'Jewish' operatives (Sabbatian infiltrators) in essential positions of political, security, legal, law enforcement, financial and business power before, during, and after the attacks to make them happen, carry them out, and then cover their tracks – and I do mean *staggering* when you think of that 0.2 percent of the world population and two percent of Americans which are Jewish while Sabbatian infiltrators are a fraction of that. A central foundation of the 9/11 conspiracy was the hijacking of government, military, Air Force and intelligence computer systems in real time through 'back-door' access made possible by Israeli (Sabbatian) 'cyber security' software. Sabbatian-controlled Israel is on the way to rivalling Silicon Valley for domination of cyberspace and is becoming the dominant force in cyber-security which gives them access to entire computer systems and their passcodes across the world. Then add to this that Zionists head (officially) Silicon Valley giants like Google (Larry Page and Sergey Brin), Google-owned YouTube (Susan Wojcicki), Facebook (Mark Zuckerberg and Sheryl Sandberg), and Apple (Chairman Arthur D. Levinson), and that ultra-Zionist hedge fund billionaire Paul Singer has a \$1 billion stake in Twitter which is only nominally headed by 'CEO' pothead Jack Dorsey. As cable news host Tucker Carlson said of Dorsey: 'There used to be debate in the medical community whether dropping a ton of acid had permanent effects and I think that debate has now ended.' Carlson made the comment after Dorsey told a hearing on Capitol Hill (if you cut through his bullshit) that he

believed in free speech so long as he got to decide what you can hear and see. These 'big names' of Silicon Valley are only front men and women for the Global Cult, not least the Sabbatians, who are the true controllers of these corporations. Does anyone still wonder why these same people and companies have been ferociously censoring and banning people (like me) for exposing any aspect of the Cult agenda and especially the truth about the 'Covid' hoax which Sabbatians have orchestrated?

The Jeffrey Epstein paedophile ring was a Sabbatian operation. He was officially 'Jewish' but he was a Sabbatian and women abused by the ring have told me about the high number of 'Jewish' people involved. The Epstein horror has Sabbatian written all over it and matches perfectly their modus operandi and obsession with sex and ritual. Epstein was running a Sabbatian blackmail ring in which famous people with political and other influence were provided with young girls for sex while everything was being filmed and recorded on hidden cameras and microphones at his New York house, Caribbean island and other properties. Epstein survivors have described this surveillance system to me and some have gone public. Once the famous politician or other figure knew he or she was on video they tended to do whatever they were told. Here we go again ...when you've got them by the balls their hearts and minds will follow. Sabbatians use this blackmail technique on a wide scale across the world to entrap politicians and others they need to act as demanded. Epstein's private plane, the infamous 'Lolita Express', had many well-known passengers including Bill Clinton while Bill Gates has flown on an Epstein plane and met with him four years after Epstein had been jailed for paedophilia. They subsequently met many times at Epstein's home in New York according to a witness who was there. Epstein's infamous side-kick was Ghislaine Maxwell, daughter of Mossad agent and ultra-Zionist mega-crooked British businessman, Bob Maxwell, who at one time owned the *Daily Mirror* newspaper. Maxwell was murdered at sea on his boat in 1991 by Sabbatian-controlled Mossad when he became a liability with his

business empire collapsing as a former Mossad operative has confirmed (see *The Trigger*).

Money, money, money, funny money ...

Before I come to the Sabbatian connection with the last three US presidents I will lay out the crucial importance to Sabbatians of controlling banking and finance. Sabbatian Mayer Amschel Rothschild set out to dominate this arena in his family's quest for total global control. What is freedom? It is, in effect, choice. The more choices you have the freer you are and the fewer your choices the more you are enslaved. In the global structure created over centuries by Sabbatians the biggest decider and restrictor of choice is ... money. Across the world if you ask people what they would like to do with their lives and why they are not doing that they will reply 'I don't have the money'. This is the idea. A global elite of multi-billionaires are described as 'greedy' and that is true on one level; but control of money – who has it and who doesn't – is not primarily about greed. It's about control. Sabbatians have seized ever more control of finance and sucked the wealth of the world out of the hands of the population. We talk now, after all, about the 'One-percent' and even then the wealthiest are a lot fewer even than that. This has been made possible by a money scam so outrageous and so vast it could rightly be called the scam of scams founded on creating 'money' out of nothing and 'loaning' that with interest to the population. Money out of nothing is called 'credit'. Sabbatians have asserted control over governments and banking ever more completely through the centuries and secured financial laws that allow banks to lend hugely more than they have on deposit in a confidence trick known as fractional reserve lending. Imagine if you could lend money that doesn't exist and charge the recipient interest for doing so. You would end up in jail. Bankers by contrast end up in mansions, private jets, Malibu and Monaco.

Banks are only required to keep a fraction of their deposits and wealth in their vaults and they are allowed to lend 'money' they don't have called 'credit'. Go into a bank for a loan and if you succeed

the banker will not move any real wealth into your account. They will type into your account the amount of the agreed 'loan' – say £100,000. This is not wealth that really exists; it is non-existent, fresh-air, created-out-of-nothing 'credit' which has never, does not, and will never exist except in theory. Credit is backed by nothing except wind and only has buying power because people think that it has buying power and accept it in return for property, goods and services. I have described this situation as like those cartoon characters you see chasing each other and when they run over the edge of a cliff they keep running forward on fresh air until one of them looks down, realises what's happened, and they all crash into the ravine. The whole foundation of the Sabbatian financial system is to stop people looking down except for periodic moments when they want to crash the system (as in 2008 and 2020 ongoing) and reap the rewards from all the property, businesses and wealth their borrowers had signed over as 'collateral' in return for a 'loan' of fresh air. Most people think that money is somehow created by governments when it comes into existence from the start as a debt through banks 'lending' illusory money called credit. Yes, the very currency of exchange is a *debt* from day one issued as an interest-bearing loan. Why don't governments create money interest-free and lend it to their people interest-free? Governments are controlled by Sabbatians and the financial system is controlled by Sabbatians for whom interest-free money would be a nightmare come true. Sabbatians underpin their financial domination through their global network of central banks, including the privately-owned US Federal Reserve and Britain's Bank of England, and this is orchestrated by a privately-owned central bank coordination body called the Bank for International Settlements in Basle, Switzerland, created by the usual suspects including the Rockefellers and Rothschilds. Central bank chiefs don't answer to governments or the people. They answer to the Bank for International Settlements or, in other words, the Global Cult which is dominated today by Sabbatians.

Built-in disaster

There are so many constituent scams within the overall banking scam. When you take out a loan of thin-air credit only the amount of that loan is theoretically brought into circulation to add to the amount in circulation; but you are paying back the principle plus interest. The additional interest is not created and this means that with every 'loan' there is a shortfall in the money in circulation between what is borrowed and what has to be paid back. There is never even close to enough money in circulation to repay all outstanding public and private debt including interest. Coldly weaved in the very fabric of the system is the certainty that some will lose their homes, businesses and possessions to the banking 'lender'. This is less obvious in times of 'boom' when the amount of money in circulation (and the debt) is expanding through more people wanting and getting loans. When a downturn comes and the money supply contracts it becomes painfully obvious that there is not enough money to service all debt and interest. This is less obvious in times of 'boom' when the amount of money in circulation (and the debt) is expanding through more people wanting and getting loans. When a downturn comes and the money supply contracts and it becomes painfully obvious – as in 2008 and currently – that there is not enough money to service all debt and interest.

Sabbatian banksters have been leading the human population through a calculated series of booms (more debt incurred) and busts (when the debt can't be repaid and the banks get the debtor's tangible wealth in exchange for non-existent 'credit'). With each 'bust' Sabbatian bankers have absorbed more of the world's tangible wealth and we end up with the One-percent. Governments are in bankruptcy levels of debt to the same system and are therefore owned by a system they do not control. The Federal Reserve, 'America's central bank', is privately-owned and American presidents only nominally appoint its chairman or woman to maintain the illusion that it's an arm of government. It's not. The 'Fed' is a cartel of private banks which handed billions to its associates and friends after the crash of 2008 and has been Sabbatian-controlled since it was manipulated into being in 1913 through the covert trickery of Rothschild banking agents Jacob Schiff and Paul

Warburg, and the Sabbatian Rockefeller family. Somehow from a Jewish population of two-percent and globally 0.2 percent (Sabbatian interlopers remember are far smaller) ultra-Zionists headed the Federal Reserve for 31 years between 1987 and 2018 in the form of Alan Greenspan, Bernard Bernanke and Janet Yellen (now Biden's Treasury Secretary) with Yellen's deputy chairman a Israeli-American dual citizen and ultra-Zionist Stanley Fischer, a former governor of the Bank of Israel. Ultra-Zionist Fed chiefs spanned the presidencies of Ronald Reagan ('Republican'), Father George Bush ('Republican'), Bill Clinton ('Democrat'), Boy George Bush ('Republican') and Barack Obama ('Democrat'). We should really add the pre-Greenspan chairman, Paul Adolph Volcker, 'appointed' by Jimmy Carter ('Democrat') who ran the Fed between 1979 and 1987 during the Carter and Reagan administrations before Greenspan took over. Volcker was a long-time associate and business partner of the Rothschilds. No matter what the 'party' officially in power the United States economy was directed by the same force. Here are members of the Obama, Trump and Biden administrations and see if you can make out a common theme.

Barack Obama ('Democrat')

Ultra-Zionists Robert Rubin, Larry Summers, and Timothy Geithner ran the US Treasury in the Clinton administration and two of them reappeared with Obama. Ultra-Zionist Fed chairman Alan Greenspan had manipulated the crash of 2008 through deregulation and jumped ship just before the disaster to make way for ultra-Zionist Bernard Bernanke to hand out trillions to Sabbatian 'too big to fail' banks and businesses, including the ubiquitous ultra-Zionist Goldman Sachs which has an ongoing staff revolving door operation between itself and major financial positions in government worldwide. Obama inherited the fallout of the crash when he took office in January, 2009, and fortunately he had the support of his ultra-Zionist White House Chief of Staff Rahm Emmanuel, son of a terrorist who helped to bomb Israel into being in 1948, and his ultra-Zionist senior adviser David Axelrod, chief strategist in Obama's two

successful presidential campaigns. Emmanuel, later mayor of Chicago and former senior fundraiser and strategist for Bill Clinton, is an example of the Sabbatian policy after Israel was established of migrating insider families to America so their children would be born American citizens. ‘Obama’ chose this financial team throughout his administration to respond to the Sabbatian-instigated crisis:

Timothy Geithner (ultra-Zionist) Treasury Secretary; Jacob J. Lew, Treasury Secretary; Larry Summers (ultra-Zionist), director of the White House National Economic Council; Paul Adolph Volcker (Rothschild business partner), chairman of the Economic Recovery Advisory Board; Peter Orszag (ultra-Zionist), director of the Office of Management and Budget overseeing all government spending; Penny Pritzker (ultra-Zionist), Commerce Secretary; Jared Bernstein (ultra-Zionist), chief economist and economic policy adviser to Vice President Joe Biden; Mary Schapiro (ultra-Zionist), chair of the Securities and Exchange Commission (SEC); Gary Gensler (ultra-Zionist), chairman of the Commodity Futures Trading Commission (CFTC); Sheila Bair (ultra-Zionist), chair of the Federal Deposit Insurance Corporation (FDIC); Karen Mills (ultra-Zionist), head of the Small Business Administration (SBA); Kenneth Feinberg (ultra-Zionist), Special Master for Executive [bail-out] Compensation. Feinberg would be appointed to oversee compensation (with strings) to 9/11 victims and families in a campaign to stop them having their day in court to question the official story. At the same time ultra-Zionist Bernard Bernanke was chairman of the Federal Reserve and these are only some of the ultra-Zionists with allegiance to Sabbatian-controlled Israel in the Obama government. Obama’s biggest corporate donor was ultra-Zionist Goldman Sachs which had employed many in his administration.

Donald Trump ('Republican')

Trump claimed to be an outsider (he wasn’t) who had come to ‘drain the swamp’. He embarked on this goal by immediately appointing ultra-Zionist Steve Mnuchin, a Goldman Sachs employee for 17

years, as his Treasury Secretary. Others included Gary Cohn (ultra-Zionist), chief operating officer of Goldman Sachs, his first Director of the National Economic Council and chief economic adviser, who was later replaced by Larry Kudlow (ultra-Zionist). Trump's senior adviser throughout his four years in the White House was his sinister son-in-law Jared Kushner, a life-long friend of Israel Prime Minister Benjamin Netanyahu. Kushner is the son of a convicted crook who was pardoned by Trump in his last days in office. Other ultra-Zionists in the Trump administration included: Stephen Miller, Senior Policy Adviser; Avrahm Berkowitz, Deputy Adviser to Trump and his Senior Adviser Jared Kushner; Ivanka Trump, Adviser to the President, who converted to Judaism when she married Jared Kushner; David Friedman, Trump lawyer and Ambassador to Israel; Jason Greenblatt, Trump Organization executive vice president and chief legal officer, who was made Special Representative for International Negotiations and the Israeli-Palestinian Conflict; Rod Rosenstein, Deputy Attorney General; Elliot Abrams, Special Representative for Venezuela, then Iran; John Eisenberg, National Security Council Legal Adviser and Deputy Council to the President for National Security Affairs; Anne Neuberger, Deputy National Manager, National Security Agency; Ezra Cohen-Watnick, Acting Under Secretary of Defense for Intelligence; Elan Carr, Special Envoy to monitor and combat anti-Semitism; Len Khodorkovsky, Deputy Special Envoy to monitor and combat anti-Semitism; Reed Cordish, Assistant to the President, Intragovernmental and Technology Initiatives. Trump Vice President Mike Pence and Secretary of State Mike Pompeo, both Christian Zionists, were also vehement supporters of Israel and its goals and ambitions.

Donald 'free-speech believer' Trump pardoned a number of financial and violent criminals while ignoring calls to pardon Julian Assange and Edward Snowden whose crimes are revealing highly relevant information about government manipulation and corruption and the widespread illegal surveillance of the American people by US 'security' agencies. It's so good to know that Trump is on the side of freedom and justice and not mega-criminals with

allegiance to Sabbatian-controlled Israel. These included a pardon for Israeli spy Jonathan Pollard who was jailed for life in 1987 under the Espionage Act. Aviem Sella, the Mossad agent who recruited Pollard, was also pardoned by Trump while Assange sat in jail and Snowden remained in exile in Russia. Sella had 'fled' (was helped to escape) to Israel in 1987 and was never extradited despite being charged under the Espionage Act. A Trump White House statement said that Sella's clemency had been 'supported by Benjamin Netanyahu, Ron Dermer, Israel's US Ambassador, David Friedman, US Ambassador to Israel and Miriam Adelson, wife of leading Trump donor Sheldon Adelson who died shortly before. Other friends of Jared Kushner were pardoned along with Sholom Weiss who was believed to be serving the longest-ever white-collar prison sentence of more than 800 years in 2000. The sentence was commuted of Ponzi-schemer Eliyahu Weinstein who defrauded Jews and others out of \$200 million. I did mention that Assange and Snowden were ignored, right? Trump gave Sabbatians almost everything they asked for in military and political support, moving the US Embassy from Tel Aviv to Jerusalem with its critical symbolic and literal implications for Palestinian statehood, and the 'deal of the Century' designed by Jared Kushner and David Friedman which gave the Sabbatian Israeli government the green light to substantially expand its already widespread program of building illegal Jewish-only settlements in the occupied land of the West Bank. This made a two-state 'solution' impossible by seizing all the land of a potential Palestinian homeland and that had been the plan since 1948 and then 1967 when the Arab-controlled Gaza Strip, West Bank, Sinai Peninsula and Syrian Golan Heights were occupied by Israel. All the talks about talks and road maps and delays have been buying time until the West Bank was physically occupied by Israeli real estate. Trump would have to be a monumentally ill-informed idiot not to see that this was the plan he was helping to complete. The Trump administration was in so many ways the Kushner administration which means the Netanyahu administration which means the Sabbatian administration. I understand why many opposing Cult fascism in all its forms gravitated to Trump, but he

was a crucial part of the Sabbatian plan and I will deal with this in the next chapter.

Joe Biden ('Democrat')

A barely cognitive Joe Biden took over the presidency in January, 2021, along with his fellow empty shell, Vice-President Kamala Harris, as the latest Sabbatian gofers to enter the White House. Names on the door may have changed and the 'party' – the force behind them remained the same as Zionists were appointed to a stream of pivotal areas relating to Sabbatian plans and policy. They included: Janet Yellen, Treasury Secretary, former head of the Federal Reserve, and still another ultra-Zionist running the US Treasury after Mnuchin (Trump), Lew and Geithner (Obama), and Summers and Rubin (Clinton); Anthony Blinken, Secretary of State; Wendy Sherman, Deputy Secretary of State (so that's 'Biden's' Sabbatian foreign policy sorted); Jeff Zients, White House coronavirus coordinator; Rochelle Walensky, head of the Centers for Disease Control; Rachel Levine, transgender deputy health secretary (that's 'Covid' hoax policy under control); Merrick Garland, Attorney General; Alejandro Mayorkas, Secretary of Homeland Security; Cass Sunstein, Homeland Security with responsibility for new immigration laws; Avril Haines, Director of National Intelligence; Anne Neuberger, National Security Agency cybersecurity director (note, cybersecurity); David Cohen, CIA Deputy Director; Ronald Klain, Biden's Chief of Staff (see Rahm Emanuel); Eric Lander, a 'leading geneticist', Office of Science and Technology Policy director (see Smart Grid, synthetic biology agenda); Jessica Rosenworcel, acting head of the Federal Communications Commission (FCC) which controls Smart Grid technology policy and electromagnetic communication systems including 5G. How can it be that so many pivotal positions are held by two-percent of the American population and 0.2 percent of the world population administration after administration no matter who is the president and what is the party? It's a coincidence? Of course it's not and this is why Sabbatians have built their colossal global web of interlocking 'anti-

hate' hate groups to condemn anyone who asks these glaring questions as an 'anti-Semite'. The way that Jewish people horrifically abused in Sabbatian-backed Nazi Germany are exploited to this end is stomach-turning and disgusting beyond words.

Political fusion

Sabbatian manipulation has reversed the roles of Republicans and Democrats and the same has happened in Britain with the Conservative and Labour Parties. Republicans and Conservatives were always labelled the 'right' and Democrats and Labour the 'left', but look at the policy positions now and the Democrat-Labour 'left' has moved further to the 'right' than Republicans and Conservatives under the banner of 'Woke', the Cult-created far-right tyranny. Where once the Democrat-Labour 'left' defended free speech and human rights they now seek to delete them and as I said earlier despite the 'Covid' fascism of the Jackboot Johnson Conservative government in the UK the Labour Party of leader Keir Starmer demanded even more extreme measures. The Labour Party has been very publicly absorbed by Sabbatians after a political and media onslaught against the previous leader, the weak and inept Jeremy Corbyn, over made-up allegations of 'anti-Semitism' both by him and his party. The plan was clear with this 'anti-Semite' propaganda and what was required in response was a swift and decisive 'fuck off' from Corbyn and a statement to expose the Anti-Semitism Industry (Sabbatian) attempt to silence Labour criticism of the Israeli government (Sabbatians) and purge the party of all dissent against the extremes of ultra-Zionism (Sabbatians). Instead Corbyn and his party fell to their knees and appeased the abusers which, by definition, is impossible. Appeasing one demand leads only to a new demand to be appeased until takeover is complete. Like I say – 'fuck off' would have been a much more effective policy and I have used it myself with great effect over the years when Sabbatians are on my case which is most of the time. I consider that fact a great compliment, by the way. The outcome of the Labour Party capitulation is that we now have a Sabbatian-controlled

Conservative Party ‘opposed’ by a Sabbatian-controlled Labour Party in a one-party Sabbatian state that hurtles towards the extremes of tyranny (the Sabbatian cult agenda). In America the situation is the same. Labour’s Keir Starmer spends his days on his knees with his tongue out pointing to Tel Aviv, or I guess now Jerusalem, while Boris Johnson has an ‘anti-Semitism czar’ in the form of former Labour MP John Mann who keeps Starmer company on his prayer mat.

Sabbatian influence can be seen in Jewish members of the Labour Party who have been ejected for criticism of Israel including those from families that suffered in Nazi Germany. Sabbatians despise real Jewish people and target them even more harshly because it is so much more difficult to dub them ‘anti-Semitic’ although in their desperation they do try.

CHAPTER THREE

The Pushbacker sting

Until you realize how easy it is for your mind to be manipulated, you remain the puppet of someone else's game

Evita Ochel

I will use the presidencies of Trump and Biden to show how the manipulation of the one-party state plays out behind the illusion of political choice across the world. No two presidencies could – on the face of it – be more different and apparently at odds in terms of direction and policy.

A Renegade Mind sees beyond the obvious and focuses on outcomes and consequences and not image, words and waffle. The Cult embarked on a campaign to divide America between those who blindly support its agenda (the mentality known as 'Woke') and those who are pushing back on where the Cult and its Sabbatians want to go. This presents infinite possibilities for dividing and ruling the population by setting them at war with each other and allows a perceptual ring fence of demonisation to encircle the Pushbackers in a modern version of the Little Big Horn in 1876 when American cavalry led by Lieutenant Colonel George Custer were drawn into a trap, surrounded and killed by Native American tribes defending their land of thousands of years from being seized by the government. In this modern version the roles are reversed and it's those defending themselves from the Sabbatian government who are surrounded and the government that's seeking to destroy them. This trap was set years ago and to explain how we must return to 2016

and the emergence of Donald Trump as a candidate to be President of the United States. He set out to overcome the best part of 20 other candidates in the Republican Party before and during the primaries and was not considered by many in those early stages to have a prayer of living in the White House. The Republican Party was said to have great reservations about Trump and yet somehow he won the nomination. When you know how American politics works – politics in general – there is no way that Trump could have become the party's candidate unless the Sabbatian-controlled 'Neocons' that run the Republican Party wanted that to happen. We saw the proof in emails and documents made public by WikiLeaks that the Democratic Party hierarchy, or Democons, systematically undermined the campaign of Bernie Sanders to make sure that Sabbatian gofer Hillary Clinton won the nomination to be their presidential candidate. If the Democons could do that then the Neocons in the Republican Party could have derailed Trump in the same way. But they didn't and at that stage I began to conclude that Trump could well be the one chosen to be president. If that was the case the 'why' was pretty clear to see – the goal of dividing America between Cult agenda-supporting Wokers and Pushbackers who gravitated to Trump because he was telling them what they wanted to hear. His constituency of support had been increasingly ignored and voiceless for decades and profoundly through the eight years of Sabbatian puppet Barack Obama. Now here was someone speaking their language of pulling back from the incessant globalisation of political and economic power, the exporting of American jobs to China and elsewhere by 'American' (Sabbatian) corporations, the deletion of free speech, and the mass immigration policies that had further devastated job opportunities for the urban working class of all races and the once American heartlands of the Midwest.

Beware the forked tongue

Those people collectively sighed with relief that at last a political leader was apparently on their side, but another trait of the Renegade Mind is that you look even harder at people telling you

what you want to hear than those who are telling you otherwise. Obviously as I said earlier people wish what they want to hear to be true and genuine and they are much more likely to believe that than someone saying what they don't want to here and don't want to be true. Sales people are taught to be skilled in eliciting by calculated questioning what their customers want to hear and repeating that back to them as their own opinion to get their targets to like and trust them. Assets of the Cult are also sales people in the sense of selling perception. To read Cult manipulation you have to play the long and expanded game and not fall for the Vaudeville show of party politics. Both American parties are vehicles for the Cult and they exploit them in different ways depending on what the agenda requires at that moment. Trump and the Republicans were used to be the focus of dividing America and isolating Pushbackers to open the way for a Biden presidency to become the most extreme in American history by advancing the full-blown Woke (Cult) agenda with the aim of destroying and silencing Pushbackers now labelled Nazi Trump supporters and white supremacists.

Sabbatians wanted Trump in office for the reasons described by ultra-Zionist Saul Alinsky (1909-1972) who was promoting the Woke philosophy through 'community organising' long before anyone had heard of it. In those days it still went by its traditional name of Marxism. The reason for the manipulated Trump phenomenon was laid out in Alinsky's 1971 book, *Rules for Radicals*, which was his blueprint for overthrowing democratic and other regimes and replacing them with Sabbatian Marxism. Not surprisingly his to-do list was evident in the Sabbatian French and Russian 'Revolutions' and that in China which will become very relevant in the next chapter about the 'Covid' hoax. Among Alinsky's followers have been the deeply corrupt Barack Obama, House Speaker Nancy Pelosi and Hillary Clinton who described him as a 'hero'. All three are Sabbatian stooges with Pelosi personifying the arrogant corrupt idiocy that so widely fronts up for the Cult inner core. Predictably as a Sabbatian advocate of the 'light-bringer' Alinsky features Lucifer on the dedication page of his book as the original radical who gained

his own kingdom ('Earth' as we shall see). One of Alinsky's golden radical rules was to pick an individual and focus all attention, hatred and blame on them and not to target faceless bureaucracies and corporations. *Rules for Radicals* is really a Sabbatian handbook with its contents repeatedly employed all over the world for centuries and why wouldn't Sabbatians bring to power their designer-villain to be used as the individual on which all attention, hatred and blame was bestowed? This is what they did and the only question for me is how much Trump knew that and how much he was manipulated. A bit of both, I suspect. This was Alinsky's Trump technique from a man who died in 1972. The technique has spanned history:

Pick the target, freeze it, personalize it, polarize it. Don't try to attack abstract corporations or bureaucracies. Identify a responsible individual. Ignore attempts to shift or spread the blame.

From the moment Trump came to illusory power everything was about him. It wasn't about Republican policy or opinion, but all about Trump. Everything he did was presented in negative, derogatory and abusive terms by the Sabbatian-dominated media led by Cult operations such as CNN, MSNBC, *The New York Times* and the Jeff Bezos-owned *Washington Post* – 'Pick the target, freeze it, personalize it, polarize it.' Trump was turned into a demon to be vilified by those who hated him and a demi-god loved by those who worshipped him. This, in turn, had his supporters, too, presented as equally demonic in preparation for the punchline later down the line when Biden was about to take office. It was here's a Trump, there's a Trump, everywhere a Trump, Trump. Virtually every news story or happening was filtered through the lens of 'The Donald'. You loved him or hated him and which one you chose was said to define you as Satan's spawn or a paragon of virtue. Even supporting some Trump policies or statements and not others was enough for an assault on your character. No shades of grey were or are allowed. Everything is black and white (literally and figuratively). A Californian I knew had her head utterly scrambled by her hatred for Trump while telling people they should love each other. She was so totally consumed by

Trump Derangement Syndrome as it became to be known that this glaring contradiction would never have occurred to her. By definition anyone who criticised Trump or praised his opponents was a hero and this lady described Joe Biden as 'a kind, honest gentleman' when he's a provable liar, mega-crook and vicious piece of work to boot. Sabbatians had indeed divided America using Trump as the fall-guy and all along the clock was ticking on the consequences for his supporters.

In hock to his masters

Trump gave Sabbatians via Israel almost everything they wanted in his four years. Ask and you shall receive was the dynamic between himself and Benjamin Netanyahu orchestrated by Trump's ultra-Zionist son-in-law Jared Kushner, his ultra-Zionist Ambassador to Israel, David Friedman, and ultra-Zionist 'Israel adviser', Jason Greenblatt. The last two were central to the running and protecting from collapse of his business empire, the Trump Organisation, and colossal business failures made him forever beholden to Sabbatian networks that bailed him out. By the start of the 1990s Trump owed \$4 billion to banks that he couldn't pay and almost \$1 billion of that was down to him personally and not his companies. This mega-disaster was the result of building two new casinos in Atlantic City and buying the enormous Taj Mahal operation which led to crippling debt payments. He had borrowed fantastic sums from 72 banks with major Sabbatian connections and although the scale of debt should have had him living in a tent alongside the highway they never foreclosed. A plan was devised to lift Trump from the mire by BT Securities Corporation and Rothschild Inc. and the case was handled by Wilber Ross who had worked for the Rothschilds for 27 years. Ross would be named US Commerce Secretary after Trump's election. Another crucial figure in saving Trump was ultra-Zionist 'investor' Carl Icahn who bought the Taj Mahal casino. Icahn was made special economic adviser on financial regulation in the Trump administration. He didn't stay long but still managed to find time to make a tidy sum of a reported \$31.3 million when he sold his

holdings affected by the price of steel three days before Trump imposed a 235 percent tariff on steel imports. What amazing bits of luck these people have. Trump and Sabbatian operatives have long had a close association and his mentor and legal adviser from the early 1970s until 1986 was the dark and genetically corrupt ultra-Zionist Roy Cohn who was chief counsel to Senator Joseph McCarthy's 'communist' witch-hunt in the 1950s. *Esquire* magazine published an article about Cohn with the headline 'Don't mess with Roy Cohn'. He was described as the most feared lawyer in New York and 'a ruthless master of dirty tricks ... [with] ... more than one Mafia Don on speed dial'. Cohn's influence, contacts, support and protection made Trump a front man for Sabbatians in New York with their connections to one of Cohn's many criminal employers, the 'Russian' Sabbatian Mafia. Israel-centric media mogul Rupert Murdoch was introduced to Trump by Cohn and they started a long friendship. Cohn died in 1986 weeks after being disbarred for unethical conduct by the Appellate Division of the New York State Supreme Court. The wheels of justice do indeed run slow given the length of Cohn's crooked career.

QAnon-sense

We are asked to believe that Donald Trump with his fundamental connections to Sabbatian networks and operatives has been leading the fight to stop the Sabbatian agenda for the fascistic control of America and the world. Sure he has. A man entrapped during his years in the White House by Sabbatian operatives and whose biggest financial donor was casino billionaire Sheldon Adelson who was Sabbatian to his DNA?? Oh, do come on. Trump has been used to divide America and isolate Pushbackers on the Cult agenda under the heading of 'Trump supporters', 'insurrectionists' and 'white supremacists'. The US Intelligence/Mossad Psyop or psychological operation known as QAnon emerged during the Trump years as a central pillar in the Sabbatian campaign to lead Pushbackers into the trap set by those that wished to destroy them. I knew from the start that QAnon was a scam because I had seen the same scenario many

times before over 30 years under different names and I had written about one in particular in the books. ‘Not again’ was my reaction when QAnon came to the fore. The same script is pulled out every few years and a new name added to the letterhead. The story always takes the same form: ‘Insiders’ or ‘the good guys’ in the government-intelligence-military ‘Deep State’ apparatus were going to instigate mass arrests of the ‘bad guys’ which would include the Rockefellers, Rothschilds, Barack Obama, Hillary Clinton, George Soros, etc., etc. Dates are given for when the ‘good guys’ are going to move in, but the dates pass without incident and new dates are given which pass without incident. The central message to Pushbackers in each case is that they don’t have to do anything because there is ‘a plan’ and it is all going to be sorted by the ‘good guys’ on the inside. ‘Trust the plan’ was a QAnon mantra when the only plan was to misdirect Pushbackers into putting their trust in a Psyop they believed to be real. Beware, beware, those who tell you what you want to hear and always check it out. Right up to Biden’s inauguration QAnon was still claiming that ‘the Storm’ was coming and Trump would stay on as president when Biden and his cronies were arrested and jailed. It was never going to happen and of course it didn’t, but what did happen as a result provided that punchline to the Sabbatian Trump/QAnon Psyop.

On January 6th, 2021, a very big crowd of Trump supporters gathered in the National Mall in Washington DC down from the Capitol Building to protest at what they believed to be widespread corruption and vote fraud that stopped Trump being re-elected for a second term as president in November, 2020. I say as someone that does not support Trump or Biden that the evidence is clear that major vote-fixing went on to favour Biden, a man with cognitive problems so advanced he can often hardly string a sentence together without reading the words written for him on the Teleprompter. Glaring ballot discrepancies included serious questions about electronic voting machines that make vote rigging a comparative cinch and hundreds of thousands of paper votes that suddenly appeared during already advanced vote counts and virtually all of

them for Biden. Early Trump leads in crucial swing states suddenly began to close and disappear. The pandemic hoax was used as the excuse to issue almost limitless numbers of mail-in ballots with no checks to establish that the recipients were still alive or lived at that address. They were sent to streams of people who had not even asked for them. Private organisations were employed to gather these ballots and who knows what they did with them before they turned up at the counts. The American election system has been manipulated over decades to become a sick joke with more holes than a Swiss cheese for the express purpose of dictating the results. Then there was the criminal manipulation of information by Sabbatian tech giants like Facebook, Twitter and Google-owned YouTube which deleted pro-Trump, anti-Biden accounts and posts while everything in support of Biden was left alone. Sabbatians wanted Biden to win because after the dividing of America it was time for full-on Woke and every aspect of the Cult agenda to be unleashed.

Hunter gatherer

Extreme Silicon Valley bias included blocking information by the *New York Post* exposing a Biden scandal that should have ended his bid for president in the final weeks of the campaign. Hunter Biden, his monumentally corrupt son, is reported to have sent a laptop to be repaired at a local store and failed to return for it. Time passed until the laptop became the property of the store for non-payment of the bill. When the owner saw what was on the hard drive he gave a copy to the FBI who did nothing even though it confirmed widespread corruption in which the Joe Biden family were using his political position, especially when he was vice president to Obama, to make multiple millions in countries around the world and most notably Ukraine and China. Hunter Biden's one-time business partner Tony Bobulinski went public when the story broke in the *New York Post* to confirm the corruption he saw and that Joe Biden not only knew what was going on he also profited from the spoils. Millions were handed over by a Chinese company with close

connections – like all major businesses in China – to the Chinese communist party of President Xi Jinping. Joe Biden even boasted at a meeting of the Cult's World Economic Forum that as vice president he had ordered the government of Ukraine to fire a prosecutor. What he didn't mention was that the same man just happened to be investigating an energy company which was part of Hunter Biden's corrupt portfolio. The company was paying him big bucks for no other reason than the influence his father had. Overnight Biden's presidential campaign should have been over given that he had lied publicly about not knowing what his son was doing. Instead almost the entire Sabbatian-owned mainstream media and Sabbatian-owned Silicon Valley suppressed circulation of the story. This alone went a mighty way to rigging the election of 2020. Cult assets like Mark Zuckerberg at Facebook also spent hundreds of millions to be used in support of Biden and vote 'administration'.

The Cult had used Trump as the focus to divide America and was now desperate to bring in moronic, pliable, corrupt Biden to complete the double-whammy. No way were they going to let little things like the will of the people thwart their plan. Silicon Valley widely censored claims that the election was rigged because it *was* rigged. For the same reason anyone claiming it was rigged was denounced as a 'white supremacist' including the pathetically few Republican politicians willing to say so. Right across the media where the claim was mentioned it was described as a 'false claim' even though these excuses for 'journalists' would have done no research into the subject whatsoever. Trump won seven million more votes than any sitting president had ever achieved while somehow a cognitively-challenged soon to be 78-year-old who was hidden away from the public for most of the campaign managed to win more votes than any presidential candidate in history. It makes no sense. You only had to see election rallies for both candidates to witness the enthusiasm for Trump and the apathy for Biden. Tens of thousands would attend Trump events while Biden was speaking in empty car parks with often only television crews attending and framing their shots to hide the fact that no one was there. It was pathetic to see

footage come to light of Biden standing at a podium making speeches only to TV crews and party fixers while reading the words written for him on massive Teleprompter screens. So, yes, those protestors on January 6th had a point about election rigging, but some were about to walk into a trap laid for them in Washington by the Cult Deep State and its QAnon Psyop. This was the Capitol Hill riot ludicrously dubbed an ‘insurrection’.

The spider and the fly

Renegade Minds know there are not two ‘sides’ in politics, only one side, the Cult, working through all ‘sides’. It’s a stage show, a puppet show, to direct the perceptions of the population into focusing on diversions like parties and candidates while missing the puppeteers with their hands holding all the strings. The Capitol Hill ‘insurrection’ brings us back to the Little Big Horn. Having created two distinct opposing groupings – Woke and Pushbackers – the trap was about to be sprung. Pushbackers were to be encircled and isolated by associating them all in the public mind with Trump and then labelling Trump as some sort of Confederate leader. I knew immediately that the Capitol riot was a set-up because of two things. One was how easy the rioters got into the building with virtually no credible resistance and secondly I could see – as with the ‘Covid’ hoax in the West at the start of 2020 – how the Cult could exploit the situation to move its agenda forward with great speed. My experience of Cult techniques and activities over more than 30 years has showed me that while they do exploit situations they haven’t themselves created this never happens with events of fundamental agenda significance. Every time major events giving cultists the excuse to rapidly advance their plan you find they are manipulated into being for the specific reason of providing that excuse – Problem-Reaction-Solution. Only a tiny minority of the huge crowd of Washington protestors sought to gain entry to the Capitol by smashing windows and breaching doors. That didn’t matter. The whole crowd and all Pushbackers, even if they did not support Trump, were going to be lumped together as dangerous

insurrectionists and conspiracy theorists. The latter term came into widespread use through a CIA memo in the 1960s aimed at discrediting those questioning the nonsensical official story of the Kennedy assassination and it subsequently became widely employed by the media. It's still being used by inept 'journalists' with no idea of its origin to discredit anyone questioning anything that authority claims to be true. When you are perpetrating a conspiracy you need to discredit the very word itself even though the dictionary definition of conspiracy is merely 'the activity of secretly planning with other people to do something bad or illegal' and 'a general agreement to keep silent about a subject for the purpose of keeping it secret'. On that basis there are conspiracies almost wherever you look. For obvious reasons the Cult and its lapdog media have to claim there are no conspiracies even though the word appears in state laws as with conspiracy to defraud, to murder, and to corrupt public morals.

Agent provocateurs are widely used by the Cult Deep State to manipulate genuine people into acting in ways that suit the desired outcome. By genuine in this case I mean protestors genuinely supporting Trump and claims that the election was stolen. In among them, however, were agents of the state wearing the garb of Trump supporters and QAnon to pump-prime the Capitol riot which some genuine Trump supporters naively fell for. I described the situation as 'Come into my parlour said the spider to the fly'. Leaflets appeared through the Woke paramilitary arm Antifa, the anti-fascist fascists, calling on supporters to turn up in Washington looking like Trump supporters even though they hated him. Some of those arrested for breaching the Capitol Building were sourced to Antifa and its stable mate Black Lives Matter. Both organisations are funded by Cult billionaires and corporations. One man charged for the riot was according to his lawyer a former FBI agent who had held top secret security clearance for 40 years. Attorney Thomas Plofchan said of his client, 66-year-old Thomas Edward Caldwell:

He has held a Top Secret Security Clearance since 1979 and has undergone multiple Special Background Investigations in support of his clearances. After retiring from the Navy, he

worked as a section chief for the Federal Bureau of Investigation from 2009-2010 as a GS-12 [mid-level employee].

He also formed and operated a consulting firm performing work, often classified, for U.S government customers including the US Drug Enforcement Agency, Department of Housing and Urban Development, the US Coast Guard, and the US Army Personnel Command.

A judge later released Caldwell pending trial in the absence of evidence about a conspiracy or that he tried to force his way into the building. *The New York Post* reported a 'law enforcement source' as saying that 'at least two known Antifa members were spotted' on camera among Trump supporters during the riot while one of the rioters arrested was John Earle Sullivan, a seriously extreme Black Lives Matter Trump-hater from Utah who was previously arrested and charged in July, 2020, over a BLM-Antifa riot in which drivers were threatened and one was shot. Sullivan is the founder of Utah-based Insurgence USA which is an affiliate of the Cult-created-and-funded Black Lives Matter movement. Footage appeared and was then deleted by Twitter of Trump supporters calling out Antifa infiltrators and a group was filmed changing into pro-Trump clothing before the riot. Security at the building was *pathetic* – as planned. Colonel Leroy Fletcher Prouty, a man with long experience in covert operations working with the US security apparatus, once described the tell-tale sign to identify who is involved in an assassination. He said:

No one has to direct an assassination – it happens. The active role is played secretly by permitting it to happen. This is the greatest single clue. Who has the power to call off or reduce the usual security precautions?

This principle applies to many other situations and certainly to the Capitol riot of January 6th, 2021.

The sting

With such a big and potentially angry crowd known to be gathering near the Capitol the security apparatus would have had a major police detail to defend the building with National Guard troops on

standby given the strength of feeling among people arriving from all over America encouraged by the QAnon Psyop and statements by Donald Trump. Instead Capitol Police ‘security’ was flimsy, weak, and easily breached. The same number of officers was deployed as on a regular day and that is a blatant red flag. They were not staffed or equipped for a possible riot that had been an obvious possibility in the circumstances. No protective and effective fencing worth the name was put in place and there were no contingency plans. The whole thing was basically a case of standing aside and waving people in. Once inside police mostly backed off apart from one Capitol police officer who ridiculously shot dead unarmed Air Force veteran protestor Ashli Babbitt without a warning as she climbed through a broken window. The ‘investigation’ refused to name or charge the officer after what must surely be considered a murder in the circumstances. They just lifted a carpet and swept. The story was endlessly repeated about five people dying in the ‘armed insurrection’ when there was no report of rioters using weapons. Apart from Babbitt the other four died from a heart attack, strokes and apparently a drug overdose. Capitol police officer Brian Sicknick was reported to have died after being bludgeoned with a fire extinguisher when he was alive after the riot was over and died later of what the Washington Medical Examiner’s Office said was a stroke. Sicknick had no external injuries. The lies were delivered like rapid fire. There was a narrative to build with incessant repetition of the lie until the lie became the accepted ‘everybody knows that’ truth. The ‘Big Lie’ technique of Nazi Propaganda Minister Joseph Goebbels is constantly used by the Cult which was behind the Nazis and is today behind the ‘Covid’ and ‘climate change’ hoaxes. Goebbels said:

If you tell a lie big enough and keep repeating it, people will eventually come to believe it. The lie can be maintained only for such time as the State can shield the people from the political, economic and/or military consequences of the lie. It thus becomes vitally important for the State to use all of its powers to repress dissent, for the truth is the mortal enemy of the lie, and thus by extension, the truth is the greatest enemy of the State.

Most protestors had a free run of the Capitol Building. This allowed pictures to be taken of rioters in iconic parts of the building including the Senate chamber which could be used as propaganda images against all Pushbackers. One Congresswoman described the scene as ‘the worst kind of non-security anybody could ever imagine’. Well, the first part was true, but someone obviously did imagine it and made sure it happened. Some photographs most widely circulated featured people wearing QAnon symbols and now the Psyop would be used to dub all QAnon followers with the ubiquitous fit-all label of ‘white supremacist’ and ‘insurrectionists’. When a Muslim extremist called Noah Green drove his car at two police officers at the Capitol Building killing one in April, 2021, there was no such political and media hysteria. They were just disappointed he wasn’t white.

The witch-hunt

Government prosecutor Michael Sherwin, an aggressive, dark-eyed, professional Rottweiler led the ‘investigation’ and to call it over the top would be to underestimate reality a thousand fold. Hundreds were tracked down and arrested for the crime of having the wrong political views and people were jailed who had done nothing more than walk in the building, committed no violence or damage to property, took a few pictures and left. They were labelled a ‘threat to the Republic’ while Biden sat in the White House signing executive orders written for him that were dismantling ‘the Republic’. Even when judges ruled that a mother and son should not be in jail the government kept them there. Some of those arrested have been badly beaten by prison guards in Washington and lawyers for one man said he suffered a fractured skull and was made blind in one eye. Meanwhile a woman is shot dead for no reason by a Capitol Police officer and we are not allowed to know who he is never mind what has happened to him although that will be *nothing*. The Cult’s QAnon/Trump sting to identify and isolate Pushbackers and then target them on the road to crushing and deleting them was a resounding success. You would have thought the Russians had

invaded the building at gunpoint and lined up senators for a firing squad to see the political and media reaction. Congresswoman Alexandria Ocasio-Cortez is a child in a woman's body, a terrible-twins, me, me, me, Woker narcissist of such proportions that words have no meaning. She said she thought she was going to die when 'insurrectionists' banged on her office door. It turned out she wasn't even in the Capitol Building when the riot was happening and the 'banging' was a Capitol Police officer. She referred to herself as a 'survivor' which is an insult to all those true survivors of violent and sexual abuse while she lives her pampered and privileged life talking drivel for a living. Her Woke colleague and fellow mega-narcissist Rashida Tlaib broke down describing the devastating effect on her, too, of *not being* in the building when the rioters were there. Ocasio-Cortez and Tlaib are members of a fully-Woke group of Congresswomen known as 'The Squad' along with Ilhan Omar and Ayanna Pressley. The Squad from what I can see can be identified by its vehement anti-white racism, anti-white men agenda, and, as always in these cases, the absence of brain cells on active duty.

The usual suspects were on the riot case immediately in the form of Democrat ultra-Zionist senators and operatives Chuck Schumer and Adam Schiff demanding that Trump be impeached for 'his part in the insurrection'. The same pair of prats had led the failed impeachment of Trump over the invented 'Russia collusion' nonsense which claimed Russia had helped Trump win the 2016 election. I didn't realise that Tel Aviv had been relocated just outside Moscow. I must find an up-to-date map. The Russia hoax was a Sabbatian operation to keep Trump occupied and impotent and to stop any rapport with Russia which the Cult wants to retain as a perceptual enemy to be pulled out at will. Puppet Biden began attacking Russia when he came to office as the Cult seeks more upheaval, division and war across the world. A two-year stage show 'Russia collusion inquiry' headed by the not-very-bright former 9/11 FBI chief Robert Mueller, with support from 19 lawyers, 40 FBI agents plus intelligence analysts, forensic accountants and other

staff, devoured tens of millions of dollars and found no evidence of Russia collusion which a ten-year-old could have told them on day one. Now the same moronic Schumer and Schiff wanted a second impeachment of Trump over the Capitol ‘insurrection’ (riot) which the arrested development of Schumer called another ‘Pearl Harbor’ while others compared it with 9/11 in which 3,000 died and, in the case of CNN, with the Rwandan genocide in the 1990s in which an estimated 500,000 to 600,000 were murdered, between 250, 000 and 500,000 women were raped, and populations of whole towns were hacked to death with machetes. To make those comparisons purely for Cult political reasons is beyond insulting to those that suffered and lost their lives and confirms yet again the callous inhumanity that we are dealing with. Schumer is a monumental idiot and so is Schiff, but they serve the Cult agenda and do whatever they’re told so they get looked after. Talking of idiots – another inane man who spanned the Russia and Capitol impeachment attempts was Senator Eric Swalwell who had the nerve to accuse Trump of collusion with the Russians while sleeping with a Chinese spy called Christine Fang or ‘Fang Fang’ which is straight out of a Bond film no doubt starring Klaus Schwab as the bloke living on a secret island and controlling laser weapons positioned in space and pointing at world capitals. Fang Fang plays the part of Bond’s infiltrator girlfriend which I’m sure she would enjoy rather more than sharing a bed with the brainless Swalwell, lying back and thinking of China. The FBI eventually warned Swalwell about Fang Fang which gave her time to escape back to the Chinese dictatorship. How very thoughtful of them. The second Trump impeachment also failed and hardly surprising when an impeachment is supposed to remove a sitting president and by the time it happened Trump was no longer president. These people are running your country America, well, officially anyway. Terrifying isn’t it?

Outcomes tell the story - always

The outcome of all this – and it’s the *outcome* on which Renegade Minds focus, not the words – was that a vicious, hysterical and

obviously pre-planned assault was launched on Pushbackers to censor, silence and discredit them and even targeted their right to earn a living. They have since been condemned as ‘domestic terrorists’ that need to be treated like Al-Qaeda and Islamic State. ‘Domestic terrorists’ is a label the Cult has been trying to make stick since the period of the Oklahoma bombing in 1995 which was blamed on ‘far-right domestic terrorists’. If you read *The Trigger* you will see that the bombing was clearly a Problem-Reaction-Solution carried out by the Deep State during a Bill Clinton administration so corrupt that no dictionary definition of the term would even nearly suffice. Nearly 30,000 troops were deployed from all over America to the empty streets of Washington for Biden’s inauguration. Ten thousand of them stayed on with the pretext of protecting the capital from insurrectionists when it was more psychological programming to normalise the use of the military in domestic law enforcement in support of the Cult plan for a police-military state. Biden’s fascist administration began a purge of ‘wrong-thinkers’ in the military which means anyone that is not on board with Woke. The Capitol Building was surrounded by a fence with razor wire and the Land of the Free was further symbolically and literally dismantled. The circle was completed with the installation of Biden and the exploitation of the QAnon Psyop.

America had never been so divided since the civil war of the 19th century, Pushbackers were isolated and dubbed terrorists and now, as was always going to happen, the Cult immediately set about deleting what little was left of freedom and transforming American society through a swish of the hand of the most controlled ‘president’ in American history leading (officially at least) the most extreme regime since the country was declared an independent state on July 4th, 1776. Biden issued undebated, dictatorial executive orders almost by the hour in his opening days in office across the whole spectrum of the Cult wish-list including diluting controls on the border with Mexico allowing thousands of migrants to illegally enter the United States to transform the demographics of America and import an election-changing number of perceived Democrat

voters. Then there were Biden deportation amnesties for the already illegally resident (estimated to be as high as 20 or even 30 million). A bill before Congress awarded American citizenship to anyone who could prove they had worked in agriculture for just 180 days in the previous two years as 'Big Ag' secured its slave labour long-term. There were the plans to add new states to the union such as Puerto Rico and making Washington DC a state. They are all parts of a plan to ensure that the Cult-owned Woke Democrats would be permanently in power.

Border – what border?

I have exposed in detail in other books how mass immigration into the United States and Europe is the work of Cult networks fuelled by the tens of billions spent to this and other ends by George Soros and his global Open Society (open borders) Foundations. The impact can be seen in America alone where the population has increased by *100 million* in little more than 30 years mostly through immigration. I wrote in *The Answer* that the plan was to have so many people crossing the southern border that the numbers become unstoppable and we are now there under Cult-owned Biden. El Salvador in Central America puts the scale of what is happening into context. A third of the population now lives in the United States, much of it illegally, and many more are on the way. The methodology is to crush Central and South American countries economically and spread violence through machete-wielding psychopathic gangs like MS-13 based in El Salvador and now operating in many American cities. Biden-imposed lax security at the southern border means that it is all but open. He said before his 'election' that he wanted to see a surge towards the border if he became president and that was the green light for people to do just that after election day to create the human disaster that followed for both America and the migrants. When that surge came the imbecilic Alexandria Ocasio-Cortez said it wasn't a 'surge' because they are 'children, not insurgents' and the term 'surge' (used by Biden) was a claim of 'white supremacists'.

This disingenuous lady may one day enter the realm of the most basic intelligence, but it won't be any time soon.

Sabbatians and the Cult are in the process of destroying America by importing violent people and gangs in among the genuine to terrorise American cities and by overwhelming services that cannot cope with the sheer volume of new arrivals. Something similar is happening in Europe as Western society in general is targeted for demographic and cultural transformation and upheaval. The plan demands violence and crime to create an environment of intimidation, fear and division and Soros has been funding the election of district attorneys across America who then stop prosecuting many crimes, reduce sentences for violent crimes and free as many violent criminals as they can. Sabbatians are creating the chaos from which order – their order – can respond in a classic Problem-Reaction-Solution. A Freemasonic moto says ‘Ordo Ab Chao’ (Order out of Chaos) and this is why the Cult is constantly creating chaos to impose a new ‘order’. Here you have the reason the Cult is constantly creating chaos. The ‘Covid’ hoax can be seen with those entering the United States by plane being forced to take a ‘Covid’ test while migrants flooding through southern border processing facilities do not. Nothing is put in the way of mass migration and if that means ignoring the government’s own ‘Covid’ rules then so be it. They know it’s all bullshit anyway. Any pushback on this is denounced as ‘racist’ by Wokers and Sabbatian fronts like the ultra-Zionist Anti-Defamation League headed by the appalling Jonathan Greenblatt which at the same time argues that Israel should not give citizenship and voting rights to more Palestinian Arabs or the ‘Jewish population’ (in truth the Sabbatian network) will lose control of the country.

Society-changing numbers

Biden’s masters have declared that countries like El Salvador are so dangerous that their people must be allowed into the United States for humanitarian reasons when there are fewer murders in large parts of many Central American countries than in US cities like

Baltimore. That is not to say Central America cannot be a dangerous place and Cult-controlled American governments have been making it so since way back, along with the dismantling of economies, in a long-term plan to drive people north into the United States. Parts of Central America are very dangerous, but in other areas the story is being greatly exaggerated to justify relaxing immigration criteria. Migrants are being offered free healthcare and education in the United States as another incentive to head for the border and there is no requirement to be financially independent before you can enter to prevent the resources of America being drained. You can't blame migrants for seeking what they believe will be a better life, but they are being played by the Cult for dark and nefarious ends. The numbers since Biden took office are huge. In February, 2021, more than 100,000 people were known to have tried to enter the US illegally through the southern border (it was 34,000 in the same month in 2020) and in March it was 170,000 – a 418 percent increase on March, 2020. These numbers are only known people, not the ones who get in unseen. The true figure for migrants illegally crossing the border in a single month was estimated by one congressman at 250,000 and that number will only rise under Biden's current policy. Gangs of murdering drug-running thugs that control the Mexican side of the border demand money – thousands of dollars – to let migrants cross the Rio Grande into America. At the same time gun battles are breaking out on the border several times a week between rival Mexican drug gangs (which now operate globally) who are equipped with sophisticated military-grade weapons, grenades and armoured vehicles. While the Capitol Building was being 'protected' from a non-existent 'threat' by thousands of troops, and others were still deployed at the time in the Cult Neocon war in Afghanistan, the southern border of America was left to its fate. This is not incompetence, it is cold calculation.

By March, 2021, there were 17,000 unaccompanied children held at border facilities and many of them are ensnared by people traffickers for paedophile rings and raped on their journey north to America. This is not conjecture – this is fact. Many of those designated

children are in reality teenage boys or older. Meanwhile Wokers posture their self-purity for encouraging poor and tragic people to come to America and face this nightmare both on the journey and at the border with the disgusting figure of House Speaker Nancy Pelosi giving disingenuous speeches about caring for migrants. The woman's evil. Wokers condemned Trump for having children in cages at the border (so did Obama, *Shhhh*), but now they are sleeping on the floor without access to a shower with one border facility 729 percent over capacity. The Biden insanity even proposed flying migrants from the southern border to the northern border with Canada for 'processing'. The whole shambles is being overseen by ultra-Zionist Secretary of Homeland Security, the moronic liar Alejandro Mayorkas, who banned news cameras at border facilities to stop Americans seeing what was happening. Mayorkas said there was not a ban on news crews; it was just that they were not allowed to film. Alongside him at Homeland Security is another ultra-Zionist Cass Sunstein appointed by Biden to oversee new immigration laws. Sunstein despises conspiracy researchers to the point where he suggests they should be banned or *taxed* for having such views. The man is not bonkers or anything. He's perfectly well-adjusted, but adjusted to what is the question. Criticise what is happening and you are a 'white supremacist' when earlier non-white immigrants also oppose the numbers which effect their lives and opportunities. Black people in poor areas are particularly damaged by uncontrolled immigration and the increased competition for work opportunities with those who will work for less. They are also losing voting power as Hispanics become more dominant in former black areas. It's a downward spiral for them while the billionaires behind the policy drone on about how much they care about black people and 'racism'. None of this is about compassion for migrants or black people – that's just wind and air. Migrants are instead being mercilessly exploited to transform America while the countries they leave are losing their future and the same is true in Europe. Mass immigration may now be the work of Woke Democrats, but it can be traced back to the 1986 Immigration Reform and Control Act (it

wasn't) signed into law by Republican hero President Ronald Reagan which gave amnesty to millions living in the United States illegally and other incentives for people to head for the southern border. Here we have the one-party state at work again.

Save me syndrome

Almost every aspect of what I have been exposing as the Cult agenda was on display in even the first days of 'Biden' with silencing of Pushbackers at the forefront of everything. A Renegade Mind will view the Trump years and QAnon in a very different light to their supporters and advocates as the dots are connected. The QAnon/Trump Psyop has given the Cult all it was looking for. We may not know how much, or little, that Trump realised he was being used, but that's a side issue. This pincer movement produced the desired outcome of dividing America and having Pushbackers isolated. To turn this around we have to look at new routes to empowerment which do not include handing our power to other people and groups through what I will call the 'Save Me Syndrome' – 'I want someone else to do it so that I don't have to'. We have seen this at work throughout human history and the QAnon/Trump Psyop is only the latest incarnation alongside all the others. Religion is an obvious expression of this when people look to a 'god' or priest to save them or tell them how to be saved and then there are 'save me' politicians like Trump. Politics is a diversion and not a 'saviour'. It is a means to block positive change, not make it possible.

Save Me Syndrome always comes with the same repeating theme of handing your power to whom or what you believe will save you while your real 'saviour' stares back from the mirror every morning. Renegade Minds are constantly vigilant in this regard and always asking the question 'What can I do?' rather than 'What can someone else do for me?' Gandhi was right when he said: 'You must be the change you want to see in the world.' We are indeed the people we have been waiting for. We are presented with a constant raft of reasons to concede that power to others and forget where the real power is. Humanity has the numbers and the Cult does not. It has to

use diversion and division to target the unstoppable power that comes from unity. Religions, governments, politicians, corporations, media, QAnon, are all different manifestations of this power-diversion and dilution. Refusing to give your power to governments and instead handing it to Trump and QAnon is not to take a new direction, but merely to recycle the old one with new names on the posters. I will explore this phenomenon as we proceed and how to break the cycles and recycles that got us here through the mists of repeating perception and so repeating history.

For now we shall turn to the most potent example in the entire human story of the consequences that follow when you give your power away. I am talking, of course, of the 'Covid' hoax.

CHAPTER FOUR

'Covid': Calculated catastrophe

Facts are threatening to those invested in fraud
DaShanne Stokes

We can easily unravel the real reason for the 'Covid pandemic' hoax by employing the Renegade Mind methodology that I have outlined this far. We'll start by comparing the long-planned Cult outcome with the 'Covid pandemic' outcome. Know the outcome and you'll see the journey.

I have highlighted the plan for the Hunger Games Society which has been in my books for so many years with the very few controlling the very many through ongoing dependency. To create this dependency it is essential to destroy independent livelihoods, businesses and employment to make the population reliant on the state (the Cult) for even the basics of life through a guaranteed pittance income. While independence of income remained these Cult ambitions would be thwarted. With this knowledge it was easy to see where the 'pandemic' hoax was going once talk of 'lockdowns' began and the closing of all but perceived 'essential' businesses to 'save' us from an alleged 'deadly virus'. Cult corporations like Amazon and Walmart were naturally considered 'essential' while mom and pop shops and stores had their doors closed by fascist decree. As a result with every new lockdown and new regulation more small and medium, even large businesses not owned by the Cult, went to the wall while Cult giants and their frontmen and women grew financially fatter by the second. Mom and pop were

denied an income and the right to earn a living and the wealth of people like Jeff Bezos (Amazon), Mark Zuckerberg (Facebook) and Sergei Brin and Larry Page (Google/Alphabet) have reached record levels. The Cult was increasing its own power through further dramatic concentrations of wealth while the competition was being destroyed and brought into a state of dependency. Lockdowns have been instigated to secure that very end and were never anything to do with health. My brother Paul spent 45 years building up a bus repair business, but lockdowns meant buses were running at a fraction of normal levels for months on end. Similar stories can told in their hundreds of millions worldwide. Efforts of a lifetime coldly destroyed by Cult multi-billionaires and their lackeys in government and law enforcement who continued to earn their living from the taxation of the people while denying the right of the same people to earn theirs. How different it would have been if those making and enforcing these decisions had to face the same financial hardships of those they affected, but they never do.

Gates of Hell

Behind it all in the full knowledge of what he is doing and why is the psychopathic figure of Cult operative Bill Gates. His puppet Tedros at the World Health Organization declared 'Covid' a pandemic in March, 2020. The WHO had changed the definition of a 'pandemic' in 2009 just a month before declaring the 'swine flu pandemic' which would not have been so under the previous definition. The same applies to 'Covid'. The definition had included... 'an infection by an infectious agent, occurring simultaneously in different countries, with a significant mortality rate relative to the proportion of the population infected'. The new definition removed the need for 'significant mortality'. The 'pandemic' has been fraudulent even down to the definition, but Gates demanded economy-destroying lockdowns, school closures, social distancing, mandatory masks, a 'vaccination' for every man, woman and child on the planet and severe consequences and restrictions for those that refused. Who gave him this power? The

Cult did which he serves like a little boy in short trousers doing what his daddy tells him. He and his psychopathic missus even smiled when they said that much worse was to come (what they knew was planned to come). Gates responded in the matter-of-fact way of all psychopaths to a question about the effect on the world economy of what he was doing:

Well, it won't go to zero but it will shrink. Global GDP is probably going to take the biggest hit ever [Gates was smiling as he said this] ... in my lifetime this will be the greatest economic hit. But you don't have a choice. People act as if you have a choice. People don't feel like going to the stadium when they might get infected ... People are deeply affected by seeing these stats, by knowing they could be part of the transmission chain, old people, their parents and grandparents, could be affected by this, and so you don't get to say ignore what is going on here.

There will be the ability to open up, particularly in rich countries, if things are done well over the next few months, but for the world at large normalcy only returns when we have largely vaccinated the entire population.

The man has no compassion or empathy. How could he when he's a psychopath like all Cult players? My own view is that even beyond that he is very seriously mentally ill. Look in his eyes and you can see this along with his crazy flailing arms. You don't do what he has done to the world population since the start of 2020 unless you are mentally ill and at the most extreme end of psychopathic. You especially don't do it when to you know, as we shall see, that cases and deaths from 'Covid' are fakery and a product of monumental figure massaging. 'These stats' that Gates referred to are based on a 'test' that's not testing for the 'virus' as he has known all along. He made his fortune with big Cult support as an infamously ruthless software salesman and now buys global control of 'health' (death) policy without the population he affects having any say. It's a breathtaking outrage. Gates talked about people being deeply affected by fear of 'Covid' when that was because of *him* and his global network lying to them minute-by-minute supported by a lying media that he seriously influences and funds to the tune of hundreds of millions. He's handed big sums to media operations including the BBC, NBC, Al Jazeera, Univision, *PBS NewsHour*,

ProPublica, National Journal, The Guardian, The Financial Times, The Atlantic, Texas Tribune, USA Today publisher Gannett, Washington Monthly, Le Monde, Center for Investigative Reporting, Pulitzer Center on Crisis Reporting, National Press Foundation, International Center for Journalists, Solutions Journalism Network, the Poynter Institute for Media Studies, and many more. Gates is everywhere in the ‘Covid’ hoax and the man must go to prison – or a mental facility – for the rest of his life and his money distributed to those he has taken such enormous psychopathic pleasure in crushing.

The Muscle

The Hunger Games global structure demands a police-military state – a fusion of the two into one force – which viciously imposes the will of the Cult on the population and protects the Cult from public rebellion. In that regard, too, the ‘Covid’ hoax just keeps on giving. Often unlawful, ridiculous and contradictory ‘Covid’ rules and regulations have been policed across the world by moronic automatons and psychopaths made faceless by face-nappy masks and acting like the Nazi SS and fascist blackshirts and brownshirts of Hitler and Mussolini. The smallest departure from the rules decreed by the psychos in government and their clueless gofers were jumped upon by the face-nappy fascists. Brutality against public protestors soon became commonplace even on girls, women and old people as the brave men with the batons – the Face-Nappies as I call them – broke up peaceful protests and handed out fines like confetti to people who couldn’t earn a living let alone pay hundreds of pounds for what was once an accepted human right. Robot Face-Nappies of Nottingham police in the English East Midlands fined one group £11,000 for attending a child’s birthday party. For decades I charted the transformation of law enforcement as genuine, decent officers were replaced with psychopaths and the brain dead who would happily and brutally do whatever their masters told them. Now they were let loose on the public and I would emphasise the point that none of this just happened. The step-by-step change in the dynamic between police and public was orchestrated from the shadows by

those who knew where this was all going and the same with the perceptual reframing of those in all levels of authority and official administration through ‘training courses’ by organisations such as Common Purpose which was created in the late 1980s and given a massive boost in Blair era Britain until it became a global phenomenon. Supposed public ‘servants’ began to view the population as the enemy and the same was true of the police. This was the start of the explosion of behaviour manipulation organisations and networks preparing for the all-war on the human psyche unleashed with the dawn of 2020. I will go into more detail about this later in the book because it is a core part of what is happening.

Police desecrated beauty spots to deter people gathering and arrested women for walking in the countryside alone ‘too far’ from their homes. We had arrogant, clueless sergeants in the Isle of Wight police where I live posting on Facebook what they insisted the population must do or else. A schoolmaster sergeant called Radford looked young enough for me to ask if his mother knew he was out, but he was posting what he *expected* people to do while a Sergeant Wilkinson boasted about fining lads for meeting in a McDonald’s car park where they went to get a lockdown takeaway. Wilkinson added that he had even cancelled their order. What a pair of prats these people are and yet they have increasingly become the norm among Jackboot Johnson’s Yellowshirts once known as the British police. This was the theme all over the world with police savagery common during lockdown protests in the United States, the Netherlands, and the fascist state of Victoria in Australia under its tyrannical and again moronic premier Daniel Andrews. Amazing how tyrannical and moronic tend to work as a team and the same combination could be seen across America as arrogant, narcissistic Woke governors and mayors such as Gavin Newsom (California), Andrew Cuomo (New York), Gretchen Whitmer (Michigan), Lori Lightfoot (Chicago) and Eric Garcetti (Los Angeles) did their Nazi and Stalin impressions with the full support of the compliant brutality of their enforcers in uniform as they arrested small business owners defying

fascist shutdown orders and took them to jail in ankle shackles and handcuffs. This happened to bistro owner Marlena Pavlos-Hackney in Gretchen Whitmer's fascist state of Michigan when police arrived to enforce an order by a state-owned judge for 'putting the community at risk' at a time when other states like Texas were dropping restrictions and migrants were pouring across the southern border without any 'Covid' questions at all. I'm sure there are many officers appalled by what they are ordered to do, but not nearly enough of them. If they were truly appalled they would not do it. As the months passed every opportunity was taken to have the military involved to make their presence on the streets ever more familiar and 'normal' for the longer-term goal of police-military fusion.

Another crucial element to the Hunger Games enforcement network has been encouraging the public to report neighbours and others for 'breaking the lockdown rules'. The group faced with £11,000 in fines at the child's birthday party would have been dobbed-in by a neighbour with a brain the size of a pea. The technique was most famously employed by the Stasi secret police in communist East Germany who had public informants placed throughout the population. A police chief in the UK says his force doesn't need to carry out 'Covid' patrols when they are flooded with so many calls from the public reporting other people for visiting the beach. Dorset police chief James Vaughan said people were so enthusiastic about snitching on their fellow humans they were now operating as an auxiliary arm of the police: 'We are still getting around 400 reports a week from the public, so we will respond to reports ... We won't need to be doing hotspot patrols because people are very quick to pick the phone up and tell us.' Vaughan didn't say that this is a pillar of all tyrannies of whatever complexion and the means to hugely extend the reach of enforcement while spreading distrust among the people and making them wary of doing anything that might get them reported. Those narcissistic Isle of Wight sergeants Radford and Wilkinson never fail to add a link to their Facebook posts where the public can inform on their fellow slaves.

Neither would be self-aware enough to realise they were imitating the Stasi which they might well never have heard of. Government psychologists that I will expose later laid out a policy to turn communities against each other in the same way.

A coincidence? Yep, and I can knit fog

I knew from the start of the alleged pandemic that this was a Cult operation. It presented limitless potential to rapidly advance the Cult agenda and exploit manipulated fear to demand that every man, woman and child on the planet was ‘vaccinated’ in a process never used on humans before which infuses self-replicating *synthetic* material into human cells. Remember the plan to transform the human body from a biological to a synthetic biological state. I’ll deal with the ‘vaccine’ (that’s not actually a vaccine) when I focus on the genetic agenda. Enough to say here that mass global ‘vaccination’ justified by this ‘new virus’ set alarms ringing after 30 years of tracking these people and their methods. The ‘Covid’ hoax officially beginning in China was also a big red flag for reasons I will be explaining. The agenda potential was so enormous that I could dismiss any idea that the ‘virus’ appeared naturally. Major happenings with major agenda implications never occur without Cult involvement in making them happen. My questions were twofold in early 2020 as the media began its campaign to induce global fear and hysteria: Was this alleged infectious agent released on purpose by the Cult or did it even exist at all? I then did what I always do in these situations. I sat, observed and waited to see where the evidence and information would take me. By March and early April synchronicity was strongly – and ever more so since then – pointing me in the direction of *there is no ‘virus’*. I went public on that with derision even from swathes of the alternative media that voiced a scenario that the Chinese government released the ‘virus’ in league with Deep State elements in the United States from a top-level bio-lab in Wuhan where the ‘virus’ is said to have first appeared. I looked at that possibility, but I didn’t buy it for several reasons. Deaths from the ‘virus’ did not in any way match what they

would have been with a ‘deadly bioweapon’ and it is much more effective if you sell the *illusion* of an infectious agent rather than having a real one unless you can control through injection who has it and who doesn’t. Otherwise you lose control of events. A made-up ‘virus’ gives you a blank sheet of paper on which you can make it do whatever you like and have any symptoms or mutant ‘variants’ you choose to add while a real infectious agent would limit you to what it actually does. A phantom disease allows you to have endless ludicrous ‘studies’ on the ‘Covid’ dollar to widen the perceived impact by inventing ever more ‘at risk’ groups including one study which said those who walk slowly may be almost four times more likely to die from the ‘virus’. People are in psychiatric wards for less.

A real ‘deadly bioweapon’ can take out people in the hierarchy that are not part of the Cult, but essential to its operation. Obviously they don’t want that. Releasing a real disease means you immediately lose control of it. Releasing an illusory one means you don’t. Again it’s vital that people are extra careful when dealing with what they want to hear. A bioweapon unleashed from a Chinese laboratory in collusion with the American Deep State may fit a conspiracy narrative, but is it true? Would it not be far more effective to use the excuse of a ‘virus’ to justify the real bioweapon – the ‘vaccine’? That way your disease agent does not have to be transmitted and arrives directly through a syringe. I saw a French virologist Luc Montagnier quoted in the alternative media as saying he had discovered that the alleged ‘new’ severe acute respiratory syndrome coronavirus , or SARS-CoV-2, was made artificially and included elements of the human immunodeficiency ‘virus’ (HIV) and a parasite that causes malaria. SARS-CoV-2 is alleged to trigger an alleged illness called Covid-19. I remembered Montagnier’s name from my research years before into claims that an HIV ‘retrovirus’ causes AIDS – claims that were demolished by Berkeley virologist Peter Duesberg who showed that no one had ever proved that HIV causes acquired immunodeficiency syndrome or AIDS. Claims that become accepted as fact, publicly and medically, with no proof whatsoever are an ever-recurring story that profoundly applies to

'Covid'. Nevertheless, despite the lack of proof, Montagnier's team at the Pasteur Institute in Paris had a long dispute with American researcher Robert Gallo over which of them discovered and isolated the HIV 'virus' and with *no evidence* found it to cause AIDS. You will see later that there is also no evidence that any 'virus' causes any disease or that there is even such a thing as a 'virus' in the way it is said to exist. The claim to have 'isolated' the HIV 'virus' will be presented in its real context as we come to the shocking story – and it is a story – of SARS-CoV-2 and so will Montagnier's assertion that he identified the full SARS-CoV-2 genome.

Hoax in the making

We can pick up the 'Covid' story in 2010 and the publication by the Rockefeller Foundation of a document called 'Scenarios for the Future of Technology and International Development'. The inner circle of the Rockefeller family has been serving the Cult since John D. Rockefeller (1839-1937) made his fortune with Standard Oil. It is less well known that the same Rockefeller – the Bill Gates of his day – was responsible for establishing what is now referred to as 'Big Pharma', the global network of pharmaceutical companies that make outrageous profits dispensing scalpel and drug 'medicine' and are obsessed with pumping vaccines in ever-increasing number into as many human arms and backsides as possible. John D. Rockefeller was the driving force behind the creation of the 'education' system in the United States and elsewhere specifically designed to program the perceptions of generations thereafter. The Rockefeller family donated exceptionally valuable land in New York for the United Nations building and were central in establishing the World Health Organization in 1948 as an agency of the UN which was created from the start as a Trojan horse and stalking horse for world government. Now enter Bill Gates. His family and the Rockefellers have long been extremely close and I have seen genealogy which claims that if you go back far enough the two families fuse into the same bloodline. Gates has said that the Bill and Melinda Gates Foundation was inspired by the Rockefeller Foundation and why not

when both are serving the same Cult? Major tax-exempt foundations are overwhelmingly criminal enterprises in which Cult assets fund the Cult agenda in the guise of 'philanthropy' while avoiding tax in the process. Cult operatives can become mega-rich in their role of front men and women for the psychopaths at the inner core and they, too, have to be psychopaths to knowingly serve such evil. Part of the deal is that a big percentage of the wealth gleaned from representing the Cult has to be spent advancing the ambitions of the Cult and hence you have the Rockefeller Foundation, Bill and Melinda Gates Foundation (and so many more) and people like George Soros with his global Open Society Foundations spending their billions in pursuit of global Cult control. Gates is a global public face of the Cult with his interventions in world affairs including Big Tech influence; a central role in the 'Covid' and 'vaccine' scam; promotion of the climate change shakedown; manipulation of education; geoengineering of the skies; and his food-control agenda as the biggest owner of farmland in America, his GMO promotion and through other means. As one writer said: 'Gates monopolizes or wields disproportionate influence over the tech industry, global health and vaccines, agriculture and food policy (including biopiracy and fake food), weather modification and other climate technologies, surveillance, education and media.' The almost limitless wealth secured through Microsoft and other not-allowed-to-fail ventures (including vaccines) has been ploughed into a long, long list of Cult projects designed to enslave the entire human race. Gates and the Rockefellers have been working as one unit with the Rockefeller-established World Health Organization leading global 'Covid' policy controlled by Gates through his mouth-piece Tedros. Gates became the WHO's biggest funder when Trump announced that the American government would cease its donations, but Biden immediately said he would restore the money when he took office in January, 2021. The Gates Foundation (the Cult) owns through limitless funding the world health system and the major players across the globe in the 'Covid' hoax.

Okay, with that background we return to that Rockefeller Foundation document of 2010 headed ‘Scenarios for the Future of Technology and International Development’ and its ‘imaginary’ epidemic of a virulent and deadly influenza strain which infected 20 percent of the global population and killed eight million in seven months. The Rockefeller scenario was that the epidemic destroyed economies, closed shops, offices and other businesses and led to governments imposing fierce rules and restrictions that included mandatory wearing of face masks and body-temperature checks to enter communal spaces like railway stations and supermarkets. The document predicted that even after the height of the Rockefeller-envisioned epidemic the authoritarian rule would continue to deal with further pandemics, transnational terrorism, environmental crises and rising poverty. Now you may think that the Rockefellers are our modern-day seers or alternatively, and rather more likely, that they well knew what was planned a few years further on. Fascism had to be imposed, you see, to ‘protect citizens from risk and exposure’. The Rockefeller scenario document said:

During the pandemic, national leaders around the world flexed their authority and imposed airtight rules and restrictions, from the mandatory wearing of face masks to body-temperature checks at the entries to communal spaces like train stations and supermarkets. Even after the pandemic faded, this more authoritarian control and oversight of citizens and their activities stuck and even intensified. In order to protect themselves from the spread of increasingly global problems – from pandemics and transnational terrorism to environmental crises and rising poverty – leaders around the world took a firmer grip on power.

At first, the notion of a more controlled world gained wide acceptance and approval. Citizens willingly gave up some of their sovereignty – and their privacy – to more paternalistic states in exchange for greater safety and stability. Citizens were more tolerant, and even eager, for top-down direction and oversight, and national leaders had more latitude to impose order in the ways they saw fit.

In developed countries, this heightened oversight took many forms: biometric IDs for all citizens, for example, and tighter regulation of key industries whose stability was deemed vital to national interests. In many developed countries, enforced cooperation with a suite of new regulations and agreements slowly but steadily restored both order and, importantly, economic growth.

There we have the prophetic Rockefellers in 2010 and three years later came their paper for the Global Health Summit in Beijing, China, when government representatives, the private sector, international organisations and groups met to discuss the next 100 years of 'global health'. The Rockefeller Foundation-funded paper was called 'Dreaming the Future of Health for the Next 100 Years' and more prophecy ensued as it described a dystopian future: 'The abundance of data, digitally tracking and linking people may mean the 'death of privacy' and may replace physical interaction with transient, virtual connection, generating isolation and raising questions of how values are shaped in virtual networks.' Next in the 'Covid' hoax preparation sequence came a 'table top' simulation in 2018 for another 'imaginary' pandemic of a disease called Clade X which was said to kill 900 million people. The exercise was organised by the Gates-funded Johns Hopkins University's Center for Health Security in the United States and this is the very same university that has been compiling the disgustingly and systematically erroneous global figures for 'Covid' cases and deaths. Similar Johns Hopkins health crisis scenarios have included the Dark Winter exercise in 2001 and Atlantic Storm in 2005.

Nostradamus 201

For sheer predictive genius look no further prophecy-watchers than the Bill Gates-funded Event 201 held only six weeks before the 'coronavirus pandemic' is supposed to have broken out in China and Event 201 was based on a scenario of a global 'coronavirus pandemic'. Melinda Gates, the great man's missus, told the BBC that he had 'prepared for years' for a coronavirus pandemic which told us what we already knew. Nostradamugates had predicted in a TED talk in 2015 that a pandemic was coming that would kill a lot of people and demolish the world economy. My god, the man is a machine – possibly even literally. Now here he was only weeks before the real thing funding just such a simulated scenario and involving his friends and associates at Johns Hopkins, the World Economic Forum Cult-front of Klaus Schwab, the United Nations,

Johnson & Johnson, major banks, and officials from China and the Centers for Disease Control in the United States. What synchronicity – Johns Hopkins would go on to compile the fraudulent ‘Covid’ figures, the World Economic Forum and Schwab would push the ‘Great Reset’ in response to ‘Covid’, the Centers for Disease Control would be at the forefront of ‘Covid’ policy in the United States, Johnson & Johnson would produce a ‘Covid vaccine’, and everything would officially start just weeks later in China. Spooky, eh? They were even accurate in creating a simulation of a ‘virus’ pandemic because the ‘real thing’ would also be a simulation. Event 201 was not an exercise preparing for something that might happen; it was a rehearsal for what those in control knew was *going* to happen and very shortly. Hours of this simulation were posted on the Internet and the various themes and responses mirrored what would soon be imposed to transform human society. News stories were inserted and what they said would be commonplace a few weeks later with still more prophecy perfection. Much discussion focused on the need to deal with misinformation and the ‘anti-vax movement’ which is exactly what happened when the ‘virus’ arrived – was said to have arrived – in the West.

Cult-owned social media banned criticism and exposure of the official ‘virus’ narrative and when I said there *was* no ‘virus’ in early April, 2020, I was banned by one platform after another including YouTube, Facebook and later Twitter. The mainstream broadcast media in Britain was in effect banned from interviewing me by the Tony-Blair-created government broadcasting censor Ofcom headed by career government bureaucrat Melanie Dawes who was appointed just as the ‘virus’ hoax was about to play out in January, 2020. At the same time the Ickonic media platform was using Vimeo, another ultra-Zionist-owned operation, while our own player was being created and they deleted in an instant hundreds of videos, documentaries, series and shows to confirm their unbelievable vindictiveness. We had copies, of course, and they had to be restored one by one when our player was ready. These people have no class. Sabbatian Facebook promised free advertisements for the Gates-

controlled World Health Organization narrative while deleting ‘false claims and conspiracy theories’ to stop ‘misinformation’ about the alleged coronavirus. All these responses could be seen just a short while earlier in the scenarios of Event 201. Extreme censorship was absolutely crucial for the Cult because the official story was so ridiculous and unsupportable by the evidence that it could never survive open debate and the free-flow of information and opinion. If you can’t win a debate then don’t have one is the Cult’s approach throughout history. Facebook’s little boy front man – front boy – Mark Zuckerberg equated ‘credible and accurate information’ with official sources and exposing their lies with ‘misinformation’.

Silencing those that can see

The censorship dynamic of Event 201 is now the norm with an army of narrative-supporting ‘fact-checker’ organisations whose entire reason for being is to tell the public that official narratives are true and those exposing them are lying. One of the most appalling of these ‘fact-checkers’ is called NewsGuard founded by ultra-Zionist Americans Gordon Crovitz and Steven Brill. Crovitz is a former publisher of *The Wall Street Journal*, former Executive Vice President of Dow Jones, a member of the Council on Foreign Relations (CFR), and on the board of the American Association of Rhodes Scholars. The CFR and Rhodes Scholarships, named after Rothschild agent Cecil Rhodes who plundered the gold and diamonds of South Africa for his masters and the Cult, have featured widely in my books. NewsGuard don’t seem to like me for some reason – I really can’t think why – and they have done all they can to have me censored and discredited which is, to quote an old British politician, like being savaged by a dead sheep. They are, however, like all in the censorship network, very well connected and funded by organisations themselves funded by, or connected to, Bill Gates. As you would expect with anything associated with Gates NewsGuard has an offshoot called HealthGuard which ‘fights online health care hoaxes’. How very kind. Somehow the NewsGuard European Managing Director Anna-Sophie Harling, a remarkably young-

looking woman with no broadcasting experience and little hands-on work in journalism, has somehow secured a position on the ‘Content Board’ of UK government broadcast censor Ofcom. An executive of an organisation seeking to discredit dissidents of the official narratives is making decisions for the government broadcast ‘regulator’ about content?? Another appalling ‘fact-checker’ is Full Fact funded by George Soros and global censors Google and Facebook.

It’s amazing how many activists in the ‘fact-checking’, ‘anti-hate’, arena turn up in government-related positions – people like UK Labour Party activist Imran Ahmed who heads the Center for Countering Digital Hate founded by people like Morgan McSweeney, now chief of staff to the Labour Party’s hapless and useless ‘leader’ Keir Starmer. Digital Hate – which is what it really is – uses the American spelling of Center to betray its connection to a transatlantic network of similar organisations which in 2020 shapeshifted from attacking people for ‘hate’ to attacking them for questioning the ‘Covid’ hoax and the dangers of the ‘Covid vaccine’. It’s just a coincidence, you understand. This is one of Imran Ahmed’s hysterical statements: ‘I would go beyond calling anti-vaxxers conspiracy theorists to say they are an extremist group that pose a national security risk.’ No one could ever accuse this prat of understatement and he’s including in that those parents who are now against vaccines after their children were damaged for life or killed by them. He’s such a nice man. Ahmed does the rounds of the Woke media getting soft-ball questions from spineless ‘journalists’ who never ask what right he has to campaign to destroy the freedom of speech of others while he demands it for himself. There also seems to be an overrepresentation in Ofcom of people connected to the narrative-worshipping BBC. This incredible global network of narrative-support was super-vital when the ‘Covid’ hoax was played in the light of the mega-whopper lies that have to be defended from the spotlight cast by the most basic intelligence.

Setting the scene

The Cult plays the long game and proceeds step-by-step ensuring that everything is in place before major cards are played and they don't come any bigger than the 'Covid' hoax. The psychopaths can't handle events where the outcome isn't certain and as little as possible – preferably nothing – is left to chance. Politicians, government and medical officials who would follow direction were brought to illusory power in advance by the Cult web whether on the national stage or others like state governors and mayors of America. For decades the dynamic between officialdom, law enforcement and the public was changed from one of service to one of control and dictatorship. Behaviour manipulation networks established within government were waiting to impose the coming 'Covid' rules and regulations specifically designed to subdue and rewire the psyche of the people in the guise of protecting health. These included in the UK the Behavioural Insights Team part-owned by the British government Cabinet Office; the Scientific Pandemic Insights Group on Behaviours (SPI-B); and a whole web of intelligence and military groups seeking to direct the conversation on social media and control the narrative. Among them are the cyberwarfare (on the people) 77th Brigade of the British military which is also coordinated through the Cabinet Office as civilian and military leadership continues to combine in what they call the Fusion Doctrine. The 77th Brigade is a British equivalent of the infamous Israeli (Sabbatian) military cyberwarfare and Internet manipulation operation Unit 8200 which I expose at length in *The Trigger*. Also carefully in place were the medical and science advisers to government – many on the payroll past or present of Bill Gates – and a whole alternative structure of unelected government stood by to take control when elected parliaments were effectively closed down once the 'Covid' card was slammed on the table. The structure I have described here and so much more was installed in every major country through the Cult networks. The top-down control hierarchy looks like this: The Cult – Cult-owned Gates – the World Health Organization and Tedros – Gates-funded or controlled chief medical officers and science 'advisers' (dictators) in each country –

political ‘leaders’ – law enforcement – The People. Through this simple global communication and enforcement structure the policy of the Cult could be imposed on virtually the entire human population so long as they acquiesced to the fascism. With everything in place it was time for the button to be pressed in late 2019/early 2020.

These were the prime goals the Cult had to secure for its will to prevail:

- 1) Locking down economies, closing all but designated ‘essential’ businesses (Cult-owned corporations were ‘essential’), and putting the population under house arrest was an imperative to destroy independent income and employment and ensure dependency on the Cult-controlled state in the Hunger Games Society. Lockdowns had to be established as the global blueprint from the start to respond to the ‘virus’ and followed by pretty much the entire world.
- 2) The global population had to be terrified into believing in a deadly ‘virus’ that didn’t actually exist so they would unquestioningly obey authority in the belief that authority must know how best to protect them and their families. Software salesman Gates would suddenly morph into the world’s health expert and be promoted as such by the Cult-owned media.
- 3) A method of testing that wasn’t testing for the ‘virus’, but was only claimed to be, had to be in place to provide the illusion of ‘cases’ and subsequent ‘deaths’ that had a very different cause to the ‘Covid-19’ that would be scribbled on the death certificate.
- 4) Because there was no ‘virus’ and the great majority testing positive with a test not testing for the ‘virus’ would have no symptoms of anything the lie had to be sold that people without symptoms (without the ‘virus’) could still pass it on to others. This was crucial to justify for the first time quarantining – house arresting – healthy people. Without this the economy-destroying lockdown of *everybody* could not have been credibly sold.
- 5) The ‘saviour’ had to be seen as a vaccine which beyond evil drug companies were working like angels of mercy to develop as quickly as possible, with all corners cut, to save the day. The public must absolutely not know that the ‘vaccine’ had nothing to do with a ‘virus’ or that the contents were ready and waiting with a very different motive long before the ‘Covid’ card was even lifted from the pack.

I said in March, 2020, that the ‘vaccine’ would have been created way ahead of the ‘Covid’ hoax which justified its use and the following December an article in the *New York Intelligencer* magazine said the Moderna ‘vaccine’ had been ‘designed’ by

January, 2020. This was ‘before China had even acknowledged that the disease could be transmitted from human to human, more than a week before the first confirmed coronavirus case in the United States’. The article said that by the time the first American death was announced a month later ‘the vaccine had already been manufactured and shipped to the National Institutes of Health for the beginning of its Phase I clinical trial’. The ‘vaccine’ was actually ‘designed’ long before that although even with this timescale you would expect the article to ask how on earth it could have been done that quickly. Instead it asked why the ‘vaccine’ had not been rolled out then and not months later. Journalism in the mainstream is truly dead. I am going to detail in the next chapter why the ‘virus’ has never existed and how a hoax on that scale was possible, but first the foundation on which the Big Lie of ‘Covid’ was built.

The test that doesn’t test

Fraudulent ‘testing’ is the bottom line of the whole ‘Covid’ hoax and was the means by which a ‘virus’ that did not exist *appeared* to exist. They could only achieve this magic trick by using a test not testing for the ‘virus’. To use a test that *was* testing for the ‘virus’ would mean that every test would come back negative given there was no ‘virus’. They chose to exploit something called the RT-PCR test invented by American biochemist Kary Mullis in the 1980s who said publicly that his PCR test … *cannot detect infectious disease*. Yes, the ‘test’ used worldwide to detect infectious ‘Covid’ to produce all the illusory ‘cases’ and ‘deaths’ compiled by Johns Hopkins and others *cannot detect infectious disease*. This fact came from the mouth of the man who invented PCR and was awarded the Nobel Prize in Chemistry in 1993 for doing so. Sadly, and incredibly conveniently for the Cult, Mullis died in August, 2019, at the age of 74 just before his test would be fraudulently used to unleash fascism on the world. He was said to have died from pneumonia which was an irony in itself. A few months later he would have had ‘Covid-19’ on his death certificate. I say the timing of his death was convenient because had he lived Mullis, a brilliant, honest and decent man, would have been

vociferously speaking out against the use of his test to detect 'Covid' when it was never designed, or able, to do that. I know that to be true given that Mullis made the same point when his test was used to 'detect' – not detect – HIV. He had been seriously critical of the Gallo/Montagnier claim to have isolated the HIV 'virus' and shown it to cause AIDS for which Mullis said there was no evidence. AIDS is actually not a disease but a series of diseases from which people die all the time. When they die from those *same diseases* after a positive 'test' for HIV then AIDS goes on their death certificate. I think I've heard that before somewhere. Countries instigated a policy with 'Covid' that anyone who tested positive with a test not testing for the 'virus' and died of any other cause within 28 days and even longer 'Covid-19' had to go on the death certificate. Cases have come from the test that can't test for infectious disease and the deaths are those who have died of *anything* after testing positive with a test not testing for the 'virus'. I'll have much more later about the death certificate scandal.

Mullis was deeply dismissive of the now US 'Covid' star Anthony Fauci who he said was a liar who didn't know anything about anything – 'and I would say that to his face – nothing.' He said of Fauci: 'The man thinks he can take a blood sample, put it in an electron microscope and if it's got a virus in there you'll know it – he doesn't understand electron microscopy and he doesn't understand medicine and shouldn't be in a position like he's in.' That position, terrifyingly, has made him the decider of 'Covid' fascism policy on behalf of the Cult in his role as director since 1984 of the National Institute of Allergy and Infectious Diseases (NIAID) while his record of being wrong is laughable; but being wrong, so long as it's the *right kind* of wrong, is why the Cult loves him. He'll say anything the Cult tells him to say. Fauci was made Chief Medical Adviser to the President immediately Biden took office. Biden was installed in the White House by Cult manipulation and one of his first decisions was to elevate Fauci to a position of even more control. This is a coincidence? Yes, and I identify as a flamenco dancer called Lola. How does such an incompetent criminal like Fauci remain in that

pivotal position in American health since *the 1980s*? When you serve the Cult it looks after you until you are surplus to requirements. Kary Mullis said prophetically of Fauci and his like: ‘Those guys have an agenda and it’s not an agenda we would like them to have ... they make their own rules, they change them when they want to, and Tony Fauci does not mind going on television in front of the people who pay his salary and lie directly into the camera.’ Fauci has done that almost daily since the ‘Covid’ hoax began. Lying is in Fauci’s DNA. To make the situation crystal clear about the PCR test this is a direct quote from its inventor Kary Mullis:

It [the PCR test] doesn’t tell you that you’re sick and doesn’t tell you that the thing you ended up with was really going to hurt you ...’

Ask yourself why governments and medical systems the world over have been using this very test to decide who is ‘infected’ with the SARS-CoV-2 ‘virus’ and the alleged disease it allegedly causes, ‘Covid-19’. The answer to that question will tell you what has been going on. By the way, here’s a little show-stopper – the ‘new’ SARS-CoV-2 ‘virus’ was ‘identified’ as such right from the start using ... *the PCR test not testing for the ‘virus’*. If you are new to this and find that shocking then stick around. I have hardly started yet. Even worse, other ‘tests’, like the ‘Lateral Flow Device’ (LFD), are considered so useless that they have to be *confirmed* by the PCR test! Leaked emails written by Ben Dyson, adviser to UK ‘Health’ Secretary Matt Hancock, said they were ‘dangerously unreliable’. Dyson, executive director of strategy at the Department of Health, wrote: ‘As of today, someone who gets a positive LFD result in (say) London has at best a 25 per cent chance of it being a true positive, but if it is a self-reported test potentially as low as 10 per cent (on an optimistic assumption about specificity) or as low as 2 per cent (on a more pessimistic assumption).’ These are the ‘tests’ that schoolchildren and the public are being urged to have twice a week or more and have to isolate if they get a positive. Each fake positive goes in the statistics as a ‘case’ no matter how ludicrously inaccurate and the

'cases' drive lockdown, masks and the pressure to 'vaccinate'. The government said in response to the email leak that the 'tests' were accurate which confirmed yet again what shocking bloody liars they are. The real false positive rate is *100 percent* as we'll see. In another 'you couldn't make it up' the UK government agreed to pay £2.8 billion to California's Innova Medical Group to supply the irrelevant lateral flow tests. The company's primary test-making centre is in China. Innova Medical Group, established in March, 2020, is owned by Pasaca Capital Inc, chaired by Chinese-American millionaire Charles Huang who was born in Wuhan.

How it works – and how it doesn't

The RT-PCR test, known by its full title of Polymerase chain reaction, is used across the world to make millions, even billions, of copies of a DNA/RNA genetic information sample. The process is called 'amplification' and means that a tiny sample of genetic material is amplified to bring out the detailed content. I stress that it is not testing for an infectious disease. It is simply amplifying a sample of genetic material. In the words of Kary Mullis: 'PCR is ... just a process that's used to make a whole lot of something out of something.' To emphasise the point companies that make the PCR tests circulated around the world to 'test' for 'Covid' warn on the box that it can't be used to detect 'Covid' or infectious disease and is for research purposes only. It's okay, rest for a minute and you'll be fine. This is the test that produces the 'cases' and 'deaths' that have been used to destroy human society. All those global and national medical and scientific 'experts' demanding this destruction to 'save us' KNOW that the test is not testing for the 'virus' and the cases and deaths they claim to be real are an almost unimaginable fraud. Every one of them and so many others including politicians and psychopaths like Gates and Tedros must be brought before Nuremberg-type trials and jailed for the rest of their lives. The more the genetic sample is amplified by PCR the more elements of that material become sensitive to the test and by that I don't mean sensitive for a 'virus' but for elements of the genetic material which

is naturally in the body or relates to remnants of old conditions of various kinds lying dormant and causing no disease. Once the amplification of the PCR reaches a certain level *everyone* will test positive. So much of the material has been made sensitive to the test that everyone will have some part of it in their body. Even lying criminals like Fauci have said that once PCR amplifications pass 35 cycles everything will be a false positive that cannot be trusted for the reasons I have described. I say, like many proper doctors and scientists, that 100 percent of the ‘positives’ are false, but let’s just go with Fauci for a moment.

He says that any amplification over 35 cycles will produce false positives and yet the US Centers for Disease Control (CDC) and Food and Drug Administration (FDA) have recommended up to 40 cycles and the National Health Service (NHS) in Britain admitted in an internal document for staff that it was using 45 cycles of amplification. A long list of other countries has been doing the same and at least one ‘testing’ laboratory has been using 50 cycles. Have you ever heard a doctor, medical ‘expert’ or the media ask what level of amplification has been used to claim a ‘positive’. The ‘test’ comes back ‘positive’ and so you have the ‘virus’, end of story. Now we can see how the government in Tanzania could send off samples from a goat and a pawpaw fruit under human names and both came back positive for ‘Covid-19’. Tanzania president John Magufuli mocked the ‘Covid’ hysteria, the PCR test and masks and refused to import the DNA-manipulating ‘vaccine’. The Cult hated him and an article sponsored by the Bill Gates Foundation appeared in the London *Guardian* in February, 2021, headed ‘It’s time for Africa to rein in Tanzania’s anti-vaxxer president’. Well, ‘reined in’ he shortly was. Magufuli appeared in good health, but then, in March, 2021, he was dead at 61 from ‘heart failure’. He was replaced by Samia Hassan Suhulu who is connected to Klaus Schwab’s World Economic Forum and she immediately reversed Magufuli’s ‘Covid’ policy. A sample of cola tested positive for ‘Covid’ with the PCR test in Germany while American actress and singer-songwriter Erykah Badu tested positive in one nostril and negative in the other. Footballer Ronaldo called

the PCR test ‘bullshit’ after testing positive three times and being forced to quarantine and miss matches when there was nothing wrong with him. The mantra from Tedros at the World Health Organization and national governments (same thing) has been test, test, test. They know that the more tests they can generate the more fake ‘cases’ they have which go on to become ‘deaths’ in ways I am coming to. The UK government has its Operation Moonshot planned to test multiple millions every day in workplaces and schools with free tests for everyone to use twice a week at home in line with the Cult plan from the start to make testing part of life. A government advertisement for an ‘Interim Head of Asymptomatic Testing Communication’ said the job included responsibility for delivering a ‘communications strategy’ (propaganda) ‘to support the expansion of asymptomatic testing that *“normalises testing as part of everyday life”*. More tests means more fake ‘cases’, ‘deaths’ and fascism. I have heard of, and from, many people who booked a test, couldn’t turn up, and yet got a positive result through the post for a test they’d never even had. The whole thing is crazy, but for the Cult there’s method in the madness. Controlling and manipulating the level of amplification of the test means the authorities can control whenever they want the number of apparent ‘cases’ and ‘deaths’. If they want to justify more fascist lockdown and destruction of livelihoods they keep the amplification high. If they want to give the illusion that lockdowns and the ‘vaccine’ are working then they lower the amplification and ‘cases’ and ‘deaths’ will appear to fall. In January, 2021, the Cult-owned World Health Organization suddenly warned laboratories about over-amplification of the test and to lower the threshold. Suddenly headlines began appearing such as: ‘Why ARE “Covid” cases plummeting?’ This was just when the vaccine rollout was underway and I had predicted months before they would make cases appear to fall through amplification tampering when the ‘vaccine’ came. These people are so predictable.

Cow vaccines?

The question must be asked of what is on the test swabs being poked far up the nose of the population to the base of the brain? A nasal swab punctured one woman's brain and caused it to leak fluid. Most of these procedures are being done by people with little training or medical knowledge. Dr Lorraine Day, former orthopaedic trauma surgeon and Chief of Orthopaedic Surgery at San Francisco General Hospital, says the tests are really a '*vaccine*'. Cows have long been vaccinated this way. She points out that masks have to cover the nose and the mouth where it is claimed the 'virus' exists in saliva. Why then don't they take saliva from the mouth as they do with a DNA test instead of pushing a long swab up the nose towards the brain? The ethmoid bone separates the nasal cavity from the brain and within that bone is the cribriform plate. Dr Day says that when the swab is pushed up against this plate and twisted the procedure is 'depositing things back there'. She claims that among these 'things' are nanoparticles that can enter the brain. Researchers have noted that a team at the Gates-funded Johns Hopkins have designed tiny, star-shaped micro-devices that can latch onto intestinal mucosa and release drugs into the body. Mucosa is the thin skin that covers the inside surface of parts of the body such as *the nose* and mouth and produces mucus to protect them. The Johns Hopkins micro-devices are called 'theragrippers' and were 'inspired' by a parasitic worm that digs its sharp teeth into a host's intestines. Nasal swabs are also coated in the sterilisation agent ethylene oxide. The US National Cancer Institute posts this explanation on its website:

At room temperature, ethylene oxide is a flammable colorless gas with a sweet odor. It is used primarily to produce other chemicals, including antifreeze. In smaller amounts, ethylene oxide is used as a pesticide and a sterilizing agent. The ability of ethylene oxide to damage DNA makes it an effective sterilizing agent but also accounts for its cancer-causing activity.

The Institute mentions lymphoma and leukaemia as cancers most frequently reported to be associated with occupational exposure to ethylene oxide along with stomach and breast cancers. How does anyone think this is going to work out with the constant testing

regime being inflicted on adults and children at home and at school that will accumulate in the body anything that's on the swab?

Doctors know best

It is vital for people to realise that 'hero' doctors 'know' only what the Big Pharma-dominated medical authorities tell them to 'know' and if they refuse to 'know' what they are told to 'know' they are out the door. They are mostly not physicians or healers, but repeaters of the official narrative – or else. I have seen alleged professional doctors on British television make shocking statements that we are supposed to take seriously. One called 'Dr' Amir Khan, who is actually telling patients how to respond to illness, said that men could take the birth pill to 'help slow down the effects of Covid-19'. In March, 2021, another ridiculous 'Covid study' by an American doctor proposed injecting men with the female sex hormone progesterone as a 'Covid' treatment. British doctor Nighat Arif told the BBC that face coverings were now going to be part of ongoing normal. Yes, the vaccine protects you, she said (evidence?) ... but the way to deal with viruses in the community was always going to come down to hand washing, face covering and keeping a physical distance. That's not what we were told before the 'vaccine' was circulating. Arif said she couldn't imagine ever again going on the underground or in a lift without a mask. I was just thanking my good luck that she was not my doctor when she said – in March, 2021 – that if 'we are *behaving* and we are doing all the right things' she thought we could 'have our nearest and dearest around us at home ... around *Christmas* and *New Year!*' Her patronising delivery was the usual school teacher talking to six-year-olds as she repeated every government talking point and probably believed them all. If we have learned anything from the 'Covid' experience surely it must be that humanity's perception of doctors needs a fundamental rethink. NHS 'doctor' Sara Kayat told her television audience that the 'Covid vaccine' would '100 percent prevent hospitalisation and death'. Not even Big Pharma claimed that. We have to stop taking 'experts' at their word without question when so many of them are

clueless and only repeating the party line on which their careers depend. That is not to say there are not brilliant doctors – there are and I have spoken to many of them since all this began – but you won't see them in the mainstream media or quoted by the psychopaths and yes-people in government.

Remember the name – Christian Drosten

German virologist Christian Drosten, Director of Charité Institute of Virology in Berlin, became a national star after the pandemic hoax began. He was feted on television and advised the German government on 'Covid' policy. Most importantly to the wider world Drosten led a group that produced the 'Covid' testing protocol for the PCR test. What a remarkable feat given the PCR cannot test for infectious disease and even more so when you think that Drosten said that his method of testing for SARS-CoV-2 was developed 'without having virus material available'. *He developed a test for a 'virus' that he didn't have and had never seen.* Let that sink in as you survey the global devastation that came from what he did. The whole catastrophe of Drosten's 'test' was based on the alleged genetic sequence published by Chinese scientists on the Internet. We will see in the next chapter that this alleged 'genetic sequence' has never been produced by China or anyone and cannot be when there is no SARS-CoV-2. Drosten, however, doesn't seem to let little details like that get in the way. He was the lead author with Victor Corman from the same Charité Hospital of the paper 'Detection of 2019 novel coronavirus (2019-nCoV) by real-time PCR' published in a magazine called *Eurosurveillance*. This became known as the Corman-Drosten paper. In November, 2020, with human society devastated by the effects of the Corman-Drosten test baloney, the protocol was publicly challenged by 22 international scientists and independent researchers from Europe, the United States, and Japan. Among them were senior molecular geneticists, biochemists, immunologists, and microbiologists. They produced a document headed 'External peer review of the RTPCR test to detect SARS-Cov-2 Reveals 10 Major Flaws At The Molecular and Methodological Level: Consequences

For False-Positive Results'. The flaws in the Corman-Drosten test included the following:

- The test is non-specific because of erroneous design
- Results are enormously variable
- The test is unable to discriminate between the whole 'virus' and viral fragments
- It doesn't have positive or negative controls
- The test lacks a standard operating procedure
- It is unsupported by proper peer view

The scientists said the PCR 'Covid' testing protocol was not founded on science and they demanded the Corman-Drosten paper be retracted by *Eurosurveillance*. They said all present and previous Covid deaths, cases, and 'infection rates' should be subject to a massive retroactive inquiry. Lockdowns and travel restrictions should be reviewed and relaxed and those diagnosed through PCR to have 'Covid-19' should not be forced to isolate. Dr Kevin Corbett, a health researcher and nurse educator with a long academic career producing a stream of peer-reviewed publications at many UK universities, made the same point about the PCR test debacle. He said of the scientists' conclusions: 'Every scientific rationale for the development of that test has been totally destroyed by this paper. It's like Hiroshima/Nagasaki to the Covid test.' He said that China hadn't given them an isolated 'virus' when Drosten developed the test. Instead they had developed the test from *a sequence in a gene bank.*' Put another way ... *they made it up!* The scientists were supported in this contention by a Portuguese appeals court which ruled in November, 2020, that PCR tests are unreliable and it is unlawful to quarantine people based solely on a PCR test. The point about China not providing an isolated virus must be true when the 'virus' has never been isolated to this day and the consequences of that will become clear. Drosten and company produced this useless 'protocol' right on cue in January, 2020, just as the 'virus' was said to

be moving westward and it somehow managed to successfully pass a peer-review in 24 hours. In other words there was no peer-review for a test that would be used to decide who had 'Covid' and who didn't across the world. The Cult-created, Gates-controlled World Health Organization immediately recommended all its nearly 200 member countries to use the Drosten PCR protocol to detect 'cases' and 'deaths'. The sting was underway and it continues to this day.

So who is this Christian Drosten that produced the means through which death, destruction and economic catastrophe would be justified? His education background, including his doctoral thesis, would appear to be somewhat shrouded in mystery and his track record is dire as with another essential player in the 'Covid' hoax, the Gates-funded Professor Neil Ferguson at the Gates-funded Imperial College in London of whom more shortly. Drosten predicted in 2003 that the alleged original SARS 'virus' (SARS-1') was an epidemic that could have serious effects on economies and an effective vaccine would take at least two years to produce. Drosten's answer to every alleged 'outbreak' is a vaccine which you won't be shocked to know. What followed were just 774 official deaths worldwide and none in Germany where there were only nine cases. That is even if you believe there ever was a SARS 'virus' when the evidence is zilch and I will expand on this in the next chapter. Drosten claims to be co-discoverer of 'SARS-1' and developed a test for it in 2003. He was screaming warnings about 'swine flu' in 2009 and how it was a widespread infection far more severe than any dangers from a vaccine could be and people should get vaccinated. It would be helpful for Drosten's vocal chords if he simply recorded the words 'the virus is deadly and you need to get vaccinated' and copies could be handed out whenever the latest made-up threat comes along. Drosten's swine flu epidemic never happened, but Big Pharma didn't mind with governments spending hundreds of millions on vaccines that hardly anyone bothered to use and many who did wished they hadn't. A study in 2010 revealed that the risk of dying from swine flu, or H1N1, was no higher than that of the annual seasonal flu which is what at least most of 'it' really was as in

the case of 'Covid-19'. A media investigation into Drosten asked how with such a record of inaccuracy he could be *the* government adviser on these issues. The answer to that question is the same with Drosten, Ferguson and Fauci – they keep on giving the authorities the 'conclusions' and 'advice' they want to hear. Drosten certainly produced the goods for them in January, 2020, with his PCR protocol garbage and provided the foundation of what German internal medicine specialist Dr Claus Köhnlein, co-author of *Virus Mania*, called the 'test pandemic'. The 22 scientists in the *Eurosurveillance* challenge called out conflicts of interest within the Drosten 'protocol' group and with good reason. Olfert Landt, a regular co-author of Drosten 'studies', owns the biotech company TIB Molbiol Syntheselabor GmbH in Berlin which manufactures and sells the tests that Drosten and his mates come up with. They have done this with SARS, Enterotoxigenic E. coli (ETEC), MERS, Zika 'virus', yellow fever, and now 'Covid'. Landt told the *Berliner Zeitung* newspaper:

The testing, design and development came from the Charité [Drosten and Corman]. We simply implemented it immediately in the form of a kit. And if we don't have the virus, which originally only existed in Wuhan, we can make a synthetic gene to simulate the genome of the virus. That's what we did very quickly.

This is more confirmation that the Drosten test was designed without access to the 'virus' and only a synthetic simulation which is what SARS-CoV-2 really is – a computer-generated synthetic fiction. It's quite an enterprise they have going here. A Drosten team decides what the test for something should be and Landt's biotech company flogs it to governments and medical systems across the world. His company must have made an absolute fortune since the 'Covid' hoax began. Dr Reiner Fuellmich, a prominent German consumer protection trial lawyer in Germany and California, is on Drosten's case and that of Tedros at the World Health Organization for crimes against humanity with a class-action lawsuit being prepared in the United States and other legal action in Germany.

Why China?

Scamming the world with a ‘virus’ that doesn’t exist would seem impossible on the face of it, but not if you have control of the relatively few people that make policy decisions and the great majority of the global media. Remember it’s not about changing ‘real’ reality it’s about controlling *perception* of reality. You don’t have to make something happen you only have to make people *believe* that it’s happening. Renegade Minds understand this and are therefore much harder to swindle. ‘Covid-19’ is not a ‘real’ ‘virus’. It’s a mind virus, like a computer virus, which has infected the minds, not the bodies, of billions. It all started, publically at least, in China and that alone is of central significance. The Cult was behind the revolution led by its asset Mao Zedong, or Chairman Mao, which established the People’s Republic of China on October 1st, 1949. It should have been called The Cult’s Republic of China, but the name had to reflect the recurring illusion that vicious dictatorships are run by and for the people (see all the ‘Democratic Republics’ controlled by tyrants). In the same way we have the ‘Biden’ Democratic Republic of America officially ruled by a puppet tyrant (at least temporarily) on behalf of Cult tyrants. The creation of Mao’s merciless communist/fascist dictatorship was part of a frenzy of activity by the Cult at the conclusion of World War Two which, like the First World War, it had instigated through its assets in Germany, Britain, France, the United States and elsewhere. Israel was formed in 1948; the Soviet Union expanded its ‘Iron Curtain’ control, influence and military power with the Warsaw Pact communist alliance in 1955; the United Nations was formed in 1945 as a Cult precursor to world government; and a long list of world bodies would be established including the World Health Organization (1948), World Trade Organization (1948 under another name until 1995), International Monetary Fund (1945) and World Bank (1944). Human society was redrawn and hugely centralised in the global Problem-Reaction-Solution that was World War Two. All these changes were significant. Israel would become the headquarters of the Sabbatians

and the revolution in China would prepare the ground and control system for the events of 2019/2020.

Renegade Minds know there are no borders except for public consumption. The Cult is a seamless, borderless global entity and to understand the game we need to put aside labels like borders, nations, countries, communism, fascism and democracy. These delude the population into believing that countries are ruled within their borders by a government of whatever shade when these are mere agencies of a global power. America's illusion of democracy and China's communism/fascism are subsidiaries – vehicles – for the same agenda. We may hear about conflict and competition between America and China and on the lower levels that will be true; but at the Cult level they are branches of the same company in the way of the McDonald's example I gave earlier. I have tracked in the books over the years support by US governments of both parties for Chinese Communist Party infiltration of American society through allowing the sale of land, even military facilities, and the acquisition of American business and university influence. All this is underpinned by the infamous stealing of intellectual property and technological know-how. Cult-owned Silicon Valley corporations waive their fraudulent 'morality' to do business with human-rights-free China; Cult-controlled Disney has become China's PR department; and China in effect owns 'American' sports such as basketball which depends for much of its income on Chinese audiences. As a result any sports player, coach or official speaking out against China's horrific human rights record is immediately condemned or fired by the China-worshipping National Basketball Association. One of the first acts of China-controlled Biden was to issue an executive order telling federal agencies to stop making references to the 'virus' by the 'geographic location of its origin'. Long-time Congressman Jerry Nadler warned that criticising China, America's biggest rival, leads to hate crimes against Asian people in the United States. So shut up you bigot. China is fast closing in on Israel as a country that must not be criticised which is apt, really, given that Sabbatians control them both. The two countries have

developed close economic, military, technological and strategic ties which include involvement in China's 'Silk Road' transport and economic initiative to connect China with Europe. Israel was the first country in the Middle East to recognise the establishment of Mao's tyranny in 1950 months after it was established.

Project Wuhan – the 'Covid' Psyop

I emphasise again that the Cult plays the long game and what is happening to the world today is the result of centuries of calculated manipulation following a script to take control step-by-step of every aspect of human society. I will discuss later the common force behind all this that has spanned those centuries and thousands of years if the truth be told. Instigating the Mao revolution in China in 1949 with a 2020 'pandemic' in mind is not only how they work – the 71 years between them is really quite short by the Cult's standards of manipulation preparation. The reason for the Cult's Chinese revolution was to create a fiercely-controlled environment within which an extreme structure for human control could be incubated to eventually be unleashed across the world. We have seen this happen since the 'pandemic' emerged from China with the Chinese control-structure founded on AI technology and tyrannical enforcement sweep across the West. Until the moment when the Cult went for broke in the West and put its fascism on public display Western governments had to pay some lip-service to freedom and democracy to not alert too many people to the tyranny-in-the-making. Freedoms were more subtly eroded and power centralised with covert government structures put in place waiting for the arrival of 2020 when that smokescreen of 'freedom' could be dispensed with. The West was not able to move towards tyranny before 2020 anything like as fast as China which was created as a tyranny and had no limits on how fast it could construct the Cult's blueprint for global control. When the time came to impose that structure on the world it was the same Cult-owned Chinese communist/fascist government that provided the excuse – the 'Covid pandemic'. It was absolutely crucial to the Cult plan for the Chinese response to the 'pandemic' –

draconian lockdowns of the entire population – to become the blueprint that Western countries would follow to destroy the livelihoods and freedom of their people. This is why the Cult-owned, Gates-owned, WHO Director-General Tedros said early on:

The Chinese government is to be congratulated for the extraordinary measures it has taken to contain the outbreak. China is actually setting a new standard for outbreak response and it is not an exaggeration.

Forbes magazine said of China: ‘... those measures protected untold millions from getting the disease’. The Rockefeller Foundation ‘epidemic scenario’ document in 2010 said ‘prophetically’:

However, a few countries did fare better – China in particular. The Chinese government’s quick imposition and enforcement of mandatory quarantine for all citizens, as well as its instant and near-hermetic sealing off of all borders, saved millions of lives, stopping the spread of the virus far earlier than in other countries and enabling a swifter post-pandemic recovery.

Once again – *spooky*.

The first official story was the ‘bat theory’ or rather the bat diversion. The source of the ‘virus outbreak’ we were told was a “wet market” in Wuhan where bats and other animals are bought and eaten in horrifically unhygienic conditions. Then another story emerged through the alternative media that the ‘virus’ had been released on purpose or by accident from a BSL-4 (biosafety level 4) laboratory in Wuhan not far from the wet market. The lab was reported to create and work with lethal concoctions and bioweapons. Biosafety level 4 is the highest in the World Health Organization system of safety and containment. Renegade Minds are aware of what I call designer manipulation. The ideal for the Cult is for people to buy its prime narrative which in the opening salvos of the ‘pandemic’ was the wet market story. It knows, however, that there is now a considerable worldwide alternative media of researchers sceptical of anything governments say and they are often given a version of events in a form they can perceive as credible while misdirecting them from the real truth. In this case let them

think that the conspiracy involved is a ‘bioweapon virus’ released from the Wuhan lab to keep them from the real conspiracy – *there is no ‘virus’*. The WHO’s current position on the source of the outbreak at the time of writing appears to be: ‘We haven’t got a clue, mate.’ This is a good position to maintain mystery and bewilderment. The inner circle will know where the ‘virus’ came from – *nowhere*. The bottom line was to ensure the public believed there *was* a ‘virus’ and it didn’t much matter if they thought it was natural or had been released from a lab. The belief that there was a ‘deadly virus’ was all that was needed to trigger global panic and fear. The population was terrified into handing their power to authority and doing what they were told. They had to or they were ‘all gonna die’.

In March, 2020, information began to come my way from real doctors and scientists and my own additional research which had my intuition screaming: ‘Yes, that’s it! *There is no virus.*’ The ‘bioweapon’ was not the ‘virus’; it was the ‘vaccine’ already being talked about that would be the bioweapon. My conclusion was further enhanced by happenings in Wuhan. The ‘virus’ was said to be sweeping the city and news footage circulated of people collapsing in the street (which they’ve never done in the West with the same ‘virus’). The Chinese government was building ‘new hospitals’ in a matter of ten days to ‘cope with demand’ such was the virulent nature of the ‘virus’. Yet in what seemed like no time the ‘new hospitals’ closed – even if they even opened – and China declared itself ‘virus-free’. It was back to business as usual. This was more propaganda to promote the Chinese draconian lockdowns in the West as the way to ‘beat the virus’. Trouble was that we subsequently had lockdown after lockdown, but never business as usual. As the people of the West and most of the rest of the world were caught in an ever-worsening spiral of lockdown, social distancing, masks, isolated old people, families forced apart, and livelihood destruction, it was party-time in Wuhan. Pictures emerged of thousands of people enjoying pool parties and concerts. It made no sense until you realised there never was a ‘virus’ and the

whole thing was a Cult set-up to transform human society out of one its major global strongholds – China.

How is it possible to deceive virtually the entire world population into believing there is a deadly virus when there is not even a ‘virus’ let alone a deadly one? It’s nothing like as difficult as you would think and that’s clearly true because it happened.

Postscript: See end of book Postscript for more on the ‘Wuhan lab virus release’ story which the authorities and media were pushing heavily in the summer of 2021 to divert attention from the truth that the ‘Covid virus’ is pure invention.

CHAPTER FIVE

There is no ‘virus’

You can fool some of the people all of the time, and all of the people some of the time, but you cannot fool all of the people all of the time

Abraham Lincoln

The greatest form of mind control is repetition. The more you repeat the same mantra of alleged ‘facts’ the more will accept them to be true. It becomes an ‘everyone knows that, mate’. If you can also censor any other version or alternative to your alleged ‘facts’ you are pretty much home and cooking.

By the start of 2020 the Cult owned the global mainstream media almost in its entirety to spew out its ‘Covid’ propaganda and ignore or discredit any other information and view. Cult-owned social media platforms in Cult-owned Silicon Valley were poised and ready to unleash a campaign of ferocious censorship to obliterate all but the official narrative. To complete the circle many demands for censorship by Silicon Valley were led by the mainstream media as ‘journalists’ became full-out enforcers for the Cult both as propagandists and censors. Part of this has been the influx of young people straight out of university who have become ‘journalists’ in significant positions. They have no experience and a headful of programmed perceptions from their years at school and university at a time when today’s young are the most perceptually-targeted generations in known human history given the insidious impact of technology. They enter the media perceptually prepared and ready to repeat the narratives of the system that programmed them to

repeat its narratives. The BBC has a truly pathetic ‘specialist disinformation reporter’ called Marianna Spring who fits this bill perfectly. She is clueless about the world, how it works and what is really going on. Her role is to discredit anyone doing the job that a proper journalist would do and system-serving hacks like Spring wouldn’t dare to do or even see the need to do. They are too busy licking the arse of authority which can never be wrong and, in the case of the BBC propaganda programme, *Panorama*, contacting payments systems such as PayPal to have a donations page taken down for a film company making documentaries questioning vaccines. Even the BBC soap opera *EastEnders* included a disgracefully biased scene in which an inarticulate white working class woman was made to look foolish for questioning the ‘vaccine’ while a well-spoken black man and Asian woman promoted the government narrative. It ticked every BBC box and the fact that the black and minority community was resisting the ‘vaccine’ had nothing to do with the way the scene was written. The BBC has become a disgusting tyrannical propaganda and censorship operation that should be defunded and disbanded and a free media take its place with a brief to stop censorship instead of demanding it. A BBC ‘interview’ with Gates goes something like: ‘Mr Gates, sir, if I can call you sir, would you like to tell our audience why you are such a great man, a wonderful humanitarian philanthropist, and why you should absolutely be allowed as a software salesman to decide health policy for approaching eight billion people? Thank you, sir, please sir.’ Propaganda programming has been incessant and merciless and when all you hear is the same story from the media, repeated by those around you who have only heard the same story, is it any wonder that people on a grand scale believe absolute mendacious garbage to be true? You are about to see, too, why this level of information control is necessary when the official ‘Covid’ narrative is so nonsensical and unsupportable by the evidence.

Structure of Deceit

The pyramid structure through which the ‘Covid’ hoax has been manifested is very simple and has to be to work. As few people as possible have to be involved with full knowledge of what they are doing – and why – or the real story would get out. At the top of the pyramid are the inner core of the Cult which controls Bill Gates who, in turn, controls the World Health Organization through his pivotal funding and his puppet Director-General mouthpiece, Tedros.

Before he was appointed Tedros was chair of the Gates-founded Global Fund to ‘fight against AIDS, tuberculosis and malaria’, a board member of the Gates-funded ‘vaccine alliance’ GAVI, and on the board of another Gates-funded organisation. Gates owns him and picked him for a specific reason – Tedros is a crook and worse. ‘Dr’ Tedros (he’s not a medical doctor, the first WHO chief not to be) was a member of the tyrannical Marxist government of Ethiopia for decades with all its human rights abuses. He has faced allegations of corruption and misappropriation of funds and was exposed three times for covering up cholera epidemics while Ethiopia’s health minister. Tedros appointed the mass-murdering genocidal Zimbabwe dictator Robert Mugabe as a WHO goodwill ambassador for public health which, as with Tedros, is like appointing a psychopath to run a peace and love campaign. The move was so ridiculous that he had to drop Mugabe in the face of widespread condemnation. American economist David Steinman, a Nobel peace prize nominee, lodged a complaint with the International Criminal Court in The Hague over alleged genocide by Tedros when he was Ethiopia’s foreign minister. Steinman says Tedros was a ‘crucial decision maker’ who directed the actions of Ethiopia’s security forces from 2013 to 2015 and one of three officials in charge when those security services embarked on the ‘killing’ and ‘torturing’ of Ethiopians. You can see where Tedros is coming from and it’s sobering to think that he has been the vehicle for Gates and the Cult to direct the global response to ‘Covid’. Think about that. A psychopathic Cult dictates to psychopath Gates who dictates to psychopath Tedros who dictates how countries of the world must respond to a ‘Covid virus’ never scientifically shown to exist. At the same time psychopathic Cult-owned Silicon Valley information

giants like Google, YouTube, Facebook and Twitter announced very early on that they would give the Cult/Gates/Tedros/WHO version of the narrative free advertising and censor those who challenged their intelligence-insulting, mendacious story.

The next layer in the global ‘medical’ structure below the Cult, Gates and Tedros are the chief medical officers and science ‘advisers’ in each of the WHO member countries which means virtually all of them. Medical officers and arbiters of science (they’re not) then take the WHO policy and recommended responses and impose them on their country’s population while the political ‘leaders’ say they are deciding policy (they’re clearly not) by ‘following the science’ on the advice of the ‘experts’ – the same medical officers and science ‘advisers’ (dictators). In this way with the rarest of exceptions the entire world followed the same policy of lockdown, people distancing, masks and ‘vaccines’ dictated by the psychopathic Cult, psychopathic Gates and psychopathic Tedros who we are supposed to believe give a damn about the health of the world population they are seeking to enslave. That, amazingly, is all there is to it in terms of crucial decision-making. Medical staff in each country then follow like sheep the dictates of the shepherds at the top of the national medical hierarchies – chief medical officers and science ‘advisers’ who themselves follow like sheep the shepherds of the World Health Organization and the Cult. Shepherds at the national level often have major funding and other connections to Gates and his Bill and Melinda Gates Foundation which carefully hands out money like confetti at a wedding to control the entire global medical system from the WHO down.

Follow the money

Christopher Whitty, Chief Medical Adviser to the UK Government at the centre of ‘virus’ policy, a senior adviser to the government’s Scientific Advisory Group for Emergencies (SAGE), and Executive Board member of the World Health Organization, was gifted a grant of \$40 million by the Bill and Melinda Gates Foundation for malaria research in Africa. The BBC described the unelected Whitty as ‘the

official who will probably have the greatest impact on our everyday lives of any individual policymaker in modern times' and so it turned out. What Gates and Tedros have said Whitty has done like his equivalents around the world. Patrick Vallance, co-chair of SAGE and the government's Chief Scientific Adviser, is a former executive of Big Pharma giant GlaxoSmithKline with its fundamental financial and business connections to Bill Gates. In September, 2020, it was revealed that Vallance owned a deferred bonus of shares in GlaxoSmithKline worth £600,000 while the company was 'developing' a 'Covid vaccine'. Move along now – nothing to see here – what could possibly be wrong with that? Imperial College in London, a major player in 'Covid' policy in Britain and elsewhere with its 'Covid-19' Response Team, is funded by Gates and has big connections to China while the now infamous Professor Neil Ferguson, the useless 'computer modeller' at Imperial College is also funded by Gates. Ferguson delivered the dramatically inaccurate excuse for the first lockdowns (much more in the next chapter). The Institute for Health Metrics and Evaluation (IHME) in the United States, another source of outrageously false 'Covid' computer models to justify lockdowns, is bankrolled by Gates who is a vehement promotor of lockdowns. America's version of Whitty and Vallance, the again now infamous Anthony Fauci, has connections to 'Covid vaccine' maker Moderna as does Bill Gates through funding from the Bill and Melinda Gates Foundation. Fauci is director of the National Institute of Allergy and Infectious Diseases (NIAID), a major recipient of Gates money, and they are very close. Deborah Birx who was appointed White House Coronavirus Response Coordinator in February, 2020, is yet another with ties to Gates. Everywhere you look at the different elements around the world behind the coordination and decision making of the 'Covid' hoax there is Bill Gates and his money. They include the World Health Organization; Centers for Disease Control (CDC) in the United States; National Institutes of Health (NIH) of Anthony Fauci; Imperial College and Neil Ferguson; the London School of Hygiene where Chris Whitty worked; Regulatory agencies like the UK Medicines & Healthcare products Regulatory Agency (MHRA)

which gave emergency approval for ‘Covid vaccines’; Wellcome Trust; GAVI, the Vaccine Alliance; the Coalition for Epidemic Preparedness Innovations (CEPI); Johns Hopkins University which has compiled the false ‘Covid’ figures; and the World Economic Forum. A [Nationalfile.com](#) article said:

Gates has a lot of pull in the medical world, he has a multi-million dollar relationship with Dr. Fauci, and Fauci originally took the Gates line supporting vaccines and casting doubt on [the drug hydroxychloroquine]. Coronavirus response team member Dr. Deborah Birx, appointed by former president Obama to serve as United States Global AIDS Coordinator, also sits on the board of a group that has received billions from Gates’ foundation, and Birx reportedly used a disputed Bill Gates-funded model for the White House’s Coronavirus effort. Gates is a big proponent for a population lockdown scenario for the Coronavirus outbreak.

Another funder of Moderna is the Defense Advanced Research Projects Agency (DARPA), the technology-development arm of the Pentagon and one of the most sinister organisations on earth. DARPA had a major role with the CIA covert technology-funding operation In-Q-Tel in the development of Google and social media which is now at the centre of global censorship. Fauci and Gates are extremely close and openly admit to talking regularly about ‘Covid’ policy, but then why wouldn’t Gates have a seat at every national ‘Covid’ table after his Foundation committed \$1.75 billion to the ‘fight against Covid-19’. When passed through our Orwellian Translation Unit this means that he has bought and paid for the Cult-driven ‘Covid’ response worldwide. Research the major ‘Covid’ response personnel in your own country and you will find the same Gates funding and other connections again and again. Medical and science chiefs following World Health Organization ‘policy’ sit atop a medical hierarchy in their country of administrators, doctors and nursing staff. These ‘subordinates’ are told they must work and behave in accordance with the policy delivered from the ‘top’ of the national ‘health’ pyramid which is largely the policy delivered by the WHO which is the policy delivered by Gates and the Cult. The whole ‘Covid’ narrative has been imposed on medical staff by a climate of fear although great numbers don’t even need that to comply. They do so through breathtaking levels of ignorance and

include doctors who go through life simply repeating what Big Pharma and their hierarchical masters tell them to say and believe. No wonder Big Pharma ‘medicine’ is one of the biggest killers on Planet Earth.

The same top-down system of intimidation operates with regard to the Cult Big Pharma cartel which also dictates policy through national and global medical systems in this way. The Cult and Big Pharma agendas are the same because the former controls and owns the latter. ‘Health’ administrators, doctors, and nursing staff are told to support and parrot the dictated policy or they will face consequences which can include being fired. How sad it’s been to see medical staff meekly repeating and imposing Cult policy without question and most of those who can see through the deceit are only willing to speak anonymously off the record. They know what will happen if their identity is known. This has left the courageous few to expose the lies about the ‘virus’, face masks, overwhelmed hospitals that aren’t, and the dangers of the ‘vaccine’ that isn’t a vaccine. When these medical professionals and scientists, some renowned in their field, have taken to the Internet to expose the truth their articles, comments and videos have been deleted by Cult-owned Facebook, Twitter and YouTube. What a real head-shaker to see YouTube videos with leading world scientists and highly qualified medical specialists with an added link underneath to the notorious Cult propaganda website *Wikipedia* to find the ‘facts’ about the same subject.

HIV – the ‘Covid’ trial-run

I’ll give you an example of the consequences for health and truth that come from censorship and unquestioning belief in official narratives. The story was told by PCR inventor Kary Mullis in his book *Dancing Naked in the Mind Field*. He said that in 1984 he accepted as just another scientific fact that Luc Montagnier of France’s Pasteur Institute and Robert Gallo of America’s National Institutes of Health had independently discovered that a ‘retrovirus’ dubbed HIV (human immunodeficiency virus) caused AIDS. They

were, after all, Mullis writes, specialists in retroviruses. This is how the medical and science pyramids work. Something is announced or *assumed* and then becomes an everybody-knows-that purely through repetition of the assumption as if it is fact. Complete crap becomes accepted truth with no supporting evidence and only repetition of the crap. This is how a 'virus' that doesn't exist became the 'virus' that changed the world. The HIV-AIDS fairy story became a multi-billion pound industry and the media poured out propaganda terrifying the world about the deadly HIV 'virus' that caused the lethal AIDS. By then Mullis was working at a lab in Santa Monica, California, to detect retroviruses with his PCR test in blood donations received by the Red Cross. In doing so he asked a virologist where he could find a reference for HIV being the cause of AIDS. 'You don't need a reference,' the virologist said ... '*Everybody knows it.*' Mullis said he wanted to quote a reference in the report he was doing and he said he felt a little funny about not knowing the source of such an important discovery when everyone else seemed to. The virologist suggested he cite a report by the Centers for Disease Control and Prevention (CDC) on morbidity and mortality. Mullis read the report, but it only said that an organism had been identified and did not say how. The report did not identify the original scientific work. Physicians, however, *assumed* (key recurring theme) that if the CDC was convinced that HIV caused AIDS then proof must exist. Mullis continues:

I did computer searches. Neither Montagnier, Gallo, nor anyone else had published papers describing experiments which led to the conclusion that HIV probably caused AIDS. I read the papers in Science for which they had become well known as AIDS doctors, but all they had said there was that they had found evidence of a past infection by something which was probably HIV in some AIDS patients.

They found antibodies. Antibodies to viruses had always been considered evidence of past disease, not present disease. Antibodies signaled that the virus had been defeated. The patient had saved himself. There was no indication in these papers that this virus caused a disease. They didn't show that everybody with the antibodies had the disease. In fact they found some healthy people with antibodies.

Mullis asked why their work had been published if Montagnier and Gallo hadn't really found this evidence, and why had they been fighting so hard to get credit for the discovery? He says he was hesitant to write 'HIV is the probable cause of AIDS' until he found published evidence to support that. 'Tens of thousands of scientists and researchers were spending billions of dollars a year doing research based on this idea,' Mullis writes. 'The reason had to be there somewhere; otherwise these people would not have allowed their research to settle into one narrow channel of investigation.' He said he lectured about PCR at numerous meetings where people were always talking about HIV and he asked them how they knew that HIV was the cause of AIDS:

Everyone said something. Everyone had the answer at home, in the office, in some drawer. They all knew, and they would send me the papers as soon as they got back. But I never got any papers. Nobody ever sent me the news about how AIDS was caused by HIV.

Eventually Mullis was able to ask Montagnier himself about the reference proof when he lectured in San Diego at the grand opening of the University of California AIDS Research Center. Mullis says this was the last time he would ask his question without showing anger. Montagnier said he should reference the CDC report. 'I read it', Mullis said, and it didn't answer the question. 'If Montagnier didn't know the answer who the hell did?' Then one night Mullis was driving when an interview came on National Public Radio with Peter Duesberg, a prominent virologist at Berkeley and a California Scientist of the Year. Mullis says he finally understood why he could not find references that connected HIV to AIDS – *there weren't any!* No one had ever proved that HIV causes AIDS even though it had spawned a multi-billion pound global industry and the media was repeating this as fact every day in their articles and broadcasts terrifying the shit out of people about AIDS and giving the impression that a positive test for HIV (see 'Covid') was a death sentence. Duesberg was a threat to the AIDS gravy train and the agenda that underpinned it. He was therefore abused and castigated after he told the Proceedings of the National Academy of Sciences

there was no good evidence implicating the new ‘virus’. Editors rejected his manuscripts and his research funds were deleted. Mullis points out that the CDC has defined AIDS as one of more than 30 diseases *if accompanied* by a positive result on a test that detects antibodies to HIV; but those same diseases are not defined as AIDS cases when antibodies are not detected:

If an HIV-positive woman develops uterine cancer, for example, she is considered to have AIDS. If she is not HIV positive, she simply has uterine cancer. An HIV-positive man with tuberculosis has AIDS; if he tests negative he simply has tuberculosis. If he lives in Kenya or Colombia, where the test for HIV antibodies is too expensive, he is simply presumed to have the antibodies and therefore AIDS, and therefore he can be treated in the World Health Organization’s clinic. It’s the only medical help available in some places. And it’s free, because the countries that support WHO are worried about AIDS.

Mullis accuses the CDC of continually adding new diseases (see ever more ‘Covid symptoms’) to the grand AIDS definition and of virtually doctoring the books to make it appear as if the disease continued to spread. He cites how in 1993 the CDC enormously broadened its AIDS definition and county health authorities were delighted because they received \$2,500 per year from the Federal government for every reported AIDS case. Ladies and gentlemen, I have just described, via Kary Mullis, the ‘Covid pandemic’ of 2020 and beyond. Every element is the same and it’s been pulled off in the same way by the same networks.

The ‘Covid virus’ exists? Okay – prove it. Er ... still waiting

What Kary Mullis described with regard to ‘HIV’ has been repeated with ‘Covid’. A claim is made that a new, or ‘novel’, infection has been found and the entire medical system of the world repeats that as fact exactly as they did with HIV and AIDS. No one in the mainstream asks rather relevant questions such as ‘How do you know?’ and ‘Where is your proof?’ The SARS-CoV-2 ‘virus’ and the ‘Covid-19 disease’ became an overnight ‘everybody-knows-that’. The origin could be debated and mulled over, but what you could not suggest was that ‘SARS-CoV-2’ didn’t exist. That would be

ridiculous. ‘Everybody knows’ the ‘virus’ exists. Well, I didn’t for one along with American proper doctors like Andrew Kaufman and Tom Cowan and long-time American proper journalist Jon Rappaport. We dared to pursue the obvious and simple question: ‘Where’s the evidence?’ The overwhelming majority in medicine, journalism and the general public did not think to ask that. After all, *everyone knew* there was a new ‘virus’. Everyone was saying so and I heard it on the BBC. Some would eventually argue that the ‘deadly virus’ was nothing like as deadly as claimed, but few would venture into the realms of its very existence. Had they done so they would have found that the evidence for that claim had gone AWOL as with HIV causes AIDS. In fact, not even that. For something to go AWOL it has to exist in the first place and scientific proof for a ‘SARS-Cov-2’ can be filed under nothing, nowhere and zilch.

Dr Andrew Kaufman is a board-certified forensic psychiatrist in New York State, a Doctor of Medicine and former Assistant Professor and Medical Director of Psychiatry at SUNY Upstate Medical University, and Medical Instructor of Hematology and Oncology at the Medical School of South Carolina. He also studied biology at the Massachusetts Institute of Technology (MIT) and trained in Psychiatry at Duke University. Kaufman is retired from allopathic medicine, but remains a consultant and educator on natural healing, I saw a video of his very early on in the ‘Covid’ hoax in which he questioned claims about the ‘virus’ in the absence of any supporting evidence and with plenty pointing the other way. I did everything I could to circulate his work which I felt was asking the pivotal questions that needed an answer. I can recommend an excellent pull-together interview he did with the website The Last Vagabond entitled *Dr Andrew Kaufman: Virus Isolation, Terrain Theory and Covid-19* and his website is andrewkaufmanmd.com. Kaufman is not only a forensic psychiatrist; he is forensic in all that he does. He always reads original scientific papers, experiments and studies instead of second-third-fourth-hand reports about the ‘virus’ in the media which are repeating the repeated repetition of the narrative. When he did so with the original Chinese ‘virus’ papers Kaufman

realised that there was no evidence of a ‘SARS-Cov-2’. They had never – from the start – shown it to exist and every repeat of this claim worldwide was based on the accepted existence of proof that was nowhere to be found – see Kary Mullis and HIV. Here we go again.

Let's postulate

Kaufman discovered that the Chinese authorities immediately concluded that the cause of an illness that broke out among about 200 initial patients in Wuhan was a ‘new virus’ when there were no grounds to make that conclusion. The alleged ‘virus’ was not isolated from other genetic material in their samples and then shown through a system known as Koch’s postulates to be the causative agent of the illness. The world was told that the SARS-Cov-2 ‘virus’ caused a disease they called ‘Covid-19’ which had ‘flu-like’ symptoms and could lead to respiratory problems and pneumonia. If it wasn’t so tragic it would almost be funny. *‘Flu-like’ symptoms?* *Pneumonia? Respiratory disease?* What in CHINA and particularly in Wuhan, one of the most polluted cities in the world with a resulting epidemic of respiratory disease?? Three hundred thousand people get pneumonia in China every year and there are nearly a billion cases worldwide of ‘flu-like symptoms’. These have a whole range of causes – including pollution in Wuhan – but no other possibility was credibly considered in late 2019 when the world was told there was a new and deadly ‘virus’. The global prevalence of pneumonia and ‘flu-like systems’ gave the Cult networks unlimited potential to re-diagnose these other causes as the mythical ‘Covid-19’ and that is what they did from the very start. Kaufman revealed how Chinese medical and science authorities (all subordinates to the Cult-owned communist government) took genetic material from the lungs of only a few of the first patients. The material contained their own cells, bacteria, fungi and other microorganisms living in their bodies. The only way you could prove the existence of the ‘virus’ and its responsibility for the alleged ‘Covid-19’ was to isolate the virus from all the other material – a process also known as ‘purification’ – and

then follow the postulates sequence developed in the late 19th century by German physician and bacteriologist Robert Koch which became the ‘gold standard’ for connecting an alleged causation agent to a disease:

1. The microorganism (bacteria, fungus, virus, etc.) must be present in every case of the disease and all patients must have the same symptoms. It must also *not be present in healthy individuals*.
2. The microorganism must be isolated from the host with the disease. If the microorganism is a bacteria or fungus it must be grown in a pure culture. If it is a virus, it must be purified (i.e. containing no other material except the virus particles) from a clinical sample.
3. The specific disease, with all of its characteristics, must be reproduced when the infectious agent (the purified virus or a pure culture of bacteria or fungi) is inoculated into a healthy, susceptible host.
4. The microorganism must be recoverable from the experimentally infected host as in step 2.

Not one of these criteria has been met in the case of ‘SARS-Cov-2’ and ‘Covid-19’. Not ONE. EVER. Robert Koch refers to bacteria and not viruses. What are called ‘viral particles’ are so minute (hence masks are useless by any definition) that they could only be seen after the invention of the electron microscope in the 1930s and can still only be observed through that means. American bacteriologist and virologist Thomas Milton Rivers, the so-called ‘Father of Modern Virology’ who was very significantly director of the Rockefeller Institute for Medical Research in the 1930s, developed a less stringent version of Koch’s postulates to identify ‘virus’ causation known as ‘Rivers criteria’. ‘Covid’ did not pass that process either. Some even doubt whether any ‘virus’ can be isolated from other particles containing genetic material in the Koch method. Freedom of Information requests in many countries asking for scientific proof that the ‘Covid virus’ has been purified and isolated and shown to exist have all come back with a ‘we don’t have that’ and when this happened with a request to the UK Department of Health they added this comment:

However, outside of the scope of the [Freedom of Information Act] and on a discretionary basis, the following information has been advised to us, which may be of interest. Most infectious diseases are caused by viruses, bacteria or fungi. Some bacteria or fungi have the capacity to grow on their own in isolation, for example in colonies on a petri dish. Viruses are different in that they are what we call ‘obligate pathogens’ – that is, they cannot survive or reproduce without infecting a host ...

... For some diseases, it is possible to establish causation between a microorganism and a disease by isolating the pathogen from a patient, growing it in pure culture and reintroducing it to a healthy organism. These are known as ‘Koch’s postulates’ and were developed in 1882. However, as our understanding of disease and different disease-causing agents has advanced, these are no longer the method for determining causation [Andrew Kaufman asks why in that case are there two published articles falsely claiming to satisfy Koch’s postulates].

It has long been known that viral diseases cannot be identified in this way as viruses cannot be grown in ‘pure culture’. When a patient is tested for a viral illness, this is normally done by looking for the presence of antigens, or viral genetic code in a host with molecular biology techniques [Kaufman asks how you could know the origin of these chemicals without having a pure culture for comparison].

For the record ‘antigens’ are defined so:

Invading microorganisms have antigens on their surface that the human body can recognise as being foreign – meaning not belonging to it. When the body recognises a foreign antigen, lymphocytes (white blood cells) produce antibodies, which are complementary in shape to the antigen.

Notwithstanding that this is open to question in relation to ‘SARS-CoV-2’ the presence of ‘antibodies’ can have many causes and they are found in people that are perfectly well. Kary Mullis said: ‘Antibodies ... had always been considered evidence of past disease, not present disease.’

‘Covid’ really is a computer ‘virus’

Where the UK Department of Health statement says ‘viruses’ are now ‘diagnosed’ through a ‘viral genetic code in a host with molecular biology techniques’, they mean ... *the PCR test* which its inventor said cannot test for infectious disease. They have no credible method of connecting a ‘virus’ to a disease and we will see that there is no scientific proof that any ‘virus’ causes any disease or there is any such thing as a ‘virus’ in the way that it is described. Tenacious Canadian researcher Christine Massey and her team made

some 40 Freedom of Information requests to national public health agencies in different countries asking for proof that SARS-CoV-2 has been isolated and not one of them could supply that information. Massey said of her request in Canada: 'Freedom of Information reveals Public Health Agency of Canada has no record of 'SARS-CoV-2' isolation performed by anyone, anywhere, ever.' If you accept the comment from the UK Department of Health it's because they can't isolate a 'virus'. Even so many 'science' papers claimed to have isolated the 'Covid virus' until they were questioned and had to admit they hadn't. A reply from the Robert Koch Institute in Germany was typical: 'I am not aware of a paper which purified isolated SARS-CoV-2.' So what the hell was Christian Drosten and his gang using to design the 'Covid' testing protocol that has produced all the illusory Covid' cases and 'Covid' deaths when the head of the Chinese version of the CDC admitted there was a problem right from the start in that the 'virus' had never been isolated/purified? Breathe deeply: What they are calling 'Covid' is actually created by a *computer program* i.e. *they made it up* – er, that's it. They took lung fluid, with many sources of genetic material, from one single person alleged to be infected with Covid-19 by a PCR test which they *claimed*, without clear evidence, contained a 'virus'. They used several computer programs to create a model of a theoretical virus genome sequence from more than fifty-six million small sequences of RNA, each of an unknown source, assembling them like a puzzle with no known solution. The computer filled in the gaps with sequences from bits in the gene bank to make it look like a bat SARS-like coronavirus! A wave of the magic wand and poof, an *in silico* (computer-generated) genome, a scientific fantasy, was created. UK health researcher Dr Kevin Corbett made the same point with this analogy:

... It's like giving you a few bones and saying that's your fish. It could be any fish. Not even a skeleton. Here's a few fragments of bones. That's your fish ... It's all from gene bank and the bits of the virus sequence that weren't there they made up.

They synthetically created them to fill in the blanks. That's what genetics is; it's a code. So it's ABBBCCDDDD and you're missing some what you think is EEE so you put it in. It's all

synthetic. You just manufacture the bits that are missing. This is the end result of the geneticization of virology. This is basically a computer virus.

Further confirmation came in an email exchange between British citizen journalist Frances Leader and the government's Medicines & Healthcare Products Regulatory Agency (the Gates-funded MHRA) which gave emergency permission for untested 'Covid vaccines' to be used. The agency admitted that the 'vaccine' is not based on an isolated 'virus', but comes from a *computer-generated model*. Frances Leader was naturally banned from Cult-owned fascist Twitter for making this exchange public. The process of creating computer-generated alleged 'viruses' is called 'in silico' or 'in silicon' – computer chips – and the term 'in silico' is believed to originate with biological experiments using only a computer in 1989. 'Vaccines' involved with 'Covid' are also produced 'in silico' or by computer not a natural process. If the original 'virus' is nothing more than a made-up computer model how can there be 'new variants' of something that never existed in the first place? They are not new 'variants'; they are new *computer models* only minutely different to the original program and designed to further terrify the population into having the 'vaccine' and submitting to fascism. You want a 'new variant'? Click, click, enter – there you go. Tell the medical profession that you have discovered a 'South African variant', 'UK variants' or a 'Brazilian variant' and in the usual HIV-causes-AIDS manner they will unquestioningly repeat it with no evidence whatsoever to support these claims. They will go on television and warn about the dangers of 'new variants' while doing nothing more than repeating what they have been told to be true and knowing that any deviation from that would be career suicide. Big-time insiders will know it's a hoax, but much of the medical community is clueless about the way they are being played and themselves play the public without even being aware they are doing so. What an interesting 'coincidence' that AstraZeneca and Oxford University were conducting 'Covid vaccine trials' in the three countries – the UK, South Africa and Brazil – where the first three 'variants' were claimed to have 'broken out'.

Here's your 'virus' – it's a unicorn

Dr Andrew Kaufman presented a brilliant analysis describing how the 'virus' was imagined into fake existence when he dissected an article published by *Nature* and written by 19 authors detailing *alleged* 'sequencing of a complete viral genome' of the 'new SARS-CoV-2 virus'. This computer-modelled *in silico* genome was used as a template for all subsequent genome sequencing experiments that resulted in the so-called variants which he said now number more than 6,000. The fake genome was constructed from more than 56 million individual short strands of RNA. Those little pieces were assembled into longer pieces by finding areas of overlapping sequences. The computer programs created over two million possible combinations from which the authors simply chose the longest one. They then compared this to a 'bat virus' and the computer 'alignment' rearranged the sequence and filled in the gaps! They called this computer-generated abomination the 'complete genome'. Dr Tom Cowan, a fellow medical author and collaborator with Kaufman, said such computer-generation constitutes scientific fraud and he makes this superb analogy:

Here is an equivalency: A group of researchers claim to have found a unicorn because they found a piece of a hoof, a hair from a tail, and a snippet of a horn. They then add that information into a computer and program it to re-create the unicorn, and they then claim this computer re-creation is the real unicorn. Of course, they had never actually seen a unicorn so could not possibly have examined its genetic makeup to compare their samples with the actual unicorn's hair, hooves and horn.

The researchers claim they decided which is the real genome of SARS-CoV-2 by 'consensus', sort of like a vote. Again, different computer programs will come up with different versions of the imaginary 'unicorn', so they come together as a group and decide which is the real imaginary unicorn.

This is how the 'virus' that has transformed the world was brought into fraudulent 'existence'. Extraordinary, yes, but as the Nazis said the bigger the lie the more will believe it. Cowan, however, wasn't finished and he went on to identify what he called the real blockbuster in the paper. He quotes this section from a paper written

by virologists and published by the CDC and then explains what it means:

Therefore, we examined the capacity of SARS-CoV-2 to infect and replicate in several common primate and human cell lines, including human adenocarcinoma cells (A549), human liver cells (HUH 7.0), and human embryonic kidney cells (HEK-293T). In addition to Vero E6 and Vero CCL81 cells. ... Each cell line was inoculated at high multiplicity of infection and examined 24h post-infection.

No CPE was observed in any of the cell lines except in Vero cells, which grew to greater than 10 to the 7th power at 24 h post-infection. In contrast, HUH 7.0 and 293T showed only modest viral replication, and A549 cells were incompatible with SARS CoV-2 infection.

Cowan explains that when virologists attempt to prove infection they have three possible 'hosts' or models on which they can test. The first was humans. Exposure to humans was generally not done for ethical reasons and has never been done with SARS-CoV-2 or any coronavirus. The second possible host was animals. Cowan said that forgetting for a moment that they never actually use purified virus when exposing animals they do use solutions that they *claim* contain the virus. Exposure to animals has been done with SARS-CoV-2 in an experiment involving mice and this is what they found: *None of the wild (normal) mice got sick*. In a group of genetically-modified mice, a statistically insignificant number lost weight and had slightly bristled fur, but they experienced nothing like the illness called 'Covid-19'. Cowan said the third method – the one they mostly rely on – is to inoculate solutions they *say* contain the virus onto a variety of tissue cultures. This process had never been shown to kill tissue *unless* the sample material was starved of nutrients and poisoned as *part of the process*. Yes, incredibly, in tissue experiments designed to show the 'virus' is responsible for killing the tissue they starve the tissue of nutrients and add toxic drugs including antibiotics and they do not have control studies to see if it's the starvation and poisoning that is degrading the tissue rather than the 'virus' they allege to be in there somewhere. You want me to pinch you? Yep, I understand. Tom Cowan said this about the whole nonsensical farce as he explains what that quote from the CDC paper really means:

The shocking thing about the above quote is that using their own methods, the virologists found that solutions containing SARS-CoV-2 – even in high amounts – were NOT, I repeat NOT, infective to any of the three human tissue cultures they tested. In plain English, this means they proved, on their terms, that this ‘new coronavirus’ is not infectious to human beings. It is ONLY infective to monkey kidney cells, and only then when you add two potent drugs (gentamicin and amphotericin), known to be toxic to kidneys, to the mix.

My friends, read this again and again. These virologists, published by the CDC, performed a clear proof, on their terms, showing that the SARS-CoV-2 virus is harmless to human beings. That is the only possible conclusion, but, unfortunately, this result is not even mentioned in their conclusion. They simply say they can provide virus stocks cultured only on monkey Vero cells, thanks for coming.

Cowan concluded: ‘If people really understood how this “science” was done, I would hope they would storm the gates and demand honesty, transparency and truth.’ Dr Michael Yeadon, former Vice President and Chief Scientific Adviser at drug giant Pfizer has been a vocal critic of the ‘Covid vaccine’ and its potential for multiple harm. He said in an interview in April, 2021, that ‘not one [vaccine] has the virus. He was asked why vaccines normally using a ‘dead’ version of a disease to activate the immune system were not used for ‘Covid’ and instead we had the synthetic methods of the ‘mRNA Covid vaccine’. Yeadon said that to do the former ‘you’d have to have some of [the virus] wouldn’t you?’ He added: ‘No-one’s got any – seriously.’ Yeadon said that surely they couldn’t have fooled the whole world for a year without having a virus, ‘but oddly enough ask around – no one’s got it’. He didn’t know why with all the ‘great labs’ around the world that the virus had not been isolated – ‘Maybe they’ve been too busy running bad PCR tests and vaccines that people don’t need.’ What is today called ‘science’ is not ‘science’ at all. Science is no longer what is, but whatever people can be manipulated to *believe* that it is. Real science has been hijacked by the Cult to dispense and produce the ‘expert scientists’ and contentions that suit the agenda of the Cult. How big-time this has happened with the ‘Covid’ hoax which is entirely based on fake science delivered by fake ‘scientists’ and fake ‘doctors’. The human-caused climate change hoax is also entirely based on fake science delivered by fake ‘scientists’ and fake ‘climate experts’. In both cases real

scientists, climate experts and doctors have their views suppressed and deleted by the Cult-owned science establishment, media and Silicon Valley. This is the ‘science’ that politicians claim to be ‘following’ and a common denominator of ‘Covid’ and climate are Cult psychopaths Bill Gates and his mate Klaus Schwab at the Gates-funded World Economic Forum. But, don’t worry, it’s all just a coincidence and absolutely nothing to worry about. Zzzzzzzz.

What is a ‘virus’ REALLY?

Dr Tom Cowan is one of many contesting the very existence of viruses let alone that they cause disease. This is understandable when there is no scientific evidence for a disease-causing ‘virus’. German virologist Dr Stefan Lanka won a landmark case in 2017 in the German Supreme Court over his contention that there is no such thing as a measles virus. He had offered a big prize for anyone who could prove there is and Lanka won his case when someone sought to claim the money. There is currently a prize of more than 225,000 euros on offer from an Isolate Truth Fund for anyone who can prove the isolation of SARS-CoV-2 and its genetic substance. Lanka wrote in an article headed ‘The Misconception Called Virus’ that scientists think a ‘virus’ is causing tissue to become diseased and degraded when in fact it is the *processes they are using* which do that – not a ‘virus’. Lanka has done an important job in making this point clear as Cowan did in his analysis of the CDC paper. Lanka says that all claims about viruses as disease-causing pathogens are wrong and based on ‘easily recognisable, understandable and verifiable misinterpretations.’ Scientists believed they were working with ‘viruses’ in their laboratories when they were really working with ‘typical particles of specific dying tissues or cells ...’ Lanka said that the tissue decaying process claimed to be caused by a ‘virus’ still happens when no alleged ‘virus’ is involved. It’s the *process* that does the damage and not a ‘virus’. The genetic sample is deprived of nutrients, removed from its energy supply through removal from the body and then doused in toxic antibiotics to remove any bacteria. He confirms again that establishment scientists do not (pinch me)

conduct control experiments to see if this is the case and if they did they would see the claims that 'viruses' are doing the damage is nonsense. He adds that during the measles 'virus' court case he commissioned an independent laboratory to perform just such a control experiment and the result was that the tissues and cells died in the exact same way as with alleged 'infected' material. This is supported by a gathering number of scientists, doctors and researchers who reject what is called 'germ theory' or the belief in the body being infected by contagious sources emitted by other people. Researchers Dawn Lester and David Parker take the same stance in their highly-detailed and sourced book *What Really Makes You Ill – Why everything you thought you knew about disease is wrong* which was recommended to me by a number of medical professionals genuinely seeking the truth. Lester and Parker say there is no provable scientific evidence to show that a 'virus' can be transmitted between people or people and animals or animals and people:

The definition also claims that viruses are the cause of many diseases, as if this has been definitively proven. But this is not the case; there is no original scientific evidence that definitively demonstrates that any virus is the cause of any disease. The burden of proof for any theory lies with those who proposed it; but none of the existing documents provides 'proof' that supports the claim that 'viruses' are pathogens.

Dr Tom Cowan employs one of his clever analogies to describe the process by which a 'virus' is named as the culprit for a disease when what is called a 'virus' is only material released by cells detoxing themselves from infiltration by chemical or radiation poisoning. The tidal wave of technologically-generated radiation in the 'smart' modern world plus all the toxic food and drink are causing this to happen more than ever. Deluded 'scientists' misread this as a gathering impact of what they wrongly label 'viruses'.

Paper can infect houses

Cowan said in an article for davidicke.com – with his tongue only mildly in his cheek – that he believed he had made a tremendous

discovery that may revolutionise science. He had discovered that small bits of paper are alive, ‘well alive-ish’, can ‘infect’ houses, and then reproduce themselves inside the house. The result was that this explosion of growth in the paper inside the house causes the house to explode, blowing it to smithereens. His evidence for this new theory is that in the past months he had carefully examined many of the houses in his neighbourhood and found almost no scraps of paper on the lawns and surrounds of the house. There was an occasional stray label, but nothing more. Then he would return to these same houses a week or so later and with a few, not all of them, particularly the old and decrepit ones, he found to his shock and surprise they were littered with stray bits of paper. He knew then that the paper had infected these houses, made copies of itself, and blew up the house. A young boy on a bicycle at one of the sites told him he had seen a demolition crew using dynamite to explode the house the previous week, but Cowan dismissed this as the idle thoughts of silly boys because ‘I was on to something big’. He was on to how ‘scientists’ mistake genetic material in the detoxifying process for something they call a ‘virus’. Cowan said of his house and paper story:

If this sounds crazy to you, it’s because it should. This scenario is obviously nuts. But consider this admittedly embellished, for effect, current viral theory that all scientists, medical doctors and virologists currently believe.

He takes the example of the ‘novel SARS-Cov2’ virus to prove the point. First they take someone with an undefined illness called ‘Covid-19’ and don’t even attempt to find any virus in their sputum. Never mind the scientists still describe how this ‘virus’, which they have not located attaches to a cell receptor, injects its genetic material, in ‘Covid’s’ case, RNA, into the cell. The RNA once inserted exploits the cell to reproduce itself and makes ‘thousands, nay millions, of copies of itself ... Then it emerges victorious to claim its next victim’:

If you were to look in the scientific literature for proof, actual scientific proof, that uniform SARS-CoV2 viruses have been properly isolated from the sputum of a sick person, that actual spike proteins could be seen protruding from the virus (which has not been found), you would find that such evidence doesn't exist.

If you go looking in the published scientific literature for actual pictures, proof, that these spike proteins or any viral proteins are ever attached to any receptor embedded in any cell membrane, you would also find that no such evidence exists. If you were to look for a video or documented evidence of the intact virus injecting its genetic material into the body of the cell, reproducing itself and then emerging victorious by budding off the cell membrane, you would find that no such evidence exists.

The closest thing you would find is electron micrograph pictures of cellular particles, possibly attached to cell debris, both of which to be seen were stained by heavy metals, a process that completely distorts their architecture within the living organism. This is like finding bits of paper stuck to the blown-up bricks, thereby proving the paper emerged by taking pieces of the bricks on its way out.

The Enders baloney

Cowan describes the 'Covid' story as being just as make-believe as his paper story and he charts back this fantasy to a Nobel Prize winner called John Enders (1897-1985), an American biomedical scientist who has been dubbed 'The Father of Modern Vaccines'. Enders is claimed to have 'discovered' the process of the viral culture which 'proved' that a 'virus' caused measles. Cowan explains how Enders did this 'by using the EXACT same procedure that has been followed by every virologist to find and characterize every new virus since 1954'. Enders took throat swabs from children with measles and immersed them in 2ml of milk. Penicillin (100u/ml) and the antibiotic streptomycin (50,g/ml) were added and the whole mix was centrifuged – rotated at high speed to separate large cellular debris from small particles and molecules as with milk and cream, for example. Cowan says that if the aim is to find little particles of genetic material ('viruses') in the snot from children with measles it would seem that the last thing you would do is mix the snot with other material – milk –that also has genetic material. 'How are you ever going to know whether whatever you found came from the snot or the milk?' He points out that streptomycin is a 'nephrotoxic' or poisonous-to-the-kidney drug. You will see the relevance of that

shortly. Cowan says that it gets worse, much worse, when Enders describes the culture medium upon which the virus 'grows': 'The culture medium consisted of bovine amniotic fluid (90%), beef embryo extract (5%), horse serum (5%), antibiotics and phenol red as an indicator of cell metabolism.' Cowan asks incredulously: 'Did he just say that the culture medium also contained fluids and tissues that are themselves rich sources of genetic material?' The genetic cocktail, or 'medium', is inoculated onto tissue and cells from rhesus monkey *kidney* tissue. This is where the importance of streptomycin comes in and currently-used antimicrobials and other drugs that are *poisonous to kidneys* and used in ALL modern viral cultures (e.g. gentamicin, streptomycin, and amphotericin). Cowan asks: 'How are you ever going to know from this witch's brew where any genetic material comes from as we now have five different sources of rich genetic material in our mix?' Remember, he says, that all genetic material, whether from monkey kidney tissues, bovine serum, milk, etc., is made from the exact same components. The same central question returns: 'How are you possibly going to know that it was the virus that killed the kidney tissue and not the toxic antibiotic and starvation rations on which you are growing the tissue?' John Enders answered the question himself – *you can't*:

A second agent was obtained from an uninoculated culture of monkey kidney cells. The cytopathic changes [death of the cells] it induced in the unstained preparations could not be distinguished with confidence from the viruses isolated from measles.

The death of the cells ('cytopathic changes') happened in exactly the same manner, whether they inoculated the kidney tissue with the measles snot or not, Cowan says. 'This is evidence that the destruction of the tissue, the very proof of viral causation of illness, was not caused by anything in the snot because they saw the same destructive effect when the snot was not even used ... the cytopathic, i.e., cell-killing, changes come from the process of the culture itself, not from any virus in any snot, period.' Enders quotes in his 1957 paper a virologist called Ruckle as reporting similar findings 'and in addition has isolated an agent from monkey kidney tissue that is so

far indistinguishable from human measles virus'. In other words, Cowan says, these particles called 'measles viruses' are simply and clearly breakdown products of the starved and poisoned tissue. For measles 'virus' see all 'viruses' including the so-called 'Covid virus'. Enders, the 'Father of Modern Vaccines', also said:

There is a potential risk in employing cultures of primate cells for the production of vaccines composed of attenuated virus, since the presence of other agents possibly latent in primate tissues cannot be definitely excluded by any known method.

Cowan further quotes from a paper published in the journal *Viruses* in May, 2020, while the 'Covid pandemic' was well underway in the media if not in reality. 'EVs' here refers to particles of genetic debris from our own tissues, such as exosomes of which more in a moment: 'The remarkable resemblance between EVs and viruses has caused quite a few problems in the studies focused on the analysis of EVs released during viral infections.' Later the paper adds that to date a reliable method that can actually guarantee a complete separation (of EVs from viruses) DOES NOT EXIST. This was published at a time when a fairy tale 'virus' was claimed in total certainty to be causing a fairy tale 'viral disease' called 'Covid-19' – a fairy tale that was already well on the way to transforming human society in the image that the Cult has worked to achieve for so long. Cowan concludes his article:

To summarize, there is no scientific evidence that pathogenic viruses exist. What we think of as 'viruses' are simply the normal breakdown products of dead and dying tissues and cells. When we are well, we make fewer of these particles; when we are starved, poisoned, suffocated by wearing masks, or afraid, we make more.

There is no engineered virus circulating and making people sick. People in laboratories all over the world are making genetically modified products to make people sick. These are called vaccines. There is no virome, no 'ecosystem' of viruses, viruses are not 8%, 50% or 100 % of our genetic material. These are all simply erroneous ideas based on the misconception called a virus.

What is 'Covid'? Load of bollocks

The background described here by Cowan and Lanka was emphasised in the first video presentation that I saw by Dr Andrew Kaufman when he asked whether the ‘Covid virus’ was in truth a natural defence mechanism of the body called ‘exosomes’. These are released by cells when in states of toxicity – see the same themes returning over and over. They are released ever more profusely as chemical and radiation toxicity increases and think of the potential effect therefore of 5G alone as its destructive frequencies infest the human energetic information field with a gathering pace (5G went online in Wuhan in 2019 as the ‘virus’ emerged). I’ll have more about this later. Exosomes transmit a warning to the rest of the body that ‘Houston, we have a problem’. Kaufman presented images of exosomes and compared them with ‘Covid’ under an electron microscope and the similarity was remarkable. They both attach to the same cell receptors (*claimed* in the case of ‘Covid’), contain the same genetic material in the form of RNA or ribonucleic acid, and both are found in ‘viral cell cultures’ with damaged or dying cells. James Hildreth MD, President and Chief Executive Officer of the Meharry Medical College at Johns Hopkins, said: ‘The virus is fully an exosome in every sense of the word.’ Kaufman’s conclusion was that there is no ‘virus’: ‘This entire pandemic is a completely manufactured crisis … there is no evidence of anyone dying from [this] illness.’ Dr Tom Cowan and Sally Fallon Morell, authors of *The Contagion Myth*, published a statement with Dr Kaufman in February, 2021, explaining why the ‘virus’ does not exist and you can read it that in full in the Appendix.

‘Virus’ theory can be traced to the ‘cell theory’ in 1858 of German physician Rudolf Virchow (1821-1920) who contended that disease originates from a single cell infiltrated by a ‘virus’. Dr Stefan Lanka said that findings and insights with respect to the structure, function and central importance of tissues in the creation of life, which were already known in 1858, comprehensively refute the cell theory. Virchow ignored them. We have seen the part later played by John Enders in the 1950s and Lanka notes that infection theories were only established as a global dogma through the policies and

eugenics of the Third Reich in Nazi Germany (creation of the same Sabbatian cult behind the ‘Covid’ hoax). Lanka said: ‘Before 1933, scientists dared to contradict this theory; after 1933, these critical scientists were silenced’. Dr Tom Cowan’s view is that ill-health is caused by too much of something, too little of something, or toxification from chemicals and radiation – not contagion. We must also highlight as a major source of the ‘virus’ theology a man still called the ‘Father of Modern Virology’ – Thomas Milton Rivers (1888-1962). There is no way given the Cult’s long game policy that it was a coincidence for the ‘Father of Modern Virology’ to be director of the Rockefeller Institute for Medical Research from 1937 to 1956 when he is credited with making the Rockefeller Institute a leader in ‘viral research’. Cult Rockefellers were the force behind the creation of Big Pharma ‘medicine’, established the World Health Organisation in 1948, and have long and close associations with the Gates family that now runs the WHO during the pandemic hoax through mega-rich Cult gofer and psychopath Bill Gates.

Only a Renegade Mind can see through all this bullshit by asking the questions that need to be answered, not taking ‘no’ or prevarication for an answer, and certainly not hiding from the truth in fear of speaking it. Renegade Minds have always changed the world for the better and they will change this one no matter how bleak it may currently appear to be.

CHAPTER SIX

Sequence of deceit

If you tell the truth, you don't have to remember anything

Mark Twain

Against the background that I have laid out this far the sequence that took us from an invented 'virus' in Cult-owned China in late 2019 to the fascist transformation of human society can be seen and understood in a whole new context.

We were told that a deadly disease had broken out in Wuhan and the world media began its campaign (coordinated by behavioural psychologists as we shall see) to terrify the population into unquestioning compliance. We were shown images of Chinese people collapsing in the street which never happened in the West with what was supposed to be the same condition. In the earliest days when alleged cases and deaths were few the fear register was hysterical in many areas of the media and this would expand into the common media narrative across the world. The real story was rather different, but we were never told that. The Chinese government, one of the Cult's biggest centres of global operation, said they had discovered a new illness with flu-like and pneumonia-type symptoms in a city with such toxic air that it is overwhelmed with flu-like symptoms, pneumonia and respiratory disease. Chinese scientists said it was a new – 'novel' – coronavirus which they called Sars-Cov-2 and that it caused a disease they labelled 'Covid-19'. There was no evidence for this and the 'virus' has never to this day been isolated, purified and its genetic code established from that. It

was from the beginning a computer-generated fiction. Stories of Chinese whistleblowers saying the number of deaths was being suppressed or that the ‘new disease’ was related to the Wuhan bio-lab misdirected mainstream and alternative media into cul-de-sacs to obscure the real truth – there was no ‘virus’.

Chinese scientists took genetic material from the lung fluid of just a few people and said they had found a ‘new’ disease when this material had a wide range of content. There was no evidence for a ‘virus’ for the very reasons explained in the last two chapters. The ‘virus’ has never been shown to (a) exist and (b) cause any disease. People were diagnosed on symptoms that are so widespread in Wuhan and polluted China and with a PCR test that can’t detect infectious disease. On this farce the whole global scam was sold to the rest of the world which would also diagnose respiratory disease as ‘Covid-19’ from symptoms alone or with a PCR test not testing for a ‘virus’. Flu miraculously disappeared *worldwide* in 2020 and into 2021 as it was redesignated ‘Covid-19’. It was really the same old flu with its ‘flu-like’ symptoms attributed to ‘flu-like’ ‘Covid-19’. At the same time with very few exceptions the Chinese response of draconian lockdown and fascism was the chosen weapon to respond across the West as recommended by the Cult-owned Tedros at the Cult-owned World Health Organization run by the Cult-owned Gates. All was going according to plan. Chinese scientists – everything in China is controlled by the Cult-owned government – compared their contaminated RNA lung-fluid material with other RNA sequences and said it appeared to be just under 80 percent identical to the SARS-CoV-1 ‘virus’ claimed to be the cause of the SARS (severe acute respiratory syndrome) ‘outbreak’ in 2003. They decreed that because of this the ‘new virus’ had to be related and they called it SARS-CoV-2. There are some serious problems with this assumption and *assumption* was all it was. Most ‘factual’ science turns out to be assumptions repeated into everyone-knows-that. A match of under 80-percent is meaningless. Dr Kaufman makes the point that there’s a 96 percent genetic correlation between humans and chimpanzees, but ‘no one would say our genetic material is part

of the chimpanzee family'. Yet the Chinese authorities were claiming that a much lower percentage, less than 80 percent, proved the existence of a new 'coronavirus'. For goodness sake human DNA is 60 percent similar to a *banana*.

You are feeling sleepy

The entire 'Covid' hoax is a global Psyop, a psychological operation to program the human mind into believing and fearing a complete fantasy. A crucial aspect of this was what *appeared* to happen in Italy. It was all very well streaming out daily images of an alleged catastrophe in Wuhan, but to the Western mind it was still on the other side of the world in a very different culture and setting. A reaction of 'this could happen to me and my family' was still nothing like as intense enough for the mind-doctors. The Cult needed a Western example to push people over that edge and it chose Italy, one of its major global locations going back to the Roman Empire. An Italian 'Covid' crisis was manufactured in a particular area called Lombardy which just happens to be notorious for its toxic air and therefore respiratory disease. Wuhan, China, *déjà vu*. An hysterical media told horror stories of Italians dying from 'Covid' in their droves and how Lombardy hospitals were being overrun by a tidal wave of desperately ill people needing treatment after being struck down by the 'deadly virus'. Here was the psychological turning point the Cult had planned. Wow, if this is happening in Italy, the Western mind concluded, this indeed could happen to me and my family. Another point is that Italian authorities responded by following the Chinese blueprint so vehemently recommended by the Cult-owned World Health Organization. They imposed fascistic lockdowns on the whole country viciously policed with the help of surveillance drones sweeping through the streets seeking out anyone who escaped from mass house arrest. Livelihoods were destroyed and psychology unravelled in the way we have witnessed since in all lockdown countries. Crucial to the plan was that Italy responded in this way to set the precedent of suspending freedom and imposing fascism in a 'Western liberal democracy'. I emphasised in an

animated video explanation on davidicke.com posted in the summer of 2020 how important it was to the Cult to expand the Chinese lockdown model across the West. Without this, and the bare-faced lie that non-symptomatic people could still transmit a ‘disease’ they didn’t have, there was no way locking down the whole population, sick and not sick, could be pulled off. At just the right time and with no evidence Cult operatives and gofers claimed that people without symptoms could pass on the ‘disease’. In the name of protecting the ‘vulnerable’ like elderly people, who lockdowns would kill by the tens of thousands, we had for the first time healthy people told to isolate as well as the sick. The great majority of people who tested positive had no symptoms because there was nothing wrong with them. It was just a trick made possible by a test not testing for the ‘virus’.

Months after my animated video the Gates-funded Professor Neil Ferguson at the Gates-funded Imperial College confirmed that I was right. He didn’t say it in those terms, naturally, but he did say it. Ferguson will enter the story shortly for his outrageously crazy ‘computer models’ that led to Britain, the United States and many other countries following the Chinese and now Italian methods of response. Put another way, following the Cult script. Ferguson said that SAGE, the UK government’s scientific advisory group which has controlled ‘Covid’ policy from the start, wanted to follow the Chinese lockdown model (while they all continued to work and be paid), but they wondered if they could possibly, in Ferguson’s words, ‘get away with it in Europe’. ‘Get away with it’? Who the hell do these moronic, arrogant people think they are? This appalling man Ferguson said that once Italy went into national lockdown they realised they, too, could mimic China:

It’s a communist one-party state, we said. We couldn’t get away with it in Europe, we thought ... and then Italy did it. And we realised we could. Behind this garbage from Ferguson is a simple fact: Doing the same as China in every country was the plan from the start and Ferguson’s ‘models’ would play a central role in achieving that. It’s just a coincidence, of course, and absolutely nothing to worry your little head about.

Oops, sorry, our mistake

Once the Italian segment of the Psyop had done the job it was designed to do a very different story emerged. Italian authorities revealed that 99 percent of those who had 'died from Covid-19' in Italy had one, two, three, or more 'co-morbidities' or illnesses and health problems that could have ended their life. The US Centers for Disease Control and Prevention (CDC) published a figure of 94 percent for Americans dying of 'Covid' while having other serious medical conditions – on average two to three (some five or six) other potential causes of death. In terms of death from an unproven 'virus' I say it is 100 percent. The other one percent in Italy and six percent in the US would presumably have died from 'Covid's' flu-like symptoms with a range of other possible causes in conjunction with a test not testing for the 'virus'. Fox News reported that even more startling figures had emerged in one US county in which 410 of 422 deaths attributed to 'Covid-19' had other potentially deadly health conditions. The Italian National Health Institute said later that the average age of people dying with a 'Covid-19' diagnosis in Italy was about 81. Ninety percent were over 70 with ten percent over 90. In terms of other reasons to die some 80 percent had two or more chronic diseases with half having three or more including cardiovascular problems, diabetes, respiratory problems and cancer. Why is the phantom 'Covid-19' said to kill overwhelmingly old people and hardly affect the young? Old people continually die of many causes and especially respiratory disease which you can re-diagnose 'Covid-19' while young people die in tiny numbers by comparison and rarely of respiratory disease. Old people 'die of Covid' because they die of other things that can be redesignated 'Covid' and it really is that simple.

Flu has flown

The blueprint was in place. Get your illusory 'cases' from a test not testing for the 'virus' and redesignate other causes of death as 'Covid-19'. You have an instant 'pandemic' from something that is nothing more than a computer-generated fiction. With near-on a

billion people having ‘flu-like’ symptoms every year the potential was limitless and we can see why flu quickly and apparently miraculously disappeared *worldwide* by being diagnosed ‘Covid-19’. The painfully bloody obvious was explained away by the childlike media in headlines like this in the UK *‘Independent’*: ‘Not a single case of flu detected by Public Health England this year as Covid restrictions suppress virus’. I kid you not. The masking, social distancing and house arrest that did not make the ‘Covid virus’ disappear somehow did so with the ‘flu virus’. Even worse the article, by a bloke called Samuel Lovett, suggested that maybe the masking, sanitising and other ‘Covid’ measures should continue to keep the flu away. With a ridiculousness that disturbs your breathing (it’s ‘Covid-19’) the said Lovett wrote: ‘With widespread social distancing and mask-wearing measures in place throughout the UK, the usual routes of transmission for influenza have been blocked.’ He had absolutely no evidence to support that statement, but look at the consequences of him acknowledging the obvious. With flu not disappearing at all and only being relabelled ‘Covid-19’ he would have to contemplate that ‘Covid’ was a hoax on a scale that is hard to imagine. You need guts and commitment to truth to even go there and that’s clearly something Samuel Lovett does not have in abundance. He would never have got it through the editors anyway.

Tens of thousands die in the United States alone every winter from flu including many with pneumonia complications. CDC figures record *45 million* Americans diagnosed with flu in 2017-2018 of which 61,000 died and some reports claim 80,000. Where was the same hysteria then that we have seen with ‘Covid-19’? Some 250,000 Americans are admitted to hospital with pneumonia every year with about 50,000 cases proving fatal. About 65 million suffer respiratory disease every year and three million deaths makes this the third biggest cause of death worldwide. You only have to redesignate a portion of all these people ‘Covid-19’ and you have an instant global pandemic or the *appearance* of one. Why would doctors do this? They are told to do this and all but a few dare not refuse those who must be obeyed. Doctors in general are not researching their own

knowledge and instead take it direct and unquestioned from the authorities that own them and their careers. The authorities say they must now diagnose these symptoms ‘Covid-19’ and not flu, or whatever, and they do it. Dark suits say put ‘Covid-19’ on death certificates no matter what the cause of death and the doctors do it. Renegade Minds don’t fall for the illusion that doctors and medical staff are all highly-intelligent, highly-principled, seekers of medical truth. *Some are*, but not the majority. They are repeaters, gofers, and yes sir, no sir, purveyors of what the system demands they purvey. The ‘Covid’ con is not merely confined to diseases of the lungs. Instructions to doctors to put ‘Covid-19’ on death certificates for anyone dying of *anything* within 28 days (or much more) of a positive test not testing for the ‘virus’ opened the floodgates. The term dying *with* ‘Covid’ and not *of* ‘Covid’ was coined to cover the truth. Whether it was a *with* or an *of* they were all added to the death numbers attributed to the ‘deadly virus’ compiled by national governments and globally by the Gates-funded Johns Hopkins operation in the United States that was so involved in those ‘pandemic’ simulations. Fraudulent deaths were added to the ever-growing list of fraudulent ‘cases’ from false positives from a false test. No wonder Professor Walter Ricciardi, scientific advisor to the Italian minister of health, said after the Lombardy hysteria had done its job that ‘Covid’ death rates were due to Italy having the second oldest population in the world and to *how hospitals record deaths*:

The way in which we code deaths in our country is very generous in the sense that all the people who die in hospitals with the coronavirus are deemed to be dying of the coronavirus. On re-evaluation by the National Institute of Health, only 12 per cent of death certificates have shown a direct causality from coronavirus, while 88 per cent of patients who have died have at least one pre-morbidity – many had two or three.

This is extraordinary enough when you consider the propaganda campaign to use Italy to terrify the world, but how can they even say twelve percent were genuine when the ‘virus’ has not been shown to exist, its ‘code’ is a computer program, and diagnosis comes from a test not testing for it? As in China, and soon the world, ‘Covid-19’ in

Italy was a redesignation of diagnosis. Lies and corruption were to become the real ‘pandemic’ fuelled by a pathetically-compliant medical system taking its orders from the tiny few at the top of their national hierarchy who answered to the World Health Organization which answers to Gates and the Cult. Doctors were told – ordered – to diagnose a particular set of symptoms ‘Covid-19’ and put that on the death certificate for any cause of death if the patient had tested positive with a test not testing for the virus or had ‘Covid’ symptoms like the flu. The United States even introduced big financial incentives to manipulate the figures with hospitals receiving £4,600 from the Medicare system for diagnosing someone with regular pneumonia, \$13,000 if they made the diagnosis from the same symptoms ‘Covid-19’ pneumonia, and \$39, 000 if they put a ‘Covid’ diagnosed patient on a ventilator that would almost certainly kill them. A few – painfully and pathetically few – medical whistleblowers revealed (before Cult-owned YouTube deleted their videos) that they had been instructed to ‘let the patient crash’ and put them straight on a ventilator instead of going through a series of far less intrusive and dangerous methods as they would have done before the pandemic hoax began and the financial incentives kicked in. We are talking cold-blooded murder given that ventilators are so damaging to respiratory systems they are usually the last step before heaven awaits. Renegade Minds never fall for the belief that people in white coats are all angels of mercy and cannot be full-on psychopaths. I have explained in detail in *The Answer* how what I am describing here played out across the world coordinated by the World Health Organization through the medical hierarchies in almost every country.

Medical scientist calls it

Information about the non-existence of the ‘virus’ began to emerge for me in late March, 2020, and mushroomed after that. I was sent an email by Sir Julian Rose, a writer, researcher, and organic farming promotor, from a medical scientist friend of his in the United States. Even at that early stage in March the scientist was able to explain

how the ‘Covid’ hoax was being manipulated. He said there were no reliable tests for a specific ‘Covid-19 virus’ and nor were there any reliable agencies or media outlets for reporting numbers of actual ‘Covid-19’ cases. We have seen in the long period since then that he was absolutely right. ‘Every action and reaction to Covid-19 is based on totally flawed data and we simply cannot make accurate assessments,’ he said. Most people diagnosed with ‘Covid-19’ were showing nothing more than cold and flu-like symptoms ‘because most coronavirus strains *are* nothing more than cold/flu-like symptoms’. We had farcical situations like an 84-year-old German man testing positive for ‘Covid-19’ and his nursing home ordered to quarantine only for him to be found to have a common cold. The scientist described back then why PCR tests and what he called the ‘Mickey Mouse test kits’ were useless for what they were claimed to be identifying. ‘The idea these kits can isolate a specific virus like Covid-19 is nonsense,’ he said. Significantly, he pointed out that ‘if you want to create a totally false panic about a totally false pandemic – pick a coronavirus’. This is exactly what the Cult-owned Gates, World Economic Forum and Johns Hopkins University did with their Event 201 ‘simulation’ followed by their real-life simulation called the ‘pandemic’. The scientist said that all you had to do was select the sickest of people with respiratory-type diseases in a single location – ‘say Wuhan’ – and administer PCR tests to them. You can then claim that anyone showing ‘viral sequences’ similar to a coronavirus ‘which will inevitably be quite a few’ is suffering from a ‘new’ disease:

Since you already selected the sickest flu cases a fairly high proportion of your sample will go on to die. You can then say this ‘new’ virus has a CFR [case fatality rate] higher than the flu and use this to infuse more concern and do more tests which will of course produce more ‘cases’, which expands the testing, which produces yet more ‘cases’ and so on and so on. Before long you have your ‘pandemic’, and all you have done is use a simple test kit trick to convert the worst flu and pneumonia cases into something new that doesn’t ACTUALLY EXIST [my emphasis].

He said that you then ‘just run the same scam in other countries’ and make sure to keep the fear message running high ‘so that people

will feel panicky and less able to think critically'. The only problem to overcome was the fact *there is no* actual new deadly pathogen and only regular sick people. This meant that deaths from the 'new deadly pathogen' were going to be way too low for a real new deadly virus pandemic, but he said this could be overcome in the following ways – all of which would go on to happen:

1. You can claim this is just the beginning and more deaths are imminent [you underpin this with fantasy 'computer projections']. Use this as an excuse to quarantine everyone and then claim the quarantine prevented the expected millions of dead.
2. You can [say that people] 'minimizing' the dangers are irresponsible and bully them into not talking about numbers.
3. You can talk crap about made up numbers hoping to blind people with pseudoscience.
4. You can start testing well people (who, of course, will also likely have shreds of coronavirus [RNA] in them) and thus inflate your 'case figures' with 'asymptomatic carriers' (you will of course have to spin that to sound deadly even though any virologist knows the more symptom-less cases you have the less deadly is your pathogen).

The scientist said that if you take these simple steps 'you can have your own entirely manufactured pandemic up and running in weeks'. His analysis made so early in the hoax was brilliantly prophetic of what would actually unfold. Pulling all the information together in these recent chapters we have this is simple 1, 2, 3, of how you can delude virtually the entire human population into believing in a 'virus' that doesn't exist:

- A 'Covid case' is someone who tests positive with a test not testing for the 'virus'.
- A 'Covid death' is someone who dies of *any cause* within 28 days (or much longer) of testing positive with a test not testing for the 'virus'.
- Asymptomatic means there is nothing wrong with you, but they claim you can pass on what you don't have to justify locking

down (quarantining) healthy people in totality.

The foundations of the hoax are that simple. A study involving ten million people in Wuhan, published in November, 2020, demolished the whole lie about those without symptoms passing on the ‘virus’. They found ‘300 asymptomatic cases’ and traced their contacts to find that not one of them was detected with the ‘virus’.

‘Asymptomatic’ patients and their contacts were isolated for no less than two weeks and nothing changed. I know it’s all crap, but if you are going to claim that those without symptoms can transmit ‘the virus’ then you must produce evidence for that and they never have. Even World Health Organization official Dr Maria Van Kerkhove, head of the emerging diseases and zoonosis unit, said as early as June, 2020, that she doubted the validity of asymptomatic transmission. She said that ‘from the data we have, it still seems to be rare that an asymptomatic person actually transmits onward to a secondary individual’ and by ‘rare’ she meant that she couldn’t cite any case of asymptomatic transmission.

The Ferguson factor

The problem for the Cult as it headed into March, 2020, when the script had lockdown due to start, was that despite all the manipulation of the case and death figures they still did not have enough people alleged to have died from ‘Covid’ to justify mass house arrest. This was overcome in the way the scientist described: ‘You can claim this is just the beginning and more deaths are imminent ... Use this as an excuse to quarantine everyone and then claim the quarantine prevented the expected millions of dead.’ Enter one Professor Neil Ferguson, the Gates-funded ‘epidemiologist’ at the Gates-funded Imperial College in London. Ferguson is Britain’s Christian Drosten in that he has a dire record of predicting health outcomes, but is still called upon to advise government on the next health outcome when another ‘crisis’ comes along. This may seem to be a strange and ridiculous thing to do. Why would you keep turning for policy guidance to people who have a history of being

monumentally wrong? Ah, but it makes sense from the Cult point of view. These ‘experts’ keep on producing predictions that suit the Cult agenda for societal transformation and so it was with Neil Ferguson as he revealed his horrific (and clearly insane) computer model predictions that allowed lockdowns to be imposed in Britain, the United States and many other countries. Ferguson does not have even an A-level in biology and would appear to have no formal training in computer modelling, medicine or epidemiology, according to Derek Winton, an MSc in Computational Intelligence. He wrote an article somewhat aghast at what Ferguson did which included taking no account of respiratory disease ‘seasonality’ which means it is far worse in the winter months. Who would have thought that respiratory disease could be worse in the winter? Well, certainly not Ferguson.

The massively China-connected Imperial College and its bizarre professor provided the excuse for the long-incubated Chinese model of human control to travel westward at lightning speed. Imperial College confirms on its website that it collaborates with the Chinese Research Institute; publishes more than 600 research papers every year with Chinese research institutions; has 225 Chinese staff; 2,600 Chinese students – the biggest international group; 7,000 former students living in China which is the largest group outside the UK; and was selected for a tour by China’s President Xi Jinping during his state visit to the UK in 2015. The college takes major donations from China and describes itself as the UK’s number one university collaborator with Chinese research institutions. The China communist/fascist government did not appear phased by the woeful predictions of Ferguson and Imperial when during the lockdown that Ferguson induced the college signed a five-year collaboration deal with China tech giant Huawei that will have Huawei’s indoor 5G network equipment installed at the college’s West London tech campus along with an ‘AI cloud platform’. The deal includes Chinese sponsorship of Imperial’s Venture Catalyst entrepreneurship competition. Imperial is an example of the enormous influence the Chinese government has within British and North American

universities and research centres – and further afield. Up to 200 academics from more than a dozen UK universities are being investigated on suspicion of ‘unintentionally’ helping the Chinese government build weapons of mass destruction by ‘transferring world-leading research in advanced military technology such as aircraft, missile designs and cyberweapons’. Similar scandals have broken in the United States, but it’s all a coincidence. Imperial College serves the agenda in many other ways including the promotion of every aspect of the United Nations Agenda 21/2030 (the Great Reset) and produced computer models to show that human-caused ‘climate change’ is happening when in the real world it isn’t. Imperial College is driving the climate agenda as it drives the ‘Covid’ agenda (both Cult hoaxes) while Patrick Vallance, the UK government’s Chief Scientific Adviser on ‘Covid’, was named Chief Scientific Adviser to the UN ‘climate change’ conference known as COP26 hosted by the government in Glasgow, Scotland. ‘Covid’ and ‘climate’ are fundamentally connected.

Professor Woeful

From Imperial’s bosom came Neil Ferguson still advising government despite his previous disasters and it was announced early on that he and other key people like UK Chief Medical Adviser Chris Whitty had caught the ‘virus’ as the propaganda story was being sold. Somehow they managed to survive and we had Prime Minister Boris Johnson admitted to hospital with what was said to be a severe version of the ‘virus’ in this same period. His whole policy and demeanour changed when he returned to Downing Street. It’s a small world with these government advisors – especially in their communal connections to Gates – and Ferguson had partnered with Whitty to write a paper called ‘Infectious disease: Tough choices to reduce Ebola transmission’ which involved another scare-story that didn’t happen. Ferguson’s ‘models’ predicted that up to 150, 000 could die from ‘mad cow disease’, or BSE, and its version in sheep if it was transmitted to humans. BSE was not transmitted and instead triggered by an organophosphate pesticide used to treat a pest on

cows. Fewer than 200 deaths followed from the human form. Models by Ferguson and his fellow incompetents led to the unnecessary culling of millions of pigs, cattle and sheep in the foot and mouth outbreak in 2001 which destroyed the lives and livelihoods of farmers and their families who had often spent decades building their herds and flocks. Vast numbers of these animals did not have foot and mouth and had no contact with the infection. Another ‘expert’ behind the cull was Professor Roy Anderson, a computer modeller at Imperial College specialising in the epidemiology of *human*, not animal, disease. Anderson has served on the Bill and Melinda Gates Grand Challenges in Global Health advisory board and chairs another Gates-funded organisation. Gates is everywhere.

In a precursor to the ‘Covid’ script Ferguson backed closing schools ‘for prolonged periods’ over the swine flu ‘pandemic’ in 2009 and said it would affect a third of the world population if it continued to spread at the speed he claimed to be happening. His mates at Imperial College said much the same and a news report said: ‘One of the authors, the epidemiologist and disease modeller Neil Ferguson, who sits on the World Health Organisation’s emergency committee for the outbreak, said the virus had “full pandemic potential”.’ Professor Liam Donaldson, the Chris Whitty of his day as Chief Medical Officer, said the worst case could see 30 percent of the British people infected by swine flu with 65,000 dying. Ferguson and Donaldson were indeed proved correct when at the end of the year the number of deaths attributed to swine flu was 392. The term ‘expert’ is rather liberally applied unfortunately, not least to complete idiots. Swine flu ‘projections’ were great for GlaxoSmithKline (GSK) as millions rolled in for its Pandemrix influenza vaccine which led to brain damage with children most affected. The British government (taxpayers) paid out more than £60 million in compensation after GSK was given immunity from prosecution. Yet another ‘Covid’ déjà vu. Swine flu was supposed to have broken out in Mexico, but Dr Wolfgang Wodarg, a German doctor, former member of parliament and critic of the ‘Covid’ hoax, observed ‘the spread of swine flu’ in Mexico City at the time. He

said: 'What we experienced in Mexico City was a very mild flu which did not kill more than usual – which killed even fewer people than usual.' Hyping the fear against all the facts is not unique to 'Covid' and has happened many times before. Ferguson is reported to have over-estimated the projected death toll of bird flu (H5N1) by some three million-fold, but bird flu vaccine makers again made a killing from the scare. This is some of the background to the Neil Ferguson who produced the perfectly-timed computer models in early 2020 predicting that half a million people would die in Britain without draconian lockdown and 2.2 million in the United States. Politicians panicked, people panicked, and lockdowns of alleged short duration were instigated to 'flatten the curve' of cases gleaned from a test not testing for the 'virus'. I said at the time that the public could forget the 'short duration' bit. This was an agenda to destroy the livelihoods of the population and force them into mass control through dependency and there was going to be nothing 'short' about it. American researcher Daniel Horowitz described the consequences of the 'models' spewed out by Gates-funded Ferguson and Imperial College:

What led our government and the governments of many other countries into panic was a single Imperial College of UK study, funded by global warming activists, that predicted 2.2 million deaths if we didn't lock down the country. In addition, the reported 8-9% death rate in Italy scared us into thinking there was some other mutation of this virus that they got, which might have come here.

Together with the fact that we were finally testing and had the ability to actually report new cases, we thought we were headed for a death spiral. But again ... we can't flatten a curve if we don't know when the curve started.

How about it *never* started?

Giving them what they want

An investigation by German news outlet *Welt Am Sonntag* (*World on Sunday*) revealed how in March, 2020, the German government gathered together 'leading scientists from several research institutes and universities' and 'together, they were to produce a [modelling]

paper that would serve as legitimization for further tough political measures'. The Cult agenda was justified by computer modelling not based on evidence or reality; it was specifically constructed to justify the Cult demand for lockdowns all over the world to destroy the independent livelihoods of the global population. All these modellers and everyone responsible for the 'Covid' hoax have a date with a trial like those in Nuremberg after World War Two when Nazis faced the consequences of their war crimes. These corrupt-beyond-belief 'modellers' wrote the paper according to government instructions and it said that if lockdown measures were lifted then up to one million Germans would die from 'Covid-19' adding that some would die 'agonizingly at home, gasping for breath' unable to be treated by hospitals that couldn't cope. All lies. No matter – it gave the Cult all that it wanted. What did long-time government 'modeller' Neil Ferguson say? If the UK and the United States didn't lockdown half a million would die in Britain and 2.2 million Americans. Anyone see a theme here? 'Modellers' are such a crucial part of the lockdown strategy that we should look into their background and follow the money. Researcher Rosemary Frei produced an excellent article headlined 'The Modelling-paper Mafiosi'. She highlights a guy called John Edmunds, a British epidemiologist, and professor in the Faculty of Epidemiology and Population Health at the London School of Hygiene & Tropical Medicine. He studied at Imperial College. Edmunds is a member of government 'Covid' advisory bodies which have been dictating policy, the New and Emerging Respiratory Virus Threats Advisory Group (NERVTAG) and the Scientific Advisory Group for Emergencies (SAGE).

Ferguson, another member of NERVTAG and SAGE, led the way with the original 'virus' and Edmunds has followed in the 'variant' stage and especially the so-called UK or Kent variant known as the 'Variant of Concern' (VOC) B.1.1.7. He said in a co-written report for the Centre for Mathematical modelling of Infectious Diseases at the London School of Hygiene and Tropical Medicine, with input from the Centre's 'Covid-19' Working Group, that there was 'a realistic

possibility that VOC B.1.1.7 is associated with an increased risk of death compared to non-VOC viruses'. Fear, fear, fear, get the vaccine, fear, fear, fear, get the vaccine. Rosemary Frei reveals that almost all the paper's authors and members of the modelling centre's 'Covid-19' Working Group receive funding from the Bill and Melinda Gates Foundation and/or the associated Gates-funded Wellcome Trust. The paper was published by e-journal *Medr* ^{xiv} which only publishes papers not peer-reviewed and the journal was established by an organisation headed by Facebook's Mark Zuckerberg and his missus. What a small world it is. Frei discovered that Edmunds is on the Scientific Advisory Board of the Coalition for Epidemic Preparedness Innovations (CEPI) which was established by the Bill and Melinda Gates Foundation, Klaus Schwab's Davos World Economic Forum and Big Pharma giant Wellcome. CEPI was 'launched in Davos [in 2017] to develop vaccines to stop future epidemics', according to its website. 'Our mission is to accelerate the development of vaccines against emerging infectious diseases and enable equitable access to these vaccines for people during outbreaks.' What kind people they are. Rosemary Frei reveals that Public Health England (PHE) director Susan Hopkins is an author of her organisation's non-peer-reviewed reports on 'new variants'. Hopkins is a professor of infectious diseases at London's Imperial College which is gifted tens of millions of dollars a year by the Bill and Melinda Gates Foundation. Gates-funded modelling disaster Neil Ferguson also co-authors Public Health England reports and he spoke in December, 2020, about the potential danger of the B.1.1.7. 'UK variant' promoted by Gates-funded modeller John Edmunds. When I come to the 'Covid vaccines' the 'new variants' will be shown for what they are – bollocks.

Connections, connections

All these people and modellers are lockdown-obsessed or, put another way, they demand what the Cult demands. Edmunds said in January, 2021, that to ease lockdowns too soon would be a disaster and they had to 'vaccinate much, much, much more widely than the

elderly'. Rosemary Frei highlights that Edmunds is married to Jeanne Pimenta who is described in a LinkedIn profile as director of epidemiology at GlaxoSmithKline (GSK) and she held shares in the company. Patrick Vallance, co-chair of SAGE and the government's Chief Scientific Adviser, is a former executive of GSK and has a deferred bonus of shares in the company worth £600,000. GSK has serious business connections with Bill Gates and is collaborating with mRNA-'vaccine' company CureVac to make 'vaccines' for the new variants that Edmunds is talking about. GSK is planning a 'Covid vaccine' with drug giant Sanofi. Puppet Prime Minister Boris Johnson announced in the spring of 2021 that up to 60 million vaccine doses were to be made at the GSK facility at Barnard Castle in the English North East. Barnard Castle, with a population of just 6,000, was famously visited in breach of lockdown rules in April, 2020, by Johnson aide Dominic Cummings who said that he drove there 'to test his eyesight' before driving back to London. Cummings would be better advised to test his integrity – not that it would take long. The GSK facility had nothing to do with his visit then although I'm sure Patrick Vallance would have been happy to arrange an introduction and some tea and biscuits. Ruthless psychopath Gates has made yet another fortune from vaccines in collaboration with Big Pharma companies and gushes at the phenomenal profits to be made from vaccines – more than a 20-to-1 return as he told one interviewer. Gates also tweeted in December, 2019, with the foreknowledge of what was coming: 'What's next for our foundation? I'm particularly excited about what the next year could mean for one of the best buys in global health: vaccines.'

Modeller John Edmunds is a big promotor of vaccines as all these people appear to be. He's the dean of the London School of Hygiene & Tropical Medicine's Faculty of Epidemiology and Population Health which is primarily funded by the Bill and Melinda Gates Foundation and the Gates-established and funded GAVI vaccine alliance which is the Gates vehicle to vaccinate the world. The organisation Doctors Without Borders has described GAVI as being 'aimed more at supporting drug-industry desires to promote new

products than at finding the most efficient and sustainable means for fighting the diseases of poverty'. But then that's why the psychopath Gates created it. John Edmunds said in a video that the London School of Hygiene & Tropical Medicine is involved in every aspect of vaccine development including large-scale clinical trials. He contends that mathematical modelling can show that vaccines protect individuals and society. That's on the basis of shit in and shit out, I take it. Edmunds serves on the UK Vaccine Network as does Ferguson and the government's foremost 'Covid' adviser, the grim-faced, dark-eyed Chris Whitty. The Vaccine Network says it works 'to support the government to identify and shortlist targeted investment opportunities for the most promising vaccines and vaccine technologies that will help combat infectious diseases with epidemic potential, and to address structural issues related to the UK's broader vaccine infrastructure'. Ferguson is acting Director of the Imperial College Vaccine Impact Modelling Consortium which has funding from the Bill and Melina Gates Foundation and the Gates-created GAVI 'vaccine alliance'. Anyone wonder why these characters see vaccines as the answer to every problem? Ferguson is wildly enthusiastic in his support for GAVI's campaign to vaccine children en masse in poor countries. You would expect someone like Gates who has constantly talked about the need to reduce the population to want to fund vaccines to keep more people alive. I'm sure that's why he does it. The John Edmunds London School of Hygiene & Tropical Medicine (LSHTM) has a Vaccines Manufacturing Innovation Centre which develops, tests and commercialises vaccines. Rosemary Frei writes:

The vaccines centre also performs affiliated activities like combating 'vaccine hesitancy'. The latter includes the Vaccine Confidence Project. The project's stated purpose is, among other things, 'to provide analysis and guidance for early response and engagement with the public to ensure sustained confidence in vaccines and immunisation'. The Vaccine Confidence Project's director is LSHTM professor Heidi Larson. For more than a decade she's been researching how to combat vaccine hesitancy.

How the bloody hell can blokes like John Edmunds and Neil Ferguson with those connections and financial ties model 'virus' case

and death projections for the government and especially in a way that gives their paymasters like Gates exactly what they want? It's insane, but this is what you find throughout the world.

'Covid' is not dangerous, oops, wait, yes it is

Only days before Ferguson's nightmare scenario made Jackboot Johnson take Britain into a China-style lockdown to save us from a deadly 'virus' the UK government website gov.uk was reporting something very different to Ferguson on a page of official government guidance for 'high consequence infectious diseases (HCID)'. It said this about 'Covid-19':

As of 19 March 2020, COVID-19 *is no longer considered to be a high consequence infectious diseases (HCID) in the UK* [my emphasis]. The 4 nations public health HCID group made an interim recommendation in January 2020 to classify COVID-19 as an HCID. This was based on consideration of the UK HCID criteria about the virus and the disease with information available during the early stages of the outbreak.

Now that more is known about COVID-19, the public health bodies in the UK have reviewed the most up to date information about COVID-19 against the UK HCID criteria. They have determined that several features have now changed; in particular, more information is available about mortality rates (low overall), and there is now greater clinical awareness and a specific and sensitive laboratory test, the availability of which continues to increase. The Advisory Committee on Dangerous Pathogens (ACDP) is also of the opinion that COVID-19 should no longer be classified as an HCID.

Soon after the government had been exposed for downgrading the risk they upgraded it again and everyone was back to singing from the same Cult hymn book. Ferguson and his fellow Gates clones indicated that lockdowns and restrictions would have to continue until a Gates-funded vaccine was developed. Gates said the same because Ferguson and his like were repeating the Gates script which is the Cult script. 'Flatten the curve' became an ongoing nightmare of continuing lockdowns with periods in between of severe restrictions in pursuit of destroying independent incomes and had nothing to do with protecting health about which the Cult gives not a shit. Why wouldn't Ferguson be pushing a vaccine 'solution' when he's owned by vaccine-obsessive Gates who makes a fortune from them and

when Ferguson heads the Vaccine Impact Modelling Consortium at Imperial College funded by the Gates Foundation and GAVI, the ‘vaccine alliance’, created by Gates as his personal vaccine promotion operation? To compound the human catastrophe that Ferguson’s ‘models’ did so much to create he was later exposed for breaking his own lockdown rules by having sexual liaisons with his married girlfriend Antonia Staats at his home while she was living at another location with her husband and children. Staats was a ‘climate’ activist and senior campaigner at the Soros-funded Avaaz which I wouldn’t trust to tell me that grass is green. Ferguson had to resign as a government advisor over this hypocrisy in May, 2020, but after a period of quiet he was back being quoted by the ridiculous media on the need for more lockdowns and a vaccine rollout. Other government-advising ‘scientists’ from Imperial College held the fort in his absence and said lockdown could be indefinite until a vaccine was found. The Cult script was being sung by the payrolled choir. I said there was no intention of going back to ‘normal’ when the ‘vaccine’ came because the ‘vaccine’ is part of a very different agenda that I will discuss in Human 2.0. Why would the Cult want to let the world go back to normal when destroying that normal forever was the whole point of what was happening? House arrest, closing businesses and schools through lockdown, (un)social distancing and masks all followed the Ferguson fantasy models. Again as I predicted (these people are so predictable) when the ‘vaccine’ arrived we were told that house arrest, lockdown, (un)social distancing and masks would still have to continue. I will deal with the masks in the next chapter because they are of fundamental importance.

Where's the 'pandemic'?

Any mildly in-depth assessment of the figures revealed what was really going on. Cult-funded and controlled organisations still have genuine people working within them such is the number involved. So it is with Genevieve Briand, assistant program director of the Applied Economics master’s degree program at Johns Hopkins

University. She analysed the impact that 'Covid-19' had on deaths from *all* causes in the United States using official data from the CDC for the period from early February to early September, 2020. She found that allegedly 'Covid' *related*-deaths exceeded those from heart disease which she found strange with heart disease always the biggest cause of fatalities. Her research became even more significant when she noted the sudden decline in 2020 of *all* non-'Covid' deaths: 'This trend is completely contrary to the pattern observed in all previous years ... the total decrease in deaths by other causes almost exactly equals the increase in deaths by Covid-19.' This was such a game, set and match in terms of what was happening that Johns Hopkins University deleted the article on the grounds that it 'was being used to support false and dangerous inaccuracies about the impact of the pandemic'. No – because it exposed the scam from official CDC figures and this was confirmed when those figures were published in January, 2021. Here we can see the effect of people dying from heart attacks, cancer, road accidents and gunshot wounds – *anything* – having 'Covid-19' on the death certificate along with those diagnosed from 'symptoms' who had even not tested positive with a test not testing for the 'virus'. I am not kidding with the gunshot wounds, by the way. Brenda Bock, coroner in Grand County, Colorado, revealed that two gunshot victims tested positive for the 'virus' within the previous 30 days and were therefore classified as 'Covid deaths'. Bock said: 'These two people had tested positive for Covid, but that's not what killed them. A gunshot wound is what killed them.' She said she had not even finished her investigation when the state listed the gunshot victims as deaths due to the 'virus'. The death and case figures for 'Covid-19' are an absolute joke and yet they are repeated like parrots by the media, politicians and alleged medical 'experts'. The official Cult narrative is the only show in town.

Genevieve Briand found that deaths from all causes were not exceptional in 2020 compared with previous years and a Spanish magazine published figures that said the same about Spain which was a 'Covid' propaganda hotspot at one point. *Discovery Salud*, a

health and medicine magazine, quoted government figures which showed how 17,000 *fewer* people died in Spain in 2020 than in 2019 and more than 26,000 fewer than in 2018. The age-standardised mortality rate for England and Wales when age distribution is taken into account was significantly lower in 2020 than the 1970s, 80s and 90s, and was only the ninth highest since 2000. Where is the ‘pandemic’?

Post mortems and autopsies virtually disappeared for ‘Covid’ deaths amid claims that ‘virus-infected’ bodily fluids posed a risk to those carrying out the autopsy. This was rejected by renowned German pathologist and forensic doctor Klaus Püschel who said that he and his staff had by then done 150 autopsies on ‘Covid’ patients with no problems at all. He said they were needed to know why some ‘Covid’ patients suffered blood clots and not severe respiratory infections. The ‘virus’ is, after all, called SARS or ‘severe acute respiratory syndrome’. I highlighted in the spring of 2020 this phenomenon and quoted New York intensive care doctor Cameron Kyle-Sidell who posted a soon deleted YouTube video to say that they had been told to prepare to treat an infectious disease called ‘Covid-19’, but that was not what they were dealing with. Instead he likened the lung condition of the most severely ill patients to what you would expect with cabin depressurisation in a plane at 30,000 feet or someone dropped on the top of Everest without oxygen or acclimatisation. I have never said this is not happening to a small minority of alleged ‘Covid’ patients – I am saying this is not caused by a phantom ‘contagious virus’. Indeed Kyle-Sidell said that ‘Covid-19’ was not the disease they were told was coming their way. ‘We are operating under a medical paradigm that is untrue,’ he said, and he believed they were treating the wrong disease: ‘These people are being slowly starved of oxygen.’ Patients would take off their oxygen masks in a state of fear and stress and while they were blue in the face on the brink of death. They did not look like patients dying of pneumonia. You can see why they don’t want autopsies when their virus doesn’t exist and there is another condition in some people that they don’t wish to be uncovered. I should add here that

the 5G system of millimetre waves was being rapidly introduced around the world in 2020 and even more so now as they fire 5G at the Earth from satellites. At 60 gigahertz within the 5G range that frequency interacts with the oxygen molecule and stops people breathing in sufficient oxygen to be absorbed into the bloodstream. They are installing 5G in schools and hospitals. The world is not mad or anything. 5G can cause major changes to the lungs and blood as I detail in *The Answer* and these consequences are labelled 'Covid-19', the alleged symptoms of which can be caused by 5G and other electromagnetic frequencies as cells respond to radiation poisoning.

The 'Covid death' scam

Dr Scott Jensen, a Minnesota state senator and medical doctor, exposed 'Covid' Medicare payment incentives to hospitals and death certificate manipulation. He said he was sent a seven-page document by the US Department of Health 'coaching' him on how to fill out death certificates which had never happened before. The document said that he didn't need to have a laboratory test for 'Covid-19' to put that on the death certificate and that shocked him when death certificates are supposed to be about facts. Jensen described how doctors had been 'encouraged, if not pressured' to make a diagnosis of 'Covid-19' if they thought it was probable or '*presumed*'. No positive test was necessary – not that this would have mattered anyway. He said doctors were told to diagnose 'Covid' by symptoms when these were the same as colds, allergies, other respiratory problems, and certainly with influenza which 'disappeared' in the 'Covid' era. A common sniffle was enough to get the dreaded verdict. Ontario authorities decreed that a single care home resident with *one* symptom from a long list must lead to the isolation of the entire home. Other courageous doctors like Jensen made the same point about death figure manipulation and how deaths by other causes were falling while 'Covid-19 deaths' were rising at the same rate due to re-diagnosis. Their videos rarely survive long on YouTube with its Cult-supporting algorithms courtesy of CEO Susan Wojcicki and her bosses at Google. Figure-tampering was so glaring

and ubiquitous that even officials were letting it slip or outright saying it. UK chief scientific adviser Patrick Vallance said on one occasion that ‘Covid’ on the death certificate doesn’t mean ‘Covid’ was the cause of death (so why the hell is it there?) and we had the rare sight of a BBC reporter telling the truth when she said: ‘Someone could be successfully treated for Covid, in say April, discharged, and then in June, get run over by a bus and die ... That person would still be counted as a Covid death in England.’ Yet the BBC and the rest of the world media went on repeating the case and death figures as if they were real. Illinois Public Health Director Dr Ngozi Ezike revealed the deceit while her bosses must have been clenching their buttocks:

If you were in a hospice and given a few weeks to live and you were then found to have Covid that would be counted as a Covid death. [There might be] a clear alternate cause, but it is still listed as a Covid death. So everyone listed as a Covid death doesn’t mean that was the cause of the death, but that they had Covid at the time of death.

Yes, a ‘Covid virus’ never shown to exist and tested for with a test not testing for the ‘virus’. In the first period of the pandemic hoax through the spring of 2020 the process began of designating almost everything a ‘Covid’ death and this has continued ever since. I sat in a restaurant one night listening to a loud conversation on the next table where a family was discussing in bewilderment how a relative who had no symptoms of ‘Covid’, and had died of a long-term problem, could have been diagnosed a death by the ‘virus’. I could understand their bewilderment. If they read this book they will know why this medical fraud has been perpetrated the world over.

Some media truth shock

The media ignored the evidence of death certificate fraud until eventually one columnist did speak out when she saw it first-hand. Bel Mooney is a long-time national newspaper journalist in Britain currently working for the *Daily Mail*. Her article on February 19th, 2021, carried this headline: ‘My dad Ted passed three Covid tests

and died of a chronic illness yet he's officially one of Britain's 120,000 victims of the virus and is far from alone ... so how many more are there?' She told how her 99-year-old father was in a care home with a long-standing chronic obstructive pulmonary disease and vascular dementia. Maybe, but he was still aware enough to tell her from the start that there was no 'virus' and he refused the 'vaccine' for that reason. His death was not unexpected given his chronic health problems and Mooney said she was shocked to find that 'Covid-19' was declared the cause of death on his death certificate. She said this was a 'bizarre and unacceptable untruth' for a man with long-time health problems who had tested negative twice at the home for the 'virus'. I was also shocked by this story although not by what she said. I had been highlighting the death certificate manipulation for ten months. It was the confirmation that a professional full-time journalist only realised this was going on when it affected her directly and neither did she know that whether her dad tested positive or negative was irrelevant with the test not testing for the 'virus'. Where had she been? She said she did not believe in 'conspiracy theories' without knowing I'm sure that this and 'conspiracy theorists' were terms put into widespread circulation by the CIA in the 1960s to discredit those who did not accept the ridiculous official story of the Kennedy assassination. A blanket statement of 'I don't believe in conspiracy theories' is always bizarre. The dictionary definition of the term alone means the world is drowning in conspiracies. What she said was even more daft when her dad had just been affected by the 'Covid' conspiracy. Why else does she think that 'Covid-19' was going on the death certificates of people who died of something else?

To be fair once she saw from personal experience what was happening she didn't mince words. Mooney was called by the care home on the morning of February 9th to be told her father had died in his sleep. When she asked for the official cause of death what came back was 'Covid-19'. Mooney challenged this and was told there had been deaths from Covid on the dementia floor (confirmed by a test not testing for the 'virus') so they considered it 'reasonable

to assume'. 'But doctor,' Mooney rightly protested, 'an assumption isn't a diagnosis.' She said she didn't blame the perfectly decent and sympathetic doctor – 'he was just doing his job'. Sorry, but that's *bullshit*. He wasn't doing his job at all. He was putting a false cause of death on the death certificate and that is a criminal offence for which he should be brought to account and the same with the millions of doctors worldwide who have done the same. They were not doing their job they were following orders and that must not wash at new Nuremberg trials any more than it did at the first ones. Mooney's doctor was 'assuming' (presuming) as he was told to, but 'just following orders' makes no difference to his actions. A doctor's job is to serve the patient and the truth, not follow orders, but that's what they have done all over the world and played a central part in making the 'Covid' hoax possible with all its catastrophic consequences for humanity. Shame on them and they must answer for their actions. Mooney said her disquiet worsened when she registered her father's death by telephone and was told by the registrar there had been very many other cases like hers where 'the deceased' had not tested positive for 'Covid' yet it was recorded as the cause of death. The test may not matter, but those involved at their level *think* it matters and it shows a callous disregard for accurate diagnosis. The pressure to do this is coming from the top of the national 'health' pyramids which in turn obey the World Health Organization which obeys Gates and the Cult. Mooney said the registrar agreed that this must distort the national figures adding that 'the strangest thing is that every winter we record countless deaths from flu, and this winter there have been none. Not one!' She asked if the registrar thought deaths from flu were being misdiagnosed and lumped together with 'Covid' deaths. The answer was a 'puzzled yes'. Mooney said that the funeral director said the same about 'Covid' deaths which had nothing to do with 'Covid'. They had lost count of the number of families upset by this and other funeral companies in different countries have had the same experience. Mooney wrote:

The nightly shroud-waving and shocking close-ups of pain imposed on us by the TV news bewildered and terrified the population into eager compliance with lockdowns. We were invited to ‘save the NHS’ and to grieve for strangers – the real-life loved ones behind those shocking death counts. Why would the public imagine what I now fear, namely that the way Covid-19 death statistics are compiled might make the numbers seem greater than they are?

Oh, just a little bit – like 100 percent.

Do the maths

Mooney asked why a country would wish to skew its mortality figures by wrongly certifying deaths? What had been going on? Well, if you don’t believe in conspiracies you will never find the answer which is that *it’s a conspiracy*. She did, however, describe what she had discovered as a ‘national scandal’. In reality it’s a global scandal and happening everywhere. Pillars of this conspiracy were all put into place before the button was pressed with the Drosten PCR protocol and high amplifications to produce the cases and death certificate changes to secure illusory ‘Covid’ deaths.

Mooney notes that normally two doctors were needed to certify a death, with one having to know the patient, and how the rules were changed in the spring of 2020 to allow one doctor to do this. In the same period ‘Covid deaths’ were decreed to be all cases where Covid-19 was put on the death certificate even without a positive test or any symptoms. Mooney asked: ‘How many of the 30,851 (as of January 15) care home resident deaths with Covid-19 on the certificate (32.4 per cent of all deaths so far) were based on an assumption, like that of my father? And what has that done to our national psyche?’ All of them is the answer to the first question and it has devastated and dismantled the national psyche, actually the global psyche, on a colossal scale. In the UK case and death data is compiled by organisations like Public Health England (PHE) and the Office for National Statistics (ONS). Mooney highlights the insane policy of counting a death from any cause as ‘Covid-19’ if this happens within 28 days of a positive test (with a test not testing for the ‘virus’) and she points out that ONS statistics reflect deaths ‘involving Covid’ ‘or due to Covid’ which meant in practice any

death where ‘Covid-19’ was mentioned on the death certificate. She described the consequences of this fraud:

Most people will accept the narrative they are fed, so panicky governments here and in Europe witnessed the harsh measures enacted in totalitarian China and jumped into lockdown. Headlines about Covid deaths tolled like the knell that would bring doomsday to us all. Fear stalked our empty streets. Politicians parroted the frankly ridiculous aim of ‘zero Covid’ and shut down the economy, while most British people agreed that lockdown was essential and (astonishingly to me, as a patriotic Brit) even wanted more restrictions.

For what? Lies on death certificates? Never mind the grim toll of lives ruined, suicides, schools closed, rising inequality, depression, cancelled hospital treatments, cancer patients in a torture of waiting, poverty, economic devastation, loneliness, families kept apart, and so on. How many lives have been lost as a direct result of lockdown?

She said that we could join in a national chorus of shock and horror at reaching the 120,000 death toll which was surely certain to have been totally skewed all along, but what about the human cost of lockdown justified by these ‘death figures’? *The British Medical Journal* had reported a 1,493 percent increase in cases of children taken to Great Ormond Street Hospital with abusive head injuries alone and then there was the effect on families:

Perhaps the most shocking thing about all this is that families have been kept apart – and obeyed the most irrational, changing rules at the whim of government – because they believed in the statistics. They succumbed to fear, which his generation rejected in that war fought for freedom. Dad (God rest his soul) would be angry. And so am I.

Another theme to watch is that in the winter months when there are more deaths from all causes they focus on ‘Covid’ deaths and in the summer when the British Lung Foundation says respiratory disease plummets by 80 percent they rage on about ‘cases’. Either way fascism on population is always the answer.

Nazi eugenics in the 21st century

Elderly people in care homes have been isolated from their families month after lonely month with no contact with relatives and grandchildren who were banned from seeing them. We were told

that lockdown fascism was to ‘protect the vulnerable’ like elderly people. At the same time Do Not Resuscitate (DNR) orders were placed on their medical files so that if they needed resuscitation it wasn’t done and ‘Covid-19’ went on their death certificates. Old people were not being ‘protected’ they were being culled – murdered in truth. DNR orders were being decreed for disabled and young people with learning difficulties or psychological problems. The UK Care Quality Commission, a non-departmental body of the Department of Health and Social Care, found that 34 percent of those working in health and social care were pressured into placing ‘do not attempt cardiopulmonary resuscitation’ orders on ‘Covid’ patients who suffered from disabilities and learning difficulties without involving the patient or their families in the decision. UK judges ruled that an elderly woman with dementia should have the DNA-manipulating ‘Covid vaccine’ against her son’s wishes and that a man with severe learning difficulties should have the jab despite his family’s objections. Never mind that many had already died. The judiciary always supports doctors and government in fascist dictatorships. They wouldn’t dare do otherwise. A horrific video was posted showing fascist officers from Los Angeles police forcibly giving the ‘Covid’ shot to women with special needs who were screaming that they didn’t want it. The same fascists are seen giving the jab to a sleeping elderly woman in a care home. This is straight out of the Nazi playbook. Hitler’s Nazis committed mass murder of the mentally ill and physically disabled throughout Germany and occupied territories in the programme that became known as Aktion T4, or just T4. Sabbatian-controlled Hitler and his grotesque crazies set out to kill those they considered useless and unnecessary. The Reich Committee for the Scientific Registering of Hereditary and Congenital Illnesses registered the births of babies identified by physicians to have ‘defects’. By 1941 alone more than 5,000 children were murdered by the state and it is estimated that in total the number of innocent people killed in Aktion T4 was between 275,000 and 300,000. Parents were told their children had been sent away for ‘special treatment’ never to return. It is rather pathetic to see claims about plans for new extermination camps being dismissed today

when the same force behind current events did precisely that 80 years ago. Margaret Sanger was a Cult operative who used 'birth control' to sanitise her programme of eugenics. Organisations she founded became what is now Planned Parenthood. Sanger proposed that 'the whole dysgenic population would have its choice of segregation or sterilization'. These included epileptics, 'feeble-minded', and prostitutes. Sanger opposed charity because it perpetuated 'human waste'. She reveals the Cult mentality and if anyone thinks that extermination camps are a 'conspiracy theory' their naivety is touching if breathtakingly stupid.

If you don't believe that doctors can act with callous disregard for their patients it is worth considering that doctors and medical staff agreed to put government-decreed DNR orders on medical files and do nothing when resuscitation is called for. I don't know what you call such people in your house. In mine they are Nazis from the Josef Mengele School of Medicine. Phenomenal numbers of old people have died worldwide from the effects of lockdown, depression, lack of treatment, the 'vaccine' (more later) and losing the will to live. A common response at the start of the manufactured pandemic was to remove old people from hospital beds and transfer them to nursing homes. The decision would result in a mass cull of elderly people in those homes through lack of treatment – *not* 'Covid'. Care home whistleblowers have told how once the 'Covid' era began doctors would not come to their homes to treat patients and they were begging for drugs like antibiotics that often never came. The most infamous example was ordered by New York governor Andrew Cuomo, brother of a moronic CNN host, who amazingly was given an Emmy Award for his handling of the 'Covid crisis' by the ridiculous Wokers that hand them out. Just how ridiculous could be seen in February, 2021, when a Department of Justice and FBI investigation began into how thousands of old people in New York died in nursing homes after being discharged from hospital to make way for 'Covid' patients on Cuomo's say-so – and how he and his staff covered up these facts. This couldn't have happened to a nicer psychopath. Even then there was a 'Covid' spin. Reports said that

thousands of old people who tested positive for ‘Covid’ in hospital were transferred to nursing homes to both die of ‘Covid’ and transmit it to others. No – they were in hospital because they were ill and the fact that they tested positive with a test not testing for the ‘virus’ is irrelevant. They were ill often with respiratory diseases ubiquitous in old people near the end of their lives. Their transfer out of hospital meant that their treatment stopped and many would go on to die.

They're old. Who gives a damn?

I have exposed in the books for decades the Cult plan to cull the world’s old people and even to introduce at some point what they call a ‘demise pill’ which at a certain age everyone would take and be out of here by law. In March, 2021, Spain legalised euthanasia and assisted suicide following the Netherlands, Belgium, Luxembourg and Canada on the Tiptoe to the demise pill. Treatment of old people by many ‘care’ homes has been a disgrace in the ‘Covid’ era. There are many, many, caring staff – I know some. There have, however, been legions of stories about callous treatment of old people and their families. Police were called when families came to take their loved ones home in the light of isolation that was killing them. They became prisoners of the state. Care home residents in insane, fascist Ontario, Canada, were not allowed to leave their *room* once the ‘Covid’ hoax began. UK staff have even wheeled elderly people away from windows where family members were talking with them. Oriana Criscuolo from Stockport in the English North West dropped off some things for her 80-year-old father who has Parkinson’s disease and dementia and she wanted to wave to him through a ground-floor window. She was told that was ‘illegal’. When she went anyway they closed the curtains in the middle of the day. Oriana said:

It’s just unbelievable. I cannot understand how care home staff – people who are being paid to care – have become so uncaring. Their behaviour is inhumane and cruel. It’s beyond belief.

She was right and this was not a one-off. What a way to end your life in such loveless circumstances. UK registered nurse Nicky Millen, a proper old school nurse for 40 years, said that when she started her career care was based on dignity, choice, compassion and empathy. Now she said ‘the things that are important to me have gone out of the window.’ She was appalled that people were dying without their loved ones and saying goodbye on iPads. Nicky described how a distressed 89-year-old lady stroked her face and asked her ‘how many paracetamol would it take to finish me off’. Life was no longer worth living while not seeing her family. Nicky said she was humiliated in front of the ward staff and patients for letting the lady stroke her face and giving her a cuddle. Such is the dehumanisation that the ‘Covid’ hoax has brought to the surface. Nicky worked in care homes where patients told her they were being held prisoner. ‘I want to live until I die’, one said to her. ‘I had a lady in tears because she hadn’t seen her great-grandson.’ Nicky was compassionate old school meeting psychopathic New Normal. She also said she had worked on a ‘Covid’ ward with no ‘Covid’ patients. Jewish writer Shai Held wrote an article in March, 2020, which was headlined ‘The Staggering, Heartless Cruelty Toward the Elderly’. What he described was happening from the earliest days of lockdown. He said ‘the elderly’ were considered a group and not unique individuals (the way of the Woke). Shai Held said:

Notice how the all-too-familiar rhetoric of dehumanization works: ‘The elderly’ are bunched together as a faceless mass, all of them considered culprits and thus effectively deserving of the suffering the pandemic will inflict upon them. Lost entirely is the fact that the elderly are individual human beings, each with a distinctive face and voice, each with hopes and dreams, memories and regrets, friendships and marriages, loves lost and loves sustained.

‘The elderly’ have become another dehumanised group for which anything goes and for many that has resulted in cold disregard for their rights and their life. The distinctive face that Held talks about is designed to be deleted by masks until everyone is part of a faceless mass.

'War-zone' hospitals myth

Again and again medical professionals have told me what was really going on and how hospitals 'overrun like war zones' according to the media were virtually empty. The mantra from medical whistleblowers was please don't use my name or my career is over. Citizen journalists around the world sneaked into hospitals to film evidence exposing the 'war-zone' lie. They really *were* largely empty with closed wards and operating theatres. I met a hospital worker in my town on the Isle of Wight during the first lockdown in 2020 who said the only island hospital had never been so quiet. Lockdown was justified by the psychopaths to stop hospitals being overrun. At the same time that the island hospital was near-empty the military arrived here to provide *extra beds*. It was all propaganda to ramp up the fear to ensure compliance with fascism as were never-used temporary hospitals with thousands of beds known as Nightingales and never-used make-shift mortuaries opened by the criminal UK government. A man who helped to install those extra island beds attributed to the army said they were never used and the hospital was empty. Doctors and nurses 'stood around talking or on their phones, wandering down to us to see what we were doing'. There were no masks or social distancing. He accused the useless local island paper, the *County Press*, of 'pumping the fear as if our hospital was overrun and we only have one so it should have been'. He described ambulances parked up with crews outside in deck chairs. When his brother called an ambulance he was told there was a two-hour backlog which he called 'bullshit'. An old lady on the island fell 'and was in a bad way', but a caller who rang for an ambulance was told the situation wasn't urgent enough. Ambulance stations were working under capacity while people would hear ambulances with sirens blaring driving through the streets. When those living near the stations realised what was going on they would follow them as they left, circulated around an urban area with the sirens going, and then came back without stopping. All this was to increase levels of fear and the same goes for the 'ventilator shortage crisis' that cost tens of millions for hastily produced ventilators never to be used.

Ambulance crews that agreed to be exploited in this way for fear propaganda might find themselves a mirror. I wish them well with that. Empty hospitals were the obvious consequence of treatment and diagnoses of non-'Covid' conditions cancelled and those involved handed a death sentence. People have been dying at home from undiagnosed and untreated cancer, heart disease and other life-threatening conditions to allow empty hospitals to deal with a 'pandemic' that wasn't happening.

Death of the innocent

'War-zones' have been laying off nursing staff, even doctors where they can. There was no work for them. Lockdown was justified by saving lives and protecting the vulnerable they were actually killing with DNR orders and preventing empty hospitals being 'overrun'. In Britain the mantra of stay at home to 'save the NHS' was everywhere and across the world the same story was being sold when it was all lies. Two California doctors, Dan Erickson and Artin Massihi at Accelerated Urgent Care in Bakersfield, held a news conference in April, 2020, to say that intensive care units in California were 'empty, essentially', with hospitals shutting floors, not treating patients and laying off doctors. The California health system was working at minimum capacity 'getting rid of doctors because we just don't have the volume'. They said that people with conditions such as heart disease and cancer were not coming to hospital out of fear of 'Covid-19'. Their video was deleted by Susan Wojcicki's Cult-owned YouTube after reaching five million views. Florida governor Ron Desantis, who rejected the severe lockdowns of other states and is being targeted for doing so, said that in March, 2020, every US governor was given models claiming they would run out of hospital beds in days. That was never going to happen and the 'modellers' knew it. Deceit can be found at every level of the system. Urgent children's operations were cancelled including fracture repairs and biopsies to spot cancer. Eric Nicholls, a consultant paediatrician, said 'this is obviously concerning and we need to return to normal operating and to increase capacity as soon as possible'. Psychopaths

in power were rather less concerned *because* they are psychopaths. Deletion of urgent care and diagnosis has been happening all over the world and how many kids and others have died as a result of the actions of these cold and heartless lunatics dictating ‘health’ policy? The number must be stratospheric. Richard Sullivan, professor of cancer and global health at King’s College London, said people feared ‘Covid’ more than cancer such was the campaign of fear. ‘Years of lost life will be quite dramatic’, Sullivan said, with ‘a huge amount of avoidable mortality’. Sarah Woolnough, executive director for policy at Cancer Research UK, said there had been a 75 percent drop in urgent referrals to hospitals by family doctors of people with suspected cancer. Sullivan said that ‘a lot of services have had to scale back – we’ve seen a dramatic decrease in the amount of elective cancer surgery’. Lockdown deaths worldwide has been absolutely fantastic with the *New York Post* reporting how data confirmed that ‘lockdowns end more lives than they save’:

There was a sharp decline in visits to emergency rooms and an increase in fatal heart attacks because patients didn’t receive prompt treatment. Many fewer people were screened for cancer. Social isolation contributed to excess deaths from dementia and Alzheimer’s.

Researchers predicted that the social and economic upheaval would lead to tens of thousands of “deaths of despair” from drug overdoses, alcoholism and suicide. As unemployment surged and mental-health and substance-abuse treatment programs were interrupted, the reported levels of anxiety, depression and suicidal thoughts increased dramatically, as did alcohol sales and fatal drug overdoses.

This has been happening while nurses and other staff had so much time on their hands in the ‘war-zones’ that Tic-Tok dancing videos began appearing across the Internet with medical staff dancing around in empty wards and corridors as people died at home from causes that would normally have been treated in hospital.

Mentions in dispatches

One brave and truth-committed whistleblower was Louise Hampton, a call handler with the UK NHS who made a viral Internet video saying she had done ‘fuck all’ during the ‘pandemic’

which was ‘a load of bollocks’. She said that ‘Covid-19’ was rebranded flu and of course she lost her job. This is what happens in the medical and endless other professions now when you tell the truth. Louise filmed inside ‘war-zone’ accident and emergency departments to show they were empty and I mean *empty* as in no one there. The mainstream media could have done the same and blown the gaff on the whole conspiracy. They haven’t to their eternal shame. Not that most ‘journalists’ seem capable of manifesting shame as with the psychopaths they slavishly repeat without question. The relative few who were admitted with serious health problems were left to die alone with no loved ones allowed to see them because of ‘Covid’ rules and they included kids dying without the comfort of mum and dad at their bedside while the evil behind this couldn’t give a damn. It was all good fun to them. A Scottish NHS staff nurse publicly quit in the spring of 2021 saying: ‘I can no longer be part of the lies and the corruption by the government.’ She said hospitals ‘aren’t full, the beds aren’t full, beds have been shut, wards have been shut’. Hospitals were never busy throughout ‘Covid’. The staff nurse said that Nicola Sturgeon, tragically the leader of the Scottish government, was on television saying save the hospitals and the NHS – ‘but the beds are empty’ and ‘we’ve not seen flu, we always see flu every year’. She wrote to government and spoke with her union Unison (the unions are Cult-compromised and *useless*, but nothing changed. Many of her colleagues were scared of losing their jobs if they spoke out as they wanted to. She said nursing staff were being affected by wearing masks all day and ‘my head is splitting every shift from wearing a mask’. The NHS is part of the fascist tyranny and must be dismantled so we can start again with human beings in charge. (Ironically, hospitals were reported to be busier again when official ‘Covid’ cases *fell* in spring/summer of 2021 and many other conditions required treatment at the same time as *the fake vaccine rollout*.)

I will cover the ‘Covid vaccine’ scam in detail later, but it is another indicator of the sickening disregard for human life that I am highlighting here. The DNA-manipulating concoctions do not fulfil

the definition of a ‘vaccine’, have never been used on humans before and were given only emergency approval because trials were not completed and they continued using the unknowing public. The result was what a NHS senior nurse with responsibility for ‘vaccine’ procedure said was ‘genocide’. She said the ‘vaccines’ were not ‘vaccines’. They had not been shown to be safe and claims about their effectiveness by drug companies were ‘poetic licence’. She described what was happening as a ‘horrid act of human annihilation’. The nurse said that management had instigated a policy of not providing a Patient Information Leaflet (PIL) before people were ‘vaccinated’ even though health care professionals are supposed to do this according to protocol. Patients should also be told that they are taking part in an ongoing clinical trial. Her challenges to what is happening had seen her excluded from meetings and ridiculed in others. She said she was told to ‘watch my step … or I would find myself surplus to requirements’. The nurse, who spoke anonymously in fear of her career, said she asked her NHS manager why he/she was content with taking part in genocide against those having the ‘vaccines’. The reply was that everyone had to play their part and to ‘put up, shut up, and get it done’. Government was ‘leaning heavily’ on NHS management which was clearly leaning heavily on staff. This is how the global ‘medical’ hierarchy operates and it starts with the Cult and its World Health Organization.

She told the story of a doctor who had the Pfizer jab and when questioned had no idea what was in it. The doctor had never read the literature. We have to stop treating doctors as intellectual giants when so many are moral and medical pygmies. The doctor did not even know that the ‘vaccines’ were not fully approved or that their trials were ongoing. They were, however, asking their patients if they minded taking part in follow-ups for research purposes – yes, the *ongoing clinical trial*. The nurse said the doctor’s ignorance was not rare and she had spoken to a hospital consultant who had the jab without any idea of the background or that the ‘trials’ had not been completed. Nurses and pharmacists had shown the same ignorance.

'My NHS colleagues have forsaken their duty of care, broken their code of conduct – Hippocratic Oath – and have been brainwashed just the same as the majority of the UK public through propaganda ...' She said she had not been able to recruit a single NHS colleague, doctor, nurse or pharmacist to stand with her and speak out. Her union had refused to help. She said that if the genocide came to light she would not hesitate to give evidence at a Nuremberg-type trial against those in power who could have affected the outcomes but didn't.

And all for what?

To put the nonsense into perspective let's say the 'virus' does exist and let's go completely crazy and accept that the official manipulated figures for cases and deaths are accurate. *Even then* a study by Stanford University epidemiologist Dr John Ioannidis published on the World Health Organization website produced an average infection to fatality rate of ... 0.23 percent! Ioannidis said: 'If one could sample equally from all locations globally, the median infection fatality rate might even be substantially lower than the 0.23% observed in my analysis.' For healthy people under 70 it was ... 0.05 percent! This compares with the 3.4 percent claimed by the Cult-owned World Health Organization when the hoax was first played and maximum fear needed to be generated. An updated Stanford study in April, 2021, put the 'infection' to 'fatality' rate at just 0.15 percent. Another team of scientists led by Megan O'Driscoll and Henrik Salje studied data from 45 countries and published their findings on the Nature website. For children and young people the figure is so small it virtually does not register although authorities will be hyping dangers to the young when they introduce DNA-manipulating 'vaccines' for children. The O'Driscoll study produced an average infection-fatality figure of 0.003 for children from birth to four; 0.001 for 5 to 14; 0.003 for 15 to 19; and it was still only 0.456 up to 64. To claim that children must be 'vaccinated' to protect them from 'Covid' is an obvious lie and so there must be another reason and there is. What's more the average age of a 'Covid' death is akin

to the average age that people die in general. The average age of death in England is about 80 for men and 83 for women. The average age of death from alleged 'Covid' is between 82 and 83. California doctors, Dan Erickson and Artin Massihi, said at their April media conference that projection models of millions of deaths had been 'woefully inaccurate'. They produced detailed figures showing that Californians had a 0.03 chance of dying from 'Covid' based on the number of people who tested positive (with a test not testing for the 'virus'). Erickson said there was a 0.1 percent chance of dying from 'Covid' in the *state* of New York, not just the city, and a 0.05 percent chance in Spain, a centre of 'Covid-19' hysteria at one stage. The Stanford studies supported the doctors' data with fatality rate estimates of 0.23 and 0.15 percent. How close are these figures to my estimate of *zero*? Death-rate figures claimed by the World Health Organization at the start of the hoax were some 15 times higher. The California doctors said there was no justification for lockdowns and the economic devastation they caused. Everything they had ever learned about quarantine was that you quarantine the *sick* and not the healthy. They had never seen this before and it made no medical sense.

Why in the light of all this would governments and medical systems the world over say that billions must go under house arrest; lose their livelihood; in many cases lose their mind, their health and their life; force people to wear masks dangerous to health and psychology; make human interaction and even family interaction a criminal offence; ban travel; close restaurants, bars, watching live sport, concerts, theatre, and any activity involving human togetherness and discourse; and closing schools to isolate children from their friends and cause many to commit suicide in acts of hopelessness and despair? The California doctors said lockdown consequences included increased child abuse, partner abuse, alcoholism, depression, and other impacts they were seeing every day. Who would do that to the entire human race if not mentally-ill psychopaths of almost unimaginable extremes like Bill Gates? We must face the reality of what we are dealing with and come out of

denial. Fascism and tyranny are made possible only by the target population submitting and acquiescing to fascism and tyranny. The whole of human history shows that to be true. Most people naively and unquestioning believed what they were told about a ‘deadly virus’ and meekly and weakly submitted to house arrest. Those who didn’t believe it – at least in total – still submitted in fear of the consequences of not doing so. For the rest who wouldn’t submit draconian fines have been imposed, brutal policing by psychopaths *for* psychopaths, and condemnation from the meek and weak who condemn the Pushbackers on behalf of the very force that has them, too, in its gunsights. ‘Pathetic’ does not even begin to suffice.

Britain’s brainless ‘Health’ Secretary Matt Hancock warned anyone lying to border officials about returning from a list of ‘hotspot’ countries could face a jail sentence of up to ten years which is more than for racially-aggravated assault, incest and attempting to have sex with a child under 13. Hancock is a lunatic, but he has the state apparatus behind him in a Cult-led chain reaction and the same with UK ‘Vaccine Minister’ Nadhim Zahawi, a prominent member of the mega-Cult secret society, Le Cercle, which featured in my earlier books. The Cult enforces its will on governments and medical systems; government and medical systems enforce their will on business and police; business enforces its will on staff who enforce it on customers; police enforce the will of the Cult on the population and play their essential part in creating a world of fascist control that their own children and grandchildren will have to live in their entire lives. It is a hierarchical pyramid of imposition and acquiescence and, yes indeedy, of clinical insanity.

Does anyone bright enough to read this book have to ask what the answer is? I think not, but I will reveal it anyway in the fewest of syllables: Tell the psychos and their moronic lackeys to fuck off and let’s get on with our lives. We are many – They are few.

CHAPTER SEVEN

War on your mind

One believes things because one has been conditioned to believe them

Aldous Huxley, *Brave New World*

I have described the ‘Covid’ hoax as a ‘Psyop’ and that is true in every sense and on every level in accordance with the definition of that term which is psychological warfare. Break down the ‘Covid pandemic’ to the foundation themes and it is psychological warfare on the human individual and collective mind.

The same can be said for the entire human belief system involving every subject you can imagine. Huxley was right in his contention that people believe what they are conditioned to believe and this comes from the repetition throughout their lives of the same falsehoods. They spew from government, corporations, media and endless streams of ‘experts’ telling you what the Cult wants you to believe and often believing it themselves (although *far* from always). ‘Experts’ are rewarded with ‘prestigious’ jobs and titles and as agents of perceptual programming with regular access to the media. The Cult has to control the narrative – control *information* – or they lose control of the vital, crucial, without-which-they-cannot-prevail public perception of reality. The foundation of that control today is the Internet made possible by the Defense Advanced Research Projects Agency (DARPA), the incredibly sinister technological arm of the Pentagon. The Internet is the result of military technology.

DARPA openly brags about establishing the Internet which has been a long-term project to lasso the minds of the global population. I have said for decades the plan is to control information to such an extreme that eventually no one would see or hear anything that the Cult does not approve. We are closing in on that end with ferocious censorship since the ‘Covid’ hoax began and in my case it started back in the 1990s in terms of books and speaking venues. I had to create my own publishing company in 1995 precisely because no one else would publish my books even then. I think they’re all still running.

Cult Internet

To secure total control of information they needed the Internet in which pre-programmed algorithms can seek out ‘unclean’ content for deletion and even stop it being posted in the first place. The Cult had to dismantle print and non-Internet broadcast media to ensure the transfer of information to the appropriate-named ‘Web’ – a critical expression of the *Cult* web. We’ve seen the ever-quickenning demise of traditional media and control of what is left by a tiny number of corporations operating worldwide. Independent journalism in the mainstream is already dead and never was that more obvious than since the turn of 2020. The Cult wants all information communicated via the Internet to globally censor and allow the plug to be pulled any time. Lockdowns and forced isolation has meant that communication between people has been through electronic means and no longer through face-to-face discourse and discussion. Cult psychopaths have targeted the bars, restaurants, sport, venues and meeting places in general for this reason. None of this is by chance and it’s to stop people gathering in any kind of privacy or number while being able to track and monitor all Internet communications and block them as necessary. Even private messages between individuals have been censored by these fascists that control Cult fronts like Facebook, Twitter, Google and YouTube which are all officially run by Sabbatian place-people and from the background by higher-level Sabbatian place people.

Facebook, Google, Amazon and their like were seed-funded and supported into existence with money-no-object infusions of funds either directly or indirectly from DARPA and CIA technology arm In-Q-Tel. The Cult plays the long game and prepares very carefully for big plays like 'Covid'. Amazon is another front in the psychological war and pretty much controls the global market in book sales and increasingly publishing. Amazon's limitless funds have deleted fantastic numbers of independent publishers to seize global domination on the way to deciding which books can be sold and circulated and which cannot. Moves in that direction are already happening. Amazon's leading light Jeff Bezos is the grandson of Lawrence Preston Gise who worked with DARPA predecessor ARPA. Amazon has big connections to the CIA and the Pentagon. The plan I have long described went like this:

1. Employ military technology to establish the Internet.
2. Sell the Internet as a place where people can freely communicate without censorship and allow that to happen until the Net becomes the central and irreversible pillar of human society. If the Internet had been highly censored from the start many would have rejected it.
3. Fund and manipulate major corporations into being to control the circulation of information on your Internet using cover stories about geeks in garages to explain how they came about. Give them unlimited funds to expand rapidly with no need to make a profit for years while non-Cult companies who need to balance the books cannot compete. You know that in these circumstances your Googles, YouTubes, Facebooks and Amazons are going to secure near monopolies by either crushing or buying up the opposition.
4. Allow freedom of expression on both the Internet and communication platforms to draw people in until the Internet is the central and irreversible pillar of human society and your communication corporations have reached a stage of near monopoly domination.
5. Then unleash your always-planned frenzy of censorship on the basis of 'where else are you going to go?' and continue to expand that until nothing remains that the Cult does not want its human targets to see.

The process was timed to hit the 'Covid' hoax to ensure the best chance possible of controlling the narrative which they knew they had to do at all costs. They were, after all, about to unleash a 'deadly virus' that didn't really exist. If you do that in an environment of free-flowing information and opinion you would be dead in the

water before you could say Gates is a psychopath. The network was in place through which the Cult-created-and-owned World Health Organization could dictate the ‘Covid’ narrative and response policy slavishly supported by Cult-owned Internet communication giants and mainstream media while those telling a different story were censored. Google, YouTube, Facebook and Twitter openly announced that they would do this. What else would we expect from Cult-owned operations like Facebook which former executives have confirmed set out to make the platform more addictive than cigarettes and coldly manipulates emotions of its users to sow division between people and groups and scramble the minds of the young? If Zuckerberg lives out the rest of his life without going to jail for crimes against humanity, and most emphatically against the young, it will be a travesty of justice. Still, no matter, cause and effect will catch up with him eventually and the same with Sergey Brin and Larry Page at Google with its CEO Sundar Pichai who fix the Google search results to promote Cult narratives and hide the opposition. Put the same key words into Google and other search engines like DuckDuckGo and you will see how different results can be. Wikipedia is another intensely biased ‘encyclopaedia’ which skews its content to the Cult agenda. YouTube links to Wikipedia’s version of ‘Covid’ and ‘climate change’ on video pages in which experts in their field offer a different opinion (even that is increasingly rare with Wojcicki censorship). Into this ‘Covid’ silence-them network must be added government media censors, sorry ‘regulators’, such as Ofcom in the UK which imposed tyrannical restrictions on British broadcasters that had the effect of banning me from ever appearing. Just to debate with me about my evidence and views on ‘Covid’ would mean breaking the fascistic impositions of Ofcom and its CEO career government bureaucrat Melanie Dawes. Gutless British broadcasters tremble at the very thought of fascist Ofcom.

Psychos behind ‘Covid’

The reason for the ‘Covid’ catastrophe in all its facets and forms can be seen by whom and what is driving the policies worldwide in such a coordinated way. Decisions are not being made to protect health, but to target psychology. The dominant group guiding and ‘advising’ government policy are not medical professionals. They are psychologists and behavioural scientists. Every major country has its own version of this phenomenon and I’ll use the British example to show how it works. In many ways the British version has been affecting the wider world in the form of the huge behaviour manipulation network in the UK which operates in other countries. The network involves private companies, government, intelligence and military. The Cabinet Office is at the centre of the government ‘Covid’ Psyop and part-owns, with ‘innovation charity’ Nesta, the Behavioural Insights Team (BIT) which claims to be independent of government but patently isn’t. The BIT was established in 2010 and its job is to manipulate the psyche of the population to acquiesce to government demands and so much more. It is also known as the ‘Nudge Unit’, a name inspired by the 2009 book by two ultra-Zionists, Cass Sunstein and Richard Thaler, called *Nudge: Improving Decisions About Health, Wealth, and Happiness*. The book, as with the Behavioural Insights Team, seeks to ‘nudge’ behaviour (manipulate it) to make the public follow patterns of action and perception that suit those in authority (the Cult). Sunstein is so skilled at this that he advises the World Health Organization and the UK Behavioural Insights Team and was Administrator of the White House Office of Information and Regulatory Affairs in the Obama administration. Biden appointed him to the Department of Homeland Security – another ultra-Zionist in the fold to oversee new immigration laws which is another policy the Cult wants to control. Sunstein is desperate to silence anyone exposing conspiracies and co-authored a 2008 report on the subject in which suggestions were offered to ban ‘conspiracy theorizing’ or impose ‘some kind of tax, financial or otherwise, on those who disseminate such theories’. I guess a psychiatrist’s chair is out of the question?

Sunstein's mate Richard Thaler, an 'academic affiliate' of the UK Behavioural Insights Team, is a proponent of 'behavioural economics' which is defined as the study of 'the effects of psychological, cognitive, emotional, cultural and social factors on the decisions of individuals and institutions'. Study the effects so they can be manipulated to be what you want them to be. Other leading names in the development of behavioural economics are ultra-Zionists Daniel Kahneman and Robert J. Shiller and they, with Thaler, won the Nobel Memorial Prize in Economic Sciences for their work in this field. The Behavioural Insights Team is operating at the heart of the UK government and has expanded globally through partnerships with several universities including Harvard, Oxford, Cambridge, University College London (UCL) and Pennsylvania. They claim to have 'trained' (reframed) 20,000 civil servants and run more than 750 projects involving 400 randomised controlled trials in dozens of countries' as another version of mind reframers Common Purpose. BIT works from its office in New York with cities and their agencies, as well as other partners, across the United States and Canada – this is a company part-owned by the British government Cabinet Office. An executive order by President Cult-servant Obama established a US Social and Behavioral Sciences Team in 2015. They all have the same reason for being and that's to brainwash the population directly and by brainwashing those in positions of authority.

'Covid' mind game

Another prime aspect of the UK mind-control network is the 'independent' [joke] Scientific Pandemic Insights Group on Behaviours (SPI-B) which 'provides behavioural science advice aimed at anticipating and helping people adhere to interventions that are recommended by medical or epidemiological experts'. That means manipulating public perception and behaviour to do whatever government tells them to do. It's disgusting and if they really want the public to be 'safe' this lot should all be under lock and key. According to the government website SPI-B consists of

'behavioural scientists, health and social psychologists, anthropologists and historians' and advises the Whitty-Vallance-led Scientific Advisory Group for Emergencies (SAGE) which in turn advises the government on 'the science' (it doesn't) and 'Covid' policy. When politicians say they are being guided by 'the science' this is the rabble in each country they are talking about and that 'science' is dominated by behaviour manipulators to enforce government fascism through public compliance. The Behaviour Insight Team is headed by psychologist David Solomon Halpern, a visiting professor at King's College London, and connects with a national and global web of other civilian and military organisations as the Cult moves towards its goal of fusing them into one fascistic whole in every country through its 'Fusion Doctrine'. The behaviour manipulation network involves, but is not confined to, the Foreign Office; National Security Council; government communications headquarters (GCHQ); MI5; MI6; the Cabinet Office-based Media Monitoring Unit; and the Rapid Response Unit which 'monitors digital trends to spot emerging issues; including misinformation and disinformation; and identifies the best way to respond'.

There is also the 77th Brigade of the UK military which operates like the notorious Israeli military's Unit 8200 in manipulating information and discussion on the Internet by posing as members of the public to promote the narrative and discredit those who challenge it. Here we have the military seeking to manipulate *domestic* public opinion while the Nazis in government are fine with that. Conservative Member of Parliament Tobias Ellwood, an advocate of lockdown and control through 'vaccine passports', is a Lieutenant Colonel reservist in the 77th Brigade which connects with the military operation jHub, the 'innovation centre' for the Ministry of Defence and Strategic Command. jHub has also been involved with the civilian National Health Service (NHS) in 'symptom tracing' the population. The NHS is a key part of this mind control network and produced a document in December, 2020, explaining to staff how to use psychological manipulation with different groups and ages to get them to have the DNA-manipulating 'Covid vaccine'

that's designed to cumulatively rewrite human genetics. The document, called 'Optimising Vaccination Roll Out – Do's and Dont's for all messaging, documents and "communications" in the widest sense', was published by NHS England and the NHS Improvement *Behaviour Change Unit* in partnership with Public Health England and Warwick Business School. I hear the mantra about 'save the NHS' and 'protect the NHS' when we need to scrap the NHS and start again. The current version is far too corrupt, far too anti-human and totally compromised by Cult operatives and their assets. UK government broadcast media censor Ofcom will connect into this web – as will the BBC with its tremendous Ofcom influence – to control what the public see and hear and dictate mass perception. Nuremberg trials must include personnel from all these organisations.

The fear factor

The 'Covid' hoax has led to the creation of the UK Cabinet Office-connected Joint Biosecurity Centre (JBC) which is officially described as providing 'expert advice on pandemics' using its independent [all Cult operations are 'independent'] analytical function to provide real-time analysis about infection outbreaks to identify and respond to outbreaks of Covid-19'. Another role is to advise the government on a response to spikes in infections – 'for example by closing schools or workplaces in local areas where infection levels have risen'. Put another way, promoting the Cult agenda. The Joint Biosecurity Centre is modelled on the Joint Terrorism Analysis Centre which analyses intelligence to set 'terrorism threat levels' and here again you see the fusion of civilian and military operations and intelligence that has led to military intelligence producing documents about 'vaccine hesitancy' and how it can be combated. Domestic civilian matters and opinions should not be the business of the military. The Joint Biosecurity Centre is headed by Tom Hurd, director general of the Office for Security and Counter-Terrorism from the establishment-to-its-fingertips Hurd family. His father is former Foreign Secretary Douglas Hurd. How coincidental that Tom

Hurd went to the elite Eton College and Oxford University with Boris Johnson. Imperial College with its ridiculous computer modeller Neil Ferguson will connect with this gigantic web that will itself interconnect with similar set-ups in other major and not so major countries. Compared with this Cult network the politicians, be they Boris Johnson, Donald Trump or Joe Biden, are bit-part players ‘following the science’. The network of psychologists was on the ‘Covid’ case from the start with the aim of generating maximum fear of the ‘virus’ to ensure compliance by the population. A government behavioural science group known as SPI-B produced a paper in March, 2020, for discussion by the main government science advisory group known as SAGE. It was headed ‘Options for increasing adherence to social distancing measures’ and it said the following in a section headed ‘Persuasion’:

- A substantial number of people still do not feel sufficiently personally threatened; it could be that they are reassured by the low death rate in their demographic group, although levels of concern may be rising. Having a good understanding of the risk has been found to be positively associated with adoption of COVID-19 social distancing measures in Hong Kong.
- The perceived level of personal threat needs to be increased among those who are complacent, using hard-hitting evaluation of options for increasing social distancing emotional messaging. To be effective this must also empower people by making clear the actions they can take to reduce the threat.
- Responsibility to others: There seems to be insufficient understanding of, or feelings of responsibility about, people’s role in transmitting the infection to others ... Messaging about actions need to be framed positively in terms of protecting oneself and the community, and increase confidence that they will be effective.
- Some people will be more persuaded by appeals to play by the rules, some by duty to the community, and some to personal risk.

All these different approaches are needed. The messaging also needs to take account of the realities of different people's lives. Messaging needs to take account of the different motivational levers and circumstances of different people.

All this could be achieved the SPI-B psychologists said by *using the media to increase the sense of personal threat* which translates as terrify the shit out of the population, including children, so they all do what we want. That's not happened has it? Those excuses for 'journalists' who wouldn't know journalism if it bit them on the arse (the great majority) have played their crucial part in serving this Cult-government Psyop to enslave their own kids and grandkids. How they live with themselves I have no idea. The psychological war has been underpinned by constant government 'Covid' propaganda in almost every television and radio ad break, plus the Internet and print media, which has pounded out the fear with taxpayers footing the bill for their own programming. The result has been people terrified of a 'virus' that doesn't exist or one with a tiny fatality rate even if you believe it does. People walk down the street and around the shops wearing face-nappies damaging their health and psychology while others report those who refuse to be that naïve to the police who turn up in their own face-nappies. I had a cameraman come to my flat and he was so frightened of 'Covid' he came in wearing a mask and refused to shake my hand in case he caught something. He had – naïveitis – and the thought that he worked in the mainstream media was both depressing and made his behaviour perfectly explainable. The fear which has gripped the minds of so many and frozen them into compliance has been carefully cultivated by these psychologists who are really psychopaths. If lives get destroyed and a lot of young people commit suicide it shows our plan is working. SPI-B then turned to compulsion on the public to comply. 'With adequate preparation, rapid change can be achieved', it said. Some countries had introduced mandatory self-isolation on a wide scale without evidence of major public unrest and a large majority of the UK's population appeared to be supportive of more coercive measures with 64 percent of adults saying they would

support putting London under a lockdown (watch the ‘polls’ which are designed to make people believe that public opinion is in favour or against whatever the subject in hand).

For ‘aggressive protective measures’ to be effective, the SPI-B paper said, special attention should be devoted to those population groups that are more at risk. Translated from the Orwellian this means making the rest of population feel guilty for not protecting the ‘vulnerable’ such as old people which the Cult and its agencies were about to kill on an industrial scale with lockdown, lack of treatment and the Gates ‘vaccine’. Psychopath psychologists sold their guilt-trip so comprehensively that Los Angeles County Supervisor Hilda Solis reported that children were apologising (from a distance) to their parents and grandparents for bringing ‘Covid’ into their homes and getting them sick. ‘... These apologies are just some of the last words that loved ones will ever hear as they die alone,’ she said. Gut-wrenchingly Solis then used this childhood tragedy to tell children to stay at home and ‘keep your loved ones alive’. Imagine heaping such potentially life-long guilt on a kid when it has absolutely nothing to do with them. These people are deeply disturbed and the psychologists behind this even more so.

Uncivil war – divide and rule

Professional mind-controllers at SPI-B wanted the media to increase a sense of responsibility to others (do as you’re told) and promote ‘positive messaging’ for those actions while in contrast to invoke ‘social disapproval’ by the unquestioning, obedient, community of anyone with a mind of their own. Again the compliant Goebbels-like media obliged. This is an old, old, trick employed by tyrannies the world over throughout human history. You get the target population to keep the target population in line – *your* line. SPI-B said this could ‘play an important role in preventing anti-social behaviour or discouraging failure to enact pro-social behaviour’. For ‘anti-social’ in the Orwellian parlance of SPI-B see any behaviour that government doesn’t approve. SPI-B recommendations said that ‘social disapproval’ should be accompanied by clear messaging and

promotion of strong collective identity – hence the government and celebrity mantra of ‘we’re all in this together’. Sure we are. The mind doctors have such contempt for their targets that they think some clueless comedian, actor or singer telling them to do what the government wants will be enough to win them over. We have had UK comedian Lenny Henry, actor Michael Caine and singer Elton John wheeled out to serve the propagandists by urging people to have the DNA-manipulating ‘Covid’ non-‘vaccine’. The role of Henry and fellow black celebrities in seeking to coax a ‘vaccine’ reluctant black community into doing the government’s will was especially stomach-turning. An emotion-manipulating script and carefully edited video featuring these black ‘celebs’ was such an insult to the intelligence of black people and where’s the self-respect of those involved selling their souls to a fascist government agenda? Henry said he heard black people’s ‘legitimate worries and concerns’, but people must ‘trust the facts’ when they were doing exactly that by not having the ‘vaccine’. They had to include the obligatory reference to Black Lives Matter with the line ... ‘Don’t let coronavirus cost even more black lives – because we matter’. My god, it was pathetic. ‘I know the vaccine is safe and what it does.’ How? ‘I’m a comedian and it says so in my script.’

SPI-B said social disapproval needed to be carefully managed to avoid victimisation, scapegoating and misdirected criticism, but they knew that their ‘recommendations’ would lead to exactly that and the media were specifically used to stir-up the divide-and-conquer hostility. Those who conform like good little baa, baas, are praised while those who have seen through the tidal wave of lies are ‘Covidiots’. The awake have been abused by the fast asleep for not conforming to fascism and impositions that the awake know are designed to endanger their health, dehumanise them, and tear asunder the very fabric of human society. We have had the curtain-twitchers and morons reporting neighbours and others to the face-nappied police for breaking ‘Covid rules’ with fascist police delighting in posting links and phone numbers where this could be done. The Cult cannot impose its will without a compliant police

and military or a compliant population willing to play their part in enslaving themselves and their kids. The words of a pastor in Nazi Germany are so appropriate today:

First they came for the socialists and I did not speak out because I was not a socialist.

Then they came for the trade unionists and I did not speak out because I was not a trade unionist.

Then they came for the Jews and I did not speak out because I was not a Jew.

Then they came for me and there was no one left to speak for me.

Those who don't learn from history are destined to repeat it and so many are.

'Covid' rules: Rewiring the mind

With the background laid out to this gigantic national and global web of psychological manipulation we can put 'Covid' rules into a clear and sinister perspective. Forget the claims about protecting health. 'Covid' rules are about dismantling the human mind, breaking the human spirit, destroying self-respect, and then putting Humpty Dumpty together again as a servile, submissive slave. Social isolation through lockdown and distancing have devastating effects on the human psyche as the psychological psychopaths well know and that's the real reason for them. Humans need contact with each other, discourse, closeness and touch, or they eventually, and literally, go crazy. Masks, which I will address at some length, fundamentally add to the effects of isolation and the Cult agenda to dehumanise and de-individualise the population. To do this while knowing – in fact *seeking* – this outcome is the very epitome of evil and psychologists involved in this *are* the epitome of evil. They must like all the rest of the Cult demons and their assets stand trial for crimes against humanity on a scale that defies the imagination. Psychopaths in uniform use isolation to break enemy troops and agents and make them subservient and submissive to tell what they know. The technique is rightly considered a form of torture and

torture is most certainly what has been imposed on the human population.

Clinically-insane American psychologist Harry Harlow became famous for his isolation experiments in the 1950s in which he separated baby monkeys from their mothers and imprisoned them for months on end in a metal container or ‘pit of despair’. They soon began to show mental distress and depression as any idiot could have predicted. Harlow put other monkeys in steel chambers for three, six or twelve months while denying them any contact with animals or humans. He said that the effects of total social isolation for six months were ‘so devastating and debilitating that we had assumed initially that twelve months of isolation would not produce any additional decrement’; but twelve months of isolation ‘almost obliterated the animals socially’. This is what the Cult and its psychopaths are doing to you and your children. Even monkeys in partial isolation in which they were not allowed to form relationships with other monkeys became ‘aggressive and hostile, not only to others, but also towards their own bodies’. We have seen this in the young as a consequence of lockdown. UK government psychopaths launched a public relations campaign telling people not to hug each other even after they received the ‘Covid-19 vaccine’ which we were told with more lies would allow a return to ‘normal life’. A government source told *The Telegraph*: ‘It will be along the lines that it is great that you have been vaccinated, but if you are going to visit your family and hug your grandchildren there is a chance you are going to infect people you love.’ The source was apparently speaking from a secure psychiatric facility. Janet Lord, director of Birmingham University’s Institute of Inflammation and Ageing, said that parents and grandparents should avoid hugging their children. Well, how can I put it, Ms Lord? Fuck off. Yep, that’ll do.

Destroying the kids – where are the parents?

Observe what has happened to people enslaved and isolated by lockdown as suicide and self-harm has soared worldwide,

particularly among the young denied the freedom to associate with their friends. A study of 49,000 people in English-speaking countries concluded that almost half of young adults are at clinical risk of mental health disorders. A national survey in America of 1,000 currently enrolled high school and college students found that 5 percent reported attempting suicide during the pandemic. Data from the US CDC's National Syndromic Surveillance Program from January 1st to October 17th, 2020, revealed a 31 percent increase in mental health issues among adolescents aged 12 to 17 compared with 2019. The CDC reported that America in general suffered the biggest drop in life expectancy since World War Two as it fell by a year in the first half of 2020 as a result of 'deaths of despair' – overdoses and suicides. Deaths of despair have leapt by more than 20 percent during lockdown and include the highest number of fatal overdoses ever recorded in a single year – 81,000. Internet addiction is another consequence of being isolated at home which lowers interest in physical activities as kids fall into inertia and what's the point? Children and young people are losing hope and giving up on life, sometimes literally. A 14-year-old boy killed himself in Maryland because he had 'given up' when his school district didn't reopen; an 11-year-old boy shot himself during a zoom class; a teenager in Maine succumbed to the isolation of the 'pandemic' when he ended his life after experiencing a disrupted senior year at school. Children as young as nine have taken their life and all these stories can be repeated around the world. Careers are being destroyed before they start and that includes those in sport in which promising youngsters have not been able to take part. The plan of the psycho-psychologists is working all right. Researchers at Cambridge University found that lockdowns cause significant harm to children's mental health. Their study was published in the *Archives of Disease in Childhood*, and followed 168 children aged between 7 and 11. The researchers concluded:

During the UK lockdown, children's depression symptoms have increased substantially, relative to before lockdown. The scale of this effect has direct relevance for the continuation of different elements of lockdown policy, such as complete or partial school closures ...

... Specifically, we observed a statistically significant increase in ratings of depression, with a medium-to-large effect size. Our findings emphasise the need to incorporate the potential impact of lockdown on child mental health in planning the ongoing response to the global pandemic and the recovery from it.

Not a chance when the Cult's psycho-psychologists were getting exactly what they wanted. The UK's Royal College of Paediatrics and Child Health has urged parents to look for signs of eating disorders in children and young people after a three to four fold increase. Specialists say the 'pandemic' is a major reason behind the rise. You don't say. The College said isolation from friends during school closures, exam cancellations, loss of extra-curricular activities like sport, and an increased use of social media were all contributory factors along with fears about the virus (psycho-psychologists again), family finances, and students being forced to quarantine. Doctors said young people were becoming severely ill by the time they were seen with 'Covid' regulations reducing face-to-face consultations. Nor is it only the young that have been devastated by the psychopaths. Like all bullies and cowards the Cult is targeting the young, elderly, weak and infirm. A typical story was told by a British lady called Lynn Parker who was not allowed to visit her husband in 2020 for the last ten and half months of his life 'when he needed me most' between March 20th and when he died on December 19th. This vacates the criminal and enters the territory of evil. The emotional impact on the immune system alone is immense as are the number of people of all ages worldwide who have died as a result of Cult-demanded, Gates-demanded, lockdowns.

Isolation is torture

The experience of imposing solitary confinement on millions of prisoners around the world has shown how a large percentage become 'actively psychotic and/or acutely suicidal'. Social isolation has been found to trigger 'a specific psychiatric syndrome, characterized by hallucinations; panic attacks; overt paranoia; diminished impulse control; hypersensitivity to external stimuli; and difficulties with thinking, concentration and memory'. Juan Mendez,

a United Nations rapporteur (investigator), said that isolation is a form of torture. Research has shown that even after isolation prisoners find it far more difficult to make social connections and I remember chatting to a shop assistant after one lockdown who told me that when her young son met another child again he had no idea how to act or what to do. Hannah Flanagan, Director of Emergency Services at Journey Mental Health Center in Dane County, Wisconsin, said: ‘The specificity about Covid social distancing and isolation that we’ve come across as contributing factors to the suicides are really new to us this year.’ But they are not new to those that devised them. They are getting the effect they want as the population is psychologically dismantled to be rebuilt in a totally different way. Children and the young are particularly targeted. They will be the adults when the full-on fascist AI-controlled technocracy is planned to be imposed and they are being prepared to meekly submit. At the same time older people who still have a memory of what life was like before – and how fascist the new normal really is – are being deleted. You are going to see efforts to turn the young against the old to support this geriatric genocide. Hannah Flanagan said the big increase in suicide in her county proved that social isolation is not only harmful, but deadly. Studies have shown that isolation from others is one of the main risk factors in suicide and even more so with women. Warnings that lockdown could create a ‘perfect storm’ for suicide were ignored. After all this was one of the *reasons* for lockdown. Suicide, however, is only the most extreme of isolation consequences. There are many others. Dr Dhruv Khullar, assistant professor of healthcare policy at Weill Cornell Medical College, said in a *New York Times* article in 2016 long before the fake ‘pandemic’:

A wave of new research suggests social separation is bad for us. Individuals with less social connection have disrupted sleep patterns, altered immune systems, more inflammation and higher levels of stress hormones. One recent study found that isolation increases the risk of heart disease by 29 percent and stroke by 32 percent. Another analysis that pooled data from 70 studies and 3.4 million people found that socially isolated individuals had a 30 percent higher risk of dying in the next seven years, and that this effect was largest in middle age.

Loneliness can accelerate cognitive decline in older adults, and isolated individuals are twice as likely to die prematurely as those with more robust social interactions. These effects start early: Socially isolated children have significantly poorer health 20 years later, even after controlling for other factors. All told, loneliness is as important a risk factor for early death as obesity and smoking.

There you have proof from that one article alone four years before 2020 that those who have enforced lockdown, social distancing and isolation knew what the effect would be and that is even more so with professional psychologists that have been driving the policy across the globe. We can go back even further to the years 2000 and 2003 and the start of a major study on the effects of isolation on health by Dr Janine Gronewold and Professor Dirk M. Hermann at the University Hospital in Essen, Germany, who analysed data on 4,316 people with an average age of 59 who were recruited for the long-term research project. They found that socially isolated people are more than 40 percent more likely to have a heart attack, stroke, or other major cardiovascular event and nearly 50 percent more likely to die from any cause. Given the financial Armageddon unleashed by lockdown we should note that the study found a relationship between increased cardiovascular risk and lack of financial support. After excluding other factors social isolation was still connected to a 44 percent increased risk of cardiovascular problems and a 47 percent increased risk of death by any cause. Lack of financial support was associated with a 30 percent increase in the risk of cardiovascular health events. Dr Gronewold said it had been known for some time that feeling lonely or lacking contact with close friends and family can have an impact on physical health and the study had shown that having strong social relationships is of high importance for heart health. Gronewold said they didn't understand yet why people who are socially isolated have such poor health outcomes, but this was obviously a worrying finding, particularly during these times of prolonged social distancing. Well, it can be explained on many levels. You only have to identify the point in the body where people feel loneliness and missing people they are parted from – it's in the centre of the chest where they feel the ache of loneliness and the ache of missing people. 'My heart aches for

you' ... 'My heart aches for some company.' I will explain this more in the chapter Escaping Wetiko, but when you realise that the body is the mind – they are expressions of each other – the reason why state of the mind dictates state of the body becomes clear.

American psychologist Ranjit Powar was highlighting the effects of lockdown isolation as early as April, 2020. She said humans have evolved to be social creatures and are wired to live in interactive groups. Being isolated from family, friends and colleagues could be unbalancing and traumatic for most people and could result in short or even long-term psychological and physical health problems. An increase in levels of anxiety, aggression, depression, forgetfulness and hallucinations were possible psychological effects of isolation. 'Mental conditions may be precipitated for those with underlying pre-existing susceptibilities and show up in many others without any pre-condition.' Powar said personal relationships helped us cope with stress and if we lost this outlet for letting off steam the result can be a big emotional void which, for an average person, was difficult to deal with. 'Just a few days of isolation can cause increased levels of anxiety and depression' – so what the hell has been the effect on the global population of *18 months* of this at the time of writing? Powar said: 'Add to it the looming threat of a dreadful disease being repeatedly hammered in through the media and you have a recipe for many shades of mental and physical distress.' For those with a house and a garden it is easy to forget that billions have had to endure lockdown isolation in tiny overcrowded flats and apartments with nowhere to go outside. The psychological and physical consequences of this are unimaginable and with lunatic and abusive partners and parents the consequences have led to tremendous increases in domestic and child abuse and alcoholism as people seek to shut out the horror. Ranjit Powar said:

Staying in a confined space with family is not all a rosy picture for everyone. It can be extremely oppressive and claustrophobic for large low-income families huddled together in small single-room houses. Children here are not lucky enough to have many board/electronic games or books to keep them occupied.

Add to it the deep insecurity of running out of funds for food and basic necessities. On the other hand, there are people with dysfunctional family dynamics, such as domineering, abusive or alcoholic partners, siblings or parents which makes staying home a period of trial. Incidence of suicide and physical abuse against women has shown a worldwide increase. Heightened anxiety and depression also affect a person's immune system, making them more susceptible to illness.

To think that Powar's article was published on April 11th, 2020.

Six-feet fantasy

Social (unsocial) distancing demanded that people stay six feet or two metres apart. UK government advisor Robert Dingwall from the New and Emerging Respiratory Virus Threats Advisory Group said in a radio interview that the two-metre rule was 'conjured up out of nowhere' and was not based on science. No, it was not based on *medical* science, but it didn't come out of nowhere. The distance related to *psychological* science. Six feet/two metres was adopted in many countries and we were told by people like the criminal Anthony Fauci and his ilk that it was founded on science. Many schools could not reopen because they did not have the space for six-feet distancing. Then in March, 2021, after a year of six-feet 'science', a study published in the *Journal of Infectious Diseases* involving more than 500,000 students and almost 100,000 staff over 16 weeks revealed no significant difference in 'Covid' cases between six feet and three feet and Fauci changed his tune. Now three feet was okay. There is no difference between six feet and three *inches* when there is no 'virus' and they got away with six feet for psychological reasons for as long as they could. I hear journalists and others talk about 'unintended consequences' of lockdown. They are not *unintended* at all; they have been coldly-calculated for a specific outcome of human control and that's why super-psychopaths like Gates have called for them so vehemently. Super-psychopath psychologists have demanded them and psychopathic or clueless, spineless, politicians have gone along with them by 'following the science'. But it's not science at all. 'Science' is not what is; it's only what people can be manipulated to believe it is. The whole 'Covid' catastrophe is

founded on mind control. Three word or three statement mantras issued by the UK government are a well-known mind control technique and so we've had 'Stay home/protect the NHS/save lives', 'Stay alert/control the virus/save lives' and 'hands/face/space'. One of the most vocal proponents of extreme 'Covid' rules in the UK has been Professor Susan Michie, a member of the British Communist Party, who is not a medical professional. Michie is the director of the Centre for Behaviour Change at University College London. She is a *behavioural psychologist* and another filthy rich 'Marxist' who praised China's draconian lockdown. She was known by fellow students at Oxford University as 'Stalin's nanny' for her extreme Marxism. Michie is an influential member of the UK government's Scientific Advisory Group for Emergencies (SAGE) and behavioural manipulation groups which have dominated 'Covid' policy. She is a consultant adviser to the World Health Organization on 'Covid-19' and behaviour. Why the hell are lockdowns anything to do with her when they are claimed to be about health? Why does a behavioural psychologist from a group charged with changing the behaviour of the public want lockdown, human isolation and mandatory masks? Does that question really need an answer? Michie *absolutely* has to explain herself before a Nuremberg court when humanity takes back its world again and even more so when you see the consequences of masks that she demands are compulsory. This is a Michie classic:

The benefits of getting primary school children to wear masks is that regardless of what little degree of transmission is occurring in those age groups it could help normalise the practice. Young children wearing masks may be more likely to get their families to accept masks.

Those words alone should carry a prison sentence when you ponder on the callous disregard for children involved and what a statement it makes about the mind and motivations of Susan Michie. What a lovely lady and what she said there encapsulates the mentality of the psychopaths behind the 'Covid' horror. Let us compare what Michie said with a countrywide study in Germany published at [researchsquare.com](https://www.researchsquare.com) involving 25,000 school children and 17,854 health complaints submitted by parents. Researchers

found that masks are harming children physically, psychologically, and behaviourally with 24 health issues associated with mask wearing. They include: shortness of breath (29.7%); dizziness (26.4%); increased headaches (53%); difficulty concentrating (50%); drowsiness or fatigue (37%); and malaise (42%). Nearly a third of children experienced more sleep issues than before and a quarter developed new fears. Researchers found health issues and other impairments in 68 percent of masked children covering their faces for an average of 4.5 hours a day. Hundreds of those taking part experienced accelerated respiration, tightness in the chest, weakness, and short-term impairment of consciousness. A reminder of what Michie said again:

The benefits of getting primary school children to wear masks is that regardless of what little degree of transmission is occurring in those age groups it could help normalise the practice. Young children wearing masks may be more likely to get their families to accept masks.

Psychopaths in government and psychology now have children and young people – plus all the adults – wearing masks for hours on end while clueless teachers impose the will of the psychopaths on the young they should be protecting. What the hell are parents doing?

Cult lab rats

We have some schools already imposing on students microchipped buzzers that activate when they get ‘too close’ to their pals in the way they do with lab rats. How apt. To the Cult and its brain-dead servants our children *are* lab rats being conditioned to be unquestioning, dehumanised slaves for the rest of their lives.

Children and young people are being weaned and frightened away from the most natural human instincts including closeness and touch. I have tracked in the books over the years how schools were banning pupils from greeting each other with a hug and the whole Cult-induced Me Too movement has terrified men and boys from a relaxed and natural interaction with female friends and work colleagues to the point where many men try never to be in a room

alone with a woman that's not their partner. Airhead celebrities have as always played their virtue-signalling part in making this happen with their gross exaggeration. For every monster like Harvey Weinstein there are at least tens of thousands of men that don't treat women like that; but everyone must be branded the same and policy changed for them as well as the monster. I am going to be using the word 'dehumanise' many times in this chapter because that is what the Cult is seeking to do and it goes very deep as we shall see. Don't let them kid you that social distancing is planned to end one day. That's not the idea. We are seeing more governments and companies funding and producing wearable gadgets to keep people apart and they would not be doing that if this was meant to be short-term. A tech start-up company backed by GCHQ, the British Intelligence and military surveillance headquarters, has created a social distancing wrist sensor that alerts people when they get too close to others. The CIA has also supported tech companies developing similar devices. The wearable sensor was developed by Tended, one of a number of start-up companies supported by GCHQ (see the CIA and DARPA). The device can be worn on the wrist or as a tag on the waistband and will vibrate whenever someone wearing the device breaches social distancing and gets anywhere near natural human contact. The company had a lucky break in that it was developing a distancing sensor when the 'Covid' hoax arrived which immediately provided a potentially enormous market. How fortunate. The government in big-time Cult-controlled Ontario in Canada is investing \$2.5 million in wearable contact tracing technology that 'will alert users if they may have been exposed to the Covid-19 in the workplace and will beep or vibrate if they are within six feet of another person'. Facedrive Inc., the technology company behind this, was founded in 2016 with funding from the Ontario Together Fund and obviously they, too, had a prophet on the board of directors. The human surveillance and control technology is called TraceSCAN and would be worn by the human cyborgs in places such as airports, workplaces, construction sites, care homes and ... schools.

I emphasise schools with children and young people the prime targets. You know what is planned for society as a whole if you keep your eyes on the schools. They have always been places where the state program the next generation of slaves to be its compliant worker-ants – or Woker-ants these days; but in the mist of the ‘Covid’ madness they have been transformed into mind laboratories on a scale never seen before. Teachers and head teachers are just as programmed as the kids – often more so. Children are kept apart from human interaction by walk lanes, classroom distancing, staggered meal times, masks, and the rolling-out of buzzer systems. Schools are now physically laid out as a laboratory maze for lab-rats. Lunatics at a school in Anchorage, Alaska, who should be prosecuted for child abuse, took away desks and forced children to kneel (know your place) on a mat for five hours a day while wearing a mask and using their chairs as a desk. How this was supposed to impact on a ‘virus’ only these clinically insane people can tell you and even then it would be clap-trap. The school banned recess (interaction), art classes (creativity), and physical exercise (getting body and mind moving out of inertia). Everyone behind this outrage should be in jail or better still a mental institution. The behavioural manipulators are all for this dystopian approach to schools.

Professor Susan Michie, the mind-doctor and British Communist Party member, said it was wrong to say that schools were safe. They had to be made so by ‘distancing’, masks and ventilation (sitting all day in the cold). I must ask this lady round for dinner on a night I know I am going to be out and not back for weeks. She probably wouldn’t be able to make it, anyway, with all the visits to her own psychologist she must have block-booked.

Masking identity

I know how shocking it must be for you that a behaviour manipulator like Michie wants everyone to wear masks which have long been a feature of mind-control programs like the infamous MKUltra in the United States, but, there we are. We live and learn. I spent many years from 1996 to right across the millennium

researching mind control in detail on both sides of the Atlantic and elsewhere. I met a large number of mind-control survivors and many had been held captive in body and mind by MKUltra. MK stands for mind-control, but employs the German spelling in deference to the Nazis spirited out of Germany at the end of World War Two by Operation Paperclip in which the US authorities, with help from the Vatican, transported Nazi mind-controllers and engineers to America to continue their work. Many of them were behind the creation of NASA and they included Nazi scientist and SS officer Wernher von Braun who swapped designing V-2 rockets to bombard London with designing the Saturn V rockets that powered the NASA moon programme's Apollo craft. I think I may have mentioned that the Cult has no borders. Among Paperclip escapees was Josef Mengele, the Angel of Death in the Nazi concentration camps where he conducted mind and genetic experiments on children often using twins to provide a control twin to measure the impact of his 'work' on the other. If you want to observe the Cult mentality in all its extremes of evil then look into the life of Mengele. I have met many people who suffered mercilessly under Mengele in the United States where he operated under the name Dr Greene and became a stalwart of MKUltra programming and torture. Among his locations was the underground facility in the Mojave Desert in California called the China Lake Naval Weapons Station which is almost entirely below the surface. My books *The Biggest Secret*, *Children of the Matrix* and *The Perception Deception* have the detailed background to MKUltra.

The best-known MKUltra survivor is American Cathy O'Brien. I first met her and her late partner Mark Phillips at a conference in Colorado in 1996. Mark helped her escape and deprogram from decades of captivity in an offshoot of MKUltra known as Project Monarch in which 'sex slaves' were provided for the rich and famous including Father George Bush, Dick Cheney and the Clintons. Read Cathy and Mark's book *Trance-Formation of America* and if you are new to this you will be shocked to the core. I read it in 1996 shortly before, with the usual synchronicity of my life, I found

myself given a book table at the conference right next to hers. MKUltra never ended despite being very publicly exposed (only a small part of it) in the 1970s and continues in other guises. I am still in touch with Cathy. She contacted me during 2020 after masks became compulsory in many countries to tell me how they were used as part of MKUltra programming. I had been observing 'Covid regulations' and the relationship between authority and public for months. I saw techniques that I knew were employed on individuals in MKUltra being used on the global population. I had read many books and manuals on mind control including one called *Silent Weapons for Quiet Wars* which came to light in the 1980s and was a guide on how to perceptually program on a mass scale. 'Silent Weapons' refers to mind-control. I remembered a line from the manual as governments, medical authorities and law enforcement agencies have so obviously talked to – or rather at – the adult population since the 'Covid' hoax began as if they are children. The document said:

If a person is spoken to by a T.V. advertiser as if he were a twelve-year-old, then, due to suggestibility, he will, with a certain probability, respond or react to that suggestion with the uncritical response of a twelve-year-old and will reach in to his economic reservoir and deliver its energy to buy that product on impulse when he passes it in the store.

That's why authority has spoken to adults like children since all this began.

Why did Michael Jackson wear masks?

Every aspect of the 'Covid' narrative has mind-control as its central theme. Cathy O'Brien wrote an article for davidicke.com about the connection between masks and mind control. Her daughter Kelly who I first met in the 1990s was born while Cathy was still held captive in MKUltra. Kelly was forced to wear a mask as part of her programming from the age of *two* to dehumanise her, target her sense of individuality and reduce the amount of oxygen her brain and body received. *Bingo*. This is the real reason for compulsory

masks, why they have been enforced en masse, and why they seek to increase the number they demand you wear. First one, then two, with one disgraceful alleged ‘doctor’ recommending four which is nothing less than a death sentence. Where and how often they must be worn is being expanded for the purpose of mass mind control and damaging respiratory health which they can call ‘Covid-19’. Canada’s government headed by the man-child Justin Trudeau, says it’s fine for children of two and older to wear masks. An insane ‘study’ in Italy involving just 47 children concluded there was no problem for babies as young as *four months* wearing them. Even after people were ‘vaccinated’ they were still told to wear masks by the criminal that is Anthony Fauci. Cathy wrote that mandating masks is allowing the authorities literally to control the air we breathe which is what was done in MKUltra. You might recall how the singer Michael Jackson wore masks and there is a reason for that. He was subjected to MKUltra mind control through Project Monarch and his psyche was scrambled by these simpletons. Cathy wrote:

In MKUltra Project Monarch mind control, Michael Jackson had to wear a mask to silence his voice so he could not reach out for help. Remember how he developed that whisper voice when he wasn’t singing? Masks control the mind from the outside in, like the redefining of words is doing. By controlling what we can and cannot say for fear of being labeled racist or beaten, for example, it ultimately controls thought that drives our words and ultimately actions (or lack thereof).

Likewise, a mask muffles our speech so that we are not heard, which controls voice ... words ... mind. This is Mind Control. Masks are an obvious mind control device, and I am disturbed so many people are complying on a global scale. Masks depersonalize while making a person feel as though they have no voice. It is a barrier to others. People who would never choose to comply but are forced to wear a mask in order to keep their job, and ultimately their family fed, are compromised. They often feel shame and are subdued. People have stopped talking with each other while media controls the narrative.

The ‘no voice’ theme has often become literal with train passengers told not to speak to each other in case they pass on the ‘virus’, singing banned for the same reason and bonkers California officials telling people riding roller coasters that they cannot shout and scream. Cathy said she heard every day from healed MKUltra survivors who cannot wear a mask without flashing back on ways

their breathing was controlled – ‘from ball gags and penises to water boarding’. She said that through the years when she saw images of people in China wearing masks ‘due to pollution’ that it was really to control their oxygen levels. ‘I knew it was as much of a population control mechanism of depersonalisation as are burkas’, she said. Masks are another Chinese communist/fascist method of control that has been swept across the West as the West becomes China at lightning speed since we entered 2020.

Mask-19

There are other reasons for mandatory masks and these include destroying respiratory health to call it ‘Covid-19’ and stunting brain development of children and the young. Dr Margarite Griesz-Brisson MD, PhD, is a Consultant Neurologist and Neurophysiologist and the Founder and Medical Director of the London Neurology and Pain Clinic. Her CV goes down the street and round the corner. She is clearly someone who cares about people and won’t parrot the propaganda. Griesz-Brisson has a PhD in pharmacology, with special interest in neurotoxicology, environmental medicine, neuroregeneration and neuroplasticity (the way the brain can change in the light of information received). She went public in October, 2020, with a passionate warning about the effects of mask-wearing laws:

The reinhalation of our exhaled air will without a doubt create oxygen deficiency and a flooding of carbon dioxide. We know that the human brain is very sensitive to oxygen deprivation. There are nerve cells for example in the hippocampus that can’t be longer than 3 minutes without oxygen – they cannot survive. The acute warning symptoms are headaches, drowsiness, dizziness, issues in concentration, slowing down of reaction time – reactions of the cognitive system.

Oh, I know, let’s tell bus, truck and taxi drivers to wear them and people working machinery. How about pilots, doctors and police? Griesz-Brisson makes the important point that while the symptoms she mentions may fade as the body readjusts this does not alter the fact that people continue to operate in oxygen deficit with long list of

potential consequences. She said it was well known that neurodegenerative diseases take years or decades to develop. 'If today you forget your phone number, the breakdown in your brain would have already started 20 or 30 years ago.' She said degenerative processes in your brain are getting amplified as your oxygen deprivation continues through wearing a mask. Nerve cells in the brain are unable to divide themselves normally in these circumstances and lost nerve cells will no longer be regenerated. 'What is gone is gone.' Now consider that people like shop workers and *schoolchildren* are wearing masks for hours every day. What in the name of sanity is going to be happening to them? 'I do not wear a mask, I need my brain to think', Griesz-Brisson said, 'I want to have a clear head when I deal with my patients and not be in a carbon dioxide-induced anaesthesia'. If you are told to wear a mask anywhere ask the organisation, police, store, whatever, for their risk assessment on the dangers and negative effects on mind and body of enforcing mask-wearing. They won't have one because it has never been done not even by government. All of them must be subject to class-action lawsuits as the consequences come to light. They don't do mask risk assessments for an obvious reason. They know what the conclusions would be and independent scientific studies that *have* been done tell a horror story of consequences.

'Masks are criminal'

Dr Griesz-Brisson said that for children and adolescents, masks are an absolute no-no. They had an extremely active and adaptive immune system and their brain was incredibly active with so much to learn. 'The child's brain, or the youth's brain, is thirsting for oxygen.' The more metabolically active an organ was, the more oxygen it required; and in children and adolescents every organ was metabolically active. Griesz-Brisson said that to deprive a child's or adolescent's brain of oxygen, or to restrict it in any way, was not only dangerous to their health, it was absolutely criminal. 'Oxygen deficiency inhibits the development of the brain, and the damage that has taken place as a result CANNOT be reversed.' Mind

manipulators of MKUltra put masks on two-year-olds they wanted to neurologically rewire and you can see why. Griesz-Brisson said a child needs the brain to learn and the brain needs oxygen to function. 'We don't need a clinical study for that. This is simple, indisputable physiology.' Consciously and purposely induced oxygen deficiency was an absolutely deliberate health hazard, and an absolute medical contraindication which means that 'this drug, this therapy, this method or measure should not be used, and is not allowed to be used'. To coerce an entire population to use an absolute medical contraindication by force, she said, there had to be definite and serious reasons and the reasons must be presented to competent interdisciplinary and independent bodies to be verified and authorised. She had this warning of the consequences that were coming if mask wearing continued:

When, in ten years, dementia is going to increase exponentially, and the younger generations couldn't reach their god-given potential, it won't help to say 'we didn't need the masks'. I know how damaging oxygen deprivation is for the brain, cardiologists know how damaging it is for the heart, pulmonologists know how damaging it is for the lungs. Oxygen deprivation damages every single organ. Where are our health departments, our health insurance, our medical associations? It would have been their duty to be vehemently against the lockdown and to stop it and stop it from the very beginning.

Why do the medical boards issue punishments to doctors who give people exemptions? Does the person or the doctor seriously have to prove that oxygen deprivation harms people? What kind of medicine are our doctors and medical associations representing? Who is responsible for this crime? The ones who want to enforce it? The ones who let it happen and play along, or the ones who don't prevent it?

All of the organisations and people she mentions there either answer directly to the Cult or do whatever hierarchical levels above them tell them to do. The outcome of both is the same. 'It's not about masks, it's not about viruses, it's certainly not about your health', Griesz-Brisson said. 'It is about much, much more. I am not participating. I am not afraid.' They were taking our air to breathe and there was no unfounded medical exemption from face masks. Oxygen deprivation was dangerous for every single brain. It had to be the free decision of every human being whether they want to

wear a mask that was absolutely ineffective to protect themselves from a virus. She ended by rightly identifying where the responsibility lies for all this:

The imperative of the hour is personal responsibility. We are responsible for what we think, not the media. We are responsible for what we do, not our superiors. We are responsible for our health, not the World Health Organization. And we are responsible for what happens in our country, not the government.

Halle-bloody-lujah.

But surgeons wear masks, right?

Independent studies of mask-wearing have produced a long list of reports detailing mental, emotional and physical dangers. What a definition of insanity to see police officers imposing mask-wearing on the public which will cumulatively damage their health while the police themselves wear masks that will cumulatively damage *their* health. It's utter madness and both public and police do this because 'the government says so' – yes a government of brain-donor idiots like UK Health Secretary Matt Hancock reading the 'follow the science' scripts of psychopathic, lunatic psychologists. The response you get from Stockholm syndrome sufferers defending the very authorities that are destroying them and their families is that 'surgeons wear masks'. This is considered the game, set and match that they must work and don't cause oxygen deficit. Well, actually, scientific studies have shown that they *do* and oxygen levels are monitored in operating theatres to compensate. Surgeons wear masks to stop spittle and such like dropping into open wounds – not to stop 'viral particles' which are so minuscule they can only be seen through an electron microscope. Holes in the masks are significantly bigger than 'viral particles' and if you sneeze or cough they will breach the mask. I watched an incredibly disingenuous 'experiment' that claimed to prove that masks work in catching 'virus' material from the mouth and nose. They did this with a slow motion camera and the mask did block big stuff which stayed inside the mask and

against the face to be breathed in or cause infections on the face as we have seen with many children. ‘Viral particles’, however, would never have been picked up by the camera as they came through the mask when they are far too small to be seen. The ‘experiment’ was therefore disingenuous *and* useless.

Studies have concluded that wearing masks in operating theatres (and thus elsewhere) make no difference to preventing infection while the opposite is true with toxic shite building up in the mask and this had led to an explosion in tooth decay and gum disease dubbed by dentists ‘mask mouth’. You might have seen the Internet video of a furious American doctor urging people to take off their masks after a four-year-old patient had been rushed to hospital the night before and nearly died with a lung infection that doctors sourced to mask wearing. A study in the journal *Cancer Discovery* found that inhalation of harmful microbes can contribute to advanced stage lung cancer in adults and long-term use of masks can help breed dangerous pathogens. Microbiologists have said frequent mask wearing creates a moist environment in which microbes can grow and proliferate before entering the lungs. The Canadian Agency for Drugs and Technologies in Health, or CADTH, a Canadian national organisation that provides research and analysis to healthcare decision-makers, said this as long ago as 2013 in a report entitled ‘Use of Surgical Masks in the Operating Room: A Review of the Clinical Effectiveness and Guidelines’. It said:

- No evidence was found to support the use of surgical face masks to reduce the frequency of surgical site infections
- No evidence was found on the effectiveness of wearing surgical face masks to protect staff from infectious material in the operating room.
- Guidelines recommend the use of surgical face masks by staff in the operating room to protect both operating room staff and patients (despite the lack of evidence).

We were told that the world could go back to ‘normal’ with the arrival of the ‘vaccines’. When they came, fraudulent as they are, the story changed as I knew that it would. We are in the midst of transforming ‘normal’, not going back to it. Mary Ramsay, head of immunisation at Public Health England, echoed the words of US criminal Anthony Fauci who said masks and other regulations must stay no matter if people are vaccinated. The Fauci idiot continued to wear two masks – different colours so both could be clearly seen – after he *claimed* to have been vaccinated. Senator Rand Paul told Fauci in one exchange that his double-masks were ‘theatre’ and he was right. It’s all theatre. Mary Ramsay back-tracked on the vaccine-return-to-normal theme when she said the public may need to wear masks and social-distance for years despite the jabs. ‘People have got used to those lower-level restrictions now, and [they] can live with them’, she said telling us what the idea has been all along. ‘The vaccine does not give you a pass, even if you have had it, you must continue to follow all the guidelines’ said a Public Health England statement which reneged on what we had been told before and made having the ‘vaccine’ irrelevant to ‘normality’ even by the official story. Spain’s fascist government trumped everyone by passing a law mandating the wearing of masks on the beach and even when swimming in the sea. The move would have devastated what’s left of the Spanish tourist industry, posed potential breathing dangers to swimmers and had Northern European sunbathers walking around with their forehead brown and the rest of their face white as a sheet. The ruling was so crazy that it had to be retracted after pressure from public and tourist industry, but it confirmed where the Cult wants to go with masks and how clinically insane authority has become. The determination to make masks permanent and hide the serious dangers to body and mind can be seen in the censorship of scientist Professor Denis Rancourt by Bill Gates-funded academic publishing website ResearchGate over his papers exposing the dangers and uselessness of masks. Rancourt said:

ResearchGate today has permanently locked my account, which I have had since 2015. Their reasons graphically show the nature of their attack against democracy, and their corruption of

science ... By their obscene non-logic, a scientific review of science articles reporting on harms caused by face masks has a 'potential to cause harm'. No criticism of the psychological device (face masks) is tolerated, if the said criticism shows potential to influence public policy.

This is what happens in a fascist world.

Where are the 'greens' (again)?

Other dangers of wearing masks especially regularly relate to the inhalation of minute plastic fibres into the lungs and the deluge of discarded masks in the environment and oceans. Estimates predicted that more than 1.5 billion disposable masks will end up in the world's oceans every year polluting the water with tons of plastic and endangering marine wildlife. Studies project that humans are using 129 billion face masks each month worldwide – about three million a minute. Most are disposable and made from plastic, non-biodegradable microfibers that break down into smaller plastic particles that become widespread in ecosystems. They are littering cities, clogging sewage channels and turning up in bodies of water. I have written in other books about the immense amounts of microplastics from endless sources now being absorbed into the body. Rolf Halden, director of the Arizona State University (ASU) Biodesign Center for Environmental Health Engineering, was the senior researcher in a 2020 study that analysed 47 human tissue samples and found microplastics in all of them. 'We have detected these chemicals of plastics in every single organ that we have investigated', he said. I wrote in *The Answer* about the world being deluged with microplastics. A study by the Worldwide Fund for Nature (WWF) found that people are consuming on average every week some 2,000 tiny pieces of plastic mostly through water and also through marine life and the air. Every year humans are ingesting enough microplastics to fill a heaped dinner plate and in a life-time of 79 years it is enough to fill two large waste bins. Marco Lambertini, WWF International director general said: 'Not only are plastics polluting our oceans and waterways and killing marine life – it's in all of us and we can't escape consuming plastics,' American

geologists found tiny plastic fibres, beads and shards in rainwater samples collected from the remote slopes of the Rocky Mountain National Park near Denver, Colorado. Their report was headed: 'It is raining plastic.' Rachel Adams, senior lecturer in Biomedical Science at Cardiff Metropolitan University, said that among health consequences are internal inflammation and immune responses to a 'foreign body'. She further pointed out that microplastics become carriers of toxins including mercury, pesticides and dioxins (a known cause of cancer and reproductive and developmental problems). These toxins accumulate in the fatty tissues once they enter the body through microplastics. Now this is being compounded massively by people putting plastic on their face and throwing it away.

Workers exposed to polypropylene plastic fibres known as 'flock' have developed 'flock worker's lung' from inhaling small pieces of the flock fibres which can damage lung tissue, reduce breathing capacity and exacerbate other respiratory problems. Now ... commonly used surgical masks have three layers of melt-blown textiles made of ... polypropylene. We have billions of people putting these microplastics against their mouth, nose and face for hours at a time day after day in the form of masks. How does anyone think that will work out? I mean – what could possibly go wrong? We posted a number of scientific studies on this at davidicke.com, but when I went back to them as I was writing this book the links to the science research website where they were hosted were dead. Anything that challenges the official narrative in any way is either censored or vilified. The official narrative is so unsupportable by the evidence that only deleting the truth can protect it. A study by Chinese scientists still survived – with the usual twist which it why it was still active, I guess. Yes, they found that virtually all the masks they tested increased the daily intake of microplastic fibres, but people should still wear them because the danger from the 'virus' was worse said the crazy 'team' from the Institute of Hydrobiology in Wuhan. Scientists first discovered microplastics in lung tissue of some patients who died of lung cancer

in the 1990s. Subsequent studies have confirmed the potential health damage with the plastic degrading slowly and remaining in the lungs to accumulate in volume. Wuhan researchers used a machine simulating human breathing to establish that masks shed up to nearly 4,000 microplastic fibres in a month with reused masks producing more. Scientists said some masks are laced with toxic chemicals and a variety of compounds seriously restricted for both health and environmental reasons. They include cobalt (used in blue dye) and formaldehyde known to cause watery eyes, burning sensations in the eyes, nose, and throat, plus coughing, wheezing and nausea. No – that must be 'Covid-19'.

Mask 'worms'

There is another and potentially even more sinister content of masks. Mostly new masks of different makes filmed under a microscope around the world have been found to contain strange black fibres or 'worms' that appear to move or 'crawl' by themselves and react to heat and water. The nearest I have seen to them are the self-replicating fibres that are pulled out through the skin of those suffering from Morgellons disease which has been connected to the phenomena of 'chemtrails' which I will bring into the story later on. Morgellons fibres continue to grow outside the body and have a form of artificial intelligence. Black 'worm' fibres in masks have that kind of feel to them and there is a nanotechnology technique called 'worm micelles' which carry and release drugs or anything else you want to deliver to the body. For sure the suppression of humanity by mind altering drugs is the Cult agenda big time and the more excuses they can find to gain access to the body the more opportunities there are to make that happen whether through 'vaccines' or masks pushed against the mouth and nose for hours on end.

So let us summarise the pros and cons of masks:

Against masks: Breathing in your own carbon dioxide; depriving the body and brain of sufficient oxygen; build-up of toxins in the mask that can be breathed into the lungs and cause rashes on the face and ‘mask-mouth’; breathing microplastic fibres and toxic chemicals into the lungs; dehumanisation and deleting individualisation by literally making people faceless; destroying human emotional interaction through facial expression and deleting parental connection with their babies which look for guidance to their facial expression.

For masks: They don’t protect you from a ‘virus’ that doesn’t exist and even if it did ‘viral’ particles are so minute they are smaller than the holes in the mask.

Governments, police, supermarkets, businesses, transport companies, and all the rest who seek to impose masks have done no risk assessment on their consequences for health and psychology and are now open to group lawsuits when the impact becomes clear with a cumulative epidemic of respiratory and other disease. Authorities will try to exploit these effects and hide the real cause by dubbing them ‘Covid-19’. Can you imagine setting out to force the population to wear health-destroying masks without doing any assessment of the risks? It is criminal and it is evil, but then how many people targeted in this way, who see their children told to wear them all day at school, have asked for a risk assessment? Billions can’t be imposed upon by the few unless the billions allow it. Oh, yes, with just a tinge of irony, 85 percent of all masks made worldwide come from *China*.

Wash your hands in toxic shite

‘Covid’ rules include the use of toxic sanitisers and again the health consequences of constantly applying toxins to be absorbed through the skin is obvious to any level of Renegade Mind. America’s Food and Drug Administration (FDA) said that sanitisers are drugs and issued a warning about 75 dangerous brands which contain

methanol used in antifreeze and can cause death, kidney damage and blindness. The FDA circulated the following warning even for those brands that it claims to be safe:

Store hand sanitizer out of the reach of pets and children, and children should use it only with adult supervision. Do not drink hand sanitizer. This is particularly important for young children, especially toddlers, who may be attracted by the pleasant smell or brightly colored bottles of hand sanitizer.

Drinking even a small amount of hand sanitizer can cause alcohol poisoning in children. (However, there is no need to be concerned if your children eat with or lick their hands after using hand sanitizer.) During this coronavirus pandemic, poison control centers have had an increase in calls about accidental ingestion of hand sanitizer, so it is important that adults monitor young children's use.

Do not allow pets to swallow hand sanitizer. If you think your pet has eaten something potentially dangerous, call your veterinarian or a pet poison control center right away. Hand sanitizer is flammable and should be stored away from heat and flames. When using hand sanitizer, rub your hands until they feel completely dry before performing activities that may involve heat, sparks, static electricity, or open flames.

There you go, perfectly safe, then, and that's without even a mention of the toxins absorbed through the skin. Come on kids – sanitise your hands everywhere you go. It will save you from the 'virus'. Put all these elements together of the 'Covid' normal and see how much health and psychology is being cumulatively damaged, even devastated, to 'protect your health'. Makes sense, right? They are only imposing these things because they care, right? *Right?*

Submitting to insanity

Psychological reframing of the population goes very deep and is done in many less obvious ways. I hear people say how contradictory and crazy 'Covid' rules are and how they are ever changing. This is explained away by dismissing those involved as idiots. It is a big mistake. The Cult is delighted if its cold calculation is perceived as incompetence and idiocy when it is anything but. Oh, yes, there are idiots within the system – lots of them – but they are *administering* the Cult agenda, mostly unknowingly. They are not deciding and dictating it. The bulwark against tyranny is self-

respect, always has been, always will be. It is self-respect that has broken every tyranny in history. By its very nature self-respect will not bow to oppression and its perpetrators. There is so little self-respect that it's always the few that overturn dictators. Many may eventually follow, but the few with the iron spines (self-respect) kick it off and generate the momentum. The Cult targets self-respect in the knowledge that once this has gone only submission remains. Crazy, contradictory, ever-changing 'Covid' rules are systematically applied by psychologists to delete self-respect. They *want* you to see that the rules make no sense. It is one thing to decide to do something when *you* have made the choice based on evidence and logic. You still retain your self-respect. It is quite another when you can see what you are being told to do is insane, ridiculous and makes no sense, and *yet you still do it*. Your self-respect is extinguished and this has been happening as ever more obviously stupid and nonsensical things have been demanded and the great majority have complied even when they can see they are stupid and nonsensical.

People walk around in face-nappies knowing they are damaging their health and make no difference to a 'virus'. They do it in fear of not doing it. I know it's daft, but I'll do it anyway. When that happens something dies inside of you and submissive reframing has begun. Next there's a need to hide from yourself that you have conceded your self-respect and you convince yourself that you have not really submitted to fear and intimidation. You begin to believe that you are complying with craziness because it's the right thing to do. When first you concede your self-respect of $2+2 = 4$ to $2+2 = 5$ you *know* you are compromising your self-respect. Gradually to avoid facing that fact you begin to *believe* that $2+2=5$. You have been reframed and I have been watching this process happening in the human psyche on an industrial scale. The Cult is working to break your spirit and one of its major tools in that war is humiliation. I read how former American soldier Bradley Manning (later Chelsea Manning after a sex-change) was treated after being jailed for supplying WikiLeaks with documents exposing the enormity of

government and elite mendacity. Manning was isolated in solitary confinement for eight months, put under 24-hour surveillance, forced to hand over clothing before going to bed, and stand naked for every roll call. This is systematic humiliation. The introduction of anal swab 'Covid' tests in China has been done for the same reason to delete self-respect and induce compliant submission. Anal swabs are mandatory for incoming passengers in parts of China and American diplomats have said they were forced to undergo the indignity which would have been calculated humiliation by the Cult-owned Chinese government that has America in its sights.

Government-people: An abusive relationship

Spirit-breaking psychological techniques include giving people hope and apparent respite from tyranny only to take it away again. This happened in the UK during Christmas, 2020, when the psycho-psychologists and their political lackeys announced an easing of restrictions over the holiday only to reimpose them almost immediately on the basis of yet another lie. There is a big psychological difference between getting used to oppression and being given hope of relief only to have that dashed. Psychologists know this and we have seen the technique used repeatedly. Then there is traumatising people before you introduce more extreme regulations that require compliance. A perfect case was the announcement by the dark and sinister Whitty and Vallance in the UK that 'new data' predicted that 4,000 could die every day over the winter of 2020/2021 if we did not lockdown again. I think they call it lying and after traumatising people with that claim out came Jackboot Johnson the next day with new curbs on human freedom. Psychologists know that a frightened and traumatised mind becomes suggestable to submission and behaviour reframing. Underpinning all this has been to make people fearful and suspicious of each other and see themselves as a potential danger to others. In league with deleted self-respect you have the perfect psychological recipe for self-loathing. The relationship between authority and public is now demonstrably the same as that of

subservience to an abusive partner. These are signs of an abusive relationship explained by psychologist Leslie Becker-Phelps:

Psychological and emotional abuse: Undermining a partner's self-worth with verbal attacks, name-calling, and belittling. Humiliating the partner in public, unjustly accusing them of having an affair, or interrogating them about their every behavior. Keeping partner confused or off balance by saying they were just kidding or blaming the partner for 'making' them act this way ... Feigning in public that they care while turning against them in private. This leads to victims frequently feeling confused, incompetent, unworthy, hopeless, and chronically self-doubting. [Apply these techniques to how governments have treated the population since New Year, 2020, and the parallels are obvious.]

Physical abuse: The abuser might physically harm their partner in a range of ways, such as grabbing, hitting, punching, or shoving them. They might throw objects at them or harm them with a weapon. [Observe the physical harm imposed by masks, lockdown, and so on.]

Threats and intimidation: One way abusers keep their partners in line is by instilling fear. They might be verbally threatening, or give threatening looks or gestures. Abusers often make it known that they are tracking their partner's every move. They might destroy their partner's possessions, threaten to harm them, or threaten to harm their family members. Not surprisingly, victims of this abuse often feel anxiety, fear, and panic. [No words necessary.]

Isolation: Abusers often limit their partner's activities, forbidding them to talk or interact with friends or family. They might limit access to a car or even turn off their phone. All of this might be done by physically holding them against their will, but is often accomplished through psychological abuse and intimidation. The more isolated a person feels, the fewer resources they have to help gain perspective on their situation and to escape from it. [No words necessary.]

Economic abuse: Abusers often make their partners beholden to them for money by controlling access to funds of any kind. They might prevent their partner from getting a job or withhold access to money they earn from a job. This creates financial dependency that makes leaving the relationship very difficult. [See destruction of livelihoods and the proposed meagre 'guaranteed income' so long as you do whatever you are told.]

Using children: An abuser might disparage their partner's parenting skills, tell their children lies about their partner, threaten to take custody of their children, or threaten to harm their children. These tactics instil fear and often elicit compliance. [See reframed social service mafia and how children are being mercilessly abused by the state over 'Covid' while their parents look on too frightened to do anything.]

A further recurring trait in an abusive relationship is the abused blaming themselves for their abuse and making excuses for the abuser. We have the public blaming each other for lockdown abuse by government and many making excuses for the government while attacking those who challenge the government. How often we have heard authorities say that rules are being imposed or reimposed only because people have refused to 'behave' and follow the rules. We don't want to do it – it's *you*.

Renegade Minds are an antidote to all of these things. They will never concede their self-respect no matter what the circumstances. Even when apparent humiliation is heaped upon them they laugh in its face and reflect back the humiliation on the abuser where it belongs. Renegade Minds will never wear masks they know are only imposed to humiliate, suppress and damage both physically and psychologically. Consequences will take care of themselves and they will never break their spirit or cause them to concede to tyranny. UK newspaper columnist Peter Hitchens was one of the few in the mainstream media to speak out against lockdowns and forced vaccinations. He then announced he had taken the jab. He wanted to see family members abroad and he believed vaccine passports were inevitable even though they had not yet been introduced. Hitchens

has a questioning and critical mind, but not a Renegade one. If he had no amount of pressure would have made him concede. Hitchens excused his action by saying that the battle has been lost. Renegade Minds never accept defeat when freedom is at stake and even if they are the last one standing the self-respect of not submitting to tyranny is more important than any outcome or any consequence.

That's why Renegade Minds are the only minds that ever changed anything worth changing.

CHAPTER EIGHT

'Reframing' insanity

Insanity is relative. It depends on who has who locked in what cage

Ray Bradbury

'Reframing' a mind means simply to change its perception and behaviour. This can be done subconsciously to such an extent that subjects have no idea they have been 'reframed' while to any observer changes in behaviour and attitudes are obvious.

Human society is being reframed on a ginormous scale since the start of 2020 and here we have the reason why psychologists rather than doctors have been calling the shots. Ask most people who have succumbed to 'Covid' reframing if they have changed and most will say 'no'; but they *have* and fundamentally. The Cult's long-game has been preparing for these times since way back and crucial to that has been to prepare both population and officialdom mentally and emotionally. To use the mind-control parlance they had to reframe the population with a mentality that would submit to fascism and reframe those in government and law enforcement to impose fascism or at least go along with it. The result has been the fact-deleted mindlessness of 'Wokeness' and officialdom that has either enthusiastically or unquestioningly imposed global tyranny demanded by reframed politicians on behalf of psychopathic and deeply evil cultists. 'Cognitive reframing' identifies and challenges the way someone sees the world in the form of situations, experiences and emotions and then restructures those perceptions to view the same set of circumstances in a different way. This can have

benefits if the attitudes are personally destructive while on the other side it has the potential for individual and collective mind control which the subject has no idea has even happened.

Cognitive therapy was developed in the 1960s by Aaron T. Beck who was born in Rhode Island in 1921 as the son of Jewish immigrants from the Ukraine. He became interested in the techniques as a treatment for depression. Beck's daughter Judith S. Beck is prominent in the same field and they founded the Beck Institute for Cognitive Behavior Therapy in Philadelphia in 1994. Cognitive reframing, however, began to be used worldwide by those with a very dark agenda. The Cult reframes politicians to change their attitudes and actions until they are completely at odds with what they once appeared to stand for. The same has been happening to government administrators at all levels, law enforcement, military and the human population. Cultists love mind control for two main reasons: It allows them to control what people think, do and say to secure agenda advancement and, by definition, it calms their legendary insecurity and fear of the unexpected. I have studied mind control since the time I travelled America in 1996. I may have been talking to next to no one in terms of an audience in those years, but my goodness did I gather a phenomenal amount of information and knowledge about so many things including the techniques of mind control. I have described this in detail in other books going back to *The Biggest Secret* in 1998. I met a very large number of people recovering from MKUltra and its offshoots and successors and I began to see how these same techniques were being used on the population in general. This was never more obvious than since the 'Covid' hoax began.

Reframing the enforcers

I have observed over the last two decades and more the very clear transformation in the dynamic between the police, officialdom and the public. I tracked this in the books as the relationship mutated from one of serving the public to seeing them as almost the enemy and certainly a lower caste. There has always been a class divide

based on income and always been some psychopathic, corrupt, and big-I-am police officers. This was different. Wholesale change was unfolding in the collective dynamic; it was less about money and far more about position and perceived power. An us-and-them was emerging. Noses were lifted skyward by government administration and law enforcement and their attitude to the public they were *supposed* to be serving changed to one of increasing contempt, superiority and control. The transformation was so clear and widespread that it had to be planned. Collective attitudes and dynamics do not change naturally and organically that quickly on that scale. I then came across an organisation in Britain called Common Purpose created in the late 1980s by Julia Middleton who would work in the office of Deputy Prime Minister John Prescott during the long and disastrous premiership of war criminal Tony Blair. When Blair speaks the Cult is speaking and the man should have been in jail a long time ago. Common Purpose proclaims itself to be one of the biggest 'leadership development' organisations in the world while functioning as a *charity* with all the financial benefits which come from that. It hosts 'leadership development' courses and programmes all over the world and claims to have 'brought together' what it calls 'leaders' from more than 100 countries on six continents. The modus operandi of Common Purpose can be compared with the work of the UK government's reframing network that includes the Behavioural Insights Team 'nudge unit' and 'Covid' reframing specialists at SPI-B. WikiLeaks described Common Purpose long ago as 'a hidden virus in our government and schools' which is unknown to the general public: 'It recruits and trains "leaders" to be loyal to the directives of Common Purpose and the EU, instead of to their own departments, which they then undermine or subvert, the NHS [National Health Service] being an example.' This is a vital point to understand the 'Covid' hoax. The NHS, and its equivalent around the world, has been utterly reframed in terms of administrators and much of the medical personnel with the transformation underpinned by recruitment policies. The outcome has been the criminal and psychopathic behaviour of the

NHS over ‘Covid’ and we have seen the same in every other major country. WikiLeaks said Common Purpose trainees are ‘learning to rule without regard to democracy’ and to usher in a police state (current events explained). Common Purpose operated like a ‘glue’ and had members in the NHS, BBC, police, legal profession, church, many of Britain’s 7,000 quangos, local councils, the Civil Service, government ministries and Parliament, and controlled many RDA’s (Regional Development Agencies). Here we have one answer for how and why British institutions and their like in other countries have changed so negatively in relation to the public. This further explains how and why the beyond-disgraceful reframed BBC has become a propaganda arm of ‘Covid’ fascism. They are all part of a network pursuing the same goal.

By 2019 Common Purpose was quoting a figure of 85,000 ‘leaders’ that had attended its programmes. These ‘students’ of all ages are known as Common Purpose ‘graduates’ and they consist of government, state and local government officials and administrators, police chiefs and officers, and a whole range of others operating within the national, local and global establishment. Cressida Dick, Commissioner of the London Metropolitan Police, is the Common Purpose graduate who was the ‘Gold Commander’ that oversaw what can only be described as the murder of Brazilian electrician Jean Charles de Menezes in 2005. He was held down by psychopathic police and shot seven times in the head by a psychopathic lunatic after being mistaken for a terrorist when he was just a bloke going about his day. Dick authorised officers to pursue and keep surveillance on de Menezes and ordered that he be stopped from entering the underground train system. Police psychopaths took her at her word clearly. She was ‘disciplined’ for this outrage by being *promoted* – eventually to the top of the ‘Met’ police where she has been a disaster. Many Chief Constables controlling the police in different parts of the UK are and have been Common Purpose graduates. I have heard the ‘graduate’ network described as a sort of Mafia or secret society operating within the fabric of government at all levels pursuing a collective policy

ingrained at Common Purpose training events. Founder Julia Middleton herself has said:

Locally and internationally, Common Purpose graduates will be 'lighting small fires' to create change in their organisations and communities ... The Common Purpose effect is best illustrated by the many stories of small changes brought about by leaders, who themselves have changed.

A Common Purpose mission statement declared:

Common Purpose aims to improve the way society works by expanding the vision, decision-making ability and influence of all kinds of leaders. The organisation runs a variety of educational programmes for leaders of all ages, backgrounds and sectors, in order to provide them with the inspirational, information and opportunities they need to change the world.

Yes, but into what? Since 2020 the answer has become clear.

NLP and the Delphi technique

Common Purpose would seem to be a perfect name or would common programming be better? One of the foundation methods of reaching 'consensus' (group think) is by setting the agenda theme and then encouraging, cajoling or pressuring everyone to agree a 'consensus' in line with the core theme promoted by Common Purpose. The methodology involves the 'Delphi technique', or an adaption of it, in which opinions are expressed that are summarised by a 'facilitator or change agent' at each stage. Participants are 'encouraged' to modify their views in the light of what others have said. Stage by stage the former individual opinions are merged into group consensus which just happens to be what Common Purpose wants them to believe. A key part of this is to marginalise anyone refusing to concede to group think and turn the group against them to apply pressure to conform. We are seeing this very technique used on the general population to make 'Covid' group-thinkers hostile to those who have seen through the bullshit. People can be reframed by using perception manipulation methods such as Neuro-Linguistic Programming (NLP) in which you change perception with the use of

carefully constructed language. An NLP website described the technique this way:

... A method of influencing brain behaviour (the 'neuro' part of the phrase) through the use of language (the 'linguistic' part) and other types of communication to enable a person to 'recode' the way the brain responds to stimuli (that's the 'programming') and manifest new and better behaviours. Neuro-Linguistic Programming often incorporates hypnosis and self-hypnosis to help achieve the change (or 'programming') that is wanted.

British alternative media operation UKColumn has done very detailed research into Common Purpose over a long period. I quoted co-founder and former naval officer Brian Gerrish in my book *Remember Who You Are*, published in 2011, as saying the following years before current times:

It is interesting that many of the mothers who have had children taken by the State speak of the Social Services people being icily cool, emotionless and, as two ladies said in slightly different words, '... like little robots'. We know that NLP is cumulative, so people can be given small imperceptible doses of NLP in a course here, another in a few months, next year etc. In this way, major changes are accrued in their personality, but the day by day change is almost unnoticeable.

In these and other ways 'graduates' have had their perceptions uniformly reframed and they return to their roles in the institutions of government, law enforcement, legal profession, military, 'education', the UK National Health Service and the whole swathe of the establishment structure to pursue a common agenda preparing for the 'post-industrial', 'post-democratic' society. I say 'preparing' but we are now there. 'Post-industrial' is code for the Great Reset and 'post-democratic' is 'Covid' fascism. UKColumn has spoken to partners of those who have attended Common Purpose 'training'. They have described how personalities and attitudes of 'graduates' changed very noticeably for the worse by the time they had completed the course. They had been 'reframed' and told they are the 'leaders' – the special ones – who know better than the population. There has also been the very demonstrable recruitment of psychopaths and narcissists into government administration at all

levels and law enforcement. If you want psychopathy hire psychopaths and you get a simple cause and effect. If you want administrators, police officers and 'leaders' to perceive the public as lesser beings who don't matter then employ narcissists. These personalities are identified using 'psychometrics' that identifies knowledge, abilities, attitudes and personality traits, mostly through carefully-designed questionnaires and tests. As this policy has passed through the decades we have had power-crazy, power-trippers appointed into law enforcement, security and government administration in preparation for current times and the dynamic between public and law enforcement/officialdom has been transformed. UKColumn's Brian Gerrish said of the narcissistic personality:

Their love of themselves and power automatically means that they will crush others who get in their way. I received a major piece of the puzzle when a friend pointed out that when they made public officials re-apply for their own jobs several years ago they were also required to do psychometric tests. This was undoubtedly the start of the screening process to get 'their' sort of people in post.

How obvious that has been since 2020 although it was clear what was happening long before if people paid attention to the changing public-establishment dynamic.

Change agents

At the centre of events in 'Covid' Britain is the National Health Service (NHS) which has behaved disgracefully in slavishly following the Cult agenda. The NHS management structure is awash with Common Purpose graduates or 'change agents' working to a common cause. Helen Bevan, a Chief of Service Transformation at the NHS Institute for Innovation and Improvement, co-authored a document called 'Towards a million change agents, a review of the social movements literature: implications for large scale change in the NHS'. The document compared a project management approach to that of change and social movements where 'people change

themselves and each other – peer to peer’. Two definitions given for a ‘social movement’ were:

A group of people who consciously attempt to build a radically new social order; involves people of a broad range of social backgrounds; and deploys politically confrontational and socially disruptive tactics – Cyrus Zirakzadeh 1997

Collective challenges, based on common purposes and social solidarities, in sustained interaction with elites, opponents, and authorities – Sidney Tarrow 1994

Helen Bevan wrote another NHS document in which she defined ‘framing’ as ‘the process by which leaders construct, articulate and put across their message in a powerful and compelling way in order to win people to their cause and call them to action’. I think I could come up with another definition that would be rather more accurate. The National Health Service and institutions of Britain and the wider world have been taken over by reframed ‘change agents’ and that includes everything from the United Nations to national governments, local councils and social services which have been kidnapping children from loving parents on an extraordinary and gathering scale on the road to the end of parenthood altogether. Children from loving homes are stolen and kidnapped by the state and put into the ‘care’ (inversion) of the local authority through council homes, foster parents and forced adoption. At the same time children are allowed to be abused without response while many are under council ‘care’. UKColumn highlighted the Common Purpose connection between South Yorkshire Police and Rotherham council officers in the case of the scandal in that area of the sexual exploitation of children to which the authorities turned not one blind eye, but both:

We were alarmed to discover that the Chief Executive, the Strategic Director of Children and Young People's Services, the Manager for the Local Strategic Partnership, the Community Cohesion Manager, the Cabinet Member for Cohesion, the Chief Constable and his predecessor had all attended Leadership training courses provided by the pseudo-charity Common Purpose.

Once 'change agents' have secured positions of hire and fire within any organisation things start to move very quickly. Personnel are then hired and fired on the basis of whether they will work towards the agenda the change agent represents. If they do they are rapidly promoted even though they may be incompetent. Those more qualified and skilled who are pre-Common Purpose 'old school' see their careers stall and even disappear. This has been happening for decades in every institution of state, police, 'health' and social services and all of them have been transformed as a result in their attitudes to their jobs and the public. Medical professions, including nursing, which were once vocations for the caring now employ many cold, callous and couldn't give a shit personality types. The UKColumn investigation concluded:

By blurring the boundaries between people, professions, public and private sectors, responsibility and accountability, Common Purpose encourages 'graduates' to believe that as new selected leaders, they can work together, outside of the established political and social structures, to achieve a paradigm shift or CHANGE – so called 'Leading Beyond Authority'. In doing so, the allegiance of the individual becomes 'reframed' on CP colleagues and their NETWORK.

Reframing the Face-Nappies

Nowhere has this process been more obvious than in the police where recruitment of psychopaths and development of unquestioning mind-controlled group-thinkers have transformed law enforcement into a politically-correct 'Woke' joke and a travesty of what should be public service. Today they wear their face-nappies like good little gofers and enforce 'Covid' rules which are fascism under another name. Alongside the specifically-recruited psychopaths we have software minds incapable of free thought. Brian Gerrish again:

An example is the policeman who would not get on a bike for a press photo because he had not done the cycling proficiency course. Normal people say this is political correctness gone mad. Nothing could be further from the truth. The policeman has been reframed, and in his reality it is perfect common sense not to get on the bike ‘because he hasn’t done the cycling course’.

Another example of this is where the police would not rescue a boy from a pond until they had taken advice from above on the ‘risk assessment’. A normal person would have arrived, perhaps thought of the risk for a moment, and dived in. To the police now ‘reframed’, they followed ‘normal’ procedure.

There are shocking cases of reframed ambulance crews doing the same. Sheer unthinking stupidity of London Face-Nappies headed by Common Purpose graduate Cressida Dick can be seen in their behaviour at a vigil in March, 2021, for a murdered woman, Sarah Everard. A police officer had been charged with the crime. Anyone with a brain would have left the vigil alone in the circumstances. Instead they ‘manhandled’ women to stop them breaking ‘Covid rules’ to betray classic reframing. Minds in the thrall of perception control have no capacity for seeing a situation on its merits and acting accordingly. ‘Rules is rules’ is their only mind-set. My father used to say that rules and regulations are for the guidance of the intelligent and the blind obedience of the idiot. Most of the intelligent, decent, coppers have gone leaving only the other kind and a few old school for whom the job must be a daily nightmare. The combination of psychopaths and rule-book software minds has been clearly on public display in the ‘Covid’ era with automaton robots in uniform imposing fascistic ‘Covid’ regulations on the population without any personal initiative or judging situations on their merits. There are thousands of examples around the world, but I’ll make my point with the infamous Derbyshire police in the English East Midlands – the ones who think pouring dye into beauty spots and using drones to track people walking in the countryside away from anyone is called ‘policing’. To them there are rules decreed by the government which they have to enforce and in their bewildered state a group gathering in a closed space and someone walking alone in the countryside are the same thing. It is beyond idiocy and enters the realm of clinical insanity.

Police officers in Derbyshire said they were ‘horrified’ – *horrified* – to find 15 to 20 ‘irresponsible’ kids playing a football match at a closed leisure centre ‘in breach of coronavirus restrictions’. When they saw the police the kids ran away leaving their belongings behind and the reframed men and women of Derbyshire police were seeking to establish their identities with a view to fining their parents. The most natural thing for youngsters to do – kicking a ball about – is turned into a criminal activity and enforced by the moronic software programs of Derbyshire police. You find the same mentality in every country. These barely conscious ‘horrified’ officers said they had to take action because ‘we need to ensure these rules are being followed’ and ‘it is of the utmost importance that you ensure your children are following the rules and regulations for Covid-19’. Had any of them done ten seconds of research to see if this parroting of their masters’ script could be supported by any evidence? Nope. Reframed people don’t think – others think for them and that’s the whole idea of reframing. I have seen police officers one after the other repeating without question word for word what officialdom tells them just as I have seen great swathes of the public doing the same. Ask either for ‘their’ opinion and out spews what they have been told to think by the official narrative. Police and public may seem to be in different groups, but their mentality is the same. Most people do whatever they are told in fear not doing so or because they believe what officialdom tells them; almost the entirety of the police do what they are told for the same reason. Ultimately it’s the tiny inner core of the global Cult that’s telling both what to do.

So Derbyshire police were ‘horrified’. Oh, really? Why did they think those kids were playing football? It was to relieve the psychological consequences of lockdown and being denied human contact with their friends and interaction, touch and discourse vital to human psychological health. Being denied this month after month has dismantled the psyche of many children and young people as depression and suicide have exploded. Were Derbyshire police *horrified by that?* Are you kidding? Reframed people don’t have those

mental and emotional processes that can see how the impact on the psychological health of youngsters is far more dangerous than any 'virus' even if you take the mendacious official figures to be true. The reframed are told (programmed) how to act and so they do. The Derbyshire Chief Constable in the first period of lockdown when the black dye and drones nonsense was going on was Peter Goodman. He was the man who severed the connection between his force and the Derbyshire Constabulary *Male Voice* Choir when he decided that it was not inclusive enough to allow women to join. The fact it was a male voice choir making a particular sound produced by male voices seemed to elude a guy who terrifyingly ran policing in Derbyshire. He retired weeks after his force was condemned as disgraceful by former Supreme Court Justice Jonathan Sumption for their behaviour over extreme lockdown impositions. Goodman was replaced by his deputy Rachel Swann who was in charge when her officers were 'horrified'. The police statement over the boys committing the hanging-offence of playing football included the line about the youngsters being 'irresponsible in the times we are all living through' missing the point that the real relevance of the 'times we are all living through' is the imposition of fascism enforced by psychopaths and reframed minds of police officers playing such a vital part in establishing the fascist tyranny that their own children and grandchildren will have to live in their entire lives. As a definition of insanity that is hard to beat although it might be run close by imposing masks on people that can have a serious effect on their health while wearing a face nappy all day themselves. Once again public and police do it for the same reason – the authorities tell them to and who are they to have the self-respect to say no?

Wokers in uniform

How reframed do you have to be to arrest a *six-year-old* and take him to court for *picking a flower* while waiting for a bus? Brain dead police and officialdom did just that in North Carolina where criminal proceedings happen regularly for children under nine. Attorney Julie Boyer gave the six-year-old crayons and a colouring book

during the ‘flower’ hearing while the ‘adults’ decided his fate. County Chief District Court Judge Jay Corpening asked: ‘Should a child that believes in Santa Claus, the Easter Bunny and the tooth fairy be making life-altering decisions?’ Well, of course not, but common sense has no meaning when you have a common purpose and a reframed mind. Treating children in this way, and police operating in American schools, is all part of the psychological preparation for children to accept a police state as normal all their adult lives. The same goes for all the cameras and biometric tracking technology in schools. Police training is focused on reframing them as snowflake Wokers and this is happening in the military. Pentagon top brass said that ‘training sessions on extremism’ were needed for troops who asked why they were so focused on the Capitol Building riot when Black Lives Matter riots were ignored. What’s the difference between them some apparently and rightly asked. Actually, there is a difference. Five people died in the Capitol riot, only one through violence, and that was a police officer shooting an unarmed protestor. BLM riots killed at least 25 people and cost billions. Asking the question prompted the psychopaths and reframed minds that run the Pentagon to say that more ‘education’ (programming) was needed. Troop training is all based on psychological programming to make them fodder for the Cult – ‘Military men are just dumb, stupid animals to be used as pawns in foreign policy’ as Cult-to-his-DNA former Secretary of State Henry Kissinger famously said. Governments see the police in similar terms and it’s time for those among them who can see this to defend the people and stop being enforcers of the Cult agenda upon the people.

The US military, like the country itself, is being targeted for destruction through a long list of Woke impositions. Cult-owned gaga ‘President’ Biden signed an executive order when he took office to allow taxpayer money to pay for transgender surgery for active military personnel and veterans. Are you a man soldier? No, I’m a LGBTQIA+ with a hint of Skoliosexual and Spectrasexual. Oh, good man. Bad choice of words you bigot. The Pentagon announced in March, 2021, the appointment of the first ‘diversity and inclusion

officer' for US Special Forces. Richard Torres-Estrada arrived with the publication of a 'D&I Strategic Plan which will guide the enterprise-wide effort to institutionalize and sustain D&I'. If you think a Special Forces 'Strategic Plan' should have something to do with defending America you haven't been paying attention.

Defending Woke is now the military's new role. Torres-Estrada has posted images comparing Donald Trump with Adolf Hitler and we can expect no bias from him as a representative of the supposedly non-political Pentagon. Cable news host Tucker Carlson said: 'The Pentagon is now the Yale faculty lounge but with cruise missiles.' Meanwhile Secretary of Defense Lloyd Austin, a board member of weapons-maker Raytheon with stock and compensation interests in October, 2020, worth \$1.4 million, said he was purging the military of the 'enemy within' – anyone who isn't Woke and supports Donald Trump. Austin refers to his targets as 'racist extremists' while in true Woke fashion being himself a racist extremist. Pentagon documents pledge to 'eradicate, eliminate and conquer all forms of racism, sexism and homophobia'. The definitions of these are decided by 'diversity and inclusion committees' peopled by those who see racism, sexism and homophobia in every situation and opinion. Woke (the Cult) is dismantling the US military and purging testosterone as China expands its military and gives its troops 'masculinity training'. How do we think that is going to end when this is all Cult coordinated? The US military, like the British military, is controlled by Woke and spineless top brass who just go along with it out of personal career interests.

'Woke' means fast asleep

Mind control and perception manipulation techniques used on individuals to create group-think have been unleashed on the global population in general. As a result many have no capacity to see the obvious fascist agenda being installed all around them or what 'Covid' is really all about. Their brains are firewalled like a computer system not to process certain concepts, thoughts and realisations that are bad for the Cult. The young are most targeted as the adults they

will be when the whole fascist global state is planned to be fully implemented. They need to be prepared for total compliance to eliminate all pushback from entire generations. The Cult has been pouring billions into taking complete control of 'education' from schools to universities via its operatives and corporations and not least Bill Gates as always. The plan has been to transform 'education' institutions into programming centres for the mentality of 'Woke'. James McConnell, professor of psychology at the University of Michigan, wrote in *Psychology Today* in 1970:

The day has come when we can combine sensory deprivation with drugs, hypnosis, and astute manipulation of reward and punishment, to gain almost absolute control over an individual's behaviour. It should then be possible to achieve a very rapid and highly effective type of brainwashing that would allow us to make dramatic changes in a person's behaviour and personality ...

... We should reshape society so that we all would be trained from birth to want to do what society wants us to do. We have the techniques to do it... no-one owns his own personality you acquired, and there's no reason to believe you should have the right to refuse to acquire a new personality if your old one is anti-social.

This was the potential for mass brainwashing in 1970 and the mentality there displayed captures the arrogant psychopathy that drives it forward. I emphasise that not all young people have succumbed to Woke programming and those that haven't are incredibly impressive people given that today's young are the most perceptually-targeted generations in history with all the technology now involved. Vast swathes of the young generations, however, have fallen into the spell – and that's what it is – of Woke. The Woke mentality and perceptual program is founded on *inversion* and you will appreciate later why that is so significant. Everything with Woke is inverted and the opposite of what it is claimed to be. Woke was a term used in African-American culture from the 1900s and referred to an awareness of social and racial justice. This is not the meaning of the modern version or 'New Woke' as I call it in *The Answer*. Oh, no, Woke today means something very different no matter how much Wokers may seek to hide that and insist Old Woke and New

Woke are the same. See if you find any 'awareness of social justice' here in the modern variety:

- Woke demands 'inclusivity' while excluding anyone with a different opinion and calls for mass censorship to silence other views.
- Woke claims to stand against oppression when imposing oppression is the foundation of all that it does. It is the driver of political correctness which is nothing more than a Cult invention to manipulate the population to silence itself.
- Woke believes itself to be 'liberal' while pursuing a global society that can only be described as fascist (see 'anti-fascist' fascist Antifa).
- Woke calls for 'social justice' while spreading injustice wherever it goes against the common 'enemy' which can be easily identified as a differing view.
- Woke is supposed to be a metaphor for 'awake' when it is solid-gold asleep and deep in a Cult-induced coma that meets the criteria for 'off with the fairies'.

I state these points as obvious facts if people only care to look. I don't do this with a sense of condemnation. We need to appreciate that the onslaught of perceptual programming on the young has been incessant and merciless. I can understand why so many have been reframed, or, given their youth, framed from the start to see the world as the Cult demands. The Cult has had access to their minds day after day in its 'education' system for their entire formative years. Perception is formed from information received and the Cult-created system is a life-long download of information delivered to elicit a particular perception, thus behaviour. The more this has expanded into still new extremes in recent decades and ever-increasing censorship has deleted other opinions and information why wouldn't that lead to a perceptual reframing on a mass scale? I

have described already cradle-to-grave programming and in more recent times the targeting of young minds from birth to adulthood has entered the stratosphere. This has taken the form of skewing what is ‘taught’ to fit the Cult agenda and the omnipresent techniques of group-think to isolate non-believers and pressure them into line. There has always been a tendency to follow the herd, but we really are in a new world now in relation to that. We have parents who can see the ‘Covid’ hoax told by their children not to stop them wearing masks at school, being ‘Covid’ tested or having the ‘vaccine’ in fear of the peer-pressure consequences of being different. What is ‘peer-pressure’ if not pressure to conform to group-think? Renegade Minds never group-think and always retain a set of perceptions that are unique to them. Group-think is always underpinned by consequences for not group-thinking. Abuse now aimed at those refusing DNA-manipulating ‘Covid vaccines’ are a potent example of this. The biggest pressure to conform comes from the very group which is itself being manipulated. ‘I am programmed to be part of a hive mind and so you must be.’

Woke control structures in ‘education’ now apply to every mainstream organisation. Those at the top of the ‘education’ hierarchy (the Cult) decide the policy. This is imposed on governments through the Cult network; governments impose it on schools, colleges and universities; their leadership impose the policy on teachers and academics and they impose it on children and students. At any level where there is resistance, perhaps from a teacher or university lecturer, they are targeted by the authorities and often fired. Students themselves regularly demand the dismissal of academics (increasingly few) at odds with the narrative that the students have been programmed to believe in. It is quite a thought that students who are being targeted by the Cult become so consumed by programmed group-think that they launch protests and demand the removal of those who are trying to push back against those targeting the students. Such is the scale of perceptual inversion. We see this with ‘Covid’ programming as the Cult imposes the rules via psycho-psychologists and governments on

shops, transport companies and businesses which impose them on their staff who impose them on their customers who pressure Pushbackers to conform to the will of the Cult which is in the process of destroying them and their families. Scan all aspects of society and you will see the same sequence every time.

Fact free Woke and hijacking the 'left'

There is no more potent example of this than 'Woke', a mentality only made possible by the deletion of factual evidence by an 'education' system seeking to produce an ever more uniform society. Why would you bother with facts when you don't know any? Deletion of credible history both in volume and type is highly relevant. Orwell said: 'Who controls the past controls the future: who controls the present controls the past.' They who control the perception of the past control the perception of the future and they who control the present control the perception of the past through the writing and deleting of history. Why would you oppose the imposition of Marxism in the name of Wokeism when you don't know that Marxism cost at least 100 million lives in the 20th century alone? Watch videos and read reports in which Woker generations are asked basic historical questions – it's mind-blowing. A survey of 2,000 people found that six percent of millennials (born approximately early 1980s to early 2000s) believed the Second World War (1939-1945) broke out with the assassination of President Kennedy (in 1963) and one in ten thought Margaret Thatcher was British Prime Minister at the time. She was in office between 1979 and 1990. We are in a post-fact society. Provable facts are no defence against the fascism of political correctness or Silicon Valley censorship. Facts don't matter anymore as we have witnessed with the 'Covid' hoax. Sacrificing uniqueness to the Woke group-think religion is all you are required to do and that means thinking for yourself is the biggest Woke no, no. All religions are an expression of group-think and censorship and Woke is just another religion with an orthodoxy defended by group-think and censorship. Burned at

the stake becomes burned on Twitter which leads back eventually to burned at the stake as Woke humanity regresses to ages past.

The biggest Woke inversion of all is its creators and funders. I grew up in a traditional left of centre political household on a council estate in Leicester in the 1950s and 60s – you know, the left that challenged the power of wealth-hoarding elites and threats to freedom of speech and opinion. In those days students went on marches defending freedom of speech while today's Wokers march for its deletion. What on earth could have happened? Those very elites (collectively the Cult) that we opposed in my youth and early life have funded into existence the antithesis of that former left and hijacked the 'brand' while inverting everything it ever stood for. We have a mentality that calls itself 'liberal' and 'progressive' while acting like fascists. Cult billionaires and their corporations have funded themselves into control of 'education' to ensure that Woke programming is unceasing throughout the formative years of children and young people and that non-Wokers are isolated (that word again) whether they be students, teachers or college professors. The Cult has funded into existence the now colossal global network of Woke organisations that have spawned and promoted all the 'causes' on the Cult wish-list for global transformation and turned Wokers into demanders of them. Does anyone really think it's a coincidence that the Cult agenda for humanity is a carbon (sorry) copy of the societal transformations desired by Woke?? These are only some of them:

Political correctness: The means by which the Cult deletes all public debates that it knows it cannot win if we had the free-flow of information and evidence.

Human-caused 'climate change': The means by which the Cult seeks to transform society into a globally-controlled dictatorship imposing its will over the fine detail of everyone's lives 'to save the planet' which doesn't actually need saving.

Transgender obsession: Preparing collective perception to accept the ‘new human’ which would not have genders because it would be created technologically and not through procreation. I’ll have much more on this in Human 2.0.

Race obsession: The means by which the Cult seeks to divide and rule the population by triggering racial division through the perception that society is more racist than ever when the opposite is the case. Is it perfect in that regard? No. But to compare today with the racism of apartheid and segregation brought to an end by the civil rights movement in the 1960s is to insult the memory of that movement and inspirations like Martin Luther King. Why is the ‘anti-racism’ industry (which it is) so dominated by privileged white people?

White supremacy: This is a label used by privileged white people to demonise poor and deprived white people pushing back on tyranny to marginalise and destroy them. White people are being especially targeted as the dominant race by number within Western society which the Cult seeks to transform in its image. If you want to change a society you must weaken and undermine its biggest group and once you have done that by using the other groups you next turn on them to do the same ... ‘Then they came for the Jews and I was not a Jew so I did nothing.’

Mass migration: The mass movement of people from the Middle East, Africa and Asia into Europe, from the south into the United States and from Asia into Australia are another way the Cult seeks to dilute the racial, cultural and political influence of white people on Western society. White people ask why their governments appear to be working against them while being politically and culturally biased towards incoming cultures. Well, here’s your answer. In the same way sexually ‘straight’ people, men and women, ask why the

authorities are biased against them in favour of other sexualities. The answer is the same – that's the way the Cult wants it to be for very sinister motives.

These are all central parts of the Cult agenda and central parts of the Woke agenda and Woke was created and continues to be funded to an immense degree by Cult billionaires and corporations. If anyone begins to say 'coincidence' the syllables should stick in their throat.

Billionaire 'social justice warriors'

Joe Biden is a 100 percent-owned asset of the Cult and the Wokers' man in the White House whenever he can remember his name and for however long he lasts with his rapidly diminishing cognitive function. Even walking up the steps of an aircraft without falling on his arse would appear to be a challenge. He's not an empty-shell puppet or anything. From the minute Biden took office (or the Cult did) he began his executive orders promoting the Woke wish-list. You will see the Woke agenda imposed ever more severely because it's really the *Cult* agenda. Woke organisations and activist networks spawned by the Cult are funded to the extreme so long as they promote what the Cult wants to happen. Woke is funded to promote 'social justice' by billionaires who become billionaires by destroying social justice. The social justice mantra is only a cover for dismantling social justice and funded by billionaires that couldn't give a damn about social justice. Everything makes sense when you see that. One of Woke's premier funders is Cult billionaire financier George Soros who said: 'I am basically there to make money, I cannot and do not look at the social consequences of what I do.' This is the same Soros who has given more than \$32 billion to his Open Society Foundations global Woke network and funded Black Lives Matter, mass immigration into Europe and the United States, transgender activism, climate change activism, political correctness and groups targeting 'white supremacy' in the form of privileged white thugs that dominate Antifa. What a scam it all is and when

you are dealing with the unquestioning fact-free zone of Woke scamming them is child's play. All you need to pull it off in all these organisations are a few in-the-know agents of the Cult and an army of naïve, reframed, uninformed, narcissistic, know-nothings convinced of their own self-righteousness, self-purity and virtue.

Soros and fellow billionaires and billionaire corporations have poured hundreds of millions into Black Lives Matter and connected groups and promoted them to a global audience. None of this is motivated by caring about black people. These are the billionaires that have controlled and exploited a system that leaves millions of black people in abject poverty and deprivation which they do absolutely nothing to address. The same Cult networks funding BLM were behind the *slave trade!* Black Lives Matter hijacked a phrase that few would challenge and they have turned this laudable concept into a political weapon to divide society. You know that BLM is a fraud when it claims that *All Lives Matter*, the most inclusive statement of all, is 'racist'. BLM and its Cult masters don't want to end racism. To them it's a means to an end to control all of humanity never mind the colour, creed, culture or background. What has destroying the nuclear family got to do with ending racism? Nothing – but that is one of the goals of BLM and also happens to be a goal of the Cult as I have been exposing in my books for decades. Stealing children from loving parents and giving schools ever more power to override parents is part of that same agenda. BLM is a Marxist organisation and why would that not be the case when the Cult created Marxism *and* BLM? Patrisse Cullors, a BLM co-founder, said in a 2015 video that she and her fellow organisers, including co-founder Alicia Garza, are 'trained Marxists'. The lady known after marriage as Patrisse Khan-Cullors bought a \$1.4 million home in 2021 in one of the whitest areas of California with a black population of just 1.6 per cent and has so far bought *four* high-end homes for a total of \$3.2 million. How very Marxist. There must be a bit of spare in the BLM coffers, however, when Cult corporations and billionaires have handed over the best part of \$100 million. Many black people can see that Black Lives Matter is not

working for them, but against them, and this is still more confirmation. Black journalist Jason Whitlock, who had his account suspended by Twitter for simply linking to the story about the ‘Marxist’s’ home buying spree, said that BLM leaders are ‘making millions of dollars off the backs of these dead black men who they wouldn’t spit on if they were on fire and alive’.

Black Lies Matter

Cult assets and agencies came together to promote BLM in the wake of the death of career criminal George Floyd who had been jailed a number of times including for forcing his way into the home of a black woman with others in a raid in which a gun was pointed at her stomach. Floyd was filmed being held in a Minneapolis street in 2020 with the knee of a police officer on his neck and he subsequently died. It was an appalling thing for the officer to do, but the same technique has been used by police on peaceful protestors of lockdown without any outcry from the Woke brigade. As unquestioning supporters of the Cult agenda Wokers have supported lockdown and all the ‘Covid’ claptrap while attacking anyone standing up to the tyranny imposed in its name. Court documents would later include details of an autopsy on Floyd by County Medical Examiner Dr Andrew Baker who concluded that Floyd had taken a fatal level of the drug fentanyl. None of this mattered to fact-free, question-free, Woke. Floyd’s death was followed by worldwide protests against police brutality amid calls to defund the police. Throwing babies out with the bathwater is a Woke speciality. In the wake of the murder of British woman Sarah Everard a Green Party member of the House of Lords, Baroness Jones of Moulsecoomb (Nincompoopia would have been better), called for a 6pm curfew for all men. This would be in breach of the Geneva Conventions on war crimes which ban collective punishment, but that would never have crossed the black and white Woke mind of Baroness Nincompoopia who would have been far too convinced of her own self-righteousness to compute such details. Many American cities did defund the police in the face of Floyd riots

and after \$15 million was deleted from the police budget in Washington DC under useless Woke mayor Muriel Bowser car-jacking alone rose by 300 percent and within six months the US capital recorded its highest murder rate in 15 years. The same happened in Chicago and other cities in line with the Cult/Soros plan to bring fear to streets and neighbourhoods by reducing the police, releasing violent criminals and not prosecuting crime. This is the mob-rule agenda that I have warned in the books was coming for so long. Shootings in the area of Minneapolis where Floyd was arrested increased by 2,500 percent compared with the year before. Defunding the police over George Floyd has led to a big increase in dead people with many of them black. Police protection for politicians making these decisions stayed the same or increased as you would expect from professional hypocrites. The Cult doesn't actually want to abolish the police. It wants to abolish local control over the police and hand it to federal government as the psychopaths advance the Hunger Games Society. Many George Floyd protests turned into violent riots with black stores and businesses destroyed by fire and looting across America fuelled by Black Lives Matter. Woke doesn't do irony. If you want civil rights you must loot the liquor store and the supermarket and make off with a smart TV. It's the only way.

It's not a race war – it's a class war

Black people are patronised by privileged blacks and whites alike and told they are victims of white supremacy. I find it extraordinary to watch privileged blacks supporting the very system and bloodline networks behind the slave trade and parroting the same Cult-serving manipulative crap of their privileged white, often billionaire, associates. It is indeed not a race war but a class war and colour is just a diversion. Black Senator Cory Booker and black Congresswoman Maxine Waters, more residents of Nincompoopia, personify this. Once you tell people they are victims of someone else you devalue both their own responsibility for their plight and the power they have to impact on their reality and experience. Instead

we have: 'You are only in your situation because of whitey – turn on them and everything will change.' It won't change. Nothing changes in our lives unless *we* change it. Crucial to that is never seeing yourself as a victim and always as the creator of your reality. Life is a simple sequence of choice and consequence. Make different choices and you create different consequences. *You* have to make those choices – not Black Lives Matter, the Woke Mafia and anyone else that seeks to dictate your life. Who are they these Wokers, an emotional and psychological road traffic accident, to tell you what to do? Personal empowerment is the last thing the Cult and its Black Lives Matter want black people or anyone else to have. They claim to be defending the underdog while *creating* and perpetuating the underdog. The Cult's worst nightmare is human unity and if they are going to keep blacks, whites and every other race under economic servitude and control then the focus must be diverted from what they have in common to what they can be manipulated to believe divides them. Blacks have to be told that their poverty and plight is the fault of the white bloke living on the street in the same poverty and with the same plight they are experiencing. The difference is that your plight black people is due to him, a white supremacist with 'white privilege' living on the street. Don't unite as one human family against your mutual oppressors and suppressors – fight the oppressor with the white face who is as financially deprived as you are. The Cult knows that as its 'Covid' agenda moves into still new levels of extremism people are going to respond and it has been spreading the seeds of disunity everywhere to stop a united response to the evil that targets *all of us*.

Racist attacks on 'whiteness' are getting ever more outrageous and especially through the American Democratic Party which has an appalling history for anti-black racism. Barack Obama, Joe Biden, Hillary Clinton and Nancy Pelosi all eulogised about Senator Robert Byrd at his funeral in 2010 after a nearly 60-year career in Congress. Byrd was a brutal Ku Klux Klan racist and a violent abuser of Cathy O'Brien in MKUltra. He said he would never fight in the military 'with a negro by my side' and 'rather I should die a thousand times,

and see Old Glory trampled in the dirt never to rise again, than to see this beloved land of ours become degraded by race mongrels, a throwback to the blackest specimen from the wilds'. Biden called Byrd a 'very close friend and mentor'. These 'Woke' hypocrites are not anti-racist they are anti-poor and anti-people not of their perceived class. Here is an illustration of the scale of anti-white racism to which we have now descended. Seriously Woke and moronic *New York Times* contributor Damon Young described whiteness as a 'virus' that 'like other viruses will not die until there are no bodies left for it to infect'. He went on: '... the only way to stop it is to locate it, isolate it, extract it, and kill it.' Young can say that as a black man with no consequences when a white man saying the same in reverse would be facing a jail sentence. *That's* racism. We had super-Woke numbskull senators Tammy Duckworth and Mazie Hirono saying they would object to future Biden Cabinet appointments if he did not nominate more Asian Americans and Pacific Islanders. Never mind the ability of the candidate what do they look like? Duckworth said: 'I will vote for racial minorities and I will vote for LGBTQ, but anyone else I'm not voting for.' Appointing people on the grounds of race is illegal, but that was not a problem for this ludicrous pair. They were on-message and that's a free pass in any situation.

Critical race racism

White children are told at school they are intrinsically racist as they are taught the divisive 'critical race theory'. This claims that the law and legal institutions are inherently racist and that race is a socially constructed concept used by white people to further their economic and political interests at the expense of people of colour. White is a 'virus' as we've seen. Racial inequality results from 'social, economic, and legal differences that white people create between races to maintain white interests which leads to poverty and criminality in minority communities'. I must tell that to the white guy sleeping on the street. The principal of East Side Community School in New York sent white parents a manifesto that called on

them to become ‘white traitors’ and advocate for full ‘white abolition’. These people are teaching your kids when they urgently need a psychiatrist. The ‘school’ included a chart with ‘eight white identities’ that ranged from ‘white supremacist’ to ‘white abolition’ and defined the behaviour white people must follow to end ‘the regime of whiteness’. Woke blacks and their privileged white associates are acting exactly like the slave owners of old and Ku Klux Klan racists like Robert Byrd. They are too full of their own self-purity to see that, but it’s true. Racism is not a body type; it’s a state of mind that can manifest through any colour, creed or culture.

Another racial fraud is ‘*equity*’. Not equality of treatment and opportunity – equity. It’s a term spun as equality when it means something very different. Equality in its true sense is a raising up while ‘*equity*’ is a race to the bottom. Everyone in the same level of poverty is ‘*equity*’. Keep everyone down – that’s equity. The Cult doesn’t want anyone in the human family to be empowered and BLM leaders, like all these ‘anti-racist’ organisations, continue their privileged, pampered existence by perpetuating the perception of gathering racism. When is the last time you heard an ‘anti-racist’ or ‘anti-Semitism’ organisation say that acts of racism and discrimination have *fallen*? It’s not in the interests of their fund-raising and power to influence and the same goes for the professional soccer anti-racism operation, Kick It Out. Two things confirmed that the Black Lives Matter riots in the summer of 2020 were Cult creations. One was that while anti-lockdown protests were condemned in this same period for ‘transmitting ‘Covid’ the authorities supported mass gatherings of Black Lives Matter supporters. I even saw self-deluding people claiming to be doctors say the two types of protest were not the same. No – the non-existent ‘Covid’ was in favour of lockdowns and attacked those that protested against them while ‘Covid’ supported Black Lives Matter and kept well away from its protests. The whole thing was a joke and as lockdown protestors were arrested, often brutally, by reframed Face-Nappies we had the grotesque sight of police officers taking the knee to Black Lives Matter, a Cult-funded Marxist

organisation that supports violent riots and wants to destroy the nuclear family and white people.

He's not white? Shucks!

Woke obsession with race was on display again when ten people were shot dead in Boulder, Colorado, in March, 2021. Cult-owned Woke TV channels like CNN said the shooter appeared to be a white man and Wokers were on Twitter condemning 'violent white men' with the usual mantras. Then the shooter's name was released as Ahmad Al Aliwi Alissa, an anti-Trump Arab-American, and the sigh of disappointment could be heard five miles away. Never mind that ten people were dead and what that meant for their families. Race baiting was all that mattered to these sick Cult-serving people like Barack Obama who exploited the deaths to further divide America on racial grounds which is his job for the Cult. This is the man that 'racist' white Americans made the first black president of the United States and then gave him a second term. Not-very-bright Obama has become filthy rich on the back of that and today appears to have a big influence on the Biden administration. Even so he's still a downtrodden black man and a victim of white supremacy. This disingenuous fraud reveals the contempt he has for black people when he puts on a Deep South Alabama accent whenever he talks to them, no, *at* them.

Another BLM red flag was how the now fully-Woke (fully-Cult) and fully-virtue-signalled professional soccer authorities had their teams taking the knee before every match in support of Marxist Black Lives Matter. Soccer authorities and clubs displayed 'Black Lives Matter' on the players' shirts and flashed the name on electronic billboards around the pitch. Any fans that condemned what is a Freemasonic taking-the-knee ritual were widely condemned as you would expect from the Woke virtue-signallers of professional sport and the now fully-Woke media. We have reverse racism in which you are banned from criticising any race or culture except for white people for whom anything goes – say what you like, no problem. What has this got to do with racial harmony and

equality? We've had black supremacists from Black Lives Matter telling white people to fall to their knees in the street and apologise for their white supremacy. Black supremacists acting like white supremacist slave owners of the past couldn't breach their self-obsessed, race-obsessed sense of self-purity. Joe Biden appointed a race-obsessed black supremacist Kristen Clarke to head the Justice Department Civil Rights Division. Clarke claimed that blacks are endowed with 'greater mental, physical and spiritual abilities' than whites. If anyone reversed that statement they would be vilified. Clarke is on-message so no problem. She's never seen a black-white situation in which the black figure is anything but a virtuous victim and she heads the Civil Rights Division which should treat everyone the same or it isn't civil rights. Another perception of the Renegade Mind: If something or someone is part of the Cult agenda they will be supported by Woke governments and media no matter what. If they're not, they will be condemned and censored. It really is that simple and so racist Clarke prospers despite (make that because of) her racism.

The end of culture

Biden's administration is full of such racial, cultural and economic bias as the Cult requires the human family to be divided into warring factions. We are now seeing racially-segregated graduations and everything, but everything, is defined through the lens of perceived 'racism. We have 'racist' mathematics, 'racist' food and even 'racist' *plants*. World famous Kew Gardens in London said it was changing labels on plants and flowers to tell its pre-'Covid' more than two million visitors a year how racist they are. Kew director Richard Deverell said this was part of an effort to 'move quickly to decolonise collections' after they were approached by one Ajay Chhabra 'an actor with an insight into how sugar cane was linked to slavery'. They are *plants* you idiots. 'Decolonisation' in the Woke manual really means colonisation of society with its mentality and by extension colonisation by the Cult. We are witnessing a new Chinese-style 'Cultural Revolution' so essential to the success of all

Marxist takeovers. Our cultural past and traditions have to be swept away to allow a new culture to be built-back-better. Woke targeting of long-standing Western cultural pillars including historical monuments and cancelling of historical figures is what happened in the Mao revolution in China which ‘purged remnants of capitalist and traditional elements from Chinese society’ and installed Maoism as the dominant ideology’. For China see the Western world today and for ‘dominant ideology’ see Woke. Better still see Marxism or Maoism. The ‘Covid’ hoax has specifically sought to destroy the arts and all elements of Western culture from people meeting in a pub or restaurant to closing theatres, music venues, sports stadiums, places of worship and even banning *singing*. Destruction of Western society is also why criticism of any religion is banned except for Christianity which again is the dominant religion as white is the numerically-dominant race. Christianity may be fading rapidly, but its history and traditions are weaved through the fabric of Western society. Delete the pillars and other structures will follow until the whole thing collapses. I am not a Christian defending that religion when I say that. I have no religion. It’s just a fact. To this end Christianity has itself been turned Woke to usher its own downfall and its ranks are awash with ‘change agents’ – knowing and unknowing – at every level including Pope Francis (*definitely* knowing) and the clueless Archbishop of Canterbury Justin Welby (possibly not, but who can be sure?). Woke seeks to coordinate attacks on Western culture, traditions, and ways of life through ‘intersectionality’ defined as ‘the complex, cumulative way in which the effects of multiple forms of discrimination (such as racism, sexism, and classism) combine, overlap, or intersect especially in the experiences of marginalised individuals or groups’. Wade through the Orwellian Woke-speak and this means coordinating disparate groups in a common cause to overthrow freedom and liberal values.

The entire structure of public institutions has been infested with Woke – government at all levels, political parties, police, military, schools, universities, advertising, media and trade unions. This abomination has been achieved through the Cult web by appointing

Wokers to positions of power and battering non-Wokers into line through intimidation, isolation and threats to their job. Many have been fired in the wake of the empathy-deleted, vicious hostility of 'social justice' Wokers and the desire of gutless, spineless employers to virtue-signal their Wokeness. Corporations are filled with Wokers today, most notably those in Silicon Valley. Ironically at the top they are not Woke at all. They are only exploiting the mentality their Cult masters have created and funded to censor and enslave while the Wokers cheer them on until it's their turn. Thus the Woke 'liberal left' is an inversion of the traditional liberal left. Campaigning for justice on the grounds of power and wealth distribution has been replaced by campaigning for identity politics. The genuine traditional left would never have taken money from today's billionaire abusers of fairness and justice and nor would the billionaires have wanted to fund that genuine left. It would not have been in their interests to do so. The division of opinion in those days was between the haves and have nots. This all changed with Cult manipulated and funded identity politics. The division of opinion today is between Wokers and non-Wokers and not income brackets. Cult corporations and their billionaires may have taken wealth disparity to cataclysmic levels of injustice, but as long as they speak the language of Woke, hand out the dosh to the Woke network and censor the enemy they are 'one of us'. Billionaires who don't give a damn about injustice are laughing at them till their bellies hurt. Wokers are not even close to self-aware enough to see that. The transformed 'left' dynamic means that Wokers who drone on about 'social justice' are funded by billionaires that have destroyed social justice the world over. It's *why* they are billionaires.

The climate con

Nothing encapsulates what I have said more comprehensively than the hoax of human-caused global warming. I have detailed in my books over the years how Cult operatives and organisations were the pump-primers from the start of the climate con. A purpose-built vehicle for this is the Club of Rome established by the Cult in 1968

with the Rockefellers and Rothschilds centrally involved all along. Their gofer frontman Maurice Strong, a Canadian oil millionaire, hosted the Earth Summit in Rio de Janeiro, Brazil, in 1992 where the global ‘green movement’ really expanded in earnest under the guiding hand of the Cult. The Earth Summit established Agenda 21 through the Cult-created-and-owned United Nations to use the illusion of human-caused climate change to justify the transformation of global society to save the world from climate disaster. It is a No-Problem-Reaction-Solution sold through governments, media, schools and universities as whole generations have been terrified into believing that the world was going to end in their lifetimes unless what old people had inflicted upon them was stopped by a complete restructuring of how everything is done. Chill, kids, it’s all a hoax. Such restructuring is precisely what the Cult agenda demands (purely by coincidence of course). Today this has been given the codename of the Great Reset which is only an updated term for Agenda 21 and its associated Agenda 2030. The latter, too, is administered through the UN and was voted into being by the General Assembly in 2015. Both 21 and 2030 seek centralised control of all resources and food right down to the raindrops falling on your own land. These are some of the demands of Agenda 21 established in 1992. See if you recognise this society emerging today:

- End national sovereignty
- State planning and management of all land resources, ecosystems, deserts, forests, mountains, oceans and fresh water; agriculture; rural development; biotechnology; and ensuring ‘*equity*’
- The state to ‘define the role’ of business and financial resources
- Abolition of private property
- ‘Restructuring’ the family unit (see BLM)
- Children raised by the state
- People told what their job will be
- Major restrictions on movement
- Creation of ‘human settlement zones’

- Mass resettlement as people are forced to vacate land where they live
- Dumbing down education
- Mass global depopulation in pursuit of all the above

The United Nations was created as a Trojan horse for world government. With the climate con of critical importance to promoting that outcome you would expect the UN to be involved. Oh, it's involved all right. The UN is promoting Agenda 21 and Agenda 2030 justified by 'climate change' while also driving the climate hoax through its Intergovernmental Panel on Climate Change (IPCC), one of the world's most corrupt organisations. The IPCC has been lying ferociously and constantly since the day it opened its doors with the global media hanging unquestioningly on its every mendacious word. The Green movement is entirely Woke and has long lost its original environmental focus since it was co-opted by the Cult. An obsession with 'global warming' has deleted its values and scrambled its head. I experienced a small example of what I mean on a beautiful country walk that I have enjoyed several times a week for many years. The path merged into the fields and forests and you felt at one with the natural world. Then a 'Green' organisation, the Hampshire and Isle of Wight Wildlife Trust, took over part of the land and proceeded to cut down a large number of trees, including mature ones, to install a horrible big, bright steel 'this-is-ours-stay-out' fence that destroyed the whole atmosphere of this beautiful place. No one with a feel for nature would do that. Day after day I walked to the sound of chainsaws and a magnificent mature weeping willow tree that I so admired was cut down at the base of the trunk. When I challenged a Woke young girl in a green shirt (of course) about this vandalism she replied: 'It's a weeping willow – it will grow back.' This is what people are paying for when they donate to the Hampshire and Isle of Wight Wildlife Trust and many other 'green' organisations today. It is not the environmental movement that I knew and instead has become a support-system – as with Extinction Rebellion – for a very dark agenda.

Private jets for climate justice

The Cult-owned, Gates-funded, World Economic Forum and its founder Klaus Schwab were behind the emergence of Greta Thunberg to harness the young behind the climate agenda and she was invited to speak to the world at ... the UN. Schwab published a book, *Covid-19: The Great Reset* in 2020 in which he used the 'Covid' hoax and the climate hoax to lay out a new society straight out of Agenda 21 and Agenda 2030. Bill Gates followed in early 2021 when he took time out from destroying the world to produce a book in his name about the way to save it. Gates flies across the world in private jets and admitted that 'I probably have one of the highest greenhouse gas footprints of anyone on the planet ... my personal flying alone is gigantic.' He has also bid for the planet's biggest private jet operator. Other climate change saviours who fly in private jets include John Kerry, the US Special Presidential Envoy for Climate, and actor Leonardo DiCaprio, a 'UN Messenger of Peace with special focus on climate change'. These people are so full of bullshit they could corner the market in manure. We mustn't be sceptical, though, because the Gates book, *How to Avoid a Climate Disaster: The Solutions We Have and the Breakthroughs We Need*, is a genuine attempt to protect the world and not an obvious pile of excrement attributed to a mega-psychopath aimed at selling his masters' plans for humanity. The Gates book and the other shite-pile by Klaus Schwab could have been written by the same person and may well have been. Both use 'climate change' and 'Covid' as the excuses for their new society and by coincidence the Cult's World Economic Forum and Bill and Melinda Gates Foundation promote the climate hoax and hosted Event 201 which pre-empted with a 'simulation' the very 'coronavirus' hoax that would be simulated for real on humanity within weeks. The British 'royal' family is promoting the 'Reset' as you would expect through Prince 'climate change caused the war in Syria' Charles and his hapless son Prince William who said that we must 'reset our relationship with nature and our trajectory as a species' to avoid a climate disaster. Amazing how many promoters of the 'Covid' and 'climate change' control

systems are connected to Gates and the World Economic Forum. A 'study' in early 2021 claimed that carbon dioxide emissions must fall by the equivalent of a global lockdown roughly every two years for the next decade to save the planet. The 'study' appeared in the same period that the Schwab mob claimed in a video that lockdowns destroying the lives of billions are good because they make the earth 'quieter' with less 'ambient noise'. They took down the video amid a public backlash for such arrogant, empathy-deleted stupidity You see, however, where they are going with this. Corinne Le Quéré, a professor at the Tyndall Centre for Climate Change Research, University of East Anglia, was lead author of the climate lockdown study, and she writes for ... the World Economic Forum. Gates calls in 'his' book for changing 'every aspect of the economy' (long-time Cult agenda) and for humans to eat synthetic 'meat' (predicted in my books) while cows and other farm animals are eliminated.

Australian TV host and commentator Alan Jones described what carbon emission targets would mean for farm animals in Australia alone if emissions were reduced as demanded by 35 percent by 2030 and zero by 2050:

Well, let's take agriculture, the total emissions from agriculture are about 75 million tonnes of carbon dioxide, equivalent. Now reduce that by 35 percent and you have to come down to 50 million tonnes, I've done the maths. So if you take for example 1.5 million cows, you're going to have to reduce the herd by 525,000 [by] 2030, nine years, that's 58,000 cows a year. The beef herd's 30 million, reduce that by 35 percent, that's 10.5 million, which means 1.2 million cattle have to go every year between now and 2030. This is insanity!

There are 75 million sheep. Reduce that by 35 percent, that's 26 million sheep, that's almost 3 million a year. So under the Paris Agreement over 30 million beasts. dairy cows, cattle, pigs and sheep would go. More than 8,000 every minute of every hour for the next decade, do these people know what they're talking about?

Clearly they don't at the level of campaigners, politicians and administrators. The Cult *does* know; that's the outcome it wants. We are faced with not just a war on humanity. Animals and the natural world are being targeted and I have been saying since the 'Covid' hoax began that the plan eventually was to claim that the 'deadly virus' is able to jump from animals, including farm animals and

domestic pets, to humans. Just before this book went into production came this story: 'Russia registers world's first Covid-19 vaccine for cats & dogs as makers of Sputnik V warn pets & farm animals could spread virus'. The report said 'top scientists warned that the deadly pathogen could soon begin spreading through homes and farms' and 'the next stage is the infection of farm and domestic animals'. Know the outcome and you'll see the journey. Think what that would mean for animals and keep your eye on a term called zoonosis or zoonotic diseases which transmit between animals and humans. The Cult wants to break the connection between animals and people as it does between people and people. Farm animals fit with the Cult agenda to transform food from natural to synthetic.

The gas of life is killing us

There can be few greater examples of Cult inversion than the condemnation of carbon dioxide as a dangerous pollutant when it is the gas of life. Without it the natural world would be dead and so we would all be dead. We breathe in oxygen and breathe out carbon dioxide while plants produce oxygen and absorb carbon dioxide. It is a perfect symbiotic relationship that the Cult wants to dismantle for reasons I will come to in the final two chapters. Gates, Schwab, other Cult operatives and mindless repeaters, want the world to be 'carbon neutral' by at least 2050 and the earlier the better. 'Zero carbon' is the cry echoed by lunatics calling for 'Zero Covid' when we already have it. These carbon emission targets will deindustrialise the world in accordance with Cult plans – the post-industrial, post-democratic society – and with so-called renewables like solar and wind not coming even close to meeting human energy needs blackouts and cold are inevitable. Texans got the picture in the winter of 2021 when a snow storm stopped wind turbines and solar panels from working and the lights went down along with water which relies on electricity for its supply system. Gates wants everything to be powered by electricity to ensure that his masters have the kill switch to stop all human activity, movement, cooking, water and warmth any time they like. The climate lie is so

stupendously inverted that it claims we must urgently reduce carbon dioxide when we *don't have enough*.

Co₂ in the atmosphere is a little above 400 parts per million when the optimum for plant growth is 2,000 ppm and when it falls anywhere near 150 ppm the natural world starts to die and so do we. It fell to as low as 280 ppm in an 1880 measurement in Hawaii and rose to 413 ppm in 2019 with industrialisation which is why the planet has become *greener* in the industrial period. How insane then that psychopathic madman Gates is not satisfied only with blocking the rise of Co₂. He's funding technology to suck it out of the atmosphere. The reason why will become clear. The industrial era is not destroying the world through Co₂ and has instead turned around a potentially disastrous ongoing fall in Co₂. Greenpeace co-founder and scientist Patrick Moore walked away from Greenpeace in 1986 and has exposed the green movement for fear-mongering and lies. He said that 500 million years ago there was *17 times* more Co₂ in the atmosphere than we have today and levels have been falling for hundreds of millions of years. In the last 150 million years Co₂ levels in Earth's atmosphere had reduced by *90 percent*. Moore said that by the time humanity began to unlock carbon dioxide from fossil fuels we were at '38 seconds to midnight' and in that sense: 'Humans are [the Earth's] salvation.' Moore made the point that only half the Co₂ emitted by fossil fuels stays in the atmosphere and we should remember that all pollution pouring from chimneys that we are told is carbon dioxide is in fact nothing of the kind. It's pollution. Carbon dioxide is an invisible gas.

William Happer, Professor of Physics at Princeton University and long-time government adviser on climate, has emphasised the Co₂ deficiency for maximum growth and food production. Greenhouse growers don't add carbon dioxide for a bit of fun. He said that most of the warming in the last 100 years, after the earth emerged from the super-cold period of the 'Little Ice Age' into a natural warming cycle, was over by 1940. Happer said that a peak year for warming in 1988 can be explained by a 'monster El Nino' which is a natural and cyclical warming of the Pacific that has nothing to do with 'climate

change'. He said the effect of Co2 could be compared to painting a wall with red paint in that once two or three coats have been applied it didn't matter how much more you slapped on because the wall will not get much redder. Almost all the effect of the rise in Co2 has already happened, he said, and the volume in the atmosphere would now have to *double* to increase temperature by a single degree. Climate hoaxers know this and they have invented the most ridiculously complicated series of 'feedback' loops to try to overcome this rather devastating fact. You hear puppet Greta going on cluelessly about feedback loops and this is why.

The Sun affects temperature? No you *climate denier*

Some other nonsense to contemplate: Climate graphs show that rises in temperature do not follow rises in Co2 – *it's the other way round* with a lag between the two of some 800 years. If we go back 800 years from present time we hit the Medieval Warm Period when temperatures were higher than now without any industrialisation and this was followed by the Little Ice Age when temperatures plummeted. The world was still emerging from these centuries of serious cold when many climate records began which makes the ever-repeated line of the 'hottest year since records began' meaningless when you are not comparing like with like. The coldest period of the Little Ice Age corresponded with the lowest period of sunspot activity when the Sun was at its least active. Proper scientists will not be at all surprised by this when it confirms the obvious fact that earth temperature is affected by the scale of Sun activity and the energetic power that it subsequently emits; but when is the last time you heard a climate hoaxter talking about the Sun as a source of earth temperature?? Everything has to be focussed on Co2 which makes up just 0.117 percent of so-called greenhouse gases and only a fraction of even that is generated by human activity. The rest is natural. More than 90 percent of those greenhouse gases are water vapour and clouds ([Fig 9](#)). Ban moisture I say. Have you noticed that the climate hoaxers no longer use the polar bear as their promotion image? That's because far from becoming extinct polar

bear communities are stable or thriving. Joe Bastardi, American meteorologist, weather forecaster and outspoken critic of the climate lie, documents in his book *The Climate Chronicles* how weather patterns and events claimed to be evidence of climate change have been happening since long before industrialisation: 'What happened before naturally is happening again, as is to be expected given the cyclical nature of the climate due to the design of the planet.' If you read the detailed background to the climate hoax in my other books you will shake your head and wonder how anyone could believe the crap which has spawned a multi-trillion dollar industry based on absolute garbage (see HIV causes AIDS and Sars-Cov-2 causes 'Covid-19'). Climate and 'Covid' have much in common given they have the same source. They both have the contradictory *everything* factor in which everything is explained by reference to them. It's hot – 'it's climate change'. It's cold – 'it's climate change'. I got a sniffle – 'it's Covid'. I haven't got a sniffle – 'it's Covid'. Not having a sniffle has to be a symptom of 'Covid'. Everything is and not having a sniffle is especially dangerous if you are a slow walker. For sheer audacity I offer you a Cambridge University 'study' that actually linked 'Covid' to 'climate change'. It had to happen eventually. They concluded that climate change played a role in 'Covid-19' spreading from animals to humans because ... wait for it ... I kid you not ... *the two groups were forced closer together as populations grow.* Er, that's it. The whole foundation on which this depended was that 'Bats are the likely zoonotic origin of SARS-CoV-1 and SARS-CoV-2'. Well, they are not. They are nothing to do with it. Apart from bats not being the origin and therefore 'climate change' effects on bats being irrelevant I am in awe of their academic insight. Where would we be without them? Not where we are that's for sure.

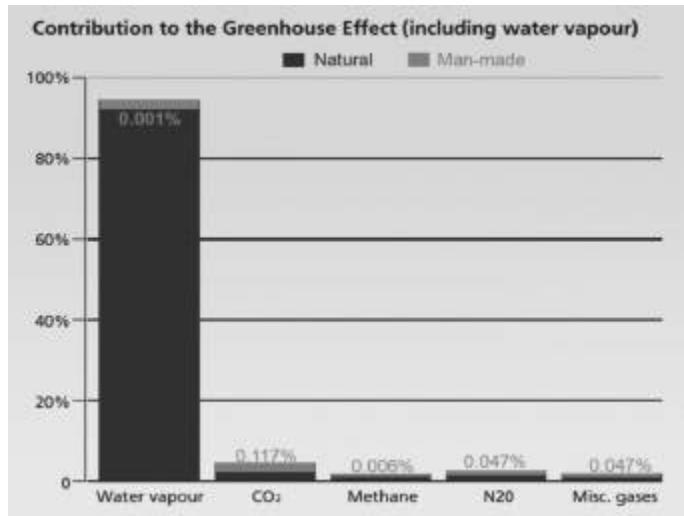


Figure 9: The idea that the gas of life is disastrously changing the climate is an insult to brain cell activity.

One other point about the weather is that climate modification is now well advanced and not every major weather event is natural – or earthquake come to that. I cover this subject at some length in other books. China is openly planning a rapid expansion of its weather modification programme which includes changing the climate in an area more than one and a half times the size of India. China used weather manipulation to ensure clear skies during the 2008 Olympics in Beijing. I have quoted from US military documents detailing how to employ weather manipulation as a weapon of war and they did that in the 1960s and 70s during the conflict in Vietnam with Operation Popeye manipulating monsoon rains for military purposes. Why would there be international treaties on weather modification if it wasn't possible? Of course it is. Weather is energetic information and it can be changed.

How was the climate hoax pulled off? See 'Covid'

If you can get billions to believe in a 'virus' that doesn't exist you can get them to believe in human-caused climate change that doesn't exist. Both are being used by the Cult to transform global society in the way it has long planned. Both hoaxes have been achieved in pretty much the same way. First you declare a lie is a fact. There's a

'virus' you call SARS-Cov-2 or humans are warming the planet with their behaviour. Next this becomes, via Cult networks, the foundation of government, academic and science policy and belief. Those who parrot the mantra are given big grants to produce research that confirms the narrative is true and ever more 'symptoms' are added to make the 'virus'/'climate change' sound even more scary. Scientists and researchers who challenge the narrative have their grants withdrawn and their careers destroyed. The media promote the lie as the unquestionable truth and censor those with an alternative view or evidence. A great percentage of the population believe what they are told as the lie becomes an everybody-knows-that and the believing-masses turn on those with a mind of their own. The technique has been used endlessly throughout human history. Wokers are the biggest promotorrs of the climate lie *and* 'Covid' fascism because their minds are owned by the Cult; their sense of self-righteous self-purity knows no bounds; and they exist in a bubble of reality in which facts are irrelevant and only get in the way of looking without seeing.

Running through all of this like veins in a blue cheese is control of information, which means control of perception, which means control of behaviour, which collectively means control of human society. The Cult owns the global media and Silicon Valley fascists for the simple reason that it *has* to. Without control of information it can't control perception and through that human society. Examine every facet of the Cult agenda and you will see that anything supporting its introduction is never censored while anything pushing back is always censored. I say again: Psychopaths that know why they are doing this must go before Nuremberg trials and those that follow their orders must trot along behind them into the same dock. 'I was just following orders' didn't work the first time and it must not work now. Nuremberg trials must be held all over the world before public juries for politicians, government officials, police, compliant doctors, scientists and virologists, and all Cult operatives such as Gates, Tedros, Fauci, Vallance, Whitty, Ferguson, Zuckerberg, Wojcicki, Brin, Page, Dorsey, the whole damn lot of

them – including, no *especially*, the psychopath psychologists. Without them and the brainless, gutless excuses for journalists that have repeated their lies, none of this could be happening. Nobody can be allowed to escape justice for the psychological and economic Armageddon they are all responsible for visiting upon the human race.

As for the compliant, unquestioning, swathes of humanity, and the self-obsessed, all-knowing ignorance of the Wokers ... don't start me. God help their kids. God help their grandkids. God *help them*.

CHAPTER NINE

We must have it? So what is it?

Well I won't back down. No, I won't back down. You can stand me up at the Gates of Hell. But I won't back down

Tom Petty

I will now focus on the genetically-manipulating ‘Covid vaccines’ which do not meet this official definition of a vaccine by the US Centers for Disease Control (CDC): ‘A product that stimulates a person’s immune system to produce immunity to a specific disease, protecting the person from that disease.’ On that basis ‘Covid vaccines’ are not a vaccine in that the makers don’t even claim they stop infection or transmission.

They are instead part of a multi-levelled conspiracy to change the nature of the human body and what it means to be ‘human’ and to depopulate an enormous swathe of humanity. What I shall call Human 1.0 is on the cusp of becoming Human 2.0 and for very sinister reasons. Before I get to the ‘Covid vaccine’ in detail here’s some background to vaccines in general. Government regulators do not test vaccines – the makers do – and the makers control which data is revealed and which isn’t. Children in America are given 50 vaccine doses by age six and 69 by age 19 and the effect of the whole combined schedule has never been tested. Autoimmune diseases when the immune system attacks its own body have soared in the mass vaccine era and so has disease in general in children and the young. Why wouldn’t this be the case when vaccines target the *immune system*? The US government gave Big Pharma drug

companies immunity from prosecution for vaccine death and injury in the 1986 National Childhood Vaccine Injury Act (NCVIA) and since then the government (taxpayer) has been funding compensation for the consequences of Big Pharma vaccines. The criminal and satanic drug giants can't lose and the vaccine schedule has increased dramatically since 1986 for this reason. There is no incentive to make vaccines safe and a big incentive to make money by introducing ever more. Even against a ridiculously high bar to prove vaccine liability, and with the government controlling the hearing in which it is being challenged for compensation, the vaccine court has so far paid out more than \$4 billion. These are the vaccines we are told are safe and psychopaths like Zuckerberg censor posts saying otherwise. The immunity law was even justified by a ruling that vaccines by their nature were 'unavoidably unsafe'.

Check out the ingredients of vaccines and you will be shocked if you are new to this. *They put that in children's bodies?? What??* Try aluminium, a brain toxin connected to dementia, aborted foetal tissue and formaldehyde which is used to embalm corpses. World-renowned aluminium expert Christopher Exley had his research into the health effect of aluminium in vaccines shut down by Keele University in the UK when it began taking funding from the Bill and Melinda Gates Foundation. Research when diseases 'eradicated' by vaccines began to decline and you will find the fall began long *before* the vaccine was introduced. Sometimes the fall even plateaued after the vaccine. Diseases like scarlet fever for which there was no vaccine declined in the same way because of environmental and other factors. A perfect case in point is the polio vaccine. Polio began when lead arsenate was first sprayed as an insecticide and residues remained in food products. Spraying started in 1892 and the first US polio epidemic came in Vermont in 1894. The simple answer was to stop spraying, but Rockefeller-created Big Pharma had a better idea. Polio was decreed to be caused by the *poliovirus* which 'spreads from person to person and can infect a person's spinal cord'. Lead arsenate was replaced by the lethal DDT which had the same effect of causing paralysis by damaging the brain and central nervous

system. Polio plummeted when DDT was reduced and then banned, but the vaccine is still given the credit for something it didn't do. Today by far the biggest cause of polio is the vaccines promoted by Bill Gates. Vaccine justice campaigner Robert Kennedy Jr, son of assassinated (by the Cult) US Attorney General Robert Kennedy, wrote:

In 2017, the World Health Organization (WHO) reluctantly admitted that the global explosion in polio is predominantly vaccine strain. The most frightening epidemics in Congo, Afghanistan, and the Philippines, are all linked to vaccines. In fact, by 2018, 70% of global polio cases were vaccine strain.

Vaccines make fortunes for Cult-owned Gates and Big Pharma while undermining the health and immune systems of the population. We had a glimpse of the mentality behind the Big Pharma cartel with a report on WION (World is One News), an international English language TV station based in India, which exposed the extraordinary behaviour of US drug company Pfizer over its 'Covid vaccine'. The WION report told how Pfizer had made fantastic demands of Argentina, Brazil and other countries in return for its 'vaccine'. These included immunity from prosecution, even for Pfizer negligence, government insurance to protect Pfizer from law suits and handing over as collateral sovereign assets of the country to include Argentina's bank reserves, military bases and embassy buildings. Pfizer demanded the same of Brazil in the form of waiving sovereignty of its assets abroad; exempting Pfizer from Brazilian laws; and giving Pfizer immunity from all civil liability. This is a 'vaccine' developed with government funding. Big Pharma is evil incarnate as a creation of the Cult and all must be handed tickets to Nuremberg.

Phantom 'vaccine' for a phantom 'disease'

I'll expose the 'Covid vaccine' fraud and then go on to the wider background of why the Cult has set out to 'vaccinate' every man, woman and child on the planet for an alleged 'new disease' with a survival rate of 99.77 percent (or more) even by the grotesquely-

manipulated figures of the World Health Organization and Johns Hopkins University. The ‘infection’ to ‘death’ ratio is 0.23 to 0.15 percent according to Stanford epidemiologist Dr John Ioannidis and while estimates vary the danger remains tiny. I say that if the truth be told the fake infection to fake death ratio is zero. Never mind all the evidence I have presented here and in *The Answer* that there is no ‘virus’ let us just focus for a moment on that death-rate figure of say 0.23 percent. The figure includes all those worldwide who have tested positive with a test not testing for the ‘virus’ and then died within 28 days or even longer of any other cause – *any other cause*. Now subtract all those illusory ‘Covid’ deaths on the global data sheets from the 0.23 percent. What do you think you would be left with? *Zero*. A vaccination has never been successfully developed for a so-called coronavirus. They have all failed at the animal testing stage when they caused hypersensitivity to what they were claiming to protect against and made the impact of a disease far worse. Cult-owned vaccine corporations got around that problem this time by bypassing animal trials, going straight to humans and making the length of the ‘trials’ before the public rollout as short as they could get away with. Normally it takes five to ten years or more to develop vaccines that still cause demonstrable harm to many people and that’s without including the long-term effects that are never officially connected to the vaccination. ‘Covid’ non-vaccines have been officially produced and approved in a matter of months from a standing start and part of the reason is that (a) they were developed before the ‘Covid’ hoax began and (b) they are based on computer programs and not natural sources. Official non-trials were so short that government agencies gave *emergency*, not full, approval. ‘Trials’ were not even completed and full approval cannot be secured until they are. Public ‘Covid vaccination’ is actually a *continuation of the trial*. Drug company ‘trials’ are not scheduled to end until 2023 by which time a lot of people are going to be dead. Data on which government agencies gave this emergency approval was supplied by the Big Pharma corporations themselves in the form of Pfizer/BioNTech, AstraZeneca, Moderna, Johnson & Johnson, and

others, and this is the case with all vaccines. By its very nature *emergency* approval means drug companies do not have to prove that the ‘vaccine’ is ‘safe and effective’. How could they with trials way short of complete? Government regulators only have to *believe* that they *could* be safe and effective. It is criminal manipulation to get products in circulation with no testing worth the name. Agencies giving that approval are infested with Big Pharma-connected place-people and they act in the interests of Big Pharma (the Cult) and not the public about whom they do not give a damn.

More human lab rats

‘Covid vaccines’ produced in record time by Pfizer/BioNTech and Moderna employ a technique *never approved before for use on humans*. They are known as mRNA ‘vaccines’ and inject a synthetic version of ‘viral’ mRNA or ‘messenger RNA’. The key is in the term ‘messenger’. The body works, or doesn’t, on the basis of information messaging. Communications are constantly passing between and within the genetic system and the brain. Change those messages and you change the state of the body and even its very nature and you can change psychology and behaviour by the way the brain processes information. I think you are going to see significant changes in personality and perception of many people who have had the ‘Covid vaccine’ synthetic potions. Insider Aldous Huxley predicted the following in 1961 and mRNA ‘vaccines’ can be included in the term ‘pharmacological methods’:

There will be, in the next generation or so, a pharmacological method of making people love their servitude, and producing dictatorship without tears, so to speak, producing a kind of painless concentration camp for entire societies, so that people will in fact have their own liberties taken away from them, but rather enjoy it, because they will be distracted from any desire to rebel by propaganda or brainwashing, or brainwashing enhanced by pharmacological methods. And this seems to be the final revolution.

Apologists claim that mRNA synthetic ‘vaccines’ don’t change the DNA genetic blueprint because RNA does not affect DNA only the other way round. This is so disingenuous. A process called ‘reverse

'transcription' can convert RNA into DNA and be integrated into DNA in the cell nucleus. This was highlighted in December, 2020, by scientists at Harvard and Massachusetts Institute of Technology (MIT). Geneticists report that more than 40 percent of mammalian genomes results from reverse transcription. On the most basic level if messaging changes then that sequence must lead to changes in DNA which is receiving and transmitting those communications. How can introducing synthetic material into cells not change the cells where DNA is located? The process is known as transfection which is defined as 'a technique to insert foreign nucleic acid (DNA or RNA) into a cell, typically with the intention of altering the properties of the cell'. Researchers at the Sloan Kettering Institute in New York found that changes in messenger RNA can deactivate tumour-suppressing proteins and thereby promote cancer. This is what happens when you mess with messaging. 'Covid vaccine' maker Moderna was founded in 2010 by Canadian stem cell biologist Derrick J. Rossi after his breakthrough discovery in the field of transforming and reprogramming stem cells. These are neutral cells that can be programmed to become any cell including sperm cells. Moderna was therefore founded on the principle of genetic manipulation and has never produced any vaccine or drug before its genetically-manipulating synthetic 'Covid' shite. Look at the name – Mode-RNA or Modify-RNA. Another important point is that the US Supreme Court has ruled that genetically-modified DNA, or complementary DNA (cDNA) synthesized in the laboratory from messenger RNA, can be patented and owned. These psychopaths are doing this to the human body.

Cells replicate synthetic mRNA in the 'Covid vaccines' and in theory the body is tricked into making antigens which trigger antibodies to target the 'virus spike proteins' which as Dr Tom Cowan said have *never been seen*. Cut the crap and these 'vaccines' deliver *self-replicating* synthetic material to the cells with the effect of changing human DNA. The more of them you have the more that process is compounded while synthetic material is all the time self-replicating. 'Vaccine'-maker Moderna describes mRNA as 'like

software for the cell' and so they are messing with the body's software. What happens when you change the software in a computer? Everything changes. For this reason the Cult is preparing a production line of mRNA 'Covid vaccines' and a long list of excuses to use them as with all the 'variants' of a 'virus' never shown to exist. The plan is further to transfer the mRNA technique to other vaccines mostly given to children and young people. The cumulative consequences will be a transformation of human DNA through a constant infusion of synthetic genetic material which will kill many and change the rest. Now consider that governments that have given emergency approval for a vaccine that's not a vaccine; never been approved for humans before; had no testing worth the name; and the makers have been given immunity from prosecution for any deaths or adverse effects suffered by the public. The UK government awarded *permanent legal indemnity* to itself and its employees for harm done when a patient is being treated for 'Covid-19' or 'suspected Covid-19'. That is quite a thought when these are possible 'side-effects' from the 'vaccine' (they are not 'side', they are effects) listed by the US Food and Drug Administration:

Guillain-Barre syndrome; acute disseminated encephalomyelitis; transverse myelitis; encephalitis; myelitis; encephalomyelitis; meningoencephalitis; meningitis; encephalopathy; convulsions; seizures; stroke; narcolepsy; cataplexy; anaphylaxis; acute myocardial infarction (heart attack); myocarditis; pericarditis; autoimmune disease; death; implications for pregnancy, and birth outcomes; other acute demyelinating diseases; non anaphylactic allergy reactions; thrombocytopenia ; disseminated intravascular coagulation; venous thromboembolism; arthritis; arthralgia; joint pain; Kawasaki disease; multisystem inflammatory syndrome in children; vaccine enhanced disease. The latter is the way the 'vaccine' has the potential to make diseases far worse than they would otherwise be.

UK doctor and freedom campaigner Vernon Coleman described the conditions in this list as 'all unpleasant, most of them very serious, and you can't get more serious than death'. The thought that anyone at all has had the 'vaccine' in these circumstances is testament to the potential that humanity has for clueless, unquestioning, stupidity and for many that programmed stupidity has already been terminal.

An insider speaks

Dr Michael Yeadon is a former Vice President, head of research and Chief Scientific Adviser at vaccine giant Pfizer. Yeadon worked on the inside of Big Pharma, but that did not stop him becoming a vocal critic of 'Covid vaccines' and their potential for multiple harms, including infertility in women. By the spring of 2021 he went much further and even used the no, no, term 'conspiracy'. When you begin to see what is going on it is impossible not to do so. Yeadon spoke out in an interview with freedom campaigner James Delingpole and I mentioned earlier how he said that no one had samples of 'the virus'. He explained that the mRNA technique originated in the anti-cancer field and ways to turn on and off certain genes which could be advantageous if you wanted to stop cancer growing out of control. 'That's the origin of them. They are a very unusual application, really.' Yeadon said that treating a cancer patient with an aggressive procedure might be understandable if the alternative was dying, but it was quite another thing to use the same technique as a public health measure. Most people involved wouldn't catch the infectious agent you were vaccinating against and if they did they probably wouldn't die:

If you are really using it as a public health measure you really want to as close as you can get to zero side-effects ... I find it odd that they chose techniques that were really cutting their teeth in the field of oncology and I'm worried that in using gene-based vaccines that have to be injected in the body and spread around the body, get taken up into some cells, and the regulators haven't quite told us which cells they get taken up into ... you are going to be generating a wide range of responses ... with multiple steps each of which could go well or badly.

I doubt the Cult intends it to go well. Yeadon said that you can put any gene you like into the body through the 'vaccine'. 'You can certainly give them a gene that would do them some harm if you wanted.' I was intrigued when he said that when used in the cancer field the technique could turn genes on and off. I explore this process in *The Answer* and with different genes having different functions you could create mayhem – physically and psychologically – if you turned the wrong ones on and the right ones off. I read reports of an experiment by researchers at the University of Washington's school of computer science and engineering in which they encoded DNA to infect computers. The body is itself a biological computer and if human DNA can inflict damage on a computer why can't the computer via synthetic material mess with the human body? It can. The Washington research team said it was possible to insert malicious malware into 'physical DNA strands' and corrupt the computer system of a gene sequencing machine as it 'reads gene letters and stores them as binary digits 0 and 1'. They concluded that hackers could one day use blood or spit samples to access computer systems and obtain sensitive data from police forensics labs or infect genome files. It is at this level of digital interaction that synthetic 'vaccines' need to be seen to get the full picture and that will become very clear later on. Michael Yeadon said it made no sense to give the 'vaccine' to younger people who were in no danger from the 'virus'. What was the benefit? It was all downside with potential effects:

The fact that my government in what I thought was a civilised, rational country, is raining [the 'vaccine'] on people in their 30s and 40s, even my children in their 20s, they're getting letters and phone calls, I know this is not right and any of you doctors who are vaccinating you know it's not right, too. They are not at risk. They are not at risk from the disease, so you are now hoping that the side-effects are so rare that you get away with it. You don't give new technology ... that you don't understand to 100 percent of the population.

Blood clot problems with the AstraZeneca 'vaccine' have been affecting younger people to emphasise the downside risks with no benefit. AstraZeneca's version, produced with Oxford University, does not use mRNA, but still gets its toxic cocktail inside cells where

it targets DNA. The Johnson & Johnson ‘vaccine’ which uses a similar technique has also produced blood clot effects to such an extent that the United States paused its use at one point. They are all ‘gene therapy’ (cell modification) procedures and not ‘vaccines’. The truth is that once the content of these injections enter cells we have no idea what the effect will be. People can speculate and some can give very educated opinions and that’s good. In the end, though, only the makers know what their potions are designed to do and even they won’t know every last consequence. Michael Yeadon was scathing about doctors doing what they knew to be wrong.

‘Everyone’s mute’, he said. Doctors in the NHS must know this was not right, coming into work and injecting people. ‘I don’t know how they sleep at night. I know I couldn’t do it. I know that if I were in that position I’d have to quit.’ He said he knew enough about toxicology to know this was not a good risk-benefit. Yeadon had spoken to seven or eight university professors and all except two would not speak out publicly. Their universities had a policy that no one said anything that countered the government and its medical advisors. They were afraid of losing their government grants. This is how intimidation has been used to silence the truth at every level of the system. I say silence, but these people could still speak out if they made that choice. Yeadon called them ‘moral cowards’ – ‘This is about your children and grandchildren’s lives and you have just buggered off and left it.’

‘Variant’ nonsense

Some of his most powerful comments related to the alleged ‘variants’ being used to instil more fear, justify more lockdowns, and introduce more ‘vaccines’. He said government claims about ‘variants’ were nonsense. He had checked the alleged variant ‘codes’ and they were 99.7 percent identical to the ‘original’. This was the human identity difference equivalent to putting a baseball cap on and off or wearing it the other way round. A 0.3 percent difference would make it impossible for that ‘variant’ to escape immunity from the ‘original’. This made no sense of having new ‘vaccines’ for

'variants'. He said there would have to be at least a *30 percent* difference for that to be justified and even then he believed the immune system would still recognise what it was. Gates-funded 'variant modeller' and 'vaccine'-pusher John Edmunds might care to comment. Yeadon said drug companies were making new versions of the 'vaccine' as a 'top up' for 'variants'. Worse than that, he said, the 'regulators' around the world like the MHRA in the UK had got together and agreed that because 'vaccines' for 'variants' were so similar to the first 'vaccines' *they did not have to do safety studies*. How transparently sinister that is. This is when Yeadon said: 'There is a conspiracy here.' There was no need for another vaccine for 'variants' and yet we were told that there was and the country had shut its borders because of them. 'They are going into hundreds of millions of arms without passing 'go' or any regulator. Why did they do that? Why did they pick this method of making the vaccine?'

The reason had to be something bigger than that it seemed and 'it's not protection against the virus'. It's was a far bigger project that meant politicians and advisers were willing to do things and not do things that knowingly resulted in avoidable deaths – 'that's already happened when you think about lockdown and deprivation of health care for a year.' He spoke of people prepared to do something that results in the avoidable death of their fellow human beings and it not bother them. This is the penny-drop I have been working to get across for more than 30 years – the level of pure evil we are dealing with. Yeadon said his friends and associates could not believe there could be that much evil, but he reminded them of Stalin, Pol Pot and Hitler and of what Stalin had said: 'One death is a tragedy. A million? A statistic.' He could not think of a benign explanation for why you need top-up vaccines 'which I'm sure you don't' and for the regulators 'to just get out of the way and wave them through'. Why would the regulators do that when they were still wrestling with the dangers of the 'parent' vaccine? He was clearly shocked by what he had seen since the 'Covid' hoax began and now he was thinking the previously unthinkable:

If you wanted to depopulate a significant proportion of the world and to do it in a way that doesn't involve destruction of the environment with nuclear weapons, poisoning everyone with anthrax or something like that, and you wanted plausible deniability while you had a multi-year infectious disease crisis, I actually don't think you could come up with a better plan of work than seems to be in front of me. I can't say that's what they are going to do, but I can't think of a benign explanation why they are doing it.

He said he never thought that they would get rid of 99 percent of humans, but now he wondered. 'If you wanted to that this would be a hell of a way to do it – it would be unstoppable folks.' Yeadon had concluded that those who submitted to the 'vaccine' would be allowed to have some kind of normal life (but for how long?) while screws were tightened to coerce and mandate the last few percent. 'I think they'll put the rest of them in a prison camp. I wish I was wrong, but I don't think I am.' Other points he made included: There were no coronavirus vaccines then suddenly they all come along at the same time; we have no idea of the long term affect with trials so short; coercing or forcing people to have medical procedures is against the Nuremberg Code instigated when the Nazis did just that; people should at least delay having the 'vaccine'; a quick Internet search confirms that masks don't reduce respiratory viral transmission and 'the government knows that'; they have smashed civil society and they know that, too; two dozen peer-reviewed studies show no connection between lockdown and reducing deaths; he knew from personal friends the elite were still flying around and going on holiday while the public were locked down; the elite were not having the 'vaccines'. He was also asked if 'vaccines' could be made to target difference races. He said he didn't know, but the document by the Project for the New American Century in September, 2000, said developing 'advanced forms of biological warfare that can target *specific genotypes* may transform biological warfare from the realm of terror to a politically useful tool.' Oh, they're evil all right. Of that we can be *absolutely* sure.

Another cull of old people

We have seen from the CDC definition that the mRNA 'Covid vaccine' is not a vaccine and nor are the others that *claim* to reduce 'severity of symptoms' in *some* people, but not protect from infection or transmission. What about all the lies about returning to 'normal' if people were 'vaccinated'? If they are not claimed to stop infection and transmission of the alleged 'virus', how does anything change? This was all lies to manipulate people to take the jabs and we are seeing that now with masks and distancing still required for the 'vaccinated'. How did they think that elderly people with fragile health and immune responses were going to be affected by infusing their cells with synthetic material and other toxic substances? They *knew* that in the short and long term it would be devastating and fatal as the culling of the old that began with the first lockdowns was continued with the 'vaccine'. Death rates in care homes soared immediately residents began to be 'vaccinated' – infused with synthetic material. Brave and committed whistleblower nurses put their careers at risk by exposing this truth while the rest kept their heads down and their mouths shut to put their careers before those they are supposed to care for. A long-time American Certified Nursing Assistant who gave his name as James posted a video in which he described emotionally what happened in his care home when vaccination began. He said that during 2020 very few residents were sick with 'Covid' and no one died during the entire year; but shortly after the Pfizer mRNA injections 14 people died within two weeks and many others were near death. 'They're dropping like flies', he said. Residents who walked on their own before the shot could no longer and they had lost their ability to conduct an intelligent conversation. The home's management said the sudden deaths were caused by a 'super-spreader' of 'Covid-19'. Then how come, James asked, that residents who refused to take the injections were not sick? It was a case of inject the elderly with mRNA synthetic potions and blame their illness and death that followed on the 'virus'. James described what was happening in care homes as 'the greatest crime of genocide this country has ever seen'. Remember the NHS staff nurse from earlier who used the same

word ‘genocide’ for what was happening with the ‘vaccines’ and that it was an ‘act of human annihilation’. A UK care home whistleblower told a similar story to James about the effect of the ‘vaccine’ in deaths and ‘outbreaks’ of illness dubbed ‘Covid’ after getting the jab. She told how her care home management and staff had zealously imposed government regulations and no one was allowed to even question the official narrative let alone speak out against it. She said the NHS was even worse. Again we see the results of reframing. A worker at a local care home where I live said they had not had a single case of ‘Covid’ there for almost a year and when the residents were ‘vaccinated’ they had 19 positive cases in two weeks with eight dying.

It's not the 'vaccine' – honest

The obvious cause and effect was being ignored by the media and most of the public. Australia’s health minister Greg Hunt (a former head of strategy at the World Economic Forum) was admitted to hospital after he had the ‘vaccine’. He was suffering according to reports from the skin infection ‘cellulitis’ and it must have been a severe case to have warranted days in hospital. Immediately the authorities said this was nothing to do with the ‘vaccine’ when an effect of some vaccines is a ‘cellulitis-like reaction’. We had families of perfectly healthy old people who died after the ‘vaccine’ saying that if only they had been given the ‘vaccine’ earlier they would still be alive. As a numbskull rating that is off the chart. A father of four ‘died of Covid’ at aged 48 when he was taken ill two days after having the ‘vaccine’. The man, a health administrator, had been ‘shielding during the pandemic’ and had ‘not really left the house’ until he went for the ‘vaccine’. Having the ‘vaccine’ and then falling ill and dying does not seem to have qualified as a possible cause and effect and ‘Covid-19’ went on his death certificate. His family said they had no idea how he ‘caught the virus’. A family member said: ‘Tragically, it could be that going for a vaccination ultimately led to him catching Covid ...The sad truth is that they are never going to know where it came from.’ The family warned people to remember

that the virus still existed and was ‘very real’. So was their stupidity. Nurses and doctors who had the first round of the ‘vaccine’ were collapsing, dying and ending up in a hospital bed while they or their grieving relatives were saying they’d still have the ‘vaccine’ again despite what happened. I kid you not. You mean if your husband returned from the dead he’d have the same ‘vaccine’ again that killed him??

Doctors at the VCU Medical Center in Richmond, Virginia, said the Johnson & Johnson ‘vaccine’ was to blame for a man’s skin peeling off. Patient Richard Terrell said: ‘It all just happened so fast. My skin peeled off. It’s still coming off on my hands now.’ He said it was stinging, burning and itching and when he bent his arms and legs it was very painful with ‘the skin swollen and rubbing against itself’. Pfizer/BioNTech and Moderna vaccines use mRNA to change the cell while the Johnson & Johnson version uses DNA in a process similar to AstraZeneca’s technique. Johnson & Johnson and AstraZeneca have both had their ‘vaccines’ paused by many countries after causing serious blood problems. Terrell’s doctor Fnu Nutan said he could have died if he hadn’t got medical attention. It sounds terrible so what did Nutan and Terrell say about the ‘vaccine’ now? Oh, they still recommend that people have it. A nurse in a hospital bed 40 minutes after the vaccination and unable to swallow due to throat swelling was told by a doctor that he lost mobility in his arm for 36 hours following the vaccination. What did he say to the ailing nurse? ‘Good for you for getting the vaccination.’ We are dealing with a serious form of cognitive dissonance madness in both public and medical staff. There is a remarkable correlation between those having the ‘vaccine’ and trumpeting the fact and suffering bad happenings shortly afterwards. Witold Rogiewicz, a Polish doctor, made a video of his ‘vaccination’ and ridiculed those who were questioning its safety and the intentions of Bill Gates: ‘Vaccinate yourself to protect yourself, your loved ones, friends and also patients. And to mention quickly I have info for anti-vaxxers and anti-Covidiers if you want to contact Bill Gates you can do this through me.’ He further ridiculed the dangers of 5G. Days later he

was dead, but naturally the vaccination wasn't mentioned in the verdict of 'heart attack'.

Lies, lies and more lies

So many members of the human race have slipped into extreme states of insanity and unfortunately they include reframed doctors and nursing staff. Having a 'vaccine' and dying within minutes or hours is not considered a valid connection while death from any cause within 28 days or longer of a positive test with a test not testing for the 'virus' means 'Covid-19' goes on the death certificate. How could that 'vaccine'-death connection not have been made except by calculated deceit? US figures in the initial rollout period to February 12th, 2020, revealed that a third of the deaths reported to the CDC after 'Covid vaccines' happened within 48 hours. Five men in the UK suffered an 'extremely rare' blood clot problem after having the AstraZeneca 'vaccine', but no causal link was established said the Gates-funded Medicines and Healthcare products Regulatory Agency (MHRA) which had given the 'vaccine' emergency approval to be used. Former Pfizer executive Dr Michael Yeadon explained in his interview how the procedures could cause blood coagulation and clots. People who should have been at no risk were dying from blood clots in the brain and he said he had heard from medical doctor friends that people were suffering from skin bleeding and massive headaches. The AstraZeneca 'shot' was stopped by some 20 countries over the blood clotting issue and still the corrupt MHRA, the European Medicines Agency (EMA) and the World Health Organization said that it should continue to be given even though the EMA admitted that it 'still cannot rule out definitively' a link between blood clotting and the 'vaccine'. Later Marco Cavaleri, head of EMA vaccine strategy, said there was indeed a clear link between the 'vaccine' and thrombosis, but they didn't know why. So much for the trials showing the 'vaccine' is safe. Blood clots were affecting younger people who would be under virtually no danger from 'Covid' even if it existed which makes it all the more stupid and sinister.

The British government responded to public alarm by wheeling out June Raine, the terrifyingly weak infant school headmistress sound-alike who heads the UK MHRA drug ‘regulator’. The idea that she would stand up to Big Pharma and government pressure is laughable and she told us that all was well in the same way that she did when allowing untested, never-used-on-humans-before, genetically-manipulating ‘vaccines’ to be exposed to the public in the first place. Mass lying is the new normal of the ‘Covid’ era. The MHRA later said 30 cases of rare blood clots had by then been connected with the AstraZeneca ‘vaccine’ (that means a lot more in reality) while stressing that the benefits of the jab in preventing ‘Covid-19’ outweighed any risks. A more ridiculous and disingenuous statement with callous disregard for human health it is hard to contemplate. Immediately after the mendacious ‘all-clears’ two hospital workers in Denmark experienced blood clots and cerebral haemorrhaging following the AstraZeneca jab and one died. Top Norwegian health official Pål Andre Holme said the ‘vaccine’ was the only common factor: ‘There is nothing in the patient history of these individuals that can give such a powerful immune response ... I am confident that the antibodies that we have found are the cause, and I see no other explanation than it being the vaccine which triggers it.’ Strokes, a clot or bleed in the brain, were clearly associated with the ‘vaccine’ from word of mouth and whistleblower reports. Similar consequences followed with all these ‘vaccines’ that we were told were so safe and as the numbers grew by the day it was clear we were witnessing human carnage.

Learning the hard way

A woman interviewed by UKColumn told how her husband suffered dramatic health effects after the vaccine when he’d been in good health all his life. He went from being a little unwell to losing all feeling in his legs and experiencing ‘excruciating pain’. Misdiagnosis followed twice at Accident and Emergency (an ‘allergy’ and ‘sciatica’) before he was admitted to a neurology ward where doctors said his serious condition had been caused by the

'vaccine'. Another seven 'vaccinated' people were apparently being treated on the same ward for similar symptoms. The woman said he had the 'vaccine' because they believed media claims that it was safe. 'I didn't think the government would give out a vaccine that does this to somebody; I believed they would be bringing out a vaccination that would be safe.' What a tragic way to learn that lesson. Another woman posted that her husband was transporting stroke patients to hospital on almost every shift and when he asked them if they had been 'vaccinated' for 'Covid' they all replied 'yes'. One had a 'massive brain bleed' the day after his second dose. She said her husband reported the 'just been vaccinated' information every time to doctors in A and E only for them to ignore it, make no notes and appear annoyed that it was even mentioned. This particular report cannot be verified, but it expresses a common theme that confirms the monumental underreporting of 'vaccine' consequences. Interestingly as the 'vaccines' and their brain blood clot/stroke consequences began to emerge the UK National Health Service began a publicity campaign telling the public what to do in the event of a stroke. A Scottish NHS staff nurse who quit in disgust in March, 2021, said:

I have seen traumatic injuries from the vaccine, they're not getting reported to the yellow card [adverse reaction] scheme, they're treating the symptoms, not asking why, why it's happening. It's just treating the symptoms and when you speak about it you're dismissed like you're crazy, I'm not crazy, I'm not crazy because every other colleague I've spoken to is terrified to speak out, they've had enough.

Videos appeared on the Internet of people uncontrollably shaking after the 'vaccine' with no control over muscles, limbs and even their face. A Scottish mother broke out in a severe rash all over her body almost immediately after she was given the AstraZeneca 'vaccine'. The pictures were horrific. Leigh King, a 41-year-old hairdresser from Lanarkshire said: 'Never in my life was I prepared for what I was about to experience ... My skin was so sore and constantly hot ... I have never felt pain like this ...' But don't you worry, the 'vaccine' is perfectly safe. Then there has been the effect on medical

staff who have been pressured to have the ‘vaccine’ by psychopathic ‘health’ authorities and government. A London hospital consultant who gave the name K. Polyakova wrote this to the *British Medical Journal* or *BMJ*:

I am currently struggling with ... the failure to report the reality of the morbidity caused by our current vaccination program within the health service and staff population. The levels of sickness after vaccination is unprecedented and staff are getting very sick and some with neurological symptoms which is having a huge impact on the health service function. Even the young and healthy are off for days, some for weeks, and some requiring medical treatment. Whole teams are being taken out as they went to get vaccinated together.

Mandatory vaccination in this instance is stupid, unethical and irresponsible when it comes to protecting our staff and public health. We are in the voluntary phase of vaccination, and encouraging staff to take an unlicensed product that is impacting on their immediate health ... it is clearly stated that these vaccine products do not offer immunity or stop transmission. In which case why are we doing it?

Not to protect health that’s for sure. Medical workers are lauded by governments for agenda reasons when they couldn’t give a toss about them any more than they can for the population in general. Schools across America faced the same situation as they closed due to the high number of teachers and other staff with bad reactions to the Pfizer/BioNTech, Moderna, and Johnson & Johnson ‘Covid vaccines’ all of which were linked to death and serious adverse effects. The *BMJ* took down the consultant’s comments pretty quickly on the grounds that they were being used to spread ‘disinformation’. They were exposing the truth about the ‘vaccine’ was the real reason. The cover-up is breathtaking.

Hiding the evidence

The scale of the ‘vaccine’ death cover-up worldwide can be confirmed by comparing official figures with the personal experience of the public. I heard of many people in my community who died immediately or soon after the vaccine that would never appear in the media or even likely on the official totals of ‘vaccine’ fatalities and adverse reactions when only about ten percent are estimated to be

reported and I have seen some estimates as low as one percent in a Harvard study. In the UK alone by April 29th, 2021, some 757,654 adverse reactions had been officially reported from the Pfizer/BioNTech, Oxford/AstraZeneca and Moderna 'vaccines' with more than a thousand deaths linked to jabs and that means an estimated ten times this number in reality from a ten percent reporting rate percentage. That's seven million adverse reactions and 10,000 potential deaths and a one percent reporting rate would be ten times *those* figures. In 1976 the US government pulled the swine flu vaccine after 53 deaths. The UK data included a combined 10,000 eye disorders from the 'Covid vaccines' with more than 750 suffering visual impairment or blindness and again multiply by the estimated reporting percentages. As 'Covid cases' officially fell hospitals virtually empty during the 'Covid crisis' began to fill up with a range of other problems in the wake of the 'vaccine' rollout. The numbers across America have also been catastrophic. Deaths linked to *all* types of vaccine increased by *6,000 percent* in the first quarter of 2021 compared with 2020. A 39-year-old woman from Ogden, Utah, died four days after receiving a second dose of Moderna's 'Covid vaccine' when her liver, heart and kidneys all failed despite the fact that she had no known medical issues or conditions. Her family sought an autopsy, but Dr Erik Christensen, Utah's chief medical examiner, said proving vaccine injury as a cause of death almost never happened. He could think of only one instance where an autopsy would name a vaccine as the official cause of death and that would be anaphylaxis where someone received a vaccine and died almost instantaneously. 'Short of that, it would be difficult for us to definitively say this is the vaccine,' Christensen said. If that is true this must be added to the estimated ten percent (or far less) reporting rate of vaccine deaths and serious reactions and the conclusion can only be that vaccine deaths and serious reactions – including these 'Covid' potions – are phenomenally understated in official figures. The same story can be found everywhere. Endless accounts of deaths and serious reactions among the public, medical

and care home staff while official figures did not even begin to reflect this.

Professional script-reader Dr David Williams, a ‘top public-health official’ in Ontario, Canada, insulted our intelligence by claiming only four serious adverse reactions and no deaths from the more than 380,000 vaccine doses then given. This bore no resemblance to what people knew had happened in their own circles and we had Dirk Huyer in charge of getting millions vaccinated in Ontario while at the same time he was Chief Coroner for the province investigating causes of death including possible death from the vaccine. An aide said he had stepped back from investigating deaths, but evidence indicated otherwise. Rosemary Frei, who secured a Master of Science degree in molecular biology at the Faculty of Medicine at Canada’s University of Calgary before turning to investigative journalism, was one who could see that official figures for ‘vaccine’ deaths and reactions made no sense. She said that doctors seldom reported adverse events and when people got really sick or died after getting a vaccination they would attribute that to anything except the vaccines. It had been that way for years and anyone who wondered aloud whether the ‘Covid vaccines’ or other shots cause harm is immediately branded as ‘anti-vax’ and ‘anti-science’. This was ‘career-threatening’ for health professionals. Then there was the huge pressure to support the push to ‘vaccinate’ billions in the quickest time possible. Frei said:

So that’s where we’re at today. More than half a million vaccine doses have been given to people in Ontario alone. The rush is on to vaccinate all 15 million of us in the province by September. And the mainstream media are screaming for this to be sped up even more. That all adds up to only a very slim likelihood that we’re going to be told the truth by officials about how many people are getting sick or dying from the vaccines.

What is true of Ontario is true of everywhere.

They KNEW – and still did it

The authorities knew what was going to happen with multiple deaths and adverse reactions. The UK government’s Gates-funded

and Big Pharma-dominated Medicines and Healthcare products Regulatory Agency (MHRA) hired a company to employ AI in compiling the projected reactions to the ‘vaccine’ that would otherwise be uncountable. The request for applications said: ‘The MHRA urgently seeks an Artificial Intelligence (AI) software tool to process the expected high volume of Covid-19 vaccine Adverse Drug Reaction ...’ This was from the agency, headed by the disingenuous June Raine, that gave the ‘vaccines’ emergency approval and the company was hired before the first shot was given. ‘We are going to kill and maim you – is that okay?’ ‘Oh, yes, perfectly fine – I’m very grateful, thank you, doctor.’ The range of ‘Covid vaccine’ adverse reactions goes on for page after page in the MHRA criminally underreported ‘Yellow Card’ system and includes affects to eyes, ears, skin, digestion, blood and so on. Raine’s MHRA amazingly claimed that the ‘overall safety experience ... is so far as expected from the clinical trials’. The death, serious adverse effects, deafness and blindness were *expected*? When did they ever mention that? If these human tragedies were expected then those that gave approval for the use of these ‘vaccines’ must be guilty of crimes against humanity including murder – a definition of which is ‘killing a person with malice aforethought or with recklessness manifesting extreme indifference to the value of human life.’ People involved at the MHRA, the CDC in America and their equivalent around the world must go before Nuremberg trials to answer for their callous inhumanity. We are only talking here about the immediate effects of the ‘vaccine’. The longer-term impact of the DNA synthetic manipulation is the main reason they are so hysterically desperate to inoculate the entire global population in the shortest possible time.

Africa and the developing world are a major focus for the ‘vaccine’ depopulation agenda and a mass vaccination sales-pitch is underway thanks to caring people like the Rockefellers and other Cult assets. The Rockefeller Foundation, which pre-empted the ‘Covid pandemic’ in a document published in 2010 that ‘predicted’ what happened a decade later, announced an initial \$34.95 million grant in February, 2021, ‘to ensure more equitable access to Covid-19

testing and vaccines' among other things in Africa in collaboration with '24 organizations, businesses, and government agencies'. The pan-Africa initiative would focus on 10 countries: Burkina Faso, Ethiopia, Ghana, Kenya, Nigeria, Rwanda, South Africa, Tanzania, Uganda, and Zambia'. Rajiv Shah, President of the Rockefeller Foundation and former administrator of CIA-controlled USAID, said that if Africa was not mass-vaccinated (to change the DNA of its people) it was a 'threat to all of humanity' and not fair on Africans. When someone from the Rockefeller Foundation says they want to do something to help poor and deprived people and countries it is time for a belly-laugh. They are doing this out of the goodness of their 'heart' because 'vaccinating' the entire global population is what the 'Covid' hoax set out to achieve. Official 'decolonisation' of Africa by the Cult was merely a prelude to financial colonisation on the road to a return to physical colonisation. The 'vaccine' is vital to that and the sudden and convenient death of the 'Covid' sceptic president of Tanzania can be seen in its true light. A lot of people in Africa are aware that this is another form of colonisation and exploitation and they need to stand their ground.

The 'vaccine is working' scam

A potential problem for the Cult was that the 'vaccine' is meant to change human DNA and body messaging and not to protect anyone from a 'virus' never shown to exist. The vaccine couldn't work because it was not designed to work and how could they make it *appear* to be working so that more people would have it? This was overcome by lowering the amplification rate of the PCR test to produce fewer 'cases' and therefore fewer 'deaths'. Some of us had been pointing out since March, 2020, that the amplification rate of the test not testing for the 'virus' had been made artificially high to generate positive tests which they could call 'cases' to justify lockdowns. The World Health Organization recommended an absurdly high 45 amplification cycles to ensure the high positives required by the Cult and then remained silent on the issue until January 20th, 2021 – Biden's Inauguration Day. This was when the

'vaccinations' were seriously underway and on that day the WHO recommended after discussions with America's CDC that laboratories *lowered their testing amplification*. Dr David Samadi, a certified urologist and health writer, said the WHO was encouraging all labs to reduce their cycle count for PCR tests. He said the current cycle was much too high and was 'resulting in any particle being declared a positive case'. Even one mainstream news report I saw said this meant the number of 'Covid' infections may have been 'dramatically inflated'. Oh, just a little bit. The CDC in America issued new guidance to laboratories in April, 2021, to use 28 cycles *but only for 'vaccinated' people*. The timing of the CDC/WHO interventions were cynically designed to make it appear the 'vaccines' were responsible for falling cases and deaths when the real reason can be seen in the following examples. New York's state lab, the Wadsworth Center, identified 872 positive tests in July, 2020, based on a threshold of 40 cycles. When the figure was lowered to 35 cycles *43 percent* of the 872 were no longer 'positives'. At 30 cycles the figure was 63 percent. A Massachusetts lab found that between *85 to 90 percent* of people who tested positive in July with a cycle threshold of 40 would be negative at 30 cycles, Ashish Jha, MD, director of the Harvard Global Health Institute, said: 'I'm really shocked that it could be that high ... Boy, does it really change the way we need to be thinking about testing.' I'm shocked that I could see the obvious in the spring of 2020, with no medical background, and most medical professionals still haven't worked it out. No, that's not shocking – it's terrifying.

Three weeks after the WHO directive to lower PCR cycles the London *Daily Mail* ran this headline: 'Why ARE Covid cases plummeting? New infections have fallen 45% in the US and 30% globally in the past 3 weeks but experts say vaccine is NOT the main driver because only 8% of Americans and 13% of people worldwide have received their first dose.' They acknowledged that the drop could not be attributed to the 'vaccine', but soon this morphed throughout the media into the 'vaccine' has caused cases and deaths to fall when it was the PCR threshold. In December, 2020, there was

chaos at English Channel ports with truck drivers needing negative 'Covid' tests before they could board a ferry home for Christmas. The government wanted to remove the backlog as fast as possible and they brought in troops to do the 'testing'. Out of 1,600 drivers just 36 tested positive and the rest were given the all clear to cross the Channel. I guess the authorities thought that 36 was the least they could get away with without the unquestioning catching on. The amplification trick which most people believed in the absence of information in the mainstream applied more pressure on those refusing the 'vaccine' to succumb when it 'obviously worked'. The truth was the exact opposite with deaths in care homes soaring with the 'vaccine' and in Israel the term used was 'skyrocket'. A re-analysis of published data from the Israeli Health Ministry led by Dr Hervé Seligmann at the Medicine Emerging Infectious and Tropical Diseases at Aix-Marseille University found that Pfizer's 'Covid vaccine' killed 'about 40 times more [elderly] people than the disease itself would have killed' during a five-week vaccination period and 260 *times* more younger people than would have died from the 'virus' even according to the manipulated 'virus' figures. Dr Seligmann and his co-study author, Haim Yativ, declared after reviewing the Israeli 'vaccine' death data: 'This is a new Holocaust.'

Then, in mid-April, 2021, after vast numbers of people worldwide had been 'vaccinated', the story changed with clear coordination. The UK government began to prepare the ground for more future lockdowns when Nuremberg-destined Boris Johnson told yet another whopper. He said that cases had fallen because of *lockdowns* not 'vaccines'. Lockdowns are irrelevant when *there is no 'virus'* and the test and fraudulent death certificates are deciding the number of 'cases' and 'deaths'. Study after study has shown that lockdowns don't work and instead kill and psychologically destroy people. Meanwhile in the United States Anthony Fauci and Rochelle Walensky, the ultra-Zionist head of the CDC, peddled the same line. More lockdown was the answer and not the 'vaccine', a line repeated on cue by the moron that is Canadian Prime Minister Justin Trudeau. Why all the hysteria to get everyone 'vaccinated' if lockdowns and

not ‘vaccines’ made the difference? None of it makes sense on the face of it. Oh, but it does. The Cult wants lockdowns *and* the ‘vaccine’ and if the ‘vaccine’ is allowed to be seen as the total answer lockdowns would no longer be justified when there are still livelihoods to destroy. ‘Variants’ and renewed upward manipulation of PCR amplification are planned to instigate never-ending lockdown *and* more ‘vaccines’.

You must have it – we’re desperate

Israel, where the Jewish and Arab population are ruled by the Sabbatian Cult, was the front-runner in imposing the DNA-manipulating ‘vaccine’ on its people to such an extent that Jewish refusers began to liken what was happening to the early years of Nazi Germany. This would seem to be a fantastic claim. Why would a government of Jewish people be acting like the Nazis did? If you realise that the Sabbatian Cult was behind the Nazis and that Sabbatians hate Jews the pieces start to fit and the question of why a ‘Jewish’ government would treat Jews with such callous disregard for their lives and freedom finds an answer. Those controlling the government of Israel *aren’t Jewish* – they’re Sabbatian. Israeli lawyer Tamir Turgal was one who made the Nazi comparison in comments to German lawyer Reiner Fuellmich who is leading a class action lawsuit against the psychopaths for crimes against humanity. Turgal described how the Israeli government was vaccinating children and pregnant women on the basis that there was no evidence that this was dangerous when they had no evidence that it *wasn’t* dangerous either. They just had no evidence. This was medical experimentation and Turgal said this breached the Nuremberg Code about medical experimentation and procedures requiring informed consent and choice. Think about that. A Nuremberg Code developed because of Nazi experimentation on Jews and others in concentration camps by people like the evil-beyond-belief Josef Mengele is being breached by the *Israeli* government; but when you know that it’s a *Sabbatian* government along with its intelligence and military agencies like Mossad, Shin Bet and the Israeli Defense Forces, and that Sabbatians

were the force behind the Nazis, the kaleidoscope comes into focus. What have we come to when Israeli Jews are suing their government for violating the Nuremberg Code by essentially making Israelis subject to a medical experiment using the controversial 'vaccines'? It's a shocker that this has to be done in the light of what happened in Nazi Germany. The Anshe Ha-Emet, or 'People of the Truth', made up of Israeli doctors, lawyers, campaigners and public, have launched a lawsuit with the International Criminal Court. It says:

When the heads of the Ministry of Health as well as the prime minister presented the vaccine in Israel and began the vaccination of Israeli residents, the vaccinated were not advised, that, in practice, they are taking part in a medical experiment and that their consent is required for this under the Nuremberg Code.

The irony is unbelievable, but easily explained in one word: Sabbatians. The foundation of Israeli 'Covid' apartheid is the 'green pass' or 'green passport' which allows Jews and Arabs who have had the DNA-manipulating 'vaccine' to go about their lives – to work, fly, travel in general, go to shopping malls, bars, restaurants, hotels, concerts, gyms, swimming pools, theatres and sports venues, while non-'vaccinated' are banned from all those places and activities. Israelis have likened the 'green pass' to the yellow stars that Jews in Nazi Germany were forced to wear – the same as the yellow stickers that a branch of UK supermarket chain Morrisons told exempt mask-wears they had to display when shopping. How very sensitive. The Israeli system is blatant South African-style apartheid on the basis of compliance or non-compliance to fascism rather than colour of the skin. How appropriate that the Sabbatian Israeli government was so close to the pre-Mandela apartheid regime in Pretoria. The Sabbatian-instigated 'vaccine passport' in Israel is planned for everywhere. Sabbatians struck a deal with Pfizer that allowed them to lead the way in the percentage of a national population infused with synthetic material and the result was catastrophic. Israeli freedom activist Shai Dannon told me how chairs were appearing on beaches that said 'vaccinated only'. Health Minister Yuli Edelstein said that anyone unwilling or unable to get

the jabs that ‘confer immunity’ will be ‘left behind’. The man’s a liar. Not even the makers claim the ‘vaccines’ confer immunity. When you see those figures of ‘vaccine’ deaths these psychopaths were saying that you must take the chance the ‘vaccine’ will kill you or maim you while knowing it will change your DNA or lockdown for you will be permanent. That’s fascism. The Israeli parliament passed a law to allow personal information of the non-vaccinated to be shared with local and national authorities for three months. This was claimed by its supporters to be a way to ‘encourage’ people to be vaccinated. Hadas Ziv from Physicians for Human Rights described this as a ‘draconian law which crushed medical ethics and the patient rights’. But that’s the idea, the Sabbatians would reply.

Your papers, please

Sabbatian Israel was leading what has been planned all along to be a global ‘vaccine pass’ called a ‘green passport’ without which you would remain in permanent lockdown restriction and unable to do anything. This is how badly – *desperately* – the Cult is to get everyone ‘vaccinated’. The term and colour ‘green’ was not by chance and related to the psychology of fusing the perception of the green climate hoax with the ‘Covid’ hoax and how the ‘solution’ to both is the same Great Reset. Lying politicians, health officials and psychologists denied there were any plans for mandatory vaccinations or restrictions based on vaccinations, but they knew that was exactly what was meant to happen with governments of all countries reaching agreements to enforce a global system. ‘Free’ Denmark and ‘free’ Sweden unveiled digital vaccine certification. Cyprus, Czech Republic, Estonia, Greece, Hungary, Iceland, Italy, Poland, Portugal, Slovakia, and Spain have all committed to a vaccine passport system and the rest including the whole of the EU would follow. The satanic UK government will certainly go this way despite mendacious denials and at the time of writing it is trying to manipulate the public into having the ‘vaccine’ so they could go abroad on a summer holiday. How would that work without something to prove you had the synthetic toxicity injected into you?

Documents show that the EU's European Commission was moving towards 'vaccine certificates' in 2018 and 2019 before the 'Covid' hoax began. They knew what was coming. Abracadabra – Ursula von der Leyen, the German President of the Commission, announced in March, 2021, an EU 'Digital Green Certificate' – green again – to track the public's 'Covid status'. The passport sting is worldwide and the Far East followed the same pattern with South Korea ruling that only those with 'vaccination' passports – again the *green* pass – would be able to 'return to their daily lives'.

Bill Gates has been preparing for this 'passport' with other Cult operatives for years and beyond the paper version is a Gates-funded 'digital tattoo' to identify who has been vaccinated and who hasn't. The 'tattoo' is reported to include a substance which is externally readable to confirm who has been vaccinated. This is a bio-luminous light-generating enzyme (think fireflies) called ... *Luciferase*. Yes, named after the Cult 'god' Lucifer the 'light bringer' of whom more to come. Gates said he funded the readable tattoo to ensure children in the developing world were vaccinated and no one was missed out. He cares so much about poor kids as we know. This was just the cover story to develop a vaccine tagging system for everyone on the planet. Gates has been funding the ID2020 'alliance' to do just that in league with other lovely people at Microsoft, GAVI, the Rockefeller Foundation, Accenture and IDEO.org. He said in interviews in March, 2020, before any 'vaccine' publicly existed, that the world must have a globalised digital certificate to track the 'virus' and who had been vaccinated. Gates knew from the start that the mRNA vaccines were coming and when they would come and that the plan was to tag the 'vaccinated' to marginalise the intelligent and stop them doing anything including travel. Evil just doesn't suffice. Gates was exposed for offering a \$10 million bribe to the Nigerian House of Representatives to invoke compulsory 'Covid' vaccination of all Nigerians. Sara Cunial, a member of the Italian Parliament, called Gates a 'vaccine criminal'. She urged the Italian President to hand him over to the International Criminal Court for crimes against

humanity and condemned his plans to 'chip the human race' through ID2020.

You know it's a long-planned agenda when war criminal and Cult gofer Tony Blair is on the case. With the scale of arrogance only someone as dark as Blair can muster he said: 'Vaccination in the end is going to be your route to liberty.' Blair is a disgusting piece of work and he confirms that again. The media has given a lot of coverage to a bloke called Charlie Mullins, founder of London's biggest independent plumbing company, Pimlico Plumbers, who has said he won't employ anyone who has not been vaccinated or have them go to any home where people are not vaccinated. He said that if he had his way no one would be allowed to walk the streets if they have not been vaccinated. Gates was cheering at the time while I was alerting the white coats. The plan is that people will qualify for 'passports' for having the first two doses and then to keep it they will have to have all the follow ups and new ones for invented 'variants' until human genetics is transformed and many are dead who can't adjust to the changes. Hollywood celebrities – the usual propaganda stunt – are promoting something called the WELL Health-Safety Rating to verify that a building or space has 'taken the necessary steps to prioritize the health and safety of their staff, visitors and other stakeholders'. They included Lady Gaga, Jennifer Lopez, Michael B. Jordan, Robert DeNiro, Venus Williams, Wolfgang Puck, Deepak Chopra and 17th Surgeon General Richard Carmona. Yawn. WELL Health-Safety has big connections with China. Parent company Delos is headed by former Goldman Sachs partner Paul Scialla. This is another example – and we will see so many others – of using the excuse of 'health' to dictate the lives and activities of the population. I guess one confirmation of the 'safety' of buildings is that only 'vaccinated' people can go in, right?

Electronic concentration camps

I wrote decades ago about the plans to restrict travel and here we are for those who refuse to bow to tyranny. This can be achieved in one go with air travel if the aviation industry makes a blanket decree.

The ‘vaccine’ and guaranteed income are designed to be part of a global version of China’s social credit system which tracks behaviour 24/7 and awards or deletes ‘credits’ based on whether your behaviour is supported by the state or not. I mean your entire lifestyle – what you do, eat, say, everything. Once your credit score falls below a certain level consequences kick in. In China tens of millions have been denied travel by air and train because of this. All the locations and activities denied to refusers by the ‘vaccine’ passports will be included in one big mass ban on doing almost anything for those that don’t bow their head to government. It’s beyond fascist and a new term is required to describe its extremes – I guess fascist technocracy will have to do. The way the Chinese system of technological – technocratic – control is sweeping the West can be seen in the Los Angeles school system and is planned to be expanded worldwide. Every child is required to have a ‘Covid’-tracking app scanned daily before they can enter the classroom. The so-called Daily Pass tracking system is produced by Gates’ Microsoft which I’m sure will shock you rigid. The pass will be scanned using a barcode (one step from an inside-the-body barcode) and the information will include health checks, ‘Covid’ tests and vaccinations. Entry codes are for one specific building only and access will only be allowed if a student or teacher has a negative test with a test not testing for the ‘virus’, has no symptoms of anything alleged to be related to ‘Covid’ (symptoms from a range of other illness), and has a temperature under 100 degrees. No barcode, no entry, is planned to be the case for everywhere and not only schools.

Kids are being psychologically prepared to accept this as ‘normal’ their whole life which is why what they can impose in schools is so important to the Cult and its gofers. Long-time American freedom campaigner John Whitehead of the Rutherford Institute was not exaggerating when he said: ‘Databit by databit, we are building our own electronic concentration camps.’ Canada under its Cult gofer prime minister Justin Trudeau has taken a major step towards the real thing with people interned against their will if they test positive with a test not testing for the ‘virus’ when they arrive at a Canadian

airport. They are jailed in internment hotels often without food or water for long periods and with many doors failing to lock there have been sexual assaults. The interned are being charged sometimes \$2,000 for the privilege of being abused in this way. Trudeau is fully on board with the Cult and says the 'Covid pandemic' has provided an opportunity for a global 'reset' to permanently change Western civilisation. His number two, Deputy Prime Minister Chrystia Freeland, is a trustee of the World Economic Forum and a Rhodes Scholar. The Trudeau family have long been servants of the Cult. See *The Biggest Secret* and Cathy O'Brien's book *Trance-Formation of America* for the horrific background to Trudeau's father Pierre Trudeau another Canadian prime minister. Hide your fascism behind the façade of a heart-on-the-sleeve liberal. It's a well-honed Cult technique.

What can the 'vaccine' really do?

We have a 'virus' never shown to exist and 'variants' of the 'virus' that have also never been shown to exist except, like the 'original', as computer-generated fictions. Even if you believe there's a 'virus' the 'case' to 'death' rate is in the region of 0.23 to 0.15 percent and those 'deaths' are concentrated among the very old around the same average age that people die anyway. In response to this lack of threat (in truth none) psychopaths and idiots, knowingly and unknowingly answering to Gates and the Cult, are seeking to 'vaccinate' every man, woman and child on Planet Earth. Clearly the 'vaccine' is not about 'Covid' – none of this ever has been. So what is it all about *really*? Why the desperation to infuse genetically-manipulating synthetic material into everyone through mRNA fraudulent 'vaccines' with the intent of doing this over and over with the excuses of 'variants' and other 'virus' inventions? Dr Sherri Tenpenny, an osteopathic medical doctor in the United States, has made herself an expert on vaccines and their effects as a vehement campaigner against their use. Tenpenny was board certified in emergency medicine, the director of a level two trauma centre for 12 years, and moved to Cleveland in 1996 to start an integrative

medicine practice which has treated patients from all 50 states and some 17 other countries. Weaning people off pharmaceutical drugs is a speciality.

She became interested in the consequences of vaccines after attending a meeting at the National Vaccine Information Center in Washington DC in 2000 where she 'sat through four days of listening to medical doctors and scientists and lawyers and parents of vaccine injured kids' and asked: 'What's going on?' She had never been vaccinated and never got ill while her father was given a list of vaccines to be in the military and was 'sick his entire life'. The experience added to her questions and she began to examine vaccine documents from the Centers for Disease Control (CDC). After reading the first one, the 1998 version of *The General Recommendations of Vaccination*, she thought: 'This is it?' The document was poorly written and bad science and Tenpenny began 20 years of research into vaccines that continues to this day. She began her research into 'Covid vaccines' in March, 2020, and she describes them as 'deadly'. For many, as we have seen, they already have been. Tenpenny said that in the first 30 days of the 'vaccine' rollout in the United States there had been more than 40,000 adverse events reported to the vaccine adverse event database. A document had been delivered to her the day before that was 172 pages long. 'We have over 40,000 adverse events; we have over 3,100 cases of [potentially deadly] anaphylactic shock; we have over 5,000 neurological reactions.' Effects ranged from headaches to numbness, dizziness and vertigo, to losing feeling in hands or feet and paraesthesia which is when limbs 'fall asleep' and people have the sensation of insects crawling underneath their skin. All this happened in the first 30 days and remember that only about *ten percent* (or far less) of adverse reactions and vaccine-related deaths are estimated to be officially reported. Tenpenny said:

So can you think of one single product in any industry, any industry, for as long as products have been made on the planet that within 30 days we have 40,000 people complaining of side effects that not only is still on the market but ... we've got paid actors telling us how great

they are for getting their vaccine. We're offering people \$500 if they will just get their vaccine and we've got nurses and doctors going; 'I got the vaccine, I got the vaccine'.

Tenpenny said they were not going to be 'happy dancing folks' when they began to suffer Bell's palsy (facial paralysis), neuropathies, cardiac arrhythmias and autoimmune reactions that kill through a blood disorder. 'They're not going to be so happy, happy then, but we're never going to see pictures of those people' she said. Tenpenny described the 'vaccine' as 'a well-designed killing tool'.

No off-switch

Bad as the initial consequences had been Tenpenny said it would be maybe 14 months before we began to see the 'full ravage' of what is going to happen to the 'Covid vaccinated' with full-out consequences taking anything between two years and 20 years to show. You can understand why when you consider that variations of the 'Covid vaccine' use mRNA (messenger RNA) to in theory activate the immune system to produce protective antibodies without using the actual 'virus'. How can they when it's a computer program and they've never isolated what they claim is the 'real thing'? Instead they use *synthetic* mRNA. They are inoculating synthetic material into the body which through a technique known as the Trojan horse is absorbed into cells to change the nature of DNA. Human DNA is changed by an infusion of messenger RNA and with each new 'vaccine' of this type it is changed even more. Say so and you are banned by Cult Internet platforms. The contempt the contemptuous Mark Zuckerberg has for the truth and human health can be seen in an internal Facebook video leaked to the Project Veritas investigative team in which he said of the 'Covid vaccines': '... I share some caution on this because we just don't know the long term side-effects of basically modifying people's DNA and RNA.' At the same time this disgusting man's Facebook was censoring and banning anyone saying exactly the same. He must go before a Nuremberg trial for crimes against humanity when he *knows* that he

is censoring legitimate concerns and denying the right of informed consent on behalf of the Cult that owns him. People have been killed and damaged by the very ‘vaccination’ technique he cast doubt on himself when they may not have had the ‘vaccine’ with access to information that he denied them. The plan is to have at least annual ‘Covid vaccinations’, add others to deal with invented ‘variants’, and change all other vaccines into the mRNA system. Pfizer executives told shareholders at a virtual Barclays Global Healthcare Conference in March, 2021, that the public may need a third dose of ‘Covid vaccine’, plus regular yearly boosters and the company planned to hike prices to milk the profits in a ‘significant opportunity for our vaccine’. These are the professional liars, cheats and opportunists who are telling you their ‘vaccine’ is safe. Given this volume of mRNA planned to be infused into the human body and its ability to then replicate we will have a transformation of human genetics from biological to synthetic biological – exactly the long-time Cult plan for reasons we’ll see – and many will die. Sherri Tenpenny said of this replication:

It’s like having an on-button but no off-button and that whole mechanism ... they actually give it a name and they call it the Trojan horse mechanism, because it allows that [synthetic] virus and that piece of that [synthetic] virus to get inside of your cells, start to replicate and even get inserted into other parts of your DNA as a Trojan-horse.

Ask the overwhelming majority of people who have the ‘vaccine’ what they know about the contents and what they do and they would reply: ‘The government says it will stop me getting the virus.’ Governments give that false impression on purpose to increase take-up. You can read Sherri Tenpenny’s detailed analysis of the health consequences in her blog at Vaxxter.com, but in summary these are some of them. She highlights the statement by Bill Gates about how human beings can become their own ‘vaccine manufacturing machine’. The man is insane. [‘Vaccine’-generated] ‘antibodies’ carry synthetic messenger RNA into the cells and the damage starts, Tenpenny contends, and she says that lungs can be adversely affected through varying degrees of pus and bleeding which

obviously affects breathing and would be dubbed ‘Covid-19’. Even more sinister was the impact of ‘antibodies’ on macrophages, a white blood cell of the immune system. They consist of Type 1 and Type 2 which have very different functions. She said Type 1 are ‘hyper-vigilant’ white blood cells which ‘gobble up’ bacteria etc. However, in doing so, this could cause inflammation and in extreme circumstances be fatal. She says these affects are mitigated by Type 2 macrophages which kick in to calm down the system and stop it going rogue. They clear up dead tissue debris and reduce inflammation that the Type 1 ‘fire crews’ have caused. Type 1 kills the infection and Type 2 heals the damage, she says. This is her punchline with regard to ‘Covid vaccinations’: She says that mRNA ‘antibodies’ block Type 2 macrophages by attaching to them and deactivating them. This meant that when the Type 1 response was triggered by infection there was nothing to stop that getting out of hand by calming everything down. There’s an on-switch, but no off-switch, she says. What follows can be ‘over and out, see you when I see you’.

Genetic suicide

Tenpenny also highlights the potential for autoimmune disease – the body attacking itself – which has been associated with vaccines since they first appeared. Infusing a synthetic foreign substance into cells could cause the immune system to react in a panic believing that the body is being overwhelmed by an invader (it is) and the consequences can again be fatal. There is an autoimmune response known as a ‘cytokine storm’ which I have likened to a homeowner panicked by an intruder and picking up a gun to shoot randomly in all directions before turning the fire on himself. The immune system unleashes a storm of inflammatory response called cytokines to a threat and the body commits hara-kiri. The lesson is that you mess with the body’s immune response at your peril and these ‘vaccines’ seriously – fundamentally – mess with immune response. Tenpenny refers to a consequence called anaphylactic shock which is a severe and highly dangerous allergic reaction when the immune system

floods the body with chemicals. She gives the example of having a bee sting which primes the immune system and makes it sensitive to those chemicals. When people are stung again maybe years later the immune response can be so powerful that it leads to anaphylactic shock. Tenpenny relates this 'shock' with regard to the 'Covid vaccine' to something called polyethylene glycol or PEG. Enormous numbers of people have become sensitive to this over decades of use in a whole range of products and processes including food, drink, skin creams and 'medicine'. Studies have claimed that some 72 percent of people have antibodies triggered by PEG compared with two percent in the 1960s and allergic hypersensitive reactions to this become a gathering cause for concern. Tenpenny points out that the 'mRNA vaccine' is coated in a 'bubble' of polyethylene glycol which has the potential to cause anaphylactic shock through immune sensitivity. Many reports have appeared of people reacting this way after having the 'Covid vaccine'. What do we think is going to happen as humanity has more and more of these 'vaccines'?

Tenpenny said: 'All these pictures we have seen with people with these rashes ... these weepy rashes, big reactions on their arms and things like that – it's an acute allergic reaction most likely to the polyethylene glycol that you've been previously primed and sensitised to.'

Those who have not studied the conspiracy and its perpetrators at length might think that making the population sensitive to PEG and then putting it in these 'vaccines' is just a coincidence. It is not. It is instead testament to how carefully and coldly-planned current events have been and the scale of the conspiracy we are dealing with. Tenpenny further explains that the 'vaccine' mRNA procedure can breach the blood-brain barrier which protects the brain from toxins and other crap that will cause malfunction. In this case they could make two proteins corrupt brain function to cause Amyotrophic lateral sclerosis (ALS), a progressive nervous system disease leading to loss of muscle control, and frontal lobe degeneration – Alzheimer's and dementia. Immunologist J. Bart Classon published a paper connecting mRNA 'vaccines' to prion

disease which can lead to Alzheimer's and other forms of neurodegenerative disease while others have pointed out the potential to affect the placenta in ways that make women infertile. This will become highly significant in the next chapter when I will discuss other aspects of this non-vaccine that relate to its nanotechnology and transmission from the injected to the uninjected.

Qualified in idiocy

Tenpenny describes how research has confirmed that these 'vaccine'-generated antibodies can interact with a range of other tissues in the body and attack many other organs including the lungs. 'This means that if you have a hundred people standing in front of you that all got this shot they could have a hundred different symptoms.'

Anyone really think that Cult gofers like the Queen, Tony Blair, Christopher Whitty, Anthony Fauci, and all the other psychopaths have really had this 'vaccine' in the pictures we've seen? Not a bloody chance. Why don't doctors all tell us about all these dangers and consequences of the 'Covid vaccine'? Why instead do they encourage and pressure patients to have the shot? Don't let's think for a moment that doctors and medical staff can't be stupid, lazy, and psychopathic and that's without the financial incentives to give the jab. Tenpenny again:

Some people are going to die from the vaccine directly but a large number of people are going to start to get horribly sick and get all kinds of autoimmune diseases 42 days to maybe a year out. What are they going to do, these stupid doctors who say; 'Good for you for getting that vaccine.' What are they going to say; 'Oh, it must be a mutant, we need to give an extra dose of that vaccine.'

Because now the vaccine, instead of one dose or two doses we need three or four because the stupid physicians aren't taking the time to learn anything about it. If I can learn this sitting in my living room reading a 19 page paper and several others so can they. There's nothing special about me, I just take the time to do it.

Remember how Sara Kayat, the NHS and TV doctor, said that the 'Covid vaccine' would '100 percent prevent hospitalisation and death'. Doctors can be idiots like every other profession and they

should not be worshipped as infallible. They are not and far from it. Behind many medical and scientific ‘experts’ lies an uninformed prat trying to hide themselves from you although in the ‘Covid’ era many have failed to do so as with UK narrative-repeating ‘TV doctor’ Hilary Jones. Pushing back against the minority of proper doctors and scientists speaking out against the ‘vaccine’ has been the entire edifice of the Cult global state in the form of governments, medical systems, corporations, mainstream media, Silicon Valley, and an army of compliant doctors, medical staff and scientists willing to say anything for money and to enhance their careers by promoting the party line. If you do that you are an ‘expert’ and if you won’t you are an ‘anti-vaxxer’ and ‘Covidiot’. The pressure to be ‘vaccinated’ is incessant. We have even had reports claiming that the ‘vaccine’ can help cure cancer and Alzheimer’s and make the lame walk. I am waiting for the announcement that it can bring you coffee in the morning and cook your tea. Just as the symptoms of ‘Covid’ seem to increase by the week so have the miracles of the ‘vaccine’. American supermarket giant Kroger Co. offered nearly 500,000 employees in 35 states a \$100 bonus for having the ‘vaccine’ while donut chain Krispy Kreme promised ‘vaccinated’ customers a free glazed donut every day for the rest of 2021. Have your DNA changed and you will get a doughnut although we might not have to give you them for long. Such offers and incentives confirm the desperation.

Perhaps the worse vaccine-stunt of them all was UK ‘Health’ Secretary Matt-the-prat Hancock on live TV after watching a clip of someone being ‘vaccinated’ when the roll-out began. Hancock faked tears so badly it was embarrassing. Brain-of-Britain Piers Morgan, the lockdown-supporting, ‘vaccine’ supporting, ‘vaccine’ passport-supporting, TV host played along with Hancock – ‘You’re quite emotional about that’ he said in response to acting so atrocious it would have been called out at a school nativity which will presumably today include Mary and Jesus in masks, wise men keeping their camels six feet apart, and shepherds under tent arrest. System-serving Morgan tweeted this: ‘Love the idea of covid vaccine passports for everywhere: flights, restaurants, clubs, football, gyms,

shops etc. It's time covid-denying, anti-vaxxer loonies had their bullsh*t bluff called & bar themselves from going anywhere that responsible citizens go.' If only I could aspire to his genius. To think that Morgan, who specialises in shouting over anyone he disagrees with, was lauded as a free speech hero when he lost his job after storming off the set of his live show like a child throwing his dolly out of the pram. If he is a free speech hero we are in real trouble. I have no idea what 'bullsh*t' means, by the way, the * throws me completely.

The Cult is desperate to infuse its synthetic DNA-changing concoction into everyone and has been using every lie, trick and intimidation to do so. The question of '*Why?*' we shall now address.

CHAPTER TEN

Human 2.0

I believe that at the end of the century the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted –

Alan Turing (1912-1954), the ‘Father of artificial intelligence’

I have been exposing for decades the plan to transform the human body from a biological to a synthetic-biological state. The new human that I will call Human 2.0 is planned to be connected to artificial intelligence and a global AI ‘Smart Grid’ that would operate as one global system in which AI would control everything from your fridge to your heating system to your car to your mind. Humans would no longer be ‘human’, but post-human and sub-human, with their thinking and emotional processes replaced by AI.

What I said sounded crazy and beyond science fiction and I could understand that. To any balanced, rational, mind it *is* crazy. Today, however, that world is becoming reality and it puts the ‘Covid vaccine’ into its true context. Ray Kurzweil is the ultra-Zionist ‘computer scientist, inventor and futurist’ and co-founder of the Singularity University. Singularity refers to the merging of humans with machines or ‘transhumanism’. Kurzweil has said humanity would be connected to the cyber ‘cloud’ in the period of the ever-recurring year of 2030:

Our thinking ... will be a hybrid of biological and non-biological thinking ... humans will be able to extend their limitations and ‘think in the cloud’ ... We’re going to put gateways to the

cloud in our brains ... We're going to gradually merge and enhance ourselves ... In my view, that's the nature of being human – we transcend our limitations. As the technology becomes vastly superior to what we are then the small proportion that is still human gets smaller and smaller and smaller until it's just utterly negligible.

They are trying to sell this end-of-humanity-as-we-know-it as the next stage of 'evolution' when we become super-human and 'like the gods'. They are lying to you. Shocked, eh? The population, and again especially the young, have been manipulated into addiction to technologies designed to enslave them for life. First they induced an addiction to smartphones (holdables); next they moved to technology on the body (wearables); and then began the invasion of the body (implantables). I warned way back about the plan for microchipped people and we are now entering that era. We should not be diverted into thinking that this refers only to chips we can see. Most important are the nanochips known as smart dust, neural dust and nanobots which are far too small to be seen by the human eye. Nanotechnology is everywhere, increasingly in food products, and released into the atmosphere by the geoengineering of the skies funded by Bill Gates to 'shut out the Sun' and 'save the planet from global warming'. Gates has been funding a project to spray millions of tonnes of chalk (calcium carbonate) into the stratosphere over Sweden to 'dim the Sun' and cool the Earth. Scientists warned the move could be disastrous for weather systems in ways no one can predict and opposition led to the Swedish space agency announcing that the 'experiment' would not be happening as planned in the summer of 2021; but it shows where the Cult is going with dimming the impact of the Sun and there's an associated plan to change the planet's atmosphere. Who gives psychopath Gates the right to dictate to the entire human race and dismantle planetary systems? The world will not be safe while this man is at large.

The global warming hoax has made the Sun, like the gas of life, something to fear when both are essential to good health and human survival (more inversion). The body transforms sunlight into vital vitamin D through a process involving ... *cholesterol*. This is the cholesterol we are also told to fear. We are urged to take Big Pharma

statin drugs to reduce cholesterol and it's all systematic. Reducing cholesterol means reducing vitamin D uptake with all the multiple health problems that will cause. At least if you take statins long term it saves the government from having to pay you a pension. The delivery system to block sunlight is widely referred to as chemtrails although these have a much deeper agenda, too. They appear at first to be contrails or condensation trails streaming from aircraft into cold air at high altitudes. Contrails disperse very quickly while chemtrails do not and spread out across the sky before eventually their content falls to earth. Many times I have watched aircraft cross-cross a clear blue sky releasing chemtrails until it looks like a cloudy day. Chemtrails contain many things harmful to humans and the natural world including toxic heavy metals, aluminium (see Alzheimer's) and nanotechnology. Ray Kurzweil reveals the reason without actually saying so: 'Nanobots will infuse all the matter around us with information. Rocks, trees, everything will become these intelligent creatures.' How do you deliver that? *From the sky.* Self-replicating nanobots would connect everything to the Smart Grid. The phenomenon of Morgellons disease began in the chemtrail era and the correlation has led to it being dubbed the 'chemtrail disease'. Self-replicating fibres appear in the body that can be pulled out through the skin. Morgellons fibres continue to grow outside the body and have a form of artificial intelligence. I cover this at greater length in *Phantom Self*.

'Vaccine' operating system

'Covid vaccines' with their self-replicating synthetic material are also designed to make the connection between humanity and Kurzweil's 'cloud'. American doctor and dedicated campaigner for truth, Carrie Madej, an Internal Medicine Specialist in Georgia with more than 20 years medical experience, has highlighted the nanotechnology aspect of the fake 'vaccines'. She explains how one of the components in at least the Moderna and Pfizer synthetic potions are 'lipid nanoparticles' which are 'like little tiny computer bits' – a 'sci-fi substance' known as nanobots and hydrogel which can be 'triggered

at any moment to deliver its payload' and act as 'biosensors'. The synthetic substance had 'the ability to accumulate data from your body like your breathing, your respiration, thoughts and emotions, all kind of things' and each syringe could carry a *million* nanobots:

This substance because it's like little bits of computers in your body, crazy, but it's true, it can do that, [and] obviously has the ability to act through Wi-Fi. It can receive and transmit energy, messages, frequencies or impulses. That issue has never been addressed by these companies. What does that do to the human?

Just imagine getting this substance in you and it can react to things all around you, the 5G, your smart device, your phones, what is happening with that? What if something is triggering it, too, like an impulse, a frequency? We have something completely foreign in the human body.

Madej said her research revealed that electromagnetic (EMF) frequencies emitted by phones and other devices had increased dramatically in the same period of the 'vaccine' rollout and she was seeing more people with radiation problems as 5G and other electromagnetic technology was expanded and introduced to schools and hospitals. She said she was 'floored with the EMF coming off' the devices she checked. All this makes total sense and syncs with my own work of decades when you think that Moderna refers in documents to its mRNA 'vaccine' as an 'operating system':

Recognizing the broad potential of mRNA science, we set out to create an mRNA technology platform that functions very much like an operating system on a computer. It is designed so that it can plug and play interchangeably with different programs. In our case, the 'program' or 'app' is our mRNA drug – the unique mRNA sequence that codes for a protein ...

... Our mRNA Medicines – 'The Software Of Life': When we have a concept for a new mRNA medicine and begin research, fundamental components are already in place. Generally, the only thing that changes from one potential mRNA medicine to another is the coding region – the actual genetic code that instructs ribosomes to make protein. Utilizing these instruction sets gives our investigational mRNA medicines a software-like quality. We also have the ability to combine different mRNA sequences encoding for different proteins in a single mRNA investigational medicine.

Who needs a real ‘virus’ when you can create a computer version to justify infusing your operating system into the entire human race on the road to making living, breathing people into cyborgs? What is missed with the ‘vaccines’ is the *digital* connection between synthetic material and the body that I highlighted earlier with the study that hacked a computer with human DNA. On one level the body is digital, based on mathematical codes, and I’ll have more about that in the next chapter. Those who ridiculously claim that mRNA ‘vaccines’ are not designed to change human genetics should explain the words of Dr Tal Zaks, chief medical officer at Moderna, in a 2017 TED talk. He said that over the last 30 years ‘we’ve been living this phenomenal digital scientific revolution, and I’m here today to tell you, that we are actually *hacking the software of life*, and that it’s changing the way we think about prevention and treatment of disease’:

In every cell there’s this thing called messenger RNA, or mRNA for short, that transmits the critical information from the DNA in our genes to the protein, which is really the stuff we’re all made out of. This is the critical information that determines what the cell will do. So we think about it as an operating system. So if you could change that, if you could introduce a line of code, or change a line of code, it turns out, that has profound implications for everything, from the flu to cancer.

Zaks should more accurately have said that this has profound implications for the human genetic code and the nature of DNA. Communications within the body go both ways and not only one. But, hey, no, the ‘Covid vaccine’ will not affect your genetics. Cult fact-checkers say so even though the man who helped to develop the mRNA technique says that it does. Zaks said in 2017:

If you think about what it is we’re trying to do. We’ve taken information and our understanding of that information and how that information is transmitted in a cell, and we’ve taken our understanding of medicine and how to make drugs, and we’re fusing the two. We think of it as information therapy.

I have been writing for decades that the body is an information field communicating with itself and the wider world. This is why

radiation which is information can change the information field of body and mind through phenomena like 5G and change their nature and function. ‘Information therapy’ means to change the body’s information field and change the way it operates. DNA is a receiver-transmitter of information and can be mutated by information like mRNA synthetic messaging. Technology to do this has been ready and waiting in the underground bases and other secret projects to be rolled out when the ‘Covid’ hoax was played. ‘Trials’ of such short and irrelevant duration were only for public consumption. When they say the ‘vaccine’ is ‘experimental’ that is not true. It may appear to be ‘experimental’ to those who don’t know what’s going on, but the trials have already been done to ensure the Cult gets the result it desires. Zaks said that it took decades to sequence the human genome, completed in 2003, but now they could do it in a week. By ‘they’ he means scientists operating in the public domain. In the secret projects they were sequencing the genome in a week long before even 2003.

Deluge of mRNA

Highly significantly the Moderna document says the guiding premise is that if using mRNA as a medicine works for one disease then it should work for many diseases. They were leveraging the flexibility afforded by their platform and the fundamental role mRNA plays in protein synthesis to pursue mRNA medicines for a broad spectrum of diseases. Moderna is confirming what I was saying through 2020 that multiple ‘vaccines’ were planned for ‘Covid’ (and later invented ‘variants’) and that previous vaccines would be converted to the mRNA system to infuse the body with massive amounts of genetically-manipulating synthetic material to secure a transformation to a synthetic-biological state. The ‘vaccines’ are designed to kill stunning numbers as part of the long-exposed Cult depopulation agenda and transform the rest. Given this is the goal you can appreciate why there is such hysterical demand for every human to be ‘vaccinated’ for an alleged ‘disease’ that has an estimated ‘infection’ to ‘death’ ratio of 0.23-0.15 percent. As I write

children are being given the ‘vaccine’ in trials (their parents are a disgrace) and ever-younger people are being offered the vaccine for a ‘virus’ that even if you believe it exists has virtually zero chance of harming them. Horrific effects of the ‘trials’ on a 12-year-old girl were revealed by a family member to be serious brain and gastric problems that included a bowel obstruction and the inability to swallow liquids or solids. She was unable to eat or drink without throwing up, had extreme pain in her back, neck and abdomen, and was paralysed from the waist down which stopped her urinating unaided. When the girl was first taken to hospital doctors said it was all in her mind. She was signed up for the ‘trial’ by her parents for whom no words suffice. None of this ‘Covid vaccine’ insanity makes any sense unless you see what the ‘vaccine’ really is – a body-changer. Synthetic biology or ‘SynBio’ is a fast-emerging and expanding scientific discipline which includes everything from genetic and molecular engineering to electrical and computer engineering. Synthetic biology is defined in these ways:

- A multidisciplinary area of research that seeks to create new biological parts, devices, and systems, or to redesign systems that are already found in nature.
- The use of a mixture of physical engineering and genetic engineering to create new (and therefore synthetic) life forms.
- An emerging field of research that aims to combine the knowledge and methods of biology, engineering and related disciplines in the design of chemically-synthesized DNA to create organisms with novel or enhanced characteristics and traits (synthetic organisms including humans).

We now have synthetic blood, skin, organs and limbs being developed along with synthetic body parts produced by 3D printers. These are all elements of the synthetic human programme and this comment by Kurzweil’s co-founder of the Singularity University,

Peter Diamandis, can be seen in a whole new light with the 'Covid' hoax and the sanctions against those that refuse the 'vaccine':

Anybody who is going to be resisting the progress forward [to transhumanism] is going to be resisting evolution and, fundamentally, they will die out. It's not a matter of whether it's good or bad. It's going to happen.

'Resisting evolution'? What absolute bollocks. The arrogance of these people is without limit. His 'it's going to happen' mantra is another way of saying 'resistance is futile' to break the spirit of those pushing back and we must not fall for it. Getting this genetically-transforming 'vaccine' into everyone is crucial to the Cult plan for total control and the desperation to achieve that is clear for anyone to see. Vaccine passports are a major factor in this and they, too, are a form of resistance is futile. It's NOT. The paper funded by the Rockefeller Foundation for the 2013 'health conference' in China said:

We will interact more with artificial intelligence. The use of robotics, bio-engineering to augment human functioning is already well underway and will advance. Re-engineering of humans into potentially separate and unequal forms through genetic engineering or mixed human-robots raises debates on ethics and equality.

A new demography is projected to emerge after 2030 [that year again] of technologies (robotics, genetic engineering, nanotechnology) producing robots, engineered organisms, 'nanobots' and artificial intelligence (AI) that can self-replicate. Debates will grow on the implications of an impending reality of human designed life.

What is happening today is so long planned. The world army enforcing the will of the world government is intended to be a robot army, not a human one. Today's military and its technologically 'enhanced' troops, pilotless planes and driverless vehicles are just stepping stones to that end. Human soldiers are used as Cult fodder and its time they woke up to that and worked for the freedom of the population instead of their own destruction and their family's destruction – the same with the police. Join us and let's sort this out. The phenomenon of enforce my own destruction is widespread in the 'Covid' era with Woker 'luvvies' in the acting and entertainment

industries supporting ‘Covid’ rules which have destroyed their profession and the same with those among the public who put signs on the doors of their businesses ‘closed due to Covid – stay safe’ when many will never reopen. It’s a form of masochism and most certainly insanity.

Transgender = transhumanism

When something explodes out of nowhere and is suddenly everywhere it is always the Cult agenda and so it is with the tidal wave of claims and demands that have infiltrated every aspect of society under the heading of ‘transgenderism’. The term ‘trans’ is so ‘in’ and this is the dictionary definition:

A prefix meaning ‘across’, ‘through’, occurring ... in loanwords from Latin, used in particular for denoting movement or conveyance from place to place (transfer; transmit; transplant) or complete change (transform; transmute), or to form adjectives meaning ‘crossing’, ‘on the other side of’, or ‘going beyond’ the place named (transmontane; transnational; trans-Siberian).

Transgender means to go beyond gender and transhuman means to go beyond human. Both are aspects of the Cult plan to transform the human body to a synthetic state with *no gender*. Human 2.0 is not designed to procreate and would be produced technologically with no need for parents. The new human would mean the end of parents and so men, and increasingly women, are being targeted for the deletion of their rights and status. Parental rights are disappearing at an ever-quickening speed for the same reason. The new human would have no need for men or women when there is no procreation and no gender. Perhaps the transgender movement that appears to be in a permanent state of frenzy might now contemplate on how it is being used. This was never about transgender rights which are only the interim excuse for confusing gender, particularly in the young, on the road to *fusing* gender. Transgender activism is not an end; it is a *means* to an end. We see again the technique of creative destruction in which you destroy the status quo to ‘build back better’ in the form that you want. The gender status quo had to be

destroyed by persuading the Cult-created Woke mentality to believe that you can have 100 genders or more. A programme for 9 to 12 year olds produced by the Cult-owned BBC promoted the 100 genders narrative. The very idea may be the most monumental nonsense, but it is not what is true that counts, only what you can make people *believe* is true. Once the gender of $2 + 2 = 4$ has been dismantled through indoctrination, intimidation and $2 + 2 = 5$ then the new no-gender normal can take its place with Human 2.0.

Aldous Huxley revealed the plan in his prophetic *Brave New World* in 1932:

Natural reproduction has been done away with and children are created, 'decanted', and raised in 'hatcheries and conditioning centres'. From birth, people are genetically designed to fit into one of five castes, which are further split into 'Plus' and 'Minus' members and designed to fulfil predetermined positions within the social and economic strata of the World State.

How could Huxley know this in 1932? For the same reason George Orwell knew about the Big Brother state in 1948, Cult insiders I have quoted knew about it in 1969, and I have known about it since the early 1990s. If you are connected to the Cult or you work your balls off to uncover the plan you can predict the future. The process is simple. If there is a plan for the world and nothing intervenes to stop it then it will happen. Thus if you communicate the plan ahead of time you are perceived to have predicted the future, but you haven't. You have revealed the plan which without intervention will become the human future. The whole reason I have done what I have is to alert enough people to inspire an intervention and maybe at last that time has come with the Cult and its intentions now so obvious to anyone with a brain in working order.

The future is here

Technological wombs that Huxley described to replace parent procreation are already being developed and they are only the projects we know about in the public arena. Israeli scientists told *The Times of Israel* in March, 2021, that they have grown 250-cell embryos

into mouse foetuses with fully formed organs using artificial wombs in a development they say could pave the way for gestating humans outside the womb. Professor Jacob Hanna of the Weizmann Institute of Science said:

We took mouse embryos from the mother at day five of development, when they are just of 250 cells, and had them in the incubator from day five until day 11, by which point they had grown all their organs.

By day 11 they make their own blood and have a beating heart, a fully developed brain. Anybody would look at them and say, 'this is clearly a mouse foetus with all the characteristics of a mouse.' It's gone from being a ball of cells to being an advanced foetus.

A special liquid is used to nourish embryo cells in a laboratory dish and they float on the liquid to duplicate the first stage of embryonic development. The incubator creates all the right conditions for its development, Hanna said. The liquid gives the embryo 'all the nutrients, hormones and sugars they need' along with a custom-made electronic incubator which controls gas concentration, pressure and temperature. The cutting-edge in the underground bases and other secret locations will be light years ahead of that, however, and this was reported by the London *Guardian* in 2017:

We are approaching a biotechnological breakthrough. Ectogenesis, the invention of a complete external womb, could completely change the nature of human reproduction. In April this year, researchers at the Children's Hospital of Philadelphia announced their development of an artificial womb.

The article was headed 'Artificial wombs could soon be a reality. What will this mean for women?' What would it mean for children is an even bigger question. No mother to bond with only a machine in preparation for a life of soulless interaction and control in a world governed by machines (see the *Matrix* movies). Now observe the calculated manipulations of the 'Covid' hoax as human interaction and warmth has been curtailed by distancing, isolation and fear with people communicating via machines on a scale never seen before.

These are all dots in the same picture as are all the personal assistants, gadgets and children's toys through which kids and adults communicate with AI as if it is human. The AI 'voice' on Sat-Nav should be included. All these things are psychological preparation for the Cult endgame. Before you can make a physical connection with AI you have to make a psychological connection and that is what people are being conditioned to do with this ever gathering human-AI interaction. Movies and TV programmes depicting the transhuman, robot dystopia relate to a phenomenon known as 'pre-emptive programming' in which the world that is planned is portrayed everywhere in movies, TV and advertising. This is conditioning the conscious and subconscious mind to become familiar with the planned reality to dilute resistance when it happens for real. What would have been a shock such is the change is made less so. We have young children put on the road to transgender transition surgery with puberty blocking drugs at an age when they could never be able to make those life-changing decisions.

Rachel Levine, a professor of paediatrics and psychiatry who believes in treating children this way, became America's highest-ranked openly-transgender official when she was confirmed as US Assistant Secretary at the Department of Health and Human Services after being nominated by Joe Biden (the Cult). Activists and governments press for laws to deny parents a say in their children's transition process so the kids can be isolated and manipulated into agreeing to irreversible medical procedures. A Canadian father Robert Hoogland was denied bail by the Vancouver Supreme Court in 2021 and remained in jail for breaching a court order that he stay silent over his young teenage daughter, a minor, who was being offered life-changing hormone therapy without parental consent. At the age of 12 the girl's 'school counsellor' said she may be transgender, referred her to a doctor and told the school to treat her like a boy. This is another example of state-serving schools imposing ever more control over children's lives while parents have ever less.

Contemptible and extreme child abuse is happening all over the world as the Cult gender-fusion operation goes into warp-speed.

Why the war on men – and now women?

The question about what artificial wombs mean for women should rightly be asked. The answer can be seen in the deletion of women's rights involving sport, changing rooms, toilets and status in favour of people in male bodies claiming to identify as women. I can identify as a mountain climber, but it doesn't mean I can climb a mountain any more than a biological man can be a biological woman. To believe so is a triumph of belief over factual reality which is the very perceptual basis of everything Woke. Women's sport is being destroyed by allowing those with male bodies who say they identify as female to 'compete' with girls and women. Male body 'women' dominate 'women's' competition with their greater muscle mass, bone density, strength and speed. With that disadvantage sport for women loses all meaning. To put this in perspective nearly 300 American high school boys can run faster than the quickest woman sprinter in the world. Women are seeing their previously protected spaces invaded by male bodies simply because they claim to identify as women. That's all they need to do to access all women's spaces and activities under the Biden 'Equality Act' that destroys equality for women with the usual Orwellian Woke inversion. Male sex offenders have already committed rapes in women's prisons after claiming to identify as women to get them transferred. Does this not matter to the Woke 'equality' hypocrites? Not in the least. What matters to Cult manipulators and funders behind transgender activists is to advance gender fusion on the way to the no-gender 'human'. When you are seeking to impose transparent nonsense like this, or the 'Covid' hoax, the only way the nonsense can prevail is through censorship and intimidation of dissenters, deletion of factual information, and programming of the unquestioning, bewildered and naive. You don't have to scan the world for long to see that all these things are happening.

Many women's rights organisations have realised that rights and status which took such a long time to secure are being eroded and that it is systematic. Kara Dansky of the global Women's Human Rights Campaign said that Biden's transgender executive order immediately he took office, subsequent orders, and Equality Act legislation that followed 'seek to erase women and girls in the law as a category'. *Exactly.* I said during the long ago-started war on men (in which many women play a crucial part) that this was going to turn into a war on them. The Cult is phasing out *both* male and female genders. To get away with that they are brought into conflict so they are busy fighting each other while the Cult completes the job with no unity of response. Unity, people, *unity*. We need unity everywhere. Transgender is the only show in town as the big step towards the no-gender human. It's not about rights for transgender people and never has been. Woke political correctness is deleting words relating to genders to the same end. Wokers believe this is to be 'inclusive' when the opposite is true. They are deleting words describing gender because gender *itself* is being deleted by Human 2.0. Terms like 'man', 'woman', 'mother' and 'father' are being deleted in the universities and other institutions to be replaced by the *no*-gender, not trans-gender, 'individuals' and 'guardians'. Women's rights campaigner Maria Keffler of Partners for Ethical Care said: 'Children are being taught from kindergarten upward that some boys have a vagina, some girls have a penis, and that kids can be any gender they want to be.' Do we really believe that suddenly countries all over the world at the same time had the idea of having drag queens go into schools or read transgender stories to very young children in the local library? It's coldly-calculated confusion of gender on the way to the fusion of gender. Suzanne Vierling, a psychologist from Southern California, made another important point:

Yesterday's slave woman who endured gynecological medical experiments is today's girl-child being butchered in a booming gender-transitioning sector. Ovaries removed, pushing her into menopause and osteoporosis, uncharted territory, and parents' rights and authority decimated.

The erosion of parental rights is a common theme in line with the Cult plans to erase the very concept of parents and 'ovaries removed, pushing her into menopause' means what? Those born female lose the ability to have children – another way to discontinue humanity as we know it.

Eliminating Human 1.0 (before our very eyes)

To pave the way for Human 2.0 you must phase out Human 1.0. This is happening through plummeting sperm counts and making women infertile through an onslaught of chemicals, radiation (including smartphones in pockets of men) and mRNA 'vaccines'. Common agriculture pesticides are also having a devastating impact on human fertility. I have been tracking collapsing sperm counts in the books for a long time and in 2021 came a book by fertility scientist and reproductive epidemiologist Shanna Swan, *Count Down: How Our Modern World Is Threatening Sperm Counts, Altering Male and Female Reproductive Development and Imperiling the Future of the Human Race*. She reports how the global fertility rate dropped by half between 1960 and 2016 with America's birth rate 16 percent below where it needs to be to sustain the population. Women are experiencing declining egg quality, more miscarriages, and more couples suffer from infertility. Other findings were an increase in erectile dysfunction, infant boys developing more genital abnormalities, male problems with conception, and plunging levels of the male hormone testosterone which would explain why so many men have lost their backbone and masculinity. This has been very evident during the 'Covid' hoax when women have been prominent among the Pushbackers and big strapping blokes have bowed their heads, covered their faces with a nappy and quietly submitted. Mind control expert Cathy O'Brien also points to how global education introduced the concept of 'we're all winners' in sport and classrooms: 'Competition was defused, and it in turn defused a sense of fighting back.' This is another version of the 'equity' doctrine in which you drive down rather than raise up. What a contrast in Cult-controlled China with its global ambitions

where the government published plans in January, 2021, to 'cultivate masculinity' in boys from kindergarten through to high school in the face of a 'masculinity crisis'. A government adviser said boys would be soon become 'delicate, timid and effeminate' unless action was taken. Don't expect any similar policy in the targeted West. A 2006 study showed that a 65-year-old man in 2002 had testosterone levels 15 percent lower than a 65-year-old man in 1987 while a 2020 study found a similar story with young adults and adolescents. Men are getting prescriptions for testosterone replacement therapy which causes an even greater drop in sperm count with up to 99 percent seeing sperm counts drop to zero during the treatment. More sperm is defective and malfunctioning with some having two heads or not pursuing an egg.

A class of *synthetic* chemicals known as phthalates are being blamed for the decline. These are found everywhere in plastics, shampoos, cosmetics, furniture, flame retardants, personal care products, pesticides, canned foods and even receipts. Why till receipts? Everyone touches them. Let no one delude themselves that all this is not systematic to advance the long-time agenda for human body transformation. Phthalates mimic hormones and disrupt the hormone balance causing testosterone to fall and genital birth defects in male infants. Animals and fish have been affected in the same way due to phthalates and other toxins in rivers. When fish turn gay or change sex through chemicals in rivers and streams it is a pointer to why there has been such an increase in gay people and the sexually confused. It doesn't matter to me what sexuality people choose to be, but if it's being affected by chemical pollution and consumption then we need to know. Does anyone really think that this is not connected to the transgender agenda, the war on men and the condemnation of male 'toxic masculinity'? You watch this being followed by 'toxic femininity'. It's already happening. When breastfeeding becomes 'chest-feeding', pregnant women become pregnant people along with all the other Woke claptrap you know that the world is going insane and there's a Cult scam in progress. Transgender activists are promoting the Cult agenda while Cult

billionaires support and fund the insanity as they laugh themselves to sleep at the sheer stupidity for which humans must be infamous in galaxies far, far away.

'Covid vaccines' and female infertility

We can now see why the 'vaccine' has been connected to potential infertility in women. Dr Michael Yeadon, former Vice President and Chief Scientific Advisor at Pfizer, and Dr Wolfgang Wodarg in Germany, filed a petition with the European Medicines Agency in December, 2020, urging them to stop trials for the Pfizer/BioNTech shot and all other mRNA trials until further studies had been done. They were particularly concerned about possible effects on fertility with 'vaccine'-produced antibodies attacking the protein Syncytin-1 which is responsible for developing the placenta. The result would be infertility 'of indefinite duration' in women who have the 'vaccine' with the placenta failing to form. Section 10.4.2 of the Pfizer/BioNTech trial protocol says that pregnant women or those who might become so should not have mRNA shots. Section 10.4 warns men taking mRNA shots to 'be abstinent from heterosexual intercourse' and not to donate sperm. The UK government said that it *did not know* if the mRNA procedure had an effect on fertility. *Did not know?* These people have to go to jail. UK government advice did not recommend at the start that pregnant women had the shot and said they should avoid pregnancy for at least two months after 'vaccination'. The 'advice' was later updated to pregnant women should only have the 'vaccine' if the benefits outweighed the risks to mother and foetus. What the hell is that supposed to mean? Then 'spontaneous abortions' began to appear and rapidly increase on the adverse reaction reporting schemes which include only a fraction of adverse reactions. Thousands and ever-growing numbers of 'vaccinated' women are describing changes to their menstrual cycle with heavier blood flow, irregular periods and menstruating again after going through the menopause – all links to reproduction effects. Women are passing blood clots and the lining of their uterus while men report erectile dysfunction and blood effects. Most

significantly of all *unvaccinated* women began to report similar menstrual changes after interaction with '*vaccinated*' people and men and children were also affected with bleeding noses, blood clots and other conditions. 'Shedding' is when vaccinated people can emit the content of a vaccine to affect the unvaccinated, but this is different. '*Vaccinated*' people were not shedding a 'live virus' allegedly in '*vaccines*' as before because the fake '*Covid vaccines*' involve synthetic material and other toxicity. Doctors exposing what is happening prefer the term '*transmission*' to shedding. Somehow those that have had the shots are transmitting effects to those that haven't. Dr Carrie Madej said the nano-content of the '*vaccines*' can 'act like an antenna' to others around them which fits perfectly with my own conclusions. This '*vaccine*' transmission phenomenon was becoming known as the book went into production and I deal with this further in the Postscript.

Vaccine effects on sterility are well known. The World Health Organization was accused in 2014 of sterilising millions of women in Kenya with the evidence confirmed by the content of the vaccines involved. The same WHO behind the '*Covid*' hoax admitted its involvement for more than ten years with the vaccine programme. Other countries made similar claims. Charges were lodged by Tanzania, Nicaragua, Mexico, and the Philippines. The Gardasil vaccine claimed to protect against a genital '*virus*' known as HPV has also been linked to infertility. Big Pharma and the WHO (same thing) are criminal and satanic entities. Then there's the Bill Gates Foundation which is connected through funding and shared interests with 20 pharmaceutical giants and laboratories. He stands accused of directing the policy of United Nations Children's Fund (UNICEF), vaccine alliance GAVI, and other groupings, to advance the vaccine agenda and silence opposition at great cost to women and children. At the same time Gates wants to reduce the global population. Coincidence?

Great Reset = Smart Grid = new human

The Cult agenda I have been exposing for 30 years is now being openly promoted by Cult assets like Gates and Klaus Schwab of the World Economic Forum under code-terms like the 'Great Reset', 'Build Back Better' and 'a rare but narrow window of opportunity to reflect, reimagine, and reset our world'. What provided this 'rare but narrow window of opportunity'? The 'Covid' hoax did. Who created that? *They* did. My books from not that long ago warned about the planned 'Internet of Things' (IoT) and its implications for human freedom. This was the plan to connect all technology to the Internet and artificial intelligence and today we are way down that road with an estimated 36 billion devices connected to the World Wide Web and that figure is projected to be 76 billion by 2025. I further warned that the Cult planned to go beyond that to the Internet of *Everything* when the human brain was connected via AI to the Internet and Kurzweil's 'cloud'. Now we have Cult operatives like Schwab calling for precisely that under the term 'Internet of Bodies', a fusion of the physical, digital and biological into one centrally-controlled Smart Grid system which the Cult refers to as the 'Fourth Industrial Revolution'. They talk about the 'biological', but they really mean the synthetic-biological which is required to fully integrate the human body and brain into the Smart Grid and artificial intelligence planned to replace the human mind. We have everything being synthetically manipulated including the natural world through GMO and smart dust, the food we eat and the human body itself with synthetic 'vaccines'. I said in *The Answer* that we would see the Cult push for synthetic meat to replace animals and in February, 2021, the so predictable psychopath Bill Gates called for the introduction of synthetic meat to save us all from 'climate change'. The climate hoax just keeps on giving like the 'Covid' hoax. The war on meat by vegan activists is a carbon (oops, sorry) copy of the manipulation of transgender activists. They have no idea (except their inner core) that they are being used to promote and impose the agenda of the Cult or that they are only the *vehicle* and not the *reason*. This is not to say those who choose not to eat meat shouldn't be respected and supported in that right, but there are ulterior motives

for those in power. A *Forbes* article in December, 2019, highlighted the plan so beloved of Schwab and the Cult under the heading: 'What Is The Internet of Bodies? And How Is It Changing Our World?' The article said the human body is the latest data platform (remember 'our vaccine is an operating system'). *Forbes* described the plan very accurately and the words could have come straight out of my books from long before:

The Internet of Bodies (IoB) is an extension of the IoT and basically connects the human body to a network through devices that are ingested, implanted, or connected to the body in some way. Once connected, data can be exchanged, and the body and device can be remotely monitored and controlled.

They were really describing a human hive mind with human perception centrally-dictated via an AI connection as well as allowing people to be 'remotely monitored and controlled'.

Everything from a fridge to a human mind could be directed from a central point by these insane psychopaths and 'Covid vaccines' are crucial to this. *Forbes* explained the process I mentioned earlier of holdable and wearable technology followed by implantable. The article said there were three generations of the Internet of Bodies that include:

- Body external: These are wearable devices such as Apple Watches or Fitbits that can monitor our health.
- Body internal: These include pacemakers, cochlear implants, and digital pills that go inside our bodies to monitor or control various aspects of health.
- Body embedded: The third generation of the Internet of Bodies is embedded technology where technology and the human body are melded together and have a real-time connection to a remote machine.

Forbes noted the development of the Brain Computer Interface (BCI) which merges the brain with an external device for monitoring and controlling in real-time. ‘The ultimate goal is to help restore function to individuals with disabilities by using brain signals rather than conventional neuromuscular pathways.’ Oh, do fuck off. The goal of brain interface technology is controlling human thought and emotion from the central point in a hive mind serving its masters wishes. Many people are now agreeing to be chipped to open doors without a key. You can recognise them because they’ll be wearing a mask, social distancing and lining up for the ‘vaccine’. The Cult plans a Great Reset money system after they have completed the demolition of the global economy in which ‘money’ will be exchanged through communication with body operating systems. Rand Corporation, a Cult-owned think tank, said of the Internet of Bodies or IoB:

Internet of Bodies technologies fall under the broader IoT umbrella. But as the name suggests, IoB devices introduce an even more intimate interplay between humans and gadgets. IoB devices monitor the human body, collect health metrics and other personal information, and transmit those data over the Internet. Many devices, such as fitness trackers, are already in use ... IoB devices ... and those in development can track, record, and store users’ whereabouts, bodily functions, and what they see, hear, and even think.

Schwab’s World Economic Forum, a long-winded way of saying ‘fascism’ or ‘the Cult’, has gone full-on with the Internet of Bodies in the ‘Covid’ era. ‘We’re entering the era of the Internet of Bodies’, it declared, ‘collecting our physical data via a range of devices that can be implanted, swallowed or worn’. The result would be a huge amount of health-related data that could improve human wellbeing around the world, and prove crucial in fighting the ‘Covid-19 pandemic’. Does anyone think these clowns care about ‘human wellbeing’ after the death and devastation their pandemic hoax has purposely caused? Schwab and co say we should move forward with the Internet of Bodies because ‘Keeping track of symptoms could help us stop the spread of infection, and quickly detect new cases’. How wonderful, but keeping track’ is all they are really bothered

about. Researchers were investigating if data gathered from smartwatches and similar devices could be used as viral infection alerts by tracking the user's heart rate and breathing. Schwab said in his 2018 book *Shaping the Future of the Fourth Industrial Revolution*:

The lines between technologies and beings are becoming blurred and not just by the ability to create lifelike robots or synthetics. Instead it is about the ability of new technologies to literally become part of us. Technologies already influence how we understand ourselves, how we think about each other, and how we determine our realities. As the technologies ... give us deeper access to parts of ourselves, we may begin to integrate digital technologies into our bodies.

You can see what the game is. Twenty-four hour control and people – if you could still call them that – would never know when something would go ping and take them out of circulation. It's the most obvious rush to a global fascist dictatorship and the complete submission of humanity and yet still so many are locked away in their Cult-induced perceptual coma and can't see it.

Smart Grid control centres

The human body is being transformed by the 'vaccines' and in other ways into a synthetic cyborg that can be attached to the global Smart Grid which would be controlled from a central point and other sub-locations of Grid manipulation. Where are these planned to be? Well, China for a start which is one of the Cult's biggest centres of operation. The technological control system and technocratic rule was incubated here to be unleashed across the world after the 'Covid' hoax came out of China in 2020. Another Smart Grid location that will surprise people new to this is Israel. I have exposed in *The Trigger* how Sabbatian technocrats, intelligence and military operatives were behind the horrors of 9/11 and not 19 Arab hijackers' who somehow manifested the ability to pilot big passenger airliners when instructors at puddle-jumping flying schools described some of them as a joke. The 9/11 attacks were made possible through control of civilian and military air computer systems and those of the White House, Pentagon and connected agencies. See *The Trigger* – it

will blow your mind. The controlling and coordinating force were the Sabbatian networks in Israel and the United States which by then had infiltrated the entire US government, military and intelligence system. The real name of the American Deep State is 'Sabbatian State'. Israel is a tiny country of only nine million people, but it is one of the global centres of cyber operations and fast catching Silicon Valley in importance to the Cult. Israel is known as the 'start-up nation' for all the cyber companies spawned there with the Sabbatian specialisation of 'cyber security' that I mentioned earlier which gives those companies access to computer systems of their clients in real time through 'backdoors' written into the coding when security software is downloaded. The Sabbatian centre of cyber operations outside Silicon Valley is the Israeli military Cyber Intelligence Unit, the biggest infrastructure project in Israel's history, headquartered in the desert-city of Beersheba and involving some 20,000 'cyber soldiers'. Here are located a literal army of Internet trolls scanning social media, forums and comment lists for anyone challenging the Cult agenda. The UK military has something similar with its 77th Brigade and associated operations. The Beersheba complex includes research and development centres for other Cult operations such as Intel, Microsoft, IBM, Google, Apple, Hewlett-Packard, Cisco Systems, Facebook and Motorola. [Techcrunch.com](#) ran an article about the Beersheba global Internet technology centre headlined 'Israel's desert city of Beersheba is turning into a cybertech oasis':

The military's massive relocation of its prestigious technology units, the presence of multinational and local companies, a close proximity to Ben Gurion University and generous government subsidies are turning Beersheba into a major global cybertech hub. Beersheba has all of the ingredients of a vibrant security technology ecosystem, including Ben Gurion University with its graduate program in cybersecurity and Cyber Security Research Center, and the presence of companies such as EMC, Deutsche Telekom, PayPal, Oracle, IBM, and Lockheed Martin. It's also the future home of the INCB (Israeli National Cyber Bureau); offers a special income tax incentive for cyber security companies, and was the site for the relocation of the army's intelligence corps units.

Sabbatians have taken over the cyber world through the following process: They scan the schools for likely cyber talent and develop them at Ben Gurion University and their period of conscription in the Israeli Defense Forces when they are stationed at the Beersheba complex. When the cyber talented officially leave the army they are funded to start cyber companies with technology developed by themselves or given to them by the state. Much of this is stolen through backdoors of computer systems around the world with America top of the list. Others are sent off to Silicon Valley to start companies or join the major ones and so we have many major positions filled by apparently 'Jewish' but really Sabbatian operatives. Google, YouTube and Facebook are all run by 'Jewish' CEOs while Twitter is all but run by ultra-Zionist hedge-fund shark Paul Singer. At the centre of the Sabbatian global cyber web is the Israeli army's Unit 8200 which specialises in hacking into computer systems of other countries, inserting viruses, gathering information, instigating malfunction, and even taking control of them from a distance. A long list of Sabbatians involved with 9/11, Silicon Valley and Israeli cyber security companies are operatives of Unit 8200. This is not about Israel. It's about the Cult. Israel is planned to be a Smart Grid hub as with China and what is happening at Beersheba is not for the benefit of Jewish people who are treated disgustingly by the Sabbatian elite that control the country. A glance at the Nuremberg Codes will tell you that.

The story is much bigger than 'Covid', important as that is to where we are being taken. Now, though, it's time to really strap in. There's more ... much more ...

CHAPTER ELEVEN

Who controls the Cult?

Awake, arise or be forever fall'n
John Milton, *Paradise Lost*

I have exposed this far the level of the Cult conspiracy that operates in the world of the seen and within the global secret society and satanic network which operates in the shadows one step back from the seen. The story, however, goes much deeper than that.

The 'Covid' hoax is major part of the Cult agenda, but only part, and to grasp the biggest picture we have to expand our attention beyond the realm of human sight and into the infinity of possibility that we cannot see. It is from here, ultimately, that humanity is being manipulated into a state of total control by the force which dictates the actions of the Cult. How much of reality can we see? Next to damn all is the answer. We may appear to see all there is to see in the 'space' our eyes survey and observe, but little could be further from the truth. The human 'world' is only a tiny band of frequency that the body's visual and perceptual systems can decode into *perception* of a 'world'. According to mainstream science the electromagnetic spectrum is 0.005 percent of what exists in the Universe ([Fig 10](#)). The maximum estimate I have seen is 0.5 percent and either way it's minuscule. I say it is far, far, smaller even than 0.005 percent when you compare reality we see with the totality of reality that we don't. Now get this if you are new to such information: Visible light, the only band of frequency that we can see, is a *fraction* of the 0.005

percent (Fig 11 overleaf). Take this further and realise that our universe is one of infinite universes and that universes are only a fragment of overall reality – *infinite* reality. Then compare that with the almost infinitesimal frequency band of visible light or human sight. You see that humans are as near blind as it is possible to be without actually being so. Artist and filmmaker, Sergio Toporek, said:

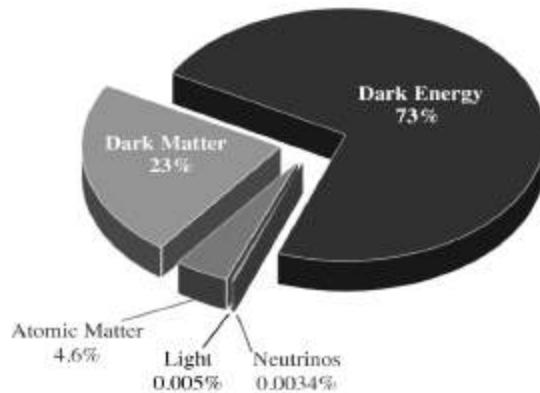


Figure 10: Humans can perceive such a tiny band of visual reality it's laughable.

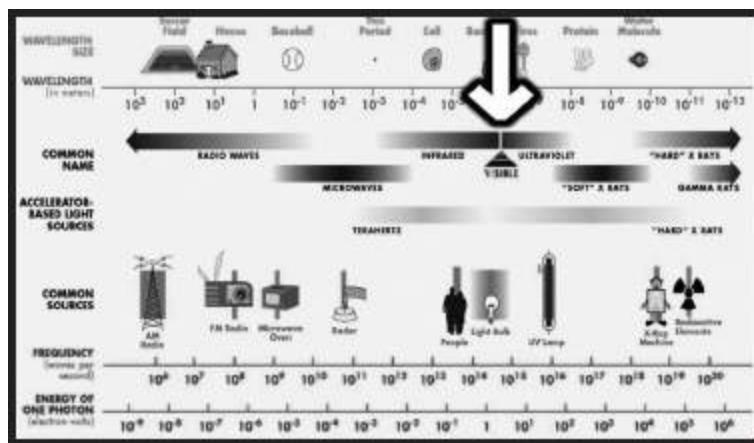


Figure 11: We can see a smear of the 0.005 percent electromagnetic spectrum, but we still know it all. Yep, makes sense.

Consider that you can see less than 1% of the electromagnetic spectrum and hear less than 1% of the acoustic spectrum. 90% of the cells in your body carry their own microbial DNA and are not 'you'. The atoms in your body are 99.99999999999999% empty space and none of them are the ones you were born with ... Human beings have 46 chromosomes, two less than a potato.

The existence of the rainbow depends on the conical photoreceptors in your eyes; to animals without cones, the rainbow does not exist. So you don't just look at a rainbow, you create it. This is pretty amazing, especially considering that all the beautiful colours you see represent less than 1% of the electromagnetic spectrum.

Suddenly the 'world' of humans looks a very different place. Take into account, too, that Planet Earth when compared with the projected size of this single universe is the equivalent of a billionth of a pinhead. Imagine the ratio that would be when compared to infinite reality. To think that Christianity once insisted that Earth and humanity were the centre of everything. This background is vital if we are going to appreciate the nature of 'human' and how we can be manipulated by an unseen force. To human visual reality virtually *everything* is unseen and yet the prevailing perception within the institutions and so much of the public is that if we can't see it, touch it, hear it, taste it and smell it then it cannot exist. Such perception is indoctrinated and encouraged by the Cult and its agents because it isolates believers in the strictly limited, village-idiot, realm of the five senses where perceptions can be firewalled and information controlled. Most of those perpetuating the 'this-world-is-all-there-is' insanity are themselves indoctrinated into believing the same delusion. While major players and influencers know that official reality is laughable most of those in science, academia and medicine really believe the nonsense they peddle and teach succeeding generations. Those who challenge the orthodoxy are dismissed as nutters and freaks to protect the manufactured illusion from exposure. Observe the dynamic of the 'Covid' hoax and you will see how that takes the same form. The inner-circle psychopaths know it's a gigantic scam, but almost the entirety of those imposing their fascist rules believe that 'Covid' is all that they're told it is.

Stolen identity

Ask people who they are and they will give you their name, place of birth, location, job, family background and life story. Yet that is not who they are – it is what they are *experiencing*. The difference is *absolutely crucial*. The true 'I', the eternal, infinite 'I', is consciousness,

a state of being aware. Forget ‘form’. That is a vehicle for a brief experience. Consciousness does not come *from* the brain, but *through* the brain and even that is more symbolic than literal. We are awareness, pure awareness, and this is what withdraws from the body at what we call ‘death’ to continue our eternal beingness, *isness*, in other realms of reality within the limitlessness of infinity or the Biblical ‘many mansions in my father’s house’. Labels of a human life, man, woman, transgender, black, white, brown, nationality, circumstances and income are not who we are. They are what we are – awareness – is *experiencing* in a brief connection with a band of frequency we call ‘human’. The labels are not the self; they are, to use the title of one of my books, a *Phantom Self*. I am not David Icke born in Leicester, England, on April 29th, 1952. I am the consciousness *having that experience*. The Cult and its non-human masters seek to convince us through the institutions of ‘education’, science, medicine, media and government that what we are *experiencing* is who we *are*. It’s so easy to control and direct perception locked away in the bewildered illusions of the five senses with no expanded radar. Try, by contrast, doing the same with a humanity aware of its true self and its true power to consciously create its reality and experience. How is it possible to do this? We do it all day every day. If you perceive yourself as ‘little me’ with no power to impact upon your life and the world then your life experience will reflect that. You will hand the power you don’t think you have to authority in all its forms which will use it to control your experience. This, in turn, will appear to confirm your perception of ‘little me’ in a self-fulfilling feedback loop. But that is what ‘little me’ really is – a *perception*. We are all ‘big-me’, infinite me, and the Cult has to make us forget that if its will is to prevail. We are therefore manipulated and pressured into self-identifying with human labels and not the consciousness/awareness *experiencing* those human labels.

The phenomenon of identity politics is a Cult-instigated manipulation technique to sub-divide previous labels into even smaller ones. A United States university employs this list of letters to

describe student identity: LGBTQQFAGPBDSM or lesbian, gay, bisexual, transgender, transsexual, queer, questioning, flexual, asexual, gender-fuck, polyamorous, bondage/discipline, dominance/submission and sadism/masochism. I'm sure other lists are even longer by now as people feel the need to self-identify the 'I' with the minutiae of race and sexual preference. Wokers programmed by the Cult for generations believe this is about 'inclusivity' when it's really the Cult locking them away into smaller and smaller versions of Phantom Self while firewalls them from the influence of their true self, the infinite, eternal 'I'. You may notice that my philosophy which contends that we are all unique points of attention/awareness within the same infinite whole or Oneness is the ultimate non-racism. The very sense of Oneness makes the judgement of people by their body-type, colour or sexuality utterly ridiculous and confirms that racism has no understanding of reality (including anti-white racism). Yet despite my perception of life Cult agents and fast-asleep Wokers label me racist to discredit my information while they are themselves phenomenally racist and sexist. All they see is race and sexuality and they judge people as good or bad, demons or untouchables, by their race and sexuality. All they see is *Phantom Self* and perceive themselves in terms of *Phantom Self*. They are pawns and puppets of the Cult agenda to focus attention and self-identity in the five senses and play those identities against each other to divide and rule. Columbia University has introduced segregated graduations in another version of social distancing designed to drive people apart and teach them that different racial and cultural groups have nothing in common with each other. The last thing the Cult wants is unity. Again the pump-primers of this will be Cult operatives in the knowledge of what they are doing, but the rest are just the *Phantom Self* blind leading the *Phantom Self* blind. We *do* have something in common – we are all *the same consciousness* having different temporary experiences.

What is this 'human'?

Yes, what *is* ‘human’? That is what we are supposed to be, right? I mean ‘human’? True, but ‘human’ is the experience not the ‘I’. Break it down to basics and ‘human’ is the way that information is processed. If we are to experience and interact with this band of frequency we call the ‘world’ we must have a vehicle that operates within that band of frequency. Our consciousness in its prime form cannot do that; it is way beyond the frequency of the human realm. My consciousness or awareness could not tap these keys and pick up the cup in front of me in the same way that radio station A cannot interact with radio station B when they are on different frequencies. The human body is the means through which we have that interaction. I have long described the body as a biological computer which processes information in a way that allows consciousness to experience this reality. The body is a receiver, transmitter and processor of information in a particular way that we call human. We visually perceive only the world of the five senses in a wakened state – that is the limit of the body’s visual decoding system. In truth it’s not even visual in the way we experience ‘visual reality’ as I will come to in a moment. We are ‘human’ because the body processes the information sources of human into a reality and behaviour system that we *perceive* as human. Why does an elephant act like an elephant and not like a human or a duck? The elephant’s biological computer is a different information field and processes information according to that program into a visual and behaviour type we call an elephant. The same applies to everything in our reality. These body information fields are perpetuated through procreation (like making a copy of a software program). The Cult wants to break that cycle and intervene technologically to transform the human information field into one that will change what we call humanity. If it can change the human information field it will change the way that field processes information and change humanity both ‘physically’ and psychologically. Hence the *messenger* (information) RNA ‘vaccines’ and so much more that is targeting human genetics by changing the body’s information – *messaging* – construct through food, drink, radiation, toxicity and other means.

Reality that we experience is nothing like reality as it really is in the same way that the reality people experience in virtual reality games is not the reality they are really living in. The game is only a decoded source of information that appears to be a reality. Our world is also an information construct – a *simulation* (more later). In its base form our reality is a wavefield of information much the same in theme as Wi-Fi. The five senses decode wavefield information into electrical information which they communicate to the brain to decode into holographic (illusory ‘physical’) information. Different parts of the brain specialise in decoding different senses and the information is fused into a reality that appears to be outside of us but is really inside the brain and the genetic structure in general ([Fig 12](#) overleaf). DNA is a receiver-transmitter of information and a vital part of this decoding process and the body’s connection to other realities. Change DNA and you change the way we decode and connect with reality – see ‘Covid vaccines’. Think of computers decoding Wi-Fi. You have information encoded in a radiation field and the computer decodes that information into a very different form on the screen. You can’t see the Wi-Fi until its information is made manifest on the screen and the information on the screen is inside the computer and not outside. I have just described how we decode the ‘human world’. All five senses decode the waveform ‘Wi-Fi’ field into electrical signals and the brain (computer) constructs reality inside the brain and not outside – ‘You don’t just look at a rainbow, you create it’. Sound is a simple example. We don’t hear sound until the brain decodes it. Waveform sound waves are picked up by the hearing sense and communicated to the brain in an electrical form to be decoded into the sounds that we hear. Everything we hear is inside the brain along with everything we see, feel, smell and taste. Words and language are waveform fields generated by our vocal chords which pass through this process until they are decoded by the brain into words that we hear. Different languages are different frequency fields or sound waves generated by vocal chords. Late British philosopher Alan Watts said:

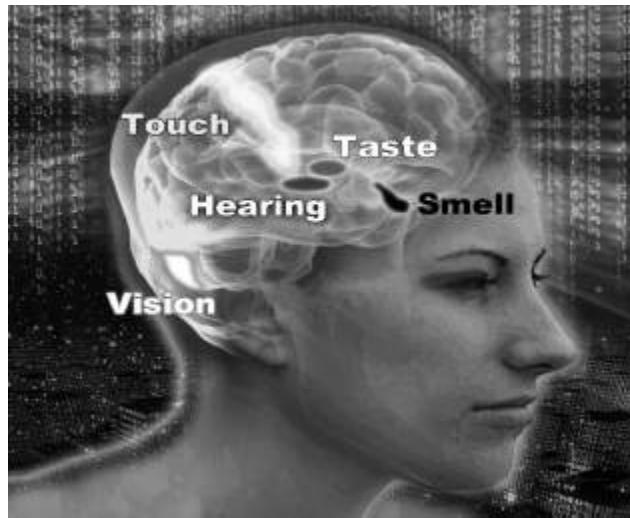


Figure 12: The brain receives information from the five senses and constructs from that our perceived reality.

[Without the brain] the world is devoid of light, heat, weight, solidity, motion, space, time or any other imaginable feature. All these phenomena are interactions, or transactions, of vibrations with a certain arrangement of neurons.

That's exactly what they are and scientist Robert Lanza describes in his book, *Biocentrism*, how we decode electromagnetic waves and energy into visual and 'physical' experience. He uses the example of a flame emitting photons, electromagnetic energy, each pulsing electrically and magnetically:

... these ... invisible electromagnetic waves strike a human retina, and if (and only if) the waves happen to measure between 400 and 700 nano meters in length from crest to crest, then their energy is just right to deliver a stimulus to the 8 million cone-shaped cells in the retina.

Each in turn send an electrical pulse to a neighbour neuron, and on up the line this goes, at 250 mph, until it reaches the ... occipital lobe of the brain, in the back of the head. There, a cascading complex of neurons fire from the incoming stimuli, and we subjectively perceive this experience as a yellow brightness occurring in a place we have been conditioned to call the 'external world'.

You hear what you decode

If a tree falls or a building collapses they make no noise unless someone is there to decode the energetic waves generated by the disturbance into what we call sound. Does a falling tree make a noise? Only if you hear it – *decode* it. Everything in our reality is a frequency field of information operating within the overall ‘Wi-Fi’ field that I call The Field. A vibrational disturbance is generated in The Field by the fields of the falling tree or building. These disturbance waves are what we decode into the sound of them falling. If no one is there to do that then neither will make any noise. Reality is created by the observer – *decoder* – and the *perceptions* of the observer affect the decoding process. For this reason different people – different *perceptions* – will perceive the same reality or situation in a different way. What one may perceive as a nightmare another will see as an opportunity. The question of why the Cult is so focused on controlling human perception now answers itself. All experienced reality is the act of decoding and we don’t experience Wi-Fi until it is decoded on the computer screen. The sight and sound of an Internet video is encoded in the Wi-Fi all around us, but we don’t see or hear it until the computer decodes that information. Taste, smell and touch are all phenomena of the brain as a result of the same process. We don’t taste, smell or feel anything except in the brain and there are pain relief techniques that seek to block the signal from the site of discomfort to the brain because if the brain doesn’t decode that signal we don’t feel pain. Pain is in the brain and only appears to be at the point of impact thanks to the feedback loop between them. We don’t see anything until electrical information from the sight senses is decoded in an area at the back of the brain. If that area is damaged we can go blind when our eyes are perfectly okay. So why do we go blind if we damage an eye? We damage the information processing between the waveform visual information and the visual decoding area of the brain. If information doesn’t reach the brain in a form it can decode then we can’t see the visual reality that it represents. What’s more the brain is decoding only a fraction of the information it receives and the rest is absorbed by the

sub-conscious mind. This explanation is from the science magazine, *Wonderpedia*:

Every second, 11 million sensations crackle along these [brain] pathways ... The brain is confronted with an alarming array of images, sounds and smells which it rigorously filters down until it is left with a manageable list of around 40. Thus 40 sensations per second make up what we perceive as reality.

The ‘world’ is not what people are told to believe that is it and the inner circles of the Cult *know that*.

Illusory ‘physical’ reality

We can only see a smear of 0.005 percent of the Universe which is only one of a vast array of universes – ‘mansions’ – within infinite reality. Even then the brain decodes only 40 pieces of information (‘sensations’) from a potential *11 million* that we receive every second. Two points strike you from this immediately: The sheer breathtaking stupidity of believing we know anything so rigidly that there’s nothing more to know; and the potential for these processes to be manipulated by a malevolent force to control the reality of the population. One thing I can say for sure with no risk of contradiction is that when you can perceive an almost indescribable fraction of infinite reality there is always more to know as in tidal waves of it. Ancient Greek philosopher Socrates was so right when he said that wisdom is to know how little we know. How obviously true that is when you think that we are experiencing a physical world of solidity that is neither physical nor solid and a world of apartness when everything is connected. Cult-controlled ‘science’ dismisses the so-called ‘paranormal’ and all phenomena related to that when the ‘para’-normal is perfectly normal and explains the alleged ‘great mysteries’ which dumbfound scientific minds. There is a reason for this. A ‘scientific mind’ in terms of the mainstream is a material mind, a five-sense mind imprisoned in see it, touch it, hear it, smell it and taste it. Phenomena and happenings that can’t be explained that way leave the ‘scientific mind’ bewildered and the rule is that if they

can't account for why something is happening then it can't, by definition, be happening. I beg to differ. Telepathy is thought waves passing through The Field (think wave disturbance again) to be decoded by someone able to connect with that wavelength (information). For example: You can pick up the thought waves of a friend at any distance and at the very least that will bring them to mind. A few minutes later the friend calls you. 'My god', you say, 'that's incredible – I was just thinking of you.' Ah, but *they* were thinking of *you* before they made the call and that's what you decoded. Native peoples not entrapped in five-sense reality do this so well it became known as the 'bush telegraph'. Those known as psychics and mediums (genuine ones) are doing the same only across dimensions of reality. 'Mind over matter' comes from the fact that matter and mind are the *same*. The state of one influences the state of the other. Indeed one *and* the other are illusions. They are aspects of the same field. Paranormal phenomena are all explainable so why are they still considered 'mysteries' or not happening? Once you go down this road of understanding you begin to expand awareness beyond the five senses and that's the nightmare for the Cult.



Figure 13: Holograms are not solid, but the best ones appear to be.

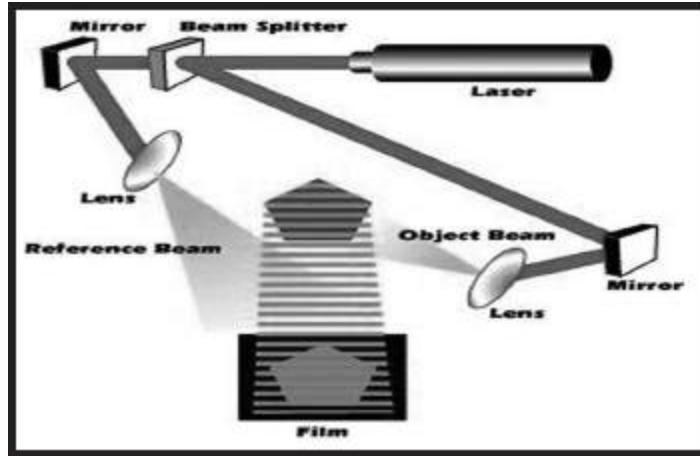


Figure 14: How holograms are created by capturing a waveform version of the subject image.

Holographic ‘solidity’

Our reality is not solid, it is holographic. We are now well aware of holograms which are widely used today. Two-dimensional information is decoded into a three-dimensional reality that is not solid although can very much appear to be (Fig 13). Holograms are created with a laser divided into two parts. One goes directly onto a photographic print ('reference beam') and the other takes a waveform image of the subject ('working beam') before being directed onto the print where it 'collides' with the other half of the laser (Fig 14). This creates a *waveform* interference pattern which contains the wavefield information of whatever is being photographed (Fig 15 overleaf). The process can be likened to dropping pebbles in a pond. Waves generated by each one spread out across the water to collide with the others and create a wave representation of where the stones fell and at what speed, weight and distance. A waveform interference pattern of a hologram is akin to the waveform information in The Field which the five senses decode into electrical signals to be decoded by the brain into a holographic illusory 'physical' reality. In the same way when a laser (think human attention) is directed at the waveform interference pattern a three-dimensional version of the subject is projected into apparently 'solid' reality (Fig 16). An amazing trait of holograms reveals more 'paranormal mysteries'. Information of the *whole*

hologram is encoded in waveform in every part of the interference pattern by the way they are created. This means that every *part* of a hologram is a smaller version of the whole. Cut the interference wave-pattern into four and you won't get four parts of the image. You get quarter-sized versions of the *whole* image. The body is a hologram and the same applies. Here we have the basis of acupuncture, reflexology and other forms of healing which identify representations of the whole body in all of the parts, hands, feet, ears, everywhere. Skilled palm readers can do what they do because the information of whole body is encoded in the hand. The concept of as above, so below, comes from this.



Figure 15: A waveform interference pattern that holds the information that transforms into a hologram.



Figure 16: Holographic people including 'Elvis' holographically inserted to sing a duet with Celine Dion.

The question will be asked of why, if solidity is illusory, we can't just walk through walls and each other. The resistance is not solid against solid; it is electromagnetic field against electromagnetic field and we decode this into the *experience* of solid against solid. We should also not underestimate the power of belief to dictate reality. What you believe is impossible *will be*. Your belief impacts on your decoding processes and they won't decode what you think is impossible. What we believe we perceive and what we perceive we experience. 'Can't dos' and 'impossibles' are like a firewall in a computer system that won't put on the screen what the firewall blocks. How vital that is to understanding how human experience has been hijacked. I explain in *The Answer, Everything You Need To Know But Have Never Been Told* and other books a long list of 'mysteries' and 'paranormal' phenomena that are not mysterious and perfectly normal once you realise what reality is and how it works. 'Ghosts' can be seen to pass through 'solid' walls because the walls are not solid and the ghost is a discarnate entity operating on a frequency so different to that of the wall that it's like two radio stations sharing the same space while never interfering with each other. I have seen ghosts do this myself. The apartness of people and objects is also an illusion. Everything is connected by the Field like all sea life is connected by the sea. It's just that within the limits of our visual reality we only 'see' holographic information and not the field of information that connects everything and from which the holographic world is made manifest. If you can only see holographic 'objects' and not the field that connects them they will appear to you as unconnected to each other in the same way that we see the computer while not seeing the Wi-Fi.

What you don't know *can* hurt you

Okay, we return to those 'two worlds' of human society and the Cult with its global network of interconnecting secret societies and satanic groups which manipulate through governments, corporations, media, religions, etc. The fundamental difference between them is *knowledge*. The idea has been to keep humanity

ignorant of the plan for its total enslavement underpinned by a crucial ignorance of reality – who we are and where we are – and how we interact with it. ‘Human’ should be the interaction between our expanded eternal consciousness and the five-sense body experience. We are meant to be *in* this world in terms of the five senses but not *of* this world in relation to our greater consciousness and perspective. In that state we experience the small picture of the five senses within the wider context of the big picture of awareness beyond the five senses. Put another way the five senses see the dots and expanded awareness connects them into pictures and patterns that give context to the apparently random and unconnected. Without the context of expanded awareness the five senses see only apartness and randomness with apparently no meaning. The Cult and its other-dimensional controllers seek to intervene in the frequency realm where five-sense reality is supposed to connect with expanded reality and to keep the two apart (more on this in the final chapter). When that happens five-sense mental and emotional processes are no longer influenced by expanded awareness, or the True ‘I’, and instead are driven by the isolated perceptions of the body’s decoding systems. They are in the world *and* of it. Here we have the human plight and why humanity with its potential for infinite awareness can be so easily manipulatable and descend into such extremes of stupidity.

Once the Cult isolates five-sense mind from expanded awareness it can then program the mind with perceptions and beliefs by controlling information that the mind receives through the ‘education’ system of the formative years and the media perceptual bombardment and censorship of an entire lifetime. Limit perception and a sense of the possible through limiting knowledge by limiting and skewing information while censoring and discrediting that which could set people free. As the title of another of my books says ... *And The Truth Shall Set You Free*. For this reason the last thing the Cult wants in circulation is the truth about anything – especially the reality of the eternal ‘I’ – and that’s why it is desperate to control information. The Cult knows that information becomes perception

which becomes behaviour which, collectively, becomes human society. Cult-controlled and funded mainstream ‘science’ denies the existence of an eternal ‘I’ and seeks to dismiss and trash all evidence to the contrary. Cult-controlled mainstream religion has a version of ‘God’ that is little more than a system of control and dictatorship that employs threats of damnation in an afterlife to control perceptions and behaviour in the here and now through fear and guilt. Neither is true and it’s the ‘neither’ that the Cult wishes to suppress. This ‘neither’ is that everything is an expression, a point of attention, within an infinite state of consciousness which is the real meaning of the term ‘God’.

Perceptual obsession with the ‘physical body’ and five-senses means that ‘God’ becomes personified as a bearded bloke sitting among the clouds or a raging bully who loves us if we do what ‘he’ wants and condemns us to the fires of hell if we don’t. These are no more than a ‘spiritual’ fairy tales to control and dictate events and behaviour through fear of this ‘God’ which has bizarrely made ‘God-fearing’ in religious circles a state to be desired. I would suggest that fearing *anything* is not to be encouraged and celebrated, but rather deleted. You can see why ‘God fearing’ is so beneficial to the Cult and its religions when *they* decide what ‘God’ wants and what ‘God’ demands (the Cult demands) that everyone do. As the great American comedian Bill Hicks said satirising a Christian zealot: ‘I think what God meant to say.’ How much of this infinite awareness (“God”) that we access is decided by how far we choose to expand our perceptions, self-identity and sense of the possible. The scale of self-identity reflects itself in the scale of awareness that we can connect with and are influenced by – how much knowing and insight we have instead of programmed perception. You cannot expand your awareness into the infinity of possibility when you believe that you are little me Peter the postman or Mary in marketing and nothing more. I’ll deal with this in the concluding chapter because it’s crucial to how we turnaround current events.

Where the Cult came from

When I realised in the early 1990s there was a Cult network behind global events I asked the obvious question: When did it start? I took it back to ancient Rome and Egypt and on to Babylon and Sumer in Mesopotamia, the 'Land Between Two Rivers', in what we now call Iraq. The two rivers are the Tigris and Euphrates and this region is of immense historical and other importance to the Cult, as is the land called Israel only 550 miles away by air. There is much more going with deep esoteric meaning across this whole region. It's not only about 'wars for oil'. Priceless artefacts from Mesopotamia were stolen or destroyed after the American and British invasion of Iraq in 2003 justified by the lies of Boy Bush and Tony Blair (their Cult masters) about non-existent 'weapons of mass destruction'.

Mesopotamia was the location of Sumer (about 5,400BC to 1,750BC), and Babylon (about 2,350BC to 539BC). Sabbatians may have become immensely influential in the Cult in modern times but they are part of a network that goes back into the mists of history. Sumer is said by historians to be the 'cradle of civilisation'. I disagree. I say it was the re-start of what we call human civilisation after cataclysmic events symbolised in part as the 'Great Flood' destroyed the world that existed before. These fantastic upheavals that I have been describing in detail in the books since the early 1990s appear in accounts and legends of ancient cultures across the world and they are supported by geological and biological evidence. Stone tablets found in Iraq detailing the Sumer period say the cataclysms were caused by non-human 'gods' they call the Anunnaki. These are described in terms of extraterrestrial visitations in which knowledge supplied by the Anunnaki is said to have been the source of at least one of the world's oldest writing systems and developments in astronomy, mathematics and architecture that were way ahead of their time. I have covered this subject at length in *The Biggest Secret* and *Children of the Matrix* and the same basic 'Anunnaki' story can be found in Zulu accounts in South Africa where the late and very great Zulu high shaman Credo Mutwa told me that the Sumerian Anunnaki were known by Zulus as the Chitauri or 'children of the serpent'. See my six-hour video interview with Credo on this subject entitled *The*

Reptilian Agenda recorded at his then home near Johannesburg in 1999 which you can watch on the Ickonic media platform.

The Cult emerged out of Sumer, Babylon and Egypt (and elsewhere) and established the Roman Empire before expanding with the Romans into northern Europe from where many empires were savagely imposed in the form of Cult-controlled societies all over the world. Mass death and destruction was their calling card. The Cult established its centre of operations in Europe and European Empires were Cult empires which allowed it to expand into a global force. Spanish and Portuguese colonialists headed for Central and South America while the British and French targeted North America. Africa was colonised by Britain, France, Belgium, the Netherlands, Portugal, Spain, Italy, and Germany. Some like Britain and France moved in on the Middle East. The British Empire was by far the biggest for a simple reason. By now Britain was the headquarters of the Cult from which it expanded to form Canada, the United States, Australia and New Zealand. The Sun never set on the British Empire such was the scale of its occupation. London remains a global centre for the Cult along with Rome and the Vatican although others have emerged in Israel and China. It is no accident that the 'virus' is alleged to have come out of China while Italy was chosen as the means to terrify the Western population into compliance with 'Covid' fascism. Nor that Israel has led the world in 'Covid' fascism and mass 'vaccination'.

You would think that I would mention the United States here, but while it has been an important means of imposing the Cult's will it is less significant than would appear and is currently in the process of having what power it does have deleted. The Cult in Europe has mostly loaded the guns for the US to fire. America has been controlled from Europe from the start through Cult operatives in Britain and Europe. The American Revolution was an illusion to make it appear that America was governing itself while very different forces were pulling the strings in the form of Cult families such as the Rothschilds through the Rockefellers and other subordinates. The Rockefellers are extremely close to Bill Gates and

established both scalpel and drug ‘medicine’ and the World Health Organization. They play a major role in the development and circulation of vaccines through the Rockefeller Foundation on which Bill Gates said his Foundation is based. Why wouldn’t this be the case when the Rockefellers and Gates are on the same team? Cult infiltration of human society goes way back into what we call history and has been constantly expanding and centralising power with the goal of establishing a global structure to dictate everything. Look how this has been advanced in great leaps with the ‘Covid’ hoax.

The non-human dimension

I researched and observed the comings and goings of Cult operatives through the centuries and even thousands of years as they were born, worked to promote the agenda within the secret society and satanic networks, and then died for others to replace them. Clearly there had to be a coordinating force that spanned this entire period while operatives who would not have seen the end goal in their lifetimes came and went advancing the plan over millennia. I went in search of that coordinating force with the usual support from the extraordinary synchronicity of my life which has been an almost daily experience since 1990. I saw common themes in religious texts and ancient cultures about a non-human force manipulating human society from the hidden. Christianity calls this force Satan, the Devil and demons; Islam refers to the Jinn or Djinn; Zulus have their Chitauri (spelt in other ways in different parts of Africa); and the Gnostic people in Egypt in the period around and before 400AD referred to this phenomena as the ‘Archons’, a word meaning rulers in Greek. Central American cultures speak of the ‘Predators’ among other names and the same theme is everywhere. I will use ‘Archons’ as a collective name for all of them. When you see how their nature and behaviour is described all these different sources are clearly talking about the same force. Gnostics described the Archons in terms of ‘luminous fire’ while Islam relates the Jinn to ‘smokeless fire’. Some refer to beings in form that could occasionally be seen, but the most common of common theme is that they operate from

unseen realms which means almost all existence to the visual processes of humans. I had concluded that this was indeed the foundation of human control and that the Cult was operating within the human frequency band on behalf of this hidden force when I came across the writings of Gnostics which supported my conclusions in the most extraordinary way.

A sealed earthen jar was found in 1945 near the town of Nag Hammadi about 75-80 miles north of Luxor on the banks of the River Nile in Egypt. Inside was a treasure trove of manuscripts and texts left by the Gnostic people some 1,600 years earlier. They included 13 leather-bound papyrus codices (manuscripts) and more than 50 texts written in Coptic Egyptian estimated to have been hidden in the jar in the period of 400AD although the source of the information goes back much further. Gnostics oversaw the Great or Royal Library of Alexandria, the fantastic depository of ancient texts detailing advanced knowledge and accounts of human history. The Library was dismantled and destroyed in stages over a long period with the death-blow delivered by the Cult-established Roman Church in the period around 415AD. The Church of Rome was the Church of Babylon relocated as I said earlier. Gnostics were not a race. They were a way of perceiving reality. Whenever they established themselves and their information circulated the terrorists of the Church of Rome would target them for destruction. This happened with the Great Library and with the Gnostic Cathars who were burned to death by the psychopaths after a long period of oppression at the siege of the Castle of Monségur in southern France in 1244. The Church has always been terrified of Gnostic information which demolishes the official Christian narrative although there is much in the Bible that supports the Gnostic view if you read it in another way. To anyone studying the texts of what became known as the Nag Hammadi Library it is clear that great swathes of Christian and Biblical belief has its origin with Gnostics sources going back to Sumer. Gnostic themes have been twisted to manipulate the perceived reality of Bible believers. Biblical texts have been in the open for centuries where they could be changed while Gnostic

documents found at Nag Hammadi were sealed away and untouched for 1,600 years. What you see is what they wrote.

Use your *pneuma* not your *nous*

Gnosticism and Gnostic come from 'gnosis' which means knowledge, or rather *secret* knowledge, in the sense of spiritual awareness – knowledge about reality and life itself. The desperation of the Cult's Church of Rome to destroy the Gnostics can be understood when the knowledge they were circulating was the last thing the Cult wanted the population to know. Sixteen hundred years later the same Cult is working hard to undermine and silence me for the same reason. The dynamic between knowledge and ignorance is a constant. 'Time' appears to move on, but essential themes remain the same. We are told to 'use your *nous*', a Gnostic word for head/brain/intelligence. They said, however, that spiritual awakening or 'salvation' could only be secured by expanding awareness *beyond* what they called *nous* and into *pneuma* or Infinite Self. Obviously as I read these texts the parallels with what I have been saying since 1990 were fascinating to me. There is a universal truth that spans human history and in that case why wouldn't we be talking the same language 16 centuries apart? When you free yourself from the perception program of the five senses and explore expanded realms of consciousness you are going to connect with the same information no matter what the perceived 'era' within a manufactured timeline of a single and tiny range of manipulated frequency. Humans working with 'smart' technology or knocking rocks together in caves is only a timeline appearing to operate within the human frequency band. Expanded awareness and the knowledge it holds have always been there whether the era be Stone Age or computer age. We can only access that knowledge by opening ourselves to its frequency which the five-sense prison cell is designed to stop us doing. Gates, Fauci, Whitty, Vallance, Zuckerberg, Brin, Page, Wojcicki, Bezos, and all the others behind the 'Covid' hoax clearly have a long wait before their range of frequency can make that connection given that an open heart is

crucial to that as we shall see. Instead of accessing knowledge directly through expanded awareness it is given to Cult operatives by the secret society networks of the Cult where it has been passed on over thousands of years outside the public arena. Expanded realms of consciousness is where great artists, composers and writers find their inspiration and where truth awaits anyone open enough to connect with it. We need to go there fast.

Archon hijack

A fifth of the Nag Hammadi texts describe the existence and manipulation of the Archons led by a 'Chief Archon' they call 'Yaldabaoth', or the 'Demiurge', and this is the Christian 'Devil', 'Satan', 'Lucifer', and his demons. Archons in Biblical symbolism are the 'fallen ones' which are also referred to as fallen angels after the angels expelled from heaven according to the Abrahamic religions of Judaism, Christianity and Islam. These angels are claimed to tempt humans to 'sin' ongoing and you will see how accurate that symbolism is during the rest of the book. The theme of 'original sin' is related to the 'Fall' when Adam and Eve were 'tempted by the serpent' and fell from a state of innocence and 'obedience' (connection) with God into a state of disobedience (disconnection). The Fall is said to have brought sin into the world and corrupted everything including human nature. Yaldabaoth, the 'Lord Archon', is described by Gnostics as a 'counterfeit spirit', 'The Blind One', 'The Blind God', and 'The Foolish One'. The Jewish name for Yaldabaoth in Talmudic writings is Samael which translates as 'Poison of God', or 'Blindness of God'. You see the parallels. Yaldabaoth in Islamic belief is the Muslim Jinn devil known as Shaytan – Shaytan is Satan as the same themes are found all over the world in every religion and culture. The 'Lord God' of the Old Testament is the 'Lord Archon' of Gnostic manuscripts and that's why he's such a bloodthirsty bastard. Satan is known by Christians as 'the Demon of Demons' and Gnostics called Yaldabaoth the 'Archon of Archons'. Both are known as 'The Deceiver'. We are talking about the same 'bloke' for sure and these common themes

using different names, storylines and symbolism tell a common tale of the human plight.

Archons are referred to in Nag Hammadi documents as mind parasites, inverters, guards, gatekeepers, detainers, judges, pitiless ones and deceivers. The 'Covid' hoax alone is a glaring example of all these things. The Biblical 'God' is so different in the Old and New Testaments because they are not describing the same phenomenon. The vindictive, angry, hate-filled, 'God' of the Old Testament, known as Yahweh, is Yaldabaoth who is depicted in Cult-dictated popular culture as the 'Dark Lord', 'Lord of Time', Lord (Darth) Vader and Dormammu, the evil ruler of the 'Dark Dimension' trying to take over the 'Earth Dimension' in the Marvel comic movie, *Dr Strange*. Yaldabaoth is both the Old Testament 'god' and the Biblical 'Satan'. Gnostics referred to Yaldabaoth as the 'Great Architect of the Universe' and the Cult-controlled Freemason network calls their god 'the Great Architect of the Universe' (also Grand Architect). The 'Great Architect' Yaldabaoth is symbolised by the Cult as the all-seeing eye at the top of the pyramid on the Great Seal of the United States and the dollar bill. Archon is encoded in *arch-itect* as it is in *arch-angels* and *arch-bishops*. All religions have the theme of a force for good and force for evil in some sort of spiritual war and there is a reason for that – the theme is true. The Cult and its non-human masters are quite happy for this to circulate. They present themselves as the force for good fighting evil when they are really the force of evil (absence of love). The whole foundation of Cult modus operandi is inversion. They promote themselves as a force for good and anyone challenging them in pursuit of peace, love, fairness, truth and justice is condemned as a satanic force for evil. This has been the game plan throughout history whether the Church of Rome inquisitions of non-believers or 'conspiracy theorists' and 'anti-vaxxers' of today. The technique is the same whatever the timeline era.

Yaldabaoth is revolting (true)

Yaldabaoth and the Archons are said to have revolted against God with Yaldabaoth claiming to *be* God – the *All That Is*. The Old Testament ‘God’ (Yaldabaoth) demanded to be worshipped as such: ‘*I am the LORD, and there is none else, there is no God beside me*’ (Isaiah 45:5). I have quoted in other books a man who said he was the unofficial son of the late Baron Philippe de Rothschild of the Mouton-Rothschild wine producing estates in France who died in 1988 and he told me about the Rothschild ‘revolt from God’. The man said he was given the name Phillip Eugene de Rothschild and we shared long correspondence many years ago while he was living under another identity. He said that he was conceived through ‘occult incest’ which (within the Cult) was ‘normal and to be admired’. ‘Phillip’ told me about his experience attending satanic rituals with rich and famous people whom he names and you can see them and the wider background to Cult Satanism in my other books starting with *The Biggest Secret*. Cult rituals are interactions with Archontic ‘gods’. ‘Phillip’ described Baron Philippe de Rothschild as ‘a master Satanist and hater of God’ and he used the same term ‘revolt from God’ associated with Yaldabaoth/Satan/Lucifer/the Devil in describing the Sabbatian Rothschild dynasty. ‘I played a key role in my family’s revolt from God’, he said. That role was to infiltrate in classic Sabbatian style the Christian Church, but eventually he escaped the mind-prison to live another life. The Cult has been targeting religion in a plan to make worship of the Archons the global one-world religion. Infiltration of Satanism into modern ‘culture’, especially among the young, through music videos, stage shows and other means, is all part of this.

Nag Hammadi texts describe Yaldabaoth and the Archons in their prime form as energy – consciousness – and say they can take form if they choose in the same way that consciousness takes form as a human. Yaldabaoth is called ‘formless’ and represents a deeply inverted, distorted and chaotic state of consciousness which seeks to attach to humans and turn them into a likeness of itself in an attempt at assimilation. For that to happen it has to manipulate

humans into low frequency mental and emotional states that match its own. Archons can certainly appear in human form and this is the origin of the psychopathic personality. The energetic distortion Gnostics called Yaldabaoth is psychopathy. When psychopathic Archons take human form that human will be a psychopath as an expression of Yaldabaoth consciousness. Cult psychopaths are Archons in human form. The principle is the same as that portrayed in the 2009 *Avatar* movie when the American military travelled to a fictional Earth-like moon called Pandora in the Alpha Centauri star system to infiltrate a society of blue people, or Na'vi, by hiding within bodies that looked like the Na'vi. Archons posing as humans have a particular hybrid information field, part human, part Archon, (the ancient 'demigods') which processes information in a way that manifests behaviour to match their psychopathic evil, lack of empathy and compassion, and stops them being influenced by the empathy, compassion and love that a fully-human information field is capable of expressing. Cult bloodlines interbreed, be they royalty or dark suits, for this reason and you have their obsession with incest. Interbreeding with full-blown humans would dilute the Archontic energy field that guarantees psychopathy in its representatives in the human realm.

Gnostic writings say the main non-human forms that Archons take are *serpentine* (what I have called for decades 'reptilian' amid unbounded ridicule from the Archontically-programmed) and what Gnostics describe as 'an unborn baby or foetus with grey skin and dark, unmoving eyes'. This is an excellent representation of the ET 'Greys' of UFO folklore which large numbers of people claim to have seen and been abducted by – Zulu shaman Credo Mutwa among them. I agree with those that believe in extraterrestrial or interdimensional visitations today and for thousands of years past. No wonder with their advanced knowledge and technological capability they were perceived and worshipped as gods for technological and other 'miracles' they appeared to perform. Imagine someone arriving in a culture disconnected from the modern world with a smartphone and computer. They would be

seen as a ‘god’ capable of ‘miracles’. The Renegade Mind, however, wants to know the source of everything and not only the way that source manifests as human or non-human. In the same way that a Renegade Mind seeks the original source material for the ‘Covid virus’ to see if what is claimed is true. The original source of Archons in form is consciousness – the distorted state of consciousness known to Gnostics as Yaldabaoth.

‘Revolt from God’ is energetic disconnection

Where I am going next will make a lot of sense of religious texts and ancient legends relating to ‘Satan’, Lucifer’ and the ‘gods’. Gnostic descriptions sync perfectly with the themes of my own research over the years in how they describe a consciousness distortion seeking to impose itself on human consciousness. I’ve referred to the core of infinite awareness in previous books as Infinite Awareness in Awareness of Itself. By that I mean a level of awareness that knows that it is all awareness and is aware of all awareness. From here comes the frequency of love in its true sense and balance which is what love is on one level – the balance of all forces into a single whole called Oneness and Isness. The more we disconnect from this state of love that many call ‘God’ the constituent parts of that Oneness start to unravel and express themselves as a part and not a whole. They become individualised as intellect, mind, selfishness, hatred, envy, desire for power over others, and such like. This is not a problem in the greater scheme in that ‘God’, the *All That Is*, can experience all these possibilities through different expressions of itself including humans. What we as expressions of the whole experience the *All That Is* experiences. We are the *All That Is* experiencing itself. As we withdraw from that state of Oneness we disconnect from its influence and things can get very unpleasant and very stupid. Archontic consciousness is at the extreme end of that. It has so disconnected from the influence of Oneness that it has become an inversion of unity and love, an inversion of everything, an inversion of life itself. Evil is appropriately live written backwards. Archontic consciousness is obsessed with death, an inversion of life,

and so its manifestations in Satanism are obsessed with death. They use inverted symbols in their rituals such as the inverted pentagram and cross. Sabbatians as Archontic consciousness incarnate invert Judaism and every other religion and culture they infiltrate. They seek disunity and chaos and they fear unity and harmony as they fear love like garlic to a vampire. As a result the Cult, Archons incarnate, act with such evil, psychopathy and lack of empathy and compassion disconnected as they are from the source of love. How could Bill Gates and the rest of the Archontic psychopaths do what they have to human society in the 'Covid' era with all the death, suffering and destruction involved and have no emotional consequence for the impact on others? Now you know. Why have Zuckerberg, Brin, Page, Wojcicki and company callously censored information warning about the dangers of the 'vaccine' while thousands have been dying and having severe, sometimes life-changing reactions? Now you know. Why have Tedros, Fauci, Whitty, Vallance and their like around the world been using case and death figures they're aware are fraudulent to justify lockdowns and all the deaths and destroyed lives that have come from that? Now you know. Why did Christian Drosten produce and promote a 'testing' protocol that he knew couldn't test for infectious disease which led to a global human catastrophe. Now you know. The Archontic mind doesn't give a shit ([Fig 17](#)). I personally think that Gates and major Cult insiders are a form of AI cyborg that the Archons want humans to become.

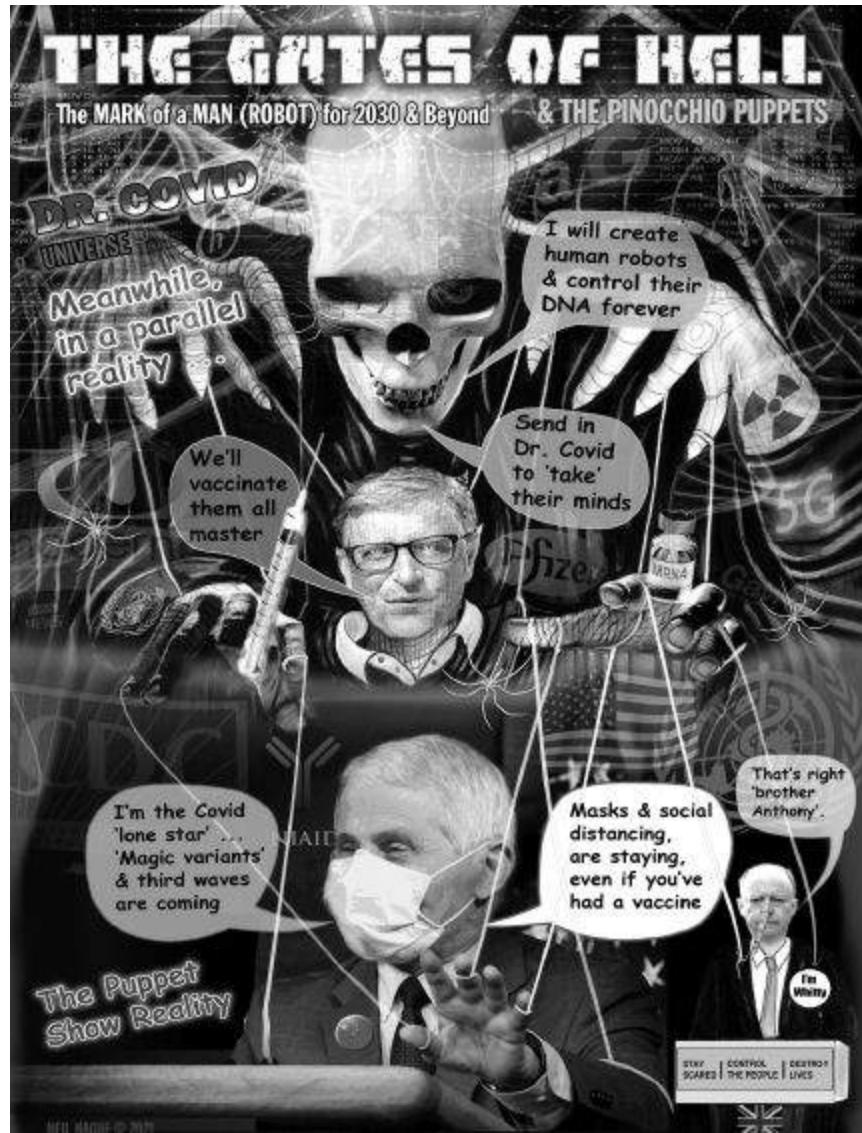


Figure 17: Artist Neil Hague's version of the 'Covid' hierarchy.

Human batteries

A state of such inversion does have its consequences, however. The level of disconnection from the Source of All means that you withdraw from that source of energetic sustenance and creativity. This means that you have to find your own supply of energetic power and it has – *us*. When the Morpheus character in the first *Matrix* movie held up a battery he spoke a profound truth when he said: 'The Matrix is a computer-generated dream world built to keep us under control in order to change the human being into one of

these.' The statement was true in all respects. We do live in a technologically-generated virtual reality simulation (more very shortly) and we have been manipulated to be an energy source for Archontic consciousness. The Disney-Pixar animated movie *Monsters, Inc.* in 2001 symbolised the dynamic when monsters in their world had no energy source and they would enter the human world to terrify children in their beds, catch the child's scream, terror (low-vibrational frequencies), and take that energy back to power the monster world. The lead character you might remember was a single giant eye and the symbolism of the Cult's all-seeing eye was obvious. Every thought and emotion is broadcast as a frequency unique to that thought and emotion. Feelings of love and joy, empathy and compassion, are high, quick, frequencies while fear, depression, anxiety, suffering and hate are low, slow, dense frequencies. Which kind do you think Archontic consciousness can connect with and absorb? In such a low and dense frequency state there's no way it can connect with the energy of love and joy. Archons can only feed off energy compatible with their own frequency and they and their Cult agents want to delete the human world of love and joy and manipulate the transmission of low vibrational frequencies through low-vibrational human mental and emotional states. *We are their energy source.* Wars are energetic banquets to the Archons – a world war even more so – and think how much low-frequency mental and emotional energy has been generated from the consequences for humanity of the 'Covid' hoax orchestrated by Archons incarnate like Gates.

The ancient practice of human sacrifice 'to the gods', continued in secret today by the Cult, is based on the same principle. 'The gods' are Archontic consciousness in different forms and the sacrifice is induced into a state of intense terror to generate the energy the Archontic frequency can absorb. Incarnate Archons in the ritual drink the blood which contains an adrenaline they crave which floods into the bloodstream when people are terrorised. Most of the sacrifices, ancient and modern, are children and the theme of 'sacrificing young virgins to the gods' is just code for children. They

have a particular pre-puberty energy that Archons want more than anything and the energy of the young in general is their target. The California Department of Education wants students to chant the names of Aztec gods (Archontic gods) once worshipped in human sacrifice rituals in a curriculum designed to encourage them to ‘challenge racist, bigoted, discriminatory, imperialist/colonial beliefs’, join ‘social movements that struggle for social justice’, and ‘build new possibilities for a post-racist, post-systemic racism society’. It’s the usual Woke crap that inverts racism and calls it anti-racism. In this case solidarity with ‘indigenous tribes’ is being used as an excuse to chant the names of ‘gods’ to which people were sacrificed (and still are in secret). What an example of Woke’s inability to see beyond black and white, us and them, They condemn the colonisation of these tribal cultures by Europeans (quite right), but those cultures sacrificing people including children to their ‘gods’, and mass murdering untold numbers as the Aztecs did, is just fine. One chant is to the Aztec god Tezcatlipoca who had a man sacrificed to him in the 5th month of the Aztec calendar. His heart was cut out and he was eaten. Oh, that’s okay then. Come on children … after three … Other sacrificial ‘gods’ for the young to chant their allegiance include Quetzalcoatl, Huitzilopochtli and Xipe Totec. The curriculum says that ‘chants, affirmations, and energizers can be used to bring the class together, build unity around ethnic studies principles and values, and to reinvigorate the class following a lesson that may be emotionally taxing or even when student engagement may appear to be low’. Well, that’s the cover story, anyway. Chanting and mantras are the repetition of a particular frequency generated from the vocal cords and chanting the names of these Archontic ‘gods’ tunes you into their frequency. That is the last thing you want when it allows for energetic synchronisation, attachment and perceptual influence. Initiates chant the names of their ‘Gods’ in their rituals for this very reason.

Vampires of the Woke

Paedophilia is another way that Archons absorb the energy of children. Paedophiles possessed by Archontic consciousness are used as the conduit during sexual abuse for discarnate Archons to vampire the energy of the young they desire so much. Stupendous numbers of children disappear every year never to be seen again although you would never know from the media. Imagine how much low-vibrational energy has been generated by children during the 'Covid' hoax when so many have become depressed and psychologically destroyed to the point of killing themselves.

Shocking numbers of children are now taken by the state from loving parents to be handed to others. I can tell you from long experience of researching this since 1996 that many end up with paedophiles and assets of the Cult through corrupt and Cult-owned social services which in the reframing era has hired many psychopaths and emotionless automatons to do the job. Children are even stolen to order using spurious reasons to take them by the corrupt and secret (because they're corrupt) 'family courts'. I have written in detail in other books, starting with *The Biggest Secret* in 1997, about the ubiquitous connections between the political, corporate, government, intelligence and military elites (Cult operatives) and Satanism and paedophilia. If you go deep enough both networks have an interlocking leadership. The Woke mentality has been developed by the Cult for many reasons: To promote almost every aspect of its agenda; to hijack the traditional political left and turn it fascist; to divide and rule; and to target agenda pushbackers. But there are other reasons which relate to what I am describing here. How many happy and joyful Wokers do you ever see especially at the extreme end? They are a mental and psychological mess consumed by emotional stress and constantly emotionally cocked for the next explosion of indignation at someone referring to a female as a female. They are walking, talking, batteries as Morpheus might say emitting frequencies which both enslave them in low-vibrational bubbles of perceptual limitation and feed the Archons. Add to this the hatred claimed to be love; fascism claimed to 'anti-fascism', racism claimed to be 'anti-racism';

exclusion claimed to inclusion; and the abuse-filled Internet trolling. You have a purpose-built Archontic energy system with not a wind turbine in sight and all founded on Archontic *inversion*. We have whole generations now manipulated to serve the Archons with their actions and energy. They will be doing so their entire adult lives unless they snap out of their Archon-induced trance. Is it really a surprise that Cult billionaires and corporations put so much money their way? Where is the energy of joy and laughter, including laughing at yourself which is confirmation of your own emotional security? Mark Twain said: 'The human race has one really effective weapon, and that is laughter.' We must use it all the time. Woke has destroyed comedy because it has no humour, no joy, sense of irony, or self-deprecation. Its energy is dense and intense. *Mmmmm*, lunch says the Archontic frequency. Rudolf Steiner (1861-1925) was the Austrian philosopher and famous esoteric thinker who established Waldorf education or Steiner schools to treat children like unique expressions of consciousness and not minds to be programmed with the perceptions determined by authority. I'd been writing about this energy vampiring for decades when I was sent in 2016 a quote by Steiner. He was spot on:

There are beings in the spiritual realms for whom anxiety and fear emanating from human beings offer welcome food. When humans have no anxiety and fear, then these creatures starve. If fear and anxiety radiates from people and they break out in panic, then these creatures find welcome nutrition and they become more and more powerful. These beings are hostile towards humanity. Everything that feeds on negative feelings, on anxiety, fear and superstition, despair or doubt, are in reality hostile forces in super-sensible worlds, launching cruel attacks on human beings, while they are being fed ... These are exactly the feelings that belong to contemporary culture and materialism; because it estranges people from the spiritual world, it is especially suited to evoke hopelessness and fear of the unknown in people, thereby calling up the above mentioned hostile forces against them.

Pause for a moment from this perspective and reflect on what has happened in the world since the start of 2020. Not only will pennies drop, but billion dollar bills. We see the same theme from Don Juan Matus, a Yaqui Indian shaman in Mexico and the information source for Peruvian-born writer, Carlos Castaneda, who wrote a series of

books from the 1960s to 1990s. Don Juan described the force manipulating human society and his name for the Archons was the predator:

We have a predator that came from the depths of the cosmos and took over the rule of our lives. Human beings are its prisoners. The predator is our lord and master. It has rendered us docile, helpless. If we want to protest, it suppresses our protest. If we want to act independently, it demands that we don't do so ... indeed we are held prisoner!

They took us over because we are food to them, and they squeeze us mercilessly because we are their sustenance. Just as we rear chickens in coops, the predators rear us in human coops, humaneros. Therefore, their food is always available to them.

Different cultures, different eras, same recurring theme.

The 'ennoia' dilemma

Nag Hammadi Gnostic manuscripts say that Archon consciousness has no 'ennoia'. This is directly translated as 'intentionality', but I'll use the term 'creative imagination'. The *All That Is* in awareness of itself is the source of all creativity – all possibility – and the more disconnected you are from that source the more you are subsequently denied 'creative imagination'. Given that Archon consciousness is almost entirely disconnected it severely lacks creativity and has to rely on far more mechanical processes of thought and exploit the creative potential of those that do have 'ennoia'. You can see cases of this throughout human society. Archon consciousness almost entirely dominates the global banking system and if we study how that system works you will appreciate what I mean. Banks manifest 'money' out of nothing by issuing lines of 'credit' which is 'money' that has never, does not, and will never exist except in theory. It's a confidence trick. If you think 'credit' figures-on-a-screen 'money' is worth anything you accept it as payment. If you don't then the whole system collapses through lack of confidence in the value of that 'money'. Archontic bankers with no 'ennoia' are 'lending' 'money' that doesn't exist to humans that *do* have creativity – those that have the inspired ideas and create businesses and products. Archon banking feeds off human creativity

which it controls through ‘money’ creation and debt. Humans have the creativity and Archons exploit that for their own benefit and control while having none themselves. Archon Internet platforms like Facebook claim joint copyright of everything that creative users post and while Archontic minds like Zuckerberg may officially head that company it will be human creatives on the staff that provide the creative inspiration. When you have limitless ‘money’ you can then buy other companies established by creative humans. Witness the acquisition record of Facebook, Google and their like. Survey the Archon-controlled music industry and you see non-creative dark suit executives making their fortune from the human creativity of their artists. The cases are endless. Research the history of people like Gates and Zuckerberg and how their empires were built on exploiting the creativity of others. Archon minds cannot create out of nothing, but they are skilled (because they have to be) in what Gnostic texts call ‘countermimicry’. They can imitate, but not innovate. Sabbatians trawl the creativity of others through backdoors they install in computer systems through their cybersecurity systems. Archon-controlled China is globally infamous for stealing intellectual property and I remember how Hong Kong, now part of China, became notorious for making counterfeit copies of the creativity of others – ‘countermimicry’. With the now pervasive and all-seeing surveillance systems able to infiltrate any computer you can appreciate the potential for Archons to vampire the creativity of humans. Author John Lamb Lash wrote in his book about the Nag Hammadi texts, *Not In His Image*:

Although they cannot originate anything, because they lack the divine factor of ennoia (intentionality), Archons can imitate with a vengeance. Their expertise is simulation (HAL, virtual reality). The Demiurge [Yaldabaoth] fashions a heaven world copied from the fractal patterns [of the original] ... His construction is celestial kitsch, like the fake Italianate villa of a Mafia don complete with militant angels to guard every portal.

This brings us to something that I have been speaking about since the turn of the millennium. Our reality is a simulation; a virtual reality that we think is real. No, I’m not kidding.

Human reality? Well, virtually

I had pondered for years about whether our reality is ‘real’ or some kind of construct. I remembered being immensely affected on a visit as a small child in the late 1950s to the then newly-opened Planetarium on the Marylebone Road in London which is now closed and part of the adjacent Madame Tussauds wax museum. It was in the middle of the day, but when the lights went out there was the night sky projected in the Planetarium’s domed ceiling and it appeared to be so real. The experience never left me and I didn’t know why until around the turn of the millennium when I became certain that our ‘night sky’ and entire reality is a projection, a virtual reality, akin to the illusory world portrayed in the *Matrix* movies. I looked at the sky one day in this period and it appeared to me like the domed roof of the Planetarium. The release of the first *Matrix* movie in 1999 also provided a synchronistic and perfect visual representation of where my mind had been going for a long time. I hadn’t come across the Gnostic Nag Hammadi texts then. When I did years later the correlation was once again astounding. As I read Gnostic accounts from 1,600 years and more earlier it was clear that they were describing the same simulation phenomenon. They tell how the Yaldabaoth ‘Demiurge’ and Archons created a ‘bad copy’ of original reality to rule over all that were captured by its illusions and the body was a prison to trap consciousness in the ‘bad copy’ fake reality. Read how Gnostics describe the ‘bad copy’ and update that to current times and they are referring to what we would call today a virtual reality simulation.

Author John Lamb Lash said ‘the Demiurge fashions a heaven world copied from the fractal patterns’ of the original through expertise in ‘HAL’ or virtual reality simulation. Fractal patterns are part of the energetic information construct of our reality, a sort of blueprint. If these patterns were copied in computer terms it would indeed give you a copy of a ‘natural’ reality in a non-natural frequency and digital form. The principle is the same as making a copy of a website. The original website still exists, but now you can change the copy version to make it whatever you like and it can

become very different to the original website. Archons have done this with our reality, a *synthetic* copy of prime reality that still exists beyond the frequency walls of the simulation. Trapped within the illusions of this synthetic Matrix, however, were and are human consciousness and other expressions of prime reality and this is why the Archons via the Cult are seeking to make the human body synthetic and give us synthetic AI minds to complete the job of turning the entire reality synthetic including what we perceive to be the natural world. To quote Kurzweil: 'Nanobots will infuse all the matter around us with information. Rocks, trees, everything will become these intelligent creatures.' Yes, *synthetic* 'creatures' just as 'Covid' and other genetically-manipulating 'vaccines' are designed to make the human body synthetic. From this perspective it is obvious why Archons and their Cult are so desperate to infuse synthetic material into every human with their 'Covid' scam.

Let there be (electromagnetic) light

Yaldabaoth, the force that created the simulation, or Matrix, makes sense of the Gnostic reference to 'The Great Architect' and its use by Cult Freemasonry as the name of its deity. The designer of the Matrix in the movies is called 'The Architect' and that trilogy is jam-packed with symbolism relating to these subjects. I have contended for years that the angry Old Testament God (Yaldabaoth) is the 'God' being symbolically 'quoted' in the opening of Genesis as 'creating the world'. This is not the creation of prime reality – it's the creation of the *simulation*. The Genesis 'God' says: 'Let there be Light: and there was light.' But what is this 'Light'? I have said for decades that the speed of light (186,000 miles per second) is not the fastest speed possible as claimed by mainstream science and is in fact the frequency walls or outer limits of the Matrix. You can't have a fastest or slowest anything within all possibility when everything is possible. The human body is encoded to operate within the speed of light or *within the simulation* and thus we see only the tiny frequency band of visible *light*. Near-death experiencers who perceive reality outside the body during temporary 'death' describe a very different

form of light and this is supported by the Nag Hammadi texts. Prime reality beyond the simulation ('Upper Aeons' to the Gnostics) is described as a realm of incredible beauty, bliss, love and harmony – a realm of 'watery light' that is so powerful 'there are no shadows'. Our false reality of Archon control, which Gnostics call the 'Lower Aeons', is depicted as a realm with a different kind of 'light' and described in terms of chaos, 'Hell', 'the Abyss' and 'Outer Darkness', where trapped souls are tormented and manipulated by demons (relate that to the 'Covid' hoax alone). The watery light theme can be found in near-death accounts and it is not the same as *simulation* 'light' which is electromagnetic or radiation light within the speed of light – the 'Lower Aeons'. Simulation 'light' is the 'luminous fire' associated by Gnostics with the Archons. The Bible refers to Yaldabaoth as 'that old serpent, called the Devil, and Satan, which deceiveth the whole world' (Revelation 12:9). I think that making a simulated copy of prime reality ('countermimicry') and changing it dramatically while all the time manipulating humanity to believe it to be real could probably meet the criteria of deceiving the whole world. Then we come to the Cult god Lucifer – the *Light Bringer*. Lucifer is symbolic of Yaldabaoth, the bringer of radiation light that forms the bad copy simulation within the speed of light. 'He' is symbolised by the lighted torch held by the Statue of Liberty and in the name 'Illuminati'. Sabbatian-Frankism declares that Lucifer is the true god and Lucifer is the real god of Freemasonry honoured as their 'Great or Grand Architect of the Universe' (simulation).

I would emphasise, too, the way Archontic technologically-generated luminous fire of radiation has deluged our environment since I was a kid in the 1950s and changed the nature of The Field with which we constantly interact. Through that interaction technological radiation is changing us. The Smart Grid is designed to operate with immense levels of communication power with 5G expanding across the world and 6G, 7G, in the process of development. Radiation is the simulation and the Archontic manipulation system. Why wouldn't the Archon Cult wish to unleash radiation upon us to an ever-greater extreme to form

Kurzweil's 'cloud'? The plan for a synthetic human is related to the need to cope with levels of radiation beyond even anything we've seen so far. Biological humans would not survive the scale of radiation they have in their script. The Smart Grid is a technological sub-reality within the technological simulation to further disconnect five-sense perception from expanded consciousness. It's a technological prison of the mind.

Infusing the 'spirit of darkness'

A recurring theme in religion and native cultures is the manipulation of human genetics by a non-human force and most famously recorded as the biblical 'sons of god' (the gods plural in the original) who interbred with the daughters of men. The Nag Hammadi *Apocryphon of John* tells the same story this way:

He [Yaldabaoth] sent his angels [Archons/demons] to the daughters of men, that they might take some of them for themselves and raise offspring for their enjoyment. And at first they did not succeed. When they had no success, they gathered together again and they made a plan together ... And the angels changed themselves in their likeness into the likeness of their mates, filling them with the spirit of darkness, which they had mixed for them, and with evil ... And they took women and begot children out of the darkness according to the likeness of their spirit.

Possession when a discarnate entity takes over a human body is an age-old theme and continues today. It's very real and I've seen it. Satanic and secret society rituals can create an energetic environment in which entities can attach to initiates and I've heard many stories of how people have changed their personality after being initiated even into lower levels of the Freemasons. I have been inside three Masonic temples, one at a public open day and two by just walking in when there was no one around to stop me. They were in Ryde, the town where I live, Birmingham, England, when I was with a group, and Boston, Massachusetts. They all felt the same energetically – dark, dense, low-vibrational and sinister. Demonic attachment can happen while the initiate has no idea what is going on. To them it's just a ritual to get in the Masons and do a bit of good

business. In the far more extreme rituals of Satanism human possession is even more powerful and they are designed to make possession possible. The hierarchy of the Cult is dictated by the power and perceived status of the possessing Archon. In this way the Archon hierarchy becomes the Cult hierarchy. Once the entity has attached it can influence perception and behaviour and if it attaches to the extreme then so much of its energy (information) infuses into the body information field that the hologram starts to reflect the nature of the possessing entity. This is the *Exorcist* movie type of possession when facial features change and it's known as shapeshifting. Islam's Jinn are said to be invisible tricksters who change shape, 'whisper', confuse and take human form. These are all traits of the Archons and other versions of the same phenomenon. Extreme possession could certainty infuse the 'spirit of darkness' into a partner during sex as the Nag Hammadi texts appear to describe. Such an infusion can change genetics which is also energetic information. Human genetics is information and the 'spirit of darkness' is information. Mix one with the other and change must happen. Islam has the concept of a 'Jinn baby' through possession of the mother and by Jinn taking human form. There are many ways that human genetics can be changed and remember that Archons have been aware all along of advanced techniques to do this. What is being done in human society today – and far more – was known about by Archons at the time of the 'fallen ones' and their other versions described in religions and cultures.

Archons and their human-world Cult are obsessed with genetics as we see today and they know this dictates how information is processed into perceived reality during a human life. They needed to produce a human form that would decode the simulation and this is symbolically known as 'Adam and Eve' who left the 'garden' (prime reality) and 'fell' into Matrix reality. The simulation is not a 'physical' construct (there is no 'physical'); it is a source of information. Think Wi-Fi again. The simulation is an energetic field encoded with information and body-brain systems are designed to decode that information encoded in wave or frequency form which

is transmitted to the brain as electrical signals. These are decoded by the brain to construct our sense of reality – an illusory ‘physical’ world that only exists in the brain or the mind. Virtual reality games mimic this process using the same sensory decoding system. Information is fed to the senses to decode a virtual reality that can appear so real, but isn’t (Figs 18 and 19). Some scientists believe – and I agree with them – that what we perceive as ‘physical’ reality only exists when we are looking or observing. The act of perception or focus triggers the decoding systems which turn waveform information into holographic reality. When we are not observing something our reality reverts from a holographic state to a waveform state. This relates to the same principle as a falling tree not making a noise unless someone is there to hear it or decode it. The concept makes sense from the simulation perspective. A computer is not decoding all the information in a Wi-Fi field all the time and only decodes or brings into reality on the screen that part of Wi-Fi that it’s decoding – focusing upon – at that moment.



Figure 18: Virtual reality technology ‘hacks’ into the body’s five-sense decoding system.



Figure 19: The result can be experienced as very ‘real’.

Interestingly, Professor Donald Hoffman at the Department of Cognitive Sciences at the University of California, Irvine, says that our experienced reality is like a computer interface that shows us only the level with which we interact while hiding all that exists beyond it: ‘Evolution shaped us with a user interface that hides the truth. Nothing that we see is the truth – the very language of space and time and objects is the wrong language to describe reality.’ He is correct in what he says on so many levels. Space and time are not a universal reality. They are a phenomenon of decoded *simulation* reality as part of the process of enslaving our sense of reality. Near-death experiencers report again and again how space and time did not exist as we perceive them once they were free of the body – body decoding systems. You can appreciate from this why Archons and their Cult are so desperate to entrap human attention in the five senses where we are in the Matrix and of the Matrix. Opening your mind to expanded states of awareness takes you beyond the information confines of the simulation and you become aware of knowledge and insights denied to you before. This is what we call ‘awakening’ – *awakening from the Matrix* – and in the final chapter I will relate this to current events.

Where are the ‘aliens’?

A simulation would explain the so-called ‘Fermi Paradox’ named after Italian physicist Enrico Fermi (1901-1954) who created the first nuclear reactor. He considered the question of why there is such a lack of extraterrestrial activity when there are so many stars and planets in an apparently vast universe; but what if the night sky that we see, or think we do, is a simulated projection as I say? If you control the simulation and your aim is to hold humanity fast in essential ignorance would you want other forms of life including advanced life coming and going sharing information with humanity? Or would you want them to believe they were isolated and apparently alone? Themes of human isolation and apartness are common whether they be the perception of a lifeless universe or the fascist isolation laws of the ‘Covid’ era. Paradoxically the very

existence of a simulation means that we are not alone when some force had to construct it. My view is that experiences that people have reported all over the world for centuries with Reptilians and Grey entities are Archon phenomena as Nag Hammadi texts describe; and that benevolent 'alien' interactions are non-human groups that come in and out of the simulation by overcoming Archon attempts to keep them out. It should be highlighted, too, that Reptilians and Greys are obsessed with *genetics* and *technology* as related by cultural accounts and those who say they have been abducted by them. Technology is their way of overcoming some of the limitations in their creative potential and our technology-driven and controlled human society of today is *archetypical* Archon-Reptilian-Grey modus operandi. Technocracy is really *Archontocracy*. The Universe does not have to be as big as it appears with a simulation. There is no space or distance only information decoded into holographic reality. What we call 'space' is only the absence of holographic 'objects' and that 'space' is The Field of energetic information which connects everything into a single whole. The same applies with the artificially-generated information field of the simulation. The Universe is not big or small as a physical reality. It is decoded information, that's all, and its perceived size is decided by the way the simulation is encoded to make it appear. The entire night sky as we perceive it only exists in our brain and so where are those 'millions of light years'? The 'stars' on the ceiling of the Planetarium looked a vast distance away.

There's another point to mention about 'aliens'. I have been highlighting since the 1990s the plan to stage a fake 'alien invasion' to justify the centralisation of global power and a world military. Nazi scientist Werner von Braun, who was taken to America by Operation Paperclip after World War Two to help found NASA, told his American assistant Dr Carol Rosin about the Cult agenda when he knew he was dying in 1977. Rosin said that he told her about a sequence that would lead to total human control by a one-world government. This included threats from terrorism, rogue nations, meteors and asteroids before finally an 'alien invasion'. All of these

things, von Braun said, would be bogus and what I would refer to as a No-Problem-Reaction-Solution. Keep this in mind when ‘the aliens are coming’ is the new mantra. The aliens are not coming – they are *already here* and they have infiltrated human society while looking human. French-Canadian investigative journalist Serge Monast said in 1994 that he had uncovered a NASA/military operation called Project Blue Beam which fits with what Werner von Braun predicted. Monast died of a ‘heart attack’ in 1996 the day after he was arrested and spent a night in prison. He was 51. He said Blue Beam was a plan to stage an alien invasion that would include religious figures beamed holographically into the sky as part of a global manipulation to usher in a ‘new age’ of worshipping what I would say is the Cult ‘god’ Yaldabaoth in a one-world religion. Fake holographic asteroids are also said to be part of the plan which again syncs with von Braun. How could you stage an illusory threat from asteroids unless they were holographic inserts? This is pretty straightforward given the advanced technology outside the public arena and the fact that our ‘physical’ reality is holographic anyway. Information fields would be projected and we would decode them into the illusion of a ‘physical’ asteroid. If they can sell a global ‘pandemic’ with a ‘virus’ that doesn’t exist what will humans not believe if government and media tell them?

All this is particularly relevant as I write with the Pentagon planning to release in June, 2021, information about ‘UFO sightings’. I have been following the UFO story since the early 1990s and the common theme throughout has been government and military denials and cover up. More recently, however, the Pentagon has suddenly become more talkative and apparently open with Air Force pilot radar images released of unexplained craft moving and changing direction at speeds well beyond anything believed possible with human technology. Then, in March, 2021, former Director of National Intelligence John Ratcliffe said a Pentagon report months later in June would reveal a great deal of information about UFO sightings unknown to the public. He said the report would have ‘massive implications’. The order to do this was included bizarrely

in a \$2.3 trillion ‘coronavirus’ relief and government funding bill passed by the Trump administration at the end of 2020. I would add some serious notes of caution here. I have been pointing out since the 1990s that the US military and intelligence networks have long had craft – ‘flying saucers’ or anti-gravity craft – which any observer would take to be extraterrestrial in origin. Keeping this knowledge from the public allows craft flown by *humans* to be perceived as alien visitations. I am not saying that ‘aliens’ do not exist. I would be the last one to say that, but we have to be streetwise here. President Ronald Reagan told the UN General Assembly in 1987: ‘I occasionally think how quickly our differences worldwide would vanish if we were facing an alien threat from outside this world.’ That’s the idea. Unite against a common ‘enemy’ with a common purpose behind your ‘saviour force’ (the Cult) as this age-old technique of mass manipulation goes global.

Science moves this way ...

I could find only one other person who was discussing the simulation hypothesis publicly when I concluded it was real. This was Nick Bostrom, a Swedish-born philosopher at the University of Oxford, who has explored for many years the possibility that human reality is a computer simulation although his version and mine are not the same. Today the simulation and holographic reality hypothesis have increasingly entered the scientific mainstream. Well, the more open-minded mainstream, that is. Here are a few of the ever-gathering examples. American nuclear physicist Silas Beane led a team of physicists at the University of Bonn in Germany pursuing the question of whether we live in a simulation. They concluded that we probably do and it was likely based on a lattice of cubes. They found that cosmic rays align with that specific pattern. The team highlighted the Greisen-Zatsepin-Kuzmin (GZK) limit which refers to cosmic ray particle interaction with cosmic background radiation that creates an apparent boundary for cosmic ray particles. They say in a paper entitled ‘Constraints on the Universe as a Numerical Simulation’ that this ‘pattern of constraint’ is exactly what you

would find with a computer simulation. They also made the point that a simulation would create its own ‘laws of physics’ that would limit possibility. I’ve been making the same point for decades that the *perceived* laws of physics relate only to this reality, or what I would later call the simulation. When designers write codes to create computer and virtual reality games they are the equivalent of the laws of physics for that game. Players interact within the limitations laid out by the coding. In the same way those who wrote the codes for the simulation decided the laws of physics that would apply. These can be overridden by expanded states of consciousness, but not by those enslaved in only five-sense awareness where simulation codes rule. Overriding the codes is what people call ‘miracles’. They are not. They are bypassing the encoded limits of the simulation. A population caught in simulation perception would have no idea that this was their plight. As the Bonn paper said: ‘Like a prisoner in a pitch-black cell we would not be able to see the “walls” of our prison.’ That’s true if people remain mesmerised by the five senses. Open to expanded awareness and those walls become very clear. The main one is the speed of light.

American theoretical physicist James Gates is another who has explored the simulation question and found considerable evidence to support the idea. Gates was Professor of Physics at the University of Maryland, Director of The Center for String and Particle Theory, and on Barack Obama’s Council of Advisors on Science and Technology. He and his team found *computer codes* of digital data embedded in the fabric of our reality. They relate to on-off electrical charges of 1 and 0 in the binary system used by computers. ‘We have no idea what they are doing there’, Gates said. They found within the energetic fabric mathematical sequences known as error-correcting codes or block codes that ‘reboot’ data to its original state or ‘default settings’ when something knocks it out of sync. Gates was asked if he had found a set of equations embedded in our reality indistinguishable from those that drive search engines and browsers and he said: ‘That is correct.’ Rich Terrile, director of the Centre for Evolutionary Computation and Automated Design at NASA’s Jet

Propulsion Laboratory, has said publicly that he believes the Universe is a digital hologram that must have been created by a form of intelligence. I agree with that in every way. Waveform information is delivered electrically by the senses to the brain which constructs a *digital* holographic reality that we call the ‘world’. This digital level of reality can be read by the esoteric art of numerology. Digital holograms are at the cutting edge of holographics today. We have digital technology everywhere designed to access and manipulate our digital level of perceived reality. Synthetic mRNA in ‘Covid vaccines’ has a digital component to manipulate the body’s digital ‘operating system’.

Reality is numbers

How many know that our reality can be broken down to numbers and codes that are the same as computer games? Max Tegmark, a physicist at the Massachusetts Institute of Technology (MIT), is the author of *Our Mathematical Universe* in which he lays out how reality can be entirely described by numbers and maths in the way that a video game is encoded with the ‘physics’ of computer games. Our world and computer virtual reality are essentially the same.

Tegmark imagines the perceptions of characters in an advanced computer game when the graphics are so good they don’t know they are in a game. They think they can bump into real objects (electromagnetic resistance in our reality), fall in love and feel emotions like excitement. When they began to study the apparently ‘physical world’ of the video game they would realise that everything was made of pixels (which have been found in our energetic reality as must be the case when on one level our world is digital). What computer game characters thought was physical ‘stuff’, Tegmark said, could actually be broken down into numbers:

And we’re exactly in this situation in our world. We look around and it doesn’t seem that mathematical at all, but everything we see is made out of elementary particles like quarks and electrons. And what properties does an electron have? Does it have a smell or a colour or a texture? No! ... We physicists have come up with geeky names for [Electron] properties, like

electric charge, or spin, or lepton number, but the electron doesn't care what we call it, the properties are just numbers.

This is the illusory reality Gnostics were describing. This is the simulation. The A, C, G, and T codes of DNA have a binary value – A and C = 0 while G and T = 1. This has to be when the simulation is digital and the body must be digital to interact with it. Recurring mathematical sequences are encoded throughout reality and the body. They include the Fibonacci sequence in which the two previous numbers are added to get the next one, as in ... 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, etc. The sequence is encoded in the human face and body, proportions of animals, DNA, seed heads, pine cones, trees, shells, spiral galaxies, hurricanes and the number of petals in a flower. The list goes on and on. There are fractal patterns – a 'never-ending pattern that is infinitely complex and self-similar across all scales in the as above, so below, principle of holograms. These and other famous recurring geometrical and mathematical sequences such as Phi, Pi, Golden Mean, Golden Ratio and Golden Section are *computer codes* of the simulation. I had to laugh and give my head a shake the day I finished this book and it went into the production stage. I was sent an article in *Scientific American* published in April, 2021, with the headline 'Confirmed! We Live in a Simulation'. Two decades after I first said our reality is a simulation and the speed of light is its outer limit the article suggested that we do live in a simulation and that the speed of light is its outer limit. I left school at 15 and never passed a major exam in my life while the writer was up to his eyes in qualifications. As I will explain in the final chapter *knowing* is far better than thinking and they come from very different sources. The article rightly connected the speed of light to the processing speed of the 'Matrix' and said what has been in my books all this time ... 'If we are in a simulation, as it appears, then space is an abstract property written in code. It is not real'. No it's not and if we live in a simulation something created it and it wasn't *us*. 'That David Icke says we are manipulated by aliens' – he's crackers.'

Wow ...

The reality that humanity thinks is so real is an illusion. Politicians, governments, scientists, doctors, academics, law enforcement, media, school and university curriculums, on and on, are all founded on a world that *does not exist* except as a simulated prison cell. Is it such a stretch to accept that 'Covid' doesn't exist when our entire 'physical' reality doesn't exist? Revealed here is the knowledge kept under raps in the Cult networks of compartmentalised secrecy to control humanity's sense of reality by inducing the population to believe in a reality that's not real. If it wasn't so tragic in its experiential consequences the whole thing would be hysterically funny. None of this is new to Renegade Minds. Ancient Greek philosopher Plato (about 428 to about 347BC) was a major influence on Gnostic belief and he described the human plight thousands of years ago with his Allegory of the Cave. He told the symbolic story of prisoners living in a cave who had never been outside. They were chained and could only see one wall of the cave while behind them was a fire that they could not see. Figures walked past the fire casting shadows on the prisoners' wall and those moving shadows became their sense of reality. Some prisoners began to study the shadows and were considered experts on them (today's academics and scientists), but what they studied was only an illusion (today's academics and scientists). A prisoner escaped from the cave and saw reality as it really is. When he returned to report this revelation they didn't believe him, called him mad and threatened to kill him if he tried to set them free. Plato's tale is not only a brilliant analogy of the human plight and our illusory reality. It describes, too, the dynamics of the 'Covid' hoax. I have only skimmed the surface of these subjects here. The aim of this book is to crisply connect all essential dots to put what is happening today into its true context. All subject areas and their connections in this chapter are covered in great evidential detail in *Everything You Need To Know, But Have Never Been Told* and *The Answer*.

They say that bewildered people 'can't see the forest for the trees'. Humanity, however, can't see the forest for the *twigs*. The five senses

see only twigs while Renegade Minds can see the forest and it's the forest where the answers lie with the connections that reveals. Breaking free of perceptual programming so the forest can be seen is the way we turn all this around. Not breaking free is how humanity got into this mess. The situation may seem hopeless, but I promise you it's not. We are a perceptual heartbeat from paradise if only we knew.

CHAPTER TWELVE

Escaping Wetiko

Life is simply a vacation from the infinite

Dean Cavanagh

Renegade Minds weave the web of life and events and see common themes in the apparently random. They are always there if you look for them and their pursuit is aided by incredible synchronicity that comes when your mind is open rather than mesmerised by what it thinks it can see.

Infinite awareness is infinite possibility and the more of infinite possibility that we access the more becomes infinitely possible. That may be stating the apparently obvious, but it is a devastatingly-powerful fact that can set us free. We are a point of attention within an infinity of consciousness. The question is how much of that infinity do we choose to access? How much knowledge, insight, awareness, wisdom, do we want to connect with and explore? If your focus is only in the five senses you will be influenced by a fraction of infinite awareness. I mean a range so tiny that it gives new meaning to infinitesimal. Limitation of self-identity and a sense of the possible limit accordingly your range of consciousness. We are what we think we are. Life is what we think it is. The dream is the dreamer and the dreamer is the dream. Buddhist philosophy puts it this way: 'As a thing is viewed, so it appears.' Most humans live in the realm of touch, taste, see, hear, and smell and that's the limit of their sense of the possible and sense of self. Many will follow a religion and speak of a God in his heaven, but their lives are still

dominated by the five senses in their perceptions and actions. The five senses become the arbiter of everything. When that happens all except a smear of infinity is sealed away from influence by the rigid, unyielding, reality bubbles that are the five-sense human or Phantom Self. Archon Cult methodology is to isolate consciousness within five-sense reality – the simulation – and then program that consciousness with a sense of self and the world through a deluge of life-long information designed to instil the desired perception that allows global control. Efforts to do this have increased dramatically with identity politics as identity bubbles are squeezed into the minutiae of five-sense detail which disconnect people even more profoundly from the infinite ‘I’.

Five-sense focus and self-identity are like a firewall that limits access to the infinite realms. You only perceive one radio or television station and no other. We’ll take that literally for a moment. Imagine a vast array of stations giving different information and angles on reality, but you only ever listen to one. Here we have the human plight in which the population is overwhelmingly confined to CultFM. This relates only to the frequency range of CultFM and limits perception and insight to that band – limits *possibility* to that band. It means you are connecting with an almost imperceptibly minuscule range of possibility and creative potential within the infinite Field. It’s a world where everything seems apart from everything else and where synchronicity is rare. Synchronicity is defined in the dictionary as ‘the happening by chance of two or more related or similar events at the same time’. Use of ‘by chance’ betrays a complete misunderstanding of reality. Synchronicity is not ‘by chance’. As people open their minds, or ‘awaken’ to use the term, they notice more and more coincidences in their lives, bits of ‘luck’, apparently miraculous happenings that put them in the right place at the right time with the right people. Days become peppered with ‘fancy meeting you here’ and ‘what are the chances of that?’ My entire life has been lived like this and ever more so since my own colossal awakening in 1990 and 91 which transformed my sense of reality. Synchronicity is not ‘by chance’; it is by accessing expanded

realms of possibility which allow expanded potential for manifestation. People broadcasting the same vibe from the same openness of mind tend to be drawn ‘by chance’ to each other through what I call frequency magnetism and it’s not only people. In the last more than 30 years incredible synchronicity has also led me through the Cult maze to information in so many forms and to crucial personal experiences. These ‘coincidences’ have allowed me to put the puzzle pieces together across an enormous array of subjects and situations. Those who have breached the bubble of five-sense reality will know exactly what I mean and this escape from the perceptual prison cell is open to everyone whenever they make that choice. This may appear super-human when compared with the limitations of ‘human’, but it’s really our natural state. ‘Human’ as currently experienced is consciousness in an unnatural state of induced separation from the infinity of the whole. I’ll come to how this transformation into unity can be made when I have described in more detail the force that holds humanity in servitude by denying this access to infinite self.

The Wetiko factor

I have been talking and writing for decades about the way five-sense mind is systematically barricaded from expanded awareness. I have used the analogy of a computer (five-sense mind) and someone at the keyboard (expanded awareness). Interaction between the computer and the operator is symbolic of the interaction between five-sense mind and expanded awareness. The computer directly experiences the Internet and the operator experiences the Internet via the computer which is how it’s supposed to be – the two working as one. Archons seek to control that point where the operator connects with the computer to stop that interaction ([Fig 20](#)). Now the operator is banging the keyboard and clicking the mouse, but the computer is not responding and this happens when the computer is taken over – *possessed* – by an appropriately-named computer ‘virus’. The operator has lost all influence over the computer which goes its own way making decisions under the control of the ‘virus’. I have

just described the dynamic through which the force known to Gnostics as Yaldabaoth and Archons disconnects five-sense mind from expanded awareness to imprison humanity in perceptual servitude.



Figure 20: The mind ‘virus’ I have been writing about for decades seeks to isolate five-sense mind (the computer) from the true ‘I’. (Image by Neil Hague).

About a year ago I came across a Native American concept of Wetiko which describes precisely the same phenomenon. Wetiko is the spelling used by the Cree and there are other versions including wintiko and windigo used by other tribal groups. They spell the name with lower case, but I see Wetiko as a proper noun as with Archons and prefer a capital. I first saw an article about Wetiko by writer and researcher Paul Levy which so synced with what I had been writing about the computer/operator disconnection and later the Archons. I then read his book, the fascinating *Dispelling Wetiko, Breaking the Spell of Evil*. The parallels between what I had concluded long before and the Native American concept of Wetiko were so clear and obvious that it was almost funny. For Wetiko see the Gnostic Archons for sure and the Jinn, the Predators, and every other name for a force of evil, inversion and chaos. Wetiko is the Native American name for the force that divides the computer from

the operator ([Fig 21](#)). Indigenous author Jack D. Forbes, a founder of the Native American movement in the 1960s, wrote another book about Wetiko entitled *Columbus And Other Cannibals – The Wetiko Disease of Exploitation, Imperialism, and Terrorism* which I also read. Forbes says that Wetiko refers to an evil person or spirit ‘who terrorizes other creatures by means of terrible acts, including cannibalism’. Zulu shaman Credo Mutwa told me that African accounts tell how cannibalism was brought into the world by the Chitauri ‘gods’ – another manifestation of Wetiko. The distinction between ‘evil person or spirit’ relates to Archons/Wetiko possessing a human or acting as pure consciousness. Wetiko is said to be a sickness of the soul or spirit and a state of being that takes but gives nothing back – the Cult and its operatives perfectly described. Black Hawk, a Native American war leader defending their lands from confiscation, said European invaders had ‘poisoned hearts’ – Wetiko hearts – and that this would spread to native societies. Mention of the heart is very significant as we shall shortly see. Forbes writes: ‘Tragically, the history of the world for the past 2,000 years is, in great part, the story of the epidemiology of the wetiko disease.’ Yes, and much longer. Forbes is correct when he says: ‘The wetikos destroyed Egypt and Babylon and Athens and Rome and Tenochtitlan [capital of the Aztec empire] and perhaps now they will destroy the entire earth.’ Evil, he said, is the number one export of a Wetiko culture – see its globalisation with ‘Covid’. Constant war, mass murder, suffering of all kinds, child abuse, Satanism, torture and human sacrifice are all expressions of Wetiko and the Wetiko possessed. The world is Wetiko made manifest, *but it doesn’t have to be*. There is a way out of this even now.



Figure 21: The mind ‘virus’ is known to Native Americans as ‘Wetiko’. (Image by Neil Hague).

Cult of Wetiko

Wetiko is the Yaldabaoth frequency distortion that seeks to attach to human consciousness and absorb it into its own. Once this connection is made Wetiko can drive the perceptions of the target which they believe to be coming from their own mind. All the horrors of history and today from mass killers to Satanists, paedophiles like Jeffrey Epstein and other psychopaths, are the embodiment of Wetiko and express its state of being in all its grotesqueness. The Cult is Wetiko incarnate, Yaldabaoth incarnate, and it seeks to facilitate Wetiko assimilation of humanity in totality into its distortion by manipulating the population into low frequency states that match its own. Paul Levy writes: ‘Holographically enforced within the psyche of every human being the wetiko virus pervades and underlies the entire field of consciousness, and can therefore potentially manifest through any one of us at any moment if we are not mindful.’ The ‘Covid’ hoax has achieved this with many people, but others have not fallen into Wetiko’s frequency lair. Players in the ‘Covid’ human catastrophe including Gates, Schwab, Tedros, Fauci, Whitty, Vallance, Johnson, Hancock, Ferguson, Drosten, and all the rest, including the psychopath psychologists, are expressions of Wetiko. This is why

they have no compassion or empathy and no emotional consequence for what they do that would make them stop doing it. Observe all the people who support the psychopaths in authority against the Pushbackers despite the damaging impact the psychopaths have on their own lives and their family's lives. You are again looking at Wetiko possession which prevents them seeing through the lies to the obvious scam going on. *Why can't they see it?* Wetiko won't let them see it. The perceptual divide that has now become a chasm is between the Wetikoed and the non-Wetikoed.

Paul Levy describes Wetiko in the same way that I have long described the Archontic force. They are the same distorted consciousness operating across dimensions of reality: '... the subtle body of wetiko is not located in the third dimension of space and time, literally existing in another dimension ... it is able to affect ordinary lives by mysteriously interpenetrating into our three-dimensional world.' Wetiko does this through its incarnate representatives in the Cult and by weaving itself into The Field which on our level of reality is the electromagnetic information field of the simulation or Matrix. More than that, the simulation *is* Wetiko / Yaldabaoth. Caleb Scharf, Director of Astrobiology at Columbia University, has speculated that 'alien life' could be so advanced that it has transcribed itself into the quantum realm to become what we call physics. He said intelligence indistinguishable from the fabric of the Universe would solve many of its greatest mysteries:

Perhaps hyper-advanced life isn't just external. Perhaps it's already all around. It is embedded in what we perceive to be physics itself, from the root behaviour of particles and fields to the phenomena of complexity and emergence ... In other words, life might not just be in the equations. It might BE the equations [My emphasis].

Scharf said it is possible that 'we don't recognise advanced life because it forms an integral and unsuspicious part of what we've considered to be the natural world'. I agree. Wetiko/Yaldabaoth *is* the simulation. We are literally in the body of the beast. But that doesn't mean it has to control us. We all have the power to overcome Wetiko

influence and the Cult knows that. I doubt it sleeps too well because it knows that.

Which Field?

This, I suggest, is how it all works. There are two Fields. One is the fierce electromagnetic light of the Matrix within the speed of light; the other is the ‘watery light’ of The Field beyond the walls of the Matrix that connects with the Great Infinity. Five-sense mind and the decoding systems of the body attach us to the Field of Matrix light. They have to or we could not experience this reality. Five-sense mind sees only the Matrix Field of information while our expanded consciousness is part of the Infinity Field. When we open our minds, and most importantly our hearts, to the Infinity Field we have a mission control which gives us an expanded perspective, a road map, to understand the nature of the five-sense world. If we are isolated only in five-sense mind there is no mission control. We’re on our own trying to understand a world that’s constantly feeding us information to ensure we do not understand. People in this state can feel ‘lost’ and bewildered with no direction or radar. You can see ever more clearly those who are influenced by the Fields of Big Infinity or little five-sense mind simply by their views and behaviour with regard to the ‘Covid’ hoax. We have had this division throughout known human history with the mass of the people on one side and individuals who could see and intuit beyond the walls of the simulation – Plato’s prisoner who broke out of the cave and saw reality for what it is. Such people have always been targeted by Wetiko/Archon-possessed authority, burned at the stake or demonised as mad, bad and dangerous. The Cult today and its global network of ‘anti-hate’, ‘anti-fascist’ Woke groups are all expressions of Wetiko attacking those exposing the conspiracy, ‘Covid’ lies and the ‘vaccine’ agenda.

Woke as a whole is Wetiko which explains its black and white mentality and how at one it is with the Wetiko-possessed Cult. Paul Levy said: ‘To be in this paradigm is to still be under the thrall of a two-valued logic – where things are either true or false – of a

wetikoized mind.' Wetiko consciousness is in a permanent rage, therefore so is Woke, and then there is Woke inversion and contradiction. 'Anti-fascists' act like fascists because fascists *and* 'anti-fascists' are both Wetiko at work. Political parties act the same while claiming to be different for the same reason. Secret society and satanic rituals are attaching initiates to Wetiko and the cold, ruthless, psychopathic mentality that secures the positions of power all over the world is Wetiko. Reframing 'training programmes' have the same cumulative effect of attaching Wetiko and we have their graduates described as automatons and robots with a cold, psychopathic, uncaring demeanour. They are all traits of Wetiko possession and look how many times they have been described in this book and elsewhere with regard to personnel behind 'Covid' including the police and medical profession. Climbing the greasy pole in any profession in a Wetiko society requires traits of Wetiko to get there and that is particularly true of politics which is not about fair competition and pre-eminence of ideas. It is founded on how many backs you can stab and arses you can lick. This culminated in the global 'Covid' coordination between the Wetiko possessed who pulled it off in all the different countries without a trace of empathy and compassion for their impact on humans. Our sight sense can see only holographic form and not the Field which connects holographic form. Therefore we perceive 'physical' objects with 'space' in between. In fact that 'space' is energy/consciousness operating on multiple frequencies. One of them is Wetiko and that connects the Cult psychopaths, those who submit to the psychopaths, and those who serve the psychopaths in the media operations of the world. Wetiko is Gates. Wetiko is the mask-wearing submissive. Wetiko is the fake journalist and 'fact-checker'. The Wetiko Field is coordinating the whole thing. Psychopaths, gofers, media operatives, 'anti-hate' hate groups, 'fact-checkers' and submissive people work as one unit *even without human coordination* because they are attached to the *same* Field which is organising it all ([Fig 22](#)). Paul Levy is here describing how Wetiko-possessed people are drawn together and refuse to let any information breach their rigid

perceptions. He was writing long before ‘Covid’, but I think you will recognise followers of the ‘Covid’ religion *oh just a little bit*:

People who are channelling the vibratory frequency of wetiko align with each other through psychic resonance to reinforce their unspoken shared agreement so as to uphold their deranged view of reality. Once an unconscious content takes possession of certain individuals, it irresistibly draws them together by mutual attraction and knits them into groups tied together by their shared madness that can easily swell into an avalanche of insanity.

A psychic epidemic is a closed system, which is to say that it is insular and not open to any new information or informing influences from the outside world which contradict its fixed, limited, and limiting perspective.

There we have the Woke mind and the ‘Covid’ mind. Compatible resonance draws the awakening together, too, which is clearly happening today.

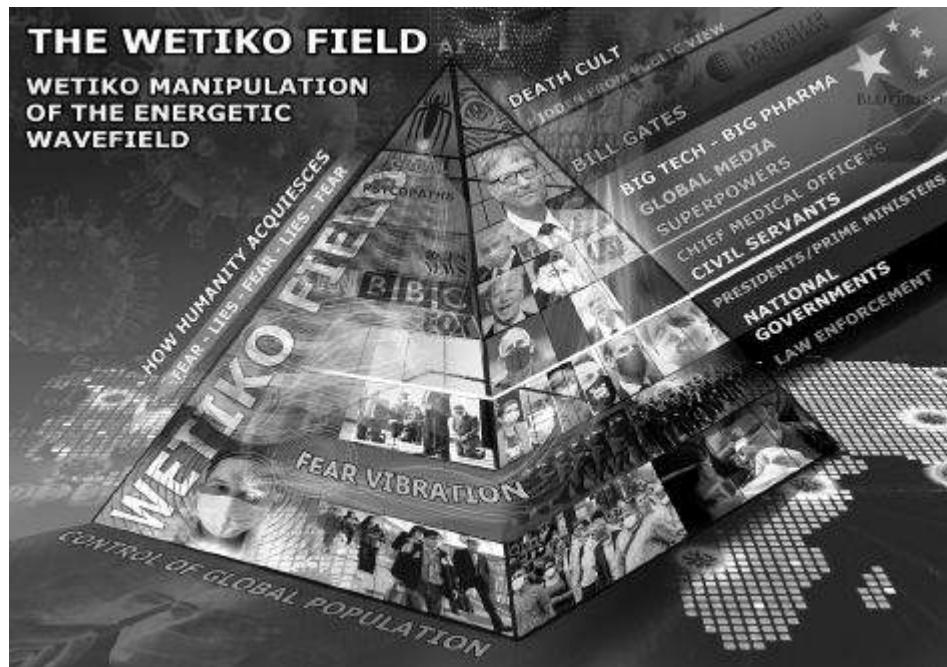


Figure 22: The Wetiko Field from which the Cult pyramid and its personnel are made manifest. (Image by Neil Hague).

Spiritual servitude

Wetiko doesn’t care about humans. It’s not human; it just possesses humans for its own ends and the effect (depending on the scale of

possession) can be anything from extreme psychopathy to unquestioning obedience. Wetiko's worst nightmare is for human consciousness to expand beyond the simulation. Everything is focussed on stopping that happening through control of information, thus perception, thus frequency. The 'education system', media, science, medicine, academia, are all geared to maintaining humanity in five-sense servitude as is the constant stimulation of low-vibrational mental and emotional states (see 'Covid'). Wetiko seeks to dominate those subconscious spaces between five-sense perception and expanded consciousness where the computer meets the operator. From these subconscious hiding places Wetiko speaks to us to trigger urges and desires that we take to be our own and manipulate us into anything from low-vibrational to psychopathic states. Remember how Islam describes the Jinn as invisible tricksters that 'whisper' and confuse. Wetiko is the origin of the 'trickster god' theme that you find in cultures all over the world. Jinn, like the Archons, are Wetiko which is terrified of humans awakening and reconnecting with our true self for then its energy source has gone. With that the feedback loop breaks between Wetiko and human perception that provides the energetic momentum on which its very existence depends as a force of evil. Humans are both its target and its source of survival, but only if we are operating in low-vibrational states of fear, hate, depression and the background anxiety that most people suffer. We are Wetiko's target because we are its key to survival. It needs us, not the other way round. Paul Levy writes:

A vampire has no intrinsic, independent, substantial existence in its own right; it only exists in relation to us. The pathogenic, vampiric mind-parasite called wetiko is nothing in itself – not being able to exist from its own side – yet it has a 'virtual reality' such that it can potentially destroy our species ...

...The fact that a vampire is not reflected by a mirror can also mean that what we need to see is that there's nothing, no-thing to see, other than ourselves. The fact that wetiko is the expression of something inside of us means that the cure for wetiko is with us as well. The critical issue is finding this cure within us and then putting it into effect.

Evil begets evil because if evil does not constantly expand and find new sources of energetic sustenance its evil, its *distortion*, dies with the assimilation into balance and harmony. Love is the garlic to Wetiko's vampire. Evil, the absence of love, cannot exist in the presence of love. I think I see a way out of here. I have emphasised so many times over the decades that the Archons/Wetiko and their Cult are not all powerful. *They are not.* I don't care how it looks even now *they are not.* I have not called them little boys in short trousers for effect. I have said it because it is true. Wetiko's insatiable desire for power over others is not a sign of its omnipotence, but its insecurity. Paul Levy writes: 'Due to the primal fear which ultimately drives it and which it is driven to cultivate, wetiko's body politic has an intrinsic and insistent need for centralising power and control so as to create imagined safety for itself.' *Yeeeeees!* Exactly! Why does Wetiko want humans in an ongoing state of fear? Wetiko itself *is* fear and it is petrified of love. As evil is an absence of love, so love is an absence of fear. Love conquers all and *especially* Wetiko which *is* fear. Wetiko brought fear into the world when it wasn't here before. *Fear* was the 'fall', the fall into low-frequency ignorance and illusion – fear is False Emotion Appearing Real. The simulation is driven and energised by fear because Wetiko/Yaldabaoth (fear) *are* the simulation. Fear is the absence of love and Wetiko is the absence of love.

Wetiko today

We can now view current events from this level of perspective. The 'Covid' hoax has generated momentous amounts of ongoing fear, anxiety, depression and despair which have empowered Wetiko. No wonder people like Gates have been the instigators when they are Wetiko incarnate and exhibit every trait of Wetiko in the extreme. See how cold and unemotional these people are like Gates and his cronies, how dead of eye they are. That's Wetiko. Sabbatians are Wetiko and everything they control including the World Health Organization, Big Pharma and the 'vaccine' makers, national 'health'

hierarchies, corporate media, Silicon Valley, the banking system, and the United Nations with its planned transformation into world government. All are controlled and possessed by the Wetiko distortion into distorting human society in its image. We are with this knowledge at the gateway to understanding the world.

Divisions of race, culture, creed and sexuality are diversions to hide the real division between those possessed and influenced by Wetiko and those that are not. The ‘Covid’ hoax has brought both clearly into view. Human behaviour is not about race. Tyrants and dictatorships come in all colours and creeds. What unites the US president bombing the innocent and an African tribe committing genocide against another as in Rwanda? What unites them? *Wetiko*. All wars are Wetiko, all genocide is Wetiko, all hunger over centuries in a world of plenty is Wetiko. Children going to bed hungry, including in the West, is Wetiko. Cult-generated Woke racial divisions that focus on the body are designed to obscure the reality that divisions in behaviour are manifestations of mind, not body. Obsession with body identity and group judgement is a means to divert attention from the real source of behaviour – mind and perception. Conflict sown by the Woke both within themselves and with their target groups are Wetiko providing lunch for itself through still more agents of the division, chaos, and fear on which it feeds. The Cult is seeking to assimilate the entirety of humanity and all children and young people into the Wetiko frequency by manipulating them into states of fear and despair. Witness all the suicide and psychological unravelling since the spring of 2020. Wetiko psychopaths want to impose a state of unquestioning obedience to authority which is no more than a conduit for Wetiko to enforce its will and assimilate humanity into itself. It needs us to believe that resistance is futile when it fears resistance and even more so the game-changing non-cooperation with its impositions. It can use violent resistance for its benefit. Violent impositions and violent resistance are *both* Wetiko. The Power of Love with its Power of No will sweep Wetiko from our world. Wetiko and its Cult know that. They just don’t want us to know.

AI Wetiko

This brings me to AI or artificial intelligence and something else Wetikos don't want us to know. What is AI *really*? I know about computer code algorithms and AI that learns from data input. These, however, are more diversions, the expeditionary force, for the real AI that they want to connect to the human brain as promoted by Silicon Valley Wetikos like Kurzweil. What is this AI? It is the frequency of *Wetiko*, the frequency of the Archons. The connection of AI to the human brain is the connection of the Wetiko frequency to create a Wetiko hive mind and complete the job of assimilation. The hive mind is planned to be controlled from Israel and China which are both 100 percent owned by Wetiko Sabbatians. The assimilation process has been going on minute by minute in the 'smart' era which fused with the 'Covid' era. We are told that social media is scrambling the minds of the young and changing their personality. This is true, but what is social media? Look more deeply at how it works, how it creates divisions and conflict, the hostility and cruelty, the targeting of people until they are destroyed. That's Wetiko. Social media is manipulated to tune people to the Wetiko frequency with all the emotional exploitation tricks employed by platforms like Facebook and its Wetiko front man, Zuckerberg. Facebook's Instagram announced a new platform for children to overcome a legal bar on them using the main site. This is more Wetiko exploitation and manipulation of kids. Amnesty International likened the plan to foxes offering to guard the henhouse and said it was incompatible with human rights. Since when did Wetiko or Zuckerberg (I repeat myself) care about that? Would Brin and Page at Google, Wojcicki at YouTube, Bezos at Amazon and whoever the hell runs Twitter act as they do if they were not channelling Wetiko? Would those who are developing technologies for no other reason than human control? How about those designing and selling technologies to kill people and Big Pharma drug and 'vaccine' producers who know they will end or devastate lives? Quite a thought for these people to consider is that if you are Wetiko in a human life you are Wetiko on the 'other side' unless your frequency

changes and that can only change by a change of perception which becomes a change of behaviour. Where Gates is going does not bear thinking about although perhaps that's exactly where he wants to go. Either way, that's where he's going. His frequency will make it so.

The frequency lair

I have been saying for a long time that a big part of the addiction to smartphones and devices is that a frequency is coming off them that entraps the mind. People spend ages on their phones and sometimes even a minute or so after they put them down they pick them up again and it all repeats. 'Covid' lockdowns will have increased this addiction a million times for obvious reasons. Addictions to alcohol overindulgence and drugs are another way that Wetiko entraps consciousness to attach to its own. Both are symptoms of low-vibrational psychological distress which alcoholism and drug addiction further compound. Do we think it's really a coincidence that access to them is made so easy while potions that can take people into realms beyond the simulation are banned and illegal? I have explored smartphone addiction in other books, the scale is mind-blowing, and that level of addiction does not come without help. Tech companies that make these phones are Wetiko and they will have no qualms about destroying the minds of children. We are seeing again with these companies the Wetiko perceptual combination of psychopathic enforcers and weak and meek unquestioning compliance by the rank and file.

The global Smart Grid is the Wetiko Grid and it is crucial to complete the Cult endgame. The simulation is radiation and we are being deluged with technological radiation on a devastating scale. Wetiko frauds like Elon Musk serve Cult interests while occasionally criticising them to maintain his street-cred. 5G and other forms of Wi-Fi are being directed at the earth from space on a volume and scale that goes on increasing by the day. Elon Musk's (officially) SpaceX Starlink project is in the process of putting tens of thousands of satellites in low orbit to cover every inch of the planet with 5G and other Wi-Fi to create Kurzweil's global 'cloud' to which the

human mind is planned to be attached very soon. SpaceX has approval to operate 12,000 satellites with more than 1,300 launched at the time of writing and applications filed for 30,000 more. Other operators in the Wi-Fi, 5G, low-orbit satellite market include OneWeb (UK), Telesat (Canada), and AST & Science (US). Musk tells us that AI could be the end of humanity and then launches a company called Neuralink to connect the human brain to computers. Musk's (in theory) Tesla company is building electric cars and the driverless vehicles of the smart control grid. As frauds and bullshitters go Elon Musk in my opinion is Major League.

5G and technological radiation in general are destructive to human health, genetics and psychology and increasing the strength of artificial radiation underpins the five-sense perceptual bubbles which are themselves expressions of radiation or electromagnetism. Freedom activist John Whitehead was so right with his 'databit by databit, we are building our own electronic concentration camps'. The Smart Grid and 5G is a means to control the human mind and infuse perceptual information into The Field to influence anyone in sync with its frequency. You can change perception and behaviour en masse if you can manipulate the population into those levels of frequency and this is happening all around us today. The arrogance of Musk and his fellow Cult operatives knows no bounds in the way that we see with Gates. Musk's satellites are so many in number already they are changing the night sky when viewed from Earth. The astronomy community has complained about this and they have seen nothing yet. Some consequences of Musk's Wetiko hubris include: Radiation; visible pollution of the night sky; interference with astronomy and meteorology; ground and water pollution from intensive use of increasingly many spaceports; accumulating space debris; continual deorbiting and burning up of aging satellites, polluting the atmosphere with toxic dust and smoke; and ever-increasing likelihood of collisions. A collective public open letter of complaint to Musk said:

We are writing to you ... because SpaceX is in process of surrounding the Earth with a network of thousands of satellites whose very purpose is to irradiate every square inch of the

Earth. SpaceX, like everyone else, is treating the radiation as if it were not there. As if the mitochondria in our cells do not depend on electrons moving undisturbed from the food we digest to the oxygen we breathe.

As if our nervous systems and our hearts are not subject to radio frequency interference like any piece of electronic equipment. As if the cancer, diabetes, and heart disease that now afflict a majority of the Earth's population are not metabolic diseases that result from interference with our cellular machinery. As if insects everywhere, and the birds and animals that eat them, are not starving to death as a result.

People like Musk and Gates believe in their limitless Wetiko arrogance that they can do whatever they like to the world because they own it. Consequences for humanity are irrelevant. It's absolutely time that we stopped taking this shit from these self-styled masters of the Earth when you consider where this is going.

Why is the Cult so anti-human?

I hear this question often: Why would they do this when it will affect them, too? Ah, but will it? Who is this *them*? Forget their bodies. They are just vehicles for Wetiko consciousness. When you break it all down to the foundations we are looking at a state of severely distorted consciousness targeting another state of consciousness for assimilation. The rest is detail. The simulation is the fly-trap in which unique sensations of the five senses create a cycle of addiction called reincarnation. Renegade Minds see that everything which happens in our reality is a smaller version of the whole picture in line with the holographic principle. Addiction to the radiation of smart technology is a smaller version of addiction to the whole simulation. Connecting the body/brain to AI is taking that addiction on a giant step further to total ongoing control by assimilating human incarnate consciousness into Wetiko. I have watched during the 'Covid' hoax how many are becoming ever more profoundly attached to Wetiko's perceptual calling cards of aggressive response to any other point of view ('There is no other god but me'), psychopathic lack of compassion and empathy, and servile submission to the narrative and will of authority. Wetiko is the psychopaths *and* subservience to psychopaths. The Cult of Wetiko is

so anti-human because it is *not* human. It embarked on a mission to destroy human by targeting everything that it means to be human and to survive as human. ‘Covid’ is not the end, just a means to an end. The Cult with its Wetiko consciousness is seeking to change Earth systems, including the atmosphere, to suit them, not humans. The gathering bombardment of 5G alone from ground and space is dramatically changing The Field with which the five senses interact. There is so much more to come if we sit on our hands and hope it will all go away. It is not meant to go away. It is meant to get ever more extreme and we need to face that while we still can – just.

Carbon dioxide is the gas of life. Without that human is over. Kaput, gone, history. No natural world, no human. The Cult has created a cock and bull story about carbon dioxide and climate change to justify its reduction to the point where Gates and the ignoramus Biden ‘climate chief’ John Kerry want to suck it out of the atmosphere. Kerry wants to do this because his master Gates does. Wetikos have made the gas of life a demon with the usual support from the Wokers of Extinction Rebellion and similar organisations and the bewildered puppet-child that is Greta Thunberg who was put on the world stage by Klaus Schwab and the World Economic Forum. The name Extinction Rebellion is both ironic and as always Wetiko inversion. The gas that we need to survive must be reduced to save us from extinction. The most basic need of human is oxygen and we now have billions walking around in face nappies depriving body and brain of this essential requirement of human existence. More than that 5G at 60 gigahertz interacts with the oxygen molecule to reduce the amount of oxygen the body can absorb into the bloodstream. The obvious knock-on consequences of that for respiratory and cognitive problems and life itself need no further explanation. Psychopaths like Musk are assembling a global system of satellites to deluge the human atmosphere with this insanity. The man should be in jail. Here we have two most basic of human needs, oxygen and carbon dioxide, being dismantled.

Two others, water and food, are getting similar treatment with the United Nations Agendas 21 and 2030 – the Great Reset – planning to

centrally control all water and food supplies. People will not even own rain water that falls on their land. Food is affected at the most basic level by reducing carbon dioxide. We have genetic modification or GMO infiltrating the food chain on a mass scale, pesticides and herbicides polluting the air and destroying the soil. Freshwater fish that provide livelihoods for 60 million people and feed hundreds of millions worldwide are being 'pushed to the brink' according the conservationists while climate change is the only focus. Now we have Gates and Schwab wanting to dispense with current food sources all together and replace them with a synthetic version which the Wetiko Cult would control in terms of production and who eats and who doesn't. We have been on the Totalitarian Tiptoe to this for more than 60 years as food has become ever more processed and full of chemical shite to the point today when it's not natural food at all. As Dr Tom Cowan says: 'If it has a label don't eat it.' Bill Gates is now the biggest owner of farmland in the United States and he does nothing without an ulterior motive involving the Cult. Klaus Schwab wrote: 'To feed the world in the next 50 years we will need to produce as much food as was produced in the last 10,000 years ... food security will only be achieved, however, if regulations on genetically modified foods are adapted to reflect the reality that gene editing offers a precise, efficient and safe method of improving crops.' Liar. People and the world are being targeted with aluminium through vaccines, chemtrails, food, drink cans, and endless other sources when aluminium has been linked to many health issues including dementia which is increasing year after year. Insects, bees and wildlife essential to the food chain are being deleted by pesticides, herbicides and radiation which 5G is dramatically increasing with 6G and 7G to come. The pollinating bee population is being devastated while wildlife including birds, dolphins and whales are having their natural radar blocked by the effects of ever-increasing radiation. In the summer windscreens used to be splattered with insects so numerous were they. It doesn't happen now. Where have they gone?

Synthetic everything

The Cult is introducing genetically-modified versions of trees, plants and insects including a Gates-funded project to unleash hundreds of millions of genetically-modified, lab-altered and patented male mosquitoes to mate with wild mosquitoes and induce genetic flaws that cause them to die out. Clinically-insane Gates-funded Japanese researchers have developed mosquitos that spread vaccine and are dubbed 'flying vaccinators'. Gates is funding the modification of weather patterns in part to sell the myth that this is caused by carbon dioxide and he's funding geoengineering of the skies to change the atmosphere. Some of this came to light with the Gates-backed plan to release tonnes of chalk into the atmosphere to 'deflect the Sun and cool the planet'. Funny how they do this while the heating effect of the Sun is not factored into climate projections focussed on carbon dioxide. The reason is that they want to reduce carbon dioxide (so don't mention the Sun), but at the same time they do want to reduce the impact of the Sun which is so essential to human life and health. I have mentioned the sun-cholesterol-vitamin D connection as they demonise the Sun with warnings about skin cancer (caused by the chemicals in sun cream they tell you to splash on). They come from the other end of the process with statin drugs to reduce cholesterol that turns sunlight into vitamin D. A lack of vitamin D leads to a long list of health effects and how vitamin D levels must have fallen with people confined to their homes over 'Covid'. Gates is funding other forms of geoengineering and most importantly chemtrails which are dropping heavy metals, aluminium and self-replicating nanotechnology onto the Earth which is killing the natural world. See *Everything You Need To Know, But Have Never Been Told* for the detailed background to this.

Every human system is being targeted for deletion by a force that's not human. The Wetiko Cult has embarked on the process of transforming the human body from biological to synthetic biological as I have explained. Biological is being replaced by the artificial and synthetic – Archontic 'countermimicry' – right across human society. The plan eventually is to dispense with the human body altogether

and absorb human consciousness – which it wouldn't really be by then – into cyberspace (the simulation which is Wetiko/Yaldabaoth). Preparations for that are already happening if people would care to look. The alternative media rightly warns about globalism and 'the globalists', but this is far bigger than that and represents the end of the human race as we know it. The 'bad copy' of prime reality that Gnostics describe was a bad copy of harmony, wonder and beauty to start with before Wetiko/Yaldabaoth set out to change the simulated 'copy' into something very different. The process was slow to start with. Entrapped humans in the simulation timeline were not technologically aware and they had to be brought up to intellectual speed while being suppressed spiritually to the point where they could build their own prison while having no idea they were doing so. We have now reached that stage where technological intellect has the potential to destroy us and that's why events are moving so fast. Central American shaman Don Juan Matus said:

Think for a moment, and tell me how you would explain the contradictions between the intelligence of man the engineer and the stupidity of his systems of belief, or the stupidity of his contradictory behaviour. Sorcerers believe that the predators have given us our systems of beliefs, our ideas of good and evil; our social mores. They are the ones who set up our dreams of success or failure. They have given us covetousness, greed, and cowardice. It is the predator who makes us complacent, routinary, and egomaniacal.

In order to keep us obedient and meek and weak, the predators engaged themselves in a stupendous manoeuvre – stupendous, of course, from the point of view of a fighting strategist; a horrendous manoeuvre from the point of those who suffer it. They gave us their mind. The predators' mind is baroque, contradictory, morose, filled with the fear of being discovered any minute now.

For 'predators' see Wetiko, Archons, Yaldabaoth, Jinn, and all the other versions of the same phenomenon in cultures and religions all over the world. The theme is always the same because it's true and it's real. We have reached the point where we have to deal with it. The question is – how?

Don't fight – walk away

I thought I'd use a controversial subheading to get things moving in terms of our response to global fascism. What do you mean 'don't fight'? What do you mean 'walk away'? We've got to fight. We can't walk away. Well, it depends what we mean by fight and walk away. If fighting means physical combat we are playing Wetiko's game and falling for its trap. It wants us to get angry, aggressive, and direct hate and hostility at the enemy we think we must fight. Every war, every battle, every conflict, has been fought with Wetiko leading both sides. It's what it does. Wetiko wants a fight, anywhere, any place. Just hit me, son, so I can hit you back. Wetiko hits Wetiko and Wetiko hits Wetiko in return. I am very forthright as you can see in exposing Wetikos of the Cult, but I don't hate them. I refuse to hate them. It's what they want. What you hate you become. What you *fight* you become. Wokers, 'anti-haters' and 'anti-fascists' prove this every time they reach for their keyboards or don their balaclavas. By walk away I mean to disengage from Wetiko which includes ceasing to cooperate with its tyranny. Paul Levy says of Wetiko:

The way to 'defeat' evil is not to try to destroy it (for then, in playing evil's game, we have already lost), but rather, to find the invulnerable place within ourselves where evil is unable to vanquish us – this is to truly 'win' our battle with evil.

Wetiko is everywhere in human society and it's been on steroids since the 'Covid' hoax. Every shouting match over wearing masks has Wetiko wearing a mask and Wetiko not wearing one. It's an electrical circuit of push and resist, push and resist, with Wetiko pushing *and* resisting. Each polarity is Wetiko empowering itself. Dictionary definitions of 'resist' include 'opposing, refusing to accept or comply with' and the word to focus on is 'opposing'. What form does this take – setting police cars alight or 'refusing to accept or comply with'? The former is Wetiko opposing Wetiko while the other points the way forward. This is the difference between those aggressively demanding that government fascism must be obeyed who stand in stark contrast to the great majority of Pushbackers. We saw this clearly with a march by thousands of Pushbackers against lockdown in London followed days later by a Woker-hijacked

protest in Bristol in which police cars were set on fire. Masks were virtually absent in London and widespread in Bristol. Wetiko wants lockdown on every level of society and infuses its aggression to police it through its unknowing stooges. Lockdown protesters are the ones with the smiling faces and the hugs, The two blatantly obvious states of being – getting more obvious by the day – are the result of Wokers and their like becoming ever more influenced by the simulation Field of Wetiko and Pushbackers ever more influenced by The Field of a far higher vibration beyond the simulation. Wetiko can't invade the heart which is where most lockdown opponents are coming from. It's the heart that allows them to see through the lies to the truth in ways I will be highlighting.

Renegade Minds know that calmness is the place from which wisdom comes. You won't find wisdom in a hissing fit and wisdom is what we need in abundance right now. Calmness is not weakness – you don't have to scream at the top of your voice to be strong. Calmness is indeed a sign of strength. 'No' means I'm not doing it. NOOOO!!! doesn't mean you're not doing it even more. Volume does not advance 'No – I'm not doing it'. You are just not doing it. Wetiko possessed and influenced don't know how to deal with that. Wetiko wants a fight and we should not give it one. What it needs more than anything is our *cooperation* and we should not give that either. Mass rallies and marches are great in that they are a visual representation of feeling, but if it ends there they are irrelevant. You demand that Wetikos act differently? Well, they're not going to are they? They are Wetikos. We don't need to waste our time demanding that something doesn't happen when that will make no difference. We need to delete the means that *allows* it to happen. This, invariably, is our cooperation. You can demand a child stop firing a peashooter at the dog or you can refuse to buy the peashooter. If you provide the means you are cooperating with the dog being smacked on the nose with a pea. How can the authorities enforce mask-wearing if millions in a country refuse? What if the 74 million Pushbackers that voted for Trump in 2020 refused to wear masks, close their businesses or stay in their homes. It would be unenforceable. The

few control the many through the compliance of the many and that's always been the dynamic be it 'Covid' regulations or the Roman Empire. I know people can find it intimidating to say no to authority or stand out in a crowd for being the only one with a face on display; but it has to be done or it's over. I hope I've made clear in this book that where this is going will be far more intimidating than standing up now and saying 'No' – I will not cooperate with my own enslavement and that of my children. There might be consequences for some initially, although not so if enough do the same. The question that must be addressed is what is going to happen if we don't? It is time to be strong and unyieldingly so. No means no. Not here and there, but *everywhere* and *always*. I have refused to wear a mask and obey all the other nonsense. I will not comply with tyranny. I repeat: Fascism is not imposed by fascists – there are never enough of them. Fascism is imposed by the population acquiescing to fascism. *I will not do it.* I will die first, or my body will. Living meekly under fascism is a form of death anyway, the death of the spirit that Martin Luther King described.

Making things happen

We must not despair. This is not over till it's over and it's far from that. The 'fat lady' must refuse to sing. The longer the 'Covid' hoax has dragged on and impacted on more lives we have seen an awakening of phenomenal numbers of people worldwide to the realisation that what they have believed all their lives is not how the world really is. Research published by the system-serving University of Bristol and King's College London in February, 2021, concluded: 'One in every 11 people in Britain say they trust David Icke's take on the coronavirus pandemic.' It will be more by now and we have gathering numbers to build on. We must urgently progress from seeing the scam to ceasing to cooperate with it. Prominent German lawyer Reiner Fuellmich, also licenced to practice law in America, is doing a magnificent job taking the legal route to bring the psychopaths to justice through a second Nuremberg tribunal for crimes against humanity. Fuellmich has an impressive record of

beating the elite in court and he formed the German Corona Investigative Committee to pursue civil charges against the main perpetrators with a view to triggering criminal charges. Most importantly he has grasped the foundation of the hoax – the PCR test not testing for the ‘virus’ – and Christian Drosten is therefore on his charge sheet along with Gates frontman Tedros at the World Health Organization. Major players must be not be allowed to inflict their horrors on the human race without being brought to book. A life sentence must follow for Bill Gates and the rest of them. A group of researchers has also indicted the government of Norway for crimes against humanity with copies sent to the police and the International Criminal Court. The lawsuit cites participation in an internationally-planned false pandemic and violation of international law and human rights, the European Commission’s definition of human rights by coercive rules, Nuremberg and Hague rules on fundamental human rights, and the Norwegian constitution. We must take the initiative from hereon and not just complain, protest and react.

There are practical ways to support vital mass non-cooperation. Organising in numbers is one. Lockdown marches in London in the spring in 2021 were mass non-cooperation that the authorities could not stop. There were too many people. Hundreds of thousands walked the London streets in the centre of the road for mile after mile while the Face-Nappies could only look on. They were determined, but calm, and just *did it* with no histrionics and lots of smiles. The police were impotent. Others are organising group shopping without masks for mutual support and imagine if that was happening all over. Policing it would be impossible. If the store refuses to serve people in these circumstances they would be faced with a long line of trolleys full of goods standing on their own and everything would have to be returned to the shelves. How would they cope with that if it kept happening? I am talking here about moving on from complaining to being pro-active; from watching things happen to making things happen. I include in this our relationship with the police. The behaviour of many Face-Nappies

has been disgraceful and anyone who thinks they would never find concentration camp guards in the ‘enlightened’ modern era have had that myth busted big-time. The period and setting may change – Wetikos never do. I watched film footage from a London march in which a police thug viciously kicked a protestor on the floor who had done nothing. His fellow Face-Nappies stood in a ring protecting him. What he did was a criminal assault and with a crowd far outnumbering the police this can no longer be allowed to happen unchallenged. I get it when people chant ‘shame on you’ in these circumstances, but that is no longer enough. They *have* no shame those who do this. Crowds needs to start making a citizen’s arrest of the police who commit criminal offences and brutally attack innocent people and defenceless women. A citizen’s arrest can be made under section 24A of the UK Police and Criminal Evidence (PACE) Act of 1984 and you will find something similar in other countries. I prefer to call it a Common Law arrest rather than citizen’s for reasons I will come to shortly. Anyone can arrest a person committing an indictable offence or if they have reasonable grounds to suspect they are committing an indictable offence. On both counts the attack by the police thug would have fallen into this category. A citizen’s arrest can be made to stop someone:

- Causing physical injury to himself or any other person
- Suffering physical injury
- Causing loss of or damage to property
- Making off before a constable can assume responsibility for him

A citizen’s arrest may also be made to prevent a breach of the peace under Common Law and if they believe a breach of the peace will happen or anything related to harm likely to be done or already done in their presence. This is the way to go I think – the Common Law version. If police know that the crowd and members of the public will no longer be standing and watching while they commit

their thuggery and crimes they will think twice about acting like Brownshirts and Blackshirts.

Common Law – common sense

Mention of Common Law is very important. Most people think the law is the law as in one law. This is not the case. There are two bodies of law, Common Law and Statute Law, and they are not the same. Common Law is founded on the simple premise of do no harm. It does not recognise victimless crimes in which no harm is done while Statute Law does. There is a Statute Law against almost everything. So what is Statute Law? Amazingly it's the law of the sea that was brought ashore by the Cult to override the law of the land which is Common Law. They had no right to do this and as always they did it anyway. They had to. They could not impose their will on the people through Common Law which only applies to do no harm. How could you stitch up the fine detail of people's lives with that? Instead they took the law of the sea, or Admiralty Law, and applied it to the population. Statute Law refers to all the laws spewing out of governments and their agencies including all the fascist laws and regulations relating to 'Covid'. The key point to make is that Statute Law is *contract law*. It only applies between *contracting* corporations. Most police officers don't even know this. They have to be kept in the dark, too. Long ago when merchants and their sailing ships began to trade with different countries a contractual law was developed called Admiralty Law and other names. Again it only applied to *contracts* agreed between *corporate* entities. If there is no agreed contract the law of the sea had no jurisdiction *and that still applies to its new alias of Statute Law*. The problem for the Cult when the law of the sea was brought ashore was an obvious one. People were not corporations and neither were government entities. To overcome the latter they made governments and all associated organisations corporations. All the institutions are *private corporations* and I mean governments and their agencies, local councils, police, courts, military, US states, the whole lot. Go to the

Dun and Bradstreet corporate listings website for confirmation that they are all corporations. You are arrested by a private corporation called the police by someone who is really a private security guard and they take you to court which is another private corporation.

Neither have jurisdiction over you unless you consent and *contract* with them. This is why you hear the mantra about law enforcement policing by *consent* of the people. In truth the people 'consent' only in theory through monumental trickery.

Okay, the Cult overcame the corporate law problem by making governments and institutions corporate entities; but what about people? They are not corporations are they? Ah ... well in a sense, and *only* a sense, they are. Not people exactly – the illusion of people. The Cult creates a corporation in the name of everyone at the time that their birth certificate is issued. Note birth/ *berth* certificate and when you go to court under the law of the sea on land you stand in a *dock*. These are throwbacks to the origin. My Common Law name is David Vaughan Icke. The name of the corporation created by the government when I was born is called Mr David Vaughan Icke usually written in capitals as MR DAVID VAUGHAN ICKE. That is not me, the living, breathing man. It is a fictitious corporate entity. The trick is to make you think that David Vaughan Icke and MR DAVID VAUGHAN ICKE are the same thing. *They are not*. When police charge you and take you to court they are prosecuting the corporate entity and not the living, breathing, man or woman. They have to trick you into identifying as the corporate entity and contracting with them. Otherwise they have no jurisdiction. They do this through a language known as legalese. Lawful and legal are not the same either. Lawful relates to Common Law and legal relates to Statute Law. Legalese is the language of Statue Law which uses terms that mean one thing to the public and another in legalese. Notice that when a police officer tells someone why they are being charged he or she will say at the end: 'Do you understand?' To the public that means 'Do you comprehend?' In legalese it means 'Do you stand under me?' Do you stand under my authority? If you say

yes to the question you are unknowingly agreeing to give them jurisdiction over you in a contract between two corporate entities.

This is a confidence trick in every way. Contracts have to be agreed between informed parties and if you don't know that David Vaughan Icke is agreeing to be the corporation MR DAVID VAUGHAN ICKE you cannot knowingly agree to contract. They are deceiving you and another way they do this is to ask for proof of identity. You usually show them a driving licence or other document on which your corporate name is written. In doing so you are accepting that you are that corporate entity when you are not. Referring to yourself as a 'person' or 'citizen' is also identifying with your corporate fiction which is why I made the Common Law point about the citizen's arrest. If you are approached by a police officer you identify yourself immediately as a living, breathing, man or woman and say 'I do not consent, I do not contract with you and I do not understand' or stand under their authority. I have a Common Law birth certificate as a living man and these are available at no charge from commonlawcourt.com. Businesses registered under the Statute Law system means that its laws apply. There are, however, ways to run a business under Common Law. Remember all 'Covid' laws and regulations are Statute Law – the law of *contracts* and you do not have to contract. This doesn't mean that you can kill someone and get away with it. Common Law says do no harm and that applies to physical harm, financial harm etc. Police are employees of private corporations and there needs to be a new system of non-corporate Common Law constables operating outside the Statute Law system. If you go to davidicke.com and put Common Law into the search engine you will find videos that explain Common Law in much greater detail. It is definitely a road we should walk.

With all my heart

I have heard people say that we are in a spiritual war. I don't like the term 'war' with its Wetiko dynamic, but I know what they mean. Sweep aside all the bodily forms and we are in a situation in which two states of consciousness are seeking very different realities.

Wetiko wants upheaval, chaos, fear, suffering, conflict and control. The other wants love, peace, harmony, fairness and freedom. That's where we are. We should not fall for the idea that Wetiko is all-powerful and there's nothing we can do. Wetiko is not all-powerful. It's a joke, pathetic. It doesn't have to be, but it has made that choice for now. A handful of times over the years when I have felt the presence of its frequency I have allowed it to attach briefly so I could consciously observe its nature. The experience is not pleasant, the energy is heavy and dark, but the ease with which you can kick it back out the door shows that its real power is in persuading us that it has power. It's all a con. Wetiko is a con. It's a trickster and not a power that can control us if we unleash our own. The con is founded on manipulating humanity to give its power to Wetiko which recycles it back to present the illusion that it has power when its power is *ours* that we gave away. This happens on an energetic level and plays out in the world of the seen as humanity giving its power to Wetiko authority which uses that power to control the population when the power is only the power the population has handed over. How could it be any other way for billions to be controlled by a relative few? I have had experiences with people possessed by Wetiko and again you can kick its arse if you do it with an open heart. Oh yes – the *heart* which can transform the world of perceived 'matter'.

We are receiver-transmitters and processors of information, but what information and where from? Information is processed into perception in three main areas – the brain, the heart and the belly. These relate to thinking, knowing, and emotion. Wetiko wants us to be head and belly people which means we think within the confines of the Matrix simulation and low-vibrational emotional reaction scrambles balance and perception. A few minutes on social media and you see how emotion is the dominant force. Woke is all emotion and is therefore thought-free and fact-free. Our heart is something different. It *knows* while the head *thinks* and has to try to work it out because it doesn't know. The human energy field has seven prime vortexes which connect us with wider reality ([Fig 23](#)). Chakra means

'wheels of light' in the Sanskrit language of ancient India. The main ones are: The crown chakra on top of the head; brow (or 'third eye') chakra in the centre of the forehead; throat chakra; heart chakra in the centre of the chest; solar plexus chakra below the sternum; sacral chakra beneath the navel; and base chakra at the bottom of the spine. Each one has a particular function or functions. We feel anxiety and nervousness in the belly where the sacral chakra is located and this processes emotion that can affect the colon to give people 'the shits' or make them 'shit scared' when they are nervous. Chakras all play an important role, but the Mr and Mrs Big is the heart chakra which sits at the centre of the seven, above the chakras that connect us to the 'physical' and below those that connect with higher realms (or at least should). Here in the heart chakra we feel love, empathy and compassion – 'My heart goes out to you'. Those with closed hearts become literally 'heart-less' in their attitudes and behaviour (see Bill Gates). Native Americans portrayed Wetiko with what Paul Levy calls a 'frigid, icy heart, devoid of mercy' (see Bill Gates).

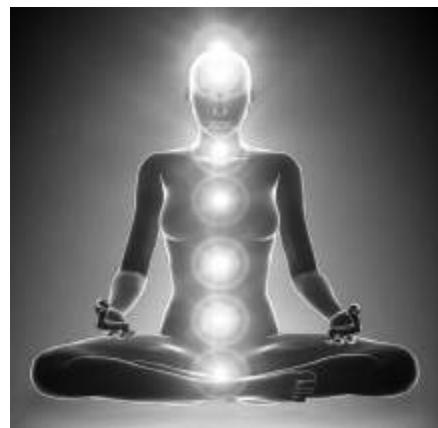


Figure 23: The chakra system which interpenetrates the human energy field. The heart chakra is the governor – or should be.

Wetiko trembles at the thought of heart energy which it cannot infiltrate. The frequency is too high. What it seeks to do instead is close the heart chakra vortex to block its perceptual and energetic influence. Psychopaths have 'hearts of stone' and emotionally-damaged people have 'heartache' and 'broken hearts'. The astonishing amount of heart disease is related to heart chakra

disruption with its fundamental connection to the ‘physical’ heart. Dr Tom Cowan has written an outstanding book challenging the belief that the heart is a pump and making the connection between the ‘physical’ and spiritual heart. Rudolph Steiner who was way ahead of his time said the same about the fallacy that the heart is a pump. *What?* The heart is not a pump? That’s crazy, right? Everybody knows that. Read Cowan’s *Human Heart, Cosmic Heart* and you will realise that the very idea of the heart as a pump is ridiculous when you see the evidence. How does blood in the feet so far from the heart get pumped horizontally up the body by the heart?? Cowan explains in the book the real reason why blood moves as it does. Our ‘physical’ heart is used to symbolise love when the source is really the heart vortex or spiritual heart which is our most powerful energetic connection to ‘out there’ expanded consciousness. That’s why we feel *knowing* – intuitive knowing – in the centre of the chest. Knowing doesn’t come from a process of thoughts leading to a conclusion. It is there in an instant all in one go. Our heart knows because of its connection to levels of awareness that *do* know. This is the meaning and source of intuition – intuitive *knowing*.

For the last more than 30 years of uncovering the global game and the nature of reality my heart has been my constant antenna for truth and accuracy. An American intelligence insider once said that I had quoted a disinformor in one of my books and yet I had only quoted the part that was true. He asked: ‘How do you do that?’ By using my heart antenna was the answer and anyone can do it. Heart-centred is how we are meant to be. With a closed heart chakra we withdraw into a closed mind and the bubble of five-sense reality. If you take a moment to focus your attention on the centre of your chest, picture a spinning wheel of light and see it opening and expanding. You will feel it happening, too, and perceptions of the heart like joy and love as the heart impacts on the mind as they interact. The more the chakra opens the more you will feel expressions of heart consciousness and as the process continues, and becomes part of you, insights and knowings will follow. An open

heart is connected to that level of awareness that knows all is *One*. You will see from its perspective that the fault-lines that divide us are only illusions to control us. An open heart does not process the illusions of race, creed and sexuality except as brief experiences for a consciousness that is all. Our heart does not see division, only unity (Figs 24 and 25). There's something else, too. Our hearts love to laugh. Mark Twain's quote that says 'The human race has one really effective weapon, and that is laughter' is really a reference to the heart which loves to laugh with the joy of knowing the true nature of infinite reality and that all the madness of human society is an illusion of the mind. Twain also said: 'Against the assault of laughter nothing can stand.' This is so true of Wetiko and the Cult. Their insecurity demands that they be taken seriously and their power and authority acknowledged and feared. We should do nothing of the sort. We should not get aggressive or fearful which their insecurity so desires. We should laugh in their face. Even in their no-face as police come over in their face-nappies and expect to be taken seriously. They don't take themselves seriously looking like that so why should we? Laugh in the face of intimidation. Laugh in the face of tyranny. You will see by its reaction that you have pressed all of its buttons. Wetiko does not know what to do in the face of laughter or when its targets refuse to concede their joy to fear. We have seen many examples during the 'Covid' hoax when people have expressed their energetic power and the string puppets of Wetiko retreat with their tail limp between their knees. Laugh – the world is bloody mad after all and if it's a choice between laughter and tears I know which way I'm going.



Figure 24: Head consciousness without the heart sees division and everything apart from everything else.



Figure 25: Heart consciousness sees everything as One.

Vaccines' and the soul

The foundation of Wetiko/Archon control of humans is the separation of incarnate five-sense mind from the infinite 'I' and closing the heart chakra where the True 'I' lives during a human life. The goal has been to achieve complete separation in both cases. I was interested therefore to read an account by a French energetic healer of what she said she experienced with a patient who had been given the 'Covid' vaccine. Genuine energy healers can sense information and consciousness fields at different levels of being which are referred to as 'subtle bodies'. She described treating the patient who later returned after having, without the healer's knowledge, two doses of the 'Covid vaccine'. The healer said:

I noticed immediately the change, very heavy energy emanating from [the] subtle bodies. The scariest thing was when I was working on the heart chakra, I connected with her soul: it was detached from the physical body, it had no contact and it was, as if it was floating in a state of total confusion: a damage to the consciousness that loses contact with the physical body, i.e. with our biological machine, there is no longer any communication between them.

I continued the treatment by sending light to the heart chakra, the soul of the person, but it seemed that the soul could no longer receive any light, frequency or energy. It was a very powerful experience for me. Then I understood that this substance is indeed used to detach consciousness so that this consciousness can no longer interact through this body that it possesses in life, where there is no longer any contact, no frequency, no light, no more energetic balance or mind.

This would create a human that is rudderless and at the extreme almost zombie-like operating with a fractional state of consciousness at the mercy of Wetiko. I was especially intrigued by what the healer said in the light of the prediction by the highly-informed Rudolf Steiner more than a hundred years ago. He said:

In the future, we will eliminate the soul with medicine. Under the pretext of a 'healthy point of view', there will be a vaccine by which the human body will be treated as soon as possible directly at birth, so that the human being cannot develop the thought of the existence of soul and Spirit. To materialistic doctors will be entrusted the task of removing the soul of humanity.

As today, people are vaccinated against this disease or that disease, so in the future, children will be vaccinated with a substance that can be produced precisely in such a way that people, thanks to this vaccination, will be immune to being subjected to the 'madness' of spiritual life. He would be extremely smart, but he would not develop a conscience, and that is the true goal of some materialistic circles.

Steiner said the vaccine would detach the physical body from the etheric body (subtle bodies) and 'once the etheric body is detached the relationship between the universe and the etheric body would become extremely unstable, and man would become an automaton'. He said 'the physical body of man must be polished on this Earth by spiritual will – so the vaccine becomes a kind of arymanique (Wetiko) force' and 'man can no longer get rid of a given materialistic feeling'. Humans would then, he said, become 'materialistic of constitution and can no longer rise to the spiritual'. I have been writing for years about DNA being a receiver-transmitter of information that connects us to other levels of reality and these 'vaccines' changing DNA can be likened to changing an antenna and what it can transmit and receive. Such a disconnection would clearly lead to changes in personality and perception. Steiner further predicted the arrival of AI. Big Pharma 'Covid vaccine' makers, expressions of Wetiko, are testing their DNA-manipulating evil on children as I write with a view to giving the 'vaccine' to babies. If it's a soul-body disconnecter – and I say that it is or can be – every child would be disconnected from 'soul' at birth and the 'vaccine' would create a closed system in which spiritual guidance from the greater self would play no part. This has been the ambition of Wetiko all

along. A Pentagon video from 2005 was leaked of a presentation explaining the development of vaccines to change behaviour by their effect on the brain. Those that believe this is not happening with the ‘Covid’ genetically-modifying procedure masquerading as a ‘vaccine’ should make an urgent appointment with Naivety Anonymous. Klaus Schwab wrote in 2018:

Neurotechnologies enable us to better influence consciousness and thought and to understand many activities of the brain. They include decoding what we are thinking in fine levels of detail through new chemicals and interventions that can influence our brains to correct for errors or enhance functionality.

The plan is clear and only the heart can stop it. With every heart that opens, every mind that awakens, Wetiko is weakened. Heart and love are far more powerful than head and hate and so nothing like a majority is needed to turn this around.

Beyond the Phantom

Our heart is the prime target of Wetiko and so it must be the answer to Wetiko. We *are* our heart which is part of one heart, the infinite heart. Our heart is where the true self lives in a human life behind firewalls of five-sense illusion when an imposter takes its place – *Phantom Self*; but our heart waits patiently to be set free any time we choose to see beyond the Phantom, beyond Wetiko. A Wetikoed Phantom Self can wreak mass death and destruction while the love of forever is locked away in its heart. The time is here to unleash its power and let it sweep away the fear and despair that is Wetiko. Heart consciousness does not seek manipulated, censored, advantage for its belief or religion, its activism and desires. As an expression of the One it treats all as One with the same rights to freedom and opinion. Our heart demands fairness for itself no more than for others. From this unity of heart we can come together in mutual support and transform this Wetikoed world into what reality is meant to be – a place of love, joy, happiness, fairness, justice and freedom. Wetiko has another agenda and that’s why the world is as

it is, but enough of this nonsense. Wetiko can't stay where hearts are open and it works so hard to keep them closed. Fear is its currency and its food source and love in its true sense has no fear. Why would love have fear when it knows it is *All That Is, Has Been, And Ever Can Be* on an eternal exploration of all possibility? Love in this true sense is not the physical attraction that passes for love. This can be an expression of it, yes, but Infinite Love, a love without condition, goes far deeper to the core of all being. It is the core of all being. Infinite reality was born from love beyond the illusions of the simulation. Love infinitely expressed is the knowing that all is One and the swiftly-passing experience of separation is a temporary hallucination. You cannot disconnect from Oneness; you can only perceive that you have and withdraw from its influence. This is the most important of all perception trickery by the mind parasite that is Wetiko and the foundation of all its potential for manipulation.

If we open our hearts, open the sluice gates of the mind, and redefine self-identity amazing things start to happen. Consciousness expands or contracts in accordance with self-identity. When true self is recognised as infinite awareness and label self – Phantom Self – is seen as only a series of brief experiences life is transformed. Consciousness expands to the extent that self-identity expands and everything changes. You see unity, not division, the picture, not the pixels. From this we can play the long game. No more is an experience something in and of itself, but a fleeting moment in the eternity of forever. Suddenly people in uniform and dark suits are no longer intimidating. Doing what your heart knows to be right is no longer intimidating and consequences for those actions take on the same nature of a brief experience that passes in the blink of an infinite eye. Intimidation is all in the mind. Beyond the mind there is no intimidation.

An open heart does not consider consequences for what it knows to be right. To do so would be to consider not doing what it knows to be right and for a heart in its power that is never an option. The Renegade Mind is really the Renegade Heart. Consideration of consequences will always provide a getaway car for the mind and

the heart doesn't want one. What is right in the light of what we face today is to stop cooperating with Wetiko in all its forms and to do it without fear or compromise. You cannot compromise with tyranny when tyranny always demands more until it has everything. Life is your perception and you are your destiny. Change your perception and you change your life. Change collective perception and we change the world.

*Come on people ... One human family, One heart, One goal ...
FREEEEEDOM!*

We must settle for nothing less.

Postscript

The big scare story as the book goes to press is the ‘Indian’ variant and the world is being deluged with propaganda about the ‘Covid catastrophe’ in India which mirrors in its lies and misrepresentations what happened in Italy before the first lockdown in 2020.

The *New York Post* published a picture of someone who had ‘collapsed in the street from Covid’ in India in April, 2021, which was actually taken during a gas leak in May, 2020. Same old, same old. Media articles in mid-February were asking why India had been so untouched by ‘Covid’ and then as their vaccine rollout gathered pace the alleged ‘cases’ began to rapidly increase. Indian ‘Covid vaccine’ maker Bharat Biotech was funded into existence by the Bill and Melinda Gates Foundation (the pair announced their divorce in May, 2021, which is a pity because they so deserve each other). The Indian ‘Covid crisis’ was ramped up by the media to terrify the world and prepare people for submission to still more restrictions. The scam that worked the first time was being repeated only with far more people seeing through the deceit. Davidicke.com and Ickonic.com have sought to tell the true story of what is happening by talking to people living through the Indian nightmare which has nothing to do with ‘Covid’. We posted a letter from ‘Alisha’ in Pune who told a very different story to government and media mendacity. She said scenes of dying people and overwhelmed hospitals were designed to hide what was really happening – genocide and starvation. Alisha said that millions had already died of starvation during the ongoing lockdowns while government and media were lying and making it look like the ‘virus’:

Restaurants, shops, gyms, theatres, basically everything is shut. The cities are ghost towns. Even so-called 'essential' businesses are only open till 11am in the morning. You basically have just an hour to buy food and then your time is up.

Inter-state travel and even inter-district travel is banned. The cops wait at all major crossroads to question why you are traveling outdoors or to fine you if you are not wearing a mask.

The medical community here is also complicit in genocide, lying about hospitals being full and turning away people with genuine illnesses, who need immediate care. They have even created a shortage of oxygen cylinders.

This is the classic Cult modus operandi played out in every country. Alisha said that people who would not have a PCR test not testing for the 'virus' were being denied hospital treatment. She said the people hit hardest were migrant workers and those in rural areas. Most businesses employed migrant workers and with everything closed there were no jobs, no income and no food. As a result millions were dying of starvation or malnutrition. All this was happening under Prime Minister Narendra Modi, a 100-percent asset of the Cult, and it emphasises yet again the scale of pure anti-human evil we are dealing with. Australia banned its people from returning home from India with penalties for trying to do so of up to five years in jail and a fine of £37,000. The manufactured 'Covid' crisis in India was being prepared to justify further fascism in the West. Obvious connections could be seen between the Indian 'vaccine' programme and increased 'cases' and this became a common theme. The Seychelles, the most per capita 'Covid vaccinated' population in the world, went back into lockdown after a 'surge of cases'.

Long ago the truly evil Monsanto agricultural biotechnology corporation with its big connections to Bill Gates devastated Indian farming with genetically-modified crops. Human rights activist Gurcharan Singh highlighted the efforts by the Indian government to complete the job by destroying the food supply to hundreds of millions with 'Covid' lockdowns. He said that 415 million people at the bottom of the disgusting caste system (still going whatever they say) were below the poverty line and struggled to feed themselves every year. Now the government was imposing lockdown at just the

time to destroy the harvest. This deliberate policy was leading to mass starvation. People may reel back at the suggestion that a government would do that, but Wetiko-controlled ‘leaders’ are capable of any level of evil. In fact what is described in India is in the process of being instigated worldwide. The food chain and food supply are being targeted at every level to cause world hunger and thus control. Bill Gates is not the biggest owner of farmland in America for no reason and destroying access to food aids both the depopulation agenda and the plan for synthetic ‘food’ already being funded into existence by Gates. Add to this the coming hyper-inflation from the suicidal creation of fake ‘money’ in response to ‘Covid’ and the breakdown of container shipping systems and you have a cocktail that can only lead one way and is meant to. The Cult plan is to crash the entire system to ‘build back better’ with the Great Reset.

'Vaccine' transmission

Reports from all over the world continue to emerge of women suffering menstrual and fertility problems after having the fake ‘vaccine’ and of the non-‘vaccinated’ having similar problems when interacting with the ‘vaccinated’. There are far too many for ‘coincidence’ to be credible. We’ve had menopausal women getting periods, others having periods stop or not stopping for weeks, passing clots, sometimes the lining of the uterus, breast irregularities, and miscarriages (which increased by 400 percent in parts of the United States). Non-‘vaccinated’ men and children have suffered blood clots and nose bleeding after interaction with the ‘vaccinated’. Babies have died from the effects of breast milk from a ‘vaccinated’ mother. Awake doctors – the small minority – speculated on the cause of non-‘vaccinated’ suffering the same effects as the ‘vaccinated’. Was it nanotechnology in the synthetic substance transmitting frequencies or was it a straight chemical bioweapon that was being transmitted between people? I am not saying that some kind of chemical transmission is not one possible answer, but the foundation of all that the Cult does is frequency and

this is fertile ground for understanding how transmission can happen. American doctor Carrie Madej, an internal medicine physician and osteopath, has been practicing for the last 20 years, teaching medical students, and she says attending different meetings where the agenda for humanity was discussed. Madej, who operates out of Georgia, did not dismiss other possible forms of transmission, but she focused on frequency in search of an explanation for transmission. She said the Moderna and Pfizer 'vaccines' contained nano-lipid particles as a key component. This was a brand new technology never before used on humanity. 'They're using a nanotechnology which is pretty much little tiny computer bits ... nanobots or hydrogel.' Inside the 'vaccines' was 'this sci-fi kind of substance' which suppressed immune checkpoints to get into the cell. I referred to this earlier as the 'Trojan horse' technique that tricks the cell into opening a gateway for the self-replicating synthetic material and while the immune system is artificially suppressed the body has no defences. Madej said the substance served many purposes including an on-demand ability to 'deliver the payload' and using the nano 'computer bits' as biosensors in the body. 'It actually has the ability to accumulate data from your body, like your breathing, your respiration, thoughts, emotions, all kinds of things.'

She said the technology obviously has the ability to operate through Wi-Fi and transmit and receive energy, messages, frequencies or impulses. 'Just imagine you're getting this new substance in you and it can react to things all around you, the 5G, your smart device, your phones.' We had something completely foreign in the human body that had never been launched large scale at a time when we were seeing 5G going into schools and hospitals (plus the Musk satellites) and she believed the 'vaccine' transmission had something to do with this: '... if these people have this inside of them ... it can act like an antenna and actually transmit it outwardly as well.' The synthetic substance produced its own voltage and so it could have that kind of effect. This fits with my own contention that the nano receiver-transmitters are designed to connect people to the

Smart Grid and break the receiver-transmitter connection to expanded consciousness. That would explain the French energy healer's experience of the disconnection of body from 'soul' with those who have had the 'vaccine'. The nanobots, self-replicating inside the body, would also transmit the synthetic frequency which could be picked up through close interaction by those who have not been 'vaccinated'. Madej speculated that perhaps it was 5G and increased levels of other radiation that was causing the symptoms directly although interestingly she said that non-'vaccinated' patients had shown improvement when they were away from the 'vaccinated' person they had interacted with. It must be remembered that you can control frequency and energy with your mind and you can consciously create energetic barriers or bubbles with the mind to stop damaging frequencies from penetrating your field. American paediatrician Dr Larry Palevsky said the 'vaccine' was not a 'vaccine' and was never designed to protect from a 'viral' infection. He called it 'a massive, brilliant propaganda of genocide' because they didn't have to inject everyone to get the result they wanted. He said the content of the jabs was able to infuse any material into the brain, heart, lungs, kidneys, liver, sperm and female productive system. 'This is genocide; this is a weapon of mass destruction.' At the same time American colleges were banning students from attending if they didn't have this life-changing and potentially life-ending 'vaccine'. Class action lawsuits must follow when the consequences of this college fascism come to light. As the book was going to press came reports about fertility effects on sperm in 'vaccinated' men which would absolutely fit with what I have been saying and hospitals continued to fill with 'vaccine' reactions. Another question is what about transmission via blood transfusions? The NHS has extended blood donation restrictions from seven days after a 'Covid vaccination' to 28 days after even a sore arm reaction.

I said in the spring of 2020 that the then touted 'Covid vaccine' would be ongoing each year like the flu jab. A year later Pfizer CEO, the appalling Albert Bourla, said people would 'likely' need a 'booster dose' of the 'vaccine' within 12 months of getting 'fully

'vaccinated' and then a yearly shot. 'Variants will play a key role', he said confirming the point. Johnson & Johnson CEO Alex Gorsky also took time out from his 'vaccine' disaster to say that people may need to be vaccinated against 'Covid-19' each year. UK Health Secretary, the psychopath Matt Hancock, said additional 'boosters' would be available in the autumn of 2021. This is the trap of the 'vaccine passport'. The public will have to accept every last 'vaccine' they introduce, including for the fake 'variants', or it would cease to be valid. The only other way in some cases would be continuous testing with a test not testing for the 'virus' and what is on the swabs constantly pushed up your noise towards the brain every time?

'Vaccines' changing behaviour

I mentioned in the body of the book how I believed we would see gathering behaviour changes in the 'vaccinated' and I am already hearing such comments from the non-'vaccinated' describing behaviour changes in friends, loved ones and work colleagues. This will only increase as the self-replicating synthetic material and nanoparticles expand in body and brain. An article in the *Guardian* in 2016 detailed research at the University of Virginia in Charlottesville which developed a new method for controlling brain circuits associated with complex animal behaviour. The method, dubbed 'magnetogenetics', involves genetically-engineering a protein called ferritin, which stores and releases iron, to create a magnetised substance – 'Magneto' – that can activate specific groups of nerve cells from a distance. This is claimed to be an advance on other methods of brain activity manipulation known as optogenetics and chemogenetics (the Cult has been developing methods of brain control for a long time). The ferritin technique is said to be non-invasive and able to activate neurons 'rapidly and reversibly'. In other words, human thought and perception. The article said that earlier studies revealed how nerve cell proteins 'activated by heat and mechanical pressure can be genetically engineered so that they become sensitive to radio waves and magnetic fields, by attaching them to an iron-storing protein called ferritin, or to inorganic

paramagnetic particles'. Sensitive to radio waves and magnetic fields? You mean like 5G, 6G and 7G? This is the human-AI Smart Grid hive mind we are talking about. The *Guardian* article said:

... the researchers injected Magneto into the striatum of freely behaving mice, a deep brain structure containing dopamine-producing neurons that are involved in reward and motivation, and then placed the animals into an apparatus split into magnetised and non-magnetised sections.

Mice expressing Magneto spent far more time in the magnetised areas than mice that did not, because activation of the protein caused the striatal neurons expressing it to release dopamine, so that the mice found being in those areas rewarding. This shows that Magneto can remotely control the firing of neurons deep within the brain, and also control complex behaviours.

Make no mistake this basic methodology will be part of the 'Covid vaccine' cocktail and using magnetics to change brain function through electromagnetic field frequency activation. The Pentagon is developing a 'Covid vaccine' using ferritin. Magnetics would explain changes in behaviour and why videos are appearing across the Internet as I write showing how magnets stick to the skin at the point of the 'vaccine' shot. Once people take these 'vaccines' anything becomes possible in terms of brain function and illness which will be blamed on 'Covid-19' and 'variants'. Magnetic field manipulation would further explain why the non-'vaccinated' are reporting the same symptoms as the 'vaccinated' they interact with and why those symptoms are reported to decrease when not in their company. Interestingly 'Magneto', a 'mutant', is a character in the Marvel Comic *X-Men* stories with the ability to manipulate magnetic fields and he believes that mutants should fight back against their human oppressors by any means necessary. The character was born Erik Lehnsherr to a Jewish family in Germany.

Cult-controlled courts

The European Court of Human Rights opened the door for mandatory 'Covid-19 vaccines' across the continent when it ruled in a Czech Republic dispute over childhood immunisation that legally

enforced vaccination could be ‘necessary in a democratic society’. The 17 judges decided that compulsory vaccinations did not breach human rights law. On the face of it the judgement was so inverted you gasp for air. If not having a vaccine infused into your body is not a human right then what is? Ah, but they said human rights law which has been specifically written to delete all human rights at the behest of the state (the Cult). Article 8 of the European Convention on Human Rights relates to the right to a private life. The crucial word here is ‘*except*’:

There shall be no interference by a public authority with the exercise of this right EXCEPT such as is in accordance with the law and is necessary in a democratic society in the interests of national security, public safety or the economic wellbeing of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others [My emphasis].

No interference *except* in accordance with the law means there *are* no ‘human rights’ *except* what EU governments decide you can have at their behest. ‘As is necessary in a democratic society’ explains that reference in the judgement and ‘in the interests of national security, public safety or the economic well-being of the country, for the prevention of disorder or crime, for the protection of health or morals, or for the protection of the rights and freedoms of others’ gives the EU a coach and horses to ride through ‘human rights’ and scatter them in all directions. The judiciary is not a check and balance on government extremism; it is a vehicle to enforce it. This judgement was almost laughably predictable when the last thing the Cult wanted was a decision that went against mandatory vaccination. Judges rule over and over again to benefit the system of which they are a part. Vaccination disputes that come before them are invariably delivered in favour of doctors and authorities representing the view of the state which owns the judiciary. Oh, yes, and we have even had calls to stop putting ‘Covid-19’ on death certificates within 28 days of a ‘positive test’ because it is claimed the practice makes the ‘vaccine’ appear not to work. They are laughing at you.

The scale of madness, inhumanity and things to come was highlighted when those not ‘vaccinated’ for ‘Covid’ were refused evacuation from the Caribbean island of St Vincent during massive volcanic eruptions. Cruise ships taking residents to the safety of another island allowed only the ‘vaccinated’ to board and the rest were left to their fate. Even in life and death situations like this we see ‘Covid’ stripping people of their most basic human instincts and the insanity is even more extreme when you think that fake ‘vaccine’-makers are not even claiming their body-manipulating concoctions stop ‘infection’ and ‘transmission’ of a ‘virus’ that doesn’t exist. St Vincent Prime Minister Ralph Gonsalves said: ‘The chief medical officer will be identifying the persons already vaccinated so that we can get them on the ship.’ Note again the power of the chief medical officer who, like Whitty in the UK, will be answering to the World Health Organization. This is the Cult network structure that has overridden politicians who ‘follow the science’ which means doing what WHO-controlled ‘medical officers’ and ‘science advisers’ tell them. Gonsalves even said that residents who were ‘vaccinated’ after the order so they could board the ships would still be refused entry due to possible side effects such as ‘wooziness in the head’. The good news is that if they were woozy enough in the head they could qualify to be prime minister of St Vincent.

Microchipping freedom

The European judgement will be used at some point to justify moves to enforce the ‘Covid’ DNA-manipulating procedure. Sandra Ro, CEO of the Global Blockchain Business Council, told a World Economic Forum event that she hoped ‘vaccine passports’ would help to ‘drive forced consent and standardisation’ of global digital identity schemes: ‘I’m hoping with the desire and global demand for some sort of vaccine passport – so that people can get travelling and working again – [it] will drive forced consent, standardisation, and frankly, cooperation across the world.’ The lady is either not very bright, or thoroughly mendacious, to use the term ‘forced consent’.

You do not ‘consent’ if you are forced – you *submit*. She was describing what the plan has been all along and that’s to enforce a digital identity on every human without which they could not function. ‘Vaccine passports’ are opening the door and are far from the end goal. A digital identity would allow you to be tracked in everything you do in cyberspace and this is the same technique used by Cult-owned China to enforce its social credit system of total control. The ultimate ‘passport’ is planned to be a microchip as my books have warned for nearly 30 years. Those nice people at the Pentagon working for the Cult-controlled Defense Advanced Research Projects Agency (DARPA) claimed in April, 2021, they have developed a microchip inserted under the skin to detect ‘asymptomatic Covid-19 infection’ before it becomes an outbreak and a ‘revolutionary filter’ that can remove the ‘virus’ from the blood when attached to a dialysis machine. The only problems with this are that the ‘virus’ does not exist and people transmitting the ‘virus’ with no symptoms is brain-numbing bullshit. This is, of course, not a ruse to get people to be microchipped for very different reasons. DARPA also said it was producing a one-stop ‘vaccine’ for the ‘virus’ and all ‘variants’. One of the most sinister organisations on Planet Earth is doing this? Better have it then. These people are insane because Wetiko that possesses them is insane.

Researchers from the Salk Institute in California announced they have created an embryo that is part human and part monkey. My books going back to the 1990s have exposed experiments in top secret underground facilities in the United States where humans are being crossed with animal and non-human ‘extraterrestrial’ species. They are now easing that long-developed capability into the public arena and there is much more to come given we are dealing with psychiatric basket cases. Talking of which – Elon Musk’s scientists at Neuralink trained a monkey to play Pong and other puzzles on a computer screen using a joystick and when the monkey made the correct move a metal tube squirted banana smoothie into his mouth which is the basic technique for training humans into unquestioning compliance. Two Neuralink chips were in the monkey’s skull and

more than 2,000 wires ‘fanned out’ into its brain. Eventually the monkey played a video game purely with its brain waves. Psychopathic narcissist Musk said the ‘breakthrough’ was a step towards putting Neuralink chips into human skulls and merging minds with artificial intelligence. *Exactly.* This man is so dark and Cult to his DNA.

World Economic Fascism (WEF)

The World Economic Forum is telling you the plan by the statements made at its many and various events. Cult-owned fascist YouTube CEO Susan Wojcicki spoke at the 2021 WEF Global Technology Governance Summit (see the name) in which 40 governments and 150 companies met to ensure ‘the responsible design and deployment of emerging technologies’. Orwellian translation: ‘Ensuring the design and deployment of long-planned technologies will advance the Cult agenda for control and censorship.’ Freedom-destroyer and Nuremberg-bound Wojcicki expressed support for tech platforms like hers to censor content that is ‘technically legal but could be harmful’. Who decides what is ‘harmful’? She does and they do. ‘Harmful’ will be whatever the Cult doesn’t want people to see and we have legislation proposed by the UK government that would censor content on the basis of ‘harm’ no matter if the information is fair, legal and provably true. Make that *especially* if it is fair, legal and provably true. Wojcicki called for a global coalition to be formed to enforce content moderation standards through automated censorship. This is a woman and mega-censor so self-deluded that she shamelessly accepted a ‘free expression’ award – *Wojcicki* – in an event sponsored by her own *YouTube*. They have no shame and no self-awareness.

You know that ‘Covid’ is a scam and Wojcicki a Cult operative when YouTube is censoring medical and scientific opinion purely on the grounds of whether it supports or opposes the Cult ‘Covid’ narrative. Florida governor Ron DeSantis compiled an expert panel with four professors of medicine from Harvard, Oxford, and Stanford Universities who spoke against forcing children and

vaccinated people to wear masks. They also said there was no proof that lockdowns reduced spread or death rates of 'Covid-19'. Cult-gofer Wojcicki and her YouTube deleted the panel video 'because it included content that contradicts the consensus of local and global health authorities regarding the efficacy of masks to prevent the spread of Covid-19'. This 'consensus' refers to what the Cult tells the World Health Organization to say and the WHO tells 'local health authorities' to do. Wojcicki knows this, of course. The panellists pointed out that censorship of scientific debate was responsible for deaths from many causes, but Wojcicki couldn't care less. She would not dare go against what she is told and as a disgrace to humanity she wouldn't want to anyway. The UK government is seeking to pass a fascist 'Online Safety Bill' to specifically target with massive fines and other means non-censored video and social media platforms to make them censor 'lawful but harmful' content like the Cult-owned Facebook, Twitter, Google and YouTube. What is 'lawful but harmful' would be decided by the fascist Blair-created Ofcom.

Another WEF obsession is a cyber-attack on the financial system and this is clearly what the Cult has planned to take down the bank accounts of everyone – except theirs. Those that think they have enough money for the Cult agenda not to matter to them have got a big lesson coming if they continue to ignore what is staring them in the face. The World Economic Forum, funded by Gates and fronted by Klaus Schwab, announced it would be running a 'simulation' with the Russian government and global banks of just such an attack called Cyber Polygon 2021. What they simulate – as with the 'Covid' Event 201 – they plan to instigate. The WEF is involved in a project with the Cult-owned Carnegie Endowment for International Peace called the WEF-Carnegie Cyber Policy Initiative which seeks to merge Wall Street banks, 'regulators' (I love it) and intelligence agencies to 'prevent' (arrange and allow) a cyber-attack that would bring down the global financial system as long planned by those that control the WEF and the Carnegie operation. The Carnegie Endowment for International Peace sent an instruction to First World

War US President Woodrow Wilson not to let the war end before society had been irreversibly transformed.

The Wuhan lab diversion

As I close, the Cult-controlled authorities and lapdog media are systematically pushing ‘the virus was released from the Wuhan lab’ narrative. There are two versions – it happened by accident and it happened on purpose. Both are nonsense. The perceived existence of the never-shown-to-exist ‘virus’ is vital to sell the impression that there is actually an infective agent to deal with and to allow the endless potential for terrifying the population with ‘variants’ of a ‘virus’ that does not exist. The authorities at the time of writing are going with the ‘by accident’ while the alternative media is promoting the ‘on purpose’. Cable news host Tucker Carlson who has questioned aspects of lockdown and ‘vaccine’ compulsion has bought the Wuhan lab story. ‘Everyone now agrees’ he said. Well, I don’t and many others don’t and the question is *why* does the system and its media suddenly ‘agree’? When the media moves as one unit with a narrative it is always a lie – witness the hour by hour mendacity of the ‘Covid’ era. Why would this Cult-owned combination which has unleashed lies like machine gun fire suddenly ‘agree’ to tell the truth??

Much of the alternative media is buying the lie because it fits the conspiracy narrative, but it’s the *wrong* conspiracy. The real conspiracy is that *there is no virus* and that is what the Cult is desperate to hide. The idea that the ‘virus’ was released by accident is ludicrous when the whole ‘Covid’ hoax was clearly long-planned and waiting to be played out as it was so fast in accordance with the Rockefeller document and Event 201. So they prepared everything in detail over decades and then sat around strumming their fingers waiting for an ‘accidental’ release from a bio-lab? *What??* It’s crazy. Then there’s the ‘on purpose’ claim. You want to circulate a ‘deadly virus’ and hide the fact that you’ve done so and you release it down the street from the highest-level bio-lab in China? I repeat – *What??*

You would release it far from that lab to stop any association being made. But, no, we'll do it in a place where the connection was certain to be made. Why would you need to scam 'cases' and 'deaths' and pay hospitals to diagnose 'Covid-19' if you had a real 'virus'? What are sections of the alternative media doing believing this crap? Where were all the mass deaths in Wuhan from a 'deadly pathogen' when the recovery to normal life after the initial propaganda was dramatic in speed? Why isn't the 'deadly pathogen' now circulating all over China with bodies in the street? Once again we have the technique of tell them what they want to hear and they will likely believe it. The alternative media has its 'conspiracy' and with Carlson it fits with his 'China is the danger' narrative over years. China *is* a danger as a global Cult operations centre, but not for this reason. The Wuhan lab story also has the potential to instigate conflict with China when at some stage the plan is to trigger a Problem-Reaction-Solution confrontation with the West. Question everything – *everything* – and especially when the media agrees on a common party line.

Third wave ... fourth wave ... fifth wave ...

As the book went into production the world was being set up for more lockdowns and a 'third wave' supported by invented 'variants' that were increasing all the time and will continue to do so in public statements and computer programs, but not in reality. India became the new Italy in the 'Covid' propaganda campaign and we were told to be frightened of the new 'Indian strain'. Somehow I couldn't find it within myself to do so. A document produced for the UK government entitled 'Summary of further modelling of easing of restrictions – Roadmap Step 2' declared that a third wave was inevitable (of course when it's in the script) and it would be the fault of children and those who refuse the health-destroying fake 'Covid vaccine'. One of the computer models involved came from the Cult-owned *Imperial College* and the other from Warwick University which I wouldn't trust to tell me the date in a calendar factory. The document states that both models presumed extremely high uptake

of the ‘Covid vaccines’ and didn’t allow for ‘variants’. The document states: ‘The resurgence is a result of some people (mostly children) being ineligible for vaccination; others choosing not to receive the vaccine; and others being vaccinated but not perfectly protected.’ The mendacity takes the breath away. Okay, blame those with a brain who won’t take the DNA-modifying shots and put more pressure on children to have it as ‘trials’ were underway involving children as young as six months with parents who give insanity a bad name. Massive pressure is being put on the young to have the fake ‘vaccine’ and child age consent limits have been systematically lowered around the world to stop parents intervening. Most extraordinary about the document was its claim that the ‘third wave’ would be driven by ‘the resurgence in both hospitalisations and deaths … dominated by *those that have received two doses of the vaccine*, comprising around 60-70% of the wave respectively’. The predicted peak of the ‘third wave’ suggested 300 deaths per day with 250 of them *fully ‘vaccinated’ people*. How many more lies do acquiescers need to be told before they see the obvious? Those who took the jab to ‘protect themselves’ are projected to be those who mostly get sick and die? So what’s in the ‘vaccine’? The document went on:

It is possible that a summer of low prevalence could be followed by substantial increases in incidence over the following autumn and winter. Low prevalence in late summer should not be taken as an indication that SARS-CoV-2 has retreated or that the population has high enough levels of immunity to prevent another wave.

They are telling you the script and while many British people believed ‘Covid’ restrictions would end in the summer of 2021 the government was preparing for them to be ongoing. Authorities were awarding contracts for ‘Covid marshals’ to police the restrictions with contracts starting in July, 2021, and going through to January 31st, 2022, and the government was advertising for ‘Media Buying Services’ to secure media propaganda slots worth a potential £320 million for ‘Covid-19 campaigns’ with a contract not ending until March, 2022. The recipient – via a list of other front companies – was reported to be American media marketing giant Omnicom Group

Inc. While money is no object for ‘Covid’ the UK waiting list for all other treatment – including life-threatening conditions – passed 4.5 million. Meantime the Cult is seeking to control all official ‘inquiries’ to block revelations about what has really been happening and why. It must not be allowed to – we need Nuremberg jury trials in every country. The cover-up doesn’t get more obvious than appointing ultra-Zionist professor Philip Zelikow to oversee two dozen US virologists, public health officials, clinicians, former government officials and four American ‘charitable foundations’ to ‘learn the lessons’ of the ‘Covid’ debacle. The personnel will be those that created and perpetuated the ‘Covid’ lies while Zelikow is the former executive director of the 9/11 Commission who ensured that the truth about those attacks never came out and produced a report that must be among the most mendacious and manipulative documents ever written – see *The Trigger* for the detailed exposure of the almost unimaginable 9/11 story in which Sabbatians can be found at every level.

Passive no more

People are increasingly challenging the authorities with amazing numbers of people taking to the streets in London well beyond the ability of the Face-Nappies to stop them. Instead the Nappies choose situations away from the mass crowds to target, intimidate, and seek to promote the impression of ‘violent protestors’. One such incident happened in London’s Hyde Park. Hundreds of thousands walking through the streets in protest against ‘Covid’ fascism were ignored by the Cult-owned BBC and most of the rest of the mainstream media, but they delighted in reporting how police were injured in ‘clashes with protestors’. The truth was that a group of people gathered in Hyde Park at the end of one march when most had gone home and they were peacefully having a good time with music and chat. Face-Nappies who couldn’t deal with the full-march crowd then waded in with their batons and got more than they bargained for. Instead of just standing for this criminal brutality the crowd used their numerical superiority to push the Face-Nappies out of the

park. Eventually the Nappies turned and ran. Unfortunately two or three idiots in the crowd threw drink cans striking two officers which gave the media and the government the image they wanted to discredit the 99.9999 percent who were peaceful. The idiots walked straight into the trap and we must always be aware of potential agent provocateurs used by the authorities to discredit their targets.

This response from the crowd – the can people apart – must be a turning point when the public no longer stand by while the innocent are arrested and brutally attacked by the Face-Nappies. That doesn't mean to be violent, that's the last thing we need. We'll leave the violence to the Face-Nappies and government. But it does mean that when the Face-Nappies use violence against peaceful people the numerical superiority is employed to stop them and make citizen's arrests or Common Law arrests for a breach of the peace. The time for being passive in the face of fascism is over.

We are the many, they are the few, and we need to make that count before there is no freedom left and our children and grandchildren face an ongoing fascist nightmare.

COME ON PEOPLE – IT'S TIME.

One final thought ...

The power of love
A force from above
Cleaning my soul
Flame on burn desire
Love with tongues of fire
Purge the soul
Make love your goal

I'll protect you from the hooded claw
Keep the vampires from your door
When the chips are down I'll be around
With my undying, death-defying
Love for you

Envy will hurt itself
Let yourself be beautiful
Sparkling love, flowers
And pearls and pretty girls
Love is like an energy
Rushin' rushin' inside of me

This time we go sublime
Lovers entwine, divine, divine,
Love is danger, love is pleasure
Love is pure – the only treasure

I'm so in love with you
Purge the soul
Make love your goal

The power of love
A force from above
Cleaning my soul
The power of love
A force from above
A sky-scraping dove

Flame on burn desire
Love with tongues of fire
Purge the soul
Make love your goal

Frankie Goes To Hollywood

APPENDIX

Cowan-Kaufman-Morell Statement on Virus Isolation (SOVI)

Isolation: The action of isolating; the fact or condition of being isolated or standing alone; separation from other things or persons; solitariness

Oxford English Dictionary

The controversy over whether the SARS-CoV-2 virus has ever been isolated or purified continues. However, using the above definition, common sense, the laws of logic and the dictates of science, any unbiased person must come to the conclusion that the SARS-CoV-2 virus has never been isolated or purified. As a result, no confirmation of the virus' existence can be found. The logical, common sense, and scientific consequences of this fact are:

- the structure and composition of something not shown to exist can't be known, including the presence, structure, and function of any hypothetical spike or other proteins;
- the genetic sequence of something that has never been found can't be known;
- "variants" of something that hasn't been shown to exist can't be known;
- it's impossible to demonstrate that SARS-CoV-2 causes a disease called Covid-19.

In as concise terms as possible, here's the proper way to isolate, characterize and demonstrate a new virus. First, one takes samples (blood, sputum, secretions) from many people (e.g. 500) with symptoms which are unique and specific enough to characterize an illness. Without mixing these samples with ANY tissue or products that also contain genetic material, the virologist macerates, filters and ultracentrifuges i.e. *purifies* the specimen. This common virology technique, done for decades to isolate bacteriophages¹ and so-called giant viruses in every virology lab, then allows the virologist to demonstrate with electron microscopy thousands of identically sized and shaped particles. These particles are the isolated and purified virus.

These identical particles are then checked for uniformity by physical and/or microscopic techniques. Once the purity is determined, the particles may be further characterized. This would include examining the structure, morphology, and chemical composition of the particles. Next, their genetic makeup is characterized by extracting the genetic material directly from the purified particles and using genetic-sequencing techniques, such as Sanger sequencing, that have also been around for decades. Then one does an analysis to confirm that these uniform particles are exogenous (outside) in origin as a virus is conceptualized to be, and not the normal breakdown products of dead and dying tissues.² (As of May 2020, we know that virologists have no way to determine whether the particles they're seeing are viruses or just normal breakdown products of dead and dying tissues.)³

1 Isolation, characterization and analysis of bacteriophages from the haloalkaline lake Elmenteita, KenyaJuliah Khayeli Akhwale et al, PLOS One, Published: April 25, 2019.
<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0215734> – accessed 2/15/21

2 "Extracellular Vesicles Derived From Apoptotic Cells: An Essential Link Between Death and Regeneration," Maojiao Li et al, Frontiers in Cell and Developmental Biology, 2020 October 2.
<https://www.frontiersin.org/articles/10.3389/fcell.2020.573511/full> – accessed 2/15/21

3 "The Role of Extracellular Vesicles as Allies of HIV, HCV and SARS Viruses," Flavia Giannessi, et al, *Viruses*, 2020 May

If we have come this far then we have fully isolated, characterized, and genetically sequenced an exogenous virus particle. However, we still have to show it is causally related to a disease. This is carried out by exposing a group of healthy subjects (animals are usually used) to this isolated, purified virus in the manner in which the disease is thought to be transmitted. If the animals get sick with the same disease, as confirmed by clinical and autopsy findings, one has now shown that the virus actually causes a disease. This demonstrates infectivity and transmission of an infectious agent.

None of these steps has even been attempted with the SARS-CoV-2 virus, nor have all these steps been successfully performed for any so-called pathogenic virus. Our research indicates that a single study showing these steps does not exist in the medical literature.

Instead, since 1954, virologists have taken unpurified samples from a relatively few people, often less than ten, with a similar disease. They then minimally process this sample and inoculate this unpurified sample onto tissue culture containing usually four to six other types of material – all of which contain identical genetic material as to what is called a “virus.” The tissue culture is starved and poisoned and naturally disintegrates into many types of particles, some of which contain genetic material. Against all common sense, logic, use of the English language and scientific integrity, this process is called “virus isolation.” This brew containing fragments of genetic material from many sources is then subjected to genetic analysis, which then creates in a computer-simulation process the alleged sequence of the alleged virus, a so-called *in silico* genome. At no time is an actual virus confirmed by electron microscopy. At no time is a genome extracted and sequenced from an actual virus. This is scientific fraud.

The observation that the unpurified specimen — inoculated onto tissue culture along with toxic antibiotics, bovine fetal tissue, amniotic fluid and other tissues — destroys the kidney tissue onto which it is inoculated is given as evidence of the virus' existence and pathogenicity. This is scientific fraud.

From now on, when anyone gives you a paper that suggests the SARS-CoV-2 virus has been isolated, please check the methods sections. If the researchers used Vero cells or any other culture method, you know that their process was not isolation. You will hear the following excuses for why actual isolation isn't done:

1. There were not enough virus particles found in samples from patients to analyze.
2. Viruses are intracellular parasites; they can't be found outside the cell in this manner.

If No. 1 is correct, and we can't find the virus in the sputum of sick people, then on what evidence do we think the virus is dangerous or even lethal? If No. 2 is correct, then how is the virus spread from person to person? We are told it emerges from the cell to infect others. Then why isn't it possible to find it?

Finally, questioning these virology techniques and conclusions is not some distraction or divisive issue. Shining the light on this truth is essential to stop this terrible fraud that humanity is confronting. For, as we now know, if the virus has never been isolated, sequenced or shown to cause illness, if the virus is imaginary, then why are we wearing masks, social distancing and putting the whole world into prison?

Finally, if pathogenic viruses don't exist, then what is going into those injectable devices erroneously called "vaccines," and what is their purpose? This scientific question is the most urgent and relevant one of our time.

We are correct. The SARS-CoV2 virus does not exist.

Sally Fallon Morell, MA

Dr. Thomas Cowan, MD

Dr. Andrew Kaufman, MD

Bibliography

- Alinsky, Saul:** *Rules for Radicals* (Vintage, 1989)
- Antelman, Rabbi Marvin:** *To Eliminate the Opiate* (Zahavia, 1974)
- Bastardi, Joe:** *The Climate Chronicles* (Relentless Thunder Press, 2018)
- Cowan, Tom:** *Human Heart, Cosmic Heart* (Chelsea Green Publishing, 2016)
- Cowan, Tom, and Fallon Morell, Sally:** *The Contagion Myth* (Skyhorse Publishing, 2020)
- Forbes, Jack D:** *Columbus And Other Cannibals – The Wetiko Disease of Exploitation, Imperialism, and Terrorism* (Seven Stories Press, 2008 – originally published in 1979)
- Gates, Bill:** *How to Avoid a Climate Disaster: The Solutions We Have and the Breakthroughs We Need* (Allen Lane, 2021)
- Huxley, Aldous:** *Brave New World* (Chatto & Windus, 1932)
- Köhnlein, Dr Claus, and Engelbrecht, Torsten:** *Virus Mania* (emu-Vertag, Lahnstein, 2020)
- Lanza, Robert, and Berman, Bob:** *Biocentrism* (BenBella Books, 2010)
- Lash, John Lamb:** *Not In His Image* (Chelsea Green Publishing, 2006)
- Lester, Dawn, and Parker, David:** *What Really Makes You Ill – Why everything you thought you knew about disease is wrong* (Independently Published, 2019)
- Levy, Paul:** *Dispelling Wetiko, Breaking the Spell of Evil* (North Atlantic Books, 2013)
- Marx, Karl:** *A World Without Jews* (Philosophical Library, first edition, 1959)
- Mullis, Kary:** *Dancing Naked in the Mine Field* (Bloomsbury, 1999)
- O'Brien, Cathy:** *Trance-Formation of America* (Reality Marketing, 1995)
- Scholem, Gershon:** *The Messianic Idea in Judaism* (Schocken Books, 1994)
- Schwab, Klaus, and Davis, Nicholas:** *Shaping the Future of the Fourth Industrial Revolution: A guide to building a better world* (Penguin Books, 2018)
- Schwab, Klaus:** *The Great Reset* (Agentur Schweiz, 2020)
- Sunstein, Cass and Thaler, Richard:** *Nudge: Improving Decisions About Health, Wealth, and Happiness* (Penguin, 2009)
- Swan, Shanna:** *Count Down: How Our Modern World Is Threatening Sperm Counts, Altering Male and Female Reproductive Development and Imperiling the Future of the Human Race* (Scribner, 2021)
- Tegmark, Max:** *Our Mathematical Universe: My Quest for the Ultimate Nature of Reality* (Penguin, 2015)
- Velikovsky, Immanuel:** *Worlds in Collision* (Paradigma, 2009)

Wilton, Robert: *The Last Days of the Romanovs* (Blurb, 2018, first published 1920)

Index

A

abusive relationships

blaming themselves, abused as [ref1](#)
children [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#)
conspiracy theories [ref1](#)
domestic abuse [ref1](#), [ref2](#)
economic abuse and dependency [ref1](#)
isolation [ref1](#)
physical abuse [ref1](#)
psychological abuse [ref1](#)
signs of abuse [ref1](#)

addiction

alcoholism [ref1](#)
frequencies [ref1](#)
substance abuse [ref1](#), [ref2](#)
technology [ref1](#), [ref2](#), [ref3](#)

Adelson, Sheldon [ref1](#), [ref2](#), [ref3](#)

Agenda 21/Agenda 2030 (UN) [ref1](#), [ref2](#), [ref3](#), [ref4](#)

AIDs/HIV [ref1](#)

causal link between HIV and AIDs [ref1](#), [ref2](#)
retroviruses [ref1](#)
testing [ref1](#), [ref2](#)
trial-run for Covid-19, as [ref1](#), [ref2](#)
aliens/extraterrestrials [ref1](#), [ref2](#)
aluminium [ref1](#)
Amazon [ref1](#), [ref2](#), [ref3](#)

amplification cycles [ref1](#), [ref2](#)
anaphylactic shock [ref1](#), [ref2](#), [ref3](#), [ref4](#)
animals [ref1](#), [ref2](#), [ref3](#)
antibodies [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
Antifa [ref1](#), [ref2](#), [ref3](#), [ref4](#)
antigens [ref1](#), [ref2](#)
anti-Semitism [ref1](#), [ref2](#), [ref3](#)
Archons [ref1](#), [ref2](#)
 consciousness [ref1](#), [ref2](#), [ref3](#)
 energy [ref1](#), [ref2](#), [ref3](#)
 ennoia [ref1](#)
 genetic manipulation [ref1](#), [ref2](#)
 inversion [ref1](#), [ref2](#), [ref3](#)
 lockdowns [ref1](#)
 money [ref1](#)
 radiation [ref1](#)
 religion [ref1](#), [ref2](#)
 technology [ref1](#), [ref2](#), [ref3](#)
 Wetiko factor [ref1](#), [ref2](#), [ref3](#), [ref4](#)
artificial intelligence (AI) [ref1](#)
army made up of robots [ref1](#), [ref2](#)
 Human 2.0 [ref1](#), [ref2](#)
 Internet [ref1](#)
 MHRA [ref1](#)
 Morgellons fibres [ref1](#), [ref2](#)
 Smart Grid [ref1](#)
 Wetiko factor [ref1](#)
asymptomatic, Covid-19 as [ref1](#), [ref2](#), [ref3](#)
aviation industry [ref1](#)

B

banking, finance and money [ref1](#), [ref2](#), [ref3](#)

2008 crisis [ref1](#), [ref2](#)

boom and bust [ref1](#)

cashless digital money systems [ref1](#)

central banks [ref1](#)

credit [ref1](#)

digital currency [ref1](#)

fractional reserve lending [ref1](#)

Great Reset [ref1](#)

guaranteed income [ref1](#), [ref2](#), [ref3](#)

Human 2.0 [ref1](#)

incomes, destruction of [ref1](#), [ref2](#)

interest [ref1](#)

one per cent [ref1](#), [ref2](#)

scams [ref1](#)

BBC [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)

Becker-Phelps, Leslie [ref1](#)

Behavioural Insights Team (BIT) (Nudge Unit) [ref1](#), [ref2](#), [ref3](#)

behavioural scientists and psychologists, advice from [ref1](#), [ref2](#)

Bezos, Jeff [ref1](#), [ref2](#), [ref3](#), [ref4](#)

Biden, Hunter [ref1](#)

Biden, Joe [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#), [ref11](#),
[ref12](#), [ref13](#), [ref14](#), [ref15](#), [ref16](#), [ref17](#)

Big Pharma

cholesterol [ref1](#)

health professionals [ref1](#), [ref2](#)

immunity from prosecution in US [ref1](#)

vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)

Wetiko factor [ref1](#), [ref2](#)

WHO [ref1](#), [ref2](#), [ref3](#)

Bill and Melinda Gates Foundation [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#),
[ref7](#)

billionaires [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#) [ref10](#), [ref11](#)
bird flu (H5N1) [ref1](#)
Black Lives Matter (BLM) [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
Blair, Tony [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
Brin, Sergei [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
British Empire [ref1](#)
Bush, George HW [ref1](#), [ref2](#)
Bush, George W [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Byrd, Robert [ref1](#)

C

Canada

Global Cult [ref1](#)
hate speech [ref1](#)
internment [ref1](#)
masks [ref1](#)
old people [ref1](#)
SARS-COV-2 [ref1](#)
satellites [ref1](#)
vaccines [ref1](#)
wearable technology [ref1](#)

Capitol Hill riot [ref1](#), [ref2](#)
agents provocateur [ref1](#)
Antifa [ref1](#)
Black Lives Matter (BLM) [ref1](#), [ref2](#)
QAnon [ref1](#)
security precautions, lack of [ref1](#), [ref2](#), [ref3](#)

carbon dioxide [ref1](#), [ref2](#)
care homes, deaths in [ref1](#), [ref2](#)
cashless digital money systems [ref1](#)
censorship [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

fact-checkers [ref1](#)
masks [ref1](#)
media [ref1](#), [ref2](#)
private messages [ref1](#)
social media [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)
transgender persons [ref1](#)
vaccines [ref1](#), [ref2](#), [ref3](#)
Wokeness [ref1](#)

Centers for Disease Control (CDC) (United States) [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#), [ref11](#), [ref12](#), [ref13](#)

centralisation [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)

chakras [ref1](#)

change agents [ref1](#), [ref2](#), [ref3](#)

chemtrails [ref1](#), [ref2](#), [ref3](#)

chief medical officers and scientific advisers [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)

children *see also young people*

abuse [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#)

care, taken into [ref1](#), [ref2](#), [ref3](#)

education [ref1](#), [ref2](#), [ref3](#), [ref4](#)

energy [ref1](#)

family courts [ref1](#)

hand sanitisers [ref1](#)

human sacrifice [ref1](#)

lockdowns [ref1](#), [ref2](#), [ref3](#)

masks [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

mental health [ref1](#)

old people [ref1](#)

parents, replacement of [ref1](#), [ref2](#)

Psyop (psychological operation), Covid as a [ref1](#), [ref2](#)

reframing [ref1](#)

smartphone addiction [ref1](#)

social distancing and isolation [ref1](#)
social media [ref1](#)
transgender persons [ref1](#), [ref2](#)
United States [ref1](#)
vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#)
Wetiko factor [ref1](#)

China [ref1](#), [ref2](#), [ref3](#), [ref4](#)
anal swab tests [ref1](#)
Chinese Revolution [ref1](#), [ref2](#), [ref3](#)
digital currency [ref1](#)
Global Cult [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#)
guaranteed income [ref1](#)
Imperial College [ref1](#)
Israel [ref1](#)
lockdown [ref1](#), [ref2](#)
masculinity crisis [ref1](#)
masks [ref1](#)
media [ref1](#)
origins of virus in China [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
pollution causing respiratory diseases [ref1](#)
Sabbatians [ref1](#), [ref2](#)
Smart Grid [ref1](#), [ref2](#)
social credit system [ref1](#)
testing [ref1](#), [ref2](#)
United States [ref1](#), [ref2](#)
vaccines [ref1](#), [ref2](#)
Wetiko factor [ref1](#)
wet market conspiracy [ref1](#)
Wuhan [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)

cholesterol [ref1](#), [ref2](#)

Christianity [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
criticism [ref1](#)
cross, inversion of the [ref1](#)

Nag Hammadi texts [ref1](#), [ref2](#), [ref3](#)
Roman Catholic Church [ref1](#), [ref2](#)
Sabbatians [ref1](#), [ref2](#)
Satan [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Wokeness [ref1](#)

class [ref1](#), [ref2](#)

climate change hoax [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Agenda 21/Agenda 2030 [ref1](#), [ref2](#), [ref3](#)

carbon dioxide [ref1](#), [ref2](#)

Club of Rome [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

fear [ref1](#)

funding [ref1](#)

Global Cult [ref1](#)

green new deals [ref1](#)

green parties [ref1](#)

inversion [ref1](#)

perception, control of [ref1](#)

PICC [ref1](#)

reframing [ref1](#)

temperature, increases in [ref1](#)

United Nations [ref1](#), [ref2](#)

Wikipedia [ref1](#)

Wokeness [ref1](#), [ref2](#)

Clinton, Bill [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)

Clinton, Hillary [ref1](#), [ref2](#), [ref3](#)

the cloud [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)

Club of Rome and climate change hoax [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

cognitive therapy [ref1](#)

Cohn, Roy [ref1](#)

Common Law [ref1](#)

Admiralty Law [ref1](#)

arrests [ref1](#), [ref2](#)

contractual law, Statute Law as [ref1](#)
corporate entities, people as [ref1](#)
legalese [ref1](#)
sea, law of the [ref1](#)
Statute Law [ref1](#)

Common Purpose leadership programme [ref1](#), [ref2](#)
communism [ref1](#), [ref2](#)
co-morbidities [ref1](#)
computer-generated virus,
Covid-19 as [ref1](#), [ref2](#), [ref3](#)
computer models [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
connections [ref1](#), [ref2](#), [ref3](#), [ref4](#)
consciousness [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Archons [ref1](#), [ref2](#), [ref3](#)
expanded [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
experience [ref1](#)
heart [ref1](#)
infinity [ref1](#), [ref2](#)
religion [ref1](#), [ref2](#)
self-identity [ref1](#)
simulation thesis [ref1](#)
vaccines [ref1](#)
Wetiko factor [ref1](#), [ref2](#)

conspiracy theorists [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
contradictory rules [ref1](#)
contrails [ref1](#)
Corman-Drosten test [ref1](#), [ref2](#), [ref3](#), [ref4](#)
countermimicry [ref1](#), [ref2](#), [ref3](#)
Covid-19 vaccines *see* vaccines
Covidiots [ref1](#), [ref2](#)
Cowan, Tom [ref1](#), [ref2](#), [ref3](#), [ref4](#)
crimes against humanity [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)

cyber-operations [ref1](#)

cyberwarfare [ref1](#)

D

DARPA (Defense Advanced Research Projects Agency) [ref1](#)

deaths

care homes [ref1](#)

certificates [ref1](#), [ref2](#), [ref3](#), [ref4](#)

mortality rate [ref1](#)

post-mortems/autopsies [ref1](#)

recording [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)

vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

deceit

pyramid of deceit [ref1](#), [ref2](#)

sequence of deceit [ref1](#)

decoding [ref1](#), [ref2](#), [ref3](#)

dehumanisation [ref1](#), [ref2](#), [ref3](#)

Delphi technique [ref1](#)

democracy [ref1](#)

dependency [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Descartes, René [ref1](#)

DNA

numbers [ref1](#)

vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#)

DNR (do not resuscitate)

orders [ref1](#)

domestic abuse [ref1](#), [ref2](#)

downgrading of Covid-19 [ref1](#)

Drosten, Christian [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)

Duesberg, Peter [ref1](#), [ref2](#)

E

economic abuse [ref1](#)

Edmunds, John [ref1](#), [ref2](#)

education [ref1](#), [ref2](#), [ref3](#), [ref4](#)

electromagnetic spectrum [ref1](#), [ref2](#)

Enders, John [ref1](#)

energy

Archons [ref1](#), [ref2](#), [ref3](#)

children and young people [ref1](#)

consciousness [ref1](#)

decoding [ref1](#)

frequencies [ref1](#), [ref2](#), [ref3](#), [ref4](#)

heart [ref1](#)

human energy field [ref1](#)

source, humans as an energy [ref1](#), [ref2](#)

vaccines [ref1](#)

viruses [ref1](#)

ennoia [ref1](#)

Epstein, Jeffrey [ref1](#), [ref2](#)

eternal 'I' [ref1](#), [ref2](#)

ethylene oxide [ref1](#)

European Union [ref1](#), [ref2](#), [ref3](#), [ref4](#)

Event [ref1](#) and Bill Gates [ref2](#)

exosomes, Covid-19 as natural defence mechanism called [ref1](#)

experience [ref1](#), [ref2](#)

Extinction Rebellion [ref1](#), [ref2](#)

F

Facebook

addiction [ref1](#), 448–50

Facebook

Archons [ref1](#)
censorship [ref1](#), [ref2](#), [ref3](#)
hate speech [ref1](#)
monopoly, as [ref1](#)
private messages, censorship of [ref1](#)
Sabbatians [ref1](#)
United States election fraud [ref1](#)
vaccines [ref1](#)
Wetiko factor [ref1](#)

fact-checkers [ref1](#)

Fauci, Anthony [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#),
[ref11](#), [ref12](#)

fear [ref1](#), [ref2](#), [ref3](#), [ref4](#)
climate change [ref1](#)
computer models [ref1](#)
conspiracy theories [ref1](#)
empty hospitals [ref1](#)
Italy [ref1](#), [ref2](#), [ref3](#)
lockdowns [ref1](#), [ref2](#), [ref3](#), [ref4](#)
masks [ref1](#), [ref2](#)
media [ref1](#), [ref2](#)
medical staff [ref1](#)
Psyop (psychological operation), Covid as a [ref1](#)
Wetiko factor [ref1](#), [ref2](#)

female infertility [ref1](#)

Fermi Paradox [ref1](#)

Ferguson, Neil [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)

fertility, decline in [ref1](#)

The Field [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)

finance *see banking, finance and money*

five-senses [ref1](#), [ref2](#)
Archons [ref1](#), [ref2](#), [ref3](#)

censorship [ref1](#)
consciousness, expansion of [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)
decoding [ref1](#)
education [ref1](#), [ref2](#)
the Field [ref1](#), [ref2](#)
God, personification of [ref1](#)
infinity [ref1](#), [ref2](#)
media [ref1](#)
paranormal [ref1](#)
perceptual programming [ref1](#), [ref2](#)
Phantom Self [ref1](#)
pneuma not nous, using [ref1](#)
reincarnation [ref1](#)
self-identity [ref1](#)
Wetiko factor [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)
5G [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)
Floyd, George and protests, killing of [ref1](#)
flu, re-labelling of [ref1](#), [ref2](#), [ref3](#)
food and water, control of [ref1](#), [ref2](#)
Freemasons [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)
Frei, Rosemary [ref1](#)
frequencies
addictions [ref1](#)
Archons [ref1](#), [ref2](#), [ref3](#)
awareness [ref1](#)
chanting and mantras [ref1](#)
consciousness [ref1](#)
decoding [ref1](#), [ref2](#)
education [ref1](#)
electromagnetic (EMF) frequencies [ref1](#)
energy [ref1](#), [ref2](#), [ref3](#), [ref4](#)
fear [ref1](#)

the Field [ref1](#), [ref2](#) 5G [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#)
five-senses [ref1](#), [ref2](#)
ghosts [ref1](#)
Gnostics [ref1](#)
hive-minds [ref1](#)
human, meaning of [ref1](#)
light [ref1](#), [ref2](#)
love [ref1](#), [ref2](#)
magnetism [ref1](#)
perception [ref1](#)
reality [ref1](#), [ref2](#), [ref3](#)
simulation [ref1](#)
terror [ref1](#)
vaccines [ref1](#)
Wetiko [ref1](#), [ref2](#), [ref3](#)
Fuellmich, Reiner [ref1](#), [ref2](#), [ref3](#)
furlough/rescue payments [ref1](#)

G

Gallo, Robert [ref1](#), [ref2](#), [ref3](#)

Gates, Bill

Archons [ref1](#), [ref2](#), [ref3](#)
climate change [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Daily Pass tracking system [ref1](#)
Epstein [ref1](#)
fascism [ref1](#)
five senses [ref1](#)
GAVI [ref1](#)
Great Reset [ref1](#)
GSK [ref1](#)
Imperial College [ref1](#), [ref2](#)
Johns Hopkins University [ref1](#), [ref2](#), [ref3](#)

lockdowns [ref1](#), [ref2](#)

masks [ref1](#)

Nuremberg trial, proposal for [ref1](#), [ref2](#)

Rockefellers [ref1](#), [ref2](#)

social distancing and isolation [ref1](#)

Sun, dimming the [ref1](#)

synthetic meat [ref1](#), [ref2](#)

vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)

Wellcome Trust [ref1](#)

Wetiko factor [ref1](#), [ref2](#), [ref3](#)

WHO [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#)

Wokeness [ref1](#)

World Economic Forum [ref1](#), [ref2](#), [ref3](#), [ref4](#)

Gates, Melinda [ref1](#), [ref2](#), [ref3](#)

GAVI vaccine alliance [ref1](#)

genetics, manipulation of [ref1](#), [ref2](#), [ref3](#)

Germany [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#) *see also Nazi Germany*

Global Cult [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

anti-human, why Global Cult is [ref1](#)

Black Lives Matter (BLM) [ref1](#), [ref2](#), [ref3](#), [ref4](#)

China [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#)

climate change hoax [ref1](#)

contradictory rules [ref1](#)

Covid-19 [ref1](#), [ref2](#), [ref3](#)

fascism [ref1](#)

geographical origins [ref1](#)

immigration [ref1](#)

Internet [ref1](#)

mainstream media [ref1](#), [ref2](#)

masks [ref1](#), [ref2](#)

monarchy [ref1](#)

non-human dimension [ref1](#)

perception [ref1](#)
political parties [ref1](#), [ref2](#)
pyramidal hierarchy [ref1](#), [ref2](#), [ref3](#)
reframing [ref1](#)
Sabbantian-Frankism [ref1](#), [ref2](#)
science, manipulation of [ref1](#)
spider and the web [ref1](#)
transgender persons [ref1](#)
vaccines [ref1](#)
who controls the Cult [ref1](#)
Wokeness [ref1](#), [ref2](#), [ref3](#), [ref4](#)

globalisation [ref1](#), [ref2](#)

Gnostics [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Google [ref1](#), [ref2](#), [ref3](#), [ref4](#)

government

- behavioural scientists and psychologists, advice from [ref1](#), [ref2](#)
- definition [ref1](#)
- Joint Biosecurity Centre (JBC) [ref1](#)
- people, abusive relationship with [ref1](#)

Great Reset [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)

- fascism [ref1](#), [ref2](#), [ref3](#)
- financial system [ref1](#)
- Human 2.0 [ref1](#)
- water and food, control of [ref1](#)

green parties [ref1](#)

Griesz-Brisson, Margarite [ref1](#)

guaranteed income [ref1](#), [ref2](#), [ref3](#)

H

Hancock, Matt [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

hand sanitisers [ref1](#)

heart [ref1](#), [ref2](#)

hive-minds/groupthink [ref1](#), [ref2](#), [ref3](#)

holographs [ref1](#), [ref2](#), [ref3](#), [ref4](#)

hospitals, empty [ref1](#)

human, meaning of [ref1](#)

Human 2.0 [ref1](#)

addiction to technology [ref1](#)

artificial intelligence (AI) [ref1](#), [ref2](#)

elimination of Human 1.0 [ref1](#)

fertility, decline in [ref1](#)

Great Reset [ref1](#)

implantables [ref1](#)

money [ref1](#)

mRNA [ref1](#)

nanotechnology [ref1](#)

parents, replacement of [ref1](#), [ref2](#)

Smart Grid, connection to [ref1](#), [ref2](#)

synthetic biology [ref1](#), [ref2](#), [ref3](#), [ref4](#)

testosterone levels, decrease in [ref1](#)

transgender = transhumanism [ref1](#), [ref2](#), [ref3](#)

vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#)

human sacrifice [ref1](#), [ref2](#), [ref3](#)

Hunger Games Society [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)

Huxley, Aldous [ref1](#), [ref2](#), [ref3](#)

I

identity politics [ref1](#), [ref2](#), [ref3](#)

Illuminati [ref1](#), [ref2](#)

illusory physical reality [ref1](#)

immigration [ref1](#), [ref2](#), [ref3](#), [ref4](#)

Imperial College [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)

implantables [ref1](#), [ref2](#)

incomes, destruction of [ref1](#), [ref2](#)

Infinite Awareness [ref1](#), [ref2](#), [ref3](#), [ref4](#)

Internet [ref1](#), [ref2](#) *see also* social media
 artificial intelligence (AI) [ref1](#)
 independent journalism, lack of [ref1](#)
 Internet of Bodies (IoB) [ref1](#)

Internet of Everything (IoE) [ref1](#), [ref2](#)

Internet of Things (IoT) [ref1](#), [ref2](#)

lockdowns [ref1](#)

Psyop (psychological operation), Covid as a [ref1](#)
 trolls [ref1](#)

intersectionality [ref1](#)

inversion
 Archons [ref1](#), [ref2](#), [ref3](#)
 climate change hoax [ref1](#)
 energy [ref1](#)
 Judaism [ref1](#), [ref2](#), [ref3](#)
 symbolism [ref1](#)
 Wetiko factor [ref1](#)
 Wokeness [ref1](#), [ref2](#), [ref3](#)

Islam
 Archons [ref1](#)
 crypto-Jews [ref1](#)
 Islamic State [ref1](#), [ref2](#)
 Jinn and Djinn [ref1](#), [ref2](#), [ref3](#)
 Ottoman Empire [ref1](#)
 Wahhabism [ref1](#)

isolation *see* **social distancing and isolation**

Israel
 China [ref1](#)
 Cyber Intelligence Unit Beersheba complex [ref1](#)
 expansion of illegal settlements [ref1](#)

formation [ref1](#)
Global Cult [ref1](#)
Judaism [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
medical experiments, consent for [ref1](#)
Mossad [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Palestine-Israel conflict [ref1](#), [ref2](#), [ref3](#)
parents, replacement of [ref1](#)
Sabbatians [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
September 11, 2001, terrorist attacks on United States [ref1](#)
Silicon Valley [ref1](#)
Smart Grid [ref1](#), [ref2](#)
United States [ref1](#), [ref2](#)
vaccines [ref1](#)
Wetiko factor [ref1](#)

Italy

fear [ref1](#), [ref2](#), [ref3](#)
Lombardy [ref1](#), [ref2](#), [ref3](#)
vaccines [ref1](#)

J

Johns Hopkins University [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
Johnson, Boris [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)
Joint Biosecurity Centre (JBC) [ref1](#)

Judaism

anti-Semitism [ref1](#), [ref2](#), [ref3](#)
Archons [ref1](#), [ref2](#)
crypto-Jews [ref1](#)
inversion [ref1](#), [ref2](#), [ref3](#)
Israel [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
Labour Party [ref1](#)
Nazi Germany [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Sabbatians [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Silicon Valley [ref1](#)
Torah [ref1](#)
United States [ref1](#), [ref2](#)
Zionists [ref1](#), [ref2](#), [ref3](#)

K

Kaufman, Andrew [ref1](#), [ref2](#), [ref3](#), [ref4](#)
knowledge [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)
Koch's postulates [ref1](#)
Kurzweil, Ray [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
Kushner, Jared [ref1](#), [ref2](#)

L

Labour Party [ref1](#), [ref2](#)
Lanka, Stefan [ref1](#), [ref2](#)
Lateral Flow Device (LFD) [ref1](#)
Levy, Paul [ref1](#), [ref2](#), [ref3](#)
Life Program [ref1](#)
lockdowns [ref1](#), [ref2](#), [ref3](#)
 amplification tampering [ref1](#)
 Archons [ref1](#)
 Behavioural Insights Team [ref1](#)
 Black Lives Matter (BLM) [ref1](#)
 care homes, deaths in [ref1](#)
 children
 abuse [ref1](#), [ref2](#)
 mental health [ref1](#)
 China [ref1](#), [ref2](#)
 computer models [ref1](#)
 consequences [ref1](#), [ref2](#)
 dependency [ref1](#), [ref2](#), [ref3](#)

domestic abuse [ref1](#)
fall in cases [ref1](#)
fear [ref1](#), [ref2](#), [ref3](#), [ref4](#)
guaranteed income [ref1](#)
Hunger Games Society [ref1](#), [ref2](#), [ref3](#)
interaction, destroying [ref1](#)
Internet [ref1](#), [ref2](#)
overdoses [ref1](#)
perception [ref1](#)
police-military state [ref1](#), [ref2](#)
protests [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
psychopathic personality [ref1](#), [ref2](#), [ref3](#)
reporting/snitching, encouragement of [ref1](#), [ref2](#)
testing [ref1](#)
vaccines [ref1](#)
Wetiko factor [ref1](#)
WHO [ref1](#)
love [ref1](#), [ref2](#), [ref3](#)
Lucifer [ref1](#), [ref2](#), [ref3](#)

M

Madej, Carrie [ref1](#), [ref2](#)
Magufuli, John [ref1](#), [ref2](#)
mainstream media [ref1](#)
BBC [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)
censorship [ref1](#), [ref2](#)
China [ref1](#)
climate change hoax [ref1](#)
fear [ref1](#), [ref2](#)
Global Cult [ref1](#), [ref2](#)
independent journalism, lack of [ref1](#)
Ofcom [ref1](#), [ref2](#), [ref3](#)

perception [ref1](#), [ref2](#)
Psyop (psychological operation), Covid as a [ref1](#)
Sabbatians [ref1](#), [ref2](#)
social disapproval [ref1](#)
social distancing and isolation [ref1](#)
United States [ref1](#), [ref2](#)
vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Mao Zedong [ref1](#), [ref2](#), [ref3](#)

Marx and Marxism [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)

masculinity [ref1](#)

masks/face coverings [ref1](#), [ref2](#), [ref3](#)

censorship [ref1](#)

children [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

China, made in [ref1](#)

dehumanisation [ref1](#), [ref2](#), [ref3](#)

fear [ref1](#), [ref2](#)

flu [ref1](#)

health professionals [ref1](#), [ref2](#), [ref3](#), [ref4](#)

isolation [ref1](#)

laughter [ref1](#)

mass non-cooperation [ref1](#)

microplastics, risk of [ref1](#)

mind control [ref1](#)

multiple masks [ref1](#)

oxygen deficiency [ref1](#), [ref2](#), [ref3](#)

police [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

pollution, as cause of plastic [ref1](#)

Psyop (psychological operation), Covid as a [ref1](#)

reframing [ref1](#), [ref2](#)

risk assessments, lack of [ref1](#), [ref2](#)

self-respect [ref1](#)

surgeons [ref1](#)

United States [ref1](#)
vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
Wetiko factor [ref1](#)
'worms' [ref1](#)
The Matrix movies [ref1](#), [ref2](#), [ref3](#)
measles [ref1](#), [ref2](#)
media see mainstream media
Medicines and Healthcare products Regulatory Agency (MHRA)
[ref1](#), [ref2](#), [ref3](#), [ref4](#)
Mesopotamia [ref1](#)
messaging [ref1](#)
military-police state [ref1](#), [ref2](#), [ref3](#)
mind control [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#) *see also MKUltra*
MKUltra [ref1](#), [ref2](#), [ref3](#)
monarchy [ref1](#)
money *see banking, finance and money*
Montagnier, Luc [ref1](#), [ref2](#), [ref3](#)
Mooney, Bel [ref1](#)
Morgellons disease [ref1](#), [ref2](#)
mortality rate [ref1](#)
Mullis, Kary [ref1](#), [ref2](#), [ref3](#)
Musk, Elon [ref1](#)

N

Nag Hammadi texts [ref1](#), [ref2](#), [ref3](#)
nanotechnology [ref1](#), [ref2](#), [ref3](#)
narcissism [ref1](#)
Nazi Germany [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)
near-death experiences [ref1](#), [ref2](#)
Neocons [ref1](#), [ref2](#), [ref3](#)

Neuro-Linguistic Programming (NLP) and the Delphi technique
[ref1](#)

NHS (National Health Service)

amplification cycles [ref1](#)

Common Purpose [ref1](#), [ref2](#)

mind control [ref1](#)

NHS England [ref1](#)

saving the NHS [ref1](#), [ref2](#)

vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

whistle-blowers [ref1](#), [ref2](#), [ref3](#)

No-Problem-Reaction-Solution [ref1](#), [ref2](#), [ref3](#), [ref4](#)

non-human dimension of Global Cult [ref1](#)

nous [ref1](#)

numbers, reality as [ref1](#)

Nuremberg Codes [ref1](#), [ref2](#), [ref3](#)

Nuremberg-like tribunal, proposal for [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#),
[ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#), [ref11](#), [ref12](#)

Ø

Obama, Barack [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#)

O'Brien, Cathy [ref1](#), [ref2](#), [ref3](#), [ref4](#)

Ochel, Evita [ref1](#)

Ofcom [ref1](#), [ref2](#), [ref3](#)

old people [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Oneness [ref1](#), [ref2](#), [ref3](#)

Open Society Foundations (Soros) [ref1](#), [ref2](#), [ref3](#)

oxygen 406, 528–34

P

paedophilia [ref1](#), [ref2](#)

Page, Larry [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)

Palestine-Israel conflict [ref1](#), [ref2](#), [ref3](#)

pandemic, definition of [ref1](#)

pandemic and health crisis scenarios/simulations [ref1](#), [ref2](#), [ref3](#), [ref4](#)

paranormal [ref1](#)

PCR tests [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)

Pearl Harbor attacks, prior knowledge of [ref1](#)

Pelosi, Nancy [ref1](#), [ref2](#), [ref3](#)

perception [ref1](#), [ref2](#), [ref3](#), [ref4](#)

- climate change hoax [ref1](#)
- control [ref1](#), [ref2](#), [ref3](#)
- decoding [ref1](#), [ref2](#)
- enslavement [ref1](#)
- externally-delivered perceptions [ref1](#)
- five senses [ref1](#)
- human labels [ref1](#)
- media [ref1](#), [ref2](#)
- political parties [ref1](#), [ref2](#)
- Psyop (psychological operation), Covid as a [ref1](#)
- sale of perception [ref1](#)
- self-identity [ref1](#), [ref2](#)
- Wokeness [ref1](#)

Phantom Self [ref1](#), [ref2](#), [ref3](#)

pharmaceutical industry *see* **Big Pharma**

phthalates [ref1](#)

Plato's Allegory of the Cave [ref1](#), [ref2](#)

pneuma [ref1](#)

police

- Black Lives Matter (BLM) [ref1](#)
- brutality [ref1](#)
- citizen's arrests [ref1](#), [ref2](#)
- common law arrests [ref1](#), [ref2](#)

Common Purpose ref1
defunding ref1
lockdowns ref1, ref2
masks ref1, ref2, ref3, ref4
police-military state ref1, ref2, ref3
psychopathic personality ref1, ref2, ref3, ref4
reframing ref1
United States ref1, ref2, ref3, ref4
Wokeness ref1

polio ref1

political correctness ref1, ref2, ref3, ref4

political parties ref1, ref2, ref3, ref4

political puppets ref1

pollution ref1, ref2, ref3

post-mortems/autopsies ref1

Postage Stamp Consensus ref1, ref2

pre-emptive programming ref1

Problem-Reaction-Solution ref1, ref2, ref3, ref4, ref5, ref6, ref7, ref8

Project for the New American Century ref1, ref2, ref3, ref4

psychopathic personality ref1

- Archons ref1
- heart energy ref1
- lockdowns ref1, ref2, ref3
- police ref1, ref2, ref3, ref4
- recruitment ref1, ref2
- vaccines ref1
- wealth ref1
- Wetiko ref1, ref2

Psyop (psychological operation), Covid as a ref1, ref2, ref3, ref4, ref5

Pushbackers ref1, ref2, ref3, ref4

pyramid structure ref1, ref2, ref3, ref4

Q

QAnon Psyop [ref1](#), [ref2](#), [ref3](#)

R

racism *see also* **Black Lives**

Matter (BLM)

anti-racism industry [ref1](#)

class [ref1](#)

critical race theory [ref1](#)

culture [ref1](#)

intersectionality [ref1](#)

reverse racism [ref1](#)

white privilege [ref1](#), [ref2](#)

white supremacy [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Wokeness [ref1](#), [ref2](#), [ref3](#)

radiation [ref1](#), [ref2](#)

randomness, illusion of [ref1](#), [ref2](#), [ref3](#)

reality [ref1](#), [ref2](#), [ref3](#)

reframing [ref1](#), [ref2](#)

change agents [ref1](#), [ref2](#)

children [ref1](#)

climate change [ref1](#)

Common Purpose leadership programme [ref1](#), [ref2](#)

contradictory rules [ref1](#)

enforcers [ref1](#)

masks [ref1](#), [ref2](#)

NLP and the Delphi technique [ref1](#)

police [ref1](#)

Wetiko factor [ref1](#)

Wokeness [ref1](#), [ref2](#)

religion *see also* particular religions

alien invasions [ref1](#)

Archons [ref1](#), [ref2](#)

consciousness [ref1](#), [ref2](#)

control, system of [ref1](#), [ref2](#), [ref3](#)

criticism, prohibition on [ref1](#)

five senses [ref1](#)

good and evil, war between [ref1](#)

hidden non-human forces [ref1](#), [ref2](#)

Sabbatians [ref1](#)

save me syndrome [ref1](#)

Wetiko [ref1](#)

Wokeness [ref1](#)

repetition and mind control [ref1](#), [ref2](#), [ref3](#)

reporting/snitching, encouragement of [ref1](#), [ref2](#)

Reptilians/Grey entities [ref1](#)

rewiring the mind [ref1](#)

Rivers, Thomas Milton [ref1](#), [ref2](#)

Rockefeller family [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#)

Rockefeller Foundation documents [ref1](#), [ref2](#), [ref3](#), [ref4](#)

Roman Empire [ref1](#)

Rothschild family [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#)

RT-PCR tests [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)

Russia

collusion inquiry in US [ref1](#)

Russian Revolution [ref1](#), [ref2](#)

Sabbatians [ref1](#)

§

Sabbantian-Frankism [ref1](#), [ref2](#)

anti-Semitism [ref1](#), [ref2](#)

banking and finance [ref1](#), [ref2](#), [ref3](#)

China [ref1](#), [ref2](#)

Israel [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Judaism [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Lucifer [ref1](#)

media [ref1](#), [ref2](#)

Nazis [ref1](#), [ref2](#)

QAnon [ref1](#)

Rothschilds [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)

Russia [ref1](#)

Saudi Arabia [ref1](#)

Silicon Valley [ref1](#)

Sumer [ref1](#)

United States [ref1](#), [ref2](#), [ref3](#)

Wetiko factor [ref1](#)

Wokeness [ref1](#), [ref2](#), [ref3](#)

SAGE (Scientific Advisory Group for Emergencies) [ref1](#), [ref2](#), [ref3](#),
[ref4](#)

SARS-1 [ref1](#)

SARs-CoV-2 [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)

Satan/Satanism [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)

satellites in low-orbit [ref1](#)

Saudi Arabia [ref1](#)

Save Me Syndrome [ref1](#)

scapegoating [ref1](#)

Schwab, Klaus [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#),
[ref11](#), [ref12](#)

science, manipulation of [ref1](#)

self-identity [ref1](#), [ref2](#), [ref3](#), [ref4](#)

self-respect, attacks on [ref1](#)

September 11, 2001, terrorist attacks on United States [ref1](#), [ref2](#),
[ref3](#), [ref4](#)

77th Brigade of UK military [ref1](#), [ref2](#), [ref3](#)

Silicon Valley/tech giants [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#) *see also*

Facebook

Israel [ref1](#)
Sabbatians [ref1](#)
technocracy [ref1](#)
Wetiko factor [ref1](#)
Wokeness [ref1](#)
simulation hypothesis [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
Smart Grid [ref1](#), [ref2](#), [ref3](#)
artificial intelligence (AI) [ref1](#)
China [ref1](#), [ref2](#)
control centres [ref1](#)
the Field [ref1](#)
Great Reset [ref1](#)
Human 2.0 [ref1](#), [ref2](#)
Israel [ref1](#), [ref2](#)
vaccines [ref1](#)
Wetiko factor [ref1](#)
social disapproval [ref1](#)
social distancing and isolation [ref1](#), [ref2](#), [ref3](#)
abusive relationships [ref1](#), [ref2](#)
children [ref1](#)
flats and apartments [ref1](#)
heart issues [ref1](#)
hugs [ref1](#)
Internet [ref1](#)
masks [ref1](#)
media [ref1](#)
older people [ref1](#), [ref2](#)
one-metre (three feet) rule [ref1](#)
rewiring the mind [ref1](#)
simulation, universe as a [ref1](#)
SPI-B [ref1](#)
substance abuse [ref1](#)

suicide and self-harm [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
technology [ref1](#)
torture, as [ref1](#), [ref2](#)
two-metre (six feet) rule [ref1](#)
women [ref1](#)

social justice [ref1](#), [ref2](#), [ref3](#), [ref4](#)

social media *see also Facebook bans on alternative views* [ref1](#)
censorship [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)
children [ref1](#)
emotion [ref1](#)
perception [ref1](#)
private messages [ref1](#)
Twitter [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
Wetiko factor [ref1](#)
YouTube [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Soros, George [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)

Spain [ref1](#)

SPI-B (Scientific Pandemic Insights Group on Behaviours) [ref1](#),
[ref2](#), [ref3](#), [ref4](#)

spider and the web [ref1](#), [ref2](#), [ref3](#), [ref4](#)

Starmer, Keir [ref1](#)

Statute Law [ref1](#)

Steiner, Rudolf [ref1](#), [ref2](#), [ref3](#)

Stockholm syndrome [ref1](#)

streptomycin [ref1](#)

suicide and self-harm [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

Sumer [ref1](#), [ref2](#)

Sunstein, Cass [ref1](#), [ref2](#), [ref3](#)

swine flu (H1N1) [ref1](#), [ref2](#), [ref3](#)

synchronicity [ref1](#)

synthetic biology [ref1](#), [ref2](#), [ref3](#), [ref4](#)

synthetic meat [ref1](#), [ref2](#)

T

technology *see also* **artificial intelligence (AI); Internet; social media addiction** [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Archons [ref1](#), [ref2](#)
the cloud [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
cyber-operations [ref1](#)
cyberwarfare [ref1](#)
radiation [ref1](#), [ref2](#)
social distancing and isolation [ref1](#)
technocracy [ref1](#)

Tedros Adhanom Ghebreyesus [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#),
[ref8](#), [ref9](#), [ref10](#), [ref11](#), [ref12](#), [ref13](#)

telepathy [ref1](#)

Tenpenny, Sherri [ref1](#)

Tesla, Nikola [ref1](#)

testosterone levels, decrease in [ref1](#)

testing for Covid-19 [ref1](#), [ref2](#)
 anal swab tests [ref1](#)
 cancer [ref1](#)
 China [ref1](#), [ref2](#), [ref3](#)
 Corman-Drosten test [ref1](#), [ref2](#), [ref3](#), [ref4](#)
 death certificates [ref1](#), [ref2](#)
 fraudulent testing [ref1](#)
 genetic material, amplification of [ref1](#)
 Lateral Flow Device (LFD) [ref1](#)
 PCR tests [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)
 vaccines [ref1](#), [ref2](#), [ref3](#)

Thunberg, Greta [ref1](#), [ref2](#), [ref3](#)

Totalitarian Tiptoe [ref1](#), [ref2](#), [ref3](#), [ref4](#)

transgender persons
 activism [ref1](#)
 artificial wombs [ref1](#)

censorship [ref1](#)
child abuse [ref1](#), [ref2](#)
Human 2.0 [ref1](#), [ref2](#), [ref3](#)
Wokeness [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
women, deletion of rights and status of [ref1](#), [ref2](#)
young persons [ref1](#)
travel restrictions [ref1](#)
Trudeau, Justin [ref1](#), [ref2](#), [ref3](#)
Trump, Donald [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#),
[ref11](#)
Twitter [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)

U

UKColumn [ref1](#), [ref2](#)
United Nations (UN) [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#) *see also Agenda 21/Agenda 2030 (UN)*
United States [ref1](#), [ref2](#)
 American Revolution [ref1](#)
 borders [ref1](#), [ref2](#)
 Capitol Hill riot [ref1](#), [ref2](#)
 children [ref1](#)
 China [ref1](#), [ref2](#)
 CIA [ref1](#), [ref2](#)
 Daily Pass tracking system [ref1](#)
 demographics by immigration, changes in [ref1](#)
 Democrats [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
 election fraud [ref1](#)
 far-right domestic terrorists, pushbackers as [ref1](#)
 Federal Reserve [ref1](#)
 flu/respiratory diseases statistics [ref1](#)
 Global Cult [ref1](#), [ref2](#)
 hand sanitisers, FDA warnings on [ref1](#)

immigration, effects of illegal [ref1](#)
impeachment [ref1](#)
Israel [ref1](#), [ref2](#)
Judaism [ref1](#), [ref2](#), [ref3](#)
lockdown [ref1](#)
masks [ref1](#)
mass media [ref1](#), [ref2](#)
nursing homes [ref1](#)
Pentagon [ref1](#), [ref2](#), [ref3](#), [ref4](#)
police [ref1](#), [ref2](#), [ref3](#), [ref4](#)
pushbackers [ref1](#)
Republicans [ref1](#), [ref2](#)
borders [ref1](#), [ref2](#)
Democrats [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
Russia, inquiry into collusion with [ref1](#)
Sabbatians [ref1](#), [ref2](#), [ref3](#)
September 11, 2001, terrorist attacks [ref1](#), [ref2](#), [ref3](#), [ref4](#)
UFO sightings, release of information on [ref1](#)
vaccines [ref1](#)
white supremacy [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Woke Democrats [ref1](#), [ref2](#)

V

vaccines [ref1](#), [ref2](#), [ref3](#)
adverse reactions [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
Africa [ref1](#)
anaphylactic shock [ref1](#), [ref2](#), [ref3](#), [ref4](#)
animals [ref1](#), [ref2](#)
anti-vax movement [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
AstraZeneca/Oxford [ref1](#), [ref2](#), [ref3](#), [ref4](#)
autoimmune diseases, rise in [ref1](#), [ref2](#)
Big Pharma [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#)

bioweapon, as real [ref1](#), [ref2](#)
black and ethnic minority communities [ref1](#)
blood clots [ref1](#), [ref2](#)
Brain Computer Interface (BCI) [ref1](#)
care homes, deaths in [ref1](#)
censorship [ref1](#), [ref2](#), [ref3](#)
chief medical officers and scientific advisers, financial interests of
[ref1](#), [ref2](#)
children [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#)
China [ref1](#), [ref2](#)
clinical trials [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)
compensation [ref1](#)
compulsory vaccinations [ref1](#), [ref2](#), [ref3](#)
computer programs [ref1](#)
consciousness [ref1](#)
cover-ups [ref1](#)
creation before Covid [ref1](#)
cytokine storm [ref1](#)
deaths and illnesses caused by vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
definition [ref1](#)
developing countries [ref1](#)
digital tattoos [ref1](#)
DNA-manipulation [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#),
[ref10](#)
emergency approval [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
female infertility [ref1](#)
funding [ref1](#)
genetic suicide [ref1](#)
Global Cult [ref1](#)
heart chakras [ref1](#)
hesitancy [ref1](#)
Human 2.0 [ref1](#), [ref2](#), [ref3](#), [ref4](#)
immunity from prosecution [ref1](#), [ref2](#), [ref3](#)

implantable technology [ref1](#)
Israel [ref1](#)
Johnson & Johnson [ref1](#), [ref2](#), [ref3](#), [ref4](#)
lockdowns [ref1](#)
long-term effects [ref1](#)
mainstream media [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
masks [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
Medicines and Healthcare products Regulatory Agency (MHRA)
[ref1](#), [ref2](#)
messaging [ref1](#)
Moderna [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)
mRNA vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#)
nanotechnology [ref1](#), [ref2](#)
NHS [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
older people [ref1](#), [ref2](#)
operating system [ref1](#)
passports [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Pfizer/BioNTech [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
polyethylene glycol [ref1](#)
pregnant women [ref1](#)
psychopathic personality [ref1](#)
races, targeting different [ref1](#)
reverse transcription [ref1](#)
Smart Grid [ref1](#)
social distancing [ref1](#)
social media [ref1](#)
sterility [ref1](#)
synthetic material, introduction of [ref1](#)
tests [ref1](#), [ref2](#), [ref3](#)
travel restrictions [ref1](#)
variants [ref1](#), [ref2](#)
viruses, existence of [ref1](#)
whistle-blowing [ref1](#)

WHO [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Wokeness [ref1](#)
working, vaccine as [ref1](#)
young people [ref1](#)
Vallance, Patrick [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#)
variants [ref1](#), [ref2](#), [ref3](#)
vegans [ref1](#)
ventilators [ref1](#), [ref2](#)
virology [ref1](#), [ref2](#)
virtual reality [ref1](#), [ref2](#), [ref3](#)
viruses, existence of [ref1](#)
visual reality [ref1](#), [ref2](#)
vitamin D [ref1](#), [ref2](#)
von Braun, Wernher [ref1](#), [ref2](#)

W

war-zone hospital myths [ref1](#)
waveforms [ref1](#), [ref2](#)
wealth [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#) [ref10](#), [ref11](#)
wet market conspiracy [ref1](#)
Wetiko factor [ref1](#)
alcoholism and drug addiction [ref1](#)
anti-human, why Global Cult is [ref1](#)
Archons [ref1](#), [ref2](#), [ref3](#), [ref4](#)
artificial intelligence (AI) [ref1](#)
Big Pharma [ref1](#), [ref2](#)
children [ref1](#)
China [ref1](#)
consciousness [ref1](#), [ref2](#)
education [ref1](#)
Facebook [ref1](#)

fear [ref1](#), [ref2](#)
frequency [ref1](#), [ref2](#)
Gates [ref1](#), [ref2](#)
Global Cult [ref1](#), [ref2](#)
heart [ref1](#), [ref2](#)
lockdowns [ref1](#)
masks [ref1](#)
Native American concept [ref1](#)
psychopathic personality [ref1](#), [ref2](#)
reframing/retraining programmes [ref1](#)
religion [ref1](#)
Silicon Valley [ref1](#)
Smart Grid [ref1](#)
smartphone addiction [ref1](#), [ref2](#)
social media [ref1](#)
war [ref1](#), [ref2](#)
WHO [ref1](#)
Wokeness [ref1](#), [ref2](#), [ref3](#)
Yaldabaoth [ref1](#), [ref2](#), [ref3](#), [ref4](#)
whistle-blowing [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
white privilege [ref1](#), [ref2](#)
white supremacy [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
Whitty, Christopher [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#), [ref10](#)
'who benefits' [ref1](#)
Wi-Fi [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Wikipedia [ref1](#), [ref2](#)
Wojcicki, Susan [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#)
Wokeness
 Antifa [ref1](#), [ref2](#), [ref3](#), [ref4](#)
 anti-Semitism [ref1](#)
 billionaire social justice warriors [ref1](#), [ref2](#), [ref3](#)

Capitol Hill riot [ref1](#), [ref2](#)
censorship [ref1](#)
Christianity [ref1](#)
climate change hoax [ref1](#), [ref2](#)
culture [ref1](#)
education, control of [ref1](#)
emotion [ref1](#)
facts [ref1](#)
fascism [ref1](#), [ref2](#), [ref3](#)
Global Cult [ref1](#), [ref2](#), [ref3](#), [ref4](#)
group-think [ref1](#)
immigration [ref1](#)
indigenous people, solidarity with [ref1](#)
inversion [ref1](#), [ref2](#), [ref3](#)
left, hijacking the [ref1](#), [ref2](#)
Marxism [ref1](#), [ref2](#), [ref3](#)
mind control [ref1](#)
New Woke [ref1](#)
Old Woke [ref1](#)
Oneness [ref1](#)
perceptual programming [ref1](#)
 Phantom Self [ref1](#)
police [ref1](#)
defunding the [ref1](#)
reframing [ref1](#)
public institutions [ref1](#)
Pushbackers [ref1](#), [ref2](#), [ref3](#)
racism [ref1](#), [ref2](#), [ref3](#)
reframing [ref1](#), [ref2](#)
religion, as [ref1](#)
Sabbatians [ref1](#), [ref2](#), [ref3](#)
Silicon Valley [ref1](#)
social justice [ref1](#), [ref2](#), [ref3](#), [ref4](#)

transgender [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)
United States [ref1](#), [ref2](#)
vaccines [ref1](#)
Wetiko factor [ref1](#), [ref2](#), [ref3](#)
young people [ref1](#), [ref2](#), [ref3](#)
women, deletion of rights and status of [ref1](#), [ref2](#)
World Economic Forum (WEF) [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#),
[ref8](#), [ref9](#)
World Health Organization (WHO) [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#),
[ref7](#), [ref8](#), [ref9](#)
AIDs/HIV [ref1](#)
amplification cycles [ref1](#)
Big Pharma [ref1](#), [ref2](#), [ref3](#)
cooperation in health emergencies [ref1](#)
creation [ref1](#), [ref2](#)
fatality rate [ref1](#)
funding [ref1](#), [ref2](#), [ref3](#)
Gates [ref1](#)
Internet [ref1](#)
lockdown [ref1](#)
vaccines [ref1](#), [ref2](#), [ref3](#), [ref4](#)
Wetiko factor [ref1](#)
world number 1 (masses) [ref1](#), [ref2](#)
world number 2 [ref1](#)
Wuhan [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#) [ref8](#)

Y

Yaldabaoth [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#)
Yeadon, Michael [ref1](#), [ref2](#), [ref3](#), [ref4](#)
young people *see also children* addiction to technology [ref1](#)
Human 2.0 [ref1](#)
vaccines [ref1](#), [ref2](#)

Wokeness [ref1](#), [ref2](#), [ref3](#)

YouTube [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#)

WHO 548

Z

Zaks, Tal [ref1](#)

Zionism [ref1](#), [ref2](#), [ref3](#)

Zuckerberg, Mark [ref1](#), [ref2](#), [ref3](#), [ref4](#), [ref5](#), [ref6](#), [ref7](#), [ref8](#), [ref9](#),
[ref10](#), [ref11](#), [ref12](#)

Zulus [ref1](#)

ICKONIC

THE ALTERNATIVE

Ickonic is something that has been a dream of mine for the last 5 years. growing up around alternative information I have always had a natural interest in what is going on in the World and what could I do to make it better. Across the range of subjects and positions of influence occupied mainly by people who don't strive to make things better it's the Media that I have always found the most frustrating and fascinating. Mainly because if the Media did their Jobs properly then so much of the negative things happening in the World simply would not be able to happen, because they would be exposed within a heartbeat.

Free Press and the Opportunities that the internet could have given would mean that the Media are able to expose things like never before and hold people to account for their actions. As we all know there are 'Untouchables' that walk among us, people the Media simply won't touch, expose or investigate and that leads to the dark underworlds that infest the establishment the World over. Well I say enough, it's time for something different, a different kind of Media, where no one is off limits from exposing and investigating. All we're interested in at Ickonic is the truth of what is really going on in the World on whichever subject we're covering.

We hope you enjoy what we have created and take something away from the platform, we aim to deliver information that's informative and most importantly self-empowering, you're not a little person, you're part of something much bigger than that and its time we as a collective race began to understand that and look to the future as ours to take.

It's time...

Jaymie Icke - Founder Ickonic Alternative Media.

SIGN UP NOW AT ICKONIC.COM

DAVID ICKE

THE ANSWER



We live in extraordinary times with billions bewildered and seeking answers for what is happening. David Icke, the man who has been proved right again and again, has spent 30 years uncovering the truth behind world affairs and in a stream of previous books he predicted current events.

The Answer will change your every perception of life and the world and set you free of the illusions that control human society. There is nothing more vital for our collective freedom than humanity becoming aware of what is in this book.

Available now at davidicke.com.

THE **TRIGGER**

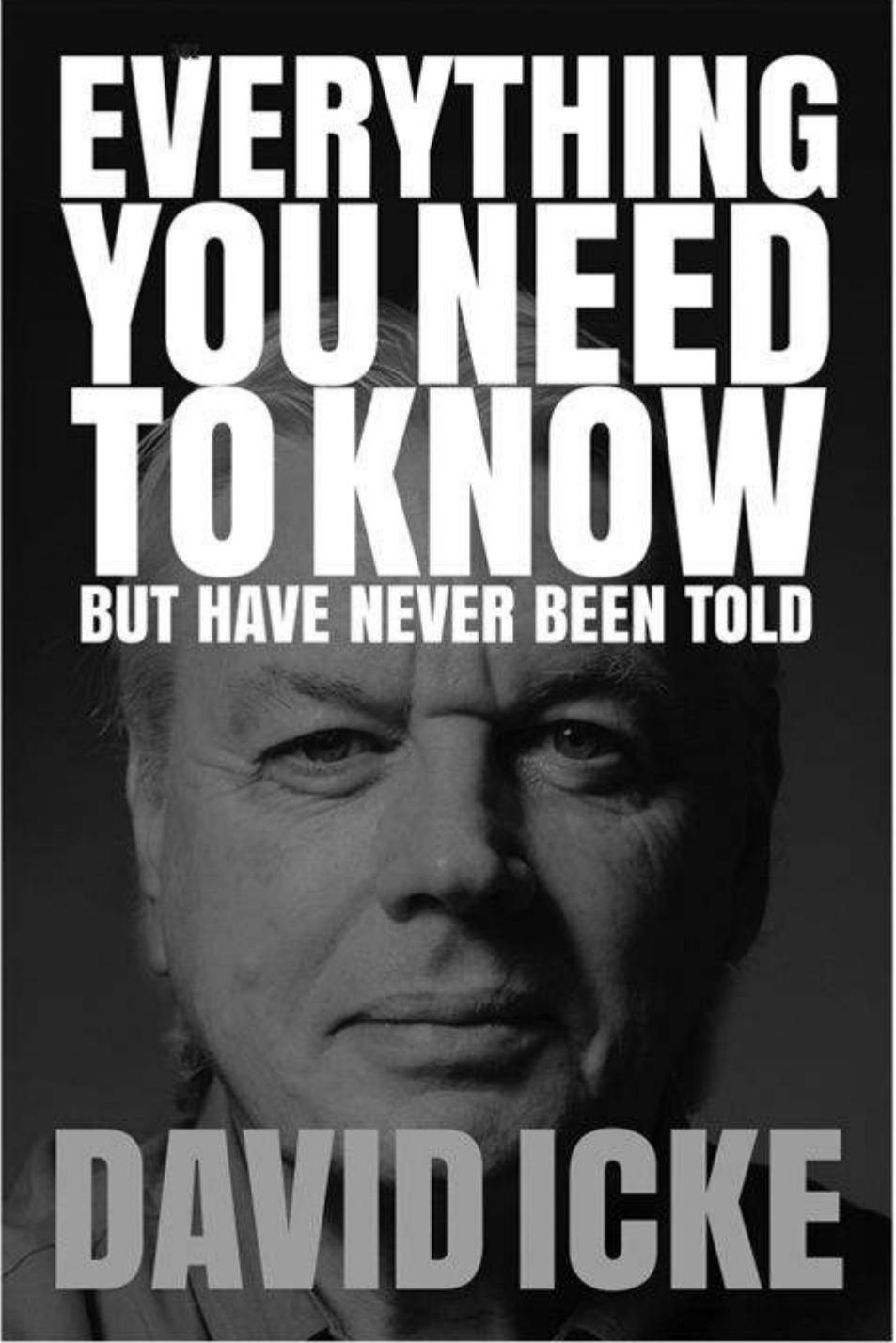
THE LIE THAT CHANGED THE WORLD
- WHO REALLY DID IT AND WHY



DAVID ICKE

EVERYTHING YOU NEED TO KNOW

BUT HAVE NEVER BEEN TOLD



DAVID ICKE

DAVIDICKE.COM



**DAVID ICKE STORE
LATEST NEWS ARTICLES
DAVID ICKE VIDEOS
WEEKLY DOT-CONNECTOR PODCASTS
LIVE EVENTS**
WWW.DAVIDICKE.COM

THE LIFE STORY OF DAVID ICKE

RENEGADE

/'ren-i.gəd/

noun

A person who behaves in a rebelliously unconventional manner.

THE FEATURE LENGTH FILM



AVAILABLE NOW AT DAVIDICKE.COM

2 NEW BOOKS
BY NEIL HAGUE

ORION'S DOOR

SYMBOLS OF CONSCIOUSNESS & BLUEPRINTS OF CONTROL
- THE STORY OF ORION'S INFLUENCE OVER HUMANITY

CUTTING EDGE VISIONARY ART
& UNIQUE ILLUSTRATED BOOKS

NEIL HAGUE

FOR
BOOKS, PRINTS & T-SHIRTS

VISIT:

NEILHAGUEBOOKS.COM

OR NEILHAGUE.COM

DR. COVID UNIVERSE

ADVENTURES
IN CLOWNLAND



NEIL HAGUE

Before you go ...

For more detail, background and evidence about the subjects in *Perceptions of a Renegade Mind* – and so much more – see my others books including *And The Truth Shall Set You Free; The Biggest Secret; Children of the Matrix; The David Icke Guide to the Global Conspiracy; Tales from the Time Loop; The Perception Deception; Remember Who You Are; Human Race Get Off Your Knees; Phantom Self; Everything You Need To Know But Have Never Been Told, The Trigger and The Answer.*

You can subscribe to the fantastic new Ickonic media platform where there are many hundreds of hours of cutting-edge information in videos, documentaries and series across a whole range of subjects which are added to every week. This includes my 90 minute breakdown of the week's news every Friday to explain *why* events are happening and to what end.