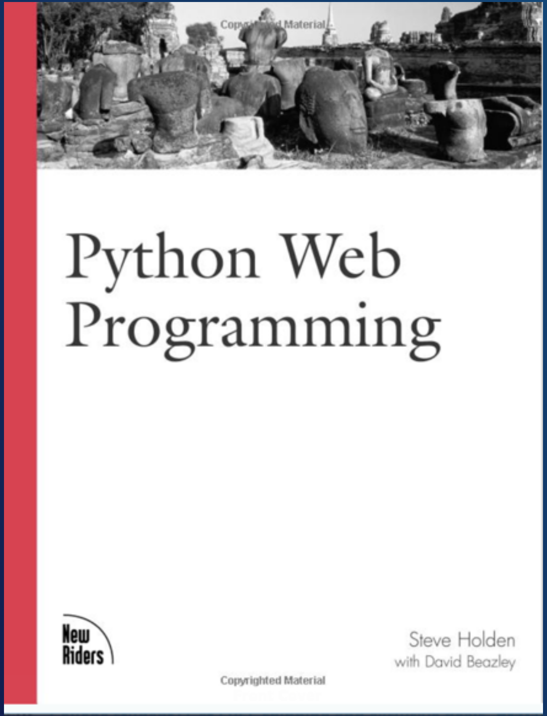
How Python is Winning New Friends

Steve Holden
CTO, Global Stress Index Limited
sh@felix.com

Introductions

- Programmer since 1967
- Computational scientist by training
- Engineer at heart
- Python user since Python 1.4 (c. 1995)
- Enjoy helping people to learn

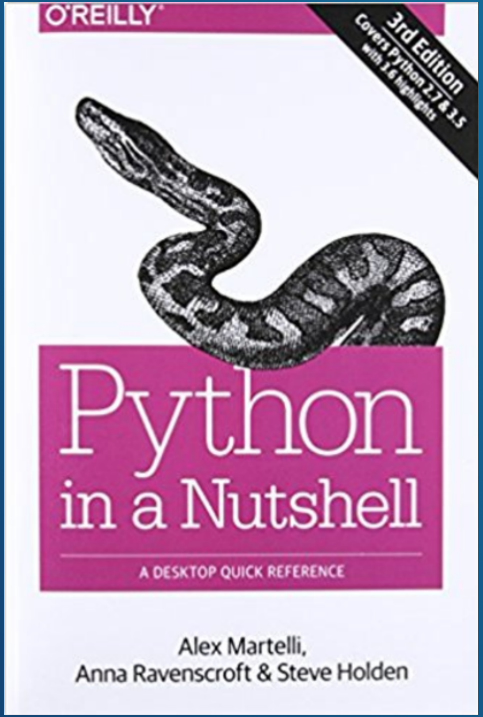
I’ve Written about Python



Python Web Programming

New Riders

Steve Holden
with David Beazley



O'REILLY

3rd Edition
Covers Python 2.7 & 3.5
with 3.6 highlights

Python in a Nutshell

A DESKTOP QUICK REFERENCE

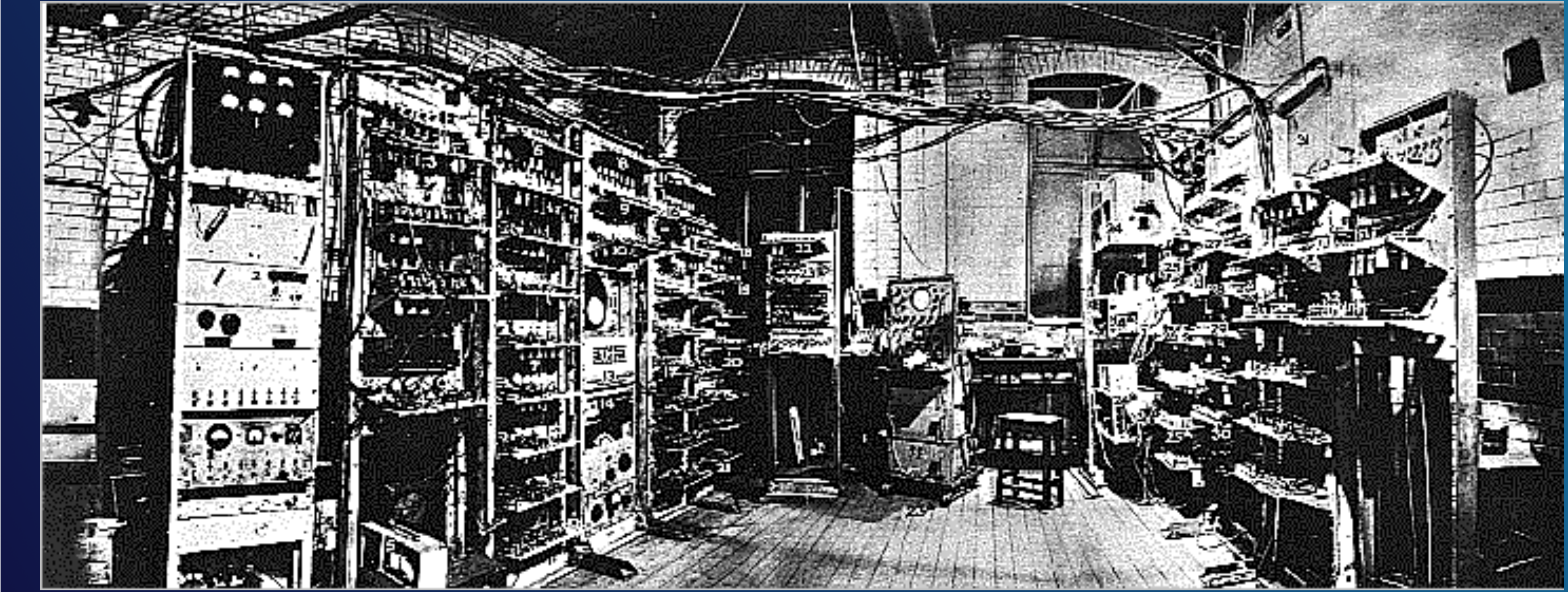
Alex Martelli,
Anna Ravenscroft & Steve Holden

Any Python users out there?

Developments in Computing

SOME HISTORY

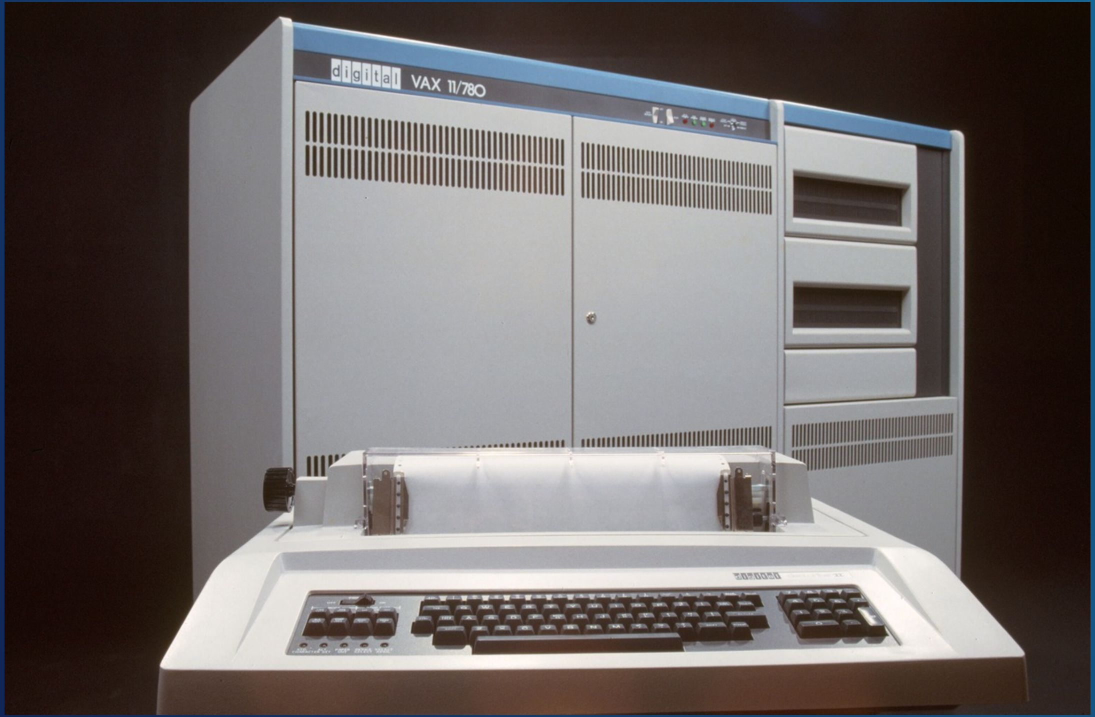
1948



Programming Was Hard

- No operating system
- No libraries
- No compilers
- No assemblers
- The painful process of abstraction layering began

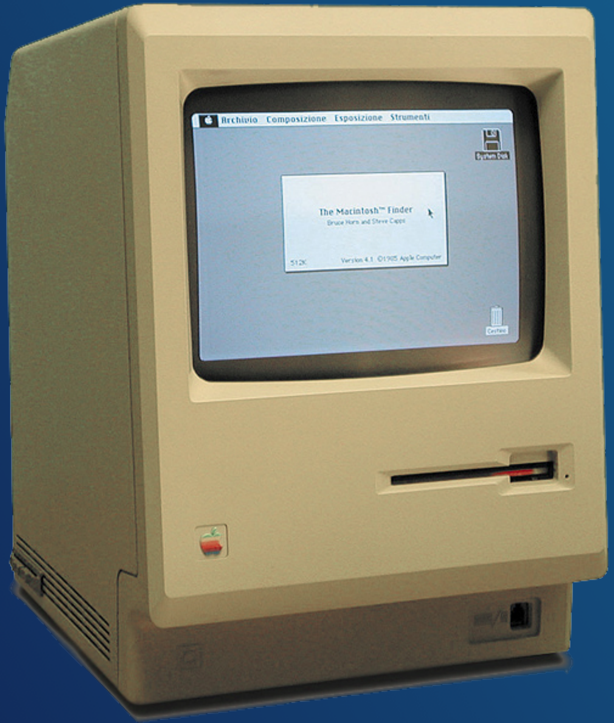
1977



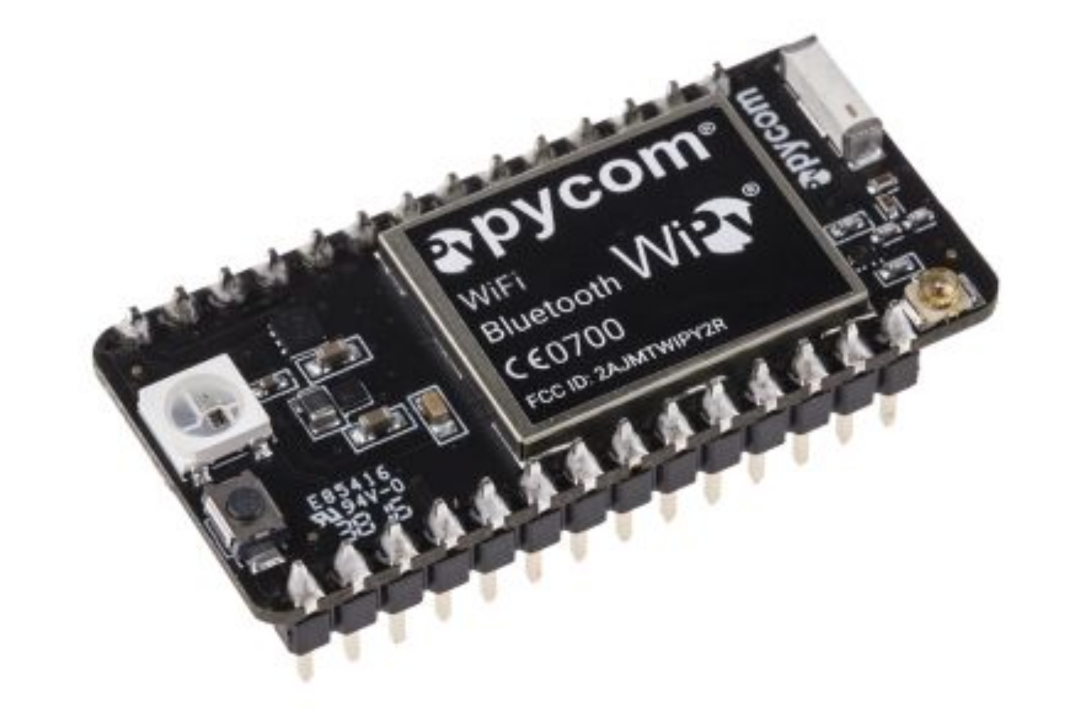
Easier to Program

- Assemblers/compilers available
- UNIX starting to emerge as a common base
  - Microprogramming handled hardware complexity
- Storage flexibly handled by the OS
- Networking heading to ubiquity

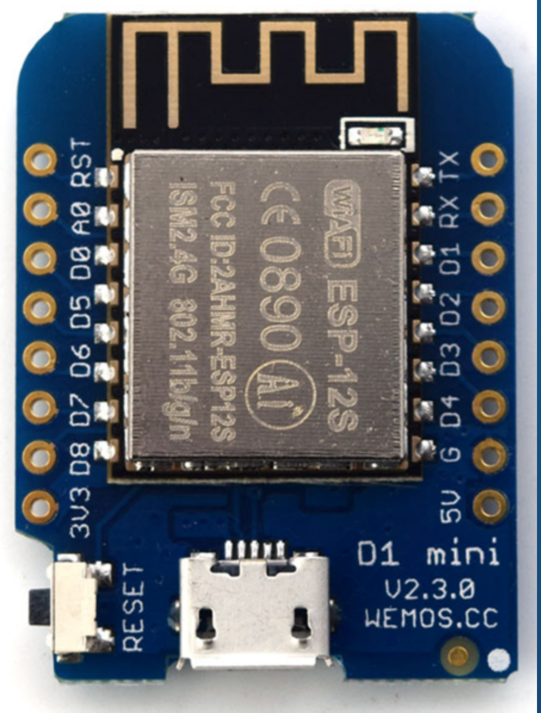
1984


Archivio
Composizione
Esposizione
Strumenti
System Disk
The Macintosh™ Finder
Bruce Horn and Steve Capps
512K
Version 4.1 ©1985 Apple Computer
Cestino

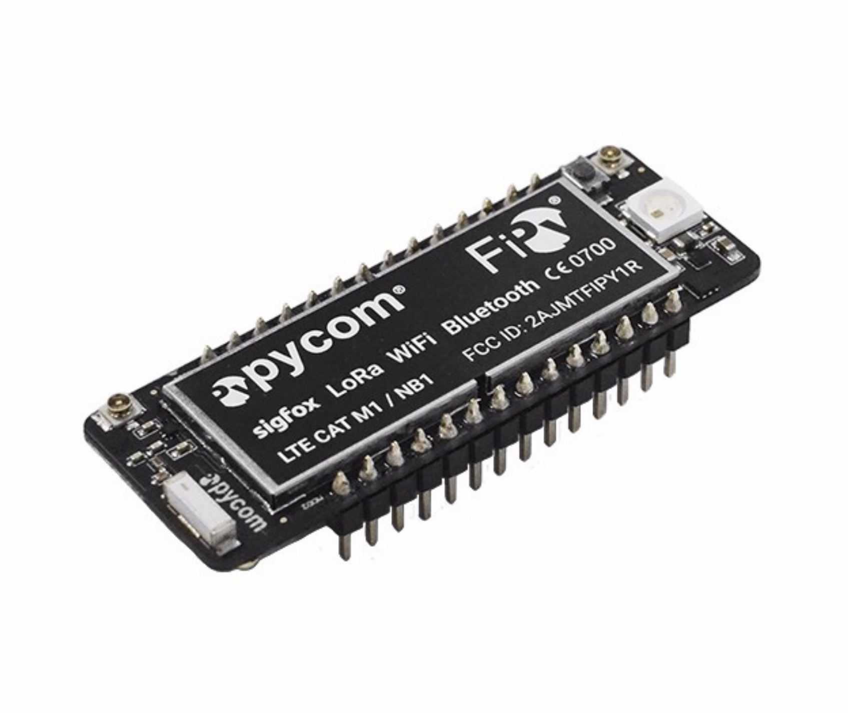
2015

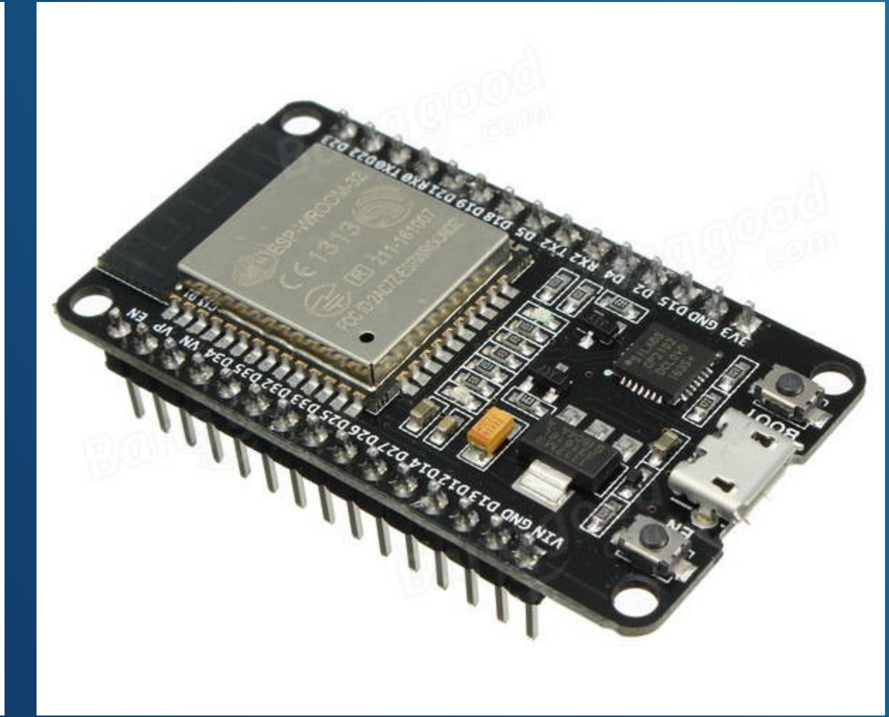

pycom
pycom
WiFi
Bluetooth
WiPy
CE0700
FCC ID: 2AJMTWIPY2R
E85416
94V-0

2016


RST
A0
D0
D5
D6
D7
D8
3V3
TX
RX
D1
D2
D3
D4
G
5V
WiFi
ESP-12S
CE 0890
FCC ID:2AHMR-ESP12S
ISM2.4G 802.11b/g/n
Ai
RESET
D1 mini
V2.3.0
WEMOS.CC

2017


pycom
sigfox LoRa WiFi Bluetooth
LTE CAT M1 / NB1
FiPy
CE 0700
FCC ID: 2AJMTFIPY1R


ESP-WROOM-32
EN
VP
VN
D34
D35
D32
D33
D25
D26
D27
D14
D12
D13
GND
VIN
D23
D22
TX0
RX0
D21
D19
D18
D5
TX2
RX2
D4
D2
D15
GND
3V3
BOOT
EN

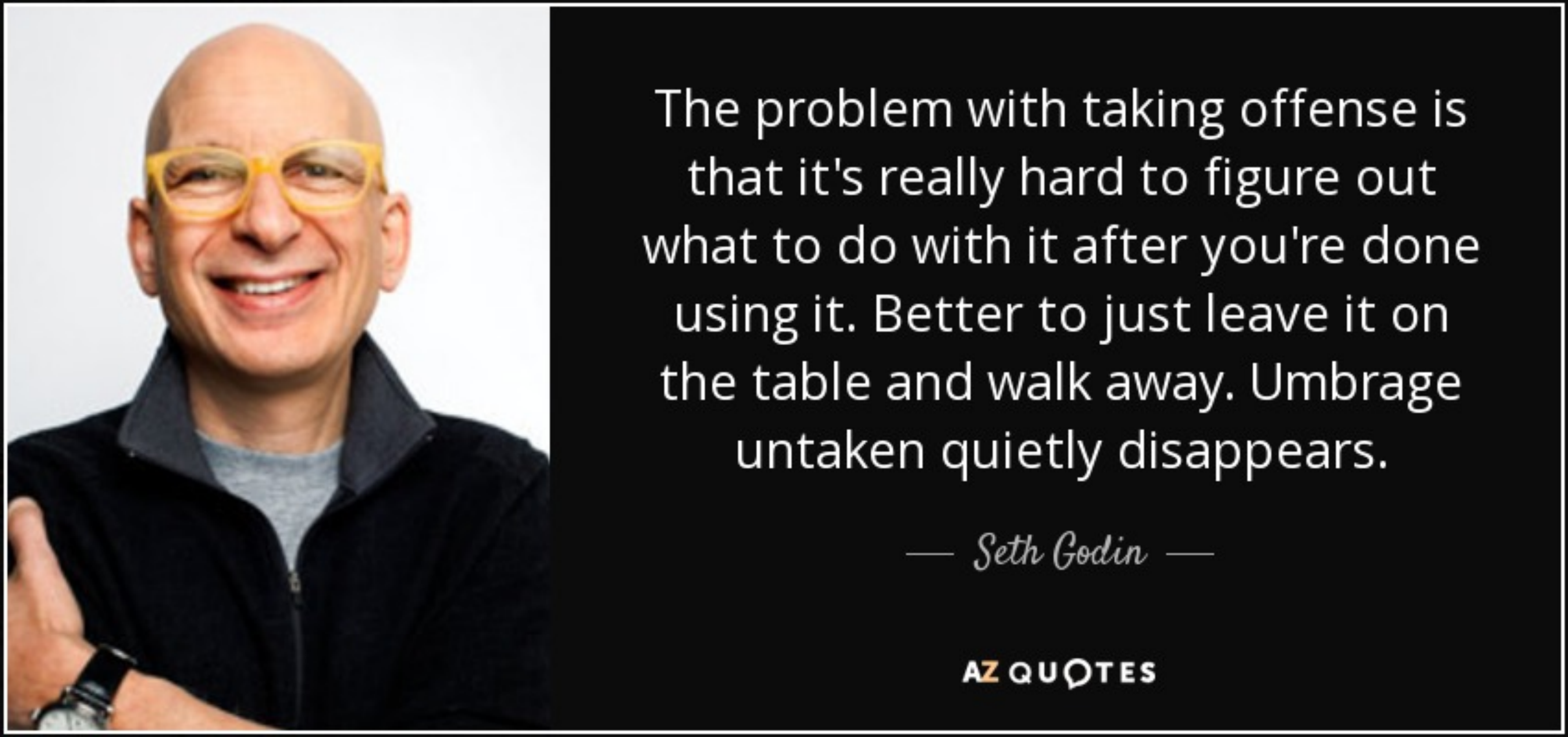
2020

?

Whatever it is, it will be complex!

And so to Python

“BUT IT’S [JUST] A SCRIPTING LANGUAGE …”



The problem with taking offense is that it's really hard to figure out what to do with it after you're done using it. Better to just leave it on the table and walk away. Umbrage untaken quietly disappears.

— Seth Godin —

What’s a “Scripting Language”?

- “First they ignore you; then they abuse you; then they crack down on you and then you win.” – ***not*** *Mahatma Ghandi*

What’s a “Scripting Language”?

- “First they ignore you; then they abuse you; then they crack down on you and then you win.” – ***not*** *Mahatma Ghandi*
- “Ridicule is like repression. Both give place to respect when they fail to produce the intended effect.” – *Mahatma Ghandi*

Note to Purists

- *Learners* do not have complex needs
  - Simplicity and consistency are important
  - Execution speed mostly isn't
- Direct hands-on experience *enables*
- Large resources not required
  - Wide availability and ease of access are critical

The Programming Audience

- Professional software engineers
- Scientists
- Lab technicians
- Teachers and students
- Self-guided learners
- Anyone who wants to control the billions of IoT devices
- …

Python’s Popularity

WHY DO PEOPLE USE PYTHON?

Easy for Beginners

- Simple Object Model
  - Abstracts memory allocation away
- *Everything* is an object
- Names are references to objects
  - Names live in *namespaces*
  - Objects live in the *heap*

Simple Assignment Semantics

- References keep objects alive
  - Object lifetime management is a non-problem
  - Dangling references therefore impossible
- Data is *never* copied on assignment
  - Python instead “binds names to values”

The REPL

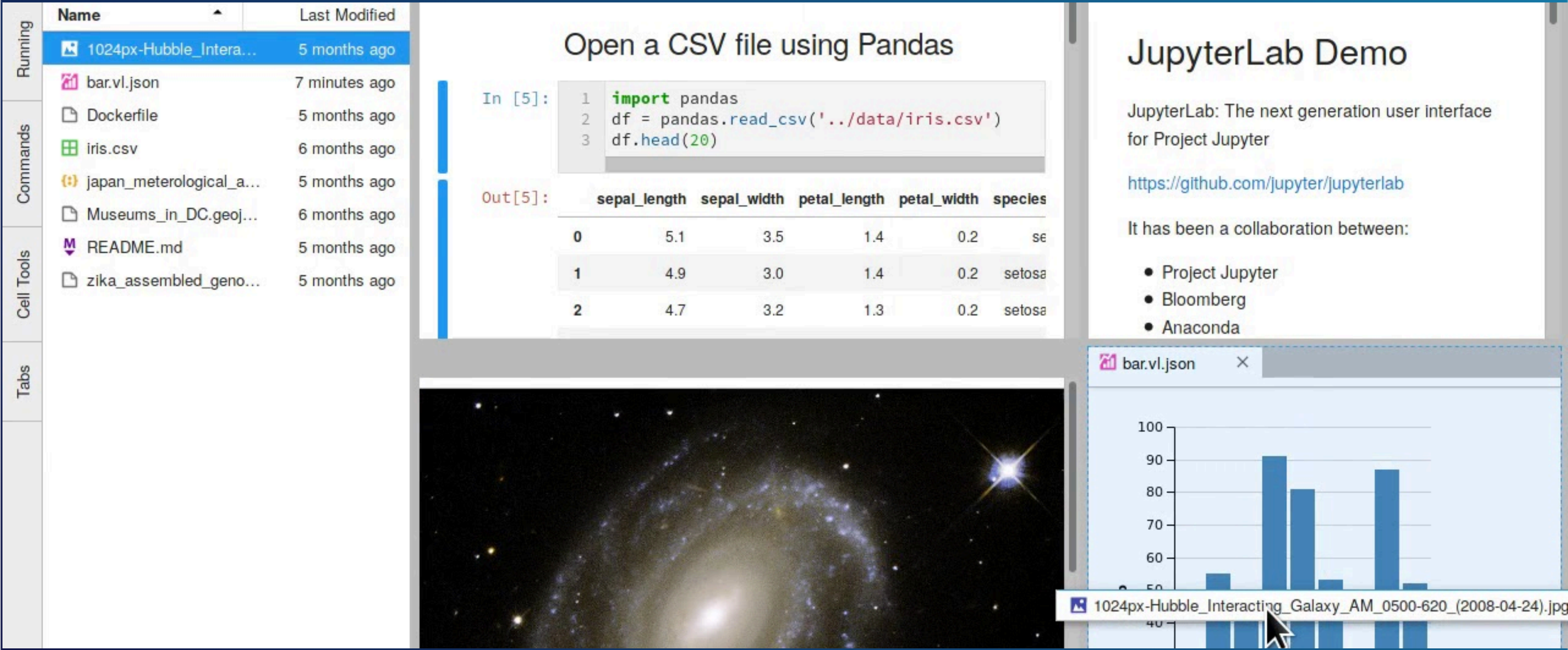
- Interactively manipulate objects – live!
- Allows *direct* learning
  - Answer your own questions authoritatively

The Ecosystem

HOW MANY PYTHONS?

Jupyter Notebook/Lab

- Heading towards “literate programming”
- Integrates graphical and other outputs with code and commentary in Markdown
- Great way to communicate executable code solutions



Name	Last Modified
1024px-Hubble_Intera…	5 months ago
bar.vl.json	7 minutes ago
Dockerfile	5 months ago
iris.csv	6 months ago
japan_meterological_a…	5 months ago
Museums_in_DC.geoj…	6 months ago
README.md	5 months ago
zika_assembled_geno…	5 months ago

Open a CSV file using Pandas

In [5]:

```
import pandas
df = pandas.read_csv('../data/iris.csv')
df.head(20)
```

Out[5]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	se
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa

JupyterLab Demo

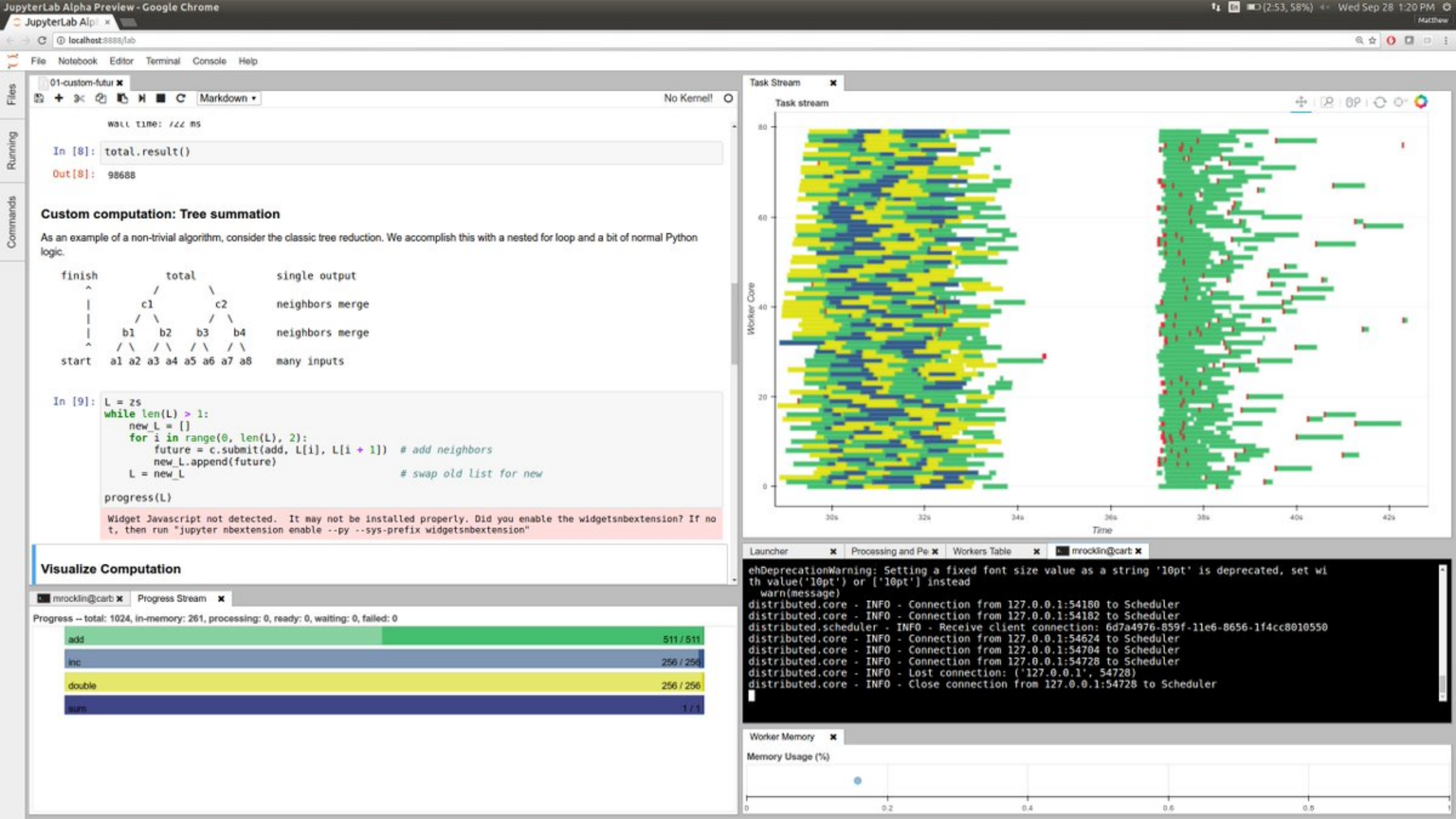
JupyterLab: The next generation user interface for Project Jupyter

https://github.com/jupyter/jupyterlab

It has been a collaboration between:

- Project Jupyter
- Bloomberg
- Anaconda

bar.vl.json
100
90
80
70
60
50
40
1024px-Hubble_Interacting_Galaxy_AM_0500-620_(2008-04-24).jpg



```
Wall time: 722 ms
```

```
In [8]: total.result()
```

```
Out[8]: 98688
```

Custom computation: Tree summation

As an example of a non-trivial algorithm, consider the classic tree reduction. We accomplish this with a nested for loop and a bit of normal Python logic.

```
finish           total             single output
   ^            /     \
   |        c1          c2         neighbors merge
   |       /  \        /  \
   |     b1    b2    b3    b4      neighbors merge
   ^    / \   / \   / \   / \
start  a1 a2 a3 a4 a5 a6 a7 a8     many inputs
```

```
In [9]: L = zs
        while len(L) > 1:
            new_L = []
            for i in range(0, len(L), 2):
                future = c.submit(add, L[i], L[i + 1])  # add neighbors
                new_L.append(future)
            L = new_L                                   # swap old list for new

        progress(L)
```

```
Widget Javascript not detected.  It may not be installed properly. Did you enable the widgetsnbextension? If no
t, then run "jupyter nbextension enable --py --sys-prefix widgetsnbextension"
```

Visualize Computation

Progress -- total: 1024, in-memory: 261, processing: 0, ready: 0, waiting: 0, failed: 0

```
ehDeprecationWarning: Setting a fixed font size value as a string '10pt' is deprecated, set wi
th value('10pt') or ['10pt'] instead
  warn(message)
distributed.core - INFO - Connection from 127.0.0.1:54180 to Scheduler
distributed.core - INFO - Connection from 127.0.0.1:54182 to Scheduler
distributed.scheduler - INFO - Receive client connection: 6d7a4976-859f-11e6-8656-1f4cc8010550
distributed.core - INFO - Connection from 127.0.0.1:54624 to Scheduler
distributed.core - INFO - Connection from 127.0.0.1:54704 to Scheduler
distributed.core - INFO - Connection from 127.0.0.1:54728 to Scheduler
distributed.core - INFO - Lost connection: ('127.0.0.1', 54728)
distributed.core - INFO - Close connection from 127.0.0.1:54728 to Scheduler
```

PyPy

- “Python written in Python”
- Implementation based on *Rpython*
  - Restricted, compilable language subset
- Gives C-like speeds on regular Python code
  - Retains Python-like clarity

Cython

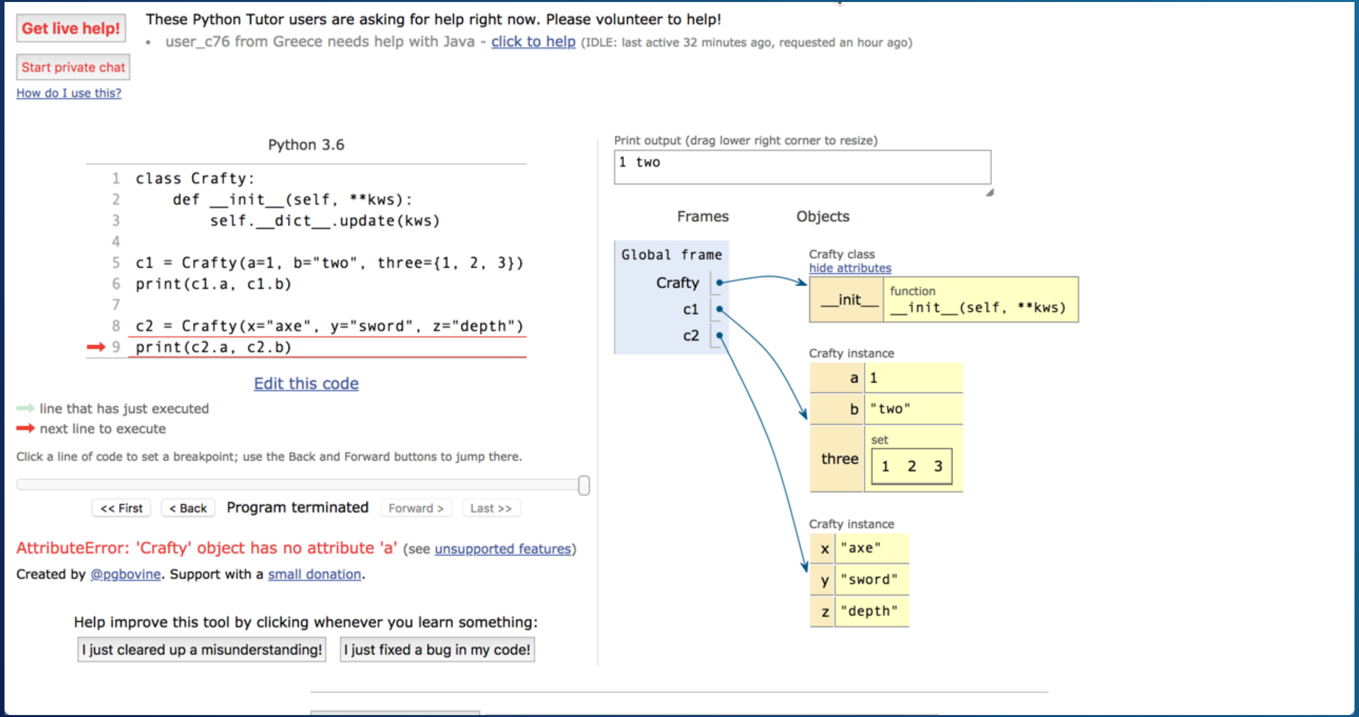
- Optimising static compiler
- Compiles Python (with C typing information) into C
- Great for wrapping existing C/C++ code in Python

MicroPython

- The *entire* Python 3.4 syntax, including
  - Exceptions
  - `with` , `yield from`, *etc.*
- Also adds 3.5’s `async` and `await`
- Optional machine code!

- Types include str, bytes, bytearray, tuple, list, dict, set, frozenset, array.array, collections.namedtuple
- Classes and instances
- And the REPL!

pythontutor.com



These Python Tutor users are asking for help right now. Please volunteer to help!

- user_c76 from Greece needs help with Java - click to help (IDLE: last active 32 minutes ago, requested an hour ago)

```
class Crafty:
    def __init__(self, **kws):
        self.__dict__.update(kws)

c1 = Crafty(a=1, b="two", three={1, 2, 3})
print(c1.a, c1.b)

c2 = Crafty(x="axe", y="sword", z="depth")
print(c2.a, c2.b)
```

line that has just executed

next line to execute

Click a line of code to set a breakpoint; use the Back and Forward buttons to jump there.

Program terminated

AttributeError: 'Crafty' object has no attribute 'a' (see unsupported features)

Created by @pgbovine. Support with a small donation.

Help improve this tool by clicking whenever you learn something:

Print output (drag lower right corner to resize)

```
1 two
```

Summary

MOST OF ALL

Python is FREE and FUN!

- Direct interaction with complex objects
- Ability to hook DIY classes into standard language syntax
- Easy for the motivated student to learn

Possibilities …

- Robot control
- Toys and games
- Weather stations
- Light patterns
- Science instrumentation/data collection
- Home automation

Final Thoughts

- Computers *don't* just belong in mathematics
  - Computer programming is ***not*** computer science

Final Thoughts

- Computers *don’t* just belong in mathematics
  - Computer programming is ***not*** computer science
- Python gives learners *direct, hands-on* experience
  - *Puts them in control*

Final Thoughts

- Computers *don’t* just belong in mathematics
  - Computer programming is ***not*** computer science
- Python gives learners *direct, hands-on* experience
  - *Puts them in control*
- Let people find their *own uses* for computers

Questions?

sh@felix.com

@holdenweb

Slides available (soon, promise) at

http://github.com/holdenweb/ACCU2018