

Python

A Problem Solving Kit

10 May 2021(Monday)

11-1pm

Alvin Alexander


Geotechnician @ JX Nippon

<https://github.com/elvinado/Problem-Solving-Kit>

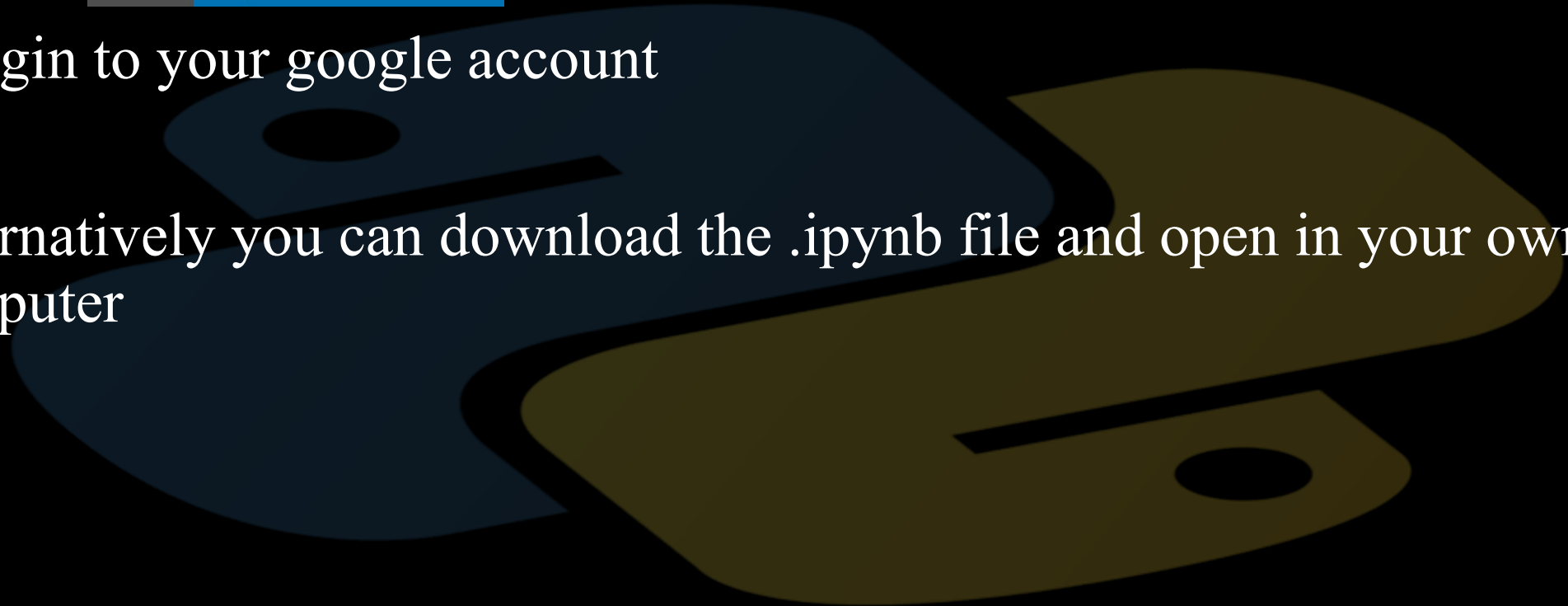
- Go to menti.com
- Use this code 8409 3511




Before we start

- Go to <http://bit.ly/3qq7iNU> (.ipynb file)
- Click  Open in Colab
- Login to your google account

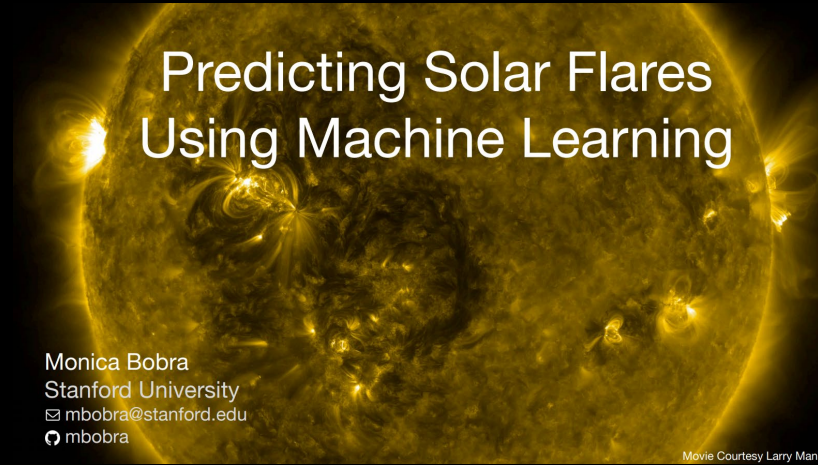
Alternatively you can download the .ipynb file and open in your own computer





Let's see how people use Python out there

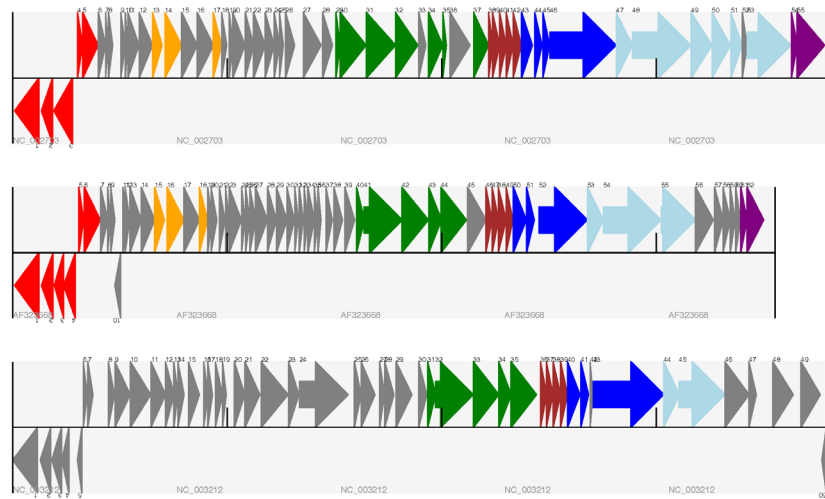
A to Z of Python Usage



Predicting Solar Flares Using
Machine Learning Methods by
Monica Bobra

Astrophysics

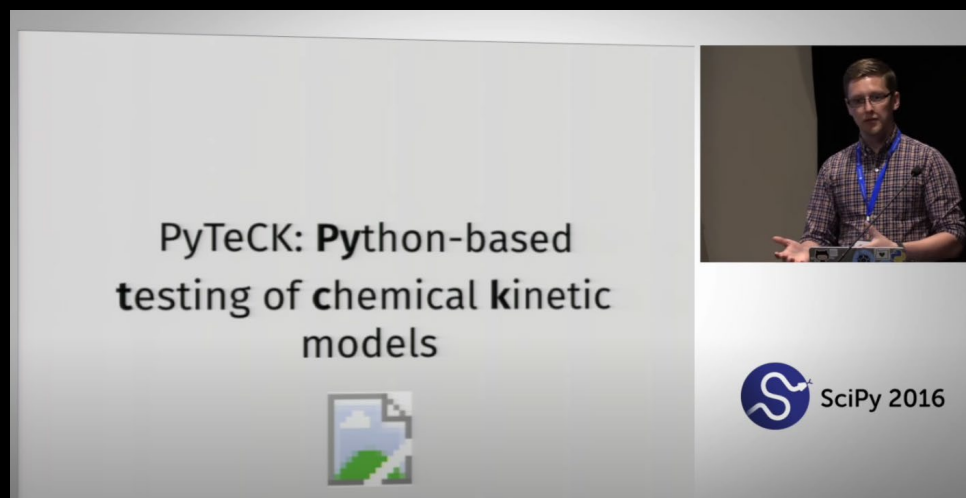
Notable Package – Astropy



Three phage
genomes diagram
side by side
(BioPython Tutorial)

Biology/Genomic

Notable Package – BioPython



PyTeCK: A Python-based automatic testing package for chemical kinetic models by Kyle Nie

Chemistry

Notable Packages – ChemPy, Chemlib, etc. check [Awesome Python Chemistry](#)

```
from tensorflow import keras
from tensorflow.keras import layers

# Instantiate a trained vision model
vision_model = keras.applications.ResNet50()

# This is our video encoding branch using the trained vision_model
video_input = keras.Input(shape=(100, None, None, 3))
encoded_frame_sequence = layers.TimeDistributed(vision_model)(video_input)
encoded_video = layers.LSTM(256)(encoded_frame_sequence)

# This is our text-processing branch for the question input
question_input = keras.Input(shape=(100,), dtype='int32')
embedded_question = layers.Embedding(10000, 256)(question_input)
encoded_question = layers.LSTM(256)(embedded_question)

# And this is our video question answering model:
merged = keras.layers.concatenate([encoded_video, encoded_question])
output = keras.layers.Dense(1000, activation='softmax')(merged)
video_qa_model = keras.Model(inputs=[video_input, question_input],
                              outputs=output)
```

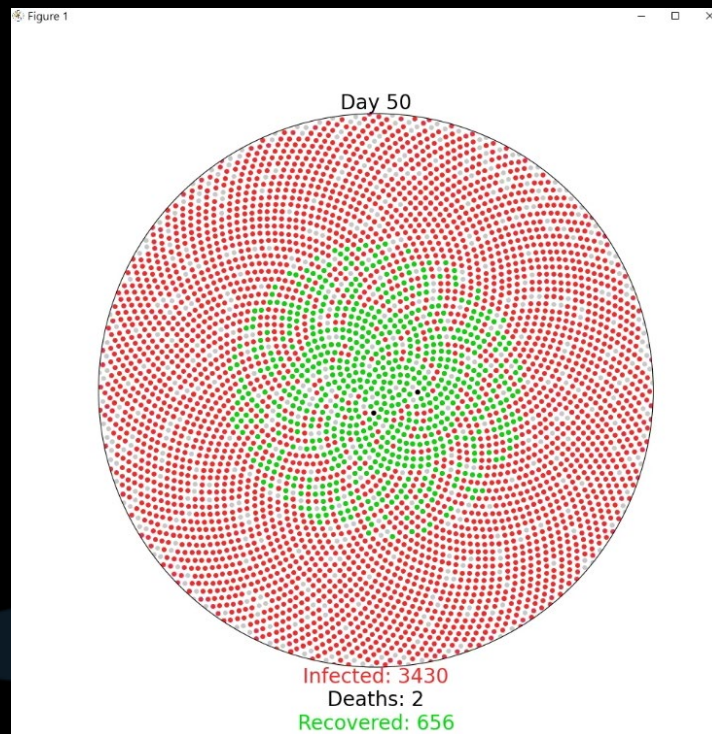
Deep learning for humans.

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow.

Deep Neural Network

Notable Packages – Keras, TensorFlow, PyTorch



Simulating Epidemic
using Python, NumPy
& Matplotlib Tutorial
by KITE youtube

Epidemiology

Notable Packages – NumPy, Matplotlib

pygame

pygame 827 2d 754 arcade 733 game 393 python 337 puzzle 333 shooter 264

strategy 253 action 213 other 151 libraries 150 space 150 simple 143 platformer 126

multiplayer 124 rpg 114 applications 92 retro 89 gpl 82 3d 81 pyopengl 73 snake 71

pyweek 71 geometrian 68 library 65 gui 61 engine 58 physics 57 simulation 53

adventure 47

Nannoid
A basic ball and paddle game with nice space graphics.

PGU - Phil's pyGame Utilities
A collection of handy modules and scripts for PyGame.

Astrocrash (Discontinued)
An arcade classic, a game of asteroids.

watermelons
pekuja, philhassey, and treeform got together on irc this evening and cranked out this whacky game about bouncing watermelons on a trampoline in about 4 hours. enjoy!

Blocker
A game of falling blocks.

Python Text Editor
A text editor in python.

OpenGL Fruit Machine
4-reel 3D Fruit Machine Demo. Tested on Windows and Linux.

pygame-blender converter
A script for incorporating Blender created graphics in Pygame applications.

Ants2d
Simple ant-road simulation.

Pygame is a set of Python modules designed for writing video games.

Fun

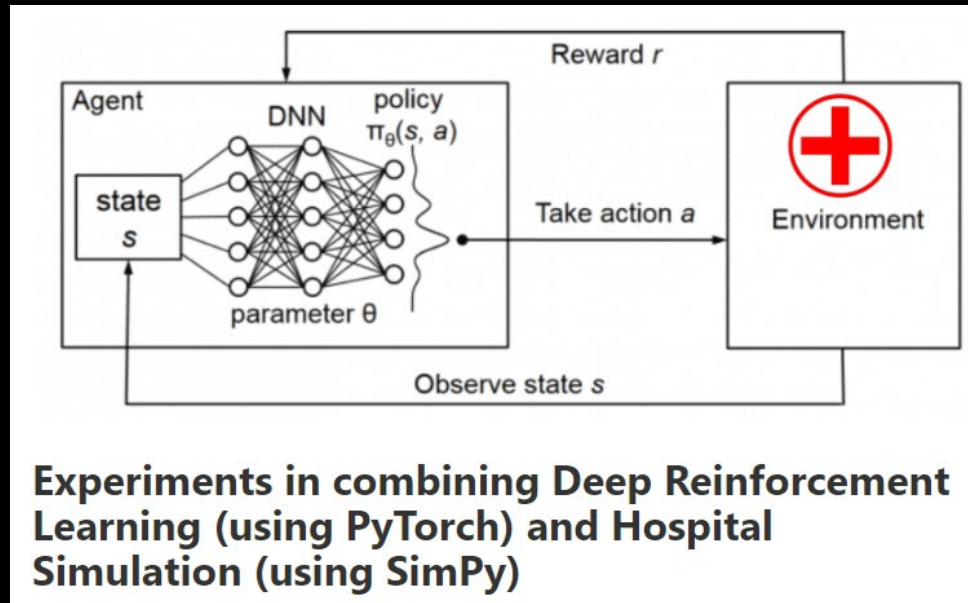
Notable Package – PyGame



Introduction to
Spatial Analysis in
Python with
Geopandas – Tutorial
By Hatari Lab

Geographic Information System

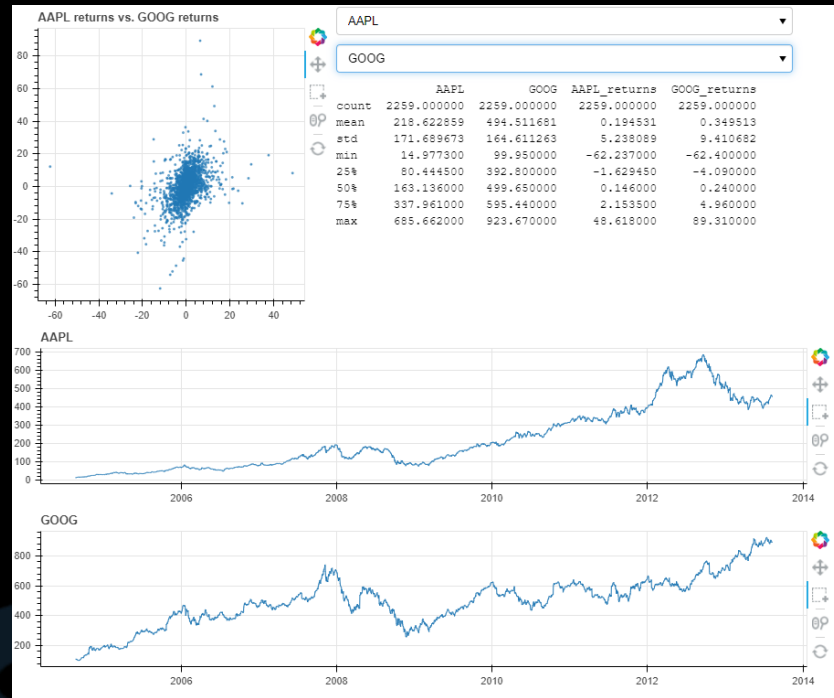
Notable Packages – GeoPandas



The Learning Hospital:
learn how to manage the
bed stock

Healthcare

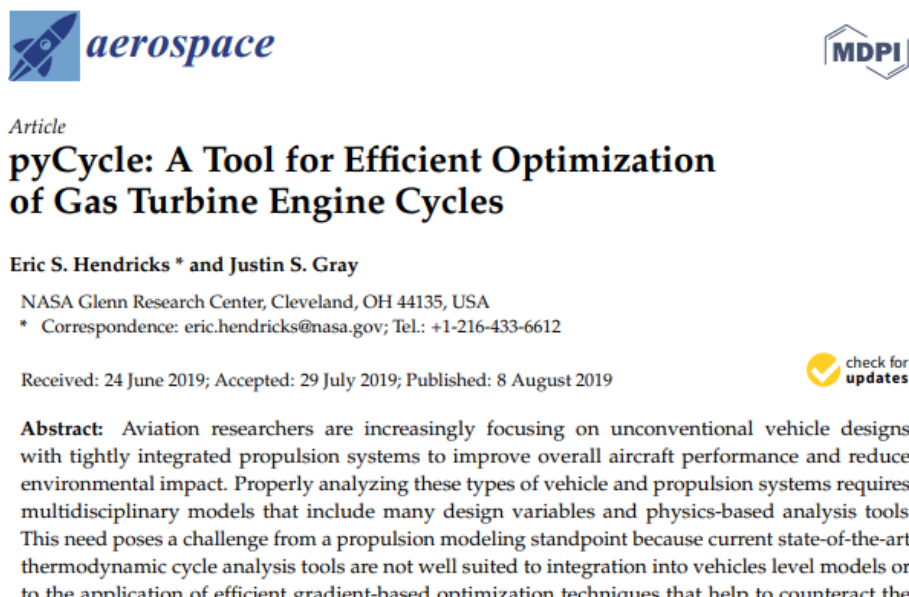
Notable Packages – Simpy (for simulation)



Linked plots,
summary statistics,
and correlations for
market data

Interactivity & Visualization

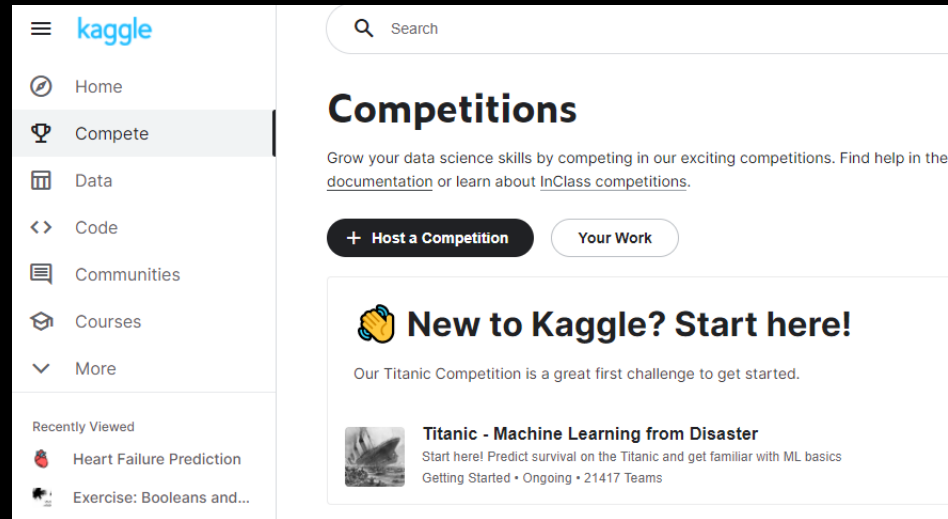
Notable Packages – Bokey, Plotly



Thermodynamic cycle modeling library, designed primarily to model jet engine performance

Jet

Notable Packages – pyCycle



Kaggle is online data science competition website

Kaggle

Notable Packages – Any



Legal


Notable Project – OpenEDGAR

OpenEDGAR: Open Source Software for SEC EDGAR Analysis

Legal instruments such as judicial opinions and contracts predominantly manifest in paper. This article explores OpenEDGAR, an open source tool that helps move law into a more computational direction by automating the process for retrieving, parsing, and indexing SEC Filings.

by Michael J. Bommarito II, Daniel Martin Katz, and Eric M. Detterman



6.3 Training and using a word2vec model from press releases

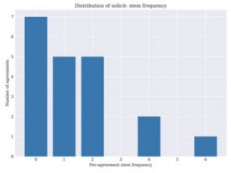
```
from lemmnlp.en.sentence import get_sentence_list
from lemmnlp.en.tokenize import get_token_list
from openedgar.modules import FilingDocument
from openedgar.clients.s3 import get_s3_client

# Use Django ORM to retrieve sample of agreements
sample_size = 1000
search_string = "press release"
release_id_list = FilingDocument.objects.filter(description__icontains=search_string)[:sample_size]

# Retrieve text contents of each agreement and track data
sentence_list = []
for release_id in release_id_list:
    release_path = "documents/text/{}".format(release_id)
    release_content = get_buffer(release_path, decoder="utf-8")

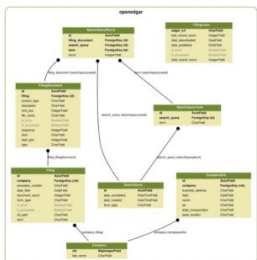
# Use Levenshtein to loop through sentences and build stopword sets
for sentence in get_sentence_list(release_content):
    stems = get_token_list(sentence, lowercase=True, stopwords=True)
    sentence_list.append(stems)

import gensim.models.word2vec
word2vec_model = gensim.models.word2vec.Word2Vec(sentence_list)
word2vec_model.wv.most_similar(positive=[DEFAULT_STOPWORD_STEM, "revenue"])
```



Rank	Stem	Similarity
1	segments	0.59705
2	sale	0.57048
3	profits	0.54328

Sample word2vec results for "revenue", see listing 6.2



MIT Computational Law Report

Automating the process for retrieving, parsing, and indexing SEC Filings

Welcome to NumPy!

NumPy (**N**umerical **P**ython) is an open source Python library that's used in almost every field of science and engineering. It's the universal standard for working with numerical data in Python, and it's at the core of the scientific Python and PyData ecosystems. NumPy users include everyone from beginning coders to experienced researchers doing state-of-the-art scientific and industrial research and development. The NumPy API is used extensively in Pandas, SciPy, Matplotlib, scikit-learn, scikit-image and most other data science and scientific Python packages.

The NumPy library contains multidimensional array and matrix data structures (you'll find more information about this in later sections). It provides **ndarray**, a homogeneous n-dimensional array object, with methods to efficiently operate on it. NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

NumPy: The
fundamental package
for scientific
computing with
Python

M

Mathematics

Notable Package – Numpy

DROPS OF JUPYTER NOTEBOOKS: HOW TO KEEP NOTES IN THE INFORMATION AGE

by: Al Williams

32 Comments



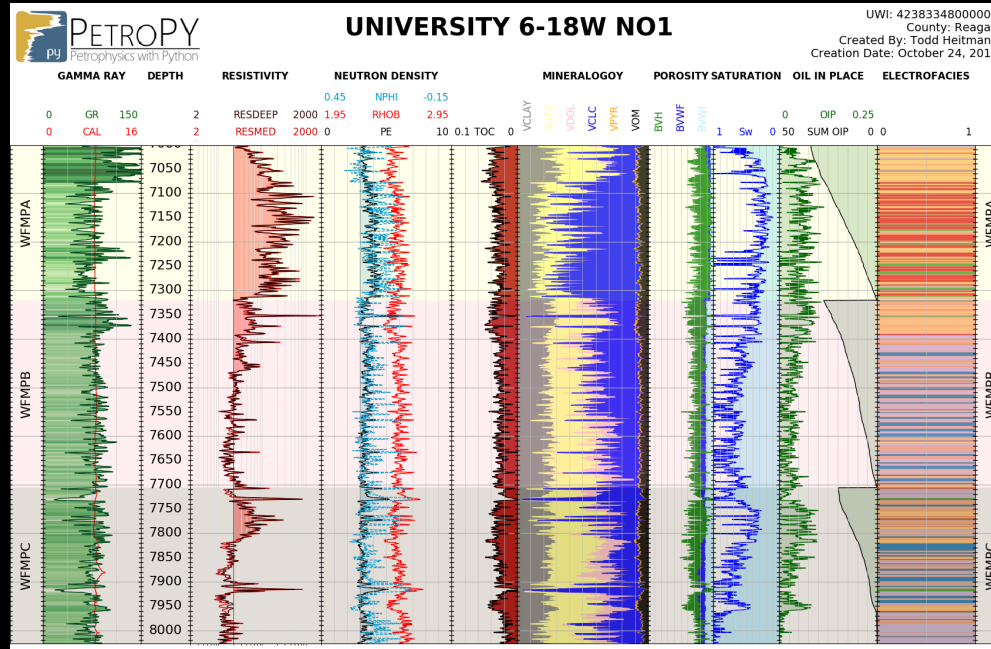
February 22, 2019



Keeping Notes using
Jupyter Notebook

Notebooks

Notable Package – Jupyter



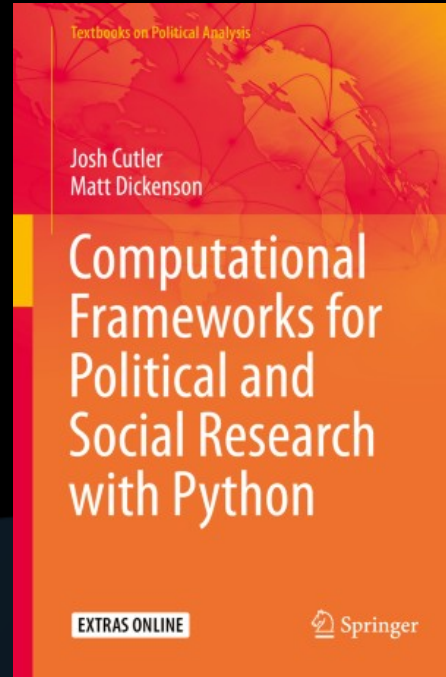
Scientific python
computing of
subsurface
formation evaluation

Oil & Gas

Notable Package – PetroPy, Lasio, Segyio

Politics

Notable Package – Python



Computational
Frameworks for
Political and Social
Research with Python

BUILDING QUANT EQUITY STRATEGIES IN PYTHON

PRESENTED BY DR. JESS STAUTH


- VP OF QUANT STRATEGY AT QUANTOPIAN
- FORMER QUANT RESEARCH ANALYST AT STARMINE
- FORMER DIRECTOR OF QUANT PRODUCT STRATEGY FOR THOMSON-REUTERS





Building Quant Equity
Strategies in Python
by Dr. Jess Stauth

Quantitative Finance

Notable Package – check [awesome-quant](https://github.com/awesome-quant)


SALIND

[About](#)
[Problems](#)
[Statistics](#)
[Glossary](#)

elvinad

Log out

Problems

Bioinformatics Stronghold

List

Tree

Rosalind is a platform for learning bioinformatics and programming through problem solving. [Take a tour](#) to get the hang of how Rosalind works.

Last win: [Jackiedduarte](#) vs. "Complementing a Strand of DNA", just now

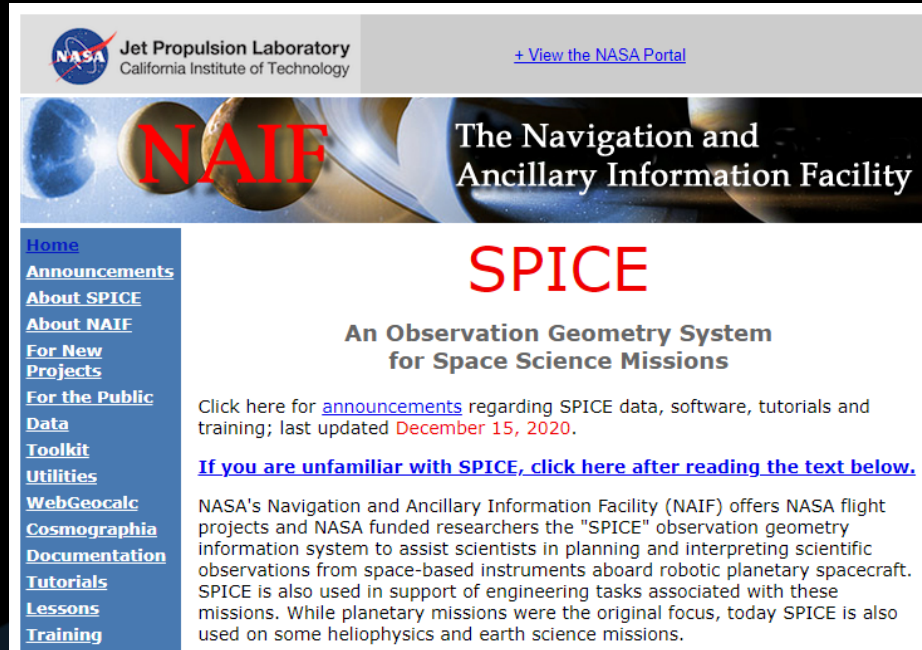
Problems: 285 (total), users: 83600, attempts: 1378093, correct: 762620

ID	Title	Solved By	Correct Ratio	Questions	Solutions	Explanation
DNA	Counting DNA Nucleotides	48311	<div><div></div></div>	<div><div></div></div> 5 months	<div><div></div></div> 3 days	<div><div></div></div> 7 months
RNA	Transcribing DNA into RNA	43116	<div><div></div></div>	<div><div></div></div> 1 year	<div><div></div></div> 5 days	<div><div></div></div> 1 year
REVC	Complementing a Strand of DNA	39022	<div><div></div></div>	<div><div></div></div> 9 months	<div><div></div></div> 3 days	<div><div></div></div> 4 years
FIB	Rabbits and Recurrence Relations	22540	<div><div></div></div>	<div><div></div></div> 4 weeks	<div><div></div></div> 1 week	<div><div></div></div> 2 years
GC	Computing GC Content	22657	<div><div></div></div>	<div><div></div></div> 2 months	<div><div></div></div> 2 weeks	<div><div></div></div> 7 years
HAMM	Counting Point Mutations	25510	<div><div></div></div>	<div><div></div></div> 1 month	<div><div></div></div> 1 month	<div><div></div></div> 4 years
IPRB	Mendel's First Law	15073	<div><div></div></div>	<div><div></div></div> 17 hours	<div><div></div></div> 17 hours	<div><div></div></div> 1 year
PROT	Translating RNA into Protein	19986	<div><div></div></div>			
SUBS	Finding a Motif in DNA	20304	<div><div></div></div>			
FIBD	Mortal Fibonacci Rabbits	9592	<div><div></div></div>			
IEV	Calculating Expected Offspring	8660	<div><div></div></div>			
LIA	Independent Alleles	4588	<div><div></div></div>			
PERM	Enumerating Gene Orders	10113	<div><div></div></div>			

Rosalind is a platform for learning bioinformatics through problem solving.

Rosalind.info

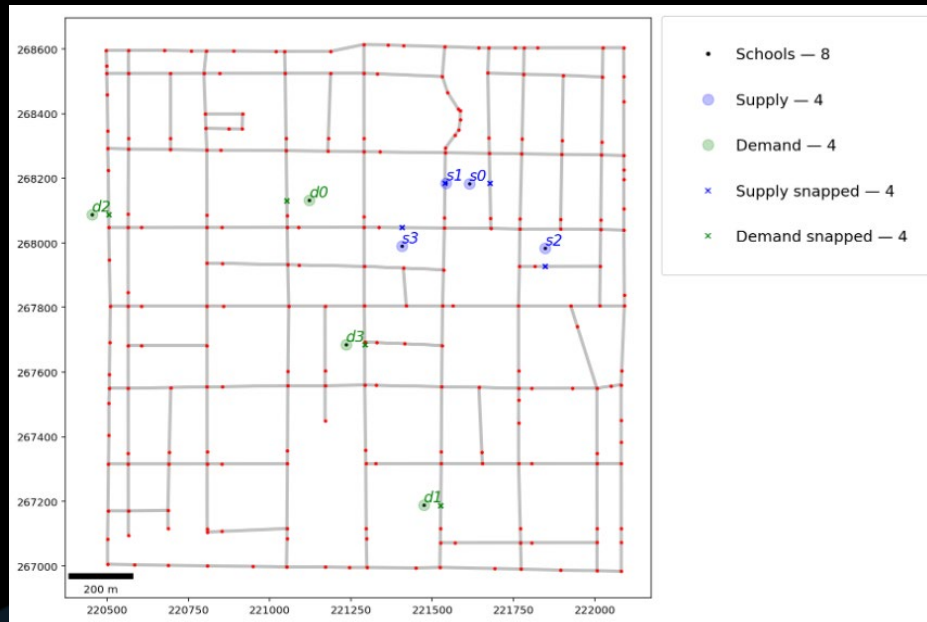
Useful Package – biopython



Planetary and spacecraft ephemeris (navigation) and ancillary engineering information

Space

Notable Package – SpiceyPy

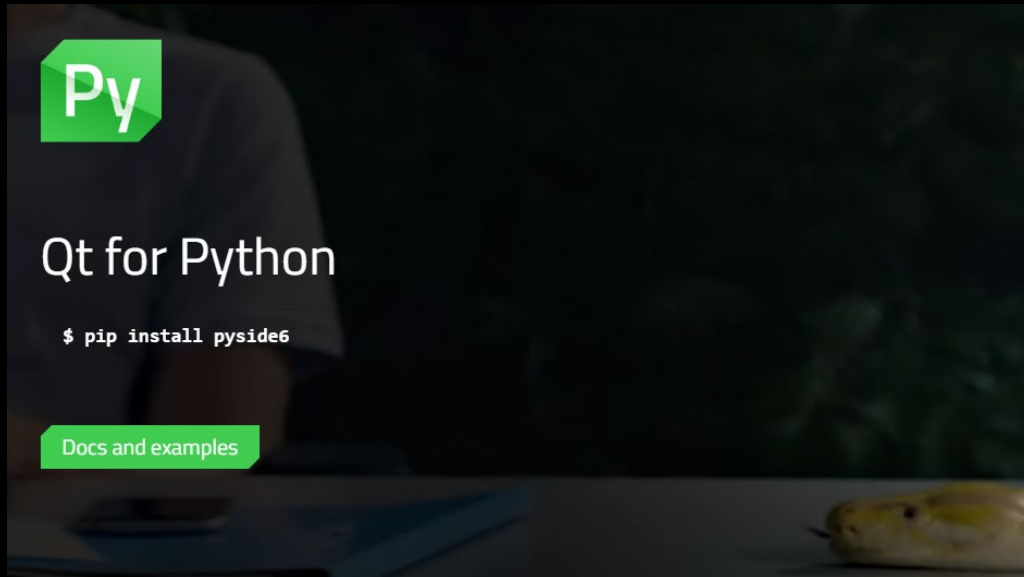


Allocate supply to demand while minimizing transportation costs

T

ransportation

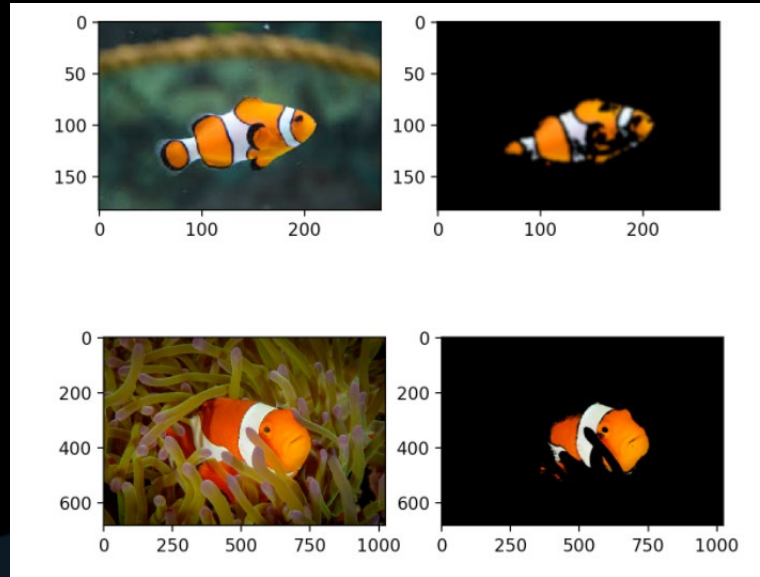
Notable Package — PySal



Qt for Python is the project that provides the official set of Python bindings

User Interface (UI)

Notable Packages – Qt, Tkinter, wxPython, PySimple GUI



Solving computer vision problem such as image segmentation

Vision

Notable Package – OpenCV, PyTorch


Beautiful Soup 4.9.0 documentation » Beautiful Soup Documentation index

Table of Contents

- Beautiful Soup Documentation
 - Getting help
- Quick Start
- Installing Beautiful Soup
 - Problems after installation
 - Installing a parser
- Making the soup
- Kinds of objects
 - Tag
 - Name
 - Attributes
 - Multi-valued attributes
 - NavigableString
 - BeautifulSoup
 - Comments and other special strings
- Navigating the tree
 - Going down
 - Navigating using tag names
 - .contents and .children
 - .descendants
 - .string
 - .strings and .stripped_strings

Beautiful Soup Documentation

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.



These instructions illustrate all major features of Beautiful Soup 4, with examples. I show you what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.

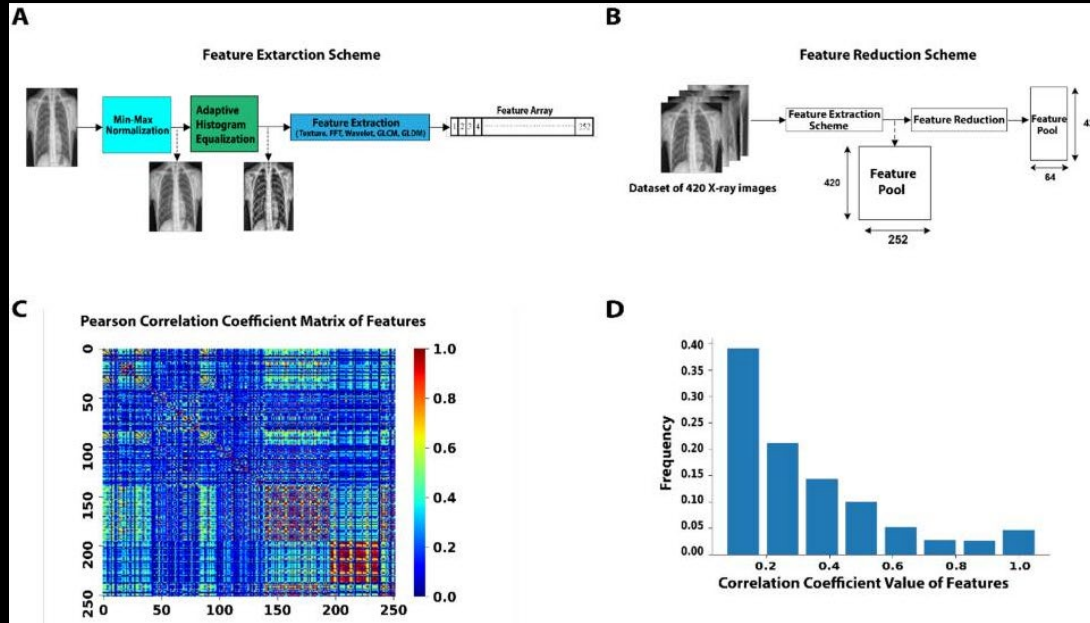
This document covers Beautiful Soup version 4.9.2. The examples in this documentation should work the same way in Python 2.7 and Python 3.8.

You might be looking for the documentation for Beautiful Soup 3. If so, you should know that Beautiful Soup 3 is no longer being developed and that support for it will be dropped on or after December 31, 2020. If you want to learn about the differences between Beautiful Soup 3 and Beautiful Soup 4, see [Porting code to BS4](#).

Beautiful Soup is a Python library for pulling data out of HTML and XML files.

Web Scraping

Notable Packages – BeautifulSoup, Selenium, Scrapy



COVID-Classifier: An automated machine learning model to assist in the diagnosis of COVID-19 infection in chest x-ray images

X-ray

Notable Packages – Numpy, Scipy, Sklearn, Keras

You?

Y

I just could not find anything starting with Y



The screenshot shows a web page from 'tutorialspoint' with the tagline 'SIMPLY EASY LEARNING'. The top navigation bar includes 'Library', 'Videos', 'Q/A', and 'eBooks'. The main title is 'Python program to display Astrological sign or Zodiac sign for a given data of birth.' Below the title are tags for 'Python', 'Server Side Programming', and 'Programming'. A small image of a person is visible. The 'Examples' section shows 'Input : Day = 13, Month = November' and 'Output : Scorpio.'. The 'Algorithm' section lists three steps: 'Step 1 : input date of birth.', 'Step 2 : checks month and date within the valid range of a specified zodiac.', and 'Step 3 : display zodiac sign.'. The 'Example Code' section is partially visible. The sidebar on the left has a 'Q/A' icon and a list of related questions under the heading 'Related Questions & Answers'.

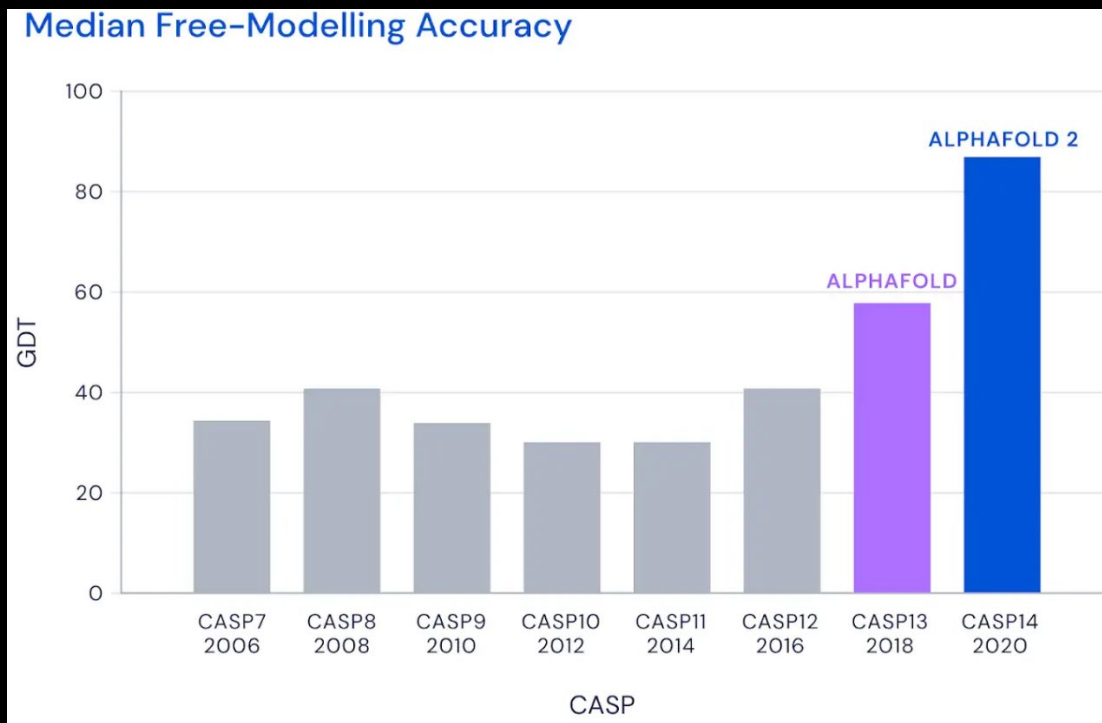
Python program to display Astrological sign or Zodiac sign for a given data of birth.

Zodiac sign

This is getting a little bit contrived.

Luckily we managed to get to Z.

On this note.... Now do you see how important coding literacy is?



CASP - biennial
Critical Assessment
of protein Structure
Prediction

AlphaFold 2

A notable example that may be pertinent to you.

Now sure you asks yourself...
What can I use this for?





WHY?

Try this exercise... ask why...

Train your curiosity

Why Python for me?

I need to do the following repetitively:

- Drag & Drop
- Download
- Digitize

I know there should be a better way.



Python@work

- Digitize graph
- Image cleaning
- Download data from websites
- Data format/structure transformation (ETL)
- Generate maps
- Extract coordinates from geospatial data
- Renaming files
- Generate folders from list
- Generate plots and graphs
- Zipping files
- Digitize seismic image into seismic data
- Digitize “log” image into log data
- Finding matching information from multiple files
- Summarizing data

My offer to you today



"Not so much about filling you up with knowledge as it is about *'teaching you how to think.'*"

— Author David F. Wallace

"Everything should be made as simple as possible, but no simpler."

— Albert Einstein

"Be less curious about people and more curious about ideas"

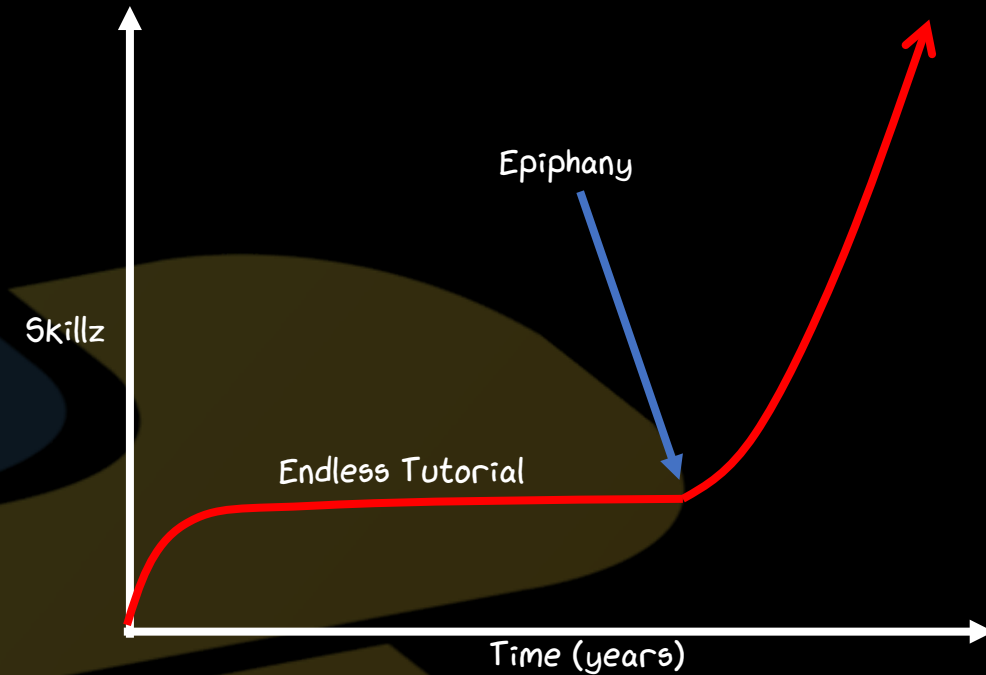
— Marie Curie

What to expect?


It is easy...but not going to be too easy

Expect doing tutorial for years but unable to go beyond it

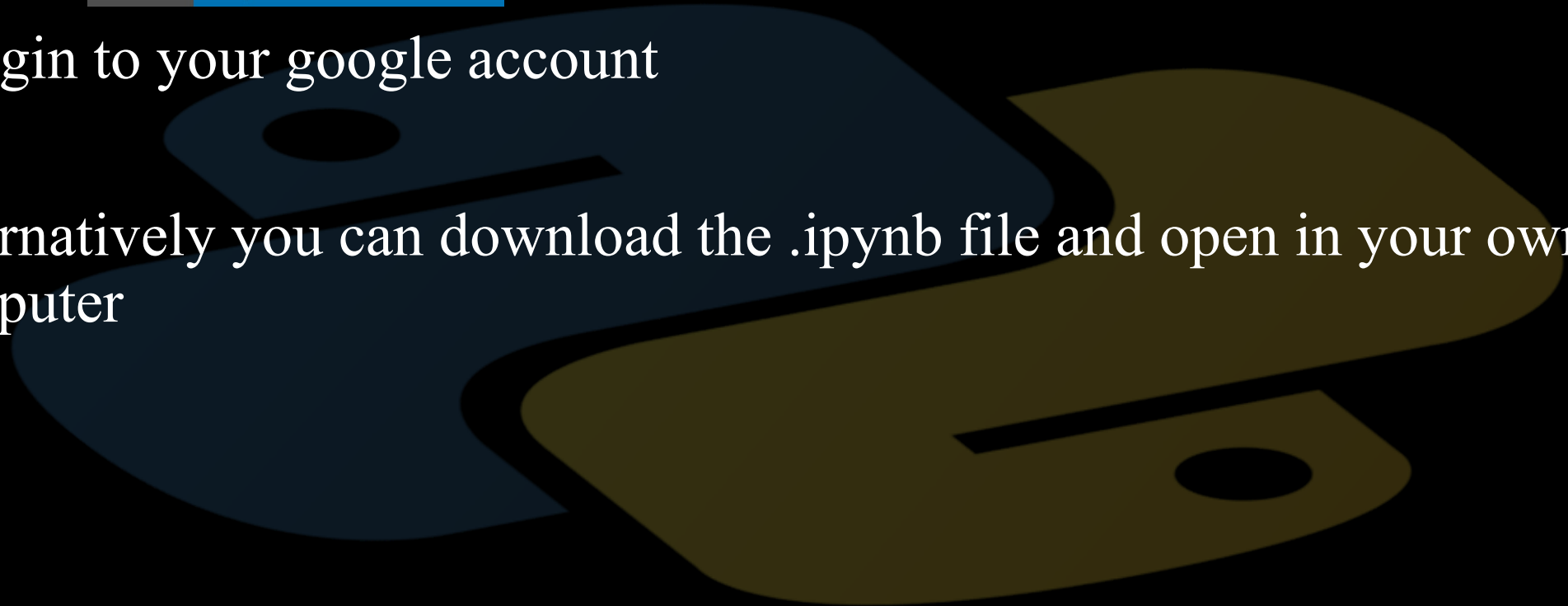
Then, one day... you magically knows how to do everything



Logistics before we start

- Go to <http://bit.ly/3qq7iNU> (.ipynb file)
- Click  Open in Colab
- Login to your google account

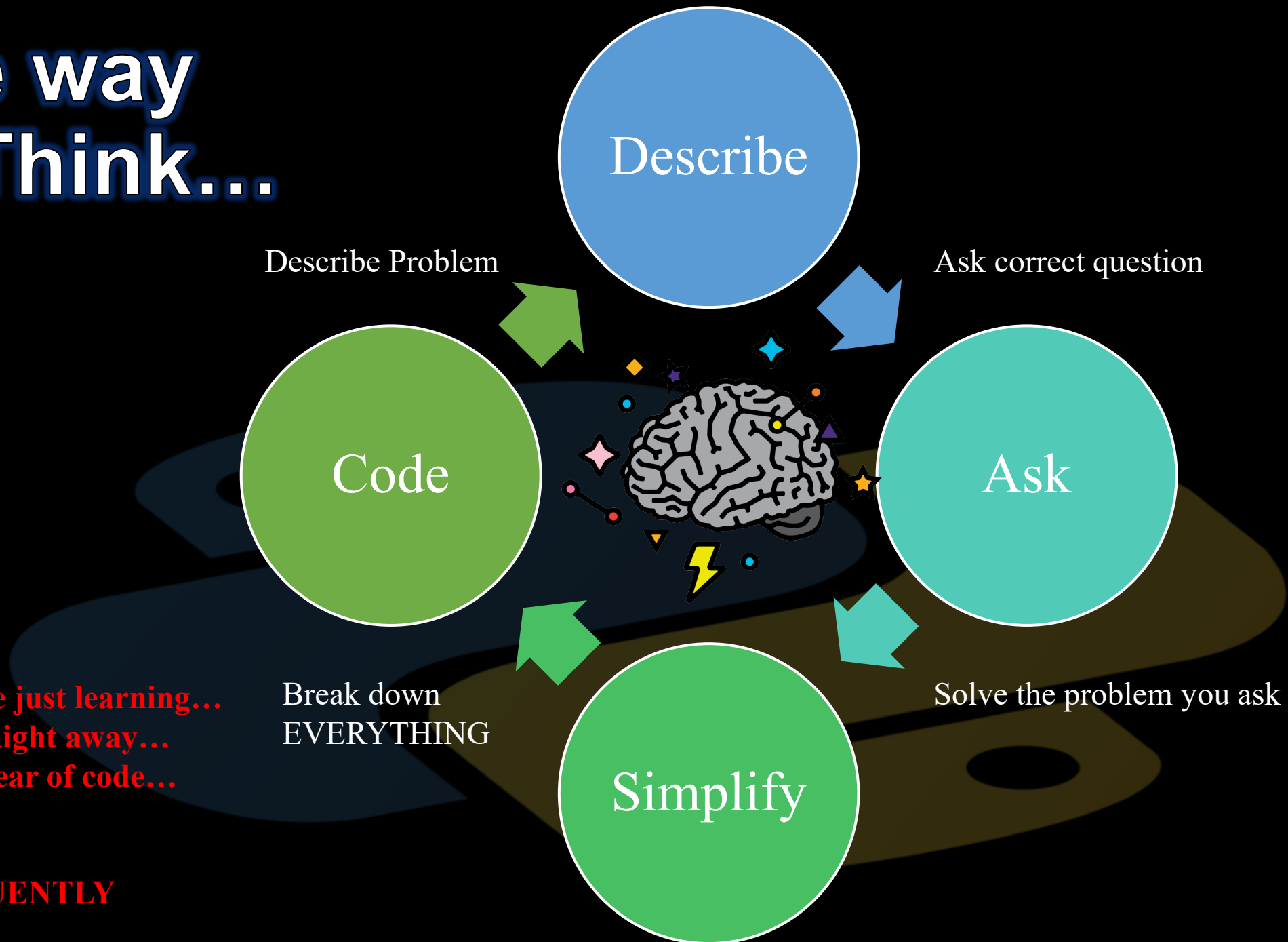
Alternatively you can download the .ipynb file and open in your own computer





import this

One way To Think...



When you are just learning...
Just code straight away...
Remove the fear of code...

FAIL FAST
FAIL FREQUENTLY



Are we ready to leave the
fuzzy philosophical platitudes?



CODING SESSION



WHY?

I can promise you when you solve problem...

You will get your dopamine dose...

- Stringing things together is the trick.





Thank you very much

<https://github.com/elvinado/Problem-Solving-Kit>