

# (Intro to ) Data Analysis in Python

John Serences, [jserences@ucsd.edu](mailto:jserences@ucsd.edu)

January 9<sup>th</sup>, Winter 2019

Class 00

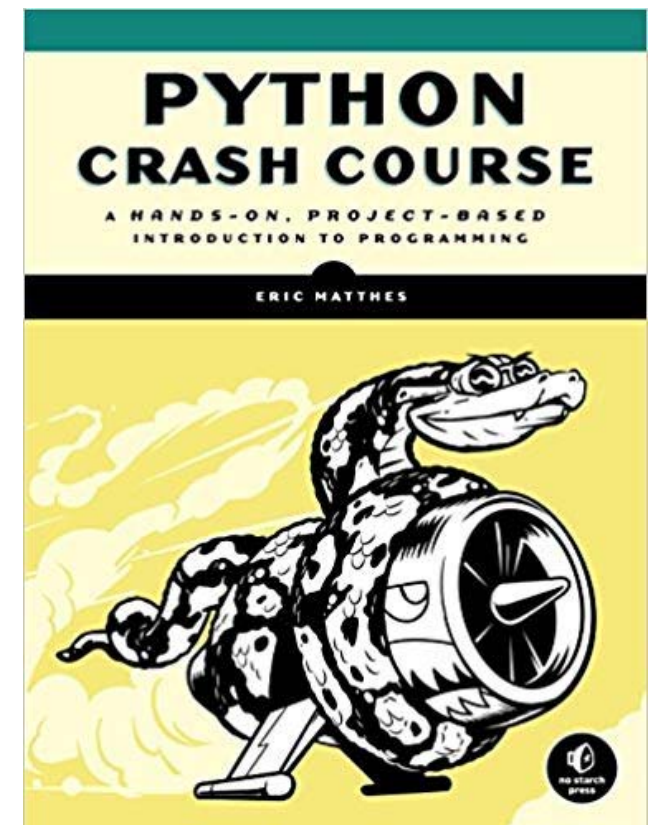
# Goal of the course

[https://github.com/JohnSerences/PSYC193\\_IntroPython\\_W2019](https://github.com/JohnSerences/PSYC193_IntroPython_W2019)

- Develop solid understanding of the Python language and the Jupyter environment.
  - Open science, data and code sharing
- Introductory course for people new to Python and new to coding
  - Experience in another language may help, but no programming experience is necessary!
- Why Python?
  - Incredibly flexible for data analysis (modules/libraries)
  - Quick development for prototyping/production, excellent GUI support
  - Support for generating and compiling C code (faster execution)
  - Good balance of flexibility and power against complexity of language/constructs (e.g. Visual Basic/Matlab vs C/C++)

# Textbook

- Python crash course : a hands-on, project-based introduction to programming by Eric Matthes, **Nov 1, 2015**
- First 6-7 weeks of the course will generally follow chapters 1-11 of this book (Part I)
- Not essential, but will likely help, especially if you are brand new to programming
- New on Amazon for \$27.00
- Used on eBay for \$2.95



# Grading: Exams

- Short quizzes – will have a very short quiz at the start of most classes (except today and for class with midterm)
  - Can drop lowest two grades
  - 10 points for a total of 60 points (best 6/8 quiz scores)
- Midterm – Feb 13<sup>th</sup>, will cover material from the first half of the quarter
  - 50 points
- Final – cumulative exam during finals week (time/date TBA)
  - 100 points

# Grading: In-class exercises

- Each week we will have an in-class problem set.
- These assignments are meant to give you the hands-on practice that you need to develop fluency in the language.
- You will usually work on this collaboratively in groups, although some assignments may be individual.
- At the end of the quarter, you must have all assignments done and each of your notebooks saved in your course folder.
- I will check all problem sets periodically throughout the quarter.
  - Completion of each assignment will be worth 5 points, for a total of 35-40 points (for 7-8 in-class problem sets)

# Grading: scale

- 60 (best 6/8 quizzes) + 50 (midterm) + 100 (final) + 40 (in-class problem sets) = 250\* points available to earn for the course
- \*Might only be 245 if we have 7 instead of 8 problem sets
- To compute:
  - $(\text{your points}) / (\text{total points})$
- Grades will not be rounded!

A+	97-100
A	93-96.99999
A-	90-92.99999
B+	87-89.99999
B	83-86.99999
B-	80-82.99999
C+	77-79.99999
C	73-76.99999
C-	70-72.99999
D+	67-69.99999
D	63-66.99999
D-	60-62.99999
F	0-59.99999

# Academic Integrity

- "Integrity of scholarship is essential for an academic community. The University expects that both faculty and students will honor this principle and in so doing protect the validity of University intellectual work. For students, this means that all academic work will be done by the individual to whom it is assigned, without unauthorized aid of any kind."
- This course will also make use of online quizzes and exams via Google Forms.
- Taking a quiz or exam logged in as another student will be treated as a violation and you will be referred for disciplinary action.
- Similarly, emailing with or otherwise communicating with other students or anyone else during a quiz or exam will be treated as a violation and also referred for disciplinary action.

# Course Schedule (approximate)























- Week00, January 9: What is Python?, Jupyter Environment (Google Colab), First Program, Intro to object types and methods
- Week01, January 16: More on object types, lists, for loops, list comprehensions, slicing lists
- Week02, January 23: If...elif...else statements, dictionaries
- Week03, January 30: User input, while statements
- Week04, February 6: NO CLASS
- Week05, February 13: Midterm, writing functions
- Week06, February 20: Classes, object-oriented programming
- Week07, February 27: File Input/Output, data formats for files (e.g. JSON, HDF5)
- -----end of following book-----
- Week08, March 6: NumPy (numerical computing), Plotting (Matplotlib/Seaborn)
- Week09, March 13: Pandas (data frames)
- Final: Room/Time TBD



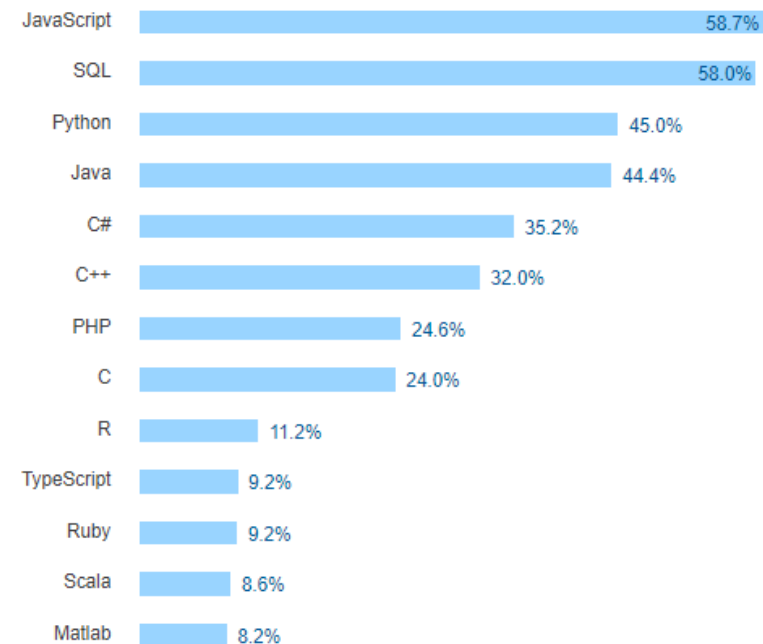
# Python vs Matlab

- Programming style
  - 0's and 1's
  - block indent
  - () vs [] for function calls, array indexing
- Use in industry/academia
  - Python is more common in industry – prototyping to full development
  - In academia, Python use rapidly growing
  - Many new branches of analysis/computing are led by the Python community with Matlab playing catch-up
- Bleeding edge – good and bad
- Open development community – good and bad.
- Some references (note who wrote each page...)
  - <https://www.mathworks.com/products/matlab/matlab-vs-python.html>
  - [https://pyzo.org/python\\_vs\\_matlab.html](https://pyzo.org/python_vs_matlab.html)
  - [http://phillipmfeldman.org/Python/Advantages\\_of\\_Python\\_Over\\_Matlab.html](http://phillipmfeldman.org/Python/Advantages_of_Python_Over_Matlab.html)
  - Perhaps the most balanced (and relevant): <https://blog.thedataincubator.com/2017/10/matlab-vs-python-numpy-for-academics-transitioning-into-data-science/>

# Bottom line on Python vs Matlab (and other languages)

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

IEEE Spectrum 2017



<https://insights.stackoverflow.com/survey/2017>

Most popular language for data scientist/engineer  
[check out this page for other interesting stats]

# File structure for in-class work

- Shared folder on google drive
  - Use this to store your in-class work from each week
  - Most assignments are group work with goal of completing most of it in class
- These folders are shared only with me
  - Do not share code notebooks with classmates

# Jupyter Notebooks

- All coding will be done in the Jupyter Notebook environment as implemented by Google Colab
- [https://github.com/JohnSerences/PSYC-NEU-231/blob/master/Tutorial\\_080\\_SensoryInferenceModel.ipynb](https://github.com/JohnSerences/PSYC-NEU-231/blob/master/Tutorial_080_SensoryInferenceModel.ipynb)

# Google Colaboratory (with 1 L)

- Login with @ucsd.edu username and active directory credentials
- Jupyter Notebook environment (<https://jupyter.org/>)
  - Contains live code, equations, visualizations and narrative text all in one place
  - Easy to share – cross platform and (should) run on any computer and any OS and will produce the same output
  - Google colab is a Jupyter notebook environment that requires no additional setup
    - Runs on virtual machine that is set up when your session starts (and is recycled after session idle)
    - Supports Python 2.7 (deprecated) and Python 3.6 (current active version)
    - All major extensions (modules/libraries)
    - Easy to share directly on drive or after downloading in open source .ipynb format

# In-class work

- To launch a new notebook and do basic setup
  - Log in to: <https://colab.research.google.com/notebooks/welcome.ipynb>
    - Can just google: “google colab”
    - Use your AD credentials to login
  - File menu -> “New Python 3 Notebook”
  - File menu -> “Rename” (or just type in the name filed in upper left corner)
    - File name should be: Lastname\_PSYC193\_Class00.ipynb
    - Use exactly this convention EVERY time, only updating the 00 counter (so next class is 01, etc).
  - File menu -> “Locate in drive”
    - This will launch a new window with a file list
    - Right click, Move, navigate to our shared folder and move file
    - Now all your work will be saved in our shared folder...

# Key concepts for today

- Variable: symbolic name that refers to an **object** (or to a chunk of data)
  - Objects can be a letter string, number, list of letter strings or numbers, etc
  - Many specialized types of object: str, int, float, list, dictionary, etc.
  - The data is contained within the object
  - A **variable** is a useful (i.e. readable/memorable) label for an object
  - Different objects take up different amounts of memory
    - Example – it takes less memory to represent a whole number (e.g. 3) than it does to represent a long “floating point” decimal number (e.g. 3.141592653589793)

# Key concepts for today

- Method: a function that is available for a given type of object (or to the variable that refers to the object)
  - You can use methods to manipulate the data that are assigned to a variable
  - Example: if you have a list of words, the `sort()` method will re-arrange the list in alphabetical order
  - Object oriented programming!
  - More formally: a method is a function that is the member of a class (this won't make sense now, but it will by week 7 or so)



# Pilot course

- Going to be trying out various new ways of disseminating course materials and giving exams
- Please be patient – some things will surely go wrong the first time, but I'll get them straightened out
- **ASK QUESTIONS!!!!**
- Thank you!



## Some shortcut keys (to start with)

- On a PC cntrl = control key, on Mac cntrl = “apple” key
  - New cell above: cntrl+M A
  - New cell below: : cntrl+M B
  - Convert to code cell: cntrl+M Y
  - Convert to text cell: cntrl+M M
- Run a cell (execute code or display markdown): cntrl+ENTER