
Welcome to Astro 9
Introduction to Python


Who Are We?

Daniel Klyde

- He/him/his pronouns
- Graduated from Berkeley in 2021
  - Physics/Astrophysics Major
- Worked for the last six months as a research assistant for the Ion Beam Technology group at LBL
- Research Interests
  - Exoplanets
  - Supernovae

BERKELEY LAB

Raphael Baer-Way

-He/him/his pronouns

-4th year Astrophysics+Physics

-Co-president of Club Soccer

-Research Interests(Filippenko group):

i. Supernovae
ii. Variable Stars

Course Goals

- Introduce you to the basics of the python coding language
- This includes learning to:
  - Write Functions
  - Create Plots
  - Work with large datasets
  - Understand where/how to use loops
- Conduct your own research project by the end of the course

What Are We Going to Learn?

- Week 1
  - Python basics: variables, arithmetic functions, etc.
  - Introduction to Numpy
  - Introduction to plotting and Matplotlib
- Week 2
  - Intro to Functions
  - Global vs Local variables
  - The “While” Loop
  - The “For” loop
  - “If” statements

What Are We Going to Learn?

- Week 3
  - Data Processing and Analysis
  - Introduction to Pandas
  - Analyzing large data-sets
- Week 4
  - Introduction to statistics in python
  - Mean and standard deviation
  - How to incorporate error estimates
  - Data Fitting

What Are We Going to Learn?

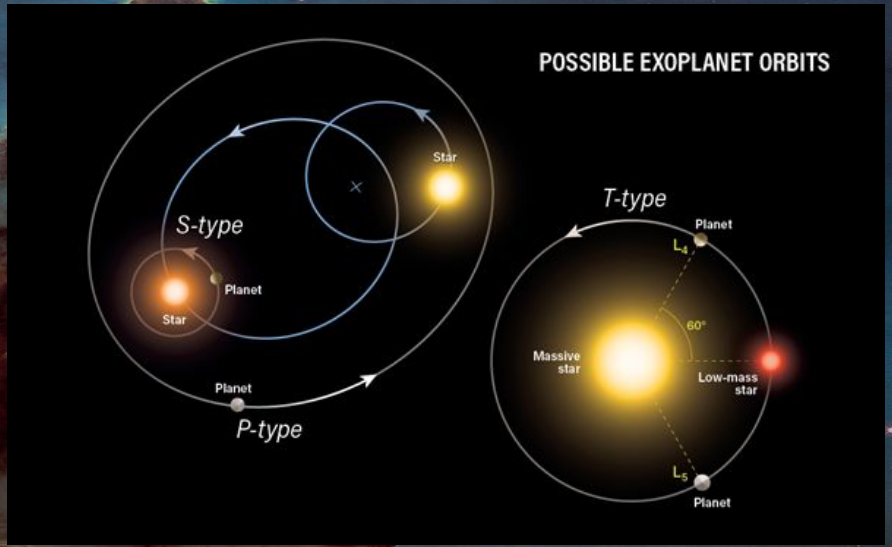
- Week 5
  - Calculus in Python
  - Numerical Differentiation/Integration
  - Linear Algebra in Python
  - Matrix Manipulation
- Week 6
  - Differential Equations
  - Using Python in astronomy research
  - Final Project Presentations!

The Final Project

- Your chance to demonstrate everything you learned over the next 6 weeks
- In small groups, you will choose a topic for your computational projects
  - Each topic needs to be brought to me for approval
  - Your topic proposals will be due next week as a part of your first homework assignment
- On our last day, each group will present their project to the class

The Final Project: Example Topics

- Data Analysis Projects
  - Create your own HR diagram
  - Create visualizations of exoplanet datasets
- Simulation Projects
  - Simulate the orbit of an exoplanet around its star
  - Model Shallow Water Equations
- More Final Project Examples:
  - https://pythondecal.github.io/



Homework

- Each week a new homework will be assigned to help you practice the skills learned during that week
- Every completed homework turned in will automatically given 50% credit, the rest will be based on correctness
- Homeworks will be due at midnight on the next Monday after they were assigned
- Collaboration is encouraged but everyone must turn in their own homework

Workshops

- Each week there will also be an assigned workshop
- These will be graded on completion rather than correctness
- At the end of every lecture some time will be given to work on the workshops to provide the chance to work collaboratively and ask questions
- The workshops will be due at midnight the first Friday after they were assigned

Grading

- 5% Participation
- 50% Homework Sets
- 20% Workshops
- 25% Final Project
- This class will not be curved, but there will also be no exams
- We have no set average grade, if everyone earns an A then everyone will get an A

Some Housekeeping Work…

Supplemental Lectures

- Every week Raphael will be leading a short supplemental lecture on a topic that we will not require you to know but are otherwise good things to learn if you will continue to use python in the future
- https://www.when2meet.com/?15971634-xSvbO

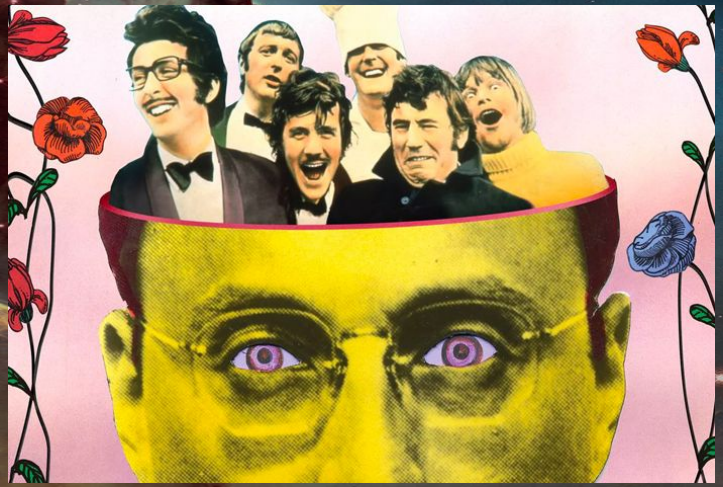
Office Hours

- Raphael and I will both hold weekly (possibly bi-weekly depending on demand) office hours
- What times would work best for everybody?
- https://www.when2meet.com/?15971670-twxnx


Ready to Start Learning Python?

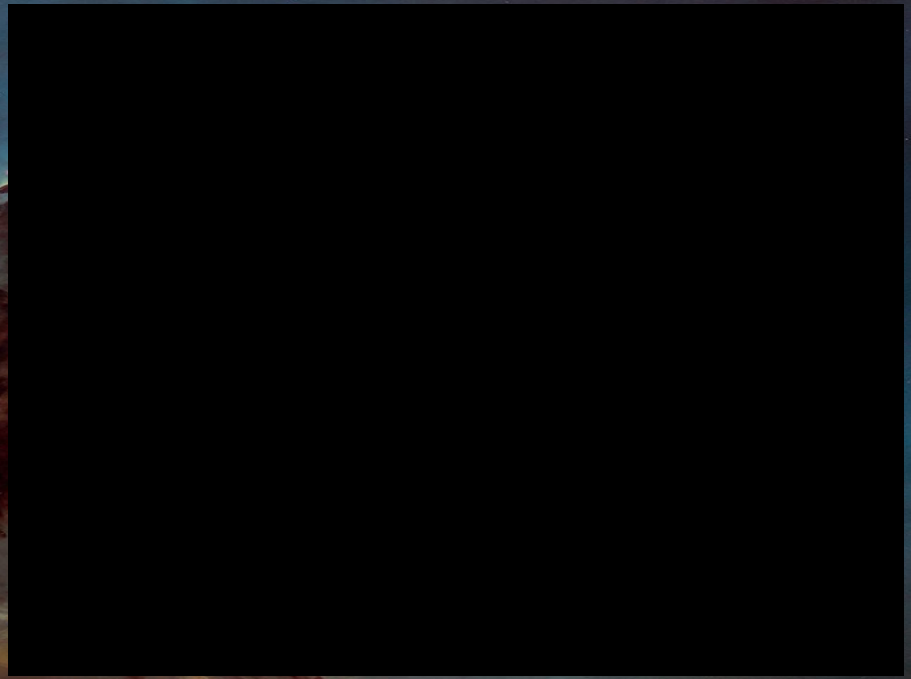
Why Python?

- Simple Syntax, relatively easy to learn for beginners
- Versatile
- Improved further by outside development
- For Astronomers:
  - Works well with large data sets
  - Can be used to interact with instruments
  - Can be used to make plots and simulations



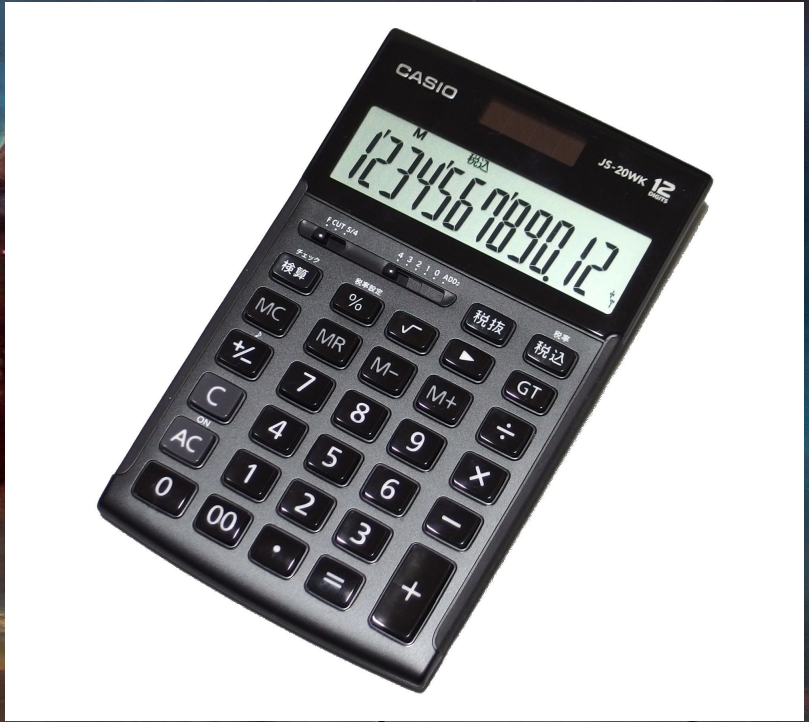
Anything you can do it can do better

- What can your best calculator do?
- You might have even had some experience with excel, which can plot and work with data fairly well
- Neither of these tools can hold up to what you will be able to accomplish once you become a proficient coder

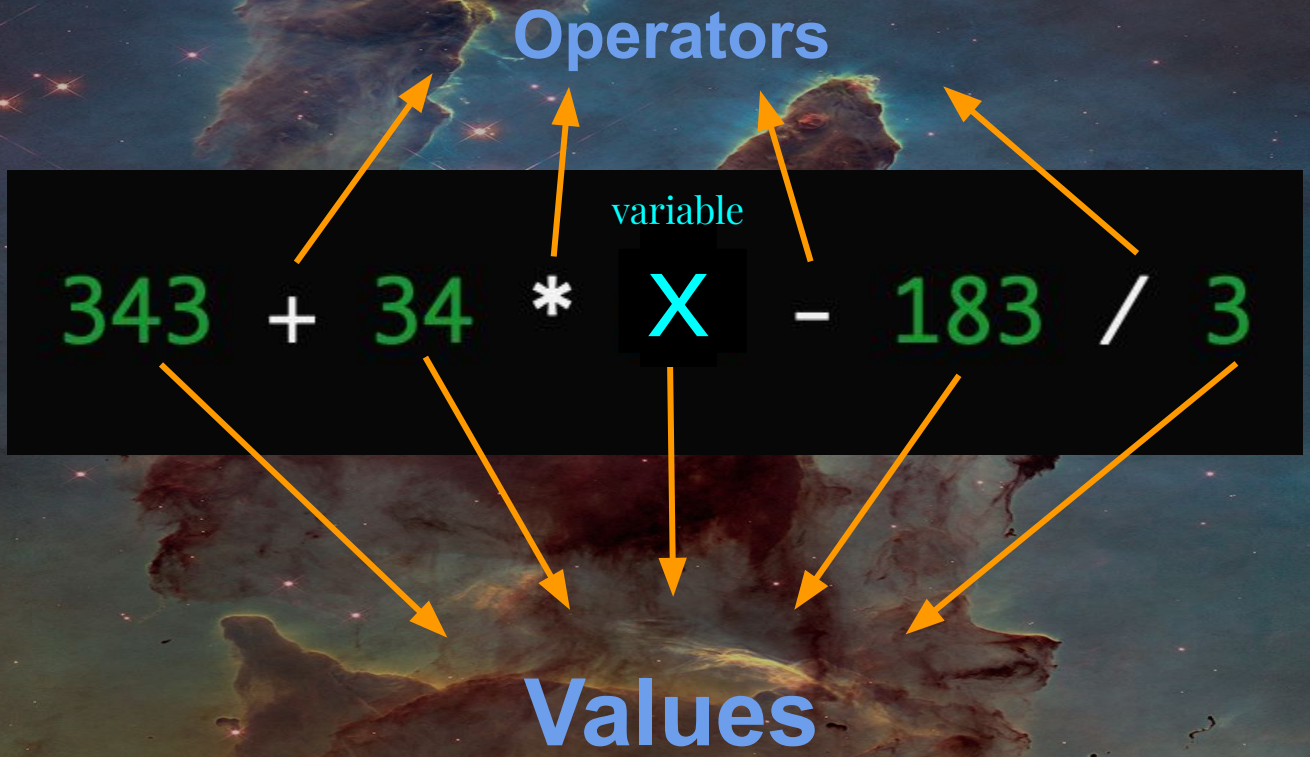


Where to start?

- Coding is meant to make your life easier
- In order to get there you need to learn its grammar
  - There's a reason they call it a language
- Start off with something easy for STEM students to understand: use it as a calculator



Expressions



What is a variable?

- A way of storing information in the computer
- We call this information DATA
- This data can have different DATA TYPES

```
x = 7

y = 7.0

Z = ‘words, words they’re all we have to go on’
```

What are the different Data Types?

- Integers
- Floats
- Strings
- Lists
- Tuples
- Dictionaries

Data Types: Numbers

- Integers (**`int`**) and Floats (**`float`**) - “NUMBERS”
  - Integers: **`-3  -2  -1  0  1  2  3`**
  - Floats: **`3.3  13451.133434  98.7`**
- How to check the type of an object?

```
type(your_object)
```

- ?
  - What is the output of **`type(183884913830484010372448 2.)`**?

+	Addition	4 + 7 ⟶ 11
-	Subtraction	12 - 5 ⟶ 7
*	Multiplication	6 * 6 ⟶ 36
/	Division	30 / 5 ⟶ 6
%	Modulus	10 % 4 ⟶ 2
//	Quotient	18 // 5 ⟶ 3
**	Exponent	3 ** 5 ⟶ 243

Data Types : Strings

- Strings (**`str`**) - “WORDS”
  - **`‘Hello World’`**
  - **`“Go Bears!”`**
  - Either type of quotation marks is fine so long as they are consistent
- You can add two strings
  - `>>> “134” + “34”`
  - `“13434”`

What is the output of **`7 * “1”`**?

Indexing (Python index starts from 0, not 1!)

```
>>> “Hello World”[0]
“H”
```

```
>>> “Hello World”[5]
“ ”
```

Data Types: Lists/Tuples/Dictionaries

- List (**`list`**) **`[4, 9, 7.5, ‘astronomy’, [“Berkeley”, True]]`**
  - Calling certain elements (INDEXING)
    - `[1,2,3,4,5,6][0]` → `1` `[1,2,3,4,5,6][-1]` → `6`
    - `[1,2,3,4,5,6][2:5]` → `[3,4,5]`
  - You can modify the elements in a list
    - `my_list = [1,2,3]` → `my_list[0] = 37`
      `my_list` → `[37,2,3]`
- Operations
  - `3 * [1,2]` → `[1,2,1,2,1,2]`
  - `[1,2] + [1,2,3]` → `[1,2,1,2,3]`

Data Types: Lists/Tuples/Dictionaries

- Tuple (**`tuple`**) **`(1,2,3,4)`**
  - Indexing and operations are the same as a list
  - You CANNOT change the elements
- Dictionary (**`dict`**)

```
d = {Stars: [‘Sirius’, ‘Sun’], Planets: [‘Venus’, ‘Saturn’]}
```

  - Accessing items:

```
>>> d[‘Stars’]
    [‘Sirius’, ‘Sun’]
```

Data Types: Booleans

- Booleans (**`bool`**) - “True or False”
  - **`True`**
    - A value is true unless it is not
  - **`False`**
    - **`0, [], (), {}, “”, None`**

- Operations:
  - **`3 > 2`** = ? **`0 <= -100`** = ?
  - **`23 == 23`** = ? **`73 != 9`** = ?

Where do you write your code?

- Where ever you want to…

- Our default answer will be datahub
  - Everyone at berkeley gets a free datahub account so just log in with your cal id
  - https://datahub.berkeley.edu

- All of your homeworks and workshops will be in the format of Jupyter Notebooks


First Coding Demo