Python Crash Course checklist

1. Use Python and Jupyter Notebooks to do basic arithmetic operations
2. Assign variables and do arithmetic operations using variables
3. Create lists and arrays
4. Write a function, use function to operate on lists/arrays
5. Plot x,y data using Matplotlib
6. Load external data for analysis and manipulation
   1. Plot, run data through function, etc.

**Pre-basics**

1. Performing operations in the terminal/command prompt
   1. Open Anaconda Prompt by searching in the Windows search bar (or opening a Terminal for Unix-based systems)
   2. Perform simple operations
      1. Type python to start a python session
      2. print(“Hello, World!”); terminal should respond by outputting the text Hello, World!
      3. Basic arithmetic – try +,-,\*,/,\*\* operations
      4. Assign local variables – x = 5, y = 7; perform operations
      5. Type exit to leave (or close the Command prompt)
   3. Write a script in a text editor
      1. Open your favorite text editor
      2. Write a few lines of script!
         1. E.g. x = 10; y = 5; x+y\*\*2
      3. Save your script as filename.py – save it to your Desktop.
      4. Open a new command prompt
      5. Use the command prompt to navigate to the Desktop. (Type ‘dir directory’ to navigate from one directory into another; directory=folder)
      6. Type python filename.py – it should run your script! :D

We will not often use line-by-line terminal-based coding, but it is good to know how to run scripts as a saved file (script.py). More often than not, we will use Jupyter Notebooks to run scripts.

**Basics I in Jupyter Notebooks**

1. Operations in Jupyter Notebooks
   1. Open Anaconda Navigator
   2. Launch Jupyter Notebooks (if you don’t have it, click the Channels button and search for it)

Jupyter notebook uses your browser as a graphical user interface and uses a “cell” structure to break up scripts into discrete chunks of code. Super useful for debugging. Also, the outputs are “in line”.

* 1. File > Open: Crash\_course\_UNPOPULATED.ipynb
  2. Perform simple operations by making a cell active and then clicking “Play” or Shift-Enter to run the lines of code in that cell
     1. Print(“Hello, World!”)
     2. Basic arithmetic – try +,-,\*,/,\*\* operations
     3. Define local variables
     4. Types of items
     5. Other functionalities: syntax, comments, headers, markdown, etc.

**Basics II**

1. Packages and extras
   1. Common packages: numpy, scipy, matplotlib, pandas: these give you access to other Python functionalities beyond the base package
      1. Numpy gives you access to things like exp, pi, log, log10, etc. – try it!
   2. Install via conda or pip
      1. Open Anaconda Prompt
         1. Conda install \_\_\_\_\_\_
         2. Or, pip install \_\_\_\_\_\_
   3. In the first cell, include:
      1. Import numpy as np
      2. Import matplotlib.pyplot as plt
      3. Include: %matplotlib notebook
2. Try some examples with exp, pi, log, etc.
3. Lists
   1. Defining lists
   2. Indices
   3. Operating on lists
   4. Making data

**More Advanced Topics I**

1. For loops
   1. Looping over lists
   2. Looping operations in a list
2. Conditional Statements
   1. If
   2. If-else
   3. If-or
3. Functions
   1. Basic print function
   2. Basic mathematical functions
4. Visualizing Data
   1. Matplotlib
5. Exporting data and saving images
   1. np.column\_stack
   2. np.savetxt

**More Advanced Topics II**

1. Loading data from an external source
   1. Numpy
   2. Others (pandas, read\_csv)
2. Fit data to a function using least squares minimization
   1. Linear regression
   2. Fit a written function, extract fitted parameters