

Introduction to Python:

Background:

Python is a programming language lauded for its readability. Even if you do not know the syntax, it is often not too difficult to understand what python code does. This makes it an excellent place to start when learning how to write programs.

In this class, we'll be learning python in an environment called a notebook. A notebook is a series of cells, each one containing some python code. If we run one cell, the notebook will remember any information we've stored and it will be available in another cell. Altogether, the notebook contains the python program, which typically performs some useful function or analysis. For example, one might analyze the signal from an Electrocardiogram (EKG) in order to identify heart conditions, visualize the positions of stars in the Milky Way, or compute the likelihood that a soccer player scores a goal based on their location on the soccer pitch.

There are many different types of python notebooks, but we will be using Kaggle notebooks. Kaggle is a free, online, data science platform that anyone can use. This assignment assumes you have already created a Kaggle account for the course. Inside the Kaggle notebook environment you will find the main notebook window where the cells of code are located, a data and options panel on the right, and a toolbar at the top.

Skills:

First, we'll need to learn a few new things about python syntax and Kaggle:

- We'll be writing python programs in Kaggle notebooks.
 - Each little block of code is called a cell.
 - Run each cell sequentially – running cells out of order can cause trouble

```
[ ]: # Python 3 environment

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt # for plots

[ ]: #This will read in the csv file and create an object called a Data Frame
data = pd.read_csv("/kaggle/input/galacticcoordswithgaia/gaiaDataNearSun.csv")

[ ]: # Inspect data:
data.head(10)
```

- We'll make use of various libraries, which contain useful tools we can use:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt # for plots
```

- You can assign a variable with the following syntax, and the program will know what that variable represents:

```
m = 21

v, w = 0, 1

print(m, v, w)

21 0 1
```

- One can also assign an array of values to a variable. One way to do this is to use a “numpy array.”

```
planets = np.array(["Mercury", "Venus", "Earth", "Mars", "Jupiter", "Saturn", "Uranus", "Neptune"])
a_in_AU = np.array([0.387, 0.723, 1.000, 1.524, 5.20, 9.6, 19.2, 30]) #The semi-major axis length in AU for each planet
P_in_yr = np.array([0.241, 0.615, 1.000, 1.881, 11.9, 29.5, 84.0, 164.80]) #The orbital period in years for each planet
```

- We can select a particular entry from an array of numbers by specifying its index by adding a [n] after the array, where n is the index number. (The index is essentially its address within the array). Careful, the index starts at zero!

```
[5]: print(planets[0])

Mercury
```

- An if statement only runs the code inside it if certain conditions are met:

```
[18]: x = 31

if x > 10:
    #If x is bigger than 10 this indented code will run, but the others will not.
    print("Bigger than 10!")
elif x == 10:
    #If x is 10 this indented code will run, but the others will not.
    print("Equals 10.")
else:
    #If the other cases do not run, this case will run (i.e., x is less than 10)
    print("Not bigger than 10.")

Bigger than 10!
```

- A for loop “loops” through a set number of iterations
 - Notice the example loop below starts at $i = 0$ and ends at $i = 6$. The loop loops through values up to but not including the second number in the `range()` command.
 - A for loop changes the value of i each time we go through the loop.

```
[2]: #This is a for loop.  
#i is a variable that tells us the number of the loop we are currently on  
#i will go from the first number in the range up to but not including the last number:  
for i in range(0, 7):  
    print("Currently on loop number", i)  
  
Currently on loop number 0  
Currently on loop number 1  
Currently on loop number 2  
Currently on loop number 3  
Currently on loop number 4  
Currently on loop number 5  
Currently on loop number 6
```

Problems:

1. In Kaggle, go to the following notebook template that has been created for you:
<https://www.kaggle.com/code/austinhinkel/intropythontutorial/notebook>. Click the three dots at the top right and select copy and edit notebook.
2. Run the first few cells in your Kaggle notebook template – they have been completed for you. Be sure to read the commentary in between the cells of code.
3. When you get to cell 5, you will have to write your own code. Complete the code as prompted.
4. Do the same for every part of the notebook that asks you to change a value, edit the code, or write your own.

Your instructor will check your Kaggle notebook and will sign off here when it is complete:
