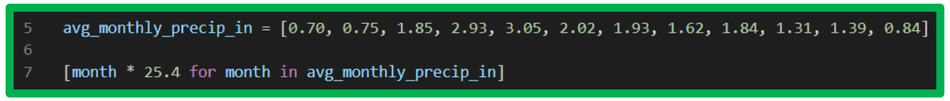
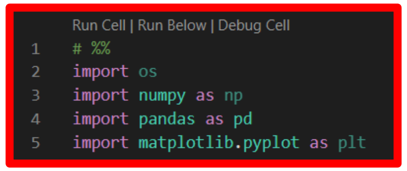
PYTHON CHEAT SHEET

- Lists: collections of values or objects in a specified order
  - How to make them:
    - Making a list depends only on the syntax used:
      - Have to use square brackets in order for python to recognize a list is being constructed, and commas must separate each item inside the brackets
      - Example: my_list = [1, 2, 3, 3, 4]
  - Lists can contain any data type:
    - Float (i.e. decimals: 1.0, 5.8, 346.890, etc.)
    - Int (i.e. integers: 5, 3, 49, 286489, etc.)
    - String (i.e. text strings: “hello”, “what’s new”, etc.)
  - Indexing:
    - When a list is created, it is automatically assigned an index number by Python starting from the left at index number 0 and counting upward by 1’s
    - Any value can be pulled from the list using its index number and the following syntax:
      - list_name[start : stop : step]
      - Ex: my_list[2]
        - If this code were printed, it would yield the value [3] as that is the number in the index space 2 in the example above
    - Indexing can also be much more complicated:
      - my_list[:3] when printed would give the result [1, 2, 3] as the start is assumed to be the beginning of the list since it is not specified, the stop is index space 3 (non-inclusive) and the step is automatically set to 1 by python unless specified
  - Method List Examples:
    - .append
    - .insert
  - Function List Examples:
    - type()
    - mean()



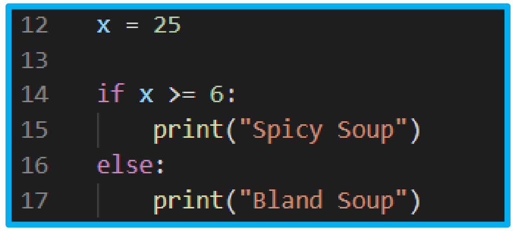
```
avg_monthly_precip_in = [0.70, 0.75, 1.85, 2.93, 3.05, 2.02, 1.93, 1.62, 1.84, 1.31, 1.39, 0.84]

[month * 25.4 for month in avg_monthly_precip_in]
```



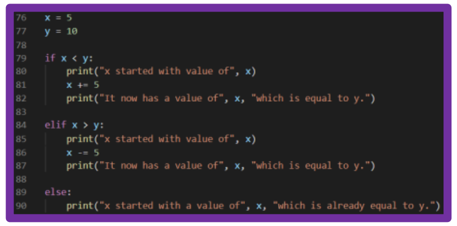
```
Run Cell | Run Below | Debug Cell
# %%
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

- Packages: a package is a collection of tools that must be imported to python before each use.
- Objects: the way we put code together in python to execute certain functionalities. These collections of code have functions connected to them.
- Functions: functions are routines in python that do specific things. They are placed before the object in python code.
- Methods: methods are functions attached to objects. They will only run properly for their appropriate object types. They are placed after the object in python code.
- Attributes: these are properties of objects. Examples include length (len()), and dimensions (.dim).

- Conditional Statements: conditional statements depend on syntax to work, and run code according to certain conditions
  - if else conditionals the format seen in the picture below, if “some code” is true then the indented code will be executed, otherwise (or “else”) the final code will be executed



```
x = 25

if x >= 6:
    print("Spicy Soup")
else:
    print("Bland Soup")
```



```
x = 5
y = 10

if x < y:
    print("x started with value of", x)
    x += 5
    print("It now has a value of", x, "which is equal to y.")

elif x > y:
    print("x started with value of", x)
    x -= 5
    print("It now has a value of", x, "which is equal to y.")

else:
    print("x started with a value of", x, "which is already equal to y.")
```

  - elif conditionals allow for multiple lines of code to be run between the if and the else. The same rules around indentation still apply and the structure of falling into bins if the conditions are not met still apply as well.
- For Loops: for loops iteratively execute code for each item in a predefined list.
  - They follow the syntax seen in the picture below, and similar to conditionals, indentation is very important to python’s recognition of this as a for loop. See picture for details.
- List Comprehensions: list comprehensions simplify for loops into one line of code. See syntax below in picture.
  - i can be placeholder in for loops and represents any item in a list. It can be anything except numbers (banana, month, j, etc.)