Packages:

- Packages are a directory of modules for a specific “program”.
    - Examples: Panda or Numpy
        - Import numpy as np
        - Import panda as pd

Objects:

- An object is a collection of data/variables. This includes integers, strings, and floats.
    - Example: filename = 'streamflow_week3.txt'

Functions:

- Functions are operations that we want to do. They are normally something to create an action. Think multiplication, division, subtraction, and addition.
    - Examples: package.functionname (arg. 1, arg. 2,... arg. n)

Methods:

- Methods are actions associated with an object
    - Example: object.method (arg. 1, arg. 2,... arg. n)

Attributes:

- Attributes return properties of an object
    - Examples:
        myarray.shape = size in every dimension
        myarray.size = total # of elements
        myarray.ndim = # of dimensions
        myarray.dtype = data type

Lists:

- A list is used to store different comma-separated values/data in square [ ] brackets.
    - Example: mylist = [1, 2, 3, 4]
    - Example: mylist.append
    - Example: mylist.index
    - Example: mylist.sort
    - Example: mylist.insert

Indexing:

- Indexing is a way to give individual values/words/data a position within a list and be able to refer to various parts of the list without having to look through every single value within the list (slicing).

Index Position	0	1	2	3	4	5	6	7	8	9
List	1	2	3	4	5	6	7	8	9	10
Index Position	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

- What is slicing and how does it work Slicing:
    - Slicing is a way to look at a range of items in a list by “slicing” the list into different parts.
        - Example:

            mylist = [start:stop: step]
            - (Start: Stop: Step) and has to be repeated for every dimension [rows, column]
            - If nothing is listed at step, it is assumed to be in intervals of one.
            - If nothing is listed for stop, it just goes to the end.

            mylist = [1, 2, 3, 4, 5, 6, 7, 8, 9 , 10]

            print(mylist[2:6]) gets the output: [3, 4, 5, 6]

- Array
    - Array is different from lists as they must all be the same data type (so we can do math)
        - Example:

            Numpy array = object stores grid #’s
        - Example:

            Array1 = np.array([1, 2, 3])
            - Initial array definition using 3 floats

            Array1[0]
            - Gives us the value “1”

            Array[1:3]
            - Gives us: [2, 3]

Conditional Statements:

- Conditional Statements are statements that are only true when the given parameters are met. Nested statements are conditional statements within a conditional statement.
    - Example: If Statement or If Else Statement
    - Example: If a > 6 AND b == 8:

        print(“howdy”)

- Example:

```
If a == 7:
    print("a equals 7"
else:
    print("a does not equal 7")
```

- Example:

```
if a > 7
    print("howdy")
        if a > 12:
            print("yall")
```

For Loops:

- For Loops are a command that looks for certain criteria in a code until that criteria is met.
- Example:

```
for range(3):
    print("Howdy")
```

    - (this should print howdy 3 times)
- Example:

```
List = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
for x in List:
    print(x)
```

List Comprehensions:

- List comprehension is a way to condense/simplify for loops, conditional statements, and other sequences that might be in a code into one single line of code.
    - Example:

```
mylist = [ i for i in range(20) if i == 10)
print(mylist)
```

    - Example:

```
Mylist = ["Even" if i = 3 else "Odd" for i in range(20)]
print(mylist)
```