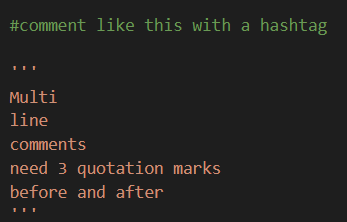
**Initial Learning Topics**

General things to know:

* All python files are .py, for example: HelloWorld.py
* The main function of a Python file looks like:



* Comments are made using #, and multi-line comments with ‘’’



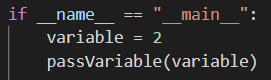
* Output to the console using print()



* You don’t need to specify the end of a line of code (some languages use a semi-colon)
* Initialize a function using def – note that you need a colon after the function name

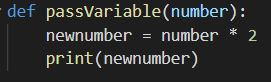


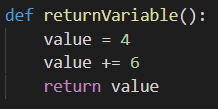
* Variables can be passed from function to function as long as the function call and definition reflect this, as shown below in the passVariable function





* Note that the variable names do not necessarily have to match up in the function call and the function itself





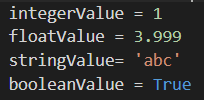
* Functions can also return variables to where the function was called, by setting a variable equal to the function call, that variable will then become whatever is returned by the function



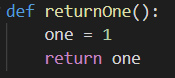
* Import statements go at the beginning of the file to allow use of Python standard library features and any 3rd party package software (more info on this later)



* Initialize variables with 1 equals sign by setting the variable name (left of =) to a value (right of =)

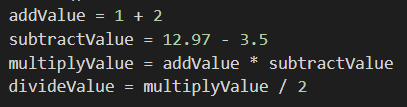


* Return a variable from a function using the return statement
* Here is an example of a full function:

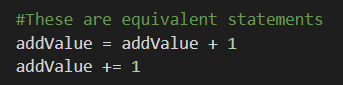


Mathematical Operations:

* Adding, multiplying, subtracting, and dividing are all relatively straightforward



* Modifying a variable can be done simply by using the += or -=, etc.



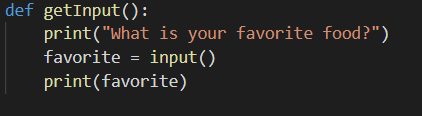
* Exponents are applied using 2 asterisks



* Modulus (calculates the remainder of a division operation) is implemented using the percent sign, for example:
  + 7 mod 2 would be 1 because is you divide 7 by 2 you get 3 remainder 1
  + 15 mod 5 would be 0 because 15 is divisible by 5



* Single vs. Double equal signs
  + Single equals (=) is used to set variables to a value (variable = 10)
  + Double equals (==) is an operation that checks whether the two values are equal, commonly used in if statements (if value == 10 …)
  + They can not be used interchangeably
* Input
  + A python file can also take in user input from the terminal using the input() command:



Lists:

* Python has several built-in data structures, the most basic being lists
* Lists are used to store multiple values in the same variable, and are useful for several reasons
* Lists use square brackets in Python, and can be initialized with or without data



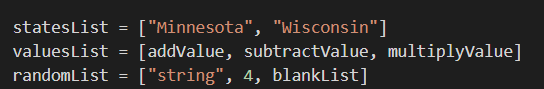
* Python is a zero-based indexing language, meaning the first element in a list is accessible using the 0 index
* Access an element of a list using square brackets and the name of the list



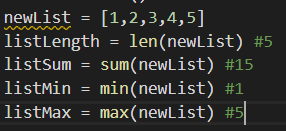
* Add an element to an existing list using the append feature (note that this is done using parentheses)



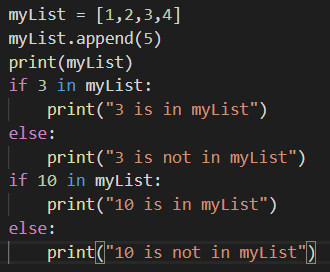
* Lists can contain more than just numbers – they can hold strings, variables, and even different variable types within the same list. Lists can also contain lists!

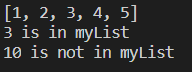


* Lists also have quite a few built-in functions for analysis – including sum, min, max, len, and more!



* Additionally, lists can be modified using the .append() list function, which adds an element to an existing list
* To check if a value is part of an existing list, use “in”



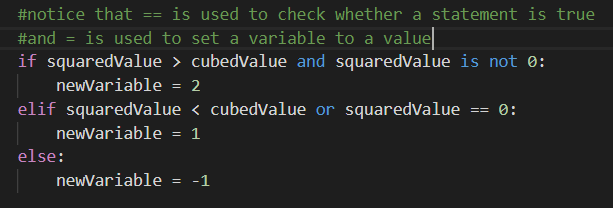


Conditional Statements (if):

* Conditional statements run if certain specified criteria are met, if statements are the most common conditional statement
* Some common Python vocab associated with if statements:
  + and: used to check whether multiple conditions are met (Python equivalent of && if familiar with C#)
  + or: used to check if either of multiple conditions are met (Python equivalent of ||)
  + not: used to interpret the opposite of a statement (Python equivalent of !)
  + <, >, <=, and >= are used in Python as normal
  + Remember that == is used in if statements
* The structure of an if statement is relatively straightforward, no parentheses needed, just a colon at the end and an indentation for any code to execute

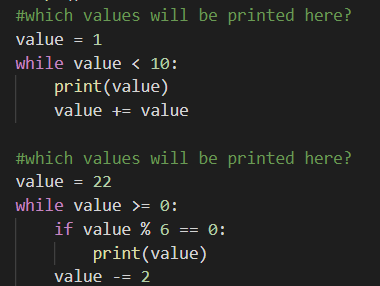


* Python’s version of else if statements is shortened to elif, else is used for else



Loops:

* Loops are used to repeat a process
* Python has for loops and while loops
* While loops repeat an action until a condition is met
* The structure of a while loop is like an if statement – again no parentheses needed, just a colon and indentation for the code that follows



* For loops repeat an action over a specified range of values, using the “in” command, which references either a range of values or a structure
* The structure of a for loop: for [variable being referenced] in [place values are referenced from]:
* To run a traditional loop with a start, end, and increment value (like in C#), the range feature of Python will be necessary
* Note that the range feature is inclusive of the first element, but exclusive of the last element (the following will print 1 but not 11):



* The default starting value of range is 0, and the default increment is 1. The follow loop will print 0,1,2,3,4:



* For loops are also helpful for accessing information in a structure (a list for example)

