Monroe Township Library Coding Bootcamp

Class 3 Notes

* Conditional statements
* Boolean operators
* While loops
* For loops
* Using ranges

**Conditional Statements (if):**

* Conditional statements are used to check if a certain condition is True before running the block of code contained within the statement
* All conditionals start with the if keyword, followed by a statement that can be evaluated to True or False, with a colon placed after the statement
  + The code contained within this block will only run if the preceding statement is True, otherwise the computer will skip it
  + The code inside of the statement must be indented! Python is a whitespace language, meaning that failing to use proper indentation will cause your program to stop and throw an IndentationError
    - This is different than most languages where indentation is encouraged but not required
* You can use the following operators for comparison when constructing a conditional statement
  + Less than (<), greater than (>), \*equal to (==), less than or equal to (<=), greater than or equal to (>=), not equal (!=)
    - The equal to operator uses 2 equals sign, this is to differentiate it from the assignment operator (=)
* You are not limited to using these comparison options though, *anything* that can be evaluated to True or False is acceptable when creating a conditional statement (including functions, methods, truthy and falsey values)
* Conditional statements can also be ‘nested’, meaning you can have an if statement within another if statement

**Conditional statements (elif):**

* You can include additional conditional statements in your code that will run if the previous statement evaluates to False
* These statements come after the initial statement, start with the elif keyword and should be indented on the same line as the if statement
* You can include as many elif statements in a single block of code as you want
  + The program will run from top to bottom until one of the statements evaluates to True, in which case any other elif statements in the same block will be skipped

**Conditional statements (else):**

* You can choose to run a different block of code in the instance where all other statements evaluate to False
* The else keyword must be placed at the bottom of your block, followed by a colon, indented at the same level as any if and elif statements

**Boolean operators:**

* Conditional statements can be combined using the Boolean operators: and, or, not
  + The and operator takes the values of each statement and returns True only if ALL are True
    - True and False 🡪 False
  + The or operator takes the values of each statement and returns True if ANY are True
    - True or False 🡪 True
  + The not operator returns the opposite of a statement’s value
    - not True 🡪 False

**Loops (while):**

* While loops are constructed using a conditional, very similar to the if statement
  + When your program reaches a while loop, it will continue running the code within the loop until the conditional statement evaluates to False
    - Because of this, you should be careful not to create an *infinite loop* where the condition in the while loop never evaluates to False
* You can also exit a while loop by using the break keyword
  + If your code reaches a break keyword, it will immediately jump out of the loop regardless of if the original conditional is True or False
* Similar to the break keyword, the continue keyword will stop executing the code in your loop and start back at the beginning of the loop

**For loops and ranges:**

* For loops are used to loop through a specific set of data that is iterable, meaning that each of its components can be accessed individually
  + Currently, we’ve learned of only one iterable data type: strings
* A for loop can be separated into two sections:
  + The for keyword followed by a new variable - this variable will take the value of one of the elements in the iterable at the corresponding step of the iteration
  + Then the in keyword followed by the data you want to loop through
* We can also create a range of numbers to run a for loop a specific number of times
  + A range must include at least one integer to represent the end position and will start counting at zero by default
    - Optionally, you can give the range a start and stop position, as well as a step amount

**Project: Rock, Paper, Scissors**

**Check class files at** [**github.com/monroecoding**](https://github.com/monroecoding)