**Python Loops**

**for Loops**

In Python **for** loops allow you to iterate over variables that are iterable (can be counted over) or sequences. The following datatypes can be used in a Python **for** loop: **list**, **tuple**, **set**, **dictionary** and a **string**. Essentially any collection variable or string. Unlike JavaScript there is no need for a complicated initiation, conditional and iteration phase in a Python **for** loop. They are more similar to JavaScript’s **for…of** loop.

The loop will run through each element in the collection, or character in the string until it gets to the end. The name you place between for and in (in the below case it is fruit and letter) holds the value for each item or character.

**fruitList = ["apple", "banana"] myString = "James"**

**for  fruit  in  fruitList: for letter in myString:**

**print( fruit ) print( letter )**

***#*** *“apple” “banana”* ***#*** *“J” “a” “m” “e” “s”*

**How to Loop a Specific Number of Times**

In order to create a loop that runs for a certain number of times you will need to use a **range()** function in a **for** loop.

**for num in range( 1, 6 ):**

**print( num ) *#*** *will print 1 2 3 4 5*

**range()**

The **range()** function returns a sequence of numbers. It can take in 3 parameters specifying the **start** number, the **last** number (not included) and the optional **steps** number specifying how much you increment between each number. Think of the **steps** like the **i += 1** phase of a for loop in JavaScript.

***# range(*** *start****,*** *last****,*** *steps* ***)***

**range( 1, 10, 2 ) *#*** *1 3 5 7 9*

**range( 5, 11 ) *#*** *5 6 7 8 9 10*

**range( 6 ) *#*** *0 1 2 3 4 5*

**continue and break in Loops**

Just like in JavaScript, we can use **continue** to skip over a current iteration in a loop to move on to the next iteration and **break** to exit out of a loop altogether.

**colourList = [ "red", "green", "blue", "purple", "black", "white" ]**

**for colour in colourList:**

**if colour == "green":**

**continue**

**elif colour == "purple":**

**break**

**print( colour ) *#*** *“red” “blue”*

**Else at the End of a Loop**

If you want to run some code after a loop has completed, you can do this by placing **else:** at the end of the loop. Take note that the **else:** has to be placed at the same indentation of the loop after it. Also note that the **else:** code block will not run if the loop is exited with a **break** or **return**.

**for num in range( 0, 6):**

**print( num )**

**else:**

**print( "Counting has finished" )**

**while Loops**

In Python **while** loops act very similarly to while loops in JavaScript. The same loop keywords used above in for loops apply here as well: **break**, **continue** and **else**.

**num = 0**

**while num <= 10:**

**print( num )**

**num += 1**

***#*** *0 1 2 3 4 5 6 7 8 9 10*