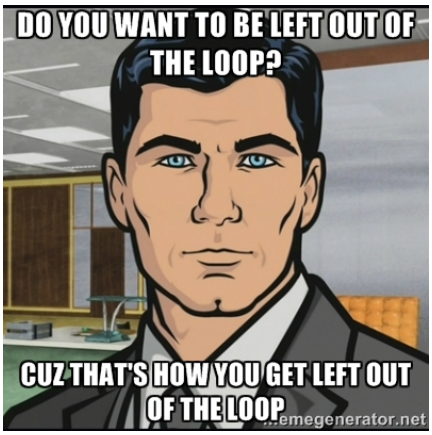
WEEK 3 REVIEW



While Loops

- Think of this like an if statement that repeats. Use the same symbols and notations as you would in an if statement. Only difference is that the code block repeats until the conditional evaluates to false.
    - Good while loop: while userInput != “exit”: { ….userInput = input(“What would you like to do now?”) }. Within this while loop, you need to make sure that you update userInput at the end of the code block .
    - Bad while loop: while userInput != “exit”: { do something }. This is BAD and you will get an INFINITE loop because your code never updates userInput so userInput will always have its initial value.
- While loops are most useful when you don’t know how many times you want the loop to repeat. For example, you want a user to be able to keep picking a choice off of a menu until they enter the designated exit command.

Exercise:

Write a while loop that will print out the menu “1) Print ‘Hello World’ 2) Print ‘Python is awesome’ 3) Print ‘Exit’” and ask the user for a number. Print this menu WHILE the user input is not equal to 3

For Loops

- Useful when you know (or can deduce) how many times the loop should run
- Similar to a while loop in that a code block repeats itself
- Unlike a while loop, a for loop is a better option if you know how many times you’d like to repeat the code block
- Format of for loop: for varName in range(int start, int end, [optional]int step).
    - for num in range(6): num will equal numbers 0-5
    - for num in range(1,6): num will equal 1-5
    - for num in range(1,6,2): num will equal all odd numbers up until 6
- End parameter is EXCLUSIVE which means “up until.” So always remember to subtract 1 from end to find the last possible number
- varName is essentially a counter. You can choose to use varName in your code block

Exercise:

1. Write a for loop that prints out the numbers 6 through 10
2. Write a for loop that prints every third number between 1 and 20
3. Write a for loop that prints every even number, and print “divisible by 4” if the number is divisible by 4

Strings

- Strings are called “sequences.” This means you can access an individual letter of a string.
- Two ways to access the individual letter:
    - for letter in string: { print(letter”) }
        - With this style, you don’t need to pay attention to location of the letter in the string
    - for index in range(len(string)): { print(string[index]) }
        - With this style, you can access the individual element of the string based on its position.

Exercise:

Write two for loops that implement both ways to access individual letters to print out each character in the phrase “Python is Awesome!” on a new line.

Errors of the Week

Time permitting, each week I’d like to talk about some common error messages you might receive while implementing the week’s material. We’ll talk about what the error means, and how to find and fix it. These will help you learn a general strategy for debugging, an invaluable tool.