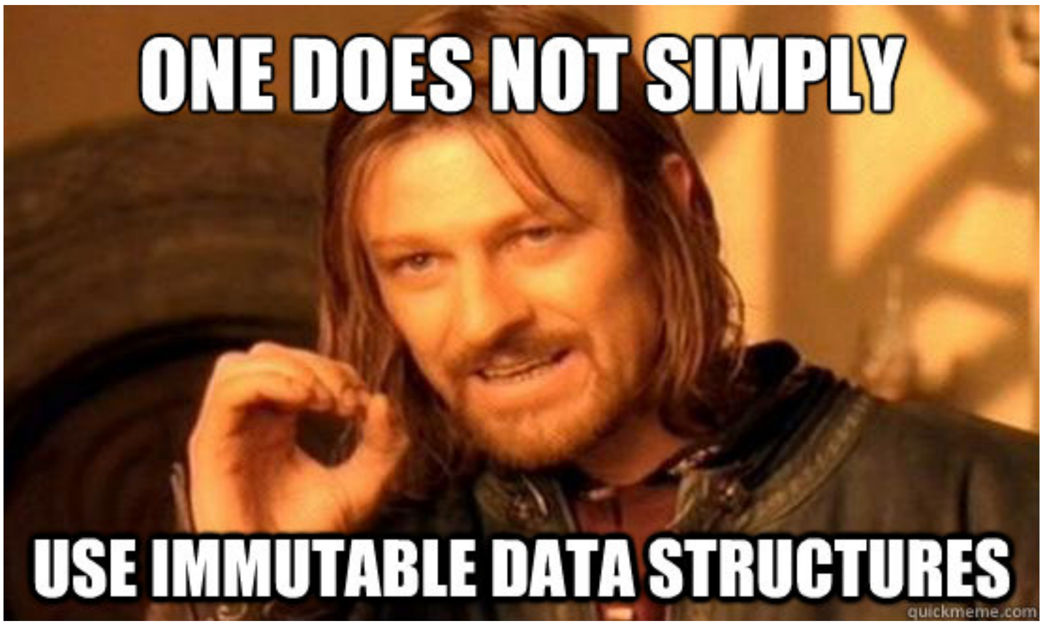
<u>WEEK 4 REVIEW</u>



For Loops: The Return of the Repeating Code

- **Iterates** through elements in a **sequence** or **collection**
    - Iterate: to perform or utter repeatedly, to loop. Also, iteration. I will use this word frequently when talking about loops.
    - Sequence: an arrangement, also called a collection
- Has a pre-defined start and end (e.g. start at 10, go until 20, start at the first letter in a word, go until the 5th letter, etc). It is bounded

Exercises:

1. Write a loop that prints all the numbers between 7 and 30, incrementing by 5
2. Ask the user for a word. Loop through the word and only print the letters that are vowels.
3. Write a for loop that has 6 iterations, and each time it asks the user for a word and adds that word to a sentence. Print the sentence every time.

Strings, Sequences, Lists aka Collections

- A collection is just an arrangement of items. These items can be numbers, letters, words, etc.
- They all have indices and can be iterated through with a for loop. Syntax is as follows: for index in range(len(collection))
- Indices in collections start at 0, not 1. So the first element in a collection is at position/index 0. The last element of a collection can ALWAYS be accessed by collection_name[len(collection)-1]
    - Side note: in order to access the element at a location. use square bracket notation. EG stringName[5], listName[0].
- Another way to iterate through a collection: for element in collection_name. This is super convenient if you don't care about the location of the element
- Strings

    - Immutable: means you can’t change individual letters of the string. EG if myString = “hello”, I can’t say myString[2] = “s”.
    - Can get substrings, also called slicing. So, myString2 = myString[1:4], myString2 would equal “ell.”
- Lists
    - Lists are a collection of elements. The type of these elements can be anything.
    - You can technically store different types of elements in the same list, however, to make your life easier, keep the type stored in the list consistent. So you can technically have a list where the first element is a string, the second is a double/float, and the third is an int. This will ultimately make your life difficult so keep the type consistent
    - Lists operate much in the same way in terms of indexing as strings do. However, lists are MUTABLE. This means that that you can change the value of an element at a specified index. So if myList = [3, 4, 2, 6, 10], I can say myList[3]=5.
    - You can have a list of lists. For example, my2DMatrix = [[2,3,4], [1,2,3], [50,60,70]].
- Tuples
    - Essentially what would happen if a list and a string had a baby
    - Tuples act like lists in that they are a collection of items and a tuple can contain words. However, like strings, they are immutable.
    - Tuple syntax: myTuple = (“word”, “wut”, “why”)
    - You can also have a tuple of lists. So myTupleOfLists = ([2,3,4], [1,3,5], [6,8,10])

<u>Exercises</u>

1. Write a for loop that asks for 5 words from the user. Store these words in a list
2. Randomly select a word from your list and only print out the vowels in the word