CSCA20 Lab 4

Objectives

- Practice using loops
- Practice writing functions that involve loops
- Practice working with lists

Basic Iteration

Working in the Python shell, begin by assigning the string `'hippopotamus'` to a variable `s`. (If your fingers are up to it, you may use the string `'hippopotomonstrosesquippedaliophobia'`.) Now do these exercises:

1. Write a loop that prints each character in `s` on a separate line
2. Write a loop that prints each character in `s` on the same line with a space after each character
3. Write a loop that prints each character in s on the same line, with a space and a comma after each character: `h , i , p , p , o , p ,` etc.
4. Write a loop similar to the previous but without a space between each character and the comma that follows it: `h, i, p, p, o, p,` etc. Notice that there is a comma after the last character. We won't worry about this for now

Loops

This section asks you to write the code for the functions specified below. There are some important aspects to these exercises:

- Check your code in the style checker
- For each of these functions, you will need to complete the full docstring first!!
- Test your functions appropriately

The functions (write them in a file called `lab4.py`):

- `count_letter(string, ch)`: Given a string and a single-character string, count all occurrences of the second string in the first and return the count
- `remove_digits(string)`: Return a new string that is the same as the given string, but with digits removed.
- `repeat_character(string, ch)`: Given a string and a single-character string, return a string consisting of the second, single character string repeated as many times as it appears in the first string.

Lists

In this section, you will write short functions or statements that involve lists. In some of the exercises, you will be expected to use list methods so that you can become familiar with the tools available to you. Remember to use the Python functions `dir` and `help` to get information about methods. You can also look at the documentation for lists:
https://docs.python.org/3/library/stdtypes.html#list
https://docs.python.org/3/library/stdtypes.html#typesseq-common
https://docs.python.org/3/library/stdtypes.html#typesseq-mutable

In all of the following instructions, try and figure out the simplest way to accomplish the task:

1. Type this assignment statement into the Python shell:
   `names = ['Menace', 'Clones', 'Sith', 'Hope', 'Back', 'Jedi']`
   For the following steps, use `names` and index/slice notation:
   a. Write an expression that produces this new list: `['Sith', 'Hope', 'Back']`
   b. Write an expression that produces this new list: `['Clones']`
   c. Write an expression that produces this new list: `['Back', 'Jedi']`
2. Given a list `L` and a value `v`, write an expression that removes the first occurrence of `v` from `L`.
3. Write an expression that prepends the string `'How are you?'` to the front of the list `['I am fine, thank you']`
4. Write code that turns `[2, 4, 99, 0, -3.5, 86.9, -101]` into `[99, 86.9, 4, 2, 0, -3.5, -101]`. You should use just two method calls. (See if you can do the transformation in one statement! If you're having trouble, ask your TA.)
5. Add to your file lab4.py a function `every_third` that takes a list as an argument and returns a new list that contains every third element of the original list, starting at index 0. Don't use slice notation

Submit your file lab4.py on MarkUs.