420-LCU-05 Programming in Python - Lab Exercise 5

February 16, 2022

Goals for this lab:

Practice with Lists, List methods and functions. Introduce For loop for strings and lists. Submit 1 python file for all parts.

Part 1 - List Basics

1. Create a variable named `ls1` that refers to a list of any six integers of your choice.
2. Create another variable named `ls2` that refers to a list of six more integers (be sure it is not the exact same as `ls1`).
3. Check whether Python considers your `ls1` to be greater than `ls2` or not. Print an appropriate message to indicate the result. I.e. `ls1` is greater than `ls2` or `ls2` is greater than `ls1`. In a comment indicate why.
4. Write the statement to subtract 2 from the final element of `ls1`. Print `ls1`. Your code prints: "ls1 = ..."
5. Write a Python statement to combine `ls1` and `ls2` into a single new list `ls3`. Print `ls3`. Your code prints: "ls3 = ..."
6. Append a value of your choosing to the list `ls1`. Check whether this affects the value of `ls3` by printing both `ls1 and` `ls3`. Your code prints 2 lines: "ls1 = ..." and "ls3 = ...". In a comment indicate if `ls3` changed.
7. Now make `ls4` an alias for `ls2` by writing: `ls4 = ls2`. Print both lists. Your code prints 2 lines: "ls2 = ..." and Your code prints: "ls4 = ...".
8. Append something to `ls2` and check and print the new value of `ls4`. Your code prints 2 lines: "ls2 = ..." and "ls4 = ...". In a comment, write what you think has happened here.

Part 2 - List Methods

1. Create a variable named `ID` whose value is a **string** that represents your ID number.
2. Create a variable named `ID_digits` and assign to it a value that is a list consisting of all of the digits in your ID number. Hint: Use the `list()` function to accomplish this. print ID and ID_digits. Your code prints 2 lines: "ID = ..." and "ID_digits = ...".
3. Write a statement or expression to count and print the number of occurrences of the digit ’1’ in `ID_digits`. Your code prints: There are x occurrences of the digit ’1’ in my ID, where x represents the actual number of occurrences.
4. Write the statement or expression to add the digit ’9’ to the **beginning** of the list ID_digits.
5. Write the statement or expression to combine all of the elements of the list back into a single string variable `new_ID` consisting just of the characters in the list (no separators). **Hint:** use `join()`. print the value of `new_ID`. Your code prints "`new_ID` = ..."

Part 3 - introducing the `for` loop

One of the most basic aspects of programming is *iteration*. Iteration is just the repetition of a computation over some sequence of values. Most programming languages express iteration using one or more *loop* statements. There are two different loop statements in Python, `while and for`. Here, we will use a few examples to show how a `for` loop can be used to access each of the elements of a string or list in order. The `for` loop works by repeating one or more statements, called the loop *body*, for each element in a sequence variable such as a string or list.

```
for x in sequence :
    statement1
    statement2
    ...
```

The variable `x` (you can use any name you like here) will be assigned the value of the first element in the sequence, and the body statements will be executed. Then the next element of the sequence will be assigned to `x`, and the list of statements will be executed again. This process continues until the last element of the sequence is used.

For this part and next, continue in the same python file and type the following code:

1. As in previous lab, define the variable `full_name` with the value of your full name.

    ```
    for letter in full_name:
        print(letter)
        print(letter.lower())
    ```

    In a comment describe what both statements inside the `for` loop do.

2. Now we'll do something a bit more interesting. Type in the following:

    ```
    result = 0
    for letter in full_name:
        print(letter, result)
        result += 1

    print(full_name, result)
    ```

    In a comment, describe what is printed, and why?

3. Here is our final example. Type in the following:

    ```
    result = ''
    for letter in full_name:
        result = letter + result

    print(result)
    ```

    In a comment, describe what is printed, and why.

Part 4 - More `for` loops

1. Assuming you still have the variable `ID_digits` defined, type

    ```
    for digit in ID_digits:
        print(digit, int(element) % 2 != 0)
    ```

    Briefly explain what this statement does.

2. Now try the following. It will do something vaguely familiar:

    ```
    result = []
    for digit in ID_digits:
        result = [digit] + result
    print(result)
    ```

    Again, observe and briefly describe what this does. In particular, are the brackets necessary in the expression `[digit] + result`? Can you leave them out?

3. Now combine the elements of `result` back into a single string:

    ```
    id_backward = ''.join(result)
    print(id_backward, 'x'.join(result))
    ```

    Be sure to show what is printed out (in a comment).