CSCA20 - Lab 2

Data Types & Strings

Learning Objectives

In this lab, we're working with converting between data types, working with built-in string methods, and learning how to find new information in the Python Docs.

Prelab

You can now get basic input from and send output to the user, so let's make sure we're ready to start playing with strings. For the start of your lab, you should show up with code that can ask the user for a word and a number, and will then print True if the word has more characters than the number. e.g.;

```
Type a word: hi
How many letters? 7
False
Type a word: supercalifragilisticexpialidocious
How many letters? 7
True
```

Hint: `print(3 > 7)` will print `False`, `print(3 < 7)` will print `True`

Demonstration & Evaluation

Successfully completing this lab and the accompanying quiz will demonstrate competency in user input/output, and variables. Note that if you have already demonstrated competency in input/output in Lab 1, you don't **need** to demonstrate it again, but as we mentioned last week, some of the fundamentals will have many chances for demonstration.

The Scenario

The Unbelievably Trained Security Council (UTSC) wants you to build them a program to help their password crackers learn how to... well.. crack passwords[1]. The program will allow the trainer to input a password, and then the trainee will go through 3 rounds of attempts to crack the password.

Round 1

After the trainer has entered the password, the trainee will be allowed to guess 5 letters. For each letter, the system will print `True` if that letter is in the password, or `False` if it isn't

Round 2

Now that the trainee knows (hopefully) the letters in the password, they will be prompted to enter the letters and their predicted positions (e.g., I think 'X' is in position 3), and will be told `True` if the letter is in that position, and `False` otherwise. The trainee will get 5 chances to guess in this round.

Round 3

At this point, the trainee has 5 chances to input the whole password. If the guessed password is correct, the system will print `True`. If a guess is incorrect, the program will print `False` (note: even if the trainee guesses correctly on the first try, the system will still prompt for all five attempts).

Starter Code

`lab2_starter.py` has been uploaded to Quercus. You should start by reading through it, and running it to make sure you understand how and why it prints the `True` and `False` values that it does. If you don't understand something, ask your TA for help. Once you understand what the starter code is doing, your job is to modify it so that it prints `True` and `False` correctly based on the input of the users.

[1] don't worry, they told me they would only use them for good... and the shadowy man in the trench-coat seemed very trustworthy

Hints

Here are a few hints that might help you with this assignment:

- Your code only needs to print out `True` or `False`, later on, when we learn about loops and selection, we can do fancy things like keeping track of the number of correct guesses, or giving them feedback on how they're doing
- You can assume that the trainee understands enough about computer science that they will know that computer scientists start counting at 0 (or you can try to change it so they don't have to, it's up to you)
- You will need to store the passcode and the trainee's guesses. But do you really need 5 separate variables for each guess of the trainee? If you won't need a value anymore, you don't need to store it and can re-use the variable to point to a new value
- We haven't covered all the features of strings you need in order to complete this lab in lectures. But we HAVE covered where you should go to find out the features of strings.

Extra Practice

If you get this working and want to move beyond the basics, there are several things you can do:

- Not all passwords are case-sensitive. Try to make a version of the program where it doesn't matter if the trainer/trainee uses upper or lower case letters
- (once you've learned loops) improve this code so that each input/print statement is only in the code once for each round
- (once you've learned selection) improve this code so that it keeps track of the number of correct guesses and gives better feedback than just `True` or `False`
- (once you've learned loops & selection) make a better version of this program that actually shows the user the parts of the password they've figured out already. e.g., if they've guessed the positions of `A` and `L` and the password is `APPLE`, prompt them with `A__L_`