

15-110 Recitation Week 4

Reminders

- HW2 due Monday, Feb. 9 @ noon
- Small groups this weekend for drawIllusion on HW2
- Recitation activity
- [Recitation feedback form](#)

Overview

- Quick string questions
- While loops
- For loops
- Strings code writing

Problems

Quick String Questions

Assuming that the following two lines have code have been run:

```
s1 = "coding is cool"
```

```
s2 = "CMU rocks!"
```

Fill out the following table with the value that each expression will evaluate to.

Expression	Value
<code>s1[1] + s2[2]</code>	
<code>s1[len(s1)-1] + s2[5]</code>	
<code>s1[3:10]</code>	
<code>s2[4:len(s2)-1]</code>	
<code>s1[::-1]</code>	

While Loops vs. For Loops

For loops allow you to set a specific range of values to iterate through ahead of time:

- General format: `for i in range(x, y, z)`
 - `i`: loop variable - contains current value of iteration
 - Can use any variable name here as long as we keep it consistent
 - `range(x,y,z)`: start value, end value, step size
 - Start value is inclusive, end value is exclusive

Python

```
for i in range(0,10,2):
    print(i)
# what would be printed?
```

While loops allow you to set a certain condition under which we keep iterating:

- General format: `while (condition)`
 - Condition will generally be some boolean expression. As long as this expression evaluates to `True`, we continue to iterate.
 - Make sure the expression evaluates to `False` at some point otherwise we end up in an infinite loop!

```
Python
i = 0
while i <= 10:
    print(i)
    i = i + 2

# what would be printed?
```

Note: the two examples are not equivalent! The for loop will not print 10 because the end value of the range is exclusive. Also, after the while loop, $i = 12$, whereas after the for loop, $i = 8$.

While vs. For Loops:

- For loops are used for a fixed number of iterations, which helps you avoid infinite loops.
- While loops require declaring an iterator variable outside the loop and updating that variable within the loop. That is abstracted away when we use for loops and `range()`.
- While loops are more versatile since the condition statement gives you more flexibility.
- You can write any for loop as a while loop!

For + While Loop Code Tracing

Open the Week 4 Python starter file and examine the function `nestedFor`. What does `nestedFor(2, 3)` print? Pay close attention to the order in which the values are printed!

Next, code trace through the function `mystery`. Pay close attention to the condition we're checking in the while loop and the update action we're using. Also consider at what point the loop stops. Try to come up with a general idea of what this function is doing.

Code Writing with Strings

Write a function `isPalindrome(s)` that takes in a string and returns **True** if it is a palindrome and **False** if it is not. Recall that a palindrome is a string that is the same front-to-back as it is back-to-front. For this solution, we are going to consider what we call the “two pointers” approach.

Ex: `isPalindrome("level")` should return **True**

Ex: `isPalindrome("lever")` should return **False**

Optional Extra Practice

Now, let's modify our `isPalindrome(s)` so that instead of using two pointers, we use one. This condenses our code a little bit.