

# 15-110 Recitation Week 4

## Reminders

- HW2 due Monday 9/27 @ Noon EDT
- Recitation feedback form: <https://forms.gle/WKrrbawKktmRu1xp9>

## Overview

- While loops
- For Loops
- Strings

# Problems

## While Loops vs. For Loops + Q/A

For loops! These 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 → iterator variable - contains current value of iteration
  - range(x,y,z) → start value, end value, step size
  - Start value is inclusive, end value is exclusive

While loops! These 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.

For loop example:

```
for i in range(0,10,2):  
    print(i)
```

Now recall an example of a similar while loop:

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

**Note:** these two examples are not equivalent! The for loop will not print 10, need to increase the end value to greater than 10. Also after the while loop, the value of i will be 12, whereas after the for loop (without changing the end value), the value of i will be 8.

While vs For Loops:

- For loops are used for a fixed number of iterations, help 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 in for loops by using range()
- While loops are more versatile, condition statement gives you more flexibility
- You can write any for loop as a while loop!

## Binary + For Loop Code Trace

Trace through the function calls below and identify the print output.

```
def f(n, bits):  
    maxValue = 2**bits - 1  
    if (n > maxValue):  
        return 'Not enough bits!'  
    result = ''  
    for k in range(bits-1, -1, -1):  
        kthBit = n // 2**k % 2  
        result = result + str(kthBit)  
    return result
```

```
print(f(3,4))
```

```
print(f(6,4))
```

```
print(f(16,4))
```

## Quick String Questions

Given the string `s = "abcdefghi"`, answer the following short answer questions:

- 1) How do I access the character "i" from string `s`?
- 2) How do I create a string `x` which is equal to "cdef" using `s`?
- 3) How do I create a string `x` which is equal to "beh" using `s`?
- 4) How do I create a string `x` which is equal to the reverse of `s`?

## Strings Functions Code Writing

Write a function `everyOther` that takes in a string `s` and prints every other letter of `s`.

Write a function `removeExclamations` that takes in a string `s` and returns a new result string with all of the exclamation marks from string `s` removed.

For example:

`removeExclamations("H!el!!lo !wo:r!ld!")` should return "Hello world"