



Property of GWC Week 5 Loops and Lists University



Something to Note



In computer programming, the order of numbers is different than you have learned in school.

School teaches you:

0,1,2,3,4,5 means you have 5 numbers

Programming:

0,1,2,3,4,5 means you have 6 numbers (count the 0)

Lists



- A list is a collection of values
 - You can have any value, and can mix values
 - `["a",3,True,4.5]`
 - Lists are mutable: you can change the elements in the list
 - Use `[]` notation for declaring lists

Lists (continued)



- You can create a list like this:
`alist = [1,2,3,"a","b"]`
- Lists have indexes: for each element stored in a list, you can access it by knowing what order (or location) in the list the element is stored
- For example in our list above “a” is at index 3
 - `alist = [1,2,3,"a","b"]`

↑ ↑ ↑ ↑ ↑
0 1 2 3 4

Lists (continued)



- You can access (or test) what is at an index in a list:
- You can also do the same using slices to see more than one element:
 - `alist[0:4]` would give you `[1,"b"]`

Your Code ...

```
1 alist = [1,2,3,"a","b"]
2 print(alist[0])
3 print(alist[4])
```

Result...

CPU Time: 0.02 sec(s), Memory: 8436 kilobyte(s)

```
1
b
```

Lists



- You can reassign elements in a list:

```
alist = [1,4,5,6]
```

```
alist[3] = "a"
```

Your new list would now be: [1,4,5,"a"]

Group Discussion



- Turn to someone next to you and talk about something that you have done that was repetitive and took a long time.
- What are some electronics (does not have to be a laptop/computer) that execute some function/activity over and over.

What is a Loop and Why are they Useful



- Using repetition in programming allows us to have a computer execute a function without needing to type it over and over.
- There are two types of loops in python:
 - While loop
 - For loop

For loops

- For loops allow you to define the number of times you want to repeat certain code:
 - You define the range in which the loop will run
 - You define an arbitrary or changing index *i*

Your Code ...

```
1 for i in range(0,5):  
2     print(i)  
3  
4  
5  
6  
-
```

Result...

CPU Time: 0.03 sec(s), Memory: 8432 kilobyte(s)

```
0  
1  
2  
3  
4
```

Range function

- The most common way to write a simple for loop uses the range() function
- `range(start,end)`
 - This will start your code at the number you give, and end the code one number less than you specified

Example:

`range(0,5) = [0,1,2,3,4]`

`range (1,6) = [1,2,3,4,5]`

Range (continued)



Another way to write the range() function is: range(end)

```
for i in range(7):
```

```
    print(i)
```

The range here would be [0,1,2,3,4,5,6] (from 0-6)
(you only need to specify your ending value)

Range (continued)



- Another detail to range is that you can alter the step size when computing with a third parameter to the function:
- $\text{range}(0,20,2) = [0,2,4,6,8,10,12,14,16,18]$ (skips by 2)
 $\text{range}(0,20,5) = [0,5,10,15]$
- You can also decrement the step by using a negative number
 $\text{range}(10,0,-1) = [10,9,8,7,6,5,4,3,2,1]$ (note the range started at 10 and ended at 0)

Activites

1. What element is at index 3 in the list: ["katie",23,4.5,"A"]
2. What is the output of the following code?
 - i. for i in range(0,4):
 - ii. print("hi")
3. Take this list: alist = [1,2,3,4,5] and change index 4 to "a"
4. Write a loop that will print out your name 5 times
5. Review question: Make a loop that will print out ("Hi" + your name) 5 times if the user input is between 0 and 5. If user input is something else, print out the user input
(hint: <= means greater than or equal to)

Bonus Questions



1. Write a loop that will find the sum of the numbers in the range(0,5)
 - a. Hint: create a variable outside the loop that equals 0 to start
2. Find the sum of the numbers in this list:
`alist = [2,3,4,5,6]`

Hint: you can do this with a loop, think about the indexes being a range