

Recitation 13 - Python (Loops, Conditions, Functions)

CSCI 1300 Fall 2017

Instructor: Dr. David Knox

Python Lists:

A list in python is an important data type. To store elements as a list, place all the elements enclosed within square brackets and separated by comma. Unlike in C++, we can have different types of elements in the same python list. Below are some of the examples of lists

```
myList = [] # empty list
```

```
myList = [1, 2, 3] # list of integers
```

```
myList = [1, "Hello", 3.4] # list with mixed datatypes
```

```
myList = ["csci1300", [2, 4, 6]] # nested list
```

Access elements in a list

To access the elements in a list we need to use the index operator []. Similar to arrays in C++, python lists index starts from 0. For example, a list having 10 elements will have index from 0 to 9.

Example:

```
myList = [1, "Hello", 3.4]
```

```
print(myList[0]) #Output is 1
```

```
print(myList[2]) #Output is 3.4
```

```
myList_1 = ["csci1300", [2, 4, 6]]
```

```
print(myList_1[0][1]) #Output is s
```

```
print(myList_1[1][2]) #Output is 6
```

Slice Lists in Python:

Python also defines a convenient method of extracting sublists of a list using slice notation. This method takes several forms. In general, we may extract every *i*th element of a list **L** starting at index **x** and going to index **y** with **L[x:y:i]**. The step size (**i**) is often excluded. The default is **i=1**. In order to get a sublist starting at the beginning of the list we may exclude the start (**x**) and to get a sublist ending at the end of the list we may exclude the end (**y**).

```
myList = ['p','y','t','h','o','n']
```

```
print(myList[1:4]) # elements from 2nd to 5th. Same as print(myList[1:4:1])... Output is ['y', 't', 'h']
```

```
print(myList[1:6:2]) # every other element starting at the 2nd going to the 6th... Output is ['y', 'h', 'n']
```

```
print(myList[2:]) # elements from 3rd to end... Output is ['t', 'h', 'o', 'n']
```

```
print(myList[:5]) # elements from the start to the 6th... Output is ['p', 'y', 't', 'h', 'o']
```

Add/Change Elements in a list:

Just like in C++ we may alter the value contained in each index of a list. This can be accomplished with a simple assignment.

```
myEven = [2, 5, 6, 8]
```

```
# change the 2nd item
```

```
myEven[1] = 4
```

```
print(myEven) #Output is [2, 4, 6, 8]
```

```
# change 2nd to 4th items
```

```
myEven[1:4] = [10, 12, 14]
```

```
print(myEven) #Output is [2, 10, 12, 14]
```

For examples of extra commands used with lists see the following:

<https://docs.python.org/3/tutorial/datastructures.html>

File I/O in Python:

The file I/O operations are similar to C++. The steps followed to open and close a file are as shown below.

- 1) Open a file in the correct mode and store it in a file object.
- 2) Use this file object to modify or read from the file as required.
- 3) Once done with all the operations in a file, make sure to close the file.

Modes Description

r : Opens a file for reading only. The file pointer is placed at the beginning of the file. This is the default mode.

r+ : Opens a file for both reading and writing. The file pointer placed at the beginning of the file.

w : Opens a file for writing only. Overwrites the file if the file exists. If the file does not exist, creates a new file for writing.

w+ : Opens a file for both writing and reading. Overwrites the existing file if the file exists. If the file does not exist, creates a new file for reading and writing.

a : Opens a file for appending. The file pointer is at the end of the file if the file exists. That is, the file is in the append mode. If the file does not exist, it creates a new file for writing.

a+ : Opens a file for both appending and reading. The file pointer is at the end of the file if the file exists. The file opens in the append mode. If the file does not exist, it creates a new file for reading and writing.

Reference: https://www.tutorialspoint.com/python/python_files_io.htm

To Write lines to a file:

```
fruits = ['Apple\n', 'Pineapple\n']  
fw = open('fruits_file.txt', 'w')  
fw.writelines(fruits)  
fw.close()
```

To Read lines from a file:

```
fr = open('fruits_file.txt')  
fr.readline() #Output is 'Apple\n'  
fr.readline() #Output is 'Pineapple\n'  
fr.close()
```

To Read lines in a loop:

```
fr = open('fruits_file.txt')  
for line in fr:  
    print(line)  
#Output is:  
#Apple  
#Pineapple  
fr.close()
```

Recitation Activity

The recitation activity this week consists of 4 Code Runner questions on Moodle. These are due by 6pm on Sunday, December 3rd.