Problem Solving and Programming in python - Day-4

Date- 14 June 2019

Day Objectives

- Python Data structures
  - Lists
  - Tuples
  - Dictionaries
- Basic problem set on Data structures
- Advanced problem set
- Packages and Modules in python

In [ ]:

Python Data Structures

Lists

In [14]:
```
li = [123, 978, 654]

li # Accessing the entire list

li[1]   # Accessing the element throw index

li[1:]   # Accessing the 2 element to end element to last element

li[ : : -1]   # reverse order in the list

li = li[::-1]     # update the orginal list to reverse order list

li  = li[::-1]     # reassigned the list to orginal list

li[::2]            # Accessing even index elements

li[1::2]        # Accessing odd index elements

#Lists can be accessed, manipulated in two different ways
 #Direct Referencing - [index]
 #Indirect Referencing - through functions


li.append(345)           #Adding an element to end of the list

li.insert(1,456)         #Adding an element at a particular index

li.sort()             # sort in asending order

li.pop()               #Remove the last element using pop

li2 = [234,456,789]


li.extend(li2)     # merge list 2 into list1

sum(li)

max(li)

min(li)

len (li)

 #try:
#cathe:

#except

sum(li[1::2])/len(li[1::2])         #Average of list elements
```

Out[14]: 485.0

In [62]:
```
#Function to identify the second largest element in a list

def secondLargest(li):
    li.sort()
    return (li[-2])

secondLargest([123,345,567])

#Function to identify the n Largest element in a list

def genericLargest(li,n):
    li.sort()
    print(li[-n])
    return
genericLargest([123,245,345,456],3)
```

```
245
```

In [12]:
```
#Function to search for data in a list and return the index return -1 if key not

def linearSearch(li,key):
    for n in  range(0,len(li)):
         if li[n] == key:
            return li[n]
    return -1

linearSearch([1,2,3,4,5],1)

def linearSearch2(li,key):
    for element in li:
        if element == key:
            return li.index(element)
    return -1

linearSearch2([1,2,34],2)


def linearSearch3(li,key):
    return

linearSearch3
```

Out[12]:
```
1
```

In [18]:
```
# Function to count the occurances of a charater in string
#"python programming", m -> 2
def Howmanytimes(s,c):
    count=0
    for ch in s:
        if ch == c:
            count += 1
    return count
def countcharacterOccurances2(s,c):
    return s.count(c)
countcharacterOccurances2("abcabcabc",'ab')
```

Out[18]: 3

In [ ]:
```
#Function to find the number of occurnaces of substring
#"abcabcabcda"   "ab"  -> 2
def characterOccurances()
```