Problem Solving and Programming in Python

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

Python Data Structures

Lists

In [39]:

```
li=[123, 345, 567]

li  # Acess the entire list

li[0] # Acess the first element in a list

li[1] # Acess the second element in a list though a index

li[1:] #Acess the all elements form second element to last element

li=li[::-1] #Acess the all elements in reverse order

li

li=li[::-1] #Acess the again the same list

li

li[::2] #Acees even index elements

li[1::2] #Acess the odd index element

###Data Manipulation
   #   Direct Reference
   #   indircet reference


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

li

li.insert(1,234)  #Adding an element at a particular index

li

li.sort()  #Sorting elements in ascending order

li

li.pop()  #Remove the last element in a list

li

li.pop(1) #Remove an element at a particular index

li

li2=[234,456,678]

li.extend(li2)
li

sum(li)
max(li)
min(li)
len(li)
```

```
57
58 ###Average of list elements
59 sum(li)
60 len(li)
61 sum(li)/len(li)
62 ###Average of all alternate elements
63 sum(li[::2])/len(li[1::2])
64 li.index(123)
65
```

Out[39]: 0

In [25]:

```
1  ### Second largest number by unsing Function
2    #sort the data and select the second last element
3    #sort the data in reverse order, and select the second
4    #Remove the max element and then get the max of the
5  li=[23,110,10,34,21]
6  def secondlargest(li):
7          li.sort()
8          return li[-2]
9
10 #secondlargest(li)
11
12 #function that returns the nth largest
13 def genericlargest(li,n):
14     li.sort()
15     return li[-n]
16 secondlargest(li)
17 genericlargest(li,5)
18
```

Out[25]: 10

In [12]:

```
1  ## Second Largest elements in lists
2  a=[]
3  n=int(input("Enter number of elements:"))
4  for i in range(1,n+1):
5      b=int(input("Enter element:"))
6      a.append(b)
7  a.sort()
8  print("Second largest element is:",a[n-2])
9
10
```

```
Enter number of elements:5
Enter element:12
Enter element:67
Enter element:87
Enter element:90
Enter element:45
Second largest element is: 87
```

In [44]:

```
##Function to search for data in a list
#Search for the key in th elist and return the index values or -1

li=[12,10,23,34,21]
def linearsearch(li,key):
    for i in range(0,len(li)):
        if li[i]==key:
            return i+1
    return -1
    #if key in li:
        #return li.index(key)
    #else:
        #return -1

def linearsearch2(li,key):
    for i in li:
        if i==key:
            return li.index(i)
        return -1

def linearsearch3(li,key):
    if key in li:

        return li.index(key)
    return -1

#linearsearch(li,10)
#linearsearch2(li,12)
linearsearch3(li,12)

```

Out[44]: 0

In [59]:

```
#Function to count the occurances of a character in a string
#"python Programming",m ->2
def countoccurance(s,c):
    count=0
    for ch in s:
        if ch == c:
            count +=1
    return count
def countoccurance2(s,c):
    return s.count(c)
#countoccurance("Python Programming",'m')
countoccurance2("Python Programming",'py')

#Function to find the number of occurances of a substring
#abcabcddcba",  ->"ab"->2

def substring(s,c):
    count=0
    for ch in s:
        if ch == c:
            count=count+1
    return count
substring("abcabcddcba","ab")
```

Out[59]: 0

In [57]:
```
1 dir(list)
```

Out[57]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__delitem__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__gt__',
 '__hash__',
 '__iadd__',
 '__imul__',
 '__init__',
 '__init_subclass__',
 '__iter__',
 '__le__',
 '__len__',
 '__lt__',
 '__mul__',
 '__ne__',
 '__new__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__reversed__',
 '__rmul__',
 '__setattr__',
 '__setitem__',
 '__sizeof__',
 '__str__',
 '__subclasshook__',
 'append',
 'clear',
 'copy',
 'count',
 'extend',
 'index',
 'insert',
 'pop',
 'remove',
 'reverse',
 'sort']
```

In [ ]:
```
1
```