Problem Solving and Programming in python-Day4

Date: 14-June-2019

Day Objectives:

- Python Data Structures
  - Lists
  - Tuples
  - Dictionaries
- Basic Problem set on Data structure
- Advance problem set
- Packages and modules in python

In [ ]:
```
1
```

Python Data Structures

Lists

In [36]:

```
## Create a list:
li = [123,234,2323]
li                      ## Accessing the entire list

li[1]                  ## Accessing an element with index in a list

li[1:]                ## Accessing all elements from second to last

li[::-1]             ## eversing the entire list

li=li[::-1]         ## Changing the original to reverse the elements in list

li=li[::-1]

li[::2]           ## Accessing even elements in the list

li[1::2]

## Lists can be accessed, manipulated in two different ways.
    ## Direct Referencing  --> Using [] brackets
    ## Indirect Referencing --> Through the functions

li.append(345)      ## Adding an element at the end of the List

li.insert(1,121)   ## Adding an element at the required index position

li.sort()          ## Sort elements in ascending oreder

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

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

li2 = [234,456,789]   ## Merge list2 into list1

li.extend(li2)

sum(li)
max(li)
len(li)

## Average of elements in the given list
sum(li)/len(li)

## Average of all the alternate elements in the even positions
sum(li[::2])/len(li[::2])
sum(li[1::2])/len(li[1::2])     ## Average of all elements in odd position


```

Out[36]: 382.0

In [95]:
```
## Function to identify the second largest element in the list
li = [123,23,12,23,23333]

def secondLargestElement(li):
    temp=0
    x=len(li)
    m=max(li)
    s=li[0]
    for j in li:
        if j > s and j <m:
            s=j
    return s



secondLargestElement(li)

```

Out[95]: 123

In [67]:
```
## Function to identify the second largest element in the list
li=[123,23,45,56,345]
def secondLargestElement(li):
    li.sort()
    li.pop()
    ma=max(li)
    return ma
secondLargestElement(li)
```

Out[67]: 123

In [97]:
```
 ## Any largest number in the given list
li=[123,23,45,56,345]
n=int(input("enter required highest position number"))
def genericlargest(li,n):
    li.sort()
    li=li[::-1]
    return ("the nth largest number is",li[n-1])

genericlargest(li,n)
```

```
enter required highest position number2
```

Out[97]: ('the nth largest number is', 123)

In [120]:

```
## Function to search for data in a list
## Search for the key in the list
## Return -1 if the data is not exists

li=[23,34,56,67,4545]
key=int(input("enter the number"))
def linearsearch(li,key):
    for i in range (0,len(li)):
        if li[i]==key:
            return i
        else:
            return -1

linearsearch(li,key)


# ## Another Method:

li=[23,45,56,4545]
def linearsearch2(li,key):
    for i in li:
        if i==key:
            return li.index(key)
    return -1

linearsearch2(li,45)



def linearsearch3(li,key):
    if key in li:
        return li.index(key)
    return -1

linearsearch3(li,45)

```

Out[120]: 1

In [129]:
```
## Function to count the occurances of a character in the given string
## "Python Programming",--> search for m is repeating....-->2

def charcheck(s,c):
    count=0
    for ch in s:
        if ch==c:
            count+=1
    return count
charcheck("programming",'m')

def countcharoccurance(s,c):
    return s.count(c)
countcharoccurance("programmingmmm","m")


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

```

Out[129]: 5

In [134]:
```
s = 'nunununbbdfdf'
sb = 'nun'
results = 0
sub_len = len(sb)
for i in range(len(s)):
    if s[i:i+sub_len] == sb:
        results += 1
print (results)
```

3

In [ ]:
```

```