Problem Solving and Programming in Python June 2019

Date - 14th 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 [58]:
```
list =[123,987,654,234]

list  # Access the entire list

list[1] # Access an element with index in a list

list[-2:] # Access all element from second element

list[::-1]  # Accessing the reverse order

list = list[::-1] # Assigning the reverse order to other variable
list  # Calling the  reverse list varible

list=list[::-1] # Again assigning the list which is going to reversing
list # Calling the reversed list

list[::2]  #Accessing the even index elements

list[1::2] # Accessing the odd index elements



# Lists can be accessed ,manipulated in
        # Direct Referencing  - [index ] it can be a list ,string,tuples
        # Indirect Referencing (this is done by using the functions. Through func

list.insert(1,424) # Adding an element at a particular position
list


list.append(241)     # Adding an element to the end of the list
list    # Element 241 is added to the end of the list


list.insert(1,424) # Adding an element at a particular position
list


list.sort() # sort elements in ascendng order
list

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

list.pop(4) # Removing the element at a particular position
list

list1=[12,13,14]

list.extend(list1) # Merge list1 into list
list


sum(list) # Gives out put of the adding the all elements in the list
```

```
max(list) # Gives the output of highest element

len(list)

list

list2=['a','b']
list.extend(list2)
list




list.pop(-2)
list

list.pop(-1)
list

list



#Average of list elements

sum(list)#Average of list elements
len(list)
avg=sum(list)/len(list)
avg


# Average of all alternate elements
avg = sum(list[::2])/len(list[::2])
avg

# Average of all odd elements

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

Out[58]: 171.0

In [87]:
```
# Function to identity the second largest element in a list
            # Sort the data and select the second last element
            # Sort the data in reverse order,and select
            # Remove the max element and then get the max of the next element

def secondLargest(l):
    l.sort()
    return l[-2]
l=[23,45,78,1,56]
secondLargest(l)

def genericLargest(l,n):
    l.sort()
    return l[-n]
secondLargest(l)
genericLargest(l,3)
```

Out[87]: 45

In [96]:
```
# Function to search for  data in alist

def linearSearch(li,key):
    if key in li:
        return True
    else:
        return False
li=[2,9,8,3,5,7]
key=int(input("key"))
linearSearch(li,key)


# short cut

def linear(list,key):
    for index in range(0,len(li)): #  or for i in li
        if li[index]==key:
            return index
    return -1
li=[34,56,2,567,234,233]
linearSearch(li,234)
```

```
key1
```

Out[96]: True

In [103]:
```
def linearSearch2(li,key):
    for element in li:
        if element == key:
            return li.index(element)
        return -1
li=[34,56,2,567,234,233]
linearSearch2(li,34)
```

Out[103]: 0

In [106]:
```
def linearSearch3(li,key):
    return li.index(key)
li=[34,56,2,567,234,233]
linearSearch3(li,234)
```

Out[106]: 4

In [111]:
```
def linearSearch4(li,key):
    if key in li:
        return li.index(key)
    else:
        return -1
li=[34,56,2,567,234,233]
key=int(input("key"))
linearSearch4(li,key)
```

```
key2222
```

Out[111]: -1

In [125]:
```
# Fucntion to count the occurances of a character in a given string
# "Python  Programming ",m--->2

def stringOccurance(String,Substring):
    return String.count(Substring)
String="Python Programming"
Substring=input("enter key")
stringOccurance(String,Substring)
```

```
enter keym
```

Out[125]: 2

In [119]:
```
# Fucntion to count the occurances of a character in a given string
# "Python  Programming ",m--->2

def countCharOccurances(s,c):
    count=0
    for ch in s:
        if ch==c:
            count +=1
    return count
s="Python  Programming"
c=input("c")
countCharOccurances(s,c)
```

```
cm
```

Out[119]:
```
2
```

In [ ]:
```
# Functions to find the number of occurances of a substring in given string

# "abcabcddcbaabdcab",------->4
```

In [ ]:

In [ ]: