Python Data structures

1)List

In [1]:
```
li=[]
type(li)
```

Out[1]:
```
list
```

In [2]:
```
li #Acces the entire list
```

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

In [3]:
```
li=[1,2,3,4] #list indexing starts from 0
li[1]
```

Out[3]:
```
2
```

In [4]:
```
li[0:]# accesing all the elements
```

Out[4]:
```
[1, 2, 3, 4]
```

In [5]:
```
li[::-1] #Reverse of a list
```

Out[5]:
```
[4, 3, 2, 1]
```

In [6]:
```
li[::2]
```

Out[6]:
```
[1, 3]
```

In [7]:
```
li[1::2]
```

Out[7]:
```
[2, 4]
```

In [9]:
```
name="sravani"
Mylist=[]
print(type(name),type(Mylist))
```

```
<class 'str'> <class 'list'>
```

In [10]:
```
names=["sravani","sravs"]
print(type(names))
print(len(names))
```

```
<class 'list'>
2
```

In [12]:
```
names[0]
```

Out[12]:
```
'sravani'
```

In [13]:
```
names[9] # since index out of range
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-13-7a8f519d9902> in <module>
----> 1 names[9]

IndexError: list index out of range
```

List Methods

Append: -To add new iten in the available list

In [14]:
```
names.append("gopi")
```

In [16]:
```
new_names=['Bharu','Naveen','Durga']
names.append(new_names)
```

```
None
```

In [17]:
```
print(names)
```

```
['sravani', 'sravs', 'gopi', ['Bharu', 'Naveen', 'Durga'], ['Bharu', 'Navee
n', 'Durga']]
```

Extend Method

In [19]:
```
names1=["sravani","sravs","gopi"]
new_names1=["Bharu","Naveen","Durga"]
names1.extend(new_names1)
print(names1)
```

```
['sravani', 'sravs', 'gopi', 'Bharu', 'Naveen', 'Durga']
```

In [20]:
```
len(names1)
```

Out[20]: 6

copy()

To take new backup of available list

In [21]:
```
sravani=[19,234,32545]
durga=sravani.copy()
print(durga)
```

```
[19, 234, 32545]
```

In [22]:
```
print(sravani)
```

```
[19, 234, 32545]
```

In [23]:
```
girls=["Sravani","Naveen"]
girls[0]="Sravs"
print(girls)
```

```
['Sravs', 'Naveen']
```

clear()

To remove all items in list

In [24]:
```
sravani.clear()
print(sravani)
```

```
[]
```

count()

It can search for given item and return the number of occurances in a list

In [28]:
```
sravani=["Kumari","Pavani","Gangadhar","Kumari"]
print(sravani)
```

```
['Kumari', 'Pavani', 'Gangadhar', 'Kumari']
```

In [33]:
```
print(sravani.count('Kumari'))
```

```
2
```

In [34]:
```
sravani.count("Pavani")
```

Out[34]:
```
1
```

In [36]:
```
numbers=[1,2,3,4,45,[2,45]]
numbers.count(2)
```

Out[36]:
```
1
```

index(item,start_index_for_searching)

return index of first occurance of the item in list if available otherwise value shows error

In [37]:
```
numbers.index(45)
```

Out[37]:
```
4
```

In [39]:
```
num=[1,2,3,1,2,3,1,2,3]
num.index(2,4)# 2,3,1,2,3
```

Out[39]: 4

Insert(index_value,item)

If given index available it can add item at given index otherwise it adds last index

In [40]:
```
num=[1,2,3,4,5,567,78]
num.insert(3,4)
print(num)
```

```
[1, 2, 3, 4, 4, 5, 567, 78]
```

In [41]:
```
num1=[10,20,30,40]
num1.insert(100,50)
```

In [42]:
```
print(num1)
```

```
[10, 20, 30, 40, 50]
```

pop()

To remove the element at the specific position

In [43]:
```
a=["s","st","rt","tre","etry","ety","etery","fgh"]
a.pop(2)
```

Out[43]: 'rt'

In [44]:
```
print(a)
```

```
['s', 'st', 'tre', 'etry', 'ety', 'etery', 'fgh']
```

remove()

removes specific item

In [47]:
```
absent=['w4rt',"sgf","wt","ert","wte","wt"]
absent.remove("wt")
```

In [48]:
```
print(absent)
```

```
['w4rt', 'sgf', 'ert', 'wte', 'wt']
```

Reverse()

Reverse the items

In [49]:
```
active=["sd","sder","rte","reter"]
active.reverse()
print(active)
```

```
['reter', 'rte', 'sder', 'sd']
```

sort()

Sort the list alphabetically

In [50]:
```
cars=['Ford','BMW','Volvo']
cars.sort()
print(cars)
```

```
['BMW', 'Ford', 'Volvo']
```

In [51]:
```
cars.sort(reverse=True)
print(cars)
```

```
['Volvo', 'Ford', 'BMW']
```

In [52]:
```
marks=[[10,20],[20,30],[40,50]]
marks.sort()
print(marks)
```

```
[[10, 20], [20, 30], [40, 50]]
```

In [64]:
```
# Function to identify the second largest element in a unique list
    #Sort the data and select the second last element
    #sort the data in reverse order and select second element
    #Remove the max element and the get the max of the new list

# Function to identify the second largest element in a unique list
def secondLargests(lis):
    lis.sort()
    print(lis[-2])
#Remove the max element and the get the max of the new list
def lar(lis,n):
    lis.sort()
    return lis[-n]
lis=[2445,456,75]
secondLargests(lis)
lar(lis,1)
```

```
456
```

Out[64]:
```
2445
```

In [1]:
```
#Given list contains all electronic items[mobile,laptop,speakers,door,files].
#If you find any item which is not related to electronics then remove it using
slicing method.
electronic=['mobile','laptop','speakers','doors','files']
print(electronic[-2:])
```

```
['doors', 'files']
```

In [2]:
```
#Find the length of the given list. Also print the minimum and maximum values
 in the list. list1 = [10,20,34,44,88]
l=[10,20,34,44,88]
print(len(l))
print(min(l))
print(max(l))
```

```
5
10
88
```

In [4]:
```
#Take a list of elements and add that list into another empty list
l1=[1,2,3,4,5]
l2=[]
l2=l1.copy()
print(l2)
```

```
[1, 2, 3, 4, 5]
```

In [5]:
```
#Sort the list of elements in ascending order and descending order.
l1.sort()
l1
```

Out[5]:
```
[1, 2, 3, 4, 5]
```

In [8]:
```
l1.sort(reverse=1)
print(l1)
```

```
[5, 4, 3, 2, 1]
```

In [10]:
```
#Given a two list of equal size create a set such that it shows the element fr
om both lists in the pair.
s=['sr','we','n']
s2=['mr','er','r']
print(*s[::-1])
```

```
n we sr
```

In [11]:
```
#Write a python program to print the sum and average of elements in the list.
m=[1,2,3,4,5]
s=sum(m)
print(s/len(m))
```

```
3.0
```

In [14]:
```
#You are given three numbers, a, b and c.
#Write a program to find the largest number
# which is less than or equal to c and leaves
# remainder b when divided by a.

a=int(input())
b=int(input())
c=int(input())
print("The largest no is",max(a,b,c))
for i in range(1,c+1):
    if(i%a==b):
        print(i)
```

```
3
2
9
The largest no is 9
2
5
8
```

In [ ]:
```

```