4th march List Data Structure Functions

In [13]:
```
l=[]
l
```

Out[13]:
```
[]
```

In [14]:
```
l.append(10)      #Adding elements in the list
l.append(30)
l.append(20)
l.append(60)
l.append(30)
l
```

Out[14]:
```
[10, 30, 20, 60, 30]
```

In [15]:
```
len(l)
```

Out[15]:
```
5
```

In [16]:
```
id(l)
```

Out[16]:
```
2290522408960
```

In [17]:
```
for i in l:
    print(i)
```

```
10
30
20
60
30
```

In [18]:
```
print(l)
```

```
[10, 30, 20, 60, 30]
```

In [19]:
```
l.append([10,20,30,'hello']) # Nested list (a list contains other list inside it)
```

In [20]:
```
l
```

Out[20]:
```
[10, 30, 20, 60, 30, [10, 20, 30, 'hello']]
```

In [21]:
```
l.remove(30)
l
```

Out[21]:
```
[10, 20, 60, 30, [10, 20, 30, 'hello']]
```

In [22]:
```
l.clear() # clear the list
```

In [23]:
```
l
```

Out[23]: []

In [24]:
```
l=[10,30,60,10,20,30]
l
```

Out[24]: [10, 30, 60, 10, 20, 30]

In [25]:
```
l1=l.copy() #copy the list
l1
```

Out[25]: [10, 30, 60, 10, 20, 30]

In [26]:
```
l.count(60) # no of times a element is occured
```

Out[26]: 1

In [27]:
```
l1.count(10)
```

Out[27]: 2

In [28]:
```
l.extend(l1) # l+l1=l
l
```

Out[28]: [10, 30, 60, 10, 20, 30, 10, 30, 60, 10, 20, 30]

In [29]:
```
print(l)
print(l1)
```

```
[10, 30, 60, 10, 20, 30, 10, 30, 60, 10, 20, 30]
[10, 30, 60, 10, 20, 30]
```

In [30]:
```
l.index(10) # it will print the index value of the element
```

Out[30]: 0

In [31]:
```
l1.index(20)
```

Out[31]: 4

In [32]:
```
l.insert(2,15) #adding elements with index no (1st index no,next element)
```

In [33]:
```
l
```

Out[33]: [10, 30, 15, 60, 10, 20, 30, 10, 30, 60, 10, 20, 30]

In [34]:
```
l1.insert(0,5)
l1
```

Out[34]: [5, 10, 30, 60, 10, 20, 30]

In [35]:
```
l.append(1+2j)
l
```

Out[35]: [10, 30, 15, 60, 10, 20, 30, 10, 30, 60, 10, 20, 30, (1+2j)]

In [36]:
```
l.pop() # removes the element by index wise,if no argument is passed by default las
l
```

Out[36]: [10, 30, 15, 60, 10, 20, 30, 10, 30, 60, 10, 20, 30]

In [37]:
```
l1.pop()
l1
```

Out[37]: [5, 10, 30, 60, 10, 20]

In [38]:
```
l
```

Out[38]: [10, 30, 15, 60, 10, 20, 30, 10, 30, 60, 10, 20, 30]

In [39]:
```
l.pop(1)
```

Out[39]: 30

In [40]:
```
l
```

Out[40]: [10, 15, 60, 10, 20, 30, 10, 30, 60, 10, 20, 30]

In [41]:
```
l.pop()
```

Out[41]: 30

In [42]:
```
l
```

Out[42]: [10, 15, 60, 10, 20, 30, 10, 30, 60, 10, 20]

In [43]:
```
print(l)
print(l1)
```

```
[10, 15, 60, 10, 20, 30, 10, 30, 60, 10, 20]
[5, 10, 30, 60, 10, 20]
```

In [44]:
```
l.remove(10) # removes the element
l
```

Out[44]: [15, 60, 10, 20, 30, 10, 30, 60, 10, 20]

In [45]:
```
l1.remove(20)
l1
```

Out[45]: [5, 10, 30, 60, 10]

In [46]:
```
print(l)
print(l1)
```

```
[15, 60, 10, 20, 30, 10, 30, 60, 10, 20]
[5, 10, 30, 60, 10]
```

In [47]:
```
l.reverse()
```

In [48]:
```
l
```

Out[48]: [20, 10, 60, 30, 10, 30, 20, 10, 60, 15]

In [49]:
```
l.sort() # Sorting (Ascending order)
```

In [50]:
```
l
```

Out[50]: [10, 10, 10, 15, 20, 20, 30, 30, 60, 60]

In [51]:
```
l.sort()
```

In [52]:
```
l
```

Out[52]: [10, 10, 10, 15, 20, 20, 30, 30, 60, 60]

In [53]:
```
l1.sort()
```

In [54]:
```
l1
```

Out[54]: [5, 10, 10, 30, 60]

In [55]:
```
print(l)
print(l1)
```

```
[10, 10, 10, 15, 20, 20, 30, 30, 60, 60]
[5, 10, 10, 30, 60]
```

In [56]:
```
l.remove(10)
```

In [57]:
```
l
```

Out[57]: [10, 10, 15, 20, 20, 30, 30, 60, 60]

In [58]:
```
l.sort(reverse = True) # Sorting (Descending order)
```

In [59]:
```
l
```

Out[59]: [60, 60, 30, 30, 20, 20, 15, 10, 10]

In [61]:
```
print(l)
print(l1)
```

```
[60, 60, 30, 30, 20, 20, 15, 10, 10]
[5, 10, 10, 30, 60]
```

In [62]:
```
l2=[1,2,3,'hi',(2+5j)]
l2
```

Out[62]:
```
[1, 2, 3, 'hi', (2+5j)]
```

In [63]:
```
l2.sort()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[63], line 1
----> 1 l2.sort()

TypeError: '<' not supported between instances of 'str' and 'int'
```

In [64]:
```
l3=['v','i','s','h','a','l']
l3
```

Out[64]:
```
['v', 'i', 's', 'h', 'a', 'l']
```

In [65]:
```
l3.sort()
```

In [66]:
```
l3
```

Out[66]:
```
['a', 'h', 'i', 'l', 's', 'v']
```

In [67]:
```
l
```

Out[67]:
```
[60, 60, 30, 30, 20, 20, 15, 10, 10]
```

In [70]:
```
l[::-1] # Start from the end and go backwards to the beginning
```

Out[70]:
```
[10, 10, 15, 20, 20, 30, 30, 60, 60]
```

In [ ]:
```

```