LIST- iN CLASS 3RD MARCH

In [2]:
```
l=[]
l
```

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

In [5]:
```
len(l)
```

Out[5]:
```
0
```

In [6]:
```
l.append(10)
```

In [7]:
```
l
```

Out[7]:
```
[10]
```

In [8]:
```
len(l)
```

Out[8]:
```
1
```

In [9]:
```
l
```

Out[9]:
```
[10]
```

In [10]:
```
l.append(20)
l.append(30)
l.append(40)
l.append(50)
```

In [11]:
```
l
```

Out[11]:
```
[10, 20, 30, 40, 50]
```

In [12]:
```
print(type(l))
```

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

In [13]:
```
len(l)
```

Out[13]:
```
5
```

In [14]:
```
l
```

Out[14]:
```
[10, 20, 30, 40, 50]
```

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

Out[15]:
```
1790217331200
```

In [17]:
```
a= True
type(a)
```

Out[17]:
```
bool
```

In [18]:
```
import keyword
keyword.kwlist
```

Out[18]:
```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

In [19]:
```
len(keyword.kwlist)
```

Out[19]:
```
35
```

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

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

In [21]:
```
l[:]
```

Out[21]:
```
[10, 20, 30, 40, 50]
```

In [22]:
```
l[0]
```

Out[22]:
```
10
```

In [23]:
```
l[1]
```

Out[23]: 20

In [24]:
```
l[-3]
```

Out[24]: 30

In [25]:
```
l
```

Out[25]: [10, 20, 30, 40, 50]

In [43]:
```
l1=l.copy()
l1
```

Out[43]: [10, 20, 30, 40, 50]

In [44]:
```
l==l1
```

Out[44]: True

In [45]:
```
print(len(l))
print(len(l1))
```

```
5
5
```

In [46]:
```
l1
```

Out[46]: [10, 20, 30, 40, 50]

In [47]:
```
l1.append(2.3)
l1.append(True)
l1.append(1+2j)
```

In [48]:
```
l1
```

Out[48]: [10, 20, 30, 40, 50, 2.3, True, (1+2j)]

In [49]:
```
l1.append(50)
l1
```

Out[49]: [10, 20, 30, 40, 50, 2.3, True, (1+2j), 50]

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

Out[50]: [10, 20, 30, 40, 50]

In [51]:
```
l.count(10)
```

Out[51]: 1

In [52]:
```
l.count(40)
```

Out[52]: 1

In [53]:
```
l
```

Out[53]: [10, 20, 30, 40, 50]

In [54]:
```
l.count(100)
```

Out[54]: 0

In [55]:
```
l
```

Out[55]: [10, 20, 30, 40, 50]

In [56]:
```
l1
```

Out[56]: [10, 20, 30, 40, 50, 2.3, True, (1+2j), 50]

In [57]:
```
l2=l1.copy()
```

In [58]:
```
l2
```

Out[58]: [10, 20, 30, 40, 50, 2.3, True, (1+2j), 50]

In [59]:
```
l2.remove(True)
```

In [60]:
```
l2
```

Out[60]: [10, 20, 30, 40, 50, 2.3, (1+2j), 50]

In [62]:
```
l2.remove(1+2j)
```

In [63]:
```
l2
```

Out[63]: [10, 20, 30, 40, 50, 2.3, 50]

In [64]:
```
l2.clear()
```

In [65]:
```
l2
```

Out[65]: []

In [66]:
```
del l2
```

In [67]:
```
l2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[67], line 1
----> 1 l2

NameError: name 'l2' is not defined
```

In [68]:
```
l1
```

Out[68]: [10, 20, 30, 40, 50, 2.3, True, (1+2j), 50]

In [69]:
```
l1[0]
```

Out[69]: 10

In [71]:
```
type(l1[0])
```

Out[71]: int

In [72]:
```
type(l1[7])
```

Out[72]: complex

In [73]:
```
print(type(l1[7]))
```

```
<class 'complex'>
```

4th march - list

In [74]:
```
print(l)
print(l1)
```

```
[10, 20, 30, 40, 50]
[10, 20, 30, 40, 50, 2.3, True, (1+2j), 50]
```

In [75]:
```
print(len(l))
print(len(l1))
```

```
5
9
```

In [76]:
```
l
```

Out[76]: [10, 20, 30, 40, 50]

In [77]:
```
for i in l:
    print(i)
```

```
10
20
30
40
50
```

In [78]:
```
l.append([1,2,3,'hi']) #nested list
l
```

Out[78]: [10, 20, 30, 40, 50, [1, 2, 3, 'hi']]

In [79]:
```
l.remove(40)
```

In [80]:
```
l
```

Out[80]: [10, 20, 30, 50, [1, 2, 3, 'hi']]

In [81]:
```
l[4]
```

Out[81]: [1, 2, 3, 'hi']

In [85]:
```
l
```

Out[85]: [10, 20, 30, 50, [1, 2, 3, 'hi']]

In [86]:
```
l.pop()
```

Out[86]: [1, 2, 3, 'hi']

In [87]:
```
l
```

Out[87]: [10, 20, 30, 50]

In [88]:
```
l1
```

Out[88]: [10, 20, 30, 40, 50, 2.3, True, (1+2j), 50]

In [89]:
```
l1.pop()
```

Out[89]: 50

In [90]:
```
l1
```

Out[90]: [10, 20, 30, 40, 50, 2.3, True, (1+2j)]

In [91]:
```
l1.pop()
```

Out[91]: (1+2j)

In [92]:
```
l1
```

Out[92]: [10, 20, 30, 40, 50, 2.3, True]

In [94]:
```
l1.pop(-1)
```

Out[94]: True

In [95]:
```
l1
```

Out[95]: [10, 20, 30, 40, 50, 2.3]

In [96]:
```
l1.pop(3)
```

Out[96]: 40

In [97]:
```
l1
```

Out[97]: [10, 20, 30, 50, 2.3]

In [98]:
```
print(l)
print(l1)
```

```
[10, 20, 30, 50]
[10, 20, 30, 50, 2.3]
```

In [99]:
```
l
```

Out[99]: [10, 20, 30, 50]

In [100…
```
l.insert(35,3)
```

In [101…
```
l
```

Out[101… [10, 20, 30, 50, 3]

In [102…
```
l.insert(3,35)
```

In [103…
```
l
```

Out[103… [10, 20, 30, 35, 50, 3]

In [104…
```
l1
```

Out[104… [10, 20, 30, 50, 2.3]

In [105…
```
l1.insert(15,1)
l1
```

Out[105… [10, 20, 30, 50, 2.3, 1]

In [106…
```
l1.insert(1,15)
```

In [107…
```
l1
```

Out[107… [10, 15, 20, 30, 50, 2.3, 1]

In [108…
```
l2=[]
l2
```

Out[108… []

In [109…
```
l2.extend(l1)
```

In [110…
```
l2
```

Out[110… [10, 15, 20, 30, 50, 2.3, 1]

In [111…
```
l
```

Out[111… [10, 20, 30, 35, 50, 3]

In [112…
```
l1
```

Out[112… [10, 15, 20, 30, 50, 2.3, 1]

In [113…
```
l
```

Out[113… [10, 20, 30, 35, 50, 3]

In [114…
```
l.extend(l1)
```

In [115…
```
l
```

Out[115… [10, 20, 30, 35, 50, 3, 10, 15, 20, 30, 50, 2.3, 1]

In [116…
```
print(l)
```

```
[10, 20, 30, 35, 50, 3, 10, 15, 20, 30, 50, 2.3, 1]
```

In [117…
```
print(l1)
```

```
[10, 15, 20, 30, 50, 2.3, 1]
```

In [118…
```
print(l2)
```

```
[10, 15, 20, 30, 50, 2.3, 1]
```

In [119…
```
l2.index(30)
```

Out[119… 3

In [120…
```
l
```

Out[120… [10, 20, 30, 35, 50, 3, 10, 15, 20, 30, 50, 2.3, 1]

In [121…
```
l.index(30)
```

Out[121… 2

In [122…
```
l
```

Out[122… [10, 20, 30, 35, 50, 3, 10, 15, 20, 30, 50, 2.3, 1]

In [123…
```
l1
```

Out[123… [10, 15, 20, 30, 50, 2.3, 1]

In [124…
```
l1.sort()
```

In [125…
```
l1
```

Out[125… [1, 2.3, 10, 15, 20, 30, 50]

In [126…
```
l1.sort(reverse=True) # Descending ORder
```

In [127…
```
l1
```

Out[127… [50, 30, 20, 15, 10, 2.3, 1]

In [128…
```
l3=[3,100,4]
l3
```

Out[128… [3, 100, 4]

In [129…
```
type(l3)
```

Out[129… list

In [130…
```
l3.sort(reverse=True)
```

In [131…
```
l3
```

Out[131…
```
[100, 4, 3]
```

In [132…
```
l6 = [3, 5.6, 'a', 1+2j]
```

In [133…
```
l6.sort()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[133], line 1
----> 1 l6.sort()

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

In [134…
```
l5 = ['z', 'm', 'n', 'b']
l5
```

Out[134…
```
['z', 'm', 'n', 'b']
```

In [135…
```
l5.sort()
l5
```

Out[135…
```
['b', 'm', 'n', 'z']
```

In [136…
```
l1
```

Out[136…
```
[50, 30, 20, 15, 10, 2.3, 1]
```

In [137…
```
l1.reverse()
```

In [138…
```
l1
```

Out[138…
```
[1, 2.3, 10, 15, 20, 30, 50]
```

In [139…
```
l
```

Out[139…
```
[10, 20, 30, 35, 50, 3, 10, 15, 20, 30, 50, 2.3, 1]
```

In [140…
```
l[::-1]
```

Out[140…
```
[1, 2.3, 50, 30, 20, 15, 10, 3, 50, 35, 30, 20, 10]
```