In [ ]:
```
#Data Structure
#List, Tuple, Set, Dict, Range
```

In [ ]:
```
#List
```

In [1]:
```
#Empty List
l = []
type(l)
```

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

In [2]:
```
len(l)
```

Out[2]:
```
0
```

In [ ]:
```
#List Functions
```

In [3]:
```
l.append()
l
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[3], line 1
----> 1 l.append()
      2 l

TypeError: list.append() takes exactly one argument (0 given)
```

In [5]:
```
l.append(10)
l
```

Out[5]:
```
[10]
```

In [6]:
```
l.append(20)
l.append(30)
l
```

Out[6]:
```
[10, 20, 30]
```

In [7]:
```
len(l)
```

Out[7]:
```
3
```

In [9]:
```
l1 = [10, 'A', True, 1+2j, 10.5]
l1
```

Out[9]:
```
[10, 'A', True, (1+2j), 10.5]
```

In [10]:
```
len(l1)
```

Out[10]:
```
5
```

In [11]:
```
print(id(l))
print(id(l1))
```

```
2580456997760
2580480887552
```

In [12]:
```
l2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[12], line 1
----> 1 l2

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

In [14]:
```
l2 = l1.copy()
l2
```

Out[14]: [10, 'A', True, (1+2j), 10.5]

In [15]:
```
l1 == l2
```

Out[15]: True

In [16]:
```
l == l2
```

Out[16]: False

In [17]:
```
print(l)
print(l1)
print(l2)
```

```
[10, 20, 30]
[10, 'A', True, (1+2j), 10.5]
[10, 'A', True, (1+2j), 10.5]
```

In [18]:
```
l != l2
```

Out[18]: True

In [ ]:
```
#Indexing in List
```

In [ ]:
```
#Slicing
```

In [19]:
```
l[:]
```

Out[19]: [10, 20, 30]

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

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

In [ ]:
```
#Forward index
```

In [21]:
```
l[0]
```

Out[21]: 10

In [ ]:
```
#Backward index
```

In [22]:
```
l[-1]
```

Out[22]: 50

In [23]:
```
l[10]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[23], line 1
----> 1 l[10]

IndexError: list index out of range
```

In [24]:
```
#Mutable/Changeable/Hashable
l
```

Out[24]: [10, 20, 30, 40, 50]

In [25]:
```
l[0] = 100
l
```

Out[25]: [100, 20, 30, 40, 50]

In [26]:
```
l[-1] = 'Jan'
l
```

Out[26]: [100, 20, 30, 40, 'Jan']

In [27]:
```
l.append('Feb')
l
```

Out[27]: [100, 20, 30, 40, 'Jan', 'Feb']

In [ ]:
```
#Duplicates are allowed
```

In [28]:
```
l.append(40)
l
```

Out[28]: [100, 20, 30, 40, 'Jan', 'Feb', 40]

In [29]:
```
l2
```

Out[29]: [10, 'A', True, (1+2j), 10.5]

In [30]:
```
len(l2)
```

Out[30]: 5

In [31]:
```
l2.clear()
l2
```

Out[31]: []

In [32]:
```
len(l2)
```

Out[32]: 0

In [33]:
```
del l2
l2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[33], line 2
      1 del l2
----> 2 l2

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

In [34]:
```
l
```

Out[34]: [100, 20, 30, 40, 'Jan', 'Feb', 40]

In [35]:
```
l.count(100)
```

Out[35]: 1

In [36]:
```
l.count(40)
```

Out[36]: 2

In [37]:
```
l.count(1000)
```

Out[37]: 0

In [38]:
```
print(l)
print(l1)
```

```
[100, 20, 30, 40, 'Jan', 'Feb', 40]
[10, 'A', True, (1+2j), 10.5]
```

In [39]:
```
print(len(l))
print(len(l1))
```

```
7
5
```

In [40]:
```
#List membership
100 in l
```

Out[40]: True

In [54]:
```
l2 = []
print(l)
print(l1)
print(l2)
```

```
[100, 20, 30, 40, 'Jan', 'Feb', 40]
[10, 'A', True, (1+2j), 10.5]
[]
```

In [55]:
```
l2.extend(l)
l2
```

Out[55]: [100, 20, 30, 40, 'Jan', 'Feb', 40]

In [56]:
```
len(l2)
```

Out[56]: 7

In [57]:
```
l2.extend(l1)
```

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

Out[58]: [100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A', True, (1+2j), 10.5]

In [59]:
```
len(l2)
```

Out[59]: 12

In [60]:
```
l
```

Out[60]: [100, 20, 30, 40, 'Jan', 'Feb', 40]

In [61]:
```
l.index('Jan')
```

Out[61]: 4

In [ ]:
```
#Forward Index & Slicing
```

In [62]:
```
l2
```

Out[62]: [100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A', True, (1+2j), 10.5]

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

Out[63]: [100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A', True, (1+2j), 10.5]

In [64]:
```
l2[0:6]
```

Out[64]: [100, 20, 30, 40, 'Jan', 'Feb']

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

Out[65]:
```
[100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A', True, (1+2j), 10.5]
```

In [66]:
```
l2[3:10]
```

Out[66]:
```
[40, 'Jan', 'Feb', 40, 10, 'A', True]
```

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

Out[67]:
```
[100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A', True, (1+2j), 10.5]
```

In [68]:
```
l2[::2]
```

Out[68]:
```
[100, 30, 'Jan', 40, 'A', (1+2j)]
```

In [70]:
```
l2[::5]
```

Out[70]:
```
[100, 'Feb', (1+2j)]
```

In [71]:
```
l2
```

Out[71]:
```
[100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A', True, (1+2j), 10.5]
```

In [72]:
```
l2[::4]
```

Out[72]:
```
[100, 'Jan', 'A']
```

In [ ]:
```
#Backward Index & Slicing
```

In [73]:
```
l2
```

Out[73]:
```
[100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A', True, (1+2j), 10.5]
```

In [74]:
```
l2[-5]
```

Out[74]:
```
10
```

In [75]:
```
l2[:-9]
```

Out[75]:
```
[100, 20, 30]
```

In [76]:
```
l2[:-3]
```

Out[76]:
```
[100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A']
```

In [77]:
```
l2[-3:]
```

Out[77]:
```
[True, (1+2j), 10.5]
```

In [78]:
```
l2[3:-3]
```

Out[78]:
```
[40, 'Jan', 'Feb', 40, 10, 'A']
```

In [ ]:
```
#Reverse
```

In [79]:
```
l2[::-1]
```

Out[79]:
```
[10.5, (1+2j), True, 'A', 10, 40, 'Feb', 'Jan', 40, 30, 20, 100]
```

In [81]:
```
l2
```

Out[81]:
```
[100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A', True, (1+2j), 10.5]
```

In [80]:
```
l2[::-3]
```

Out[80]:
```
[10.5, 'A', 'Feb', 30]
```

In [82]:
```
l2
```

Out[82]:
```
[100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A', True, (1+2j), 10.5]
```

In [83]:
```
l2.remove(1+2j)
l2
```

Out[83]:
```
[100, 20, 30, 40, 'Jan', 'Feb', 40, 10, 'A', True, 10.5]
```

In [84]:
```
l2.remove(40)
l2
```

Out[84]:
```
[100, 20, 30, 'Jan', 'Feb', 40, 10, 'A', True, 10.5]
```

In [ ]:
```
#23rd Oct Lecture
```

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

Out[85]:
```
[100, 20, 30, 40, 'Jan', 'Feb', 40]
```

In [86]:
```
l.append(10)
```

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

Out[87]:
```
[100, 20, 30, 40, 'Jan', 'Feb', 40, 10]
```

In [88]:
```
l.insert(2,25)
l
```

Out[88]:
```
[100, 20, 25, 30, 40, 'Jan', 'Feb', 40, 10]
```

In [89]:
```
l.insert(4,35)
l
```

Out[89]:
```
[100, 20, 25, 30, 35, 40, 'Jan', 'Feb', 40, 10]
```

In [ ]:
```
#Nested List
```

In [90]:
```
l.insert(2,'ABC')
```

In [91]:
```
l
```

Out[91]:
```
[100, 20, 'ABC', 25, 30, 35, 40, 'Jan', 'Feb', 40, 10]
```

In [92]:
```
l.insert(3,1+2j)
l
```

Out[92]:
```
[100, 20, 'ABC', (1+2j), 25, 30, 35, 40, 'Jan', 'Feb', 40, 10]
```

In [93]:
```
l.insert(4,[1,2,3])
```

In [94]:
```
l
```

Out[94]:
```
[100, 20, 'ABC', (1+2j), [1, 2, 3], 25, 30, 35, 40, 'Jan', 'Feb', 40, 10]
```

In [95]:
```
l.insert(12,[2,3,4])
l
```

Out[95]:
```
[100,
 20,
 'ABC',
 (1+2j),
 [1, 2, 3],
 25,
 30,
 35,
 40,
 'Jan',
 'Feb',
 40,
 [2, 3, 4],
 10]
```

In [96]:
```
l.append([7,8,9])
l
```

Out[96]:
```
[100,
 20,
 'ABC',
 (1+2j),
 [1, 2, 3],
 25,
 30,
 35,
 40,
 'Jan',
 'Feb',
 40,
 [2, 3, 4],
 10,
 [7, 8, 9]]
```

In [97]:
```
l4 = [1,2,3,4,5]
l4
```

Out[97]:
```
[1, 2, 3, 4, 5]
```

In [98]:
```
l4.insert([1,2,3])
l
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[98], line 1
----> 1 l4.insert([1,2,3])
      2 l

TypeError: insert expected 2 arguments, got 1
```

In [99]:
```
l4.append('ABC')
l4
```

Out[99]:
```
[1, 2, 3, 4, 5, 'ABC']
```

In [100…
```
l4.insert(2,[1,2,3])
```

In [101…
```
l4
```

Out[101…
```
[1, 2, [1, 2, 3], 3, 4, 5, 'ABC']
```

In [102…
```
l4.remove([1,2,3])
```

In [103…
```
l4
```

Out[103…
```
[1, 2, 3, 4, 5, 'ABC']
```

In [105…
```
l4.remove('ABC')
l4
```

Out[105…
```
[1, 2, 3, 4, 5]
```

In [106…
```
l4.insert(4,[1,2,3])
l4
```

Out[106…
```
[1, 2, 3, 4, [1, 2, 3], 5]
```

In [ ]:
```
#Differnce between pop and remove functions
#remove() - give the element value
#pop() - give the index value
```

In [107…
```
l
```

Out[107…
```
[100,
 20,
 'ABC',
 (1+2j),
 [1, 2, 3],
 25,
 30,
 35,
 40,
 'Jan',
 'Feb',
 40,
 [2, 3, 4],
 10,
 [7, 8, 9]]
```

In [108…
```
l.pop()
l
```

Out[108…
```
[100,
 20,
 'ABC',
 (1+2j),
 [1, 2, 3],
 25,
 30,
 35,
 40,
 'Jan',
 'Feb',
 40,
 [2, 3, 4],
 10]
```

In [ ]:
```
#Means if pop() is given with no parameters, will remove the last element
```

In [ ]:
```
# if I want to remove 'ABC' then using pop, give index value ie 2. pop first pops t
```

In [109…
```
l.pop(2)
```

Out[109…
```
'ABC'
```

In [110…
```
l
```

Out[110…
```
[100, 20, (1+2j), [1, 2, 3], 25, 30, 35, 40, 'Jan', 'Feb', 40, [2, 3, 4], 10]
```

In [ ]:
```
# if I want to remove 'Jan' using remove, give the element value
```

In [111…
```
l.remove('Jan')
l
```

Out[111…
```
[100, 20, (1+2j), [1, 2, 3], 25, 30, 35, 40, 'Feb', 40, [2, 3, 4], 10]
```

In [112…
```
l.remove()
l
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[112], line 1
----> 1 l.remove()
      2 l

TypeError: list.remove() takes exactly one argument (0 given)
```

In [113…
```
l.sort()   #works on same datatype
l
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[113], line 1
----> 1 l.sort()
      2 l

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

In [114…
```
l5 = [30,40,10,1000,70,30]
l5.sort()
l5
```

Out[114…
```
[10, 30, 30, 40, 70, 1000]
```

In [116…
```
l6 = ['Sidhesh','Rashmi','Balurkar','Saanchi']
l6.sort()
l6
```

Out[116…
```
['Balurkar', 'Rashmi', 'Saanchi', 'Sidhesh']
```

In [117…
```
l7 = [(1+2j),(8+6j),(8+5j)]
l7.sort
l7
```

Out[117…
```
[(1+2j), (8+6j), (8+5j)]
```

In [118…
```
l8 = [2,6.5,8]
l8.sort
l8
```

Out[118… [2, 6.5, 8]

In [ ]:
```
#Descending order
```

In [119…
```
l7.sort(reverse=True)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[119], line 1
----> 1 l7.sort(reverse=True)

TypeError: '<' not supported between instances of 'complex' and 'complex'
```

In [120…
```
l8.sort(reverse=True)
```

In [121…
```
l8
```

Out[121… [8, 6.5, 2]

In [122…
```
l6.sort(reverse=True)
l6
```

Out[122… ['Sidhesh', 'Saanchi', 'Rashmi', 'Balurkar']

In [123…
```
l5
```

Out[123… [10, 30, 30, 40, 70, 1000]

In [124…
```
l5.sort(reverse=True)
l5
```

Out[124… [1000, 70, 40, 30, 30, 10]

In [125…
```
l5.index(4)
l5
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[125], line 1
----> 1 l5.index(4)
      2 l5

ValueError: 4 is not in list
```

In [126…
```
l5.index(70)
```

Out[126… 1

In [127…
```
l
```

Out[127… [100, 20, (1+2j), [1, 2, 3], 25, 30, 35, 40, 'Feb', 40, [2, 3, 4], 10]

In [129…
```
l9 = [1,4,8,9]
print(l9[0])
print(l9[1])
print(l9[2])
print(l9[3])
```

```
1
4
8
9
```

In [131…
```
for i in l9:
    print(i)
```

```
1
4
8
9
```

In [132…
```
l
```

Out[132…
```
[100, 20, (1+2j), [1, 2, 3], 25, 30, 35, 40, 'Feb', 40, [2, 3, 4], 10]
```

In [133…
```
l[8]
```

Out[133…
```
'Feb'
```

In [136…
```
print(l[8][0])
print(l[8][1])
print(l[8][2])
```

```
F
e
b
```

In [137…
```
print(l[2][0])
print(l[2][1])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[137], line 1
----> 1 print(l[2][0])
      2 print(l[2][1])

TypeError: 'complex' object is not subscriptable
```

In [138…
```
print(l[3][0])
print(l[3][1])
print(l[3][2])
```

```
1
2
3
```

In [139…
```
print(l[4][0])
print(l[4][1])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[139], line 1
----> 1 print(l[4][0])
      2 print(l[4][1])

TypeError: 'int' object is not subscriptable
```

In [140…
```
l
```

Out[140…
```
[100, 20, (1+2j), [1, 2, 3], 25, 30, 35, 40, 'Feb', 40, [2, 3, 4], 10]
```

In [141…
```
for i in l:
    print(i)
```

```
100
20
(1+2j)
[1, 2, 3]
25
30
35
40
Feb
40
[2, 3, 4]
10
```

In [142…
```
for i in enumerate(l):
    print(i)
```

```
(0, 100)
(1, 20)
(2, (1+2j))
(3, [1, 2, 3])
(4, 25)
(5, 30)
(6, 35)
(7, 40)
(8, 'Feb')
(9, 40)
(10, [2, 3, 4])
(11, 10)
```

In [ ]:
```
# enumerate create pairs
```

In [143…
```
all(l)
```

Out[143…
```
True
```

In [144…
```
any(l)
```

Out[144…
```
True
```

In [145…
```
l11 = [1,2,3,4,5,0]
all(l11)
```

Out[145… False

In [146…
```
any(l11)
```

Out[146… True

In [147…
```
l12 = [1,2,3,4,5]
all(l12)
```

Out[147… True

In [148…
```
any(l12)
```

Out[148… True

In [149…
```
l12.append('ABC')
l12
```

Out[149… [1, 2, 3, 4, 5, 'ABC']

In [151…
```
all(l12)
```

Out[151… True

In [152…
```
l12.pop()
l12
```

Out[152… [1, 2, 3, 4, 5]

In [153…
```
l12.append(3.5)
l12
```

Out[153… [1, 2, 3, 4, 5, 3.5]

In [154…
```
all(l12)
```

Out[154… True

In [ ]:
```
# List Code - Practice exercise
```

In [155…
```
txt = "    Rashmi is my name    "
txt.lstrip()
```

Out[155… 'Rashmi is my name   '

In [156…
```
txt.rstrip()
```

Out[156… '   Rashmi is my name'

In [157…
```
txt.strip()
```

Out[157…
```
'Rashmi is my name'
```

In [158…
```
txt
```

Out[158…
```
'    Rashmi is my name    '
```

In [ ]:
```
#Using escape character
```

In [159…
```
txt = "My favourite TV serial is "Kaun Banega Crorepati""
```

```
  Cell In[159], line 1
    txt = "My favourite TV serial is "Kaun Banega Crorepati""
                                      ^
SyntaxError: invalid syntax
```

In [160…
```
txt = "My favourite TV serial is \"Kaun Banega Crorepati\""
txt
```

Out[160…
```
'My favourite TV serial is "Kaun Banega Crorepati"'
```

In [ ]:
```
#List creation
```

In [161…
```
list1 = []
print(type(list1))
```

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

In [162…
```
#List of integers
list2 = [1,5,8,9]
list2
```

Out[162…
```
[1, 5, 8, 9]
```

In [163…
```
#List of float numbers
list3 = [3.7, 8.9, 1.4]
list3
```

Out[163…
```
[3.7, 8.9, 1.4]
```

In [164…
```
#List of string
list4 = ['ABC', 'PQR', "Rashmi"]
list4
```

Out[164…
```
['ABC', 'PQR', 'Rashmi']
```

In [165…
```
#Mixed datatypes list
list5 = [10,40.8,"Rashmi",1+2j,True]
list5
```

Out[165…
```
[10, 40.8, 'Rashmi', (1+2j), True]
```

In [167…
```
#Nested list
list6 = [1,5,4,3, [1,2,3],[3,4,5],['Rashmi','Balurkar']]
list6
```

Out[167…
```
[1, 5, 4, 3, [1, 2, 3], [3, 4, 5], ['Rashmi', 'Balurkar']]
```

In [168…
```
len(list6)
```

Out[168…
```
7
```

In [ ]:
```
#Indexing
```

In [169…
```
#Retrieve 1st element
list6[0]
```

Out[169…
```
1
```

In [170…
```
list5[2]
```

Out[170…
```
'Rashmi'
```

In [171…
```
#Nested indexing
list5[2][0]
list5[2][1]
list5[2][2]
list5[2][3]
list5[2][4]
list5[2][5]
```

Out[171…
```
'i'
```

In [172…
```
#Nested indexing
print(list5[2][0])
print(list5[2][1])
print(list5[2][2])
print(list5[2][3])
print(list5[2][4])
print(list5[2][5])
```

```
R
a
s
h
m
i
```

In [173…
```
print(list5[2][6])
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[173], line 1
----> 1 print(list5[2][6])

IndexError: string index out of range
```

In [174…
```
#Access last element of list
list5[-1]
```

Out[174…
```
True
```

In [175…
```
#Slicing
mylist = ['one','two','three','four','five','six','seven','eight']
mylist
```

Out[175…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [176…
```
mylist[:]
```

Out[176…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [177…
```
mylist[0:3]  #index 0 to 'n-1' means 2
```

Out[177…
```
['one', 'two', 'three']
```

In [178…
```
mylist[2:5]
```

Out[178…
```
['three', 'four', 'five']
```

In [179…
```
mylist[:3]  #means index 0 to 2
```

Out[179…
```
['one', 'two', 'three']
```

In [180…
```
mylist[:2]  #means 0 to 1
```

Out[180…
```
['one', 'two']
```

In [182…
```
mylist[:-3]    #means index 0 to (-3-1) ie -4
```

Out[182…
```
['one', 'two', 'three', 'four', 'five']
```

In [183…
```
mylist[:-2]           #means index 0 to -2-1 ie -3
```

Out[183…
```
['one', 'two', 'three', 'four', 'five', 'six']
```

In [184…
```
mylist[-3:]     #means from -3 till last
```

Out[184…
```
['six', 'seven', 'eight']
```

In [185…
```
mylist[-2:]        #means from -2 till last
```

Out[185…
```
['seven', 'eight']
```

In [186…
```
mylist[-1]      #return last element
```

Out[186…
```
'eight'
```

In [187…
```
#Add, remove and change items
mylist
```

Out[187…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [188…
```
mylist.append('nine')       #Add item to the end of the list
mylist
```

Out[188…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [189…
```
mylist.insert(9,'ten')     #Add iem at index 9
mylist
```

Out[189…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

In [191…
```
mylist.insert(1,'ONE')        #Add item at index 1
mylist
```

Out[191…
```
['one',
 'ONE',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten']
```

In [192…
```
mylist.remove('ONE')     #Remove item 'ONE'
mylist
```

Out[192…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

In [193…
```
mylist.pop()        #Removes last item
```

Out[193…
```
'ten'
```

In [194…
```
mylist
```

Out[194…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [195…
```
mylist.pop(8)      #Remove item at index 8
```

Out[195…
```
'nine'
```

In [196…
```
mylist
```

Out[196…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [197…
```
del mylist[7]     #Remove item at index 7
mylist
```

Out[197… ['one', 'two', 'three', 'four', 'five', 'six', 'seven']

In [198…
```
#Change value of the list ie List is mutable
mylist[0] = 1
mylist[1] = 2
mylist[2] = 3
mylist
```

Out[198… [1, 2, 3, 'four', 'five', 'six', 'seven']

In [199…
```
mylist.clear()     #Clear all elements of the list
mylist
```

Out[199… []

In [201…
```
del mylist  #delete the list
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[201], line 1
----> 1 del mylist  #delete the list
      2 mylist

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

In [202…
```
#Copy list
mylist = ['one','two','three','four','five','six','seven','eight','nine']
mylist
```

Out[202… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']

In [203…
```
mylist1 = mylist     #Create a new reference mylist1
mylist1
```

Out[203… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']

In [204…
```
#Here address of mylist and mylist1 is same
print(id(mylist))
print(id(mylist1))
```

```
2580483075392
2580483075392
```

In [205…
```
mylist2 = mylist.copy()      #Create a copy. Here address will be different
mylist2
```

Out[205… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']

In [206…
```
print(id(mylist))
print(id(mylist2))
```

```
2580483075392
2580481787072
```

In [207…
```
mylist[0] = 1    #change the element
mylist
```

Out[207…
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [208…
```
mylist1     #mylist1 will also be changes as it is pointing to the same address
```

Out[208…
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [209…
```
mylist2     #copied list won't have any impact
```

Out[209…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [210…
```
#Join list
list1 = ['one', 'two', 'three', 'four']
list2 = ['five', 'six', 'seven', 'eight']
list3 = list1 + list2     #using + operator
list3
```

Out[210…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [211…
```
list1.append(list2)      #Append list1 with list2 elements
list1
```

Out[211…
```
['one', 'two', 'three', 'four', ['five', 'six', 'seven', 'eight']]
```

In [212…
```
list1.pop()
list1
```

Out[212…
```
['one', 'two', 'three', 'four']
```

In [213…
```
list1.extend(list2)      #Add the list2 elements in list1, same like appending. Her
list1
```

Out[213…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [214…
```
#List membership
'one' in list1    #check 'one' exists in list1
```

Out[214…
```
True
```

In [215…
```
'ten' in list1
```

Out[215…
```
False
```

In [216…
```
'ten' not in list1
```

Out[216…
```
True
```

In [217…
```
#Check if 'three' exists in the list
if 'three' in list1:
    print('Three is present in list1 : ',list1)
```

```
else:
    print('Three is not present in list1 : ',list1)
```

```
Three is present in list1 :  ['one', 'two', 'three', 'four', 'five', 'six', 'seven',
'eight']
```

In [218…
```
#Check if 'eleven' exists in the list
if 'eleven' in list1:
    print('Eleven is present in list1 : ',list1)
else:
    print('Eleven is not present in list1 : ',list1)
```

```
Eleven is not present in list1 :  ['one', 'two', 'three', 'four', 'five', 'six', 'se
ven', 'eight']
```

In [226…
```
#Reverse and sort list
list1 = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
list1
```

Out[226…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [227…
```
list1.reverse()     #Reverse the list
list1
```

Out[227…
```
['eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one']
```

In [228…
```
list1 = list1[::-1]    #Reverse the list
list1
```

Out[228…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [229…
```
mylist3 = [9,5,2,99,12,88,34]
mylist3
```

Out[229…
```
[9, 5, 2, 99, 12, 88, 34]
```

In [231…
```
mylist3.sort()    #Sort in ascending order
```

In [234…
```
mylist3 = [9,5,2,99,12,88,34]
mylist3
```

Out[234…
```
[9, 5, 2, 99, 12, 88, 34]
```

In [235…
```
mylist3.sort(reverse = False)      #Sort in ascending order
mylist3
```

Out[235…
```
[2, 5, 9, 12, 34, 88, 99]
```

In [236…
```
mylist3 = [9,5,2,99,12,88,34]
mylist3
```

Out[236…
```
[9, 5, 2, 99, 12, 88, 34]
```

In [237…
```
mylist3.sort(reverse = True)      #Sort in descending order
mylist3
```

Out[237…
```
[99, 88, 34, 12, 9, 5, 2]
```

In [238…
```
mylist4 = [9,5,2,99,12,88,34]
sorted(mylist4)    #Returns sorted list but doesn't change the original list
```

Out[238…
```
[2, 5, 9, 12, 34, 88, 99]
```

In [239…
```
mylist4
```

Out[239…
```
[9, 5, 2, 99, 12, 88, 34]
```

In [240…
```
#Loop through list
mylist
```

Out[240…
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [241…
```
for i in mylist:
    print(i)
```

```
1
two
three
four
five
six
seven
eight
nine
```

In [242…
```
for i in enumerate mylist:
    print(i)
```

```
  Cell In[242], line 1
    for i in enumerate mylist:
                       ^
SyntaxError: invalid syntax
```

In [243…
```
#enumerate is a function
for i in enumerate(mylist):
    print(i)
```

```
(0, 1)
(1, 'two')
(2, 'three')
(3, 'four')
(4, 'five')
(5, 'six')
(6, 'seven')
(7, 'eight')
(8, 'nine')
```

In [244…
```
#Count
mylist = ['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
mylist
```

Out[244…
```
['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
```

In [245…
```
mylist.count('one')     #count the occurrence or frequency of 'one' in the list
```

Out[245…
```
3
```

In [246…
```
mylist.count('two')     #count the occurrence or frequency of 'two' in the list
```

Out[246…
```
2
```

In [247…
```
mylist.count('three')     #count the occurrence or frequency of 'three' in the list
```

Out[247…
```
2
```

In [248…
```
mylist.count('four')     #count the occurrence or frequency of 'four' in the list
```

Out[248…
```
1
```

In [249…
```
#All/Any
l1 = [1,2,3,4,0]
l1
```

Out[249…
```
[1, 2, 3, 4, 0]
```

In [250…
```
all(l1)      #0 is present so it is False as 0 means False
```

Out[250…
```
False
```

In [251…
```
any(l1)     #always returns True as there are items in the list
```

Out[251…
```
True
```

In [252…
```
l2 = [1,2,3,4]
l2
```

Out[252…
```
[1, 2, 3, 4]
```

In [253…
```
all(l2)      #No 0 here, so True
```

Out[253…
```
True
```

In [254…
```
any(l2)
```

Out[254…
```
True
```

In [255…
```
l3 = [1,2,3,4,True,False]
l3
```

Out[255… [1, 2, 3, 4, True, False]

In [256…
```
all(l3)     #returns False has False value
```

Out[256… False

In [257…
```
any(l3)
```

Out[257… True

In [258…
```
l4 = [1,2,3,4,True]
l4
```

Out[258… [1, 2, 3, 4, True]

In [259…
```
all(l4)     #returns True as no False element
```

Out[259… True

In [260…
```
any(l4)
```

Out[260… True

In [261…
```
l5 = []
l5
```

Out[261… []

In [263…
```
all(l5)     #returns True as no False element
```

Out[263… True

In [265…
```
any(l5)    #only this case will return False as there are no elements
```

Out[265… False

In [ ]: