In [3]:
```
l=[] #list always given with []
l
```

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

In [5]:
```
l.append(10) #append is to add values at the end of the list
l
```

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

In [7]:
```
l.append(20) #we can add any data type to the list
l.append(20.5)
l.append(True)
l.append('akhil')
l
```

Out[7]:
```
[10, 20, 20.5, True, 'akhil']
```

In [9]:
```
l.remove(10) #we can remove the values using remove
l
```

Out[9]:
```
[20, 20.5, True, 'akhil']
```

In [15]:
```
l1=l.copy() #we can copy the whole list
l1
```

Out[15]:
```
[20, 20.5, True, 'akhil']
```

In [17]:
```
l.append(10)
l
```

Out[17]:
```
[20, 20.5, True, 'akhil', 10]
```

In [13]:
```
l.count(10) #count key counts number of values
```

Out[13]:
```
1
```

In [19]:
```
l==l1 #since 10 is added later in the list
```

Out[19]:
```
False
```

In [21]:
```
len(l)
```

Out[21]:
```
5
```

In [23]:
```
len(l1)
```

Out[23]:
```
4
```

In [25]:
```
l.append(20)
l
```

Out[25]:
```
[20, 20.5, True, 'akhil', 10, 20]
```

```
In [27]: l.remove(20) #if there is an duplicate value the available value will be removed
         l

Out[27]: [20.5, True, 'akhil', 10, 20]

In [29]: l

Out[29]: [20.5, True, 'akhil', 10, 20]

In [31]: l[:]

Out[31]: [20.5, True, 'akhil', 10, 20]

In [33]: l[:4]

Out[33]: [20.5, True, 'akhil', 10]

In [35]: l[4]

Out[35]: 20

In [37]: l

Out[37]: [20.5, True, 'akhil', 10, 20]

In [35]: l1

Out[35]: [20, 20.5, True, 'akhil']

In [39]: l2=[]
         l2

Out[39]: []

In [41]: l2.append(1)
         l2.append(2.3)
         l2.append(True)
         l2.append(1+2j)
         l2.append('nit')
         l2

Out[41]: [1, 2.3, True, (1+2j), 'nit']

In [43]: l3=l2.copy()
         l3

Out[43]: [1, 2.3, True, (1+2j), 'nit']

In [45]: len(l3)

Out[45]: 5

 In [9]: l3.clear() #this function clear the elements from list

In [47]: len(l3)
```

```
Out[47]:  5

In [49]:  l2.remove(2.3)
          l2

Out[49]:  [1, True, (1+2j), 'nit']

In [51]:  del (l3)

In [53]:  l3=[]
          l3

Out[53]:  []

In [55]:  l3.append(10)
          l3

Out[55]:  [10]

In [57]:  l3.extend(l2)
          l3

Out[57]:  [10, 1, True, (1+2j), 'nit']

In [59]:  l3.index(1)

Out[59]:  1

In [61]:  l3.index(1+2j)

Out[61]:  3

In [63]:  l3.index('nit')

Out[63]:  4

In [65]:  l3

Out[65]:  [10, 1, True, (1+2j), 'nit']

In [67]:  l3.append(1)
          l3

Out[67]:  [10, 1, True, (1+2j), 'nit', 1]

In [69]:  l3.insert(5,'technology')

In [35]:  l3

Out[35]:  [10, 1, True, (1+2j), 'nit', 'technology', 1]

In [71]:  l3.insert(3,False)

In [39]:  l3
```

```
Out[39]: [10, 1, True, False, (1+2j), 'nit', 'technology', 1]

In [73]: l3.pop()

Out[73]: 1

In [43]: l3

Out[43]: [10, 1, True, False, (1+2j), 'nit', 'technology']

In [75]: l3.pop(True)

Out[75]: 1

In [77]: l3

Out[77]: [10, True, False, (1+2j), 'nit', 'technology']

In [79]: l3.pop(2)

Out[79]: False

In [81]: l3

Out[81]: [10, True, (1+2j), 'nit', 'technology']

In [83]: l4=[10,100,3,45,76,24]
         l4

Out[83]: [10, 100, 3, 45, 76, 24]

In [85]: l4.sort()
         l4

Out[85]: [3, 10, 24, 45, 76, 100]

In [87]: l4.sort(reverse=False)
         l4

Out[87]: [3, 10, 24, 45, 76, 100]

In [89]: l4.sort(reverse=True)
         l4

Out[89]: [100, 76, 45, 24, 10, 3]

In [91]: l5=['z','s','d']
         l5

Out[91]: ['z', 's', 'd']

In [93]: l6=[1,2,3,4,'a','s','d']
         l6

Out[93]: [1, 2, 3, 4, 'a', 's', 'd']
```

```
In [95]: l6.sort() #as sort will not work when there are more than one data type
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[95], line 1
----> 1 l6.sort()

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

```
In [97]: l3
```

```
Out[97]: [10, True, (1+2j), 'nit', 'technology']
```

```
In [99]: l3.reverse()
         l3
```

```
Out[99]: ['technology', 'nit', (1+2j), True, 10]
```

```
In [101… l2[3]
```

```
Out[101… 'nit'
```

```
In [103… print(l2[3][0]) #nested list
         print(l2[3][1])
         print(l2[3][2])
```

```
n
i
t
```

```
In [105… l3
```

```
Out[105… ['technology', 'nit', (1+2j), True, 10]
```

```
In [107… l3.insert(3,2)
```

```
In [109… l3
```

```
Out[109… ['technology', 'nit', (1+2j), 2, True, 10]
```

```
In [111… l3.pop(4)
```

```
Out[111… True
```

```
In [117… l3.reverse()
         l3
```

```
Out[117… ['technology', 'nit', (1+2j), 2, 10]
```

```
In [119… l3.index(True)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
Cell In[119], line 1
----> 1 l3.index(True)

ValueError: True is not in list
```

In [121…

```
l3.pop(2)
l3
```

Out[121…

```
['technology', 'nit', 2, 10]
```

In [123…

```
l7=[1,2,3,4,5,6,7]
l7
```

Out[123…

```
[1, 2, 3, 4, 5, 6, 7]
```

In [125…

```
l7.reverse()
l7
```

Out[125…

```
[7, 6, 5, 4, 3, 2, 1]
```

In [127…

```
l3
```

Out[127…

```
['technology', 'nit', 2, 10]
```

In [129…

```
l6=['sbi','icici']
l7=['hdfc','kotak']
l6+l7
```

Out[129…

```
['sbi', 'icici', 'hdfc', 'kotak']
```

In [131…

```
l8=[1,2,3,4,5,6]
```

In [133…

```
for i in enumerate (l3): #this key gives the values and index
    print(i)
```

```
(0, 'technology')
(1, 'nit')
(2, 2)
(3, 10)
```

In [135…

```
for i in (l3): #this key changes the values positions from row into colums
    print(i)
```

```
technology
nit
2
10
```

In [137…

```
l
```

Out[137…

```
[20.5, True, 'akhil', 10, 20]
```

In [139…

```
l[:]
```

Out[139…

```
[20.5, True, 'akhil', 10, 20]
```

In [141…

```
l[::-1] #this gives the inverted
```

Out[141…

```
[20, 10, 'akhil', True, 20.5]
```

In [143…

```
l[::-2]#this gives gives the gap
```

Out[143…
```
[20, 'akhil', 20.5]
```

In [145…
```
l[2:]
```

Out[145…
```
['akhil', 10, 20]
```

In [147…
```
l[:5]
```

Out[147…
```
[20.5, True, 'akhil', 10, 20]
```

In [149…
```
l[0:5:2]
```

Out[149…
```
[20.5, 'akhil', 20]
```

In [151…
```
l
```

Out[151…
```
[20.5, True, 'akhil', 10, 20]
```

In [187…
```
l[0]=20 #mutation ,changing the values
l
```

Out[187…
```
[20, True, 'akhil', 10, 20]
```

2 tuple

In [153…
```
t=() #tuple is assign with ()
t
```

Out[153…
```
()
```

In [155…
```
type(t)
```

Out[155…
```
tuple
```

In [157…
```
type(l)
```

Out[157…
```
list
```

In [159…
```
t1=tuple()
type(t1)
```

Out[159…
```
tuple
```

In [161…
```
t=(10,10,20,30)
t
```

Out[161…
```
(10, 10, 20, 30)
```

In [163…
```
icici=(1234,'abe',4)
icici
```

Out[163…
```
(1234, 'abe', 4)
```

In [165…
```
i=icici.copy # copy is not possible in tuple
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Cell In[165], line 1
----> 1 i=icici.copy

AttributeError: 'tuple' object has no attribute 'copy'
```

In [167…

```
t[0]=1 # tuple is notb mutable
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[167], line 1
----> 1 t[0]=1

TypeError: 'tuple' object does not support item assignment
```

In [169…

```
t1=(10,1.2,'nit',1+2j) #
t1
```

Out[169…

```
(10, 1.2, 'nit', (1+2j))
```

In [171…

```
t1.count(10) # count
```

Out[171…

```
1
```

In [173…

```
t1.index('nit') #index
```

Out[173…

```
2
```

In [175…

```
for i in t1: #for loop
    print(i)
```

```
10
1.2
nit
(1+2j)
```

In [177…

```
for i in enumerate(t1): #enumarate
    print(i)
```

```
(0, 10)
(1, 1.2)
(2, 'nit')
(3, (1+2j))
```

In [179…

```
t[:3]
```

Out[179…

```
(10, 10, 20)
```

In [181…

```
t
```

Out[181…

```
(10, 10, 20, 30)
```

In [ ]:

In [ ]:

```
In [ ]:
```

```
In [ ]:
```