

```
In [1]: l = [10]
l
```

```
Out[1]: [10]
```

```
In [3]: l.append(20)
l.append(30)
l.append(40)
l.append(50)
```

```
In [5]: l
```

```
Out[5]: [10, 20, 30, 40, 50]
```

```
In [7]: l.append(60,70,80) #it will not accept multiple values
```

```
-----
```

```
TypeError
Cell In[7], line 1
----> 1 l.append(60,70,80)
```

```
Traceback (most recent call last)
```

```
TypeError: list.append() takes exactly one argument (3 given)
```

```
In [9]: type(l)
```

```
Out[9]: list
```

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

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

```
In [15]: l.append(10)
l.append('iit')
l.append(10.3)
l.append(2+8j)
```

```
In [17]: l
```

```
Out[17]: [10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j)]
```

```
In [19]: len(l)
```

```
Out[19]: 9
```

```
In [23]: l1 = l.copy()
```

```
In [25]: l1
```

```
Out[25]: [10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j)]
```

```
In [27]: len(l1)
```

```
Out[27]: 9
```

```
In [29]: l == 11
```

```
Out[29]: True
```

```
In [31]: l.append(80)  
l
```

```
Out[31]: [10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j), 80]
```

```
In [33]: l == 11
```

```
Out[33]: False
```

```
In [37]: l.index(80)
```

```
Out[37]: 9
```

```
In [39]: l2 = l.copy()
```

```
In [41]: l2
```

```
Out[41]: [10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j), 80]
```

```
In [43]: l == l2
```

```
Out[43]: True
```

```
In [45]: l1 == l2
```

```
Out[45]: False
```

```
In [47]: l
```

```
Out[47]: [10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j), 80]
```

```
In [49]: len(l2)
```

```
Out[49]: 10
```

```
In [51]: print(l)  
print(l1)  
print(l2)
```

```
[10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j), 80]  
[10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j)]  
[10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j), 80]
```

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

```
Out[53]: [10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j), 80]
```

```
In [55]: l[:3]
```

```
Out[55]: [10, 20, 30]
```

```
In [57]: l[4:]
```

```
Out[57]: [50, 10, 'iit', 10.3, (2+8j), 80]
```

```
In [59]: l[:5]
```

```
Out[59]: [10, 20, 30, 40, 50]
```

```
In [61]: l[7:]
```

```
Out[61]: [10.3, (2+8j), 80]
```

```
In [63]: l[:-1]
```

```
Out[63]: [10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j)]
```

```
In [65]: l[:-3]
```

```
Out[65]: [10, 20, 30, 40, 50, 10, 'iit']
```

```
In [67]: l2.clear()      #l2 has cleared
```

```
In [69]: l2
```

```
Out[69]: []
```

```
In [71]: del(l2)      #in this l2 deleted the variables
```

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

NameError

Cell In[73], line 1

----> 1 l2

Traceback (most recent call last)

NameError: name 'l2' is not defined

```
In [75]: l2
```

NameError

Cell In[75], line 1

----> 1 l2

Traceback (most recent call last)

NameError: name 'l2' is not defined

```
In [77]: l
```

```
Out[77]: [10, 20, 30, 40, 50, 10, 'iit', 10.3, (2+8j), 80]
```

```
In [79]: l.index(10.3)
```

```
Out[79]: 7
```

```
In [81]: l.index('iit')
```

```
Out[81]: 6
```

```
In [85]: nums = [10,20,30]
```

```
In [87]: nums
```

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

```
In [91]: nums[-1]
```

```
Out[91]: 30
```

```
In [95]: nums[:2]
```

```
Out[95]: [10, 20]
```

```
In [97]: num1 = 'hi' , 'bye'
```

```
In [99]: num1
```

```
Out[99]: ('hi', 'bye')
```

```
In [101...]: num2 = 'uday waste fellow'  
num2
```

```
Out[101...]: 'uday waste fellow'
```

```
In [105...]: nums = num1 , num2  
nums
```

```
Out[105...]: (('hi', 'bye'), 'uday waste fellow')
```

```
In [111...]: nums
```

```
Out[111...]: (('hi', 'bye'), 'uday waste fellow')
```

```
In [123...]: nums = [10,20,30,40]  
nums
```

```
Out[123...]: [10, 20, 30, 40]
```

```
In [121...]: l = [10]  
l
```

```
Out[121... [10]
```

```
In [125... nums.append(50)
```

```
In [127... nums
```

```
Out[127... [10, 20, 30, 40, 50]
```

```
In [129... nums.remove(50)  
nums
```

```
Out[129... [10, 20, 30, 40]
```

```
In [137... nums
```

```
Out[137... [10, 20]
```

```
In [141... nums = [50,60,70,80]  
nums
```

```
Out[141... [50, 60, 70, 80]
```

```
In [143... nums.pop(3)
```

```
Out[143... 80
```

```
In [145... nums
```

```
Out[145... [50, 60, 70]
```

```
In [147... nums.pop(1)
```

```
Out[147... 60
```

```
In [149... nums
```

```
Out[149... [50, 70]
```

```
In [159... nums.insert(10,20) #10 was an index number and 20 was an object or elemnt will be
```

```
In [155... nums
```

```
Out[155... [50, 70, 20]
```

```
In [161... nums.insert(1,30) #for example
```

```
In [163... nums
```

```
Out[163... [50, 30, 70, 20, 20]
```

```
In [173... nums.extend([20 , 30]) #it will extend List from the Last elemenet
```

```
In [175...]: nums
```

```
Out[175...]: [50, 30, 70, 20, 20, 20, 30]
```

```
In [177...]: nums.extend([30])      #it will extend list from the last elemenet
```

```
In [179...]: nums
```

```
Out[179...]: [50, 30, 70, 20, 20, 20, 30, 30]
```

```
In [181...]: nums.extend(['a',50])
```

```
In [183...]: nums
```

```
Out[183...]: [50, 30, 70, 20, 20, 20, 30, 30, 'a', 50]
```

```
In [187...]: len(nums)
```

```
Out[187...]: 10
```

```
In [191...]: nums.remove('a')
```

```
In [193...]: nums
```

```
Out[193...]: [20, 20, 20, 30, 30, 30, 50, 70, 50]
```

```
In [195...]: nums.sort()      # ascending order sort means
```

```
In [197...]: nums
```

```
Out[197...]: [20, 20, 20, 30, 30, 30, 50, 50, 70]
```

```
In [199...]: nums.reverse()      # it will reverse the elements
```

```
In [201...]: nums
```

```
Out[201...]: [70, 50, 50, 30, 30, 30, 20, 20, 20]
```

```
In [203...]: tup = (10,20,30)
```

```
In [205...]: tup
```

```
Out[205...]: (10, 20, 30)
```

```
In [211...]: tup.count(20)  #it will count the elements how many are in the List 10
```

```
Out[211...]: 1
```

```
In [213...]: tup = (20,30,40,20,30,40)
```

```
In [215]: tup
```

```
Out[215]: (20, 30, 40, 20, 30, 40)
```

```
In [217]: tup.count(20)
```

```
Out[217]: 2
```

```
In [227]: tup.index(30)      # index means it will count the which the number is present Like
```

```
Out[227]: 1
```

set

```
In [3]: s = {10,20,30,40,50}  
s
```

```
Out[3]: {10, 20, 30, 40, 50}
```

```
In [5]: s1 = {'nit' , 'iit' , 'akshith'}
```

```
In [7]: s1
```

```
Out[7]: {'akshith', 'iit', 'nit'}
```

```
In [9]: s == s1
```

```
Out[9]: False
```

```
In [11]: s1 = s.copy()
```

```
In [13]: s1
```

```
Out[13]: {10, 20, 30, 40, 50}
```

```
In [15]: s1 == s
```

```
Out[15]: True
```

```
In [17]: s[1] #index is not supported in sets
```

TypeError
Cell In[17], line 1
----> 1 s[1]

Traceback (most recent call last)

TypeError: 'set' object is not subscriptable

```
In [19]: s
```

```
Out[19]: {10, 20, 30, 40, 50}
```

mutable - we can modify any value list = mutable

22 nd april

```
In [5]: l.append(10)
l.append('iit')
l.append(10.3)
l.append(2+8j)
l
```

```
NameError
Cell In[5], line 1
----> 1 l.append(10)
      2 l.append('iit')
      3 l.append(10.3)

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

```
In [1]: l = []
l
```

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

```
In [3]: l.append(10)          #adding the last element in the list
l.append('akshith')
l.append(6.5)
l.append(2+7j)
l
```

```
Out[3]: [10, 'akshith', 6.5, (2+7j)]
```

```
In [5]: l
```

```
Out[5]: [10, 'akshith', 6.5, (2+7j)]
```

```
In [7]: l1 = l.copy()
```

```
In [9]: l1
```

```
Out[9]: [10, 'akshith', 6.5, (2+7j)]
```

```
In [11]: l == l1
```

```
Out[11]: True
```

```
In [15]: l1 = [20,30,40,50]
```

11

Out[15]: [20, 30, 40, 50]

In [17]: 11

Out[17]: [20, 30, 40, 50]

In [19]: 1 == 11

Out[19]: False

In [21]: 12 = [10,20,30,40]
12

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

In [67]: 12.clear()

```
NameError  
Cell In[67], line 1  
----> 1 12.clear()
```

Traceback (most recent call last)

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

In [43]: 12

In [47]: del(12) #l2 will be deleted

```
NameError  
Cell In[47], line 1  
----> 1 del(12)
```

Traceback (most recent call last)

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

In [49]: 11

Out[49]: [20, 30, 40, 50]

In [53]: print(l1)
print(l)

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

In [55]: l = [90,80,70,60,50,40]

In [57]: 1

Out[57]: [90, 80, 70, 60, 50, 40]

```
In [59]: print(l)
      print(l1)

[90, 80, 70, 60, 50, 40]
[20, 30, 40, 50]
```

```
In [63]: l = l.index(70)          #index finds the position of the number
```

```
In [65]: l
```

```
Out[65]: 2
```

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

```
Out[69]: [20, 30, 40, 50]
```

```
In [71]: l1.index(50)
```

```
Out[71]: 3
```

```
In [73]: l1
```

```
Out[73]: [20, 30, 40, 50]
```

```
In [75]: l2 = [10,20,30,40,10,10,20,20,20,30,40,40,40]
```

```
In [77]: l2
```

```
Out[77]: [10, 20, 30, 40, 10, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [79]: l2.count(10)      #it will count the numbers
```

```
Out[79]: 3
```

```
In [81]: l2.count(20)
```

```
Out[81]: 4
```

```
In [83]: print(l)
      print(l1)
      print(l2)
```

```
2
[20, 30, 40, 50]
[10, 20, 30, 40, 10, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [89]: l1.extend(l2)
```

```
In [91]: l1
```

```
Out[91]: [20, 30, 40, 50, 10, 20, 30, 40, 10, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [93]: l1
```

```
Out[93]: [20, 30, 40, 50, 10, 20, 30, 40, 10, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [95]: l2
```

```
Out[95]: [10, 20, 30, 40, 10, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [97]: l2.extend(l1)
```

```
l2
```

```
In [101...]: l2.extend(l1)
```

```
-----  
TypeError  
Cell In[101], line 1  
----> 1 l2.extend(l1)
```

```
Traceback (most recent call last)
```

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

```
In [103...]: l1 #l has only one int thats why its error
```

```
Out[103...]: 2
```

```
In [105...]: print(l1)  
print(l2)
```

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

```
In [107...]: l1.insert(2,35) # 2 is the index number and 35 is the value what we have
```

```
In [109...]: l1
```

```
Out[109...]: [20, 30, 35, 40, 50, 10, 20, 30, 40, 10, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [111...]: l2
```

```
Out[111... [10,  
20,  
30,  
40,  
10,  
10,  
20,  
20,  
20,  
30,  
40,  
40,  
40,  
20,  
30,  
40,  
50,  
10,  
20,  
30,  
40,  
10,  
10,  
20,  
20,  
20,  
30,  
40,  
40,  
40]
```

```
In [113... print(l2)
```

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

```
In [115... l2.insert(5 , 15)
```

```
In [117... 12
```

```
Out[117... [10,  
20,  
30,  
40,  
10,  
15,  
10,  
20,  
20,  
20,  
30,  
40,  
40,  
40,  
20,  
30,  
40,  
10,  
20,  
30,  
40,  
10,  
10,  
20,  
20,  
20,  
30,  
40,  
40,  
40,  
40]
```

```
In [119... print(l2)
```

```
[10, 20, 30, 40, 10, 15, 10, 20, 20, 20, 30, 40, 40, 40, 20, 30, 40, 50, 10, 20, 30,  
40, 10, 10, 20, 20, 20, 30, 40, 40]
```

```
In [121... print(l1)  
print(l2)
```

```
[20, 30, 35, 40, 50, 10, 20, 30, 40, 10, 10, 20, 20, 20, 30, 40, 40, 40]  
[10, 20, 30, 40, 10, 15, 10, 20, 20, 20, 30, 40, 40, 40, 20, 30, 40, 50, 10, 20, 30,  
40, 10, 10, 20, 20, 20, 30, 40, 40]
```

```
In [125... l1.pop(10)
```

```
Out[125... 10
```

```
In [129... 12.pop(0)  
12
```

```
Out[129... [30,  
40,  
10,  
15,  
10,  
20,  
20,  
20,  
30,  
40,  
40,  
40,  
20,  
30,  
40,  
50,  
10,  
20,  
30,  
40,  
10,  
10,  
20,  
20,  
20,  
30,  
40,  
40,  
40]
```

```
In [131... print(l1)  
print(l2)
```

```
[20, 30, 35, 40, 50, 10, 20, 30, 40, 10, 20, 20, 20, 20, 30, 40, 40, 40]  
[30, 40, 10, 15, 10, 20, 20, 20, 30, 40, 40, 40, 20, 30, 40, 50, 10, 20, 30, 40, 10,  
10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [133... l1.pop(1) # if we give the index number it will say the number and next it will be
```

```
Out[133... 30
```

```
In [135... 11
```

```
Out[135... [20, 35, 40, 50, 10, 20, 30, 40, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [137... 12.pop(0)
```

```
Out[137... 30
```

```
In [139... 12
```

```
Out[139... [40,  
10,  
15,  
10,  
20,  
20,  
20,  
20,  
30,  
40,  
40,  
40,  
20,  
30,  
40,  
50,  
10,  
20,  
30,  
40,  
10,  
10,  
20,  
20,  
20,  
20,  
30,  
40,  
40,  
40]
```

```
In [141... print(l2)
```

```
[40, 10, 15, 10, 20, 20, 20, 30, 40, 40, 40, 20, 30, 40, 50, 10, 20, 30, 40, 10, 10,  
20, 20, 20, 30, 40, 40, 40]
```

```
In [143... 11
```

```
Out[143... [20, 35, 40, 50, 10, 20, 30, 40, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [145... 11.remove(20)
```

```
In [147... 11
```

```
Out[147... [35, 40, 50, 10, 20, 30, 40, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [149... 11.remove(40)
```

```
In [151... 11
```

```
Out[151... [35, 50, 10, 20, 30, 40, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [155... 11.remove(40) #directly it will remove the number
```

```
In [157... 11
```

```
Out[157... [35, 50, 10, 20, 30, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [159... 11
```

```
Out[159... [35, 50, 10, 20, 30, 10, 20, 20, 20, 30, 40, 40, 40]
```

```
In [161... 11.reverse()      #directly it will reverse the numbers
```

```
In [163... 11
```

```
Out[163... [40, 40, 40, 30, 20, 20, 20, 10, 30, 20, 10, 50, 35]
```

```
In [167... 11
```

```
Out[167... [40, 40, 40, 30, 20, 20, 20, 10, 30, 20, 10, 50, 35]
```

```
In [169... 11.sort()      #it will give in ascending order
```

```
In [171... 11
```

```
Out[171... [10, 10, 20, 20, 20, 20, 30, 30, 35, 40, 40, 40, 50]
```

```
In [173... 11.sort(reverse = False)          #here false means ascending order
```

```
In [175... 11
```

```
Out[175... [10, 10, 20, 20, 20, 20, 30, 30, 35, 40, 40, 40, 50]
```

```
In [177... 11.sort(reverse = True )        #true = descending order
```

```
In [179... 11
```

```
Out[179... [50, 40, 40, 40, 35, 30, 30, 20, 20, 20, 20, 10, 10]
```

in these datatypes are called as user can assign only one value like int,string,bool,float,complex

datastructuers = collection of datatypes like user can declare more thamn one values

matrix = collection of rows and columns like tabular form

datastructures are divided into 2 parts are inbuid and user defined

inbuild fuctions are list , tuple , set , dict

userdefined = linked list , tree , stack , numpy.

in theses inbuild fuction list is mutable

**list functions are
append(),copy(),clear(),index(),extend(),remove()**

append() = add the element in the list at the last

copy() = copy the elements from suppose l1 to l2

clear()= it will clear the elements in the list

index() = it will give the index number of which we give the element

extend() = it will extend from list to list like the list will always increase

remove() = if we give the element in the list then it will remove the element

reverse() = in this the list will be reversed

del() = it will totally delete the variables

sort() = the list will be sorted in ascending order and if we give `l.sort(reverse = True)` it will be in descending order

False means ascending order

True means descending order

pop() = if we give the index number it will remove the element and return the entire list with removing the number what we have given

insert() = in this insert there is index and value number like we have to insert 25 value in which index that defines these insert like it has index and value

count() = in these count in the given list if we given the count 10 the count shows how many times it repeated and give the value as output.

these are we completed till now .

```
Out[181... 2
```

```
In [183... 11
```

```
Out[183... [50, 40, 40, 40, 35, 30, 30, 20, 20, 20, 20, 10, 10]
```

```
In [187... print(l2)
```

```
[40, 10, 15, 10, 20, 20, 20, 30, 40, 40, 40, 20, 30, 40, 50, 10, 20, 30, 40, 10, 10,  
20, 20, 20, 30, 40, 40]
```

```
In [193... print(l1)  
print(l2)
```

```
[50, 40, 40, 40, 35, 30, 30, 20, 20, 20, 20, 10, 10]  
[40, 10, 15, 10, 20, 20, 20, 30, 40, 40, 40, 20, 30, 40, 50, 10, 20, 30, 40, 10, 10,  
20, 20, 20, 30, 40, 40]
```

```
In [197... len(l1)  
len(l2)
```

```
Out[197... 28
```

```
In [199... len(l1)
```

```
Out[199... 13
```

```
In [203... 11
```

```
Out[203... [50, 40, 40, 40, 35, 30, 30, 20, 20, 20, 20, 10, 10]
```

```
In [205... l1[0:10:2]
```

```
Out[205... [50, 40, 35, 30, 20]
```

```
In [209... print (l2)
```

```
[40, 10, 15, 10, 20, 20, 20, 30, 40, 40, 40, 20, 30, 40, 50, 10, 20, 30, 40, 10, 10,  
20, 20, 20, 30, 40, 40]
```

```
In [211... l = [10,20,30,0,50,60,70,80,90]
```

```
In [213... l
```

```
Out[213... [10, 20, 30, 0, 50, 60, 70, 80, 90]
```

```
In [215... l[0:9:3]
```

```
Out[215... [10, 0, 70]
```

```
In [217... l[0:6:2]
```

```
Out[217... [10, 30, 50]
```

```
In [219... 1[2:9:3]      #step indexing
```

```
Out[219... [30, 60, 90]
```

```
In [221... 1[:]
```

```
Out[221... [10, 20, 30, 0, 50, 60, 70, 80, 90]
```

```
In [223... 1[:2]
```

```
Out[223... [10, 20]
```

```
In [225... 1[:5]
```

```
Out[225... [10, 20, 30, 0, 50]
```

```
In [227... 1[2:6]
```

```
Out[227... [30, 0, 50, 60]
```

```
In [235... 1[-1]
```

```
Out[235... 90
```

```
In [237... 1[:-1]
```

```
Out[237... [10, 20, 30, 0, 50, 60, 70, 80]
```

```
In [239... 1[3:-1]
```

```
Out[239... [0, 50, 60, 70, 80]
```

```
In [243... 1[3:]
```

```
Out[243... [0, 50, 60, 70, 80, 90]
```

```
In [245... 1
```

```
Out[245... [10, 20, 30, 0, 50, 60, 70, 80, 90]
```

```
In [247... type(1)
```

```
Out[247... list
```

```
In [249... len(1)
```

```
Out[249... 9
```

```
In [1]: list = []
list
```

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

```
In [3]: type(list)
```

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

```
In [15]: list1 = [10,20,30,40,50]      #list of integers
```

```
In [17]: list1
```

```
Out[17]: [10, 20, 30, 40, 50]
```

```
In [19]: list2 = [10.3,2.4,3.6,7.8]      #list of float
```

```
In [21]: list2
```

```
Out[21]: [10.3, 2.4, 3.6, 7.8]
```

```
In [23]: list3 = ['one' , 'two' , 'three']      #List of string
```

```
In [25]: list3
```

```
Out[25]: ['one', 'two', 'three']
```

```
In [27]: list4 = ['akshith' , 25 , [50,100] , [150,30] ]    # nested Loops = List inside list
```

```
In [29]: list4
```

```
Out[29]: ['akshith', 25, [50, 100], [150, 30]]
```

```
In [31]: list5 = ['akshith' , 10 , 2.3]      # mixed data types
```

```
In [33]: list5
```

```
Out[33]: ['akshith', 10, 2.3]
```

```
In [45]: len(list4)
```

```
Out[45]: 4
```

```
In [47]: len(list5)
```

```
Out[47]: 3
```

```
In [51]: list2[0]
```

```
Out[51]: 10.3
```

```
In [53]: list4[2]
```

```
Out[53]: [50, 100]
```

```
In [59]: list3[0][0]
```

```
Out[59]: 'o'
```

```
In [61]: list3[1][1]      # in this first letter is index of the word and second letter i
```

```
Out[61]: 'w'
```

```
In [63]: list3[2][2]
```

```
Out[63]: 'r'
```

```
In [65]: list3
```

```
Out[65]: ['one', 'two', 'three']
```

```
In [67]: list3[0][1]
```

```
Out[67]: 'n'
```

```
In [69]: list3[1][2]
```

```
Out[69]: 'o'
```

```
In [71]: list4[-1]      #last item of the list
```

```
Out[71]: [150, 30]
```

```
In [73]: list6 = ['one', 'two', 'three', 'four', 'five', 'six', 'seven']  
list6
```

```
Out[73]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

```
In [75]: list6[0:3]
```

```
Out[75]: ['one', 'two', 'three']
```

```
In [77]: list6[1:6]
```

```
Out[77]: ['two', 'three', 'four', 'five', 'six']
```

```
In [79]: list6[3]
```

```
Out[79]: 'four'
```

```
In [83]: list6[:3]
```

```
Out[83]: ['one', 'two', 'three']
```

```
In [85]: list6[:]    # it will give the whole list
```

```
Out[85]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

```
In [87]: mylist = list6.copy()
```

```
In [89]: mylist
```

```
Out[89]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

```
In [91]: mylist.append('eight')
```

```
In [93]: mylist
```

```
Out[93]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [95]: mylist.remove('three')
```

```
In [97]: mylist
```

```
Out[97]: ['one', 'two', 'four', 'five', 'six', 'seven', 'eight']
```

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

```
In [101...]: mylist
```

```
Out[101...]: ['eight', 'seven', 'six', 'five', 'four', 'two', 'one']
```

```
In [103...]: mylist.reverse()
```

```
In [105...]: mylist
```

```
Out[105...]: ['one', 'two', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [107...]: mylist.insert(2,'three')      # here 2 is index number and three is where we can add
```

```
In [109...]: mylist
```

```
Out[109...]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [111...]: mylist.pop()
```

```
Out[111...]: 'eight'
```

```
In [114...]: mylist
```

```
Out[114...]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

```
In [124...]: mylist.append('one')
```

```
In [126...]: mylist
```

```
Out[126... ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'One', 'one']
```

```
In [128... mylist.count('one')
```

```
Out[128... 2
```

```
In [130... mylist.extend(list2)
```

```
In [132... mylist
```

```
Out[132... ['one',
            'two',
            'three',
            'four',
            'five',
            'six',
            'seven',
            'One',
            'one',
            10.3,
            2.4,
            3.6,
            7.8]
```

```
In [134... print(mylist)
```

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'One', 'one', 10.3, 2.4, 3.6, 7.8]
```

```
In [136... mylist.sort() # it will not support in float and string
```

TypeError
 Cell In[136], line 1
 ----> 1 mylist.sort()

Traceback (most recent call last)

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

```
In [138... list1.sort()
```

```
In [142... list1
```

```
Out[142... [10, 20, 30, 40, 50]
```

```
In [144... mylist
```

```
Out[144... ['One',
    'five',
    'four',
    'one',
    'one',
    'seven',
    'six',
    'three',
    'two',
    10.3,
    2.4,
    3.6,
    7.8]
```

```
In [146... print(mylist)

['One', 'five', 'four', 'one', 'one', 'seven', 'six', 'three', 'two', 10.3, 2.4, 3.6, 7.8]
```

```
In [148... mylist.pop()
```

```
Out[148... 7.8
```

```
In [150... mylist
```

```
Out[150... ['One',
    'five',
    'four',
    'one',
    'one',
    'seven',
    'six',
    'three',
    'two',
    10.3,
    2.4,
    3.6]
```

```
In [152... mylist.pop(2)
```

```
Out[152... 'four'
```

```
In [154... mylist
```

```
Out[154... ['One', 'five', 'one', 'one', 'seven', 'six', 'three', 'two', 10.3, 2.4, 3.6]
```

```
In [156... del(list1)      #List 1 is deleted
```

```
In [158... list1
```

```

-----
NameError                                         Traceback (most recent call last)
Cell In[158], line 1
----> 1 list1

NameError: name 'list1' is not defined

In [170... mylist = ['one','two','three','four','five','six','seven']
mylist

Out[170... ['one', 'two', 'three', 'four', 'five', 'six', 'seven']

In [172... mylist[0] = 1 # changing the values of the list
mylist[1] = 2
mylist[2] = 3
mylist

Out[172... [1, 2, 3, 'four', 'five', 'six', 'seven']

In [174... mylist.clear() #it will all the elements in the list

In [176... mylist

Out[176... []

In [178... del(mylist) #it will deleted all the variables in the list

In [180... mylist

-----
NameError                                         Traceback (most recent call last)
Cell In[180], line 1
----> 1 mylist

NameError: name 'mylist' is not defined

In [182... mylist = ['one','two','three','four','five','six','seven']
mylist1 = mylist

In [184... id(mylist),id(mylist1) #it is the adress of the both List and it would be t

Out[184... (2182032359040, 2182032359040)

In [186... mylist2 = mylist.copy()

In [188... mylist2

Out[188... ['one', 'two', 'three', 'four', 'five', 'six', 'seven']

In [190... mylist2 == mylist

Out[190... True

```

```
In [192... id(mylist2),id(mylist)      # adress would be different
```

```
Out[192... (2182032355968, 2182032359040)
```

```
In [194... mylist
```

```
Out[194... ['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

```
In [196... mylist[1] = 2 #it will change the value
```

```
In [198... mylist
```

```
Out[198... ['one', 2, 'three', 'four', 'five', 'six', 'seven']
```

```
In [204... list1 = ['one','two','three','four','five','six','seven'] # Join Lists
list2 = ['eight','nine','ten']
list3 = list1 + list2
print(list3)
```

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

```
In [206... list1 = ['one','two','three','four','five','six','seven'] # Join Lists
list2 = ['eight','nine','ten']
```

```
In [208... list1.extend(list2)
```

```
In [210... list1
```

```
Out[210... ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [214... 'one' in list1 #List membership checking if one is presenting in the List or not
```

```
Out[214... True
```

```
In [216... 'fouteen' in list1
```

```
Out[216... False
```

```
In [218... list1
```

```
Out[218... ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [233... if 'two' in list1:      # IT WILL PRINT THE STATEMENT LIKE IT IS PRESENTING OR NOT
    print("two is present in the list1")
else:
    print("two is not present in the list1")
```

```
two is present in the list1
```

```
In [235... if 'fourteen' in list1:
    print('fourteen is present in list1')
else:
    print('fourteen is not present in list1')
```

fourteen is not present in list1

In [237... list1

Out[237... ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']

In [243... **for** i **in** list1: # for Loop
 print(i)

one
two
three
four
five
six
seven
eight
nine
ten

In [245... mylist

Out[245... ['one', 2, 'three', 'four', 'five', 'six', 'seven']

In [247... **for** i **in** mylist:
 print(i)

one
2
three
four
five
six
seven

In [250... **for** i **in** enumerate(list1): # in this enumerate means it will give the index number
 print(i)

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

In [252... **for** i **in** enumerate(mylist):
 print(i)

```
(0, 'one')
(1, 2)
(2, 'three')
(3, 'four')
(4, 'five')
(5, 'six')
(6, 'seven')
```

In [254...]: `l1 = [1,2,3,4,5,0]` # because there is 0 in the l1 if we use all it is false
`all(l1)`

Out[254...]: False

In [256...]: `l2 = [1,2,3,44,5]` # if we use any then it will be true

In [260...]: `any(l1)`

Out[260...]: True

In [262...]: `l1 = [1,2,3,4,True,False]`

In [264...]: `all(l1)` # all the item in the list are false because in the list it contains

Out[264...]: False

In [266...]: `any(l2)` #any of the item in the list will be true because it contains true also

Out[266...]: True

In [268...]: `l2 = [1,2,3,4,True]` # in this list all items will be true because it contains true

In [270...]: `all(l2)`

Out[270...]: True

In [272...]: `l2 = [1,2,3,4,True,False]`

In [274...]: `any(l2)` #because in this list there is true also it will count one value

Out[274...]: True

In [276...]: `all(l2)` # all false because in the list false is there that why the output

Out[276...]: False

In [278...]: `l = [1,2,3,4,5,6]`
`l`

Out[278...]: [1, 2, 3, 4, 5, 6]

In [284...]: `for i in l:`
 `print(i)`

```
1
2
3
4
5
6
```

```
In [282]: for i in enumerate(l):
    print(i)
```

```
(0, 1)
(1, 2)
(2, 3)
(3, 4)
(4, 5)
(5, 6)
```

Tuple

```
In [4]: t = ()      # in the tuple concept tuple always have () open brackets
t
```

```
Out[4]: ()
```

```
In [6]: len(t)
```

```
Out[6]: 0
```

```
In [8]: tup1 = (10,20,30,40)      #tuple of integer numbers
```

```
In [10]: tup2 = (2.3,4.5,6.7,8.9)      #tuple of float numbers
```

```
In [12]: tup3 = ('akshith','bablu','rohit','sharma')      #tuple of strings
```

```
In [16]: tup4 = (10,(100,20),(30,40),'sunil')      #nested tuples
```

```
In [18]: tup5 = (20,2.4,'sunny')      #tuple of mixed datatypes
```

```
In [20]: tup6 = ('akshith',10,(200,30),(500,400) , { 'sunny','uday'} , (20,30,40))
```

```
In [22]: len(tup6)      #Length of the tuple
```

```
Out[22]: 6
```

tuple indexing

```
In [24]: tup1[0]      #it will give the first element of the tuple
```

```
Out[24]: 10
```

```
In [26]: tup2[3]
```

```
Out[26]: 8.9
```

```
In [28]: tup3[2]
```

```
Out[28]: 'rohit'
```

```
In [32]: tup3[0][0]      # it will give first letter of the first index of the tuple
```

```
Out[32]: 'a'
```

```
In [34]: tup6[0][3]
```

```
Out[34]: 'h'
```

```
In [36]: tup4[-1]      # last item of the tuple
```

```
Out[36]: 'sunil'
```

```
In [38]: tup6[-3]      # last third item of the tuple
```

```
Out[38]: (500, 400)
```

tuple slicing

```
In [42]: mytup = ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')
```

```
In [44]: mytup
```

```
Out[44]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')
```

```
In [46]: mytup[0:3]      # it will slice the tuple from 0 to 3-1= 2 then it will return the ou
```

```
Out[46]: ('one', 'two', 'three')
```

```
In [48]: mytup[2:5]      # it will give from 2 to 5-1=4 then it will return the output (n-1)
```

```
Out[48]: ('three', 'four', 'five')
```

```
In [50]: mytup[3:]      #from three onwards it will total entire tuple because there is no n
```

```
Out[50]: ('four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')
```

```
In [52]: mytup[:-3]     # after semi colon if we give -3 for suppose then it will del last thr
```

```
Out[52]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven')
```

```
In [54]: mytup[:-1]
```

```
Out[54]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine')
```

```
In [56]: mytup[-2] # only -2 will be given then the output is last second element
```

```
Out[56]: 'nine'
```

```
In [58]: mytup
```

```
Out[58]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')
```

```
In [60]: mytup[-3] #last third element
```

```
Out[60]: 'eight'
```

```
In [62]: mytup[-2:] #it will give only two values at the last
```

```
Out[62]: ('nine', 'ten')
```

```
In [64]: mytup[3:]
```

```
Out[64]: ('four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')
```

```
In [66]: mytup[:] # it will give whole tuple
```

```
Out[66]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')
```

remove and change items

```
In [11]: mytuple = ('one','two','three','four','five','six','seven','eight','nine','ten')
```

```
In [13]: mytuple
```

```
Out[13]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')
```

```
In [21]: mytuple = ('one','two','three','four','five','six','seven','eight','nine','ten')
mytuple
```

```
Out[21]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')
```

```
In [29]: del mytuple[0] #tuples are immutable which we can't delete tuple items
```

```
-----
```

```
TypeError
```

```
Cell In[29], line 1
```

```
----> 1 del mytuple[0]
```

```
Traceback (most recent call last)
```

```
TypeError: 'tuple' object doesn't support item deletion
```

```
In [27]: mytuple[0] = 1 #tuples are immutable which we can't change the items in the tuple
```

```

-----  

TypeError                                         Traceback (most recent call last)  

Cell In[27], line 1  

----> 1 mytuple[0] = 1  

  

TypeError: 'tuple' object does not support item assignment

```

In [31]: `mytuple[1] = 2`

```

-----  

TypeError                                         Traceback (most recent call last)  

Cell In[31], line 1  

----> 1 mytuple[1] = 2  

  

TypeError: 'tuple' object does not support item assignment

```

In [33]: `del mytuple #deleting entire tuple is possible`

In [35]: `mytuple`

```

-----  

NameError                                         Traceback (most recent call last)  

Cell In[35], line 1  

----> 1 mytuple  

  

NameError: name 'mytuple' is not defined

```

loop through a tuple

In [38]: `mytuple = ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')
mytuple`

Out[38]: `('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')`

In [40]: `for i in mytuple: #for Loop
 print(i)`

```

one
two
three
four
five
six
seven
eight
nine
ten

```

In [48]: `for i in enumerate(mytuple): # enumerate means it will give output as pair va
 print(i)`

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

tuple membership

In [51]: mytuple

Out[51]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')

In [53]: 'one' in mytuple # checking the number is present or not if it is present the

Out[53]: True

In [55]: 'twelve' in mytuple #it is not there in the list

Out[55]: False

In [57]: 'ten' in mytuple

Out[57]: True

```
In [59]: if 'three' in mytuple:
            print('three is present in mytuple')
        else:
            print('three is not present in mytuple')
```

three is present in mytuple

```
In [61]: if "eleven" in mytuple:
            print("eleven is present in mytuple")
        else:
            print("eleven is not present in mytuple")
```

eleven is not present in mytuple

index position

In [64]: mytuple

Out[64]: ('one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten')

In [68]: mytuple.index('one') # it will give the index number for suppose 'one' is pres

```
Out[68]: 0
```

```
In [70]: mytuple.index('ten')
```

```
Out[70]: 9
```

```
In [74]: mytuple.index('two')
```

```
Out[74]: 1
```

count function

```
In [81]: mytuple10 = ('one','two','three','four','five','six','seven','eight','nine','ten','one','two','three','four','five','six','seven','eight','nine','ten','one','one','one','two','three','four','four')
```

```
In [83]: mytuple10
```

```
Out[83]: ('one',
           'two',
           'three',
           'four',
           'five',
           'six',
           'seven',
           'eight',
           'nine',
           'ten',
           'one',
           'one',
           'one',
           'two',
           'three',
           'four',
           'four')
```

```
In [85]: mytuple10.count('two')
```

```
Out[85]: 2
```

```
In [87]: mytuple10.count('one')
```

```
Out[87]: 4
```

sorting

```
In [92]: del(mytuple10) #mytuple 10 will be deleted
```

```
In [94]: mytuple10
```

```
NameError Traceback (most recent call last)
Cell In[94], line 1
----> 1 mytuple10

NameError: name 'mytuple10' is not defined

In [102... mytuple1 = (10,20,30,10,40,80,70,60,50)
mytuple1

Out[102... (10, 20, 30, 10, 40, 80, 70, 60, 50)

In [104... sorted(mytuple1)    # it will sorted in ascending order

Out[104... [10, 10, 20, 30, 40, 50, 60, 70, 80]

In [106... sorted(mytuple1 , reverse=True)    # it will be sorted in descending order (reverse

Out[106... [80, 70, 60, 50, 40, 30, 20, 10, 10]

In [ ]: #by default it will be sorted in ascending order if we give reverse = True then it

In [ ]:
```