In [1]:
```
txt="  abc def ghi  "
txt.lstrip()
```

Out[1]:
```
'abc def ghi  '
```

In [2]:
```
txt="   abc def ghi   "
txt.strip()
```

Out[2]:
```
'abc def ghi'
```

Using Escape Character

In [3]:
```
mystr="My favourite TV Series is "Game of Thrones""
```

```
  Cell In[3], line 1
    mystr="My favourite TV Series is "Game of Thrones""
                                      ^
SyntaxError: invalid syntax
```

In [4]:
```
mystr="My favourite series is \"Game of Thrones\""
print(mystr)
```

```
My favourite series is "Game of Thrones"
```

List creation

In [1]:
```
list1=[]
```

In [6]:
```
print(type(list1))
```

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

In [7]:
```
list2=[10,20,60] #List of integer numbers
```

In [8]:
```
list3=[10.77,30.66,60.89] #list of float numbers
```

In [9]:
```
list4=['one','two','three'] #list of strings
```

In [10]:
```
list5=['dhwani',25,[50,100],[150,90]] #Nested lists
```

In [11]:
```
list6=[100,'dhwani',17.765] #listed of mixed datatypes
```

In [13]:
```
list7=['dhwani',25,[50,100],[150,90],{'John','David'}]
```

In [14]:
```
len(list6) #Length of list
```

Out[14]:
```
3
```

list indexing

In [15]:
```
list2[0] #Retrieve first element of the list
```

Out[15]: 10

In [16]:
```
list4[0] #Retrieve first element of the list
```

Out[16]: 'one'

In [17]:
```
list4[0][0] #Nested indexing - Access the first character of the first list elem
```

Out[17]: 'o'

In [18]:
```
list4[-1] #last item of the list
```

Out[18]: 'three'

In [20]:
```
list5[-1] #last item of the list
```

Out[20]: [150, 90]

list slicing

In [21]:
```
mylist=['one','two','three','four','five','six','seven','eight']
```

In [22]:
```
mylist[0:3] #Return all iitems from 0th to 3rd iindex location excluding the ite
```

Out[22]: ['one', 'two', 'three']

In [23]:
```
mylist[2:5] #list all items from 2nd to 5th index location excluding the item
```

Out[23]: ['three', 'four', 'five']

In [24]:
```
mylist[:3] #Return first three items
```

Out[24]: ['one', 'two', 'three']

In [25]:
```
mylist[:2] #Return first two items
```

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

In [26]:
```
mylist[-3:] #Return last three items
```

Out[26]: ['six', 'seven', 'eight']

In [27]:
```
mylist[-2:] #Return last two items
```

Out[27]: ['seven', 'eight']

In [28]:
```
mylist[-1] #Return last item of the list
```

Out[28]: 'eight'

In [29]:
```
mylist[:] #return whole list
```

Out[29]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

add, remove & change items

In [30]:
```
mylist
```

Out[30]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [31]:
```
mylist.append('nine')
mylist
```

Out[31]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']

In [32]:
```
mylist.insert(9,'ten')
mylist
```

Out[32]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']

In [33]:
```
mylist.insert(1,'ONE')
mylist
```

Out[33]:
```
['one',
 'ONE',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten']
```

In [34]:
```
mylist.remove('ONE') #remove item 'ONE'
mylist
```

Out[34]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']

In [35]:
```
mylist.pop() #remove last item of the list
mylist
```

Out[35]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']

In [36]:
```
mylist.pop(8) #remove item at index location 8
mylist
```

Out[36]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [37]:
```
del mylist[7] #remove item at index location 7
mylist
```

Out[37]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven']

In [38]:
```
#change value of the string
mylist[0]=1
```

```
mylist[1]=2
mylist[2]=3
mylist
```

Out[38]:
```
[1, 2, 3, 'four', 'five', 'six', 'seven']
```

In [39]:
```
mylist.clear() #empty list/ delete all items in the list
mylist
```

Out[39]:
```
[]
```

In [40]:
```
del mylist #delete the whole list
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[40], line 2
      1 del mylist #delete the whole list
----> 2 mylist

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

copy list

In [41]:
```
mylist=['one','two','three','four','five','six','seven','eight','nine']
```

In [42]:
```
mylist1=mylist #creating 'mylist1'
```

In [43]:
```
id(mylist), id(mylist1) #The address of both mylist & mylist1 will be the same
```

Out[43]:
```
(1455123967040, 1455123967040)
```

In [44]:
```
mylist2=mylist.copy() #create a copy of the list
```

In [45]:
```
id(mylist2) #the address of mylist2 will be different from mylist because mylist
```

Out[45]:
```
1455123971456
```

In [46]:
```
mylist[0]=1
```

In [47]:
```
mylist
```

Out[47]:
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [48]:
```
mylist1 #mylist1 will be also impacted as it is pointing to the same list
```

Out[48]:
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [49]:
```
mylist2 #copy of list won't be impacted due to changes made on the original list
```

Out[49]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

join lists

In [50]:
```
list1=['one','two','three','four']
list2=['five','six','seven','eight']
```

In [51]:
```
list3=list1+list2 #join two lists by '+' operator
list3
```

Out[51]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [52]:
```
list1.extend(list2) #append list2 with list1
list1
```

Out[52]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

list membership

In [53]:
```
list1
```

Out[53]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [54]:
```
'one' in list1 #check if 'one' exist in the list
```

Out[54]: True

In [55]:
```
'ten' in list1 #check if 'ten' exist in the list
```

Out[55]: False

In [56]:
```
if 'three' in list1: #check if 'three' exist in the list
    print('eleven is present in the list')
else:
    print('eleven is not present in the list')
```

eleven is present in the list

reverse & sort list

In [57]:
```
list1
```

Out[57]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [58]:
```
list1.reverse() #reverse the list
list1
```

Out[58]: ['eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one']

In [59]:
```
list1=list1[::-1] #reverse the list
list1
```

Out[59]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [60]:
```
mylist3=[9,5,2,99,12,88,34]
mylist3.sort()  #sort lsit in ascending order
mylist3
```

Out[60]:
```
[2, 5, 9, 12, 34, 88, 99]
```

In [61]:
```
mylist3=[9,5,2,99,12,88,34]
mylist3.sort(reverse=True) #sort list in descending order
mylist3
```

Out[61]:
```
[99, 88, 34, 12, 9, 5, 2]
```

In [62]:
```
mylist4=[88,65,33,21,11,98]
sorted(mylist4)
```

Out[62]:
```
[11, 21, 33, 65, 88, 98]
```

In [63]:
```
mylist4
```

Out[63]:
```
[88, 65, 33, 21, 11, 98]
```

loop through a list

In [64]:
```
list1
```

Out[64]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [65]:
```
for i in list1:
    print(i)
```

```
one
two
three
four
five
six
seven
eight
```

In [66]:
```
for i in enumerate(list1):
    print(i)
```

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

count

In [67]:
```
list10=['one','two','three','four','one','one','two','three']
```

In [68]:
```
list10.count('one') #number of times item 'one' occurred in the list
```

Out[68]: 3

In [69]:
```
list10.count('two') #occurence of item 'two' in the list
```

Out[69]: 2

In [70]:
```
list10.count('four') #occurence of item 'four' in the list
```

Out[70]: 1

all/any

In [71]:
```
L1=[1,2,3,4,0]
```

In [73]:
```
all(L1) #will return False as ine value is false(value 0)
```

Out[73]: False

In [74]:
```
any(L1) #will return True as we have items in the with True value
```

Out[74]: True

In [75]:
```
L2=[1,2,3,4,True,False]
```

In [76]:
```
all(L2) #returns False as one value is False
```

Out[76]: False

In [77]:
```
any(L2) #will return True as we have items int he list wthh True value
```

Out[77]: True

In [78]:
```
L3=[1,2,3,True]
```

In [79]:
```
all(L3) #will return True as all items in the list are True
```

Out[79]: True

In [ ]:
```

```

list datastructure

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

Out[13]: []

In [14]:
```
type(l)
```

Out[14]: list

In [15]:
```
print(type(l))
```

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

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

Out[16]: 0

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

In [18]:
```
l
```

Out[18]: [10]

In [19]:
```
l.append(20)
l.append(30)
l.append(40)
l.append(50)
```

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

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

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

Out[21]: 5

In [22]:
```
l.append(10,20)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[22], line 1
----> 1 l.append(10,20)

TypeError: list.append() takes exactly one argument (2 given)
```

In [23]:
```
l.append(10)
```

In [24]:
```
l
```

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

In [26]:
```
l[0]
```

Out[26]: 10

In [27]:
```
l[0]=100
```

In [28]:
```
l
```

Out[28]: [100, 20, 30, 40, 50, 10]

In [29]:
```
l[-1]=-20
```

In [30]:
```
l
```

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

In [31]:
```
l1=l.copy()
l1
```

Out[31]: [100, 20, 30, 40, 50, -20]

In [32]:
```
print(l)
print(l1)
```

```
[100, 20, 30, 40, 50, -20]
[100, 20, 30, 40, 50, -20]
```

In [33]:
```
l==l1
```

Out[33]: True

In [34]:
```
id(l)==id(l1)
```

Out[34]: False

In [35]:
```
id(l)!=id(l1)
```

Out[35]: True

In [36]:
```
l
```

Out[36]: [100, 20, 30, 40, 50, -20]

In [37]:
```
l.count(20)
```

Out[37]: 1

In [38]:
```
l.append(20)
```

In [39]:
```
l
```

Out[39]: [100, 20, 30, 40, 50, -20, 20]

In [40]:
```
l.count(20)
```

Out[40]: 2

In [41]:
```
l1
```

Out[41]: [100, 20, 30, 40, 50, -20]

In [42]:
```
len(l1)
```

Out[42]: 6

In [43]:
```
l1.clear()
```

In [44]:
```
l1
```

Out[44]: []

In [45]:
```
del l1
```

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

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[46], line 1
----> 1 l1

NameError: name 'l1' is not defined
```

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

Out[47]: [100, 20, 30, 40, 50, -20, 20]

In [52]:
```
l2=[]
```

In [53]:
```
l2.append(9)
l2.append('hi')
l2.append(True)
l2.append(9+2j)
l2.append(9.3)
```

In [54]:
```
l2
```

Out[54]: [9, 'hi', True, (9+2j), 9.3]

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

```
[100, 20, 30, 40, 50, -20, 20]
```

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

```
[9, 'hi', True, (9+2j), 9.3]
```

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

Out[57]: [100, 20, 30, 40, 50, -20, 20]

In [58]:
```
l.index(100)
```

Out[58]: 0

In [59]:
```
l.index(20)
```

Out[59]: 1

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

Out[60]: [9, 'hi', True, (9+2j), 9.3]

In [61]:
```
l2[1]
```

Out[61]: 'hi'

In [62]:
```
l2[1][0]
```

Out[62]: 'h'

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

Out[63]: [9, 'hi', True, (9+2j), 9.3]

In [64]:
```
print(l2[1])
```

```
hi
```

In [66]:
```
print(l2[1][0])
print(l2[1][1])
```

```
h
i
```

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

Out[67]: [100, 20, 30, 40, 50, -20, 20]

In [68]:
```
l[:]
```

Out[68]: [100, 20, 30, 40, 50, -20, 20]

In [69]:
```
l[3:]
```

Out[69]: [40, 50, -20, 20]

In [70]:
```
l
```

Out[70]: [100, 20, 30, 40, 50, -20, 20]

In [71]:
```
l[:10]
```

Out[71]: [100, 20, 30, 40, 50, -20, 20]

In [72]:
```
l
```

Out[72]: [100, 20, 30, 40, 50, -20, 20]

In [73]:
```
l[0:8:3]
```

Out[73]: [100, 40, 20]

In [74]:
```
l2
```

Out[74]: [9, 'hi', True, (9+2j), 9.3]

In [75]:
```
l2[1:6:2]
```

Out[75]: ['hi', (9+2j)]

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

Out[76]: [100, 20, 30, 40, 50, -20, 20]

In [77]:
```
l.insert(18,2)
```

In [78]:
```
l
```

Out[78]: [100, 20, 30, 40, 50, -20, 20, 2]

In [79]:
```
l.insert(2,24)
```

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

Out[80]: [100, 20, 24, 30, 40, 50, -20, 20, 2]

In [81]:
```
l.pop()
```

Out[81]: 2

In [82]:
```
l
```

Out[82]: [100, 20, 24, 30, 40, 50, -20, 20]

In [83]:
```
l.pop()
```

Out[83]: 20

In [84]:
```
l
```

Out[84]: [100, 20, 24, 30, 40, 50, -20]

In [85]:
```
l2
```

Out[85]: [9, 'hi', True, (9+2j), 9.3]

In [86]:
```
l2.pop(3)
```

Out[86]: (9+2j)

In [87]:
```
l2
```

Out[87]: [9, 'hi', True, 9.3]

In [88]:
```
l2.remove('hi')
```

In [89]:
```
l2
```

Out[89]: [9, True, 9.3]

In [ ]:
```

```