```
In [1]: txt = " abc def ghi "

        txt.lstrip()

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

```
In [3]: txt = " abc def ghi "

        txt.strip()

Out[3]: 'abc def ghi'
```

Using Escape Character

```
In [6]: #Using double quotes in the string is not allowed.
        mystr = "My favourite TV Series is "Game of Thrones""
```

```
  Cell In[6], line 2
    mystr = "My favourite TV Series is "Game of Thrones""
                                        ^
SyntaxError: invalid syntax
```

```
In [8]: #Using escape character to allow illegal characters
        mystr = "My favourite series is \"Game of Thrones\""
        print(mystr)

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

List

1) List is an ordered sequence of items.

2) We can have different data types under a list. E.g we can have integer, float and string items in a same list.

List Creation

```
In [13]: list1 = [] # Empty List
```

```
In [15]: print(type(list1))

<class 'list'>
```

```
In [19]: list2 = [10,30,60] # List of integers numbers
```

```
In [21]: list3 = [10.77,30.66,60.89] # List of float numbers
```

```
In [23]: list4 = ['one','two' , "three"] # List of strings
```

```
In [25]: list5 = ['Asif', 25 ,[50, 100],[150, 90]] # Nested Lists
In [27]: list6 = [100, 'Asif', 17.765] # List of mixed data types
```

```
In [29]: list7 = ['Asif', 25 ,[50, 100],[150, 90] , {'John' , 'David'}]
```

```
In [31]: len(list6) #Length of list
Out[31]: 3
```

List Indexing

```
In [34]: list2[0] # Retreive first element of the list
Out[34]: 10
```

```
In [36]: list4[0] # Retreive first element of the list
Out[36]: 'one'
```

```
In [40]: list4[0][0] # Nested indexing - Access the first character of the first list ele
Out[40]: 'o'
```

```
In [42]: list4[-1] # Last item of the list
Out[42]: 'three'
```

```
In [44]: list5[-1] # Last item of the list
Out[44]: [150, 90]
```

List Slicing

```
In [47]: mylist = ['one' , 'two' , 'three' , 'four' , 'five' , 'six' , 'seven' , 'eight']
```

```
In [49]: mylist[0:3] # Return all items from 0th to 3rd index location excluding the item
Out[49]: ['one', 'two', 'three']
```

```
In [51]: mylist[2:5] # List all items from 2nd to 5th index location excluding the item a
Out[51]: ['three', 'four', 'five']
```

```
In [53]: mylist[:3] # Return first three items
Out[53]: ['one', 'two', 'three']
```

```
In [55]: mylist[:2] # Return first two items
Out[55]: ['one', 'two']
```

```
In [57]: mylist[-3:] # Return last three items
```

```
Out[57]: ['six', 'seven', 'eight']
```

```
In [59]: mylist[-2:] # Return last two items
```

```
Out[59]: ['seven', 'eight']
```

```
In [61]: mylist[-1] # Return last item of the list
```

```
Out[61]: 'eight'
```

```
In [63]: mylist[:] # Return whole list
```

```
Out[63]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

Add , Remove & Change Items

```
In [66]: mylist
```

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

```
In [68]: mylist.append('nine') # Add an item to the end of the list
         mylist
```

```
Out[68]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

```
In [70]: mylist.insert(9,'ten') # Add item at index location 9
         mylist
```

```
Out[70]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [72]: mylist.insert(1,'ONE') # Add item at index location 1
         mylist
```

```
Out[72]: ['one',
          'ONE',
          'two',
          'three',
          'four',
          'five',
          'six',
          'seven',
          'eight',
          'nine',
          'ten']
```

```
In [74]: mylist.remove('ONE') # Remove item "ONE"
         mylist
```

```
Out[74]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [76]: mylist.pop() # Remove last item of the list
         mylist
```

```
Out[76]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [78]:
```
mylist.pop(8) # Remove item at index location 8
mylist
```

Out[78]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [80]:
```
del mylist[7] # Remove item at index location 7
mylist
```

Out[80]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

In [82]:
```
# Change value of the strin
mylist[0] = 1
mylist[1] = 2
mylist[2] = 3
mylist
```

Out[82]:
```
[1, 2, 3, 'four', 'five', 'six', 'seven']
```

In [84]:
```
mylist.clear() # Empty List / Delete all items in the list
mylist
```

Out[84]:
```
[]
```

In [86]:
```
del mylist # Delete the whole list
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[86], line 2
      1 del mylist # Delete the whole list
----> 2 mylist

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

Copy List

In [91]:
```
mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine'
```

In [93]:
```
mylist1 = mylist # Create a new reference "mylist1"
```

In [95]:
```
id(mylist) , id(mylist1) # The address of both mylist & mylist1 will be the same
```

Out[95]:
```
(2383703782848, 2383703782848)
```

In [97]:
```
mylist2 = mylist.copy() # Create a copy of the list
```

In [99]:
```
id(mylist2) # The address of mylist2 will be different from mylist because mylis
```

Out[99]:
```
2383703863808
```

In [101…
```
mylist[0] = 1
```

In [103…
```
mylist
```

Out[103…

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

In [105…

```python
mylist1 # mylist1 will be also impacted as it is pointing to the same list
```

Out[105…

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

In [107…

```python
mylist2 # Copy of list won't be impacted due to changes made on the original lis
```

Out[107…

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

Join List

In [110…

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

In [112…

```python
list3 = list1 + list2 # Join two lists by '+' operator
list3
```

Out[112…

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

In [114…

```python
list1.extend(list2) #Append list2 with list1
list1
```

Out[114…

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

List Membership

In [117…

```python
list1
```

Out[117…

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

In [119…

```python
'one' in list1 # Check if 'one' exist in the list
```

Out[119…

```
True
```

In [121…

```python
'ten' in list1 # Check if 'ten' exist in the list
```

Out[121…

```
False
```

In [125…

```python
if 'three' in list1: # Check if 'three' exist in the list
    print('Three is present in the list')
else:
    print('Three is not present in the list')
```

```
Three is present in the list
```

In [127…

```python
if 'eleven' in list1:
    print('eleven is present in the list')
else:
    print('eleven is not present in the list') # Check if 'eleven' exist in the
```

```
eleven is not present in the list
```

Reverse and sort list

In [130…
```
list1
```
Out[130…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```
In [132…
```
list1.reverse() # Reverse the list
list1
```
Out[132…
```
['eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one']
```
In [134…
```
list1 = list1[::-1] # Reverse the list
list1
```
Out[134…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```
In [136…
```
mylist3 = [9,5,2,99,12,88,34]
mylist3.sort() # Sort list in ascending order
mylist3
```
Out[136…
```
[2, 5, 9, 12, 34, 88, 99]
```
In [138…
```
mylist3 = [9,5,2,99,12,88,34]
mylist3.sort(reverse=True) # Sort list in descending order
mylist3
```
Out[138…
```
[99, 88, 34, 12, 9, 5, 2]
```
In [140…
```
mylist4 = [88,65,33,21,11,98]
sorted(mylist4) # Returns a new sorted list and doesn't change original list
```
Out[140…
```
[11, 21, 33, 65, 88, 98]
```
In [142…
```
mylist4
```
Out[142…
```
[88, 65, 33, 21, 11, 98]
```

Loop through list

In [145…
```
list1
```
Out[145…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```
In [149…
```
for i in list1:
    print(i)
```

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

In [151…

```
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 [154…

```
list10 =['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
```

In [156…

```
list10.count('one') # Number of times item "one" occurred in the list.
```

Out[156…

```
3
```

In [158…

```
list10.count('two') # Occurence of item 'two' in the list
```

Out[158…

```
2
```

In [160…

```
list10.count('four') #Occurence of item 'four' in the list
```

Out[160…

```
1
```

All / Any

The all() method returns:

True - If all elements in a list are true

False - If any element in a list is false

The any() function returns True if any element in the list is True. If not, any() returns False.

In [163…

```
L1 = [1,2,3,4,0]
```

In [165…

```
all(L1) # Will Return false as one value is false (Value 0)
```

```
Out[165…  False

In [167…  any(L1) # Will Return True as we have items in the list with True value

Out[167…  True

In [169…  L2 = [1,2,3,4,True,False]

In [171…  all(L2) # Returns false as one value is false

Out[171…  False

In [173…  any(L2) # Will Return True as we have items in the list with True value

Out[173…  True

In [175…  L3 = [1,2,3,True]

In [177…  all(L3) # Will return True as all items in the list are True

Out[177…  True

  In [ ]:
```