

```
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]: #Using double quotes in the string is not allowed.  
mystr = "My favourite TV Series is "Game of Thrones""  
  
File "<ipython-input-3-0fa35a74da86>", line 2  
    mystr = "My favourite TV Series is "Game of Thrones""  
                                         ^  
SyntaxError: invalid syntax
```

```
In [4]: #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 [5]: list1 = [] # Empty List
```

```
In [6]: print(type(list1))  
  
<class 'list'>
```

```
In [7]: list2 = [10,30,60] # List of integers 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 = ['Asif', 25 , [50, 100],[150, 90]] # Nested Lists
```

```
In [11]: list6 = [100, 'Asif', 17.765] # List of mixed data types
```

```
In [12]: list7 = ['Asif', 25 , [50, 100],[150, 90] , {'John' , 'David'}]
```

```
In [13]: len(list6) #Length of list
```

```
Out[13]: 3
```

List Indexing

```
In [14]: list2[0] # Retrieve first element of the list
```

```
Out[14]: 10
```

```
In [15]: list4[0] # Retrieve first element of the list
```

```
Out[15]: 'one'
```

```
In [16]: list4[0][0] # Nested indexing - Access the first character of the first element
```

```
Out[16]: 'o'
```

```
In [17]: list4[-1] # Last item of the list
```

```
Out[17]: 'three'
```

```
In [18]: list5[-1] # Last item of the list
```

```
Out[18]: [150, 90]
```

List Slicing

```
In [19]: mylist = ['one' , 'two' , 'three' , 'four' , 'five' , 'six' , 'seven']
```

```
In [20]: mylist[0:3] # Return all items from 0th to 3rd index location excluding 3rd index
```

```
Out[20]: ['one', 'two', 'three']
```

```
In [21]: mylist[2:5] # List all items from 2nd to 5th index location excluding 5th index
```

```
Out[21]: ['three', 'four', 'five']
```

```
In [22]: mylist[:3] # Return first three items
```

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

```
In [23]: mylist[:2] # Return first two items
```

```
Out[23]: ['one', 'two']
```

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

```
Out[24]: ['six', 'seven', 'eight']
```

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

```
Out[25]: ['seven', 'eight']
```

```
In [26]: mylist[-1] # Return last item of the list
```

```
Out[26]: 'eight'
```

```
In [27]: mylist[:] # Return whole list
```

```
Out[27]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

Add , Remove & Change Items

```
In [28]: mylist
```

```
Out[28]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [29]: mylist.append('nine') # Add an item to the end of the list  
mylist
```

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

```
In [30]: mylist.insert(9,'ten') # Add item at index location 9  
mylist
```

```
Out[30]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'ni  
ne', 'ten']
```

```
In [31]: mylist.insert(1,'ONE') # Add item at index location 1  
mylist
```

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

```
In [32]: mylist.remove('ONE') # Remove item "ONE"  
mylist
```

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

```
In [33]: mylist.pop() # Remove last item of the list
mylist
```

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

```
In [34]: mylist.pop(8) # Remove item at index location 8
mylist
```

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

```
In [35]: del mylist[7] # Remove item at index location 7
mylist
```

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

```
In [36]: # Change value of the string
mylist[0] = 1
mylist[1] = 2
mylist[2] = 3
mylist
```

```
Out[36]: [1, 2, 3, 'four', 'five', 'six', 'seven']
```

```
In [37]: mylist.clear() # Empty List / Delete all items in the list
mylist
```

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

```
In [38]: del mylist # Delete the whole list
mylist
```

```
-----
-----
NameError                                Traceback (most recent call last)
<ipython-input-38-50c7849aa2cb> in <module>
      1 del mylist # Delete the whole list
----> 2 mylist

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

Copy List

```
In [40]: mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [41]: mylist1 = mylist # Create a new reference "mylist1"
```

```
In [42]: id(mylist) , id(mylist1) # The address of both mylist & mylist1 will be same
```

```
Out[42]: (140270951800256, 140270951800256)
```

```
In [43]: mylist2 = mylist.copy() # Create a copy of the list
```

```
In [44]: id(mylist2) # The address of mylist2 will be different from mylist bec
```

```
Out[44]: 140270951813952
```

```
In [45]: mylist[0] = 1
```

```
In [49]: print(mylist)
```

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

```
In [50]: mylist1 # mylist1 will be also impacted as it is pointing to the same
```

```
Out[50]: [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

```
In [51]: mylist2 # Copy of list won't be impacted due to changes made on the or
```

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

Join Lists

```
In [52]: list1 = ['one', 'two', 'three', 'four']  
list2 = ['five', 'six', 'seven', 'eight']
```

```
In [53]: list3 = list1 + list2 # Join two lists by '+' operator  
list3
```

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

```
In [54]: list1.extend(list2) #Append list2 with list1  
list1
```

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

List Membership

```
In [55]: list1
```

```
Out[55]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [56]: 'one' in list1 # Check if 'one' exist in the list
```

```
Out[56]: True
```

```
In [57]: 'ten' in list1 # Check if 'ten' exist in the list
```

```
Out[57]: False
```

```
In [58]: 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 [60]: if 'eleven' in list1: # Check if 'eleven' exist in the list
          print('eleven is present in the list')
        else:
          print('eleven is not present in the list')
```

eleven is not present in the list

Reverse & Sort List

```
In [61]: list
```

```
Out[61]: list
```

```
In [62]: list1.reverse() # Reverse the list
          list1
```

```
Out[62]: ['eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one']
```

```
In [63]: list1 = list1[::-1] # Reverse the list
          list1
```

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

```
In [65]: mylist3 = [9,5,2,99,12,88,34]
          print(mylist3)
          mylist3.sort() # Sort list in ascending order
          print(mylist3)
```

```
[9, 5, 2, 99, 12, 88, 34]
[2, 5, 9, 12, 34, 88, 99]
```

```
In [66]: mylist3 = [9,5,2,99,12,88,34]
          mylist3.sort(reverse=True) # Sort list in descending order
          mylist3
```

```
Out[66]: [99, 88, 34, 12, 9, 5, 2]
```

```
In [69]: mylist4 = [88,65,33,21,11,98]
          print(sorted(mylist4)) # Returns a new sorted list and doesn't change
          print(mylist4)
```

```
[11, 21, 33, 65, 88, 98]
[88, 65, 33, 21, 11, 98]
```

Loop through a list

```
In [71]: list1
```

```
Out[71]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [72]: for i in list1:
          print(i)
```

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

```
In [73]: 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 [74]: list10 = ['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
```

```
In [75]: list10.count('one') # Number of times item "one" occurred in the list.
```

```
Out[75]: 3
```

```
In [76]: list10.count('two') # Occurrence of item 'two' in the list
```

```
Out[76]: 2
```

```
In [77]: list10.count('four') # Occurrence of item 'four' in the list
```

```
Out[77]: 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

Loading [MathJax]/jax/output/HTML-CSS/fonts/STIX-Web/fontdata.js

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

```
In [79]: L1 = [1,2,3,4,0]
```

```
In [80]: all(L1) # Will Return false as one value is false (Value 0)
```

```
Out[80]: False
```

```
In [81]: any(L1) # Will Return True as we have items in the list with True value
```

```
Out[81]: True
```

```
In [82]: L2 = [1,2,3,4,True,False]
```

```
In [83]: all(L2) # Returns false as one value is false
```

```
Out[83]: False
```

```
In [84]: any(L2) # Will Return True as we have items in the list with True value
```

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

```
In [85]: L3 = [1,2,3,True]
```

```
In [86]: all(L3) # Will return True as all items in the list are True
```

```
Out[86]: True
```

```
In [ ]:
```