```
In [265]: txt = "   abc def ghi   "

          txt.lstrip()

Out[265]: 'abc def ghi   '
```

```
In [266]: txt = "   abc def ghi   "

          txt.strip()
Out[266]: 'abc def ghi'
```

Using Escape Character

```
In [252]: #Using double quotes in the string is not allowed.
          mystr = "My favourite TV Series is "Game of Thrones""

            File "<ipython-input-252-0fa35a74da86>", line 2
              mystr = "My favourite TV Series is "Game of Thrones""
                                                     ^

          SyntaxError: invalid syntax
```

```
In [253]: #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 [423]: list1 = []       # Empty List
```

```
In [491]: print(type(list1))

          <class 'list'>
```

```
In [424]: list2 = [10,30,60]       # List of integers numbers
```

```
In [425]: list3 = [10.77,30.66,60.89]        # List of float numbers
In [426]: list4 = ['one','two' , "three"]   # List of strings
```

```
In [427]: list5 = ['Asif', 25 ,[50, 100],[150, 90]]    # Nested Lists
```

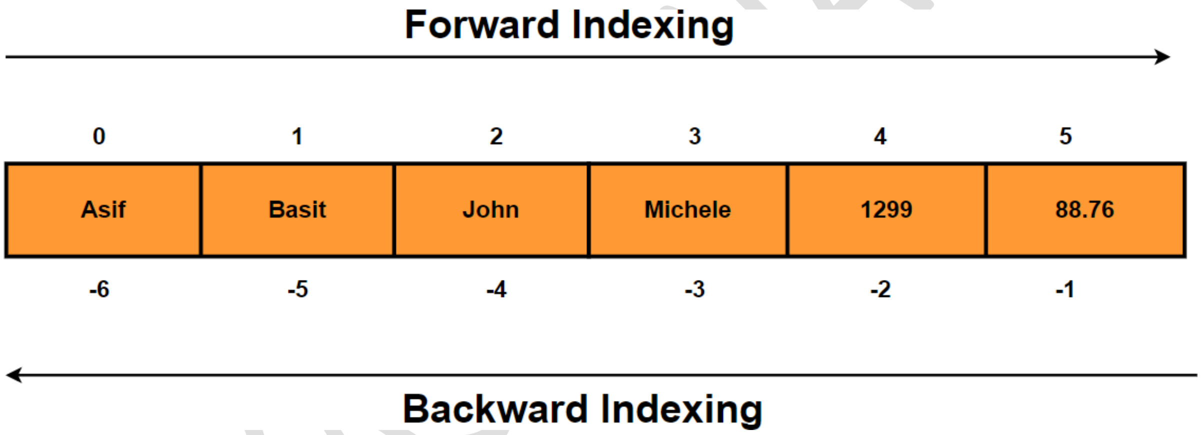
```
In [428]: list6 = [100, 'Asif', 17.765]   # List of mixed data types
```

```
In [429]: list7 = ['Asif', 25 ,[50, 100],[150, 90] , {'John' , 'David'}]
```

```
In [430]: len(list6) #Length of list

Out[430]: 3
```

List Indexing



```
In [432]: list2[0] # Retreive first element of the list

Out[432]: 10
```

```
In [433]: list4[0] # Retreive first element of the list

Out[433]: 'one'
```

```
In [434]: list4[0][0] # Nested indexing - Access the first character of the first list ele

Out[434]: 'o'
```

```
In [435]: list4[-1] # Last item of the list

Out[435]: 'three'
```

```
In [436]: list5[-1]  # Last item of the list

Out[436]: [150, 90]
```

List Slicing

```
In [437]: mylist = ['one' , 'two' , 'three' , 'four' , 'five' , 'six' , 'seven' , 'eight']
```

```
In [438]: mylist[0:3] # Return all items from 0th to 3rd index location excluding the item

Out[438]: ['one', 'two', 'three']
```

```
In [439]: mylist[2:5] # List all items from 2nd to 5th index location excluding the item a

Out[439]: ['three', 'four', 'five']
```

```
In [440]: mylist[:3] # Return first three items

Out[440]: ['one', 'two', 'three']
```

```
In [441]: mylist[:2]  # Return first two items

Out[441]: ['one', 'two']
```

```
In [442]: mylist[-3:] # Return last three items

Out[442]: ['six', 'seven', 'eight']
```

```
In [443]: mylist[-2:] # Return last two items

Out[443]: ['seven', 'eight']
```

```
In [444]: mylist[-1] # Return last item of the list

Out[444]: 'eight'
```

```
In [445]: mylist[:] # Return whole list

Out[445]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

Add , Remove & Change Items

```
In [446]: mylist

Out[446]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [447]: mylist.append('nine') # Add an item to the end of the list
          mylist

Out[447]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

```
In [448]: mylist.insert(9,'ten') # Add item at index location 9
          mylist

Out[448]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [449]: mylist.insert(1,'ONE') # Add item at index location 1
          mylist

Out[449]: ['one',
           'ONE',
           'two',
           'three',
           'four',
           'five',
           'six',
           'seven',
           'eight',
           'nine',
           'ten']
```

```
In [450]: mylist.remove('ONE') # Remove item "ONE"
          mylist

Out[450]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [451]: mylist.pop() # Remove last item of the list
          mylist

Out[451]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

```
In [452]: mylist.pop(8) # Remove item at index location 8
          mylist

Out[452]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [453]: del mylist[7] # Remove item at index location 7
          mylist

Out[453]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

```
In [454]: # Change value of the string
          mylist[0] = 1
          mylist[1] = 2
          mylist[2] = 3
          mylist

Out[454]: [1, 2, 3, 'four', 'five', 'six', 'seven']
```

```
In [455]: mylist.clear()  # Empty List / Delete all items in the list
          mylist

Out[455]: []
```

```
In [456]: del mylist # Delete the whole list
          mylist

          ---------------------------------------------------------------------------
          NameError                                 Traceback (most recent call last)
          <ipython-input-456-50c7849aa2cb> in <module>
                1 del mylist # Delete the whole list
          ----> 2 mylist

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

Copy List

```
In [457]: mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine'
```

```
In [458]: mylist1 = mylist # Create a new reference "mylist1"
```

```
In [459]: id(mylist) , id(mylist1) # The address of both mylist & mylist1 will be the same

Out[459]: (1537348392776, 1537348392776)
```

```
In [460]: mylist2 = mylist.copy() # Create a copy of the list
```

```
In [461]: id(mylist2) # The address of mylist2 will be different from mylist because mylis

Out[461]: 1537345955016
```

```
In [462]: mylist[0] = 1
```

```
In [463]: mylist

Out[463]: [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [464]: mylist1 # mylist1 will be also impacted as it is pointing to the same list

Out[464]: [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [465]: mylist2 # Copy of list won't be impacted due to changes made on the original lis

Out[465]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

Join Lists

```
In [466]: list1 = ['one', 'two', 'three', 'four']
          list2 = ['five', 'six', 'seven', 'eight']
```

```
In [467]: list3 = list1 + list2 # Join two lists by '+' operator
          list3

Out[467]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [468]: list1.extend(list2) #Append list2 with list1
          list1

Out[468]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

List Membership

```
In [469]: list1

Out[469]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [470]: 'one' in list1 # Check if 'one' exist in the list

Out[470]: True
```

```
In [471]: 'ten' in list1 # Check if 'ten' exist in the list

Out[471]: False
```

```
In [472]: 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 [473]: 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 [474]: list1

Out[474]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [475]: list1.reverse() # Reverse the list
          list1

Out[475]: ['eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one']
```

```
In [476]: list1 = list1[::-1] # Reverse the list
          list1

Out[476]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [477]: mylist3 = [9,5,2,99,12,88,34]
          mylist3.sort()   # Sort list in ascending order
          mylist3

Out[477]: [2, 5, 9, 12, 34, 88, 99]
```

```
In [478]: mylist3 = [9,5,2,99,12,88,34]
          mylist3.sort(reverse=True)  # Sort list in descending order
          mylist3

Out[478]: [99, 88, 34, 12, 9, 5, 2]
```

```
In [584]: mylist4 = [88,65,33,21,11,98]
          sorted(mylist4)       # Returns a new sorted list and doesn't change original l

Out[584]: [11, 21, 33, 65, 88, 98]
```

```
In [585]: mylist4

Out[585]: [88, 65, 33, 21, 11, 98]
```

Loop through a list

```
In [481]: list1

Out[481]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [482]: for i in list1:
              print(i)

          one
          two
          three
          four
          five
          six
          seven
          eight
```

```
In [483]: 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 [485]: list10 =['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
```

```
In [486]: list10.count('one') # Number of times item "one" occurred in the list.

Out[486]: 3
```

```
In [487]: list10.count('two') # Occurence of item 'two' in the list

Out[487]: 2
```

```
In [489]: list10.count('four') #Occurence of item 'four' in the list

Out[489]: 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 [816]: L1 = [1,2,3,4,0]
```

```
In [817]: all(L1) # Will Return false as one value is false (Value 0)

Out[817]: False
```

```
In [818]: any(L1)  # Will Return True as we have items in the list with True value

Out[818]: True
```

```
In [819]: L2 = [1,2,3,4,True,False]
```

```
In [820]: all(L2) # Returns false as one value is false

Out[820]: False
```

```
In [821]: any(L2)  # Will Return True as we have items in the list with True value

Out[821]: True
```

```
In [822]: L3 = [1,2,3,True]
```

```
In [823]: all(L3) # Will return True as all items in the list are True

Out[823]: True
```