In [335...

```
txt=" abc def ghi "
txt.lstrip()
```

Out[335...

```
'abc def ghi '
```

In [337...

```
txt= " abc, def, ghi "
txt.strip()
```

Out[337...

```
'abc, def, ghi'
```

Using Escape Character

In [340...

```
mystr=" I was born  " in the year of 2001""
mystr
```

```
  Cell In[340], line 1
    mystr=" I was born  " in the year of 2001""
                                 ^

SyntaxError: invalid syntax
```

In [342...

```
mystr=" I was born \"in the year of 2001\""
mystr
```

Out[342...

```
' I was born "in the year of 2001"'
```

LIST

1. List is an orderd 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 [346...

```
list=[]
```

In [348...

```
list1=[10,20,30,40] # List of Integer Numbers
```

In [350...

```
list2=[3.4,6.7,3.9] # List of float Numbers
```

In [352...

```
list3=['one','two','three'] # list of strings
```

In [354...

```
list4=[45,7.9,'one',True] # mixed data types
```

In [356...

```
list5=[78,26,(1,2,3),[6,7,8]] # Nested Lists
```

In [358...

```
len(list4) # Length of the list
```

Out[358...

```
4
```

List Indexing

1. Forward Indexing
2. Backward Indexing

In [361…
```
list4
```
Out[361…
```
[45, 7.9, 'one', True]
```

In [363…
```
list4[0] # retrieve the first element of the list
```
Out[363…
```
45
```

In [365…
```
list4[2] # retrieve the 3rd element of index location 2
```
Out[365…
```
'one'
```

In [367…
```
list4[-1] # retrieve the last element of the list
```
Out[367…
```
True
```

In [369…
```
list3[0][0] # nested index - access the first character of the first element
```
Out[369…
```
'o'
```

In [371…
```
list4[:] # whole list
```
Out[371…
```
[45, 7.9, 'one', True]
```

List Slicing

In [374…
```
list4
```
Out[374…
```
[45, 7.9, 'one', True]
```

In [376…
```
list4[0:2] # returns all items  from index 0th index  to 2nd index location excl
```
Out[376…
```
[45, 7.9]
```

In [378…
```
list4[1:3] # returns all items from index 1st index to 3rd index location exclud
```
Out[378…
```
[7.9, 'one']
```

In [380…
```
list4[:3] # returns first three intems
```
Out[380…
```
[45, 7.9, 'one']
```

In [382…
```
list4[:2] # returns first two items
```
Out[382…
```
[45, 7.9]
```

```
In [384…   list4[-2:] # returns last two items

Out[384…   ['one', True]

In [386…   list4[-3:] # returns last three items

Out[386…   [7.9, 'one', True]

In [388…   list4[-1:] # returns last item

Out[388…   [True]
```

Add ,Remove & change items

```
In [391…   list1

Out[391…   [10, 20, 30, 40]

In [393…   list1.append(50)  # add the element at the end of the list
           list1.append(60)
           list1

Out[393…   [10, 20, 30, 40, 50, 60]

In [395…   list1.insert(6,80) # add the element at index location 6
           list1

Out[395…   [10, 20, 30, 40, 50, 60, 80]

In [397…   list1.insert(7,90) # add the element at index loaction 7
           list1

Out[397…   [10, 20, 30, 40, 50, 60, 80, 90]

In [399…   list1.pop() # removes the last element
           list1

Out[399…   [10, 20, 30, 40, 50, 60, 80]

In [401…   list1.pop(2) # removes the element at index location 2
           list1

Out[401…   [10, 20, 40, 50, 60, 80]

In [403…   list1.remove(40) # reomoves the element 40
           list1

Out[403…   [10, 20, 50, 60, 80]

In [405…   del list1[2] # removes the element at index location 2
           list1

Out[405…   [10, 20, 60, 80]
```

```
In [407…  # Changes the value of integer numbers
          list1[2]=30
          list1[3]=40
          list1

Out[407…  [10, 20, 30, 40]
```

```
In [409…  list2.clear() # delete all items in the list
          list2

Out[409…  []
```

```
In [411…  del list # deletes the whole list
          list

Out[411…  list
```

Copy List

```
In [414…  list=['one','two','three','four','five','six','seven','eight','nine','ten']
          list

Out[414…  ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [416…  list1=list # create a new reference 'list1'
```

```
In [418…  id(list),id(list1) # the address of both list & list1 will be the same

Out[418…  (1978452745088, 1978452745088)
```

```
In [420…  list2=list.copy() #create a copy of the list
```

```
In [422…  id(list2) # address of the list2 will be different from list because list

Out[422…  1978452576256
```

```
In [424…  list[0]=1
          list

Out[424…  [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [426…  list1 # list1 will be also impacted as it is pointing to the same list

Out[426…  [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

```
In [428…  list2 # copy of list won't be impacted due to changes made on the original list

Out[428…  ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']
```

Join Lists

In [431…
```
list1=['one','two','three','four']
list2=['five','six','seven','eight']
```

In [433…
```
list3=list1+list2 # Join two lists by '+' operator
list3
```

Out[433…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [435…
```
list1.extend(list2)  # Append list2 with list1
list1
```

Out[435…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

List Membership

In [438…
```
list1
```

Out[438…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [440…
```
'one' in list1 # checks if 'one' exists in the list
```

Out[440…
```
True
```

In [442…
```
'eleven' in list1 # check if 'eleven' exists in the list
```

Out[442…
```
False
```

In [444…
```
'seven' in list1 # check if 'seven' exists in the list
```

Out[444…
```
True
```

In [446…
```
if 'three' in list1:
    print('three is present in the list')
else:
    print('three  is not present in the list')
```

```
three is present in the list
```

In [448…
```
if 'ten' in list1:
    print('ten is present in the list')
else:
    print('ten is not present in the list')
```

```
ten is not present in the list
```

Reverse & Sort List

In [451…
```
list1
```

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

In [453…
```
list1.reverse() # reverse the list
list1
```

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

```
In [455…   list1=list1[::-1] # Reverse the list
           list1
```

```
Out[455…   ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [457…   list3=[9,5,2,99,12,88,34]
           list3.sort() # Sort list in ascending order
           list3
```

```
Out[457…   [2, 5, 9, 12, 34, 88, 99]
```

```
In [459…   list3.sort(reverse=True) # Sort list in descending order
           list3
```

```
Out[459…   [99, 88, 34, 12, 9, 5, 2]
```

```
In [461…   list4=[88,65,33,21,11,98]
           sorted(list4) # Returns a new sorted list and doesn't change original list
```

```
Out[461…   [11, 21, 33, 65, 88, 98]
```

```
In [463…   list4
```

```
Out[463…   [88, 65, 33, 21, 11, 98]
```

Loop through a List

```
In [466…   list1
```

```
Out[466…   ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [468…   for i in list1:
               print(i)
```

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

```
In [470…   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 [473…
```
list10=['one','two','three','four','one','one','two','three']
list10
```

Out[473…
```
['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
```

In [475…
```
list10.count('one') # Number of items 'one' occurred in the list.
```

Out[475…
```
3
```

In [477…
```
list10.count('two') # Occurenceof item 'two' in the list
```

Out[477…
```
2
```

In [479…
```
list10.count('three') # Occurence of item 'three' in the list
```

Out[479…
```
2
```

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() returnsFalse

In [482…
```
L1=[1,2,3,4,0]
```

In [484…
```
all(L1) # will return false as one value is false (value0)
```

Out[484…
```
False
```

In [486…
```
any(L1) # Will return True as we have items in the list with True value
```

Out[486…
```
True
```

In [488…
```
L2=[1,2,3,4,True,False]
```

In [490…
```
all(L2) # Returns False as one value is False
```

Out[490…
```
False
```

In [492…
```
any(L2) # Will Return True as we have items in the list with True value
```

Out[492…
```
True
```

In [494…
```
L3=[1,2,3,True]
```

In [496…
```
all(L3) # Will return True as all items in the list are True
```

```
Out[496…  True
```

```
In [498…  any(L3) # Will return True as we have items in the list with True value
```

```
Out[498…  True
```