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 "Games of Thrones""
```

```
  Cell In[6], line 1
    mystr = "My favourite TV Series is "Games 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 [12]:
```
list1 = []  # Empty List
```

In [20]:
```
print(type(list1))
```

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

In [21]:
```
list2 = [10,30,60]  # List of integers numbers
```

In [25]:
```
list3 = [10.77,30.66,60.89]  # List of float numbers
```

In [27]:
```
list4 = ['one','two', "three"]  # List of strings
```

In [29]:
```
list5 = ['Asif', 25 ,[50, 100],[150, 90]]  # Nested Lists
```

In [33]:
```
list6 = [100, 'Asif', 17.765] # List of mixed data types
```

In [35]:
```
list7 = ['Asif', 25 ,[50, 100],[150, 90] , {'John' , 'David'}]
```

In [37]:
```
len(list6) #Length of list
```

Out[37]: 3

List Indexing

- Forward Indexing
- Backward Indexing

In [42]:
```
list2[0] # Retreive first element of the list
```

Out[42]: 10

In [44]:
```
list4[0] # Retreive first element of the list
```

Out[44]: 'one'

In [48]:
```
list4[0][0] # Nested indexing - Access the first character of the first list ele
```

Out[48]: 'o'

In [50]:
```
list4[-1] # Last item of the list
```

Out[50]: 'three'

In [52]:
```
list5[-1] # Last item of the list
```

Out[52]: [150, 90]

LIST SLICING

In [55]:
```
mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [57]:
```
mylist[0:3] # Return all items from 0th to 3rd index location excluding the item
```

Out[57]: ['one', 'two', 'three']

In [59]:
```
mylist[2:5] # List all items from 2nd to 5th index location excluding the item a
```

Out[59]: ['three', 'four', 'five']

In [61]:
```
mylist[:3] # Return first three items
```

Out[61]: ['one', 'two', 'three']

In [63]:
```
mylist[:2] # Return first two items
```

Out[63]: ['one', 'two']

In [67]:
```
mylist[-3:] # Return last three items
```

Out[67]: ['six', 'seven', 'eight']

In [71]:
```
mylist[-2:] # Return last two items
```

Out[71]: ['seven', 'eight']

In [73]:
```
mylist[-1] # Return last item of the list
```

Out[73]: 'eight'

In [75]:
```
mylist[:] # Return whole list
```

Out[75]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

Add , Remove & Change Items

In [78]:
```
mylist
```

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

In [82]:
```
mylist.append('nine') # Add an item to the end of the list
mylist
```

Out[82]:
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'nine']
```

In [86]:
```
mylist.insert(9,'ten') # Add item at index location 9
mylist
```

Out[86]:
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'ten',
 'nine']
```

In [88]:
```
mylist.insert(1,'ONE') # Add item at index location 1
mylist
```

Out[88]:
```
['one',
 'ONE',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'ten',
 'nine']
```

In [90]:
```
mylist.remove('ONE') # Remove item "ONE"
mylist
```

Out[90]:
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'ten',
 'nine']
```

In [92]:
```
mylist.pop() # Remove last item of the list
mylist
```

Out[92]:
```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten',
 'ten']
```

In [94]:
```
mylist.pop(8) # Remove item at index location 8
mylist
```

Out[94]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'ten', 'ten']
```

In [96]:
```
del mylist[7] # Remove item at index location 7
mylist
```

Out[96]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'ten', 'ten']
```

In [98]:
```
# Change value of the string
mylist[0] = 1
mylist[1] = 2
mylist[2] = 3
mylist
```

Out[98]:
```
[1, 2, 3, 'four', 'five', 'six', 'seven', 'ten', 'ten']
```

In [100…
```
mylist.clear() # Empty List / Delete all items in the list
mylist
```

Out[100…
```
[]
```

In [102…
```
del mylist # Delete the whole list
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[102], line 2
      1 del mylist
----> 2 mylist

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

COPY LIST

In [119…
```
mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', '
```

In [129…
```
mylist1 = mylist # Create a new reference "mylist1"
```

In [131…
```
id(mylist) , id(mylist1) # The address of both mylist & mylist1 will be the same
```

Out[131… (3004403640192, 3004403640192)

In [133…
```
mylist2 = mylist.copy() # Create a copy of the list
```

In [145…
```
id(mylist2) # The address of mylist2 will be different from mylist because mylis
```

Out[145… 3004403871360

In [137…
```
mylist[0] = 1
```

In [139…
```
mylist
```

Out[139… [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']

In [141…
```
mylist1 # mylist1 will be also impacted as it is pointing to the same list
```

Out[141… [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']

In [147…
```
mylist2 # Copy of list won't be impacted due to changes made on the original list
```

Out[147… [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'ten']

JOINT LISTS

In [151…
```
list1 = ['one', 'two', 'three', 'four']
list2 = ['five', 'six', 'seven', 'eight']
```

In [155…
```
list3 = list1 + list2 # Join two lists by '+' operator
list3
```

Out[155… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [157…
```
list1.extend(list2) #Append list2 with list1
list1
```

Out[157… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

LIST MEMBERSHIP

In [162…
```
list1
```

Out[162… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [164…
```
'one' in list1 # Check if 'one' exist in the list
```

Out[164… True

In [166…
```
'ten' in list1 # Check if 'ten' exist in the list
```

Out[166… False

In [170…
```
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 [172…
```
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 [175…
```
list1
```

Out[175… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [177…
```
list1.reverse() # Reverse the list
list1
```

Out[177… ['eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'one']

In [179…
```
list1 = list1[::-1] # Reverse the list
list1
```

Out[179… ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [181…
```
mylist3 = [9,5,2,99,12,88,34]
mylist3.sort() # Sort list in ascending order
mylist3
```

Out[181… [2, 5, 9, 12, 34, 88, 99]

In [183…
```
mylist3 = [9,5,2,99,12,88,34]
mylist3.sort(reverse=True) # Sort list in descending order
mylist3
```

Out[183… [99, 88, 34, 12, 9, 5, 2]

In [185…
```
mylist4 = [88,65,33,21,11,98]
sorted(mylist4) # Returns a new sorted list and doesn't change original
```

Out[185… [11, 21, 33, 65, 88, 98]

In [187…
```
mylist4
```

Out[187… [88, 65, 33, 21, 11, 98]

LOOP THROUGH A LIST

In [190…
```
list1
```

Out[190…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [192…
```
for i in list1:
    print(i)
```

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

In [194…
```
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 [197…
```
list10 = ['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
```

In [199…
```
list10.count('one') # Number of times item "one" occurred in the list.
```

Out[199…
```
3
```

In [201…
```
list10.count('two') # Occurence of item 'two' in the list
```

Out[201…
```
2
```

In [203…
```
list10.count('four') # Occurence of item 'four' in the list
```

Out[203…
```
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 [207…
```
L1 = [1,2,3,4,0]
```

In [209…
```
all(L1) # Will Return false as one value is false (Value 0)
```

Out[209… False

In [211…
```
any(L1) # Will Return True as we have items in the list with True value
```

Out[211… True

In [213…
```
L2 = [1,2,3,4,True,False]
```

In [215…
```
all(L2) # Returns false as one value is false
```

Out[215… False

In [217…
```
any(L2) # Will Return True as we have items in the list with True value
```

Out[217… True

In [219…
```
L3 = [1,2,3,True]
```

In [221…
```
all(L3) # Will return True as all items in the list are True
```

Out[221… True

In [ ]: