In [3]:
```
txt = "    abc def ghi    "
txt.lstrip()          # lstrip() function is used with strings to remove leading
```

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

In [2]:
```
txt = "    abc def ghi    "
txt.strip()           # strip() function is used to remove characters from both
```

Out[2]:
```
'abc def ghi'
```

Using Escape charecter

In [4]:
```
mystr = "My favourite movie is "Salaar"" ##Using double quotes in the string is
```

```
  Cell In[4], line 1
    mystr = "My favourite movie is "Salaar""
                                    ^
SyntaxError: invalid syntax
```

In [6]:
```
##Using escape character to allow illegal characters
mystr = "My favourite movie is \"Salaar\""
print(mystr)
```

```
My favourite movie is "Salaar"
```

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.

In [9]:
```
list1 = []
print(type(list1))
```

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

In [10]:
```
l2 = [10,20,30]
```

In [11]:
```
l3 = [20.7,40.4,50.6]
```

In [12]:
```
l4 = ['one','two','three']
```

In [13]:
```
l5 = ['prasad',25,[50,25],[70,89]]
```

In [14]:
```
l6 = [10,'prasad',58.9]
```

In [15]:
```
l7 = ['prasad',25,[50,25],[70,89],{'johny','david'}]
```

In [16]:
```
len(l5)
```

Out[16]:
```
4
```

In [17]:
```
len(l6)
```

Out[17]: 3

List Indexing

In [18]:
```
l2[0]
```

Out[18]: 10

In [19]:
```
l4[1]
```

Out[19]: 'two'

In [20]:
```
l4[0][0]
```

Out[20]: 'o'

In [21]:
```
l4[-1]
```

Out[21]: 'three'

List Slicing

In [22]:
```
mylist = [10,20,30,40,50,60,70,80]
```

In [23]:
```
mylist[0:3]
```

Out[23]: [10, 20, 30]

In [24]:
```
mylist[2:5]
```

Out[24]: [30, 40, 50]

In [25]:
```
mylist[:4]
```

Out[25]: [10, 20, 30, 40]

In [26]:
```
mylist[:-1]
```

Out[26]: [10, 20, 30, 40, 50, 60, 70]

In [27]:
```
mylist[-2:]
```

Out[27]: [70, 80]

In [28]:
```
mylist[-1]
```

Out[28]: 80

In [29]:
```
mylist[:]
```

Out[29]: [10, 20, 30, 40, 50, 60, 70, 80]

List Functions

In [30]:
```
mylist
```

Out[30]: [10, 20, 30, 40, 50, 60, 70, 80]

In [31]:
```
mylist.append(90) # Add an item to the end of the list
mylist
```

Out[31]: [10, 20, 30, 40, 50, 60, 70, 80, 90]

In [32]:
```
mylist.insert(6,65) # Add an item to the index Location 6
mylist
```

Out[32]: [10, 20, 30, 40, 50, 60, 65, 70, 80, 90]

In [33]:
```
mylist.remove(65)  # Remove item 65
mylist
```

Out[33]: [10, 20, 30, 40, 50, 60, 70, 80, 90]

In [34]:
```
mylist.pop()  # Remove last item of the list
mylist
```

Out[34]: [10, 20, 30, 40, 50, 60, 70, 80]

In [40]:
```
mylist.pop(2)  # Remove item at index location 2
mylist
```

Out[40]: [10, 20, 40, 50, 60]

In [41]:
```
mylist
```

Out[41]: [10, 20, 40, 50, 60]

In [43]:
```
del mylist[2]  # Remove item at index location 2
mylist
```

Out[43]: [10, 20, 50, 60]

In [44]:
```
mylist[0] = 1
mylist[1] = 2
mylist
```

Out[44]: [1, 2, 50, 60]

In [45]:
```
mylist.clear()
mylist
```

Out[45]: []

In [46]:
```
del mylist
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[46], line 2
      1 del mylist
----> 2 mylist

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

Copy List

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

In [48]:
```
mylist1 = mylist
mylist1
```

Out[48]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [49]:
```
id(mylist1), id(mylist) # The Adress of both lists will be same
```

Out[49]:
```
(1648287999616, 1648287999616)
```

In [50]:
```
mylist2 = mylist.copy() # create a copy of the list
```

In [51]:
```
id(mylist2)
```

Out[51]:
```
1648288011200
```

In [52]:
```
mylist[0] = 1
mylist
```

Out[52]:
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [53]:
```
mylist2
```

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

Join Lists

In [54]:
```
l1 = ['one','two','three']
l2 = ['four','five','six']
```

In [55]:
```
l3 = l1+l2
l3
```

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

In [56]:
```
l1.extend(l2)  # Apppend l2 with l1
l1
```

Out[56]:
```
['one', 'two', 'three', 'four', 'five', 'six']
```

List Membership

In [57]:
```
l1
```

Out[57]: ['one', 'two', 'three', 'four', 'five', 'six']

In [58]:
```
'one' in l1
```

Out[58]: True

In [59]:
```
'ten' in l1
```

Out[59]: False

In [60]:
```
if 'three' in l1:   # check if three is present in the list
    print('Three is present in the list')
else:
    print('There is no Three in list')
```

Three is present in the list

In [61]:
```
if 'ten' in l1:
    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 [62]:
```
l1
```

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

In [65]:
```
l1.reverse()
l1
```

Out[65]: ['six', 'five', 'four', 'three', 'two', 'one']

In [68]:
```
l1 = l1[::-1]
l1
```

Out[68]: ['one', 'two', 'three', 'four', 'five', 'six']

In [71]:
```
l3 = [50,20,56,10,5]
l3.sort()
l3
```

Out[71]: [5, 10, 20, 50, 56]

In [72]:
```
l3
```

Out[72]: [5, 10, 20, 50, 56]

In [73]:
```
l3.sort(reverse=True)
l3
```

Out[73]: [56, 50, 20, 10, 5]

Loop Through a List

In [74]:
```
l1
```

Out[74]: ['one', 'two', 'three', 'four', 'five', 'six']

In [75]:
```
for i in l1:
    print(i)
```

```
one
two
three
four
five
six
```

In [76]:
```
for i in enumerate(l1):  # It adds a counter (index) to each element of the iter
```

```
(0, 'one')
(1, 'two')
(2, 'three')
(3, 'four')
(4, 'five')
(5, 'six')
```

In [77]:
```
l1.count('one') # counts how many times an item repeats in list
```

Out[77]: 1

In [78]:
```
l1.append('four')
```

In [79]:
```
l1
```

Out[79]: ['one', 'two', 'three', 'four', 'five', 'six', 'four']

In [82]:
```
l1.count('four')
```

Out[82]: 2

All / Any

In [83]:
```
l4 = [1,2,3,4,0]
```

In [87]:
```
all(l4)
```

Out[87]: False

In [88]:
```
any(l4)
```

Out[88]: True

In [89]:
```
l5 = [1,2,3,4,True,False]
```

In [90]:
```
all(l5)
```

Out[90]: False

In [91]:
```
any(l5)
```

Out[91]: True

In [92]:
```
l6 = [1,2,3,4,True]
```

In [93]:
```
all(l6)
```

Out[93]: True

In [ ]: