lstrip and strip functions

In [136…
```
txt = "  abc def ghi "
txt.lstrip()

# txt.lstrip()` removes whitespace characters from the beginning (left side) of a s
```

Out[136…
```
'abc def ghi '
```

In [137…
```
txt = "   abc def ghi  "
txt.strip()

#`txt.strip()` removes whitespace characters from both the beginning and end of a s
```

Out[137…
```
'abc def ghi'
```

In [ ]:

Using escape character

In [138…
```
# using double quotes in the string isnot allowed

mystr = "My favorite TV series is "Game of Thrones""
```

```
  Cell In[138], line 3
    mystr = "My favorite TV series is "Game of Thrones""
                                      ^
SyntaxError: invalid syntax
```

In [ ]:

In [ ]:
```
# using escape character( \ ) to allow illegal characters

mystr = "My favorite TV series is \"Game of Thrones\""
print(mystr)
```

In [ ]:

List

In [ ]:
```
list1 = []  # empty list
```

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

In [ ]:
```
```

In [ ]:
```
list2 = [10,30,60]   #integer list
```

In [ ]:
```
list3 = [10.77,30.56,59.41]  # list of float
```

In [ ]:
```
list4 = ['one', 'two', 'three']   #list of strings
```

In [ ]:
```
list5 = ['Nit', 25, [50,100],[40,50,60]]

# NESTED LISTS
```

In [ ]:
```
list6 = [100,'nit',74.56]

#List  of mixed datatypes
```

In [ ]:
```
list7 = ['nit',25,[50,150],[10,20,30],{'John','David'}]
#Nested list
```

In [ ]:
```
len(list7)
```

In [ ]:
```
```

List indexing

In [ ]:
```
list2[0]
```

In [ ]:
```
list6[1]
```

In [ ]:
```
list7[3][2]  # Nested list indexing
```

In [ ]:
```
list5[-1]
```

In [ ]:
```
```

List slicing

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

In [ ]:
```
mylist[:3]
```

In [ ]:
```
mylist[5:]
```

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

In [ ]:
```
mylist[:-3]
```

In [ ]:
```
mylist[-6:]
```

In [ ]:
```
mylist[-5]
```

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

In [ ]:
```

```

Add, Remove & changes to list

In [ ]:
```
mylist
```

In [ ]:
```
mylist.append('nine')

# appends/adds 'nine' at the end of the list
```

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

In [ ]:
```
mylist.insert(9,'ten')
print(mylist)
mylist
```

In [ ]:
```
mylist.insert(1,'ONE')
```

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

In [ ]:
```
mylist.remove('ONE')
mylist
```

In [ ]:
```
mylist.pop()

# Removes last item
```

In [ ]:
```
mylist
```

In [ ]:
```
mylist.pop(4)
#removes 4th index item
```

In [ ]:
```
mylist
```

In [ ]:
```
mylist.pop(7)
mylist
```

In [ ]:
```
#changes values of string
mylist[0]=1
mylist[1]=2
```

```
mylist[2]=3
mylist
```

In [ ]:
```
mylist.clear()    # Remove all items from list

mylist
```

In [ ]:
```
del mylist  #Deletes the whole list
mylist
```

In [ ]:
```

```

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

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

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

print(id(mylist1))

#Address will be same for both the list as mylist1 is new refernece and NOT A COPY
```

In [ ]:
```
mylist2=mylist.copy()

# creates a copy of list and NOT a refernce list
```

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

print(id(mylist2))

#different address as its a copy list
```

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

In [ ]:
```
mylist1

# reference list is also affected by the changes in the original list
```

In [ ]:
```
mylist2

#copy list is not affected by the chnages in the origina list
```

In [ ]:
```

```

Join lists

In [139…
```
list1 = ['one','two','three','four']
list2 = ['five','six','seven']
```

```
list3 = ['eight','nine']
```

In [140…
```
list4 = list1 + list2  #joins lists by '+' operator
list4
```

Out[140…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

In [141…
```
list1.extend(list2)
list1

#appends list2 to list1 at the end
```

Out[141…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

In [ ]:
```

```

List Membership

In [142…
```
list1
```

Out[142…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

In [143…
```
'one' in list1   # Check if 'one' exist in the list
```

Out[143…
```
True
```

In [144…
```
'four' in list1
```

Out[144…
```
True
```

In [145…
```
'ten' in list1
```

Out[145…
```
False
```

In [146…
```
if 'three' in list1:
    print('Three is present in the list1')
else:
    print('Three in not present in the list1')
```

```
Three is present in the list1
```

In [149…
```
if 'Three' in list1:
    print('Three is present in the list1')
else:
    print('Three in not present in the list1')

#strings are case sensitive
```

```
Three in not present in the list1
```

In [150…
```
if 'sixty' in list1:
    print('Sixty is present in the list1')
```

```
else:
    print('Sixty in not present in the list1')
```

```
Sixty in not present in the list1
```

In [ ]:

Reverse and Sort list

In [151…
```
list1
```

Out[151…
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven']
```

In [154…
```
list1.reverse()        # Reverses the list(index places)
list1
```

Out[154…
```
['seven', 'six', 'five', 'four', 'three', 'two', 'one']
```

In [157…
```
list1 = list1[::-1]    # Reverses the list(index places)
list1
```

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

In [158…
```
list1[::-1]
```

Out[158…
```
['seven', 'six', 'five', 'four', 'three', 'two', 'one']
```

In [ ]:

In [160…
```
mylist3 = [5,8,63,78,15,42,1]
mylist3
```

Out[160…
```
[5, 8, 63, 78, 15, 42, 1]
```

In [168…
```
mylist3.sort()    # Ascending order sort
mylist3
```

Out[168…
```
[1, 5, 8, 15, 42, 63, 78]
```

In [170…
```
mylist3.sort(reverse=True)   # Reverse of ascending order => Descending order
mylist3
```

Out[170…
```
[78, 63, 42, 15, 8, 5, 1]
```

In [175…
```
mylist3    # After sort function,  Order of list changes in the origina list
```

Out[175…
```
[78, 63, 42, 15, 8, 5, 1]
```

In [176…
```
mylist4 = [4,85,32,19,0,49]
print(mylist4)
print(sorted(mylist4))
```

```
# Returns with a sorted list but doesnot change the order in the original list
```

```
[4, 85, 32, 19, 0, 49]
[0, 4, 19, 32, 49, 85]
```

In [ ]:

Loop through a list

In [177…
```
list1
```

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

In [181…
```
for i in list1:
    print(i)

#This starts a for loop that will iterate through each element in `list1`, assignin
```

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

In [184…
```
enumerate(list1)
```

Out[184…
```
<enumerate at 0x268dbfea840>
```

In [185…
```
for i in enumerate(list1):
    print(i)

# This creates a loop where `i` will be a tuple containing the index and the corres
```

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

In [ ]:

Count

In [192…
```
lst1 = ['one','two','three','four','one','one','three','two']
```

In [194…
```
lst1.count('one')   # number of 'one' repetitions in the list
```

Out[194… 3

In [195…
```
lst1.count('two')
```

Out[195… 2

In [196…
```
lst1.count('five')
```

Out[196… 0

In [ ]:

All - Any method

In [201…
```
L1 = [1,2,3,4,0]
```

In [199…
```
all(L1)

# for all() function:
# True - If all elements in a list are true(1 or True)
# False - If any element in a list is false(0 or False)
```

Out[199… False

In [205…
```
any(L1)

# True if any element in the list is True(1 or True)
# If not, any() returns False
```

Out[205… True

In [ ]:

In [203…
```
L2 = [1,2,3,4,5,True, False]
```

In [204…
```
all(L2)     # False in the list
```

Out[204… False

In [206…
```
any(L2)     # True and 1 in the list
```

Out[206… True

In [ ]:

In [207…
```
L3 = [ 1,2,3, True]
```

In [208…
```
any(L3)     # No 0 or False in the list
```

Out[208… True

In [209…
```
all(L3)     # No 0 or False in the list
```

Out[209… True

In [ ]:
```

```