LIST--

A Datastructure in Python. It is a sequence of ordered items.

A List can contain different datatypes under it. Like int, float, string ..etc

```
In [1]: list1=[]
        list

Out[1]: list
```

```
In [3]: list1

Out[3]: []
```

```
In [5]: type(list1)

Out[5]: list
```

```
In [7]: print(type(list1))

<class 'list'>
```

```
In [46]: list2 = [10,20,30,40] #list of integers
```

```
In [32]: list3 = [10.2,23.4,56.7,9.8] #list of float numbers
```

```
In [34]: list4 = ['one','two',"three"] #list of strings
```

```
In [36]: list5 = [[1,2],[2,3],'Hari',23] #Nested Lists
```

```
In [38]: list6 = [2,'Hari',2.3] #List of mixed datatypes
```

```
In [24]: len(list6) #length of list

Out[24]: 3
```

LIST INDEXING

```
In [40]: list2[0] #retreive first element of the list

Out[40]: 10
```

```
In [50]: list2[1]

Out[50]: 20
```

```
In [52]: list6[1]
```

```
Out[52]:  'Hari'
```

```
In [54]:  list4[2]
```

```
Out[54]:  'three'
```

```
In [56]:  list2[4]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[56], line 1
----> 1 list2[4]

IndexError: list index out of range
```

```
In [58]:  list5[0][0] #Nested indexing -- Access the first character of the first element of
```

```
Out[58]:  1
```

```
In [60]:  list6[2][1]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[60], line 1
----> 1 list6[2][1]

TypeError: 'float' object is not subscriptable
```

```
In [62]:  list6[1][1]
```

```
Out[62]:  'a'
```

```
In [64]:  list5[1][1]
```

```
Out[64]:  3
```

```
In [66]:  list4[-1] #last item of the list
```

```
Out[66]:  'three'
```

```
In [68]:  list6[-1]
```

```
Out[68]:  2.3
```

LIST SLICING

```
In [72]:  mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [74]:  mylist[0:2] #prints from 0 to n-1=2-1=1 so 0 and 1
```

```
Out[74]:  ['one', 'two']
```

```
In [76]:  mylist[2:5] #2 to 5-1=4 so 2,3,4

Out[76]:  ['three', 'four', 'five']

In [78]:  mylist[:3] #0 to 3-1=2 so 0,1,2 #returns first three elements

Out[78]:  ['one', 'two', 'three']

In [80]:  mylist[:2] #o to 2-1=1 so 0,1

Out[80]:  ['one', 'two']

In [82]:  mylist[-3:] #-3 to 0-1 so -3 to -1 so -3,-2,-1 #returns last 3 items

Out[82]:  ['six', 'seven', 'eight']

In [84]:  mylist[-2:] #-2 to 0-1 so -2 to -1 so -2,-1

Out[84]:  ['seven', 'eight']

In [86]:  mylist[-1]

Out[86]:  'eight'

In [88]:  mylist[:] #0 to 0-1 so 0 to -1 #retuen whole list

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

ADD, REMOVE & CHANGE ITEMS

```
In [101…  mylist

Out[101…  ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [115…  mylist.append(2)

In [117…  mylist
```

```
Out[117…  ['one',
           'two',
           'three',
           'four',
           'five',
           'six',
           'seven',
           'eight',
           2,
           2,
           'nine',
           'nine',
           'nine',
           2]
```

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

```
In [121…  mylist
```

```
Out[121…  ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

```
In [123…  mylist.append(1)
```

```
In [125…  mylist
```

```
Out[125…  ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 1]
```

```
In [127…  mylist.append(nine)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[127], line 1
----> 1 mylist.append(nine)

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

```
In [129…  mylist.append('nine')
```

```
In [131…  mylist
```

```
Out[131…  ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 1, 'nine']
```

```
In [133…  mylist.append(9,23)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[133], line 1
----> 1 mylist.append(9,23)

TypeError: list.append() takes exactly one argument (2 given)
```

```
In [135…  mylist.append(9,'ten')
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[135], line 1
----> 1 mylist.append(9,      )

TypeError: list.append() takes exactly one argument (2 given)
```

In [137…

```
mylist.append(10) #append function -- adds item to the last
```

In [139…

```
mylist
```

Out[139…

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 1, 'nine', 10]
```

In [141…

```
mylist.insert(9,'ten') #add item at index ocation 9
```

In [143…

```
mylist
```

Out[143…

```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 1,
 'ten',
 'nine',
 10]
```

In [145…

```
mylist.insert(9,23)
```

In [147…

```
mylist
```

Out[147…

```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 1,
 23,
 'ten',
 'nine',
 10]
```

In [149…

```
mylist.insert(10,'eleven')
```

In [151…

```
mylist
```

Out[151…

```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 1,
 23,
 'eleven',
 'ten',
 'nine',
 10]
```

In [153…

```
mylist.insert(0,1)
```

In [155…

```
mylist
```

Out[155…

```
[1,
 'one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 1,
 23,
 'eleven',
 'ten',
 'nine',
 10]
```

In [157…

```
mylist.remove(1) #removes that particular item
```

In [159…

```
mylist
```

Out[159…

```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 1,
 23,
 'eleven',
 'ten',
 'nine',
 10]
```

In [161…

```
mylist.remove(1)
```

In [163...

```
mylist
```

Out[163...

```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 23,
 'eleven',
 'ten',
 'nine',
 10]
```

In [165...

```
mylist.remove('eleven')
```

In [167...

```
mylist
```

Out[167...

```
['one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 23,
 'ten',
 'nine',
 10]
```

In [169...

```
mylist.pop()
```

Out[169...

```
10
```

In [171...

```
mylist.pop()
mylist
```

Out[171...

```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 23, 'ten']
```

In [173...

```
mylist.pop(8)
```

Out[173...

```
23
```

In [175...

```
mylist.pop(8)
mylist
```

Out[175...

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

In [177...

```
mylist.pop(1)
mylist
```

Out[177…

```
['one', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

pop() -- by default removes last item of the list

pop(8) -- removes item at 8th index

In [180…

```
del mylist[7]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[180], line 1
----> 1 del mylist[7]

IndexError: list assignment index out of range
```

In [182…

```
del mylist[6] ##delete/remove item at index lcation 7
```

In [184…

```
mylist
```

Out[184…

```
['one', 'three', 'four', 'five', 'six', 'seven']
```

Changing value of the string

In [187…

```
mylist[0] = 1
mylist[1] = 2
mylist[2] = 3
mylist
```

Out[187…

```
[1, 2, 3, 'five', 'six', 'seven']
```

In [189…

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

In [191…

```
mylist
```

Out[191…

```
[]
```

In [193…

```
del mylist # Delete the whole list
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[193], line 2
      1 del mylist # Delete the whole list
----> 2 mylist

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

COPY LIST

In [196…

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

In [198…

```
mylist1 = mylist # Create a new reference "mylist1"
```

In [200…

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

Out[200…

```
(2979208403904, 2979208403904)
```

In [202…

```
mylist2=mylist.copy() #create a copy of that list
```

In [206…

```
id(mylist2)  # The address of mylist2 will be different from mylist because mylist
```

Out[206…

```
2979208443904
```

In [208…

```
mylist[0]=1
mylist
```

Out[208…

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

In [210…

```
mylist
```

Out[210…

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

In [212…

```
mylist1 # mylist1 will be also impacted as it is pointing to the same list
```

Out[212…

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

In [214…

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

Out[214…

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

Join Lists

In [217…

```
list1=[1,2,3,'four']   # Join two lists by '+' operator
list2=[5,6,'seven']
list3=list1+list2
```

In [219…

```
list3
```

Out[219…

```
[1, 2, 3, 'four', 5, 6, 'seven']
```

In [221…

```
list1.extend(list2)  #Append list2 with list1
```

In [223…

```
list1
```

Out[223…

```
[1, 2, 3, 'four', 5, 6, 'seven']
```

In [225…

```
list2
```

Out[225…

```
[5, 6, 'seven']
```

List Membership

In [228...

```
list1
```

Out[228...

```
[1, 2, 3, 'four', 5, 6, 'seven']
```

In [230...

```
1 in list1
```

Out[230...

```
True
```

In [232...

```
'four' in list1
```

Out[232...

```
True
```

In [234...

```
8 in list1
```

Out[234...

```
False
```

In [242...

```
if 3 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 [246...

```
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 [249...

```
list1
```

Out[249...

```
[1, 2, 3, 'four', 5, 6, 'seven']
```

In [251...

```
list1.reverse() # Reverse the list
list1
```

Out[251...

```
['seven', 6, 5, 'four', 3, 2, 1]
```

In [253...

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

Out[253...

```
[1, 2, 3, 'four', 5, 6, 'seven']
```

In [255...

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

Out[255...

```
[2, 5, 9, 12, 34, 88, 99]
```

In [257…

```
mylist=[3,5,100,45]
mylist.sort()
mylist
```

Out[257…

```
[3, 5, 45, 100]
```

In [259…

```
mylist.sort(reverse=True)
```

In [261…

```
mylist
```

Out[261…

```
[100, 45, 5, 3]
```

In [263…

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

Out[263…

```
[11, 21, 33, 65, 88, 98]
```

In [265…

```
mylist4
```

Out[265…

```
[88, 65, 33, 21, 11, 98]
```

Count

In [268…

```
list10 =['one', 'two', 'three', 'four', 'one', 'one', 'two', 'three']
list10.count('one')  # Number of times item "one" occurred in the list.
```

Out[268…

```
3
```

In [270…

```
list10.count('three')
```

Out[270…

```
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() returns False.

In [274…

```
L1 = [1,2,3,4,0]
all(L1) # Will Return false as one value is false (Value 0)
```

Out[274…

```
False
```

In [276…

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

Out[276…

```
True
```

In [278…

```
L2 = [1,2,3,4,True,False]
all(L2)
```

Out[278…

```
False
```

In [280…

```
any(L2)
```

Out[280…

```
True
```

In [282…

```
L3 = [1,2,3,True]
all(L3)
```

Out[282…

```
True
```

In [ ]: