In [2]:
```
l = []
l
```

Out[2]:
```
[]
```

In [4]:
```
l1 = list()
l1
```

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

In [5]:
```
len(l)
```

Out[5]:
```
0
```

In [6]:
```
len(l1)
```

Out[6]:
```
0
```

In [9]:
```
l.append(10)
l.append(20)
l.append(50)
```

In [10]:
```
l
```

Out[10]:
```
[10, 10, 20, 50, 10, 20, 50]
```

In [11]:
```
l.remove(10,20,50)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[11], line 1
----> 1 l.remove(10,20,50)

TypeError: list.remove() takes exactly one argument (3 given)
```

In [12]:
```
l.remove(10)
```

In [13]:
```
l
```

Out[13]:
```
[10, 20, 50, 10, 20, 50]
```

In [14]:
```
l.remove(10)
```

In [15]:
```
l.remove(20)
```

In [16]:
```
l.remove(50)
```

In [17]:
```
l
```

Out[17]:
```
[10, 20, 50]
```

In [19]:
```
l.append('nit')
```

In [20]:
```
l.append('5.5')
```

In [21]:
```
l
```

Out[21]:
```
[10, 20, 50, 'nit', '5.5']
```

In [22]:
```
l.pop(4)
```

Out[22]:
```
'5.5'
```

In [23]:
```
l
```

Out[23]:
```
[10, 20, 50, 'nit']
```

In [24]:
```
l.append(70)
```

In [25]:
```
l
```

Out[25]:
```
[10, 20, 50, 'nit', 70]
```

In [26]:
```
print(len(l))
```

```
5
```

In [37]:
```
l1=l.copy()
```

In [38]:
```
l1
```

Out[38]:
```
[10, 20, 50, 'nit', 70]
```

In [39]:
```
l1
```

Out[39]:
```
[10, 20, 50, 'nit', 70]
```

In [40]:
```
l1.append(100)
```

In [ ]:
```
l2 = []
```

In [32]:
```
l2
```

Out[32]:
```
[]
```

In [34]:
```
l2.append(2)
l2.append(3)
l2.append(4)
```

In [35]:
```
l2.append([5,6,7])
```

In [36]:
```
l2
```

Out[36]:
```
[2, 2, 3, 4, [5, 6, 7]]
```

In [41]:
```
l1 == l2
```

Out[41]:
```
False
```

In [42]:
```
l3 = l2.copy()
```

In [43]:
```
l2 == l3
```

Out[43]:
```
True
```

In [44]:
```
id(l2) == id(l3)
```

Out[44]:
```
False
```

In [45]:
```
l3.clear()
```

In [46]:
```
del l3
```

In [47]:
```
l3
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[47], line 1
----> 1 l3

NameError: name 'l3' is not defined
```

In [48]:
```
print(l)
```

```
[10, 20, 50, 'nit', 70]
```

In [49]:
```
print(l1)
```

```
[10, 20, 50, 'nit', 70, 100]
```

In [50]:
```
print(l2)
```

```
[2, 2, 3, 4, [5, 6, 7]]
```

In [51]:
```
l1[1:8]
```

Out[51]:
```
[20, 50, 'nit', 70, 100]
```

In [52]:
```
l1[0:7:2]
```

Out[52]:
```
[10, 50, 70]
```

In [53]:
```
l1[::-1]
```

Out[53]:
```
[100, 70, 'nit', 50, 20, 10]
```

In [54]:
```
l2[::-1]
```

Out[54]:
```
[[5, 6, 7], 4, 3, 2, 2]
```

In [55]:
```
l1[::-3]
```

Out[55]:
```
[100, 50]
```

In [56]:
```
print(l)
print(l1)
print(l2)
```

```
[10, 20, 50, 'nit', 70]
[10, 20, 50, 'nit', 70, 100]
[2, 2, 3, 4, [5, 6, 7]]
```

In [57]:
```
l3 = ['red','green','blue']
l3
```

Out[57]:
```
['red', 'green', 'blue']
```

In [58]:
```
l3[::-1]
```

Out[58]:
```
['blue', 'green', 'red']
```

In [59]:
```
l.count(10)
```

Out[59]:
```
1
```

In [60]:
```
l2.count(2)
```

Out[60]:
```
2
```

In [61]:
```
l1.remove(10)
```

In [62]:
```
l1.index('nit')
```

Out[62]:
```
2
```

In [63]:
```
l2.index([5,6,7])
```

Out[63]:
```
4
```

In [65]:
```
print(l[3][1])
```

```
i
```

In [66]:
```
print(l[3][0])
print(l[3][1])
print(l[3][2])
```

```
n
i
t
```

In [67]:
```
print(l[3][0])
print(l[3][1])
print(l[3][2])
print(l[3][3])
```

```
n
i
t
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[67], line 4
      2 print(l[3][1])
      3 print(l[3][2])
----> 4 print(l[3][3])

IndexError: string index out of range
```

In [68]:
```
#To Avoid above situation use Try Catch and Final method
```

In [70]:
```
x = [1,2,3]
y = x.copy()
x.append(4)
print(y)
```

```
[1, 2, 3]
```

In [71]:
```
l3 = [10,20,30] #Integer list
```

In [72]:
```
l4= [10.77,33.22,60.66,89] #list with float values
l5 = ['one','Two','Three'] # String list
l6 = ['Asif',25,[50,100],[150,90]] #Nested List
```

In [73]:
```
len(l6)
```

Out[73]:
```
4
```

In [74]:
```
l4[0] #indexing
```

Out[74]:
```
10.77
```

In [75]:
```
l5[0][0] #Nested Indexing
```

Out[75]:
```
'o'
```

In [76]:
```
l4[-1]
```

Out[76]:
```
89
```

In [77]:
```
l4[::-1]
```

Out[77]:
```
[89, 60.66, 33.22, 10.77]
```

In [78]:
```
l6[-1]
```

Out[78]:
```
[150, 90]
```

In [79]:
```
#Slicing the list
```

In [80]:
```
mylist= ['one','two','three','four','five','six','seven','eight']
```

In [81]:
```
mylist[0:3]
```

Out[81]:
```
['one', 'two', 'three']
```

In [83]:
```
mylist[2:5]
```

Out[83]:
```
['three', 'four', 'five']
```

In [84]:
```
mylist[:3]
```

Out[84]:
```
['one', 'two', 'three']
```

In [85]:
```
mylist[:2]
```

Out[85]:
```
['one', 'two']
```

In [86]:
```
mylist[:-3]
```

Out[86]:
```
['one', 'two', 'three', 'four', 'five']
```

In [87]:
```
mylist[-3:]
```

Out[87]:
```
['six', 'seven', 'eight']
```

In [88]:
```
mylist[3:]
```

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

In [89]:
```
mylist[:3]
```

Out[89]:
```
['one', 'two', 'three']
```

In [90]:
```
mylist[-2:]
```

Out[90]:
```
['seven', 'eight']
```

In [91]:
```
mylist[-1]
```

Out[91]:
```
'eight'
```

In [92]:
```
mylist[:]
```

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

Add,Remove & Change Items

In [93]:
```
mylist
```

Out[93]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [94]:
```
mylist.append('nine') # Add an item at the end of the list
mylist
```

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

In [95]:
```
mylist.insert(0,'zero') #Inserts at a particular location based on index
mylist
```

Out[95]:
```
['zero',
 'one',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine']
```

In [96]:
```
mylist.remove('one') #Remove item "one"
mylist
```

Out[96]:
```
['zero', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [97]:
```
mylist.pop() #Remove Last item of the list
mylist
```

Out[97]:
```
['zero', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [99]:
```
mylist.pop(6) #Remove the item at index location 6
mylist
```

Out[99]:
```
['zero', 'two', 'three', 'four', 'five', 'six', 'eight']
```

In [101…
```
del mylist[2] #removes the value based on index
mylist
```

Out[101]:
```
['zero', 'two', 'four', 'five', 'six', 'eight']
```

In [104…
```
#Change the value of string
mylist[0] = 1
mylist[1]= 2
mylist[2] = 3
mylist
```

Out[104]:
```
[1, 2, 3, 'five', 'six', 'eight']
```

In [105…
```
mylist.clear() #Empty list/Delete all items in the list
mylist
```

Out[105]:
```
[]
```

In [106…
```
del mylist #delete the whole list
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[106], line 2
      1 del mylist #delete the whole list
----> 2 mylist

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

Copy list

In [107…
```
mylist = ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nin
mylist
```

Out[107]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'Te
n']
```

In [108…
```
l9 = mylist.copy()
```

In [109…
```
l9
```

Out[109]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'Te
n']
```

In [110…
```
l10 = l9 #the address of l9 and l10 are same
```

In [111…
```
l10
```

Out[111]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'Te
n']
```

In [112…
```
id(l9) == id(l10)
```

Out[112]:
```
True
```

In [113…
```
id(mylist)==id(l9)
```

Out[113]:
```
False
```

In [116…
```
l9[0] = 1
l9
```

Out[116]:
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'Ten']
```

In [117…
```
l10 #both l9 and l10 are impacted.
```

Out[117]:
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'Ten']
```

Join Lists

In [118…
```
l11 = [1,2,3,4]
l12 = ['five', 'six', 'seven', 'eight']
```

In [119…
```
l13 = l11+l12 # Two lists joined by + operator
```

In [120…
```
l13
```

Out[120]:
```
[1, 2, 3, 4, 'five', 'six', 'seven', 'eight']
```

In [122…
```
l11.extend(l12) # Append list12 with list11
l11
```

Out[122]:
```
[1, 2, 3, 4, 'five', 'six', 'seven', 'eight']
```

List Memebership

In [124…
```
l9
```

Out[124]:
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'Ten']
```

In [125…
```
l9[0]='one'
```

In [126…
```
l9
```

Out[126]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'Te
n']
```

In [127…
```
'one' in l9 # check if 'one' exist in the list
```

Out[127]:
```
True
```

In [128…
```
'ten' in l9
```

Out[128]:
```
False
```

In [129…
```
if 'three' in l9:
    print('Three is in the l9')
else:
    print('Three is not in the l9')
```

```
Three is in the l9
```

In [130…
```
if 'Twelve' in l9:
    print('Twelve is in the l9')
else:
    print('Twelve is not in l9')
```

```
Twelve is not in l9
```

Reverse and Sort list

In [131…
```
l9
```

Out[131]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'Te
n']
```

In [136…
```
l15= l9.reverse() # Reverse the list # Not reversing the list
```

In [137…
```
l15
```

In [138…
```
l16 = l9[::-1]
l16
```

Out[138]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'Te
n']
```

In [139…
```
print(l9[::-1])# List is not getting reversed
```

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

In [140…
```
l17 = [1,1,0,8,2,0,2,4,1,4,1]
```

In [142…
```
l17.sort() # Sort list in descending order
l17
```

Out[142]:
```
[0, 0, 1, 1, 1, 1, 2, 2, 4, 4, 8]
```

In [143…
```
l17.sort(reverse=True) # Descending sort
l17
```

Out[143]:
```
[8, 4, 4, 2, 2, 1, 1, 1, 1, 0, 0]
```

In [144…
```
sorted(l17)
```

Out[144]:
```
[0, 0, 1, 1, 1, 1, 2, 2, 4, 4, 8]
```

In [145…
```
l17
```

Out[145]:
```
[8, 4, 4, 2, 2, 1, 1, 1, 1, 0, 0]
```

Loop through a list

In [146…
```
l9
```

Out[146]:
```
['Ten', 'nine', 'eight', 'seven', 'six', 'five', 'four', 'three', 'two', 'on
e']
```

In [147…
```
for i in l9:
    print(i)
```

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

In [148…
```
for i in enumerate(l9):
    print(i)
```

```
(0, 'Ten')
(1, 'nine')
(2, 'eight')
(3, 'seven')
(4, 'six')
(5, 'five')
(6, 'four')
(7, 'three')
(8, 'two')
(9, 'one')
```

Count

In [150…
```
l17
```

Out[150]:
```
[8, 4, 4, 2, 2, 1, 1, 1, 1, 0, 0]
```

In [151…
```
l17.count(1)
```

Out[151]:
```
4
```

In [153…
```
l17.count(8)
```

Out[153]:
```
1
```

All/Any

The All() method returns True - if all elements in a list are true False- if one of the element in a list is false

In [154…
```
l17
```

Out[154]:
```
[8, 4, 4, 2, 2, 1, 1, 1, 1, 0, 0]
```

In [155…
```
all(l17) # Will return false as one value is false(Value 0)
```

Out[155]:
```
False
```

In [156…
```
any(l17) # Will return True as we have items in the list with True value
```

Out[156]:
```
True
```

In [160…
```
L20 = [12,2.3,300,True,False]
```

In [161…
```
all(L20)
```

Out[161]:
```
False
```

In [162…
```
any(L20)
```

Out[162]:
```
True
```

In [ ]:
```

```