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 [1]:
```
list1 = []  #empty list
```

In [2]:
```
print(type(list1))
```

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

In [3]:
```
list2 = [10,30,60] # list of integers
```

In [4]:
```
list3 = [10.77 , 30.66, 89.90]  #list of float numbers
```

In [5]:
```
list4 = ['one' , 'two', 'three'] #list of strings
```

In [6]:
```
list5 = ['Asif', 25 ,[50,100],[150,90]]  # Nested lists
```

In [7]:
```
list6 = [100, 'RAM',45.67] # list of mixed data types
```

In [8]:
```
list7 = ['Asif', 25 ,[50,100],[150,90],{'john',"kiran"}]
```

In [9]:
```
len(list6)  #Length of list
```

Out[9]:
```
3
```

list indexing 1. forward indexing 2. backward indexing

In [10]:
```
list2[0] #Retrieve first element of the list
```

Out[10]:
```
10
```

In [11]:
```
list4[0] #Retrieve first element of the list
```

Out[11]:
```
'one'
```

In [12]:
```
list4[0][0]  # Nested indexing - Access the first character of the first list
```

Out[12]:
```
'o'
```

In [13]:
```
list4[-1] #last item of the list
```

Out[13]:
```
'three'
```

In [14]:
```
list5[-1]  #last item of the list
```

Out[14]:
```
[150, 90]
```

list slicing

In [15]:
```
mylist =['one','two', 'three','four','five','six','seven','eight']
```

In [16]:
```
mylist[0:3]     # return all items from 0th to 3rd index location excluding th
```

Out[16]:
```
['one', 'two', 'three']
```

In [17]:
```
mylist[2:5]  # return all items from 2nd to 5th index location excluding the i
```

Out[17]:
```
['three', 'four', 'five']
```

In [18]:
```
mylist[:3]  # return first three items
```

Out[18]:
```
['one', 'two', 'three']
```

In [19]:
```
mylist[:2] # return first two items
```

Out[19]:
```
['one', 'two']
```

In [20]:
```
mylist[-3:] # return last two items
```

Out[20]:
```
['six', 'seven', 'eight']
```

In [21]:
```
mylist[-2:] # return last two items
```

Out[21]:
```
['seven', 'eight']
```

In [22]:
```
mylist[-1] # Return last item of the list
```

Out[22]:
```
'eight'
```

In [23]:
```
mylist[:] # Return whole list
```

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

Add Remove & change items

In [24]:
```
mylist
```

Out[24]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [25]:
```
mylist.append('nine')  # add an item to the end of the list
mylist
```

Out[25]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [26]:
```
mylist.insert(9,'ten') # Add item at index lacation 9
mylist
```

Out[26]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'te
n']
```

In [27]:
```
mylist.insert(1,'ONE') # Add item at index lacation 9
mylist
```

Out[27]:
```
['one',
 'ONE',
 'two',
 'three',
 'four',
 'five',
 'six',
 'seven',
 'eight',
 'nine',
 'ten']
```

In [28]:
```
mylist.remove('ONE') # Remove item "ONE"
mylist
```

Out[28]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine', 'te
n']
```

In [29]:
```
mylist.pop()
mylist
```

Out[29]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [30]:
```
mylist.pop(4) # Remove item at index location 4
mylist
```

Out[30]:
```
['one', 'two', 'three', 'four', 'six', 'seven', 'eight', 'nine']
```

In [31]:
```
del mylist[7]    # Remove item at index location 7
mylist
```

Out[31]:
```
['one', 'two', 'three', 'four', 'six', 'seven', 'eight']
```

In [32]:
```
# change value of the string
mylist[0] = 1
mylist[1] = 2
mylist[2] = 3
mylist
```

Out[32]:
```
[1, 2, 3, 'four', 'six', 'seven', 'eight']
```

In [33]:
```
mylist.clear() # Empty list / Delete all items in the list
mylist
```

Out[33]:
```
[]
```

In [34]:
```
del mylist     # Delete the whole list
mylist
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[34], line 2
      1 del mylist     # Delete the whole list
----> 2 mylist

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

Copy List

In [35]:
```
mylist = ['one', 'two', 'three','four','five','six','seven','eight','nine']
```

In [36]:
```
mylist1 = mylist  # Create a new reference "mylist1"
```

In [37]:
```
id(mylist), id(mylist1)   # The address of both mylist & mylist1 will be same
```

Out[37]:
```
(2969446879872, 2969446879872)
```

In [38]:
```
mylist2 = mylist.copy() #Create a cpoy of the list
```

In [39]:
```
id(mylist2)  # The address of mylist2 will be different from mylist
```

Out[39]:
```
2969446733184
```

In [40]:
```
mylist[0] = 1
```

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

Out[41]:
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [42]:
```
mylist1        # mylist1 will be also impacted as it is pointing to the same l
```

Out[42]:
```
[1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [43]:
```
mylist2 # copy of list won't be impacted due to changes made on the original l
```

Out[43]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

Join Lists

In [44]:
```
list1 = ['one','two','three','four']
list2 = ['five','six','seven','eight']
```

In [45]:
```
list3 = list1 + list2 #join two lists by '+' operator
list3
```

Out[45]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [46]:
```
list1.extend(list2) #append list2 with list1
list1
```

Out[46]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

List Membership

In [47]:
```
list1
```

Out[47]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [48]:
```
'one'in list1 #check if 'one' exist in the list
```

Out[48]:
```
True
```

In [49]:
```
'ten' in list1 #check if 'ten' in the list
```

Out[49]:
```
False
```

In [50]:
```
if 'eight' in list1:
    print('eight is present in the list')
else:
    print('eight is not present in the list')
```

```
eight is present in the list
```

In [51]:
```
if 'nine' in list1:
    print('nine is present in the list')
else:
    print('nine is not present in the list')
```

```
nine is not present in the list
```

Reverse & Sort List

In [52]:
```
list1
```

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

In [53]:
```
list1.reverse()
list1
```

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

In [54]:
```
list1 = list1[::-1 ] # reverse the list
list1
```

Out[54]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [55]:
```
list1 = list1[::-1 ] # reverse the list
list1
```

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

In [56]:
```
mylist3 = [9,5,99,12,88,34]
mylist3.sort()     # Sort list in ascending order
mylist3
```

Out[56]:
```
[5, 9, 12, 34, 88, 99]
```

In [57]:
```
mylist3 = [9,5,99,12,88,34]
mylist3.sort(reverse=True)     # Sort list in descending order
mylist3
```

Out[57]:
```
[99, 88, 34, 12, 9, 5]
```

In [58]:
```
mylist4 = [88,34,56,12,76,34,65]
sorted(mylist4)                          #Returns a new sorted list and doesn't c
```

Out[58]:
```
[12, 34, 34, 56, 65, 76, 88]
```

In [59]:
```
mylist4
```

Out[59]:
```
[88, 34, 56, 12, 76, 34, 65]
```

Loop through a list

In [62]:
```
list1 = ['one','two','three','four','five','six','seven','eight']
```

In [63]:
```
list1
```

Out[63]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [64]:
```
for i in list1:
    print(i)
```

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

In [65]:
```
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 [66]:
```
list1
```

Out[66]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [67]:
```
list1.count('one')
```

Out[67]: 1

In [68]:
```
list4 =['one','two','three','four','one','two','three','six']
```

In [69]:
```
list4.count('three')
```

Out[69]: 2

In [70]:
```
list4.count('one')
```

Out[70]: 2

In [71]:
```
list4.count('six')
```

Out[71]: 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 [90]:
```
L1 = [1,2,3,4,5,6,7,11]  # will not return false as there is no false value(0)
```

In [91]:
```
all(L1)
```

Out[91]: True

In [92]:
```
any(L1)
```

Out[92]: True

In [75]:
```
L2 = [1,2,3,4,5,6,7,0]  # Will Return false as one value is false
```

In [76]:
```
any(L2)
```

Out[76]: True

In [77]:
```
all(L2)
```

Out[77]: False

In [87]:
```
L3 = [0,1,2,3,4,5,6,7]
```

In [88]:
```
any(L3)
```

Out[88]: True

In [89]:
```
all(L3)
```

Out[89]: False

In [93]:
```
L4 = [1,2,3,4,True]
```

In [94]:
```
any(L4)
```

Out[94]: True

In [95]:
```
all(L4)
```

Out[95]: True

In [96]:
```
L5 = [1,2,3,4,True,False]
```

In [97]:
```
any(L5) # Will return true as we have items in the list with true value
```

Out[97]: True

In [98]:
```
all(L5)      # Returns false as one value is false
```

Out[98]:
```
False
```