LIST Creation

In [2]:
```
l1 = []  # empty list
l1
```

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

In [3]:
```
print(type(l1))
```

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

In [4]:
```
l2 = [10,30,60]   # list of integer numbers
l2
```

Out[4]:
```
[10, 30, 60]
```

In [5]:
```
l3 = [10.77,30.66,60.89]    # list of float numbers
l3
```

Out[5]:
```
[10.77, 30.66, 60.89]
```

In [6]:
```
l4 = ['one','two','three']   # list of strings
l4
```

Out[6]:
```
['one', 'two', 'three']
```

In [7]:
```
l5 = ['savi',25,[50,100],[150,90]]   # Nested lists
l5
```

Out[7]:
```
['savi', 25, [50, 100], [150, 90]]
```

In [8]:
```
l6 = [100,'savi',17.765]   # list of mixed data types
l6
```

Out[8]:
```
[100, 'savi', 17.765]
```

In [9]:
```
len(l6)
```

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

In [10]:
```
l2
```

Out[10]:
```
[10, 30, 60]
```

In [12]:
```
l2[0]   # retrive frst element of the LIST
```

Out[12]:
```
10
```

In [13]:
```
l4[0]
```

Out[13]: 'one'

In [14]:
```
l4[0][0]   #Nested indexing - Access the frst character of the list of ele
```

Out[14]: 'o'

In [16]:
```
l4[-1]     # last item of the list
```

Out[16]: 'three'

In [18]:
```
l5[-1]     # last item of the list
```

Out[18]: [150, 90]

LIST Slicing

In [19]:
```
myl = ['one','two','three','four','five','six','seven','eight']
myl
```

Out[19]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [21]:
```
myl[0:3]    # return all items from 0th - 3rd index loaction
```

Out[21]: ['one', 'two', 'three']

In [22]:
```
myl[2:5]    # return all items from 2nd - 5th index loaction
```

Out[22]: ['three', 'four', 'five']

In [23]:
```
myl[:3]     # return frst three items
```

Out[23]: ['one', 'two', 'three']

In [24]:
```
myl[:2]
```

Out[24]: ['one', 'two']

In [25]:
```
myl[:-3]
```

Out[25]: ['one', 'two', 'three', 'four', 'five']

In [26]:
```
myl[-3:]    # return last three items
```

Out[26]: ['six', 'seven', 'eight']

In [27]:
```
myl[-2:]    # return last two items
```

Out[27]: ['seven', 'eight']

In [28]:
```
myl[-1:]    # return last item of the list
```

Out[28]:
```
['eight']
```

In [30]:
```
myl[:]       # return whole LIST
```

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

ADD , REMOVE & CHANGE ITEMS

In [21]:
```
myl = ['one','two','three','four','five','six','seven','eight']
myl
```

Out[21]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [22]:
```
myl.append('nine')   # add item to the END of the list
myl
```

Out[22]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight', 'nine']
```

In [23]:
```
myl.insert(9,'ten')     # add item at index location 9
myl
```

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

In [24]:
```
myl.insert(1,'TWO')   #ADD item at index location 1
myl
```

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

In [25]:
```
myl.remove('TWO')   # remove item 'ONE'
myl
```

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

In [26]:
```
myl.pop()     # REMOVE last item of the LIST
```

Out[26]:
```
'ten'
```

In [27]:
```
myl.pop(8)   # REMOVE item at index location 7
myl
```

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

In [28]:
```
# change value of the STRING
myl[0]   = 1
myl[1]   = 2
myl[2]  = 3
myl
```

Out[28]:
```
[1, 2, 3, 'four', 'five', 'six', 'seven', 'eight']
```

In [29]:
```
del myl[1]
```

In [30]:
```
myl
```

Out[30]:
```
[1, 3, 'four', 'five', 'six', 'seven', 'eight']
```

COPY LIST

In [31]:
```
myl
```

Out[31]:
```
[1, 3, 'four', 'five', 'six', 'seven', 'eight']
```

In [32]:
```
myl = ['one','two','three','four','five','six','seven','eight']
myl
```

Out[32]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [34]:
```
myl1 = myl    # create a new reference  "myl1"
myl1
```

Out[34]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [35]:
```
id(myl),id(myl1)  # the address of myl & myl1 as SAME
```

Out[35]:
```
(2250391420544, 2250391420544)
```

In [37]:
```
myl2 = myl.copy()      # create a copy of the LIST
myl2
```

Out[37]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [38]:
```
id(myl2)  # the address of myl2 is different from original list (myl)
```

Out[38]:
```
2250405363968
```

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

```
myl
```

Out[40]: [1, 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [41]:
```
myl
```

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

In [42]:
```
myl1   # myl1 also impacted
```

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

In [44]:
```
myl2   # copy of the list won't be impacted due to changes made in original L1
```

Out[44]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

JOIN LISTS

In [2]:
```
l1 = ['one','two','three','four']
l2 = ['five','six','seven','eight']
```

In [3]:
```
l3 = l1+l2   # join two LISTS by "+"  operator
l3
```

Out[3]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [4]:
```
l1.extend(l2)  #append list2 with list1
l1
```

Out[4]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

LIST membership

In [38]:
```
l1 = ['one','two','three','four','five','six','seven','eight']
l1
```

Out[38]: ['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']

In [39]:
```
'one' in l1   # check if 'one' exist in the LIST
```

Out[39]: True

In [8]:
```
'ten'  in l1
```

Out[8]: False

In [40]:
```
if 'three' in l1:
```

```
    print('three is present in the list')   # check if the three is exist in t
else:
    print('three is not present in the list')
```

```
three is present in the list
```

In [41]:
```
if 'eleven' in l1:
    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 [42]:
```
l1
```

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

In [43]:
```
l1.reverse()  # reverse the list
l1
```

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

In [44]:
```
l1 = l1[::-1]   # reverse the list
l1
```

Out[44]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [45]:
```
l1
```

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

In [46]:
```
l1.reverse()
l1
```

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

In [47]:
```
l1 = l1[-1]
l1
```

Out[47]:
```
'one'
```

In [48]:
```
l1 = l1[:-1]
l1
```

Out[48]:
```
'on'
```

In [49]:
```
l3 = [9,5,2,99,12,88,34]
```

```
l3.sort()      # sort list in ascending order
l3
```

Out[49]: [2, 5, 9, 12, 34, 88, 99]

In [51]:
```
l3.sort(reverse = True)   # sort in decending order
l3
```

Out[51]: [99, 88, 34, 12, 9, 5, 2]

In [ ]:

In [52]:
```
l4 = [88,65,3321,11,98]     # return a new sorted list does't  change original
sorted(l4)
```

Out[52]: [11, 65, 88, 98, 3321]

LOOP THROUGH LIST

In [56]:
```
myl = ['one','two','three','four','five','six','seven','eight']
myl
```

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

In [59]:
```
for i in myl:
    print(i)
```

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

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

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

COUNT

In [69]:
```
myl
```

Out[69]:
```
['one', 'two', 'three', 'four', 'five', 'six', 'seven', 'eight']
```

In [70]:
```
myl.count('one')  # no.of times 'one'  occured in list
```

Out[70]:
```
1
```

In [71]:
```
myl.count('two')  # no.of times 'two'  occured in list
```

Out[71]:
```
1
```

In [72]:
```
myl.count('three')  # no.of times 'three'  occured in list
```

Out[72]:
```
1
```

ALL / ANY

In [2]:
```
l1 = [1,2,3,4,0]
l1
```

Out[2]:
```
[1, 2, 3, 4, 0]
```

In [3]:
```
all(l1)    # will return false as one value is false(value 0)
```

Out[3]:
```
False
```

In [5]:
```
any(l1)     # will return True as we have items with True value
```

Out[5]:
```
True
```

In [6]:
```
l2 = [1,2,3,4,True,False]
```

In [8]:
```
all(l2)   # will return false as one value is false(value 0)
```

Out[8]:
```
False
```

In [7]:
```
any(l1)      # will return True as we have items with True value
```

Out[7]:
```
True
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: