List

-> List is one of the Sequence Data structure

-> Lists are collection of items (Strings, integers or even other lists)

-> Lists are enclosed in [ ]

-> Each item in the list has an assigned index value.

-> Each item in a list is separated by a comma

-> Lists are mutable, which means they can be changed.

List Creation

In [2]:
```
empty_list = []
print(empty_list)

list_strings = ["one", "two", "three", "four"] # list os strings
print(list_strings)

list_integers = [1, 2, 3, 4]#list of integers
print(list_integers)

list_of_list = [[1, 2, 4, 8],["my_list", "one", "two", "three"]]
print(list_of_list)
```

```
[]
['one', 'two', 'three', 'four']
[1, 2, 3, 4]
[[1, 2, 4, 8], ['my_list', 'one', 'two', 'three']]
```

List Length

In [3]:
```
"""
Length is inbuilt funtion in python for determinig/finding the lenght of the list
"""

list_len = [1, "one", 2, "two", 3, "three"]
print(len(list_len))
```

```
6
```

List Append

In [4]:
```
"""
The append() method adds an element/item to the end of the list
"""

#Syntax:  list.append(element)

list_append = [1,"One", 2,]
list_append.append("two")
print(list_append)
```

```
[1, 'One', 2, 'two']
```

List Insert

In [6]:
```
"""
The insert() method inserts a given element at a given index in a list
"""

#syntax: lst.insert(x, y)

list_insert = [1, "One", 2]
list_insert.insert(3, "two")
print(list_insert)
```

```
[1, 'One', 2, 'two']
```

List Remove

In [7]:
```
"""
The remove method removes the first occurence of specified element in a given list
"""

#syntax: lst.remove(x)

lst_remove = ['one', 'two', 'three', 'four', 'two']

lst_remove.remove('two') #it will remove first occurence of 'two' in a given list

print(lst_remove)
```

```
['one', 'three', 'four', 'two']
```

List Extend

In [8]:
```
"""
The extend() method joins the specified list elements to the end of the current list
"""

#syntax: lst.extend(x)

list_1 = [1, 2, 3, 4, 5]
list_2 = [6, 7, 8, 9, 10]
list_1.extend(list_2)
print(list_1)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

List Reverse

In [9]:
```
"""
The reverse method is reverses objects of list in place.

"""

#syntaxlst.reverse()

lst = ['one', 'two', 'three', 'four']

lst.reverse()

print(lst)
```

```
['four', 'three', 'two', 'one']
```

List Sorting

In [16]:
```
"""
The sorted() function returns a sorted list of the specified iterable object.
"""

#syntax = sorted(x)
numbers = [3, 1, 6, 2, 8]

sorted_lst = sorted(numbers)


print("Sorted list :", sorted_lst)

#original list remain unchanged
print("Original list: ", numbers)
```

```
Sorted list : [1, 2, 3, 6, 8]
Original list:  [3, 1, 6, 2, 8]
```

String Split to create a list

In [17]:
```
string_one = "one,two,three,four,five"
string_split = string_one.split(',')
print(string_split)
```

```
['one', 'two', 'three', 'four', 'five']
```

List Slicing

In [18]:
```
slice_numbers = [10, 20, 30, 40, 50,60,70,80]

#print all numbers
print(slice_numbers[:])

#print from index 0 to index 3
print(slice_numbers[0:4])
```

```
[10, 20, 30, 40, 50, 60, 70, 80]
[10, 20, 30, 40]
```

List Deletion

In [21]:
```
lst_del = ['one', 'two', 'three', 'four', 'five']

del lst_del[1]
print(lst_del)

#or we can use pop() method
a = lst_del.pop(1)
print(a)

print(lst)
```

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

List Count

In [19]:
```
"""
The count method returns count of how many times object occurs in list.
"""

numbers = [1, 2, 3, 1, 3, 4, 2, 5]

#frequency of 1 in a list
print(numbers.count(1))

#frequency of 3 in a list
print(numbers.count(3))
```

```
2
2
```

List Comprehension

In [20]:
```
#using list comprehension
squares = [i**2 for i in range(10)]
print(squares)

lst = [-10, -20, 10, 20, 50]

#create a new list with values doubled
new_lst = [i*2 for i in lst]
print(new_lst)

#filter the list to exclude negative numbers
new_lst = [i for i in lst if i >= 0]
print(new_lst)
```

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
[-20, -40, 20, 40, 100]
[10, 20, 50]
```

In [ ]:
```

```