

## List Methods #3

- The syntax of index method is `index(x [, start [, end ]])`
- The `index()` is used to search a particular element/item in a list and when found it returns its index.
- By default index takes starting and ending value as zero if the index is not defined
- The parameter `[ ]` (`start , end`) are optional in index method
- The start and end value in index are mostly given when you want to know the index value of the duplicate element because when value is not given it takes default value as 0 and returns the index value of first occurrence of the element.

### Example :

```
>>>
>>> l1 =[5,6,7,1,2,3,6,7,9,6]
>>> l1.index(1)
3                                #return the index of 1 i.e, 3
>>> l1.index(7)
2
>>> l1.index(5)
0
>>> l1.index(6,2)
6                                #return the index of 6 there indexing is
                                starting from 2
```

```
>>>
>>> l1.index(6,2,7)
6                                # number to be search is 6 in range of
>>> |                            index from 2 to 7
```

- When you give an ending point in indexing they will stop one step before that value. If element is not found then we'll get an error

- The **count(x)** method is used to know how many times a particular value/element is appearing in a list . It will not give the index value it will count the given number for example .

```
>>>
>>> l1 =[5,6,7,1,2,3,6,7,9,6]
>>> l1.count(6)
3
>>> l1.count(5)
1
>>> l1.count(7)
2
>>> |
```

- When the **reverse()** method is called it simply reverse the contents of a list.

```
>>>
>>> l1 =[5,6,7,1,2,3,6,7,9,6]
>>> l1.reverse()
>>> l1
[6, 9, 7, 6, 3, 2, 1, 7, 6, 5]
>>> |
```

- The syntax for sort method is `sort ( *, key = none , reverse = false )`
- The `sort()` sort the elements in the list and you'll get the elements of the list In sorted order.
- understanding the working of `sort()`

**Sort ( \*, key = none , reverse = false )**

Key = none - here we are defining our own criteria / function  
Reverse = false - reverse is key and false is value

```
>>>
>>> l1 = ['yy','JJ','mm','BB','aa','zz']
>>> l1.sort()
>>> l1
['BB', 'JJ', 'aa', 'mm', 'yy', 'zz']      #Before applying key = none
>>>
>>> l1.sort(key = str.lower)
>>> l1
['aa', 'BB', 'JJ', 'mm', 'yy', 'zz']      #after applying key =str.lower
>>> |
```

- Upper case letter are always treated as smaller than lower case letter [ based on ASCII value it happens]
- There's a global function called `sorted()` , This function will modify the original list it gives a new sorted list.

```
>>> l1 = ['yy', 'JJ', 'mm', 'BB', 'aa', 'zz']
>>> l1.sort()
>>> l1
['BB', 'JJ', 'aa', 'mm', 'yy', 'zz']
>>> sorted(l1)
['BB', 'JJ', 'aa', 'mm', 'yy', 'zz']
>>> |
```

#Modifies the given list i.e , sort it

#Not modify list but create a new sorted list