4.1.6 LIST ALIASING:

Since variables refer to objects, if we assign one variable to another, both variables refer to same objects.(One or more variable can refer the same object)

```
>>>a=[1,2,3]
>>>b=a
>>>a is b                # Displays True
```

4.1.7 LIST CLONING:

If we want to modify a list and also keep copy of the original we can use cloning to copy the list and make that as a reference.

Ex:

```
a=[1,2,3]
b=a[:]
```

print(b) → **[1,2,3]**

4.1.8 LIST PARAMETER:

Passing a list as an argument actually passes a reference to the list, not the copy of the list. We can also pass a list as an argument to the function.

Ex:

```
>>> def mul(a_list):   #a_list is a list passing as a parameter
        for index,value in enumerate(a_list):
                a_list[index]=2*value
        print(a_list)
>>> a_list=[1,2,3,4,5]
>>> mul(a_list)
```

Output:

```
[2, 4, 6, 8, 10]
```

4.1.9 DELETING LIST ELEMENTS

To remove a list element, del operator can be used if an element to be deleted is known. In the following code, the element 'Chennai' is deleted by mentioning its index in the del operator.

Ex:

```
stulist = ['Rama', 'Chennai', 2018,
'CSE', 92.7] print 'Initial list is : ',
stulist
del stulist[1]
print 'Now the list is : ', stulist
```

Output:

```
Initial list is : ['Rama', 'Chennai', 2018,
'CSE', 92.7] Now the list is : ['Rama',
2018, 'CSE', 92.7]
```

pop() and remove() methods can also be used to delete list elements