List Method #1

- Methods are member function of a class
- The methods discussed here are **append(x) , extend(iterable) , insert(i, x) , copy ( )**

append(x)

- It means adding one element at a time to a list
- It modifies the same list
- We can use slicing on append method

```
>>>
>>> L1 = [5,6,7,8,9]
>>> len(L1)
5
>>> L1.append(10)
>>> len(L1)
6
>>> L1
[5, 6, 7, 8, 9, 10]
>>> L1.append(11,12,13)
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    L1.append(11,12,13)
TypeError: list.append() takes exactly one argument (3 given)
>>>
```

```
>>>
>>> L1 = [5,6,7,8,9]
>>>
>>> L1[len(L1):]=[10]
>>> L1
[5, 6, 7, 8, 9, 10]
>>> L1[6:6]=[11]
>>> L1
[5, 6, 7, 8, 9, 10, 11]
>>>
```

extend (iterable)

- It is same as append but it can take collection of element in parameter
- It will modify the same list
- Extend can take list , tuple and string as parameter because they are iterable as well
- Slicing is also applicable on extend method

```
>>>
>>> L1 = [5,6,7,8,9]
>>> L1.extend([10,11,12])
>>> L1
[5, 6, 7, 8, 9, 10, 11, 12]
>>> L1.extend('abc')
>>> L1
[5, 6, 7, 8, 9, 10, 11, 12, 'a', 'b', 'c']
>>>
>>> L1=[5,6,7,8,9]
>>> L1[len(L1):]=[10,11,12]
>>> L1
[5, 6, 7, 8, 9, 10, 11, 12]
>>>
```

Insert (i , x)

- It is used to insert any element at a given index
- Same list is modified here
- Slicing is applicable on insert method

```
>>>
>>> L1 = [5,6,7,8,9]
>>> id(L1)
140505553016192
>>> L1.insert(0,10)
>>> L1
[10, 5, 6, 7, 8, 9]
>>> id(L1)
140505553016192
>>>
>>> L1
[10, 5, 6, 7, 8, 9]
>>> L1.insert(3,20)
>>> L1
[10, 5, 6, 20, 7, 8, 9]
>>> L1[4:4]=[22]
>>> L1
[10, 5, 6, 20, 22, 7, 8, 9]
>>> L1[0:0]=[25]
>>> L1
[25, 10, 5, 6, 20, 22, 7, 8, 9]
>>>
```

Copy

- It copy( ) will create a shallow copy of the list
- It returns a new list after copying it

```
>>>
>>> L1 = [5, 6, 7, 8, 9]
>>> L2 =L1.copy()
>>> L2
[5, 6, 7, 8, 9]
>>> L1
[5, 6, 7, 8, 9]
>>> id(L1)
140265482259968
>>> id(L2)
140265443651840
>>> L1[0]
5
>>> L2[0]
5
>>> id(L1[0])
140265436354992
>>>
```