

Deleting Elements of LIST

1. pop () Method

2. del Method

3. remove ()
Method

Deleting Elements of LIST

1. pop () Method

It removes the element from the specified index, and also return the element which was removed.

Its syntax is:

```
List.pop ([index])
```

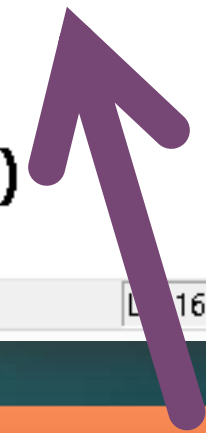
Deleting Elements of LIST

1. pop () Method

For Example:

```
Python 3.4
File Edit Shell Debug Options Windows
>>> L1=[25,67,89,103]
>>> L1.pop(2)
89
>>> print(L1)
[25, 67, 103]
```

```
Python 3.4.0 Shell
File Edit Shell Debug Options
Windows Help
>>> x=L1.pop(1)
>>> print(x)
67
>>> print(L1)
[25, 103]
```

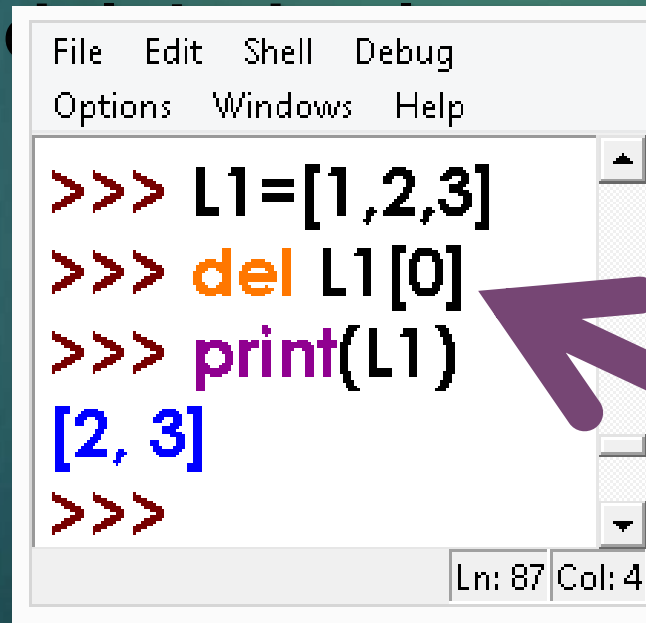


Popped element 67 is assigned to x

Deleting Elements of LIST

2. del Method

`del` removes the specified element from the list, but does not return the



```
File Edit Shell Debug
Options Windows Help
>>> L1=[1,2,3]
>>> del L1[0]
>>> print(L1)
[2, 3]
>>>
```

The screenshot shows a Python shell window with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help). The code entered is: `>>> L1=[1,2,3]`, `>>> del L1[0]`, and `>>> print(L1)`. The output is `[2, 3]`. A purple arrow points from the `del` command in the code to the `del` text in the adjacent box.

Syntax :

del

Listobject[index]

del method

example

Deleting Elements of LIST

3. remove () Method

In case, we know the element to be deleted not the index, of the element, then remove () can be used.

Syntax is:

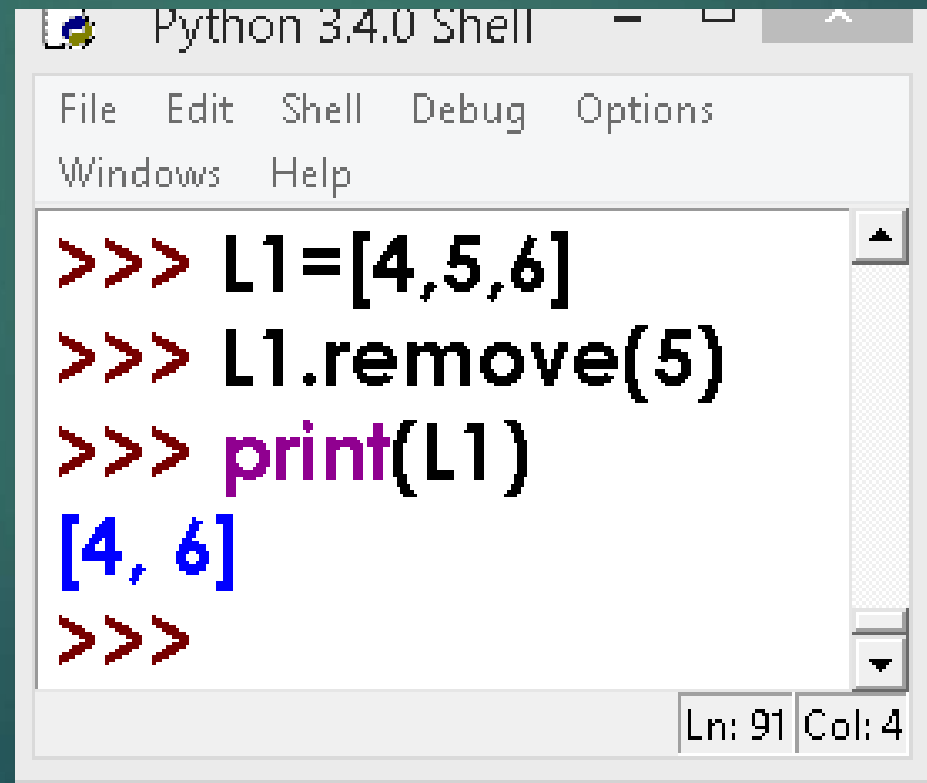
```
listobject.remove(value)
```

```
>>>L1.remove(60)
```

contd...

Deleting Elements of LIST

3. remove () Method



```
Python 3.4.0 Shell
File Edit Shell Debug Options
Windows Help
>>> L1=[4,5,6]
>>> L1.remove(5)
>>> print(L1)
[4, 6]
>>>
```

Ln: 91 Col: 4



SOME MORE METHODS

SOME MORE METHODS

1. insert () Method

2. reverse ()
Method

3. sort () Method

4. clear () Method

5. index () Method

6. extend ()
Method

OTHER METHODS

1. insert () Method

This method allows us to insert an element, at the given position specified by its index, and the remaining elements are shifted to accommodate the new element. **insert ()** requires two arguments-**index value** and **item value**.

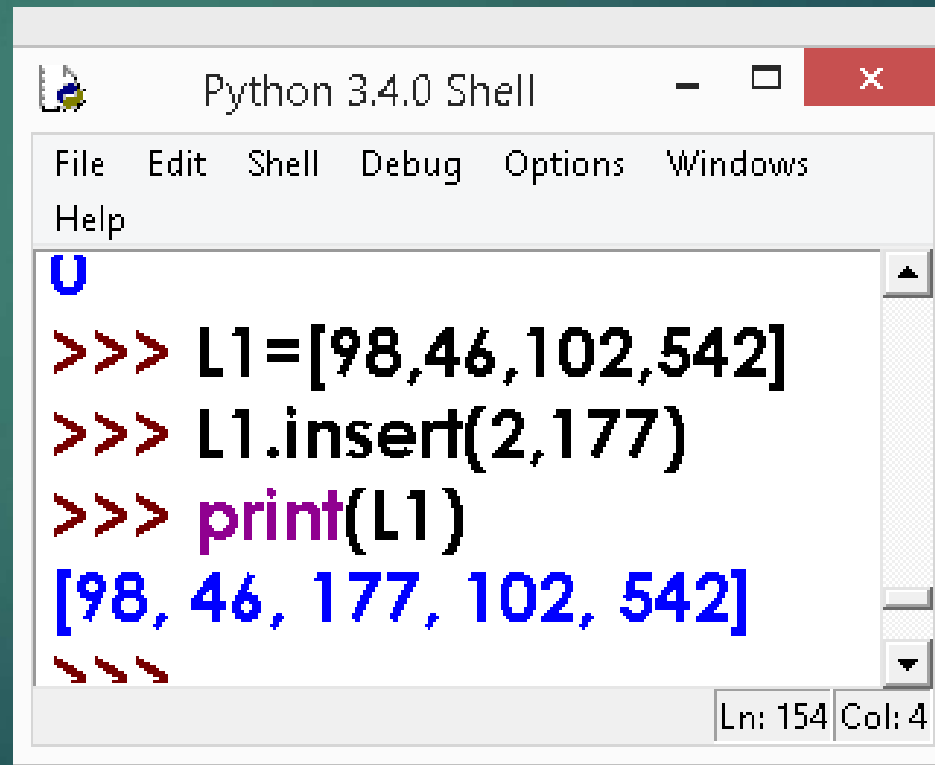
Its syntax is

```
list.insert(index, item)
```

OTHER METHODS

1. insert () Method

For Example:



```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows
Help
U
>>> L1=[98,46,102,542]
>>> L1.insert(2,177)
>>> print(L1)
[98, 46, 177, 102, 542]
>>>
```

Ln: 154 Col: 4

OTHER METHODS

2. reverse () Method

This method can be used to reverse the elements of the list in place

Its syntax is:

```
list.reverse ()
```

Method does not return anything as the reversed list is stored in the same variable.

OTHER METHODS

2. reverse () Method

For Example:

```
File Edit Shell Debug Options Windows  
Help  
  
>>> print(L1)  
[98, 46, 177, 102, 542]  
>>> L1.reverse()  
>>> print(L1)  
[542, 102, 177, 46, 98]
```

Ln: 159 Col: 4

OTHER METHODS

3. `sort ()` Method

For arranging elements in an order Python provides a method `sort ()` and a function `sorted ()`. `sort ()` modifies the list in place and `sorted ()` returns a new sorted list.

OTHER METHODS

3. sort () Method

For Example:

Default
Ascending
Order sorting

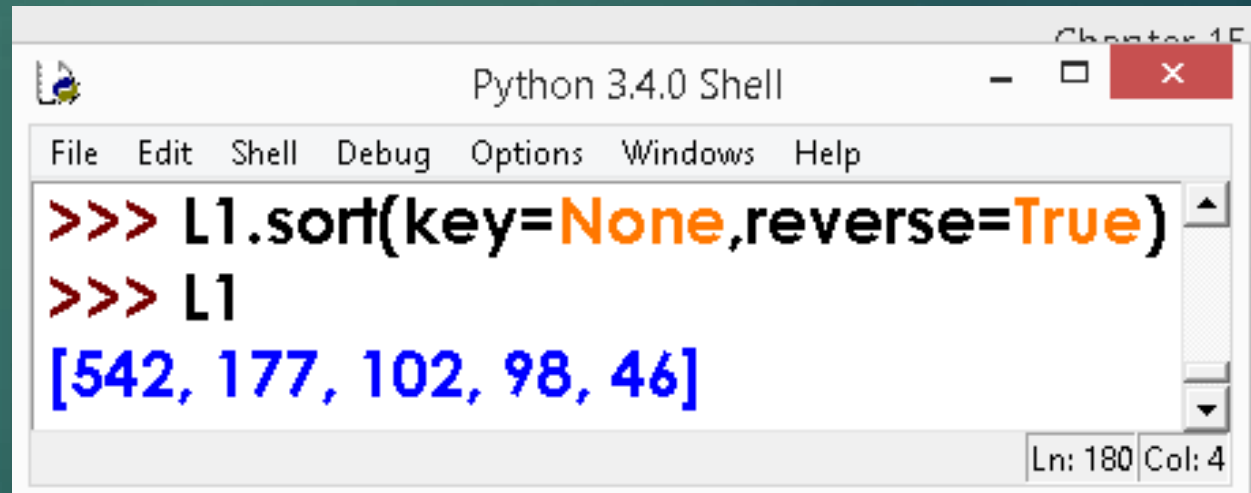


```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows
Help
>>> print(L1)
[542, 102, 177, 46, 98]
>>> L1.sort()
>>> print(L1)
[46, 98, 102, 177, 542]
Ln: 170 Col: 4
```

OTHER METHODS

3. sort () Method

For Example:



```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
>>> L1.sort(key=None,reverse=True)
>>> L1
[542, 177, 102, 98, 46]
```

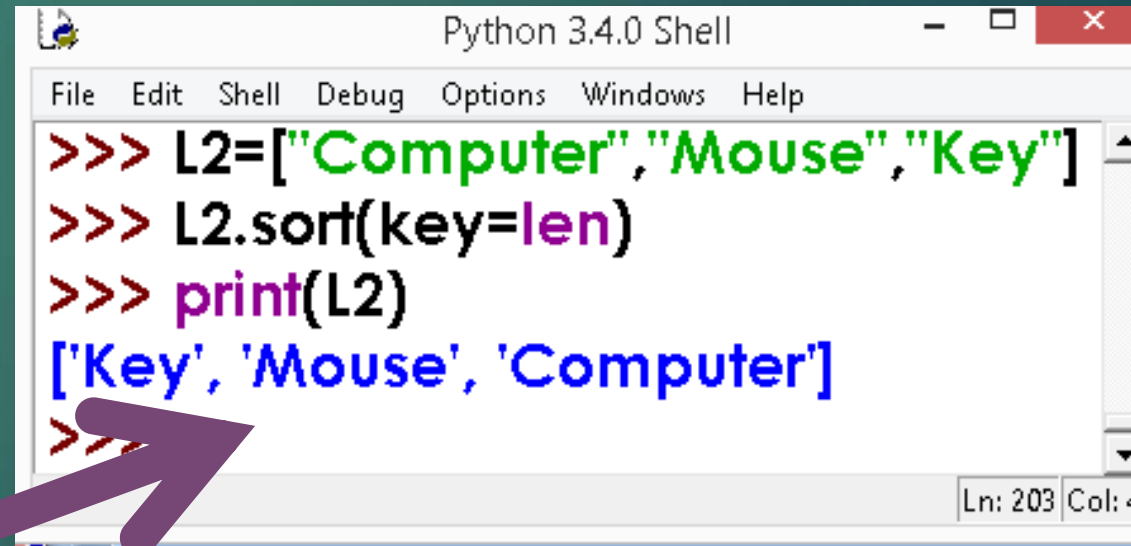
The screenshot shows a Python 3.4.0 Shell window with a menu bar (File, Edit, Shell, Debug, Options, Windows, Help). The command `L1.sort(key=None,reverse=True)` is entered, followed by `L1`. The output is `[542, 177, 102, 98, 46]`. The status bar at the bottom right indicates 'Ln: 180 Col: 4'.

Descending
order

OTHER METHODS

3. sort () Method

For Example:



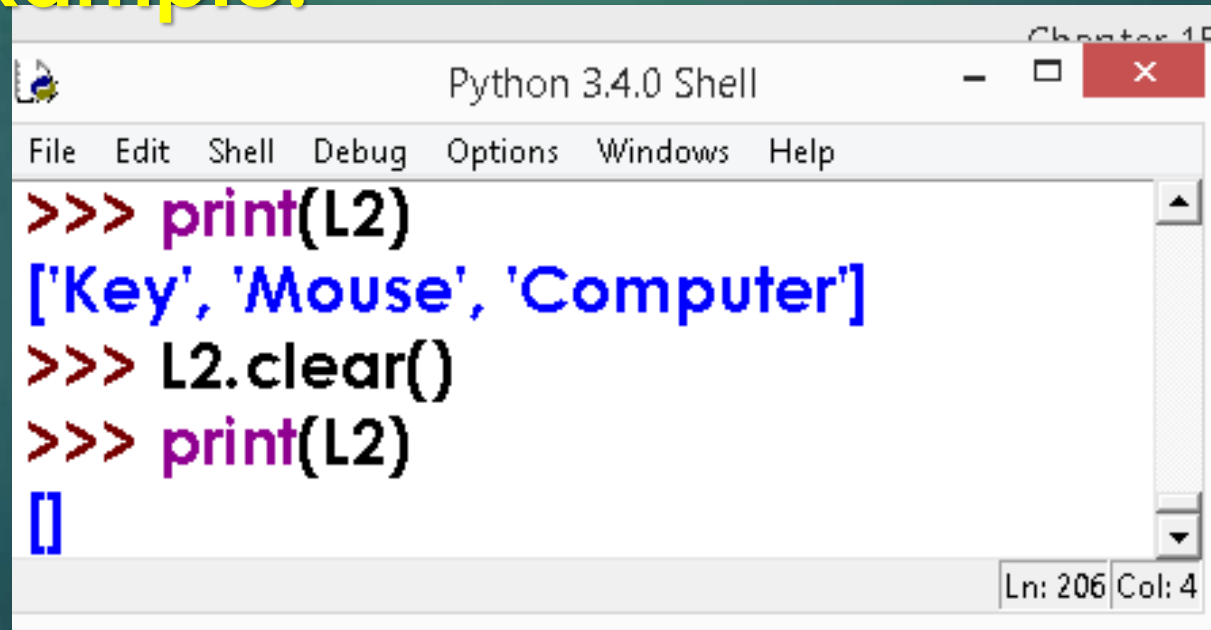
```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
>>> L2=["Computer","Mouse","Key"]
>>> L2.sort(key=len)
>>> print(L2)
['Key', 'Mouse', 'Computer']
>>>
```

A screenshot of a Python 3.4.0 Shell window. The window title is "Python 3.4.0 Shell". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Windows", and "Help". The main area shows a Python interactive session where a list L2 is defined with elements "Computer", "Mouse", and "Key". The list is then sorted in place using the sort method with the key parameter set to len. The output shows the list is now ['Key', 'Mouse', 'Computer']. A purple arrow points from the text below to the 'key=len' argument in the code.

Sorting using key value (length of string)

OTHER METHODS

4. clear () Method **clear() method**
deletes
or clears the content of list. For
Example:



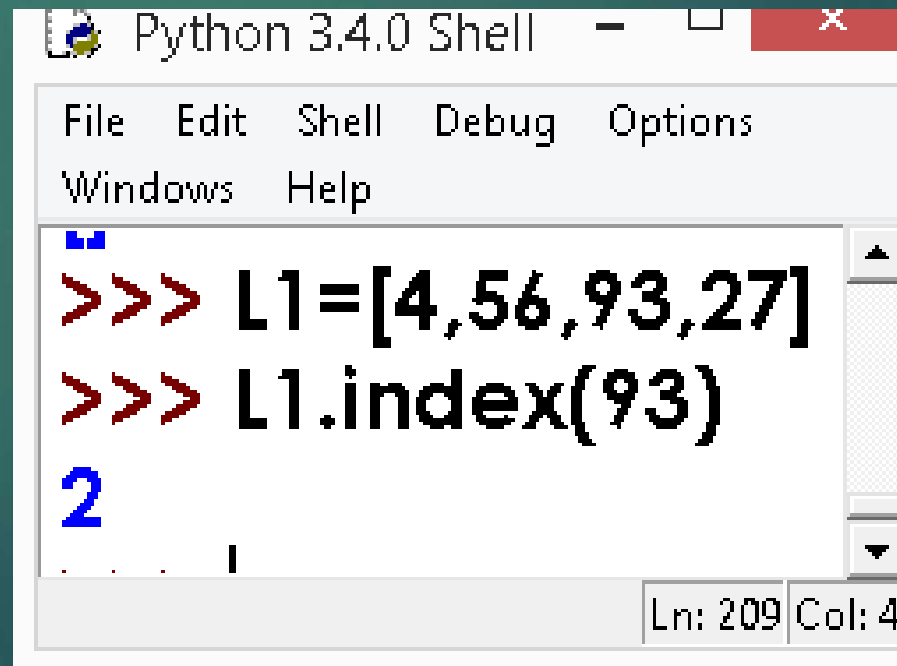
```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
>>> print(L2)
['Key', 'Mouse', 'Computer']
>>> L2.clear()
>>> print(L2)
[]
Ln: 206 Col: 4
```

OTHER METHODS

5. index () Method

To know the index of an element in a given list

For Example:



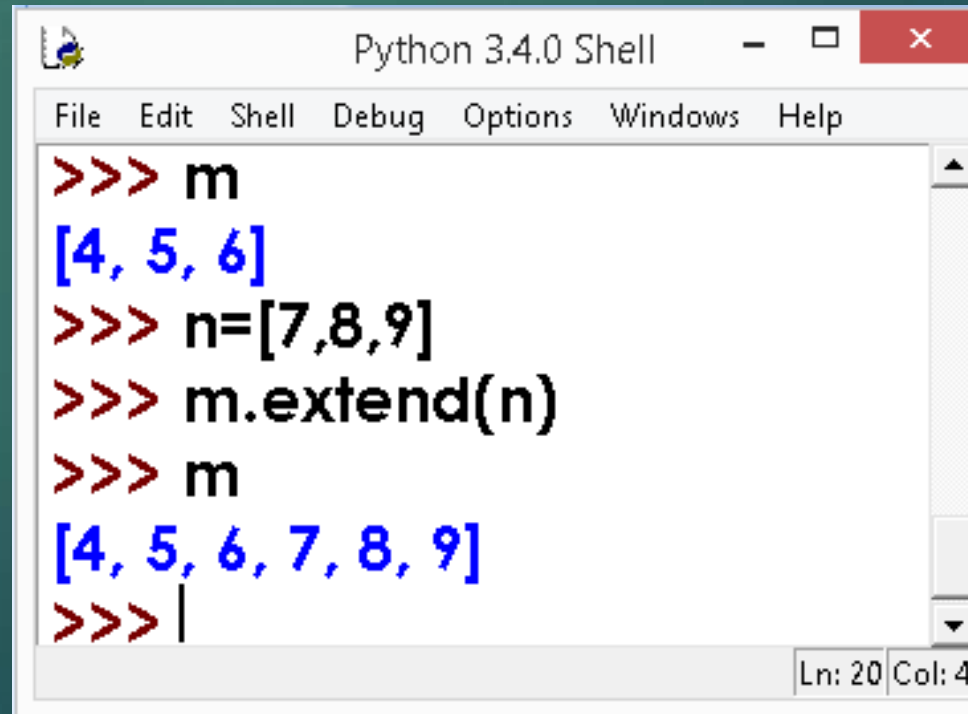
```
Python 3.4.0 Shell
File Edit Shell Debug Options
Windows Help
>>> L1=[4,56,93,27]
>>> L1.index(93)
2
Ln: 209 Col: 4
```

OTHER METHODS

6. extend () Method

Extending a given list with another list.

For Example:



```
Python 3.4.0 Shell
File Edit Shell Debug Options Windows Help
>>> m
[4, 5, 6]
>>> n=[7,8,9]
>>> m.extend(n)
>>> m
[4, 5, 6, 7, 8, 9]
>>> |
Ln: 20 Col: 4
```