

List

Lists are similar to arrays in C. However; the list can contain data of different types. The items stored in the list are separated with a comma (,) and enclosed within square brackets [].

We can use slice [:] operators to access the data of the list. The concatenation operator (+) and repetition operator (*) works with the list in the same way as they were working with the strings.

Example	Output
list = ['abc', 123 , 'hi', 10.12 , 10] print list print list[0] print list[2:4] print list[2:] print list * 2 print list + list	['abc', 123, 'hi', 10.12, 10] abc ['hi', 10.12] ['hi', 10.12, 10] ['abc', 123, 'hi', 10.12, 10, 'abc', 123, 'hi', 10.12, 10] ['abc', 123, 'hi', 10.12, 10, 'abc', 123, 'hi', 10.12, 10]

Access Python List

Python allows us to access value from the list by various ways.

Example	Output
list = ['abc', 123 , 'hi', 10.12 , 10] print list[0]	abc ['hi', 10.12] ['hi', 10.12, 10]

print list[2:4] print list[2:] print list[-3:-1]	['hi', 10.12]
--	---------------

Adding Python Lists

In Python, lists can be added by using the concatenation operator(+) to join two lists.

Example	Output
list = ['abc', 123] list1 = ['hi', 10.12 , 10] print list+list1	['abc', 123, 'hi', 10.12, 10]

Replicating lists

Replicating means repeating, It can be performed by using '*' operator by a specific number of time.

Example	Output
list = ['abc', 123] print list*2	['abc', 123, 'abc', 123]

List Slicing

A subpart of a list can be retrieved on the basis of index. This subpart is known as list slice. This feature allows us to get sub-list of specified start and end index.

Example	Output
list = ['abc', 123 , 'hi', 10.12 , 10] print list[0] print list[2:4] print list[2:]	abc ['hi', 10.12] ['hi', 10.12, 10]

Updating List

To update or change the value of particular index of a list, assign the value to that particular index of the List.

Example	Output
list = ['abc', 123 , 'hi', 10.12 , 10] print list list[2]='how' print list	['abc', 123, 'hi', 10.12, 10] ['abc', 123, 'how', 10.12, 10]

Appending Python List

Python provides, append() method which is used to append i.e., add an element at the end of the existing elements.

Example	Output
list = ['abc', 123 , 'hi', 10.12 , 10] list.append('how') print list	['abc', 123, 'hi', 10.12, 10, 'how']

Deleting Elements

In Python, **del** statement can be used to delete an element from the list. It can also be used to delete all items from startIndex to endIndex.

Example	Output
<pre>list = ['abc', 123 , 'hi', 10.12 , 10] del list[2:4] print list</pre>	<pre>['abc', 123, 10]</pre>

Following are the common list functions.

min() method

This method is used to get min value from the list.

Example	Output
<pre>list = [1, 123 , 102, 10.12 , 10] print min(list)</pre>	<pre>1</pre>

max() method

This method is used to get max value from the list.

Example	Output
<pre>list = [1, 123 , 102, 10.12 , 10] print max(list)</pre>	<pre>123</pre>

len() method

This method is used to get length of the the list.

Example	Output
list = [1, 123 , 102, 10.12 , 10] print len(list)	5

cmp() method

The method **cmp()** compares elements of two lists.

Example	Output
list = [1, 123] list1 = [10.12 , 10] print cmp(list,list1) print cmp(list1,list)	-1 1

Following are the built-in methods of List

index(object) Method

It returns the index value of the object.

Example	Output
list = [1, 123 , 102, 10.12 , 10] print list.index(102)	2

count(object) Method

It returns the number of times an object is repeated in list.

Example	Output
list = [1, 123 , 102, 10.12 , 10]	1

<pre>print list.count(102) list1 = [1, 123 , 102, 102 , 10] print list1.count(102)</pre>	2
--	---

pop()/pop(index) Method

It returns the last object or the specified indexed object. It removes the popped object.

Example	Output
<pre>list = [1, 123 , 102, 10.12 , 10] print list.pop() print list.pop(2) print list</pre>	<pre>10 102 [1, 123, 10.12]</pre>

insert(index,object) Method

It inserts an object at the given index.

Example	Output
<pre>list = [1, 123 , 102, 10.12 , 10] print list.insert(3,20) print list</pre>	<pre>[1, 123, 102, 20, 10.12, 10]</pre>

extend(sequence) Method

It adds the sequence to existing list.

Example	Output
<pre>list = [1, 123 , 102, 10.12 , 10] list1 = [3 , 4]</pre>	<pre>[1, 123, 102, 10.12, 10, [3, 4]]</pre>

print list.append(list1) print list	
--	--

remove(object) Method

It removes the object from the given List.

Example	Output
list = [1, 123 , 102, 10.12 , 10] list.remove(123) print list	[1, 102, 10.12, 10]

reverse()Method

It reverses the position of all the elements of a list.

Example	Output
list = [1, 123 , 102, 10.12 , 10] list.reverse() print list	[10, 10.12, 102, 123, 1]

sort() Method

It is used to sort the elements of the List.

Example	Output
list = [1, 123 , 102, 10.12 , 10] list.sort() print list	[1, 10, 10.12, 102, 123]