BCA – 502: Python Programming

Rahul Kumar Singh

In today's Class we have discussed on Built-in List Methods of Python.

List manipulation using in build methods:-

Python includes the following built-in methods to manipulate list –

1. List cmp() Method:-

Python list method **cmp()** compares elements of two lists.

Syntax:-

Following is the syntax for cmp() method –

cmp(list1, list2)

Parameters

list1 – This is the first list to be compared.

list2 – This is the second list to be compared.

Return Value

If elements are of the same type, perform the compare and return the result. If elements are different types, check to see if they are numbers.

- If numbers, perform numeric coercion if necessary and compare.
- If either element is a number, then the other element is "larger" (numbers are "smallest").
- Otherwise, types are sorted alphabetically by name.

If we reached the end of one of the lists, the longer list is "larger." If we exhaust both lists and share the same data, the result is a tie, meaning that 0 is returned.

Example:-

The following example shows the usage of cmp() method.

```
#!/usr/bin/python

list1, list2 = [123, 'xyz'], [456, 'abc']

print cmp(list1, list2)

print cmp(list2, list1)

list3 = list2 + [786];

print cmp(list2, list3)
```

When we run above program, it produces following result−

```
-1
```

1

-1

2. List len() Method:-

Python list method len() returns the number of elements in the list.

Syntax

Following is the syntax for len() method –

len(list)

Parameters

list – This is a list for which number of elements to be counted.

Return Value

This method returns the number of elements in the list.

Example:-

The following example shows the usage of len() method.

```
#!/usr/bin/python

list1, list2 = [123, 'xyz', 'zara'], [456, 'abc']

print "First list length : ", len(list1)
```

```
print "Second list length : ", len(list2)
```

When we run above program, it produces following result–

```
First list length :  3
Second list length :  2
```

3. List max() Method:-

Python list method max returns the elements from the list with maximum value.

Syntax:-

Following is the syntax for max() method –

```
max(list)
```

Parameters

list – This is a list from which max valued element to be returned.

Return Value

This method returns the elements from the list with maximum value.

Example:-

The following example shows the usage of max()

method.

```
#!/usr/bin/python

list1, list2 = [123, 'xyz', 'zara', 'abc'], [456, 700, 200]

print "Max value element : ", max(list1)

print "Max value element : ", max(list2)
```

When we run above program, it produces following result−

```
Max value element :  zara

Max value element :  700
```

4. List min() Method:-

Python list method **min()** returns the elements from the list with minimum value.

Syntax

Following is the syntax for min() method −

```
min(list)
```

Parameters

list − This is a list from which min valued element to be returned.

Return Value

This method returns the elements from the list with minimum value.

Example

The following example shows the usage of min() method.

```
#!/usr/bin/python

list1, list2 = [123, 'xyz', 'zara', 'abc'], [456, 700, 200]

print "min value element : ", min(list1)

print "min value element : ", min(list2)
```

When we run above program, it produces following result−

```
min value element :  123

min value element :  200
```

5. List list() Method:-

Python list method **list()** takes sequence types and converts them to lists. This is used to convert a given tuple into list.

Note − Tuple are very similar to lists with only difference

that element values of a tuple can not be changed and tuple elements are put between parentheses instead of square bracket.

Syntax

Following is the syntax for list() method −

```
list( seq )
```

Parameters

seq − This is a tuple to be converted into list.

Return Value

This method returns the list.

Example

The following example shows the usage of list() method.

```
#!/usr/bin/python

aTuple = (123, 'xyz', 'zara', 'abc');

aList = list(aTuple)

print "List elements : ", aList
```

When we run above program, it produces following result−

```
List elements :  [123, 'xyz', 'zara', 'abc']
```

6. List append() Method:-

Python list method **append()** appends a passed obj into the existing list.

Syntax

Following is the syntax for append() method –

list.append(obj)

Parameters

obj – This is the object to be appended in the list.

Return Value

This method does not return any value but updates existing list.

Example

The following example shows the usage of append() method.

```
#!/usr/bin/python

aList = [123, 'xyz', 'zara', 'abc'];

aList.append( 2009 );

print "Updated List : ", aList
```

When we run above program, it produces following result–

Updated List : [123, 'xyz', 'zara', 'abc', 2009]

7. List count() Method:-

Python list method **count()** returns count of how many times obj occurs in list.

Syntax

Following is the syntax for count() method –

list.count(obj)

Parameters

obj – This is the object to be counted in the list.

Return Value

This method returns count of how many times obj occurs in list.

Example

The following example shows the usage of count() method.

#!/usr/bin/python

```
aList = [123, 'xyz', 'zara', 'abc', 123];

print "Count for 123 : ", aList.count(123)

print "Count for zara : ", aList.count('zara')
```

When we run above program, it produces following result−

```
Count for 123 :  2

Count for zara :  1
```

8. List extend() Method:-

Python list method **extend()** appends the contents of seq to list.

Syntax

Following is the syntax for extend() method −

```
list.extend(seq)
```

Parameters

seq − This is the list of elements

Return Value

This method does not return any value but add the content to existing list.

Example

The following example shows the usage of extend() method.

```
#!/usr/bin/python

aList = [123, 'xyz', 'zara', 'abc', 123];

bList = [2009, 'manni'];

aList.extend(bList)

print "Extended List : ", aList
```

When we run above program, it produces following result–

```
Extended List :  [123, 'xyz', 'zara', 'abc', 123, 2009, 'manni']
```

9. List index() Method:-

Python list method **index()** returns the lowest index in list that obj appears.

Syntax

Following is the syntax for index() method –

```
list.index(obj)
```

Parameters

obj – This is the object to be find out.

Return Value

This method returns index of the found object otherwise raise an exception indicating that value does not find.

Example

The following example shows the usage of index() method.

```
#!/usr/bin/python

aList = [123, 'xyz', 'zara', 'abc'];

print "Index for xyz : ", aList.index( 'xyz' )

print "Index for zara : ", aList.index( 'zara' )
```

When we run above program, it produces following result–

```
Index for xyz :  1

Index for zara :  2
```

10. List insert() Method:-

Python list method **insert()** inserts object obj into list at

offset index.

Syntax

Following is the syntax for insert() method −

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

Parameters

index − This is the Index where the object obj need to be inserted.

obj − This is the Object to be inserted into the given list.

Return Value

This method does not return any value but it inserts the given element at the given index.

Example

The following example shows the usage of insert() method.

```
#!/usr/bin/python

aList = [123, 'xyz', 'zara', 'abc']

aList.insert( 3, 2009)

print "Final List : ", aList
```

When we run above program, it produces following result–

Final List : [123, 'xyz', 'zara', 2009, 'abc']

11. List pop() Method:-

Python list method **pop()** removes and returns last object or obj from the list.

Syntax

Following is the syntax for pop() method –

list.pop(obj = list[-1])

Parameters

obj – This is an optional parameter, index of the object to be removed from the list.

Return Value

This method returns the removed object from the list.

Example

The following example shows the usage of pop() method.

#!/usr/bin/python

aList = [123, 'xyz', 'zara', 'abc'];

```
print "A List : ", aList.pop()

print "B List : ", aList.pop(2)
```

When we run above program, it produces following result−

```
A List :  abc

B List :  zara
```

12. List remove() Method:-

Python list method remove() searches for the given element in the list and removes the first matching element.

Syntax

Following is the syntax for remove() method −

```
list.remove(obj)
```

Parameters

obj − This is the object to be removed from the list.

Return Value

This Python list method does not return any value but removes the given object from the list.

Example

The following example shows the usage of remove() method.

```
#!/usr/bin/python

aList = [123, 'xyz', 'zara', 'abc', 'xyz'];

aList.remove('xyz');

print "List : ", aList

aList.remove('abc');

print "List : ", aList
```

When we run above program, it produces following result–

```
List :  [123, 'zara', 'abc', 'xyz']

List :  [123, 'zara', 'xyz']
```

13. List reverse() Method:-

Python list method reverse() reverses objects of list in place.

Syntax

Following is the syntax for reverse() method –

```
list.reverse()
```

Return Value

This method does not return any value but reverse the given object from the list.

Example

The following example shows the usage of reverse() method.

```
#!/usr/bin/python

aList = [123, 'xyz', 'zara', 'abc', 'xyz'];

aList.reverse();

print "List : ", aList
```

When we run above program, it produces following result–

```
List :  ['xyz', 'abc', 'zara', 'xyz', 123]
```

14. List sort() Method:-

Python list method sort() sorts objects of list, use compare func if given.

Syntax

Following is the syntax for sort() method −

list.sort([func])

Return Value

This method does not return any value but it changes from the original list.

Example

The following example shows the usage of sort() method.

```
#!/usr/bin/python

aList = [123, 'xyz', 'zara', 'abc', 'xyz'];

aList.sort();

print "List : ", aList
```

When we run above program, it produces following result−

```
List :  [123, 'abc', 'xyz', 'xyz', 'zara']
```