

## Python tutorial-List

### Python Lists and Tuples

In Python, list and tuple are sequences. The difference between list and tuple is that the list can be changed while tuple can't be changed. You can insert, update, or delete items of the list but you can't do that with the tuple.

#### Lists: creating a list

Let's create a list to store the names of months of the year.

```
>>> months=['January',
'February','March','April','May','June','Junly','August','September','October','November',
'December']

>>> months

['January', 'February', 'March', 'April', 'May', 'June', 'Junly', 'August', 'September', 'October',
'November', 'December']

>>>
```

#### Lists: Accessing list elements

You can access a list element by specifying its number. The numbers of the list ranks from 0 upwards.

```
>>> months[0]

'January'

>>> months[1]

'February'

>>>
```

You can access the list elements from the last to the beginning by using negative number:

```
>>> months[-1]

'December'

>>> months[-2]

'November'

>>>
```

You also can access a range of the list elements by using a range of numbers:

```
>>> months[0:6]
['January', 'February', 'March', 'April', 'May', 'June']
>>> months[-6:-1]
['Junly', 'August', 'September', 'October', 'November']
```

If you want to access the elements from a position to the end, you can specify the start number and leave the end blank:

```
>>> months[5:]
['June', 'Junly', 'August', 'September', 'October', 'November', 'December']
>>>
>>> months[-3:]
['October', 'November', 'December']
```

Note: A string is also a sequence. Therefore, to access characters in a string you need to specify their numbers.

```
>>> 'Hello'[0]
'H'
>>> yourname='Channa'
>>> yourname[1]
'h'
>>>
```

In accessing the list elements you can specify the step:

```
>>> yourname[0:6:2]
'Can'
>>>
```

**List: adding lists**

In Python, you can not only add numbers but also you can add sequences.

```
>>> lst1=[1,2,3]
```

```
>>> lst2=[4,5,6]
```

```
>>> lst3=lst1+lst2
```

```
>>> lst3
```

```
[1, 2, 3, 4, 5, 6]
```

```
>>> 'Dear,'+'Mr.Dara'
```

```
'Dear,Mr.Dara'
```

```
>>>
```

### **List: multiplying lists**

You can multiply a sequence with a number.

```
>>> lst4=2*lst1
```

```
>>> lst4
```

```
[1, 2, 3, 1, 2, 3]
```

```
>>>
```

### **List: checking membership**

You can use in operator to check whether a value is in a sequence.

```
>>> 1 in lst4
```

```
True
```

```
>>> 'H' in 'Hello'
```

```
True
```

```
>>>
```

### **List: lists in list**

An element of a list can be a list.

```
>>> lst5=[[1,2],[3,4],[5,6]]
```

```
>>> lst5[0]
```

```
[1, 2]
```

```
>>>
```

### **List: len, min, and max**

len() method is used to get the number of items in a list while min() and max() are used to get the minimum, and the maximum elements of the list.

```
>>> lst1=[1,2,3]
```

```
>>> len(lst1)
```

```
3
```

```
>>> min(lst1)
```

```
1
```

```
>>> max(lst1)
```

```
3
```

```
>>
```

### **List: changing list elements**

```
>>> lst1[2]=5
```

```
>>> lst1
```

```
[1, 2, 5]
```

```
>>> lst1[0:2]=12,15
```

```
>>> lst1
```

```
[12, 15, 5]
```

```
>>>
```

### **List: deleting list elements**

You can delete list elements by using del keyword.

```
>>> del lst1[2]
```

```
>>> lst1
```

```
[12, 15]
```

```
>>>
```

### **List Methods: append, extend, count, index, insert, pop, remove, reverse, and sort methods.**

You can also append a new element to a list by using append() method:

```
>>> lst1.append(16)
```

```
>>> lst1
```

```
[12, 15, 16]
```

```
>>>
```

To append a range of values to the list, use extend() method:

```
>>> lst1.extend([17,18,19])
```

```
>>> lst1
```

```
[12, 15, 16, 17, 18, 19]
```

```
>>>
```

To count frequencies of an element in the list you will use count() method:

```
>>> lst1.count(15)
```

```
1
```

```
>>>
```

To find index of an element, you can use index() method:

```
>>> lst1=[12,13,14,15]
```

```
>>> lst1.index(12)
```

```
0
```

```
>>>
```

You can insert an object to any position of the list by using insert() method:

```
>>> lst1.insert(4,16)
```

```
>>> lst1
```

```
[12, 13, 14, 15, 16]
```

```
>>> lst1.insert(0,11)
```

```
>>> lst1
```

```
[11, 12, 13, 14, 15, 16]
```

```
>>> lst1.insert(2,'Hello')
```

```
>>> lst1
```

```
[11, 12, 'Hello', 13, 14, 15, 16]
```

```
>>>
```

List also contains pop() method that lets you delete the last element of the list:

```
>>> lst1.pop()
```

```
16
```

```
>>> lst1
```

```
[11, 12, 'Hello', 13, 14, 15]
```

With remove() method, you can remove an item from the list starting from the beginning( left-to-right). If there are more than one the same items, the first left will be removed:

```
>>> lst1.remove(11)
```

```
>>> lst1
```

```
[12, 'Hello', 13, 14, 15]
```

```
>>> lst2=[11,12,14,11,11]
```

```
>>> lst2.remove(11)
```

```
>>> lst2
```

```
[12, 14, 11, 11]
```

```
>>>
```

To reverse the display order of a list, use reverse() method:

```
>>> lst2.reverse()
```

```
>>> lst2
```

```
[11, 11, 14, 12]
```

```
>>>
```

To sort the list, use sort() method:

```
>>> lst2.sort()
```

```
>>> lst2
```

```
[11, 11, 12, 14]
```

```
>>>
```