Python List Methods

Python List Methods

Python list class provides many methods that transform or operate on the items of the List. In addition to these, we have builtin methods that operate on list objects and transform them.

In this tutorial, we will learn about all the available methods of Python List. Following is the list of methods.

1. list.append()
2. list.extend()
3. list.insert()
4. list.remove()
5. list.pop()
6. list.clear()
7. list.index()
8. list.count()
9. list.sort()
10. list.reverse()
11. list.copy()

list.append(x)

Python List append(x) method adds the item x to the end of the list.

Example Program

```
list1 = [25, 74, 19]
list1.append(87)
print(list1)
```

Program Output

```
[25, 74, 19, 87]
```

list.extend(iterable)

Python List extend(iterable) method appends the items of the iterable x to the end of the list. Iterable could be a list, a dictionary, a set, tuple or any other iterable.

In the following example, we extend list with another list.

Example Program

```
list1 = [25, 74, 19]
list2 = [52, 68]
list1.extend(list2)
print(list1)
```

Program Output

```
[25, 74, 19, 52, 68]
```

In the following example, we extend list with a tuple.

Example Program

```
list1 = [25, 74, 19]
tuple1 = (52, 68)
list1.extend(tuple1)
print(list1)
```

Program Output

```
[25, 74, 19, 52, 68]
```

In the following example, we extend list with a set.

Example Program

```
list1 = [25, 74, 19]
set1 = {52, 68}
list1.extend(set1)
print(list1)
```

Program Output

```
[25, 74, 19, 52, 68]
```

We can iterate over dictionary keys or values. In the following example, we extend list with dictionary values. Similarly, you can extend a list with dictionary keys.

Example Program

```
list1 = [25, 74, 19]
dictionary = {"a": 52, "b": 68}
list1.extend(dictionary.values())
print(list1)
```

Program Output

```
[25, 74, 19, 52, 68]
```

list.insert(i, x)

Python List insert(i, x) method inserts the item x at index specified by the x. Items from the index i are shifted to right by one position.

In the following program, we take a list with some numbers and then insert a new number at index 2.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
list1.insert(2, 88)
print(list1)
```

Program Output

```
[25, 74, 88, 19, 52, 68, 34]
```

list.remove(x)

Python List remove(x) method removes the first occurrence of item x in the list.

In the following program, we will remove an item from the list.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
list1.remove(19)
print(list1)
```

Program Output

```
[25, 74, 52, 68, 34]
```

If item x is not present in the list, you will get ValueError, as shown in the following example.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
list1.remove(88)
print(list1)
```

Program Output

```
Traceback (most recent call last):
  File "d:/workspace/python/example.py", line 2, in <module>
    list1.remove(88)
ValueError: list.remove(x): x not in list
```

list.pop(i)

Python List pop(i) method removes the item present at index i and returns the item. All the items after index i are shifted to left by one position.

In the following program, we will remove the item present at index 2.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
x = list1.pop(2)
print(x)
print(list1)
```

Program Output

```
[25, 74, 52, 68, 34]
```

If no index is passed as argument to pop() method, pop() removes and returns the last item of the list.

In the following program, we will call pop() method on the list with no argument passed. Last item of the list should be removed.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
x = list1.pop()
print(x)
print(list1)
```

Program Output

```
34
[25, 74, 19, 52, 68]
```

list.clear()

Python List clear() method removes all the items from list and makes it empty.

In the following program, we will clear all the items of the list.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
list1.clear()
print(list1)
```

Program Output

```

```

list.index(x[, start[, end]])

Python List index(x[, start[, end]]) method returns the index of the first occurrence of item x in the list. You may also specify the start and end index positions to limit the search to that specific slice.

In the following program, we will find the index of item `52` in the list.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
index = list1.index(52)
print(index)
```

Program Output

```
3
```

list.count(x)

Python List count(x) method returns the number of occurrences of item x in the list.

In the following program, we will count the occurrences of `19` in the list.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34, 19]
n = list1.count(19)
print(n)
```

Program Output

```
2
```

list.sort(key=None, reverse=False)

Python List sort(key=None, reverse=False) method sorts the items in the list. Sorting happens in-place, meaning, original list is modified. By default, sorting happens in ascending order. But if `reverse=True`, sorting happens in descending order.

sort() returns None.

In the following example, we will take a list of numbers and sort them in ascending order.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
list1.sort()
print(list1)
```

Program Output

```
[19, 25, 34, 52, 68, 74]
```

In the following example, we will take a list of numbers and sort them in descending order. The only difference from the above program is that, we have to the pass the argument `reverse=True`.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
list1.sort(reverse=True)
print(list1)
```

Program Output

```
[74, 68, 52, 34, 25, 19]
```

list.reverse()

Python List reverse() method reverses the order of items. First item becomes last item, second item becomes last but one item, and so on.

reverse() performs the operation inplace and so, the original list is modified. reverse() returns None.

In the following example, we will take a list of numbers and reverse their order.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
list1.sort()
print(list1)
```

Program Output

```
[34, 68, 52, 19, 74, 25]
```

list.copy()

Python List copy() method returns a shallow copy of the list. Any modifications to the returned list does not affect the original list.

In the following example, we will take a list of numbers, copy it to another list, and modify the items in the second list. Original list should remain unchanged while the second list is modified.

Example Program

```
list1 = [25, 74, 19, 52, 68, 34]
list2 = list1.copy()
list2[0] = 88
print("list1 :", list1)
print("list2 :", list2)
```

Program Output

```
list1 : [25, 74, 19, 52, 68, 34]
list2 : [88, 74, 19, 52, 68, 34]
```

Conclusion

In this Python Tutorial, we learned about Python List Methods available and their usage with the help of example programs.

Python Programming

- Python Tutorial
- Install Python
- Install Anaconda Python
- Python HelloWorld Program
- Python Variables
- Python Variable Data Type Conversion
- Python Comments

Control Statements

- Python If
- Python If Else
- Python While Loop
- Python For Loop

Python String

- Python String Methods
- Python String Length
- Python String Replace
- Python Split String
- Python Count Occurrences of Sub-String
- Python Sort List of Strings

Functions

- Python Functions

Python Collections

- Python List
- Python Dictionary

Advanced

- Python Multithreading

Useful Resources

- Python Interview Questions