5.1. More on Lists

The list data type has some more methods. Here are all of the methods of list objects:

`list.`**`append`**(*x*)

Add an item to the end of the list; equivalent to `a[len(a):] = [x]`.

`list.`**`extend`**(*L*)

Extend the list by appending all the items in the given list; equivalent to `a[len(a):] = L`.

`list.`**`insert`**(*i*, *x*)

Insert an item at a given position. The first argument is the index of the element before which to insert, so `a.insert(0, x)` inserts at the front of the list, and `a.insert(len(a), x)` is equivalent to `a.append(x)`.

`list.`**`remove`**(*x*)

Remove the first item from the list whose value is *x*. It is an error if there is no such item.

`list.`**`pop`**([*i*])

Remove the item at the given position in the list, and return it. If no index is specified, `a.pop()` removes and returns the last item in the list. (The square brackets around the *i* in the method signature denote that the parameter is optional, not that you should type square brackets at that position. You will see this notation frequently in the Python Library Reference.)

`list.`**`index`**(*x*)

Return the index in the list of the first item whose value is *x*. It is an error if there is no such item.

`list.`**`count`**(*x*)

Return the number of times *x* appears in the list.

`list.`**`sort`**()

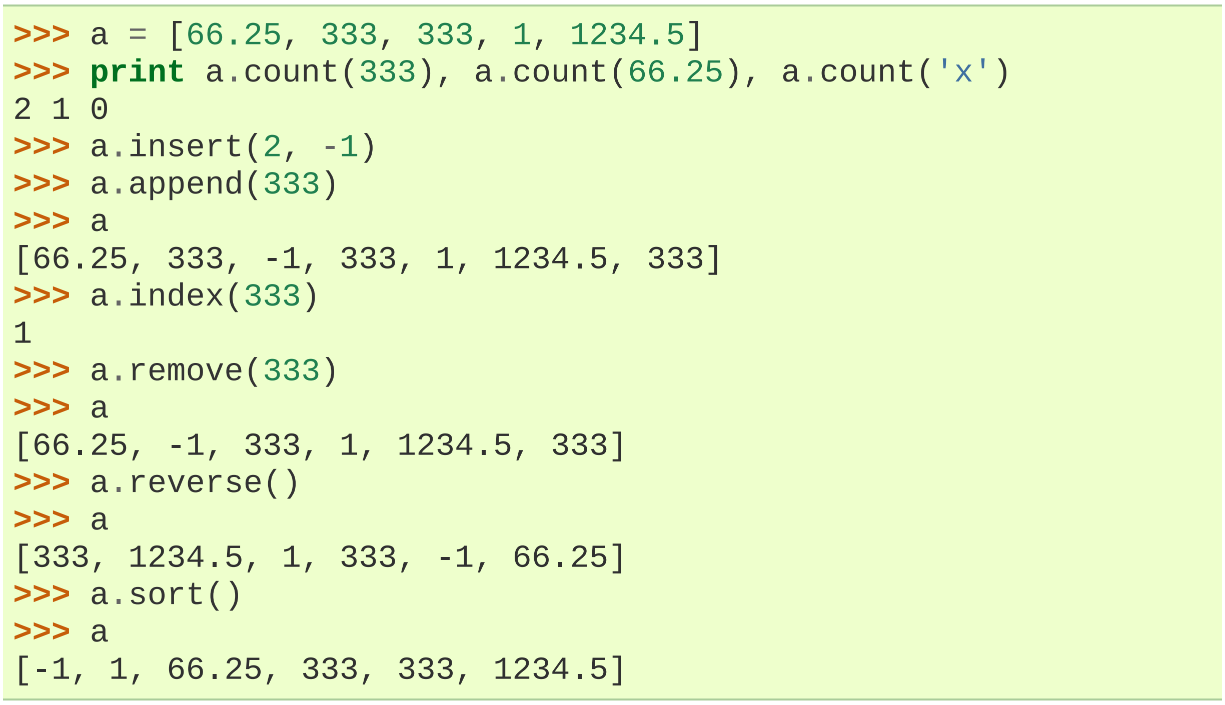
Sort the items of the list, in place.

`list.`**`reverse`**()

Reverse the elements of the list, in place.

An example that uses most of the list methods:

>>>



```
>>> a = [66.25, 333, 333, 1, 1234.5]
>>> print a.count(333), a.count(66.25), a.count('x')
2 1 0
>>> a.insert(2, -1)
>>> a.append(333)
>>> a
[66.25, 333, -1, 333, 1, 1234.5, 333]
>>> a.index(333)
1
>>> a.remove(333)
>>> a
[66.25, -1, 333, 1, 1234.5, 333]
>>> a.reverse()
>>> a
[333, 1234.5, 1, 333, -1, 66.25]
>>> a.sort()
>>> a
[-1, 1, 66.25, 333, 333, 1234.5]
```

http://docs.python.org/2/tutorial/datastructures.html