Lists

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.

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.

```
>>> 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]
```

Dictionary

c = dict(zip(['one', 'two', 'three'], [1, 2, 3]))

max(mydict, key=mydict.get)
Returns the key of the maximum value in dictionary

mydict[key] = mydict.**setdefault**(key, 0) + 1
If item in dictionary, increment it, otherwise, set it to 1

len(d)

del d[key]
Remove d[key] from *d*. Raises a **KeyError** if *key* is not in the map.

key not in d
clear()
Remove all items from the dictionary.

get(*key*[, *default*])¶
Return the value for *key* if *key* is in the dictionary, else *default*. If *default* is not given, it defaults to **None**, so that this method never raises a **KeyError**.

has_key(*key*)¶
Test for the presence of *key* in the dictionary. **has_key()** is deprecated in favor of key in d.

items()¶
Return a copy of the dictionary's list of (key, value) pairs.

keys()¶
Return a copy of the dictionary's list of keys. See the note for **dict.items()**.

pop(*key*[, *default*])¶
If *key* is in the dictionary, remove it and return its value, else return *default*. If *default* is not given and *key* is not in the dictionary, a **KeyError** is raised.

popitem()¶
Remove and return an arbitrary (key, value) pair from the dictionary.

setdefault(*key*[, *default*])¶
If *key* is in the dictionary, return its value. If not, insert *key* with a value of *default* and return *default*. *default* defaults to **None**.

Strings

ord('a') - > 97 chr(97) -> 'a'