Some Python list methods

In the “Python: Introduction for Programmers” course we describe just a few methods of lists. This more complete document is for reference and interest; you do not need to memorise these for the course.

These methods return a value and do not change the list.

`count(`*value*`)` How many times does *value* appear in the list?

```
>>> numbers = [1, 2, 3, 1, 2, 3]
>>> numbers.count(2)
3
>>> numbers
[1, 2, 3, 1, 2, 3]
```

`index(`*value*`)` Where is the first place *value* appears in the list?

```
>>> numbers = [1, 2, 3, 1, 2, 3]
>>> numbers.index(2)
1
>>> numbers[1]
2
```

`index(`*value*`, `*start*`)` Where is the first place *value* appears in the list at or after *start*?

```
>>> numbers = [1, 2, 3, 1, 2, 3]
>>> numbers.index(2,1)
1
>>> numbers.index(2,2)
4
>>> numbers[4]
2
```

These methods change the list and do not return any value.

`append(`*value*`)` Stick a single value on the end of the list.

```
>>> numbers = [1, 2, 3, 1, 2, 3]
>>> numbers.append(4)
>>> numbers
[1, 2, 3, 1, 2, 3, 4]
```

`extend(`*list*`)` Stick several values on the end of the list.

```
>>> numbers = [1, 2, 3, 1, 2, 3]
>>> numbers.extend([5,6,7])
>>> numbers
[1, 2, 3, 1, 2, 3, 4, 5, 6, 7]
```

`remove(`*value*`)` Remove the first instance of a value from the list.

```
>>> numbers = [1, 2, 3, 1, 2, 3]
>>> numbers.remove(2)
>>> numbers
[1, 3, 1, 2, 3]
```

`insert(`*index*`, `*value*`)` Insert *value* so that it gets index *index* and move everything up one to make room.

```
>>> numbers = [1, 2, 3, 1, 2, 3]
>>> numbers.insert(3, 5)
>>> numbers
[1, 2, 3, 5, 1, 2, 3]
>>> numbers.insert(0, 6)
>>> numbers
[6, 1, 2, 3, 5, 1, 2, 3]
```

`reverse()` Reverse the order of the list's items.

```
>>> numbers = [1, 2, 3, 1, 2, 3]
>>> numbers.reverse()
>>> numbers
[3, 2, 1, 3, 2, 1]
```

`sort()` Sort the items in the list.

```
>>> numbers = [1, 2, 3, 1, 2, 3]
>>> numbers.sort()
>>> numbers
[1, 1, 2, 2, 3, 3]
```

This method, exceptionally returns a value (from the list) and changes the list itself.

`pop()` Removes the last item from the list and returns it.

```
>>> numbers = [1, 2, 3, 1, 2, 3]
>>> numbers.pop()
3
>>> numbers
[1, 2, 3, 1, 2]
```