



7.11. The Gory Details: Iterables

The general syntax of a for loop is:

```
for iter_var_name in some_seq:
    # code block line 1
    # code block line 2
    # ...
```

After the word `in`, there can be any Python expression that evaluates to a sequence. You have already seen iteration over strings and lists. A string is a sequence whose items are single characters. A list is a sequence whose items can be any kind of Python object.

Actually, the for loop is a little more general. It can iterate not just over strings and lists, but any kind of Python object that acts like a sequence. So far, strings, lists, and tuples are all we have seen.

We will also see later some Python objects that act like sequences for the purposes of iteration with a for loop. These include file objects and iterators.

In fact, we have actually already seen one of these without noticing, because it hardly matters. Technically, the `range` function doesn't actually return a list. That is, `range(3)` doesn't actually create the list `[0, 1, 2]`. It returns an object that acts just like the list `[0, 1, 2]`, when used in a for loop. The difference is that the numbers 0, 1, and 2 are produced as they are needed rather than all created in advance. This hardly matters when there are only three items. For `range(10000000)` it makes a little difference for how fast the program runs and how much memory is used. That's why the items are produced as needed rather than all produced in advance.

For the purposes of this book, however, the difference will rarely matter. You will be safe to think of the `range` function as if it returns a list object. Indeed, the Python interpreter that's built into the textbook cheats a little bit and makes the `range` function actually produce a list. When you run a native Python interpreter on your computer it won't cheat in that way. Still, when you run code of the form, `for x in range(y)` you will usually do just fine to think of `range(y)` returning a list.

Don't worry about understanding these details right now. The important point is that in the activecode window above, instead of `some_seq` you can have any Python expression that evaluates to a string, a list, or certain other Python objects that act like sequences for the purposes of use in for loops. It's just something to keep in mind for later, when we see some of those other Python objects that act like sequences but aren't quite.

You have attempted 1 of 1 activities on this page



✓ Completed. Well Done!

7.10. Naming Variables in For Loops">

Naming Variables in For Loops">

7.12. Keeping Track of Your Iterator Variable and Your Iterable">

