3

Lists and Hashes

As we saw from the previous chapter, there are three types of data: scalars, lists, and hashes. So far we've only been working with scalars – single numbers or strings. We've joined two single strings together to make one, converted one currency only, and held one number in a variable.

There are times, when we'll want to group together information or express correspondences between information. Just like the ingredients in a recipe or the pieces in a jigsaw, some things belong together in a natural sequence, for example, individual lines in a file, or the names of players in a squash ladder. In Perl, we represent these relationships in lists – series of scalars. They can be stored in another type of variable called an **array**, and we call each piece of data in the list an **element**.

Alternatively, some things are better expressed as a set of one-to-one correspondences. A phone book, for example, is a set of correspondences between addresses and phone numbers. In Perl, structures like the phone book are represented as a **hash**. Some people call them 'associative arrays' because they look a bit like arrays where each element is associated with another value. Most Perl programmers find that a bit too long-winded and just call them hashes.

In this chapter, we'll see how we build up lists and hashes and what we can do with them when we've got them. We'll also begin to look at some control structures, which will enable us to step through lists and arrays. As well as all this, we'll learn how to process data more than once without having to write out the relevant sections of our program again and again.

Lists

We're all familiar with lists from everyday life. Think about a shopping list, what properties does it have? First of all, it's a single thing, one piece of paper. Secondly, it's made up of a number of values. In the case of a shopping list, you might want to say that these values are actually strings – `"ketchup"`, `"peanut butter"`, `"ice cream"`, and so on. Finally, it's also ordered, which means that there's a first item and a last item.

Lists in Perl aren't actually that much different: They're counted as a single thing, but they're made up of a number of values. In Perl, these values are scalars, rather than purely strings. They're also stored in the order they were created.