Why would you use a tuple instead of a list? Basically because sometimes it is useful to use something that you know can never change. If you create a tuple with two elements inside, it will always have those two elements inside.

PYTHON MAPS WON'T HELP YOU FIND YOUR WAY

In Python, a *map* (also referred to as a *dict*, short for *dictionary*) is a collection of things, like lists and tuples. The difference between maps and lists or tuples is that each item in a map has a *key* and a corresponding *value*.

For example, say we have a list of people and their favorite sports. We could put this information into a Python list, with the person's name followed by their sport, like so:

```
>>> favorite_sports = ['Ralph Williams, Football',
                       'Michael Tippett, Basketball',
                       'Edward Elgar, Baseball',
                       'Rebecca Clarke, Netball',
                       'Ethel Smyth, Badminton',
                       'Frank Bridge, Rugby']
```

If I asked you what Rebecca Clarke's favorite sport is, you could skim through that list and find the answer is netball. But what if the list included 100 (or many more) people?

Now, if we store this same information in a map, with the person's name as the key and their favorite sport as the value, the Python code would look like this:

```
>>> favorite_sports = {'Ralph Williams' : 'Football',
                       'Michael Tippett' : 'Basketball',
                       'Edward Elgar' : 'Baseball',
                       'Rebecca Clarke' : 'Netball',
                       'Ethel Smyth' : 'Badminton',
                       'Frank Bridge' : 'Rugby'}
```

We use colons to separate each key from its value, and each key and value is surrounded by single quotes. Notice, too, that the items in a map are enclosed in braces ({}), not parentheses or square brackets.

The result is a map (each key maps to a particular value), as shown in Table 3-1.

Table 3-1: Keys Pointing to Values in a Map of Favorite Sports

Key	Value
Ralph Williams	Football
Michael Tippett	Basketball
Edward Elgar	Baseball
Rebecca Clarke	Netball
Ethel Smyth	Badminton
Frank Bridge	Rugby

Now, to get Rebecca Clarke's favorite sport, we access our map `favorite_sports` using her name as the key, like so:

```
>>> print(favorite_sports['Rebecca Clarke'])
Netball
```

And the answer is netball.

To delete a value in a map, use its key. For example, here's how to remove Ethel Smyth:

```
>>> del favorite_sports['Ethel Smyth']
>>> print(favorite_sports)
{'Rebecca Clarke': 'Netball', 'Michael Tippett': 'Basketball', 'Ralph
Williams': 'Football', 'Edward Elgar': 'Baseball', 'Frank Bridge':
'Rugby'}
```

To replace a value in a map, we also use its key:

```
>>> favorite_sports['Ralph Williams'] = 'Ice Hockey'
>>> print(favorite_sports)
{'Rebecca Clarke': 'Netball', 'Michael Tippett': 'Basketball', 'Ralph
Williams': 'Ice Hockey', 'Edward Elgar': 'Baseball', 'Frank Bridge':
'Rugby'}
```

We replace the favorite sport of `Football` with `Ice Hockey` by using the key `Ralph Williams`.

As you can see, working with maps is kind of like working with lists and tuples, except that you can't join maps with the plus operator (+). If you try to do that, you'll get an error message:

```
>>> favorite_sports = {'Rebecca Clarke': 'Netball',
                       'Michael Tippett': 'Basketball',
                       'Ralph Williams': 'Ice Hockey',
                       'Edward Elgar': 'Baseball',
                       'Frank Bridge': 'Rugby'}
>>> favorite_colors = {'Malcolm Warner' : 'Pink polka dots',
                       'James Baxter' : 'Orange stripes',
                       'Sue Lee' : 'Purple paisley'}
>>> favorite_sports + favorite_colors
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'dict' and 'dict'
```

Joining maps doesn't make sense to Python, so it just throws up its hands.

WHAT YOU LEARNED

In this chapter, you learned how Python uses strings to store text, and how it uses lists and tuples to handle multiple items. You saw that the items in lists can be changed, and that you can join one list to another list, but that the values in a tuple cannot change. You also learned how to use maps to store values with keys that identify them.

PROGRAMMING PUZZLES

The following are a few experiments you can try yourself. The answers can be found at *http://python-for-kids.com/*.

#1: FAVORITES

Make a list of your favorite hobbies and give the list the variable name `games`. Now make a list of your favorite foods and name the variable `foods`. Join the two lists and name the result `favorites`. Finally, print the variable `favorites`.