Variables and Assignment

Chapter 3

So far, whenever we have put a string or a number, the thing we put is gone. What I mean is, if we wanted to print something out twice, we would have to type it in twice:

```
puts '...you can say that again...'
puts '...you can say that again...'
```

```
...you can say that again...
...you can say that again...
```

It would be nice if we could just type it in once and then hang on to it... store it somewhere. Well, we can, of course otherwise, I wouldn't have brought it up!

To store the string in your computer's memory, we need to give the string a name. Programmers often refer to this process as assignment, and they call the names variables. This variable can be just about any sequence of letters and numbers, but the first character needs to be a lowercase letter. Let's try that last program again, but this time I will give the string the name myString (though I could just as well have named it str or myOwnLittleString or henryTheEighth).

```
myString = '...you can say that again...'
puts myString
puts myString
```

```
...you can say that again...
...you can say that again...
```

Whenever you tried to do something to myString, the program did it to '...you can say that again...' instead. You can think of the variable myString as "pointing to" the string '...you can say that again...'. Here's a slightly more interesting example:

```
name = 'Patricia Rosanna Jessica Mildred Oppenheimer'
puts 'My name is ' + name + '.'
puts 'Wow!  ' + name + ' is a really long name!'
```

```
My name is Patricia Rosanna Jessica Mildred Oppenheimer.
Wow!  Patricia Rosanna Jessica Mildred Oppenheimer is a really long name!
```

Also, just as we can assign an object to a variable, we can reassign a different object to that variable. (This is why we call them variables: because what they point to can vary.)

```
composer = 'Mozart'
puts composer + ' was "da bomb", in his day.'
```

```
Mozart was "da bomb", in his day.
```

```
composer = 'Beethoven'
puts 'But I prefer ' + composer + ', personally.'
```

```
But I prefer Beethoven, personally.
```

Of course, variables can point to any kind of object, not just strings:

```
var = 'just another ' + 'string'
puts var
```

```
just another string
```

```
var = 5 * (1 + 2)
puts var
```

```
15
```

In fact, variables can point to just about anything... except other variables. So what happens if we try?

```
var1 = 8
var2 = var1
puts var1
puts var2

puts ''

var1 = 'eight'
puts var1
puts var2
```

```
8
8

eight
8
```

So first, when we tried to point var2 to var1, it really pointed to 8 instead (just like var1 was pointing to). Then we had var1 point to 'eight', but since var2 was never really pointing at var1, it stays pointing at 8.

So now that we've got variables, numbers, and strings, let's learn how to mix them all up!