

Writing Scripts - Defaulted Arguments

In the previous example, it might be reasonable to assume that most of the time, the quantity argument is going to be 1. It would be convenient if messages that had a quantity of 1 didn't even have to write it. For example:

```
ring-up-item cash-register price 17.29.
```

If you run the expression above, a debugger will appear with the error message "Using an undefined value" and q selected. This is because the arguments to the message didn't give a value to q, so its value is undefined. This is the same error as using a local variable before you've assigned it a value.

Scripts can handle missing values rather than just causing an error: They can default the value of the argument. In essence, we want assign 1 to q if the message didn't give it a value in the first place. There is a form of assignment, `?=`, just for this case. It is called defaulting. Change the ring-up-item script to:

```
$ price p, quantity q.
```

```
$ cost.
```

```
q ?= 1.
```

```
cost := p * q.
```

```
sub-total := sub-total + cost.
```

Now, if argument quantity is part of the message, then q will be set by the message, otherwise it will be set to 1.

»

This is actually the only way that you can assign to the variables for arguments. Argument variables, like p and q, cannot be assigned to using `:=`. Remember: argument variables are assigned by the message, local variables are assigned by your script.

