Writing Scripts - Self

Now that cash-register can perform a number of useful actions, you can tie them all together. Add several buttons to the cash-register form view as follows:

Add a button called buy-milk with the script:

```
ring-up-item self price 1.59
```

Add a button called buy-eggs with the script:

```
ring-up-item self price 0.205 quantity 12
```

Add a button called buy-cheese with the script:

```
ring-up-item self price 5.45 quantity 0.5
```

buying half a pound of cheese at 5.45/pound

Add a button called grand-total with the script:

```
$ amt-off, tax.

amt-off := self discount on sub-total.

tax := (sub-total - amt-off) * 0.0725.

total := sub-total - amt-off + tax.

daily-total := daily-total + total.

sub-total := 0.
```

Notice than in all these scripts, the messages were sent to something called self, not cash-register. Self is a predeclared variable that has the value of the receiver of the message. Since tapping on these buttons causes the a message to be sent with cash-register as the receiver, self has the value of cash-register.

If you replaced every occurrence of self in these scripts with cash-register, then they would work exactly the same. Self is better because if you renamed cash-register, and the scripts named cash-register they would no longer work. When the scripts use self, they operate on the correct object, no matter what you name it. The same is true if you make instances of cash-register: if the scripts use cash-register, they always operate on the class. If they use self, the work correctly.

You cannot use self when you send messages to the cash-register from the workspace. This is because for scripts in the workspace, the value of self is the workspace, and the workspace doesn’t understand messages like ring-up-item or discount.