Svelte automatically updates the DOM when your component's state changes. Often, some parts of a component's state need to be computed from *other* parts (such as a `fullname` derived from a `firstname` and a `lastname`), and recomputed whenever they change.

For these, we have *reactive declarations*. They look like this:

```
let count = 0;
$: doubled = count * 2;
```

> *Don't worry if this looks a little alien. It's valid (if unconventional) JavaScript, which Svelte interprets to mean 're-run this code whenever any of the referenced values change'. Once you get used to it, there's no going back.*

Let's use `doubled` in our markup:

```
<p>{count} doubled is {doubled}</p>
```

Of course, you could just write `{count * 2}` in the markup instead — you don't have to use reactive values. Reactive values become particularly valuable when you need to reference them multiple times, or you have values that depend on *other* reactive values.