Creating and Reading Signals

Deborah Kurata

Developer | Content Creator | MVP | GDE

@deborahkurata | https://www.youtube.com/@deborah_kurata

Create a Signal

signal
constructor function

```
quantity = signal<number>(1);
```

Optional type:
string
number
array
object

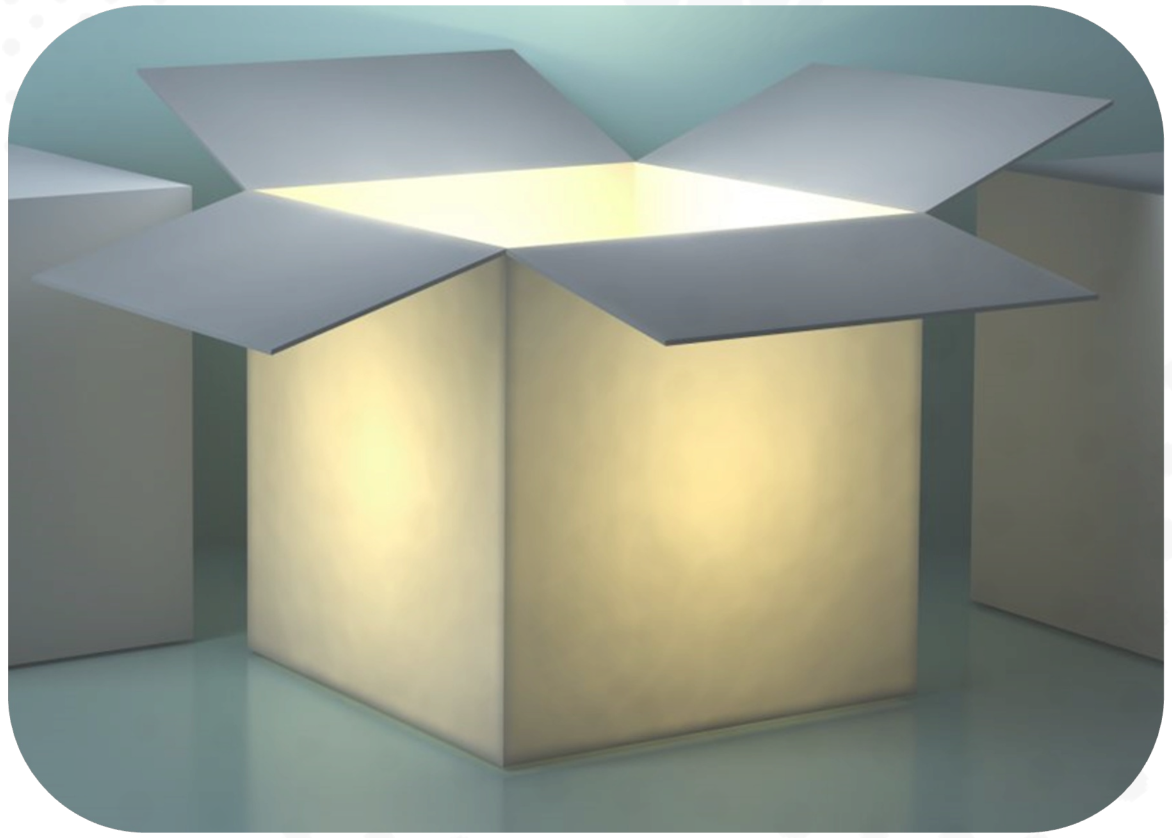
Required initial value

Read a Signal

signal name

```
quantity();
```

"open the box"



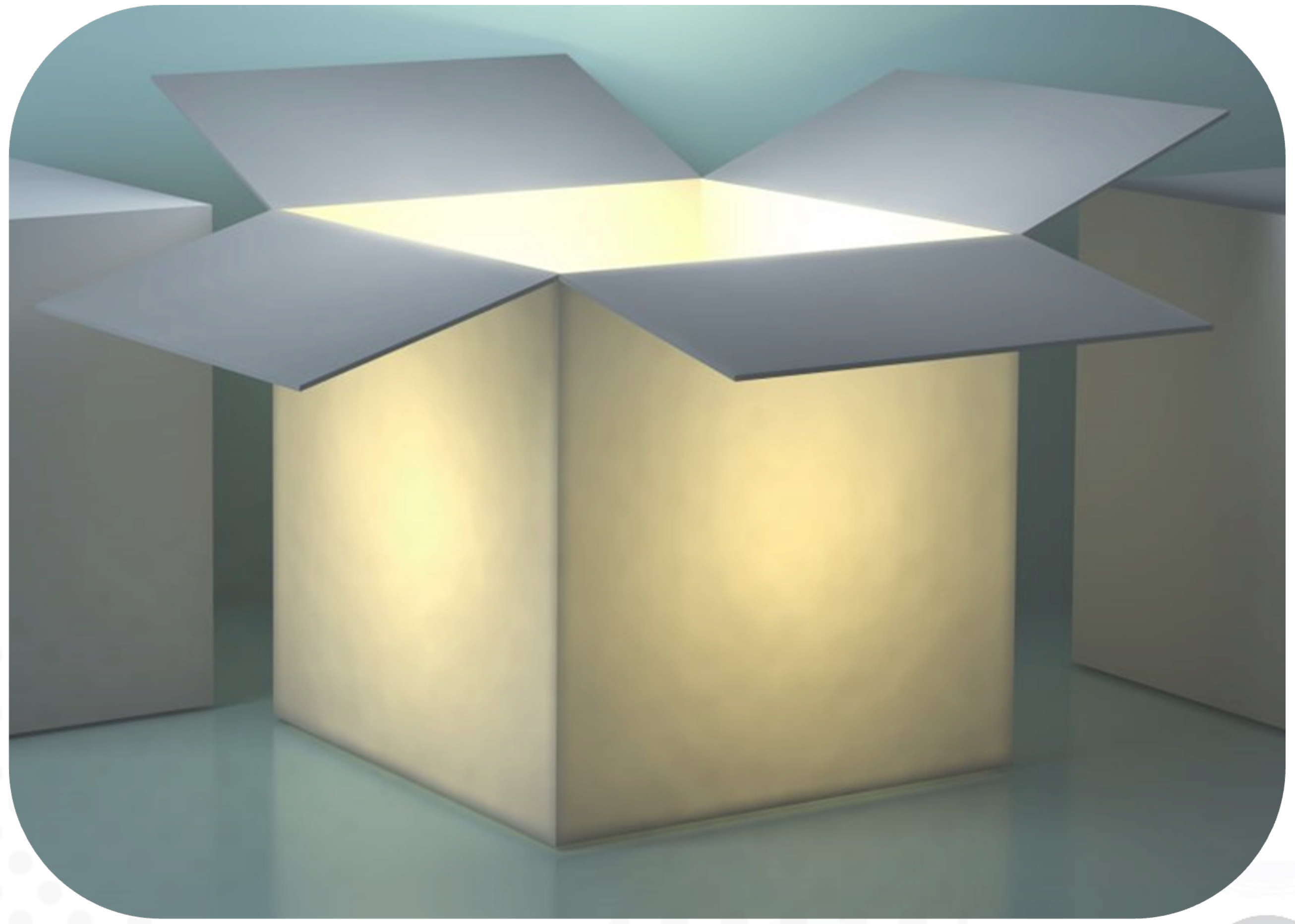
Reading a Signal

A signal only holds one value, the current value

Reading the signal reads the current value

Calls the signal's getter function

```
quantity();
```



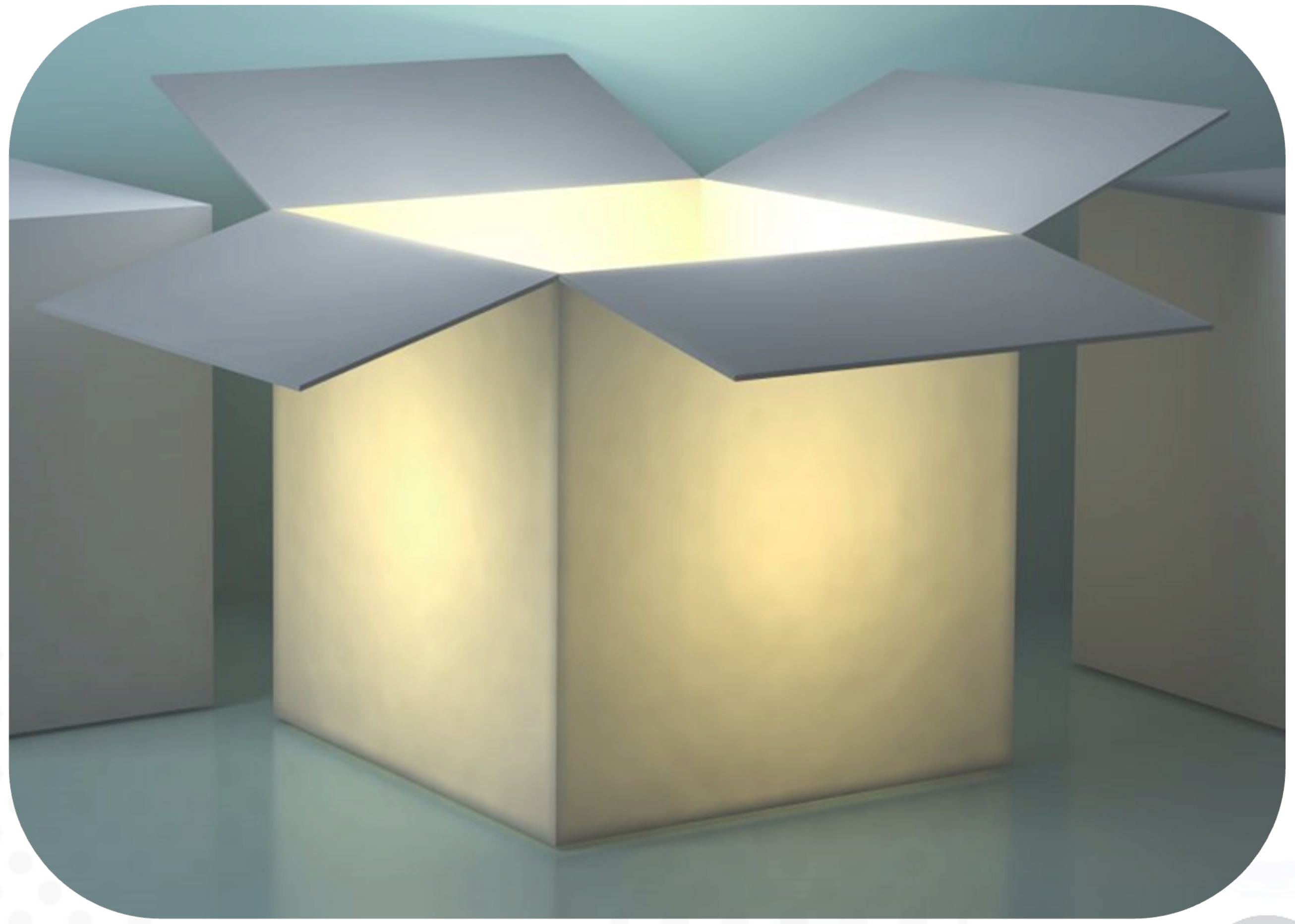
Reading a Signal in a Template

Displays the current value

Registers the signal as a dependency of the view

When the signal changes, the portion of the view is re-rendered

```
<div>
 {{ quantity() }}
</div>
```



Reading a Signal in a Reactive Function

Tracks each referenced signal as a dependency

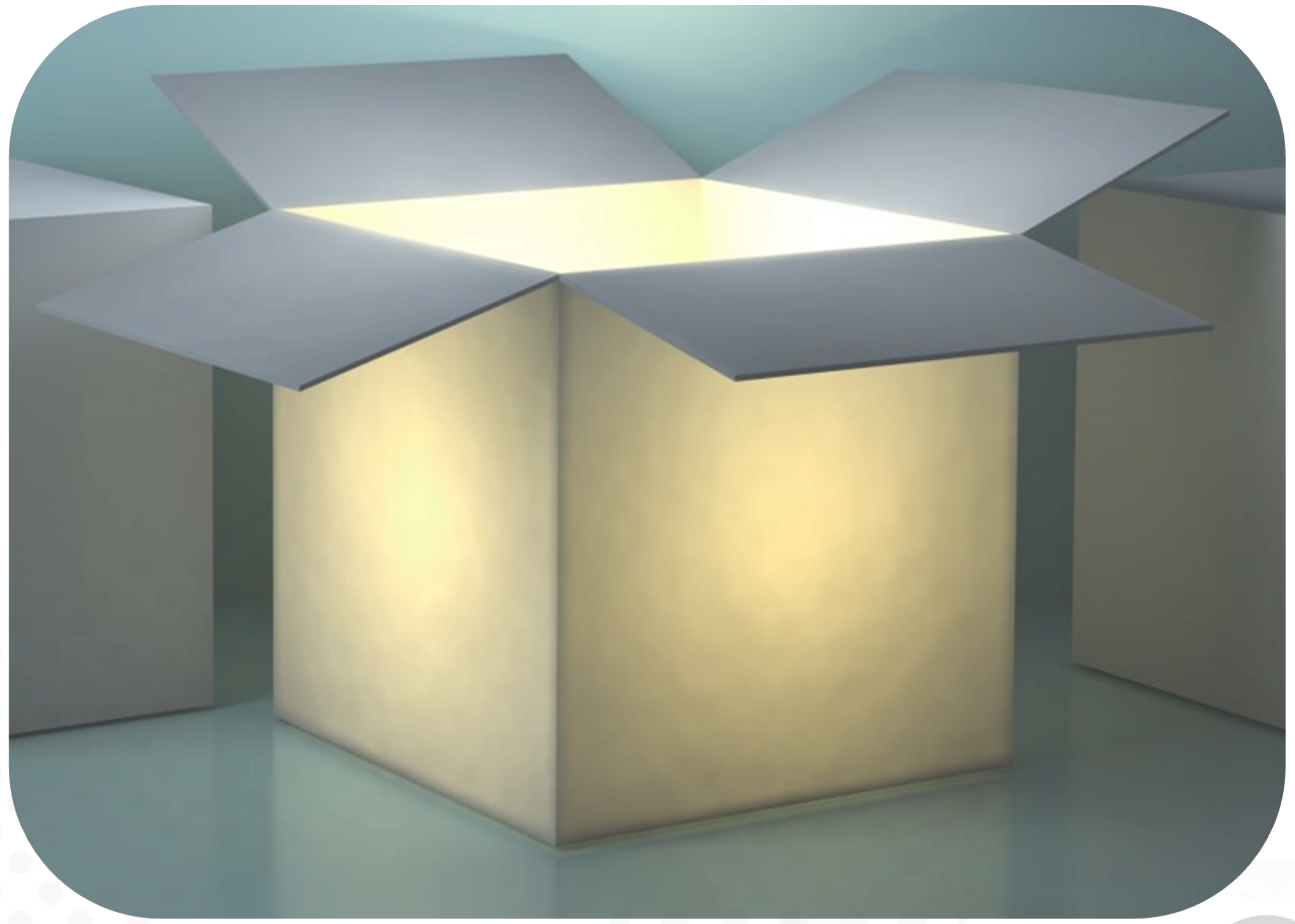
When a dependent signal changes, the function is re-executed

Computed value is memoized

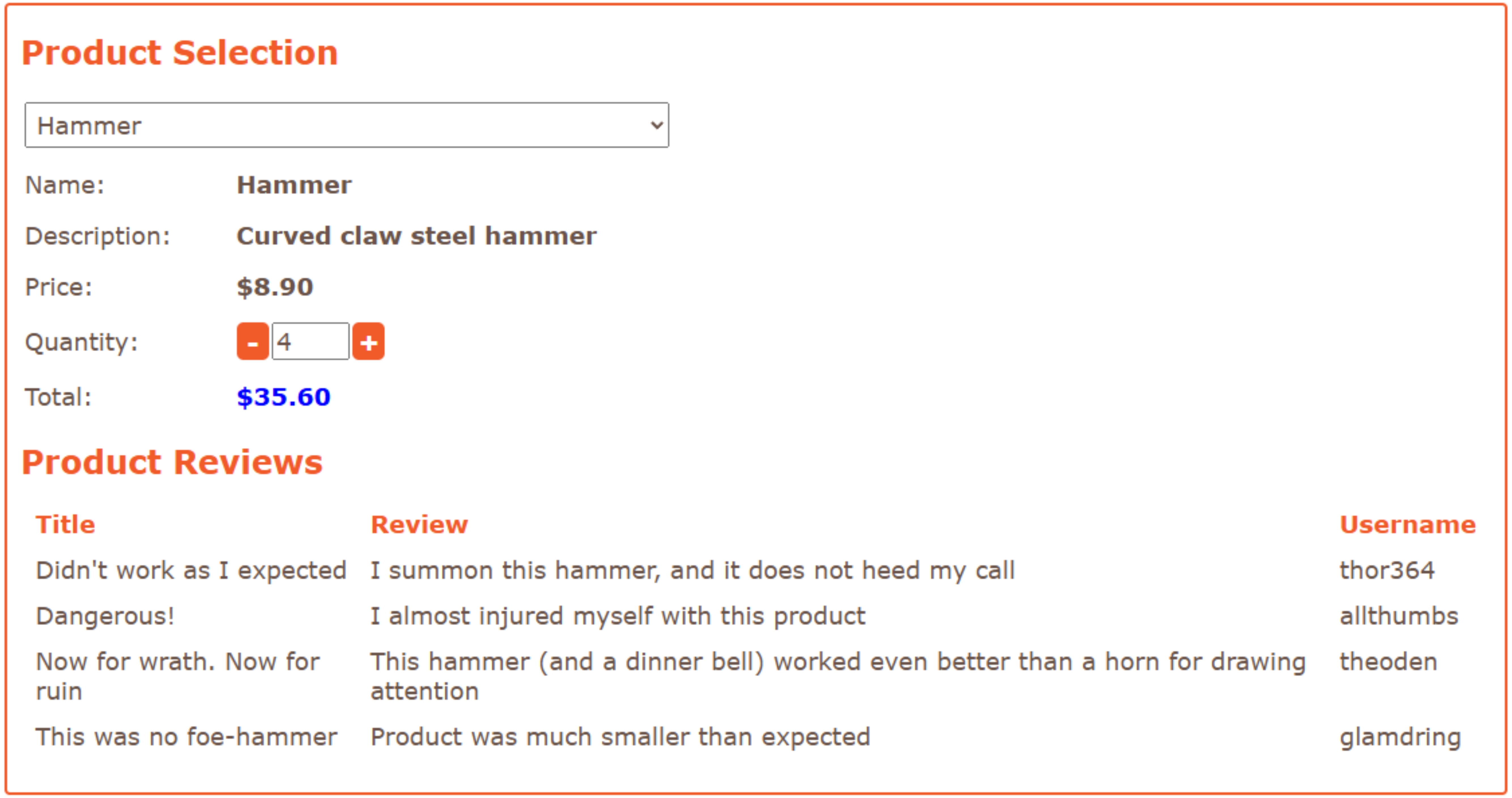
That result is reused

```
console.log(z());
console.log(z());
```

```
z = computed(()=>
        x() + y());
```



Sample Application



Product Selection

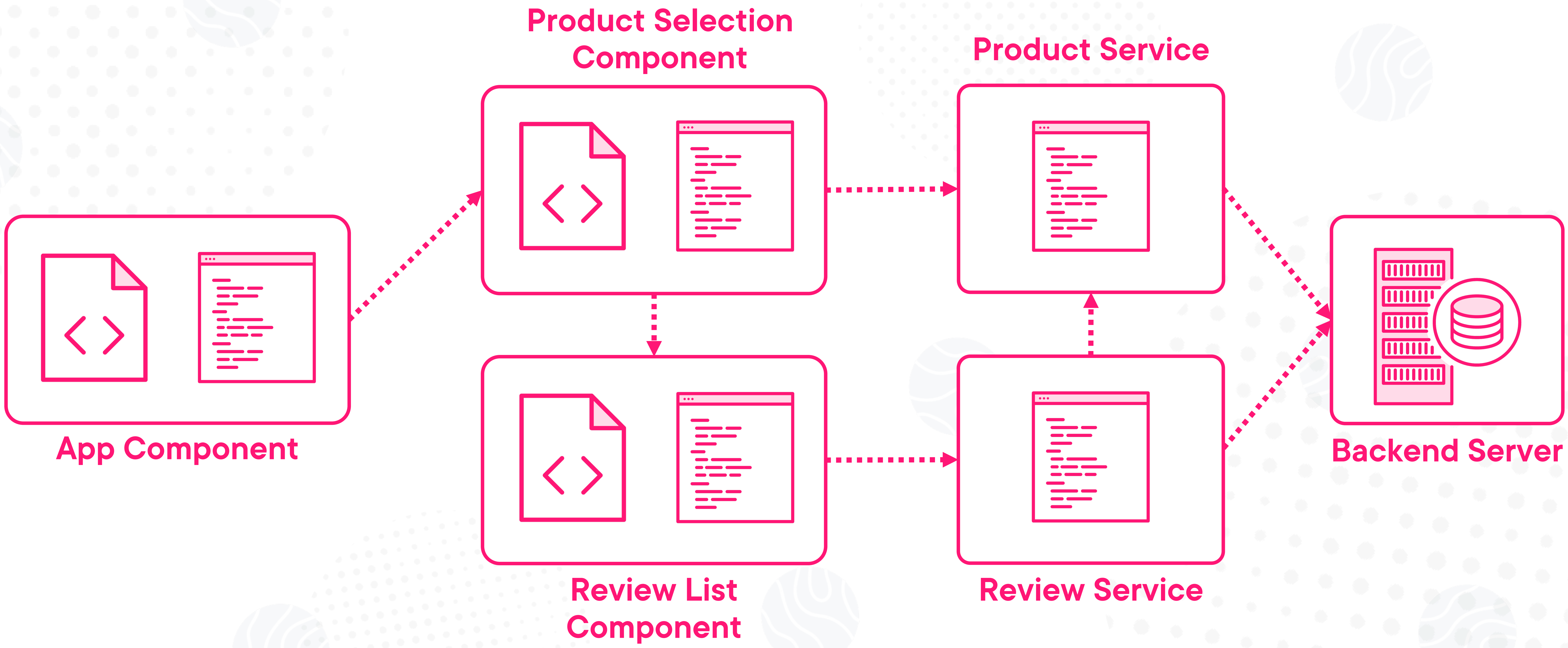
Hammer

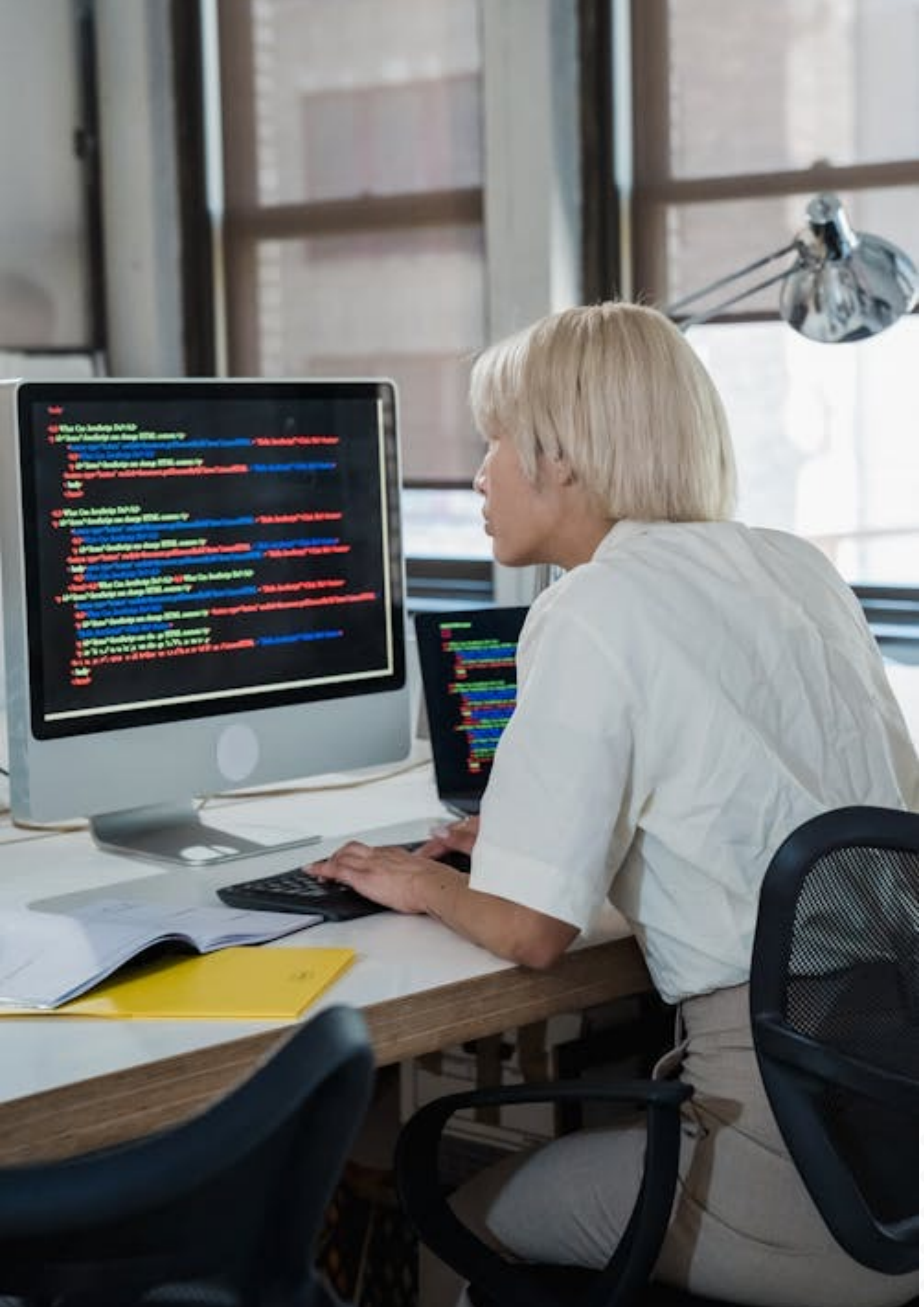
Name:	**Hammer**
Description:	**Curved claw steel hammer**
Price:	**$8.90**
Quantity:	- 4 +
Total:	**$35.60**

Product Reviews

Title	Review	Username
Didn't work as I expected	I summon this hammer, and it does not heed my call	thor364
Dangerous!	I almost injured myself with this product	allthumbs
Now for wrath. Now for ruin	This hammer (and a dinner bell) worked even better than a horn for drawing attention	theoden
This was no foe-hammer	Product was much smaller than expected	glamdring

Sample Application Structure





Debugging Tips

Use an effect to log a signal value

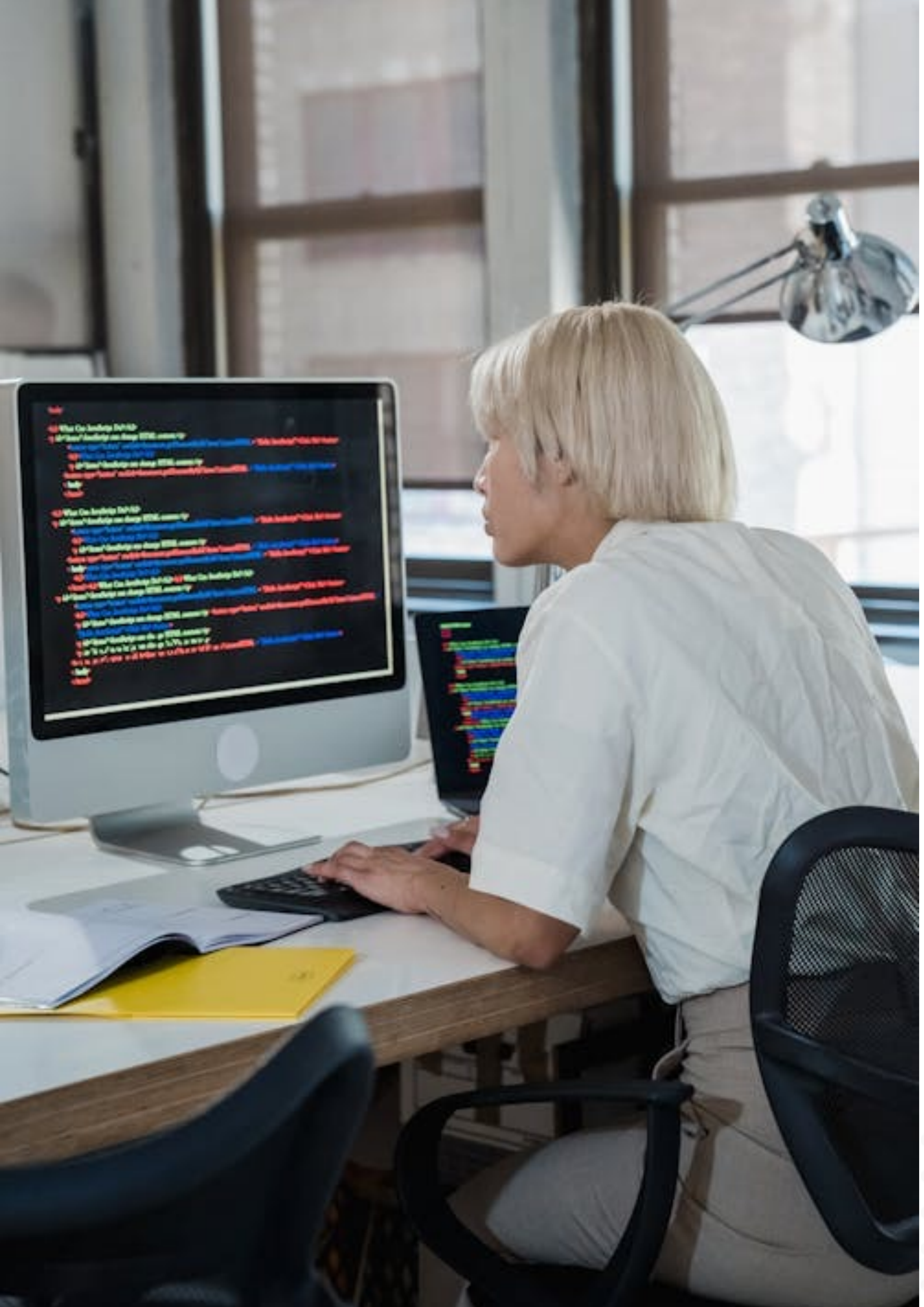
```
eff = effect(() => console.log(this.quantity()));
```

Don't reassign a signal

```
quantity = signal(1);
...
this.quantity = this.quantity() + 1;
```

Use `.set()` or `.update()` instead

```
quantity = signal(1);
...
this.quantity.set(2);
this.quantity.update(q => q + 1);
```



Debugging Tips (continued)

Notice when to use a signal vs. a signal value

Reference a signal:

- Two-way binding `[(ngModel)]='quantity'`
- Set `this.quantity.set(2);`
- Update `this.quantity.update(q => q + 1);`

Read a signal value:

- Interpolation `{{ quantity() }}`
- Logging `console.log(this.quantity());`
- Operations `() => price() * quantity();`

GitHub

`https://github.com/DeborahK/angular-signals-ps-course`

Beginning sample application files:

apm-begin

Final (completed) sample application files:

apm-end

File with clickable links to additional information:

MOREINFO.md