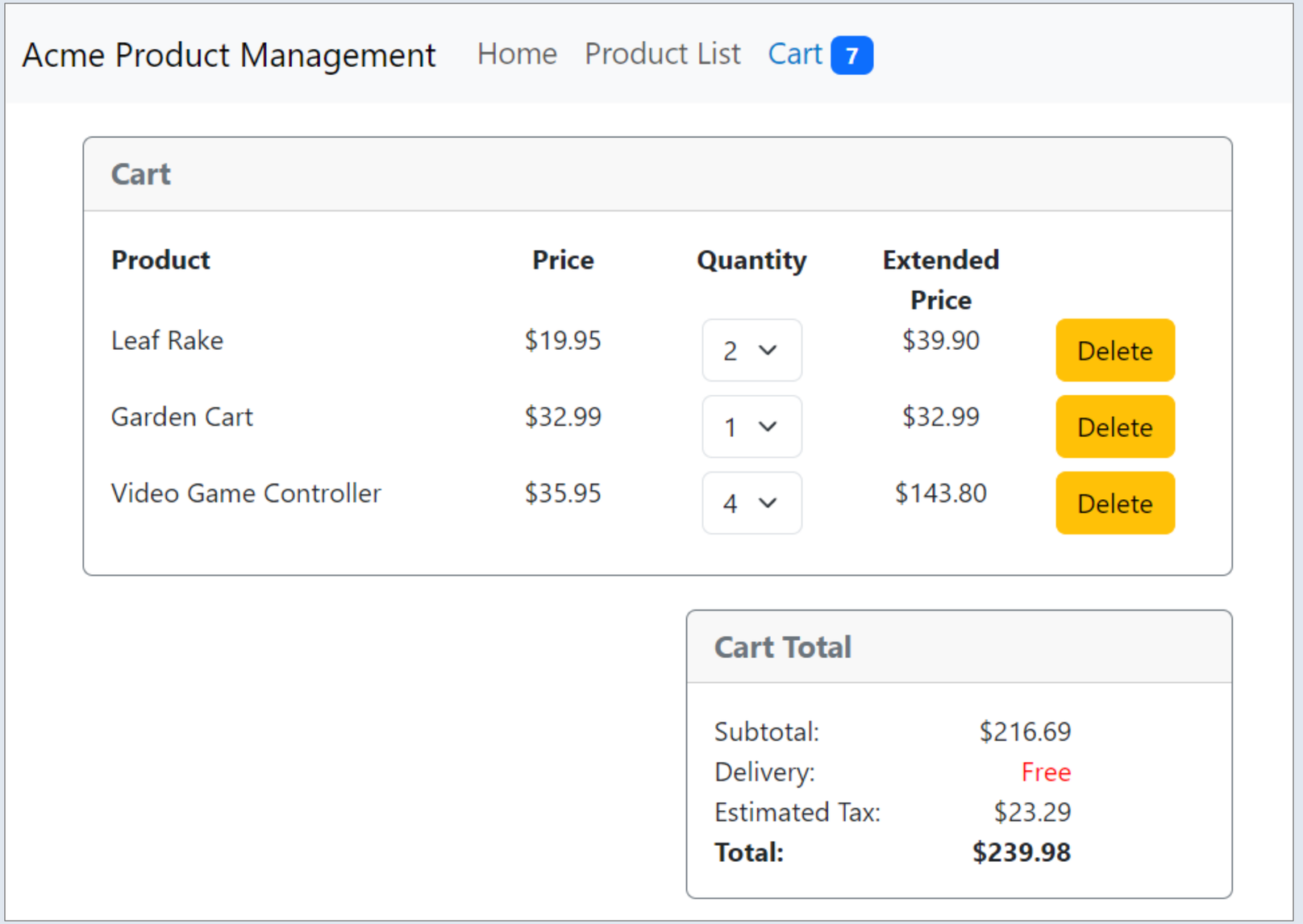
Using Signals to Build a Shopping Cart Feature

Deborah Kurata
Developer

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

Shopping Cart



Demo

Define the cart as a signal

Demo

Add products to the cart signal

Key Point

Always use one of the signal methods to modify a signal

```
// Replace the value
this.quantity.set(newQty);
```

```
// Update value based on current value
this.cartItems.update(items =>
        [...items, {product, quantity:1}]);
```

Why?

To ensure the signal is aware of the change

Demo

Declare a computed signal for the cart count

Implement the cart count badge

Demo

Declare computed signals for the cart totals

Display the totals

Demo

React to changes in the item quantity

React to changes when deleting an item

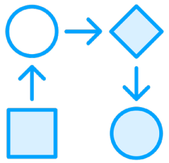
Demo

Define a signal for a cart item

Add a computed signal for the extended price

Benefits of Signals

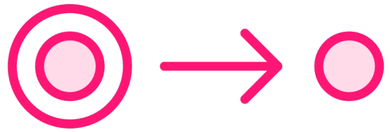
We built a shopping cart in moments



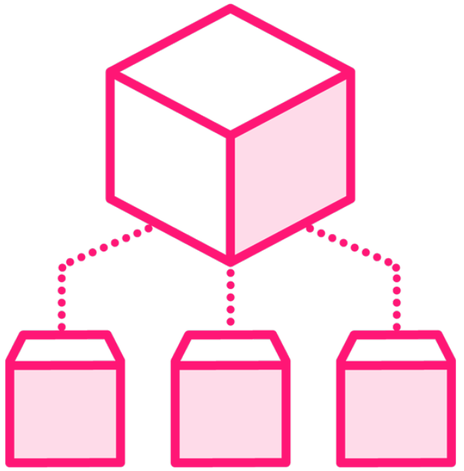
Build a fully reactive application

With a minimal amount of code

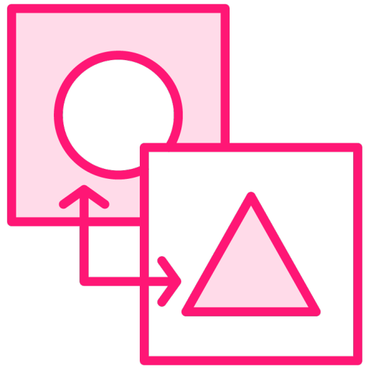
Best Practices



Use a signal method when updating a signal



Ensure computed signals reference a signal



Declare shared signals in services

For More Information

Demo code

- https://github.com/DeborahK/angular-rxjs-signals-fundamentals

"Manage State with Angular Signals"

- https://youtu.be/04avEeicarQ

Up Next:

RxJS and Angular Signals: Better Together