**Advanced Components**

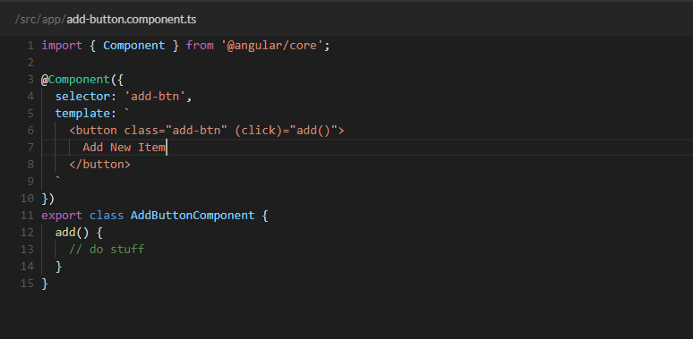
## Data projection in angular

Content projection allows you to insert a shadow DOM in your component. To put it simply, if you want to insert HTML elements or other components in a component, then you do that using the concept of content projection. In Angular, you achieve content projection using **< ng-content >< /ng-content >.**You can make reusable components and scalable applications by properly using content projection.

You use the **<ng-content></ng-content>** tag as a placeholder for that dynamic content, then when the template is parsed Angular will replace that placeholder tag with your content.

If you understand **{{myValue}}**, then you understand the basics of what ng-content does. The difference is where that value comes from. With normal curly brace interpolation the value comes from the component. With ng-content the value comes from the component **in its execution context.**

Let’s say you want to create a reusable button in your app.



Here we can see a generic add button which triggers an event when clicked. Nothing crazy here. The main thing I want to point out is the button’s text. “Add New Item" is hard-coded in the template. But, what if we wanted to get more specific with our button text? For example, “Add Coffee". We *could* put that value in the component like this:

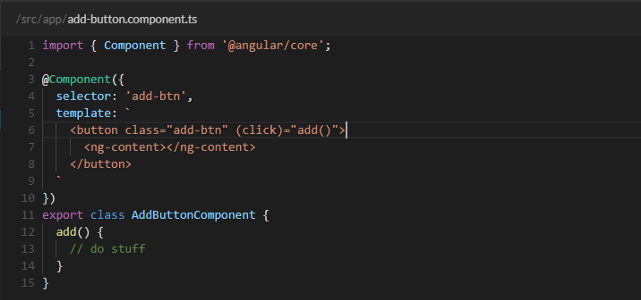


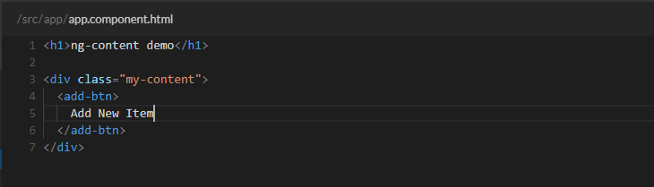
Or even as an Input from the parent component like this:





These ways work but this is where **ng-content** shines. Take a look at this:





See what’s happening here?

In the template for the reusable add button component we use the **<ng-content></ng-content>** tag as our placeholder for the button text.