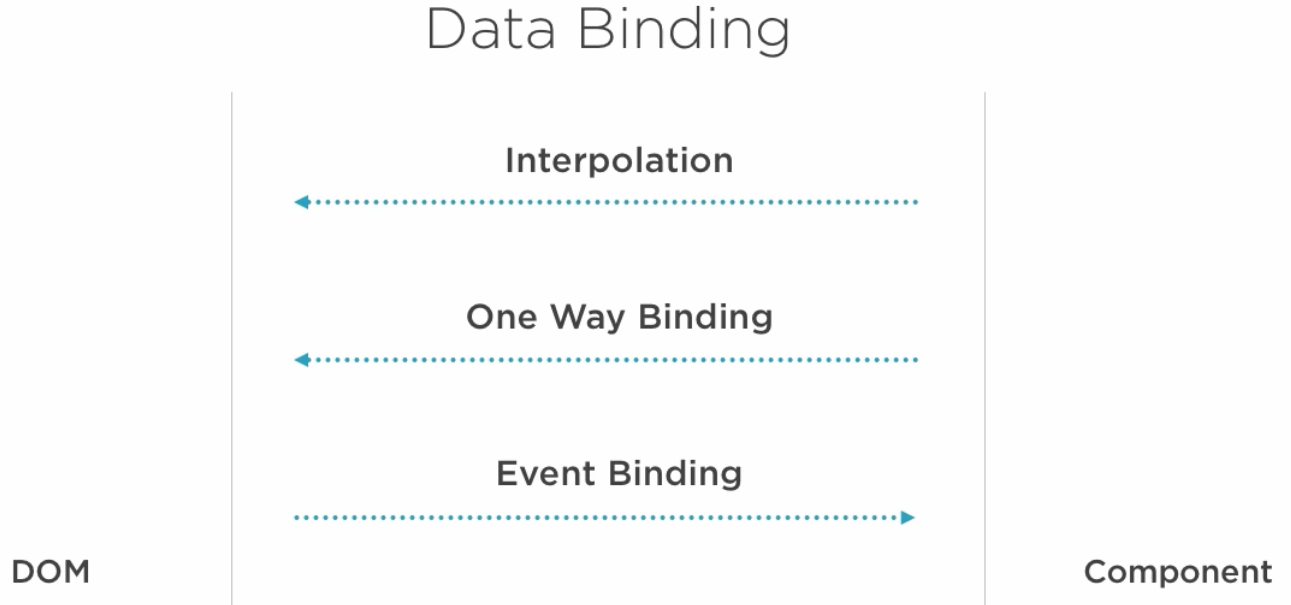
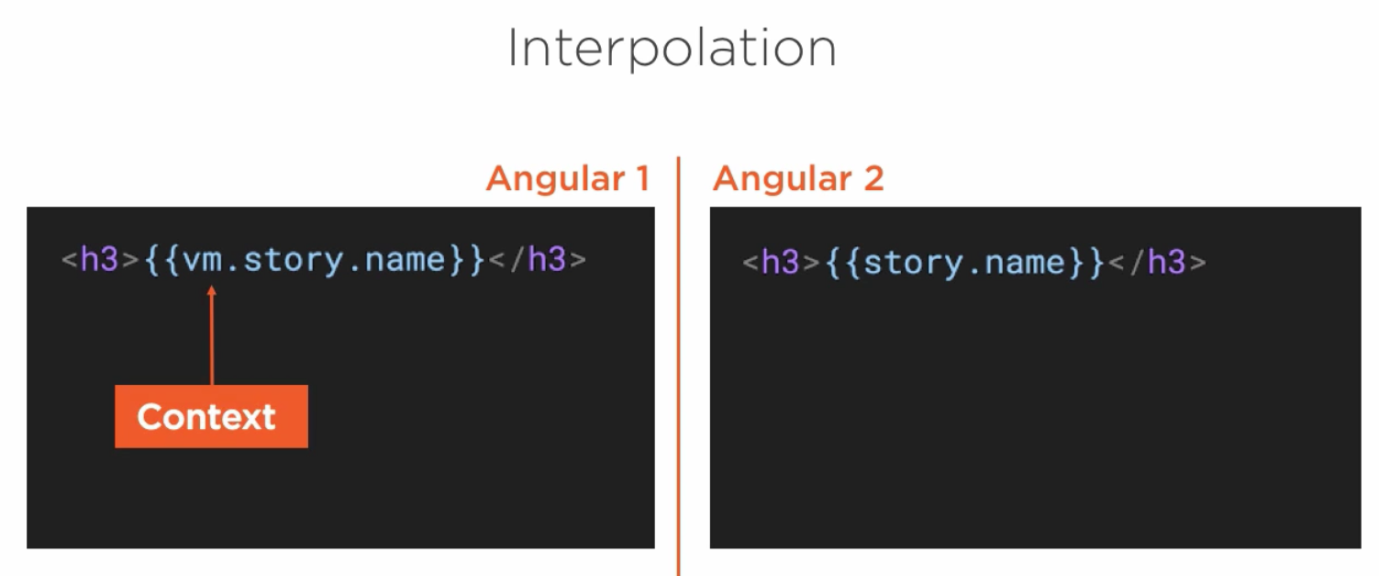
Data Binding

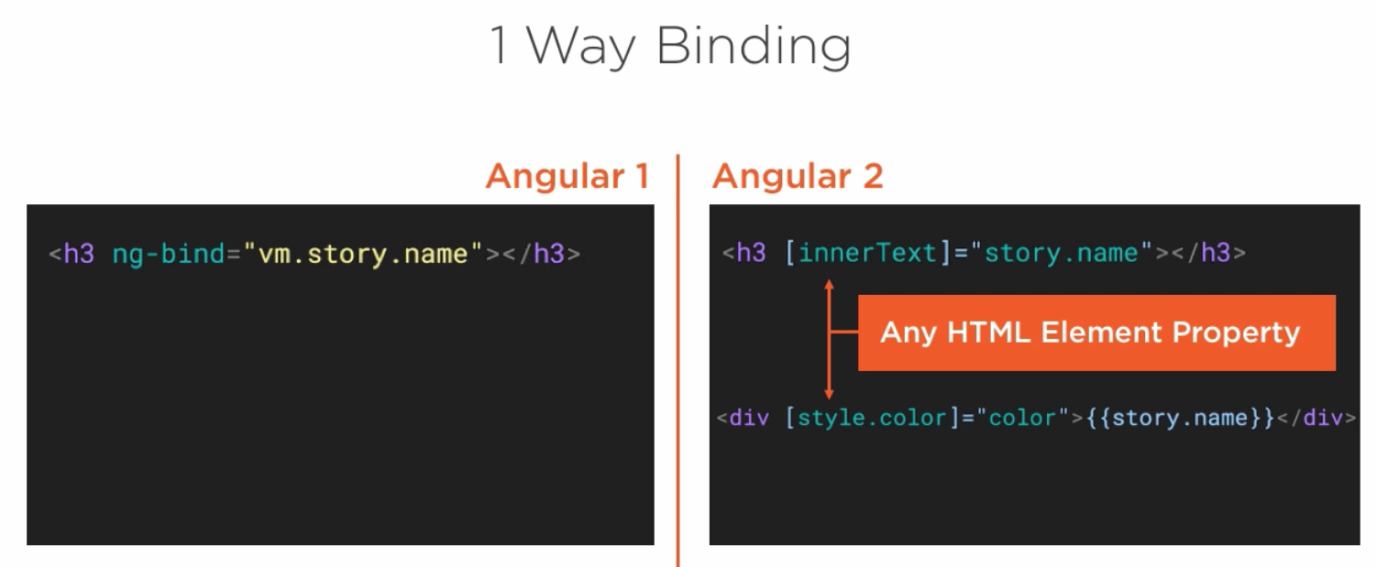
* Data binding. One of my favorite features from Angular 1 made its way into **Angular 2** in a **sense**. It behaves a **little** **bit** differently, but really let's think about what we actually want from data binding. We've got some data, in this case in a component in Angular 2, and then we want to get that data over to the DOM, which is our view. And there are a couple different ways we do that.



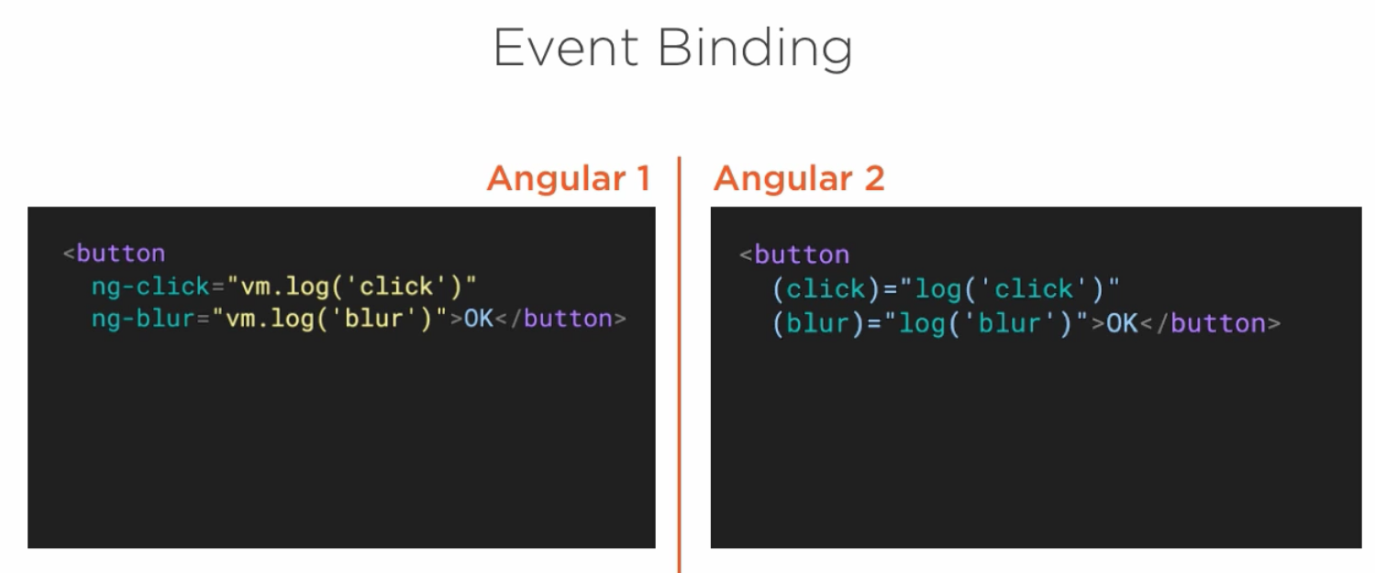
* In Angular 1 and Angular 2 we have all these options still. We have interpolation, that's the curly braces, that gets the data in read only form up to the DOM. Then we've got one way binding. We also have event binding, things like clicks. And then we have two way binding for things like the input element. Let's go through each of these.



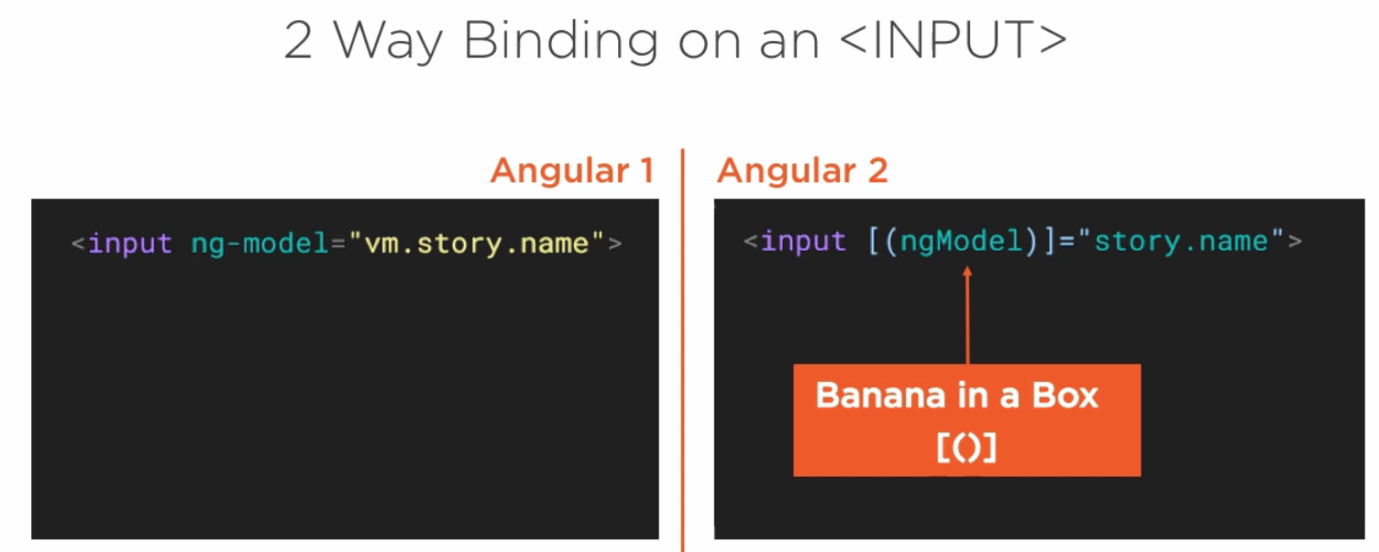
* Alright, in an **interpolation** with Angular 1, we simply just say, okay, I've got this vm.story.name, and I want to show that value here. The vm is the context from our **controller**. Angular 2, guess what? It's pretty much the same. We still use those curly braces, but instead of having to say vm, we've already got the context, so we just saved a little bit.



* Okay, that's interpolation. What about one way binding? In Angular 1, if we wanted to, we could do ng-bind. That's effectively what interpolation ended up being anyway.
* In Angular 2, we could use the square brackets around a property. Whoa, hold up. Is that valid html? Actually, yes it is. So now we can say take any HTML element property, like innerText, wrap it with the square braces, and then we can bind that to a model, story.name. Now of course I wouldn't do that with the h3 here, I'd probably just use the interpolation, but it's showing the point. What I might do instead is I might have a div where I want to set the color equal to maybe red or blue or yellow. So I might have a color model property, and I want to set that to be the style.color, so I wrap style.color with the square braces and set it to my property, which is on my component. And remember, I can do this with any HTML element property.



* Next up is event binding. These are things like clicks or on focus or blur. In Angular 1 we did ng-click or ng-blur. In Angular 2 we just take the same property that's on the HTML element, in this case it's the event called click or the event called blur, and we wrap it with parentheses. So properties on the previous one we use the square braces, and for events we use the parentheses. I want to reinforce that there are no click or blur directives in Angular 2. These are in fact HTML element events.



* Alright, the granddaddy of them all, the two way binding on an input. In Angular 1 we say input ng-model, and we set it to the model, and then we get our two way binding. We can type a name on one side in the view, and then it goes to the controller and then back and forth. In Angular 2 we have a special directive called ngModel, which we again bind to story.name. But notice we have got this syntax, which we wrap ngModel with the square brackets because it's property, but also with these parentheses. That's a special syntax that's known as Banana in a Box. No, I didn't make that up. And what this really means is when you see that syntax for an ngModel, that's going to give us two way binding. Effectively what's actually happening under the covers is that this is sugar syntax for saying you know what, we want to take that story name and bring it up into the view, that's the square brackets basically, and then we're going to use the parentheses to have an event of when I change that to go back down to the component.