

Angular for Beginners: Part 3 Data Binding in Angular.

By Tuba Mansoor

This is part 3 of the Angular for Beginners series. It is in continuation with the earlier parts mentioned below.

[Part 1](https://taagung.com/angular-tutorial-1/) of this series discussed how to get started with Angular.

[Part 2](https://taagung.com/angular-tutorial-2/) discussed about the basic tenets of Angular- Modules, Components and Directives.

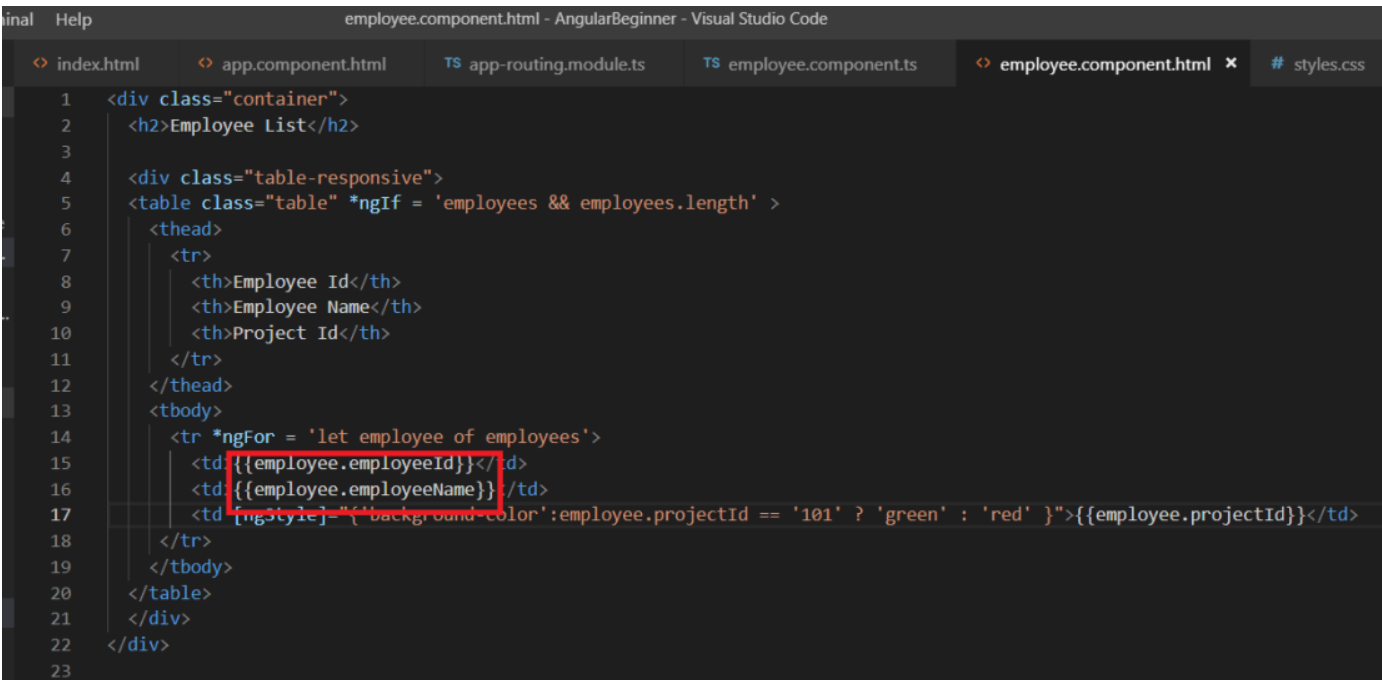
Part 3 here further builds up on the demo code created in earlier parts, so it is strongly advised that you go through them before proceeding.

# Data Binding in Angular

There are four types of binding in Angular:

## Interpolation

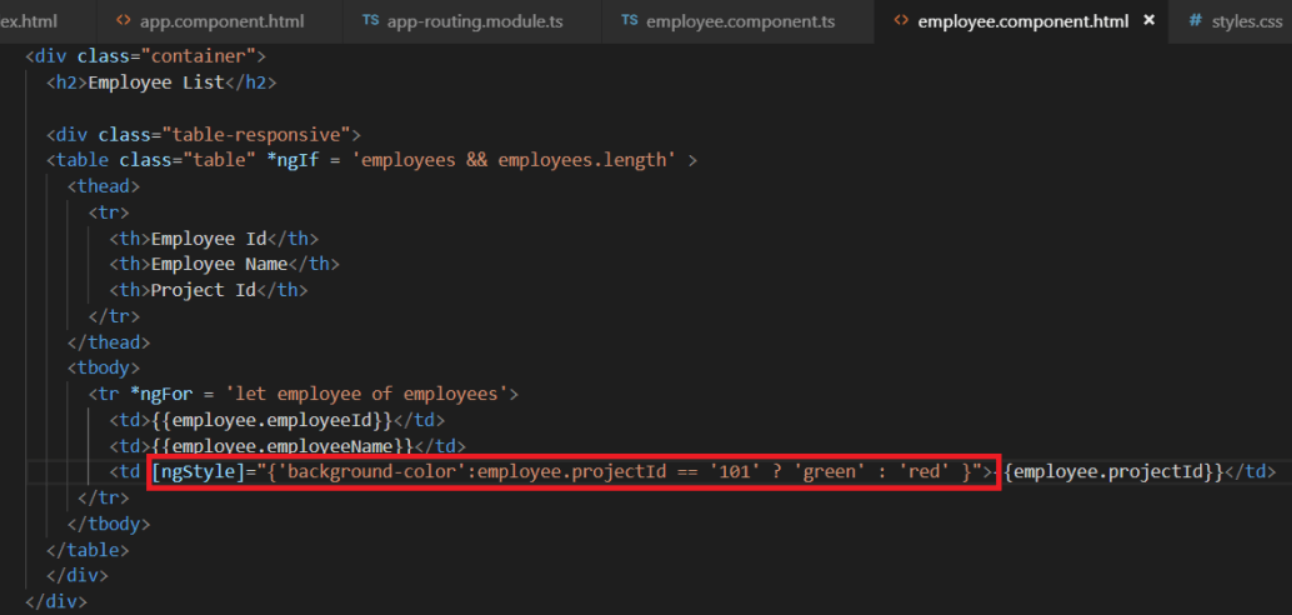
We have already used interpolation in our code, when we displayed the employee fields in our table.



It is denoted as {component.property}} and is used to pass value from the component to the DOM.

## Property Binding

We used property binding in our code, when we binded the property of the NgStyle element property to values from our component.ts file.



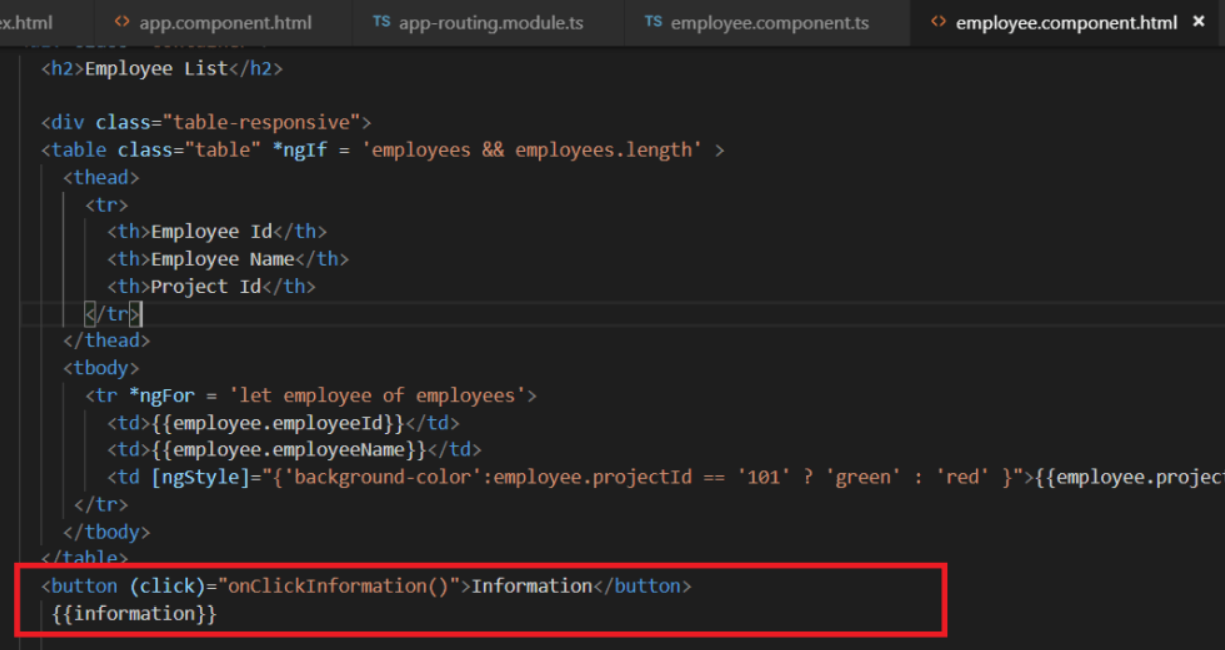
In property binding, element html property values in DOM are set from the component. The background of the row is set based on the component property value.

## Event Binding

The above two types are used to pass value from the component to the DOM. Event binding is used to pass value from the DOM to the component.

Let’s have a look at it in the code.

We will add a button in our html code as follows:



|  |  |
| --- | --- |
| 1  2 | <button (click)="onClickInformation()">Information</button>  {{information}} |

On button click we are calling a function onClickInformation(). This function is defined in the component file.

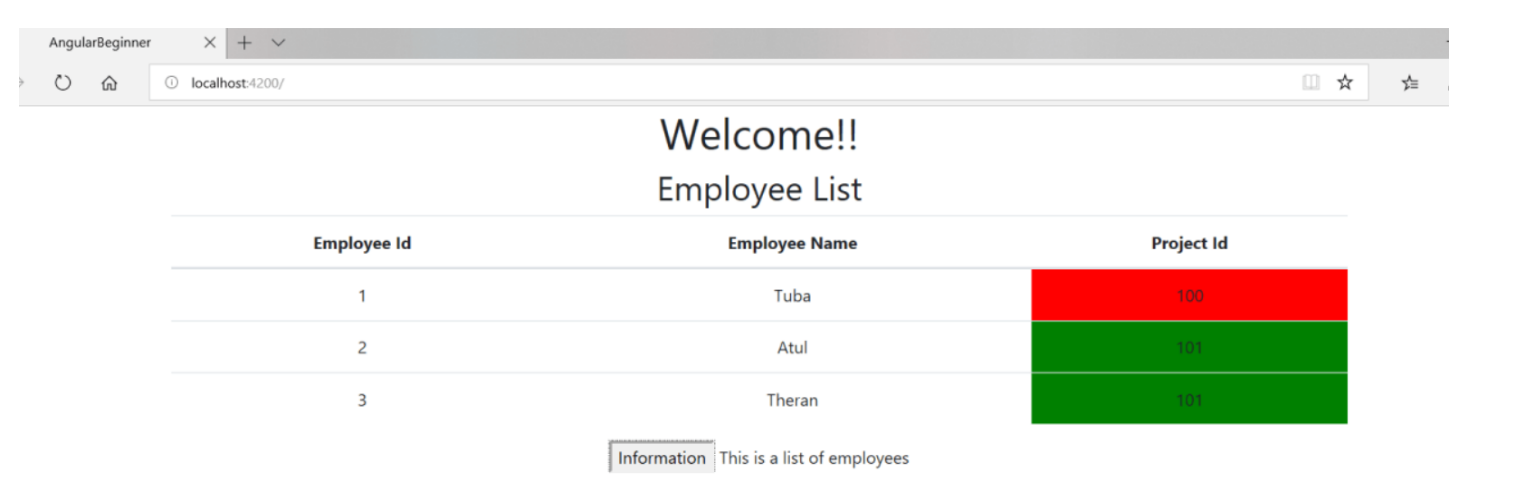
Let’s define this function in the component file.



|  |  |
| --- | --- |
| 1  2 | information = '';    onClickInformation() {    this.information = 'This is a list of employees';  } |

Here through event binding we are calling the onClickInformation function in the component. Initially the value of the ‘information’ variable is set to empty. On click of the button, some value is set in the OnClickInformation function. This value is then displayed in DOM through interpolation.

So, on button click, we can see the following in the browser.

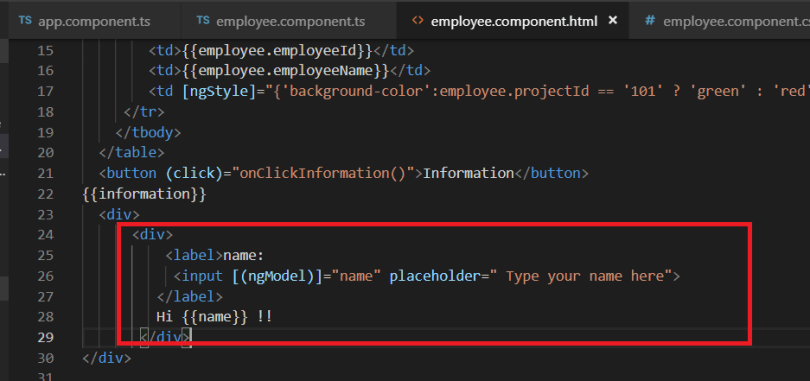


# Two way data binding

As the name suggests, this provides binding from component to DOM and vice versa. Let’s see this in action. Two way data binding uses ngModel.

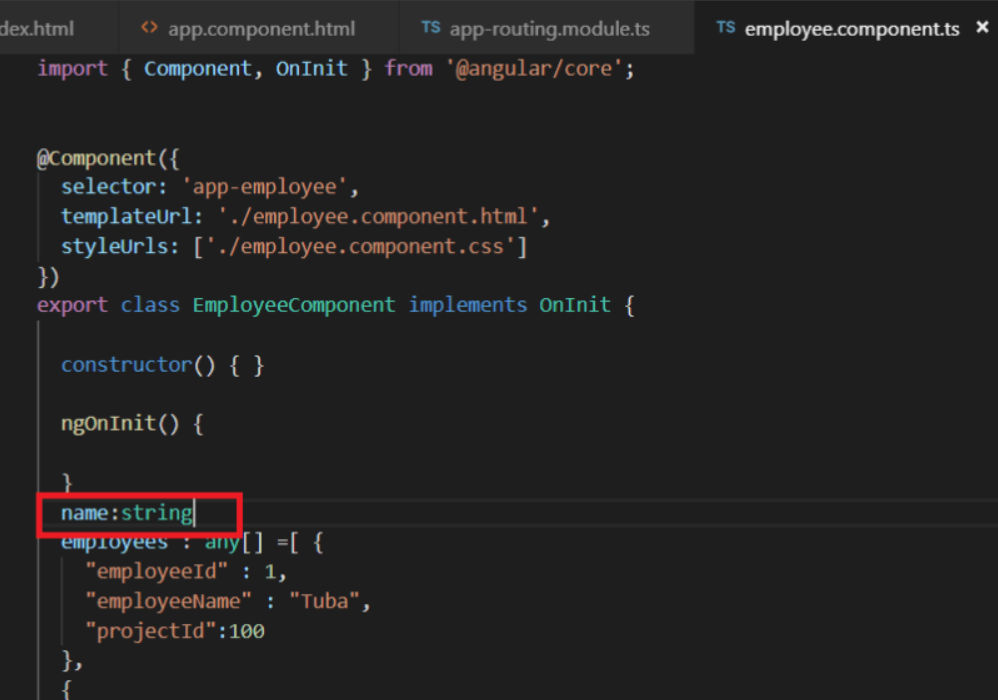
Type in the following piece of code into your html file:

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | <div>     <label>name:      <input [(ngModel)]="name" placeholder=" Type your name here">    </label>    Hi {{name}} !!  </div> |

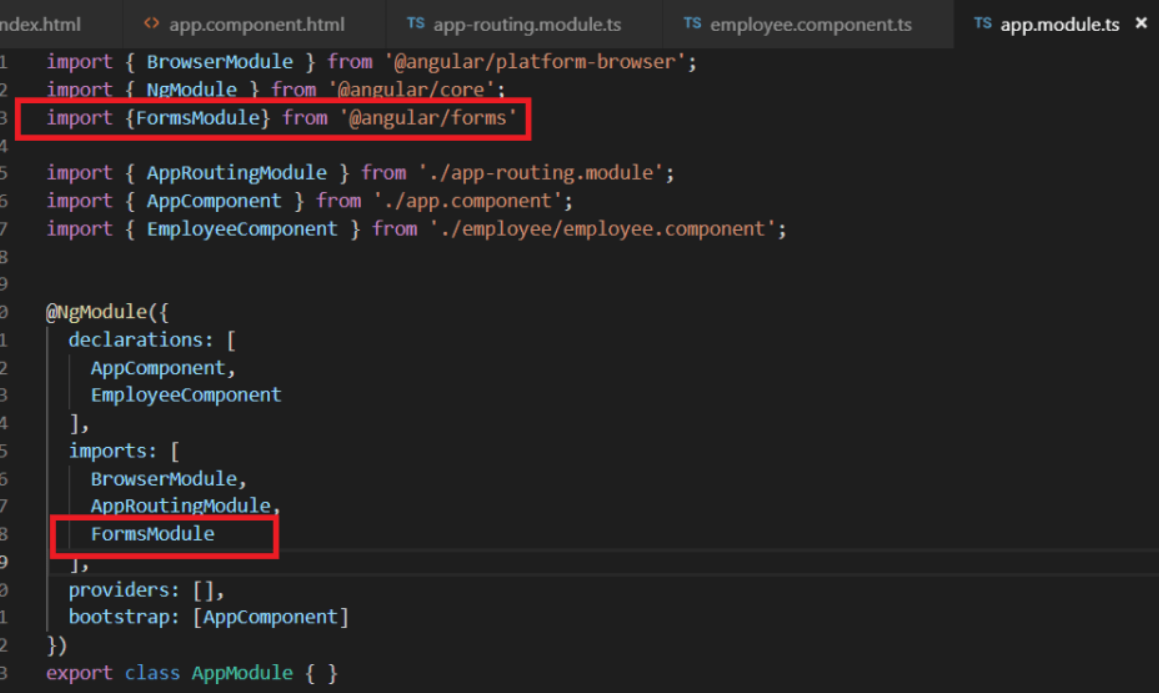


The name property is defined in the component as follows:

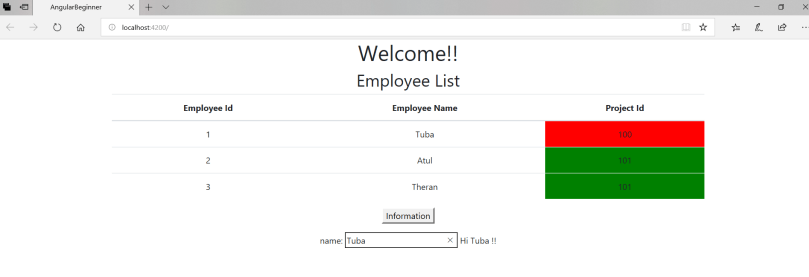
|  |  |
| --- | --- |
| 1 | name:string |



To ngModel directive that we just used, is present in the formsModule. So we need to import it in our app.module.ts file as below:



Let’s go to our browser and see what happens:



You will find that whatever you write in the textbox is reflected next to it.

So here we have a string property ‘name’ that is defined in the component file. Its value is set in the DOM equal to the value in the textbox. So, value is passed from DOM to component. At the same time its value is passed back from component to DOM through interpolation and you can see it next to it. So, the value in the view and the model is same at all times. This is two way binding.

It is denoted as [()] notation which is also known as banana in a box notation.

The [next part](https://taagung.com/angular-tutorial-4/) in this series talks about using interfaces in Angular. Happy Learning!

This article was originally published on my website [taagung](https://taagung.com/angular-tutorial-3/).