Angular

AJ

Property or Attribute Binding

```
<p class="card-text" [hidden]="true">{{joke.punchline}}</p>
```

A few things:

1. We wrapped the attribute with a [], more on that later.
2. We made the attribute equal to true, if we made it equal to false it shows the element.

We say we have bound the value true to the property called hidden.

> Important
>
> This is called **Input Property Binding** and it's a very important concept in Angular.

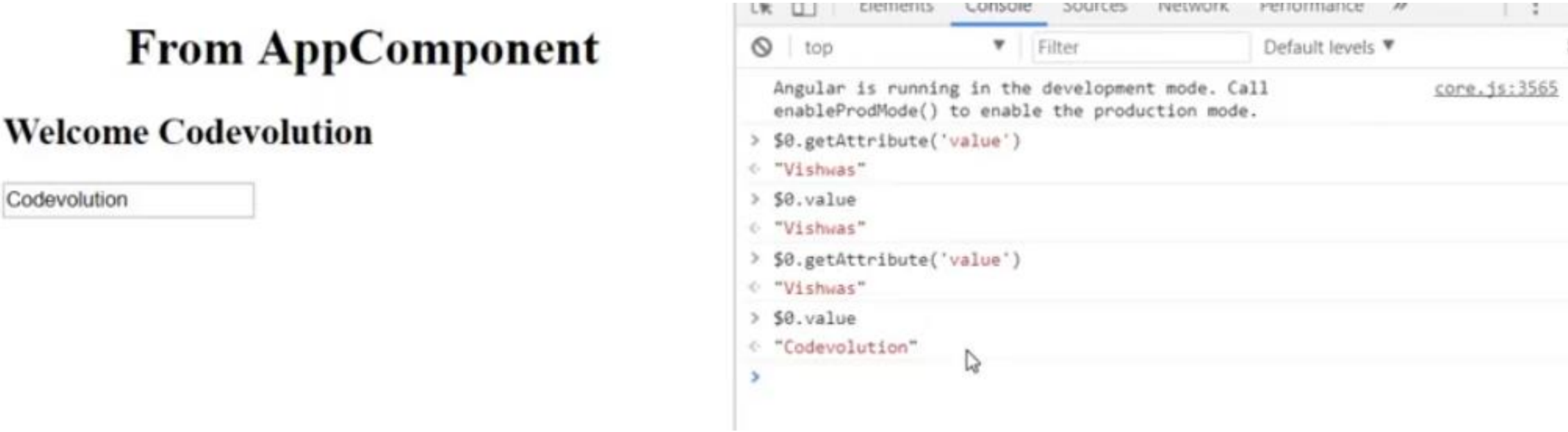
```
<input [id]="myId" type="text" value="Vishwas">
<input id="{{myId}}" type="text" value="Vishwas">

`,
styles: []
)
xport class TestComponent implements OnInit {

public name = "Codevolution";
public myId = "testId";
```

Property Binding

```
<p class="card-text" [hidden]="true">{{joke.punchline}}</p>
```

The *target* inside `[]` is the name of the property. In the example above the target is the `hidden` DOM property.

The *text* to the right of `=` is javascript code that gets executed and the resulting value is assigned to the target.



Attribute vs Property

Attributes and Properties are not the same.

Attributes - HTML

Properties - DOM (Document Object Model)

Attributes initialize DOM properties and then they are done. Attribute values cannot change once they are initialized.

Property values however can change.

Example of Property Binding

```
<input disabled="false" id="{{myId}}" type="text" value="Vishwas">
```

```
<input [disabled]="false" id="{{myId}}" type="text" value="Vishwas">
```

```
    <input [disabled]="isDisabled" id="{{myId}}" type="text"


  `,
  styles: []
})
export class TestComponent implements OnInit {

  public name = "Codevolution";
  public myId = "testId";
  public isDisabled = true;
```

Attribute, Class, and Style Bindings in Angular

```
<tr><td [attr.attribute-name]="Attribute Value"></td></tr>
<h1 [class.class-name]="Class Value"></h1>
<h1 [style.style-name]="Style Value"></h1>
```

Class Binding in Angular

Class binding is used to set a class property of a view element. We can add and remove the CSS class names from an element's class attribute with class binding.

The class binding syntax is also like property binding. In property binding, we only specify the element between brackets. But in the case of class binding, it starts with the prefix class, followed by a dot (.), and the name of the class. You then bind the class value with CSS class name like **class.class-name**.

The example below shows the standard way of setting class attribute without binding. In this case, we are setting a class attribute with a class name 'myClass' without binding.

Class Binding in Angular

```
1 <div class="myClass">Setting class without binding</div>
```

The example below shows setting all the class values with binding. In this case, we are binding class "myClassBinding" with class binding.

```
1 <div class="myClass" [class]="myClassBinding">Setting all classes with binding</div>
```

Whenever the template expression evaluates to true, Angular binds that class name to the class binding. It removes the class when the template expression evaluates to false.

Let's see another example where we are binding a specific class name with class binding. In this case, if 'isTrue' value evaluates to true then it will bind the 'myClass' to a class property. If it evaluates to false, then it will not bind the 'myClass' to a class property.
File Name: example.component.ts

Class Binding in Angular

```
import { Component } from "@angular/core";
@Component({
selector: 'app-example',
template: ` <div> <h1 [class.myClass]="isTrue">This class binding is for true value</h1> <h1
[class.myClass]="!isTrue">This class binding is for false value</h1> </div> ` })
export class ExampleComponent {
isTrue: boolean = true;
}
```

```
template: `
  <h2>
    Welcome {{name}}
  </h2>
  <h2 class="text-success">Codevolution</h2>
`,
styles: [`
  .text-success {
    color: green;
  }
  .text-danger {
    color: red;
  }
  .text-special {
    font-style: italic;
  }
`]
```

While the class binding is a fine way of binding, the ngClass directive is preferred for handling multiple class names at the same time.

```
<h2 [class]="successClass">Codevolution</h2>
public successClass = "text-success";
```

Class Binding in Angular

ngClass directive

Style Binding in Angular

Style binding is used to set a style of a view element. We can set inline styles with style binding.

Like with class and attribute binding, style binding syntax is like property binding. In property binding, we only specify the element between brackets. But in case of style binding, it starts with the prefix class, followed by a dot (.) and the name of the style. You then bind the style value with CSS style name like the **style.style-name**.

Style Binding in Angular

Let's consider an example of style binding. In this example, we are binding a color style to the 'h1' element. It will display the text within the h1 tags in a blue color.
File Name: example.component.ts

```
import { Component } from "@angular/core";
@Component({
selector: 'app-example',
template: ` <div> <h1
[style.color]="blue">This is a Blue
Heading</h1> </div> ` })
export class ExampleComponent { }
```

Style Binding in Angular

Style bindings will also have the unit. In the example below, we are setting style font size in "px" and "%" units.

```
template: ` <div> <span [style.font-size.px]="isTrue? 20
: 12">This style binding is set for true value</span>
<span [style.font-size.%]="!isTrue : 120 : 30">This
style binding is set for false value</span> </div> ` })
```

```
export class ExampleComponent { isTrue: boolean = true; }
```

Event Binding in Angular

In many cases, users will not only just view the information or data on web applications or mobile applications, but will also interact with these applications using different user actions like clicks, keystrokes, change events, etc.

Event binding syntax will have a target event name within parentheses on the left of an equal sign, and a quoted template statement on the right.

Syntax: (event)

Event Binding in Angular

```
import { Component } from "@angular/core";
@Component({
selector: 'app-example',
template: ` <div> <button
(click)="onClick()">Click me!</button> </div> `
})
export class ExampleComponent {
onClick(){ alert("You Clicked Me!");
}
}
```

Event Binding in Angular

Target Event Binding

The target event is identified by the name within the parenthesis, ex: (click), which represents the click event. In the example, above we saw the target click event bound to the 'onClick()' method, which will listen to the button's click event

```
<button (click) = "onClick()">Click me!</button>
```

We can also use the prefix on-, in event binding this is known as canonical form.

```
<button on-click = "onClick()">Click me!</button>
```

If the name of the target event does not match with the element's event, then Angular will throw an error "unknown directive".