

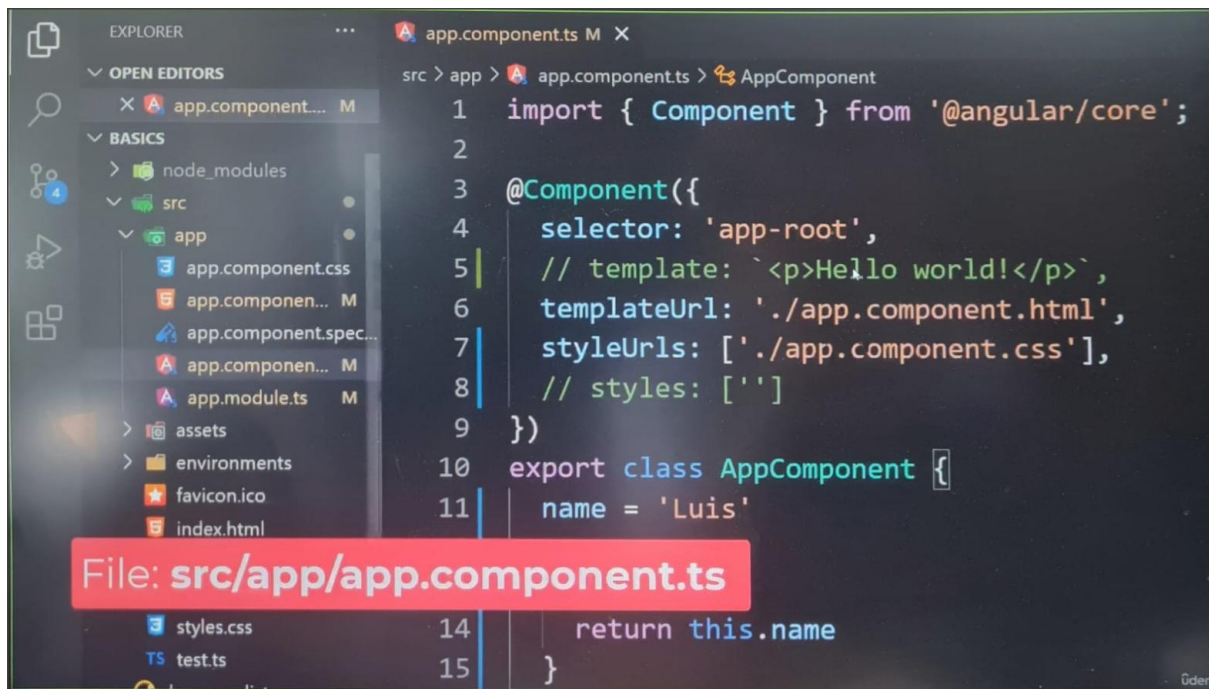
So far, we've learned how to output data.

Angular provides us with Interpolation for rendering data from a component class.

This feature is not the only option at our disposal for manipulating the document.

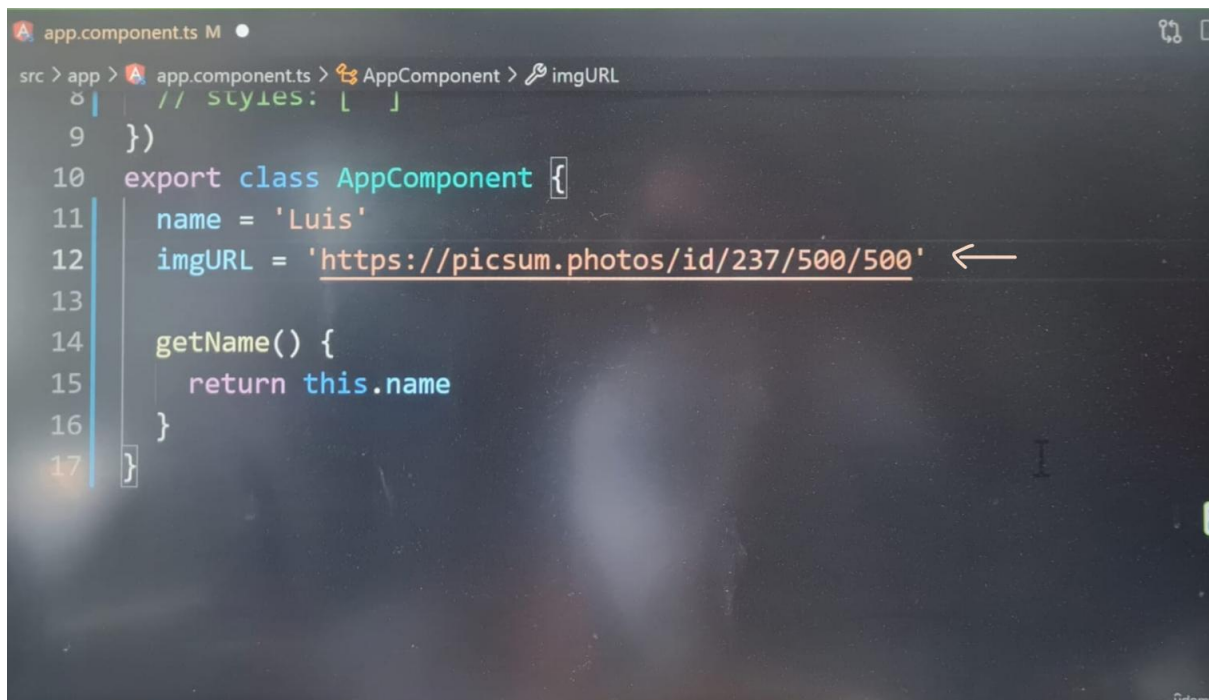
Attribute values can be tied to properties from our class; this feature is known as **Property Binding**.

Through this feature, we can do some exciting things, for example, our attributes can store values from our component class.



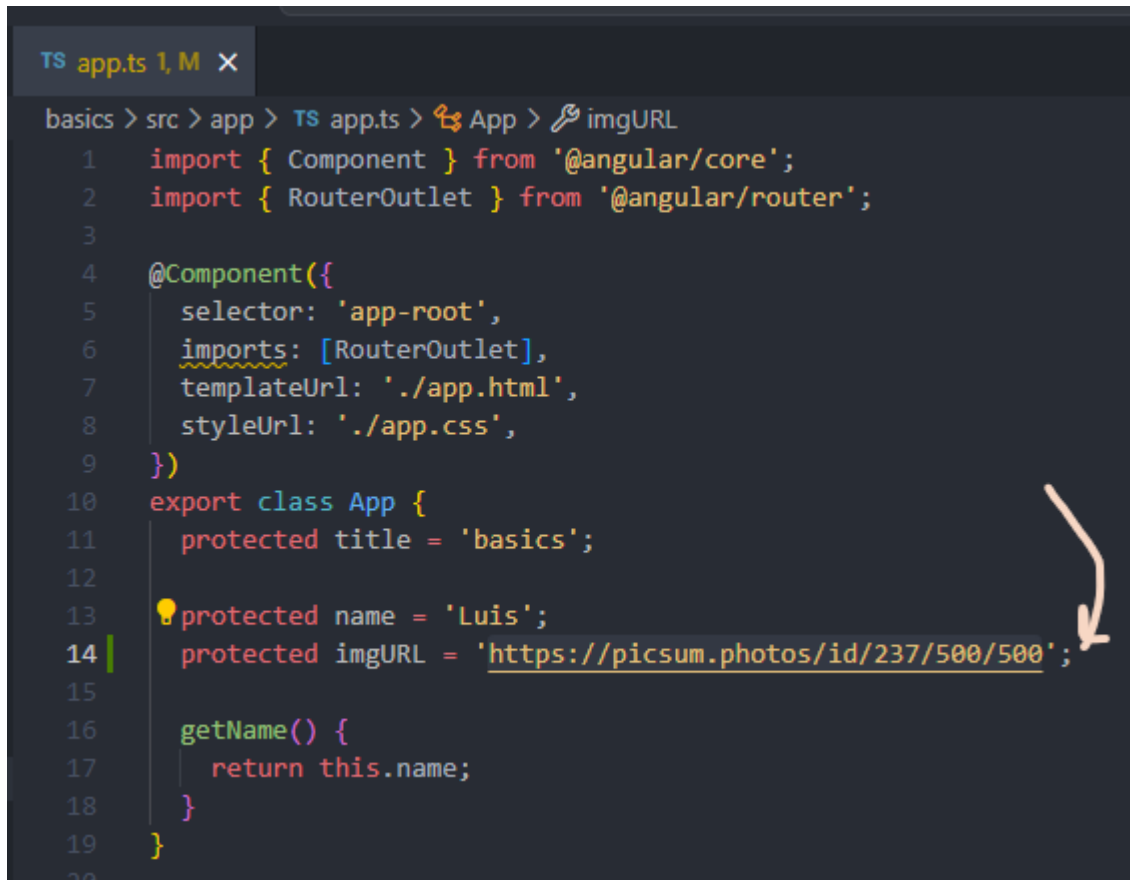
```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'app-root',
5   // template: `<p>Hello world!</p>`,
6   templateUrl: './app.component.html',
7   styleUrls: ['./app.component.css'],
8   // styles: []
9 })
10 export class AppComponent {
11   name = 'Luis'
12
13   // ...
14
15   return this.name
16 }
```

File: src/app/app.component.ts



```
9 })
10 export class AppComponent {
11   name = 'Luis'
12   imgURL = 'https://picsum.photos/id/237/500/500' ←
13
14   getName() {
15     return this.name
16   }
17 }
```

On my latest version of Angular: [app.ts]:

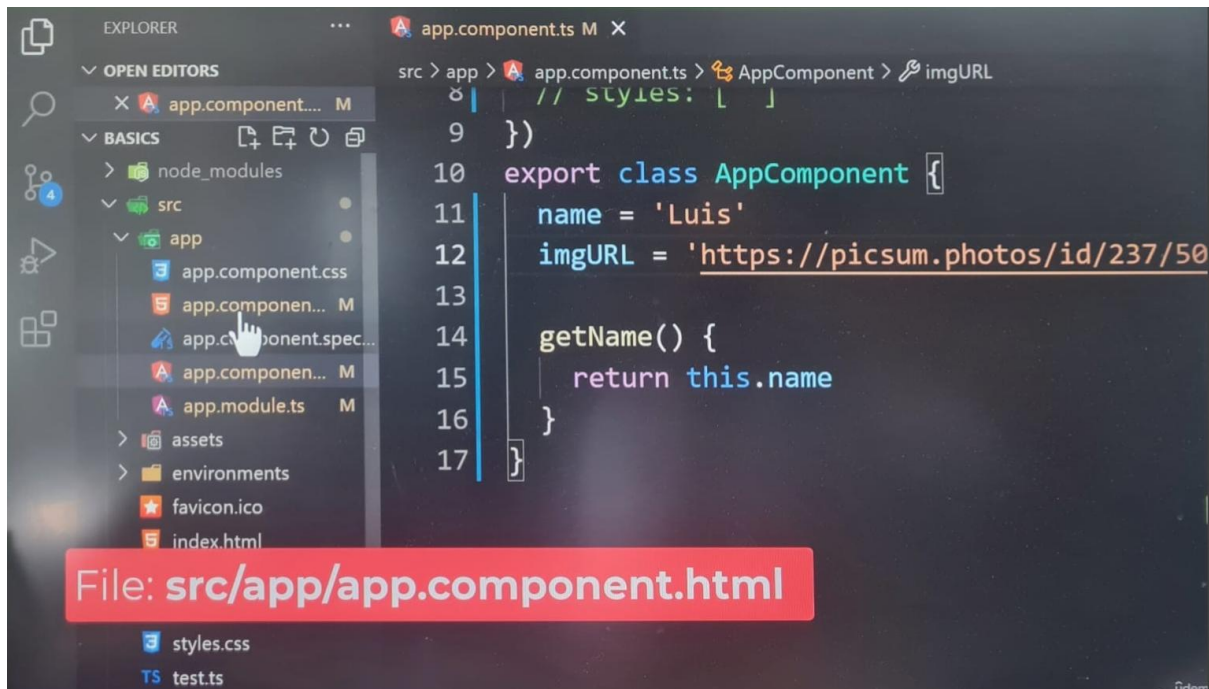


```
TS app.ts 1, M X
basics > src > app > TS app.ts > App > imgURL
1  import { Component } from '@angular/core';
2  import { RouterOutlet } from '@angular/router';
3
4  @Component({
5    selector: 'app-root',
6    imports: [RouterOutlet],
7    templateUrl: './app.html',
8    styleUrls: ['./app.css'],
9  })
10 export class App {
11   protected title = 'basics';
12
13   protected name = 'Luis';
14   protected imgURL = 'https://picsum.photos/id/237/500/500';
15
16   getName() {
17     return this.name;
18   }
19 }
20
```

Angular allows us to modify a templates attribute with a feature called **Property Binding**.

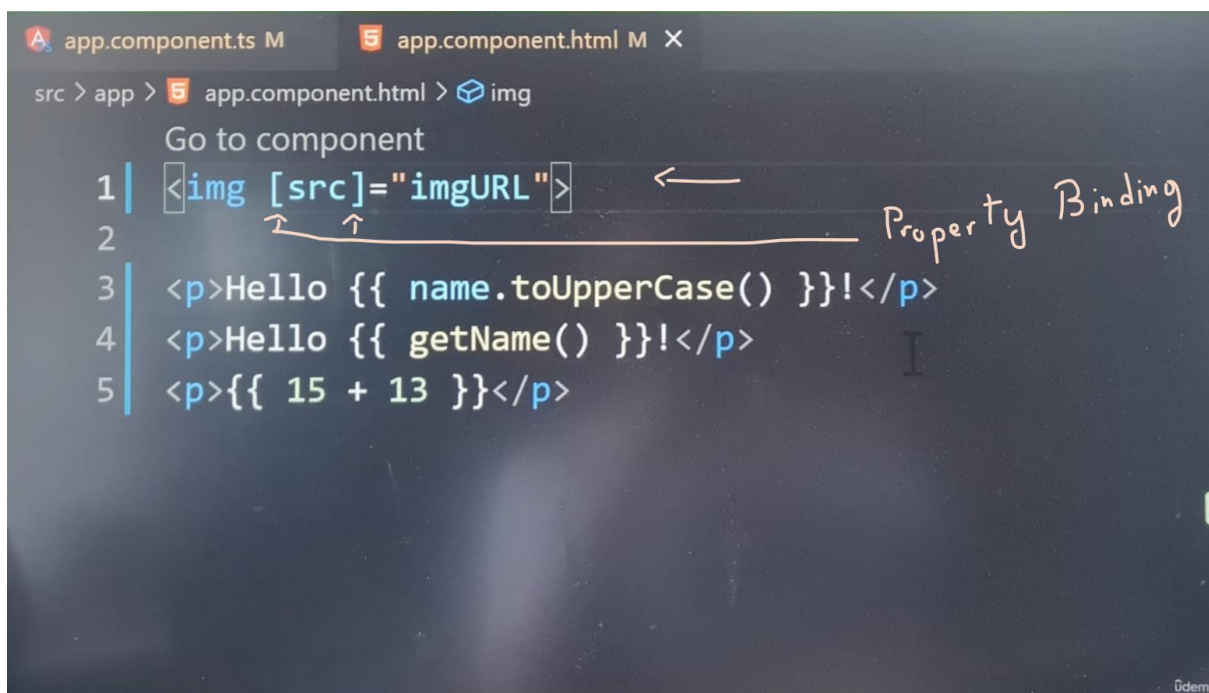
Property Binding is being able to change the document with properties from a class, if a property in our class changes, Angular will update the document with the new value. This is the **change detection system** I mentioned in the last section. Angular will automatically watch our properties for change. Those changes are communicated throughout our app. We can apply **property binding** to any attribute in our documents.

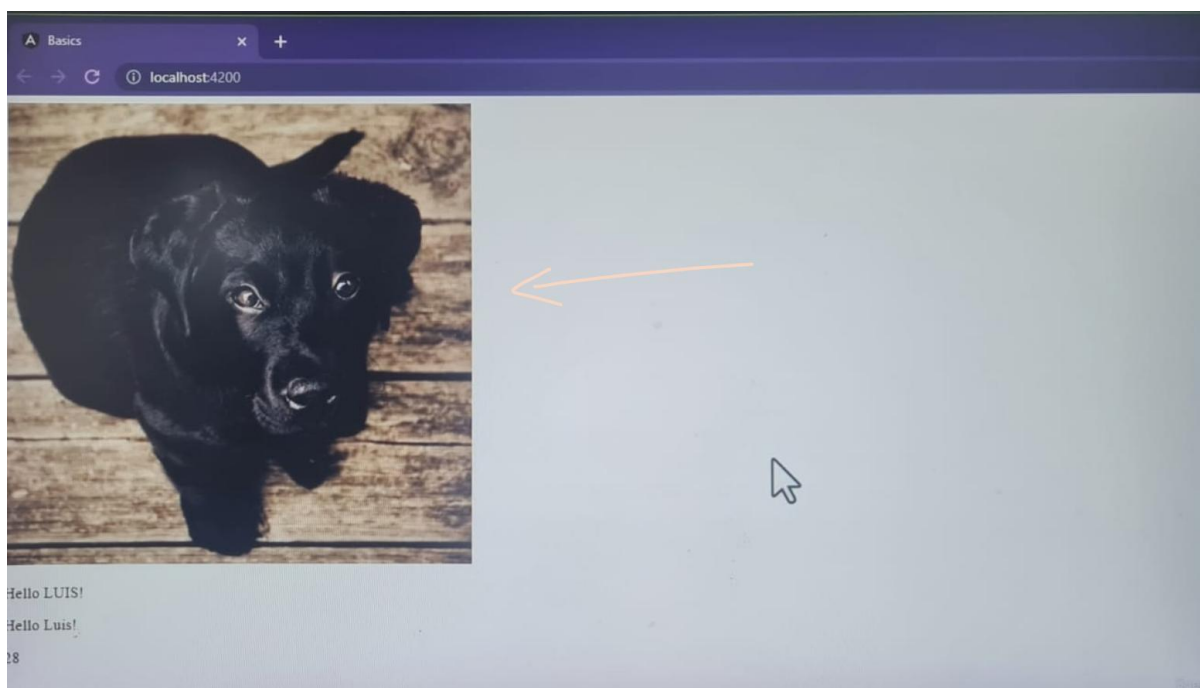
Let's try using property binding to load our image:



The syntax for **Property Binding** is the attribute name surrounded by square brackets [] during the change detection phase, Angular processes our templates, during this process, we saw an example of how it processed our template for rendering the name, if Angular comes across an attribute with square brackets, it'll process the attribute before rendering the components templates, the value of the attribute is treated as an expression. That's important to note down. The value of this attribute is no longer static. It's interpreted as an expression. Whatever value is evaluated from the expression will be used as the value for the attribute.

Let's set this attribute to the image URL property from our class:





On my latest version of Angular: [app.html]:

```
<> app.html M X
basics > src > app > <> app.html > img
Go to component
1 <img [src]="imgURL" />
2
3 <p>Hello {{ name.toLocaleUpperCase() }}</p>
4 <p>Hello {{ getName() }}</p>
5 <p>{{ 15 + 13 }}</p>
6
```



Hello LUIS

Hello Luis

28

The feature we're using is called **Property Binding**; it allows us to apply properties from a class to an attribute value. The Component class in template are tied together.