Input Aliases

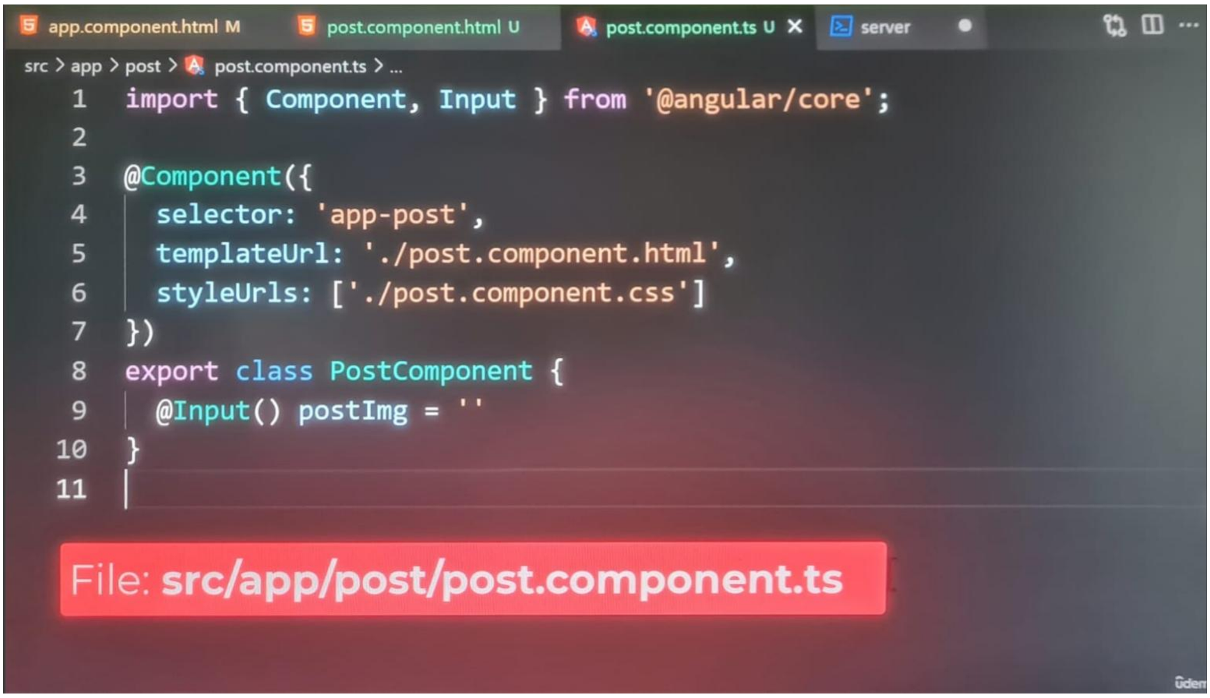
In this lecture, we're going to learn how to add an alias for an input.

We have a property called **postImg** decorated with the @Input() decorator, the parent component has the power to override this property with an attribute. Our attribute name should be short, yet descriptive.

Angular gives us the opportunity to modify the attribute name for the property with an alias, for example, instead of typing **postImg**, we can shorten the attribute name to **image**. The property name can remain the same in the class.

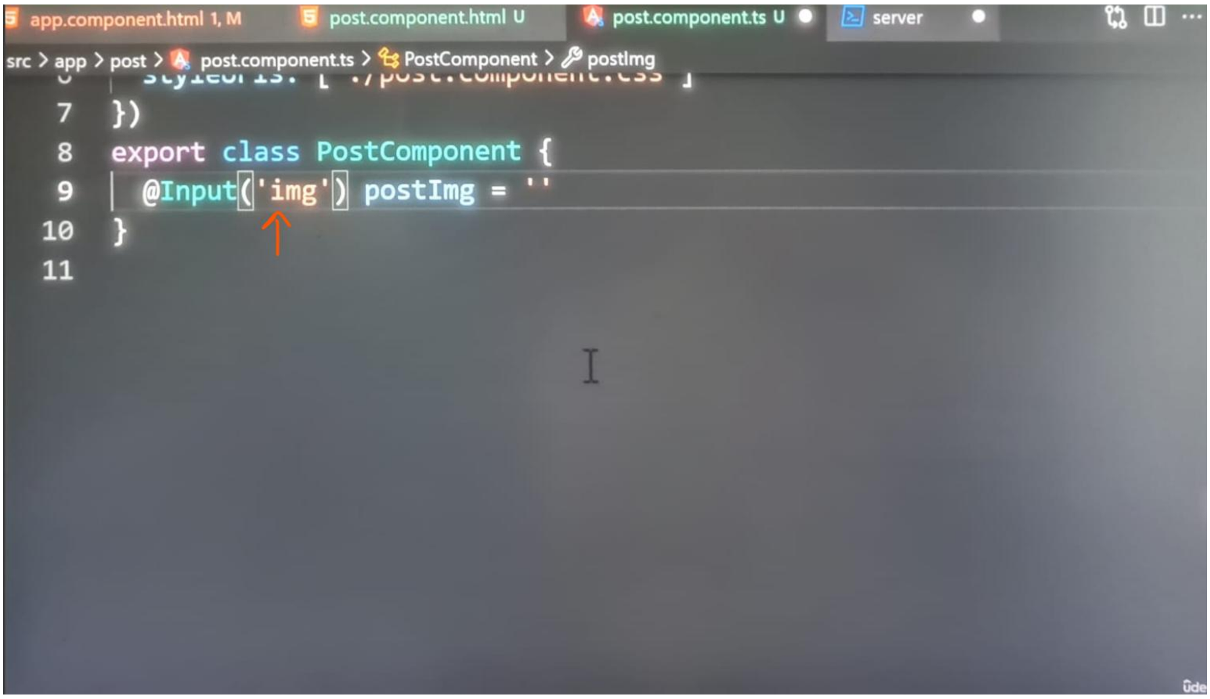
Inside the @Input() decorator function, we can pass in an alias, for this example, let's set the alias to **img**:

@Input('img') postImg = '';



```
import { Component, Input } from '@angular/core';

@Component({
  selector: 'app-post',
  templateUrl: './post.component.html',
  styleUrls: ['./post.component.css']
})
export class PostComponent {
  @Input() postImg = ''
}
```

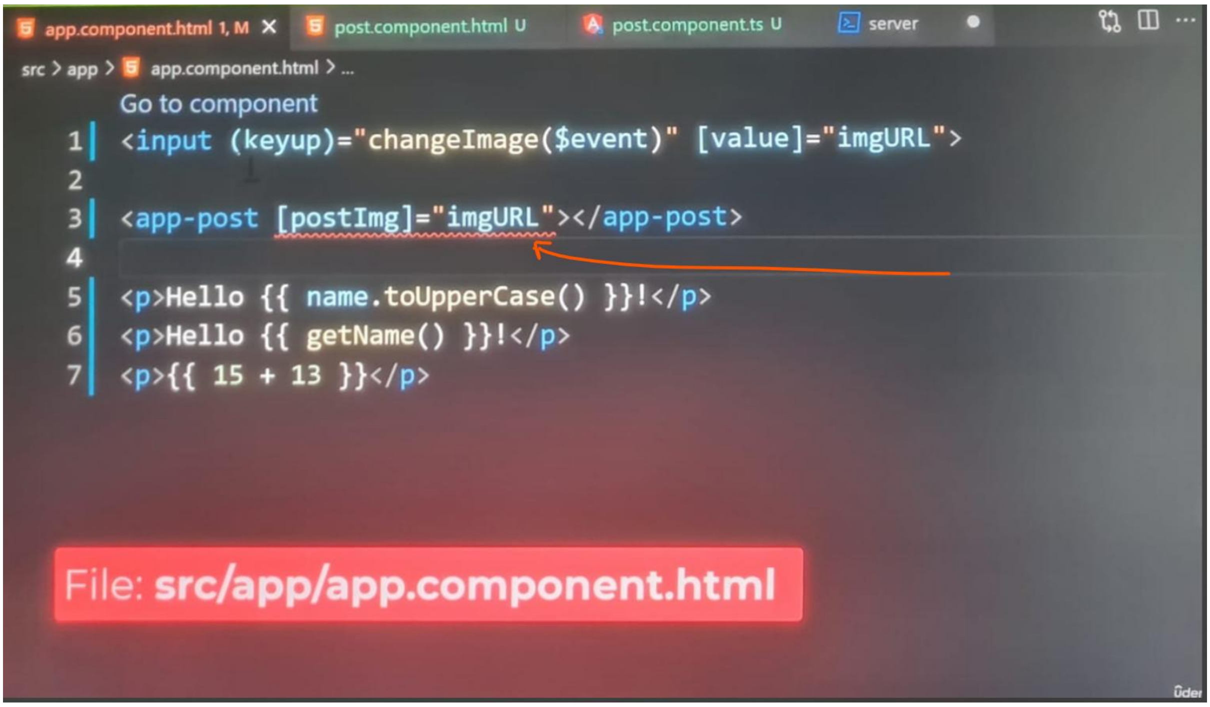
File: src/app/post/post.component.ts



```
  styleUrls: ['./post.component.css']
})
export class PostComponent {
  @Input('img') postImg = ''
}
```

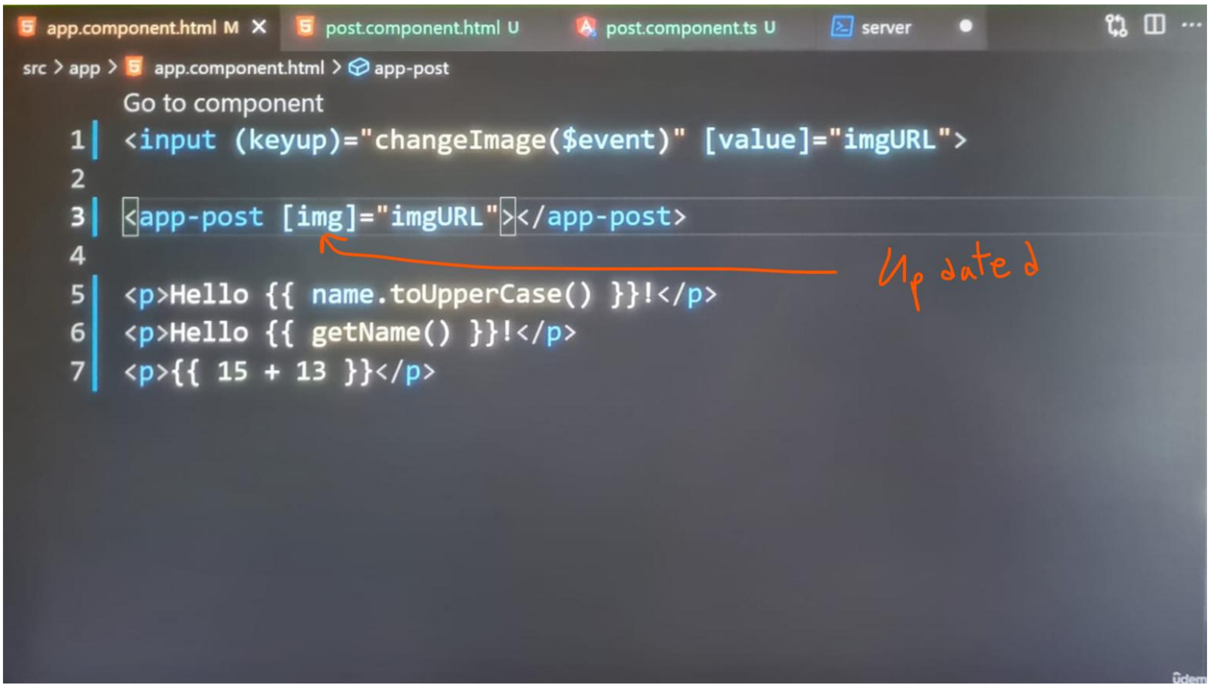
The **postImg** property will hold the **imgURL** in this component. However, if a parent component sets this property, it must use the **alias** to **bind it**.

Let’s switch over to the app template:



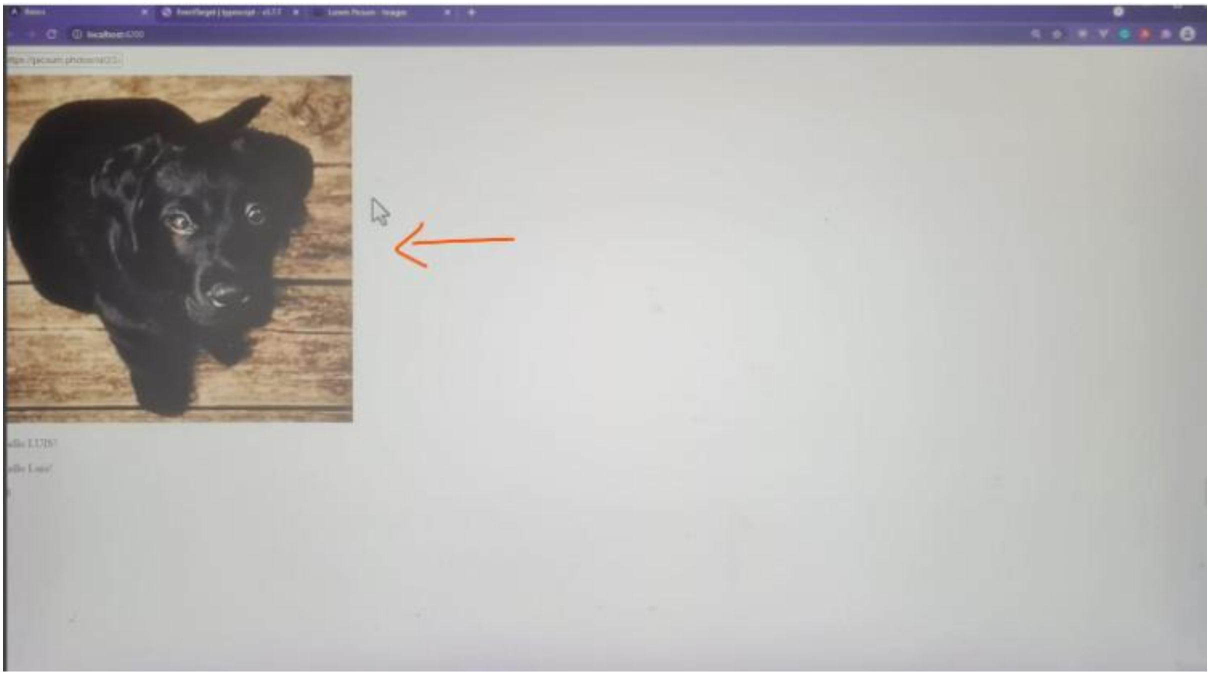
```
<input (keyup)="changeImage($event)" [value]="imgURL">

<app-post [postImg]="imgURL"></app-post>

<p>Hello {{ name.toUpperCase() }}!</p>
<p>Hello {{ getName() }}!</p>
<p>{{ 15 + 13 }}</p>
```

On the app-post component, we are going to change the binding to **img**:



```
<input (keyup)="changeImage($event)" [value]="imgURL">

<app-post [img]="imgURL"></app-post>

<p>Hello {{ name.toUpperCase() }}!</p>
<p>Hello {{ getName() }}!</p>
<p>{{ 15 + 13 }}</p>
```

The image still works on the browser:



Alias gives us even more flexibility with how we assign names to properties.

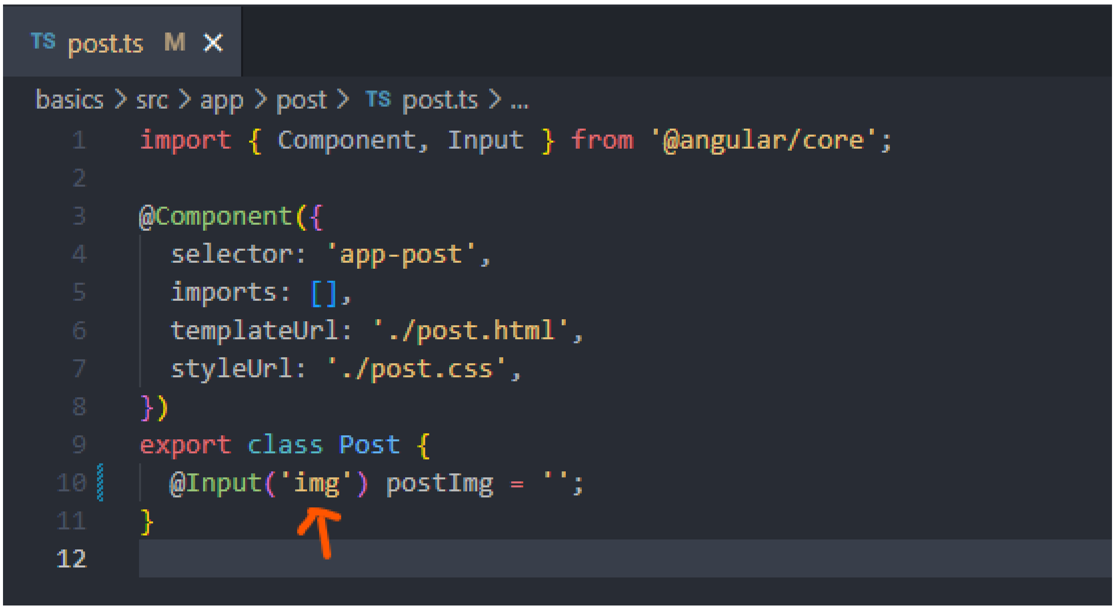
It’s not a feature we should use often.

https://v17.angular.io/guide/styleguide#avoid-aliasing-inputs-and-outputs

Avoid aliasing *inputs* and *outputs.* [Decorators]

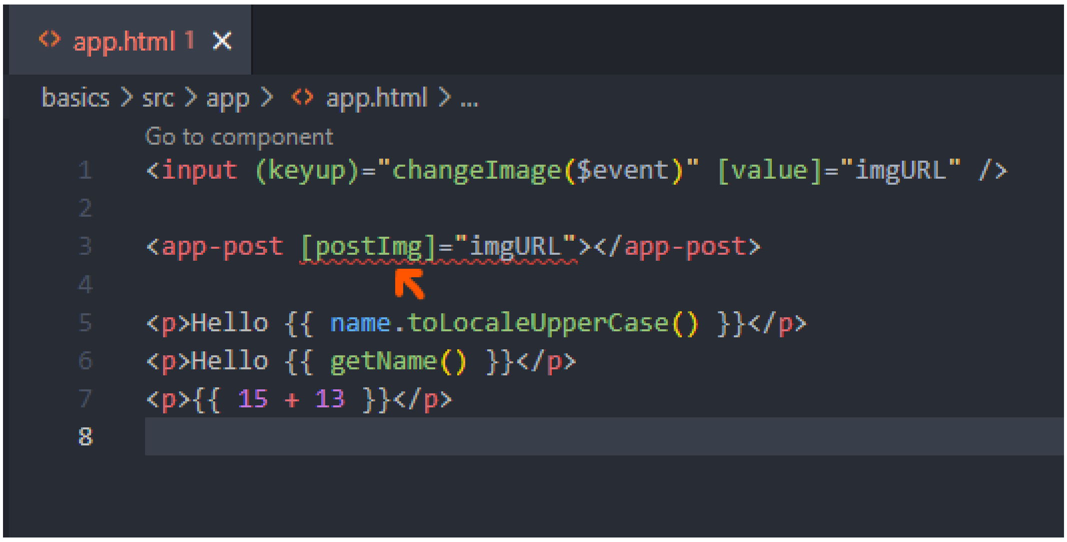
My code in latest version of Angular 20.0.5 with standalone architecture:

Post.ts:

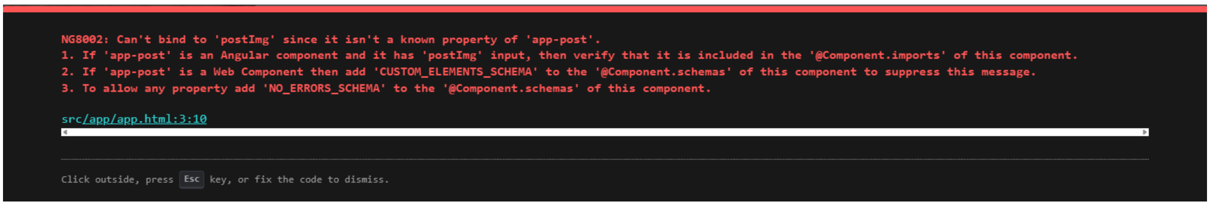


```typescript
import { Component, Input } from '@angular/core';

@Component({
  selector: 'app-post',
  imports: [],
  templateUrl: './post.html',
  styleUrl: './post.css',
})
export class Post {
  @Input('img') postImg = '';
}
```

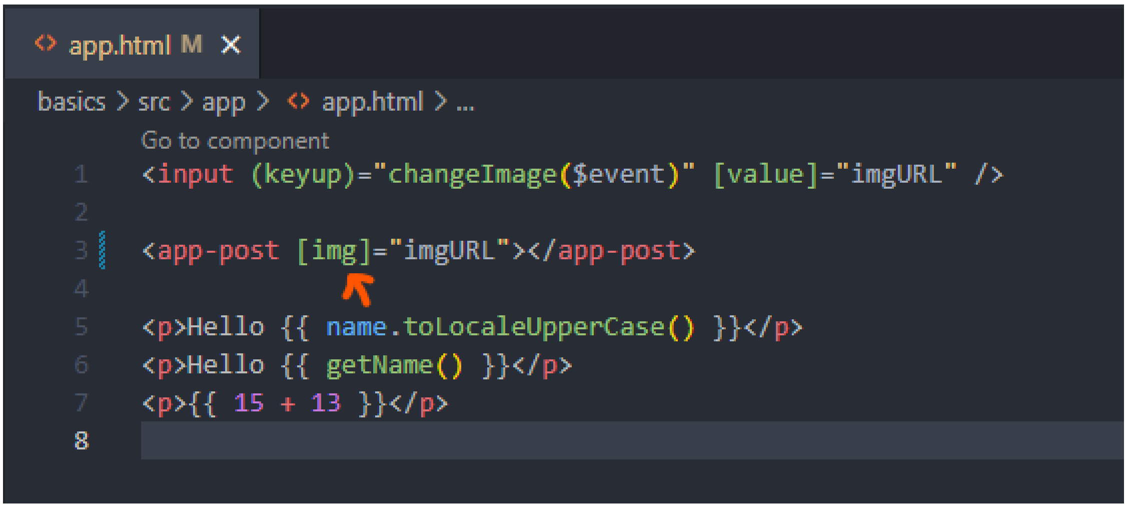
App.html:



```html
<input (keyup)="changeImage($event)" [value]="imgURL" />

<app-post [postImg]="imgURL"></app-post>

<p>Hello {{ name.toLocaleUpperCase() }}</p>
<p>Hello {{ getName() }}</p>
<p>{{ 15 + 13 }}</p>
```

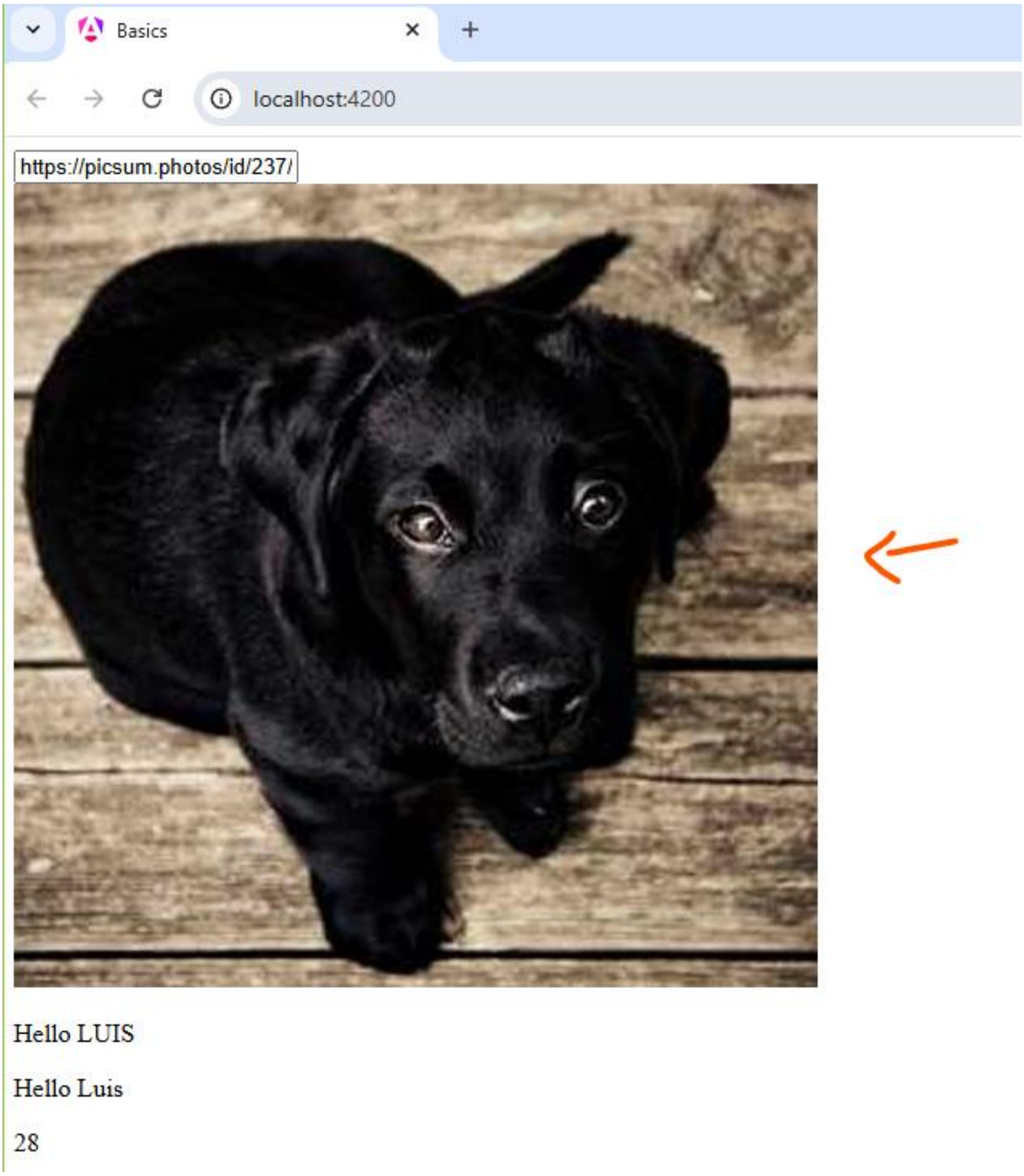


```
NG8002: Can't bind to 'postImg' since it isn't a known property of 'app-post'.
1. If 'app-post' is an Angular component and it has 'postImg' input, then verify that it is included in the '@Component.imports' of this component.
2. If 'app-post' is a Web Component then add 'CUSTOM_ELEMENTS_SCHEMA' to the '@Component.schemas' of this component to suppress this message.
3. To allow any property add 'NO_ERRORS_SCHEMA' to the '@Component.schemas' of this component.

src/app/app.html:3:10

Click outside, press Esc key, or fix the code to dismiss.
```



```
<input (keyup)="changeImage($event)" [value]="imgURL" />

<app-post [img]="imgURL"></app-post>

<p>Hello {{ name.toLocaleUpperCase() }}</p>
<p>Hello {{ getName() }}</p>
<p>{{ 15 + 13 }}</p>

```

Output:



```
https://picsum.photos/id/237/

Hello LUIS

Hello Luis

28
```