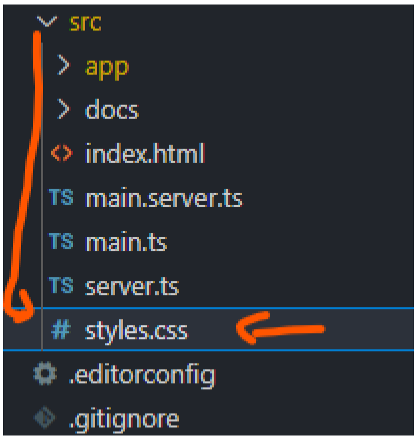
Scoped CSS

In this lecture, we are going to discuss how Angular treats CSS in components.

Behind the scenes, Angular treats component CSS differently than Global CSS. Being aware of these features will prevent you from running into problems.

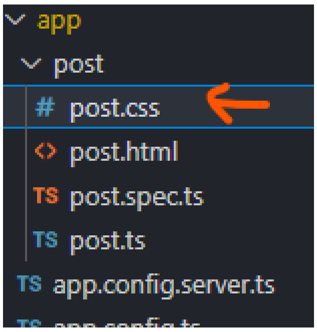
We have two options for applying CSS to a component.

1. We can add CSS to the src/styles.css:



   Styles added to this file are applied globally to an app. However, we should add styles for a specific component in its respective style sheet file.

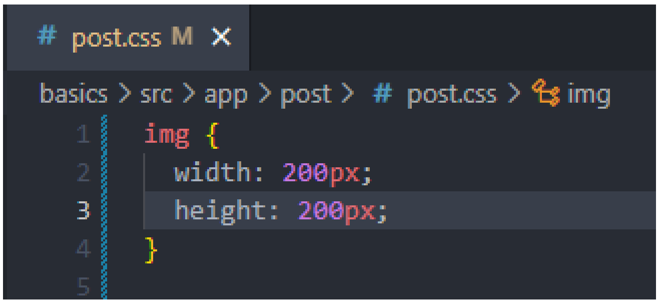
2. For example, our post component has a CSS file called **post.component.css or in my case post.css:**



   Styles created in a component CSS files are scoped. What this means is that styles can only affect a components template. Our styles cannot affect other components.

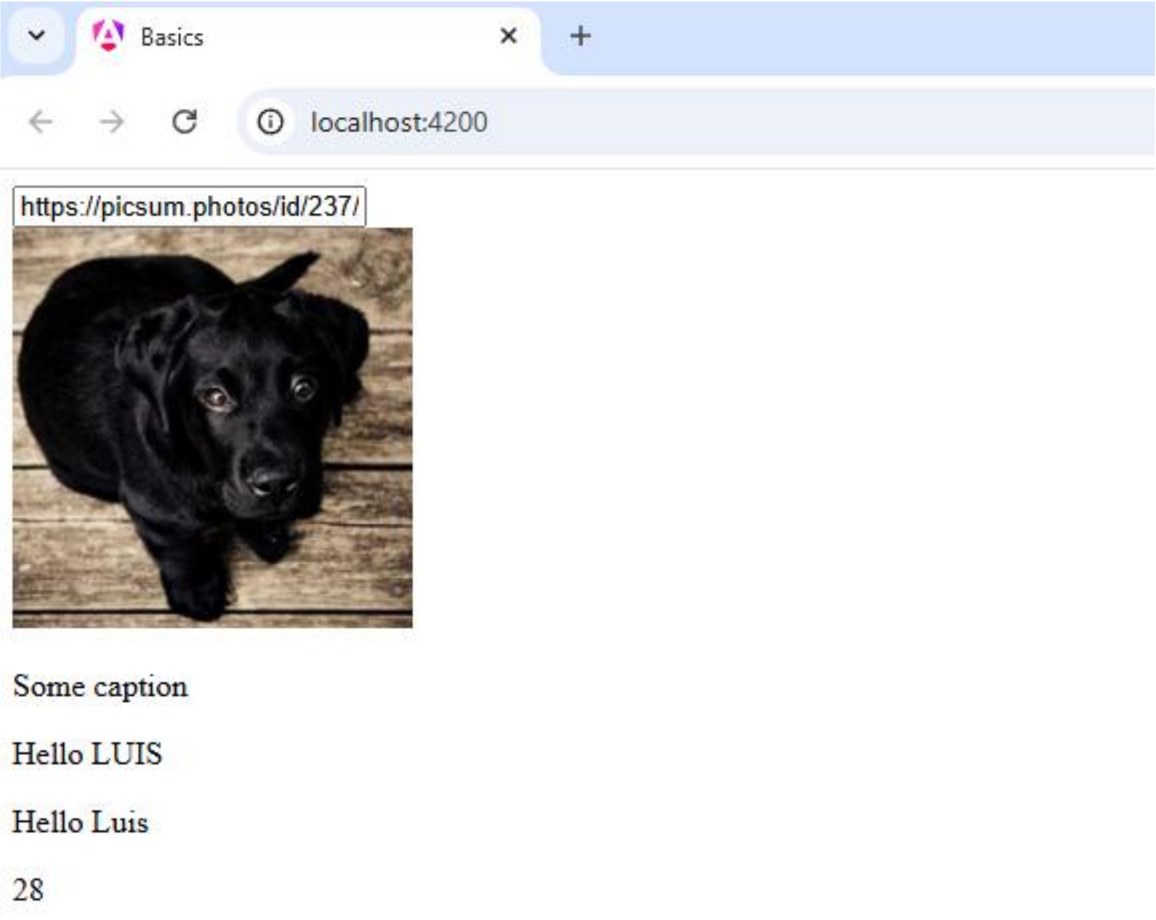
To better understand, let's look at an example:

Inside **post.component.css or post.css:**



```css
img {
  width: 200px;
  height: 200px;
}

```



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

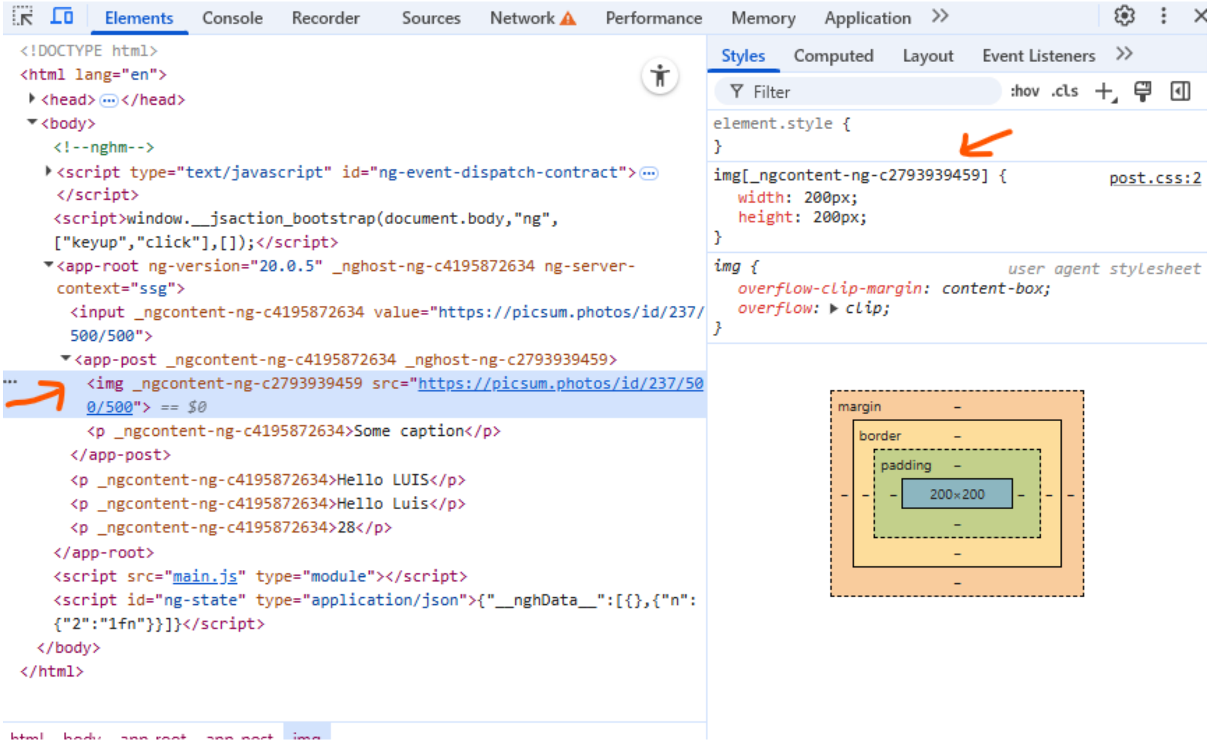
Some caption

Hello LUIS

Hello Luis

28

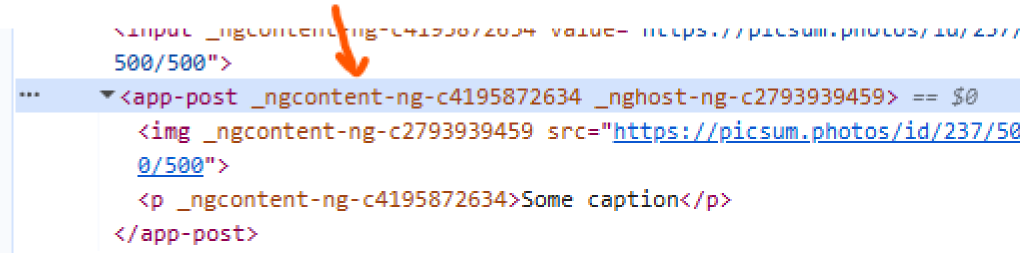
The image has been resized without a problem. However, there’s something else happening behind the scenes. Let’s inspect the element of the image:



```
<!DOCTYPE html>
<html lang="en">
 <head> ··· </head>
 <body>
   <!--nghm-->
  <script type="text/javascript" id="ng-event-dispatch-contract"> ···
   </script>
   <script>window.__jsaction_bootstrap(document.body,"ng",
   ["keyup","click"],[]);</script>
  <app-root ng-version="20.0.5" _nghost-ng-c4195872634 ng-server-
   context="ssg">
     <input _ngcontent-ng-c4195872634 value="https://picsum.photos/id/237/
     500/500">
    <app-post _ngcontent-ng-c4195872634 _nghost-ng-c2793939459>
       <img _ngcontent-ng-c2793939459 src="https://picsum.photos/id/237/50
       0/500"> == $0
       <p _ngcontent-ng-c4195872634>Some caption</p>
     </app-post>
     <p _ngcontent-ng-c4195872634>Hello LUIS</p>
     <p _ngcontent-ng-c4195872634>Hello Luis</p>
     <p _ngcontent-ng-c4195872634>28</p>
   </app-root>
   <script src="main.js" type="module"></script>
   <script id="ng-state" type="application/json">{"__nghData__":[{},{"n":
   {"2":"1fn"}}]}</script>
 </body>
</html>
```

```
element.style {
}
img[_ngcontent-ng-c2793939459] {                    post.css:2
  width: 200px;
  height: 200px;
}
img {                                  user agent stylesheet
  overflow-clip-margin: content-box;
  overflow: ▸ clip;
}
```

In the **developer tools**, Angular has added a special attribute to the image tag. It's an ID for preventing the styles from being applied to other images. Angular has modified the selector to include the **custom attribute attached to the image tag. By adding this ID, it can adequately select the image tag without selecting other image tags. Our CSS has been encapsulated. If we have images in other components, they won't be resized from this component styles. This behaviour is incredibly beneficial to us. We never have to worry about having conflicting styles.**

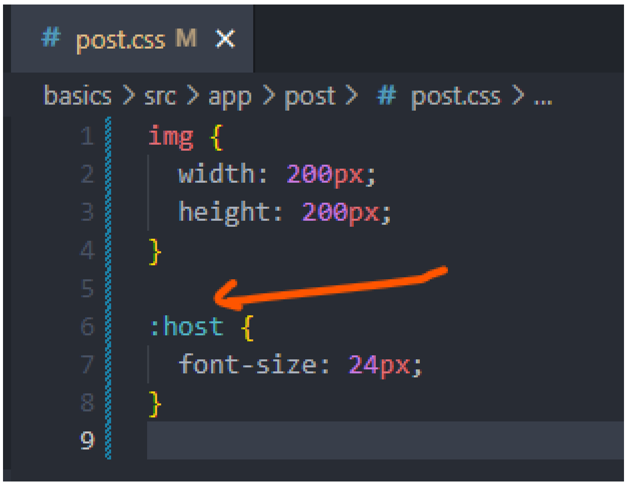


```
    <input _ngcontent-ng-c4195872634 value="https://picsum.photos/id/237/
    500/500">
··· ▼<app-post _ngcontent-ng-c4195872634 _nghost-ng-c2793939459> == $0
      <img _ngcontent-ng-c2793939459 src="https://picsum.photos/id/237/50
      0/500">
      <p _ngcontent-ng-c4195872634>Some caption</p>
    </app-post>
```

If we look back at the elements panel, we will notice something else. The template is being wrapped with the **custom component tag.** Our image is wrapped with the <app-post> components. Angular does not replace the component tag with the template. Instead, it inserts the template into the tag. In some cases, we may want to select the tag. For example, we may want to select the <app-post> component tag. If we're going to apply styles to the app-post tag from the **post.component.css file**, we will need to use a special **pseudo selector.**

Switch back to the editor.

Let's say we want to increase the font size of all text inside the components. Unfortunately, we can't select the elements inside the component with <app-post> selector. Instead, we need to replace it with a **pseudo selector called host – This selector will tell Angular to apply these styles to the component tag wrapped around the templates:**

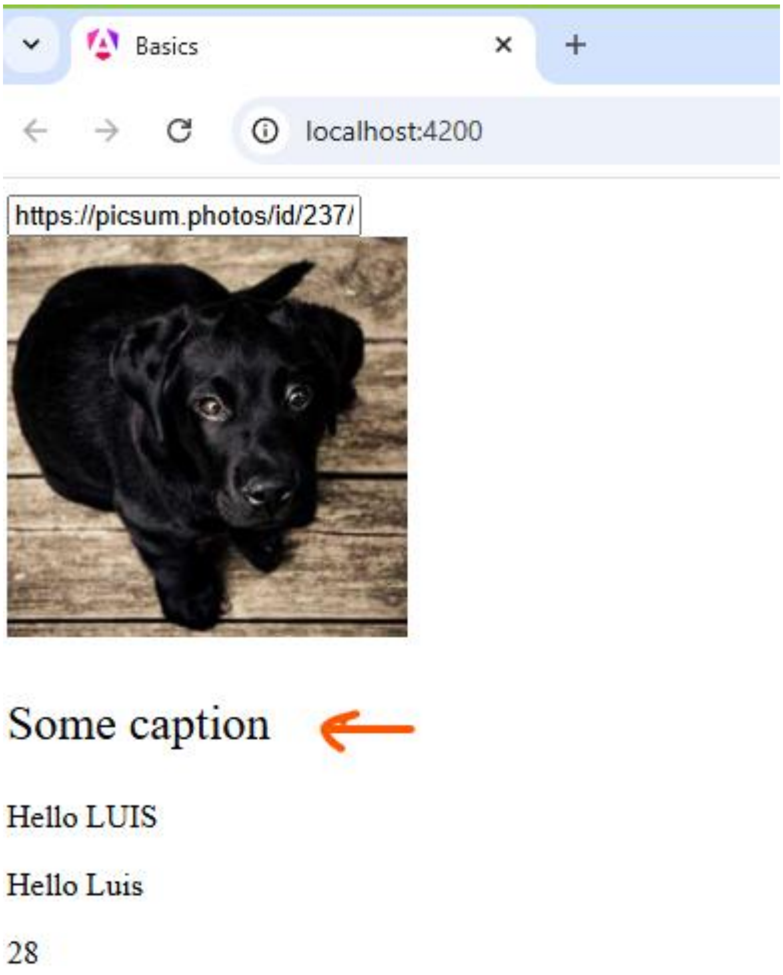


```css
img {
  width: 200px;
  height: 200px;
}

:host {
  font-size: 24px;
}

```

Refresh the page in the browser:



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

Some caption

Hello LUIS

Hello Luis

28

The caption we added to the image has been enlarged, even though it comes from the app component. It’s a part of the post component through **content projection.** That wraps up this section.