

Attribute Binding

In the previous lecture, you were introduced to **"Property Binding"** - a key Angular feature that allows you to bind element properties to dynamic values.

For example, `` binds the `src` property of the underlying [HTMLImageElement](#) DOM object to the value stored in `someSrc`.

Whilst it might look like you're binding the `src` attribute of the `` tag, you're actually NOT doing that. Instead, property binding really targets the underlying DOM object property (in this case a property that's also called `src`) and binds that.

This might look like a subtle detail (and often it indeed doesn't matter) but it's important to understand this difference between element attributes and property. [This article](#) can help with understanding this difference.

Whilst it won't make a difference in Angular apps in many cases, it DOES matter if you're trying to set attributes on elements dynamically. Attributes which don't have an equally-named underlying property.

For example, when binding [ARIA attributes](#), you can't target an underlying DOM object property.

Since "Property Binding" wants to target properties (and not attributes), that can be a problem. That's why Angular offers a slight variation of the "Property Binding" syntax that does allow you to bind attributes to dynamic values.

It looks like this:

```
<div
  role="progressbar"
  [attr.aria-valuenow]="currentVal"
```

```
[attr.aria-valuemax]="maxVal">...</div>
```

By adding `attr` in front of the attribute name you want to bind dynamically, you're *"telling"* Angular that it shouldn't try to find a property with the specified name but instead bind the respective attribute - in the example above, the `aria-valuenow` and `aria-valuemax` attributes would be bound dynamically.