

Attribute Binding

In this lesson, we're going to look at the concept of attribute binding. If you're coding along with the repo, you can checkout the `L3-start` branch or the [starting code](#) on Codepen.

Our Goal

In the starting code, we have a new `div` with the class of `product-image`.

 `index.html`

```
<div class="product-image">
  <!-- image goes here -->
</div>
```

By the end of the lesson, we're going to have an `img` element here that is reactively bound to a new `image` property on our data. Whenever the value of that image changes, our image will update in the DOM.

Adding an image to our data

Remember in our **assets** directory we have an **images** folder, with images for green and blue socks? Let's target one of those images from a new data property on our Vue app. We'll do so by setting `image` equal to a path so it can grab that image.

 `main.js`

```
const app = Vue.createApp({
  data() {
    return {
      product: 'Socks',
      image: './assets/images/socks_green.jpg'
    }
  }
})
```

Now we're ready to add an `img` element in the template.

 `index.html`

```
<div class="product-image">
  
</div>
```

In the source attribute, we'll say `image`. Right now, this won't do anything. We want `src` to pull in the `image` path from our data, similar to how we pulled in the `product` data value in the `h1` expression in the previous lesson.

So the question here is: **How do we bind the `src` attribute to the `image` data?**

Introducing Attribute Binding

To create a bond between an HTML element's attribute and a value from your Vue app's data, we'll use a Vue directive called `v-bind`.

 `index.html`

```

```

Now, we've created a reactive bond between what lives in this attribute (`"image"`) and the `image` data itself.

Looking in the browser, we'll now see our green socks image displaying.



Understanding v-bind

How exactly is the `v-bind` directive working? We use this directive to dynamically bind an attribute to an expression. In this case, the attribute is `src` and the expression is whatever is in that attribute's quotes: `"image"`

 `index.html`

```
 <!-- src attribute bound to the image data -->
```

If you're thinking this doesn't look like a typical JavaScript expression, you can imagine it looking like this: `v-bind:src="{{ image }}"`. Under the hood, Vue is going to evaluate it just the same.

v-bind: Dynamically bind an attribute to an expression



A Reactive Bond

Because of Vue's reactivity system, if we updated our `image` data to a path that points to the image of our blue socks (`image: './assets/images/socks_blue.jpg'`), the expression that our `src` attribute is bound to would update and our browser would display the blue socks image.

A shorthand for v-bind

Using `v-bind` is super common—so common that there is a shorthand for it, and it's just the colon, like so:

```

```

As you can imagine, since there's so many different HTML attributes, there are many use cases for `v-bind`. For example, you might be binding a description to an `alt` attribute, binding a URL to an `href`, binding some dynamic styles to a `class` or `style` attribute, disabling and enabling a button, and so on.

Coding Challenge

We've reached the end of the lesson and we're onto our second challenge:

Add a `url` to the data object

Bind the `url` to an anchor tag's `href` attribute

As a reminder, if you're coding along with our repo, you can check out `L3-end` branch, and you can view the [solution code](#) on Codepen.