

Inheritance and Extends

In this lesson, we'll be looking at how to implement inheritance with the extends directive.

WE'LL COVER THE FOLLOWING ^

- Definition
- Example

Definition

Another great feature of Sass is **inheritance**. We implement this using the `@extend` directive.

Inheritance is a feature of SASS that allows multiple classes to share a common set of properties with one another.

Example

Some typical CSS code for styling a button:

```
.button {  
  background-color: #0000FF; // Blue  
  border: none;  
  color: white;  
  padding: 15px 32px;  
  text-align: center;  
  text-decoration: none;  
  display: inline-block;  
  font-size: 1.5rem;  
}
```




If we happen to have a number of buttons on our website, all of which are styled in a similar manner, we would have a good case for inheritance!

We would implement a secondary button via inheritance like so:

```
.button-secondary {  
  @extend .button;  
  background-color: #4CAF50; // Green  
}
```

Our `.button-secondary` class will take on all of the properties and values set the `.button` class, with the exception of `background-color` which we've decided to set to green.

Try it out below!

Output	
HTML	
CSS (SCSS)	
<pre>.button { background-color: #0000FF; // Blue border: none; color: white; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block; font-size: 1.5rem; } .button-secondary { @extend .button; background-color: #4CAF50; // Green }</pre>	
<div></div>	

The use of inheritance helps us to keep our code neat, clean and focused on constructing reusable components.

Up next, we'll take a look at the `&` operator.