Partials and Importing

In this lesson, we'll be looking at how to modularize our files with partials & importing.

WE'LL COVER THE FOLLOWING

Definition

Partials in SASS help us to break up our files into smaller files without affecting performance.

The use of partials allows us to modularize our CSS, to help keep things maintainable.

We divide up our Sass into separate files representing different components. A partials' name will always start with an underscore `_`. We then import the partial using an `@import` directive.

Example

Let's say our Sass file is getting rather large. We might choose to make a partial file that contains just the code that's relevant to the header section; we'd call it `_header.scss` and move the appropriate code into that new file.

We would then import it back into `main.css`, like so:

```
// In main.scss
@import 'header';
```

Note: When you import a file, there's no need to include the `_` or `.scss` file extension.

@import in CSS vs SASS

The `@import` directive is of course, also available in CSS. However, there is an important difference. Whenever an import statement is used in CSS, an HTTP request is made to the server. This increases the amount of resource requests and negatively affects the performance of a web page. SASS does not do that. SASS takes the file that you want to import from and combines it with the file you're importing. So you can serve a single CSS file to the web browser, which does not affect the performance.

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