MODULE 01 - 020: SCSS `@import` directive

The `@import` directive allows to: - **Include other SCSS files** into a main file. - **Organize your code** for a better readability/scalability/maintainability. - Use both **partials** (files prefixed with `_`) and complete SCSS files.

---

1. Master File:

- `application.scss` file contains all @import directives.

In a typical setup: - You have a **master SCSS file** (e.g., `application.scss`). - This file includes multiple `@import` statements, which act as modular building blocks.

2. Output:

- After precompilation and minification, a single optimized CSS file is generated, ready for production.

In final words, managing the SCSS stylesheets "modularly" helps improving the database organisation.
Almost any framework is ready to work using multiple stylesheets directed by the master file.

Video Speech lesson

[ENG]
# Real World Example of the Scss @import Directive
This lesson walks through a real world production application and examines how the @import directive is used to organize a full project.
*** We're making our way nicely through the Scss course. And in this guide, we're going to talk about the import directive inside of Scss and instead of us going and looking at a few sample pieces of code I wanted to show you a real-world production application because the import directive specifically was made to be able to manage your code especially for larger applications.
Right here we have a decent size application. And so if I go in this is a Rails app but Scss can be used in any programming language and framework.
So you do not have if you work in something else such as PHP or dot net or something like that then you can absolutely still use Scss.

This is just an example. So I'm going to go into where I have my main application file and you can see here it is application dot Scss.
So in order to use this statement, it has to be a dot Scss file extension, and right here instead of seeing styles what you actually see are a bunch of import statements.

So what does that mean?
Well, what the import directive does is it allows you to as you may have guessed import other files.
Now sometimes these files can be partials and then other times they can be entire other sets of Scss files.
So we're going to walk through what that actually looks like.

This is a very common pattern you'll see where you have one master application Scss file and then it imports all kinds of other Scss files to be able to organize the code and it also accomplishes one other task that is very important.
If you remember Cascading Style Sheets is what represents CSS that's the acronym. The whole cascading nature of stylesheets means that it's very easy to accidentally have one style override another one if you're using a framework such as the bootstrap framework like we're using right here it's incredibly important to be able to override certain styles that you don't want to keep.

So for bootstrap it has a number of buttons and the buttons may be greyed out of the box for certain prototype applications, but they're not what you would use in a production one.
So you have to make sure that your custom styles that override this are called afterward and by leveraging import statements then it makes that much easier to control that.

This is just an example. So I'm going to go into where I have my main application file and you can see here it is application dot Scss.
So in order to use this statement, it has to be a dot Scss file extension, and right here instead of seeing styles what you actually see are a bunch of import statements.
So what does that mean? Well, what the import directive does is it allows you to as you may have guessed import other files. Now sometimes these files can be partials and then other times they can be entire other sets of Scss files.
So we're going to walk through what that actually looks like.

This is a very common pattern you'll see where you have one master application Scss file and then it imports all kinds of other Scss files to be able to organize the code and it also accomplishes one other task that is very important.
If you remember Cascading Style Sheets is what represents CSS that's the acronym. The whole cascading nature of stylesheets means that it's very easy to accidentally have one style override another one if you're using a framework such as the bootstrap framework like we're using right here it's incredibly important to be able to override certain styles that you don't want to keep.

So for bootstrap it has a number of buttons and the buttons may be greyed out of the box for certain prototype applications, but they're not what you would use in a production one.
So you have to make sure that your custom styles that override this are called afterward and by leveraging import statements then it makes that much easier to control that.

So Rails has its own compilation process but if you click on the application CSS file all of those styles all of those import statements all got brought into a single file and then from there Rail's also it condensed all of it so it minified it.
It made it removed any of the excess whitespace to help shrink the size of the file.
But then before that even happen Scss performed its precompilation process so none of these styles that you would see such as using things like variables or nested elements none of that is actually inside of here.
It is all compiled straight into regular CSSA. So that is a real-world example of how to use the import directive inside of Scss.

Reference Project

Source code
Application (Not Working)

[SPA]

Usamos @import para mejorar la estructura, visibilidad, escalabilidad y mantenimiento de nuestro código, desestructurando lo que sería un larguísimo y complejo archivo SCSS único, en diferentes y modulares archivos.

Esta directiva, por tanto, permite cargar distintos archivos SCSS en uno como módulos. No es ajeno a cómo, en otros lenguajes, se cargan piezas de código cómo módulos.

La forma habitual de usar @import es:
1.- Un archivo "director" referencia cada archivo SCSS externo (Pueden ser archivos parciales, o completos), incluyendolos usando esta directiva.
2.- Cada fichero SCSS referenciado se encuentra dentro de la root folder de nuestro poryecto.

Casi cualquier framework, como Rails, facilita la precompilación usando imports.