How to Install and Configure SASS Locally

In this lesson, we'll cover the installation of SASS onto our local machine. Including the initial configuration & live server setup.

WE'LL COVER THE FOLLOWING

Before we can write Sass code, it needs to be installed locally. As by default, it's not a language known to the browser.

Let's now go through the process to setup the environment that will allow us to write then compile Sass.

Note: When Sass is compiled, it is converted into regular CSS code that browsers can interpret and render.

Environment setup

Before we start, you must have **npm** installed on your computer, it comes bundled with Node.js; you can install it from here. Go ahead and install it if you haven't already.

If you are unsure whether you have Node.js installed or not, run `node -v` from your terminal.

If you see a version number, it's installed!

A note on terminal:

If you are new to SASS, chances are you may also be new to running commands from the terminal. It's not as daunting as it might seem! And it's a real time-saver once you gain more experience.

To open a terminal on a Windows PC, right-click the Windows Icon and select *Windows Powershell*, if you're on a Mac go to *Finder* > *Applications* > *Utilities* > *Terminal*.

Folder Structure

Let's create our project folders! They will be structured like so:

```
sass-project
   |- sass
   |- css
```

To create this structure, open terminal and change to the folder you wish to install the sass project into (via the cd command).

Then run the following commands:

```
mkdir sass-project
cd sass-project
mkdir -p sass css
```

File Structure

You will need an **index.html** and **main.scss** in this folder.

To create these files, run:

```
touch index.html
cd sass
touch main.scss
cd ..
```

You'll also need a default CSS stylesheet for the SASS to compile into:

```
cd css
touch style.css
```

Open up your index.html and paste in the following code:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Index page</title>
    <link rel="stylesheet" href="css/style.css">
  </head>
  <body>

  </body>
</html>
```

This is just our boilerplate code with the stylesheet connected!

Initializing our Project Directory

All projects that use npm need to be initialized. To do this, ensure you're still in the sass-project folder and run the following command:

```
npm init -y
```

This will create a **package.json** file for our project. We'll be learning more about the configuration of this file later in the course!

Install node-sass

node-sass is the library which allows us to compile `.scss` to `.css`.

Run the following command to install node-sass as *dev dependency*.

```
npm install node-sass --save-dev
```

Note: A **dev dependency** is only used in the build phase of our project. It's not included at runtime.

Compiling Sass Code to CSS

Next, we need to create an *npm script* to run the compilation.

Add this script inside the script section of our previously created **package.json** file:

```
"compile-sass": "node-sass sass/main.scss css/style.css"
```

Don't forget to separate each script with a comma!

We have here specified **main.scss** as our main Sass file and **style.css** as the compiled CSS file.

To compile our SASS code into CSS, all we need to do is run:

```
npm run compile-sass
```

Live Reload

Let's also add a live reload to our project! To do this run the following to install globally:

```
npm install live-server -g
```

Now, make sure you're still in the Sass project folder, and run:

```
live-server
```

And just like that, you've got a pretty neat dev environment with your project running locally on HTTP.

You'll need to keep `live-server` and `npm run compile-sass` running in two separate terminal windows.

So we now have the project environment all set up on the local machine!