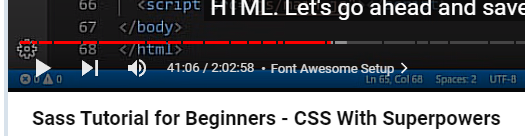
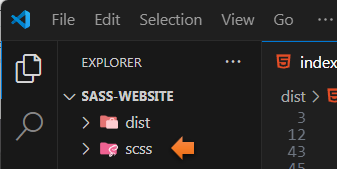
2 The SASS Folder

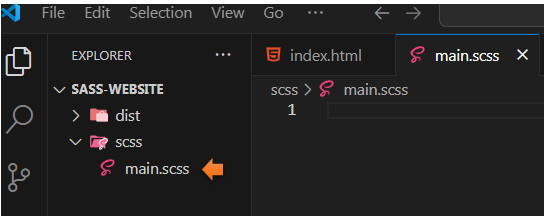
<https://www.youtube.com/watch?v=_a5j7KoflTs>



Make sure you are in the root, that means outside of that dist folder. This folder is for your SCSS and it needs to be created outside of that dist folder.

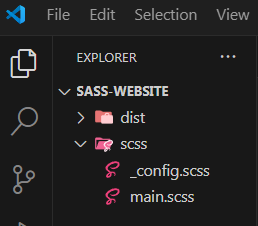


Inside of that file create a file called main.scss



Inside of that same scss folder, we will create a partial \_config

tea



In a large workflow, we would actually create a partial for all of our elements. This is not a large project, but we will still be looking at that work method in this tutorial, so we can be more familiar with it, when we do need to step into larger projects.

Writing the \_config partial file.

$primary-color:#320001;

$secondary-color:#e8854c;

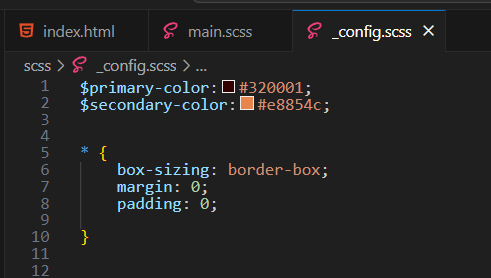
\* {

box-sizing: border-box;

margin: 0;

padding: 0;

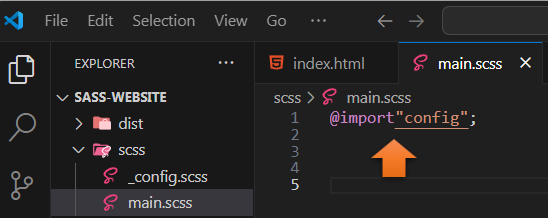
}



# The Main.scss file

Open up the main.scss file that we had just set up. We need to import that config file that we just created, so our main file will have access to it.

@import"config";



# The body code

body {

background: $primary-color;

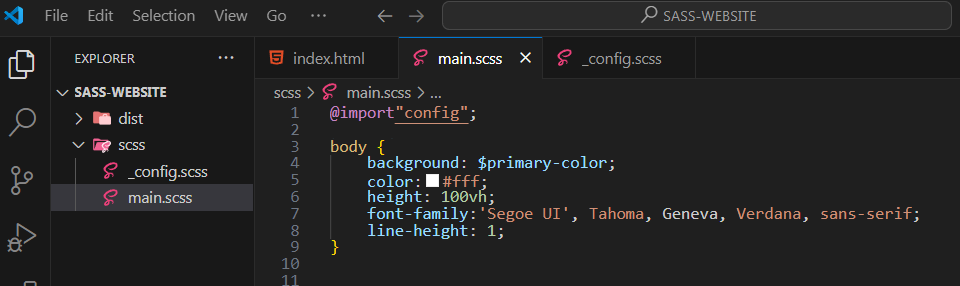
color:#fff;

height: 100vh;

font-family:'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;

line-height: 1;

}



Using vh here for the height will give us 100% of the available viewport, which is what we want.

Using rem, just means that the font-size is based on whatever the body font-size was set to. If you set your body font size to 12, setting rem to 1 will be 12, setting rem size to 2 would be 24, or twice the size and so on… The default font size is actually 16, so if you do not set the font then rem would be based on 16, and twice that would be 32…

Writing the SCSS for the rest of the webpage

h1, h2, h3 {

font-weight: 400;

padding: 8px;

}

a{

color:#fff;

text-decoration: none;

}

header {

position: fixed;

z-index: 2;

width: 100%;

padding: 1rem;

}

.nav ul li{

display: inline-flex;

justify-content: space-between;

top: 0px;

color: white;

padding: 10px 20px;

}

main {

height: 100%;

width: 100%;

padding-top: 100px;

.social-icons {

position: fixed;

bottom: 1rem;

left: 1rem;

a {

padding: 0.4rem;

&:hover {

color: $secondary-color;

}

}

}

}

footer {

font-size: 1rem;

position: fixed;

bottom: 0.4rem;

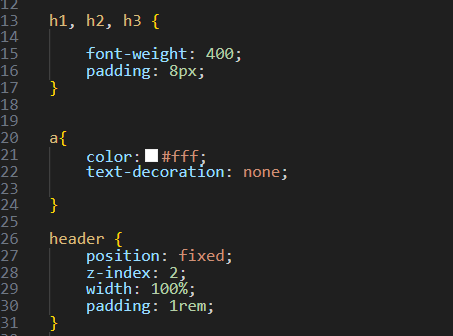
right: 1rem;

text-align: right;

padding: 1rem;

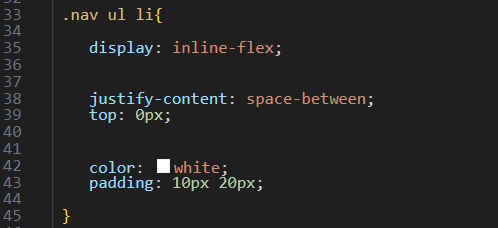
color:#fff;

}



# The Navigation Code

The navigation will default to vertical and we want it to be horizontal



The Main Code

Notice that we have nested some of the code in the .main class rule. The Social icons and the hover was nested inside of the rule.

main {

height: 100%;

width: 100%;

padding-top: 100px;

.social-icons {

position: fixed;

bottom: 3rem;

left: 1rem;

a {

padding: 0.4rem;

&:hover {

color: $secondary-color;

}

}

}

}



The Footer Code

footer {

font-size: 1rem;

position: fixed;

bottom: 0.2rem;

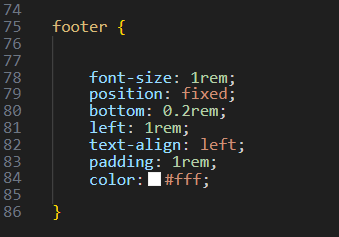
left: 1rem;

text-align: left;

padding: 1rem;

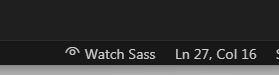
color:#fff;

}

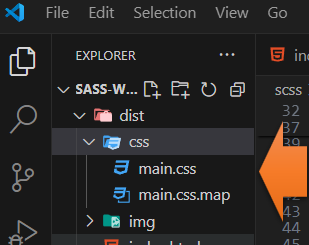


BEFORE TESTING

Turn on SASS



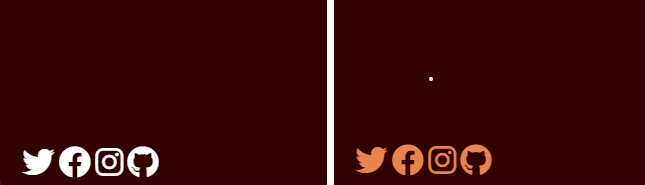
Now if everything worked correctly you can open up your dist folder and you will see your new CSS folder with file was automatically created, and its corresponding main.css.map file. And all of that SASS code was converted into regular CSS, which browsers understand.



You can test your page by using Live server, but without any text or images it still does not look great.

GO TO INDEX.HTML to view page

This is what the hover looks like



This is what the entire page looks like right now

