6 The Hamburger Menu

Contents

[Turn on Sass 1](#_Toc165481807)

[Creating the Partial Menu File 1](#_Toc165481808)

[The code for the \_menu.scss file 2](#_Toc165481809)

[@include the partial file 3](#_Toc165481810)

[The Menu-btn for the Hamburger Menu 3](#_Toc165481811)

[To Add the Before pseudo code 4](#_Toc165481812)

[To Test the menu code 5](#_Toc165481813)

[To Add the After pseudo code 6](#_Toc165481814)

[Changing the bottom bar to not be the same width 7](#_Toc165481815)

[Animate the hamburger 10](#_Toc165481816)

# Turn on Sass

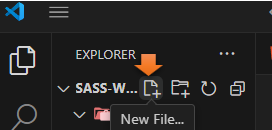
In Visual Studio

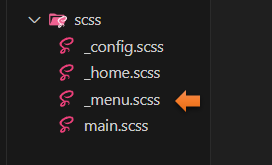
Remember to turn on your Watch Sass



# Creating the Partial Menu File

Here we are going to start working on our menu. To begin this, we will create a new partial file to deal with just the menus in this site, like we did with our home page.





# The code for the \_menu.scss file

.menu-btn {

position:absolute;

z-index: 1;

right: 1rem;

top:1rem;

height: 20px;

width: 28px;

cursor: pointer;

@include transition-ease;

&\_\_burger {

position:absolute;

right: 0;

top: 0.5;

height: 3px;

width: 28px;

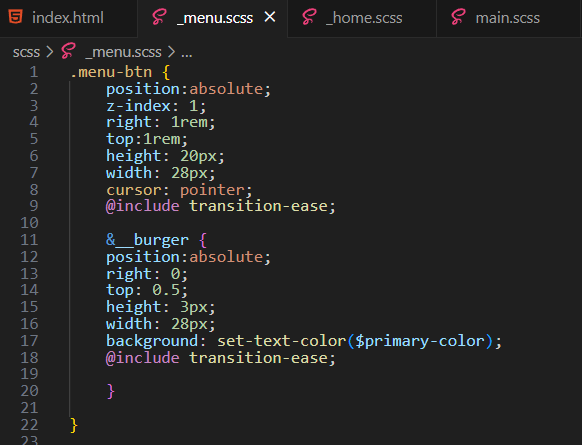
background: set-text-color($primary-color);

@include transition-ease;

}

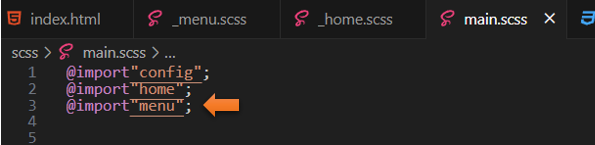
}

Notice how we next the burger menu inside of the menu-btn class using the &\_\_



# @include the partial file

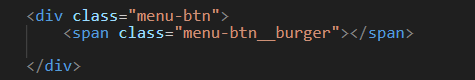
Remember to Include this partial inside of the main.scss



# The Menu-btn for the Hamburger Menu

Go to the index.html file.

If you look at the html code, you will see that the menu button is basically a wrapper. We will be using this as a click event. Within the menu-btn, we will have our bars for the hamburger.



Go to the \_menu partial file.

When we wrote the code in the menu partial for the nested hamburger, all that we have configured, so far was the middle bar. You will notice in the menu partial, that we set this middle bar to a width of 28px. Now we are going to use pseudo elements to create the other bars of the hamburger like this.

**\*** **Careful**, on this one, if you do not put in both a width and a height for our hamburger line, it will not show up

# To Add the Before pseudo code

**\*Watch it**, do not include any space when you write the &::before or it will not work.

&::before {

content: '';

position: absolute;

top: -8px;

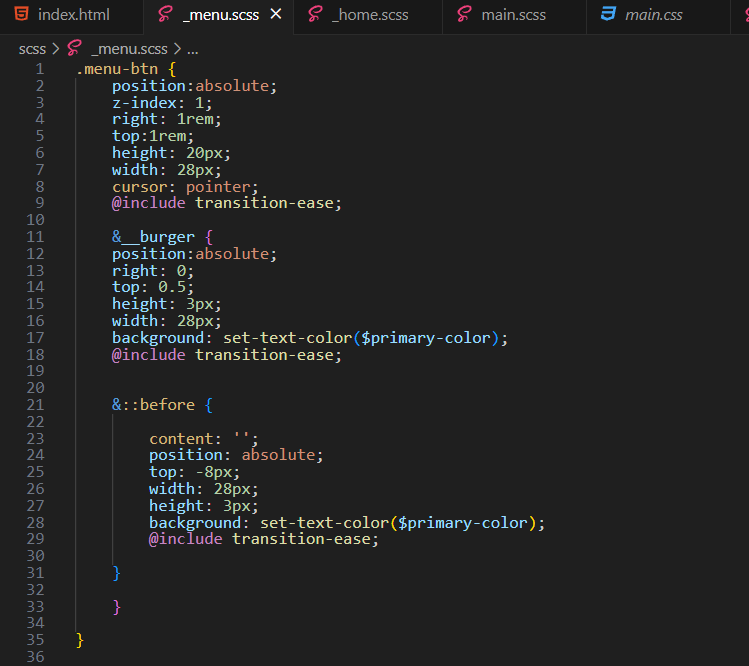
width: 28px;

height: 3px;

background: set-text-color($primary-color);

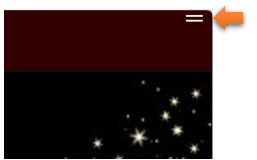
@include transition-ease;

}



# To Test the menu code

Go to the index.html file.



# To Add the After pseudo code

Instead of 8 up using the negative 8, we want to use 8 px down, so just remove the – sign in front of the 8 to achieve this.

**\*Watch it**, do not include any space when you write the &::after or it will not work.

&::after {

content: '';

position: absolute;

top: 8px;

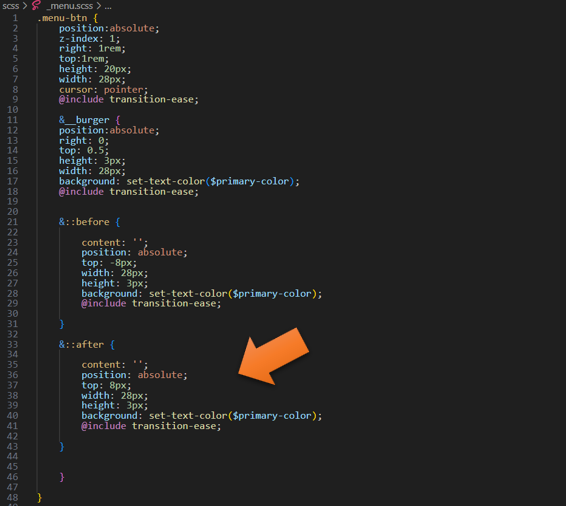
width: 28px;

height: 3px;

background: set-text-color($primary-color);

@include transition-ease;

}





# Changing the bottom bar to not be the same width

This is easy just change the width in the &::after to be:

width: 20px;

.menu-btn {

position:absolute;

z-index: 1;

right: 1rem;

top:1rem;

height: 20px;

width: 28px;

cursor: pointer;

@include transition-ease;

&\_\_burger {

position:absolute;

right: 0;

top: 0.5;

height: 3px;

width: 28px;

background: set-text-color($primary-color);

@include transition-ease;

&::before {

content: '';

position: absolute;

top: -8px;

width: 28px;

height: 3px;

background: set-text-color($primary-color);

@include transition-ease;

}

&::after {

content: '';

position: absolute;

top: 8px;

width: 20px;

height: 3px;

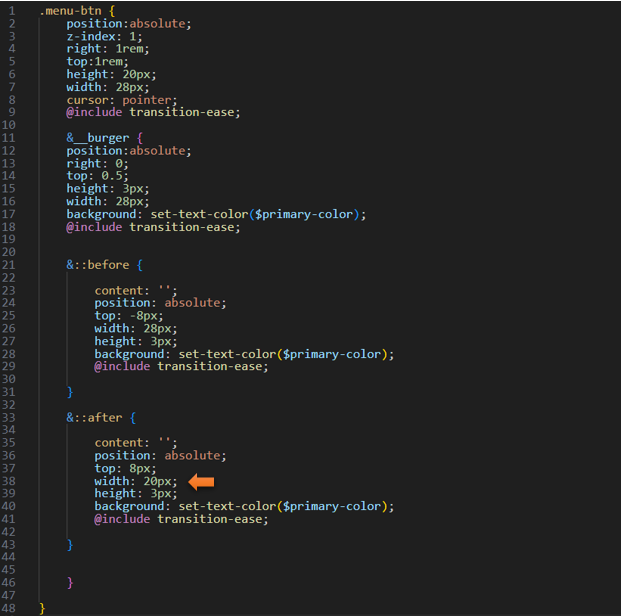
background: set-text-color($primary-color);

@include transition-ease;

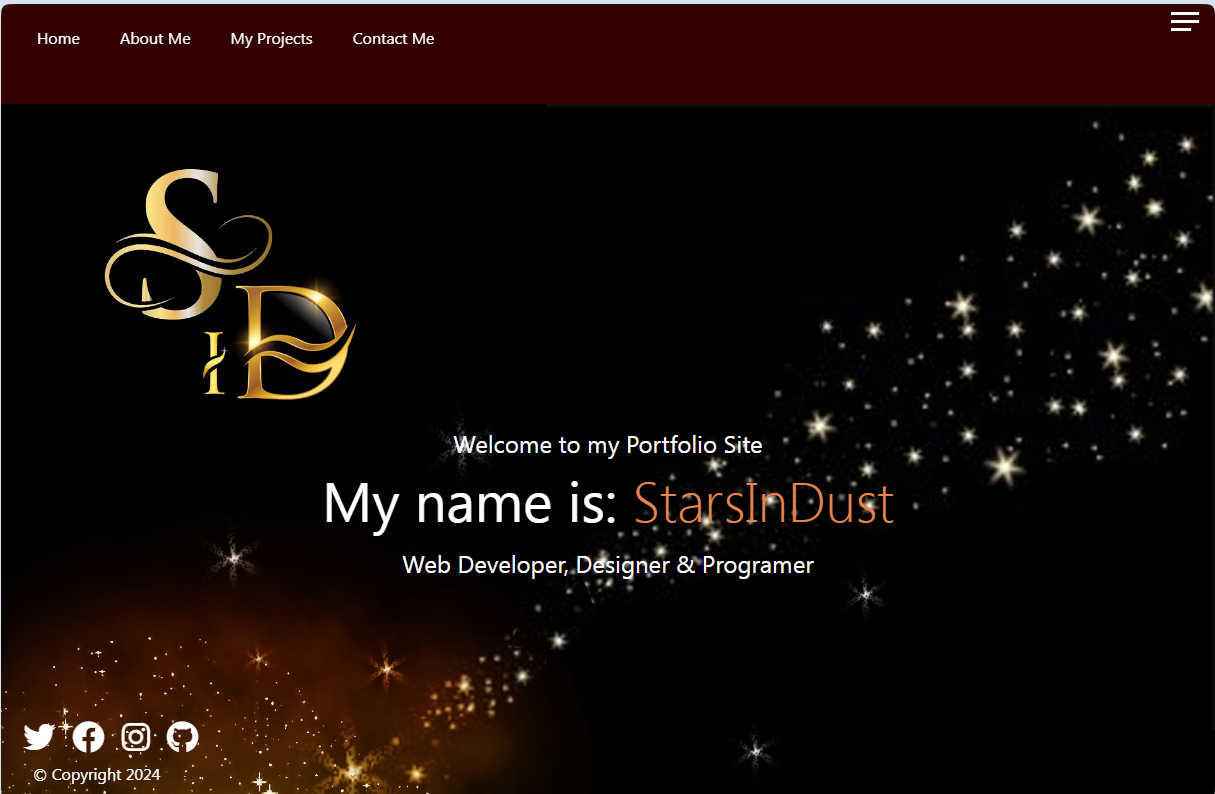
}

}

}







# Animate the hamburger

We will be using Java script to make the hamburger menu interactive

&::after **{**

content: '';

position: absolute;

top: 8px;

width: 20px;

height: 3px;

background: set-text-color($primary-color);

@include transition-ease;

&.open**{**

transform: rotate(720deg);

background: transparent;

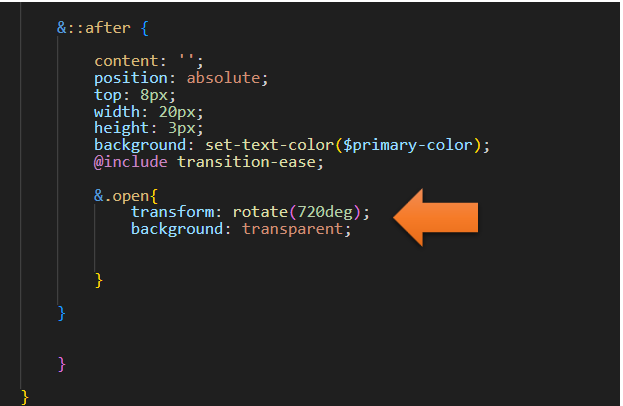
**}**

**}**

}

}

The last two closing curly braces will be closing out the code above the &::after code



\*Watch it, right now, this code will not do anything. In our next tutorial, we will be looking at doing the JavaScript for it.