9 Formatting Main Menu Links

Contents

[Turn on Sass 1](#_Toc165720174)

[Writing the code for the Navigation Links 1](#_Toc165720175)

[&\_\_item Code 2](#_Toc165720176)

[Writing the JavaScript code for &\_\_item 3](#_Toc165720177)

[Test The index.html page 5](#_Toc165720178)

[More code inside of the Menu nav 6](#_Toc165720179)

[Save and Test index.html 7](#_Toc165720180)

[Slowing down the speed the menu links come onto the page 8](#_Toc165720181)

[Allowing a variable to work inside of a function 8](#_Toc165720182)



# Turn on Sass

In Visual Studio

Remember to turn on your Watch Sass

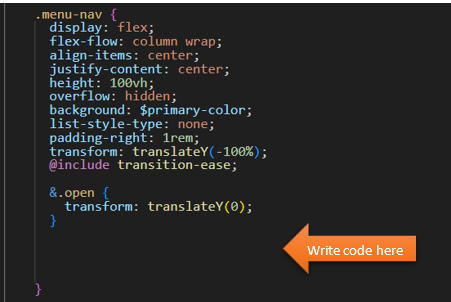


# Writing the code for the Navigation Links

Inside of the \_menu.scss



We are going to still be inside of the .menu-nav class rule.



Even though the menu comes in from the top of the page, we want the menu items to come in from the right. The links will start off from all of the way off the right side of page, when the hamburger button is pushed.

When we add the &open we will move them from the initial position and in from the right. We will also add an active class.

# &\_\_item Code

&\_\_item{

transform: translateX(100vw);

@include transition-ease;

&.open{

transform: translateX(0);

}

&.active > a {

color: $secondary-color;

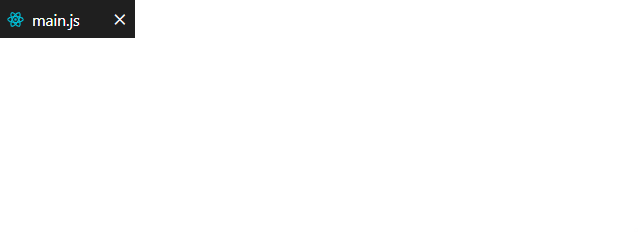
}

}



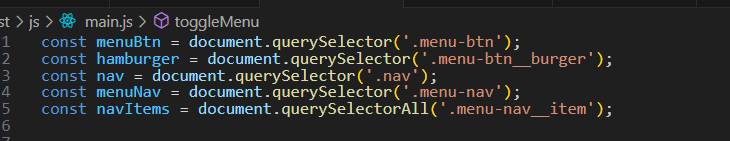
# Writing the JavaScript code for &\_\_item

in order to write the JavaScript code, we of course need to be in the JavaScript file.



Add this line to the constants at the top of the JavaScript file

const navItems = document.querySelectorAll('.menu-nav\_\_item');



These nav items also need to have the class of open added and removed from the class list.

function toggleMenu() {

if(!showMenu) {

hamburger.classList.add('open');

nav.classList.add('open');

menuNav.classList.add('open');

navItems.forEach(item => item.classList.add('open'));

showMenu = true;

} else {

hamburger.classList.remove('open');

nav.classList.remove('open');

menuNav.classList.remove('open');

navItems.forEach(item => item.classList.remove('open'));

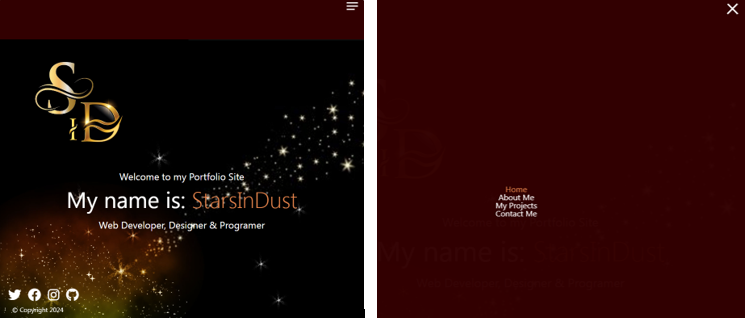
showMenu = false;

}

}



# Test The index.html page



# More code inside of the Menu nav





&\_\_link {

display: inline-block;

font-size: 2rem;

text-transform: uppercase;

padding: 2rem 0;

font-weight: 300;

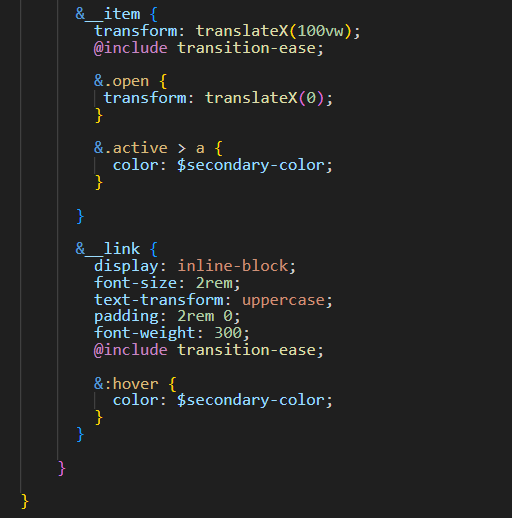
@include transition-ease;

&:hover {

color: $secondary-color;

}

}



# Save and Test index.html

Click on the hamburger menu now and you should see this



# Slowing down the speed the menu links come onto the page

Still inside of the \_menu.scss



You want to write this at the bottom of the page, after those 4 closing curly braces

@for $i from 1 through 4 {

.menu-nav\_\_item:nth-child(#{$i}) {

transition-delay: ($i \* 0.1s) + 0.15s;

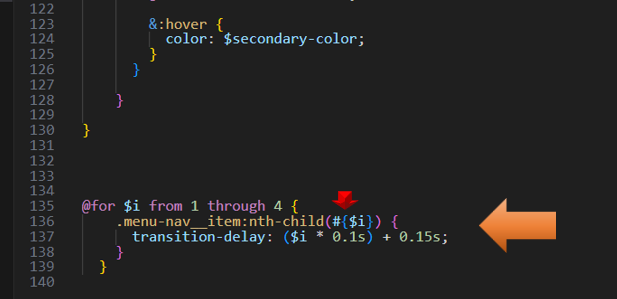
}

}

# Allowing a variable to work inside of a function

We can’ t just write a variable inside of the @for function. It will only show you an error if you do. The way that we need to do this is to wrap the variable inside of a pound sign with curly braces around the variable.

This code will allow it to loop through each link and then do the transition delay to slow it down and stagger coming in. You will however, notice that on the transition-delay, it will allow us to write the variable without the wrap of the # sign and curly braces.



Test the index.html to see the links come in one at a time.

