10 Extensions

Contents

[What is an Extension? 1](#_Toc164403310)

[Change your two Paragraph tags to be different 1](#_Toc164403311)

[Change the name of your nested paragraph in main.scss file 1](#_Toc164403312)

[The new code representing the .main class 2](#_Toc164403313)

[Test the extended copied styles 2](#_Toc164403314)

[To add additional styles to the second class 3](#_Toc164403315)

[Test the extended styles 3](#_Toc164403316)

Turn on [Watch SASS](file:///C:\xampp\htdocs\Enlightenment\Articles\2024\6-SASS-Or-SCSS\10-Extensions\2%20Creating%20the%20Project.docx)

1

# What is an Extension?

In SCSS, we can extend an element. This is basically what is called inheritance in other languages. And so, in SASS, if we are to extend an element, that element would then inherit all of the styles from the parent element.

# Change your two Paragraph tags to be different

Go to the index.html file.

Change the classes on our two paragraphs to be different from one another. Now remember as soon as you make this change to your paragraphs, and tutorial that we have done before will no longer be working.

2

# Change the name of your nested paragraph in main.scss file

Next go into the .main class and change this (#{&} \_\_p, to be p1

3

So, now we will be extending all of the styles of p1 onto p2. This way we could change one or two things on p2 without having to re-type all of the styles, especially if the 2 paragraphs will have some of the styles being duplicated in code. This is a more efficient way of approaching things.

We will be adding some new code here for p2 to represent the second paragraph.

#{&}\_\_p2 {

@extend .main\_\_p1;

}

# The new code representing the .main class

Now start writing this nested paragraph code to represent p2

.main {

    @include flexCenter(row);

    padding-top: 5%;

    width: 75%;

    margin: 0 auto;

#{&}\_\_p1 {

    font-weight:weight(medium);

    &:hover {

        color:chocolate;

    }

}

#{&}\_\_p2 {

@extend .main\_\_p1;

}

4

If you test your code now you will find that both paragraphs will behave exactly the same, because we have not made any new changes to p2, we have merely copied all of the styles onto it from p1

# Test the extended copied styles

Go to index.html to test the styles

5

# To add additional styles to the second class

This is simple, once we have copied p1 into p2 using @extend, we can just write the additional rules that we want our p2 to be able to file.

.main {

@include flexCenter(row);

padding-top: 5%;

width: 75%;

margin: 0 auto;

#{&}\_\_p1 {

font-weight:weight(medium);

&:hover {

color:chocolate;

}

}

#{&}\_\_p2 {

@extend .main\_\_p1;

&:hover {

color:$accent-color;

}

}

6

# Test the extended styles

Go to index.html to test the styles

And it is listening to our new extended style that we gave the hover in the second paragraph

7