Exercise (Instructions): Scss

Objectives and Outcomes

In this exercise you will learn to write Scss code and then automatically transform it into the corresponding CSS code. At the end of this exercise you will be able to:

- Write Scss code using many of the features of Scss
- Automatically convert the Scss code into CSS

Adding Scss Variables

- Open the *conFusion* project in a text editor of your choice. In the css folder, create a file named *styles.scss*. We will add the Scss code into this file.
- Add the following Scss variables into the file:

```
$lt-gray: #ddd;
$background-dark: #512DA8;
$background-light: #9575CD;
$background-pale: #D1C4E9;

// Height variables
$carousel-item-height: 300px;

```

We have just added a few color and a height variable. We will make use of these variables while defining the classes.

Scss Mixins

- Next we add a mixin into the file as follows:

```
@mixin zero-margin($pad-up-dn, $pad-left-right) {
  margin:0px auto;
  padding: $pad-up-dn $pad-left-right;
}

```

We will make use of this to define several row classes next.

- Using the variables and Mixin class that we defined earlier, add the following row classes to the file:

```
.row-header{
    @include zero-margin(0px,0px);
}

.row-content {
    @include zero-margin(50px,0px);
    border-bottom: 1px ridge;
    min-height:400px;
}

.footer{
    background-color: $background-pale;
    @include zero-margin(20px, 0px);
}
.jumbotron {
    @include zero-margin(70px,30px);
    background: $background-light ;
    color:floralwhite;
}

address{
    font-size:80%;
    margin:0px;
    color:#0f0f0f;
}

body{
    padding:50px 0px 0px 0px;
    z-index:0;
}

.navbar-inverse {
     background-color: $background-dark;
}
.tab-content {
    border-left: 1px solid $lt-gray;
    border-right: 1px solid $lt-gray;
    border-bottom: 1px solid $lt-gray;
    padding: 10px;
}

```

Note the use of the variables and the mixin with various parameters in defining the classes.

Nesting Selectors

- Next we add a carousel class to illustrate the use of nesting of classes in Scss, as follows:

```
.carousel {
    background:$background-dark;

    .carousel-item {
        height: $carousel-item-height;
        img {
            position: absolute;
            top: 0;
            left: 0;
            min-height: 300px;
        }
    }
}
#carousel-button {
    right:0px;
    position: absolute;
    bottom: 0px;
}
```

Installing and using the node-sass module

- Now we install the node module to support the compilation of the Scss file to a CSS file. To do this, type the following at the command prompt:

```
1       npm install --save-dev node-sass
2
```

This will install the *node-sass* NPM module into your project and also add it as a development dependency in your package.json file.

- Next open your package.json file and add the following line into the scripts object there. This adds a script to enable the compilation of the Scss file into a CSS file:

```
1      "scss": "node-sass -o css/ css/"
```

- In order to transform the Scss file to a CSS file, type the following at the prompt:

```
1       npm run scss
```

- You can now do a Git commit with the message "Sass".

Conclusions

In this exercise you learnt to write Scss code and then automatically generating the CSS file by compiling the Scss code.

Mark as completed