

SCSS Fundamentals

To install SASS (Windows):

```
npm install -g sass
```

To convert SASS to CSS file:

```
sass style.scss style.css
```

Variable:

```
$primary-color: blue;
$secondary-color: orange;
$warning-color: yellow;
$danger-color: red;
$success-color: green;
$title-font-size: 20px;

.text{
  color: $danger-color;
}

body{
  background-color: $secondary-color;
}
```

To convert SASS to CSS file:

```
sass variable.scss variable.css
```

Converted CSS:

```
.text {
  color: red;
}

body {
  background-color: orange;
}
```

```
/*# sourceMappingURL=variable.css.map */
```

```
<body>
  <h1 class="text">Hello</h1>
</body>
```

Import and Partial file:

Create another file (partial file) for variables, for partial we write like this: **__variable.scss**

```
$primary-color: blue;
$secondary-color: rgb(0, 217, 255);
$warning-color: yellow;
$danger-color: red;
$success-color: green;
$title-font-size: 20px;
```

import that partial file to main file (style. scss)

```
@import "variable";

.text{
  color: $danger-color;
}

body{
  background-color: $secondary-color;
}
```

Mixin:

Just like variables, if we have any code block which will be repeated, we can use that code block with **@mixin** keyword and can use this mixin with **@include** repeatedly.

Created a partial file **_mixin.scss**

```
@mixin btn-default-style {
  border: 2px solid $danger-color;
  color: #fff;
  padding: 40px 50px;
  border-radius: 5px;
```

```
    font-size: 16px;
}
```

main SASS file:

```
@import "variable";
@import "mixin";

.btn1{
  @include btn-default-style();
  background-color: $primary-color;
  cursor: grab;
}

.btn2 {
  @include btn-default-style();
  background-color: $secondary-color;
  cursor: pointer;
}

.btn3 {
  @include btn-default-style();
  background-color: $warning-color;
  cursor: not-allowed;
}

.btn4 {
  @include btn-default-style();
  background-color: $danger-color;
  cursor: zoom-in;
}
```

Converted CSS file:

```
.btn1 {
  border: 2px solid red;
  color: #fff;
  padding: 40px 50px;
  border-radius: 5px;
  font-size: 16px;
  background-color: blue;
  cursor: grab;
}

.btn2 {
  border: 2px solid red;
  color: #fff;
  padding: 40px 50px;
```

```

border-radius: 5px;
font-size: 16px;
background-color: rgb(0, 217, 255);
cursor: pointer;
}

.btn3 {
border: 2px solid red;
color: #fff;
padding: 40px 50px;
border-radius: 5px;
font-size: 16px;
background-color: yellow;
cursor: not-allowed;
}

.btn4 {
border: 2px solid red;
color: #fff;
padding: 40px 50px;
border-radius: 5px;
font-size: 16px;
background-color: red;
cursor: zoom-in;
}

/*# sourceMappingURL=mixin.css.map */

```

```

<body>
  <button class="btn1">Submit1</button>
  <button class="btn2">Submit2</button>
  <button class="btn3">Submit3</button>
  <button class="btn4">Submit4</button>
</body>

```

Mixin with Parameters:

```

@mixin default-button-style($color, $background-color, $cursor) {
color: $color;
border: 5px solid $fourth-color;
border-radius: 5px;
padding: 20px 25px;
background-color: $background-color;
cursor: $cursor;
}

```

main SASS file:

```
@import 'variable';
@import 'mixin';

.btn1{
  @include default-button-style(black, $fifth-color, pointer);
}

.btn2{
  @include default-button-style(black, $third-color, grab);
}

.btn3{
  @include default-button-style(white, $fourth-color, not-allowed);
}

.btn4{
  @include default-button-style(blue, $second-color, default);
}
```

Inheritance (like Mixin, used for code repetition purpose):

```
.button-base-style{
  border: 5px solid black;
  border-radius: 5px;
  color: white;
}

.button-primary{
  @extend .button-base-style;
  background-color: blue;
}

.button-secondary{
  @extend .button-base-style;
  background-color: red;
}
```

```
.button-base-style, .button-secondary, .button-primary {
  border: 5px solid black;
  border-radius: 5px;
  color: white;
}
```

```
.button-primary {  
  background-color: blue;  
}
```

```
.button-secondary {  
  background-color: red;  
}
```

```
<body>  
  <button class="button-primary">Submit</button>  
  <button class="button-secondary">Cancel</button>  
</body>
```

Mathematical Operations:

```
$default-font-size: 16px;
```

```
.text-1{  
  font-size: $default-font-size + 10px;  
}
```

```
.text-2 {  
  font-size: $default-font-size - 10px;  
}
```

```
.text-3 {  
  font-size: $default-font-size * 2;  
}
```

```
.text-4 {  
  font-size: $default-font-size / 2;  
}
```

```
.text-1 {  
  font-size: 26px;  
}
```

```
.text-2 {  
  font-size: 6px;  
}
```

```
.text-3 {  
  font-size: 32px;  
}
```

```
.text-4 {  
  font-size: 8px;  
}
```

```
<body>  
  <div class="my-div">  
    <p class="text-1">text1</p>  
    <p class="text-2">text2</p>  
    <p class="text-3">text3</p>  
    <p class="text-4">text4</p>  
  </div>  
</body>
```

For Loop:

```
$default-font-size: 16px;  
  
@for $i from 1 through 10{  
  .text-#{$i} {  
    font-size: $default-font-size + $i;  
  }  
}
```

```
.text-1 {  
  font-size: 17px;  
}  
  
.text-2 {  
  font-size: 18px;  
}  
  
.text-3 {  
  font-size: 19px;  
}  
  
.text-4 {  
  font-size: 20px;  
}  
  
.text-5 {  
  font-size: 21px;  
}  
  
.text-6 {  
  font-size: 22px;
```

```

}

.text-7 {
  font-size: 23px;
}

.text-8 {
  font-size: 24px;
}

.text-9 {
  font-size: 25px;
}

.text-10 {
  font-size: 26px;
}

```

```

<body>
  <div class="my-div">
    <p class="text-1">text1</p>
    <p class="text-2">text2</p>
    <p class="text-3">text3</p>
    <p class="text-4">text4</p>
    <p class="text-5">text5</p>
    <p class="text-6">text6</p>
    <p class="text-7">text7</p>
    <p class="text-8">text8</p>
    <p class="text-9">text9</p>
    <p class="text-10">text10</p>
  </div>
</body>

```

If-else:

```

$dark-mode: false;

.navbar{
  @if $dark-mode {
    background-color: black;
    color: white;
  }

  @else{
    background-color: white;
    color: black;
  }
}

```



```
    }  
}
```

```
.navbar {  
  background-color: white;  
  color: black;  
}
```