

Task 1: introduction to SASS

Advantages of SASS:

With a feature like nesting, It allows write a more organized stylesheet.

It allows code reusability hence minimizing the need of repeating codes.

Task 2: Installing SASS

To install SASS via Visual Studio Code the following are the steps:

Click on the Extensions Icon on the VS Code window.

Search for “Sass(.sass only)” the click install to install it.

Next Search for “Live Sass Compiler”. Install it. This help to compile live SASS into CSS.

Task 3: Variables in Sass

Assignment included in Zip file

[style.scss](#)

```
$color: red;
$bg-color: gray;
$primary-color: black;
$radius: 5px;
$border-style: solid;

body{
  background-color: $bg-color;
  font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
.heading {
  color: $color;
}
.par{
  color: $primary-color;
}
.button-next {
  color: white;
  font-style:italic;
  font-weight: 900;
  height: 30px;
  background-color: $color;
  border: 2px $border-style $primary-color;
  border-radius: $radius;
}
```

Task 4: Operators

How Operators perform calculations in SASS.

SASS operators can be used to add up values to create new ones, scale up values by multiplying. We can divide, find difference among other operations.

For instance, you may create a variable **\$width** and give it some default value. Then suppose inside our HTML we have an element that requires a width as twice as much as the value set for the variable and another element requires a width that is half the set width. In this case, for the first element will give the width value as **\$width*2** and for the second one (**\$width / 2**).

Task 5: Nesting in Sass

File [styles.scss](#) included in the Zip file

```
@mixin button($width,$height,$radius){
  width: $width;
  height: $height;
  background-color: red;
  border-radius: $radius;
  color: white;
  font-weight: 700;
  border:none;
  padding: 10px;
}
$text: 1rem;
body{
  color: white;
  font-family: sans-serif;
  background-color:rgb(59, 59, 59);
  // font-size: 16px;
  .header{
    height: auto;
    display: block;
    background-color: rgba($color: #000000, $alpha: .6);
    img{
      position: absolute;
      width: 100%;
      height: auto;
      z-index: -1;
    }
  }
  .nav{
    display: flex;
    justify-content: space-between;
    h1{
      color: red;
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    }
  }
  .sign-out{
    @include button(auto,50px, 5px);
  }
}
```

```

        font-size: 18px;
    }
}
.web-title{
    display: flex;
    flex-direction: column;
    align-items: center;
    h1{
        font-size: $text*3;
    }
    p{
        font-size: $text*1.8;
    }
    .finish-sign-up{
        @include button(auto, auto,0 );
        margin-bottom:10px ;
        font-size: $text*2;
    }
}
}
}
.watch-section{
    background-color: #000000;
    height: 500px;
    display: flex;
    direction: row;
    justify-content: space-around;
    padding: 8%;
    margin-bottom: 10px;
    h1{
        font-size: $text*3;
    }
    p{
        font-size: $text*1.8;
    }
}
}
}
}

```

Task 6: Mixin in Sass and Task 7: Parameters in Sass

File [shadow.scss](#) included in the zipped folder

```

@mixin box_des($width,$bgcolor){
    box-shadow:10px 10px 6px rgb(88, 88, 88);
    width: $width;
    background-color: $bgcolor;
    padding: 10px;
}
.box{
    @include box_des(200px, rgb(0, 102, 255) );
}

```

```
}  
.box-blue{  
  @include box_des(150px, gray);  
  h1 {  
    color: red;  
  }  
}  
.box-brown {  
  @include box_des(auto,auto );  
  p{  
    color: brown;  
  }  
}  
  
.my-boxes{  
  display: flex;  
  flex-direction: row;  
  justify-content: space-evenly;  
}
```