



edureka!
a Veranda Enterprise



Full Stack Web Development Program

Activate Windows
Go to Settings to activate Windows.



edureka!

a Veranda Enterprise



Day 11 - SASS

Titles

- Introduction to Sass
- Working of Sass
- Installing Sass
- Variables in Sass
- Operators in Sass
- Nesting in Sass
- Mixins in Sass
- Parameters in Sass
- Functions in Sass



Activate Windows
Go to Settings to activate Windows.

Learning Objectives

By the end of this module, you will be able to:

- Learn basics of Sass, including how it extends the functionality of CSS and simplifies the process of writing and maintaining stylesheets
- Learn how to use nesting in Sass to organize their stylesheets and write more efficient CSS
- Learn about mixins, which allow them to define reusable blocks of CSS code



Activate Windows
Go to Settings to activate Windows.



Syntactically Awesome Style Sheets

Activate Windows
Go to Settings to activate Windows.

Introduction to Sass

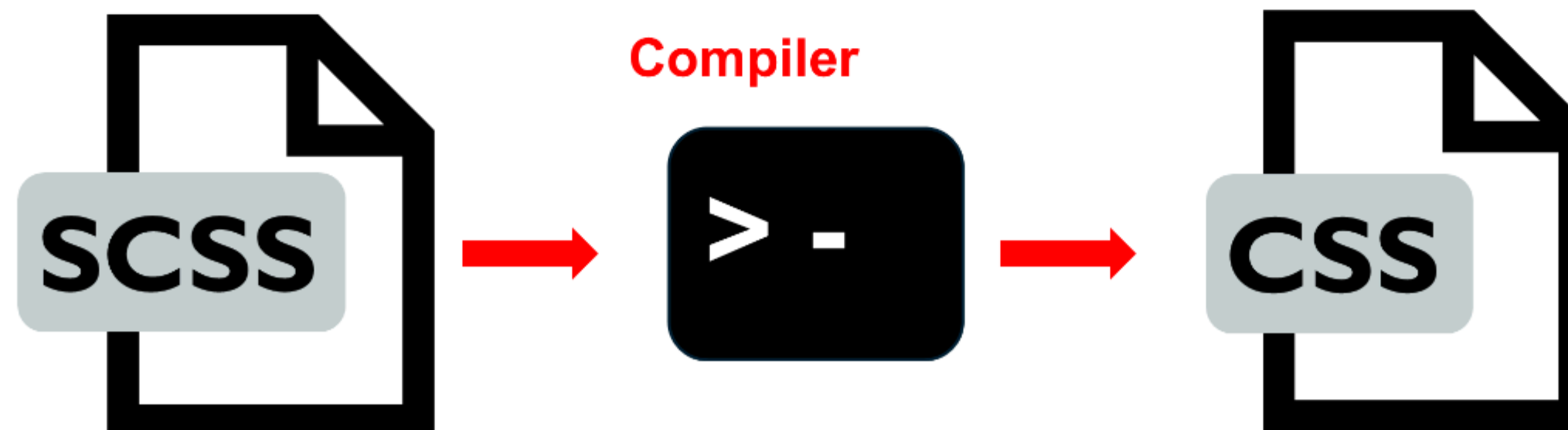
Sass (Syntactically Awesome Stylesheets) is an extension of CSS that adds powerful features and functionalities to the standard CSS language.

- **Benefits:** It enhances the way stylesheets are structured and maintained, offering a more organized and efficient approach to styling web applications.
- **Compilation:** Sass code is written in a more concise syntax and needs to be compiled into standard CSS to be interpreted by web browsers.



Introduction to Sass (contd.)

- Sass, a preprocessor-based scripting language, is compatible with all versions of CSS.
- It was initially designed by Hampton Catlin and developed by Natalie Weizenbaum. Later, Weizenbaum, along with Chris Epstein, continued working on its development.
- One of the key advantages of using Sass is its ability to interpret or compile into CSS. This feature allows for greater flexibility and efficiency in managing larger stylesheets.
- Additionally, It promotes reusability, enabling the minimization of stylesheets for improved maintainability.

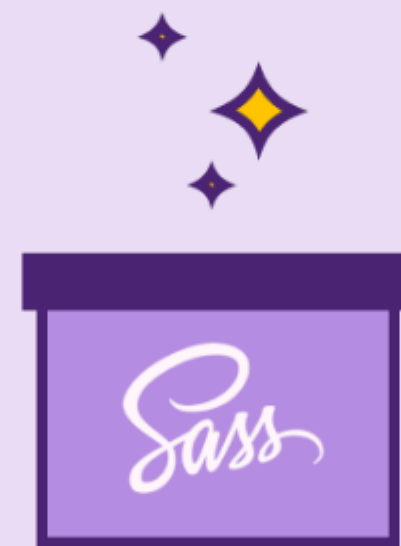


Introduction to Sass (contd.)

style.scss

```
1 $font-stack: Helvetica, sans-serif;  
2 $primary-color: #333;  
3  
4 body {  
5     font: 100% $font-stack;  
6     color: $primary-color;  
7 }  
8  
9  
10
```

Some Magic



style.css

```
1 body {  
2     font: 100% Helvetica, sans-serif;  
3     color: #333;  
4 }  
5  
6  
7  
8  
9  
10
```

Activate Windows
Go to Settings to activate Windows.

Working of Sass

- Compiling Sass code into CSS involves a process where a compiler reads Sass files and generates corresponding CSS files.
- Sass code can be written in **two formats**, .scss (Sassy CSS) and .sass (older indented syntax), both of which offer advanced features.
- However, .scss is more similar to traditional CSS than .sass.
- Once the Sass code is compiled, it is transformed into standard CSS that browsers can interpret and apply to the webpage.



Activate Windows
Go to Settings to activate Windows.

Features of Sass

- SASS provides many of the benefits of programming with code but adds the ability to apply it to stylesheets.
- In the following slide, we will explore the primary features of SASS:
 - Variables
 - Nesting
 - Mixins
 - Extends/Inheritance
 - Partials & Importing
 - Function
 - Control Directives
 - Operators

Activate Windows
Go to Settings to activate Windows.

Variables in Sass

- Variables are containers to store information that can be used later. Variable is a place in a memory that can store the data. The data can be String, Boolean, List, Color, etc.
- In sass, we use the \$(dollar) symbol before the variable's name to create a variable.
- The syntax for declaring a sass variable is as below:

```
$variable_name : value ;
```

Example:

```
$primary-color: #ffffff;
```

In this example, we've defined a variable named **\$primary-color** and assigned it the value **#ffffff**, which is having color white.

Activate Windows
Go to Settings to activate Windows.

Variables in Sass (contd.)

- In the given example we declared four variables, namely `color_1`, `color_2`, `color_3` & `color_4` with values.
- Transpiler will translate and compile the entire sass code. This will return normal CSS with variable-value, as shown below.

```
#p-1 {  
  
    background-color: black;  
  
}
```

- It does the same for all other variables.

```
// define variables for the primary colors  
$color_1: black;  
$color_2: rgb(221, 36, 36);  
$color_3: rgb(45, 192, 31);  
$color_4: blanchalmond;  
  
#p-1{  
    background-color: $color_1  
}  
  
#p-2{  
    background-color: $color_2  
}  
  
#p-3{  
    background-color: $color_3  
}  
  
#p-4{  
    background-color: $color_4;  
}
```

Activate Windows
Go to Settings to activate Windows.

Variables in Sass (contd.)

- Variables can hold various values, including numeric values that you can use in calculations.

Example:

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

.paragraph {
  font-size: $base-font-size * 1.2;
  // Font size increased by 20%
}
```

- Here, we're using the **\$base-font-size** variable in a calculation to achieve a font size that's **20%** larger than the base size.

Variables in Sass (contd.)

- **Benefits:** Using variables in your styles offers several benefits, including enhancing consistency by allowing you to update a single value (the variable) instead of modifying every instance of that value. This makes your styles more maintainable and less prone to errors.
- **Global Scope:** The global scope of variables in stylesheets allows them to be defined at the top of the stylesheet but accessed from any part of the code. This ensures consistency in the value of variables across different sections of the code.
- **Preprocessing Magic:** Sass is a preprocessor, which means that the code you write in Sass is compiled into regular CSS before being sent to web browsers. As a result, the browser only sees the final CSS output, not the original Sass code.

Activate Windows
Go to Settings to activate Windows.