**Theming in CSS** refers to applying a consistent design (colors, fonts, backgrounds) across a website or application while allowing customization, such as switching between light and dark modes. Themes are typically created by defining a set of **design variables** (like colors and fonts), which can be swapped dynamically to change the appearance of the UI.

Using **Sass** (Syntactically Awesome Stylesheets), themes can be easily implemented and managed by utilizing **variables** for common styles like background color, text color, and button styles.

**Implementing Theming with Sass**

Sass allows you to define variables for a theme, which can then be reused throughout your styles. You can create multiple themes (e.g., light and dark) by defining different sets of variables for each theme.

**Example: Theme Switcher (Light and Dark)**

We will create a theme switcher that allows users to toggle between **light** and **dark** themes using Sass.

**1. Sass Code for Light and Dark Themes**

We define the variables for both themes and then apply them to the appropriate elements.

**Defining Theme Variables in Sass:**

scss

Copy code

// Define light theme variables

$light-bg-color: #ffffff;

$light-text-color: #333333;

$light-button-bg: #3498db;

$light-button-text: #ffffff;

// Define dark theme variables

$dark-bg-color: #1e1e1e;

$dark-text-color: #f1f1f1;

$dark-button-bg: #555555;

$dark-button-text: #ffffff;

// Mixin to apply theme styles

@mixin theme($bg-color, $text-color, $button-bg, $button-text) {

body {

background-color: $bg-color;

color: $text-color;

}

button {

background-color: $button-bg;

color: $button-text;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

&:hover {

opacity: 0.8;

}

}

}

// Default light theme styles

body.light {

@include theme($light-bg-color, $light-text-color, $light-button-bg, $light-button-text);

}

// Dark theme styles

body.dark {

@include theme($dark-bg-color, $dark-text-color, $dark-button-bg, $dark-button-text);

}

**2. HTML Structure**

The HTML structure includes a button for toggling the theme and an example button element to showcase the theme change.

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="stylesheet" href="styles.css">

<title>Theme Switcher</title>

</head>

<body class="light">

<button id="theme-switcher">Switch Theme</button>

<button class="example-btn">Example Button</button>

<script>

const themeSwitcher = document.getElementById('theme-switcher');

const body = document.body;

themeSwitcher.addEventListener('click', () => {

if (body.classList.contains('light')) {

body.classList.remove('light');

body.classList.add('dark');

} else {

body.classList.remove('dark');

body.classList.add('light');

}

});

</script>

</body>

</html>

**3. Compiled CSS Output**

css

Copy code

body.light {

background-color: #ffffff;

color: #333333;

}

body.light button {

background-color: #3498db;

color: #ffffff;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

}

body.light button:hover {

opacity: 0.8;

}

body.dark {

background-color: #1e1e1e;

color: #f1f1f1;

}

body.dark button {

background-color: #555555;

color: #ffffff;

padding: 10px 20px;

border: none;

border-radius: 4px;

cursor: pointer;

}

body.dark button:hover {

opacity: 0.8;

}

**Explanation of Sass Variables and Mixins**

* **Variables**: Theme variables ($light-bg-color, $dark-bg-color, etc.) hold values like colors and are used across the styles for both themes. This allows you to easily change the theme by simply switching variable values.
* **Mixins**: The @mixin theme(...) defines a reusable block of CSS that accepts variables as parameters (background color, text color, button styles). You can apply this mixin to different themes by passing different values for each theme.
* **Theme Switching**: The **JavaScript** toggles the light and dark classes on the <body> element, which switches between the two themes. Each theme has its own styles based on the applied variables, making it easy to implement the theme change.