**CSS variables**, also known as **custom properties**, allow you to define reusable values in your stylesheet. They are particularly useful for maintaining consistency and simplifying changes across a website, as you can define values (such as colors, fonts, spacing) in one place and reuse them throughout the CSS.

**How CSS Variables Work:**

* **Definition**: CSS variables are defined using the -- prefix and are typically declared inside a :root selector or within specific elements.
* **Usage**: To use a variable, you reference it with the var() function.

**Syntax:**

css

Copy code

/\* Define a CSS variable \*/

:root {

--primary-color: blue;

--padding: 20px;

}

/\* Use the variable \*/

.element {

background-color: var(--primary-color);

padding: var(--padding);

}

**Advantages of CSS Variables:**

* **Consistency**: Use the same values across multiple elements, ensuring uniform styling.
* **Easier Maintenance**: Changing a variable in one place updates all references throughout the stylesheet.
* **Dynamic Changes**: Variables can be overridden for specific sections or media queries, allowing for flexibility in design.

**Example of CSS Variables in Action:**

html

Copy code

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>CSS Variables Example</title>

<style>

/\* Define CSS variables \*/

:root {

--primary-color: #3498db;

--secondary-color: #2ecc71;

--font-size: 16px;

--padding: 15px;

}

/\* Apply variables \*/

body {

font-size: var(--font-size);

background-color: var(--primary-color);

color: white;

padding: var(--padding);

}

.button {

background-color: var(--secondary-color);

padding: var(--padding);

border: none;

color: white;

font-size: var(--font-size);

cursor: pointer;

}

/\* Modify variables for a specific section \*/

.dark-theme {

--primary-color: #2c3e50;

--secondary-color: #e74c3c;

}

/\* Using the variables in the dark-theme class \*/

.dark-theme body {

background-color: var(--primary-color);

}

.dark-theme .button {

background-color: var(--secondary-color);

}

</style>

</head>

<body>

<h1>CSS Variables Example</h1>

<button class="button">Click Me</button>

<!-- Applying dark theme -->

<div class="dark-theme">

<h2>Dark Theme Section</h2>

<button class="button">Dark Theme Button</button>

</div>

</body>

</html>

**Explanation of the Example:**

1. **Defining Variables**:
   * CSS variables like --primary-color, --secondary-color, --font-size, and --padding are defined inside the :root selector, which makes them available throughout the document.
   * The var() function is used to apply these variables to the body and .button styles.
2. **Using Variables**:
   * The **body** element uses the --primary-color for the background color, --font-size for text size, and --padding for space around the content.
   * The **.button** element uses --secondary-color for its background and the same padding and font size.
3. **Modifying Variables for Specific Sections**:
   * The **.dark-theme** class redefines the values of --primary-color and --secondary-color, applying a different theme for the section. This shows how variables can be overridden within specific parts of a webpage.
4. **Effect of Changing Variables**:
   * If you change --primary-color in the :root selector, it will update the background color of all elements that reference this variable, such as the body. The same happens with other variables like font size and padding. For the .dark-theme section, the modified values will apply only to that section.

**Benefits of CSS Variables:**

* **Consistency**: Using the same variable across multiple elements ensures consistent styling (e.g., all buttons having the same padding or colors).
* **Easy Updates**: Changing a variable value in one place updates all references automatically, reducing the effort needed to make global design changes.
* **Flexibility**: Variables can be modified for specific themes or sections, allowing for easy creation of different styles without duplication.