**ChatGPT**

**What are CSS Variables (Custom Properties)?**

CSS variables, also known as custom properties, allow you to store values that can be reused throughout a stylesheet. They provide a way to create a consistent design and make it easier to manage and update styles across a website.

**Key Features:**

* **Defined with --:** CSS variables are defined using a double hyphen (--) followed by a name.
* **Scoped:** Variables can be scoped to a specific element or inherited throughout the document.
* **Dynamic:** You can change the values of CSS variables, which will immediately update any styles that use them.

**How CSS Variables Work**

1. **Define a Variable:** CSS variables are typically defined within a selector (commonly the :root pseudo-class for global scope) using the syntax:

css

Copy code

--variable-name: value;

1. **Use a Variable:** To use a CSS variable, use the var() function:

css

Copy code

property: var(--variable-name);

**Example of CSS Variables**

Here’s an example HTML document that demonstrates how to define and use CSS variables.

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 in the root \*/

:root {

--main-bg-color: #3498db; /\* Main background color \*/

--main-text-color: #ffffff; /\* Main text color \*/

--padding: 20px; /\* Padding for boxes \*/

--border-radius: 5px; /\* Border radius for boxes \*/

}

body {

font-family: Arial, sans-serif;

background-color: var(--main-bg-color);

color: var(--main-text-color);

margin: 0;

padding: var(--padding);

}

.container {

display: flex;

justify-content: space-around;

flex-wrap: wrap;

}

.box {

background-color: #2ecc71; /\* Box background color \*/

color: #ffffff; /\* Box text color \*/

padding: var(--padding);

margin: var(--padding);

border-radius: var(--border-radius);

width: 200px;

text-align: center;

}

</style>

</head>

<body>

<h1>CSS Variables Example</h1>

<div class="container">

<div class="box">Box 1</div>

<div class="box">Box 2</div>

<div class="box">Box 3</div>

</div>

<script>

// Changing CSS variable values dynamically

const rootStyles = document.documentElement.style;

setTimeout(() => {

rootStyles.setProperty('--main-bg-color', '#e74c3c'); // Change main background color

rootStyles.setProperty('--main-text-color', '#000000'); // Change main text color

}, 2000);

</script>

</body>

</html>

**Explanation of the Example**

1. **Defining Variables**:
   * In the :root selector, four CSS variables are defined:
     + --main-bg-color: Sets the main background color.
     + --main-text-color: Sets the main text color.
     + --padding: Sets padding for boxes.
     + --border-radius: Sets the border radius for boxes.
2. **Using Variables**:
   * The variables are used throughout the CSS. For example, background-color: var(--main-bg-color); applies the main background color to the body.
3. **Dynamic Changes**:
   * A JavaScript snippet uses setTimeout to change the values of the --main-bg-color and --main-text-color variables after 2 seconds. This demonstrates how changing a variable affects styles instantly without needing to rewrite any CSS rules.

**Effects of Changing Variable Values**

When the CSS variables change:

* The background color of the body will change from blue (#3498db) to red (#e74c3c).
* The text color will change from white (#ffffff) to black (#000000).

This example shows how CSS variables enhance maintainability and consistency across stylesheets, allowing for quick updates and modifications.