

JavaScript Code Color Flipper Example

Project: Color Flipper

Step 1: HTML Structure

Create an HTML file named index.html and set up the basic structure.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Color Flipper</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div class="container">
    <h1>Color Flipper</h1>
    <div class="color-box" id="colorBox"></div>
    <button id="flipButton">Flip Color</button>
  </div>
  <script src="script.js"></script>
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
</body>  
</html>
```

Step 2: CSS Styling

Create a CSS file named styles.css for basic styling.

```
body {  
    font-family: Arial, sans-serif;  
    background-color: #f0f0f0;  
    margin: 0;  
    padding: 0;  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    height: 100vh;  
}  
  
.container {  
    background-color: #fff;  
    padding: 20px;  
    border-radius: 5px;  
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);  
}
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
width: 300px;
text-align: center;
}
```

```
h1 {
  font-size: 24px;
  margin-bottom: 10px;
}
```

```
.color-box {
  width: 150px;
  height: 150px;
  margin: 20px auto;
  border-radius: 5px;
}
```

```
button {
  display: block;
  margin: 10px auto;
  padding: 8px 16px;
  background-color: #007bff;
  color: #fff;
  border: none;
```

```
border-radius: 3px;
cursor: pointer;
}
```

Step 3: JavaScript Logic

Create a JavaScript file named script.js for the application logic.

```
const flipButton =
document.getElementById("flipButton");
const colorBox = document.getElementById("colorBox");

const colors = ["#007bff", "#28a745", "#dc3545",
"#ffc107", "#17a2b8", "#343a40"];

flipButton.addEventListener("click", flipColor);

function flipColor() {
    const randomIndex = Math.floor(Math.random() *
colors.length);
    colorBox.style.backgroundColor =
colors[randomIndex];
}
```

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

Step 4: Testing

Open the index.html file in a web browser. You should see a Color Flipper with a colored box and a "Flip Color" button.

Congratulations! You've successfully created a simple Color Flipper using HTML, CSS, and JavaScript. This project demonstrates how you can use JavaScript to change the appearance of an element dynamically based on user interaction.

Here's the detailed breakdown of the JavaScript code:

We start by getting references to the HTML elements we'll be interacting with: flipButton and colorBox.

We attach an event listener to the "Flip Color" button (flipButton). This listener is set to call the flipColor function when the button is clicked.

The flipColor function is defined. This function:

- Generates a random index using Math.random() and Math.floor() within the range of the colors array length.
- Sets the background color of the colorBox element to the randomly selected color from the colors array.

By following these steps, the code creates a simple Color Flipper that changes the background color of a box each time the button is clicked. This project demonstrates how JavaScript can be used to modify the visual properties of elements dynamically.

Learn more about JavaScript with Examples and Source Code Laurence Svekis Courses <https://basescripts.com/>

Step 1: Get References to HTML Elements

```
const flipButton =  
document.getElementById("flipButton");  
const colorBox = document.getElementById("colorBox");
```

In this step, we're using the `document.getElementById()` method to retrieve references to the HTML elements we'll be interacting with:

- `flipButton`: The "Flip Color" button.
- `colorBox`: The colored box element where we'll change the background color.

Step 2: Attach Event Listener

```
flipButton.addEventListener("click", flipColor);
```

Here, we're attaching an event listener to the "Flip Color" button. The `flipColor` function will be called when the button is clicked.

Step 3: Define the `flipColor` Function

Learn more about JavaScript with Examples and Source Code Laurence Svekis
Courses <https://basescripts.com/>

```
function flipColor() {  
    const randomIndex = Math.floor(Math.random() *  
colors.length);  
    colorBox.style.backgroundColor =  
colors[randomIndex];  
}
```

Here's a detailed breakdown of the flipColor function:

- When the "Flip Color" button is clicked, the flipColor function is executed.
- We use Math.random() to generate a random decimal number between 0 (inclusive) and 1 (exclusive). We then multiply this number by the length of the colors array to get a random index within the array.
- The Math.floor() function is used to round down the decimal index to the nearest integer, ensuring it's a valid index within the array.
- We access the colors array using the random index to get a randomly selected color.
- We set the backgroundColor property of the colorBox element to the randomly selected color. This changes the background color of the box dynamically.

Step 4: Testing

When the user clicks the "Flip Color" button, the flipColor function is executed.

This function generates a random index, selects a color from the colors array, and changes the background color of the colorBox element to the selected color.

This project showcases how JavaScript can be used to modify the appearance of elements on a webpage based on user interactions. It's a simple yet effective demonstration of how JavaScript can create dynamic and visually engaging web experiences.