

Dark Mode toggle effect with check box.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dark Mode Toggle</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      padding: 20px;
    }
    /* Dark mode styles */
    body.dark-mode {
      background-color: #333;
      color: #fff;
    }
  </style>
</head>
<body>
  <h1>Dark Mode Toggle</h1>
  <label>
    <input type="checkbox" id="darkModeToggle"> Dark Mode
  </label>
  <script>
    // Retrieve user settings from localStorage
    const storedSettings = JSON.parse(localStorage.getItem('userSettings')) || {};
    // Apply dark mode setting if available
    if (storedSettings.darkMode) {
      document.body.classList.add('dark-mode');
      document.getElementById('darkModeToggle').checked = true;
    }
  </script>
</body>
</html>
```

```
// Toggle dark mode when the checkbox is clicked
document.getElementById('darkModeToggle').addEventListener('change',
function () {
    // Update the user settings
    const userSettings = {
        darkMode: this.checked,
    };
    // Apply dark mode styles
    if (userSettings.darkMode) {
        document.body.classList.add('dark-mode');
    } else {
        document.body.classList.remove('dark-mode');
    }
    // Store the updated settings in localStorage
    localStorage.setItem('userSettings', JSON.stringify(userSettings));
});
</script>
</body>
</html>
```

HTML Code:

1. Document Type Declaration (<!DOCTYPE html>):

- Defines the document type and version of HTML being used.

2. HTML Structure:

- `<html lang="en">`: Specifies that the document is written in English.
- `<head>`: Contains metadata about the HTML document, such as character set, viewport settings, and the title of the page.
- `<meta charset="UTF-8">`: Sets the character set to UTF-8.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: Configures the viewport for responsive design.
- `<title>Dark Mode Toggle</title>`: Sets the title of the HTML document.
- `<style>`: Contains CSS styles for the document.

3. CSS Styles:

- `body`: Applies styles to the body of the HTML document.
- `body.dark-mode`: Applies additional styles for dark mode when the body has the class "dark-mode."

Body Section:

4. Body Styles (body):

- `font-family: Arial, sans-serif;`: Sets the font family for the body text.
- `padding: 20px;`: Adds padding to the body.

5. Heading and Dark Mode Toggle:

- `<h1>Dark Mode Toggle</h1>`: Displays the heading of the page.
- `<label>`: Creates a label for the dark mode toggle.
- `<input type="checkbox" id="darkModeToggle">`: Creates a checkbox input element with the id "darkModeToggle."
- `Dark Mode`: Text next to the checkbox.

JavaScript Code:

6. Retrieve User Settings from Local Storage:

- `const storedSettings = JSON.parse(localStorage.getItem('userSettings')) || {};`: Retrieves user settings from local storage and parses it as JSON. If there are no settings, it initializes an empty object.

7. Apply Dark Mode Setting if Available:

- `if (storedSettings.darkMode) {...}`: Checks if the stored settings include a dark mode preference.
- `document.body.classList.add('dark-mode');`: Adds the 'dark-mode' class to the body, applying dark mode styles.
- `document.getElementById('darkModeToggle').checked = true;`: Sets the dark mode toggle checkbox to checked if dark mode is enabled.

8. Toggle Dark Mode on Checkbox Change:

- `document.getElementById('darkModeToggle').addEventListener('change', function () {...});`: Adds an event listener to the dark mode toggle checkbox that triggers a function when the checkbox state changes.

9. Update User Settings and Apply Dark Mode Styles:

- `const userSettings = {...};`: Defines an object containing the user's settings, including dark mode.
- `if (userSettings.darkMode) {...} else {...}`: Checks the current state of the dark mode setting and applies or removes the 'dark-mode' class accordingly.

10. Store Updated Settings in Local Storage:

- `localStorage.setItem('userSettings', JSON.stringify(userSettings));`: Stores the updated user settings in local storage as a JSON string.
-

Summary:

This code creates a simple webpage with a dark mode toggle. It uses JavaScript to handle the toggle state, store user preferences in local storage, and apply corresponding styles. The dark mode setting is persistent across page reloads, enhancing the user experience.