

JavaScript für Dark/Light Mode: Schritt-für-Schritt-Erklärung

Einführung

Dieses Dokument erklärt detailliert die JavaScript-Funktionalität, die für die Implementierung eines Dark/Light Mode-Umschalters benötigt wird. Jeder Codeabschnitt wird ausführlich erläutert, damit Schülerinnen und Schüler die Funktionsweise genau verstehen können.

1. Grundlegende JavaScript-Struktur

Hier ist der grundlegende JavaScript-Code für den Dark/Light Mode-Umschalter:

```
// DOM-Elemente auswählen
const toggleSwitch = document.querySelector('#checkbox');
const themeLabel = document.querySelector('#theme-label');
const currentTheme = localStorage.getItem('theme');

// Funktion zum Umschalten des Themes
function switchTheme(event) {
  if (event.target.checked) {
    // Auf Dark Mode umschalten
    document.documentElement.setAttribute('data-theme', 'dark');
    localStorage.setItem('theme', 'dark');
    themeLabel.textContent = 'Dark Mode';
  } else {
    // Auf Light Mode umschalten
    document.documentElement.setAttribute('data-theme', 'light');
    localStorage.setItem('theme', 'light');
  }
}
```

```

        themeLabel.textContent = 'Light Mode';
    }
}

// Event-Listener für den Toggle-Switch
toggleSwitch.addEventListener('change', switchTheme);

// Gespeichertes Theme beim Laden der Seite anwenden
if (currentTheme) {
    document.documentElement.setAttribute('data-theme', currentTheme);

    if (currentTheme === 'dark') {
        toggleSwitch.checked = true;
        themeLabel.textContent = 'Dark Mode';
    }
}
}

```

Erklärung der JavaScript-Elemente:

1. **DOM-Elemente auswählen:** `javascript const toggleSwitch = document.querySelector('#checkbox'); const themeLabel = document.querySelector('#theme-label'); const currentTheme = localStorage.getItem('theme');`
2. `document.querySelector('#checkbox')` - Wählt das HTML-Element mit der ID "checkbox" aus (unsere Toggle-Switch-Checkbox)
3. `document.querySelector('#theme-label')` - Wählt das HTML-Element mit der ID "theme-label" aus (der Text, der den aktuellen Modus anzeigt)
4. `localStorage.getItem('theme')` - Ruft den gespeicherten Theme-Wert aus dem localStorage des Browsers ab (falls vorhanden)
5. `const` - Deklariert eine Konstante, deren Wert nicht geändert werden kann
6. **Funktion zum Umschalten des Themes:** `javascript function switchTheme(event) { if (event.target.checked) { // Auf Dark Mode umschalten document.documentElement.setAttribute('data-theme', 'dark'); localStorage.setItem('theme', 'dark'); themeLabel.textContent`

```
= 'Dark Mode'; } else { // Auf Light Mode umschalten  
document.documentElement.setAttribute('data-theme', 'light');  
localStorage.setItem('theme', 'light'); themeLabel.textContent =  
'Light Mode'; } }
```

7. `function switchTheme(event) { ... }` - Definiert eine Funktion namens "switchTheme", die ein Event-Objekt als Parameter erhält
8. `if (event.target.checked) { ... } else { ... }` - Prüft, ob die Checkbox aktiviert ist (checked)
9. Wenn die Checkbox aktiviert ist (Dark Mode):
 - `document.documentElement.setAttribute('data-theme', 'dark')` - Setzt das Attribut "data-theme" auf "dark" für das HTML-Element (aktiviert die Dark Mode-CSS-Variablen)
 - `localStorage.setItem('theme', 'dark')` - Speichert die Benutzereinstellung "dark" im localStorage des Browsers
 - `themeLabel.textContent = 'Dark Mode'` - Ändert den Text des Labels zu "Dark Mode"
10. Wenn die Checkbox deaktiviert ist (Light Mode):
 - Ähnliche Aktionen, aber mit "light" statt "dark"
11. **Event-Listener für den Toggle-Switch:** `javascript toggleSwitch.addEventListener('change', switchTheme);`
12. `toggleSwitch.addEventListener('change', switchTheme)` - Fügt einen Event-Listener hinzu, der auf Änderungen der Checkbox achtet
13. Wenn sich der Zustand der Checkbox ändert (durch Klicken), wird die Funktion `switchTheme` aufgerufen
14. 'change' - Das Event, auf das reagiert wird (Änderung des Checkbox-Zustands)
15. `switchTheme` - Die Funktion, die aufgerufen wird, wenn das Event eintritt
16. **Gespeichertes Theme beim Laden der Seite anwenden:** ```javascript if (currentTheme) { document.documentElement.setAttribute('data-theme', currentTheme);`

```
if (currentTheme === 'dark') { toggleSwitch.checked = true; themeLabel.textContent = 'Dark Mode'; } }
```

17. `if (currentTheme) { ... }` - Prüft, ob ein Theme im localStorage gespeichert ist

18. Wenn ein Theme gespeichert ist:

- `document.documentElement.setAttribute('data-theme', currentTheme)`
- Wendet das gespeicherte Theme an
- `if (currentTheme === 'dark') { ... }` - Prüft, ob das gespeicherte Theme "dark" ist
- Wenn das gespeicherte Theme "dark" ist:
- `toggleSwitch.checked = true` - Aktiviert die Checkbox
- `themeLabel.textContent = 'Dark Mode'` - Ändert den Text des Labels zu "Dark Mode"

2. Erweiterung: Systemeinstellungen erkennen

Eine wichtige Erweiterung ist die Erkennung der Systemeinstellungen des Benutzers:

```
// Systemeinstellung für Farbschema erkennen
function detectColorScheme() {
    if (!localStorage.getItem('theme') && window.matchMedia('(prefers-color-scheme: dark)').matches) {
        document.documentElement.setAttribute('data-theme', 'dark');
        toggleSwitch.checked = true;
        themeLabel.textContent = 'Dark Mode';
    }
}

// Systemeinstellung beim Laden überprüfen
detectColorScheme();

// Event-Listener für Änderungen der Systemeinstellung
window.matchMedia('(prefers-color-scheme: dark)').addEventListener('change', e => {
    if (!localStorage.getItem('theme')) {
        if (e.matches) {
            document.documentElement.setAttribute('data-theme', 'dark');
        }
    }
});
```

```

        toggleSwitch.checked = true;
        themeLabel.textContent = 'Dark Mode';
    } else {
        document.documentElement.setAttribute('data-theme', 'light');
        toggleSwitch.checked = false;
        themeLabel.textContent = 'Light Mode';
    }
}
});

```

Erklärung der Systemeinstellungserkennung:

1. **Funktion zur Erkennung des Farbschemas:** javascript function
`detectColorScheme()` { if (!localStorage.getItem('theme') && window.matchMedia('(prefers-color-scheme: dark)').matches) { document.documentElement.setAttribute('data-theme', 'dark'); toggleSwitch.checked = true; themeLabel.textContent = 'Dark Mode'; } }
2. `function detectColorScheme() { ... }` - Definiert eine Funktion zur Erkennung des bevorzugten Farbschemas
3. `if (!localStorage.getItem('theme') && window.matchMedia('(prefers-color-scheme: dark)').matches) { ... }` - Prüft zwei Bedingungen:
 1. `!localStorage.getItem('theme')` - Es ist kein Theme im localStorage gespeichert (der Benutzer hat noch keine manuelle Auswahl getroffen)
 2. `window.matchMedia('(prefers-color-scheme: dark)').matches` - Die Systemeinstellung des Benutzers bevorzugt den Dark Mode
4. Wenn beide Bedingungen erfüllt sind:
 - `document.documentElement.setAttribute('data-theme', 'dark')` - Aktiviert den Dark Mode
 - `toggleSwitch.checked = true` - Aktiviert die Checkbox
 - `themeLabel.textContent = 'Dark Mode'` - Ändert den Text des Labels zu "Dark Mode"
5. **Systemeinstellung beim Laden überprüfen:** javascript `detectColorScheme();`

6. Ruft die Funktion `detectColorScheme` beim Laden der Seite auf, um die Systemeinstellung zu überprüfen
7. **Event-Listener für Änderungen der Systemeinstellung:** `javascript`

```
window.matchMedia('(prefers-color-scheme: dark)').addEventListener('change', e => { if (!localStorage.getItem('theme')) { if (e.matches) { document.documentElement.setAttribute('data-theme', 'dark'); toggleSwitch.checked = true; themeLabel.textContent = 'Dark Mode'; } else { document.documentElement.setAttribute('data-theme', 'light'); toggleSwitch.checked = false; themeLabel.textContent = 'Light Mode'; } } });
```
8. `window.matchMedia('(prefers-color-scheme: dark)').addEventListener('change', e => { ... })` - Fügt einen Event-Listener hinzu, der auf Änderungen der Systemeinstellung achtet
9. `if (!localStorage.getItem('theme')) { ... }` - Prüft, ob der Benutzer bereits eine manuelle Auswahl getroffen hat
10. Wenn keine manuelle Auswahl getroffen wurde:
 - `if (e.matches) { ... } else { ... }` - Prüft, ob die neue Systemeinstellung Dark Mode bevorzugt
 - Je nach Systemeinstellung wird der entsprechende Modus aktiviert

3. Animationseffekte hinzufügen

Um den Wechsel zwischen den Modi ansprechender zu gestalten, können Animationseffekte hinzugefügt werden:

```
// Funktion zum Umschalten des Themes mit Animation
function switchTheme(event) {
  if (event.target.checked) {
    // Auf Dark Mode umschalten
    document.documentElement.setAttribute('data-theme', 'dark');
    localStorage.setItem('theme', 'dark');
    themeLabel.textContent = 'Dark Mode';
  }
}
```

```

        // Animationseffekt hinzufügen
        document.body.classList.add('theme-transition');
        setTimeout(() => {
            document.body.classList.remove('theme-transition');
        }, 1000);
    } else {
        // Auf Light Mode umschalten
        document.documentElement.setAttribute('data-theme', 'light');
        localStorage.setItem('theme', 'light');
        themeLabel.textContent = 'Light Mode';

        // Animationseffekt hinzufügen
        document.body.classList.add('theme-transition');
        setTimeout(() => {
            document.body.classList.remove('theme-transition');
        }, 1000);
    }
}

```

Erklärung der Animationseffekte:

- 1. Animationsklasse hinzufügen:** `javascript`
`document.body.classList.add('theme-transition');`
- `document.body.classList.add('theme-transition')` - Fügt dem Body-Element die Klasse "theme-transition" hinzu
- Diese Klasse kann in CSS verwendet werden, um spezielle Animationseffekte während des Themenwechsels zu definieren
- 4. Timeout für die Entfernung der Animationsklasse:** `javascript` `setTimeout(() => { document.body.classList.remove('theme-transition'); }, 1000);`
- `setTimeout(() => { ... }, 1000)` - Führt den Code innerhalb der geschweiften Klammern nach 1000 Millisekunden (1 Sekunde) aus
- `document.body.classList.remove('theme-transition')` - Entfernt die Klasse "theme-transition" vom Body-Element

7. Dies stellt sicher, dass die Animation nach Abschluss wieder entfernt wird
8. **Zugehöriges CSS für die Animation:** `css .theme-transition { transition: background-color 1s ease, color 1s ease; }`
9. Definiert sanfte Übergänge für Hintergrund- und Textfarbe während der Themenwechsel-Animation

4. Vollständiger JavaScript-Code

Hier ist der vollständige JavaScript-Code mit allen besprochenen Funktionen:

```
// DOM-Elemente auswählen
const toggleSwitch = document.querySelector('#checkbox');
const themeLabel = document.querySelector('#theme-label');
const currentTheme = localStorage.getItem('theme');

// Funktion zum Umschalten des Themes mit Animation
function switchTheme(event) {
  if (event.target.checked) {
    // Auf Dark Mode umschalten
    document.documentElement.setAttribute('data-theme', 'dark');
    localStorage.setItem('theme', 'dark');
    themeLabel.textContent = 'Dark Mode';

    // Animationseffekt hinzufügen
    document.body.classList.add('theme-transition');
    setTimeout(() => {
      document.body.classList.remove('theme-transition');
    }, 1000);
  } else {
    // Auf Light Mode umschalten
    document.documentElement.setAttribute('data-theme', 'light');
    localStorage.setItem('theme', 'light');
    themeLabel.textContent = 'Light Mode';

    // Animationseffekt hinzufügen
```



```

        document.body.classList.add('theme-transition');
        setTimeout(() => {
            document.body.classList.remove('theme-transition');
        }, 1000);
    }
}

// Event-Listener für den Toggle-Switch
toggleSwitch.addEventListener('change', switchTheme);

// Gespeichertes Theme beim Laden der Seite anwenden
if (currentTheme) {
    document.documentElement.setAttribute('data-theme', currentTheme);

    if (currentTheme === 'dark') {
        toggleSwitch.checked = true;
        themeLabel.textContent = 'Dark Mode';
    }
}

// Systemeinstellung für Farbschema erkennen
function detectColorScheme() {
    if (!localStorage.getItem('theme') && window.matchMedia('(prefers-color-scheme: dark)').matches) {
        document.documentElement.setAttribute('data-theme', 'dark');
        toggleSwitch.checked = true;
        themeLabel.textContent = 'Dark Mode';
    }
}

// Systemeinstellung beim Laden überprüfen
detectColorScheme();

// Event-Listener für Änderungen der Systemeinstellung
window.matchMedia('(prefers-color-scheme: dark)').addEventListener('change', e => {
    if (!localStorage.getItem('theme')) {
        if (e.matches) {
            document.documentElement.setAttribute('data-theme', 'dark');

```

```
        toggleSwitch.checked = true;
        themeLabel.textContent = 'Dark Mode';
    } else {
        document.documentElement.setAttribute('data-theme', 'light');
        toggleSwitch.checked = false;
        themeLabel.textContent = 'Light Mode';
    }
}
});
```

5. Schritt-für-Schritt-Anleitung zur Implementierung

Hier ist eine Schritt-für-Schritt-Anleitung zur Implementierung des JavaScript-Codes:

Schritt 1: Erstelle eine neue JavaScript-Datei

Erstelle eine Datei namens `script.js` im Verzeichnis `js` deines Projekts.

Schritt 2: Wähle die DOM-Elemente aus

Füge den folgenden Code am Anfang der Datei ein, um die benötigten DOM-Elemente auszuwählen:

```
// DOM-Elemente auswählen
const toggleSwitch = document.querySelector('#checkbox');
const themeLabel = document.querySelector('#theme-label');
const currentTheme = localStorage.getItem('theme');
```

Schritt 3: Definiere die Funktion zum Umschalten des Themes

Füge die Funktion `switchTheme` hinzu, die den Wechsel zwischen Light und Dark Mode steuert:

```
// Funktion zum Umschalten des Themes
function switchTheme(event) {
  if (event.target.checked) {
    // Auf Dark Mode umschalten
    document.documentElement.setAttribute('data-theme', 'dark');
    localStorage.setItem('theme', 'dark');
    themeLabel.textContent = 'Dark Mode';
  } else {
    // Auf Light Mode umschalten
    document.documentElement.setAttribute('data-theme', 'light');
    localStorage.setItem('theme', 'light');
    themeLabel.textContent = 'Light Mode';
  }
}
```

Schritt 4: Füge den Event-Listener hinzu

Füge einen Event-Listener hinzu, der auf Änderungen der Checkbox reagiert:

```
// Event-Listener für den Toggle-Switch
toggleSwitch.addEventListener('change', switchTheme);
```

Schritt 5: Wende das gespeicherte Theme beim Laden an

Füge Code hinzu, der das gespeicherte Theme beim Laden der Seite anwendet:

```
// Gespeichertes Theme beim Laden der Seite anwenden
if (currentTheme) {
  document.documentElement.setAttribute('data-theme', currentTheme);

  if (currentTheme === 'dark') {
    toggleSwitch.checked = true;
    themeLabel.textContent = 'Dark Mode';
  }
}
```

```
}  
}
```

Schritt 6: Füge die Systemeinstellungserkennung hinzu

Füge Code hinzu, der die Systemeinstellungen des Benutzers erkennt:

```
// Systemeinstellung für Farbschema erkennen  
function detectColorScheme() {  
    if (!localStorage.getItem('theme') && window.matchMedia('(prefers-color-scheme: dark)').matches) {  
        document.documentElement.setAttribute('data-theme', 'dark');  
        toggleSwitch.checked = true;  
        themeLabel.textContent = 'Dark Mode';  
    }  
}  
  
// Systemeinstellung beim Laden überprüfen  
detectColorScheme();  
  
// Event-Listener für Änderungen der Systemeinstellung  
window.matchMedia('(prefers-color-scheme: dark)').addEventListener('change', e => {  
    if (!localStorage.getItem('theme')) {  
        if (e.matches) {  
            document.documentElement.setAttribute('data-theme', 'dark');  
            toggleSwitch.checked = true;  
            themeLabel.textContent = 'Dark Mode';  
        } else {  
            document.documentElement.setAttribute('data-theme', 'light');  
            toggleSwitch.checked = false;  
            themeLabel.textContent = 'Light Mode';  
        }  
    }  
});
```

Schritt 7: Binde die JavaScript-Datei in deine HTML-Datei ein

Füge am Ende deiner HTML-Datei, kurz vor dem schließenden `</body>`-Tag, den folgenden Code ein:

```
<script src="js/script.js"></script>
```

6. Häufige Fehler und Lösungen

Hier sind einige häufige Fehler und ihre Lösungen:

- 1. Die Checkbox reagiert nicht auf Klicks:**
 2. Überprüfe, ob die ID der Checkbox mit der ID im JavaScript-Code übereinstimmt
 3. Stelle sicher, dass das Label korrekt mit der Checkbox verknüpft ist
- 4. Das Theme wird nicht gespeichert:**
 5. Überprüfe, ob der localStorage-Code korrekt implementiert ist
 6. Teste, ob dein Browser localStorage unterstützt (die meisten modernen Browser tun dies)
- 7. Die Systemeinstellungserkennung funktioniert nicht:**
 8. Nicht alle Browser unterstützen `prefers-color-scheme`
 9. Teste die Funktionalität in verschiedenen Browsern
- 10. Die Animation ist zu schnell oder zu langsam:**
 11. Passe die Zeitangabe im `setTimeout`-Aufruf an (derzeit 1000 Millisekunden)
 12. Passe die Übergangszeit in der CSS-Klasse `.theme-transition` an

Zusammenfassung

In diesem Dokument haben wir die JavaScript-Funktionalität für einen Dark/Light Mode-Umschalter detailliert erklärt:

1. Auswahl der DOM-Elemente für die Interaktion mit dem Toggle-Switch
2. Implementierung der Funktion zum Umschalten zwischen Light und Dark Mode
3. Speicherung der Benutzereinstellung im localStorage des Browsers
4. Erkennung und Berücksichtigung der Systemeinstellungen des Benutzers
5. Hinzufügen von Animationseffekten für einen ansprechenden Themenwechsel
6. Schritt-für-Schritt-Anleitung zur Implementierung des Codes
7. Häufige Fehler und ihre Lösungen

Mit diesem Wissen kannst du einen vollständig funktionierenden Dark/Light Mode-Umschalter implementieren, der die Benutzereinstellungen speichert und die Systemeinstellungen berücksichtigt.