

# Praktische Übungen: Dark/Light Mode implementieren

---

## Einführung

---

Dieses Dokument enthält praktische Übungen, mit denen Schülerinnen und Schüler die Implementierung eines Dark/Light Mode-Umschalters üben können. Die Übungen bauen aufeinander auf und führen schrittweise durch den gesamten Prozess.

## Übung 1: Grundlegende HTML-Struktur erstellen

---

**Aufgabe:** Erstelle die HTML-Struktur für eine einfache Webseite mit einem Dark/Light Mode-Umschalter.

**Anleitung:**

1. Erstelle eine neue HTML-Datei namens `uebung1.html`
2. Füge die grundlegende HTML-Struktur hinzu (DOCTYPE, html, head, body)
3. Füge im head-Bereich einen Titel und einen Link zu einer CSS-Datei hinzu
4. Erstelle im body-Bereich:
5. Eine Überschrift
6. Einen Absatz mit Text
7. Einen Container für den Toggle-Switch
8. Ein Label mit einer Checkbox und einem Schieberegler

**Beispiellösung:**

```
<!DOCTYPE html>
<html lang="de">
```

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dark/Light Mode Übung 1</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div class="container">
    <h1>Dark/Light Mode Übung</h1>

    <div class="theme-switch-wrapper">
      <label class="theme-switch" for="checkbox">
        <input type="checkbox" id="checkbox" />
        <div class="slider round"></div>
      </label>
      <span id="theme-label">Light Mode</span>
    </div>

    <p>Dies ist ein Beispieltext, um den Dark/Light Mode zu demonstrieren. De
  </div>

  <script src="script.js"></script>
</body>
</html>

```

## Übung 2: CSS-Styling für den Toggle-Switch

**Aufgabe:** Erstelle das CSS-Styling für den Toggle-Switch und definiere die Farbschemata für Light und Dark Mode.

### Anleitung:

1. Erstelle eine neue CSS-Datei namens `style.css`
2. Definiere CSS-Variablen für beide Farbschemata (Light und Dark Mode)
3. Style den Toggle-Switch, sodass er wie ein moderner Schieberegler aussieht
4. Wende die Farbschemata auf die grundlegenden Elemente der Webseite an

## Beispiellösung:

```
:root {
  /* Light Mode (Standard) */
  --background-color: #f8f9fa;
  --text-color: #333333;
  --primary-color: #0066cc;
  --secondary-color: #e9ecef;
  --accent-color: #fd7e14;
  --card-bg: #ffffff;
  --border-color: #dee2e6;

  /* Übergangseffekte */
  --transition-speed: 0.3s;
}

/* Dark Mode */
[data-theme="dark"] {
  --background-color: #222222;
  --text-color: #f8f9fa;
  --primary-color: #4da6ff;
  --secondary-color: #444444;
  --accent-color: #ff9642;
  --card-bg: #333333;
  --border-color: #555555;
}

* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
  transition: background-color var(--transition-speed),
              color var(--transition-speed),
              border-color var(--transition-speed);
}

body {
```

```

    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    background-color: var(--background-color);
    color: var(--text-color);
    line-height: 1.6;
    padding: 20px;
}

.container {
    max-width: 800px;
    margin: 0 auto;
    padding: 20px;
    background-color: var(--card-bg);
    border-radius: 8px;
    box-shadow: 0 2px 10px rgba(0, 0, 0, 0.1);
}

h1 {
    color: var(--primary-color);
    margin-bottom: 20px;
    border-bottom: 2px solid var(--border-color);
    padding-bottom: 10px;
}

p {
    margin-bottom: 20px;
}

/* Toggle Switch Styling */
.theme-switch-wrapper {
    display: flex;
    align-items: center;
    margin: 20px 0;
}

.theme-switch {
    display: inline-block;
    height: 34px;

```

```
    position: relative;
    width: 60px;
}

.theme-switch input {
    display: none;
}

.slider {
    background-color: #ccc;
    bottom: 0;
    cursor: pointer;
    left: 0;
    position: absolute;
    right: 0;
    top: 0;
    transition: .4s;
}

.slider:before {
    background-color: white;
    bottom: 4px;
    content: "";
    height: 26px;
    left: 4px;
    position: absolute;
    transition: .4s;
    width: 26px;
}

input:checked + .slider {
    background-color: var(--accent-color);
}

input:checked + .slider:before {
    transform: translateX(26px);
}
```

```
.slider.round {
  border-radius: 34px;
}

.slider.round:before {
  border-radius: 50%;
}

#theme-label {
  margin-left: 10px;
  font-weight: 500;
}
```

## Übung 3: JavaScript-Funktionalität implementieren

---

**Aufgabe:** Implementiere die JavaScript-Funktionalität, um zwischen Light und Dark Mode zu wechseln und die Benutzereinstellung zu speichern.

**Anleitung:**

1. Erstelle eine neue JavaScript-Datei namens `script.js`
2. Wähle die DOM-Elemente aus (Checkbox und Label)
3. Implementiere eine Funktion zum Umschalten des Themes
4. Füge einen Event-Listener für die Checkbox hinzu
5. Implementiere die Speicherung der Benutzereinstellung im localStorage
6. Stelle sicher, dass die gespeicherte Einstellung beim Laden der Seite angewendet wird

**Beispiellösung:**

```
// 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';
    }
}
}

```

## Übung 4: Systemeinstellungen erkennen

---

**Aufgabe:** Erweitere die JavaScript-Funktionalität, um die Systemeinstellungen des Benutzers zu erkennen und zu berücksichtigen.

**Anleitung:**

1. Öffne die Datei `script.js`

2. Implementiere eine Funktion zur Erkennung der Systemeinstellungen
3. Füge einen Event-Listener hinzu, der auf Änderungen der Systemeinstellungen reagiert
4. Stelle sicher, dass die Systemeinstellungen nur berücksichtigt werden, wenn der Benutzer noch keine manuelle Auswahl getroffen hat

**Beispiellösung:**

```
// Systemeinstellung für Farbschema erkennen
function detectColorScheme() {
    if (!localStorage.getItem('theme') && window.matchMedia('(prefers-color-scheme: dark)').media === 'true') {
        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';
        }
    }
});
```



## Übung 5: Erweiterte Inhalte und Styling

---

**Aufgabe:** Erweitere die Webseite um zusätzliche Inhalte und verbessere das Styling für beide Modi.

**Anleitung:**

1. Öffne die Datei `uebung1.html` und benenne sie in `index.html` um
2. Füge weitere Inhalte hinzu:
3. Eine Infobox
4. Eine Tabelle
5. Ein Formular
6. Ein Codebeispiel
7. Öffne die Datei `style.css`
8. Erweitere das CSS-Styling für die neuen Elemente
9. Stelle sicher, dass alle Elemente in beiden Modi gut aussehen und lesbar sind

**Beispiellösung für HTML-Erweiterungen:**

```
<div class="info-box">
  <h3>Wusstest du?</h3>
  <p>Dark Mode kann die Akkulaufzeit bei OLED-Displays verlängern, da dunkle Pi
</div>

<h2>Vergleich: Light vs. Dark Mode</h2>
<table>
  <thead>
    <tr>
      <th>Eigenschaft</th>
      <th>Light Mode</th>
      <th>Dark Mode</th>
    </tr>
  </thead>
  <tbody>
    <tr>
```

```

        <td>Lesbarkeit bei Tageslicht</td>
        <td>Sehr gut</td>
        <td>Mäßig</td>
    </tr>
    <tr>
        <td>Lesbarkeit bei Nacht</td>
        <td>Schlecht</td>
        <td>Sehr gut</td>
    </tr>
    <tr>
        <td>Augenmüdigkeit</td>
        <td>Höher bei langer Nutzung</td>
        <td>Geringer bei langer Nutzung</td>
    </tr>
    <tr>
        <td>Akkulaufzeit (OLED)</td>
        <td>Normal</td>
        <td>Verbessert</td>
    </tr>
</tbody>
</table>

```

## <h2>Kontaktformular</h2>

```

<form>
    <div class="form-group">
        <label for="name">Name:</label>
        <input type="text" id="name" class="form-control">
    </div>
    <div class="form-group">
        <label for="email">E-Mail:</label>
        <input type="email" id="email" class="form-control">
    </div>
    <div class="form-group">
        <label for="message">Nachricht:</label>
        <textarea id="message" class="form-control" rows="4"></textarea>
    </div>
    <button type="submit" class="btn">Absenden</button>

```

```

</form>

<h2>Code-Beispiel</h2>
<pre><code>
// Einfache Funktion zum Umschalten des Themes
function toggleTheme() {
    const body = document.body;
    body.classList.toggle('dark-theme');
}
</code></pre>

```

### Beispiellösung für CSS-Erweiterungen:

```

/* Info Box */
.info-box {
    background-color: var(--secondary-color);
    border-left: 4px solid var(--primary-color);
    padding: 15px;
    margin: 20px 0;
    border-radius: 0 4px 4px 0;
}

.info-box h3 {
    color: var(--primary-color);
    margin-bottom: 10px;
}

/* Tabelle */
table {
    width: 100%;
    border-collapse: collapse;
    margin: 20px 0;
}

th, td {
    padding: 12px 15px;
    text-align: left;
}

```

```

        border-bottom: 1px solid var(--border-color);
    }

    th {
        background-color: var(--secondary-color);
        color: var(--primary-color);
    }

    tr:hover {
        background-color: var(--secondary-color);
    }

    /* Formular */
    .form-group {
        margin-bottom: 15px;
    }

    .form-control {
        width: 100%;
        padding: 10px;
        border: 1px solid var(--border-color);
        border-radius: 4px;
        background-color: var(--input-bg, var(--card-bg));
        color: var(--text-color);
    }

    .btn {
        padding: 10px 15px;
        background-color: var(--primary-color);
        color: white;
        border: none;
        border-radius: 4px;
        cursor: pointer;
        transition: background-color 0.2s;
    }

    .btn:hover {

```

```

        background-color: var(--btn-hover, var(--primary-color));
        opacity: 0.9;
    }

    /* Code-Block */
    pre {
        background-color: var(--code-bg, #f5f5f5);
        color: var(--code-text, #333);
        padding: 15px;
        border-radius: 4px;
        overflow-x: auto;
        margin: 20px 0;
    }

    [data-theme="dark"] pre {
        --code-bg: #2d2d2d;
        --code-text: #f8f9fa;
    }

    code {
        font-family: 'Courier New', Courier, monospace;
    }

```

## Übung 6: Animationseffekte hinzufügen

---

**Aufgabe:** Füge Animationseffekte hinzu, um den Wechsel zwischen den Modi ansprechender zu gestalten.

### Anleitung:

1. Öffne die Datei `style.css`
2. Füge CSS-Klassen für Animationen hinzu
3. Öffne die Datei `script.js`
4. Erweitere die `switchTheme`-Funktion, um Animationsklassen hinzuzufügen und zu entfernen

### Beispiellösung für CSS-Animationen:

```
.theme-transition {  
    transition: background-color 1s ease, color 1s ease;  
}  
  
.fade-in {  
    animation: fadeIn 0.5s ease-in;  
}  
  
@keyframes fadeIn {  
    from { opacity: 0; }  
    to { opacity: 1; }  
}
```

### Beispiellösung für JavaScript-Animationen:

```
// 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');  
        document.querySelectorAll('.container > *').forEach(element => {  
            element.classList.add('fade-in');  
            setTimeout(() => {  
                element.classList.remove('fade-in');  
            }, 500);  
        });  
  
        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');
        document.querySelectorAll('.container > *').forEach(element => {
            element.classList.add('fade-in');
            setTimeout(() => {
                element.classList.remove('fade-in');
            }, 500);
        });

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

```

## Übung 7: Barrierefreiheit verbessern

**Aufgabe:** Verbessere die Barrierefreiheit der Webseite, insbesondere für den Dark/Light Mode-Umschalter.

### Anleitung:

1. Öffne die Datei `index.html`
2. Füge ARIA-Attribute zum Toggle-Switch hinzu
3. Stelle sicher, dass der Kontrast in beiden Modi ausreichend ist
4. Füge eine Tastatursteuerung für den Toggle-Switch hinzu

### Beispiellösung für HTML-Barrierefreiheit:

```

<div class="theme-switch-wrapper">
  <label class="theme-switch" for="checkbox">
    <input type="checkbox" id="checkbox" aria-label="Umschalten zwischen Ligh
    <div class="slider round" role="switch" aria-checked="false" tabindex="0"
  </label>
  <span id="theme-label">Light Mode</span>
</div>

```

### Beispiellösung für JavaScript-Barrierefreiheit:

```

// Tastatursteuerung für den Toggle-Switch
const slider = document.querySelector('.slider');
slider.addEventListener('keydown', (e) => {
  if (e.key === 'Enter' || e.key === ' ') {
    e.preventDefault();
    toggleSwitch.checked = !toggleSwitch.checked;
    slider.setAttribute('aria-checked', toggleSwitch.checked);
    switchTheme({ target: toggleSwitch });
  }
});

// ARIA-Attribute aktualisieren
function updateARIA() {
  slider.setAttribute('aria-checked', toggleSwitch.checked);
}

// Event-Listener für den Toggle-Switch erweitern
toggleSwitch.addEventListener('change', (e) => {
  switchTheme(e);
  updateARIA();
});

// ARIA-Attribute beim Laden aktualisieren
if (currentTheme === 'dark') {

```



```
slider.setAttribute('aria-checked', 'true');  
}
```

## Übung 8: Responsives Design implementieren

---

**Aufgabe:** Mache die Webseite responsiv, sodass sie auf verschiedenen Geräten gut aussieht.

**Anleitung:**

1. Öffne die Datei `style.css`
2. Füge Media Queries für verschiedene Bildschirmgrößen hinzu
3. Passe das Layout und die Schriftgrößen entsprechend an

**Beispiellösung für responsives CSS:**

```
/* Grundlegendes responsives Design */  
@media (max-width: 768px) {  
  .container {  
    padding: 15px;  
  }  
  
  h1 {  
    font-size: 1.8rem;  
  }  
  
  h2 {  
    font-size: 1.5rem;  
  }  
  
  table {  
    font-size: 0.9rem;  
  }  
  
  th, td {  
    padding: 8px 10px;  
  }  
}
```

```
}  
}  
  
@media (max-width: 480px) {  
  body {  
    padding: 10px;  
  }  
  
  .container {  
    padding: 10px;  
  }  
  
  h1 {  
    font-size: 1.5rem;  
  }  
  
  h2 {  
    font-size: 1.3rem;  
  }  
  
  .theme-switch {  
    height: 28px;  
    width: 50px;  
  }  
  
  .slider:before {  
    height: 20px;  
    width: 20px;  
  }  
  
  input:checked + .slider:before {  
    transform: translateX(22px);  
  }  
  
  table {  
    font-size: 0.8rem;  
  }  
}
```

```
th, td {  
    padding: 6px 8px;  
}  
}
```

## Abschlussprojekt: Vollständige Dark/Light Mode-Webseite

---

**Aufgabe:** Erstelle eine vollständige Webseite mit Dark/Light Mode-Unterstützung zu einem Thema deiner Wahl.

### Anforderungen:

1. Die Webseite sollte mindestens folgende Elemente enthalten:
2. Eine Navigationsleiste
3. Ein Hero-Bereich
4. Mehrere Inhaltsabschnitte
5. Ein Kontaktformular
6. Ein Footer
7. Der Dark/Light Mode-Umschalter sollte in der Navigationsleiste platziert sein
8. Alle Elemente sollten in beiden Modi gut aussehen und lesbar sein
9. Die Webseite sollte responsiv sein
10. Die Benutzereinstellung sollte gespeichert werden
11. Die Systemeinstellungen sollten berücksichtigt werden
12. Übergänge zwischen den Modi sollten animiert sein
13. Die Webseite sollte barrierefrei sein

### Tipps:

1. Beginne mit der Planung der Struktur und des Designs
2. Implementiere zuerst die grundlegende HTML-Struktur

3. Füge dann das CSS-Styling hinzu, einschließlich der Variablen für beide Modi
4. Implementiere die JavaScript-Funktionalität für den Modiwechsel
5. Teste die Webseite in verschiedenen Browsern und auf verschiedenen Geräten
6. Überprüfe die Barrierefreiheit mit entsprechenden Tools

## Zusammenfassung

---

In diesen Übungen hast du gelernt, wie man einen Dark/Light Mode-Umschalter implementiert und eine Webseite erstellt, die beide Modi unterstützt. Du hast folgende Fähigkeiten erworben:

1. Erstellung der HTML-Struktur für einen Toggle-Switch
2. Definition von CSS-Variablen für verschiedene Farbschemata
3. Implementierung der JavaScript-Funktionalität für den Modiwechsel
4. Speicherung der Benutzereinstellung im localStorage
5. Erkennung und Berücksichtigung der Systemeinstellungen
6. Hinzufügen von Animationseffekten für den Modiwechsel
7. Verbesserung der Barrierefreiheit
8. Implementierung eines responsiven Designs

Diese Fähigkeiten kannst du nun in deinen eigenen Webprojekten einsetzen, um benutzerfreundliche und moderne Webseiten zu erstellen, die sowohl Light als auch Dark Mode unterstützen.