

Arbeitsblatt: Dark und White Modus

Einführung

Liebe Schülerin, lieber Schüler,

In diesem Arbeitsblatt wirst du dich mit dem Thema "Dark und White Modus" in der Webentwicklung beschäftigen. Du lernst, wie man mit HTML, CSS und JavaScript zwischen einem hellen und einem dunklen Farbschema wechseln kann und warum diese Funktion für moderne Websites wichtig ist.

Lernziele

Nach Abschluss dieses Arbeitsblatts solltest du: - Die Vorteile von Dark/Light Mode verstehen - CSS-Variablen für verschiedene Farbschemata einsetzen können - Einen Toggle-Switch mit JavaScript programmieren können - Barrierefreiheitsaspekte bei der Farbwahl berücksichtigen können

Aufgabe 1: Grundlagen verstehen

Beantworte folgende Fragen:

1. Welche Vorteile bietet ein Dark Mode für Benutzer?
2. Warum ist es wichtig, Benutzern die Wahl zwischen Light und Dark Mode zu geben?
3. Nenne drei Beispiele für Websites oder Apps, die einen Dark Mode anbieten.
4. Welche Rolle spielt Dark Mode bei der Barrierefreiheit?

Aufgabe 2: CSS-Variablen für Farbschemata

Vervollständige den folgenden CSS-Code, um Farbvariablen für Light und Dark Mode zu definieren:

```
:root {  
    /* Light Mode (Standard) */  
    --background-color: _____;  
    --text-color: _____;  
    --primary-color: _____;  
    --secondary-color: _____;  
}  
  
[data-theme="dark"] {  
    /* Dark Mode */  
    --background-color: _____;  
    --text-color: _____;  
    --primary-color: _____;  
    --secondary-color: _____;  
}
```

Erkläre, warum CSS-Variablen besonders nützlich für die Implementierung von Dark/Light Mode sind.

Aufgabe 3: Toggle-Switch erstellen

Erstelle einen Toggle-Switch für den Wechsel zwischen Light und Dark Mode. Nutze dafür die folgende HTML-Struktur und ergänze den CSS-Code:

```
<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>
```

```
.theme-switch-wrapper {
  /* Dein Code hier */
}

.theme-switch {
  /* Dein Code hier */
}

.slider {
  /* Dein Code hier */
}

.slider:before {
  /* Dein Code hier */
}

input:checked + .slider {
  /* Dein Code hier */
}

input:checked + .slider:before {
  /* Dein Code hier */
}
```

Aufgabe 4: JavaScript-Funktionalität

Schreibe eine JavaScript-Funktion, die: 1. Den Toggle-Switch überwacht 2. Zwischen Light und Dark Mode wechselt 3. Die Benutzereinstellung im localStorage speichert

```
// DOM-Elemente auswählen
const toggleSwitch = document.querySelector('#checkbox');
```

```
const themeLabel = document.querySelector('#theme-label');

// Funktion zum Umschalten des Themes
function switchTheme(event) {
    // Dein Code hier
}

// Event-Listener für den Toggle-Switch
// Dein Code hier

// Gespeichertes Theme beim Laden der Seite anwenden
// Dein Code hier
```

Aufgabe 5: Barrierefreiheit prüfen

Öffne den Kontrastrechner (<exercises/kontrastrechner.html>) und teste verschiedene Farbkombinationen:

1. Wähle eine Textfarbe und eine Hintergrundfarbe für den Light Mode
2. Wähle eine Textfarbe und eine Hintergrundfarbe für den Dark Mode
3. Prüfe, ob beide Kombinationen die WCAG-Richtlinien erfüllen
4. Notiere die Kontrastverhältnisse und ob sie die AA- und AAA-Standards erfüllen

Modus	Textfarbe	Hintergrundfarbe	Kontrastverhältnis	AA-Standard	AAA-Standard
Light					
Dark					

Aufgabe 6: Systemeinstellungen erkennen

Analysiere den folgenden Code zur Erkennung der Systemeinstellungen:

```
const prefersDarkScheme = window.matchMedia('(prefers-color-scheme: dark)');

if (prefersDarkScheme.matches) {
    // Benutzer bevorzugt Dark Mode
    document.documentElement.setAttribute('data-theme', 'dark');
} else {
    // Benutzer bevorzugt Light Mode oder hat keine Präferenz
    document.documentElement.setAttribute('data-theme', 'light');
}

// Event-Listener für Änderungen der Systemeinstellung
prefersDarkScheme.addEventListener('change', (e) => {
    if (e.matches) {
        // Wechsel zu Dark Mode
        document.documentElement.setAttribute('data-theme', 'dark');
    } else {
        // Wechsel zu Light Mode
        document.documentElement.setAttribute('data-theme', 'light');
    }
});
```

Beantworte folgende Fragen: 1. Was macht die Funktion `window.matchMedia()` ? 2. Wie wird erkannt, ob der Benutzer Dark Mode bevorzugt? 3. Warum ist es wichtig, einen Event-Listener für Änderungen der Systemeinstellung zu haben? 4. Wie würdest du diesen Code erweitern, um die Benutzereinstellung im `localStorage` zu speichern?

Aufgabe 7: Eigenes Projekt

Erstelle eine einfache Webseite mit folgenden Anforderungen:

1. Die Seite soll einen funktionierenden Dark/Light Mode-Toggle haben
2. CSS-Variablen für Farbschemata verwenden
3. Die Benutzereinstellung im `localStorage` speichern
4. Systemeinstellungen berücksichtigen

5. Barrierefreiheitsrichtlinien einhalten (ausreichender Kontrast)

Die Webseite kann ein beliebiges Thema haben (z.B. persönliche Vorstellung, Hobby, Lieblingsfilm, etc.).

Hilfreiche Ressourcen

- [MDN Web Docs: CSS-Variablen](#)
- [MDN Web Docs: localStorage](#)
- [MDN Web Docs: prefers-color-scheme](#)
- [WCAG Kontrastrichtlinien](#)

Bewertungskriterien

Dein Projekt wird nach folgenden Kriterien bewertet:

- Funktionalität des Dark/Light Mode-Toggles
- Korrekte Verwendung von CSS-Variablen
- Speicherung der Benutzereinstellung
- Berücksichtigung der Systemeinstellungen
- Einhaltung der Barrierefreiheitsrichtlinien
- Code-Qualität und Dokumentation

Viel Erfolg bei der Bearbeitung der Aufgaben!