

Exercice 8 - Toggle de thème (Dark Mode)

Objectif

Changer dynamiquement l'apparence via des classes CSS basées sur le state.

Concepts abordés

- Classes CSS dynamiques
 - Template literals avec className
 - Toggle de booléen
 - Persistance (bonus)
-

Énoncé

Créer un composant `ThemeToggle` qui :

1. Affiche un bouton pour basculer entre thème clair et sombre
2. Change l'apparence de la page selon le thème
3. Affiche l'icône appropriée (soleil/lune)

Bonus

- Persister le choix dans localStorage
 - Appliquer le thème au body
-

Correction

```
// src/exercices/ex08-toggle-theme/ThemeToggle.jsx

import { useState, useEffect } from 'react';
import './ThemeToggle.css';

function ThemeToggle() {
  // =====
  // STATE
  // On initialise avec la valeur du localStorage si elle existe
  // =====
  const [isDark, setIsDark] = useState(() => {
    // Fonction d'initialisation (exécutée une seule fois)
    const saved = localStorage.getItem('theme');
    return saved === 'dark';
  });

  // =====
  // EFFET : Synchroniser avec localStorage et le body
  // =====
}
```

```

useEffect(() => {
  // Sauvegarder le choix
  localStorage.setItem('theme', isDark ? 'dark' : 'light');

  // Appliquer au body (optionnel, pour le style global)
  document.body.classList.toggle('dark-theme', isDark);
}, [isDark]);

// =====
// HANDLER
// =====
function toggleTheme() {
  setIsDark(prev => !prev);
}

// =====
// CLASSES DYNAMIQUES
// Plusieurs façons de construire la className
// =====

// Méthode 1 : Template literal
const containerClass = `theme-container ${isDark ? 'dark' : 'light'}`;

// Méthode 2 : Concaténation
// const containerClass = 'theme-container ' + (isDark ? 'dark' : 'light');

// Méthode 3 : Array.join (utile pour plusieurs classes conditionnelles)
// const containerClass = [
//   'theme-container',
//   isDark ? 'dark' : 'light',
//   isCompact && 'compact' // Ajouté seulement si true
// ].filter(Boolean).join(' ');

return (
  <div className={containerClass}>
    <div className="theme-content">
      <h2>Bienvenue !</h2>
      <p>
        Le thème actuel est : <strong>{isDark ? 'Sombre' : 'Clair'}</strong>
      </p>

      <button
        onClick={toggleTheme}
        className="theme-toggle-btn"
        aria-label={isDark ? 'Activer le mode clair' : 'Activer le mode sombre'}
      >
        {/* Icône selon le thème */}
        <span className="icon">
          {isDark ? '☀' : '🌙'}
        </span>
        <span>
          {isDark ? 'Mode clair' : 'Mode sombre'}
        </span>
      </button>
    </div>
  </div>
);

```

```

    </div>

    { /* Démo de différents éléments */
    <div className="demo-elements">
      <div className="card">
        <h3>Carte exemple</h3>
        <p>Le style s'adapte automatiquement au thème.</p>
      </div>

      <input
        type="text"
        placeholder="Champ de texte"
        className="demo-input"
      />
    </div>
  </div>
);
}

export default ThemeToggle;

```

```

/* src/exercices/ex08-toggle-theme/ThemeToggle.css */

/* =====
  VARIABLES CSS
  Définies différemment selon le thème
  ===== */

.theme-container {
  --bg-color: #ffffff;
  --text-color: #1f2937;
  --card-bg: #f9fafb;
  --border-color: #e5e7eb;
  --primary-color: #2563eb;

  min-height: 100vh;
  padding: 2rem;
  background-color: var(--bg-color);
  color: var(--text-color);
  transition: background-color 0.3s, color 0.3s;
}

/* Thème sombre : on redéfinit les variables */
.theme-container.dark {
  --bg-color: #111827;
  --text-color: #f9fafb;
  --card-bg: #1f2937;
  --border-color: #374151;
  --primary-color: #3b82f6;
}

/* =====

```

```
BOUTON DE TOGGLE

/*

.theme-toggle-btn {
  display: flex;
  align-items: center;
  gap: 0.5rem;
  padding: 0.75rem 1.5rem;
  font-size: 1rem;
  border: 2px solid var(--border-color);
  border-radius: 9999px;
  background: var(--card-bg);
  color: var(--text-color);
  cursor: pointer;
  transition: all 0.2s;
}

.theme-toggle-btn:hover {
  border-color: var(--primary-color);
  transform: scale(1.05);
}

.icon {
  font-size: 1.25rem;
}

/* =====
ÉLÉMENTS DE DÉMO
===== */

.demo-elements {
  margin-top: 2rem;
  display: flex;
  gap: 1rem;
  flex-wrap: wrap;
}

.card {
  background: var(--card-bg);
  border: 1px solid var(--border-color);
  border-radius: 8px;
  padding: 1rem;
  flex: 1;
  min-width: 200px;
}

.demo-input {
  padding: 0.75rem;
  border: 1px solid var(--border-color);
  border-radius: 4px;
  background: var(--card-bg);
  color: var(--text-color);
  font-size: 1rem;
}
```

Points clés

Construire une className dynamique

```
// Template literal (recommandé)
<div className={`base-class ${condition ? 'classe-a' : 'classe-b'}`}>

// Avec plusieurs classes conditionnelles
<div className={`
  base-class
  ${isDark ? 'dark' : 'light'}
  ${isLarge ? 'large' : ''}
  ${isActive && 'active'}
  `.trim()}>
```

Pattern toggle

```
// Toggle simple
setIsDark(prev => !prev);

// Équivalent à
setIsDark(!isDark); // Mais moins sûr en cas de clics rapides
```

Variables CSS pour les thèmes

Définir les couleurs comme variables CSS permet de changer tout le thème en modifiant juste les valeurs des variables.