

Exercice 7 - Rendu conditionnel

Objectif

Afficher différents éléments selon des conditions.

Concepts abordés

- Opérateur ternaire `? :`
 - Opérateur `&&` (short-circuit)
 - Instructions `if` avec early return
 - Gestion des états de chargement et d'erreur
-

Énoncé

Créer un composant `StatusPanel` qui affiche différents messages selon l'état :

1. **État de chargement** : Afficher "Chargement en cours..."
2. **État d'erreur** : Afficher le message d'erreur en rouge
3. **Liste vide** : Afficher "Aucune donnée disponible"
4. **Liste avec données** : Afficher la liste normalement
5. **Bonus** : Indicateur si l'utilisateur est admin

Props attendues

```
<StatusPanel  
  isLoading={false}  
  error=null  
  items={[...]}  
  isAdmin={true}  
/>
```

Correction

```
// src/exercices/ex07-rendu-conditionnel/StatusPanel.jsx  
  
import { useState } from 'react';  
  
function StatusPanel() {  
  // _____  
  // STATE pour simuler différents états  
  // En vrai, ces données viendraient de props ou d'un fetch  
  // _____  
  const [isLoading, setIsLoading] = useState(false);
```

```
const [error, setError] = useState(null);
const [items, setItems] = useState([
  { id: 1, name: "Article 1" },
  { id: 2, name: "Article 2" },
  { id: 3, name: "Article 3" },
]);
const [isAdmin, setIsAdmin] = useState(true);

// =====
// PATTERN 1 : EARLY RETURN
// Pour les cas qui empêchent l'affichage normal
// =====

if (isLoading) {
  return (
    <div className="status-panel loading">
      <div className="spinner"></div>
      <p>Chargement en cours...</p>
    </div>
  );
}

if (error) {
  return (
    <div className="status-panel error">
      <p className="error-message">Erreur : {error}</p>
      <button onClick={() => setError(null)}>Réessayer</button>
    </div>
  );
}

// =====
// RENDU PRINCIPAL
// =====

return (
  <div className="status-panel">
    /* =====
       PATTERN 2 : && (SHORT-CIRCUIT)
       Affiche seulement si la condition est vraie
    ===== */}

    {isAdmin && (
      <div className="admin-badge">
        Vous êtes administrateur
      </div>
    )}

    <h2>Liste des articles</h2>

    /* =====
       PATTERN 3 : TERNAIRE
       Affiche A ou B selon la condition
    ===== */}

    {items.length === 0 ? (
      <p className="empty-message">Aucune donnée disponible</p>
    ) : (
```

```

<ul className="items-list">
  {items.map(item => (
    <li key={item.id}>{item.name}</li>
  )));
</ul>
)};

/*
=====
  CONTRÔLES POUR TESTER
=====
*/}

<div className="controls">
  <button onClick={() => setIsLoading(true)}>
    Simuler chargement
  </button>
  <button onClick={() => setError("Connexion impossible")}>
    Simuler erreur
  </button>
  <button onClick={() => setItems([])}>
    Vider la liste
  </button>
  <button onClick={() => setIsAdmin(!isAdmin)}>
    Toggle Admin
  </button>
</div>
</div>
);

}

export default StatusPanel;

```

Tableau récapitulatif

Situation	Pattern	Exemple
A ou B	Ternaire ? : {isLoggedIn ? <User /> : <Login />}	
A ou rien	&&	{isAdmin && <AdminPanel />}
Plusieurs cas	if + return	Chargement, erreur, vide, normal
État d'erreur	Early return	if (error) return <Error />;

Piège à éviter

```

// ATTENTION avec && et les nombres
const count = 0;

// MAUVAIS : affiche "0" dans le DOM !
{count && <span>Messages</span>}

```

```
// BON : comparaison explicite
{count > 0 && <span>Messages</span>}
```