

TP : Composants et Props

Objectif : Apprendre à découper votre interface en plusieurs fichiers et à faire circuler les données.

Concepts clés : Props (propriétés), Décomposition de fichiers, Mapping (.map).

1. Création d'un composant enfant

Actuellement, tout votre code est dans App.tsx. Dans un vrai projet, chaque élément (une carte de livre, un bouton, une barre de recherche) est un fichier séparé.

Action : Créez un dossier src/components et un fichier BookCard.tsx.

Fichier : src/components/BookCard.tsx

```
// TODO : Définissez l'interface pour les Props
interface BookProps {
  title: string;
  author: string;
}

// TODO : Créez le composant qui reçoit ces props
const BookCard = ({__ : BookProps}) => {
  return (
    <div style={{ border: '1px solid #ccc', padding: '10px', margin: '10px' }}>
      <h3>{__}</h3>
      <p>Par : {__}</p>
    </div>
  );
};

export default BookCard;
```

2. Utilisation du jeu de données (Mapping)

Dans votre page "Librairie", vous n'allez pas écrire chaque livre à la main. Vous allez transformer un tableau de données en une liste de composants.

Fichier : src/App.tsx

```
import BookCard from './components/BookCard';
```

```

const myBooks = [
  { id: 1, title: "Le Petit Prince", author: "Saint-Exupéry" },
  { id: 2, title: "1984", author: "George Orwell" },
  { id: 3, title: "L'Étranger", author: "Albert Camus" }
];

function Library() {
  return (
    <div>
      <h2>Ma Collection</h2>
      <div className="book-list">
        { /* TODO : Utilisez .map() pour transformer le tableau en composants <BookCard /> */ }
        { myBooks.map((book) => (
          <BookCard
            key={__}
            title={__}
            author={__}
          />
        )) }
      </div>
    </div>
  );
}

```

3. Le défi LocalStorage (Bonus)

Savoir utiliser le stockage local.

Action : 1. Dans votre composant Compteur (TP 1), utilisez `useEffect` pour enregistrer la valeur du compteur dans le `localStorage` à chaque fois qu'elle change.

2. Au chargement de la page, faites en sorte que le compteur reprenne la valeur sauvegardée au lieu de repartir à 0.

Indice : > - Pour sauvegarder : `localStorage.setItem('maCle', valeur)`

- Pour récupérer : `localStorage.getItem('maCle')`

Questions de réflexion

1. **Immutabilité :** Un composant enfant a-t-il le droit de modifier lui-même une Prop qu'il a

reçue du parent ?

2. **La Key** : Pourquoi est-il obligatoire de mettre une key quand on utilise `.map()` ? Que se passe-t-il si on l'oublie ?
3. **Persistence** : Quelle est la différence entre le State et le LocalStorage ?