

Composition

props spécial children

Nous allons voir une propriété qui permet de faire passer d'autres composants à un composant donné. Cette méthode s'appelle de la composition. On utilisera la propriété particulière `this.props.children` pour la mettre en place.

```
const Sidebar = (props) => {  
  return (  
    <aside className="sidebar">  
      <Widget color="red" >  
        <nav>  
          <a href="#" >Red sidebar</a>  
        </nav>  
      </Widget>  
      <Widget color="yellow" >  
        <nav>  
          <a href="#" >Yellow sidebar</a>  
        </nav>  
      </Widget>  
    </aside>  
  );  
}  
  
// Et pour les autres Components  
  
const Widget = (props) => {  
  return (  
    <div className={'sidebar' + props.color}>  
      {props.children}  
    </div>  
  );  
}
```

Tout ce qui se trouve dans la balise `Widget` est passé en `props.children`.

Création de ses propres “entrées”

Vous pouvez également passer dans les attributs de vos composants d'autre(s) composant(s) :

```
const Split = (props) => {
```

```

    return (
      <div className="side">
        <div className="side-left">
          {props.left}
        </div>
        <div className="side-right">
          {props.right}
        </div>
      </div>
    );
  }

  const App = () => {
    return (
      <Split
        left={
          <Sidebar />
        }
        right={
          <Sidebar />
        } />
    );
  }

```

Notion de spécialisation

Vous pouvez définir un composant qui sera une spécialisation d'un composant plus générique (ce point a déjà été traité dans les exercices précédents) :

```

const Post = (props) => {

  return (
    <Widget color="yellow">
      <h1 className="title">
        {props.title}
      </h1>
      <p className="content">
        {props.content}
      </p>
    </Widget>
  );
}

const App = () =>{

```

```

    return (
      <Post
        title="Hello React"
        content="React c'est génial !" />
    );
  }
}

```

1. Exercice ajouter des utilisateurs

Prendre la liste d'utilisateurs **users** ci-dessous. Créez un composant Dashboard dans lequel vous définissez un champ pour ajouter un user à la liste des users.

```

const users = [
  { id : 1, name : "Alice" },
  { id : 2, name : "Alan" },
  { id : 3, name : "Phil" },
  { id : 4, name : "Naoudi" },
  { id : 5, name : "Fenley" },
];

```

Le composant Dashboard contiendra un composant User pour afficher les utilisateurs. Vérifiez que l'utilisateur n'est pas déjà dans la liste avant de l'ajouter.

Voici un schéma de l'arbre React que l'on souhaite monter :

```

App
├── .
│   ├── .
│   │   ├── Dashboard title=" Dashboard user "
│   │   │   ├── .
│   │   │   │   ├── User
│   │   │   │   │   ├── .
│   │   │   │   │   └── /Dashboard

```

Wireframe :

```

[ ] add user
- "Alice"
- "Alan"
- "Phil"
- "Naoudi"
- "Fenley"

```