Les collections

Les composants React

Utilisation de collection dans le JSX

Pour afficher une collection d'objet JS avec React, il est nécessaire de la convertir en collection d'élément React.

Pour cela, il est possible d'utiliser la méthode « map » des arrays Javascript.

La collection JSX obtenu peut être utilisé directement dans le JSX, React générera automatique le rendu de la liste.

Exemple d'une liste constitué d'une balise « ul » avec des éléments « li ».

```
const SimpleList = function(props) {
    const listItems = props.names.map(
        (name) => <li>{name}</li>
    );

    return (
        <ul>{listItems}</ul>
    );
}

export default SimpleList;
```

Clefs nécessaire

Le rendu des éléments fonctionne, mais il y a une erreur dans la console!

React a besoin de clefs **unique** et **stable** pour chaques éléments de la liste.

Les clefs aideront React à identifier les objets qui doivent être modifiés, ajoutés, supprimés dans la liste lors des rendus.

Ajouter l'identifiant de l'objet

L'idéal est d'utiliser une valeur présente au sein des données.

Lors de la création des éléments, utiliser l'attribut « key » sur l'élément JSX.

```
const listItems = objs.map(
    (obj) => <li key={obj.key}>{obj.data}</li>
);
```

⚠️ La valeur de la « key » de l'objet doit être unique et ne doit pas être changé.

Tips : Les librairies « nanoid » et « uuid » permettent de générer des identifiants unique.

Cas particulier : Impossible d'avoir un identifiant stable

Dans le cas où il est **impossible** d'ajouter un identifiant stable.

Il est possible **en dernier recours** d'utiliser un index avec la fonction « map ».

```
const listItems = names.map(
    (name, index) => <li key={index}>{name}</li>
);
```

⚠️ L'ordre des éléments peut changer après un rendu,
cela peut avoir un impact négatif sur les performance!

Liste d'éléments complexe

Il est possible de créer un composant React qui représente un élément de liste.

Pour cela, il faut créer un composant qui a comme noeud principal la balise adaptée.

```
const DemoListItem = function(props) {
   const {firstname, lastname, urlImage} = props;
   const altImage = `Photo de ${firstname} ${lastname}`;

   return (
      <li>
         <h3>{firstname} <span>{lastname}</span></h3>
         <img src={urlImage} alt={altImage} />
      </li>
   );
}

export default DemoListItem;
```

⚠️ Attention, l'attribut « key » reste au niveau du composant « liste ».

Exemple complet de liste React - L'application

```
import './App.css';
import { nanoid } from 'nanoid'
import ElementList from './components/demo-liste/element-list';

const App = function () {
  // Demo avec une liste pré-définie
  const list = [
    {id: nanoid(), firstname: 'Riri', lastname: 'Duck'},
    {id: nanoid(), firstname: 'Balthazar', lastname: 'Picsou'},
    {id: nanoid(), firstname: 'Zaza', lastname: 'Vanderquack'}
  ];
  return (
    <div className="App">
      <h1>Demo d'utilisation de liste</h1>
      <ElementList list={list} />
    </div>
  );
}

export default App;
```

Exemple complet de liste React - La liste

```
import PropTypes from 'prop-types';
import ElementListItem from './element-list-item';

const ElementList = function(props) {
    const listItems = props.list.map(
        (elem) => <ElementListItem key={elem.id} {...elem} />
    );
    return (
        <ul>
            {listItems}
        </ul>
    )
}
ElementList.defaultProps = {
    list: []
}
ElementList.propTypes = {
    list: PropTypes.array
}
export default ElementList;
```

Exemple complet de liste React - L'élément de liste

```
import PropTypes from 'prop-types';

const ElementListItem = function(props) {
    const {firstname, lastname} = props;
    return (
        <li>
            <h2>{firstname} {lastname}</h2>
        </li>
    )
}

ElementListItem.propTypes = {
    firstname: PropTypes.string.isRequired,
    lastname: PropTypes.string.isRequired
}

export default ElementListItem;
```

Exercice • Création d'une liste

Les composants React

Exercice

- Ajouter dans l'App un tableau (JavaScript) de produit
  - Un produit possède :
    - ➢ Un nom
    - ➢ Un prix (en euro)
    - ➢ Un booléen « En promo »

- Créer un composant liste pour afficher les différents produits

- (BONUS) Afficher en rouge le prix des produits en promotion.