

Manipulation d'objets Javascript avec React

Qu'est-ce qu'un objet Javascript ?

Un objet est une collection de données apparentées et/ou de fonctionnalités (qui, souvent, se composent de plusieurs variables et fonctions, appelées propriétés et méthodes quand elles sont dans des objets). Prenons un exemple pour voir à quoi cela ressemble :

```
const voiture = {  
  marque: "Hyundai",  
  modele: "Ioniq 5",  
  annee: 2022  
}
```

Ici nous avons un objet `voiture` qui est composé des propriétés `marque`, `modele` et `annee`. Chaque propriété a une valeur qui peut être de n'importe quel type (une chaîne de caractères, un nombre ou même un autre objet!)

Objets simples

Avec React, nous pouvons manipuler des objets et les injecter directement dans une composante de la façon suivante :

```
function Voiture() {  
  const voiture = {  
    marque: "Hyundai",  
    modele: "Ioniq 5",  
    annee: 2022  
  }  
  return (  
    <div>  
      <h1>{voiture.marque}</h1>  
      <h3>{voiture.modele}</h3>  
      <p>{voiture.annee}</p>  
    </div>  
  );  
}
```

Ici, la "fonction" `Voiture` est une composante React qui affiche les propriétés de l'objet `voiture` dans des éléments distincts, en utilisant des accolades :

- `marque` est dans un élément `<h1>`
- `modele` est dans un élément `<h3>`
- `annee` est dans un élément `<p>`

Liste d'objets

De la même manière, nous pouvons déclarer une liste d'objets et les injecter en utilisant un **index** qui va permettre de référencer un des éléments de la liste. Voici un exemple avec une liste d'objets **voitures** qui a les mêmes propriétés que l'objet utilisé dans l'exemple précédent :

```
function Voiture() {
  const voitures = [
    { marque: "Hyundai", modele: "Ioniq 5", annee: 2022 },
    { marque: "Audi", modele: "R8", annee: 2019 },
  ]

  const idx=0;

  return (
    <div>
      <h1>{voitures[idx].marque}</h1>
      <h3>{voitures[idx].modele}</h3>
      <p>{voitures[idx].annee}</p>
    </div>
  );
}
```

idx est l'index qui permet de référencer un élément de la liste **voitures**. Dans cet exemple, **idx** est égal à 0. **voitures[idx]** sera donc le premier élément de la liste d'objets **voitures**.

Voici un autre exemple qui utilise le même concept, mais avec une liste d'objets **films** (un film ayant les propriétés **titre**, **genre**, **annee**, et **image**):

```
function Film() {
  const films = [
    {titre: "Alien", genre: "SciFi", annee: 1979, image:},
    {titre: "Everything Everywhere All At Once", genre: "Comédie", annee:
2022, image:}
  ];

  const idx=1;

  return (
    <div>
      <h1>{films[idx].titre} ({films[idx].annee})</h1>
      <h3>{films[idx].genre}</h3>
    </div>
  );
}
```

Objets complexes

Comme dit dans la première partie, un objet peut même être composé d'objet(s) imbriqué(s). Dans l'exemple suivant, nous avons une composante **Personne** qui affiche les propriétés d'un objet **personne**. L'objet **personne** est lui-même composé d'un objet **adresse** qui a les propriétés **numero**, **rue** et **ville**.

```
function Personne() {  
  const personne = {  
    nom: "Chicoine",  
    prenom: "Louis",  
    adresse: {  
      numero: "2222",  
      rue: "Des Prés",  
      ville: "Sherbrooke"  
    }  
  }  
  
  return (  
    <div>  
      <h3>{personne.nom}, {personne.prenom}</h3>  
      <p>{personne.adresse.numero} {personne.adresse.rue}</p>  
      <p>{personne.adresse.ville}</p>  
    </div>  
  );  
}
```