TD 4 : Approfondissement des hooks en React.JS

Objectifs de la séance :

- Comprendre les hooks d'état *useState* en profondeur.
- Explorer les hooks d'effet *useEffect* pour gérer les effets secondaires.
- Introduction aux hooks pour la gestion de l'état global *useContext*.

Etape 1: Utilisation avancée de *useState*

1. Créez un composant **PersonDetails** qui sera appelé depuis le composant principal.

2. Dans ce composant, créez une variable à état avec *useState* pour gérer les informations de cette personne.

```
const [person, setPerson] = useState({
   name: 'Cristiano Ronaldo',
   age: 38,
   email: 'cronaldo@vaiportugues.pt',
   country: 'Portugal',
 });
```

3. Affichez toutes les informations de la personne (nom, âge, mail et ville de résidence) dans des champs de saisie et permettez à l'utilisateur de les modifier. Vous ne développerez qu'une seule fonction de modification qui sera commune aux 4 informations.

Indice : vous aurez encore besoin de l'objet event.

Etape 2: Utilisation de *useEffect* avec une requête API

1. Créez un composant **PersonInfo** qui sera appelé depuis le composant principal et qui affichera des informations sur une personne (par exemple, son nom et son âge).

2. Les informations affichées devront être récupérées en ligne depuis l'API de test :

https://jsonplaceholder.typicode.com

Pour afficher les détails en JSON d'un utilisateur depuis cette API la requête GET est la suivante :

https://jsonplaceholder.typicode.com/users/n°utilisateur

Utilisez *useEffect* pour effectuer la requête à cette API.

3. Affichez certaines de ces informations dans l'interface utilisateur (nom, âge, mail, ville de résidence).

Exercice 3: Gestion de l'état global avec *useContext*

1. Créez un composant **PeopleProvider** dans lequel vous créerez un contexte PeopleContext. Dans ce même composant, créez votre variable à état contenant vos personnes. Ainsi qu'une fonction *addPerson* qui prenne en paramètre une personne et l'ajoute dans la variable à état. Pensez à faire le paramétrage adéquat dans **app.js** également pour préciser que les futurs composants **AddPerson** et **PersonList** puissent bien accéder au contexte.

```
const [people, setPeople] = useState([
   { id: 1, name: 'Alice', age: 30 },
   { id: 2, name: 'Bob', age: 25 },
 ]);
```

2. Créez un composant **PersonList** qui accède à l'objet people depuis le contexte, et affiche le nom de toutes les personnes présentes dans la liste.

3. Créer un composant **AddPerson** qui affiche un champ de saisie (pour la saisie du nom de la nouvelle personne) et un bouton Ajouter. Ce composant accèdera à la fonction *addPerson* (cf. 3.1) depuis le contexte et y fera appel pour ajouter une nouvelle personne dans la variable à état (cf. 3.1).