React.js useEffect exercise 4 : Form input validation with useEffect

1ère étape :
Importation des Hooks useState et useEffect, définition des props et initialisation des états

```
src > exercices > ValidateInput.jsx > ...
1   import { useState, useEffect } from 'react';
2
3   const ValidatedInput = ({ validationFunction, errorMessage }) => {
4     const [value, setValue] = useState('');
5     const [isValid, setIsValid] = useState(true);
6
```

La prop validationFunction est une fonction qui est utilisée pour valider la valeur saisie dans l'input. Cette fonction prend la valeur actuelle de l'input comme argument et retourne un booléen (true si la valeur est valide, false sinon).

La prop errorMessage permet d'afficher un message d'erreur si la validation échoue.

L'état value permet de garder la valeur actuelle de l'input.

L'état isValid permet d'indiquer si la valeur de l'input est valide ou non.

2ème étape :
Le render du composant

```
src > exercices > ValidateInput.jsx > ...
3   const ValidatedInput = ({ validationFunction, errorMessage }) => {
11    return (
12      <div>
13        <input
14          type="text"
15          value={value}
16          onChange={(e) => setValue(e.target.value)}
17          className={isValid ? '' : 'error'}
18        />
19        {!isValid && <p className="error-message">{errorMessage}</p>}
20      </div>
21    );
22  };
23
24  export default ValidatedInput;
```

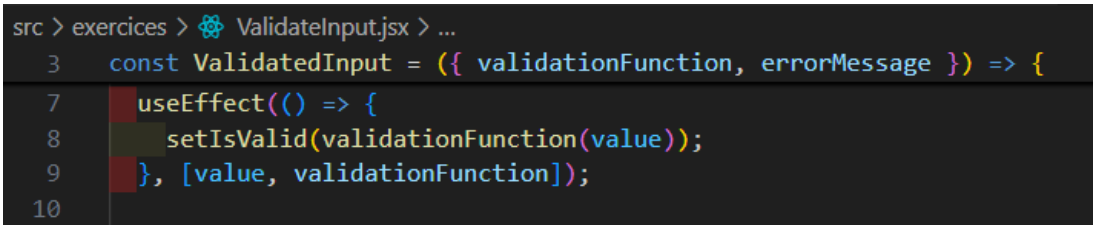
Le composant affiche un champ input dont la valeur est liée à l'état value.

La fonction liée à l'événement onChange met à jour la valeur de l'état lié à l'input chaque fois qu'une modification est apportée.

La classe CSS de l'input est déterminée par l'état de validation : si isValid est true, aucune classe spécifique n'est ajoutée, sinon la classe 'error' est ajoutée pour indiquer une erreur visuellement. Si la valeur de l'input n'est pas valide, un paragraphe contenant le message d'erreur est affiché.

3ème étape :

Ce hook useEffect est déclenché chaque fois que la valeur de l'état value ou la fonction validationFunction() change.



À chaque changement, il appelle la fonction validationFunction() avec la valeur actuelle de l'input stocké dans l'état value et met à jour l'état isValid en fonction du résultat (true si la valeur est valide, false sinon).

4ème étape :

Utilisation du composant ValidatedInput dans un composant parent (App.jsx)

ligne 4 : validateLength est la fonction de validation qui prend une chaîne de caractères (value) en argument et retourne true si la longueur de cette chaîne est supérieure ou égale à 5 caractères, ou false sinon.

ligne 9 : Cette fonction sera utilisée pour valider la longueur de la valeur entrée par l'utilisateur dans le champ de saisie, elle est passée en prop au composant ValidatedInput

ligne 10 : errorMessage est le prop définissant le message d'erreur qui sera affiché sous l'input si la validation échoue (c'est-à-dire si la chaîne de caractères est inférieure à 5 caractères).