Hook useState

Thèmes	React
Description	Déclarer et gérer l'état d'un composant.
Date de création	@16 mars 2025 17:19
Créée par	Kévin Wolff

Chapitres

Sur le même sujet

- Premier composant React
- Cycle de vie d'un composant
- Hook useEffect
- Hook custom

Introduction

🔗 Documentation officielle - useState

Le hook `useState` permet de déclarer et gérer l'état local d'un composant fonctionnel. Il est indispensable pour créer des interfaces interactives en permettant aux composants de stocker et de modifier des données sans dépendre d'un état global.

Importez `useState` depuis React avant de l'utiliser :

```
import { useState } from 'react';
```

Une fois importé, il permet de déclarer un état au sein du composant de la manière suivante :

```
const [state, setState] = useState(initialState);
```

- `state` : variable qui contient l'état actuel.
- `setState` : fonction qui permet de mettre à jour l'état.
- `initialState` : la valeur initiale de l'état.

Si vous êtes adepte de la programmation orientée objet, vous pouvez voir `state` comme une propriété et `setState` comme un setter permettant de modifier cette propriété.

```
import { useState } from 'react';

function Counter() {
```

```
  const [count, setCount] = useState(0);

  const handleIncrement = () => {
    setCount(count + 1);
  };

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={handleIncrement}>Incrémenter</button>
    </div>
  );
}
```

Dans cet exemple, le hook `useState()` crée une variable d'état `count` et une fonction `setCount()` pour la modifier. L'état est initialisé à `0` via `useState(0)`. À chaque clic sur le bouton, `setCount(count + 1)` met à jour la valeur de `count`.

Grâce à cette mise à jour de l'état, le composant est re-render (voir) pour afficher la nouvelle valeur de `count`.