

- [Pourquoi utiliser des hooks \(crochets\) ?](#)

Pourquoi utiliser des hooks (crochets) ?

Que devons-nous faire si nous voulons ajouter un état à notre composant fonction ? Et si nous voulions que notre application réponde aux changements de données ?

Dans cette leçon, nous découvrirons *React Hooks* et comment ils peuvent nous aider à utiliser puissamment les composants fonctionnels.

En termes simples, les React Hooks sont des fonctions qui nous permettent de gérer l'état interne des composants et de gérer les effets secondaires post-rendu directement à partir de nos composants de fonction. À l'aide de Hooks, nous pouvons déterminer ce que nous voulons montrer aux utilisateurs en déclarant à quoi devrait ressembler notre interface utilisateur en fonction de l'état.

React propose un certain nombre de Hooks intégrés. Quelques-uns d'entre eux incluent `useState()`, `useEffect()`, `useContext()`, `useReducer()` et `useRef()`. Consultez [la liste complète](#) dans la documentation React.

Dans cette section, nous allons apprendre à :

- Créez un composant de fonction « avec état ».
- Utilisez le crochet d'État.
- Initialisez un état et définissez un état.
- Définissez les gestionnaires d'événements.
- Utilisez les fonctions de rappel du paramètre d'état.
- Utilisez l'état avec des tableaux et des objets.