

Hooks

En React, un "hook" es como una herramienta mágica que te ayuda a agregar funcionalidades especiales a tus componentes. Piénsalo como piezas de Lego que puedes conectar a tus bloques de construcción (componentes) para hacer cosas geniales. Por ejemplo, hay "hooks" que te permiten recordar información, otros para estar al tanto de cambios y más. Son como superpoderes que le das a tus componentes para que hagan cosas increíbles.

De forma técnica, es una función especial que te permite usar características de React en componentes basados en funciones. Los hooks fueron introducidos en React 16.8 para permitir el uso de estado y otras características de clase en componentes funcionales, eliminando así la necesidad de utilizar clases para ciertos casos.

useState

Imagina que estás construyendo una casa con bloques de construcción. Cada bloque es un pedazo de información que tu casa necesita recordar, como tu nombre o el color de tus paredes. Ahora, useState en React es como un bloc de notas especial que puedes pegar en la pared. Puedes escribir algo en él y, cada vez que necesites recordar o cambiar algo, simplemente consultas ese bloc de notas. Es una forma fácil de que tu casa (o tu componente React) recuerde cosas importantes.

De forma técnica, `useState` es un hook que te permite agregar estado a un componente funcional. El estado es básicamente información que tu componente necesita recordar y, cuando ese estado cambia, React se encarga de volver a renderizar el componente.

Por ejemplo, supongamos que tienes un contador en tu aplicación. Puedes usar `useState` para declarar una variable de estado, digamos `count`, y una función que te permite actualizar ese estado, digamos `setCount`. Al principio, `count` sería 0, pero cuando llamas a `setCount(1)`, React actualiza el estado y vuelve a renderizar el componente con el nuevo valor de `count`.

```
import React, { useState } from 'react'

const Counter = () => {
  const [count, setCount] = useState(0)

  return (
    <div>
      <button onClick={() => setCount(count + 1)}>Click to increment</button>
      <button onClick={() => setCount(count - 1)}>Click to decrement</button>
      <p>{count}</p>
    </div>
  )
}
```