

React-useState

useState Hook in React

The `useState` hook is a built-in Hook in React that allows you to add state to functional components. This means you can now have stateful components without having to write class-based components.

What is State?

State refers to the data and properties of a component that change over time, as opposed to props which are read-only values passed from a parent component.

How does useState work?

When you call `useState()` in your functional component, React will add a new state variable and an update function to the component. The state variable is initially set to its default value (which you specify), and the update function allows you to change the state.

Here's an example:

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

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

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </div>
  );
}
```

In this example:

- We import the `useState` hook from React.
- Inside our `Counter` component, we call `useState(0)` to initialize a state variable called `count` with an initial value of `0`.
- The `useState` hook returns an array where the first element is the current count (i.e., the state variable), and the second element is an update function (`setCount`) that allows us to change the count.
- We use the `setCount` function in our button's `onClick` event handler to increment the count each time it's clicked.

Key Takeaways

1. The `useState` hook adds a state variable and an update function to a functional component.
2. You can specify the initial value of the state variable when you call `useState`.
3. The update function allows you to change the state, making your component interactive.
4. This example demonstrates how to use `useState` with a simple counter app.

Curated by Brajesh Kumar