

Managing Data with State

In this lecture, we will learn about state in React. While props let us pass data into components, state allows components to manage their own data and update the user interface when that data changes.

What is State?

State is like a variable that belongs to a component. Unlike normal variables, when state changes, React automatically re-renders the component to reflect the new data.

The useState Hook

React provides a special function called `useState` to create and manage state. It is called a “hook” because it hooks into React features.

Example:

```
import { useState } from "react"

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

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click Me</button>
    </div>
  )
}

export default App
```

How it works:

- `useState(0)` creates a state variable named `count` with an initial value of `0`.
- `setCount` is the function used to update the state.
- Whenever `setCount` is called, React re-renders the component with the new state value.

Multiple State Variables

You can have more than one state variable inside a component:

```
function Profile() {  
  const [name, setName] = useState("Harry")  
  const [age, setAge] = useState(19)  
  
  return (  
    <div>  
      <h2>{name}</h2>  
      <p>Age: {age}</p>  
    </div>  
  )  
}
```

Each call to `useState` creates a separate piece of state.

Recap

- State is used to store data inside a component.
- The `useState` hook provides a way to declare and update state.
- Updating state automatically re-renders the component.