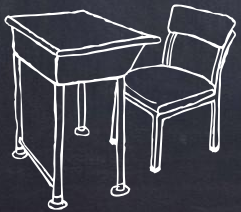


State management in React



By: T. lingesh





State Mangement

State management is a crucial aspect of building React applications, as it allows you to manage and update the internal state of your components. React provides a built-in state management system, but for more complex applications, you might need additional tools or libraries to handle state more efficiently.

Component State:

React components can have local state managed using the “useState” hook. This is suitable for simple state requirements within a component.

Code:

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

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

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


SAMPLE CODE

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

const Counter = () => {
  // Step 1: Define state using the useState hook
  const [count, setCount] = useState(0);

  // Step 2: Define functions to update the state
  const increment = () => {
    setCount(count + 1);
  };

  const decrement = () => {
    setCount(count - 1);
  };

  // Step 3: Render the component with the state
  return (
    <div>
      <h2>Counter</h2>
      <p>Count: {count}</p>

      {/* Step 4: Interact with the state using event handlers */}
      <button onClick={increment}>Increment</button>
      <button onClick={decrement}>Decrement</button>
    </div>
  );
};

export default Counter;
```

1) State Initialization:

We use the `useState` hook to initialize the state variable `count` with an initial value of 0. The `useState` hook returns an array with two elements: the current state value and a function to update the state.

2) Updating State:

We define two functions, `increment` and `decrement`, that use the `setCount` function to update the `count` state. These functions will be called when the corresponding buttons are clicked

3) Rendering State:

The current state value `count` is rendered in the component JSX, showing the count on the webpage

Interacting with State:

Two buttons trigger the increment and decrement functions when clicked, updating the state and re-rendering the component.

Exporting the Component:

```
export default Counter;
```

This exports the Counter component so that it can be imported and used in other parts of the application.

Overall, this code creates a simple React component (Counter) with a state variable (count) that can be incremented and decremented by clicking buttons. The state is managed using the useState hook, and the component re-renders whenever the state changes.

THANK YOU

