State and Hooks in Next.js

Managing Component Logic and State

What is State?

State is a way to store and manage data that can change over time in a component.

Allows components to respond to user actions, fetch data, and control UI behavior.

Example: A button click updating a counter.

Introduction to Hooks

Hooks are special functions in React (and Next.js) that let you use state and other React features in functional components.

Key Hooks:

- useState for state management
- useEffect for side effects like data fetching

Note: Hooks allow you to write cleaner, more maintainable function components.

The useState Hook

`useState` is a hook that lets you add state to function components.

Syntax:

```
const [state, setState] = useState(initialValue);
```

```
const [count, setCount] = useState(0);
```

Code Example

Button click increments the count state.

State is preserved between re-renders.

```
12    import { useState } from 'react';
13
14  function Counter() {
15    const [count, setCount] = useState(0);
16
17    return (
18      <div>
19        <p>You clicked {count} times</p>
20        <button onClick={() => setCount(count + 1)}>
21          Click me
22        </button>
23      </div>
24    );
25  }
26
27  export default Counter;
```

The useEffect Hook

useEffect allows you to perform side effects in function components (like fetching data, updating the DOM, or setting up a timer).

Syntax:

```
useEffect(() => {
    // Side effect logic here
}, [dependencies]);
```

Example with useEffect

Explanation:

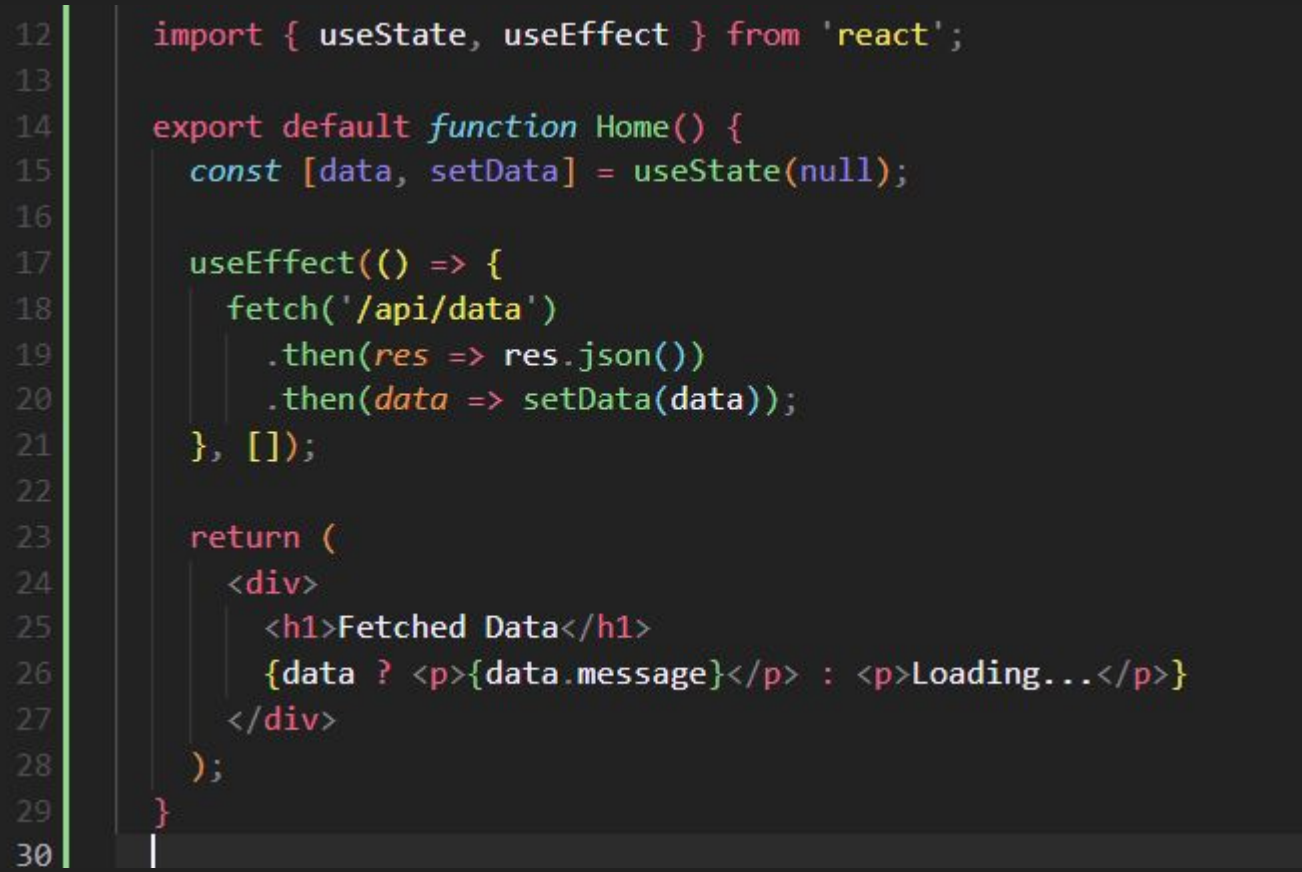
Timer increments every second using setInterval.

Cleanup function ensures interval is cleared when the component is removed.

```
12  import { useState, useEffect } from 'react';
13
14  function Timer() {
15    const [time, setTime] = useState(0);
16
17    useEffect(() => {
18      const interval = setInterval(() => {
19        setTime(time => time + 1);
20      }, 1000);
21      return () => clearInterval(interval);
22    }, []);
23
24    return <p>Timer: {time} seconds</p>;
25  }
26
27  export default Timer;
28
        You, 4 weeks ago • Uncommitted changes
```

State and Side Effects in Next.js

Hooks like useState and useEffect work in Next.js pages just like in any React app.



Multiple States and Effects

Components can have multiple

useState hooks to manage

different pieces of state.

```jsx
function Form() {
  const [name, setName] = useState('');
  const [email, setEmail] = useState('');

  return (
    <form>
      <input
        type="text"
        value={name}
        onChange={e => setName(e.target.value)}
        placeholder="Name"
      />
      <input
        type="email"
        value={email}
        onChange={e => setEmail(e.target.value)}
        placeholder="Email"
      />
    </form>
  );
}
```

You, 4 weeks ago • Uncommitted changes

Create a New Counter Page Using TypeScript (TSX)

Challenge 1: Add a new button to double the current count:

Challenger 2” Add styles to button