Full Stack Development with AI Programme

Session 10

Nida

Tuesday, 17 Jun 2025

NUS
National University of Singapore
National University of Singapore

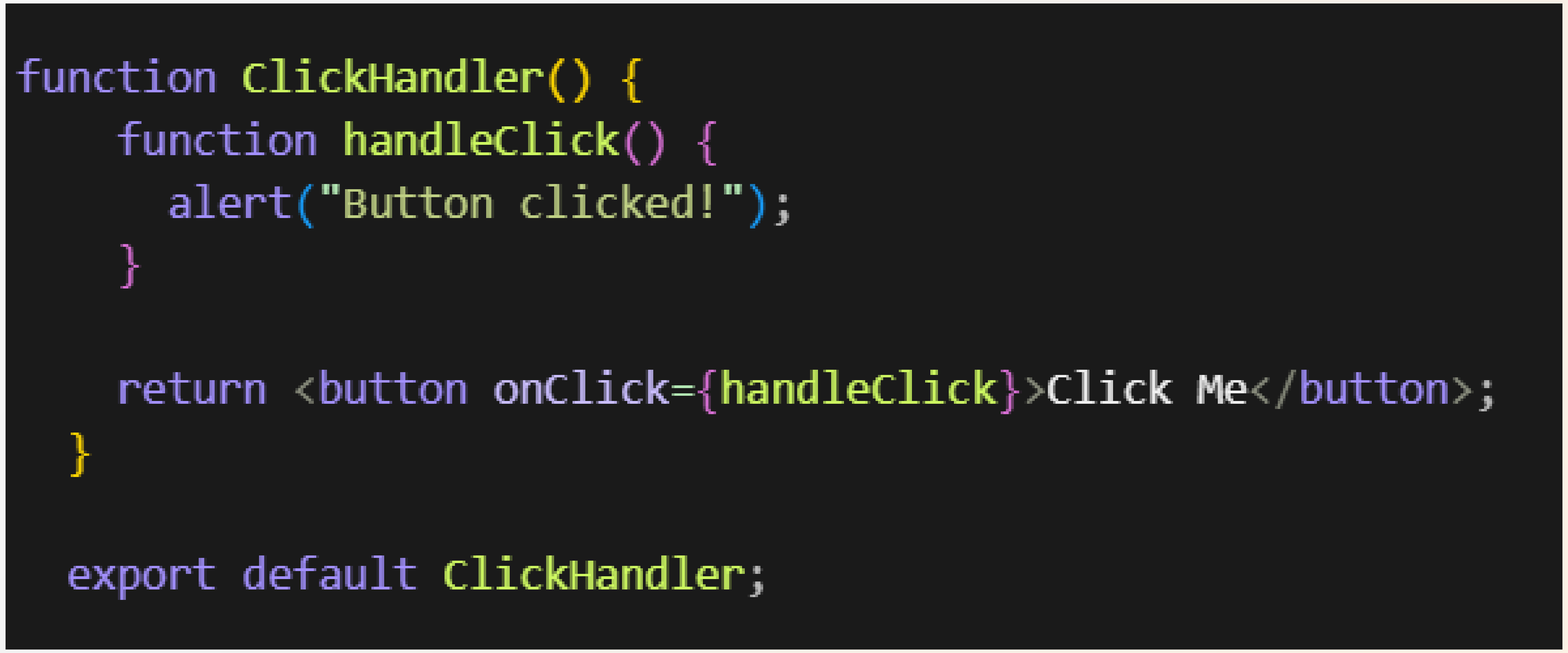
NUS National University of Singapore | Advanced Computing for Executives School of Computing

Agenda

- Handle HTML and React Events
- Difference between props and states
- React Hooks and State Variables
- Controlled vs. Uncontrolled React Components
- Fetch() API
- useEffect Hook
- React Router

Handle HTML and React Events

- HTML Events: onclick, onchange, etc
- React Events: Synthetic events (eg., onClick, onChange)
- Why use synthetic events? Performance and cross-browser compatibility.

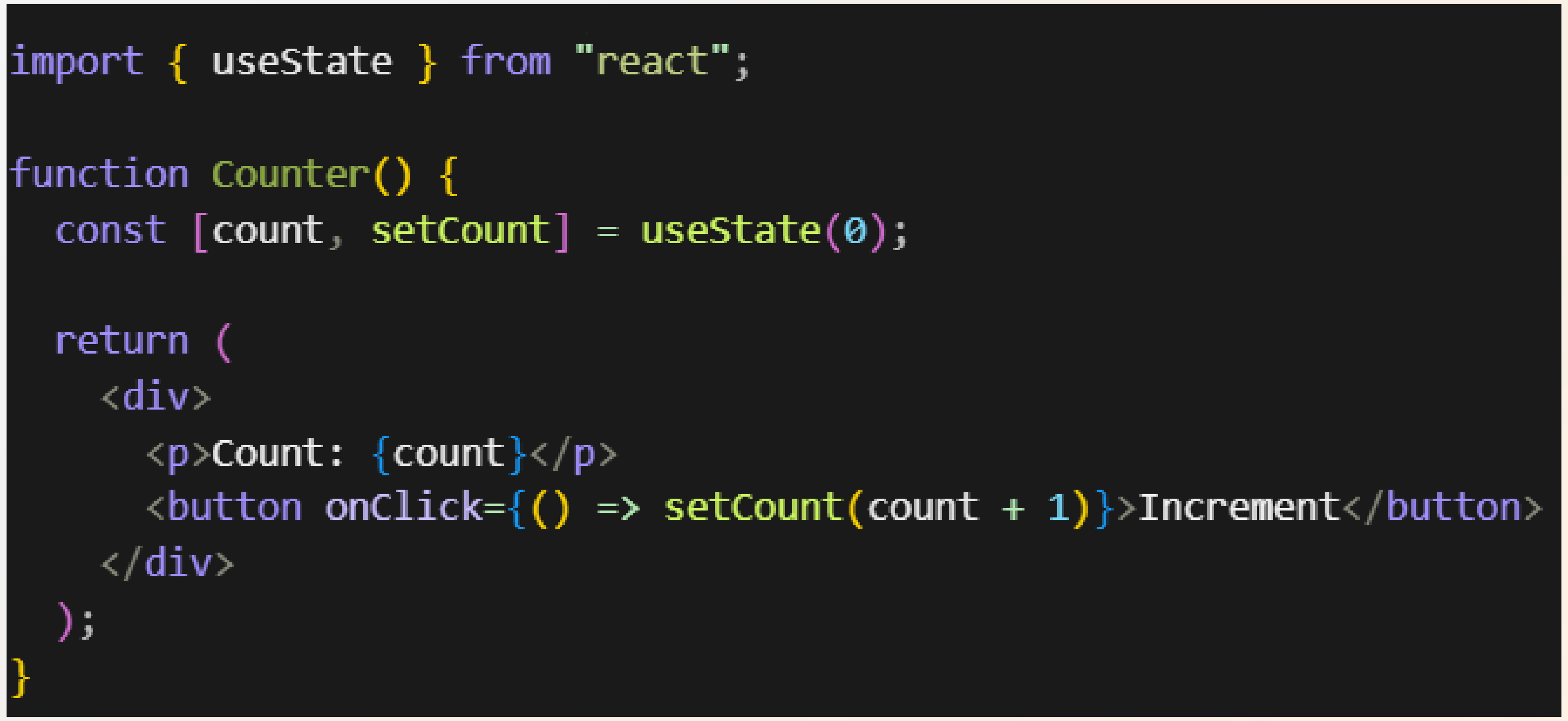


```
function ClickHandler() {
    function handleClick() {
      alert("Button clicked!");
    }

    return <button onClick={handleClick}>Click Me</button>;
  }

  export default ClickHandler;
```

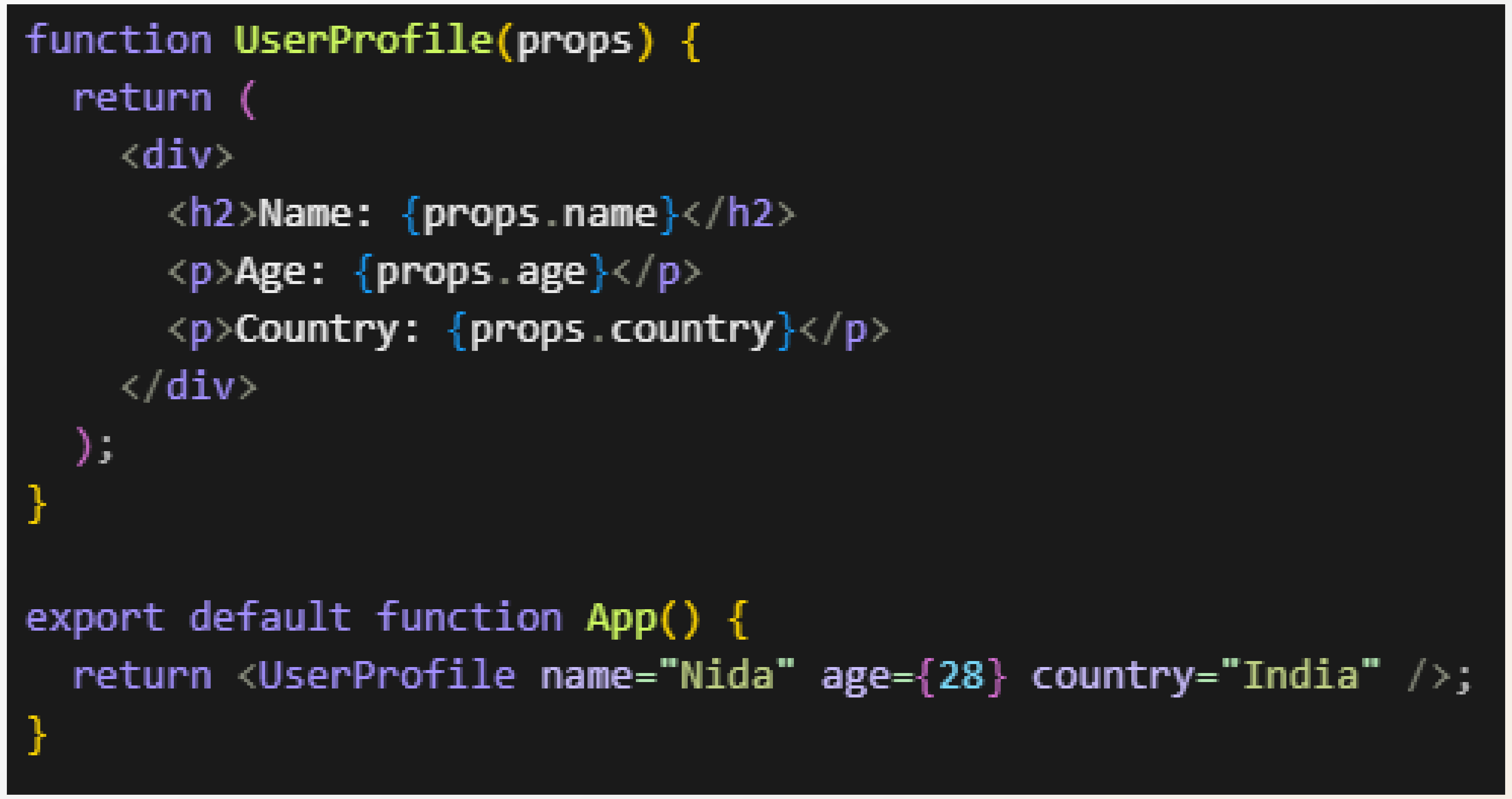
Difference between Props and State

Props: Read-only, passed from parent to child

State: Managed within a component, mutable.



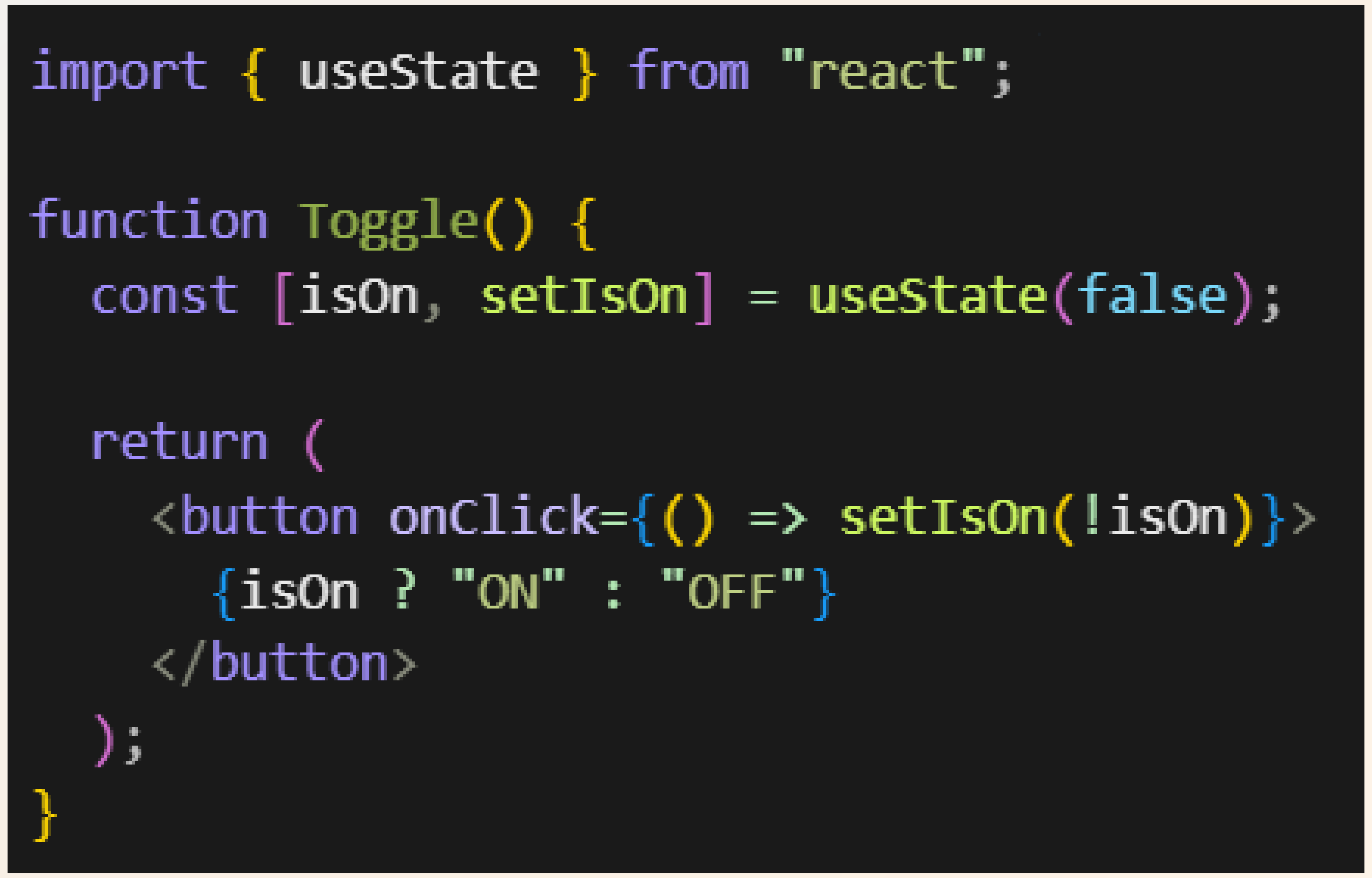
```
import { useState } from "react";

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

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



```
function UserProfile(props) {
  return (
    <div>
      <h2>Name: {props.name}</h2>
      <p>Age: {props.age}</p>
      <p>Country: {props.country}</p>
    </div>
  );
}

export default function App() {
  return <UserProfile name="Nida" age={28} country="India" />;
}
```

React Hooks and State Variables

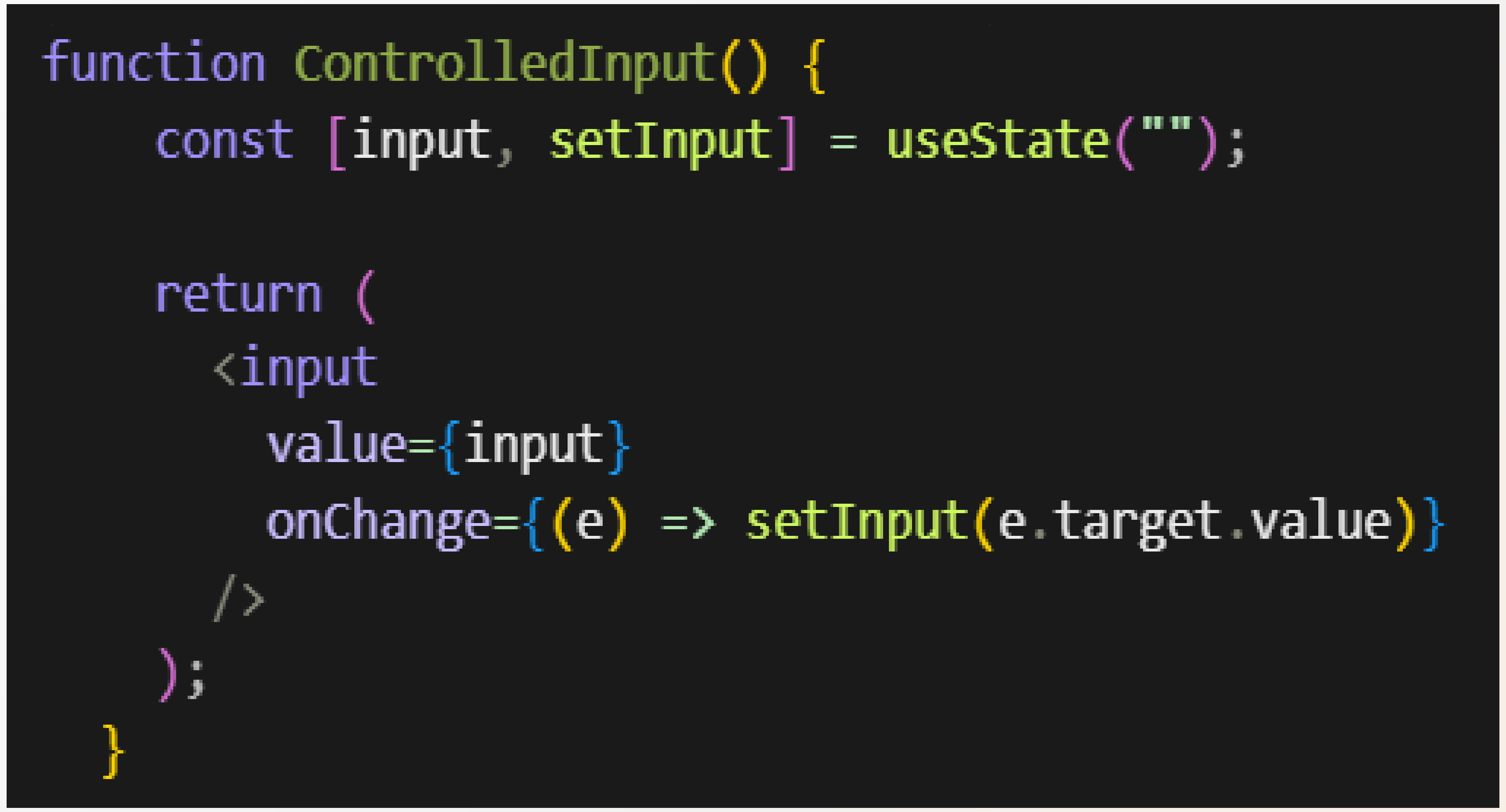
- What are hooks?
  - Hooks are special functions in React that let you **use state and other React features in functional components**. Before hooks, state and lifecycle methods were only available in class components. Hooks were introduced in **React 16.8** to simplify React development by allowing functional components to manage state and side effects.
- Common hooks: useState, useEffect, useRef



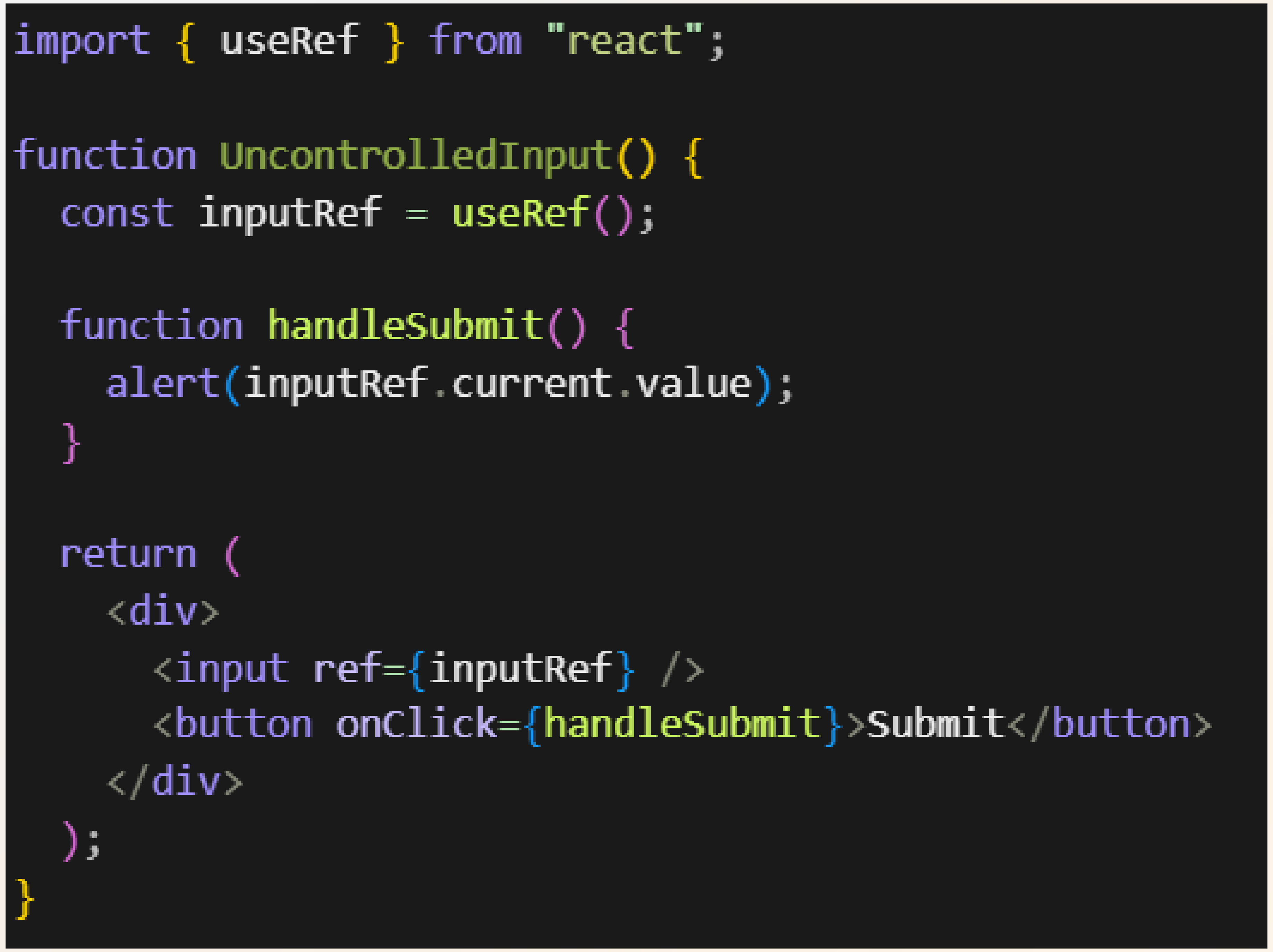
```
import { useState } from "react";

function Toggle() {
  const [isOn, setIsOn] = useState(false);

  return (
    <button onClick={() => setIsOn(!isOn)}>
      {isOn ? "ON" : "OFF"}
    </button>
  );
}
```

Controlled vs. Uncontrolled React Components

- Controlled Components:
  - State is managed by React (useState).
- Uncontrolled Components:
  - State is managed by the DOM (useRef).

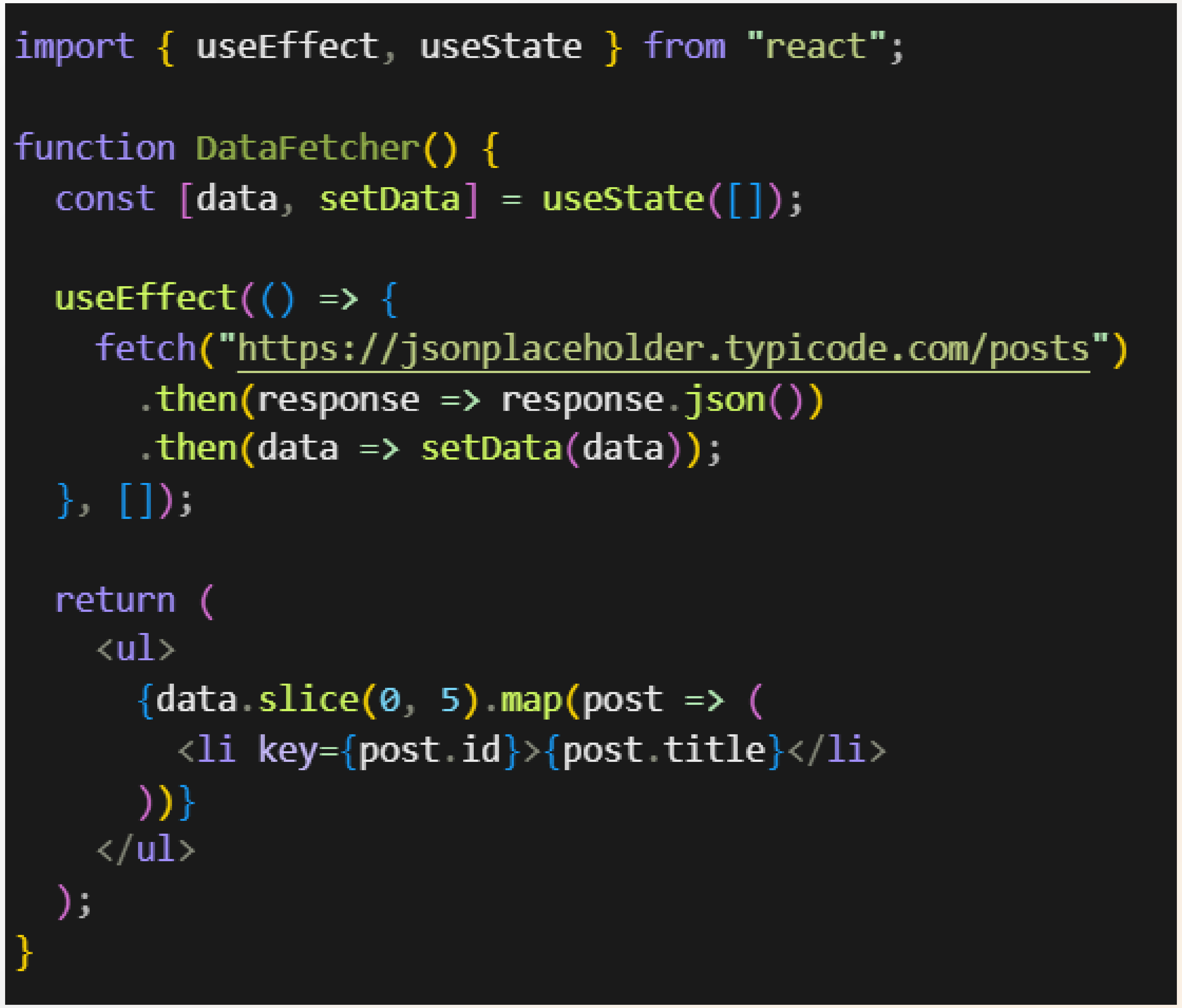


```
function ControlledInput() {
    const [input, setInput] = useState("");

    return (
      <input
        value={input}
        onChange={(e) => setInput(e.target.value)}
      />
    );
  }
```



```
import { useRef } from "react";

function UncontrolledInput() {
  const inputRef = useRef();

  function handleSubmit() {
    alert(inputRef.current.value);
  }

  return (
    <div>
      <input ref={inputRef} />
      <button onClick={handleSubmit}>Submit</button>
    </div>
  );
}
```

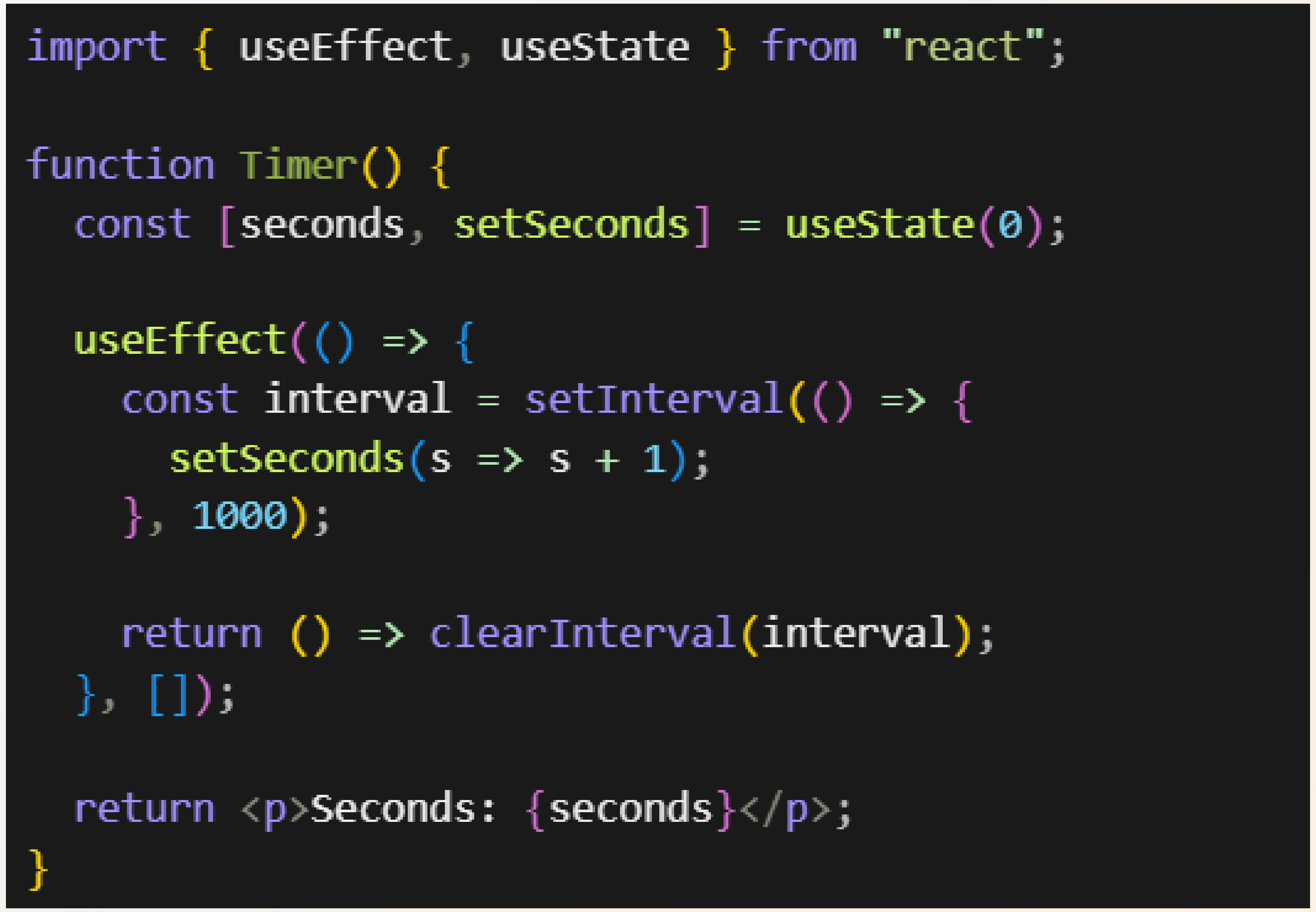
Fetch() API



```
import { useEffect, useState } from "react";

function DataFetcher() {
  const [data, setData] = useState([]);

  useEffect(() => {
    fetch("https://jsonplaceholder.typicode.com/posts")
      .then(response => response.json())
      .then(data => setData(data));
  }, []);

  return (
    <ul>
      {data.slice(0, 5).map(post => (
        <li key={post.id}>{post.title}</li>
      ))}
    </ul>
  );
}
```

useEffect Hook

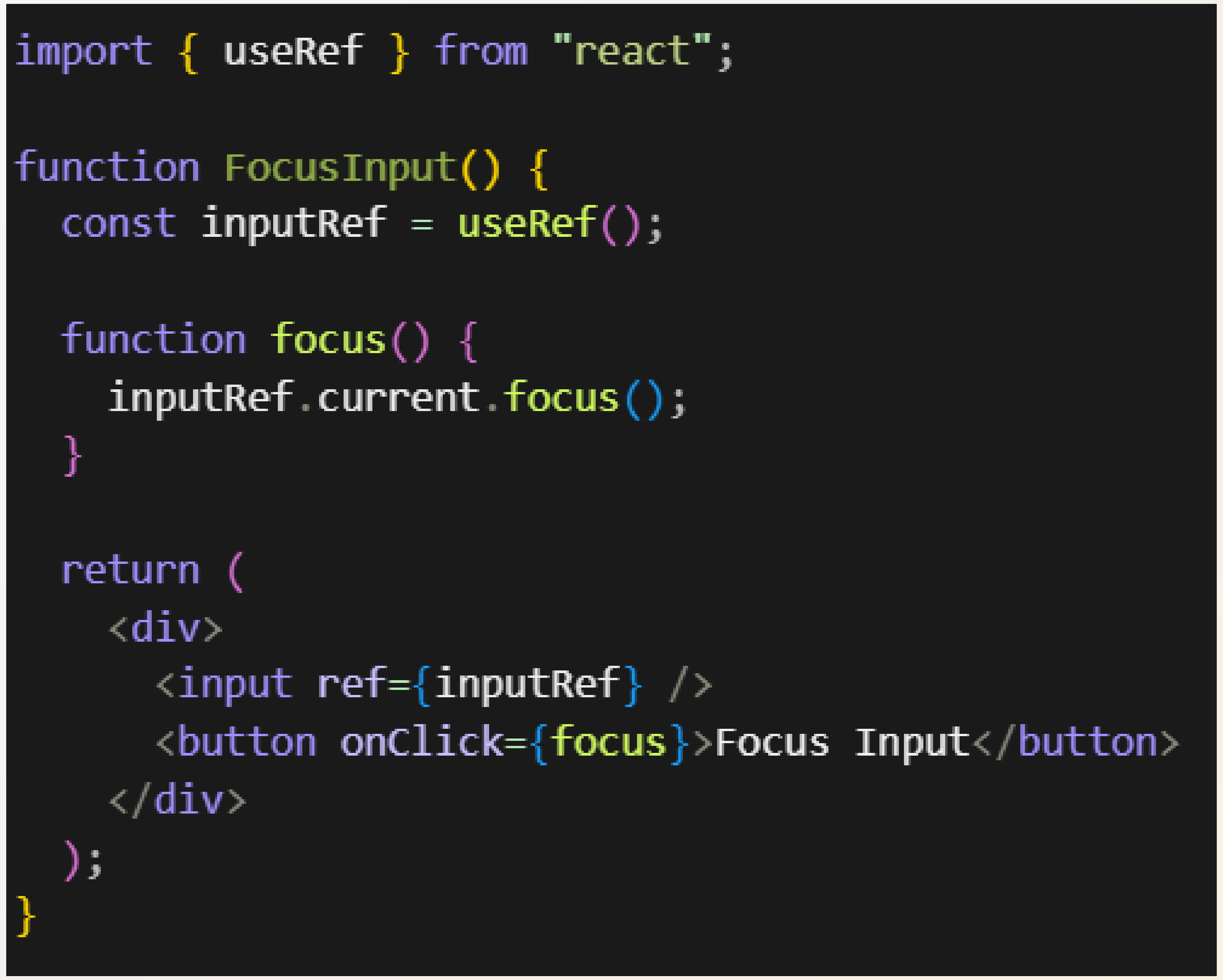
- When to use useEffect
- Dependency arrays and cleanup functions



```
import { useEffect, useState } from "react";

function Timer() {
  const [seconds, setSeconds] = useState(0);

  useEffect(() => {
    const interval = setInterval(() => {
      setSeconds(s => s + 1);
    }, 1000);

    return () => clearInterval(interval);
  }, []);

  return <p>Seconds: {seconds}</p>;
}
```

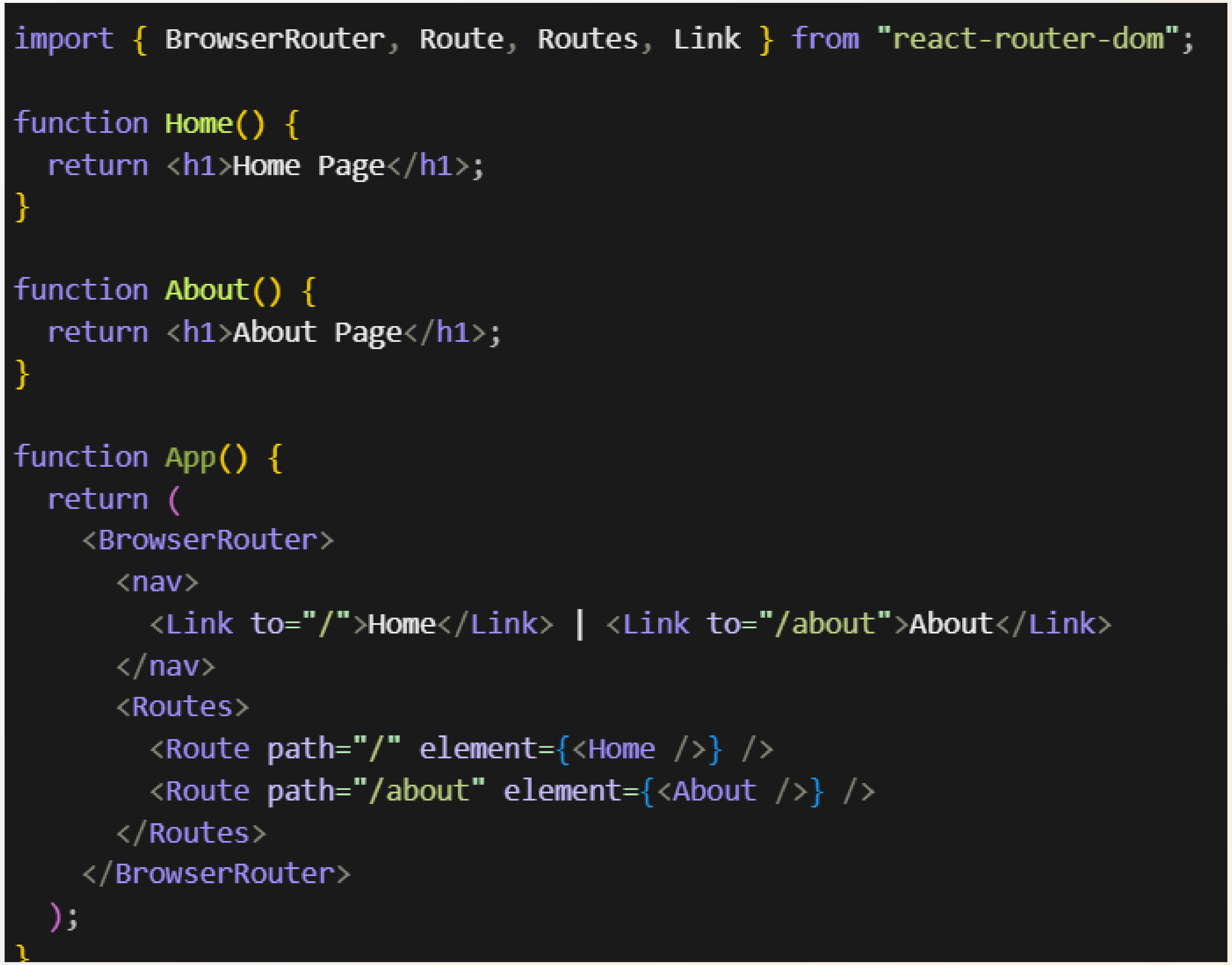
useRefHook

- useRef for referencing DOM elements
- useRef for persisting values across renders



```
import { useRef } from "react";

function FocusInput() {
  const inputRef = useRef();

  function focus() {
    inputRef.current.focus();
  }

  return (
    <div>
      <input ref={inputRef} />
      <button onClick={focus}>Focus Input</button>
    </div>
  );
}
```

React Router



```
import { BrowserRouter, Route, Routes, Link } from "react-router-dom";

function Home() {
  return <h1>Home Page</h1>;
}

function About() {
  return <h1>About Page</h1>;
}

function App() {
  return (
    <BrowserRouter>
      <nav>
        <Link to="/">Home</Link> | <Link to="/about">About</Link>
      </nav>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </BrowserRouter>
  );
}
```

LIVE SESSION EXPERIENCE SURVEY

Before we proceed to Q&A
Take 2 minutes to share your feedback with Us!

Q&A

questions

answers

NUS National University of Singapore

Advanced Computing for Executives
School of Computing

THANK YOU