CoGrammar

Welcome to this session:

Context API for State Management

The session will start shortly...

Questions? Drop them in the chat. We'll have dedicated moderators answering questions.

Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:

Ian Wyles
Designated Safeguarding Lead

Simone Botes

Nurhaan Snyman

Rafiq Manan

Ronald Munodawafa

Tevin Pitts

Scan to report a safeguarding concern

or email the Designated Safeguarding Lead:
Ian Wyles
safeguarding@hyperiondev.com

CoGrammar HyperionDev

Skills Bootcamp Cloud Web Development

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Skills Bootcamp Cloud Web Development

- For all **non-academic questions**, please submit a query:
  www.hyperiondev.com/support

- **Report a safeguarding incident: *www.hyperiondev.com/safeguardreporting***

- We would love your feedback on lectures: *Feedback on Lectures*

- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.

What is a state variable in React.js?

A. A variable that triggers a re-render every time its value changes.

B. Stores metadata about the component.

C. Variable responsible for controlling the react component lifecycle.

Learning Outcomes

- Define the concept of state management within the context of React.js applications.
- Demonstrate the usage of the useState hook to declare and manage local state.
- Identify the concept of prop drilling and its drawbacks in large React component hierarchies.
- Describe the purpose and functionality of React Context API for managing global state.

State Management

- ❖ State management is the process of handling and updating data within a React application.
- ❖ It allows components to maintain their internal state and respond to user interactions effectively.

CoGrammar

What is State in React?

- ❖ In React, state refers to an object that represents the current condition of a component.
- ❖ Stateful components have the ability to hold and modify their state, which affects their rendering and behavior.

CoGrammar

How Does State Work?

- ❖ When a component's state changes, React automatically re-renders the component to reflect the updated state.
- ❖ Changes to state trigger a re-render of the component and its child components, ensuring that the UI stays in sync with the underlying data.

CoGrammar

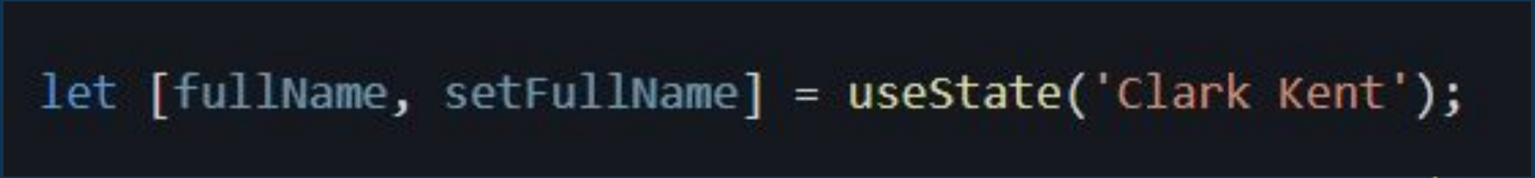
useState Hook

- ❖ In functional components, we use the useState hook to introduce stateful behavior.
- ❖ The useState hook allows us to declare state variables and update them within the component.

CoGrammar

useState Hook

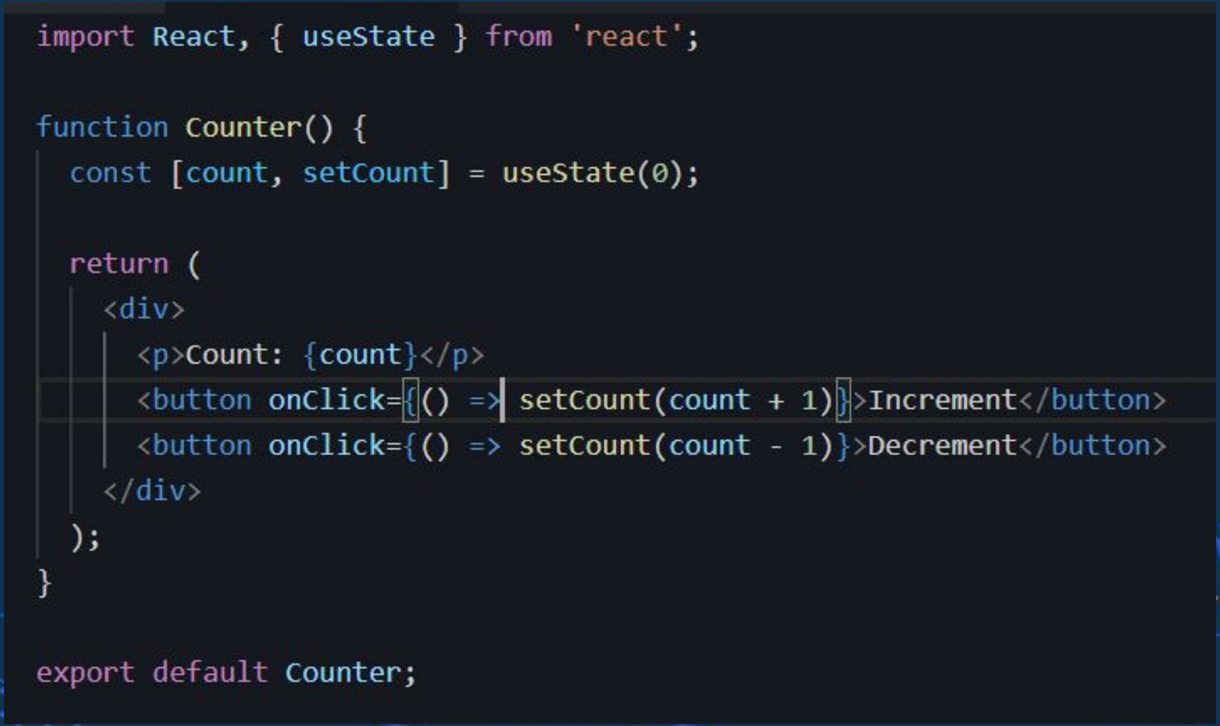
- state: Represents the current value of the state variable.
- setState: A function used to update the state variable and trigger re-rendering.



```
let [fullName, setFullName] = useState('Clark Kent');
```

CoGrammar

Example: Counter Component



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

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

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

export default Counter;
```

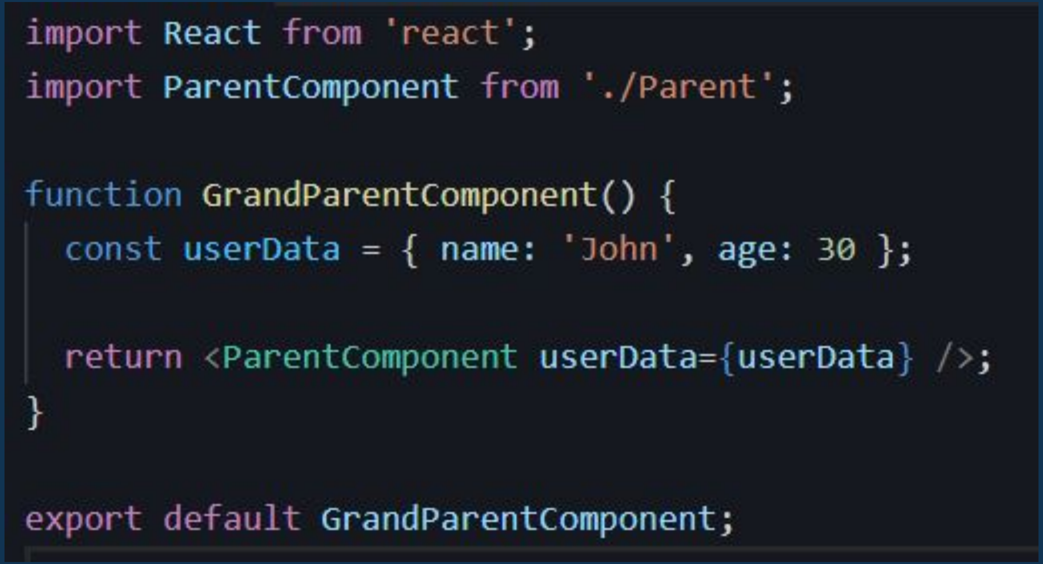
CoGrammar

Prop drilling

- Prop drilling is a common challenge in React applications where data needs to be passed through multiple layers of components.
- It arises when passing props down several levels in the component tree, leading to code complexity and maintenance issues.

CoGrammar

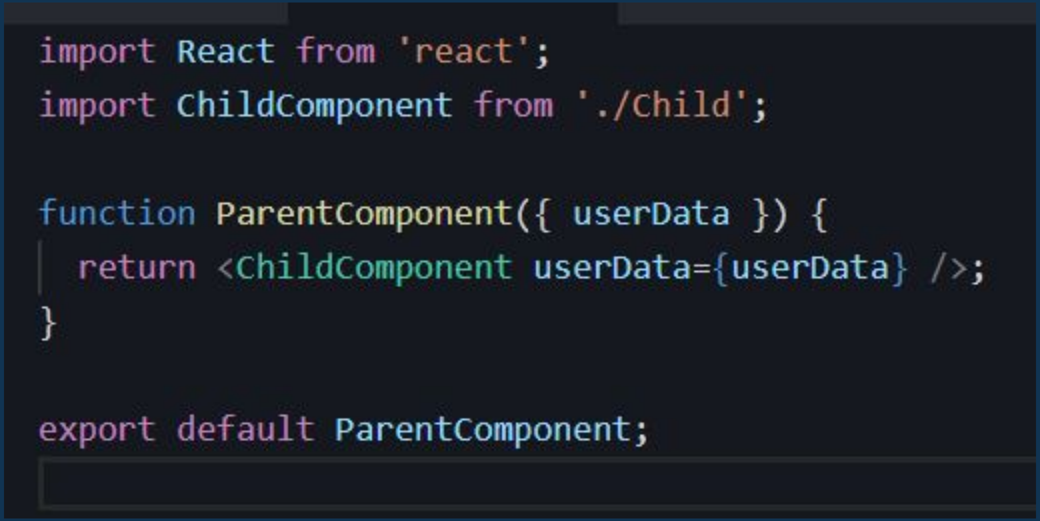
Prop drilling: Examples



```
import React from 'react';
import ParentComponent from './Parent';

function GrandParentComponent() {
  const userData = { name: 'John', age: 30 };

  return <ParentComponent userData={userData} />;
}

export default GrandParentComponent;
```

CoGrammar

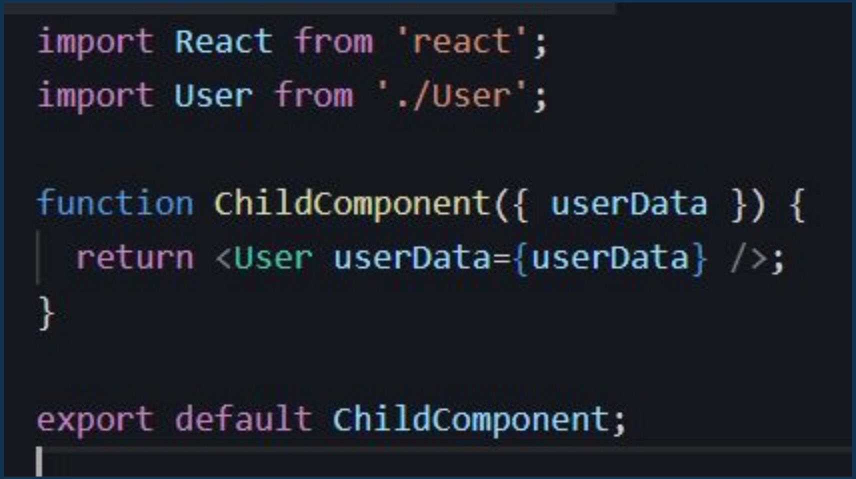
Prop drilling: Examples



```
import React from 'react';
import ChildComponent from './Child';

function ParentComponent({ userData }) {
  return <ChildComponent userData={userData} />;
}

export default ParentComponent;
```

CoGrammar

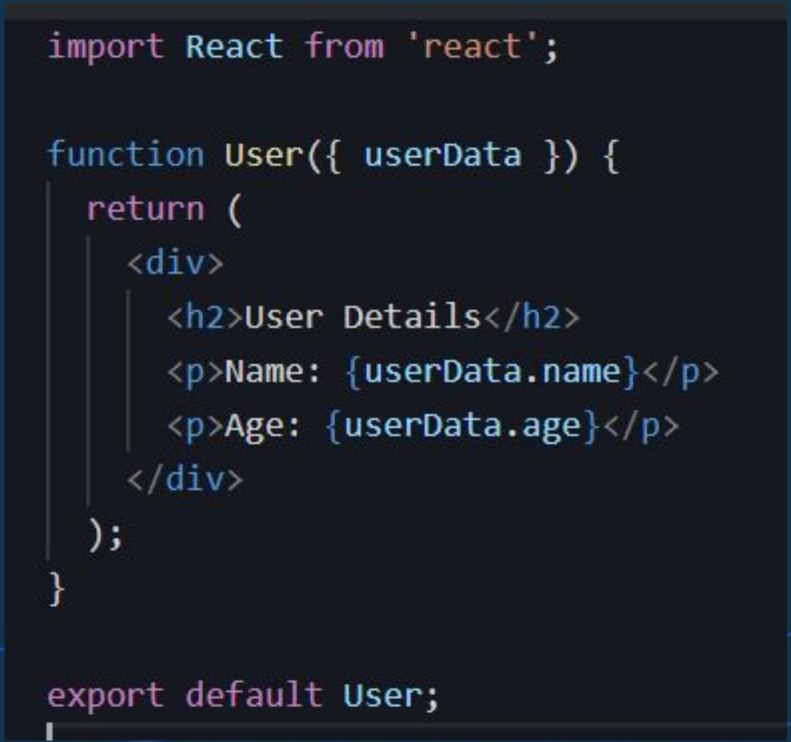
Prop drilling: Examples



```
import React from 'react';
import User from './User';

function ChildComponent({ userData }) {
  return <User userData={userData} />;
}

export default ChildComponent;
```

CoGrammar

Prop drilling: Examples



```
import React from 'react';

function User({ userData }) {
  return (
    <div>
      <h2>User Details</h2>
      <p>Name: {userData.name}</p>
      <p>Age: {userData.age}</p>
    </div>
  );
}

export default User;
```

CoGrammar

Challenges of Prop Drilling

- Prop drilling can make code harder to maintain and refactor, especially in large component hierarchies.
- It increases coupling between components and makes it difficult to track data flow.

CoGrammar

Let’s Breathe!

Let’s take a small break before moving on to the next topic.

CoGrammar

React Context API

- ❖ React Context API provides a solution for managing global state in React applications.
- ❖ It allows components to share data without explicitly passing props through each level of the component tree.

CoGrammar

Creating a Context

- ❖ To create a context, we use the createContext function provided by React.
- ❖ We define a context for our data and provide a default value.

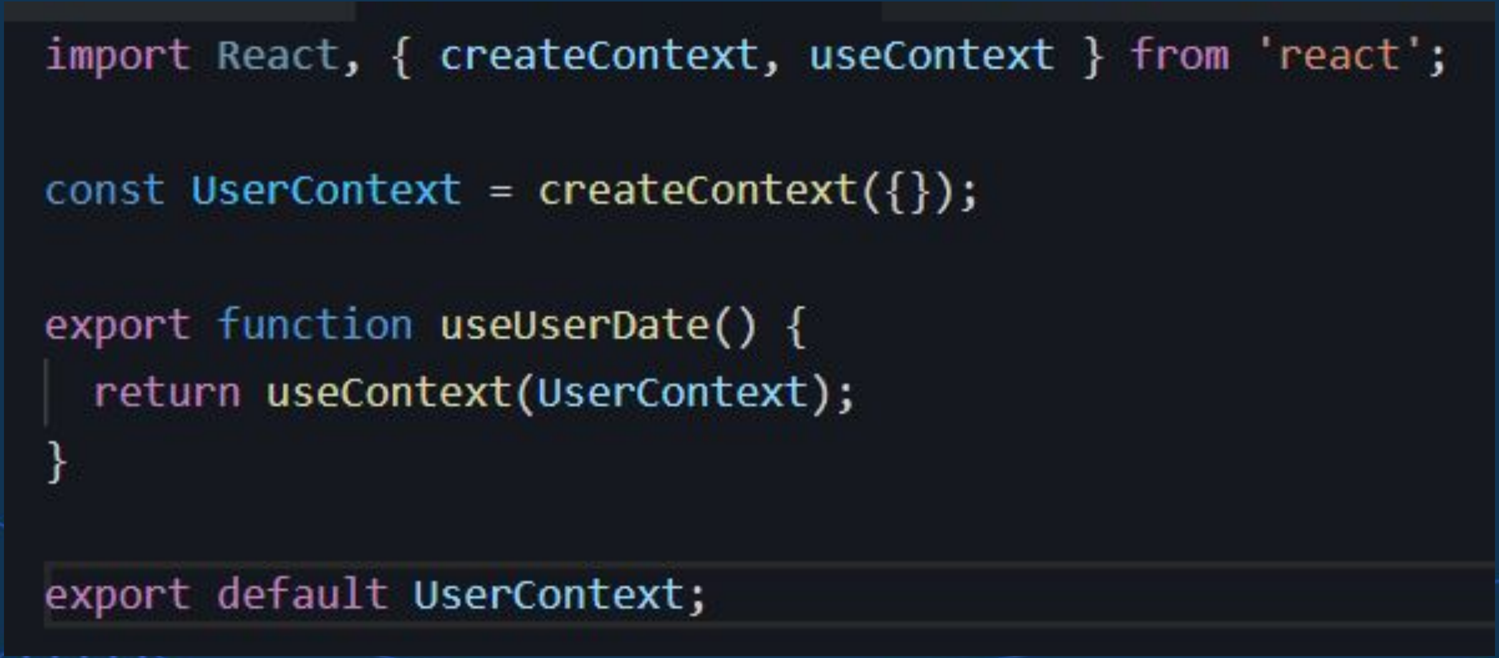
CoGrammar

Consuming Context

- Components can consume context using the useContext hook.
- This enables them to access the context value without being directly nested within the provider.

CoGrammar

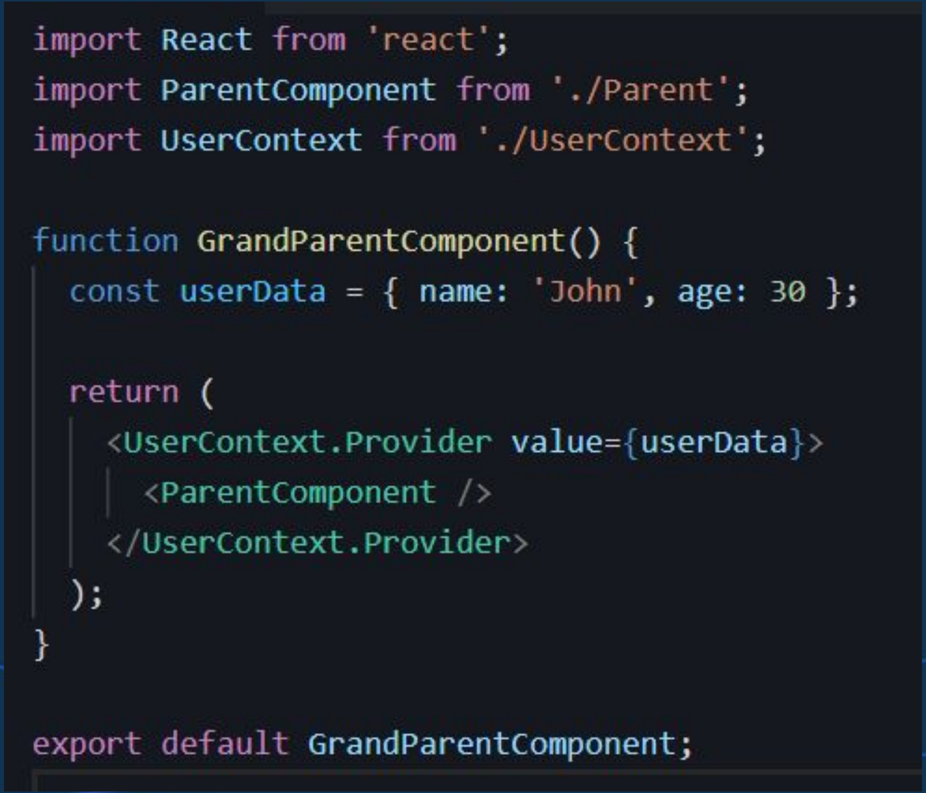
Example: Using React Context API



```
import React, { createContext, useContext } from 'react';

const UserContext = createContext({});

export function useUserDate() {
  return useContext(UserContext);
}

export default UserContext;
```

CoGrammar

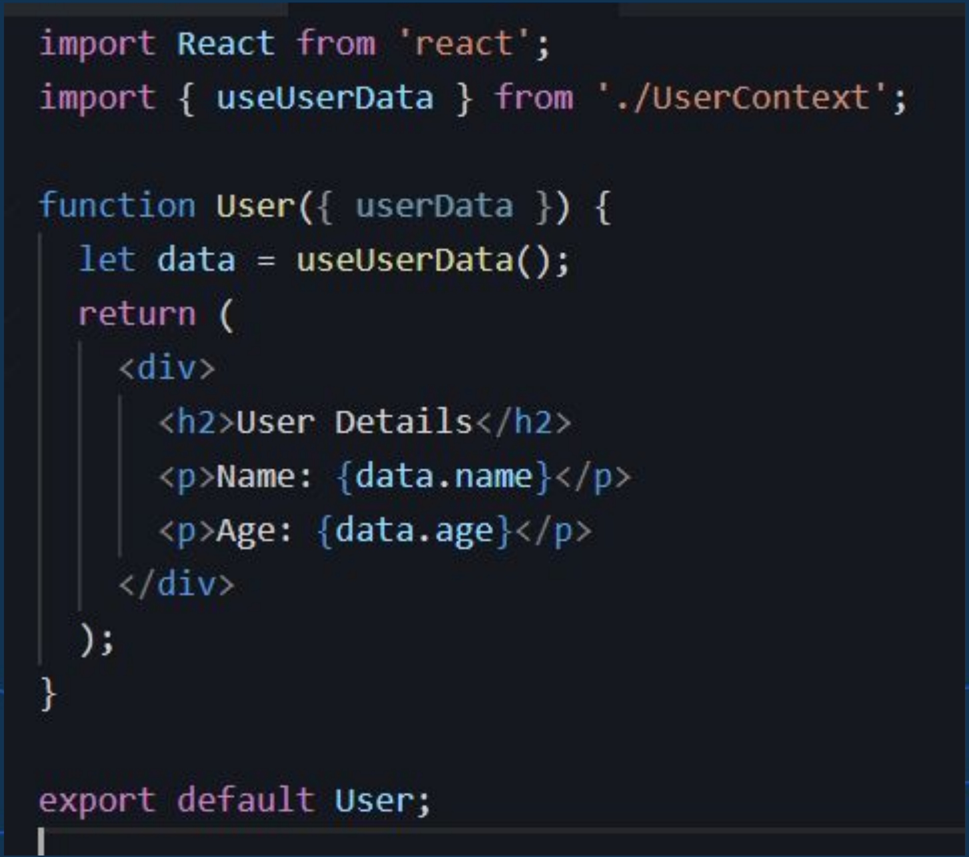
Example: Using React Context API



```jsx
import React from 'react';
import ParentComponent from './Parent';
import UserContext from './UserContext';

function GrandParentComponent() {
  const userData = { name: 'John', age: 30 };

  return (
    <UserContext.Provider value={userData}>
      <ParentComponent />
    </UserContext.Provider>
  );
}

export default GrandParentComponent;
```

CoGrammar

Example: Using React Context API



```
import React from 'react';
import { useUserData } from './UserContext';

function User({ userData }) {
  let data = useUserData();
  return (
    <div>
      <h2>User Details</h2>
      <p>Name: {data.name}</p>
      <p>Age: {data.age}</p>
    </div>
  );
}

export default User;
```

CoGrammar

How do we create a state variable in a react.js component that can store boolean values?

A. useName(true)

B. useVariable(false)

C. useState(false)

CoGrammar

To change the value of a state variable, you directly use the assignment operator (=).

A. True

B. False

CoGrammar

Summary

- **Key Takeaways:**
  - In this session, we delved into the essentials of state management in React.js.
  - We started by understanding how state works in React, leveraging the useState hook to manage local state within components.
  - We explored the challenges of prop drilling in passing data across component trees and introduced the React Context API as a solution for global state management.
- **Homework:** Try to create a state variable that determines a single component’s theme (dark or light).

CoGrammar

Questions and Answers

CoGrammar

Thank you for attending

SKILLS FOR LIFE
SKILLS BOOTCAMPS

Department for Education

CoGrammar