**Report on methods in React application**

In a React application, methods are an integral part of class components and functional components (using hooks). They allow you to define and encapsulate reusable code for handling various aspects of component behavior. Here are some common methods used in React and their corresponding code snippets:

1. **Lifecycle Methods:**

**componentDidMount():** Invoked immediately after the component is mounted (rendered to the DOM).

componentDidMount() {

// Perform initialization tasks

}

**componentDidUpdate(prevProps, prevState):** Called after the component is updated. It receives the previous props and state as arguments.

componentDidUpdate(prevProps, prevState) {

// Perform tasks based on prop or state changes

}

**componentWillUnmount():** Called right before the component is unmounted and destroyed.

componentWillUnmount() {

// Clean up resources or event listeners

}

1. **Event Handler Methods:**

**onClick:**

handleClick() {

// Handle the click event

}

render() {

return <button onClick={this.handleClick}>Click me</button>;

}

1. **Custom Methods:**

Example of a custom method to perform a specific action:

customMethod() {

// Perform a specific action

}

render() {

return <div>{this.customMethod()}</div>;

}

1. **State Manipulation Methods:**

* Using setState to update state values:

handleClick() {

this.setState({ count: this.state.count + 1 });

}

* Using useState hook to manage state in functional components:

import React, { useState } from 'react';

function Counter() {

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

const handleClick = () => {

setCount(count + 1);

};

return (

<div>

<p>Count: {count}</p>

<button onClick={handleClick}>Increment</button>

</div>

);

}

1. **Utility Methods:**

Example of a utility method for data manipulation:

function formatName(firstName, lastName) {

return `${lastName}, ${firstName}`;

}

render() {

const fullName = formatName('John', 'Doe');

return <div>{fullName}</div>;

}

These are just a few examples of the many methods you can use in React. The specific methods you'll need depend on your application's requirements and the lifecycle of your components. Remember to consider the component's class or functional component syntax when using these methods.