


How Does Conditional Rendering Work in React Components?

 freecodecamp.org/learn/front-end-development-libraries-v9/lecture-working-with-data-in-react/how-does-conditional-rendering-work-in-react-components



Conditional rendering in React allows you to create dynamic user interfaces. It is used to show different content based on certain conditions or states within your application.

The most common approaches of using conditional rendering includes using `if` statements, the ternary (`?:`) operator, and logical AND (`&&`) operator.

The simplest form of conditional rendering uses an `if` statement. Here's an example:

```
function Greeting({ isLoggedIn }) {  
  if (isLoggedIn) {  
    return <h1>Welcome back!</h1>;  
  }  
  return <h1>Please sign in</h1>;  
}
```

In this example, the `Greeting` component checks the `isLoggedIn` prop. If it's `true`, it returns a welcome message, otherwise, it prompts the user to sign in.

Here is an example using the `Greeting` component inside of the parent `App` component:

```
function App() {
  return (
    <div className="App">
      <Greeting isLoggedIn={false} />
    </div>
  );
}
```

For simpler conditions, the ternary operator (`?:`) is often used directly within JSX. It allows for inline conditional rendering, which can make your code more concise:

```
function Greeting({ isLoggedIn }) {
  return <h1>{isLoggedIn ? "Welcome back!" : "Please sign in."}</h1>;
}
```

This code achieves the same result as the previous example but in a more compact form. The ternary operator checks `isLoggedIn` and renders the appropriate message.

Another common pattern for conditional rendering is using the logical AND (`&&`) operator. This is particularly useful when you want to render something, or nothing, based on a condition:

```
function Notification({ message }) {
  return (
    <div>
      {message && <p>{message}</p>}
    </div>
  );
}
```

In this example, the paragraph element with the message is only rendered if the `message` prop is truthy. If `message` is falsy - meaning it is an empty string, `null`, or `undefined`, nothing is rendered to the screen.

By mastering these techniques of conditional rendering, you can build more interactive and user-friendly applications that adapt to changing data and user interactions.

Questions

Which of the following is NOT a common method for conditional rendering in React?

Using `if` statements.

Using the ternary operator.

Using the logical AND (`&&`) operator

Using the `switch` statement.

What will be rendered by the following code if `message` is an empty string?

```
function Alert({ message }) {  
  return (  
    <div>  
      {message && <p>{message}</p>}  
    </div>  
  );  
}
```

`<div><p></p></div>`

`<div></div>`

`<div>false</div>`

This will throw an error.

In the following code, what type of conditional rendering is being used?

```
function Greeting({ name }) {  
  return <h1>{name ? `Hello, ${name}!` : 'Hello, Guest!'}  
</h1>;  
}
```

`if` statement.

`switch` statement.

Ternary operator.

Logical AND (`&&`) operator.