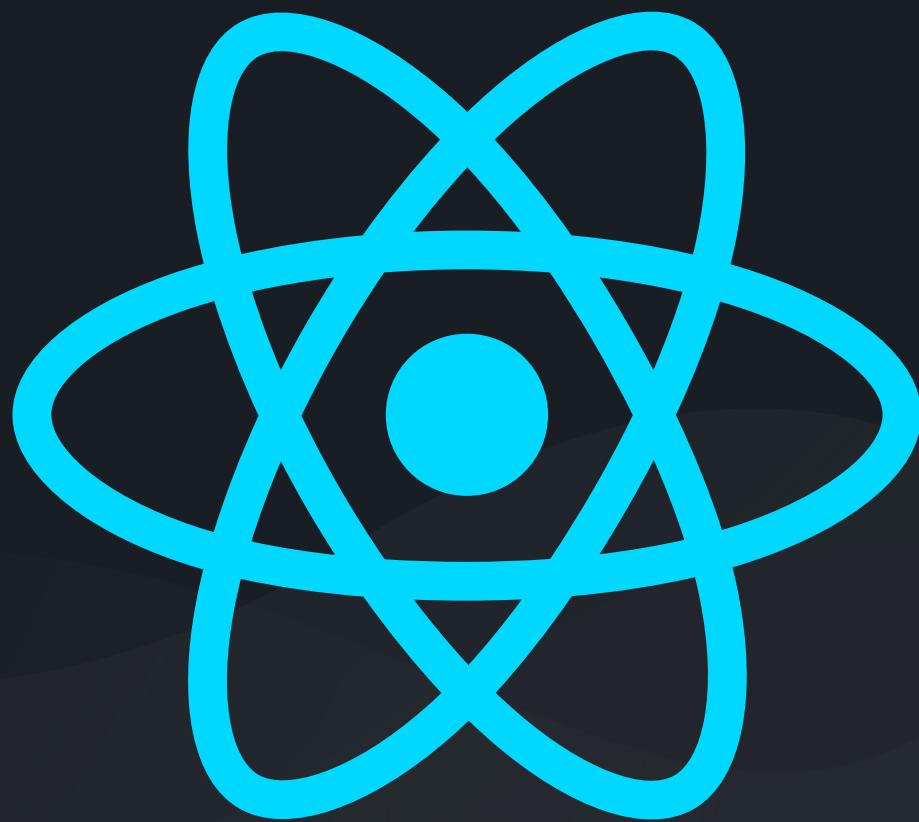


Conditional Rendering

React JS | In Detail



Mallikarjun | @CodeBustler



Conditional Rendering

In React, conditional rendering is a **powerful technique** that allows you to display **different components or content** based on certain conditions.

React provides **multiple ways** to achieve conditional rendering, depending on the complexity and requirements of your application.

- If-Else Statements
- Ternary Operators
- Logical && Operator
- Conditional Render with Components



Using **If-else** Statements

You can use JS if-else statements inside the **render()** method of a component to conditionally render different content.

```
import React from "react";

function MyComponent({ isLoggedIn }) {
  if (isLoggedIn) {
    return <div>Welcome, user!</div>;
  } else {
    return <div>Please log in.</div>;
  }
}

// Welcome, user!
```

Note

In JSX (JavaScript XML), you **cannot** directly **use statements** such as **if**, **for**, or **while** **within the curly braces** (**{ }**) as you would in regular JavaScript.

This is because JSX is a syntax extension of JavaScript that aims to provide a **declarative way of describing** the structure and logic of UI components.



@CodeBustler





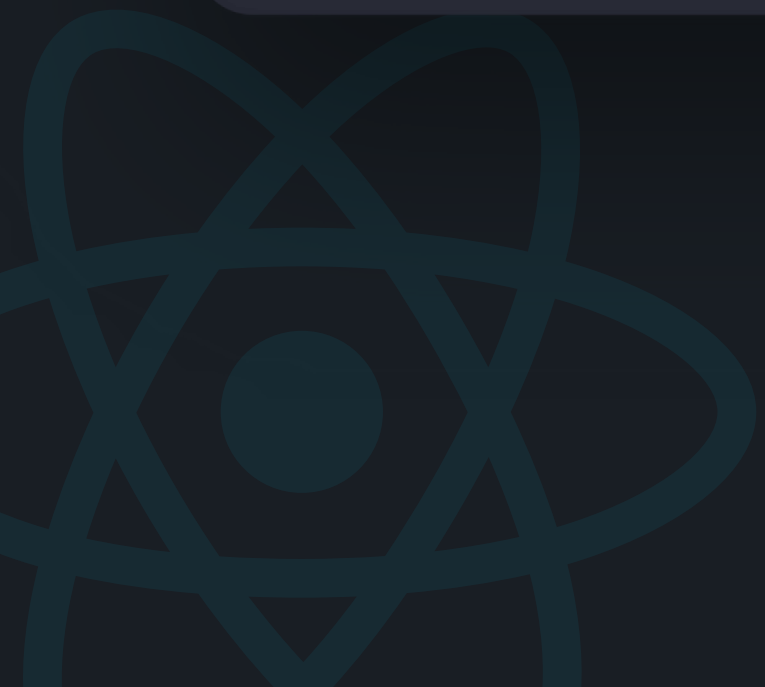
Using Ternary Operator

You can use the ternary operator (`? :`) **inside curly braces** `{ }` to conditionally render content within JSX.

```
import React from "react";

function MyComponent({ isLoggedIn }) {
  return (
    isLoggedIn ? <div>Welcome, user!</div>
              : <div>Please log in.</div>;
  )

  // Welcome, user!
```





Using **Logical &&** Oprt

When the condition is true, the component is rendered; otherwise, it is **skipped**.

This is particularly useful when you want to render **nothing or a default** state when a condition is **false**.

@CodeBustler

```
import React from "react";

function MyComponent({ isLoggedIn }) {
  return <div>
    {isLoggedIn && <div>Hello!</div>}
  </div>;
}

// Hello!
```



Conditional Rendering with Components

In more complex scenarios, you may want to render different components based on conditions.

```
import React from "react";

function UserData() {
  return <h1>Welcome to the User Dashboard!</h1>;
}

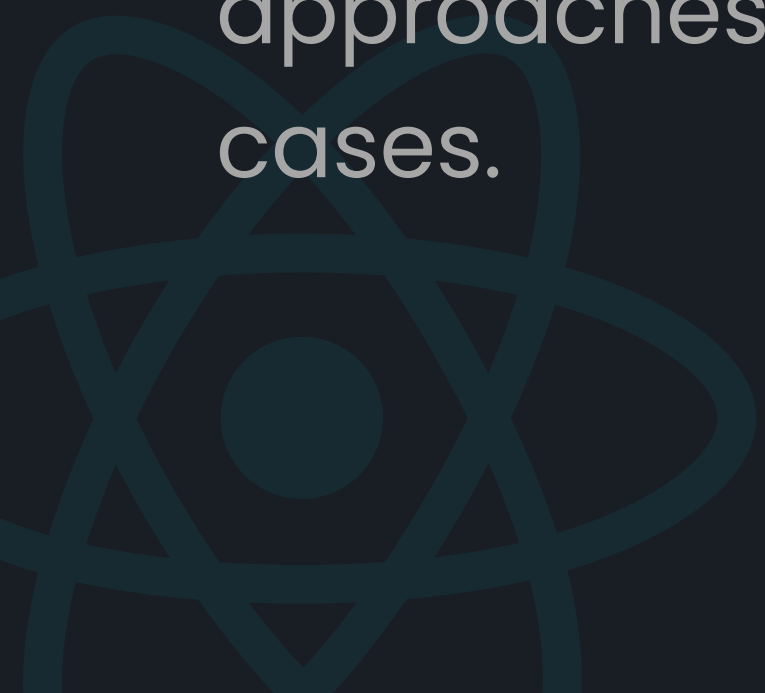
function AdminData() {
  return <h1>Welcome to the Admin Dashboard!</h1>;
}

function App({ userRole }) {
  return <div>
    {userRole === "user" ? <UserData /> : <AdminData />}
  </div>;
}
```

These are just a few examples of conditional rendering in React.

Depending on your requirements, you can combine these techniques, use switch statements, or leverage more advanced approaches such as rendering based on **state** or **iterating through arrays**.

React's flexible nature allows for various conditional rendering approaches to suit different use cases.



Other Posts



To learn **HTML, CSS, JavaScript & React JS** or Front-End **Projects**, Visit my page all concepts & Projects codes are available 🔥

@CodeBustler



@CodeBustler

Mallikarjun | Web Developer

Follow For More !

And i need dopamine ⚡

