

Conditional rendering in ReactJS (Functional Components) means showing or hiding UI elements based on a condition (state, props, variables, etc.).

Below are all common patterns, from basic to practical real-world usage.

Using if statement (outside JSX)

Best when logic is complex.

```
function UserStatus({ isLoggedIn }) {
  if (isLoggedIn) {
    return <h2>Welcome Back!</h2>;
  }
  return <h2>Please Login</h2>;
}
```

- ✓ Clean
 - ✓ Easy to debug
 - ✗ Can't be used directly inside JSX
-

Using Ternary Operator (? :)

Most commonly used for two conditions.

```
function Profile({ isAdmin }) {
  return (
    <div>
      {isAdmin ? <h3>Admin Panel</h3> : <h3>User Dashboard</h3>}
    </div>
  );
}
```

- ✓ Short
 - ✓ Readable
 - ✗ Avoid nesting ternaries
-

Using Logical AND (&&)

Used when you want to render only if condition is true.

```
function Notification({ count }) {
  return (
    <div>
      {count > 0 && <p>You have {count} new messages</p>}
    </div>
  );
}
```

```
});  
}
```

⚠️ Important

```
0 && <p>Hello</p> // renders 0 (bug)
```

- ✓ Use with boolean conditions
-

Using else if via variable

Clean approach for multiple conditions.

```
function Status({ status }) {  
  let message;  
  
  if (status === "loading") {  
    message = "Loading...";  
  } else if (status === "success") {  
    message = "Data Loaded";  
  } else {  
    message = "Error Occurred";  
  }  
  
  return <h2>{message}</h2>;  
}
```

- ✓ Very readable
 - ✓ Best for 3+ conditions
-

Conditional Rendering with useState

Common in forms, modals, toggles.

```
import { useState } from "react";  
  
function Login() {  
  const [isLoggedIn, setIsLoggedIn] = useState(false);  
  
  return (  
    <div>  
      {isLoggedIn ? <h2>Dashboard</h2> : <h2>Login Page</h2>}  
  
      <button onClick={() => setIsLoggedIn(!isLoggedIn)}>  
        Toggle  
      </button>  
    </div>  
  );  
}
```

```
};  
}
```

Show / Hide Components (Modal Example)

```
function ModalExample() {  
  const [showModal, setShowModal] = useState(false);  
  
  return (  
    <>  
      <button onClick={() => setShowModal(true)}>Open</button>  
  
      {showModal && (  
        <div className="modal">  
          <p>Modal Content</p>  
          <button onClick={() => setShowModal(false)}>Close</button>  
        </div>  
      )}  
    </>  
  );  
}
```

Conditional Rendering using Props

```
function Button({ disabled }) {  
  return (  
    <button disabled={disabled}>  
      {disabled ? "Disabled" : "Click Me"}  
    </button>  
  );  
}
```

Render Nothing (null)

React allows returning null.

```
function Warning({ show }) {  
  if (!show) return null;  
  return <p>Warning Message</p>;  
}
```

- ✓ Component exists
 - ✓ UI hidden
-

Conditional CSS (Alternative Approach)

```
function Alert({ error }) {
  return (
    <div className={error ? "error" : "success"}>
      Message
    </div>
  );
}
```
