# 📘 React Component Rendering Techniques

### 🔹 Conditional Rendering in React

* Conditional rendering lets you **control what gets displayed** based on runtime conditions.
* You can use regular if, ternary operators, logical &&, or switch cases to determine output.
* JSX allows embedding these conditions inside component returns.

function Greeting({ isLoggedIn }) {

if (isLoggedIn) {

return <h1>Welcome back!</h1>;

}

return <h1>Please sign in.</h1>;

}

#### ✅ Common Patterns

{isValid && <span>✅ Valid entry</span>}

{status === "error" ? <ErrorMsg /> : <SuccessMsg />}

### 🔹 Element Variables in React

* You can assign JSX to **variables**, helping with cleaner and flexible rendering logic.

let button;

if (isAuthenticated) {

button = <LogoutButton />;

} else {

button = <LoginButton />;

}

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

* Element variables are especially useful when combining multiple rendering paths.

### 🔹 Preventing Components from Rendering

There are several ways to **skip rendering** components:

#### ❌ Return null

* If a component returns null, React renders nothing to the DOM.

function HiddenMessage({ isVisible }) {

if (!isVisible) return null;

return <p>You can see me now!</p>;

}

#### ❌ Conditional JSX

* You can use conditions directly in JSX to suppress component rendering.

{shouldShow ? <Banner /> : null}

#### ❌ Early Return

* Short-circuit component logic and exit early.

function Profile({ user }) {

if (!user) return null; // Skip rendering if user is missing

return <h2>{user.name}</h2>;

}