1. Explain Conditional Rendering in React

Conditional rendering in React means showing or hiding parts of the UI **based on conditions (if-else logic)**.

Just like in JavaScript, you can use:

- if statements
- ternary operators
- && (logical AND) for simple conditions

Example 1: Using if-else

```
if (isLoggedIn) {
  return <UserPage />;
} else {
  return <GuestPage />;
}
```

Example 2: Ternary Operator

```
{isLoggedIn ? <UserPage /> : <GuestPage />}
```

Example 3: Logical AND (&&)

```
{error && <p style={{ color: "red" }}>Something went wrong!</p>}
```

2. Define Element Variables

Element variables are used to **store JSX elements** in variables, allowing you to build dynamic UIs more clearly.

Example:

```
let message;
if (isLoggedIn) {
  message = <h2>Welcome back!</h2>;
} else {
  message = <h2>Please sign in.</h2>;
}

return (
  <div>
```

```
    {message}
  </div>
);
```

- Use element variables to keep your return() clean and readable.

3. Prevent Components from Rendering

Sometimes, you want to **conditionally skip rendering a component**.

Option 1: Return null

If a component should render **nothing**, you can return null.

```
function WarningBanner({ show }) {
  if (!show) {
    return null; // prevents rendering
  }
  return <div className="warning">Warning!</div>;
}
```

Option 2: Skip rendering in JSX

```
{shouldRender && <SomeComponent />}
```