7. Conditional Rendering and Lists

Last updated by | Subramanya Dixit | May 5, 2025 at 7:59 PM GMT+5:30

📘 Why Conditional Rendering Is Important

It allows React components to dynamically change what's shown to the user, enabling better control over UI behavior based on app state or props.

🧠 Key Concepts

✅ Conditional Rendering

Use logical or ternary operators to render components or elements based on conditions.

```
function Greeting({ isLoggedIn }) {
  return (
    <div>
      {isLoggedIn ? <p>Welcome back!</p> : <p>Please log in.</p>}
    </div>
  );
}
```

✅ Short-circuit Evaluation

```
{isAdmin && <button>Delete User</button>}
```

✅ if-else Rendering

Use conditional statements inside functions.

```
function Message({ type }) {
  if (type === 'error') return <p>Error occurred!</p>;
  if (type === 'success') return <p>Operation successful!</p>;
  return null;
}
```

📋 Rendering Lists

Use `map()` to render multiple components from an array.

```
const users = ['Alice', 'Bob', 'Charlie'];

function UserList() {
  return (
    <ul>
      {users.map((user, index) => (
        <li key={index}>{user}</li>
      ))}
    </ul>
  );
}
```

✅ Key Prop

Use a unique `key` prop to help React identify which items changed.

```
<li key={user.id}>{user.name}</li>
```

📊 Visual Overview

```
flowchart TD
  A[App State Changes] --> B[Evaluate Condition]
  B --> C{True or False}
  C -->|True| D[Render Component A]
  C -->|False| E[Render Component B]
```

💡 Guidelines

- Always include a `key` when rendering lists.

- Use ternary operators for inline logic.

- Avoid deeply nested if-else in JSX.

- Render fallback UI for empty lists or error states.

📝 Practice Exercises

1. Show a different greeting message based on login state.

2. Render a list of items using `map()`.

3. Conditionally show a message when the list is empty.

4. Use a `key` prop correctly in a dynamic list.

❓ Quiz Questions

1. What does `isLoggedIn ? <A /> : <B />` represent?
a) useEffect hook
b) useState setter
✅ **c) Ternary conditional rendering**
d) JSX map function

2. What is the purpose of the `key` prop in lists?
a) For styling
b) For accessibility
✅ **c) To help React track changes**
d) To render conditionally

3. What happens if you forget to include a `key` prop in a list?

a) React shows an error

b) Nothing changes

✅ **c) React may inefficiently re-render items**

d) List items become read-only

---

4. What does `{isAdmin && <Button />}` do?

a) Always renders Button

✅ **b) Renders Button only if isAdmin is true**

c) Throws an error if isAdmin is false

d) Creates a loop