React: Lists, Attributes, and Components

Understanding how to render lists, use attributes, and build components is key to becoming confident in React.

List and Keys: How to Render a List of Data?

Rendering a list means displaying multiple similar UI elements from an array.

```
const fruits = ["Apple", "Banana", "Mango"];


const listItems = fruits.map((fruit, index) =>
  <li key={index}>{fruit}</li>
);
```

We use `.map()` to loop through the array and return JSX for each item.

Importance of Keys

What is the purpose of the `key` in lists?

- React uses `key` to identify which items changed, were added, or removed.
- Helps improve performance during re-rendering.

> **Never use array index as key in real-world apps where list items can change order.**

Define Array and Object and Render as a List or Table

Array Example:

```
const cities = ["Mumbai", "Delhi", "Kolkata"];
const cityList = cities.map((city, i) => <li key={i}>{city}</li>);
```

Object Array Example as Table:

```
const users = [
  { name: "Raj", age: 25 },
  { name: "Simran", age: 23 }
];

const userRows = users.map((user, index) => (
  <tr key={index}>
    <td>{user.name}</td>
    <td>{user.age}</td>
  </tr>
));

<table>
  <thead><tr><th>Name</th><th>Age</th></tr></thead>
  <tbody>{userRows}</tbody>
</table>
```

Comments in JSX

JSX uses JavaScript-style comments, but inside curly braces:

```
{/* This is a JSX comment */}
```

Attributes in JSX and Rules

- Use **camelCase**: className instead of class, htmlFor instead of for.
- Values must be in quotes or curly braces.

```
<img src="logo.png" alt="Company Logo" width={100} />
```

Creation of Custom Attribute

JSX doesn’t support custom attributes directly in the DOM.

But in React components, you can **pass custom props**:

```
<MyButton data-id="123" label="Click Me" />
```

In MyButton:

```
function MyButton(props) {
  return <button>{props.label}</button>;
}
```

Introduction to Component and Render Function as Component

What is a Component?

A **component** is a reusable piece of UI.

Function Component:

```
function Welcome() {
  return <h1>Welcome to React!</h1>;
}
```

Render Function:

React renders the component:

```
ReactDOM.render(<Welcome />, document.getElementById('root'));
```

Rules of Components

- Component name must start with a **capital letter**.
- Must return a single JSX element.
- File name and component name should match (for clarity).
- Props help make components **dynamic**.

Advantages of Component Rendering

- **Reusability:** Use the same component in multiple places.
- **Readability:** Clean code with separate logic.
- **Separation of Concerns:** Each component handles one task.
- **Testing:** Easy to test components individually.

Know More: FAQ

Q1. Why are keys important in React lists?

- They help React efficiently update the UI without reloading everything.

Q2. What if we forget to give a key?

- React will warn you and performance might degrade.

Q3. Can I use JavaScript if-else in JSX?

- Use ternary expressions or functions. You can't use regular `if` directly inside JSX.

Q4. Are components like functions?

- Yes, they are JavaScript functions that return JSX.

Q5. Can components have other components inside them?

- Yes, this is called **component composition**.