full Series 1: React Basics & Fundamentals
with **interview Q&A**, **simple code examples**, and **real-world analogies** to help you deeply understand and remember each concept.

---

1. What is React?

💬 **Interview Answer:**
React is a JavaScript library for building user interfaces, created by Meta. It uses a component-based architecture and efficiently updates the UI using a virtual DOM.

📌 **Code Example:**

```
function App() {
  return <h1>Hello React!</h1>;
}
```

🔁 **Analogy:**
React is like LEGO — each component is a reusable block to build complex structures (UIs).

---

2. What is JSX?

💬 **Interview Answer:**
JSX stands for JavaScript XML. It allows us to write HTML-like code inside JavaScript. JSX gets compiled into `React.createElement()` calls.

📌 **Code Example:**

```
const element = <h1>Hello, JSX!</h1>;
```

🔁 **Analogy:**
JSX is like a readable recipe card — looks like HTML but works behind the scenes with JavaScript.

---

3. What are Components in React?

💬 **Interview Answer:**
Components are the building blocks of React. Each component returns JSX and can be reused. Functional components are now the standard.

📌 **Code Example:**

```
function Welcome() {
  return <h2>Welcome to React</h2>;
}
```

🔁 **Analogy:**
Components are like small gadgets — each with its own function, assembled into one machine (the app).

---

4. What are Props?

💬 **Interview Answer:**
Props are read-only inputs passed from a parent to child component. They help components be dynamic and reusable.

📌 **Code Example:**

```
function Greet(props) {
  return <p>Hello, {props.name}</p>;
}
```

🔁 **Analogy:**
Props are like ingredients passed to a chef — the chef (component) uses them to create the final dish (UI).

---

5. What is State?

💬 **Interview Answer:**
State is local, mutable data managed within a component. It's used to control dynamic behavior and re-render UI on changes.

📌 **Code Example:**

```
const [count, setCount] = useState(0);
```

🔁 **Analogy:**
State is like a live scoreboard — it updates the display automatically whenever the score changes.

---

6. What is Prop Drilling?

💬 **Interview Answer:**
Prop Drilling is when data is passed through many nested components, even if only the child at the bottom needs it. It leads to cluttered code and can be solved using Context.

📌 **Code Example:**

```
<Parent>
  <Child>
    <GrandChild user={user} />
  </Child>
</Parent>
```

🔁 **Analogy:**
It's like passing a note through many people just to reach one person — inefficient and error-prone.

---

7. What is Conditional Rendering?

💬 **Interview Answer:**
Conditional rendering allows React to decide what to display based on conditions, using JavaScript logic like `if`, `&&`, or ternary operators.

📌 **Code Example:**

```
{isLoggedIn ? <Dashboard /> : <Login />}
```

🔁 **Analogy:**
Like traffic lights — green if conditions are right, red if not.

---

8. How does List Rendering work in React?

💬 **Interview Answer:**
React uses `.map()` to loop through arrays and render a list of elements. Each child must have a unique `key` prop to optimize performance.

📌 **Code Example:**

```
{items.map(item => <li key={item.id}>{item.name}</li>)}
```

🔁 **Analogy:**
It’s like printing name tags — each one must have a unique ID to avoid confusion.

---

9. How do you handle Events in React?

💬 **Interview Answer:**
React uses synthetic events — cross-browser wrappers around native events. Use camelCase and pass functions for handlers.

📌 **Code Example:**

```
<button onClick={() => alert('Clicked!')}>Click</button>
```

🔁 **Analogy:**
Synthetic events are like a universal remote — the same control works on all devices (browsers).

---

10. How do you handle Forms in React?

💬 **Interview Answer:**
React handles forms using controlled components, where form inputs are tied to state. This allows you to control and validate form data.

📌 **Code Example:**

```
const [name, setName] = useState('');
<input value={name} onChange={(e) => setName(e.target.value)} />
```

🔁 **Analogy:**
It’s like having real-time feedback while filling a form — you see the change immediately.

---

11. What is Lifting State Up?

💬 **Interview Answer:**
Lifting state up means moving shared state to the nearest common ancestor of components that need access to it. This allows siblings to share data.

📌 **Code Example:**

```
<Parent>
  <ChildA setData={setData} />
  <ChildB data={data} />
</Parent>
```

🔁 **Analogy:**
Like putting shared snacks on a central table so both kids can reach them.

---

12. What is One-Way Data Flow?

💬 **Interview Answer:**
In React, data flows from parent to child components (top-down). Child components cannot directly modify parent state, maintaining predictable state flow.

📌 **Code Example:**

```
<Child name={parentName} />
```

🔁 **Analogy:**
Like water flowing downhill — it always flows in one direction.
