



# React Full Notes

---

## ♦ 1. What is React?

- **React** is a **JavaScript library** for building **user interfaces (UIs)**.
  - Developed by **Facebook**, it's used to build **single-page applications (SPAs)**.
  - Component-based and declarative.
- 

## ♦ 2. Key Features of React

- **JSX** (JavaScript XML)
  - **Components**
  - **Virtual DOM**
  - **Unidirectional Data Flow**
  - **Hooks**
- 

## ♦ 3. Basic Setup

Install using:

bash

CopyEdit

```
npx create-react-app my-app
```

```
cd my-app
```

```
npm start
```

---

## ♦ 4. JSX (JavaScript XML)

Allows writing HTML-like syntax in JavaScript.

```
jsx
CopyEdit
const element = <h1>Hello, React!</h1>;
```

---

## ♦ 5. Components

### ♦ Functional Component

```
jsx
CopyEdit
function Welcome() {
  return <h1>Welcome!</h1>;
}
```

### ♦ Class Component

```
jsx
CopyEdit
class Welcome extends React.Component {
  render() {
    return <h1>Welcome!</h1>;
  }
}
```

---

## ♦ 6. Props (Properties)

Used to pass data to components.

```
jsx
CopyEdit
function Greet(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

---

## ♦ 7. State

Used to manage internal data in a component.

jsx

CopyEdit

```
import { useState } from 'react';
```

```
function Counter() {  
  const [count, setCount] = useState(0);  
  return (  
    <div>  
      <p>{count}</p>  
      <button onClick={() => setCount(count + 1)}>Add</button>  
    </div>  
  );  
}
```

---

## ♦ 8. Event Handling

jsx

CopyEdit

```
function ClickMe() {  
  function handleClick() {  
    alert("Clicked!");  
  }  
  
  return <button onClick={handleClick}>Click</button>;  
}
```

---

## ♦ 9. Conditional Rendering

jsx

CopyEdit

```
{isLoggedIn ? <h1>Welcome</h1> : <h1>Please Login</h1>}
```

---

## ♦ 10. Lists & Keys

jsx

CopyEdit

```
const names = ['Alice', 'Bob', 'Charlie'];
const listItems = names.map((name, index) =>
  <li key={index}>{name}</li>
);
```

---

## ◆ 11. useEffect Hook

Runs side effects like API calls.

jsx

CopyEdit

```
useEffect(() => {
  console.log("Component mounted");
}, []);
```

---

## ◆ 12. useRef Hook

Used to reference DOM elements or hold values that persist.

jsx

CopyEdit

```
const inputRef = useRef(null);
<input ref={inputRef} />
```

---

## ◆ 13. useContext Hook

Provides global state without prop drilling.

jsx

CopyEdit

```
const ThemeContext = createContext();

function App() {
  return (
    <ThemeContext.Provider value="dark">
      <Component />
    </ThemeContext.Provider>
  );
}
```

```
    </ThemeContext.Provider>
  );
}
```

---

## ◆ 14. Forms

jsx

CopyEdit

```
function Form() {
  const [name, setName] = useState('');
  return (
    <form>
      <input
        type="text"
        value={name}
        onChange={(e) => setName(e.target.value)}
      />
    </form>
  );
}
```

---

## ◆ 15. React Router (for navigation)

bash

CopyEdit

```
npm install react-router-dom
```

jsx

CopyEdit

```
import { BrowserRouter, Routes, Route } from 'react-router-dom';
```

```
<BrowserRouter>
  <Routes>
    <Route path="/" element={<Home />} />
    <Route path="/about" element={<About />} />
  </Routes>
</BrowserRouter>
```

---

## ♦ 16. Redux (For global state management)

- Actions
- Reducers
- Store
- Dispatch

---

## ♦ 17. React Best Practices

- Break UI into small reusable components
- Use keys in lists
- Avoid prop drilling (use Context)
- Keep components pure
- Use hooks for side effects