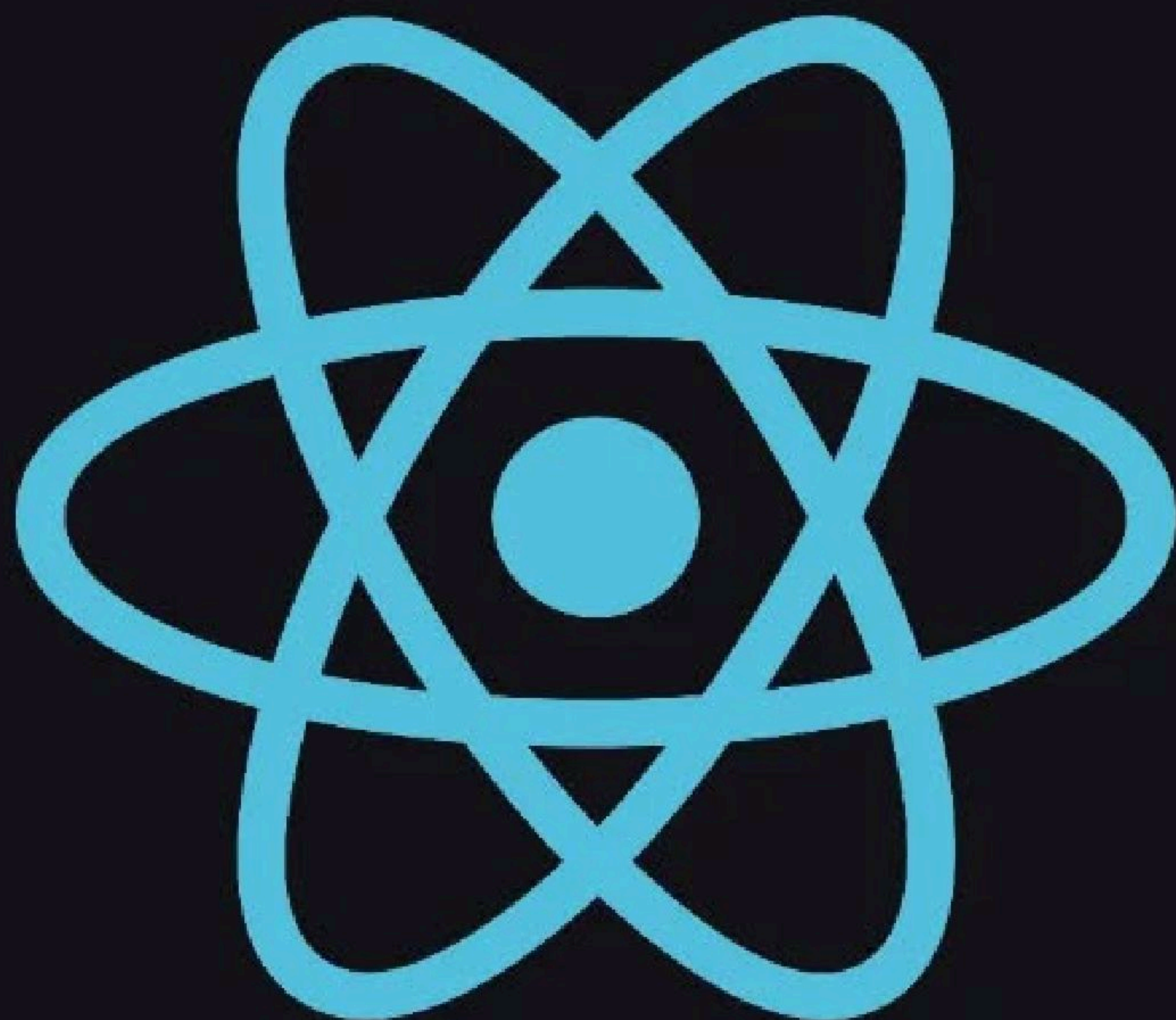




Save this post

# ReactJS

## CheatSheet



@imperio\_coders





# 1. Setup & Basics

```
// Create App
npx create-react-app my-app

// Functional Component
function Welcome() {
  return <h1>Hello React</h1>;
}

// JSX Rules
const name = "Nikhil";
return <h2>Hello, {name}</h2>;
```

# 2. Props & State

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

// State
import { useState } from 'react';

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

Comment “React” for Full PDF



### 3. useEffect Hook

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

useEffect(() => {
  console.log("Component mounted");

  return () => {
    console.log("Cleanup");
  };
}, []);
```

### 4. Event Handling

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

### 5. Conditional Rendering

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

// && operator
{loading && <p>Loading...</p>}
```

Join telegram for free notes ( Link in bio )



## 6. Lists & Keys

```
const items = ["A", "B", "C"];
return (
  <ul>
    {items.map((item, i) => (
      <li key={i}>{item}</li>
    ))}
  </ul>
);
```

## 7. Forms

```
function Form() {
  const [input, setInput] = useState("");

  return (
    <input
      value={input}
      onChange={(e) => setInput(e.target.value)}
    />
  );
}
```

Comment “**React**” for Full PDF



## 8. useContext

```
const ThemeContext = createContext();

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

function Child() {
  const theme = useContext(ThemeContext);
  return <div>{theme}</div>;
}
```

## 9. useReducer

```
const reducer = (state, action) => {
  switch (action.type) {
    case "INC": return { count: state.count + 1 };
    default: return state;
  }
};

const [state, dispatch] = useReducer(reducer, { count: 0 });
dispatch({ type: "INC" });
```



## 10. Custom Hooks

```
function useToggle(initial) {  
  const [state, setState] = useState(initial);  
  const toggle = () => setState(!state);  
  return [state, toggle];  
}  
  
// Usage  
const [show, toggleShow] = useToggle(false);
```

## 11. React Router (v6)

```
import { BrowserRouter, Routes, Route } from 'react-router-dom';  
  
<BrowserRouter>  
  <Routes>  
    <Route path="/" element={<Home />} />  
    <Route path="/about" element={<About />} />  
  </Routes>  
</BrowserRouter>
```

## 12. Memoization

```
const expensive = useMemo(() => compute(data), [data]);  
const stableFunc = useCallback(() => doSomething(), []);
```



## 13. Lifting State Up

```
// Parent Component
function App() {
  const [data, setData] = useState("");
  return <Child update={setData} />;
}

// Child Component
function Child({ update }) {
  return <button onClick={() => update("New Data")}>Send</button>;
}
```

## 14. Refs & useRef

```
const inputRef = useRef();

<input ref={inputRef} />
<button onClick={() => inputRef.current.focus()}>Focus</button>
```

## 15. Component Optimization

```
import React, { memo } from "react";

const MyComp = memo(function MyComp({ name }) {
  return <p>{name}</p>;
});
```



**Save It & Follow For More !**



**Join our Whatsapp Community  
for free Resources & More!**

**Link in Bio !**