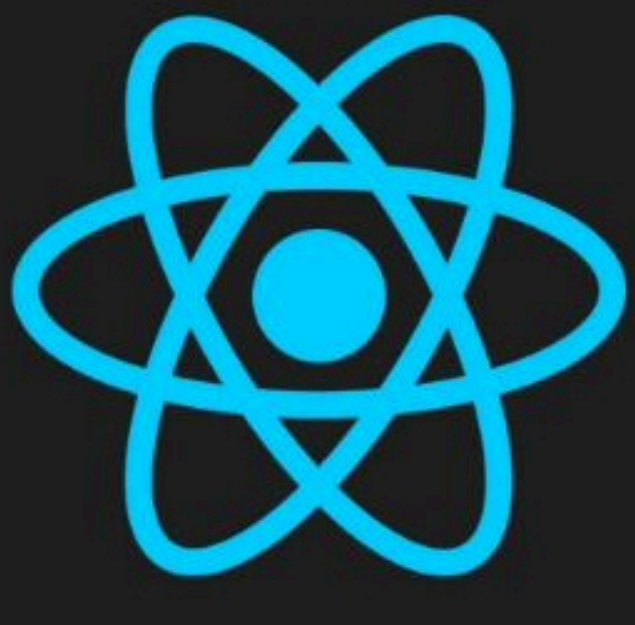
React JS CheatSheet

What is React js ?

- React is a powerful JavaScript library used for building interactive UIs and single-page applications.

Key Features of React

- **Component-Based:** Build encapsulated components that manage their own state.
- **Virtual DOM:** Faster UI updates by efficiently managing DOM changes.
- **JSX:** Write HTML in JavaScript with JSX syntax for cleaner code.
- **Unidirectional Data Flow:** Data flows downwards, ensuring stable data management.



Components

- Functional Component

```
function Greeting() {
  return <h1>Hello, User!</h1>;
}
```

- Class Component

```
class Greeting extends React.Component {
  render() {
    return <h1>Hello, User!</h1>;
  }
}
```

Basics

- **Create React App:**

```
npx create-react-app my-app
cd my-app
npm start
```

- **JSX Syntax:**

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

Props (Properties)

- Pass data to child components.
- Immutable in child components.


```
<Greeting name="John" />
function Greeting(props) {
  return <h1>Hello, {props.name}</h1>;
}
```

State

- Manages dynamic data within a component.
- Use useState in functional components.


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

Lifecycle Methods

- **componentDidMount():** Called after the component mounts.
- **componentDidUpdate():** Called after updates.
- **componentWillUnmount():** Called before unmounting.

React Hooks

- **useState:** State management.
- **useEffect:** Side effects (e.g., API calls).
- **useContext:** Global state management.

- **Example (useEffect):**


```
useEffect(() => {
  // Fetch data or other side effects
}, []);
```

Conditional Rendering

- Render based on conditions.

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


Event Handling

- Handle user inputs like clicks.

```
<button onClick={handleClick}>Click Me</button>
```

Follow