React Basics – Interview Notes

1. What is React?

React is a **JavaScript library used to build user interfaces**, especially **single-page applications (SPAs)**.

It allows developers to create **reusable UI components** and efficiently update the UI when data changes.

Key Features

- Component-based architecture
- Virtual DOM for efficient updates
- Declarative UI
- Reusable components
- Large ecosystem

---

2. Components

Components are **independent, reusable pieces of UI**.

Instead of writing a large UI file, React breaks the interface into **small reusable components**.

Functional Component

```
function Welcome() {
 return <h1>Hello World</h1>;
}
```

Class Component (Older approach)

```
class Welcome extends React.Component {
 render() {
```

```
  return <h1>Hello World</h1>;
 }
}
```

Important Interview Point

Modern React applications mainly use **functional components with hooks**.

---

3. JSX

JSX stands for **JavaScript XML**.

It allows writing **HTML-like syntax inside JavaScript**.

Example

```
const element = <h1>Hello React</h1>;
```

Behind the Scenes

JSX gets converted to:

```
React.createElement("h1", null, "Hello React");
```

JSX Rules

- Must have **one parent element**
- Use **className instead of class**
- JavaScript expressions inside { }

Example:

```
const name = "Anupama";
<h1>Hello {name}</h1>
```

---

4. Props

Props are **inputs passed from parent to child components**.

They help components become **dynamic and reusable**.

Example

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

<Greeting name="Anupama" />
```

Important Points

- Props are **read-only**
- Passed **parent → child**

---

5. State

State represents **data that changes inside a component**.

When state updates, React **re-renders the component**.

Example

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

Update state:

```
setCount(count + 1);
```

---

6. Props vs State

Props	State
Passed from parent	Managed inside component

Read-only	Can be updated
Used for configuration	Used for dynamic data

---

7. Virtual DOM

React uses a **Virtual DOM**, which is a lightweight copy of the real DOM.

Process

1. React creates Virtual DOM
2. Compares with previous Virtual DOM
3. Finds differences
4. Updates only changed elements

This process improves **performance and efficiency**.

---

8. Keys in React

Keys help React **identify elements in a list**.

Example

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

Why Keys are Important

- Help React track list items
- Improve rendering performance

Quick React Basics Summary

React Basics

React → UI library
Components → reusable UI pieces
JSX → HTML inside JavaScript
Props → data from parent
State → component data
Virtual DOM → efficient updates
Keys → identify list elements