**1) What is React.js?**

🡪React JS is Java-script libray used to build user interfaces for websites and web application.

🡪Using react JS create a fast and interactive UI like button, form etc.

**React Different from Other :-**

🡪React JS focused in UI .Other framework focused in Full-featured.

🡪React is easy to fast and other framework can be complex.

🡪React JS performance is very fast with Virtual DOM & other is slow some use real DOM.

🡪React js can be used with other libraries. other framework More opinionated, requires full setup.

**2)** **Explain the core principles of React such as the virtual DOM and component-based architecture.**

**Core principles of React:**

**1.Virtual DOM**

React creates a **virtual copy** of the real DOM.

**2.Component-Based Architecture:-**

A button, a form, or even a full web page can be a component. Components make the code reusable and easy to maintain.

**3. One-Way Data Flow:**

React flows data **from parent to child components.thats why one way data flow.**

**4.JSX:**

**Jsx makes writing UI.**

**5.State & Props:**

Stores dynamic data inside a component and props used to pass data from one component to another.

**3) What are the advantages of using React.js in web development?**

🡪React.js, developed and maintained by Facebook, is one of the most popular JavaScript libraries for building user interfaces, especially single-page applications (SPAs).

**1. Component-Based Architecture**

* **Reusable Components**: React promotes building UI using independent, reusable components. This modularity makes development more manageable and facilitates code reuse.

**2. Virtual DOM for Performance**

* React uses a **virtual DOM** to optimize rendering. Instead of reloading the entire DOM with every change, React updates only the changed elements, improving performance and speed.

**3. Declarative UI**

* With React, developers describe **what** they want the UI to look like rather than **how** to change the UI over time. This results in more predictable and easier-to-debug code.

**4. JSX Syntax**

* React uses JSX (JavaScript XML), which allows writing HTML-like code directly within JavaScript. It improves readability and makes writing components more intuitive.

**5. Strong Backing and Continuous Updates**

* React is maintained by Facebook and widely used in their products (like Instagram), ensuring long-term support and improvements.

**JSX**

**1)What is JSX in React.js? Why is it used?**

🡪jsx is special syntax to in react that allows you to write html code inside javascript.

**Why:**

🡪 **Makes UI Code Easier to Write ,**You can write HTML-like syntax directly in JavaScript.

🡪 No need to use complex document. createElement() functions.

🡪 You can use variables, functions, and expressions inside JSX. With javascript

**2) How is JSX diffrenet from regular Javascript**

🡪JSX is looks like HTML inside Java-script .regular Java-script user pure JS.

🡪JSX used in react for UL building. Regular JS is used for logic and DOM.

🡪JSX is faster .regular js is slower manipulating DOM directly.

**3) Curly braces {}in JSX expressions**

🡪Curly braces {} are very important because they allow us to insert dynamic values inside JSX.

**Example:**

const name = "krisha";

<h1>Hello, {name}!</h1>

**Outputs:**

Hello, krisha!

**Components:**

**1)What are React Component**

Component a bulding a block of react application. Each component is like a UI piece that can have its own logic and behavior.

**Difference between functionalcomponents and class components**

|  |  |
| --- | --- |
| **Function Components** | **Class Components** |
| Function components is javascript function that return UI. | Class components that extends React.Component |
| State management uses hooks like useState. | State management uses this.state. |
| Life-cycle method useEffect. | Life-cycle method like componentDidmount. |
| Function component performance is Faster. | Class component performance is more complex. |

**2)Pass data to a component using props**

React, **props** allow us to **pass data from a parent component to a child component.**

**Example:-**

Parent components:-

function App() {

return <UserInfo name="Foram" age={22} />;

}

Child Components:-

function UserInfo(props) {

return (

<div>

<h1>Name: {props.name}</h1>

<h2>Age: {props.age}</h2>

</div>

);}

**Output:**

Name: krisha

Age : 22

**3)** **The role of render()in class components**

🡪Class component react it tells react what to display on the screen.

🡪must be include in every class components.

🡪Retrun jsx. runs automatically .

**Example**

import React, { Component } from "react";

class Print extends Component {

render() {

return <h1>Hello, World!</h1>;

}

}

export default Print;

**Props and state:**

**1)What are props in React.js**

Props are used to pass data **from a parent component to a child component** in React.props are like function arguments.

🡪Propas Data passed from parent to child. state data managed within a components.

🡪Propas can’t be change. state can change.

🡪Propas are used to send data . state are used to store data.

**2)State in React**

🡪 State in React is **a built-in object that stores data** inside a component.

🡪 React provides a special function called useState to manage state.

**Example**

🡪State can mutable.

**Example**

Button clicks.

**3)** **Why is this.setState()used in class components, and how does it work?**

**🡪Class components** this.state is used to store data, but **you cannot update state directly** by writing this.state = newValue.

🡪 It **merges** the new state with the existing state.  
🡪 It **triggers a re-render,** updating the UI with the new state value.  
 🡪It **does not change state immediately.**