|  |
| --- |
| REACT BASIC****--********1.What is React?**** React is a JavaScript library for building user interfaces, maintained by Facebook. It facilitates the creation of interactive and dynamic web applications by using a component-based architecture. React efficiently updates and renders the user interface through a virtual DOM, enhancing performance and providing a seamless development experience. ****2.What are the key features of React?**** Some key features of React include:  1.Virtual DOM for efficient rendering and performance optimization.  2.Component-based architecture for reusability and modular development.  3.One-way data flow for predictable and maintainable code.  4.Unidirectional data binding with state and props management.  5.React Native for building mobile applications using React."  **3.What is JSX? Why can't browsers read JSX?**  JSX stands for JavaScript XML. It is a React extension which allows writing JavaScript code that looks similar to HTML. It makes HTML file easy to understand. The JSX file makes the React application robust and boosts its performance. JSX provides you to write XML-like syntax in the same file where you write JavaScript code, and then preprocessor (i.e., transpilers like Babel) transform these expressions into actual JavaScript code. Just like XML/HTML, JSX tags have a tag name, attributes, and children.  Browsers cannot read JSX directly because they can only understand JavaScript objects, and JSX is not a regular JavaScript object. Thus, we need to transform the JSX file into a JavaScript object using transpilers like Babel and then pass it to the browser.  **4.What is conditional rendering in React?**  Conditional rendering in React involves rendering different components or content based on certain conditions. You can use if statements or ternary operators within JSX to conditionally render elements. ****5.What is a React Router, and why is it used?**** React Router is a library for handling routing in React applications. It allows developers to create declarative routing configurations, enabling navigation between different views or pages within a single-page application (SPA). React Router provides components like BrowserRouter, Route, and Link to define routes, match URLs, and navigate between views.  **6. What is the purpose of the callback function as an argument of setState()?**  The callback function is invoked when setState is finished and the component gets rendered. Since setState() is asynchronous the callback function is used for any post action.  Note: It is recommended to use the lifecycle method rather than this callback function.  Javascript  setState({ name: "John" }, () =>  console.log("The name has updated and component re-rendered")  );  **7.What is the difference between Shadow DOM and Virtual DOM?**  The goal of React Fiber is to increase its suitability for areas like animation, layout, and gestures. Its headline feature is incremental rendering: the ability to split rendering work into chunks and spread it out over multiple frames.  from documentation  Its main goals are:   * Ability to split interruptible work in chunks. * Ability to prioritize, rebase and reuse work in progress. * Ability to yield back and forth between parents and children to support layout in React. * Ability to return multiple elements from render(). * Better support for error boundaries.   **8.Why fragments are better than container divs?**  Below are the list of reasons to prefer fragments over container DOM elements,   * Fragments are a bit faster and use less memory by not creating an extra DOM node. This only has a real benefit on very large and deep trees. * Some CSS mechanisms like Flexbox and CSS Grid have special parent-child relationships, and adding divs in the middle makes it hard to keep the desired layout. * The DOM Inspector is less cluttered.   **9.How events are different in React?**  Handling events in React elements has some syntactic differences:  -React event handlers are named using camelCase, rather than lowercase.  -With JSX you pass a function as the event handler, rather than a string.    **REACT STATE & HOOKS--**  **10.What are React hooks?**  React hooks are functions that allow developers to use state and other React features in functional components. Hooks were introduced in React 16.8 as a way to write reusable logic and manage stateful behavior in functional components. Some commonly used hooks are useState, useEffect, and useContext.  **11.What are "hooks" in React, and how do they change the way you manage state and side effects?**  Hooks in React are functions that allow you to "hook into" React state and lifecycle features from functional components. They were introduced in React 16.8 to address some of the limitations and complexities associated with managing state and side effects in class components. Hooks make it easier to reuse logic across components and promote the use of functional components over class components. ****12.What is the purpose of state in React?**** State is used to manage and store data within a component. It represents the current state of the component and can be changed over time. When the state of a component is updated, React automatically re-renders the component and its child components to reflect the changes in the UI.  **13. Why should we not update the state directly?**  If you try to update the state directly then it won't re-render the component.  //Wrong  Javascript  this.state.message = "Hello world";  Instead, use the setState() method. It schedules an update to a component's state object. When state changes, the component responds by re-rendering.  //Correct  Javascript  this.setState({ message: "Hello World" });  Note: You can directly assign to the state object either in the constructor or using the latest javascript's class field declaration syntax.  **14.What is the purpose of the useState hook?**  The useState hook is used to add state to functional components. It takes an initial state value as an argument and returns an array with two elements: the current state value and a function to update the state. By using the useState hook, functional components can maintain their own state without using class components.  **15.What is the purpose of the useEffect hook?**  The useEffect hook in React is used to perform side effects in functional components. It allows you to execute code after the component has rendered or when specific dependencies have changed. Side effects can include API calls, subscriptions, or manipulating the DOM. The useEffect hook takes a callback function as its first argument and an optional array of dependencies as its second argument. 16.When would You use usecontext hook? You would use the useContext hook in React when you need to access data or functions from a context within a functional component. It simplifies consuming context values and is useful for managing global state, such as themes, user authentication, and application settings. ****17.What is the significance of the dependency array in the useEffect hook?**** The dependency array in the useEffect hook specifies the values that the effect depends on. When any of the values in the dependency array change, the effect is re-executed. If the dependency array is empty, the effect runs only once after the initial render. By specifying dependencies, you can control when the effect should be triggered or if it should be skipped. ****18.What is the difference between props and state?**** "Props and state are both used to pass data to components, but there are some key differences:  Props are passed from parent components to child components and are read-only within the child component. They are used to configure and customize a component.  State is managed within the component itself and can be changed using the setState() method. State represents the internal data of a component and can trigger re-rendering when updated."   **19.What is the purpose of the useReducer hook in React?**  The useReducer hook is used to manage complex state logic in a more centralized and predictable way. It is an alternative to using useState when the state transitions involve multiple values or complex logic. useReducer takes a reducer function and an initial state and returns the current state and a dispatch function to trigger state transitions. ****20.What is the purpose of the useContext hook?**** The useContext hook is used to consume a React context in a functional component. It allows you to access the value provided by a context provider higher up in the component tree without the need for prop drilling. The useContext hook takes the context object as its argument and returns the current context value. |