**React interview questions**

**1. What is React and why is it used?**

Answer: React is a Javascript library that is used for building web applications. React is a component-based library. Each component shows the UI interface of the web applications. React builds lightweight interactions of the DOM in the memory. React makes a virtual DOM to render the components. In React the declaring syntax is user-friendly and also for the development process. So the UI behavior is easy to understand. React is a one-way-data flow library, it reduces complexity and is easier to understand. React has a large number of third-party components and extensions that make a React web application more attractive, interactive, and responsive. React has a large community to develop itself, and the syntax is becoming easier day by day.

**2. What are the key features of React?**

Answer: First of all I would like to say about the JSX extension file where HTML can be written within Javascript. This feature made React more popular, user-friendly, and easier to use. React make a virtual DOM be rendered and effectively make an update.

React goes with one-way-data flow, where the data flows from the parent component to the child component. This helps maintain a predictable state and simplifies debugging and testing. When the state or props of a component change, React determines the minimal set of changes needed to update the UI. React provides a set of lifecycle methods that allow developers to hook into different stages of a component. Lifecycle methods help manage component behavior and optimize performance.

**3. What is JSX in React?**

Answer: JSX is the syntax extension in React that is written in Javascript. But HTML can be used within it. It is the main benefit of JSX. It allows you to define elements, attributes, and content in a concise and declarative manner. When using JSX, each HTML-like tag corresponds to a React component. JSX also allows the use of JavaScript expressions within curly braces {}. This enables the dynamic rendering of content. The transpiler transforms JSX elements into React.createElement() function calls, which create React elements representing the desired UI structure. It improves code readability, component reusability, and the separation of concerns by combining HTML structure and JavaScript logic within a single file.

**4. Explain the concept of Virtual DOM in React.**

Answer: Virtual DOM is a key concept of React that helps to sufficient rendering and do update the user interface. It is a lightweight representation of the actual DOM. When there is a change in the component's state or props, React triggers a process called reconciliation. React identifies the differences between the two representations, determining what has changed in terms of elements. After the differences are identified, React calculates the minimum set of updates needed to reflect the changes in the real DOM. React tries to perform the smallest possible number of DOM manipulations.

**5. What is the difference between state and props in React?**

Answer: State represents the internal data and state of a component. It is managed by the component itself. Each component has its own state, which can be updated and changed over time. Props represent the external data passed to a component. They are used to provide data and configuration from a parent component to its child components. Declaration: State is declared and initialized within a component using the “useState()”, Props are passed from a parent component to its child component through attributes in JSX when rendering the child component.

**6. What are React hooks? List some commonly used hooks.**

Answer: React hook is a state that can be used in the other components without changing it. Hooks are generally two types in use. There exist some in-build hooks and also side-effects, and users can also create and use custom hooks. When we try to change the side effects of the web application, we always use the in-built hooks like useState(), useEffect(), useContext(), and useRef() are mostly used in React. If we want to make our code smaller, more interactive, and more professional, we use custom hooks. Custom hooks allow us to use a stateful manner on other components without changing codes. Custom hooks are always reusable whenever we want.

**7. Explain the lifecycle methods in React.**

Answer: There are three steps in the lifecycle of React. When a component is created and then inserted into the DOM, the mounting phase is created. The Updating Phase in React occurs when a component is re-rendered due to changes in its state or props. The Unmounting Phase in React occurs when a component is removed from the DOM. This phase involves the cleanup and removal of resources associated with the component.

**8. What is the significance of keys in React lists?**

Answer: Keys are one kind of attribute that is used in React when the component is in rendering. Keys attributes help React to update significant elements, keys give every element a unique identifier by which React can update specific elements. React also can add or remove any component with the help of the unique identifier that is given by keys. Without the help of keys, React can only update the entire component. React can reuse the components with the help of keys. Now, keys must be unique among the child elements. But it is not must to be unique in the global. In one line key is a unique name that holds the component or element as a unique value.

**9. What are controlled components in React?**

Answer: First of all controlled components in React means the input or form elements, which is belonged to HTML. The input or form elements are, like “input”, “textarea”, and “select”, etc the mostly used controlled elements. In React the values of the components are controlled by the states of React. The “onSubmit” event handler is mostly used for controlling the values of a form in React. With this event handler, the user data’s value can be passed through and always in a syncing point. The “onClick”, and “onChange” event handlers are also generally used for form. With the help of React, we can initialize the form when updating or adding all the data successfully. Overall React controlled components give powerful features to add, remove, update data, and also manipulate the values.

**10. Explain the concept of higher-order components (HOCs) in React.**

Answer: Higher-order components are the necessary function of React that takes a component as an argument and returns a developer an enhanced one. It can wrap a component and returns a new component without changing the feature of that component. HOC can be reused among several components without changing the element. HOCs also allow composition, where multiple HOCs can be combined to create a more complex and feature-rich component. HOCs can add new props, can update the props, and can give additional data to a component whenever it is needed. HOCs are a powerful pattern in React for code, and enhancing component functionality.

**11. What is the significance of the "render" method in React?**

Answer: “Render” is a very important method in React. Render defines that what should be updated, added, or removed from the DOM elements. The method is responsible for returning the JSX or React elements that represent the component's user interface. Inside the "render" method, we create React elements using JSX. The "render" method should be unique. If we give the same props and state, the "render" method will always give the same output. When a component's state or props change, React triggers a process called reconciliation, where it compares the previously rendered output with the new output from the "render" method. Also, the "render" method should be a pure function without any side effects. It only creates the UI structure based on the component's props and state.

**12. How does React handle events?**

Answer: In React, event handlers by using a type of props named after the unique event, onClick, onChange, or onSubmit. Then we have to assign a function to these props that will be called when the event is triggered. React creates multiple event updates together to optimize performance. This helps prevent unnecessary re-renders. To optimize memory usage, React uses event pooling. After an event is processed, the synthetic event object is recycled and reused for future events. React provides wrappers for common event modifiers and methods to prevent the default behavior of events. For example, you can use event.preventDefault() to prevent the default action of a form submission or event.stopPropagation() to stop the event from propagating further up the component.

**13. What is the purpose of refs in React?**

Answer: “refs” means reference in React, which gives a way to access and interact with DOM elements. “refs” generally retrieve information from a specific element, such as reading its value, focusing it, or triggering specific DOM methods. “refs” can also be used to control selection ranges within input fields or manipulate media playback, like pausing or seeking a video. Sometimes, Refs are used to integrate React with third-party libraries that rely on direct DOM access. That helps a developer build more interactive web applications. “refs” can be created using the “React.createRef()” method or by using the ref callback function. Refs are then assigned to elements or components as a prop.

**14. Explain the concept of context in React.**

Answer: context is a feature that allows data to be shared. Context can be accessed by many components without passing any props. Context revolves around two main components: the Provider and the Consumer. The Provider component allows you to define and provide the context data to be shared. The Consumer component allows components within its subtree to access and consume the provided context data. Context is created using the “createContext()” method. It returns an object with a Provider. The Provider helps for making the context data available to the components that need it. Then we have to pass the data as a prop to the Provider component, using the “value” prop.

15. What is the purpose of the "setState" method in React?

Answer: The “setState” method is used to update the state of a component and trigger a re-render of the component and its child components. It is a built-in method. It allows us to update specific properties of the component's state object. When “setState” is called and the state changes, React automatically triggers a re-render of the component. It is asynchronous, meaning that React may batch multiple “setState” calls together for performance reasons. It allows us to update the state based on the previous state. By using the “setState”, React provides a way to manage and update component state.

**16. How does React handle forms?**

Answer: First of all controlled components in React means the input or form elements, which is belonged to HTML. The input or form elements are, like “input”, “textarea”, and “select”, etc the mostly used controlled elements. In React the values of the components are controlled by the states of React. The “onSubmit” event handler is mostly used for controlling the values of a form in React. With this event handler, the user data’s value can be passed through and always in a syncing point. The “onClick”, and “onChange” event handlers are also used for form. With the help of React, we can initialize the form when updating or adding all the data successfully. Overall React controlled components give powerful features to add, remove, update data, and also manipulate the values.

**17. Explain the concept of error boundaries in React.**

Answer: Normally, if an error occurs within a component's render method or its lifecycle methods, it would propagate up the component tree, potentially causing the entire application to crash. An error boundary component is a regular React component that implements the componentDidCatch lifecycle method. This method is called when an error occurs within the component or its child components. To create an error boundary, we wrap the component or components that you want to protect with an error boundary component using JSX syntax. Error boundaries only catch errors that occur during rendering or lifecycle methods. By using error boundaries, we can isolate and handle errors within a specific part of your component tree without affecting the rest of the application.

**18. What are the differences between functional components and class components?**

Answer: Functional components and class components are two different ways of defining components in React, each with its own syntax and features. Class components have built-in support for state management and lifecycle methods. They have a dedicated state object and can define lifecycle methods such as componentDidMount, componentDidUpdate, and componentWillUnmount. Class components use this to access props, state, and other class methods within the component. Class components can be more complex and verbose compared to functional components. They require the definition of a class, and the use of the render() method to return JSX.

**19. How can you optimize performance in React?**

Answer: Optimizing performance in React involves identifying and implementing strategies to make users' applications run faster, consume fewer resources, and provide a smoother user experience. Functional components with React hooks are generally more performant than class components. Hooks, such as useState, useEffect, and useMemo, allow for better control of state and side effects, reducing unnecessary re-renders and improving performance. Wrap components with React.memo to memoize and prevent unnecessary re-renders when the component's props have not changed. Memoization helps optimize performance by skipping the rendering process when it is not required.

**20. What are React portals?**

Answer: React portals are a feature in React that provides a way to render components into a different part of the DOM tree, outside the parent component's hierarchy. They allow you to render components outside their usual parent-child relationship and enable more flexible rendering scenarios. With portals, we can build a component into a different DOM node that is not a direct parent of the component. This is useful when we need to render content into a different part of the DOM, such as a modal, a tooltip, or a portal for modals. Portals enable a separation of concerns between the logical component structure and the visual representation.

**21. Explain the concept of code splitting in React.**

Answer: Code splitting in React is a technique used to split a large JavaScript bundle into smaller, more manageable chunks. The purpose of code splitting is to optimize the initial load time of a web application by only loading the code that is necessary for the current page or component.

**22. How do you handle routing in React?**

Answer: Routing is handled by using third-party libraries that provide routing, such as React Router. React Router is the most popular and widely used routing library for React web applications. Firstly we have to install npm or yarn in the code editor. Then, In the application's entry point or the component where you want to handle routing, import the necessary components from React Router. Then Use the Link component from React Router to create navigation links within your application. React Router allows you to define route parameters to handle dynamic segments in your URLs. React Router provides additional features such as nested routes, route guards, route transitions, and more. It offers a comprehensive and flexible routing solution for building single-page applications with React.

**23. What is the purpose of React Fragments?**

Answer: React Fragments provide a way to group a list of children elements without introducing an extra wrapping element to the DOM. They allow you to render multiple elements as siblings within a component's render method without adding unnecessary markup. Fragments improve code readability, maintainability, and performance by reducing the number of unnecessary DOM nodes. Fragments can be used with two different syntaxes. The first syntax is using the <React.Fragment> component or its shorthand syntax <></>. Both syntaxes achieve the same result of grouping multiple elements without adding an extra DOM node. Fragments can accept a key prop just like any other React element. This is useful when rendering lists of fragments, allowing React to efficiently update and reconcile the list items.

**24. How can you prevent the default behavior of an event in React?**

Answer: In React, we can prevent the default behavior of an event by calling the preventDefault() method on the event object. This method stops the default action associated with the event, Like form submission, link navigation, or element finding. In the event handler function, we can call the preventDefault() method on the event object. This method helps the UI to prevent the default behavior associated with the event. After calling preventDefault(), you can write custom code to handle the event according to your requirements. This may include modifying the component's state, performing additional actions, making API requests, or navigating to a different route.

**25. Explain the concept of React Fiber.**

Answer: React Fiber is a reimplementation of the React reconciliation algorithm and its core algorithm for updating the user interface (UI). Reconciliation is the process in React that compares the previous state of the UI with the new state and determines the minimal set of changes needed to update the UI accordingly. React Fiber is a reimplementation of the reconciliation algorithm using a new data structure called "fiber." The fiber architecture allows for incremental, asynchronous rendering, enabling better performance and the ability to pause, abort, or prioritize rendering work. One of the key features of React Fiber is the ability to perform work in smaller units called "time slices" or "work chunks." React Fiber introduces the concept of "virtual time" to represent the progress of work.

**26. What are the differences between React and Angular?**

Answer: React is a JavaScript library for building user interfaces, and Angular is a complete framework for building web applications. It focuses on the component-based architecture, where you create reusable UI components that manage their own state. It follows the MVC architectural pattern, where components are organized into modules and use services for data management. React provides a virtual DOM, allowing efficient updates to the UI based on state changes, Angular has a comprehensive set of features, including dependency injection, built-in routing, and powerful templating capabilities.

**27. How can you pass data between parent and child components in React?**

Answer: In React, you can pass data from a parent component to a child component by utilizing props. Props means properties, are a way to pass data from a parent component to its child components. Firstly, In the parent component, define the data to pass as a prop. Include the child component in the parent component's JSX code and pass the data as a prop. In the child component, access the data passed from the parent component through the props object. To access the passed data, you can use props.propName within the child component's code. By passing data from a parent component to a child component through props. If the child component needs to update the data or communicate changes back to the parent component, the parent can pass callback functions as props.

**28. How would you handle state management in a large React application?**

Answer: Handling state management in a large React application requires careful consideration and the use of appropriate techniques and tools. For managing state within individual components, you can use local component state using the useState hook or the this.state and this.setState methods in class components. React's Context API allows you to create and share state across components without manually passing props through intermediate components. Context is useful for global or shared state that needs to be accessible by multiple components. We can create a context using createContext() and provide the context value at a higher level in the component tree using a Provider component. React Query is a library specifically designed for handling server state and data fetching in React applications. It simplifies the management of asynchronous data, caching, and synchronization with server data.

**29. What are the benefits of using Redux for state management in React?**

Answer: Redux offers a centralized store where you can store and manage the state of your application. This makes it easier to access and update the state from any component without passing props through intermediate components. It provides a single source of truth for your application's state. Redux follows a strict pattern of state updates through actions and reducers. Redux's strict unidirectional data flow and immutability make it well-suited for time-travel debugging. With tools like Redux DevTools, you can inspect and replay past actions, step through state changes, and debug your application's state at different points in time. Redux is designed to handle complex state management scenarios in large applications. It encourages modular and reusable code by separating state management logic from component code.

**30. Discuss the concept of React Native and explain how it differs from React in terms of development and target platforms.**

Answer: React Native is a framework that allows developers to build mobile applications using React and JavaScript. While React is used for building web applications, React Native extends the capabilities of React to enable cross-platform mobile app development. With React, we can share code and logic across web platforms. React Native allows for code sharing between iOS and Android platforms to a large extent. React renders components into the DOM, resulting in a web-based user interface. React Native renders components into native UI elements for each platform. React development typically relies on web development tools like web browsers, and web dev tools, React Native provides its own development environment and tooling.