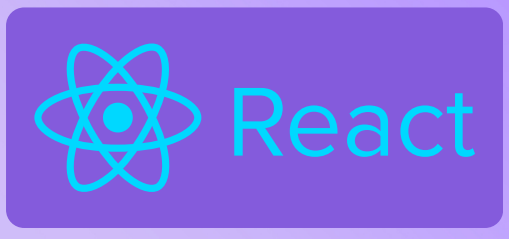
100 React js Interview Q&A



What is React.js?

React.js is an open-source JavaScript library used for building user interfaces. It allows developers to create reusable UI components and efficiently update and render them as the application state changes.

What are the key features of React.js?

React.js has several key features, including:

- Virtual DOM: React uses a virtual representation of the DOM to efficiently update and render components.
- Component-based architecture: React follows a modular approach, where UIs are built using reusable components.
- Unidirectional data flow: React follows a one-way data binding, making it easier to track and debug data changes.
- JSX: React uses JSX, a syntax extension that allows developers to write HTML-like code within JavaScript.

What is JSX?

JSX is a syntax extension for JavaScript used in React.js. It allows developers to write HTML-like code directly within JavaScript, making it easier to create and manipulate React elements.

Explain the difference between functional components and class components in React.

Functional components are stateless and are defined as JavaScript functions. They are simpler and easier to read, test, and maintain. Class components, on the other hand, are defined as ES6 classes and

have additional features like lifecycle methods and state management.

What is the significance of the "render" method in React?

The "render" method is a required method in every React component. It returns a description of what the component should render. The render method is responsible for rendering the component's JSX or other React components.

What is a React element?

A React element is a lightweight representation of a DOM element. It is a plain JavaScript object that describes what should be rendered on the screen. React elements are the building blocks of React applications.

What is a React component?

A React component is a reusable and self-contained module that represents a part of the user interface. It can be a function component or a class component. Components encapsulate logic and UI, making it easier to develop and maintain complex applications.

What is state in React?

State is an object that holds data that can change over time in a React component. It represents the mutable part of the component and can be accessed using this.state. When the state changes, React re-renders the component to reflect the updated data.

What is the difference between props and state in React?

Props (short for properties) and state are both used to manage data in React components, but they have some differences:

- Props are read-only and are passed from parent components to child components. They are used to pass data down the component tree.

- State is mutable and is managed internally within a component. It is used to manage data that can change over time within a component.

What is the significance of keys in React lists?

Keys are used to give a unique identity to each element in a list rendered by React. They help React identify which items have changed, been added, or been removed. Keys improve the efficiency of list rendering and help maintain component state correctly.

Explain the concept of the virtual DOM in React.

The virtual DOM is a lightweight copy of the actual DOM maintained by React. It is a representation of the UI components and their structure. When there are changes in the application state, React compares the virtual DOM with the real DOM and efficiently updates only the necessary parts, resulting in better performance.

What are React hooks?

React hooks are functions that allow functional components to use state and other React features without writing a class. Hooks, such as useState and useEffect, provide a simpler way to manage component state and lifecycle.

What is the purpose of the useEffect hook?

The useEffect hook is used to perform side effects in a React component. It allows you to perform actions like data fetching, subscriptions, or manually changing the DOM after the component has rendered.

What is the role of the useState hook?

The useState hook is used to add state to functional components. It returns an array with two elements: the current state value and a function to update that value. It enables functional components to have state similar to class components.

What are controlled components in React?

Controlled components are React components that manage their state using the state property and update it using setState(). They receive their current value and change callbacks as props, and they notify parent components of state changes through these callbacks.

What is the purpose of the useCallback hook?

The useCallback hook is used to memoize functions in React. It returns a memoized version of the callback function that only changes if one of the dependencies has changed. It helps optimize performance by avoiding unnecessary re-rendering of components.

Explain the concept of prop drilling in React.

Prop drilling refers to the process of passing props through multiple layers of components in order to reach a deeply nested component that needs access to the data. It can make the code more complex and lead to less maintainable code. Context API or Redux can be used to avoid prop drilling.

What is the Context API in React?

The Context API is a built-in feature of React that provides a way to share data between components without explicitly passing it through props. It allows you to create a global state accessible to any component within a designated context.

What is the significance of React Router?

React Router is a popular library used for implementing routing in React applications. It allows developers to handle navigation and rendering of different components based on the URL. React Router provides a declarative way to define routes and nested routes in the application.

What is Redux?

Redux is a predictable state container for JavaScript applications. It provides a centralized store to manage the state of an application and follows a unidirectional data flow. Redux is commonly used with React to manage the application's global state.

Explain the concept of "pure components" in React.

Pure components are React components that implement a shouldComponentUpdate method with a shallow prop and state comparison. They only re-render when the data they receive or hold has changed. Pure components optimize rendering performance by preventing unnecessary re-rendering.

What is the significance of the React Developer Tools extension?

The React Developer Tools extension is a browser extension that allows developers to inspect and debug React components. It provides a panel that displays the component hierarchy, props, state, and other useful information for React applications.

How can you optimize the performance of a React application?

To optimize the performance of a React application, you can:

Use shouldComponentUpdate or PureComponent to prevent unnecessary re-rendering.

Implement code splitting and lazy loading to load components only when needed.

Use memoization techniques such as useMemo and useCallback to avoid expensive computations or unnecessary re-renders.

Implement server-side rendering (SSR) to improve the initial load time and SEO.

Optimize the size of the bundled JavaScript and CSS files using techniques like minification and compression.

What are the different lifecycle methods in React?

React has several lifecycle methods, including:

componentDidMount: Called after the component has been mounted to the DOM.

componentDidUpdate: Called after the component has been updated and re-rendered.

componentWillUnmount: Called just before the component is unmounted and destroyed.

shouldComponentUpdate: Allows optimization by determining whether the component should re-render or not.

render: Renders the component's JSX or other React components.

What is the significance of the "key" prop in React?

The "key" prop is used to give a unique identity to elements in an array rendered by React. It helps React optimize the rendering process by efficiently updating and reordering the elements when necessary.

What is the purpose of the "dangerouslySetInnerHTML" attribute in React?

The "dangerouslySetInnerHTML" attribute is used to insert HTML content directly into a component. It should be used with caution as it poses a security risk if not used properly.

How can you prevent a component from rendering in React?

You can prevent a component from rendering by returning null or false from its render method. Alternatively, you can use lifecycle methods like shouldComponentUpdate or React.memo to control the re-rendering process.

What is the significance of the "children" prop in React?

The "children" prop is a special prop in React that allows components to display whatever is included between their opening and closing tags. It can be used to pass JSX elements or plain text as children to a component.

How can you update the state of a component in React?

To update the state of a component, you should never mutate the state directly. Instead, use the setState method provided by React. It accepts a new state object or a function that returns a new state object, and React will merge the changes and trigger a re-render.

What is the purpose of the "defaultProps" property in React?

The "defaultProps" property is used to define default values for props in a React component. If a prop is not provided when the component is used, the default value specified in defaultProps will be used instead.

What is React Fiber?

React Fiber is a reimplementation of React's core algorithm for rendering and updating components. It was introduced to improve performance, enable incremental rendering, and better support concurrent mode.

Explain the concept of "render props" in React.

Render props is a technique in React where a component accepts a function as a prop and uses it to render content. This allows the component to share its internal state and logic with the function component passed as a prop, enabling code reuse and composability.

What is the purpose of the "key" prop in a list of React components?

The "key" prop is used to provide a unique identifier for each item in a list of React components. It helps React identify which items have changed, been added, or been removed, improving the efficiency of list rendering and component updates.

What are portals in React?

Portals in React provide a way to render children components into a different DOM node that exists outside the parent component's hierarchy. It allows you to render components at a different position

in the DOM tree, which can be useful for modals, tooltips, and overlays.

What is the significance of the "dangerouslySetState" method in React?

The "dangerouslySetState" method is used to update the state of a React component. It is called directly on the component instance and bypasses the usual state update mechanism provided by setState. The use of "dangerouslySetState" is discouraged and should be avoided in most cases.

Explain the concept of code splitting in React.

Code splitting is a technique used to split a React application's JavaScript bundle into smaller chunks. It allows you to load only the required code when needed, improving the initial load time and reducing the overall bundle size. React.lazy and React.Suspense are commonly used for code splitting.

What is the purpose of the "Fragment" component in React?

The "Fragment" component is a built-in component in React that allows you to group multiple elements together without adding an extra DOM node. It is useful when you need to return multiple elements from a component's render method without wrapping them in a parent element.

How can you handle forms in React?

In React, form handling typically involves capturing user input, managing it with component state, and handling form submission. You can use controlled components, where form elements are linked to the component state, or use libraries like Formik or React Hook Form for advanced form handling.

What are React refs?

Refs in React are a way to access and interact with DOM elements or React components directly. They provide a way to reference and manipulate DOM elements outside the usual React component

hierarchy. Refs should be used sparingly and primarily for interacting with non-React libraries or imperative DOM operations.

What is the purpose of the "componentWillUnmount" method in React?

The componentWillUnmount method is a lifecycle method in React that is called just before a component is unmounted and destroyed. It provides an opportunity to perform cleanup tasks such as canceling network requests, removing event listeners, or clearing timers.

Explain the concept of error boundaries in React.

Error boundaries are React components that catch JavaScript errors in their child component tree and display fallback UI instead of crashing the whole application. They help isolate errors and provide a better user experience by gracefully handling errors at the component level.

What is the purpose of the "React.StrictMode" component?

The React.StrictMode component is a wrapper component provided by React. When used, it enables additional warnings and checks during development. It helps highlight potential problems and deprecated features in the application code.

What is the significance of the "setState" method in React?

The setState method is used to update the state of a React component. It accepts a new state object or a function that returns a new state object. React will merge the changes and trigger a re-render of the component and its children.

What is the purpose of the "shouldComponentUpdate" method in React?

The shouldComponentUpdate method is a lifecycle method that allows you to optimize performance by determining whether a component should re-render or not. It is called before rendering and

can be used to compare previous and new props or state to decide whether an update is necessary.

What is the role of the "React.Fragment" component in React?

The React.Fragment component allows you to return multiple elements from a component's render method without adding an extra DOM node. It is useful when you don't want to introduce an additional wrapper element in the rendered output.

What is the purpose of the "getDerivedStateFromProps" method in React?

The getDerivedStateFromProps is a static lifecycle method that is called before rendering when new props are received. It allows you to update the component's state based on the changes in the incoming props. It is rarely needed and can often be replaced with other patterns like componentDidUpdate or useState.

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

Higher-order components are functions that take a component as input and return an enhanced version of that component. HOCs are used for code reuse, cross-cutting concerns, and augmenting components with additional functionality. They are a powerful pattern for component composition in React.

What is the purpose of the "React.memo" function in React?

The React.memo function is a higher-order component (HOC) that memoizes a functional component. It helps optimize performance by preventing unnecessary re-renders of the component when its props haven't changed. It is similar to PureComponent for class components.

What is the difference between an element and a component in React?

In React, an element is a plain JavaScript object that represents a single instance of a DOM element or a React component. It is the smallest building block in React. A component, on the other hand, is

a reusable and self-contained module that encapsulates logic and UI. Components can be either function components or class components.

What is the purpose of the "React.createRef" method in React?

The React.createRef method is used to create a ref object in React. Refs are a way to access and interact with DOM elements or React components directly. They provide a way to reference and manipulate elements outside the usual React component hierarchy.

How can you conditionally render components in React?

Conditional rendering in React can be achieved using various techniques, such as:

Using if statements or ternary operators in the component's render method.

Using logical && operator to conditionally render an element.

Using the map method to render a list of components conditionally.

Using conditional rendering libraries like react-router or react-loadable for more complex scenarios.

What is the purpose of the "React.cloneElement" method in React?

The React.cloneElement method is used to clone and modify a React element with new props. It allows you to pass additional props to a child component created by another component. It is often used in cases where you need to enhance or modify the behavior of a child component.

What is the significance of the "React.PureComponent" class in React?

The React.PureComponent class is a base class in React that provides a convenient way to optimize performance for class components. It implements a shallow prop and state comparison in the shouldComponentUpdate method, automatically preventing unnecessary re-renders when the data hasn't changed.

How can you pass data between components in React?

Data can be passed between components in React through props. Parent components can pass data to child components by setting props on the child component elements. In more complex scenarios, you can use techniques like context API, Redux, or libraries like React Router to share data across components.

What is the purpose of the "React.Children" utility in React?

The React.Children utility provides a set of methods for working with the props.children property of a React component. It allows you to iterate, map, and count the children of a component, regardless of whether they are single elements or arrays of elements.

Explain the concept of "uncontrolled components" in React.

Uncontrolled components are form elements in React that store their own state internally, rather than being controlled by React's component state. They allow user input to be directly accessed through the DOM, which can be useful in some scenarios but can make it harder to enforce constraints or validation.

What is the purpose of the "React.createContext" method in React?

The React.createContext method is used to create a new context object in React. Context provides a way to share data between components without having to pass it explicitly through props. It is especially useful when data needs to be accessed by multiple components at different levels of the component tree.

How can you handle events in React?

In React, you can handle events by providing event handlers as props to elements. Event handlers are functions that are executed when a specific event occurs, such as a button click or a form submission. React uses synthetic events that wrap the native browser events and provide consistent behavior across different browsers.

What is the purpose of the "React.createPortal" method in React?

The React.createPortal method is used to render a React component's content into a different DOM node, which can be outside the parent component's hierarchy. It is commonly used to render modal dialogs, tooltips, or overlays that need to be rendered at a different position in the DOM tree.

What is the role of the "componentDidCatch" method in React?

The componentDidCatch method is a lifecycle method in React that is called when an error occurs within a component's child tree. It allows the component to catch the error and handle it gracefully by displaying an error boundary or a fallback UI.

Explain the concept of lazy loading in React.

Lazy loading is a technique used to load components or assets only when they are needed, rather than loading everything upfront. React.lazy and Suspense are used to implement lazy loading in React. Lazy loading can significantly improve the initial load time and performance of an application.

What is the purpose of the "React.Suspense" component in React?

The React.Suspense component is a built-in component in React that enables handling of loading states and code splitting. It is used in combination with React.lazy to specify a fallback UI or loading indicator while waiting for a component to load asynchronously.

What is the significance of the "React.memo" function in React?

The React.memo function is a higher-order component (HOC) that memoizes a functional component. It helps optimize performance by preventing unnecessary re-renders of the component when its props haven't changed. It is similar to PureComponent for class components.

How can you prevent a component from re-rendering in React?

To prevent a component from re-rendering, you can use techniques like:

Using PureComponent or React.memo to optimize performance and prevent re-rendering when the props haven't changed.

Implementing the shouldComponentUpdate lifecycle method and returning false when no updates are needed.

Using state management libraries like Redux or MobX to manage the component's state outside the component hierarchy.

What is the purpose of the "React.Fragment" component in React?

The React.Fragment component allows you to group multiple elements together without adding an extra DOM node. It is useful when you need to return multiple elements from a component's render method without wrapping them in a parent element.

Explain the concept of error boundaries in React.

Error boundaries are React components that catch JavaScript errors in their child component tree and display fallback UI instead of crashing the whole application. They help isolate errors and provide a better user experience by gracefully handling errors at the component level.

What is the purpose of the "useEffect" hook in React?

The useEffect hook is used to perform side effects in a functional component. It is similar to componentDidMount, componentDidUpdate, and componentWillUnmount combined. It allows you to perform actions like data fetching, subscriptions, or manually changing the DOM after the component has rendered.

What is the difference between useState and useReducer hooks in React?

useState is a hook used to manage state in functional components, providing a way to add local state to a component. useReducer, on the other hand, is a hook that provides a more powerful state management solution for complex state logic. It is based on the concept of a reducer function and actions.

What is the purpose of the "useContext" hook in React?

The useContext hook is used to consume values from a context in functional components. It allows you to access the current context value directly without wrapping the component in a context consumer. It simplifies the process of accessing shared data across components.

What is the purpose of the "useCallback" hook in React?

The useCallback hook is used to memoize functions in React. It returns a memoized version of the callback function that only changes if one of the dependencies has changed. It helps optimize performance by avoiding unnecessary re-rendering of components.

How can you handle forms in React?

In React, form handling typically involves capturing user input, managing it with component state, and handling form submission. You can use controlled components, where form elements are linked to the component state, or use libraries like Formik or React Hook Form for advanced form handling.

What is the purpose of the "useState" hook in React?

The useState hook is used to add state to functional components. It returns an array with two elements: the current state value and a function to update that value. It enables functional components to have state similar to class components.

What is the significance of the "key" prop in React lists?

The "key" prop is used to give a unique identity to each element in a list rendered by React. It helps React identify which items have changed, been added, or been removed. Keys improve the efficiency of list rendering and help maintain component state correctly.

What are hooks in React?

Hooks are functions that allow functional components to use state and other React features without writing a class. They were

introduced in React 16.8 and provide a simpler and more flexible way to manage component state and lifecycle.

What is the purpose of the "React.memo" function in React?

The React.memo function is a higher-order component (HOC) that memoizes a functional component. It helps optimize performance by preventing unnecessary re-renders of the component when its props haven't changed. It is similar to PureComponent for class components.

How can you handle events in React?

In React, you can handle events by providing event handlers as props to elements. Event handlers are functions that are executed when a specific event occurs, such as a button click or a form submission. React uses synthetic events that wrap the native browser events and provide consistent behavior across different browsers.

What is the significance of the "dangerouslySetInnerHTML" attribute in React?

The "dangerouslySetInnerHTML" attribute is used to insert HTML content directly into a component. It should be used with caution as it poses a security risk if not used properly.

What is the purpose of the "defaultProps" property in React?

The "defaultProps" property is used to define default values for props in a React component. If a prop is not provided when the component is used, the default value specified in defaultProps will be used instead.

What is the purpose of the "componentDidMount" method in React?

The componentDidMount method is a lifecycle method in React that is called after the component has been mounted to the DOM. It is commonly used to initiate network requests, set up subscriptions, or perform other initialization tasks.

What is the role of the "componentDidUpdate" method in React?

The componentDidUpdate method is a lifecycle method in React that is called after a component has been updated and re-rendered. It is often used to perform side effects based on changes in props or state, such as making additional network requests or updating the DOM.

What is the purpose of the "componentWillUnmount" method in React?

The componentWillUnmount method is a lifecycle method in React that is called just before a component is unmounted and destroyed. It provides an opportunity to perform cleanup tasks such as canceling network requests, removing event listeners, or clearing timers.

What is the difference between a controlled component and an uncontrolled component in React?

A controlled component is a component whose state is controlled by React. It receives its current value and change callbacks as props and notifies parent components of state changes through these callbacks. An uncontrolled component, on the other hand, stores its own state internally and manages it through the DOM, without involving React's component state.

What is the purpose of the "render" method in React?

The "render" method is a required method in every React component. It returns a description of what the component should render. The render method is responsible for rendering the component's JSX or other React components.

What is the significance of the "children" prop in React?

The "children" prop is a special prop in React that allows components to display whatever is included between their opening and closing tags. It can be used to pass JSX elements or plain text as children to a component.

How can you update the state of a component in React?

To update the state of a component, you should never mutate the state directly. Instead, use the setState method provided by React. It accepts a new state object or a function that returns a new state object, and React will merge the changes and trigger a re-render.

What is the purpose of the "shouldComponentUpdate" method in React?

The shouldComponentUpdate method is a lifecycle method that allows you to optimize performance by determining whether a component should re-render or not. It is called before rendering and can be used to compare previous and new props or state to decide whether an update is necessary.

What is the difference between props and state in React?

Props (short for properties) and state are both used to manage data in React components, but they have some differences:

Props are read-only and are passed from parent components to child components. They are used to pass data down the component tree.

State is mutable and is managed internally within a component. It is used to manage data that can change over time within a component.

What is the significance of the virtual DOM in React?

The virtual DOM is a lightweight copy of the actual DOM maintained by React. It is a representation of the UI components and their structure. When there are changes in the application state, React compares the virtual DOM with the real DOM and efficiently updates only the necessary parts, resulting in better performance.

What is JSX in React?

JSX is a syntax extension for JavaScript used in React. It allows developers to write HTML-like code directly within JavaScript, making it easier to create and manipulate React elements.

What are React fragments?

React fragments allow you to group multiple elements together without adding an extra DOM node. They provide a way to return multiple elements from a component's render method without wrapping them in a parent element.

What is the Context API in React?

The Context API is a built-in feature of React that provides a way to share data between components without explicitly passing it through props. It allows you to create a global state accessible to any component within a designated context.

What is the purpose of React Router in React applications?

React Router is a popular library used for implementing routing in React applications. It allows developers to handle navigation and rendering of different components based on the URL. React Router provides a declarative way to define routes and nested routes in the application.

What is Redux and how does it relate to React?

Redux is a predictable state container for JavaScript applications. It provides a centralized store to manage the state of an application and follows a unidirectional data flow. Redux is commonly used with React to manage the application's global state.

How can you optimize the performance of a React application?

To optimize the performance of a React application, you can:

Use shouldComponentUpdate or PureComponent to prevent unnecessary re-rendering.

Implement code splitting and lazy loading to load components only when needed.

Use memoization techniques such as useMemo and useCallback to avoid expensive computations or unnecessary re-renders.

Implement server-side rendering (SSR) to improve the initial load time and SEO.

Optimize the size of the bundled JavaScript and CSS files using techniques like minification and compression.

What are the different lifecycle methods in React?

React has several lifecycle methods, including:

componentDidMount: Called after the component has been mounted to the DOM.

componentDidUpdate: Called after the component has been updated and re-rendered.

componentWillUnmount: Called just before the component is unmounted and destroyed.

shouldComponentUpdate: Allows optimization by determining whether the component should re-render or not.

render: Renders the component's JSX or other React components.

What is the purpose of the "dangerouslySetState" method in React?

The dangerouslySetState method is used to update the state of a React component. It is called directly on the component instance and bypasses the usual state update mechanism provided by setState. The use of "dangerouslySetState" is discouraged and should be avoided in most cases.

Explain the concept of prop drilling in React.

Prop drilling refers to the process of passing props through multiple layers of components in order to reach a deeply nested component that needs access to the data. It can make the code more complex and lead to less maintainable code. Context API or Redux can be used to avoid prop drilling.

What is the purpose of the React Developer Tools extension?

The React Developer Tools extension is a browser extension that allows developers to inspect and debug React components. It provides a panel that displays the component hierarchy, props, state, and other useful information for React applications.

How does React handle lists and keys?

When rendering lists in React, each item in the list should have a unique "key" prop assigned to it. The "key" prop helps React identify which items have changed, been added, or been removed. Keys improve the efficiency of list rendering and help maintain component state correctly.

What is the role of the "defaultProps" property in React components?

The "defaultProps" property is used to define default values for props in a React component. If a prop is not provided when the component is used, the default value specified in defaultProps will be used instead.