

JavaScript and ReactJS Interview Questions

JavaScript and ReactJS Interview Questions

JavaScript - Basic

What are the differences between `var`, `let`, and `const`?

`var` is function-scoped, `let` and `const` are block-scoped. `var` can be re-declared, `let` cannot. `const` is read-only and must be initialized at declaration.

Explain the concept of hoisting in JavaScript.

Hoisting is a JavaScript mechanism where variables and function declarations are moved to the top of their containing scope during the compilation phase.

What is the difference between `==` and `===`?

`==` checks for equality with type coercion, whereas `===` checks for strict equality without type conversion.

What are closures in JavaScript?

A closure is a function that has access to its own scope, the scope of the outer function, and the global scope.

Explain event delegation in JavaScript.

Event delegation allows you to avoid adding event listeners to multiple elements by adding a single event listener to a common parent element. The event bubbles up from the target element to the parent.

JavaScript and ReactJS Interview Questions

JavaScript - Intermediate

How does the `this` keyword work in JavaScript?

`this` refers to the context in which a function is executed. In a method, it refers to the object that owns the method. In a regular function, it refers to the global object (in non-strict mode) or `undefined` (in strict mode).

What are JavaScript promises and how do they work?

Promises represent the eventual completion (or failure) of an asynchronous operation and its resulting value. They have `then`, `catch`, and `finally` methods to handle success, failure, and cleanup.

Explain the concept of prototypes and inheritance in JavaScript.

Prototypes are objects from which other objects inherit properties and methods. JavaScript uses prototype-based inheritance, meaning objects can inherit from other objects.

What is the difference between synchronous and asynchronous programming in JavaScript?

Synchronous code is executed in sequence, each operation waiting for the previous one to complete. Asynchronous code allows operations to run in parallel or to defer execution until certain conditions are met, improving performance.

What is the purpose of the `async` and `await` keywords in JavaScript?

`async` is used to define a function that returns a promise, and `await` is used to pause execution until the promise is resolved, allowing for easier and cleaner asynchronous code.

JavaScript and ReactJS Interview Questions

JavaScript - Advanced

How do you handle memory leaks in JavaScript?

By avoiding global variables, clearing intervals and timeouts, removing event listeners when no longer needed, and using tools to monitor memory usage.

Explain the concept of the event loop and call stack in JavaScript.

The event loop continuously checks the call stack and task queue. If the stack is empty, it moves the first task from the queue to the stack for execution.

What are generators and how are they used in JavaScript?

Generators are functions that can pause their execution using ``yield`` and can be resumed later, allowing for the implementation of lazy sequences and asynchronous control flows.

What is the difference between deep and shallow copying in JavaScript?

A shallow copy duplicates an object's properties without copying nested objects, whereas a deep copy recursively duplicates all nested objects, resulting in an entirely separate object.

How do you optimize performance in JavaScript applications?

Techniques include minimizing DOM access, debouncing or throttling events, using efficient data structures, code splitting, and lazy loading of assets.

JavaScript and ReactJS Interview Questions

ReactJS - Basic

What is ReactJS and why is it used?

ReactJS is a JavaScript library for building user interfaces, particularly single-page applications, by creating reusable UI components. It's used for its efficiency in updating and rendering components.

What are components in React?

Components are the building blocks of a React application. They can be functional or class-based and represent parts of the user interface that can be reused and composed to create complex UIs.

What is the difference between a class component and a functional component?

Class components are ES6 classes that extend `React.Component` and can use lifecycle methods. Functional components are functions that accept props and return JSX, and can use hooks for state and lifecycle features.

Explain the concept of state and props in React.

`state` is an object managed within a component to keep track of data that affects the rendering of the component. `props` are inputs to components that allow data to be passed from parent to child components.

What is JSX and how does it differ from HTML?

JSX is a syntax extension for JavaScript that looks similar to HTML but allows you to write HTML-like code within JavaScript. It is transformed into `React.createElement` calls that create React elements.

JavaScript and ReactJS Interview Questions

ReactJS - Intermediate

What is the Virtual DOM and how does it work in React?

The Virtual DOM is an in-memory representation of the real DOM. React uses it to efficiently update the UI by computing the minimal set of changes to apply to the actual DOM.

What are React hooks and name a few common ones?

Hooks are functions that let you use state and other React features in functional components. Common hooks include `useState`, `useEffect`, `useContext`, `useReducer`, and `useRef`.

How does React handle form inputs and state?

React handles form inputs using controlled components, where form elements' values are derived from state and updated via event handlers. This provides direct control over the form's behavior.

What is the Context API and when would you use it?

The Context API provides a way to pass data through the component tree without having to pass props down manually at every level. It's useful for global state management like themes or user authentication.

How do you handle side effects in React?

Side effects are handled using the `useEffect` hook in functional components or lifecycle methods like `componentDidMount` and `componentDidUpdate` in class components.

JavaScript and ReactJS Interview Questions

ReactJS - Advanced

Explain the concept of Higher-Order Components (HOCs) in React.

HOCs are functions that take a component and return a new component with additional props or functionality.

They are used for code reuse and composition.

What is server-side rendering (SSR) and how does it work with React?

SSR is the process of rendering a React component on the server and sending the HTML to the client. This improves initial load times and SEO. Frameworks like Next.js facilitate SSR with React.

How do you optimize performance in a React application?

Techniques include memoizing components with `React.memo`, using the `useCallback` and `useMemo` hooks, lazy loading components with `React.lazy` and `Suspense`, and minimizing re-renders.

What are React portals and how do you use them?

React portals allow you to render components outside of the current DOM hierarchy. They are useful for modal dialogs, tooltips, and other components that need to break out of the parent component's styling.

How do you manage global state in a React application?

Global state can be managed using the Context API, state management libraries like Redux or MobX, or hooks like `useReducer` combined with `useContext`.