

10 JavaScript Interview Questions



What is JavaScript, and what are its main features?	2
What is the difference between let, const, and var?	2
What is the difference between synchronous and asynchronous JavaScript, and how do you handle asynchronous operations?	3
What is a closure, and how can you use it in JavaScript?	3
What is the event loop, and how does it work in JavaScript?	4
What is the difference between == and === in JavaScript?	4
What is the purpose of the this keyword in JavaScript, and how does it work?	5
What are JavaScript closures, and how do they work?	5
How does inheritance work in JavaScript, and what is prototypal inheritance?	6
What are some of the common design patterns used in JavaScript, and why are they useful?	6

What is JavaScript, and what are its main features?

Response: JavaScript is a high-level, interpreted programming language that is used to create interactive web pages and applications. Its main features include dynamic typing, object-oriented programming, and support for functional programming.

Explanation: This question tests the candidate's basic understanding of what JavaScript is and what makes it different from other programming languages. A good response should mention some of the language's key features and how they enable web developers to create rich, interactive web applications.

What is the difference between let, const, and var?

Response: let and const are block-scoped variables introduced in ES6, while var is a function-scoped variable that has been available since earlier versions of JavaScript. let allows for the reassignment of values, while const does not.

Explanation: This question tests the candidate's knowledge of the different variable declarations in JavaScript and their scoping rules. A good response should demonstrate an understanding of how let, const, and var differ from each other and the implications of their scoping rules.

What is the difference between synchronous and asynchronous JavaScript, and how do you handle asynchronous operations?

Response: Synchronous JavaScript executes code in a blocking manner, while asynchronous JavaScript allows code to execute non-blockingly. To handle asynchronous operations, developers can use callbacks, promises, or async/await syntax.

Explanation: This question tests the candidate's understanding of how JavaScript handles asynchronous operations and how to work with them effectively. A good response should demonstrate an understanding of the different asynchronous patterns available in JavaScript and when to use each one.

What is a closure, and how can you use it in JavaScript?

Response: A closure is a function that has access to variables in its outer lexical environment. Closures can be used to create private variables and functions that are not accessible outside of the closure.

Explanation: This question tests the candidate's understanding of closures and their practical applications in JavaScript. A good response should explain how closures work and how they can be used to encapsulate code and data within a function.

What is the event loop, and how does it work in JavaScript?

Response: The event loop is a mechanism that allows JavaScript to execute code asynchronously by continuously checking for new events in the event queue. When an event is detected, the corresponding callback function is executed.

Explanation: This question tests the candidate's understanding of how JavaScript executes code asynchronously using the event loop. A good response should explain the basic mechanics of the event loop and how it allows JavaScript to handle non-blocking I/O operations.

What is the difference between == and === in JavaScript?

Response: == is a loose equality operator that compares the values of two operands for equality, while === is a strict equality operator that compares the values and types of two operands for equality.

Explanation: This question tests the candidate's understanding of the differences between loose and strict equality in JavaScript. A good response should explain the implications of using == versus === in different situations and demonstrate a clear understanding of the differences between the two operators.

What is the purpose of the this keyword in JavaScript, and how does it work?

Response: The this keyword refers to the object that a function is a method of. It can also be used to refer to the global object or to create new objects using constructor functions.

Explanation: This question tests the candidate's understanding of the this keyword in JavaScript and how it works in different contexts. A good response should explain the basic mechanics of this and how it can be used to reference different objects depending on the context in which it is used.

What are JavaScript closures, and how do they work?

Response: JavaScript closures are functions that have access to variables in their outer lexical scope, even after the outer function has returned. This is possible because JavaScript functions create a new execution context each time they are called, and each context has access to the variables in its outer scope.

Explanation: This question tests the candidate's understanding of closures and their inner workings in JavaScript. A good response should explain how closures allow functions to maintain access to their outer scope even after the outer function has returned, and provide examples of how closures can be used in practice.

How does inheritance work in JavaScript, and what is prototypal inheritance?

Response: Inheritance in JavaScript is achieved through the use of prototype chains, where objects can inherit properties and methods from other objects. Prototypal inheritance is a type of inheritance model in which objects inherit directly from other objects, rather than through the use of classes.

Explanation: This question tests the candidate's understanding of inheritance and prototype chains in JavaScript. A good response should explain how inheritance works in JavaScript and how prototypal inheritance differs from other inheritance models. The candidate should also provide examples of how prototype chains can be used to create more efficient and maintainable code.

What are some of the common design patterns used in JavaScript, and why are they useful?

Response: Some common design patterns used in JavaScript include the module pattern, the singleton pattern, and the observer pattern. These patterns are useful because they provide developers with reusable, modular code that can be easily maintained and extended.

Explanation: This question tests the candidate's knowledge of common design patterns in JavaScript and their practical applications. A good response should provide examples of how these patterns can be used to create more efficient and

maintainable code, and explain why they are useful for JavaScript developers.