

JavaScript Interview Questions with Answers (1-100)

JavaScript Interview Questions with Answers (1-100)

1. What is JavaScript?

JavaScript is a high-level, interpreted scripting language used to create dynamic and interactive web content. It runs in the browser and can also be used on the server using Node.js.

2. What are the data types in JavaScript?

JavaScript has primitive types: string, number, boolean, undefined, null, bigint, and symbol, and non-primitive types like object and array.

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

var is function-scoped, let and const are block-scoped. const cannot be reassigned, while let can.

4. What is hoisting in JavaScript?

Hoisting is JavaScript's default behavior of moving declarations to the top of the current scope before code execution.

5. What is a closure?

A closure is a function that retains access to its lexical scope even when the function is executed outside that scope.

6. What is the difference between == and ===?

== checks for value equality with type coercion, while === checks for strict equality (value and type).

7. What is the use of this keyword in JavaScript?

this refers to the object that is executing the current function. Its value depends on how the function is called.

8. What is an IIFE (Immediately Invoked Function Expression)?

An IIFE is a function that runs as soon as it is defined. Syntax: (function(){ /* code */ })();

9. What is event bubbling and capturing?

Event bubbling starts from the innermost element and bubbles up, while capturing starts from the outermost and goes inward.

10. What are arrow functions?

Arrow functions are a concise syntax for writing functions and do not bind their own this.

11. What is the difference between null and undefined?

undefined means a variable has been declared but not assigned, while null is an assignment value representing "no value".

12. What are template literals in JavaScript?

Template literals allow embedding expressions using backticks (`) and \${expression} syntax for string interpolation.

13. What is the spread operator in JavaScript?

The spread operator ... is used to expand iterable elements like arrays or objects into individual elements.

14. What is the rest parameter?

Rest parameters ...args allow a function to accept an indefinite number of arguments as an array.

15. What is a callback function?

A callback is a function passed into another function as an argument to be executed later.

16. What is the difference between synchronous and asynchronous code?

Synchronous code runs sequentially, blocking the next line until finished.
Asynchronous code can run in the background without blocking.

17. What are Promises in JavaScript?

Promises represent a value that may be available now, later, or never. They help handle async operations.

18. What are `async/await` keywords used for?

They simplify writing asynchronous code. `await` pauses execution until the promise resolves.

19. What is an event loop in JavaScript?

The event loop handles asynchronous tasks and coordinates between the call stack and callback queue.

20. What is the call stack?

The call stack is a mechanism to keep track of function execution context in a LIFO (Last In First Out) order.

21. What are higher-order functions?

Functions that take other functions as arguments or return functions are called higher-order functions.

22. What is the difference between `map()`, `filter()` and `forEach()`?

- `map()` returns a new array by transforming each element.
- `filter()` returns a new array of elements that meet a condition.
- `forEach()` just iterates without returning anything.

23. What is destructuring in JavaScript?

Destructuring allows unpacking values from arrays or properties from objects into distinct variables.

24. What is the `typeof` operator?

`typeof` returns the type of a variable, like '`number`', '`string`', or '`object`'.

25. What is `Nan` in JavaScript?

`Nan` stands for "Not a Number" and indicates an invalid number result, e.g., from `parseInt("abc")`.

26. What is the difference between `slice()` and `splice()`?

- `slice()` returns a shallow copy without modifying the array.
- `splice()` modifies the original array by removing or adding elements.

27. What is a pure function?

A pure function always returns the same output for the same input and has no side effects.

28. What is memoization?

Memoization is an optimization technique that caches the result of expensive function calls for faster reuse.

29. What is the purpose of bind() method?

bind() returns a new function with a specified this value and optional parameters.

30. What are truthy and falsy values?

Truthy values evaluate to true in a Boolean context; falsy values include false, 0, '', null, undefined, and NaN.

31. What is the difference between deep copy and shallow copy?

A shallow copy copies only references to nested objects, while a deep copy duplicates all levels of the object.

32. How do you clone an object in JavaScript?

Use Object.assign({}, obj) for shallow copy or JSON.parse(JSON.stringify(obj)) for deep copy.

33. What is the use of setTimeout()?

setTimeout() executes a function after a specified delay (in milliseconds).

34. What is the use of setInterval()?

setInterval() repeatedly calls a function with a fixed time delay between each call.

35. What are generators in JavaScript?

Generators are special functions that can pause (yield) and resume their execution, using the function* syntax.

36. What is the difference between call(), apply(), and bind()?

- call() invokes a function with arguments individually.

- `apply()` uses an array of arguments.
- `bind()` returns a new function with this bound.

37. What is scope in JavaScript?

Scope determines the accessibility of variables – either globally, functionally, or block-wise.

38. What is lexical scope?

Lexical scope means a function can access variables defined in its outer (parent) scope.

39. What is the Temporal Dead Zone (TDZ)?

TDZ is the time between block start and `let/const` declaration where the variable cannot be accessed.

40. What are the different ways to define a function in JavaScript?

Functions can be declared as function declarations, function expressions, and arrow functions.

41. What is object destructuring?

Object destructuring allows extracting properties from an object and assigning them to variables.

42. What is array destructuring?

Array destructuring lets you assign elements of an array to variables in a single statement.

43. What is the difference between function declaration and function expression?

Function declarations are hoisted, while function expressions are not.

44. What are default parameters?

Default parameters allow initializing function parameters with default values if no value is passed.

45. What is the DOM?

DOM (Document Object Model) represents the structure of a webpage as a tree of objects.

46. How do you select an element in the DOM?

Use `getElementById`, `querySelector`, `getElementsByClassName`, etc., to select DOM elements.

47. What is event delegation?

Event delegation allows handling events at a parent level instead of attaching listeners to each child.

48. What is the difference between `e.preventDefault()` and `e.stopPropagation()`?

`preventDefault()` stops default browser behavior, while `stopPropagation()` stops the event from bubbling up.

49. What are custom events in JavaScript?

Custom events allow developers to create and dispatch their own events using `CustomEvent`.

50. What is `localStorage` in JavaScript?

`localStorage` stores key-value pairs in the browser that persist even after the page reloads or closes.

51. What is `sessionStorage` in JavaScript?

`sessionStorage` stores key-value pairs for the duration of a page session and clears when the tab or window is closed.

52. What is the difference between `localStorage` and `sessionStorage`?

`localStorage` data persists even after closing the browser, while `sessionStorage` is cleared when the session ends.

53. How do you check if a variable is an array?

Use `Array.isArray(variable)` to check if a value is an array.

54. What is the difference between `Object.freeze()` and `Object.seal()`?

`Object.freeze()` prevents adding, removing, or modifying properties.

`Object.seal()` only prevents adding or removing properties.

55. What is the `typeof null` in JavaScript?

`typeof null` returns 'object' due to a historical bug in JavaScript.

56. What is the use of `Promise.all()`?

`Promise.all()` runs multiple promises in parallel and returns a single promise when all have resolved or one fails.

57. What is a ternary operator?

It's a shorthand for `if...else: condition ? trueValue : falseValue.`

58. What are template strings?

Template strings use backticks (`) and allow embedded expressions using `${}.`

59. How do you convert a string to a number in JavaScript?

Use `Number()`, `parseInt()`, or `parseFloat()` to convert strings to numbers.

60. How do you convert a number to a string in JavaScript?

Use `.toString()` or `String()` to convert numbers to strings.

61. What are falsy values in JavaScript?

Falsy values include `false`, `0`, `''`, `null`, `undefined`, and `Nan`.

62. What are template engines?

Template engines like Handlebars or EJS allow generating HTML dynamically using JS and template syntax.

63. What are JavaScript modules?

Modules are reusable pieces of code that can be imported/exported between files using `import` and `export`.

64. What is the difference between client-side and server-side JavaScript?

Client-side JS runs in the browser, server-side JS (like Node.js) runs on the server.

65. What are tagged template literals?

Tagged templates allow you to parse template literals with a function for custom processing.

66. What is a Symbol in JavaScript?

Symbols are unique and immutable primitive values used as unique object property keys.

67. What is optional chaining (?.) in JavaScript?

It allows safely accessing deeply nested properties without throwing errors if an intermediate value is null or undefined.

68. What is nullish coalescing (??) operator?

It returns the right-hand value only if the left-hand value is null or undefined.

69. How do you debounce a function in JavaScript?

Debouncing delays a function call until after a pause in calls, typically using setTimeout and clearTimeout.

70. What is throttling in JavaScript?

Throttling ensures a function runs at most once in a specified time interval, useful in scroll or resize events.

71. What is hoisting in JavaScript?

Hoisting is JavaScript's behavior of moving declarations to the top of the current scope before execution.

72. What gets hoisted in JavaScript?

Variable declarations (var) and function declarations are hoisted, but let and const are not initialized.

73. What is currying in JavaScript?

Currying transforms a function with multiple arguments into a series of functions with one argument each.

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

== compares values with type coercion, === compares both value and type (strict equality).

75. What are closures in JavaScript?

A closure is a function that remembers variables from its outer scope even after the outer function has finished executing.

76. What are arrow functions in JavaScript?

Arrow functions are shorter syntax for functions and do not have their own this or arguments binding.

77. What is the arguments object in JavaScript?

arguments is an array-like object accessible inside regular functions containing all passed arguments.

78. What is an Immediately Invoked Function Expression (IIFE)?

An IIFE is a function that runs as soon as it is defined, often used to create a private scope.

79. What are data types in JavaScript?

There are 8 data types: string, number, bigint, boolean, undefined, null, symbol, and object.

80. What is event bubbling in JavaScript?

Event bubbling is the process where an event starts from the deepest target element and bubbles up to its parents.

81. What is event capturing?

Event capturing is the opposite of bubbling, where the event is handled from the outermost element down to the target.

82. What is a WeakMap in JavaScript?

WeakMap is a collection of key-value pairs where keys are objects and values can be garbage collected if no references exist.

83. What is a WeakSet in JavaScript?

WeakSet is a collection of objects that are weakly referenced, meaning they can be garbage collected.

84. What is the global object in JavaScript?

In browsers, the global object is window; in Node.js, it's global.

85. What is the difference between window and document?

window represents the browser window, while document represents the loaded HTML content.

86. How does async function return a promise?

An async function always returns a promise, even if the return value is not explicitly a promise.

87. What are static methods in JavaScript?

Static methods are defined on a class itself, not instances, and are called using ClassName.method().

88. What is the use of super in JavaScript?

super is used to call the constructor or methods of a parent class from a child class.

89. What is prototype in JavaScript?

Every object has a prototype from which it can inherit methods and properties.

90. What is prototypal inheritance?

It's a feature where objects inherit properties and methods from another object via their prototype.

91. What is a factory function in JavaScript?

A factory function returns a new object every time it's called, without using classes or constructors.

92. What is the difference between Object.create() and new keyword?

Object.create() sets the prototype directly, while new invokes a constructor function.

93. What is the spread syntax used for objects?

It allows copying and merging objects: { ...obj1, ...obj2 }.

94. How do you check if a property exists in an object?

Use the in operator or hasOwnProperty() method.

95. What is the difference between for...in and for...of?

for...in iterates over object keys, for...of iterates over iterable values like arrays.

96. What are iterable objects in JavaScript?

Objects that implement the [Symbol.iterator] method and can be looped with for...of.

97. What is a callback hell?

Callback hell occurs when multiple nested callbacks make code hard to read and maintain.

98. How do you avoid callback hell?

Use Promises or async/await to handle asynchronous operations cleanly.

99. What is the difference between innerHTML and textContent?

innerHTML parses HTML tags, while textContent only sets or gets plain text.

100. What is the purpose of eval() in JavaScript?

eval() executes a string as JavaScript code, but it's discouraged due to security and performance issues.