

JavaScript Interview Questions and Answer

1. What is JavaScript?

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications and is primarily used to create interactive effects within web browsers.

2. What are data types in JavaScript?

JavaScript has six primitive data types:

- String
- Number
- Boolean
- Undefined
- Null
- Symbol (ES6)

Additionally, there is the Object type which includes arrays, functions, and dates.

3. What is var, let, and const?

- var: Function-scoped and can be redeclared and updated.
- let: Block-scoped and can be updated but not redeclared.
- const: Block-scoped, cannot be redeclared or updated.

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

- == compares values with type coercion.
- === compares both value and type strictly (without type coercion).

5. What are closures in JavaScript?

A closure is a function that remembers its outer scope even when the function is executed outside its lexical scope.

Code-

```
function outer() {  
  let count = 0;  
  return function inner() {  
    count++;  
    console.log(count);  
  };  
}
```

```
}  
  
const closure = outer();  
  
closure(); // 1  
  
closure(); // 2
```

6. What is hoisting in JavaScript?

Hoisting is a JavaScript mechanism where variables and function declarations are moved to the top of their scope before code execution.

7. Explain event delegation in JavaScript.

Event delegation allows you to add a single event listener to a parent element that covers all its children by utilizing event bubbling.

8. What is a callback function?

A callback function is a function passed as an argument to another function and is executed after some operation is completed.

9. What is the difference between null and undefined?

- null: Explicitly represents an empty or non-existent value.
- undefined: Represents a variable that has been declared but not assigned a value.

10. What is NaN in JavaScript?

NaN stands for "Not-a-Number." It is a property of the global object and is returned when a mathematical operation cannot be performed.

11. What are arrow functions in JavaScript?

Arrow functions provide a shorter syntax for writing functions and do not have their own this context.

Code-

```
const sum = (a, b) => a + b;
```

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

- call(): Invokes a function with a given this value and arguments passed individually.
- apply(): Invokes a function with a given this value and arguments passed as an array.
- bind(): Returns a new function with a bound this value.

13. What is the use of async and await?

async and await allow asynchronous code to be written in a synchronous style, making code easier to read.

Code-

```
async function fetchData() {
```

```
let response = await fetch('url');  
let data = await response.json();  
}
```

14. What are Promises in JavaScript?

A promise is an object that represents the eventual completion (or failure) of an asynchronous operation.

15. What is the purpose of the Array.map() method?

The map() method creates a new array by applying a function to each element of the array.

16. Explain Array.filter() and Array.reduce().

- filter(): Creates a new array with elements that pass a test.
- reduce(): Reduces an array to a single value by executing a function on each element.

17. What is the this keyword?

this refers to the object from which the function is called. In arrow functions, this is lexically scoped and refers to the context in which the function was defined.

18. What is a prototype in JavaScript?

Every JavaScript object has a prototype, which is another object from which it inherits properties and methods.

19. What is prototypal inheritance?

Prototypal inheritance is a feature of JavaScript where an object can inherit properties and methods from another object via its prototype.

20. Explain Array.splice() and Array.slice().

- splice(): Changes the contents of an array by removing or replacing existing elements and/or adding new elements.
- slice(): Returns a shallow copy of a portion of an array into a new array.

21. What is the difference between synchronous and asynchronous programming?

- Synchronous: Operations are executed sequentially, blocking further execution until the current operation finishes.
- Asynchronous: Operations do not block the execution and can be handled with callbacks, promises, or async/await.

22. What is the spread operator?

The spread operator (...) expands an iterable (like an array or object) into its individual elements.

Code-

```
const arr = [1, 2, 3];
```

```
console.log(...arr); // 1 2 3
```

23. What is the rest parameter?

The rest parameter (...) allows you to represent an indefinite number of arguments as an array.

Code-

```
function sum(...args) {  
  return args.reduce((a, b) => a + b, 0);  
}
```

24. What is typeof in JavaScript?

typeof is used to determine the data type of a variable.

Code-

```
console.log(typeof 42); // "number"
```

25. Explain the difference between forEach() and map().

- `forEach()`: Iterates over an array but does not return a new array.
- `map()`: Returns a new array with the results of calling a function on every element.

26. What is the DOM?

The Document Object Model (DOM) is a programming interface for web documents. It represents the page so that programs can change the structure, style, and content.

27. What are higher-order functions?

A higher-order function is a function that takes another function as an argument or returns a function.

28. What is setTimeout and setInterval?

- `setTimeout()`: Executes a function after a specified delay.
- `setInterval()`: Repeatedly executes a function with a fixed time delay between each call.

29. What is a Promise.all()?

`Promise.all()` takes an array of promises and returns a single promise that resolves when all the promises in the array are resolved.

30. What is use strict in JavaScript?

`use strict` is a directive that enables strict mode, which helps in writing secure JavaScript by throwing more errors for potentially unsafe actions.

31. Explain the Object.assign() method.

`Object.assign()` is used to copy the values of all enumerable properties from one or more source objects to a target object.

32. What are modules in JavaScript?

Modules allow you to break code into reusable pieces and import/export them between files.

33. What is destructuring in JavaScript?

Destructuring allows you to extract values from arrays or properties from objects into distinct variables.

Code-

```
const { name, age } = { name: "John", age: 30 };
```

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

TDZ is a behavior where variables declared with `let` or `const` are inaccessible until the code execution reaches their declaration.

35. What is `eval()` in JavaScript?

`eval()` is a function that executes a string of JavaScript code. It should be avoided due to security risks.

36. What are IIFEs (Immediately Invoked Function Expressions)?

IIFEs are functions that are executed right after they are created.

Code-

```
(function () {  
  console.log("IIFE");  
})();
```

37. What are generators in JavaScript?

Generators are functions that can be paused and resumed, allowing asynchronous-like behavior.

38. What is tail call optimization?

Tail call optimization occurs when the last action of a function is calling another function, and the current function's stack frame is replaced instead of added.

39. What are function expressions?

A function expression defines a function inside an expression.

Code-

```
const sum = function(a, b) {  
  return a + b;  
};
```

40. What is the `instanceof` operator?

The `instanceof` operator checks if an object is an instance of a particular class.

41. What is the difference between `apply()` and `call()`?

Both call a function with a given **this** value, but `apply()` takes arguments as an array, whereas `call()` takes them individually.

42. What is memoization in JavaScript?

Memoization is an optimization technique used to store the results of expensive function calls and return the cached result when the same inputs occur again.

43. What is `Object.freeze()` in JavaScript?

`Object.freeze()` freezes an object, preventing new properties from being added, existing properties from being removed, or their values from being changed.

44. What is deep copy vs shallow copy?

- Shallow copy: Only copies the first layer of an object. Changes to nested objects affect the original.
- Deep copy: Copies all nested objects and arrays, so changes do not affect the original object.

45. What are template literals in JavaScript?

Template literals are strings that allow embedded expressions. They are enclosed by backticks (`) instead of quotes.

Code-

```
const name = "John";  
console.log(`Hello, ${name}!`);
```

46. What is `JSON.stringify()` and `JSON.parse()`?

- `JSON.stringify()`: Converts a JavaScript object into a JSON string.
- `JSON.parse()`: Converts a JSON string into a JavaScript object.

47. What is function currying in JavaScript?

Currying is the process of converting a function that takes multiple arguments into a sequence of functions that each take a single argument.

Code-

```
function curry(a) {  
  return function(b) {  
    return a + b;  
  };  
}  
  
const sum = curry(5);  
console.log(sum(3)); // 8
```

48. What are weak references in JavaScript?

Weak references allow objects to be garbage-collected even if they are still referenced by a weak reference (like WeakMap or WeakSet).

49. What is Array.from() in JavaScript?

Array.from() creates a new array from an array-like or iterable object.

50. What is debouncing in JavaScript?

Debouncing is a programming technique that ensures a function is not called too frequently by delaying its execution until after a specified delay.