

100 Most Asked JavaScript Interview QnA

Made By:



Yadneyesh (Curious Coder)
CodWithCurious.com

A large yellow square containing the letters 'JS' in a bold, black, sans-serif font.

1. What is JavaScript?

JavaScript is a high-level, interpreted programming language primarily used for adding interactivity to web pages.

2. What are the data types in JavaScript?

JavaScript has six primitive data types: string, number, boolean, null, undefined, and symbol, along with a complex data type called object.

3. What is the difference between null and undefined?

null represents the intentional absence of any object value, while undefined indicates the absence of a value or an uninitialized variable.

4. What is the DOM in JavaScript?

The Document Object Model (DOM) is a programming interface that represents the structure of HTML and XML documents. It allows JavaScript to access and manipulate the content and structure of a webpage.

5. What is an event in JavaScript?

An event is an action or occurrence that happens in the browser, such as a button click or page load. JavaScript can respond to these events by executing code in response.

6. What is an anonymous function in JavaScript?

An anonymous function is a function without a name. It can be assigned to a variable or passed as an argument to another function. They are often used for one-time or callback functions.

7. What are closures in JavaScript?

Closures are functions that have access to variables from an outer function, even after the outer function has finished executing. They encapsulate data and provide a way to maintain state between function calls.

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

The == operator checks for equality after performing type coercion, while the === operator checks for equality without type coercion, ensuring both the value and type match.

9. What is hoisting in JavaScript?

Hoisting is a JavaScript behavior where variable and function declarations are moved to the top of their containing scope during the compilation phase, allowing them to be used before they are declared.

10. What is the this keyword in JavaScript?

The this keyword refers to the object that is currently executing the code. Its value is determined by how a function is called, and it provides a way to access object properties and methods within a function.

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

Functions in JavaScript can be defined using function declarations, function expressions, arrow functions, and methods within objects.

12. What is the purpose of the let keyword in JavaScript?

The let keyword is used to declare block-scoped variables in JavaScript. Variables declared with let are only accessible within the block where they are defined.

13. What is the purpose of the const keyword in JavaScript?

The const keyword is used to declare block-scoped variables in JavaScript that cannot be re-assigned. However, it does not make objects or arrays immutable.

14. What are template literals in JavaScript?

Template literals, denoted by backticks (```), are a way to create strings in JavaScript that support interpolation of variables and multi-line strings.

15. What are JavaScript promises?

Promises are used for asynchronous programming in JavaScript. They represent the eventual completion (or failure) of an asynchronous operation and allow chaining of operations using `.then()` and `.catch()`.

16. What is the async/await syntax in JavaScript?

The `async/await` syntax is a modern approach to handle asynchronous operations. It allows writing asynchronous code in a more synchronous-like manner, making it easier to read and maintain.

17. What are arrow functions in JavaScript?

Arrow functions are a concise syntax for defining functions in JavaScript. They have a shorter syntax compared to traditional function expressions and inherit the `this` value from the enclosing scope.

18. What is event delegation in JavaScript?

Event delegation is a technique where you attach an event listener to a parent element instead of individual child elements. It allows handling events efficiently, especially for dynamically added elements.

19. What is the purpose of the map() function in JavaScript?

The `map()` function is used to create a new array by applying a given function to each element of an existing array. It allows transforming and manipulating array elements easily.

20. What is the purpose of the filter() function in JavaScript?

The `filter()` function is used to create a new array containing elements that pass a certain condition defined by a provided function. It allows filtering elements from an array based on specific criteria.

21. What is the purpose of the reduce() function in JavaScript?

The `reduce()` function is used to reduce an array to a single value by applying a function to each element and accumulating the result. It is often used to perform calculations or transformations on arrays.

22. What is a callback function in JavaScript?

A callback function is a function that is passed as an argument to another function and gets executed at a later time or in response to an event. It enables asynchronous and event-driven programming.

23. What is the difference between let and var in JavaScript?

The let keyword declares block-scoped variables, while the var keyword declares function-scoped variables. Variables declared with var are hoisted, while variables declared with let are not.

24. What are JavaScript modules?

JavaScript modules are reusable pieces of code that encapsulate related functionality. They allow for better organization, encapsulation, and code reuse in larger JavaScript applications.

25. What is object destructuring in JavaScript?

Object destructuring is a feature that allows extracting properties from objects and assigning them to variables. It provides a concise way to extract values and work with object properties.

26. What are JavaScript classes?

JavaScript classes are a way to define objects with shared properties and behaviors. They provide a template for creating multiple instances of objects with similar characteristics.

27. What is inheritance in JavaScript?

Inheritance is a mechanism in JavaScript where an object can inherit properties and methods from another object. It allows for code reuse and creating hierarchical relationships between objects.

28. What are JavaScript getters and setters?

Getters and setters are special methods used to get and set the values of object properties, respectively. They provide control over property access and enable data validation and encapsulation.

29. What is the purpose of the try/catch statement in JavaScript?

The try/catch statement is used for error handling in JavaScript. It allows catching and handling exceptions that occur during the execution of a block of code.

30. What is the difference between let and const in JavaScript?

The let keyword is used to declare variables that can be reassigned, while the const keyword is used to declare variables that are read-only and cannot be reassigned.

31. What is the purpose of the forEach() function in JavaScript?

The forEach() function is used to execute a provided function once for each element in an array. It provides an easy way to iterate over array elements and perform operations on them.

32. What is the purpose of the localStorage object in JavaScript?

The localStorage object allows web applications to store key-value pairs locally within the user's browser. It provides a simple way to store and retrieve data persistently.

33. What are arrow functions? How are they different from regular functions?

Arrow functions are a concise syntax for defining functions in JavaScript. They have a

shorter syntax compared to regular functions and do not bind their own this value.

34. What is the purpose of the `setTimeout()` function in JavaScript?

The `setTimeout()` function is used to schedule the execution of a function after a specified delay in milliseconds. It allows adding time-based delays to JavaScript code.

35. What is event bubbling in JavaScript?

Event bubbling is a mechanism in which an event triggered on a specific element will also trigger the same event on all of its parent elements. It starts from the innermost element and propagates upwards in the DOM tree.

36. What is the purpose of the `fetch()` function in JavaScript?

The `fetch()` function is used to make HTTP requests and fetch resources from the network. It provides a modern and flexible way to perform asynchronous network requests.

37. What is the difference between null and undefined?

`null` is an explicitly assigned value that represents the absence of an object, while `undefined` is a value assigned by the JavaScript engine to variables that have been declared but have not been assigned a value.

38. What is event propagation in JavaScript?

Event propagation is the process of an event being triggered on an element and then propagating to its parent elements or capturing down from its parent elements. It allows handling events at different levels of the DOM hierarchy.

39. What is the purpose of the `Object.keys()` function in JavaScript?

The `Object.keys()` function is used to extract all the keys of an object and return them as an array. It provides an easy way to iterate over an object's properties.

40. What is the difference between null and undefined in JavaScript?

`null` is an assigned value that represents the intentional absence of an object value, while `undefined` represents an uninitialized or undefined value, often used as a default initial value.

41. What is the purpose of the `addEventListener()` method in JavaScript?

The `addEventListener()` method is used to attach an event listener to an element. It allows you to listen for specific events and execute a function when the event is triggered.

42. What is the purpose of the `parentNode` property in JavaScript?

The `parentNode` property is used to access the parent node of an element in the DOM. It allows traversal and manipulation of the DOM tree by accessing the immediate parent of an element.

43. What is the purpose of the `querySelector()` method in JavaScript?

The `querySelector()` method is used to select the first element that matches a specified CSS selector. It provides a powerful way to retrieve elements from the DOM based on CSS selectors.

44. What is the purpose of the `querySelectorAll()` method in JavaScript?

The `querySelectorAll()` method is used to select all elements that match a specified CSS selector. It returns a collection of elements that can be iterated over or accessed using indexing.

45. **What is the difference between `querySelector()` and `getElementById()`?**
`querySelector()` is a more versatile method that allows selecting elements based on any CSS selector, while `getElementById()` is specifically used to select an element by its unique id attribute.
46. **What is the difference between function declarations and function expressions in JavaScript?**
Function declarations are hoisted and can be called before they are defined, while function expressions are not hoisted and must be defined before they are called.
47. **What is the purpose of the `bind()` method in JavaScript?**
The `bind()` method is used to create a new function with a specified `this` value and initial arguments. It allows explicit binding of the `this` value within a function.
48. **What is the purpose of the `call()` method in JavaScript?**
The `call()` method is used to invoke a function with a specified `this` value and arguments provided individually. It allows borrowing methods from other objects and explicit invocation of functions.
49. **What is the purpose of the `apply()` method in JavaScript?**
The `apply()` method is used to invoke a function with a specified `this` value and arguments provided as an array or an array-like object. It allows borrowing methods from other objects and explicit invocation of functions.
50. **What is the purpose of the `Array.isArray()` method in JavaScript?**
The `Array.isArray()` method is used to determine whether a given value is an array or not. It returns `true` if the value is an array, and `false` otherwise.
51. **What is event capturing in JavaScript?**
Event capturing is the process of an event being triggered on an element's parent elements first, before reaching the target element. It allows capturing events at the outermost level of the DOM hierarchy.
52. **What is event delegation in JavaScript?**
Event delegation is a technique where you attach an event listener to a parent element instead of individual child elements. It allows handling events efficiently, especially for dynamically added elements.
53. **What is the purpose of the `startsWith()` method in JavaScript?**
The `startsWith()` method is used to check if a string starts with a specified substring. It returns `true` if the string starts with the substring, and `false` otherwise.
54. **What is the purpose of the `endsWith()` method in JavaScript?**
The `endsWith()` method is used to check if a string ends with a specified substring. It returns `true` if the string ends with the substring, and `false` otherwise.
55. **What is the purpose of the `includes()` method in JavaScript?**
The `includes()` method is used to check if a string contains a specified substring. It returns `true` if the substring is found, and `false` otherwise.
56. **What is the purpose of the `padStart()` method in JavaScript?**
The `padStart()` method is used to pad the beginning of a string with a specified character until it reaches a desired length. It is often used for formatting purposes.
57. **What is the purpose of the `padEnd()` method in JavaScript?**
The `padEnd()` method is used to pad the end of a string with a specified character

until it reaches a desired length. It is often used for formatting purposes.

58. What is the purpose of the `charAt()` method in JavaScript?

The `charAt()` method is used to retrieve the character at a specified index in a string. It returns the character at the specified index or an empty string if the index is out of range.

59. What is the purpose of the `charCodeAt()` method in JavaScript?

The `charCodeAt()` method is used to retrieve the Unicode value of the character at a specified index in a string. It returns the Unicode value of the character or NaN if the index is out of range.

60. What is the purpose of the `String.fromCharCode()` method in JavaScript?

The `String.fromCharCode()` method is used to create a string from a sequence of Unicode values. It allows converting Unicode values to their corresponding characters.

61. What is the purpose of the `JSON.stringify()` method in JavaScript?

The `JSON.stringify()` method is used to convert a JavaScript object or value to a JSON string. It is commonly used for data serialization and communication with web servers.

62. What is the purpose of the `JSON.parse()` method in JavaScript?

The `JSON.parse()` method is used to parse a JSON string and convert it into a JavaScript object or value. It is commonly used to deserialize JSON data received from a server.

63. What is the purpose of the `encodeURIComponent()` function in JavaScript?

The `encodeURIComponent()` function is used to encode special characters in a URL component. It ensures that the component can be included in a URL without causing any parsing errors.

64. What is the purpose of the `decodeURIComponent()` function in JavaScript?

The `decodeURIComponent()` function is used to decode URL-encoded components. It converts URL-encoded characters back to their original form.

65. What is the purpose of the `Math.random()` function in JavaScript?

The `Math.random()` function is used to generate a random floating-point number between 0 (inclusive) and 1 (exclusive). It is often used to introduce randomness in JavaScript programs.

66. What is the purpose of the `Math.floor()` function in JavaScript?

The `Math.floor()` function is used to round a number down to the nearest integer. It removes the decimal part of the number and returns the largest integer less than or equal to the given number.

67. What is the purpose of the `Math.ceil()` function in JavaScript?

The `Math.ceil()` function is used to round a number up to the nearest integer. It increases the number to the next higher integer, regardless of the decimal part.

68. What is the purpose of the `Math.round()` function in JavaScript?

The `Math.round()` function is used to round a number to the nearest integer. It rounds the number up or down based on the decimal part.

69. What is the purpose of the `Math.max()` function in JavaScript?

The `Math.max()` function is used to find the largest number among a list of

arguments. It returns the highest value passed as an argument.

70. What is the purpose of the `Math.min()` function in JavaScript?

The `Math.min()` function is used to find the smallest number among a list of arguments. It returns the lowest value passed as an argument.

71. What is the purpose of the `Math.pow()` function in JavaScript?

The `Math.pow()` function is used to calculate the power of a number. It takes a base and an exponent as arguments and returns the result of raising the base to the exponent.

72. What is the purpose of the `Math.sqrt()` function in JavaScript?

The `Math.sqrt()` function is used to calculate the square root of a number. It returns the positive square root of the given number.

73. What is the purpose of the `Math.abs()` function in JavaScript?

The `Math.abs()` function is used to calculate the absolute value of a number. It returns the magnitude of the number without considering its sign.

74. What is the purpose of the `Math.floor()` and `Math.random()` functions together?

By combining `Math.floor()` and `Math.random()` functions, you can generate a random integer within a specified range. For example, `Math.floor(Math.random() * 10)` generates a random integer between 0 and 9.

75. What is the purpose of the `Date()` constructor in JavaScript?

The `Date()` constructor is used to create a new JavaScript Date object that represents a specific date and time. It allows working with dates and performing various operations on them.

76. What is the purpose of the `getFullYear()` method in JavaScript Date objects?

The `getFullYear()` method is used to retrieve the four-digit year value of a JavaScript Date object. It returns the year as a four-digit number, such as 2023.

77. What is the purpose of the `getMonth()` method in JavaScript Date objects?

The `getMonth()` method is used to retrieve the month value of a JavaScript Date object. It returns a zero-based index, where January is represented by 0 and December by 11.

78. What is the purpose of the `getDate()` method in JavaScript Date objects?

The `getDate()` method is used to retrieve the day of the month value of a JavaScript Date object. It returns the day as a number between 1 and 31.

79. What is the purpose of the `getDay()` method in JavaScript Date objects?

The `getDay()` method is used to retrieve the day of the week value of a JavaScript Date object. It returns a zero-based index, where Sunday is represented by 0 and Saturday by 6.

80. What is the purpose of the `getHours()` method in JavaScript Date objects?

The `getHours()` method is used to retrieve the hour value of a JavaScript Date object. It returns the hour as a number between 0 and 23.

81. What is the purpose of the `getMinutes()` method in JavaScript Date objects?

The `getMinutes()` method is used to retrieve the minute value of a JavaScript Date object. It returns the minute as a number between 0 and 59.

82. What is the purpose of the `getSeconds()` method in JavaScript Date objects?

The `getSeconds()` method is used to retrieve the second value of a JavaScript Date

object. It returns the second as a number between 0 and 59.

83. What is the purpose of the `getFullYear()` and `getMonth()` methods together in JavaScript Date objects?

By combining the `getFullYear()` and `getMonth()` methods, you can retrieve the full date in a human-readable format. For example, `const currentDate = new Date(); const year = currentDate.getFullYear(); const month = currentDate.getMonth(); console.log(year + '-' + (month + 1));` will print the current year and month in the format 'YYYY-MM'.

84. What is the purpose of the `setFullYear()` method in JavaScript Date objects?

The `setFullYear()` method is used to set the year value of a JavaScript Date object. It allows modifying the year component of a date.

85. What is the purpose of the `setMonth()` method in JavaScript Date objects?

The `setMonth()` method is used to set the month value of a JavaScript Date object. It allows modifying the month component of a date.

86. What is the purpose of the `setDate()` method in JavaScript Date objects?

The `setDate()` method is used to set the day of the month value of a JavaScript Date object. It allows modifying the day component of a date.

87. What is the purpose of the `setHours()` method in JavaScript Date objects?

The `setHours()` method is used to set the hour value of a JavaScript Date object. It allows modifying the hour component of a date.

88. What is the purpose of the `setMinutes()` method in JavaScript Date objects?

The `setMinutes()` method is used to set the minute value of a JavaScript Date object. It allows modifying the minute component of a date.

89. What is the purpose of the `setSeconds()` method in JavaScript Date objects?

The `setSeconds()` method is used to set the second value of a JavaScript Date object. It allows modifying the second component of a date.

90. What is the purpose of the `toLocaleString()` method in JavaScript Date objects?

The `toLocaleString()` method is used to convert a JavaScript Date object to a localized string representation based on the current locale. It considers the user's time zone and regional settings to format the date and time.

91. What is the purpose of the `toDateString()` method in JavaScript Date objects?

The `toDateString()` method is used to convert the date portion of a JavaScript Date object to a human-readable string representation. It returns the date in the format 'Day Mon DD YYYY', such as 'Thu Jun 24 2023'.

92. What is the purpose of the `getTime()` method in JavaScript Date objects?

The `getTime()` method is used to retrieve the timestamp value of a JavaScript Date object. It returns the number of milliseconds elapsed since January 1, 1970, 00:00:00 UTC.

93. What is the purpose of the `setTime()` method in JavaScript Date objects?

The `setTime()` method is used to set the timestamp value of a JavaScript Date object. It allows modifying the date and time by providing a new timestamp.

94. What is the purpose of the `setTimeout()` function in JavaScript?

The `setTimeout()` function is used to execute a specified function or a piece of code

after a delay specified in milliseconds. It is commonly used for delaying the execution of code or creating timeouts.

95. What is the purpose of the `setInterval()` function in JavaScript?

The `setInterval()` function is used to repeatedly execute a specified function or a piece of code at a fixed interval specified in milliseconds. It is commonly used for creating intervals and timers.

96. What is the purpose of the `clearTimeout()` function in JavaScript?

The `clearTimeout()` function is used to cancel a timeout set by the `setTimeout()` function. It clears the scheduled execution of a function before it is triggered.

97. What is the purpose of the `clearInterval()` function in JavaScript?

The `clearInterval()` function is used to cancel an interval set by the `setInterval()` function. It stops the repeated execution of a function at a fixed interval.

98. What is the purpose of the `isNaN()` function in JavaScript?

The `isNaN()` function is used to check if a value is NaN (Not-a-Number). It returns true if the value is NaN, and false otherwise.

99. What is the purpose of the `isFinite()` function in JavaScript?

The `isFinite()` function is used to check if a value is a finite number. It returns true if the value is a finite number, and false otherwise. It also returns false for NaN, Infinity, and -Infinity.

100. What is the purpose of the `parseFloat()` function in JavaScript?

The `parseFloat()` function is used to parse a string and extract a floating-point number from it. It converts the initial portion of the string that represents a valid floating-point number into a number value.