Sure, here are 100 JavaScript interview questions along with short answers:

1. What is JavaScript?
   Answer: JavaScript is a high-level, interpreted programming language used to add interactivity to web pages.

2. What are the data types in JavaScript?
   Answer: JavaScript has six primitive data types: undefined, null, boolean, number, string; and one non-primitive type: object, Array.

3. What is the difference between `null` and `undefined`?
   Answer: `null` is an intentional absence of value, while `undefined` indicates a variable has been declared but not assigned a value.

4. What are the different ways to declare a variable in JavaScript?
   Answer: `var`, `let`, and `const`.

5. What is the scope of variables declared using `var`, `let`, and `const`?
   Answer: Variables declared with `var` are global, while those declared with `let` and `const` are block-scoped.

6. What is hoisting in JavaScript?
   Answer: Hoisting is a JavaScript behavior where variable and function declarations are moved to the top of their scope during compilation.

7. What is the difference between `==` and `===` operators?
   Answer: `==` compares values after type coercion, while `===` compares both values and types without coercion.

8. What is the purpose of `NaN` in JavaScript?
   Answer: `NaN` stands for "Not-a-Number" and represents the result of an invalid mathematical operation.

9. How can you check if a variable is an array in JavaScript?
   Answer: You can use the `Array.isArray()` method to check if a variable is an array.

10. What is an immediately-invoked function expression (IIFE)?
    Answer: An IIFE is a function that is defined and executed immediately after its creation.

11. What are closures in JavaScript?
    Answer: Closures are functions that have access to variables from their outer (enclosing) scope even after that scope has finished executing.

13. What is the purpose of the `this` keyword in JavaScript?
   Answer: `this` refers to the current execution context and can vary depending on how a function is called.

14. How can you change the value of `this` explicitly in a function?
   Answer: You can use `call()`, `apply()`, or `bind()` methods to change the value of `this` explicitly.

15. What is a callback function?
   Answer: A callback function is a function passed as an argument to another function to be executed later.

16. What is an arrow function in JavaScript?
   Answer: Arrow functions are a shorthand syntax for writing functions in JavaScript, and they retain the context of `this` from their surrounding code.

17. Explain the difference between `let`, `const`, and `var`.
   Answer: `let` and `const` are block-scoped and cannot be re-declared within the same scope, while `var` is function-scoped and can be re-declared.

18. How do you create a copy of an object in JavaScript?
   Answer: You can create a shallow copy of an object using `Object.assign({}, originalObject)` or the spread operator (`{...originalObject}`).

19. What are template literals in JavaScript?
   Answer: Template literals are a way to create strings using backticks (``) that allow embedded expressions using `${}`.

20. What is the use of `strict mode` in JavaScript?
   Answer: Strict mode is a way to enforce stricter rules on JavaScript code to avoid common errors and improve performance.

22. Explain the purpose of the `Promise` object in JavaScript.
   Answer: Promises are used for asynchronous programming, representing the eventual completion (or failure) of an asynchronous operation and its resulting value.

23. What is a callback hell in JavaScript?
   Answer: Callback hell refers to deeply nested callback functions that can make code hard to read and maintain in asynchronous scenarios.

24. How can you convert a string to a number in JavaScript?
   Answer: You can use `parseInt()` or `parseFloat()` functions to convert a string to a number.

25. What is the difference between `slice()` and `splice()` methods?
Answer: `slice()` returns a shallow copy of a portion of an array, while `splice()` can add or remove elements from an array.

26. Explain the usage of the `bind()` method in JavaScript.
Answer: The `bind()` method creates a new function that, when called, has its `this` keyword set to the provided value.

30. How do you check if an object has a specific property in JavaScript?
Answer: You can use the `hasOwnProperty()` method or the `in` operator to check for the existence of a property.

31. What are the rest parameters in JavaScript?
Answer: Rest parameters allow a function to accept an indefinite number of arguments as an array.

32. How do you handle errors in JavaScript?
Answer: You can use `try...catch` blocks to handle errors and prevent them from crashing the application.

33. What is the difference between `Object.keys()` and `Object.values()`?
Answer: `Object.keys()` returns an array of a given object's own enumerable property names, while `Object.values()` returns an array of its values.

34. Explain the purpose of the `async` and `await` keywords.
Answer: The `async` keyword is used to declare an asynchronous function, and `await` is used to pause the execution of an async function until a promise is resolved.

35. How do you compare two objects for equality in JavaScript?
Answer: In JavaScript, two objects are only equal if they reference the same object in memory.

36. What is the use of the `map()` method in JavaScript arrays?
Answer: The `map()` method creates a new array by applying a callback function to each element of the original array.

38. How can you remove a property from an object in JavaScript?
Answer: You can use the `delete` keyword to remove a property from an object.

40. How do you reverse a string in JavaScript?
Answer: You can use the `split()`, `reverse()`, and `join()` methods to reverse a string.

42. How do you check if an object is empty in JavaScript?
Answer: You can use `Object.keys()` or `Object.getOwnPropertyNames()` to check if an object has any properties.

45. What is the purpose of the `reduce()` method in JavaScript arrays?
Answer: The `reduce()` method is used to applying a function to each element.

47. How do you add an element to the beginning of an array in JavaScript?
Answer: You can use the `unshift()` method to add an element to the beginning of an array.

48. What are the different ways to declare and define a function in JavaScript?
Answer: You can use function declarations, function expressions, arrow functions, and methods defined within objects or classes.

49. What is the `NaN` property in JavaScript?
Answer: `NaN` is a property of the global object representing the value "Not-a-Number."

50. How can you convert a number to a string in JavaScript?
Answer: You can use the `toString()` method or the `String()` constructor to convert a number to a string.

52. What is the purpose of the `filter()` method in JavaScript arrays?
Answer: The `filter()` method is used to create a new array with elements that pass a given test (callback function).

53. How do you check if a variable is a function in JavaScript?
Answer: You can use the `typeof` operator to check if a variable is a function.

54. What is the use of the `concat()` method in JavaScript arrays?
Answer: The `concat()` method is used to merge two or more arrays, creating a new array without modifying the original arrays.

55. Explain the difference between `null`, `undefined`, and `undeclared` variables.
Answer: `null` is a value assigned to a variable, `undefined` indicates a variable has not been assigned any value, and `undeclared` variables are those that have not been declared using any variable declaration keyword (e.g., `var`, `let`, or `const`).

56. How do you check if a variable is an object in JavaScript?
Answer: You can use the `typeof` operator to check if a variable is an object.

57. What is the difference between `call()` and `apply()` methods?
Answer: Both `call()` and `apply()` are used to invoke a function with a specific `this` context and arguments, but the difference lies in how you pass arguments: `call()` takes individual arguments, while `apply()` takes an array of arguments.

58. How can you convert a string to uppercase or lowercase in JavaScript?
Answer: You can use the `toUpperCase()` method to convert a string to uppercase and the `toLowerCase()` method to convert it to lowercase.

59. What are the different ways to loop through an object's properties in JavaScript?
Answer: You can use a `for...in` loop or `Object.keys()`, `Object.values()`, or `Object.entries()` methods to loop through an object's properties.

60. What is the purpose of the `forEach()` method in JavaScript arrays?
Answer: The `forEach()` method is used to execute a provided function once for each array element.

61. Explain the difference between `let` and `const` keywords.
Answer: Both `let` and `const` are used for variable declaration, but `let` allows reassignment of the variable, while `const` creates a read-only constant variable that cannot be reassigned.

62. How do you remove an element from the middle of an array in JavaScript?
Answer: You can use the `splice()` method to remove an element from an array.

64. How do you check if an array includes a specific value in JavaScript?
Answer: You can use the `includes()` method to check if an array includes a specific value.

65. Explain the difference between `slice()` and `substring()` methods for strings.
Answer: Both `slice()` and `substring()` extract parts of a string, but the main difference is that `slice()` can accept negative indices, while `substring()` cannot.

66. What is the purpose of the `find()` method in JavaScript arrays?
Answer: The `find()` method is used to return the value of the first element in the array that satisfies the provided testing function.

67. How can you convert a number to a fixed number of decimal places in JavaScript?
Answer: You can use the `toFixed()` method to convert a number to a string with a fixed number of decimal places.

68. What is the `Math` object in JavaScript?
Answer: The `Math` object is a built-in object that provides mathematical constants and functions.

69. Explain the concept of function currying in JavaScript.
Answer: Function currying is a technique where a function that takes multiple arguments is transformed into a series of functions that take one argument each.

70. How can you merge two or more arrays in JavaScript?
Answer: You can use the `concat()` method or the spread operator (`[...array1, ...array2]`) to merge arrays.

71. What is the difference between `localStorage` and `sessionStorage` in JavaScript?
Answer: Both `localStorage` and `sessionStorage` are used to store key-value pairs in the browser, but `localStorage` persists even after the browser is closed, while `sessionStorage` is cleared when the session ends.

72. What is a generator function in JavaScript?
Answer: A generator function is a special type of function that can be paused and resumed during its execution.

73. How can you convert a string to an array in JavaScript?
Answer: You can use the `split()` method to convert a string to an array based on a separator.

75. How do you find the index of a specific element in an array in JavaScript?
Answer: You can use the `indexOf()` method to find the index of a specific element in an array.

76. What is the purpose of the `Object.keys()` method?
Answer: The `Object.keys()` method returns an array of a given object's own enumerable property names.

77. How can you copy an array or object without modifying the original?
Answer: For arrays, you can use `slice()` or the spread operator (`[...originalArray]`). For objects, you can use `Object.assign({}, originalObject)` or the spread operator (`{...originalObject}`).

79. How do you compare two arrays for equality in JavaScript?
Answer: You can't directly compare arrays using `==` or `===`, as they are reference types. You need to compare their elements individually.

81. Explain the purpose of the `split()` method for strings in JavaScript.
Answer: The `split()` method is used to split a string into an array of substrings based on a specified separator.

82. How can you convert an array to a string in JavaScript?
Answer: You can use the `join()` method to convert an array to a string.

83. What is the difference between `var`, `let`, and `const` regarding variable hoisting?
Answer: Variables declared with `var` are hoisted and initialized with `undefined`, while `let` and `const` are also hoisted but not initialized.

84. How do you sort an array in JavaScript?
Answer: You can use the `sort()` method to sort an array alphabetically or numerically.

85. What is the use of the `every()` method in JavaScript arrays?
Answer: The `every()` method tests whether all elements in an array pass a provided test (callback function).

86. How do you check if an object is an instance of a specific class in JavaScript?
Answer: You can use the `instanceof` operator to check if an object is an instance of a specific class.

87. What is a closure in JavaScript, and why is it useful?
Answer: A closure is a function that has access to variables from its outer (enclosing) scope even after that scope has finished executing. It's useful for creating private variables and data encapsulation.

88. How do you empty an array in JavaScript?
Answer: You can set the length of the array to 0 or use the `splice()` method to remove all elements from the array.

90. How do you convert an object to a JSON string in JavaScript?
Answer: You can use the `JSON.stringify()` method to convert an object to a JSON string.

91. What is the purpose of the `lastIndexOf()` method in JavaScript arrays?
Answer: The `lastIndexOf()` method returns the last index at which a given element is found in an array.

92. How do you check if a variable is a number in JavaScript?
Answer: You can use the `typeof` operator to check if a variable is a number.

93. What is the purpose of the `some()` method in JavaScript arrays?
Answer: The `some()` method tests whether at least one element in the array passes a provided test (callback function).

94. How do you get the current date and time in JavaScript?
Answer: You can use the `Date` object to get the current date and time.

95. What is the `Math.random()` method in JavaScript?
Answer: The `Math.random()` method returns a random floating-point number between 0 (inclusive) and 1 (exclusive).

96. How can you convert a string to an integer in JavaScript?
Answer: You can use `parseInt()` to convert a string to an integer.

97. What is the purpose of the `setTimeout()` function in JavaScript?
Answer: `setTimeout()` is used to execute a function or evaluate an expression after a specified delay in milliseconds.

98. How do you convert an object to an array in JavaScript?
Answer: You can use `Object.entries()` to convert an object to an array of its key-value pairs.

99. What is the difference between `localStorage` and `sessionStorage`?
Answer: `localStorage` stores data with no expiration date, while data in `sessionStorage` is cleared when the session ends.

100. How do you remove duplicates from an array in JavaScript?
Answer: You can use `Set` or `filter()` method to remove duplicates from an array.