100 JavaScript Interview Questions

🧩 JavaScript Basics

1. What is JavaScript?
2. What are the different data types in JavaScript?
3. What is the difference between let, const, and var?
4. What are primitive and non-primitive data types in JavaScript?
5. What is the difference between == and ===?
6. What is type coercion in JavaScript?
7. What is the typeof operator used for?
8. What are template literals?
9. What is the difference between null and undefined?
10. What is NaN in JavaScript?

⚙️ Functions and Scope

11. What are functions in JavaScript?
12. What is the difference between a function declaration and a function expression?
13. What are arrow functions and how are they different?
14. What is lexical scope?
15. What is a closure?
16. Give an example of a closure in JavaScript.
17. What is the purpose of the return statement?
18. What are higher-order functions?
19. What is an immediately invoked function expression (IIFE)?
20. What is recursion in JavaScript?

🧠 Objects and Arrays

21. What are objects in JavaScript?
22. How do you create an object in JavaScript?
23. What are arrays and how are they different from objects?
24. How can you iterate over an array in JavaScript?
25. What are array methods like map(), filter(), and reduce() used for?
26. How can you add a new property to an existing object?
27. What is object destructuring?
28. What is the difference between Object.keys(), Object.values(), and Object.entries()?
29. How can you merge two arrays?
30. What does the spread operator do?

📜 DOM and Browser Concepts

31. What is the DOM?
32. How can you select an element in the DOM?
33. What is the difference between document.getElementById() and document.querySelector()?
34. How do you create and append elements dynamically using JavaScript?
35. What's the difference between onclick="..." and addEventListener()?
36. How are events triggered in the browser?
37. What are event listeners?
38. What does preventDefault() do?
39. How do you change the content or style of a DOM element using JavaScript?
40. How do you handle form validation using JavaScript?

🕹️ Asynchronous JavaScript

41. What is synchronous vs asynchronous programming?
42. What are callbacks in JavaScript?

43. What are promises?
44. What are the states of a promise?
45. What is async and await?
46. What are blocking and non-blocking operations?
47. What is the event loop in JavaScript?
48. How does JavaScript handle asynchronous code if it's single-threaded?
49. What is the purpose of finally() in a Promise?
50. Can you use await outside an async function?

🧱 Operators and Logic

51. What are truthy and falsy values in JavaScript?
52. What is the difference between ++x and x++?
53. What does the ternary operator do?
54. What is short-circuit evaluation?
55. What is the difference between logical && and ||?
56. Output of: true || false && false?
57. What does the nullish coalescing operator (??) do?
58. What are bitwise operators?
59. How does JavaScript handle floating-point precision?
60. Output: [] == ![] — why?

🧰 Objects, Prototypes, and Classes

61. What is prototypal inheritance?
62. How does the this keyword work in JavaScript?
63. How does this behave differently in arrow functions?
64. Output: 5 & 1, 5 | 1, 5 ^ 1 — explain.
65. What are ES6 classes?
66. Why is NaN not equal to NaN?

67. Output: false ? 1 : 2?
68. What is method chaining?
69. Output: console.log('5' * {})?
70. Output: console.log("A" - "B" + 2)?

🧭 Advanced JavaScript Concepts

71. What is hoisting?
72. What is the temporal dead zone (TDZ)?
73. What is currying in JavaScript?
74. What is memoization?
75. What is debouncing?
76. What is throttling?
77. What are generators in JavaScript?
78. What is the Symbol type used for?
79. What is the difference between shallow and deep copy?
80. What is the difference between call(), apply(), and bind()?

💻 Error Handling and Debugging

81. What is a try-catch block used for?
82. What is the purpose of the finally block?
83. What is throw used for in JavaScript?
84. What are runtime and syntax errors?
85. How can you debug JavaScript code?
86. What is the difference between console.log(), console.error(), and console.warn()?
87. What is stack trace?
88. What are custom errors?
89. How can you handle asynchronous errors?

90. What is the purpose of the onerror event?

🧩 Logical & Output-Based Questions

91. What will be the output of:

```
console.log([] + []);
console.log([] + {});
console.log({} + []);
console.log({} + {});
```

92. Predict the output:

```
console.log(typeof null);
console.log(typeof NaN);
```

93. Output of:

```
let a = 10;
let b = (a++, ++a);
console.log(b);
```

94. Output of:

```
let x = [1, 2, 3];
let y = x;
y.push(4);
console.log(x);
```

95. What will this print?

```
console.log(0.1 + 0.2 === 0.3);
```

96. Output:

```
console.log([] == ![]);
```

97. What will be printed?

```
console.log(typeof NaN);
console.log(isNaN("Hello"));
```

98. Output:

```
console.log('5' - 2);
console.log('5' + 2);
```

99. Output:

```
setTimeout(() => console.log('A'));
Promise.resolve().then(() => console.log('B'));
console.log('C');
```

100. What will be the output?

```
var a = 1;
function test() {
 console.log(a);
 var a = 2;
}
test();
```
