

## Basics

1. What are the data types in JavaScript?
2. Explain the difference between `undefined` and `null`.
3. How does JavaScript handle types?
4. What are truthy and falsy values in JavaScript?
5. What is the difference between `==` and `===`?
6. Explain hoisting in JavaScript.
7. What is strict mode in JavaScript and how is it enabled?
8. What are global variables and how are they declared?
9. Explain the difference between `var`, `let`, and `const`.
10. How do you define constants in JavaScript?

## Functions

11. What are the different ways to create functions in JavaScript?
12. Explain the difference between function declaration and function expression.
13. What are arrow functions? How do they differ from regular functions?
14. What is function hoisting? Does it work with arrow functions?
15. How do you pass arguments to a function in JavaScript?
16. Explain function closures and how they are used.
17. What is a callback function? Provide an example of its usage.
18. Explain the concept of higher-order functions and provide an example.
19. How does `this` keyword work in JavaScript?
20. What is the difference between call, apply, and bind methods?

## Objects and Prototypes

21. How do you create objects in JavaScript?
22. Explain the concept of prototypal inheritance in JavaScript.
23. What is a prototype chain?
24. How do you add properties to an object in JavaScript?
25. What are object methods? Provide an example.
26. Explain object destructuring in JavaScript.
27. How do you clone an object in JavaScript?
28. What are ES6 classes? How do they differ from constructor functions?
29. How do you inherit from a class in JavaScript?
30. Explain the difference between `hasOwnProperty` and `in` operator.

## Arrays

31. How do you create an array in JavaScript?
32. Explain the various methods to add and remove elements from an array.
33. What are array methods like `map`, `filter`, and `reduce`? Provide examples of their usage.
34. Explain the difference between `slice` and `splice` methods.
35. How do you iterate over an array in JavaScript?
36. What are array-like objects? Provide examples.
37. Explain the `forEach` method and how it differs from a regular `for` loop.
38. How do you check if a variable is an array in JavaScript?
39. What are typed arrays in JavaScript?

## Asynchronous JavaScript

40. What is asynchronous programming in JavaScript?
41. Explain the event loop in JavaScript.
42. What are promises in JavaScript? How do they work?
43. Explain `async` and `await` keywords in JavaScript.
44. What is the difference between callbacks and promises?
45. How do you handle errors in asynchronous code?
46. Explain the concept of chaining in promises.
47. What are generator functions and iterators in JavaScript?
48. How do you handle multiple asynchronous operations in JavaScript?

## ### DOM Manipulation and Events

49. How do you access and manipulate the DOM in JavaScript?
50. Explain event handling in JavaScript.
51. What are event bubbling and capturing?
52. How do you add and remove event listeners in JavaScript?
53. How do you prevent default behavior in an event?
54. Explain event delegation and its benefits.
55. How do you create and trigger custom events in JavaScript?
56. Explain the difference between `document.getElementById` and `document.querySelector`.

## ES6+ Features

57. What are the new features introduced in ES6 (ECMAScript 2015)?
58. Explain destructuring assignment in ES6.
59. What are template literals in ES6?
60. How do you use `let` and `const` in ES6?
61. Explain arrow functions and their advantages.
62. What are default parameters in ES6?
63. How do you use classes in ES6?
64. Explain the spread (`...`) operator and its uses.
65. What are modules in ES6? How do you export and import modules?

## Error Handling and Debugging

66. How do you handle errors in JavaScript?
67. What are `try`, `catch`, and `finally` blocks?
68. How do you throw custom errors in JavaScript?
69. What are some common debugging techniques in JavaScript?
70. How do you use the JavaScript console for debugging?

## Web APIs and Browser Environment

71. How do you use localStorage and sessionStorage in JavaScript?
72. What is the difference between localStorage and cookies?
73. How do you make HTTP requests in JavaScript? Explain `fetch` API.
74. How do you handle CORS issues in JavaScript?
75. Explain the difference between synchronous and asynchronous XMLHttpRequest.

## Testing and Performance

76. How do you write unit tests for JavaScript code?
77. What are some popular testing frameworks for JavaScript?
78. How do you optimize JavaScript code for better performance?
79. What tools would you use to analyze and improve website performance?

## Security and Best Practices

80. What are some common security issues in JavaScript applications?
81. How do you prevent XSS (Cross-Site Scripting) attacks in JavaScript?
82. What is Content Security Policy (CSP) and how do you implement it?
83. How do you sanitize user input in JavaScript?
84. What are some best practices for writing secure JavaScript code?

## Miscellaneous

85. How do you create a timer in JavaScript?
86. Explain the difference between synchronous and asynchronous code execution.
87. How do you implement a carousel/slider in JavaScript?
88. Explain the concept of lazy loading images in JavaScript.
89. How do you implement pagination using JavaScript?
90. How do you detect the browser and its version using JavaScript?
91. Explain the concept of functional programming in JavaScript.
92. What are the benefits of using immutable data structures in JavaScript?
93. How do you handle routing in a single-page application (SPA) using JavaScript?
94. How do you implement animations in JavaScript?
95. How do you implement drag-and-drop functionality in JavaScript?
96. Explain the concept of memoization and provide a practical example.
97. How do you handle responsive design with JavaScript?
98. How do you implement internationalization (i18n) in JavaScript applications?
99. What are the differences between progressive enhancement and graceful degradation?
100. How do you integrate third-party libraries and plugins in JavaScript applications?