1. What is JavaScript, and how is it different from Java?
2. How do you include JavaScript in an HTML document?
3. What are data types in JavaScript?
4. Explain the difference between let, var, and const.
5. What is a variable and how do you declare a variable in JavaScript?
6. What are JavaScript functions, and how do you define a function?
7. What is the difference between undefined and null in JavaScript?
8. How do you create an object in JavaScript?
9. What is the difference between == and === in JavaScript?
10. What are arrays in JavaScript, and how do you declare them?
11. Explain what the typeof operator does.
12. How do you write comments in JavaScript?
13. What is the difference between function declarations and function expressions?
14. Explain what an anonymous function is.
15. What are template literals in JavaScript?
16. How does JavaScript handle automatic type conversion (type coercion)?
17. What are JavaScript closures?
18. How do you create a loop in JavaScript? Give examples of for and while loops.
19. What are the different types of pop-up boxes in JavaScript?
20. How do you handle errors in JavaScript using try, catch, and finally?
21. Explain the concept of scope in JavaScript.
22. What is a callback function?
23. How can you access elements from the DOM in JavaScript?
24. What is the difference between synchronous and asynchronous code in JavaScript?
25. What is event delegation in JavaScript?
26. What is the difference between the spread operator and rest parameters?
27. How do you create a promise in JavaScript?
28. What is the this keyword in JavaScript, and how does it work?
29. What is event bubbling in JavaScript?
30. How do you prevent default behavior for an event in JavaScript?
31. What are arrow functions, and how do they differ from regular functions?
32. Explain what the bind(), call(), and apply() methods do in JavaScript.
33. What is the new keyword used for in JavaScript?
34. Explain how setTimeout and setInterval work.
35. What is strict mode in JavaScript, and why is it used?
36. What are higher-order functions in JavaScript?
37. How can you use map(), filter(), and reduce() in JavaScript arrays?
38. What are JavaScript modules, and how do you export and import them?
39. How does the async/await syntax work in JavaScript?
40. What is object destructuring, and how is it used?
41. How does the for...in loop differ from the for...of loop?
42. Explain the concept of immutability in JavaScript.
43. What is a polyfill in JavaScript?
44. Explain how JavaScript handles memory management.
45. What is the purpose of the Promise.all() method?
46. What are default parameters in JavaScript functions?
47. What is hoisting in JavaScript, and how does it work?
48. What is the difference between deep copy and shallow copy in JavaScript?
49. How do you debounce or throttle a function in JavaScript?
50. Explain the concept of currying in JavaScript.
51. What is event loop in JavaScript?
52. What is the difference between == and Object.is()?
53. What is the difference between microtasks and macrotasks in the event loop?
54. How does JavaScript handle garbage collection?
55. What are generators in JavaScript, and how are they used?
56. What is memoization in JavaScript, and when would you use it?
57. What is tail call optimization in JavaScript?
58. What is the difference between a shallow and deep merge of objects?
59. How can you create a singleton pattern in JavaScript?
60. How do you handle memory leaks in JavaScript?
61. Explain what Symbol is in JavaScript and when you would use it.
62. How do you implement inheritance in JavaScript?
63. What is a Proxy object in JavaScript, and what is it used for?
64. What is the difference between function expressions and arrow functions with respect to the this keyword?
65. How do you create and manage async iterators in JavaScript?
66. What is the module pattern in JavaScript?
67. What are service workers in JavaScript, and how are they used?
68. How do you implement lazy loading in JavaScript?
69. What is a WeakMap and WeakSet in JavaScript?
70. What are tagged template literals, and how do they work?
71. How do you handle circular references in JavaScript objects?
72. What is the observer pattern in JavaScript?
73. What is the concept of monads in JavaScript?
74. How do you handle large numbers and perform precision calculations in JavaScript?
75. What is the difference between prototypal and classical inheritance in JavaScript?
76. What are pure functions in JavaScript, and why are they important?
77. How do you optimize the performance of a JavaScript application?
78. What is tail call recursion, and how does JavaScript optimize it?
79. Explain how import() for dynamic imports works.
80. What are Typed Arrays in JavaScript?
81. What is the Temporal API in JavaScript, and how is it useful?
82. How do you handle binary data in JavaScript?
83. What is the Intl object in JavaScript, and how do you use it?
84. How can you secure a JavaScript application from XSS attacks?
85. What are asynchronous generators, and how do they differ from regular generators?
86. How do you implement a debounce function in JavaScript?
87. What is optional chaining (?.), and when would you use it?
88. How does JavaScript handle tail recursion optimization?
89. What are private class fields, and how do they work?
90. How do you create immutability in JavaScript with Object.freeze()?
91. What is function composition, and how is it used in JavaScript?
92. What is the difference between Object.freeze() and Object.seal()?
93. How does JavaScript implement the call stack and heap memory model?
94. What are async iterators, and how do they differ from regular iterators?
95. What are atomic operations in JavaScript?
96. How does JavaScript handle non-blocking I/O operations?
97. What are WebSockets, and how do they work in JavaScript?
98. What is the purpose of the Reflect API in JavaScript?
99. How do you deal with cross-origin resource sharing (CORS) in JavaScript applications?
100. What are ECMAScript proposals, and how do they become part of the language?