Basics

1. What is JavaScript?
2. Difference between JavaScript and Java?
3. How is JavaScript executed in the browser?
4. What are data types in JavaScript?
5. Difference between var, let, and const?
6. What is hoisting?
7. What is scope in JavaScript?
8. What is the difference between global, function, and block scope?
9. What are truthy and falsy values?
10. What is type coercion?
11. What is NaN?
12. Difference between == and ===?
13. What is strict mode?
14. What is the use of typeof?
15. What is an IIFE?
16. What are primitive vs reference types?
17. What is pass by value vs pass by reference?
18. What is the this keyword?
19. How does this behave in arrow functions?
20. What is the JavaScript call stack?

Functions & Objects

21. What are first-class functions?
22. What is a higher-order function?
23. What are pure functions?
24. What is a callback function?
25. What is function currying?
26. What is function composition?
27. What is closure?
28. Give a real-world use case of closures.

29. What are objects in JavaScript?
30. How do you create objects?
31. What is prototype?
32. What is prototype chaining?
33. Difference between __proto__ and prototype?
34. What is inheritance in JavaScript?
35. How does Object.create() work?
36. What is bind, call, and apply?
37. What is method borrowing?
38. What is object destructuring?
39. What is spread vs rest operator?
40. How does garbage collection work?

Arrays & Strings

41. Difference between map, filter, and reduce?
42. What is forEach vs map?
43. How do you remove duplicates from an array?
44. Difference between slice and splice?
45. How do you flatten an array?
46. What is immutability?
47. How do you check if a value is an array?
48. What are array-like objects?
49. What is string immutability?
50. How do you reverse a string?
51. How do you check palindrome?
52. What is includes()?
53. What is split, join?
54. What is charAt() vs charCodeAt()?
55. How do you compare strings safely?

Asynchronous JavaScript

56. What is synchronous vs asynchronous execution?
57. What is the event loop?
58. What are Web APIs?
59. What is a callback queue?
60. What is a Promise?
61. Promise states?
62. What is promise chaining?
63. What is callback hell?
64. How do Promises solve callback hell?
65. What is async/await?
66. Error handling with async/await?
67. What is Promise.all()?
68. Difference between Promise.all() and Promise.allSettled()?
69. What is Promise.race()?
70. What is Promise.any()?
71. What are microtasks?
72. Microtask vs macrotask?
73. What is setTimeout()?
74. What is debouncing?
75. What is throttling?

Browser & DOM

76. What is DOM?
77. How does DOM differ from HTML?
78. What is DOM manipulation?
79. What is event bubbling?
80. What is event capturing?
81. What is event delegation?
82. What is preventDefault()?
83. What is stopPropagation()?
84. What is reflow and repaint?

85. How do you optimize DOM performance?
86. What is Virtual DOM?
87. Virtual DOM vs Real DOM?
88. What is shadow DOM?
89. What is Web Storage API?
90. Difference between localStorage and sessionStorage?

ES6+ & Modern JavaScript

91. What are ES6 features?
92. What are arrow functions?
93. What are default parameters?
94. What are template literals?
95. What are modules in JavaScript?
96. import vs require?
97. What is tree shaking?
98. What is Babel?
99. What is transpiling?
100. What is polyfill?
101. What is optional chaining?
102. What is nullish coalescing?
103. What is dynamic import?
104. What is Symbol?
105. What is BigInt?

Performance, Security & Best Practices

106. How do you optimize JavaScript performance?
107. What is memory leak in JavaScript?
108. Common causes of memory leaks?
109. How do you prevent XSS?
110. What is CSRF?
111. How does same-origin policy work?

112. What is CORS?

113. How do you secure APIs in JavaScript apps?

114. What is Content Security Policy (CSP)?

115. What is lazy loading?

116. What is code splitting?

117. What is bundling?

118. What is Web Workers?

119. What is Service Worker?

120. How do you structure large JavaScript applications?