63 JavaScript Interview Questions to Ask Candidates (with Answers)

Questions

1. Can you explain what closures are in JavaScript and why they are useful?

2. How does JavaScript handle asynchronous operations?

3. What is the difference between `==` and `===` in JavaScript?

4. How would you explain event delegation in JavaScript?

5. What is the purpose of the `this` keyword in JavaScript?

6. Can you explain the concept of prototypes in JavaScript?

7. What are JavaScript promises, and why are they important?

8. What is the event loop in JavaScript, and how does it work?

9. What is the difference between `let`, `const`, and `var`, and when would you use each?

10. Can you explain how the `map()`, `filter()`, and `reduce()` methods work in JavaScript?

11. What are template literals in JavaScript, and how do they differ from regular strings?

12. How can you handle errors in JavaScript code? Can you describe the use of `try`, `catch`, and `finally`?

13. What is the purpose of the `async` and `await` keywords in JavaScript?

14. How do you create a simple object in JavaScript, and what are the key differences between objects and arrays?

15. Can you explain what the `bind()` method does in JavaScript?

16. What are JavaScript modules, and why are they important?

17. How do you prevent a default action of an event in JavaScript?

18. What is the purpose of the `addEventListener()` method in JavaScript?

19. Can you explain the concept of hoisting in JavaScript?

20. What is a JavaScript generator function, and how does it work?

21. How do you create a promise in JavaScript and handle its resolution or rejection?

22. What is the difference between synchronous and asynchronous execution in JavaScript?

23. How can you check if a variable is an array in JavaScript?

24. What is event bubbling, and how does it differ from event capturing in JavaScript?

25. Can you explain what a closure is in the context of scope and execution context?

26. How do you use the `forEach()` method in JavaScript, and how does it differ from a standard `for` loop?

27. What is the purpose of the `JSON.stringify()` and `JSON.parse()` methods in JavaScript?

28. How do you handle scope in JavaScript?

29. What are the main differences between synchronous and asynchronous programming in JavaScript?

30. How do you approach debugging JavaScript code?

31. What are higher-order functions in JavaScript, and why are they useful?

32. Can you explain what 'hoisting' is in JavaScript?

33. How do you manage state in a JavaScript application?

34. What is the difference between deep and shallow copy in JavaScript?

35. What are the key differences between 'null' and 'undefined' in JavaScript?

36. How do you optimize the performance of a JavaScript application?

37. How do you handle errors and exceptions in JavaScript?

38. Can you explain the difference between event bubbling and event capturing in JavaScript?

39. How do you stop event propagation in JavaScript?

40. What is the purpose of the `stopPropagation()` method in event handling?

41. How would you implement custom events in JavaScript?

42. Can you describe how to handle multiple events with a single function?

43. What are passive event listeners and why are they significant?

44. How do you manage event listener memory leaks in JavaScript?

45. Can you explain the role of the `preventDefault()` method in event handling?

46. How do you handle keyboard events in JavaScript?

47. What strategies would you use to optimize event handling performance in a web application?

48. Can you explain the difference between a callback function and a promise?

49. What are some common pitfalls when working with asynchronous code in JavaScript?

50. How does the async/await syntax improve asynchronous code readability?

51. What is a race condition and how can it be avoided in JavaScript?

52. Explain how you would handle multiple asynchronous operations that need to be completed before moving on to the next step.

53. What is an example of when you might use a promise instead of a callback?

54. How would you handle errors in an asynchronous function using async/await?

55. What are the benefits of using promises over traditional callback functions?

56. How does JavaScript's event loop handle asynchronous operations?

57. How would you optimize the performance of a JavaScript-heavy web application?

58. Describe a situation where you had to debug a complex JavaScript issue. How did you approach it?

59. How would you explain the concept of 'this' in JavaScript to a junior developer?

60. If you were tasked with converting a large jQuery project to modern JavaScript, how would you approach it?

61. How would you implement a feature that requires real-time updates from a server?

62. Describe how you would structure a large-scale JavaScript application to ensure maintainability and scalability.

63. How would you ensure cross-browser compatibility in a JavaScript application?