JavaScript Comprehensive Interview Questions

JavaScript Basics:

1. What are the different data types in JavaScript?

2. What is the difference between var, let, and const?

3. Explain the difference between == and ===.

4. What is undefined vs null?

5. How do you check if a variable is an array in JavaScript?

6. Explain the concept of hoisting with an example.

7. What is the difference between function declaration and function expression?

8. What is a closure in JavaScript? Provide a use case.

9. Explain the concept of scope (global, local, block).

10. What are IIFE (Immediately Invoked Function Expressions)?

ES6+ Features:

11. What is destructuring? How can it be used with arrays and objects?

12. Explain template literals and their use cases.

13. What is the difference between arrow functions and regular functions?

14. Explain default parameters in JavaScript functions.

15. What is the spread operator and how is it different from the rest operator?

16. How do modules work in JavaScript (import/export)?

17. What are Promises and how are they different from async/await?

18. Explain the difference between map() and forEach().

19. What is a Set and how is it different from an Array?

20. What is WeakMap and WeakSet in JavaScript?

Functions & Closures:

21. What are higher-order functions? Provide an example.

22. What is a pure function? Why is it important?

23. Explain callback functions with an example.

24. What is a memoization function? How can you implement it?

25. How does function currying work? Provide an example.

26. What is the difference between call(), apply(), and bind()?

27. How do closures work in JavaScript? Provide a practical use case.

28. How would you implement a debounce function in JavaScript?

29. How would you implement a throttle function in JavaScript?

30. Explain recursive functions with an example.

Loops and Iterators:

31. What is the difference between for and for...of loops?

32. How does a forEach() loop work? Can you break out of it?

33. What is the difference between for...in and for...of?

34. How would you flatten a nested array using recursion or loops?

35. How would you write a function to find duplicates in an array?

Arrays and Objects:

36. How would you remove duplicates from an array in JavaScript?

37. Explain the difference between map(), filter(), and reduce().

38. How would you merge two arrays without duplicates?

39. How would you deep clone an object in JavaScript?

40. How would you check if two objects are equal?

Asynchronous JavaScript (Promises & Async/Await):

41. Explain the event loop in JavaScript.

42. What is callback hell and how can you avoid it?

43. How does async/await improve the readability of asynchronous code?

44. What is Promise.all() and how does it work?

45. How would you implement a retry mechanism using promises?

46. What is the difference between Promise.all(), Promise.allSettled(), and Promise.race()?

APIs and Fetch:

47. How would you make an API call using fetch()?

48. How can you handle errors when using fetch()?

49. What is CORS and how does it affect API requests?

50. What is the difference between GET and POST requests?

51. How would you implement a loading spinner while making an API request?

Error Handling:

52. What are try/catch blocks in JavaScript?

53. How do you handle uncaught errors in JavaScript?

54. What is throw in JavaScript and when would you use it?

55. How would you create a custom error class in JavaScript?

56. How do you handle promise rejections in async/await?

DOM Manipulation:

57. How would you select an element by its ID and class in JavaScript?

58. How would you add an event listener to an element?

59. How do you dynamically create and append elements to the DOM?

60. How would you implement a modal using JavaScript?

61. What is event delegation and how does it improve performance?

Event Loop & Execution Context:

62. Explain the concept of call stack and task queue.

63. What is the difference between microtasks and macrotasks?

64. How does JavaScript handle asynchronous code?

65. What is the difference between setTimeout() and setInterval()?

66. What is debouncing and throttling? How would you implement them?

Advanced Concepts:

67. What is the difference between deep copy and shallow copy?

68. How does prototypal inheritance work in JavaScript?

69. What are JavaScript classes? How do they differ from functions?

70. What is this in JavaScript? How does it behave in different contexts?

71. How do you implement a polyfill in JavaScript?

Security in JavaScript:

72. What is XSS (Cross-Site Scripting) and how can you prevent it?

73. What is CSP (Content Security Policy) and why is it important?

74. How would you protect your JavaScript code from man-in-the-middle attacks?

75. What are cookies, localStorage, and sessionStorage? How are they different?

Real-World Problem-Solving Questions:

76. Write a function to find the longest palindrome in a string.

77. How would you implement a binary search algorithm in JavaScript?

78. How would you shuffle an array randomly?

79. Write a function to find the first non-repeating character in a string.

80. Write a function to reverse a linked list in JavaScript.