100 JavaScript Practice Questions: Arrays, Functions & Loops

Arrays (Questions 1-35)

Basic Array Operations

1. Create an array with 5 numbers and find the sum of all elements.
2. Write code to find the largest number in an array [3, 7, 2, 9, 1].
3. Create an array of fruits and add a new fruit at the beginning.
4. Remove the last element from an array of colors.
5. Check if the number 5 exists in the array [1, 2, 3, 4, 5, 6].
6. Find the index of 'apple' in the array ['banana', 'apple', 'orange'].
7. Create an array of 10 numbers and reverse it without using reverse() method.
8. Merge two arrays [1, 2, 3] and [4, 5, 6] into one.
9. Create a copy of an array without modifying the original.
10. Replace all occurrences of 'cat' with 'dog' in an array of animals.

Array Methods & Manipulation

11. Use slice() to extract the middle 3 elements from an array of 7 elements.
12. Use splice() to remove 2 elements starting from index 3 in an array.
13. Convert the array ['h', 'e', 'l', 'l', 'o'] into a string.
14. Split the string "JavaScript is awesome" into an array of words.
15. Sort an array of names alphabetically.
16. Sort an array of numbers in descending order.
17. Filter an array to get only even numbers from [1, 2, 3, 4, 5, 6, 7, 8].
18. Use map() to square all numbers in an array [1, 2, 3, 4].
19. Use reduce() to find the product of all numbers in an array.
20. Find all numbers greater than 10 in an array using filter().

Advanced Array Operations

21. Create a 2D array (matrix) with 3 rows and 3 columns.
22. Access and modify the element at row 2, column 1 in a 2D array.
23. Find the sum of all elements in a 2D array.
24. Flatten a nested array [[1, 2], [3, 4], [5, 6]] into [1, 2, 3, 4, 5, 6].
25. Remove duplicate elements from an array [1, 2, 2, 3, 3, 4].

26. Find the intersection of two arrays [1, 2, 3, 4] and [3, 4, 5, 6].
27. Find elements that exist in the first array but not in the second.
28. Group an array of objects by a specific property.
29. Sort an array of objects by a specific property (e.g., age).
30. Find the frequency of each element in an array.

Array Challenges

31. Rotate an array to the right by k positions.
32. Find the second largest number in an array.
33. Check if an array is sorted in ascending order.
34. Find the missing number in an array of consecutive integers.
35. Move all zeros to the end of an array while maintaining order of other elements.

Functions (Questions 36-65)

Function Basics

36. Write a function that takes two numbers and returns their sum.
37. Create a function that checks if a number is even or odd.
38. Write a function that calculates the factorial of a number.
39. Create a function that converts Celsius to Fahrenheit.
40. Write a function that checks if a string is a palindrome.
41. Create a function that returns the absolute value of a number without using Math.abs().
42. Write a function that finds the maximum of three numbers.
43. Create a function that counts the number of vowels in a string.
44. Write a function that generates a random number between min and max values.
45. Create a function that capitalizes the first letter of each word in a string.

Function Parameters & Return Values

46. Write a function with default parameters that greets a person.
47. Create a function that accepts unlimited arguments and returns their average.
48. Write a function that returns multiple values using an object.
49. Create a function that accepts a callback function as parameter.
50. Write a function that returns different values based on the number of arguments passed.
51. Create a function that validates email format using parameters.
52. Write a function that calculates compound interest with optional parameters.

53. Create a function that formats currency with default locale settings.
54. Write a function that accepts an array and a transformation function.
55. Create a function that returns a function (higher-order function).

Arrow Functions & Advanced Concepts

56. Convert a regular function to an arrow function.
57. Write an arrow function that filters an array based on a condition.
58. Create an immediately invoked function expression (IIFE).
59. Write a recursive function to calculate Fibonacci sequence.
60. Create a closure that maintains a private counter.
61. Write a function that memorizes its results (memoization).
62. Create a function that can be called in different ways (function overloading).
63. Write a generator function that yields even numbers.
64. Create an async function that simulates API call with setTimeout.
65. Write a function that binds context using call, apply, or bind.

Loops (Questions 66-100)

For Loops

66. Use a for loop to print numbers from 1 to 10.
67. Use a for loop to print even numbers from 2 to 20.
68. Create a multiplication table for number 7 using for loop.
69. Use nested for loops to create a pattern of stars (*).
70. Use a for loop to iterate through an array and find the sum.
71. Use for loop to reverse a string character by character.
72. Create a for loop that counts backwards from 10 to 1.
73. Use for loop to find all prime numbers between 1 and 50.
74. Use nested for loops to multiply two matrices.
75. Use for loop with break statement to find first number divisible by 7.

While Loops

76. Use while loop to print numbers until a condition is met.
77. Create a guessing game using while loop (guess a number between 1-10).
78. Use while loop to calculate sum of digits in a number.
79. Use while loop to reverse a number.

80. Create a while loop that continues until user enters 'quit'.
81. Use while loop to find the largest digit in a number.
82. Create a while loop that simulates a countdown timer.
83. Use while loop with continue statement to skip certain values.
84. Use while loop to validate user input until correct format is entered.
85. Create a while loop that processes items from an array until it's empty.

Advanced Loop Concepts

86. Use for...in loop to iterate through object properties.
87. Use for...of loop to iterate through array values.
88. Compare performance between for, while, and forEach loops.
89. Use nested loops to find all pairs in an array that sum to a target.
90. Create a loop that handles both arrays and objects dynamically.
91. Use loop with try-catch to handle potential errors during iteration.
92. Create an infinite loop with proper exit conditions.
93. Use loops to implement binary search algorithm.
94. Use loops to sort an array using bubble sort algorithm.
95. Create a loop that processes data in chunks/batches.

Loop Challenges

96. Use loops to create a spiral pattern in a 2D array.
97. Find the longest consecutive sequence in an array using loops.
98. Use loops to implement a simple calculator that processes multiple operations.
99. Create a loop that finds all anagrams of a word from a dictionary array.
100. Use loops to simulate a simple game (like tic-tac-toe) with win condition checking.

Tips for Practice:

1. **Start Simple**: Begin with basic questions and gradually move to complex ones.
2. **Test Your Code**: Always run your code to verify it works correctly.
3. **Optimize**: After solving, think about how to make your solution more efficient.
4. **Understand**: Don't just write code; understand why it works.
5. **Debug**: Practice debugging when your code doesn't work as expected.
6. **Combine Concepts**: Many problems require using arrays, functions, and loops together.

Example Solutions Pattern:

```javascript
// Question: Find the sum of all elements in an array
function sumArray(arr) {
  let sum = 0;
  for (let i = 0; i < arr.length; i++) {
    sum += arr[i];
  }
  return sum;
}

// Test
console.log(sumArray([1, 2, 3, 4, 5])); // Output: 15
```

Practice these questions systematically, and you'll build a strong foundation in JavaScript arrays, functions, and loops!