

[HR Toolkit](#) / [Interview Questions](#)

/ Top 50 JavaScript coding Interview Questions and Answers

# Top 50 JavaScript coding Interview Questions and Answers

*Published: August 5, 2023 | Last modified: September 8, 2025*

2412

JavaScript is one of the widely used programming languages and used to create engaging and dynamic websites. Due to its vast scope, JavaScript interviews can be challenging. However, thorough preparation can help overcome these challenges. An interviewee should familiarize themselves with the most basic concepts to the most complex libraries and frameworks.

## Key Takeaways:

- Assess both fundamentals and problem-solving
- Look beyond syntax, evaluate how candidates explain their thought process
- Gauge candidates' framework familiarity in React, Node.js, etc.
- Soft skills are equally important
- Spot red flags early, watch out for reliance on copy-paste coding and lack of async knowledge





# Look for in a JavaScript Coding Interview

When hiring JavaScript developers, interviewers focus on both technical mastery and workplace behavior. A balanced mix of hard skills and soft skills helps HR determine who stands out.

## Hard Skills (Technical Expertise)

Interviewers assess hard skills to ensure candidates can write efficient, maintainable code and handle real-world challenges. Strong technical foundations prove that a developer can build, debug, and scale applications confidently.

- **Core JavaScript fundamentals:** Mastery of prototypes, closures, scope, and sync programming to solve problems without relying only on frameworks.
- **Problem-solving and algorithms:** Ability to manipulate arrays/objects, handle recursion, and apply efficient logic during live coding sessions.
- **Frameworks and ecosystem knowledge:** Hands-on experience with React, Node.js, or similar tools for end-to-end development.
- **Tools and best practices:** Familiarity with debugging tools, ES6+ features, and performance optimization techniques.
- Soft Skills (Workplace Competencies)

Technical ability alone is not enough. Interviewers also look for soft skills that show how well a developer can work with teams, communicate ideas, and adapt to change. These are traits that make someone reliable beyond just coding.

- **Clear communication:** Explaining code, decisions, and trade-offs in a way that's easy for teams to follow.
- **Collaboration and team fit:** Adapting to [team building activities](#), code reviews, and cross-functional teamwork.
- **Problem ownership:** Taking responsibility for debugging, testing, and delivering working solutions under deadlines.





These top 50 JavaScript interview questions and answers will help an interviewee to thoroughly practice and prepare for their interview.

# Top 50 JavaScript Coding Interview Questions

## Basic JavaScript coding questions

Basic JavaScript questions cover concepts like data types, variables and scoping, array, string manipulation, OOP (Object Oriented Programming), control flow, error handling, DOM manipulation, and asynchronous programming. The basic JavaScript coding interview questions are:

### 1. Write a JavaScript function to calculate the sum of two numbers.

When managers ask this question, they are looking for the candidate's basic understanding of JavaScript. They assess their understanding of basic syntax along with problem-solving skills. This also helps evaluate the candidate's coding style and attention to detail.

#### Sample Answer

[copy](#)

*I would take two parameters and the following function can be used to calculate the sum of any 2 numbers that are passed as arguments.*

```
function sumOfTwoNumbers(a, b) {  
  
    return a + b;  
  
}
```

### 2. Write a JavaScript program to find the maximum number in an array.





code step-by-step while demonstrating bug-free code.

### Sample Answer

copy

```
function findMaxNumber(arr) {  
  
    return Math.max(...arr);  
  
}
```

### 3. Write a JavaScript function to check if a given string is a palindrome (reads the same forwards and backwards).

The interviewer is looking for the candidate's familiarity with loop constructs, JavaScript string methods, and other basic JavaScript syntax. They will evaluate the candidate's skills based on the approach used to solve the palindrome problem.

### Sample Answer

copy

```
function isPalindrome(str) {  
  
    return str === str.split("").reverse().join("");  
  
}
```

### 4. Write a JavaScript program to reverse a given string.

Hiring managers are expecting an accurate solution that demonstrates the interviewee's proficiency in JavaScript programming.

```
const reverseString = (str) => str.split("").reverse().join("");
```

### 5. Write a JavaScript function that takes an array of numbers and returns a new array with only the even numbers.





logically and articulate their thought processes.

### Sample Answer

copy

*By using the filter method on the array, I can check if each element is even or not by using the modulus operator (%) with 2. The element is even if the result is 0. This can be included in the new array.*

```
function filterEvenNumbers(numbers) {  
  
    return numbers.filter(num => num % 2 === 0);  
  
}
```

## 6. Write a JavaScript program to calculate the factorial of a given number.

By asking this question, managers aim to assess the candidate's algorithmic thinking and understanding of JavaScript programming. The interviewer expects the candidate to demonstrate their knowledge of the factorial concept.

### Sample Answer

copy

*A factorial number is the product of all positive integers, which are equal to or less than the given number.*

```
function factorial(number) {  
  
    if (number === 0 || number === 1) {  
  
        return 1;  
  
    } else {  
  
        return number * factorial(number - 1);  
  
    }  
  
}
```





## 7. Write a JavaScript function to check if a given number is prime.

Interviewers can analyze the candidate's knowledge of JavaScript algorithms and mathematical concepts. They expect the candidate to translate a mathematical concept into functional code.

### Sample Answer

[copy](#)

*To check if a given number is prime, loop from 2 to the square root of the number. If any integer evenly divides it, the number is not prime.*

```
function isPrime(num) {  
  
  if (num <= 1) return false;  
  
  for (let i = 2; i <= Math.sqrt(num); i++) {  
  
    if (num % i === 0) return false;  
  
  }  
  
  return true;  
  
}
```

## 8. Write a JavaScript program to find the largest element in a nested array.

When asking this question, interviewers are looking for the candidate's ability to handle nested data structures and apply their knowledge of conditional statements, arrays, and loops. Candidates must apply their knowledge to real-world scenarios.

### Sample Answer

[copy](#)

```
function findLargestElement(nestedArray) {  
  
  let largest = nestedArray[0][0];
```





```
for (let num of arr) {  
  
    if (num > largest) {  
  
        largest = num;  
  
    }  
  
}  
  
}  
  
return largest;  
  
}
```

### 9. Write a JavaScript function that returns the Fibonacci sequence up to a given number of terms.

This question helps hiring managers assess the interviewee's understanding of fundamental algorithms in JavaScript. They expect the candidate to consider edge cases and handle errors.

#### Sample Answer

copy 

```
function fibonacciSequence(numTerms) {  
  
    if (numTerms <= 0) return [];  
  
    if (numTerms === 1) return [0];  
  
    let sequence = [0, 1];  
  
    while (sequence.length < numTerms) {  
  
        let nextNumber = sequence[sequence.length - 1] +  
sequence[sequence.length - 2];  
  
        sequence.push(nextNumber);  
  
    }  
  
}
```





```
return sequence;
```

```
}
```

## 10. Write a JavaScript program to convert a string to title case (capitalize the first letter of each word).

Interviewers analyze the candidate's ability to break down a problem into manageable steps and demonstrate knowledge of string manipulation, looping, and basic JavaScript functions.

### Sample Answer

copy 

```
function toTitleCase(str) {  
  
    return str.replace(/\b\w/g, l => l.toUpperCase());  
  
}
```

## Advanced JavaScript coding interview questions

Advanced JavaScript coding includes various complex concepts and techniques. Such key concepts are often tested in JavaScript interviews. Some of the concepts are – closure and scope, prototypal inheritance, **functional programming**, design patterns, memory management, ES6+ features, and many more.

### 1. Implement a debounce function in JavaScript that limits the frequency of a function's execution when it's called repeatedly within a specified time frame.

Interviewers expect the candidate to showcase their ability to clearly explain the purpose of the debounce function and its usage in scenarios where function calls need to be controlled. They are looking for the person's ability to articulate technical concepts clearly.







*By delaying the execution of the debounce function until the specified time frame has passed, the frequency can be limited.*

```
function debounce(func, delay) {  
  
  let timer;  
  
  return function() {  
  
    clearTimeout(timer);  
  
    timer = setTimeout(func, delay);  
  
  };  
  
}
```

**2. Write a function that takes an array of objects and a key, and returns a new array sorted based on the values of that key in ascending order.**

By asking this question, hiring managers analyze how well the candidate can discuss the sorting algorithm and its time complexity. It's also crucial for candidates to demonstrate their code's robustness.

### Sample Answer

copy 

*The following function takes an array of objects and a key to sort the array based on the values in ascending order.*

```
function sortByKey(arr, key) {  
  
  return arr.sort((a, b) => a[key] - b[key]);  
  
}
```

**3. Implement a deep clone function in JavaScript that creates a copy of a nested object or array without any reference to the original.**





issues while cloning.

### Sample Answer

copy

*By using two methods together and creating a deep clone, I can serialize the object to a JSON string. I would then parse it back into a new object, thereby removing any reference to the original object.*

```
function deepClone(obj) {  
  
    return JSON.parse(JSON.stringify(obj));  
  
}
```

### 4. Write a recursive function to calculate the factorial of a given number.

Interviewers expect the candidate to write a concise recursive function that handles edge cases. Candidates must show their understanding of how recursion works to avoid infinite loops or stack overflow errors.

### Sample Answer

copy

```
function factorial(num) {  
  
    if (num <= 1) return 1;  
  
    return num * factorial(num - 1);  
  
}
```

### 5. Implement a function that takes two sorted arrays and merges them into a single sorted array without using any built-in sorting functions.

When interviewers ask this question, they seek to assess the knowledge of algorithms and efficiency in handling sorted data. They also look for the ability to think of and execute a correct solution.



*I can implement a function that can efficiently merge two sorted arrays.*

```
function mergeSortedArrays(arr1, arr2) {  
  
    return [...arr1, ...arr2].sort((a, b) => a - b);  
  
}
```

**6. Write a function that checks if a given string is a palindrome, considering only alphanumeric characters and ignoring case.**

Interviewers analyze the interviewee's approach to execute code and demonstrate familiarity with handling case-sensitive and alphanumeric checks, regular expressions, and JavaScript string methods.

### Sample Answer

copy

```
function isPalindrome(str) {  
  
    const cleanStr = str.replace(/[\^a-zA-Z0-9]/g, "").toLowerCase();  
  
    const reversedStr = cleanStr.split("").reverse().join("");  
  
    return cleanStr === reversedStr;  
  
}
```

**7. Create a JavaScript class for a linked list with methods to insert a node at the beginning, end, or at a specific position, and to delete a node from a given position.**

By asking this question, interviewers can evaluate how well a candidate can design and implement a class for a linked list while also presenting their **problem-solving skills**.





*I would implement a linked list with methods to insert a node at the beginning, end, and at specific positions. Then, I would delete a node from a given position.*

## 8. Implement a function that flattens a nested array in JavaScript, converting it into a single-level array.

Managers can gauge the candidate's logical thinking skills and capability to handle complex data structures. Interviewees should demonstrate their knowledge of loops, recursion, and arrays.

### Sample Answer

[copy](#)

```
const flattenArray = (nestedArray) => {  
  
  return nestedArray.flat(Infinity);  
  
};
```

## 9. Write a function that determines if two strings are anagrams of each other

When interviewers present this question, they aim to measure how well the candidate can use appropriate string-related methods and identify anagrams accurately.

### Sample Answer

[copy](#)

```
function areAnagrams(str1, str2) {  
  
  return str1.split("").sort().join("") === str2.split("").sort().join("");  
  
}
```

## 10. Create a JavaScript function that returns the Fibonacci sequence up to a given number, utilizing memoization for optimized



Interviewees are expected to show their proficiency in OOP and familiarity with recursion and memoization. They can also determine the candidate's attention to detail in class design and organizing code.

### Sample Answer

[copy](#)

*By creating a function that uses an array to store the computed values, a Fibonacci sequence can be generated.*

```
function fibonacciWithMemoization(n) {  
  
  let memo = [0, 1];  
  
  for (let i = 2; i <= n; i++) {  
  
    memo[i] = memo[i - 1] + memo[i - 2];  
  
  }  
  
  return memo;  
  
}
```

## Common JavaScript coding interview questions

Some of the common JavaScript coding interview questions typically cover these topics: checking for palindrome, finding missing/largest numbers, object manipulation, removing duplicates, merging, etc.

### 1. Write a function to check if a given string is a palindrome.

Hiring managers review how well a candidate can handle edge cases while handling case sensitivity, punctuation, and whitespace.





*This function takes a string as input to convert it into lowercase and then compares it with its reverse. The string can be deemed a palindrome if the two match.*

```
function isPalindrome(str) {

    return str.toLowerCase() ===
    str.toLowerCase().split("").reverse().join("");

}
```

## 2. Implement a function to reverse a string without using the built-in reverse() method.

Hiring managers aim to analyze the candidate's knowledge of string manipulation in JavaScript while also measuring their ability to think of alternative solutions.

### Sample Answer

copy

*I would use a for loop to iterate through the characters from the end to the beginning. By appending the character to a new string, it results in the reversed output.*

```
function reverseString(str) {

    let reversed = "";

    for (let i = str.length - 1; i >= 0; i--) {

        reversed += str[i];

    }

    return reversed;

}
```





By presenting the candidates with this question, managers can gauge how well a candidate is familiar with basic JavaScript functions and array manipulation.

### Sample Answer

[copy](#)

*I would use the following functions to find the smallest and largest numbers in the array.*

```
function findMinMax(arr) {  
  
    let min = Math.min(...arr);  
  
    let max = Math.max(...arr);  
  
    return [min, max];  
  
}
```

**4. Write a function that takes an array of integers as input and returns a new array with only the unique elements.**

Hiring managers can evaluate the candidate's knowledge of JavaScript functions, array handling capabilities, and communicating technical concepts.

### Sample Answer

[copy](#)

```
function getUniqueElements(arr) {  
  
    return Array.from(new Set(arr));  
  
}
```

**5. Implement a function to find the factorial of a given number.**





Interviewers also assess the ability to use concise and effective code and provide efficient code implementation.

### Sample Answer

[copy](#)

```
function factorial(number) {  
  
  if (number === 0 || number === 1) return 1;  
  
  return number * factorial(number - 1);  
  
}
```

## 6. Write a function that determines if a given number is prime or not.

By asking this question, interviewers can understand how good the candidate is proficient in math operations and JavaScript logic. The interviewee should excute a clean and optimized solution that is efficient.

### Sample Answer

[copy](#)

```
function isPrime(num) {  
  
  if (num <= 1) return false;  
  
  for (let i = 2; i <= Math.sqrt(num); i++) {  
  
    if (num % i === 0) return false;  
  
  }  
  
  return true;  
  
}
```

## 7. Implement a function to find the sum of all the numbers in an array.







problem-solving capabilities and ability to pay attention to code efficiency.

### Sample Answer

[copy](#)

*I would use the reduce method to implement the following function:*

```
function findSum(arr) {  
  
    return arr.reduce((sum, num) => sum + num, 0);  
  
}
```

**8. Given a string, write a function to count the occurrences of each character in the string.**

Hiring managers expect the candidate to be familiar with string manipulation and loop constructs. When they ask this question, they can evaluate whether the candidate knows data structures.

### Sample Answer

[copy](#)

```
function countCharacterOccurrences(str) {  
  
    const charCount = {};  
  
    for (let char of str) {  
  
        charCount[char] = (charCount[char] || 0) + 1;  
  
    }  
  
    return charCount;  
  
}
```

**9. Implement a function to remove duplicates from an array.**





methods and different approaches to solve the problem.

### Sample Answer

copy 

*The following function duplicates from an array by converting it into a Set. This automatically removes duplicates. Next, the function converts the Set back into an array.*

```
function removeDuplicates(arr) {  
  
    return Array.from(new Set(arr));  
  
}
```

## 10. Write a function that sorts an array of numbers in ascending order.

Interviewees must show their knowledge of bubble sort, merge sort, sorting algorithms, and other approaches. The HR manager aims to measure the capability to execute strong algorithms and handle edge cases.

### Sample Answer

copy 

*I can solve this by using JavaScript's built-in sort method.*

```
function ascendingSort(numbers) {  
  
    return numbers.sort((a, b) => a - b);  
  
}
```

## Tricky JavaScript coding questions

By asking tricky JavaScript coding questions, managers can assess problem—solving skills, JavaScript concepts, and **critical thinking**. These





**1. Write a function that reverses the order of words in a sentence without using the built-in reverse() method.**

This question not only assesses the creativity of the candidates but also helps hiring managers understand how well a candidate can come up with a clean and understandable solution.

### Sample Answer

copy 

```
function reverseSentence(sentence) {  
  
  const words = sentence.split(' ');  
  
  const reversedWords = words.reverse();  
  
  return reversedWords.join(' ');  
  
}
```

**2. Implement a function that checks if a given string is a palindrome (reads the same forwards and backwards) while ignoring whitespace and punctuation.**

Interviewers can gauge the interviewee's capability to handle whitespace and punctuation gracefully while also maintaining the palindrome-checking logic. Candidates must express their knowledge of regular expressions or any other efficient approach.

### Sample Answer

copy 

```
function isPalindrome(str) {  
  
  const cleanedStr = str.replace(/[^\w]/g, "").toLowerCase();  
  
  const reversedStr = cleanedStr.split("").reverse().join("");  
  
  return cleanedStr === reversedStr;  
  
}
```





3. Write a function that takes an array of integers and returns the largest difference between any two numbers in the array.

Candidates should demonstrate their approach to finding the maximum difference between the array elements to handle edge cases and invalid inputs.

### Sample Answer

copy

```
function largestDifference(arr) {  
  
    let min = arr[0];  
  
    let maxDiff = 0;  
  
    for (let i = 1; i < arr.length; i++) {  
  
        if (arr[i] < min) {  
  
            min = arr[i];  
  
        }  
  
        else {  
  
            const diff = arr[i] - min;  
  
            if (diff > maxDiff) {  
  
                maxDiff = diff;  
  
            }  
  
        }  
  
    }  
  
    return maxDiff;  
  
}
```





Interviewers can analyze how well a candidate can effectively communicate code explanations and their familiarity with algorithmic efficiency.

### Sample Answer

copy 

```
function removeDuplicates(arr) {  
  
    return arr.filter((item, index) => arr.indexOf(item) === index);  
  
}
```

5. Write a function that accepts a number and returns its factorial (e.g., factorial of 5 is 5 x 4 x 3 x 2 x 1).

By presenting this question in the interview, hiring managers can assess the capability of the candidate to handle numeric calculations. They can also determine how well the interviewee can pay attention to handling edge cases, if applicable.

### Sample Answer

copy 

```
function factorial(num) {  
  
    if (num === 0 || num === 1) {  
  
        return 1;  
  
    } else {  
  
        return num * factorial(num - 1);  
  
    }  
  
}
```





Interviewers expect the candidates to demonstrate their ability to work with complex data structures and use appropriate techniques to accomplish tasks.

### Sample Answer

[copy](#)

```
function flattenArray(arr) {  
  
    return arr.flat();  
  
}
```

**7. Write a function that checks if a given string is an anagram of another string (contains the same characters in a different order).**

Candidates should showcase how well they can handle complex algorithms and logic. Interviewers are specifically looking for knowledge in string methods, data structures, and loop constructs.

### Sample Answer

[copy](#)

```
function isAnagram(str1, str2) {  
  
    const sortedStr1 = str1.split("").sort().join("");  
  
    const sortedStr2 = str2.split("").sort().join("");  
  
    return sortedStr1 === sortedStr2;  
  
}
```

**8. Implement a function that finds the second smallest element in an array of integers.**

Interviewers can measure the candidate's problem-solving skills and their understanding of conditional statements, loops, and arrays.



```
function secondSmallest(arr) {  
  
    const sortedArr = arr.sort((a, b) => a - b);  
  
    return sortedArr[1];  
  
}
```

### 9. Write a function that generates a random alphanumeric string of a given length.

By asking this question, interviewers can understand how well a candidate can ensure the function produces a reliable and consistent random output.

#### Sample Answer

copy 

```
function generateRandomString(length) {  
  
    const characters =  
    'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789'  
  
    let result = "";  
  
    for (let i = 0; i < length; i++) {  
  
        const randomIndex = Math.floor(Math.random() *  
characters.length);  
  
        result += characters.charAt(randomIndex);  
  
    }  
  
    return result;  
  
}
```





*Hiring managers can gauge a candidate's capability to implement coding solutions and create an efficient algorithm.*

### Sample Answer

[copy](#)

```
function toRomanNumeral(number) {  
  
  // Implement your code here  
  
}
```

## JavaScript array coding questions

JavaScript array coding interview questions are technical questions asked to gauge candidates' ability to work with arrays along with their familiarity with fundamental data structures.

### 1. Write a function that returns the sum of all numbers in an array.

By asking such a question, [hiring managers](#) can evaluate whether the candidate would be able to perform common tasks and solve basic coding challenges.

### Sample Answer

[copy](#)

```
function sumArray(arr) {  
  
  return arr.reduce((total, num) => total + num, 0);  
  
}
```

### 2. Implement a function that finds the maximum number in an array.





understand the capability to communicate technical solutions.

### Sample Answer

copy

```
function findMaxNumber(arr) {  
  
    let max = arr[0];  
  
    for (let i = 1; i < arr.length; i++) {  
  
        if (arr[i] > max) {  
  
            max = arr[i];  
  
        }  
  
    }  
  
    return max;  
  
}
```

### 3. Write a function that returns a new array containing only the unique elements from an input array.

The hiring manager is specifically looking for candidates who can demonstrate an understanding in data manipulation and JavaScript arrays. Additionally, interviewers evaluate how well the candidate strives for an optimized solution without duplicate elements.

### Sample Answer

copy

```
function getUniqueElements(inputArray) {  
  
    return [...new Set(inputArray)];  
  
}
```





By asking this question, hiring managers can assess the candidate's knowledge of arithmetic operations, array manipulation, and looping.

### Sample Answer

copy 

```
function calculateAverage(numbers) {  
  
    let sum = 0;  
  
    for (let number of numbers) {  
  
        sum += number;  
  
    }  
  
    return sum / numbers.length;  
  
}
```

## 5. Write a function that sorts an array of strings in alphabetical order.

When interviewers present this question in the interview, they expect the candidate to be familiar with sorting algorithms and JavaScript array manipulation.

### Sample Answer

copy 

```
function sortStrings(arr) {  
  
    return arr.slice().sort();  
  
}
```

## 6. Implement a function that finds the index of a specific element in an array. If the element is not found, the function should return -1.





implement the function proper error handling.

### Sample Answer

copy

```
function findElementIndex(arr, element) {  
  
    const index = arr.indexOf(element);  
  
    return index !== -1 ? index : -1;  
  
}
```

7. Write a function that removes all falsy values (false, null, 0, "", undefined, and NaN) from an array.

Candidates must showcase **communication skills** and explain their solutions logically. Interviewers analyze the interviewee's ability to write a function that filters false values from an array.

### Sample Answer

copy

```
function removeFalsyValues(arr) {  
  
    return arr.filter(Boolean);  
  
}
```

8. Implement a function that merges two arrays into a single array, alternating elements from each array.

Hiring managers determine a candidate's ability to craft efficient algorithms and knowledge of array manipulation.

### Sample Answer

copy

```
function mergeArrays(array1, array2) {  
  
    const mergedArray = [];
```





```
for (let i = 0; i < maxLength; i++) {  
  
    if (i < array1.length) mergedArray.push(array1[i]);  
  
    if (i < array2.length) mergedArray.push(array2[i]);  
  
}  
  
return mergedArray;  
  
}
```

### 9. Write a function that finds the second largest number in an array.

Such a question reveals to the interviewers how well the candidate can use loops and array methods, work with them, and utilize logic to find solutions.

#### Sample Answer

[copy !\[\]\(ec9132f1d27c8919987d92907322654d\_img.jpg\)](#)

```
function findSecondLargest(arr) {  
  
    arr.sort((a, b) => b - a);  
  
    return arr[1];  
  
}
```

### 10. Implement a function that groups elements in an array based on a given condition. For example, grouping even and odd numbers into separate arrays.

When interviews ask this question, they aim to evaluate a candidate's understanding of concepts like array methods, conditional statements, and other technical concepts. Candidate should demonstrate good coding style.





```
function groupByCondition(arr, condition) {  
  
  return [  
  
    arr.filter(element => condition(element)),  
  
    arr.filter(element => !condition(element))  
  
  ];  
  
}
```

## Tips to prepare for a JavaScript coding interview

There are 5 tips a candidate should keep in mind when attending a JavaScript coding interview:

- **Master JavaScript basics:** Candidates must have a solid understanding of core JavaScript fundamentals, including variables, loops, conditionals, objects, and data types. Additionally, [practicing coding skills](#) is important.

**Note:** Many interviewers skip frameworks initially and dive straight into raw JavaScript to test depth of knowledge.

- **Explore popular libraries and frameworks:** Familiarizing oneself with the most common JavaScript libraries and frameworks like Vue.js, React, etc., helps understand key concepts.
- **Practice whiteboard Java coding:** [Java coding skills](#) can further be sharpened by practicing whiteboard coding. Candidates should solve coding challenges and explain their thought processes loudly. They should mainly focus on simplicity, clarity, and efficiency.





than just the final solution.

- **Test code:** After writing Java solutions, test it with various inputs and ensure it works correctly and can handle edge cases. Consider the time complexity of the solutions and aim for efficiency.
- **Review JavaScript projects :** Discussing JavaScript projects in the interview is essential. Interviewees should explain their approach clearly and how they overcame the challenges.

## Red Flags

There are some red flags hiring managers should watch out for in a JavaScript coding interview:

**Lack of basic JavaScript knowledge:** Struggling with core concepts like variables, functions, or objects indicates weak foundations, making it hard to grow into advanced roles.

**Providing efficient solutions:** Writing code that doesn't scale, has high time complexity, or ignores optimization shows poor problem-solving habits.

**No practical problem-solving abilities:** Candidates who cannot translate theory into working solutions can struggle in real-world coding tasks.

**Limited knowledge of asynchronous programming:** Not understanding promises, async/wait, or callbacks shows gaps in handling modern web applications.

**Copy-pasting code:** Reliance on memorized or external snippets suggests lack of originality and inability to solve problems independently.

**Inability to articulate thought processes:** Failing to explain logic or choices makes collaboration and debugging harder in team environment.

**Unfamiliarity with Modern JavaScript:** Gaps in ES6+ features like arrow functions, destructuring, or modules reveal outdated skills.





# Common Categories of JavaScript Coding Interview Questions

JavaScript interviews are designed to test both theoretical knowledge and practical coding ability. Questions usually fall into a few broad categories, covering everything from language fundamentals to system design. Below are the most common areas interviewers can focus on:

## 1) Core JavaScript Fundamentals

Covers concepts like variables, scope, closures, hoisting, prototypes, inheritance, and asynchronous programming. These test how well the candidate understands the building blocks of the language.

## 2) Data Structures and Algorithms

Focuses on arrays, objects, strings, recursion, sorting, searching, and problem-solving efficiency. Candidates are often asked to explain time and space complexity.

## 3) Front-End Development and Frameworks

Assesses knowledge of React, Angular, or Vue, along with DOM manipulation, state management, and event handling. Employers want to see if you can build responsive and interactive UIs.

## 4) Back-End Development with Node.js

Evaluates understanding of server-side JavaScript, APIs, Express, databases, and the event loop. These questions test readiness for full-stack or back-end roles.

## 5) Tools, Best Practices, and System Design

Covers debugging with DevTools, performance optimization, modular code structure, design patterns, and sometimes high-level system design scenarios.





[Java programming interview questions and Answers](#)

[Top 40 Java Interview Questions and Answers 2023](#)

## Related Job Descriptions

[Java Developer Job Description](#)

[Full Stack Developer Job Description Template](#)

### Subscribe to our Newsletter

Curated HR content delivered to you, bi-monthly!

Subscribe







**One tool  
for all your  
HR needs**

**Book Keka Demo**



Keka is made for your people, by people like you, who care for people in the organization





## Core HR

HR Software

ESS Portal

Employee Profiles

Documents

Helpdesk

Pulse Surveys

HR Analytics

## Payroll

Payroll Software

Compliance

Expense Management

Compensation

Loans & Advances

Employee Finances

## Hiring & Onboarding

Applicant Tracking System

Hiring Software

Offer Management

Employee Onboarding

## Learn

Learning Management System

## Company

About us

Customer Stories





News

---

Marketplace

Security

**We are hiring !**

Time Attendance

Attendance Management

Leave Management

GPS / Mobile Attendance

Shift Management

Performance

Performance Management

360 degree reviews

OKR Software

One on One Meeting

Continuous Feedback

PSA

PSA Software

Resource Management

Project Management

Timesheet Software

Billing System

Analytics

Opportunity Management

Contact





For Support:

[support@keka.com](mailto:support@keka.com)

12th & 14th Floor, Vasavi Sky City,

Telecom Nagar, Gachibowli,

Hyderabad, Telangana 500032

---

2025 Keka Technologies Private Limited All  
rights reserved

[Security  
Policy](#)

[Terms  
of  
Service](#)

[Privacy  
Policy](#)

[Cookie  
Policy](#)

[DPA](#)

