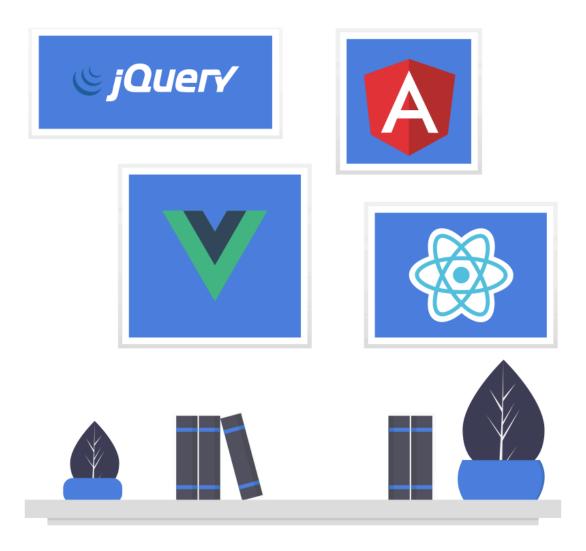# 30 JavaScript Coding Interview Questions for Beginner, Mid-Level and Expert Developers

March 29, 2023

JavaScript is a leading programming language for creating interactive and dynamic web pages. You will see a lot of popular frameworks and libraries based on

that JavaScript is the main programming language for the front-end development of software applications.

Today we will guide you on how to design coding challenges for evaluating JavaScript developers. We will suggest coding challenges for beginner, mid-level and expert JavaScript developers. Let's start with coding challenges that are suitable for beginner-level developers.

Interview Questions for Beginner JavaScript Developers

When preparing interview questions for beginner-level JavaScript developers, focus on basic JavaScript syntax, array and string manipulation, objects and JSON, DOM manipulation, error handling and asynchronous programming. Here are 10 sample questions in this regard:

Question 1

What will be the output of the below code:

```
const array = [10, 20, 30, 40];
```

```
15);
console.log(result);
```

Question 2

Find the issue in the below code snippet.

```
 let counter = 0;
for (var i = 1; i <= 10; i++) {
  counter+= i;
}
console.log(counter);
console.log(i);
```

Question 3

Analyze the below code. Do you see any issue? If yes, what is that issue?

```
const object1 = {
  a: 10,
  b: 20,
  c: function () {
      console.log(this.a + this.b);
  },
};
const func = object1.c;
func();
```

Question 4

Create a JavaScript function that calculates the tip for a given bill amount and tip

will be calculated tip value.

Question 5

What will be the output of below code snippet:

```javascript
function greetHello(name) {
   return `Hello, ${name}!`;
}
console.log(greetHello("Brian"));
```

Question 6

Will the below code return any error? If yes, identify the error.

```javascript
function fetchData(callback) {
  fetch('https://api.example.com/data')
       .then(response => response.json())
       .then(data => callback(null, data))
       .catch(error => callback(error));
}
 fetchData(function (error, data) {
  if (error) {
       console.log('Error:', error);
  } else {
       console.log('Data:', data);
  }
});
```

Question 7

Implement a simple shopping cart system with features to add items, remove items

the item name, price and quantity. Implement features to add items to the cart, remove items and calculate the total cost.

Question 8

Analyze the below code snippet and advise what will be the output:

```
const person = {
  firstName: "Helen",
  lastName: "Ryan",
  getFullName: function () {
        return this.firstName + " " + this.lastName;
  },
};
console.log(person.getFullName());
```

Question 9

Find the issue with the below code snippet:

```
setTimeout(function () {
    console.log("This will be executed after 3 seconds");
}, 3000);
clearTimeout();
```

Question 10

What issue exists in the below code:

```
const testArray = [1, 2, 3];
testArray = [4, 5, 6];
```

Interview Questions for Mid-level JavaScript Developers

When designing interview questions for mid-level JavaScript developers, you should prepare challenges to test the understanding of advanced JavaScript concepts and problem-solving skills. Some areas that should be considered for evaluation include functional programming, asynchronous programming, promises, working with APIs and advanced JavaScript features. Find below 10 coding challenges which are suited to mid-level JavaScript developers:

Question 1

What is the issue in the below code:

```
  const fetchData = async () => {
    const response = await
fetch("https://api.samplewebsite.com/data");
    const data = await response.json();
    console.log(data);
};
fetchData();
```

Question 2

What will be the output of the below code:

```
const promise1 = Promise.resolve("One");
const promise2 = new Promise((resolve) => setTimeout(() =>
```

```
Promise.allSettled([promise1, promise2, promise3]).then((results)
=> console.log(results));
```

Question 3

Develop a simple URL shortener service using JavaScript. Implement a function that takes a long URL as an input parameter and the output will be a shortened URL. Create a reverse function as well. The reverse function takes the shortened URL and returns the original long URL. You can use simple in-memory objects to store the mapping between long and short URLs.

Question 4

Implement an autocomplete feature for a search input field. Given an array of words, write a function that suggests words based on the current input. The output of the function will be an array of suggested words that start with the input characters, limiting the number of suggestions (e.g., a maximum of 7 suggestions).

Question 5

What is the issue in the below code:

```
const obj = {
    name: "Conner",
    age: 27,
    greet: () => {
        console.log(`Hey, my name is ${this.name}`);
    },
};
obj.greet();
```

Question 6

What will be the output of below code snippet:

```javascript
const object1 = {
  prop1: "value1",
  prop2: {
        prop3: "value3",
  },
};
 const newObj = { ...obj };
newObj.prop2.prop3 = "newValue3";
console.log(object1.prop2.prop3);
```

Question 7

Will the below code return any error? If yes, what will be the error?

```javascript
class Bird {
  constructor(name) {
        this.name = name;
  }
  speak() {
        console.log(`${this.name} makes a noise.`);
  }
}
 class Crow extends Bird{
  speak() {
        super.speak();
        console.log(`${this.name} sings.`);
  }
}
```

Question 8

Develop a function that throttles another function, allowing it to be called at most once every specified interval (e.g., 300ms). The throttling function will have two input parameters. One will be the function to be throttled and the second will be the interval in milliseconds. The throttled function should be called with the same arguments as the original function.

Question 9

What is wrong with the below code:

```
const arr = [1, 2, 3, 4, 5];
const sum = arr.reduce((total, num) => total + num);
console.log(sum / arr.length);
```

Question 10

CodeInterview   Product   Pricing   Contact Sales   Log In   Start for free

meeting when both people will be available. Each interval is represented as an array of two integers, where the first integer is the start time and the second integer is the end time.

Interview Questions for Expert JavaScript Developers

When preparing coding challenges for expert-level JavaScript engineers, you should test the advanced features of the language and performance optimization techniques. Some of the areas to evaluate include advanced JavaScript features, code architecture, design patterns, performance optimization and security. Below we have presented 10 coding challenges for expert JavaScript developers:

Question 1

Is there any security vulnerability in the below code? If yes, identify it:

```javascript
const username = document.getElementById('username').value;
const password = document.getElementById('password').value;
fetch('https://api.examplewebsite.com/login', {
  method: 'POST',
  body: JSON.stringify({ username, password })
})
.then(response => response.json())
.then(data => console.log(data))
.catch(error => console.log(error));
```

Question 2

Identify the output of the below code.

```javascript
const testArray = [1, 2, 3, 4, 5];
const res = testArray.reduce((acc, curr) => {
  if (curr % 2 === 0) {
        return acc + curr;
  }
  return acc;
}, 0);
```

Question 3

What is the possible performance issue in the below code?

```javascript
const arr = [];
for (let i = 0; i < 1000000; i++) {
   arr.push(Math.floor(Math.random() * 1000));
}
```

Question 4

Suggest the output of the below code:

```javascript
const arr = [1, 2, 3];
const object1 = { x: 1, y: 2, z: 3 };
console.log([...arr, ... object1]);
```

Question 5

Design a social media platform that contains features like sign up, creating a profile and posting status updates. Users should be able to follow other users and view their posts on a newsfeed.

Question 6

What is wrong with the below call to the API?

```javascript
fetch('https://api.example.com/data')
   .then(response => {
       if (!response.ok) {
```

```
      return response.json();
    })
    .then(data => console.log(data))
    .catch(error => console.log(error));
```

Question 7

What will be the output of below code snippet?

```
const promise1 = Promise.resolve(One);
const promise2 = Promise.resolve(Two);
Promise.all([promise1, promise2]).then(([result1, result2]) =>
console.log(result1 + ' ' + result2));
```

Question 8

Design an online code editor where users can write, save and run JavaScript code. The editor should include features like syntax highlighting, auto-completion and error checking.

Question 9

The below code snippet uses closures to implement a counter. How will you optimize it to minimize memory usage:

```
function counter() {
  let count = 0;
  return function() {
        count++;
        console.log(count);
    }
```
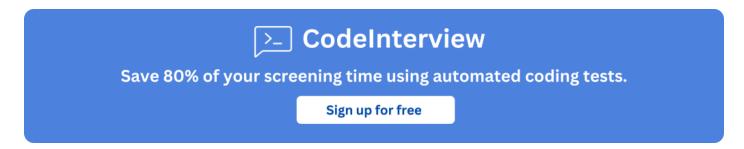
```
increment(); // 1
increment(); // 2
increment(); // 3
```

Question 10

Develop a fitness tracker application where users can enter their daily exercise routines and track their progress over time. The application should allow users to set goals, view their progress and receive reminders.

Conclusion

Evaluating JavaScript developers at varying skill levels requires well-designed code challenges. Syntax, data types and functions are some basic concepts to test first. More sophisticated topics like object-oriented programming, algorithms and data structures should be included when testing mid-level engineers. Complex issues like performance optimization, security and design patterns should be used to evaluate expert-level developers.

This article has presented numerous examples of coding challenges to help hiring managers spot promising JavaScript developers at different levels of experience.

[Java Coding Interview Questions](#)

[Python Coding Interview Questions](#)

[ReactJS Coding Interview Questions](#)

[C# Coding Interview Questions](#)

[C++ Coding Interview Questions](#)

[PHP Coding Interview Questions](#)

[73 Advanced Coding Interview Questions](#)

Share on:

Products

Tour

Pricing

API Docs

Login

Sign Up Free

Customers

Guide for Interviewers

Support

Contact Sales

Company

About

Blog

Careers

Contact us

Compilers

Online Java Compiler

Online Python Compiler

Online C Compiler

Online Scala Compiler

Online Ruby Compiler

Legal

Privacy Policy

Terms of Services

© 2024 Remote Interview, Inc