

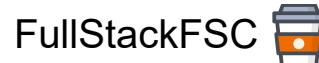


# Kill Your Tech Interview

3877 Full-Stack, Algorithms & System Design Interview Questions

Answered To Get Your Next Six-Figure Job Offer

[See All Questions](#)[Data Science & ML QAs](#) [Full-Stack, Web & Mobile](#) [Algorithms & Data Structures](#) [System Design](#) [Full-Stack](#) [Algorithms](#) [System Design](#) [.NET Core 68](#) [ADO.NET 33](#) [ASP.NET 54](#) [ASP.NET MVC 36](#) [ASP.NET Web API 33](#) [Agile & Scrum 47](#) [Android 113](#) [Angular 120](#) [AngularJS 61](#) [Azure 59](#) [Azure Service Bus 25](#) [C# 127](#) [CSS 50](#) [Cosmos DB 46](#) [Dependency Injection 17](#) [Design Patterns 68](#) [DevOps 44](#) [Entity Framework 57](#) [Flutter 67](#) [Git 36](#) [Golang 49](#) [GraphQL 25](#) [HTML5 55](#)



I'm Hiring  
Devs

Café

Laravel 41

MongoDB 77

MySQL 60

Node.js 120

OOP 58

Objective-C 43

PHP 82

PWA 22

Python 112

React 130

React Hooks 46

React Native 72

Reactive Programming 12

Redis 25

Redux 35

Ruby 84

Ruby on Rails 72

Rust 38

SQL 42

Spring 87

Swift 72

T-SQL 51

TypeScript 91

UX Design 78

Unit Testing 24

Vue.js 41

WCF 37

WPF 46

Web Security 58

WebSockets 26

Xamarin 83

iOS 36

jQuery 51



Having Machine Learning & DS Interview? Check MLStack.Cafe - 1704 Data Science & ML Interview Questions & Answers! Having ML & DS Interview? Check MLStack.Cafe - 1704 ML & DS Interview Questions and Answers

FullStackFSC

I'm Hiring  
Devs

Café



# 25+ JavaScript Coding Interview Questions (SOLVED with CODE)

JavaScript 179



and tricky JavaScript Coding Interview Questions and Challenges to crack on your next senior web developer interview and got your next six-figure job offer in no time!

## Q1: Explain what a *callback* function is and provide a simple example

Junior

JS JavaScript 179

### Answer

A `callback` function is a function that is passed to another function as an argument and is executed after some operation has been completed. Below is an example of a simple callback function that logs to the console *after* some operations have been completed.

```
function modifyArray(arr, callback) {
    // do something to arr here
    arr.push(100);
    // then execute the callback function that was passed
    callback();
}

var arr = [1, 2, 3, 4, 5];

modifyArray(arr, function() {
    console.log("array has been modified", arr);
});
```



I'm Hiring  
Devs

Café

WORD IN THE SENTENCE

## Problem

For example `Welcome to this Javascript Guide!` should be become `emocleW ot siht tpircsavaJ !ediuG`

## Answer

```
var string = "Welcome to this Javascript Guide!";

// Output becomes !ediuG tpircsavaJ siht ot emocleW
var reverseEntireSentence = reverseBySeparator(string, "");

// Output becomes emocleW ot siht tpircsavaJ !ediuG
var reverseEachWord = reverseBySeparator(reverseEntireSentence, " ");

function reverseBySeparator(string, separator) {
    return string.split(separator).reverse().join(separator);
}
```

Having Tech or Coding Interview? Check 179 JavaScript Interview Questions

Source: <https://github.com/kennymkchan>

**Q3: How to check if an object is an array or not? Provide some code.**

Junior

JS JavaScript 179

## Answer

The best way to find whether an object is instance of a particular class or not using `toString` method from `Object.prototype`



I'm Hiring  
Devs

## Café

One of the best use cases of type checking of an object is when we do method overloading in JavaScript. For understanding this let say we have a method called `greet` which takes one single string and also a list of strings, so making our `greet` method workable in both situations we need to know what kind of parameter is being passed, is it single value or list of values?

```
function greet(param) {
  if() {
    // here have to check whether param is array or not
  }
  else {
  }
}
```

However, in above implementation it might not be necessary to check type for array, we can check for single value string and put array logic code in else block, let see below code for the same.

```
function greet(param) {
  if(typeof param === 'string') {
  }
  else {
    // If param is of type array then this block of code would execute
  }
}
```

Now it's fine we can go with above two implementations, but when we have a situation like a parameter can be `single value`, `array`, and `object` type then we will be in trouble.

Coming back to checking type of object, As we mentioned that we can use `Object.prototype.toString`

```
if(Object.prototype.toString.call(arrayList) === '[object Array]') {
  console.log('Array!');
}
```



I'm Hiring  
Devs

Café

```
    console.log('Array');
} else {
    console.log('Not an array');
}
```

FYI jQuery uses `Object.prototype.toString.call` internally to check whether an object is an array or not.

In modern browser, you can also use:

```
Array.isArray(arrayList);
```

`Array.isArray` is supported by Chrome 5, Firefox 4.0, IE 9, Opera 10.5 and Safari 5

---

Having Tech or Coding Interview? Check 179 JavaScript Interview Questions      Source: [github.com/ganqqwerty](https://github.com/ganqqwerty)

## Q4: How to empty an array in JavaScript?

Junior

JS    JavaScript 179

### Problem

```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f'];
```

How could we empty the array above?

### Answer

#### Method 1



I'm Hiring  
Devs

## Café

Above code will set the variable `arrayList` to a new empty array. This is recommended if you don't have **references to the original array** `arrayList` anywhere else because It will actually create a new empty array. You should be careful with this way of empty the array, because if you have referenced this array from another variable, then the original reference array will remain unchanged, Only use this way if you have only referenced the array by its original variable `arrayList` .

For Instance:

```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f']; // Created array
var anotherArrayList = arrayList; // Referenced arrayList by another variable
arrayList = []; // Empty the array
console.log(anotherArrayList); // Output ['a', 'b', 'c', 'd', 'e', 'f']
```

## Method 2

```
arrayList.length = 0;
```

Above code will clear the existing array by setting its length to 0. This way of empty the array also update all the reference variable which pointing to the original array. This way of empty the array is useful when you want to update all the another reference variable which pointing to `arrayList` .

For Instance:

```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f']; // Created array
var anotherArrayList = arrayList; // Referenced arrayList by another variable
arrayList.length = 0; // Empty the array by setting length to 0
console.log(anotherArrayList); // Output []
```

## Method 3

```
arrayList.splice(0, arrayList.length);
```



```
var arrayList = ['a', 'b', 'c', 'd', 'e', 'f']; // Created array
var anotherArrayList = arrayList; // Referenced arrayList by another variable
arrayList.splice(0, arrayList.length); // Empty the array by setting length to 0
console.log(anotherArrayList); // Output []
```

#### Method 4

```
while(arrayList.length) {
    arrayList.pop();
}
```

Above implementation can also empty the array. But not recommended to use often.

Having Tech or Coding Interview? Check 179 JavaScript Interview Questions

Source: [github.com/ganqqwerty](https://github.com/ganqqwerty)

## Q5: How would you check if a number is an integer?

Junior

JS JavaScript 179

### Answer

A very simple way to check if a number is a decimal or integer is to see if there is a remainder left when you divide by 1.

```
function isInt(num) {
    return num % 1 === 0;
}

console.log(isInt(4)); // true
console.log(isInt(12.2)); // false
console.log(isInt(0.3)); // false
```



I'm Hiring  
Devs

Café

## 60 Java and Spring Interview Questions You Must Know

Design Patterns 68

Java 165

OOP 58

### Q6: Implement enqueue and dequeue using only two stacks

Junior

JavaScript 179

#### Answer

*Enqueue* means to add an element, *dequeue* to remove an element.

```
var inputStack = [] // First stack
var outputStack = [] // Second stack

// For enqueue, just push the item into the first stack
function enqueue(stackInput, item) {
    return stackInput.push(item);
}

function dequeue(stackInput, stackOutput) {
    // Reverse the stack such that the first element of the output stack is the
    // last element of the input stack. After that, pop the top of the output to
    // get the first element that was ever pushed into the input stack
    if (stackOutput.length <= 0) {
        while(stackInput.length > 0) {
            var elementToOutput = stackInput.pop();
            stackOutput.push(elementToOutput);
        }
    }

    return stackOutput.pop();
}
```



## Q7: Make this work

Junior

JS JavaScript 179

### Problem

```
duplicate([1, 2, 3, 4, 5]); // [1,2,3,4,5,1,2,3,4,5]
```

### Answer

```
function duplicate(arr) {
    return arr.concat(arr);
}

duplicate([1, 2, 3, 4, 5]); // [1,2,3,4,5,1,2,3,4,5]
```

Having Tech or Coding Interview? Check [👉 179 JavaScript Interview Questions](#)

Source: [github.com/yangshun](https://github.com/yangshun)

## Q8: Write a "mul" function which will properly handle the syntax below

Junior

JS JavaScript 179

### Problem

```
console.log(mul(2)(3)(4)); // output : 24
console.log(mul(4)(3)(4)); // output : 48
```

### Answer



```
        return x * y * z;
    );
);
}
}
```

Here `mul` function accept the first argument and return anonymous function which take the second parameter and return anonymous function which take the third parameter and return multiplication of arguments which is being passed in successive

In JavaScript function defined inside has access to outer function variable and function is the first class object so it can be returned by function as well and passed as argument in another function.

- A function is an instance of the Object type
- A function can have properties and has a link back to its constructor method
- Function can be stored as variable
- Function can be pass as a parameter to another function
- Function can be returned from function

*Having Tech or Coding Interview? Check 179 JavaScript Interview Questions*

Source: [github.com/ganqqwerty](https://github.com/ganqqwerty)



**Full-Stack, Web & Mobile**



**Algorithms & Data Structures**



**Full-Stack**



**Algorithms**



**System Design**



**C#** 127



**ASP.NET MVC** 36



**Unit Testing** 24



**Xamarin** 83



**MySQL** 60



**WebSockets** 26



**Dependency Injection** 17



**Vue.js** 41



**TypeScript** 91



**Ruby on Rails** 72

**MLStack.Cafe - 1704 ML & DS Interview Questions and Answers**

## Q9: Write a function that would allow you to do this?

Junior

JS JavaScript 179

### Problem

```
var addSix = createBase(6);
addSix(10); // returns 16
addSix(21); // returns 27
```

### Answer

You can create a closure to keep the value passed to the function `createBase` even after the inner function is returned. The inner function that is being returned is created within an outer function, making it a closure, and it has access to the variables within the outer function, in this case the variable `baseNumber`.

```
function createBase(baseNumber) {
    return function(N) {
        // we are referencing baseNumber here even though it was declared
        // outside of this function. Closures allow us to do this in JavaScript
        return baseNumber + N;
    }
}

var addSix = createBase(6);
addSix(10);
addSix(21);
```



## 32 Advanced Data Structures Interview Questions (ANSWERED) To Smash Your Next Programming Interview

88 Data Structures 61

### Q10: FizzBuzz Challenge

Mid



JavaScript 179

#### Problem

Create a for loop that iterates up to `100` while outputting "fizz" at multiples of `3` , "buzz" at multiples of `5` and "fizzbuzz" at multiples of `3` and `5` .

#### Answer

Check out this version of FizzBuzz:

```
for (let i = 1; i <= 100; i++) {  
  let f = i % 3 == 0,  
      b = i % 5 == 0;  
  console.log(f ? (b ? 'FizzBuzz' : 'Fizz') : b ? 'Buzz' : i);  
}
```

Having Tech or Coding Interview? Check [👉 179 JavaScript Interview Questions](#)

Source: [github.com/yangshun](https://github.com/yangshun)

### Q11: Given two strings, return true if they are anagrams of one another

Mid



JavaScript 179



## Answer

```
var firstWord = "Mary";
var secondWord = "Army";

isAnagram(firstWord, secondWord); // true

function isAnagram(first, second) {
    // For case insensitivity, change both words to lowercase.
    var a = first.toLowerCase();
    var b = second.toLowerCase();

    // Sort the strings, and join the resulting array to a string. Compare the results
    a = a.split("").sort().join("");
    b = b.split("").sort().join("");

    return a === b;
}
```

Having Tech or Coding Interview? Check 179 JavaScript Interview Questions

Source: <https://github.com/kennymkchan>

## Q12: How would you use a closure to create a private counter?

Mid



JavaScript 179

## Answer

You can create a function within an outer function (a closure) that allows you to update a private variable but the variable wouldn't be accessible from outside the function without the use of a helper function.



I'm Hiring  
Devs

Café

```
// to modify the private _counter variable
return {
    add: function(increment) { _counter += increment; },
    retrieve: function() { return 'The counter is currently at: ' + _counter; }
}

// error if we try to access the private variable like below
// _counter;

// usage of our counter function
var c = counter();
c.add(5);
c.add(9);

// now we can access the private variable in the following way
c.retrieve(); // => The counter is currently at: 14
```

Having Tech or Coding Interview? Check 179 JavaScript Interview Questions

Source: coderbyte.com

Mid



JavaScript 179

## Q13: Provide some examples of non-boolean value coercion to a boolean one

### Answer

The question is when a non-boolean value is coerced to a boolean, does it become `true` or `false`, respectively?

The specific list of "falsy" values in JavaScript is as follows:

- `""` (empty string)



Any value that's not on this "falsy" list is "truthy." Here are some examples of those:

- "hello"
- 42
- true
- [ ] , [ 1, "2", 3 ] (arrays)
- { } , { a: 42 } (objects)
- function foo() { ... } (functions)

Having Tech or Coding Interview? Check 179 JavaScript Interview Questions      Source: FullStack.Cafe

## 20 CI/CD Interview Questions DevOps Engineers Should Know

DevOps 44

Docker 64

Microservices 34

**Q14: What will be the output of the following code?**

Mid

JavaScript 179

### Problem

```
var y = 1;
if (function f() {}) {
  y += typeof f;
```



Above code would give output `undefined`. If condition statement evaluate using `eval` so `eval(function f() {})` which return `function f() {}` which is true so inside if statement code execute. `typeof f` return undefined because if statement code execute at run time, so statement inside `if` condition evaluated at run time.

```
var k = 1;
if (1) {
    eval(function foo() {});
    k += typeof foo;
}
console.log(k);
```

Above code will also output `undefined`.

```
var k = 1;
if (1) {
    function foo() {};
    k += typeof foo;
}
console.log(k); // output 1function
```

---

Having Tech or Coding Interview? Check ➔ 179 JavaScript Interview Questions

Source: [github.com/ganqqwerty](https://github.com/ganqqwerty)

## Q15: What will the following code output?

### Problem

Mid



JavaScript

179



```
console.log(b);
```

## Answer

The code above will output 5 even though it seems as if the variable was declared within a function and can't be accessed outside of it. This is because

```
var a = b = 5;
```

is interpreted the following way:

```
var a = b;  
b = 5;
```

But `b` is not declared anywhere in the function with `var` so it is set equal to 5 in the *global scope*.

---

Having Tech or Coding Interview? Check 179 JavaScript Interview Questions

Source: coderbyte.com



Full-Stack, Web & Mobile



Algorithms & Data Structures



Full-Stack



Algorithms



System Design



Laravel 41



Xamarin 83



React Hooks 46



React Native 72



Unit Testing 24



Flutter 67



JavaScript 179



Java 165



Entity Framework 57



WCF 37

**MLStack.Cafe - 1704 ML & DS Interview Questions and Answers**

## Q16: Write a function that would allow you to do this

Mid



JavaScript 179

### Problem

`multiply(5)(6);`

### Answer

You can create a *closure* to keep the value of `a` even after the inner function is returned. The inner function that is being returned is created within an outer function, making it a closure, and it has access to the variables within the outer function, in this case the variable `a`.

```
function multiply(a) {  
    return function(b) {  
        return a * b;  
    }  
}  
  
multiply(5)(6);
```

Having Tech or Coding Interview? Check [👉 179 JavaScript Interview Questions](#)

Source: [coderbyte.com](#)



I'm Hiring  
Devs

Café

WORK PROVIDE SOME CODE

## examples

### Answer

Join **FullStack.Cafe** to open this Answer. It's Free!



Get Free Access To Answer

Join 120k+ Developer Who Trust FullStack.Cafe

## The Ultimate Guide to Transitioning to Remote Web Development

\$ Career 1

### Q18: How would you create a private variable in JavaScript?

Senior

JS JavaScript 179

### Answer

Join **FullStack.Cafe** to open this Answer. It's Free!



Get Free Access To Answer

Join 120k+ Developer Who Trust FullStack.Cafe



I'm Hiring  
Devs

Café

JavaScript: Provide an example

## Answer

Join **FullStack.Cafe** to open this Answer. It's Free!

 Get Free Access To Answer

Join 120k+ Developer Who Trust FullStack.Cafe

## Q20: What will be the output of the following code?

Senior

 JavaScript 179

## Answer

Join **FullStack.Cafe** to open this Answer. It's Free!

 Get Free Access To Answer

Join 120k+ Developer Who Trust FullStack.Cafe

## Q21: What will be the output of the following code?

Senior

 JavaScript 179

## Answer

Join **FullStack.Cafe** to open this Answer. It's Free!

 Get Free Access To Answer



I'm Hiring  
Devs

Café

## 23 PowerShell Interview Questions DevOps Engineers Must Master

PowerShell 24

**Q22: What will the following code output?**

Senior

JavaScript 179

### Answer

Join **FullStack.Cafe** to open this Answer. It's Free!

Join 120k+ Developer Who Trust FullStack.Cafe



**Full-Stack, Web & Mobile**



**Algorithms & Data Structures**



**Full-Stack**



**Algorithms**



**System Design**

**Design Patterns** 68

**Android** 113

**WCF** 37

**C#** 127

**Cosmos DB** 46

**JavaScript** 179

**ADO.NET** 33

**Java** 165

**PHP** 82

**GraphQL** 25

FullStackFSC

I'm Hiring  
Devs

Café

MLStack.Cafe - 1704 ML &amp; DS Interview Questions and Answers

## Q23: When would you use the `bind` function?

Senior

JS JavaScript 179

### Answer

Join **FullStack.Cafe** to open this Answer. It's Free!



Get Free Access To Answer

Join 120k+ Developer Who Trust FullStack.Cafe

## Q24: Write a recursive function that performs a binary search

Senior

JS JavaScript 179

### Answer

Join **FullStack.Cafe** to open this Answer. It's Free!



Get Free Access To Answer

Join 120k+ Developer Who Trust FullStack.Cafe

## Q25: Describe the Revealing Module Pattern design pattern

Expert

JS JavaScript 179

FullStackFSC

I'm Hiring  
Devs

Café

Share this blog post to open Expert question!



## 14 Fibonacci Interview Questions (SOLVED) To Brush Before Coding Interview

 Fibonacci Series 14

Unlock 3877 Answers

**Top 27 Rust Interview Questions (ANSWERED) For Backend and System Developers**

 Rust 38

**15 Common Clean Architecture Interview Questions (ANSWERED) For Software Architects and Devs**

**21 Azure Service Bus Interview Questions (SOLVED) For Azure Cloud Developers**

 Azure Service Bus 25



Rust has been Stack Overflow's most loved language for four years in a row and emerged as a compelling language choice for both backend and system developers, offering a unique combination of memory safety, performance, concurrency without Data races...



Clean Architecture provides a clear and modular structure for building software systems, separating business rules from implementation details. It promotes maintainability by allowing for easier updates and changes to specific components without affe...



Azure Service Bus is a crucial component for Azure cloud developers as it provides reliable and scalable messaging capabilities. It enables decoupled communication between different components of a distributed system, promoting flexibility and resili...





I'm Hiring  
Devs

Café

## Developers

Cosmos DB has gained popularity among developers and organizations across various industries, including finance, e-commerce, gaming, IoT, and more. Follow along and learn the 24 most common and advanced Azure Cosmos DB interview questions and answers...

∅ Cosmos DB 46

## 27 Advanced MongoDB Interview Questions (ANSWERED) For Experienced Developers

More than any other NoSQL database, and dramatically more than any relational database, MongoDB's document-oriented data model makes it exceptionally easy to add or change fields, among other things. It unlocks Iteration on the project. Iteration f...

∅ MongoDB 77

## 24 Unit Testing Interview Questions (ANSWERED) For Software Developers

Unit Tests and Test Driven Development (TDD) help you really understand the design of the code you are working on. Instead of writing code to do something, you are starting by outlining all the conditions you are subjecting the code to and what output...

∅ Software Testing 29

∅ Unit Testing 24



I'm Hiring  
Devs

Café

TOP SOFTWARE DEVS AND ARCHITECTS

Domain-Driven Design is nothing magical but it is crucial to understand the importance of Ubiquitous Language, Domain Modeling, Context Mapping, extracting the Bounded Contexts correctly, designing efficient Aggregates and etc. before your next DDD p...

🕒 DDD 38

## 35 Azure Developer Interview Questions (ANSWERED) To Crack Your Cloud Interview

At its core, Microsoft Azure is a public cloud computing platform - with solutions including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) that can be used for services such as analytics, virtual c...

⚠️ Azure 59

## 52 Node.js Interview Questions (ANSWERED) for JavaScript Developers

As an asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications. Follow along to refresh your knowledge and explore the 52 most frequently asked and advanced Node JS Interview Questions and Answers every...

⌚ Node.js 126



I'm Hiring  
Devs

Café

## TOP DEVELOPERS AND SOFTWARE ARCHITECTS

Dependency Injection is most useful when you're aiming for code reuse, versatility and robustness to changes in your problem domain. DI is also useful for decoupling your system. DI also allows easier unit testing without having to hit a database and...

⚡ Dependency Injection 17

❀ Design Patterns 68

▣ Software Architecture 90

Load More...

FullStack.Cafe is a biggest hand-picked collection of top Full-Stack, Coding, Data Structures & System Design Interview Questions to land 6-figure job offer in no time.

Coded with ❤️ using React in Australia AU

by @aershov24, Full Stack Cafe Pty Ltd 🎉, 2018-2023

[Privacy](#) • [Terms of Service](#) • [Guest Posts](#) • [Contacts](#) • [MLStack.Cafe](#)

Great based on 86 Trustpilot Reviews

