Become a Certified Professional →

Top 50 JavaScript Interview Questions You Must Prepare in 2023

Last updated on Nov 29,2022 *613.3K Views*

Share

Sayantini
A Data Science Enthusiast with in-hand skills in programming languages such as...

Today, **Google** and **Facebook** use JavaScript to build complex, desktop-like web applications. With the launch of [Node.js](#), It has also become one of the most popular languages for building server-side software. Today, even the web isn't big enough to contain JavaScript's versatility. I believe that you are already aware of these facts and this has made you land on this JavaScript Interview Questions article.

So, if you are planning to start your career in [JavaScript](#) and you wish to know the skills related to it, now is the right time to dive in, when the technology is in its blossoming state. **JavaScript Interview Questions** and our [JavaScript training](#) and [Java Certification](#) training will provide you with in-depth knowledge and help you prepare for your interviews.

The JavaScript interview questions are divided into three sections:

- **Beginner Level**
- **Intermediate Level**
- **Advanced Level**

Let's begin with the first section of JavaScript interview questions.

JavaScript Interview Questions and Answers in 2023 | Edureka

This Edureka video on "JavaScript Interview Questions" will help you to prepare yourself for JavaScript Interviews.

Beginner Level JavaScript Interview Questions and Answers for Freshers

Q1. What is the difference between Java & JavaScript?

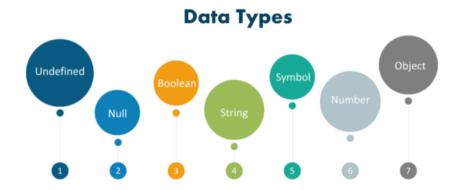| Java	JavaScript
Java is an OOP programming language.	JavaScript is an OOP scripting language.
It creates applications that run in a virtual machine or browser.	The code is run on a browser only.
Java code needs to be compiled.	JavaScript code are all in the form of text.

Q2. What is JavaScript?

- Number
- Object

Data Types

Q4. What are the features of JavaScript?
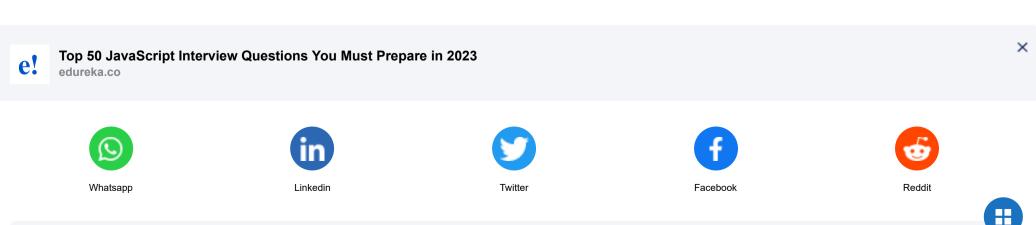
Following are the **features** of JavaScript:

- It is a **lightweight, interpreted** programming language.
- It is designed for creating **network-centric** applications.
- It is complementary to and **integrated** with Java.
- It is an **open** and **cross-platform** scripting language.

Q5. Is JavaScript a case-sensitive language?

Yes, JavaScript is a **case sensitive** language. The language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

Q6. What are the advantages of JavaScript?

Q8. How can you create an Array in JavaScript?

You can define arrays using the **array literal** as follows-

```
var x = [];
var y = [1, 2, 3, 4, 5];
```

Q9. What is a name function in JavaScript & how to define it?

A named function declares a name as soon as it is defined. It can be defined using **function** keyword as :

```
function named(){
// write code here
}
```

Q10. Can you assign an anonymous function to a variable and pass it as an argument to another function?

Yes! An anonymous function can be assigned to a variable. It can also be passed as an argument to another function.

In case you are facing any challenges with these JavaScript Interview Questions, please comment on your problems in the section below.

Q11. What is argument objects in JavaScript & how to get the type of arguments passed to a function?

JavaScript variable arguments represents the **arguments** that are passed to a function. Using **type of** operator, we can get the type of arguments passed to a function. For example –

```
function func(x){
console.log(typeof x, arguments.length);
}
func(); //==> "undefined", 0
func(7); //==> "number", 1
func("1", "2", "3"); //==> "string", 3
```

Q12. What are the scopes of a variable in JavaScript?

The scope of a variable is the **region** of your program in which it is **defined**. JavaScript variable will have only two scopes.

• **Global Variables** – A global variable has global scope which means it is visible everywhere in your JavaScript code.

Closures are created whenever a variable that is defined outside the **current scope** is accessed from within some inner scope. It gives you access to an outer function's scope from an inner function. In JavaScript, closures are created every time a function is created. To use a closure, simply define a function inside another function and expose it.

Q16. Name some of the built-in methods and the values returned by them.

Built-in Method	Values
CharAt()	It returns the character at the specified index.
Concat()	It joins two or more strings.
forEach()	It calls a function for each element in the array.
indexOf()	It returns the index within the calling String object of the first occurrence of the specified value.
length()	It returns the length of the string.
pop()	It removes the last element from an array and returns that element.
push()	It adds one or more elements to the end of an array and returns the new length of the array.
reverse()	It reverses the order of the elements of an array.

In case you are facing any challenges with these JavaScript Interview Questions, please comment on your problems in the section below.

Q17. What are the variable naming conventions in JavaScript?

The following **rules** are to be followed while **naming variables** in JavaScript:

1. You should not use any of the JavaScript **reserved keyword** as variable name. For example, break or boolean variable names are not valid.
2. JavaScript variable names should not start with a **numeral** (0-9). They must begin with a letter or the underscore character. For example, 123name is an invalid variable name but _123name or name123 is a valid one.
3. JavaScript variable names are **case sensitive**. For example, Test and test are two different variables.

Q18. How does Type Of Operator work?

The **type of** operator is used to get the data type of its operand. The operand can be either a **literal** or a **data structure** such as a variable, a function, or an object. It is a **unary** operator that is placed before its single operand, which can be of any type. Its value is a string indicating the data type of the operand.

Q19. How to create a cookie using JavaScript?

The simplest way to create a cookie is to assign a string value to the **document.cookie** object, which looks like this-

Syntax :

Top 10 Technologies to Learn in 2023 | Edureka

▶

Intermediate Level JS Interview Questions and Answers

Q22. What is the difference between Attributes and Property?

Attributes- provide more details on an element like id, type, value etc.

Property- is the value assigned to the property like type="text", value='Name' etc.

Q23. List out the different ways an HTML element can be accessed in a JavaScript code.

Here are the list of ways an HTML element can be accessed in a Javascript code:

(i) **getElementById('idname'):** Gets an element by its ID name

(ii) **getElementsByClass('classname'):** Gets all the elements that have the given classname.

(iii) **getElementsByTagName('tagname'):** Gets all the elements that have the given tag name.

(iv) **querySelector():** This function takes css style selector and returns the first selected element.

Q24. In how many ways a JavaScript code can be involved in an HTML file?

- **Statically:** in this, the variable can hold only one type, like in Java a variable declared of string can take only set of characters and nothing else.

Q27. What is the difference between Local storage & Session storage?

Local Storage – The data is not sent back to the server for every HTTP request (HTML, images, JavaScript, CSS, etc) – reducing the amount of traffic between client and server. It will stay until it is manually cleared through settings or program.

Session Storage – It is similar to local storage; the only difference is while data stored in local storage has no expiration time, data stored in session storage gets cleared when the page session ends. Session Storage will leave when the browser is closed.

In case you are facing any challenges with these JavaScript Interview Questions, please comment on your problems in the section below.

Q28. What is the difference between the operators '==' & '==='?

The main difference between "==" and "===" operator is that formerly compares variable by making **type correction** e.g. if you compare a number with a string with numeric literal, == allows that, but === doesn't allow that, because it not only checks the value but also type of two variable, if two variables are not of the same type "===" return false, while "==" return true.

Q29. What is the difference between null & undefined?

Undefined means a variable has been **declared** but has not yet been **assigned** a value. On the other hand, null is an assignment value. It can be assigned to a variable as a representation of no value. Also, undefined and null are two distinct types: undefined is a type itself (undefined) while null is an object.

Q30. What is the difference between undeclared & undefined?

Undeclared variables are those that do not **exist** in a program and are not declared. If the program tries to read the value of an undeclared variable, then a **runtime error** is encountered. Undefined variables are those that are declared in the program but have not been given any value. If the program tries to read the value of an undefined variable, an undefined value is returned.

Q31. Name some of the JavaScript Frameworks

Become a Certified Professional →

A [JavaScript framework](#) is an application framework written in JavaScript. It differs from a JavaScript library in its control flow. There are many JavaScript Frameworks available but some of the most commonly used frameworks are:

- [Angular](#)
- [React](#)
- Vue

Q32. What is the difference between window & document in JavaScript?

Window	Document
JavaScript window is a global object which holds variables, functions, history, location.	The document also comes under the window and can be considered as the property of the window.

Q33. What is the difference between innerHTML & innerText?

innerHTML – It will process an HTML tag if found in a string

innerText – It will not process an HTML tag if found in a string

Q34. What is an event bubbling in JavaScript?

Event bubbling is a way of **event propagation** in the HTML DOM API, when an event occurs in an element inside another element, and both elements have registered a handle for that event. With bubbling, the event is first captured and handled by the **innermost** element and then propagated to outer elements. The execution starts from that event and goes to its parent element. Then the execution passes to its parent element and so on till the body element.

Q35. What is NaN in JavaScript?

NaN is a short form of **Not a Number.** Since NaN always compares unequal to any number, including NaN, it is usually used to indicate an error condition for a function that should return a valid number. When a string or something else is being **converted** into a **number** and that cannot be done, then we get to see NaN.

In case you are facing any challenges with these JavaScript Interview Questions, please comment on your problems in the section below.

Q39. What are Exports & Imports?

Imports and exports help us to write modular JavaScript code. Using Imports and exports we can split our code into multiple files. For example-

```
//------ lib.js ------</span>
export const sqrt = Math.sqrt;</span>
export function square(x) {</span>
return x * x;</span>
}
export function diag(x, y) {
return sqrt(square(x) + square(y));
}

//------ main.js ------</span>
 { square, diag } from 'lib';
console.log(square(5)); // 25
console.log(diag(4, 3)); // 5
```

Now with this, we have reached the final section of JS Interview Questions.

Advanced Level JavaScript Interview Questions and Answers for Experienced Professionals

Q40. What is the 'Strict' mode in JavaScript and how can it be enabled?

Strict mode is a way to introduce better error-checking into your code.

- When you use strict mode, you cannot use implicitly declared variables, or assign a value to a read-only property, or add a property to an object that is not extensible.
- You can enable strict mode by adding "use strict" at the beginning of a file, a program, or a function.

Q41. What is a prompt box in JavaScript?

A prompt box is a box which allows the user to enter input by providing a **text box**. The prompt() method displays a dialog box that prompts the visitor for input. A prompt box is often used if you want the user to input a value before entering a page. When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

Q42. What will be the output of the code below?

```
var Y = 1;
if (function F(){})
{
y += Typeof F;</span>
```

There are a number of methods you can use to **empty** an **array**:

Method 1 –

```
arrayList = []
```

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 method of emptying the array, because if you have referenced this array from another variable, then the original reference array will remain unchanged.

Method 2 –

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

The code above will clear the existing array by setting its length to 0. This way of emptying the array also updates all the reference variables that point to the original array. Therefore, this method is useful when you want to update all reference variables pointing to arrayList.

Method 3 –

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

The implementation above will also work perfectly. This way of emptying the array will also update all the references to the original array.

Method 4 –

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

The implementation above can also empty arrays, but it is usually not recommended to use this method often.

Q45. What will be the output of the following code?

```
return X.foo;
}
)();
console.log(output);
```

The output would be undefined. The delete operator is used to delete the property of an object. Here, x is an object which has the property foo, and as it is a self-invoking function, we will delete the foo property from object x. After doing so, when we try to reference a deleted property foo, the result is undefined.

Q47. What will be the output of the following code?

```
var Employee =
{
company: 'xyz'
}
var Emp1 = Object.create(employee);
delete Emp1.company Console.log(emp1.company);
```

The output would be xyz. Here, emp1 object has company as its prototype property. The delete operator doesn't delete prototype property. emp1 object doesn't have company as its own property. However, we can delete the company property directly from the Employee object using delete Employee.company.

Q48. What will be the output of the code below?

```
//nfe (named function expression)
var Foo = Function Bar()
{
return 7;
};
typeof Bar();
```

The output would be Reference Error. A function definition can have only one reference variable as its function name.

Q49. What is the reason for wrapping the entire content of a JavaScript source file in a function book?

This is an increasingly common practice, employed by many popular JavaScript libraries. This technique creates a closure around the entire contents of the file which, perhaps most importantly, creates a private namespace and thereby helps avoid potential name clashes between different JavaScript modules and libraries.

Become a Certified Professional →

Subscribe to our Newsletter, and get personalized recommendations.

G   Sign up with Google

f   Signup with Facebook

Already have an account? Sign in.

and front-end web technologies. It includes training on Web Development, jQuery, Angular, NodeJS, ExpressJS and MongoDB.

Got a question for us? Please mention it in the comments section of "JavaScript Interview Questions" blog and we will get back to you or you can also join our Java Training in Bangalore.

Top 50 JavaScript Interview Questions You Must Prepare in 2023
edureka.co

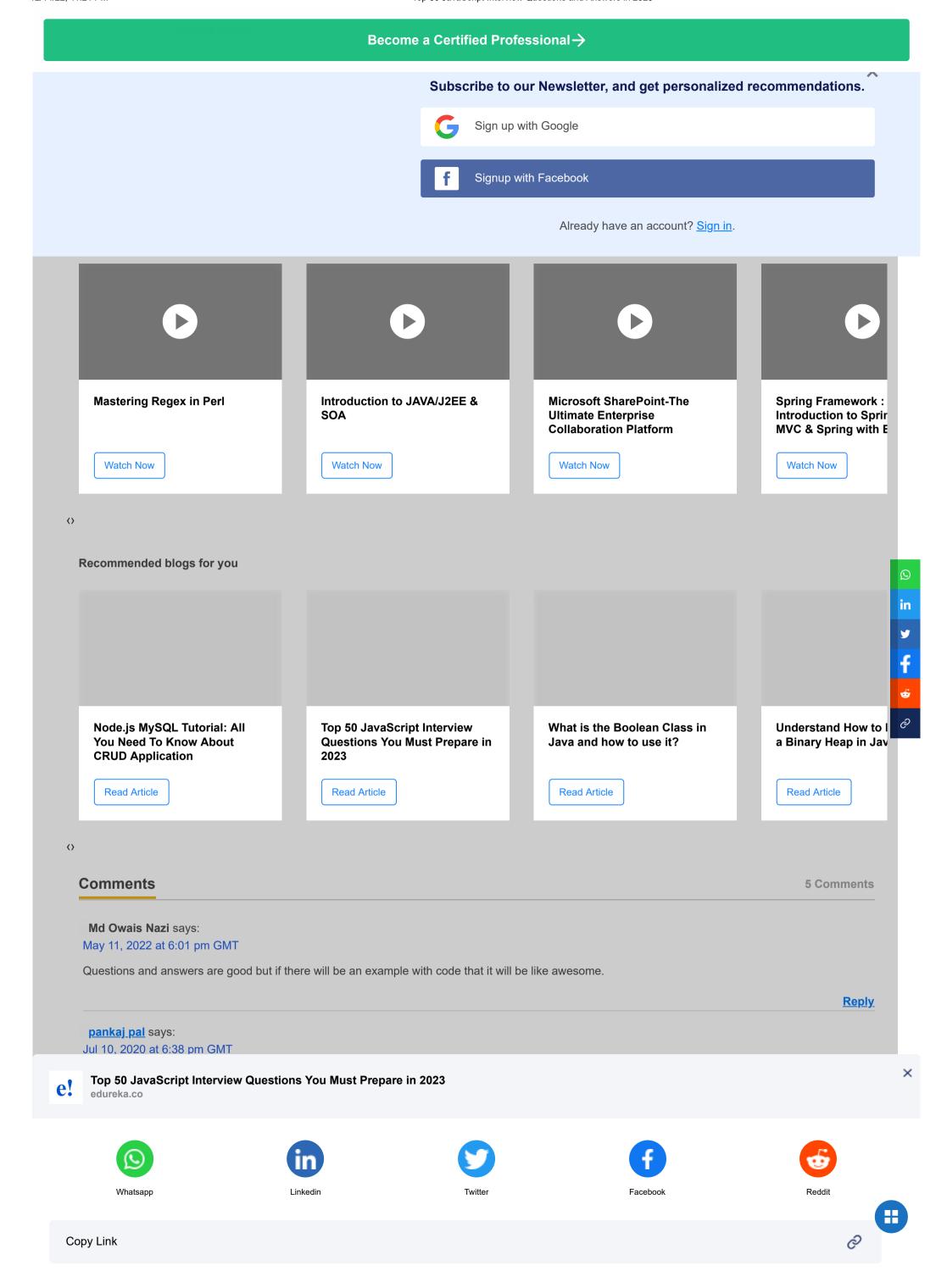| Whatsapp | Linkedin | Twitter | Facebook | Reddit |

Copy Link

Become a Certified Professional →

Subscribe to our Newsletter, and get personalized recommendations.

G   Sign up with Google

f   Signup with Facebook

Already have an account? Sign in.

Mastering Regex in Perl

Watch Now

Introduction to JAVA/J2EE & SOA

Watch Now

Microsoft SharePoint-The Ultimate Enterprise Collaboration Platform

Watch Now

Spring Framework : Introduction to Spring MVC & Spring with E

Watch Now

‹›

Recommended blogs for you

Node.js MySQL Tutorial: All You Need To Know About CRUD Application

Read Article

Top 50 JavaScript Interview Questions You Must Prepare in 2023

Read Article

What is the Boolean Class in Java and how to use it?

Read Article

Understand How to a Binary Heap in Jav

Read Article

‹›

Comments   5 Comments

Md Owais Nazi says:

May 11, 2022 at 6:01 pm GMT

Questions and answers are good but if there will be an example with code that it will be like awesome.

Reply

pankaj pal says:

Jul 10, 2020 at 6:38 pm GMT

e!   **Top 50 JavaScript Interview Questions You Must Prepare in 2023**
      edureka.co   ✕

Whatsapp   Linkedin   Twitter   Facebook   Reddit

Copy Link   🔗

Become a Certified Professional →

not defined Output != output And after all these fixes we got 0 in console.Why do you care so little about code?Q46 Output != output X.Foo != X.fooQ47 Exception: SyntaxError: unexpected token: identifier Because there is missing ; Should be: delete Emp1.company; Exception: ReferenceError: employee is not defined Employee != employeeEmp1 != emp1Exception: ReferenceError: Console is not definedAnd after all these fixes we get xyzQ48 Exception: SyntaxError: unexpected token: identifier is it Function or function? Who knows… Fix that and no error. No output also. Needs console.log(typeof Bar()); And then output number… Why? Because this works. Function can have one name, but many aliases…Dude, it's 2019! Why are you teaching people like this???

Reply

Join the discussion

Trending Courses in Programming & Frameworks

Full Stack Web Development Internship Program	Java Certification Training Course	Python Scripting Certification Training	Node.js Certification Course
3k Enrolled Learners	65k Enrolled Learners	13k Enrolled Learners	9k Enrolled Learners
Weekend/Weekday	Weekend	Weekend	Weekend
Live Class	Live Class	Self Paced	Live Class
Reviews	Reviews	Reviews	Reviews
★★★★★ 5 (850)	★★★★☆ 4 (25850)	★★★★★ 5 (5050)	★★★★★ 5 (3300

Whatsapp

Linkedin

Twitter

Facebook

Reddit

Top 50 JavaScript Interview Questions You Must Prepare in 2023

Copy Link

Become a Certified Professional →

Subscribe to our Newsletter, and get personalized recommendations.

G   Sign up with Google

f   Signup with Facebook

Already have an account? Sign in.

TRENDING CERTIFICATION COURSES

DevOps Certification Training

AWS Architect Certification Training

Big Data Hadoop Certification Training

Tableau Training & Certification

Python Certification Training for Data Science

Selenium Certification Training

PMP® Certification Exam Training

Robotic Process Automation Training using UiPath

Apache Spark and Scala Certification Training

Microsoft Power BI Training

Online Java Course and Training

Python Certification Course

TRENDING MASTERS COURSES

Data Scientist Masters Program

DevOps Engineer Masters Program

Cloud Architect Masters Program

Big Data Architect Masters Program

Machine Learning Engineer Masters Program

Full Stack Web Developer Masters Program

Business Intelligence Masters Program

Data Analyst Masters Program

Test Automation Engineer Masters Program

Post-Graduate Program in Artificial Intelligence & Machine Learning

Post-Graduate Program in Big Data Engineering

COMPANY

About us

News & Media

Reviews

Contact us

Blog

Community

Sitemap

Blog Sitemap

Community Sitemap

Webinars

WORK WITH US

Careers

Become an Instructor

Become an Affiliate

Become a Partner

Hire from Edureka

DOWNLOAD APP

CATEGORIES   ⌄

CATEGORIES

Cloud Computing | DevOps | Big Data | Data Science | BI and Visualization | Programming & Frameworks | Software Testing |

Project Management and Methodologies | Robotic Process Automation | Frontend Development | Data Warehousing and ETL | Artificial Intelligence | Blockchain

Databases | Cyber Security | Mobile Development | Operating Systems | Architecture & Design Patterns | Digital Marketing

TRENDING BLOG ARTICLES   ⌄

Whatsapp

Linkedin

Twitter

Facebook

Reddit

Copy Link

Become a Certified Professional →

Subscribe to our Newsletter, and get personalized recommendations.

G   Sign up with Google

f   Signup with Facebook

Already have an account? Sign in.