JavaScript Interview Questions

1. **What is JavaScript?**
2. **Enumerate the differences between Java and JavaScript?**
3. **What are JavaScript Data Types?**
4. **What is the use of isNaN function?**
5. **Which is faster between JavaScript and an ASP script?**
6. **Which company developed JavaScript?**
7. **What are undeclared and undefined variables?**
8. **What are global variables? How are these variable declared?**
9. **What is a prompt box?**
10. **What is 'this' keyword in JavaScript?**
11. **What is the working of timers in JavaScript?**
12. **Which symbol is used for comments in Javascript?**
13. **What is the difference between ViewState and SessionState?**
14. **What is === operator? Or what is strict equality operator?**
15. **Does JavaScript support automatic type conversion?**
16. **How to read and write a file using JavaScript?**
17. **What is called Variable typing in Javascript?**
18. **Difference between "==" and "==="?**
19. **What would be the result of 3+2+"7"?**
20. **What do you mean by NULL in Javascript?**
21. **What is an undefined value in JavaScript?**
22. **What are all the types of Pop up boxes available in JavaScript?**
23. **What is the use of Void (0)?**
24. **What is the data type of variables in JavaScript?**
25. **What is the difference between an alert box and a confirmation box?**
26. **What are escape characters?**
27. **What are JavaScript Cookies?**
28. **What a pop()method in JavaScript is? And shift() method?**
29. **What are the disadvantages of using innerHTML in JavaScript?**
30. **What is break and continue statements?**
31. **What are the two basic groups of data types in JavaScript?**
32. **Which keywords are used to handle exceptions?**
33. **What is the use of the blur function?**
34. **What are the different types of errors in JavaScript?**
35. **What is the use of the Push method in JavaScript? Unshift() method?**
36. **What is the 'Strict Mode in JavaScript, and how can it be enabled?**
37. **How can the OS of the client machine be detected?**
38. **How closures work in JavaScript?**
39. **What are the important properties of an anonymous function in JavaScript?**
40. **What is the difference between .call() and .apply()?**
41. **What is event bubbling?**
42. **Is JavaScript case sensitive? Give its example.**
43. **How are DOM utilized in JavaScript?**
44. **How are event handlers utilized in JavaScript?**
45. **What are the decodeURI() and encodeURI()?**
46. **What are the important JavaScript Array Method explain with example?**
47. **What is OOPS Concept in JavaScript?**
48. **What is JavaScript Unit Testing, and what are the challenges in JavaScript Unit Testing? What are some important JavaScript Unit Testing Frameworks?**

Link: <https://www.guru99.com/javascript-interview-questions-answers.html>

Other’s Link: <https://github.com/sudheerj/javascript-interview-questions>

<https://www.interviewbit.com/javascript-interview-questions/>

<https://vigowebs.medium.com/frequently-asked-es6-interview-questions-and-answers-e3fb7f2dba2>

<https://www.interviewbit.com/javascript-interview-questions/>

<https://dev.to/macmacky/70-javascript-interview-questions-5gfi>

<http://thatjsdude.com/interview/js2.html>

Applying Job’s in India

1. <https://internshala.com/internship/detail/web-development-work-from-home-job-internship-at-avyukt-edutech-pvt-ltd1622315302>
2. <https://internshala.com/internship/detail/full-stack-development-work-from-home-job-internship-at-peakmind1622303644>
3. <https://internshala.com/internship/detail/reactjs-development-work-from-home-job-internship-at-applore-technologies1622281427>
4. <https://internshala.com/internship/detail/full-stack-web-app-development-work-from-home-job-internship-at-shashank-b1622255011>
5. <https://internshala.com/internship/detail/frontend-development-work-from-home-job-internship-at-tgh-technologies-private-limited1622272538>

Questions With Answer

1. **What is the difference between Call, Apply and Bind**

Ans: **Call:** The call() method invokes a function with a given this value and arguments provided one by one

**Apply:** Invokes the function with a given this value and allows you to pass in arguments as an array

**bind:** returns a new function, allowing you to pass any number of arguments

1. What is JSON and its common operations?

Ans: **JSON** is a text-based data format following JavaScript object syntax,

JSON (JavaScript Object Notation) is a lightweight format that is used for data interchanging.

It is useful when you want to transmit data across a network and it is basically just a text file with an extension of .json.

**Parsing:** Converting a string to a native object

**Stringification:** converting a native object to a string so it can be transmitted across the network,

1. Difference between slice and splice?

Ans: Slice method won't mutate the original array but it returns the subset as a new array. Used to pick the elements from array.

Splice method modifies the original array and returns the deleted array. Used to insert or delete elements to/from array.

### What is the difference between == and === operators?

### Ans: (===) means strict operator and (==) means non-strict/ type converting operator.

0 == false // true

0 === false // false

1 == "1" // true

1 === "1" // false

null == undefined // true

null === undefined // false

'0' == false // true

'0' === false // false

[]==[] or []===[] //false, refer different objects in memory

{}=={} or {}==={} //false, refer different objects in memory

### What are lambda or arrow functions

Ans: An arrow function is a shorter syntax for a function expression and does not have its own **this, arguments, super, or new.target**. These functions are best suited for non-method functions, and they cannot be used as constructors

### What is a first order function

Ans: First-order function is a function that doesn’t accept another function as an argument and doesn’t return a function as its return value.

### What is a higher order function

Ans: Higher-order function is a function that accepts another function as an argument or returns a function as a return value or both.

### What is Hoisting

Ans: Hoisting is a JavaScript mechanism where variables and function declarations are moved to the top of their scope before code execution. Remember that JavaScript only hoists declarations, not initialization.

var message;

console.log(message);

message = 'The variable Has been hoisted';

### What are closures?

### Ans: The closure is a locally declared variable related to a function that stays in memory when it has returned.

Ans: A closure is the combination of a function and the lexical environment within which that function was declared. i.e, It is an inner function that has access to the outer or enclosing function’s variables. The closure has three scope chains

1. Own scope where variables defined between its curly brackets
2. Outer function’s variables
3. Global variables

Let's take an example of closure concept,

function Welcome(name){

var greetingInfo = function(message){

console.log(message+' '+name);

}

return greetingInfo;

}

var myFunction = Welcome('John');

myFunction('Welcome '); //Output: Welcome John

myFunction('Hello Mr.'); //output: Hello Mr.John

### What is scope in javascript

Ans: Scope determines the visibility of variables and other resources in areas of your code.

### What is web storage

Ans: Web storage is an API that provides a mechanism by which browsers can store key/value pairs locally within the user's browser

The web storage provides two mechanisms for storing data on the client.

1. **Local storage:** It stores data for current origin with no expiration date.
2. **Session storage:** It stores data for one session and the data is lost when the browser tab is closed.

### What is a Cookie

Ans: A cookie is a piece of data that is stored on your computer to be accessed by your browser. Cookies are saved as key/value pairs. For example, you can create a cookie named username as below,

document.cookie = "username=John";

1. **Why do you need a Cookie**

Cookies are used to remember information about the user profile(such as username). It basically involves two steps,

1. When a user visits a web page, the user profile can be stored in a cookie.
2. Next time the user visits the page, the cookie remembers the user profile.

### What are the differences between cookie, local storage and session storage

Below are some of the differences between cookie, local storage and session storage,

| **Feature** | **Cookie** | **Local storage** | **Session storage** |
| --- | --- | --- | --- |
| Accessed on client or server side | Both server-side & client-side | client-side only | client-side only |
| Lifetime | As configured using Expires option | until deleted | until tab is closed |
| SSL support | Supported | Not supported | Not supported |
| Maximum data size | 4KB | 5 MB | 5MB |

### Why do you need a promise

Promises are used to handle asynchronous operations. It has three states such as

Pending, fulfilled, rejected.

### What is a callback function

A callback function is a function passed into another function as an argument. This function is invoked inside the outer function to complete an action

function callbackFunction(name) {

console.log('Hello ' + name);

}

function outerFunction(callback) {

let name = prompt('Please enter your name.');

callback(name);

}

outerFunction(callbackFunction);

### Why do we need callbacks

The callbacks are needed because javascript is an event driven language. That means instead of waiting for a response javascript will keep executing while listening for other events

### What is the typeof operator

You can use the JavaScript typeof operator to find the type of a JavaScript variable. It returns the type of a variable or an expression.

### What is the difference between null and undefined

Below are the main differences between null and undefined,

| **Null** | **Undefined** |
| --- | --- |
| It is an assignment value which indicates that variable points to no object. | It is not an assignment value where a variable has been declared but has not yet been assigned a value. |
| Type of null is object | Type of undefined is undefined |
| The null value is a primitive value that represents the null, empty, or non-existent reference. | The undefined value is a primitive value used when a variable has not been assigned a value. |
| Indicates the absence of a value for a variable | Indicates absence of variable itself |
| Converted to zero (0) while performing primitive operations | Converted to NaN while performing primitive operations |

### What are the differences between undeclared and undefined variables

Below are the major differences between undeclared and undefined variables,

| **undeclared** | **undefined** |
| --- | --- |
| These variables do not exist in a program and are not declared | These variables declared in the program but have not assigned any value |
| If you try to read the value of an undeclared variable, then a runtime error is encountered | If you try to read the value of an undefined variable, an undefined value is returned. |

### What are global variables

Global variables are those that are available throughout the length of the code without any scope. The var keyword is used to declare a local variable but if you omit it then it will become global variable

msg = "Hello" // var is missing, it becomes global variable

### What are the problems with global variables

The problem with global variables is the conflict of variable names of local and global scope. It is also difficult to debug and test the code that relies on global variables.

### What is an event flow

Event flow is the order in which event is received on the web page. There are two ways of event flow

1. Top to Bottom(Event Capturing)
2. Bottom to Top (Event Bubbling)

### What is event bubbling

Event bubbling is a type of event propagation where the event first triggers on the innermost target element, and then successively triggers on the ancestors (parents) of the target element in the same nesting hierarchy till it reaches the outermost DOM element.

JavaScript allows DOM elements to be nested inside each other. In such a case, if the handler of the child is clicked, the handler of the parent will also work as if it were clicked too.

### What is event capturing

Event capturing is a type of event propagation where the event is first captured by the outermost element, and then successively triggers on the descendants (children) of the target element in the same nesting hierarchy till it reaches the innermost DOM element.

### What is the purpose of void 0

Void(0) is used to prevent the page from refreshing

### Is JavaScript a compiled or interpreted language

JavaScript is an interpreted language, not a compiled language. An interpreter in the browser reads over the JavaScript code, interprets each line, and runs it. Nowadays modern browsers use a technology known as Just-In-Time (JIT) compilation, which compiles JavaScript to executable bytecode just as it is about to run.

1. **What are events**

Events are "things" that happen to HTML elements. When JavaScript is used in HTML pages, JavaScript can react on these events. Some of the examples of HTML events are,

1. Web page has finished loading
2. Input field was changed
3. Button was clicked

### What is the use of stopPropagation method

The stopPropagation method is used to stop the event from bubbling up the event chain.

### What is the use of preventDefault method

The preventDefault() method cancels the event if it is cancelable, meaning that the default action or behaviour that belongs to the event will not occur.

### What is the use of setTimeout

The setTimeout() method is used to call a function or evaluate an expression after a specified number of milliseconds. For example, let's log a message after 2 seconds using setTimeout method,

setTimeout(function(){ console.log("Good morning"); }, 2000);

### What is the use of setInterval

The setInterval() method is used to call a function or evaluate an expression at specified intervals (in milliseconds). For example, let's log a message after 2 seconds using setInterval method,

setInterval(function(){ console.log("Good morning"); }, 2000);

### What are break and continue statements

The break statement is used to "jump out" of a loop.

The continue statement is used to "jump over" one iteration in the loop

### What is a rest parameter

Rest parameter allows us to represent an indefinite number of arguments as an array.

### What is a spread operator

Spread operator allows iterables( arrays / objects / strings ) to be expanded into single arguments/elements

### What is an anonymous function

An anonymous function is a function without a name! Anonymous functions are commonly assigned to a variable name or used as a callback function.

var x = function (a, b) {return a \* b};

var z = x(5, 10);

console.log(z); // 50

1. **What are primitive data types**

A primitive data type is data that has a primitive value (which has no properties or methods). There are 7 types of primitive data types.

1. string
2. number
3. boolean
4. null
5. undefined
6. bigint
7. symbol

### What is an event loop

The Event Loop is a queue of callback functions. When an async function executes, the callback function is pushed into the queue. The JavaScript engine doesn't start processing the event loop until the async function has finished executing the code. **Note:** It allows Node.js to perform non-blocking I/O operations even though JavaScript is single-threaded.

1. **What is call stack**

Call Stack is a data structure for javascript interpreters to keep track of function calls in the program. It has two major actions,

1. Whenever you call a function for its execution, you are pushing it to the stack.
2. Whenever the execution is completed, the function is popped out of the stack.
3. **List down some of the features of ES6**

Below are the list of some new features of ES6,

1. Support for constants or immutable variables
2. Block-scope support for variables, constants and functions
3. Arrow functions
4. Default parameters
5. Rest and Spread Parameters
6. Template Literals
7. Multi-line Strings
8. Destructuring Assignment
9. Enhanced Object Literals
10. Promises
11. Classes
12. Modules

### What is ES6

ES6 is the sixth edition of the javascript language and it was released in June 2015. It was initially known as ECMAScript 6 (ES6) and later renamed to ECMAScript 2015. Almost all the modern browsers support ES6 but for the old browsers there are many transpilers, like Babel.js etc.

### What is destructuring assignment

The destructuring assignment is a JavaScript expression that makes it possible to unpack values from arrays or properties from objects into distinct variables.

### What is AJAX

AJAX stands for Asynchronous JavaScript and XML and it is a group of related technologies(HTML, CSS, JavaScript, XMLHttpRequest API etc) used to display data asynchronously. i.e. We can send data to the server and get data from the server without reloading the web page.

1. **Enumerate the differences between Java and JavaScript?**

Java is a complete programming language. In contrast, JavaScript is a coded program that can be introduced to HTML pages. Java is an object-oriented programming (OOPS) or structured programming languages like C++ or C, whereas JavaScript is a client-side scripting language.

1. **What is the use of isNaN function?**

isNan function returns true if the argument is not a number; otherwise, it is false.

1. **Which company developed JavaScript?**

Netscape is the software company that developed JavaScript.

1. **What are undeclared and undefined variables?**

Undeclared variables are those that do not exist in a program and are not declared.

Undefined variables are those that are declared in the program but have not been given any value.

1. **What are global variables? How are these variable declared?**

Global variables are available throughout the length of the code so that it has no scope. The var keyword is used to declare a local variable or object. If the var keyword is omitted, a global variable is declared.

1. **What is a prompt box?**

A prompt box is a box that allows the user to enter input by providing a text box.

1. **What is 'this' keyword in JavaScript?**

'This' keyword refers to the object from where it was called.

1. **What is the working of timers in JavaScript?**

The **setTimeout(function, delay)** function is used to start a timer that calls a particular function after the mentioned delay. The **setInterval(function, delay)** function repeatedly executes the given function in the mentioned delay and only halts when canceled. The **clearInterval(id)** function instructs the timer to stop.

1. **What is === operator?**

=== is called a strict equality operator, which returns true when the two operands have the same value without conversion.

1. **What do you mean by NULL in Javascript?**

The NULL value is used to represent no value or no object. It implies no object or null string, no valid boolean value, no number, and no array object.

1. **What is an undefined value in JavaScript?**

Undefined value means the

* Variable used in the code doesn't exist
* Variable is not assigned to any value
* Property does not exist.

1. **What are all the types of Pop up boxes available in JavaScript?**

* Alert
* Confirm and
* Prompt

1. **What is the difference between an alert box and a confirmation box?**

An alert box displays only one button, which is the OK button.

But a Confirmation box displays two buttons, namely OK and cancel.

1. **What are escape characters?**

Escape characters (Backslash) is used when working with special characters like single quotes, double quotes, apostrophes, and ampersands.

1. **What are the different types of errors in JavaScript?**

There are three types of errors:

* **Load time errors**: Errors that come up when loading a web page, like improper syntax errors, are known as Load time errors and generate the errors dynamically.
* **Runtime errors**: Errors that come due to misuse of the command inside the HTML language.
* **Logical Errors**: These are the errors that occur due to the bad logic performed on a function with a different operation.

1. **How can the OS of the client machine be detected?**

The navigator. appVersion string can be used to detect the operating system on the client machine.

1. **What is JavaScript Unit Testing, and what are the challenges in JavaScript Unit Testing?**

JavaScript Unit Testing is a testing method in which JavaScript tests code written for a web page or web application module. It is combined with HTML as an inline event handler and executed in the browser to test if all functionalities work fine. These unit tests are then organized in the test suite.

Every suite contains several tests designed to be executed for a separate module. Most importantly, they don't conflict with any other module and run with fewer dependencies on each other (some critical situations may cause dependencies).

**Challenges of JavaScript Unit Testing:**

Here are important challenges of JavaScript Unit Testing:

* Many other languages support unit testing in browsers, in the stable as well as in runtime environment, but JavaScript can not
* You can understand some system actions with other languages, but this is not the case with JavaScript
* Some JavaScript are written for a web application that may have multiple dependencies.
* JavaScript is good to use in combination with HTML and CSS rather than on the web
* Difficulties with page rendering and DOM manipulation
* Sometimes you find an error message on your screen regarding 'Unable to load example.js' or any other JavaScript error regarding version control. These vulnerabilities come under Unit Testing JavaScript

**Solutions of JavaScript Unit Testing:**

To avoid such issues, what you can do is;

* Do not use global variables.
* Do not manipulate predefined objects.
* Design core functionalities based on the library.
* Try to create small pieces of functionalities with lesser dependencies

1. **What is DOM in JavaScript?**

JavaScript can access all the elements in a web page using the Document Object Model (DOM). The web browser creates a DOM of the webpage when the page is loaded.