Custom Search SEARCH

Software Testing Help

32 TOP JavaScript Interview Questions and Answers

Last Updated: October 23, 2018

JavaScript Interview Questions and Answers:

If you are preparing for an interview, here are the **32** most frequently asked JavaScript Interview Questions and Answers for your reference.

We have designed the same to get you introduced with the questions you may likely encounter during your technical interview.

Let's Explore!!

</>
</>
</>
JavaScript Interview Questions and Answers
© www.SoftwareTestingHelp.com

Menu

About JavaScript

JavaScript is a high-level programming language, probably one of the most used programming languages in the world right now. It can be used to program the web browser or even servers.

To know the importance of JavaScript, disable JavaScript on your browser and try to load the Web Page in it. Those Web Pages will not work properly. Many contents in them may misbehave.

Almost all modern browsers use the combination of JavaScript, CSS, and HTML.

JavaScript is an Interpreted Programming Language. An interpreter is embedded on Browsers like Google Chrome, Microsoft Internet Explorer etc. So, JavaScript code can be handled by JavaScript Engine of the Browser

JavaScript appeared in December 1995 and was initially called LiveScript, although the name was soon changed for marketing reasons.

JavaScript should not be confused with 'Java' which is also bearing some resemblance but is a completely different language.

List of Most Frequently Asked JavaScript Interview Questions

Q #1) What is JavaScript?

Answer: JavaScript is a scripting language developed by Netscape. It can be used to program web browser or even servers. It can dynamically update the contents of the webpage, which is the beauty of JavaScript.

Q #2) What are the advantages of using External JavaScript?

Answer: Using External JavaScript in our code has many advantages as stated below.

- Separation of Code is done.
- Code Maintainability is Easy.
- Performance is better.

Q #3) In the following Code snippet can you please predict the output or If you get an error, please explain the error?

```
elp</strong> </h2>

<script>
var studentName = "Sajeesh Sreeni"; // String 'Sajeesh Sreeni' stored in studentName
var studentName; // varaible is decalred again
document.getElementById("studentName").innerHTML =
"Redeclaring the varaible will not lose the value!.<br>"
+"Here the value in studentName is "+ studentName;

</script>

</body>
</html>
```

Answer: This code will not produce any errors.

Redeclaration of the variables is allowed in JavaScript. Hence, the value of the variable will not be lost after the execution of the statement here.

Q #4) In the following Code snippet can you please predict the output or If you get an error; please explain the error?

```
<!DOCTYPE html>
<html>
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p id="sum_first"></p>
<p id="sum_second"></p>
<script>
var sum_first =50+20+' Sajeesh Sreeni ';
var sum_second= " Sajeesh Sreeni "+50+20;
document.getElementById("sum_first").innerHTML = "The first varaible sum is :"+sum_first +
"<br>The second varaible sum is :"+sum_second ;
</script>
</body>
</html>
```

Answer: This code will not show any error!

The output of the code snippet here is:

The first variable sum is: 70 Sajeesh Sreeni
The second variable sum is: Sajeesh Sreeni 5020

Q #5) What is the difference between test () and exec () methods?

☰ Menu

en pattern, if it finds the matching text then it returns
alse'.

But in exec (), we will search a string for a given pattern, if it finds the matching text then it returns the pattern itself and else it returns 'null' value.

Q #6) What are the Advantages of JavaScript?

Answer: JavaScript Scripting language has many advantages as stated below.

- **Lightweight:** JavaScript is easy to implement. It has small memory footprints.
- **Interpreted:** It is an interpreted language. Instructions are executed directly.
- **Object-oriented:** JavaScript is an object-oriented language.
- **First class functions:** In JavaScript, a function can be used as a value.
- **Scripting Language:** It's a language in which instructions are written for a run-time environment.

Q #7) In the following Code snippet can you please predict the output or If you get an error; please explain the error?

```
<!DOCTYPE html>
<html>
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p style='text-decoration:underline'>Example Const Variable </p>
<p id="display"></p>
<script>
const first_num;
first_num =1000;
document.getElementById("display").innerHTML = "First Number:"+ first_num;
</script>
</body>
</html>
```

Answer: The 'const' variable 'first_num' is not initialized with a value, so the code will produce a Syntax Error.

The output of the code snippet here is:

Error: Uncaught SyntaxError: Missing initializer in the const declaration

Q #8) Have you used any browser for debugging? If yes, how is it done?

Answer: By, Pressing 'F12' key in the keyboard we can enable debugging in the browser. Chose the 'Console' tab to view the result.

Menu

the value in variables. All the modern browsers have a **ırome, Firefox, Opera, and Safari)**. This feature can be turned ON and OFF.

Q #9) What is the use of 'debugger' keyword in JavaScript code?

Answer: Using the 'debugger' keyword in the code is like using breakpoints in the debugger.

To test the code, the debugger must be enabled for the browser. If debugging is disabled for the browser, the code will not work. During debugging the code below should stop executing, before it goes to the next line.

Q #10) What are the distinct types of Error Name Values?

Answer: There are 6 types of values in 'Error Name' Property.

Error	Description
Range Error	We will get this error if we use a number outside the range
Syntax Error	This error raises when we use the incorrect syntax. (Please refer Ques No: 7)
Reference Error	This error is thrown if used an undeclared variable Please refer Ques No: 19
Eval Error	Thrown due to the error in eval(). New JavaScript version doesn't have this error
Type Error	Value is outside the range of types used. Please refer Ques No :22
URI Error	Due to the usage of illegal characters.

Q #11) What is JavaScript Hoisting?

Answer: Using 'JavaScript Hoisting' method, when an interpreter runs the code, all the variables are hoisted to the top of the original /current scope. If you have a variable declared anywher

≡ Menu

ation of a variable and is not applicable for hoisted to the top, whereas function explanations are not hoisted to the top.

Basically, where we declared the variable inside the code doesn't matter much.

Q #12) What is JavaScript 'Strict Mode'?

Answer: 'Strict mode' is a restricted variant of JavaScript.

Usually, JavaScript is 'not very strict' in throwing errors.

But in 'Strict mode' it will throw all types of errors, even the silent errors. Thus, the process of debugging becomes easier. And the chances for making mistake for the developer is reduced.

Q #13) What are the characteristics of JavaScript 'Strict Mode'?

Answer: Given below are the characteristics of JavaScript 'Strict Mode':

- 'Strict Mode' will stop developers from creating global variables.
- Developers are restricted from using duplicate parameters.
- Strict mode will restrict you from using JavaScript keyword as a variable name or function name.
- Strict mode is declared with 'use strict' keyword at the beginning of the script.
- All browser support strict mode.

Q #14) What are Self Invoking Functions?

Answer: They are also known as 'Immediately Invoked Function Expressions' or 'Self Executing Anonymous Functions'. These functions are invoked automatically in the code, hence they are named as 'Self Invoking Functions'.

Usually, we define a function and invoke it, but if we want to execute a function automatically where it is explained, and if we are not going to call it again, we can use anonymous functions. And these types of function have no name.

Q #15) What is the syntax of 'Self Invoking Function'? Give an example?

Answer:

The syntax for the Self-Invoking function:

```
(function () {
return () } () ;
```

☰ Menu

```
<!DOCTYPE html>
<html>
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p style='text-decoration:underline'>Example for Self-Invoking </p>
<p id="dispaly_num"></p>
<script>
(function (){
elem = document.getElementById("dispaly_num");
elem.innerHTML = "This function has no name.<br>It is called automatically";
}());
</script>
</body>
</html>
```

Here the anonymous function is automatically invoked in the code snippet.

The function is used to set the text property of <p> tag having ‘display_num’ as Id.

The output of the code snippet here is:

> *This function has no name.*
> *It is called automatically*

Q #16) In the following Code snippet, can you please predict the output or If you get an error; please explain the error?

Answer:

```
<!DOCTYPE html>
<html>
<body>
<h2> <strong> Sample : Software Testing Help</strong> </h2>
<p style='text-decoration:underline'>Example for JavaScript Hoisting </p>
<p id="dispaly_num"></p>
<script>
first_num = 100; // Assign value 100 to num
elem = document.getElementById("dispaly_num");
elem.innerHTML = " Here the variable first_num:<u> "+first_num +"</u> is taken to the top <br>"
"Since second variable is initialised the value is not taken to the top and it's value is "
+ "<u>"+second_num +"</u> “;
var first_num; // declaration only
var second_num =200; // Initialised the variable
</script>
</body>
</html>
```

Menu

d there, the interpreter will take all the variables

As per this, ‘first_num’ variable is taken to the top and ‘second_num’ variable is initialized with a value, so it is not taken to the top. The is code will not throw an error. But the value of ‘second_num’ is undefined.

The output of the code snippet here is:

Here the variable first_num: 100 is taken to the top
Since the second variable is initialized the value is not taken to the top and its value is undefined

Q #17) If you need to hide the JavaScript code from the older browser versions, how will you perform it?

Answer: In JavaScript Code, after the <script> tag add ‘<! –’ HTML tag.

This will make the browser to not execute the JavaScript code if it was being an older version of it. Also, after the end </script> tag add ‘//–>’ HTML tag.

This method will help in solving compatibility issues and UI issues to an extent.

```
<!DOCTYPE html>
<html>
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p id="display"></p>
<script> <!--
document.getElementById("display").innerHTML = "Here I am not using an older version of browse
So the code will work in my browser";
//-->
</script>
</body>
</html>
```

Here the code snippet after a <script> tag is executed in my browser as I am not using an older version of the browser.

The output of the code snippet here is:

Here I am not using an older version of the browser.
So the code will work in my browser

Q #18) In the following Code snippet can you please predict the output or If you get an error,

```
<!DOCTYPE html>
<html>
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p style='text-decoration:underline'>Find the output </p>
<p id="display"></p>
<script>
var first_num =500;
var result= function(){
document.getElementById("display").innerHTML = first_num;
var first_num =1000;
}
result();
</script>
</body>
</html>
```

Answer: Here in the code given above, the value of the 'first_num' variable will not be 1000.

In JavaScript, there is no hoisting for variable initialization. The function 'result ()' will choose the local variable 'first_num', as it is declared inside the function. Since the variable is declared after it is used, the value of 'first_num' is undefined.

The output of the code snippet here is:

Undefined

Q #19) What is the difference between 'var' and 'let' keyword?

Answer: The Differences are as follows:

Var	let
'var' keyword was introduced in JavaScript code from the beginning Stage itself.	'let' keyword is introduced in 2015 only.
'Var' keyword has function scope. The variable defined with var is available anywhere within the function	A variable declared with 'let' keyword has a scope only with in that block. So, let has a Block Scope.
The variable declared with 'var' be hoisted	The variable declared with 'let' be hoisted

Q #20) In the following Code snippet can you please predict the output or If you get an error; please explain the error?

```
<html>
```

```
<html>

                                                  elp</strong> </h2>
<p style="text-decoration:underline">Find the output </p>
<p id="display_first"></p>
<p id="display_second"></p>
<script>
if(true){
var first_num =1000;
let second_num=500;
}
document.getElementById("display_first").innerHTML = "First Number:" + first_num;
document.getElementById("display_second").innerHTML = "Second Number:" + second_num;
</script>
</body>
</html>
```

Answer:

The output of the code snippet here is:

First Number :*1000*

We will get *'First* Number :*1000'* as output. There is an 'Uncaught Reference Error' error also. In the code snippet, the scope of 'second_num' is only with in the if() block. If a developer tries to access the value outside the block, he will get a 'Uncaught Reference error'.
Uncaught Reference Error: second_num is not defined.

Q #21) What is the difference between '==' and '==='?

Answer:

Both '==' and '===' are comparison operators.

'==' operator	**'===' operator**
It is known as 'Type Converting Operator'	It is known as 'Strict Equality Operator'
It compares Value, do not compare type	It compares both value and type.

Q #22) What is the difference between 'let' and 'const'?

Answer :

let	**const**
using 'let' we can change the value of	using 'const', after the first assignment of th

Menu

	const

Consider the code
{
let first_num =1;
first_num=2;
document. write (first_num);
}
Here the code will give an output, since the change in value of first_num is possible.
	Consider the code
	{
	const second_num =1;
	second_num=2;
	document. write (second_num);
	}
	Here the code will produce an error, since the 'second_num' is assigned with a second value.

Q #23) In the following Code snippet can you please predict the output or If you get an error; please explain the error?

```
<!DOCTYPE html>
<html>
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p style='text-decoration:underline'>Example of 'Const' Keyword </p>
<p id="display_first"></p>
<p id="display_second"></p>
<script>
let first_num =500;
first_num=501;
document.getElementById("display_first").innerHTML = "First Number:"+ first_num ;
const second_num =1000;
second_num=1001;
document.getElementById("display_second").innerHTML = "Second Number :"+second_num;
</script>
</body>
</html>
```

Answer: Please refer Ques No: 21 before reading further

The output of the code snippet is:

First Number:501

We will also get an Error while running the code, as we are trying to change the value of a 'const' variable.

Error: *Uncaught TypeError: Assignment to constant variable.*

Q #24) What is the difference between 'null' and 'undefined'?

Answer: Both the keywords represent empty values**.**

, but we won't assign a value to that variable. On the
riable and assign the 'null' value to the variable.

- type of (undefined) and type of (null) object.

Q #25) What is the difference between 'function declaration' and 'function expression'?

Answer:

It can be explained with an Example:

```
<!DOCTYPE html>
<html>
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p style='text-decoration:underline'>Example Function Declaration</p>
<p id="display_add"></p>
<p id="display_sub"></p>
<script>
function add(first_num,second_num){
return first_num + second_num;
}
var substract = function sub(first_num,second_num){
return first_num - second_num;
}
var first_num=700;
var second_num=300;
document.getElementById("display_add").innerHTML = "Sum of the number is:" + add(first_num,s
document.getElementById("display_sub").innerHTML = "Difference of the number is:" + substract(
</script>
</body>
</html>
```

As shown in the example add() is a function declaration and subtract() is a function expression. The syntax of function declaration is like a function which is saved into a variable.

Function declarations are hoisted but function expressions are not hoisted.

Q #26) What are 'settimeout()'?

Answer: (It better to explain this answer with an example)

Consider the code snippet

```
Console.log ('First Line');
Console.log ('Second Line');
Console.log ('Third Line');
```

First Line
Second Line
Third Line

Now you introduce settimeout() method and wrap the same set of code in it.

```
Settimeout(function() {
Console.log (‘First Line’);
},0);
Console.log (‘Second Line’);
Console.log (‘Third Line’);
```

The output of the code snippet here is:

Second Line
Third Line
First Line

With the introduction of settimeout(), the processes become asynchronous. The first statements to be placed in the stack is Console.log (‘Second Line’), and Console.log (‘Third Line’), and they will get executed first. You need to wait until everything in the stack is completed first.

Even though ‘0’ is the timeout period, it doesn’t mean that it will be executed right away.

Q #27) What is a Closure and How do you use it?

Answer: A closure is an inner function. It can access the outer variables of a function. In Closure, within function_1 there is another function_2 which returns ‘A’ value and function_1 also returns a value; say ‘B’.

Here sum() is the outer function and add () is an inner function, it can access all the variables including ‘first_num’ ‘second_num’ and ‘third_num’. The outer function is calling the inner function add().

```
<script>
// To find the sum of two numbers using closure method
function sum( first_num, second_num )
{
var sumStr= 600;
function add(first_num , second_num)
{
return (sumStr + (first_num + second_num));
}
);
```

The output of the code snippet here is:

Result is: 500

Q #28) In the following Code snippet can you please predict the output or If you get an error; please explain the error?

```
<!DOCTYPE html>
<html>
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p style='text-decoration:underline'>Example Assignmnet Statement</p>
<p id="display"></p>
<script>
var x =500;
let y,z,p,q;
q=200;
if(true){
x=y=z=p=q;
document.getElementById("display").innerHTML = "x="+ x + "<br>y :"+ y +"<br>z :"+ z+"<br>p :"+

}
</script>
</body>
</html>
```

Answer: The assignment statements are considered from Right to left.

The output of the code snippet here is:

x=200
y:200
z:200
p:200
q:200

Q #29) Can you give an example where the code snippet shows the difference between test () and exec () methods?

```
<!DOCTYPE html>
<html>
<body>
```

Menu

```
                               ı "How“ in the given string "Hello. Good Morning. How c
                               return the pattern </p>
                               :ton>
<p id="result"></p>
<script>
function searchTxt() {
var str = "Hello. Good Morning. How do you feel today?";
var search_patt = new RegExp("How");
var res = search_patt.exec(str);
document.getElementById("result").innerHTML ="Found the pattern :"+ res;
}
</script>
</body>
</html>
```

Answer: This is an example of the test () and exec () method, Refer Ques No: 5 for more details.

The output of the code snippet here is:

Found the pattern using exec (): How
Using test () the result is: true

Q #30) Can you give an example showing JavaScript Hoisting?

Answer:

```
<!DOCTYPE html>
<html>
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p style='text-decoration:underline'>Example for JavaScript Hoisting </p>
<p id="dispaly_num"></p>
<script>
num = 100; // Assign value 100 to num
elem = document.getElementById("dispaly_num");
elem.innerHTML = "Here the variables are used before declaring it." +
" <br>The value of the variable is " + num;
var num; // Declare the varaible </script>
</body>
</html>
```

Please refer Ques No:11 for more details

Here the variable ‘num’ is used before declaring it. But JavaScript Hoisting will allow it.

The output of the code snippet here is:

Menu

declaring it.

Q #31) Can you give an example showing the use of ‘debugger’ keyword in the JavaScript code?

Answer:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p style='text-decoration:underline'> Example for debug keyword </p>
<p> Here to test the code, debugger must be enabled for the browser, <br>during debugging the
<p id="wait_result"></p>
<p id="show_result"></p>
<script>
var a = 1000;
var b = 500;
var sum = a + b;
document.getElementById("wait_result").innerHTML = "Adding numbers......<br>Select 'Resume Sc
debugger;
document.getElementById("show_result").innerHTML = "Sum of the numbers : "+sum;
</script>
</body>
</html>
```

__Note:__ The debugger must be enabled for the browser to test the code. Refer Ques No: 5 for more details

This is an example of debugging keyword (Browser used: Chrome)

The output of the code snippet here is:

Here to test the code, the debugger must be enabled for the browser,
during debugging the code below should stop executing before it goes to the next line.
Adding numbers…
Select ‘Resume Script execution’ to continue:

<Click on ‘Resume Script execution’ Button>

Sum of the numbers: 1500

Q #32) In the following Code snippet can you please predict the output or If you get an error; please explain the error?

```
<body>
<h2> <strong> Sample: Software Testing Help</strong> </h2>
<p style='text-decoration:underline'>Example Type Converting </p>
<p id="display"></p>
<script>
var first_num =500;
var first_name='500';
if(first_num == first_name){
document.getElementById("display").innerHTML = "Comparison will return 'true' by Type convertin
}
</script>
</body>
</html>
```

Answer: Consider the code

```
If (‘100’==100) {
document. write (“It’s a Type Converting Operator”);
}
Here
  typeof(‘100’) is string
   typeof(100) is number
the ‘==’ operator will convert the number type, which is on the right side of the operator to strin
```

The output of the code snippet here is:

Comparison will return ‘true’ by Type converting Operator

Conclusion

It’s better to store the JavaScript Codes, CSS, and HTML as separate External ‘js’ files. Separating the coding part and HTML part will make it easier to read and work with them. Multiple developers also find this method easier to work with simultaneously.

JavaScript Code is easy to maintain. Same set of JavaScript Codes can be used in multiple pages. If we use External JavaScript codes and if we need to change the code, then we need to change it in one place. So that we can reuse the code and maintain them in a much easier way.

JavaScript Code has better Performance. External JavaScript file will increase the page loading speed as it will be cached by the browser.

Hope you have found the answers to the most frequently asked interview questions on

☰ Menu

30 Top HTML Interview Questions and Answers

Top 20 RESTful Web Services Interview Question and Answers

Top 40 Popular J2EE Interview Questions and Answers

Top 30+ Popular CSS Interview Questions and Answers

TOP JAVA INTERVIEW QUESTIONS AND ANSWERS
Java
© www.SoftwareTestingHelp.com

Top 50+ Core Java Interview Questions and Answers

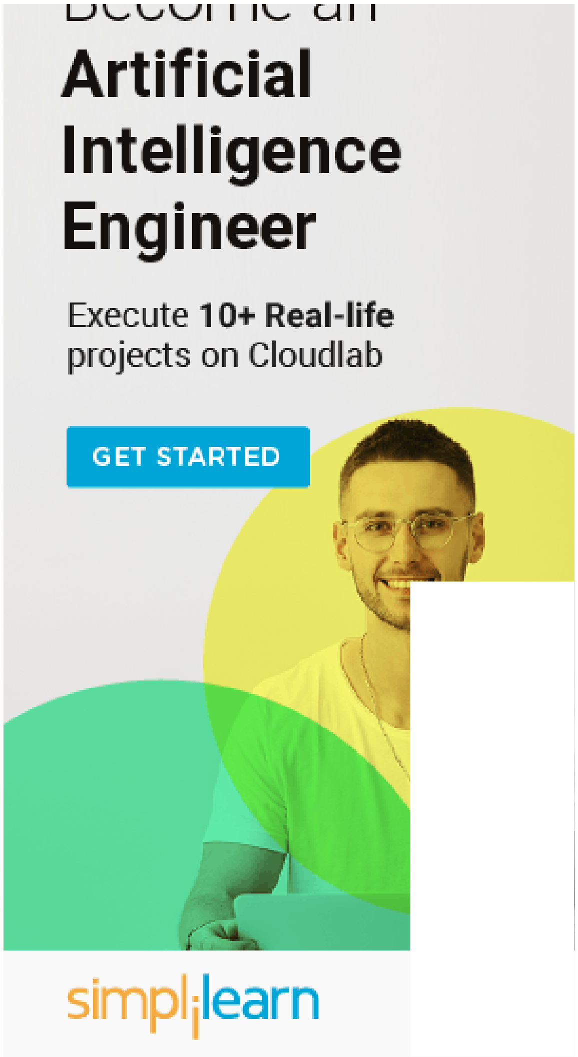
90 TOP SQL Interview Questions and Answers (UPDATED)

Most Important
C# Interview Questions With Answers
© www.SoftwareTestingHelp.com

Top 50 C# Interview Questions with Answers

Top 50 Bootst Questi

Menu


Become an
Artificial Intelligence Engineer
Execute 10+ Real-life projects on Cloudlab
GET STARTED
simplilearn

Join Over 200,000+ Testers

Get premium ebooks and testing tips.

Enter your email here...

SUBSCRIBE NOW!

Join Over 300000+ Followers!

Menu

Top FREE Training Tutorials

Load Testing Buyer’s Guide

Free QA Training

Selenium Tutorials

QTP/UFT Tutorials

ISTQB Free Study Guide

ISTQB Premium Study Guide

Quality Center Tutorials

LoadRunner Tutorials

JMeter Tutorials

JIRA Tutorials

VBScript Tutorials

Best Test Management Tools

Appium Studio Tutorials

Unix Tutorials

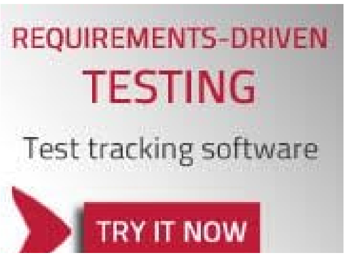
Free DevOps Training

Free JAVA Tutorials

Free Python Tutorials

101+ Interview Questions

Adv



Menu

ABOUT US | **CONTACT US** | **ADVERTISE** | **TESTING SERVICES**

Thanks For Reading! 50 Million Visits and Counting!