

JavaScript Interview Questions for Freshers

1. What is JavaScript?

JavaScript is a client-side scripting language as well as a server-side scripting language. This scripting language can be written into [HTML](#) pages (also could use CSS for styling the pages), and web browsers understand the page.

This scripting language also acts like an object-oriented programming language but not a class-based object-oriented language.

2. Who developed JavaScript, and what is the first name of JavaScript?

JavaScript was created by a Netscape programmer, Brendan Eich.

He developed this new scripting language in just ten days.

At the time of launch, it was initially named Mocha, after which it was known as Live Script and later known as JavaScript.

3. What are the differences between Java and JavaScript?

Java	JavaScript
Java is a complete programming language that can be used for backend coding.	JavaScript is a coded program that can be introduced to HTML pages (otherwise known as server-side scripting language).
Java is an object-oriented programming (OOPS) or structured programming languages like C++ or C and .net.	JavaScript is a client-side scripting language (not fully OOP).
Java creates applications that run in a virtual machine or browser	JavaScript code is run on a browser only.
Java code needs to be compiled	JavaScript code is all in text.

4. What are the JavaScript Data Types?

Following are the Data types present in JavaScript:

Basically, there are two types of JavaScript Data types.

1. Primitive Datatypes
2. Non- Primitive Datatypes

Primitive Datatypes	Non-Primitive Data types
String:-String represents the sequence of characters means the combination of characters, Ex:-‘Hello’.	Object:-Object represents instances through which we all can access members
Number:-Numbers represents the numeric values, Ex:-2000.	Array:-Array represents’ a group of similar type data.
Boolean:-Boolean represents the Boolean value, i.e. either true or false.	RegExp:-It represents Regular Expression.
Undefined:-it Represents undefined value or not defined values.	
Null:-It represents null values means there is no value.	

5. Why Should We Study JavaScript?

JavaScript is one of the three languages all web developers must learn for the following reasons:

1. HTML is used to define the content of web pages. It is otherwise known as the skeleton of web pages.
2. CSS is used to specify the layout or give styling to the web pages, otherwise known as the shape of the body or cover of the skeleton.
3. JavaScript to program the behaviour of web pages or web pages workability.

6. What is the basic use of the is NaN function in JavaScript?

The function returns true if the argument is not a number. If the argument is a number, then it returns as false.

7. Who is faster among JavaScript and ASP Script?

JavaScript is faster.

JavaScript is more rapid because, as JS is a client-side language and that it does not need any assistance or help of the webserver to execute, but on the other hand, ASP is a server-side language. That's why ASP is always slower than JavaScript.

JS now is also known as a server-side language named NodeJS.

8. What do you mean by negative infinity?

Negative Infinity is nothing but a number in JavaScript that can be derived by dividing negative numbers by zero. This could be generated by arithmetic operations.

9. Can we break JavaScript Code into several lines? If yes, then How?

Yes, We can break JavaScript code into several lines; we can break within a string statement using a backslash ('\\') at the end of the first line code.

For example,

```
document.write ("This is \\a program");
```

And when you are not within a strong statement and want to change to a new line, then JavaScript ignores the break in the line.

For Example:

```
var x=1, y=2,
```

```
z=x+y;
```

The above code is perfect for better understanding, but it might hamper our debugging, so it is not advisable to write.

10. What are undeclared and undefined variables?

When variables are not declared in a program, then it is known as Undeclared Variables.

If no variable exists in our program and the program wants to read those variables, it will generate a runtime error.

When there is the declaration of a variable given no value to the variable inside a program is known as an Undefined Variable.

When a program wants to read the variable's value, then the undefined values are returned.

JavaScript Coding Questions

11. Write a code for adding new elements dynamically in JavaScript?

```
<html>

<head>

<title>Elements Dynamically</title>

<script type="text/javascript">

    function addNode()

    {

        var newP = document.createElement("p");

        var textNode = document.createTextNode(" This is a new text node");

        newP.appendChild(textNode);

        document.getElementById("Dynamic").appendChild(newP);

    }

</script>

</head>

<body>

<p id="Dynamic">Dynamic</p>

</body>

</html>
```

This is a simple code for representing how to add new elements dynamically.

So, at first, we set the boilerplate of HTML or the HTML structure. After that, we mention or indicate to DOM that it is a JavaScript file.

After that, we tell a function `addNode`, then we create an element of `p` or paragraph type, then we create text node by applying `createTextNode` ("This is a new text node"); then we will append or assign the text to this `p`-type.

For execution, we specify `<p id="dynamic">dynamic</p>` then we can process our program by `document.getElementById("Dynamic")`; and append our `p`-type then the code will successfully be executed

12. What do you mean by global variables? Define the Declaration and Problem with Global Variable?

The variable which has no scope or available throughout the length of the code is otherwise known as the Global variable.

For declaring a local variable, a `var` keyword is used. It is also applicable for declaring an object. When there is the commission of the `var` keyword, then the global variable is declared.

For Example:

```
// Declaration of a global globalVariable = "Test";
```

13. What is a prompt box?

A prompt is a type of box. It allows the user to enter their input, provide a text box, number, and text provided by label and box.

14. What do you mean by the 'this' keyword in JavaScript?

In the case of Java, the 'This' keyword is used to point to the current object, but in JavaScript, the 'This' keyword refers to them from where it was called.

In other Words, we could say that the "this" keyword refers to the object it belongs to.

So, the 'this' keyword has different values according to or depending upon where it is used.

1. If we use it in a method then, this refers to the owner object.
2. If it is alone, then this refers to the Global object.
3. If it is used in a function, then this refers to the global object
4. If it is used in a function, in strict mode, then this remains undefined.
5. In an event, this refers to that respective element that receives the event.

Call() and apply() method refer 'this' to any object.

15. Explain the working of timers in JavaScript?

Timers are used to execute a bit of code at a set time and repeat the bit of code in a given interval.

Those working are done by using functions setTimeout, setInterval, clear interval.

Working of Functions:

The setTimeout(function, delay) function is used to start a timer, which calls a particular function after the, particularly mentioned delay.

The setInterval(function, delay) function is used to repeatedly execute the given function in the said delay and only stops when it is cancelled.

The clearInterval(id) function instructs or indicates the timer to stop (for stopping the given or mentioned function).

The whole timers are operated within a single thread, and for his events might wait or queue up, they wait for their execution.

16. How do we Define comment's in JavaScript?

For Single line comments:-"/" is used.

For Multi line comments:-"/* is used for multi-line comment */".

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

```
var myarray = new Array (1,4,3,6,10,0,22)
```

```
document.write(myarray.sort())
```

```
myarray.sort(function(a, b) { return b - a; });
```

```
document.write(myarray);
```

Ans: [0,1,10,22,3,4,6] and [22,10,6,4,3,1,0]

The `sort()` method sorts the elements of an array. The sort order can be either alphabetic or numeric, and either ascending (up) or descending (down). By default, the `sort()` method sorts the values as strings in alphabetical and ascending order.

18. Create a button element with a value attribute set to “CLICK ME and WAIT” and invoke a click event on the button, which calls a function that changes the button’s text value color to red. At the same time, this function also sets up a timed function using `setTimeout()` that sets the text color back to black after 5 seconds.

```
<script type="text/javascript">
```

```
function setRed ( )
```

```
{
```

```
document.getElementById("Button1").style.color = "#FF0000";
```

```
setTimeout ( "setBlack()", 5000 );
```

```
}
```

```
function setBlack ( )
```

```
{
```

```
document.getElementById("Button1").style.color = "#000000";
```

```
}
```

```
</script>
```

```
<input type="button" name="clickMe" id="Button1" value="Click me and wait!" onclick="setRed()"/>
```

19. What is the output of the below code?

```
function checkAge(age) {
```

```
if (age < 18) {
```

```
    const message = "Sorry, you're too young to get your driving license.";
```

```
}
```

```
else {
```

```
    const message = "Yay! You're are eligible!";
```

```
}
```

```
    return message;
```

```
}
```

```
console.log(checkAge(21));
```

Ans: Reference Error

Variables with the `const` and `let` keyword are block-scoped. A block is anything between curly brackets (`{ }`). In this case, the curly brackets of the `if/else` statements. You cannot reference a variable outside of the block it's declared in, a Reference Error gets thrown.

20. Write some JavaScript that uses the current clock and gives an alert message whether the science class is over (Assume class ends at 02:30).

Ans: Possible solution:

```
<script>
```

```
var cTime = new Date();
```



```
var hour = cTime.getHours();
```

```
var mins = cTime.getMinutes();
```

```
if (hour > 12 || hour == 12 && mins > 20) {
```

```
    alert("Yay, science class got over!!");
```

```
    } else {
```

```
        alert("Hang on, the science class is yet to finish!");
```

```
    }
```

```
</script>
```

21. How many alert dialogs will the following Javascript generate, and what will be displayed in each of them?

```
var x = "20";
```

```
function func1(){
```

```
    var x = "5";
```

```
alert(this.x);
```

```
function func2(){alert(x);}
```

```
func2();
```

```
}
```

```
func1();
```

Ans: A. There will be 2 alert dialogs. The first will display "20", and the second will display "5".

JavaScript interview questions for Intermediate

22. Write the Difference between ViewState and SessionState?

ViewState	SessionState
ViewState' is specific to only a page in a session.	SessionState' specific to user data to access all pages inside the web application.
ViewState is only visible from a single page, not from multiple pages.	In SessionState, the data availability is access through all of the web pages.
In ViewState, information is stored on the client-side.	In SessionState, information is stored on the server.
ViewState value could be lost or cleared when a new page loaded.	The programmer himself clears the SessionState value, and in other cases is when the timeouts value gets cleared.

23. What do you mean by the "===" operator?

This operator is called a strict equality operator.

It returns true when the two operands have the same value without having any type of conversion.

So, we could say that it is a strict equality operator which returns false values are similar types.

24. Explain briefly how you can submit a form using JavaScript?

In JavaScript, you can use an on click event for submitting a form, i.e., `form.submit()` method.

You can perform a submit action by using submit button, By clicking on Hyperlinks.

For submitting a form using JavaScript use `document.form[0].submit();`

For Example:

```
document.form[0].submit();
```

25. Does JavaScript support automatic type conversion (AutoConversion)?

Automatic Type conversion is otherwise known as AutoConversion. When a lower precision data type gets converted to a higher precision data type, it is called typecasting.

For Ex: When we want to convert byte type data to short type data, it could be done easily as a byte is a lower precision data type and a short data type.

ByteShort

Yes, JavaScript supports automatic type conversion; type conversion is the common way followed by JS programmers or Developers.

Automatic Type conversion is otherwise known as “Widening conversion” and “Implicit conversion”.

It is the simple concept of assigning lower data types to higher data types.

26. How should we change the style/class of an element?

First of all, we can use the `className` to assign a value directly to the class. If any such classes are already present on the element, then this will override them.

For getting the value of class on the element, we can add multiple spaces using `className`.

Given the following way, we can change the class of an element:

```
document.getElementById("myText").style.fontSize = "20";
```

or,

```
document.getElementById ("myText").className = "any class";
```

27. Explain about Read and Write of a file using JavaScript?

Basically, there are two ways to read and write a file:

1. Using JavaScript extensions.
2. Using a web page and Active X objects.

Given are the steps to read and write a file:

Step 1:

```
file=fopen(getScriptPath(),0);
```

`fread()` is used for reading the file content

Step 2:

```
str = fread(file,flength(file);
```

The function `fwrite()` is used to write the contents of the file.

Step 3:

```
file = fopen("c:\MyFile.txt", 3); // it is used for opening the file for writing.
```

```
fwrite(file, str);
```

// str which is specified within the method is the content that is to be written into the file.

28. What are all the loops available in JavaScript?

The available loops available in JavaScript are given below:

- **For loop statement:**

The JavaScript For loop is the same as for loops of Java and C. For loop continues until a specified condition evaluates to false.

Ex:

```
for(int i=0; i<=3; i++)
```

- **While loop:**

A while statement executes its statement until a specified condition evaluates to true, a while statement looks like as follows:

Ex:

```
while(Condition)
```

```
Statement
```

- **do-while loops:**

The do-while loop continues until a specified condition is false.

Ex:

```
do
```

```
statement
```

```
while(Condition)
```

29. What is called Variable typing in JavaScript?

Variable typing is nothing much firstly, it is used to assign a number to a variable, and after that, the same variable is assigned to a String.

For Example:

```
i = 10;
```

```
i = "string";
```

So this is called variable typing. This concept of JS is similar to Java.

30. Function's for converting a string of any base class to integer in JavaScript?

The `parseInt()` function's responsibility is to convert numbers between different bases.

`parseInt()` function takes the string to be converted as its first parameter, and the second parameter is the base of the given string.

31. Explain the difference between the "==" and "===" operator?

"==" operator checks only for equals of the value whereas,

"===" operator also checks equality of value but in a stricter way and returns false if either the value or the type of variables are different.

32. What Should be the answer of 3+2+ "7"?

As we all know that 3 and 2 are integers or integer types, so they will be added numerically. And after that, since 7 is a string, it doesn't add, and it will be concatenated because the string always is concatenated and not added, So the final result should be 57.

33. What do you understand by "this" keyword in javascript?

Ans: In JavaScript "this" keyword is widely used and it points to a specific object that is executing a current piece of code. It refers to the object that is executing the current function. If the function is a regular function, "this" refers to the global object. If the function referenced is a method in an object, "this" refers to the object itself.

Below are the rules that apply to "this" keyword to know which object it is referencing:

1. Global Scope

If a function is called from a global scope which includes 'this' keyword, it will always point to the window object.

For Example:

```
<script>
```

```
//global declaration of num having global scope
```

```
var num = 150;
```

```

function global_this() {

// local variable inside function has local scope

    var num = 250;

    alert("mynum = " + mynum); // answer is 250 called locally

    alert("this.mynum = " + this.mynum); // answer is 150 as "this" keyword points to //global variable
    i.e., window object.

}

global_this();

</script>

```

2. Object's Method

By using a new keyword if an object is created then "this" will point to that specific object

For example:

```

<script>

// declaring global variables 100 has global scope

var myNum = 100;

function printNum() {

    this.myNum = 300;

    this.display = function () {

        var myNum = 400;

        alert("myNum = " + myNum); // 400

        alert("this.myNum = " + this.myNum); // 300

    };

}

var objM = new printNum();

```

```
objM.display();
```

```
</script>
```

“this” keyword inside the display() method of the object ‘objM’ will point to the value outside the scope of the display() method.

3. **call() or apply() method**

A function can be invoked using the () operator or using call() and apply the () method also. The main aim of call() and apply() is to set the context of “this” keyword inside a function regardless of whether that function is being called in the global scope or as an object’s method.

4. **bind() method**

The bind() method is used to set the context of ‘this’ keyword to a particular object when a function is invoked. It is mostly helpful in placing the context of this for a callback function.

For example:

```
var myNum = 50;
```

```
function test() {
```

```
    alert(this.myNum);
```

```
}
```

```
var obj_1 = { myNum : 100 , demo1: test};
```

```
var obj_2 = { myNum : 150 , demo1:test };
```

```
test(); // ‘this’ will point to the global window object
```



```
test.call(obj_1); // 'this' will point to obj_1
```

```
test.apply(obj_2); // 'this' will point to obj_2
```

```
obj_1.demo1.call(window); // 'this' will point to global window object
```

```
test.apply(obj_2); // 'this' will point to obj_2
```

```
<script>
```

```
<script>
```

```
//global scope of variable myNum
```

```
var myNum = 200;
```

```
function test(call)
```

```
{
```

```
//local scope of variable myNum
```

```
var myNum = 250;
```

```
call();

};

var obj1 = {

    myVar: 350,

    Thisfunc : function() {

        alert(“this’ points to myNum ” + this + “ , myNum = ” + this.myNum);

    }

};

test(obj1.Thisfunc); //this points to the global window object

test(obj1.Thisfunc.bind(obj1)); // setting “this” value using bind() method
```

34. What is the result of 3+2+"7"?

Ans: The result is 57. With the + operator 3 and 2 are added. As 7 is a string, the + operator concatenates the expression.

35. List the disadvantages of using innerHTML in javascript.

Ans: The innerHTML property is a part of the DOM and is used to set or return the HTML content of an element. The return value represents the text content of the HTML element. It allows JavaScript code to make changes to a website being rendered.

The disadvantages of using innerHTML are:

1. Appending to innerHTML is not supported without reparsing the whole innerHTML.
2. As reparsing is required for innerHTML the processing is slow and takes more time.
3. The event handlers do not attach automatically to the newly created elements by setting innerHTML. One must keep track of the event handler and attach the new element manually.
4. By using innerHTML if you add, append, delete or modify contents on a webpage all contents are replaced, also all the DOM nodes inside that element are reparsed and recreated.
5. No proper validation is provided by innerHTML, hence any valid HTML code can be used. This may break the document of JavaScript.

36. What is a continue statement in javascript?

Ans: The continue statement moves or jumps over the current iteration in the loop if a particular condition occurs, and continues with the execution of the next iteration in the loop.

It can be used in looping statements such as for loop, while loop, and do-while loop. When it is used in a while loop, then it moves back to the condition. If it is used in a for loop, the flow moves to the update expression.

Using the continue statement, the program's flow immediately jumps to the conditional statements, and if the condition is true, then the next iteration will be started; else, the control comes out of the loop.

Syntax:

```
1(with or without label)
2Continue; Or continue label_name
```

37. List the differences between .call() and .apply()

.call()

.apply()

.call() method calls a function with given values and arguments are provided separately

.apply() method calls a function with given values and arguments are provided in the form of an array or array object.

Syntax: object.objectMethod.call(objectInstance, arguments)

2 parameters:

objectInstance: holds the instance of an

object.arguments: method takes arguments separated by comma .

Syntax:object.objectMethod.apply(objectInstance, arrayOfArguments)

2 Parameters:

objectInstance: holds the instance of an object.

arrayOfArguments: method takes the array of arguments.

Example:show.call(object, “1st argument”, “2nd argument”);

Example: show.apply(object, [“1st argument”, “2nd argument”, “3rd argument”]);

38. What do you understand by event bubbling?

Ans: Event flow specifies the order in which events are received on the page from the element where the event occurs and propagated through the DOM tree. There are two main event models: event bubbling and event capturing.

In the event bubbling model (bottom to top), an event starts at the most particular element and then flows upward toward the least specific element i.e., the document or even the window. For example, you have a div element and a button inside the div element when the button triggers a click event, the click event occurs in the following order:

- button
- div with the id container
- body
- html
- Document

The click event first occurs on the button, which is the element that was clicked. Then the click event goes up the DOM tree, firing on each node along its way until it reaches the document object. Few web browsers these days will bubble the event up to the window object.

39. Is JavaScript case-sensitive?

Yes, JavaScript is a case-sensitive language. JavaScript also has a set of rules for writing JavaScript programs or codes where the identifiers, variables, keywords, and function names must be written using an appropriate capitalization of letters.

For example:

```
<script>
```

```
var name,Name;
```

```
name='abc';    //variable 1
```

```
Name='def';    // variable 2
```

```
document.write(Name);
```

```
</script>
```

In JavaScript, name, and Name are not the same thing even if both variables are spelled the same.

40. What is the difference between a web garden and a web frame?

Web garden	Web frame
Web Garden is the web hosting system that encompasses multiple “processes”.	Web Farm is the web hosting system that encompasses multiple “computers”.
It has an application pool(a container of work processes) that can be configured and can define the number of work processes for that pool.	multiple web servers are available for multiple clients.
used while hosting multiple processes on a single web server.	used while hosting a single web application on multiple web servers in order to distribute the load among the web servers.

41. Describe the role of deferred scripts in javascript.

Ans: The defer Attribute:

The defer attribute tells the browser to continue to process the HTML and build DOM and does not wait till the script file is executed fully. The script loads “in the background”, and then runs later when the DOM is completely built.

For example:

```
<script defer src="samplescript.js">
```

The script is loaded asynchronously, the script file can be downloaded while the HTML document is still parsing, even if the file is fully downloaded before the HTML document is finished parsing, the script is not executed until the parsing is complete. Hence, scripts with a defer attribute will never block the page and always execute when the DOM is completely ready.

42. What are the different functional components of JavaScript?

Ans: The basic JavaScript functions are called functional components. These can be created in two ways:

1. Using function keyword:

```
function functionName (parameters) {
```

```
// Code goes here...
```

```
}
```

2. Using function expression or Anonymous function: declare a variable then assign a function it without name.

```
const show= function () {
```

```
console.log('Anonymous function!')
```

```
}
```

• Can also be declared using arrow function

```
const show=()=>{
```

```
console.log('Anonymous function!')
```

```
}
```

43. What do you mean by screen objects? Explain.

Ans: The screen object is a built-in interface that is used to get the information about the browser screen on which the current webpage is displayed. It provides information about the dimensions of the rendered screen such as screen width, height, colorDepth, pixelDepth, etc.

Property of JavaScript screen object that returns information of the browser:

1. width: returns the width of the screen
2. height: returns the height of the screen
3. availWidth: returns the available width excluding windows taskbar
4. availHeight: returns the available height excluding windows taskbar
5. colorDepth: returns the color depth of color palette, in bits, to render images
6. pixelDepth: returns the color resolution in bits per pixel of the screen.

For example:

```
<script>
```

```
document.write("<br/>The display screen width is: "+screen.width);
```

```
document.write("<br/>The display screen height is: "+screen.height);
```

```
document.write("<br/>The display screen available width is: "+screen.availWidth);
```

```
document.write("<br/>The display screen available height is: "+screen.availHeight);
```

```
document.write("<br/>The display screen color depth is: "+screen.colorDepth);
```

```
document.write("<br/>The display screen pixel depth is: "+screen.pixelDepth);
```

```
</script>
```

44. What do you mean by the unshift() method?

Ans: unshift() is a built-in object array method that will add array elements to the front of an array. It overwrites the array original array by adding new array elements to the beginning of an array.

For example:

```
<p id="test"></p>
```

```
<script>
```

```
const number= ["one", "two", "three", "four"];
```

```
number.unshift("five", "six");
```

```
document.getElementById("test").innerHTML = number;
```

```
</script>
```

Output: five, six, one, two, three, four

45. What are the unescape() and escape() functions in javascript?

Ans: The escape() function in JavaScript is used for encoding (the process of converting plaintext to ciphertext) a string.

Syntax:

```
1 | escape(string to be encoded)
```

The unescape() function is used to decode(decrypt) that string encoded by the escape() function.

Syntax:

1 |unescape(string to be decoded)

46. What do you mean by decodeURI() and encodeURI() in javascript?

Ans: The encodeURI() function encodes the complete URI. It also encodes a few special characters: , / ? : @ & = + \$ #

Syntax:

1 |encodeURI(complete uri string to be encoded)

The decodeURI() function decodes the URI generated by the encodeURI() function.

Syntax:

1 |decodeURI(complete uri string that has been encoded)

47. Are ECMAScript and javascript related? How?

Ans: ECMAScript is a standard for creating a scripting language. Introduced by ECMA International and is basically an implementation with which we learn how to create a scripting language.

Javascript is a general-purpose scripting language that follows the specification of ECMAScript. It is mostly an implementation which tells how to use a scripting language.

48. What do you mean by QuickSort Algorithm in javascript?

Ans: Quicksort algorithm is one of the most popular sorting algorithms in any programming language. QuickSort algorithm follows the divide and conquers method. It divides elements into smaller parts based on several conditions and performs the sort operations on those divided smaller parts. It works well when working with large datasets.

The steps on how the Quicksort algorithm works:

First: select a pivot element.

Second: compare all array elements with the selected pivot element.

Third: arrange them in a way that elements less than the pivot element is to its left and greater than the pivot is at its right.

Finally: execute the same operations on both left and right side elements of the pivot element.

Javascript Logical Questions

49. How to detect the operating system on the client machine?

if we want to detect the operating system on the client machine, then we have to use `navigator.appVersion` or `navigator.userAgent` property.

50. What do you mean by NULL in JavaScript?

The NULL is used to represent no-value or no-object.

It implies there shouldn't be no object or null string, no valid boolean value, no number, and no array object. The value will be nothing, or it means null.

51. What is the use of the delete operator?

The Delete keyword is used for deleting purposes. The delete keyword is used to delete the property as well as its value also.

For Example:

```
var student= {age:20, batch:"ABC"};
```

```
delete student.age;
```

52. What do you mean by undefined value in JavaScript?

First of all, an Undefined value means the variable used in the program or code doesn't exist, and another is the value that was not assigned and property doesn't exist, which is known as an Undefined value.

53. Name all the types of Pop up boxes available in JavaScript?

There are several types of pop-box are available in JavaScript.

- **Alert Box:**

An Alert box is used to ensure that the information comes through to the user end. When an alert box pops up, the user has to click the "OK" button for further proceeding.

Syntax:

The `window.alert()` method is used for pop-ups.

```
1Window.alert("Sai");
2alert("I am from great learning");
```

- **Confirm Box:**

A confirm box is used if somebody wants the user to verify or accept something.

When a pop box appears, the user only has to do either click on the "OK" button or click on the "CANCEL" button.

If the user clicks on the "OK" button, then the box returns true, and if the user clicks on the "CANCEL" button, then the box returns false.

Syntax:

```
1Window.confirm("Yes");
2If (confirm("press a button")){
3Txt="you pressed ok";
4}else{
5Txt="you pressed cancel";
6}
```

- **Prompt Box:**

A prompt box is used for the user to input a value before entering into a page.

When a prompt box pops up, the user has to click either on "OK" or "Cancel" to proceed after entering an input value.

If the user clicks on "OK", then the box returns a true value, neither on the clicking of the "cancel" button box returns null.

Syntax:

```
1 | Window.prompt("hello", " hiii");
```

54. Define the use of Void(0) in JavaScript?

The Void(0) is used to prevent/precautionary steps to prevent the page from refreshing, and the passing parameter "zero" is passed while/during calls.

After passing the parameter and calling, Void(0) is used to call another method without refreshing the page.

55. How can a page be forced to load another page in JavaScript?

The following code could be chosen to get desired output:

```
<script language="JavaScript" type="text/JavaScript" >
```

```
<!-- location.href="http://newhost/newpath/newfile.html"; //--></script>
```

56. What is the data type of variables in JavaScript?

All variables in JavaScript are object data types.

57. State the Difference between an Alert Box and a Confirmation Box?

As the name suggests, “alert” gives a pop-up displaying only one button, which is an “OK” button, but in the other case, the Confirmation box displays two buttons that contain one “OK” button and another one is the “CANCEL” button.

This is the fundamental difference between the alert box and the confirmation box.

58. What do you mean by Escape characters?

The Escape Character or backslash is placed or used before characters to make them visible.

The Escape character (Backslash) is used when working with a special type of character like quotes, double quotes, apostrophes, and ampersands.

For Example:

```
document.write "I m a "good" boy"
```

```
document.write "I m a \ "good\" boy"
```

59. What do you mean by Cookies JavaScript?

Cookies are data, which are stored in small text files on our computer.

We can Say cookies are used to visit websites faster after accepting accept the cookies.

Cookies are the small text files stored in a computer, and cookies get created when the user visits the websites to store information to use it at the time of their need.

Simply, we can say that the cookies concept is introduced for remembering information about the user.

Ex:

username: Great Learning;

60. How should we create a cookie in JavaScript?

By using JavaScript, we can create, read, and delete (crud operations) cookies with the "document.cookie" property.

To create a cookie JavaScript:

```
document.cookie = "username=John Doe";
```

You can also add an expiry date for a cookie, but, by default, the expiry date of a cookie is when you close your respective browser, the cookie gets deleted or expired.

61. How can you read a cookie in JavaScript?

By using our JavaScript concept we can read a cookie given below:

```
var x = document.cookie;
```

62. Explain what the pop() method in JavaScript is?

The simple working of pop() method to removing the last element from an array and returning that element, by working with this method, also changes the length of the array.

The pop() method is acts similar to the shift() method.

The difference is that is the Shift() method works at the start of the array, and also the pop() method takes the last element of the specified or given array and returns it. After the array is called, it is altered.

For Example:

```
var cars = ["Audi", "BMW", "Mercedes"];
```

```
cars.pop();
```

```
//Now cars become Audi, BMW.
```

63. Is JavaScript contains concept level scope?

The scope is the context on where the variables for functions can be accessed, as you all write in Java and C, C++. I.e. defined by { }.

The concept level is otherwise known as block-level scope. As JavaScript supports Function-level scope.

So, JavaScript does not have a concept-level scope.

Due to the variables declared inside or within the function have scope inside the function.

64. What are the two primary groups of data types in JavaScript?

The primary groups of data types JavaScript are mentioned below:

1. Primitive Types
2. Reference Types

Primitive Types:

Primitive types are number and Boolean type data types.

Reference Types:

Reference types are the more complex types. It is like strings and dates.

65. What is DOM?

DOM stands for Document Object Model.

When a web page is loaded, then the browser creates a Document object model of that page.

The Document Object Model defines a standard or rules for accessing documents on web pages.

66. What is the HTML DOM model?

Simply, we can say that an HTML DOM is nothing but a defining standard for how to get, change, delete, add HTML elements.

The HTML DOM is a standard object model and programming interface for HTML.

67. How can Generic objects be created?

Generic objects can be created as:

```
var I = new object();
```

68. Write the uses of the typeof operator in JavaScript?

The “typeof” operator is used to return a string description of the type of a variable.

Javascript interview questions for Experienced

69. How to handle exceptions in JavaScript?

The exception is the abnormal termination of a program is called an exception.

An unwanted or unexpected event that disturbs the normal flow of the program is otherwise called an exception.

Exceptions are caused by our program, not by our lack of system resources.

Defining an alternative way to continue the rest of the program is typically called exception handling.

So, we can handle the exception with Try-catch and finally keyword, or we can say that Try-Catch block, finally is used to handle exceptions in JavaScript.

Try-catch block is used to handle the exception, and the ‘finally’ block is bound to call either their call of try-catch block or not, but finally, the block is bound to call.

In the final block, the programmer can destroy the connection.

Syntax:

```
1Try{
2//Code
3}
4Catch(exception){
5           //Code to throw an exception
6}
7Finally{
8           //Code runs either it finishes successfully or after catch
9}
```

70. Name the keyword which is used to print the text on the screen?

We can write the text on the screen through the document. write keyword.

```
"document.write("Welcome");"
```

After that, it will write welcome on the screen.

71. What do you mean by blur function in JavaScript?

The blur event occurs when the element losses its focus or blurriness.

The primary use of the Blur function in a program is to removing focus from a specified object.

Syntax:

In HTML:

```
1 | <element onblur="HTML">
```

In JavaScript:

```
1 | object.onblur= function(){JavaScript};
```

72. What do you mean by variable typing in JavaScript?

Basically, variable typing is used for assigning a number to a variable and after that assigning string to that same variable.

The steps using which you can get more clarifications are given below through an example:

```
i= 8;
```

```
i="john";
```

73. How to print Statements in JavaScript?

By using Console.log() function in JavaScript, we can print any variables defined before int, and it is also used to print any message that needs to get displayed to the user.

The Syntax for defining or showing elements to the user is:

```
1 | console. log(A);
```


74. How to find an operating system in the client machine using JavaScript?

For finding the Operating system in the client machine using JavaScript using "Navigator. app version" is used.

75. What are the different types of errors are available in JavaScript?

There are three types of errors are in JavaScript:

(1) Load time Errors:

Those errors that come up when we load a web page with inappropriate syntax Errors are otherwise known as Load time Errors. Load time Errors are generated Dynamically.

(2) Run time Errors:

Those Errors which are generated due to misuse of command inside an HTML language are known as Run time errors.

(3) Logical Errors:

Those errors occur due to the formation or writing off bad logic inside the program, which logic has different operations known as Logical errors.

The primary benefit of the push method is that it always adds or appends one or more than one element to the end of an array, so using this method, we can append multiple elements bypassing multiple arguments.

Syntax:

```
1 array.push(item1,item2,.....,item10);
```

For Ex:

```
Var fruits= ["Banana", "apple", "orange"];
```

```
fruits.push ("grapes");
```

Output:

Banana, apple, orange, grapes.

77. What do you mean by unshift method in JavaScript?

Unshift method work is similar to the push method but pushes method to append the elements and unshift method just prepend the elements.

unshift method works beginning of the array. This method is used to prepend one or more than one element at the beginning of an array.

Syntax:

```
1 array.unshift (item1, item2,..., Item10);
```

For Ex:

```
Var fruits= ["Banana", "apple", "orange"];
```

```
fruits.unshift ("Grapes");
```

Output:

Grapes, Banana, apple, orange.

78. What is the difference between JavaScript and Jscript?

The difference between these two scripts is said to be no different.

Both are pretty similar, but the only difference is JavaScript is developed by Netscape and Jscript is developed by Microsoft.

79. How are the object properties assigned in JavaScript?

You can define a property by assigning its value.

As all of you know, all JavaScript variables like objects names and property names are case-sensitive so that we can assign value through their name and properties.

We can assign object Properties in the following ways:

```
obj["class"] = 12;
```

Or,

```
obj.class = 12;
```

80. What do you mean by 'Strict' mode in JavaScript, and how can it be enabled?

Strict mode is used to solve some of the mistakes that hamper the JavaScript engines for work efficiently,

The strict mode adds a specific constraint to JavaScript, so under the Script mode, JavaScript show errors from few codes.

For Enabling the Strict mode, you have to add the string literal "use strict" above the specified file. You can get a clear idea by the following example:

```
function myfunction() {  
  
    "use strict";  
  
    var v = "This is a strict mode function";  
  
}
```

81. Write the various ways to get the status of a Checkbox?

The status for getting status of a checkbox as follows:

```
alert(document.getElementById('checkbox1').checked);
```

If the Checkbox is checked, then the alert will return TRUE.

82. Explain the window.onload and onDocumentReady perform in JavaScript?

.onload:

The Window.onload will execute the codes when the browser loads the DOM tree and other resources like images and objects.

The .onload function is not run further & until all the information on the page is loaded, leading to some delay before any code is executed.

.onDocumentReady:

The `.onDocumentReady` is executed when the DOM is load, without waiting for the resources to load like `.onload` works.

This leads to the `.onDocumentReady` allows to executes the code faster in DOM.

In the case of `.onDocumentReady`, it loads the codes just after the Document Object Manipulation is loaded, so this allows early manipulation of the code.

83. What do you mean by closures in JavaScript? When are they used?

The closure is nothing but a locally declared variable or otherwise known as the local variable, which is related to a function, and it stays in memory at the time when the function has returned.

For Example:

```
function greet(message) {  
  
    console.log(message);  
  
}  
  
function greeter(name, age) {  
  
    return name + " says howdy!! He is " + age + " years old";  
  
}
```

// For Generating the message:-

```
var message = greeter("James", 23);
```

// Pass it Explicitly to greet:-

```
greet(message);
```

So the above Function could be better represented by using closures in this function:

```
{  
  
    var message = name + " says howdy!! He is " + age + " years old";
```

```
        return function greet()

        {

            console.log(message);

        };

    }
}
```

// For Generating the closures:-

```
var JamesGreeter = greeter("James", 23);
```

// For Using the closures:-

```
JamesGreeter();
```

84. What do you mean by anonymous function in JavaScript? Describe its properties.

A function said to be an anonymous function when the function is declared without any named identifier.

In other words, we can say that the anonymous function is inaccessible after their declaration in a program or a code.

Anonymous function declaration:

```
var anon = function()

{

    alert('I am anonymous');

};

anon();
```

Common JavaScript Interview questions

85. Explain Hoisting in JavaScript.

Hoisting is a JavaScript behaviour where anywhere the variable/functions are declared they are moved to the top of the scope. Note the scope may be local or global.

Ex: Before Hoisting

```
a = 1;
```

```
alert ( ' a = ' + a);
```

```
var a;
```

Ex: After Hoisting

```
var a;
```

```
a = 1;
```

```
alter( ' a = ' + a);
```

86. Explain Implicit Type Coercion in JavaScript.

Automatic value conversion from one data type to another data type is known as Implicit type coercion in JavaScript. There are many types Number to String, String to Number, Boolean to Number etc. When different values are applied to the datatypes that are not of that value it automatically converts them.

1. String Coercion:

The time when the number is added to a string, it converts the number type as a string type with the help of the "+" operator.

Example:

```
var a = 4;
```

```
var b = "5"; //number that's inside double quotes indicate it's been considered as a string
```

```
a + b // Returns "45"
```

Example:

```
var p = 45;
```

```
var q = "Hi"; // Here the number 45 is been concatenated with the string
```

```
p + q // Returns "45Hi";
```

NOTE: We can use the " – " operator as well, but the number which we give will be converted to string datatype and subtracted

Example:

```
var p = 5;
```

```
var q = "5";
```

```
p - q //Returns 0 since the variable q (string type) is converted to a number type
```

2. Boolean Coercion

When we use logical operators, if statements, ternary operators and loop checks Boolean coercion takes place. We have to ensure properly the truthy and falsy values.

Except 0, O(n), -0, "", Null, undefined and Nan all others are truthy values

Example:

```
var a = 0;
```

```
var b = 24; // variable declaration with value
```

```
if(a) { console.log(a) } // Since the a value is 0 the code will not run ( Falsy)
```

```
if(b) { console.log(b) } // Since the b value is 24 the code will run(Truthy)
```

3. Logical Operators:

Comparing the logical operators in the other programming languages, in JavaScript, the Logical operators does not return true or false, one of the operands is returned.

OR (||) operator – The first value is returned only when it is truthy else the second value gets returned.

AND (&&) operator – When both the values are truthy it returns always the second value. If the first value false it returns the first value. If the second value false it will return the second value.

Operator	First Value	Second Value	Returns
----------	-------------	--------------	---------

OR ()	Truthy	Falsy	First Value
Falsy	Truthy	Second Value	
Falsy	Falsy	Second Value	
Truthy	Truthy	First Value	
AND (& &)	Truthy	Falsy	Second value
Falsy	Truthy	First Value	
Truthy	Truthy	Second Value	

Example:

var a = 260;

var b = "Hi";

var c = NaN;

a || b // Returns 260 since the first value is truthy

a || c // Returns 260 since the first value is truthy

a && b // Returns "Hi" since both the values are truthy

b && c // Returns NaN since the second value is falsy


```
if( a && b )
```

```
{
```

```
console.log("Code runs" ); // This block runs because a && b returns "Hi" (Truthy)
```

```
}
```

```
if( a || c )
```

```
{
```

```
console.log("Code runs"); // This block runs because a || b returns 260(Truthy)
```

```
}
```

Equality Coercion

The operator we use for Equality coercion is " == ". It compares only values not datatypes.

If there exist two different datatypes then it converts both of them to one type and compares the value.

Example:

```
var a = 12;
```

```
var b = "12";
```

`a == b` //operands are equal since it converts both the datatypes to same and compares

87. Is JavaScript a statically typed or a dynamically typed language?

JavaScript is categorized as a dynamically typed language. Because the variable type is checked during runtime parallelly with statically typed language, where the types of a variable are checked during the compilation phase.

Static Typing	Dynamic Typing
<pre>string namename = "Peter";name = 34;</pre>	<pre>var namename = "Peter";name = 35;</pre>
Variables have types	Variables have no types
Values have types	Values have no types
Variables cannot change types	Variables can change types

This is one of the important Javascript interview questions asked in interviews.

88. Explain passed by value and passed by reference.

JavaScript provides two different categories of datatypes Primitive and Objects

Primitive Datatypes: Number, Boolean, String, Null and Undefined.

Objects: Arrays, functions, plain objects and more Anything except primitive are objects.

NOTE: All primitive data types in JavaScript is passed by value

Pass by value:

In this, the function is called by passing the value directly as an argument. So, any changes that's made inside a function won't affect the actual value.

The parameters passed as arguments are mutated (Creation of own copy). So, any changes made inside the function is made to a copied value but not for the original one.

Example:

```
<script>
```

```
let a = 1;
```

```
let change = (val) =>
```

```
{
```

```
  val = 2
```

```
}
```

```
change(a);
```

```
document.write(a);
```

```
</script>
```

In the above example variable, a is been assigned with the value 1 and then changed to 2 inside the function called change. Since JavaScript pass by value the output will be 1.

Pass by Reference:

In some instances, there arises a situation, the address is passed instead of arguments to call a function. During that time the value gets changed inside a function affects the variable passed outside the function. This is called a pass by reference. In JavaScript, mostly arrays and objects follow pass by reference.

In the following example, an object named 'b' is declared outside the function 'change'. Here one should heed that variable 'b' got mutated but not assigned with value 2, as shown in example 2. A pass by reference takes place when a mutation has occurred.

Example 1:

```
<script>
```

```
let b = {num:1};
```

```
let change = (val) =>
```

```
{
```

```
    val.num = 2
```

```
}
```

```
change(b);
```

```
document.write(JSON.stringify(b));
```

```
</script>
```

Example 2:

```
<script>
```

```
let b = {num : 1};
```

```
let change = (val) => {
```

```
  val = {num :2};
```

```
}
```

```
change(b);
```

```
document.write(JSON.stringify(b));
```

89. What is an Immediately Invoked Function in JavaScript?

Ans:

The function which runs as soon as its defined is known as the Immediately Invoked Function in JavaScript.

Syntax:

```
function ()
```

```
{
```

```
// Do something;
```

```
})();
```

90. Explain Higher-Order Functions in JavaScript

Where starting a function with the bracket is necessary else it will be considered as the normal function. The second set of parentheses is used to invoke the function because functions do not work without invoking.

Just like data types such as Number, Boolean, String is considered to be the data then functions can also work as data.

1. Functions can be passed through other functions
2. Functions can be set as object properties
3. Functions can be stored in arrays
4. Functions can be set as variables

Example:

```
function higher-order(fn)
```

```
{
```

```
  fn();
```

```
}
```

```
higherOrder(function() { console.log("Hi") });
```

91. Explain call(), apply() and, bind() methods.

Ans:

call()

It is a library method available in the JavaScript

This method invokes functions/methods by mentioning the owner object

Example:

```
function sayHi()
```

```
{
```

```
    return "Hi" + this.name;
```

```
}
```

```
var obj = {name: "Shiv"};
```

```
sayHi.call(obj); // Returns "Hi Shiv"
```

This method allows using the method (function) of another object

apply()

This method is the same as call() but apply () takes arguments in the form of array whereas call () takes arguments separately

Example

```
function saySomething(message)
```

```
{
```

```
    return this.name + " is " + message;
```

```
}
```

```
var person4 = {name: "Rita"};
```

```
saySomething.apply(person4, ["awesome"]); //Returns Rita is awesome
```

bind ()

This method uses the "this" keyword which will be bound to the parent object which is considered as a parameter. This method will always return a new function

Example:

```
var bikeDetails = {
```

```
    displayDetails: function(registrationNumber,brandName){
```

```
        return this.name+ " , "+ "bike details: "+ registrationNumber + " , " + brandName;
```

```
    }
```

```
}
```

```
var person1 = {name: "Shiv"};
```

```
var detailsOfPerson1 = bikeDetails.displayDetails.bind(person1, "KA020612", "Dio");
```



```
// Binds the displayDetails function to the person1 object
```

```
detailsOfPerson1();
```

```
// Returns Shiv, bike details: KA020612, Dio
```

92. What is Currying in JavaScript?

The transformation of functions from $f(a,b,c)$ callable to $f(a),(b),(c)$ callable is known as Currying in JavaScript. This is an advanced technique used to work with functions not only in JavaScript yet in other languages as well.

Example:

```
function curry(f) { // curry(f) does the currying transform
```

```
  return function(a) {
```

```
    return function(b) {
```

```
      return f(a, b);
```

```
    };
```

```
  };
```

```
}
```

For Example, if we have a function $f(a,b)$, then the function after currying, will be transformed to $f(a)(b)$.

This is one of the important Javascript interview questions asked in interviews.

93. Explain Scope and Scope Chain in JavaScript.

JavaScript variables are also having scope as other programming languages. The accessibility and visibility of the variables are known as Scope.

There are three types of scopes in JS:

- Global Scope

Variables that are not inside any function or the curly braces is known as Global Scope. These variables can be accessed from all parts of the code.

- Local or Function Scope

Variables declared inside a scope or function is known as function. Variables where it can be accessed only within that function. That means they cannot be accessed outside code.

- Block Scope

The variables in blocked scope are limited only to that particular block within the curly braces mentioned.

Scope Chain: The currently accessible scopes in a code are known as Scope chains. Irrespective of the scopes either it might be under global, local or block. When the JavaScript engine searches for a scope currently, the accessible scopes are termed to be scope chains.

94. What are object prototypes?

The mechanism in which the objects inherit features from one another is known as an object prototype.

Example: (only for understanding purpose non-technical)

Let's talk about a car guess what the features are, every car has an engine, a staring and 4 wheels. It might be brand X, Y, or Z every car moves using the same mechanism.

So, relating it to object prototype. Brand X is an object same prototype of a basic mechanism is inherited by brand Y with exterior changes and added features. So, this process is known as object prototyping.

95. What are callbacks?

A callback is a function passed as an argument to another function. It acts as a sequencing system for function execution. Once the function is been executed using call back, we can wait for the result and then execute the next function in the sequence.

Ex. setTimeout () timeout method is usually used to cover up the amount of time taken to execute the program.

This is one of the important Javascript interview questions asked in interviews.

96. What is memoization?

The optimization technique which speeds up the applications by storing the results needed to an immediate function calls and returning to the cached result when the same inputs are supplied again is known as memorization

Example:

```
function memoizedAddTo256() //function declaration
```

```
{
```

```
    var cache = {}; // variable declaration as cache
```

```
    return function(num){
```

```
        if(num in cache){
```

```
console.log("cached value"); // if condition is true it returns cached value
```

```
return cache[num] /
```

```
}
```

```
else{
```

```
cache[num] = num + 256;
```

```
return cache[num];
```

```
}
```

```
}
```

```
}
```

```
var memoizedFunc = memoizedAddTo256();
```

```
memoizedFunc(20); // Normal return
```

```
memoizedFunc(20); // Cached return
```

97. What is recursion in a programming language?

Ans:

Recursion is a technique in which the function calls itself again and again repeatedly until the condition gets false.

Example:

```
function countDown(number) {  
  
    // number is displayed  
  
    console.log(number);  
  
    // value of the number gets decreased in each iteration  
  
    const newNumber = number - 1;  
  
    // condition is passed until its false  
  
    if (newNumber > 0) {  
  
        countDown(newNumber);  
  
    }  
  
}
```

```
countDown(4);
```

Output: 4 3 2 1

98. What is the use of a constructor function in JavaScript?

Ans:

A function that creates an instance of a class which is called an object is known as a constructor. Whenever the object is using a new keyword then the constructor gets called. Constructor is used to creating an object and set values if there are any object properties present.

99. What are arrow functions?

Ans:

The function which allows declaring shorter syntax which was introduced during the ES6 version is known as the arrow function.

Example:

Before

```
hello = function ()  
  
  {  
  
    return "Hi!";  
  
  }
```

After:

```
hello = () =>  
  
  {  
  
    return "Hi!";  
  
  }
```

This is one of the important Javascript interview questions asked in interviews.

100. Differences between declaring variables using var, let and const.

Ans:

var is having scope only within the function

let and const are having scope within their blocks between the curly braces

Also, any variable with the keyword const cannot be changed or modified it remains constant

keyword	const	let	var
block scope	yes	yes	no
function scope	yes	yes	yes
global scope	no	no	yes
can be reassigned	no	yes	yes

101. What are the rest parameter and spread operators?

Ans:

The operator that allows to call a function with n number of arguments and access those extra arguments as an array is known as rest operator (...) it also allows to destructure array or objects.

As the name suggest the spread operator (...) allows us to expand an iterable like array into its individual elements.

Sample Code:

```
<!DOCTYPE html>
```

```
<html lang="en" >
```

```
<head>
```

```
<meta charset="UTF-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
<title>Document</title>
```

```
<style>
```

```
body {
```

```
    font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
```

```
}
```

```
.sample, .result {
```

```
    font-size: 20px;
```

```
    font-weight: 500;
```

```
}
```



```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>JavaScript Rest and spread operator</h1>
```

```
<div class="sample"></div>
```

```
<div style="color: green;" class="result"></div>;
```

```
<button class="btn">Spread Operator</button>
```

```
<h3>
```

Click on the above button to concatenate array using spread operator

```
</h3>
```

```
<button class="btn">Rest Operator</button>
```

```
<div style="color: green;" class="result"></div>
```

<h3>

Click on the above button to add some numbers using rest operator

</h3>

<script>

```
let sampleEle = document.querySelector(".sample");
```

```
let btnEle = document.querySelectorAll(".btn");
```

```
let resEle = document.querySelectorAll(".result");
```

```
let arr = [2, 3, 4, 5];
```

```
let arr1 = ["a", "b", "c", "d"];
```

```
sampleEle.innerHTML = "arr = " + arr + "<br> arr1 = " + arr1;
```

```
function addArr(num, ...ar) {
```

```
  let sum = num;
```

```
ar.forEach((item) => {
```

```
    sum += item;
```

```
});
```

```
resEle[1].innerHTML = "Sum of the elements = " + sum;
```

```
}
```

```
btnEle[0].addEventListener("click", () => {
```

```
    resEle[0].innerHTML = "Concatenated array = " + [...arr, ...arr1];
```

```
});
```

```
btnEle[1].addEventListener("click", () => {
```

```
    addArr(44, 11, 35, 44, 22, 99);
```

```
});
```

```
</script>
```

</body>

</html>

102. What is the use of promises in JavaScript?

Ans:

Any asynchronous operations that occur in JavaScript is handled by promises

There are four stages of promises in JavaScript:

1. Pending – It acts to be a waiting list neither fulfilled nor rejected. It is the initial state.
2. Fulfilled – Any Asynchronous operation is completed to ensure that the promise has been fulfilled.
3. Rejected – Asynchronous reason that's incomplete ensures that the promise has been rejected.
4. Settled – This is a neutral state where the promise is neither rejected nor fulfilled.

This is one of the important Javascript interview questions asked in interviews.

103. What are classes in JavaScript?

Ans:

The templates of JavaScript objects are Classes in JavaScript

Every class in JavaScript must have a constructor with it.

Syntax:

```
class ClassName
```

```
{
```

```
  constructor() { ... }
```

```
}
```

Example:

```
class vehicle {  
  
    constructor(name, year)  
  
    {  
  
        this.name = name;  
  
        this.year = year;  
  
    }  
  
}
```

104. What are generator functions?**Ans:**

As the name suggests this function helps to generate new values using the keyword yield. The major work of the yield keyword is to pause the execution of the function in the middle send the details to the function call and resume at the state where the yield was before the interruption.

Syntax:

```
// An example of generator function
```

```
function* gen(){  
  
  yield 1;  
  
  yield 2;  
  
  ...  
  
  ...  
  
}
```

Example:

<script>

```
var createOwnIterable = {  
  
  *[Symbol.iterator]() {  
  
    yield 'a';  
  
    yield 'b';  
  
    yield 'c';  
  
  }  
  
}
```

```
for (let value of createOwnIterable) {
```

```
    document.write(value);
```

```
    document.write("<br>");
```

```
}
```

```
<script>
```

Output:

a b c

This is one of the important Javascript interview questions asked in interviews.

105. Explain WeakSet in JavaScript.

Ans:

Set is a collection of unique and ordered components in JavaScript. WeakSet, like Set, is a collection of unique and ordered elements, but there are certainly major differences: WeakSet only holds objects and no other types. A weakly referred item is one that is contained within the weakset. This means that if an item within the weakset lacks a reference, it will be trash collected. Unlike Set, only it has three methods `add()`, `delete()` and `has()`

This is one of the important Javascript interview questions asked in interview.

Example:

```
const newSet = new Set([4, 5, 6, 7]);
```

```
console.log(newSet); // Outputs Set {4,5,6,7}
```

```
const newSet2 = new WeakSet([3, 4, 5]); //Throws an error
```

```
let obj1 = {message:"Hello world"};
```

```
const newSet3 = new WeakSet([obj1]);
```

```
console.log(newSet3.has(obj1)); // true
```

106. Explain WeakMap in JavaScript.

The map is usually used to store key-value pairs. It can be both primitive and non-primitive types. If the keys and values must be always an object in weakmap. If there are no object references then the garbage collector collects the object.

Example:

```
const map1 = new Map();
```

```
map1.set('Value', 1); // value is set
```

```
const map2 = new WeakMap();
```

```
map2.set('Value', 2.3); // Throws an error
```

```
let obj = {name:"Vivek"};
```

```
const map3 = new WeakMap();
```



```
map3.set(obj, {age:23});
```

107. What is Object Destructuring?

Ans:

Extracting object or elements of an array all in one line of code is known as object destructuring.

Example:

```
const classDetails = {
```

```
    strength: 78,
```

```
    benches: 39, // declarations
```

```
    blackBoard:1
```

```
}
```

```
const {strength:classStrength, benches:classBenches,blackBoard:classBlackBoard} = classDetails;
```

```
console.log(classStrength); // Outputs 78
```

```
console.log(classBenches); // Outputs 39
```

```
console.log(classBlackBoard); // Outputs 1
```

108. What is a Temporal Dead Zone?

Ans:

The variables declared using keyword let or const keywords faces temporal dead zone.

It is a state that occurs when the variables are tried to access before initialization.

Example:

```
x = 23; // reference error
```

```
let x;
```

```
function anotherRandomFunc(){
```

```
    message = "Hello"; // Throws a reference error
```

```
    let message;
```

```
}
```

```
anotherRandomFunc();
```

109. What are built-in methods in JavaScript

Number Methods

| Sr.No. | Method & Description |
|--------|----------------------|
| | |

| | |
|---|---|
| 1 | constructor() returns the function that's created by default |
| 2 | toExponential() Numbers will be displayed in exponential notation |
| 3 | toFixed() used to fix how many numbers should be present to the right of the decimal point. |
| 4 | toLocaleString() used to return the correct value of the number according to the settings of the system |
| 5 | toPrecision() used to define how many numbers are present towards the left and right of the number |
| 6 | toString() Returns the string representation |
| 7 | valueOf() Returns the number value |

Boolean Methods:

| Sr.No. | Method & Description |
|--------|--|
| 1 | toSource() used to return the source that contains Boolean objects |
| 2 | toString() returns true or false according to the condition |
| 3 | valueOf() primitive value of Boolean object is returned |

String Methods:

| Sr.No. | Method & Description |
|--------|--|
| 1 | charAt() : returns the character at the specific index |
| 2 | charCodeAt()Returns Unicode value of the character. |
| 3 | concat()Combines two different strings together |

| | |
|----|---|
| 4 | indexOf()Returns the index of the string if not found will return -1 |
| 5 | length() Returns string length |
| 6 | match() used to match expression with a string |
| 7 | replace() used to replace the string with the regular expression mentioned |
| 8 | search() searches the elements asked for in the string |
| 9 | slice() a new string is returned after cutting out certain elements of a string |
| 10 | split()Separates strings into two. |
| 11 | toLowerCase() converts string to lower case |
| 12 | toUpperCase() converts string to upper case |

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