**1. What is JavaScript?**

JavaScript is a client-side as well as server side scripting language that can be inserted into HTML pages and is understood by web browsers. JavaScript is also an Object based Programming language

**2. 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. These two languages are not at all inter-dependent and are designed for the different intent. Java is an object - oriented programming (OOPS) or structured programming language like C++ or C whereas JavaScript is a client-side scripting language.

**3. What are JavaScript Data Types?**

Following are the JavaScript Data types:

* Number
* String
* Boolean
* Object
* Undefined

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

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

**5. Between JavaScript and an ASP script, which is faster?**

JavaScript is faster. JavaScript is a client-side language and thus it does not need the assistance of the web server to execute. On the other hand, ASP is a server-side language and hence is always slower than JavaScript. Javascript now is also a server side language (nodejs).

**6. What is negative infinity?**

Negative Infinity is a number in JavaScript which can be derived by dividing negative number by zero.

**7. Is it possible to break JavaScript Code into several lines?**

Breaking within a string statement can be done by the use of a backslash, '\', at the end of the first line

Example:

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

And if you change to a new line when not within a string statement, then javaScript ignores break in line.

Example:

var x=1, y=2,

z=

x+y;

The above code is perfectly fine, though not advisable as it hampers debugging.

**8. Which company developed JavaScript?**

Netscape is the software company who developed JavaScript.

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

Undeclared variables are those that do not exist in a program and are not declared. If the program tries to read the value of an undeclared variable, then a runtime error is encountered.

Undefined variables are those that are declared in the program but have not been given any value. If the program tries to read the value of an undefined variable, an undefined value is returned.

**10. Write the code for adding new elements dynamically?**

<html>

<head>

<title>t1</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("firstP").appendChild(newP); }

</script> </head>

<body> <p id="firstP">firstP<p> </body>

</html>

**11. What are global variables? How are these variable declared and what are the problems associated with using them?**

Global variables are those that are available throughout the length of the code, that is, these have 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.

Example:

// Declare a global globalVariable = "Test";

The problems that are faced by using global variables are the clash of variable names of local and global scope. Also, it is difficult to debug and test the code that relies on global variables.

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

A prompt box is a box which allows the user to enter input by providing a text box. Label and box will be provided to enter the text or number.

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

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

**14. Explain the working of timers in JavaScript? Also elucidate the drawbacks of using the timer, if any?**

Timers are used to execute a piece of code at a set time or also to repeat the code in a given interval of time. This is done by using the functions **setTimeout, setInterval** and **clearInterval**.

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 is used to repeatedly execute the given function in the mentioned delay and only halts when cancelled. The **clearInterval(id)** function instructs the timer to stop.

Timers are operated within a single thread, and thus events might queue up, waiting to be executed.

**15. Which symbol is used for comments in Javascript?**

// for Single line comments and

/\* Multi

Line

Comment

\*/

**16. What is the difference between ViewState and SessionState?**

'ViewState' is specific to a page in a session.

'SessionState' is specific to user specific data that can be accessed across all pages in the web application.

**17. What is === operator?**

=== is called as strict equality operator which returns true when the two operands are having the same value without any type conversion.

**18. Explain how can you submit a form using JavaScript?**

To submit a form using JavaScript use document.form[0].submit();

document.form[0].submit();

**19. Does JavaScript support automatic type conversion?**

Yes JavaScript does support automatic type conversion, it is the common way of type conversion used by JavaScript developers

**20. How can the style/class of an element be changed?**

It can be done in the following way:

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

or

document.getElementById("myText").className = "anyclass";

**21. Explain how to read and write a file using JavaScript?**

There are two ways to read and write a file using JavaScript

* Using JavaScript extensions
* Using a web page and Active X objects

**22. What are all the looping structures in JavaScript?**

Following are looping structures in Javascript:

* For
* While
* do-while loops

**23. What is called Variable typing in Javascript?**

Variable typing is used to assign a number to a variable and the same variable can be assigned to a string.

Example

i = 10;

i = "string";

This is called variable typing.

**24. How can you convert the string of any base to integer in JavaScript?**

The parseInt() function is used to convert numbers between different bases. parseInt() takes the string to be converted as its first parameter, and the second parameter is the base of the given string.

In order to convert 4F (of base 16) to integer, the code used will be -

parseInt ("4F", 16);

**25. Explain the difference between "==" and "==="?**

"==" checks only for equality in value whereas "===" is a stricter equality test and returns false if either the value or the type of the two variables are different.

**26. What would be the result of 3+2+"7"?**

Since 3 and 2 are integers, they will be added numerically. And since 7 is a string, its concatenation will be done. So the result would be 57.

**27. Explain how to detect the operating system on the client machine?**

In order to detect the operating system on the client machine, the navigator.platform string (property) should be used.

**28. What do 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.

**29. What is the function of delete operator?**

The delete keyword is used to delete the property as well as its value.

Example

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

delete student.age;

**30. 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 doesn't exist

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

* Alert
* Confirm and
* Prompt

**32. What is the use of Void(0)?**

Void(0) is used to prevent the page from refreshing and parameter "zero" is passed while calling.

Void(0) is used to call another method without refreshing the page.

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

The following code has to be inserted to achieve the desired effect:

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

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

**34. What is the data type of variables of in JavaScript?**

All variables in the JavaScript are object data types.

**35. 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.

**36. What are escape characters?**

Escape characters (Backslash) is used when working with special characters like single quotes, double quotes, apostrophes and ampersands. Place backslash before the characters to make it display.

Example:

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

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

**37. What are JavaScript Cookies?**

Cookies are the small test files stored in a computer and it gets created when the user visits the websites to store information that they need. Example could be User Name details and shopping cart information from the previous visits.

**38. Explain what is pop()method in JavaScript?**

The pop() method is similar as the shift() method but the difference is that the Shift method works at the start of the array. Also the pop() method take the last element off of the given array and returns it. The array on which is called is then altered.

Example:

var cloths = ["Shirt", "Pant", "TShirt"];

cloths.pop();

//Now cloth becomes Shirt,Pant

**39. Whether JavaScript has concept level scope?**

No. JavaScript does not have concept level scope. The variable declared inside the function has scope inside the function.

**40. Mention what is the disadvantage of using innerHTML in JavaScript?**

If you use innerHTML in JavaScript the disadvantage is

* Content is replaced everywhere
* We cannot use like "appending to innerHTML"
* Even if you use +=like "innerHTML = innerHTML + 'html'" still the old content is replaced by html
* The entire innerHTML content is re-parsed and build into elements, therefore its much slower
* The innerHTML does not provide validation and therefore we can potentially insert valid and broken HTML in the document and break it

**41. What is break and continue statements?**

Break statement exits from the current loop.

Continue statement continues with next statement of the loop.

**42. What are the two basic groups of dataypes in JavaScript?**

They are as –

* Primitive
* Reference types.

Primitive types are number and Boolean data types. Reference types are more complex types like strings and dates.

**43. How generic objects can be created?**

Generic objects can be created as:

var I = new object();

**44. What is the use of type of operator?**

'Typeof' is an operator which is used to return a string description of the type of a variable.

**45. Which keywords are used to handle exceptions?**

Try… Catch---finally is used to handle exceptions in the JavaScript

Try{

Code

}

Catch(exp){

Code to throw an exception

}

Finally{

Code runs either it finishes successfully or after catch

}

**46. Which keyword is used to print the text in the screen?**

document.write("Welcome") is used to print the text – Welcome in the screen.

**47. What is the use of blur function?**

Blur function is used to remove the focus from the specified object.

**48. What is variable typing?**

Variable typing is used to assign a number to a variable and then assign string to the same variable. Example is as follows:

i= 8;

i="john";

**49. How to find operating system in the client machine using JavaScript?**

The '**Navigator.appversion'** is used to find the name of the operating system in the client machine.

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

There are three types of errors:

* **Load time errors**: Errors which come up when loading a web page like improper syntax errors are known as Load time errors and it generates the errors dynamically.
* **Run time 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 which is having different operation.

**51. What is the use of Push method in JavaScript?**

The push method is used to add or append one or more elements to the end of an Array. Using this method, we can append multiple elements by passing multiple arguments

**52. What is unshift method in JavaScript?**

Unshift method is like push method which works at the beginning of the array. This method is used to prepend one or more elements to the beginning of the array.

**53. What is the difference between JavaScript and Jscript?**

Both are almost similar. JavaScript is developed by Netscape and Jscript was developed by Microsoft .

**54. How are object properties assigned?**

Properties are assigned to objects in the following way -

obj["class"] = 12;

or

obj.class = 12;

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

Strict Mode adds certain compulsions to JavaScript. Under the strict mode, JavaScript shows errors for a piece of codes, which did not show an error before, but might be problematic and potentially unsafe. Strict mode also solves some mistakes that hamper the JavaScript engines to work efficiently.

Strict mode can be enabled by adding the string literal "use strict" above the file. This can be illustrated by the given example:

function myfunction() {

"use strict";

var v = "This is a strict mode function";

}

**56. What is the way to get the status of a CheckBox?**

The status can be acquired as follows -

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

If the CheckBox will be checked, this alert will return TRUE.

**57. 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.

**58. Explain window.onload and onDocumentReady?**

The onload function is not run until all the information on the page is loaded. This leads to a substantial delay before any code is executed.

onDocumentReady loads the code just after the DOM is loaded. This allows early manipulation of the code.

**59. How will you explain closures in JavaScript? When are they used?**

Closure is a locally declared variable related to a function which stays in memory 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";

}

// Generate the message

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

// Pass it explicitly to greet

greet(message);

This function can be better represented by using closures

function greeter(name, age) {

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

return function greet() {

console.log(message);

};

}

// Generate the closure

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

// Use the closure

JamesGreeter();

**60. How can a value be appended to an array?**

A value can be appended to an array in the given manner -

arr[arr.length] = value;

**61. Explain the for-in loop?**

The for-in loop is used to loop through the properties of an object.

The syntax for the for-in loop is -

for (variable name in object){

statement or block to execute

}

In each repetition, one property from the object is associated to the variable name, and the loop is continued till all the properties of the object are depleted.

**62. Describe the properties of an anonymous function in JavaScript?**

A function that is declared without any named identifier is known as an anonymous function. In general, an anonymous function is inaccessible after its declaration.

Anonymous function declaration -

var anon = function() {

alert('I am anonymous');

};

anon();

**63. What is the difference between .call() and .apply()?**

The function .call() and .apply() are very similar in their usage except a little difference. .call() is used when the number of the function's arguments are known to the programmer, as they have to be mentioned as arguments in the call statement. On the other hand, .apply() is used when the number is not known. The function .apply() expects the argument to be an array.

The basic difference between .call() and .apply() is in the way arguments are passed to the function. Their usage can be illustrated by the given example.

var someObject = {

myProperty : 'Foo',

myMethod : function(prefix, postfix) {

alert(prefix + this.myProperty + postfix);

}

};

someObject.myMethod('<', '>'); // alerts '<Foo>'

var someOtherObject = {

myProperty : 'Bar'

};

someObject.myMethod.call(someOtherObject, '<', '>'); // alerts '<Bar>'

someObject.myMethod.apply(someOtherObject, ['<', '>']); // alerts '<Bar>'

**64. Define event bubbling?**

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 parent will also work as if it were clicked too.

**65. Is JavaScript case sensitive? Give an example?**

Yes, JavaScript is case sensitive. For example, a function parseInt is not same as the function Parseint.

**66. What boolean operators can be used in JavaScript?**

The 'And' Operator (&&), 'Or' Operator (||) and the 'Not' Operator (!) can be used in JavaScript.

\*Operators are without the parenthesis.

**67. How can a particular frame be targeted, from a hyperlink, in JavaScript?**

This can be done by including the name of the required frame in the hyperlink using the 'target' attribute.

<a href="/newpage.htm" target="newframe">>New Page</a>

**68. What is the role of break and continue statements?**

Break statement is used to come out of the current loop while the continue statement continues the current loop with a new recurrence.

**69. Write the point of difference between web-garden and a web-farm?**

Both web-garden and web-farm are web hosting systems. The only difference is that web-garden is a setup that includes many processors in a single server while web-farm is a larger setup that uses more than one server.

**70. How are object properties assigned?**

Assigning properties to objects is done in the same way as a value is assigned to a variable. For example, a form object's action value is assigned as 'submit' in the following manner - Document.form.action="submit"

**71. What is the method for reading and writing a file in JavaScript?**

This can be done by Using JavaScript extensions (runs from JavaScript Editor), example for opening of a file -

fh = fopen(getScriptPath(), 0);

**72. How are DOM utilized in JavaScript?**

DOM stands for Document Object Model and is responsible for how various objects in a document interact with each other. DOM is required for developing web pages, which includes objects like paragraph, links, etc. These objects can be operated to include actions like add or delete. DOM is also required to add extra capabilities to a web page. On top of that, the use of API gives an advantage over other existing models.

**73. How are event handlers utilized in JavaScript?**

Events are the actions that result from activities, such as clicking a link or filling a form, by the user. An event handler is required to manage proper execution of all these events. Event handlers are an extra attribute of the object. This attribute includes event's name and the action taken if the event takes place.

**74. Explain the role of deferred scripts in JavaScript?**

By default, the parsing of the HTML code, during page loading, is paused until the script has not stopped executing. It means, if the server is slow or the script is particularly heavy, then the webpage is displayed with a delay. While using Deferred, scripts delays execution of the script till the time HTML parser is running. This reduces the loading time of web pages and they get displayed faster.

**75. What are the various functional components in JavaScript?**

The different functional components in JavaScript are-

**First-class functions:** Functions in JavaScript are utilized as first class objects. This usually means that these functions can be passed as arguments to other functions, returned as values from other functions, assigned to variables or can also be stored in data structures.

**Nested functions:** The functions, which are defined inside other functions, are called Nested functions. They are called 'everytime' the main function is invoked.

**76. Write about the errors shown in JavaScript?**

JavaScript gives a message if it encounters an error. The recognized errors are -

* Load-time errors: The errors shown at the time of the page loading are counted under Load-time errors. These errors are encountered by the use of improper syntax, and thus are detected while the page is getting loaded.
* Run-time errors: This is the error that comes up while the program is running. It is caused by illegal operations, for example, division of a number by zero, or trying to access a non-existent area of the memory.
* Logic errors: It is caused by the use of syntactically correct code, which does not fulfill the required task. For example, an infinite loop.

**77. What are Screen objects?**

Screen objects are used to read the information from the client's screen. The properties of screen objects are -

* AvailHeight: Gives the height of client's screen
* AvailWidth: Gives the width of client's screen.
* ColorDepth: Gives the bit depth of images on the client's screen
* Height: Gives the total height of the client's screen, including the taskbar
* Width: Gives the total width of the client's screen, including the taskbar

**78. Explain the unshift() method ?**

This method is functional at the starting of the array, unlike the push(). It adds the desired number of elements to the top of an array. For example -

var name = [ "john" ];

name.unshift( "charlie" );

name.unshift( "joseph", "Jane" );

console.log(name);

The output is shown below:

[" joseph "," Jane ", " charlie ", " john "]

**79. Define unescape() and escape() functions?**

The escape () function is responsible for coding a string so as to make the transfer of the information from one computer to the other, across a network.

For Example:

<script>

document.write(escape("Hello? How are you!"));

</script>

Output: Hello%3F%20How%20are%20you%21

The unescape() function is very important as it decodes the coded string.

It works in the following way. For example:

<script>

document.write(unescape("Hello%3F%20How%20are%20you%21"));

</script>

Output: Hello? How are you!

**80. What are the decodeURI() and encodeURI()?**

EncodeURl() is used to convert URL into their hex coding. And DecodeURI() is used to convert the encoded URL back to normal.

<script>

var uri="my test.asp?name=ståle&car=saab";

document.write(encodeURI(uri)+ "<br>");

document.write(decodeURI(uri));

</script>

Output -

my%20test.asp?name=st%C3%A5le&car=saab

my test.asp?name=ståle&car=saab

**81. Why it is not advised to use innerHTML in JavaScript?**

innerHTML content is refreshed every time and thus is slower. There is no scope for validation in innerHTML and, therefore, it is easier to insert rouge code in the document and, thus, make the web page unstable.

**82. What does the following statement declares?**

var myArray = [[[]]];

It declares a three dimensional array.

**83. How are JavaScript and ECMA Script related?**

ECMA Script are like rules and guideline while Javascript is a scripting language used for web development.

**84. What is namespacing in JavaScript and how is it used?**

Namespacing is used for grouping the desired functions, variables etc. under a unique name. It is a name that has been attached to the desired functions, objects and properties. This improves modularity in the coding and enables code reuse.

**85. How can JavaScript codes be hidden from old browsers that don't support JavaScript?**

For hiding JavaScript codes from old browsers:

Add "<!--" without the quotes in the code just after the <script> tag.

Add "//-->" without the quotes in the code just before the <script> tag.

Old browsers will now treat this JavaScript code as a long HTML comment. While, a browser that supports JavaScript, will take the "<!--" and "//-->" as one-line comments.

What is JavaScript?

JavaScript is a lightweight, interpreted programming language with object-oriented capabilities that allows you to build interactivity into otherwise static HTML pages.

The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

Name some of the JavaScript features.

Following are the features of JavaScript −

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

What are the advantages of using JavaScript?

Following are the advantages of using JavaScript −

* **Less server interaction −** You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
* **Immediate feedback to the visitors −** They don't have to wait for a page reload to see if they have forgotten to enter something.
* **Increased interactivity −** You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
* **Richer interfaces −** You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

What are disadvantages of using JavaScript?

We can not treat JavaScript as a full fledged programming language. It lacks the following important features −

* Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
* JavaScript can not be used for Networking applications because there is no such support available.
* JavaScript doesn't have any multithreading or multiprocess capabilities.

Is JavaScript a case-sensitive language?

Yes! JavaScript is a case-sensitive language. This means that language keywords, variables, function names, and any other identifiers must always be typed with a consistent capitalization of letters.

How can you create an Object in JavaScript?

JavaScript supports Object concept very well. You can create an object using the object literal as follows −

var emp = {

name: "Zara",

age: 10

};

How can you read properties of an Object in JavaScript?

You can write and read properties of an object using the dot notation as follows −

// Getting object properties

emp.name // ==> Zara

emp.age // ==> 10

// Setting object properties

emp.name = "Daisy" // <== Daisy

emp.age = 20 // <== 20

How can you create an Array in JavaScript?

You can define arrays using the array literal as follows −

var x = [];

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

How to read elements of an array in JavaScript?

An array has a length property that is useful for iteration. We can read elements of an array as follows −

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

for (var i = 0; i < x.length; i++) {

// Do something with x[i]

}

What is a named function in JavaScript? How to define a named function?

A named function has a name when it is defined. A named function can be defined using function keyword as follows −

function named(){

// do some stuff here

}

How many types of functions JavaScript supports?

A function in JavaScript can be either named or anonymous.

How to define a anonymous function?

An anonymous function can be defined in similar way as a normal function but it would not have any name.

Can you assign a anonymous function to a variable?

Yes! An anonymous function can be assigned to a variable.

Can you pass a anonymous function as an argument to another function?

Yes! An anonymous function can be passed as an argument to another function.

What is arguments object in JavaScript?

JavaScript variable arguments represents the arguments passed to a function.

How can you get the type of arguments passed to a function?

Using typeof operator, we can get the type of arguments passed to a function. For example −

function func(x){

console.log(typeof x, arguments.length);

}

func(); //==> "undefined", 0

func(1); //==> "number", 1

func("1", "2", "3"); //==> "string", 3

How can you get the total number of arguments passed to a function?

Using arguments.length property, we can get the total number of arguments passed to a function. For example −

function func(x){

console.log(typeof x, arguments.length);

}

func(); //==> "undefined", 0

func(1); //==> "number", 1

func("1", "2", "3"); //==> "string", 3

How can you get the reference of a caller function inside a function?

The arguments object has a callee property, which refers to the function you're inside of. For example −

function func() {

return arguments.callee;

}

func(); // ==> func

What is the purpose of 'this' operator in JavaScript?

JavaScript famous keyword this always refers to the current context.

What are the valid scopes of a variable in JavaScript?

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

* **Global Variables −** A global variable has global scope which means it is visible everywhere in your JavaScript code.
* **Local Variables −** A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

Which type of variable among global and local, takes precedence over other if names are same?

A local variable takes precedence over a global variable with the same name.

What is callback?

A callback is a plain JavaScript function passed to some method as an argument or option. Some callbacks are just events, called to give the user a chance to react when a certain state is triggered.

What is closure?

Closures are created whenever a variable that is defined outside the current scope is accessed from within some inner scope.

Give an example of closure?

Following example shows how the variable counter is visible within the create, increment, and print functions, but not outside of them −

function create() {

var counter = 0;

return {

increment: function() {

counter++;

},

print: function() {

console.log(counter);

}

}

}

var c = create();

c.increment();

c.print(); // ==> 1

Which built-in method returns the character at the specified index?

charAt() method returns the character at the specified index.

Which built-in method combines the text of two strings and returns a new string?

concat() method returns the character at the specified index.

Which built-in method calls a function for each element in the array?

forEach() method calls a function for each element in the array.

Which built-in method returns the index within the calling String object of the first occurrence of the specified value?

indexOf() method returns the index within the calling String object of the first occurrence of the specified value, or −1 if not found.

Which built-in method returns the length of the string?

length() method returns the length of the string.

Which built-in method removes the last element from an array and returns that element?

pop() method removes the last element from an array and returns that element.

Which built-in method adds one or more elements to the end of an array and returns the new length of the array?

push() method adds one or more elements to the end of an array and returns the new length of the array.

Which built-in method reverses the order of the elements of an array?

reverse() method reverses the order of the elements of an array −− the first becomes the last, and the last becomes the first.

Which built-in method sorts the elements of an array?

sort() method sorts the elements of an array.

Which built-in method returns the characters in a string beginning at the specified location?

substr() method returns the characters in a string beginning at the specified location through the specified number of characters.

Which built-in method returns the calling string value converted to lower case?

toLowerCase() method returns the calling string value converted to lower case.

Which built-in method returns the calling string value converted to upper case?

toUpperCase() method returns the calling string value converted to upper case.

Which built-in method returns the string representation of the number's value?

toString() method returns the string representation of the number's value.

What are the variable naming conventions in JavaScript?

While naming your variables in JavaScript keep following rules in mind.

You should not use any of the JavaScript reserved keyword as variable name. These keywords are mentioned in the next section. For example, break or boolean variable names are not valid.

JavaScript variable names should not start with a numeral (0-9). They must begin with a letter or the underscore character. For example, 123test is an invalid variable name but \_123test is a valid one.

JavaScript variable names are case sensitive. For example, Name and name are two different variables.

How typeof operator works?

The typeof is a unary operator that is placed before its single operand, which can be of any type. Its value is a string indicating the data type of the operand.

The typeof operator evaluates to "number", "string", or "boolean" if its operand is a number, string, or boolean value and returns true or false based on the evaluation.

What typeof returns for a null value?

It returns "object".

Can you access Cookie using javascript?

JavaScript can also manipulate cookies using the cookie property of the Document object. JavaScript can read, create, modify, and delete the cookie or cookies that apply to the current web page.

How to create a Cookie using JavaScript?

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

Syntax −

document.cookie = "key1 = value1; key2 = value2; expires = date";

Here expires attribute is option. If you provide this attribute with a valid date or time then cookie will expire at the given date or time and after that cookies' value will not be accessible.

How to read a Cookie using JavaScript?

Reading a cookie is just as simple as writing one, because the value of the document.cookie object is the cookie. So you can use this string whenever you want to access the cookie.

The document.cookie string will keep a list of name = value pairs separated by semicolons, where name is the name of a cookie and value is its string value.

You can use strings' split() function to break the string into key and values.

How to delete a Cookie using JavaScript?

Sometimes you will want to delete a cookie so that subsequent attempts to read the cookie return nothing. To do this, you just need to set the expiration date to a time in the past.

How to redirect a url using JavaScript?

his is very simple to do a page redirect using JavaScript at client side. To redirect your site visitors to a new page, you just need to add a line in your head section as follows −

<head>

<script type="text/javascript">

<!--

window.location="http://www.newlocation.com";

//-->

</script>

</head>

How to print a web page using javascript?

JavaScript helps you to implement this functionality using print function of window object. The JavaScript print function window.print() will print the current web page when executed.

What is Date object in JavaScript?

The Date object is a datatype built into the JavaScript language. Date objects are created with the new Date( ).

Once a Date object is created, a number of methods allow you to operate on it. Most methods simply allow you to get and set the year, month, day, hour, minute, second, and millisecond fields of the object, using either local time or UTC (universal, or GMT) time.

What is Number object in JavaScript?

he Number object represents numerical date, either integers or floating-point numbers. In general, you do not need to worry about Number objects because the browser automatically converts number literals to instances of the number class.

Syntax −

Creating a number object −

var val = new Number(number);

If the argument cannot be converted into a number, it returns NaN (Not-a-Number).

How to handle exceptions in JavaScript?

The latest versions of JavaScript added exception handling capabilities. JavaScript implements the try...catch...finally construct as well as the throw operator to handle exceptions.

You can catch programmer-generated and runtime exceptions, but you cannot catch JavaScript syntax errors.

What is purpose of onError event handler in JavaScript?

The onerror event handler was the first feature to facilitate error handling for JavaScript. The error event is fired on the window object whenever an exception occurs on the page.

The onerror event handler provides three pieces of information to identify the exact nature of the error −

* **Error message −** The same message that the browser would display for the given error.
* **URL −** The file in which the error occurred.
* **Line number −** The line number in the given URL that caused the error.

## What is Next ?

Further you can go through your past assignments you have done with the subject and make sure you are able to speak confidently on them. If you are fresher then interviewer does not expect you will answer very complex questions, rather you have to make your basics concepts very strong.

Second it really doesn't matter much if you could not answer few questions but it matters that whatever you answered, you must have answered with confidence. So just feel confident during your interview. We at tutorialspoint wish you best luck to have a good interviewer and all the very best for your future endeavor. Cheers :-)

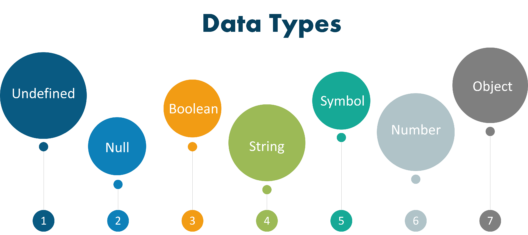
### ****1. What is the difference between Java & JavaScript?****

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

### ****Q2. What is JavaScript?****

[JavaScript](https://www.edureka.co/blog/what-is-javascript/) is a **lightweight**, **interpreted** programming language with object-oriented capabilities that allows you to build interactivity into otherwise static HTML pages. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

### ****Q3. What are the data types supported by JavaScript?****



The **data types** supported by JavaScript are:

* Undefined
* Null
* Boolean
* String
* Symbol
* Number
* Object

### ****Q4. What are the features of JavaScript?****

Following are the **features** of JavaScript:

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

### ****Q5. Is JavaScript a case-sensitive language?****

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

### ****Q6. What are the advantages of JavaScript?****

Following are the **advantages** of using JavaScript −

* **Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
* **Immediate feedback to the visitors** − They don’t have to wait for a page reload to see if they have forgotten to enter something.
* **Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
* **Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

### ****Q7. How can you create an object in JavaScript?****

JavaScript supports **Object** concept very well. You can create an object using the **object literal** as follows −

|  |  |
| --- | --- |
| 1  2  3  4 | var emp = {  name: "Daniel",  age: 23  }; |

### ****Q8. How can you create an Array in JavaScript?****

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

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

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

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

|  |  |
| --- | --- |
| 1  2  3 | function named(){  // write code here  } |

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

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

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

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | function func(x){  console.log(typeof x, arguments.length);  }  func(); //==> "undefined", 0  func(7); //==> "number", 7  func("1", "2", "3"); //==> "string", 3 |

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

The scope of a variable is the **region** of your program in which it is **defined**. JavaScript variable will have only two scopes.  
• **Global Variables** − A global variable has global scope which means it is visible everywhere in your JavaScript code.  
• **Local Variables** − A local variable will be visible only within a function where it is defined. Function parameters are always local to that function.

### ****Q13. What is the purpose of ‘This’ operator in JavaScript?****

The JavaScript **this** keyword refers to the object it belongs to. This has different values depending on where it is used. In a method, this refers to the owner object and in a function, this refers to the global object.

[Powered by Edureka [](https://www.edureka.co/mymock-interview-service)](https://www.edureka.co/mymock-interview-service" \t "_blank)

### [80% Interview rejections happen in first 90 seconds](https://www.edureka.co/mymock-interview-service" \t "_blank)

[Take Javascript Mock Interview](https://www.edureka.co/mymock-interview-service" \t "_blank)

* [Get Interviewed by Industry Experts](https://www.edureka.co/mymock-interview-service" \t "_blank)
* [Personalized interview feedback](https://www.edureka.co/mymock-interview-service" \t "_blank)

### ****Q14. What is Callback?****

A **callback** is a plain JavaScript function passed to some method as an argument or option. It is a function that is to be **executed** after another function has finished executing, hence the name ‘**call back**‘. In JavaScript, functions are objects. Because of this, functions can take functions as arguments, and can be returned by other functions.

### ****Q15. What is Closure? Give an example.****

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

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

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

### ****Q17. What are the variable naming conventions in JavaScript?****

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

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

### ****Q18. How does TypeOf Operator work?****

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

### ****Q19. How to create a cookie using JavaScript?****

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

**Syntax :**

|  |  |
| --- | --- |
| 1 | document.cookie = "key1 = value1; key2 = value2; expires = date"; |

### ****Q20. How to read a cookie using JavaScript?****

Reading a cookie is just as simple as writing one, because the value of the document.cookie object is the cookie. So you can use this string whenever you want to access the cookie.

* The **document.cookie** string will keep a list of name = value pairs separated by semicolons, where name is the name of a cookie and value is its string value.
* You can use strings’ **split()** function to break the string into key and values.

### ****Q21. How to delete a cookie using JavaScript?****

If you want to delete a cookie so that subsequent attempts to read the cookie return nothing, you just need to set the expiration date to a time in the past. You should define the cookie path to ensure that you delete the right cookie. Some browsers will not let you delete a cookie if you don’t specify the path.

Now let’s move on to the next section of JavaScript interview questions.

## ****Intermediate Level JavaScript Interview Questions****

### ****Q22. What is the difference between Attributes and Property?****

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

**Property-**  is the value assigned to the property like type=”text”, value=’Name’ etc.

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

Here are the list of ways an HTML element can be accessed in a Javascript code:  
(i) **getElementById(‘idname’):** Gets an element by its ID name  
(ii) **getElementsByClass(‘classname’):** Gets all the elements that have the given classname.  
(iii) **getElementsByTagName(‘tagname’):** Gets all the elements that have the given tag name.  
(iv) **querySelector():** This function takes css style selector and returns the first selected element.

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

There are 3 different ways in which a JavaScript code can be involved in an HTML file:

* **Inline**
* **Internal**
* **External**

An **inline** function is a JavaScript function, which is assigned to a variable created at runtime. You can differentiate between Inline Functions and Anonymous since an inline function is assigned to a variable and can be easily reused. When you need a JavaScript for a function, you can either have the script **integrated** in the page you are working on, or you can have it placed in a **separate** file that you call, when needed. This is the difference between an **internal** script and an **external** script.

### ****Q25. What are the ways to define a variable in JavaScript?****

The three possible ways of defining a variable in JavaScript are:

* **Var** – The JavaScript variables statement is used to declare a variable and, optionally, we can initialize the value of that variable. Example: var a =10; Variable declarations are processed before the execution of the code.
* **Const** – The idea of const functions is not allow them to modify the object on which they are called. When a function is declared as const, it can be called on any type of object.
* **Let** – It is a signal that the variable may be reassigned, such as a counter in a loop, or a value swap in an algorithm. It also signals that the variable will be used only in the block it’s defined in.

### ****Q26. What is a Typed language?****

Typed Language is in which the values are associated with **values** and not with **variables**. It is of two types:

* **Dynamically:** in this, the variable can hold multiple types; like in JS a variable can take number, chars.
* **Statically:** in this, the variable can hold only one type, like in Java a variable declared of string can take only set of characters and nothing else.

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

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

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

### ****Q28. What is the difference between the operators ‘==‘ & ‘===‘?****

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

### ****Q29. What is the difference between null & undefined?****

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

### ****Q30. What is the difference between undeclared & undefined?****

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

### ****Q31. Name some of the JavaScript Frameworks****

A [JavaScript framework](https://www.edureka.co/blog/top-10-javascript-frameworks/) is an application framework written in JavaScript. It differs from a JavaScript library in its control flow. There are many JavaScript Frameworks available but some of the most commonly used frameworks are:

* [Angular](https://www.edureka.co/angular-training)
* [React](https://www.edureka.co/blog/reactjs-tutorial)
* Vue

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

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

### ****Q33. What is the difference between innerHTML & innerText?****

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

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

### ****Q34. What is an event bubbling in JavaScript?****

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

[Powered by Edureka](https://www.edureka.co/mymock-interview-service" \t "_blank)

### [Javascript Mock interviews for you](https://www.edureka.co/mymock-interview-service" \t "_blank)

* [Interviews by Industry Experts](https://www.edureka.co/mymock-interview-service" \t "_blank)
* [Personalized detailed interview feedback](https://www.edureka.co/mymock-interview-service" \t "_blank)
* [Access to exclusive and curated content](https://www.edureka.co/mymock-interview-service" \t "_blank)

### ****Q35. What is NaN in JavaScript?****

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

### ****Q36. How do JavaScript primitive/object types passed in functions?****

One of the differences between the two is that Primitive Data Types are passed By Value and Objects are passed By Reference.

* **By Value** means creating a COPY of the original. Picture it like twins: they are born exactly the same, but the first twin doesn’t lose a leg when the second twin loses his in the war.
* **By Reference** means creating an ALIAS to the original. When your Mom calls you “Pumpkin Pie” although your name is Margaret, this doesn’t suddenly give birth to a clone of yourself: you are still one, but you can be called by these two very different names.

### ****Q37. How can you convert the string of any base to integer in JavaScript?****

The **parseInt()** function is used to convert numbers between different bases. It takes the string to be converted as its first parameter, and the second parameter is the base of the given string.

For example-

|  |  |
| --- | --- |
| 1 | parseInt("4F", 16) |

### ****Q38. What would be the result of 2+5+”3″?****

Since 2 and 5 are integers, they will be added numerically. And since 3 is a string, its concatenation will be done. So the result would be 73. The ” ” makes all the difference here and represents 3 as a string and not a number.

### ****Q39. What are Exports & Imports?****

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | //------ lib.js ------</span>  export const sqrt = Math.sqrt;</span>  export function square(x) {</span>  return x \* x;</span>  }  export function diag(x, y) {  return sqrt(square(x) + square(y));  }    //------ main.js ------</span>   { square, diag } from 'lib';  console.log(square(5)); // 25  console.log(diag(4, 3)); // 5 |

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

## ****Advanced Level JavaScript Interview Questions****

### ****Q40. What is the ‘Strict’ mode in JavaScript and how can it be enabled?****

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

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

### ****Q41. What is a prompt box in JavaScript?****

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

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var Y = 1;  if (function F(){})  {  y += Typeof F;</span>  }  console.log(y); |

The output would be 1undefined. The if condition statement evaluates using eval, so eval(function f(){}) returns function f(){} (which is true). Therefore, inside the if statement, executing typeof f returns undefined because the if statement code executes at run time, and the statement inside the if condition is evaluated during run time.

### ****Q43. What is the difference between Call & Apply?****

The **call()** method calls a function with a given this value and arguments provided individually.

**Syntax-**

|  |  |
| --- | --- |
| 1 | fun.call(thisArg[, arg1[, arg2[, ...]]]) |

The **apply()** method calls a function with a given this value, and arguments provided as an array.

**Syntax-**

|  |  |
| --- | --- |
| 1 | fun.apply(thisArg, [argsArray]) |

### ****Q44. How to empty an Array in JavaScript?****

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

**Method 1 –**

|  |  |
| --- | --- |
| 1 | arrayList = [] |

Above code will set the variable arrayList to a new empty array. This is recommended if you don’t have references to the original array arrayList anywhere else, because it will actually create a new, empty array. You should be careful with this method of emptying the array, because if you have referenced this array from another variable, then the original reference array will remain unchanged.

**Method 2 –**

|  |  |
| --- | --- |
| 1 | arrayList.length = 0; |

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

**Method 3 –**

|  |  |
| --- | --- |
| 1 | arrayList.splice(0, arrayList.length); |

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

[Powered by Edureka](https://www.edureka.co/mymock-interview-service" \t "_blank)

### [Need help for your upcoming interview?](https://www.edureka.co/mymock-interview-service" \t "_blank)

[Take Javascript Mock Interview](https://www.edureka.co/mymock-interview-service" \t "_blank)

* [Get Interviewed by Industry Experts](https://www.edureka.co/mymock-interview-service" \t "_blank)
* [Personalized interview feedback](https://www.edureka.co/mymock-interview-service" \t "_blank)

**Method 4 –**

|  |  |
| --- | --- |
| 1  2  3  4 | while(arrayList.length)  {  arrayList.pop();  } |

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

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | var Output = (function(x)  {  Delete X;  return X;  }  )(0);  console.log(output); |

The output would be 0. The delete operator is used to delete properties from an object. Here x is not an object but a local variable. delete operators don’t affect local variables.

### ****Q46. What will be the output of the following code?****

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | var X = { Foo : 1};  var Output = (function()  {  delete X.foo;  return X.foo;  }  )();  console.log(output); |

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

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | var Employee =  {  company: 'xyz'  }  var Emp1 = Object.create(employee);  delete Emp1.company Console.log(emp1.company); |

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

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

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | //nfe (named function expression)  var Foo = Function Bar()  {  return 7;  };  typeof Bar(); |

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

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

This is an increasingly common practice, employed by many popular JavaScript libraries. This technique creates a closure around the entire contents of the file which, perhaps most importantly, creates a private namespace and thereby helps avoid potential name clashes between different JavaScript modules and libraries.  
Another feature of this technique is to allow for an easy alias for a global variable. This is often used in jQuery plugins.

### ****Q50. What are escape characters in JavaScript?****

JavaScript escape characters enable you to write special characters without breaking your application. Escape characters (Backslash) is used when working with special characters like single quotes, double quotes, apostrophes and ampersands. Place backslash before the characters to make it display.

**For example-**

|  |  |
| --- | --- |
| 1  2 | document.write "I am a "good" boy"  document.write "I am a "good" boy" |

With this, we have come to the end of JavaScript interview questions blog. I Hope these **JavaScript Interview Questions** will help you in your interviews. In case you have attended any JavaScript interview in the recent past, do paste those interview questions in the comments section and we’ll answer them. You can also comment below if you have any questions in your mind, which you might face in your JavaScript interview.

If you wish to learn JavaScript and build your own applications, then check out our [***Full Stack Web Developer Masters Program***](https://www.edureka.co/masters-program/full-stack-developer-training)which comes with instructor-led live training and real-life project experience. This training makes you proficient in skills to work with back-end and front-end web technologies. It includes training on Web Development, jQuery, Angular, NodeJS, ExpressJS and MongoDB.

Got a question for us? Please mention it in the comments section of “JavaScript Interview Questions” blog and we will get back to you.

### uestion 1. Explain what is Javascript? List some data types supported by Javascript?

#### Javascript

Javascript is an object-oriented computer programming language commonly used to create interactive effects within web browsers.It is first used by the Netscape browser, that provides access to the HTML document object model (DOM), provides access to the browser object model (BOM). Javascript syntax looks a lot like java, c or c++ syntax.

Below is the list of data types supported by Javascript:-

* Undefined
* Null
* Boolean
* String
* Symbol
* Number
* Object

### Question 2. What close() does in Javascript?

In Javascript close() method is used to close the current window. You must write window.close() to ensure that this command is associated with a window object and not some other JavaScript object.

### Question 3. What is the difference between let and var?

Both var and let are used for variable/ method declaration in javascript but the main difference between let and var is that **var** is function scoped whereas **let** is block scoped.

### Question 4. Explain Closures in JavaScript?

Closures are the combination of lexical environment and function within which the function was declared. This allows JavaScript programmers to write better, more creative, concise and expressive codes. The closure will consist of all the local variables that were in-scope when the closure was created.

Sure, closures appear to be complex and beyond the scope, but after you read this article, closures will be much more easy to understand and more simple for your everyday [JavaScript](https://www.onlineinterviewquestions.com/advanced-javascript-interview-questions/) programming tasks. JavaScript is  a very function-oriented language it gives the user freedom to use functions as the wish of the programmer.

### Question 5. Explain JavaScript Event Delegation Model?

In JavaScript, there is some cool stuff that makes it the best of all. One of them is Delegation Model. When capturing and bubbling, allow functions to implement one single handler to many elements at one particular time then that is called event delegation. Event delegation allows you to add event listeners to one parent instead of specified nodes. That particular listener analyzes bubbled events to find a match on the child elements. Many people think it to be complicated but in reality, it is very simple if one starts understanding it.

### Question 6. Describe negative infinity in JavaScript?

NEGATIVE\_INFINITY property represents negative infinity and is a number in javascript, which is derived by ‘dividing negative number by zero’. It can be better understood as a number that is lower than any other number. Its properties are as follows:  
– A number of objects need not to be created to access this static property.  
– The value of negative infinity is the same as the negative value of the infinity property of the global object.

The values behave differently than the mathematical infinity:

1. Any positive value, including POSITIVE\_INFINITY, multiplied by NEGATIVE\_INFINITY is NEGATIVE\_INFINITY.
2. Any negative value, including NEGATIVE\_INFINITY, multiplied by NEGATIVE\_INFINITY is POSITIVE\_INFINITY.
3. Zero multiplied by NEGATIVE\_INFINITY is NaN.
4. NaN multiplied by NEGATIVE\_INFINITY is NaN.
5. NEGATIVE\_INFINITY, divided by any negative value except NEGATIVE\_INFINITY, is POSITIVE\_INFINITY.
6. NEGATIVE\_INFINITY, divided by any positive value except POSITIVE\_INFINITY, is NEGATIVE\_INFINITY.
7. NEGATIVE\_INFINITY, divided by either NEGATIVE\_INFINITY or POSITIVE\_INFINITY, is NaN.
8. Any number divided by NEGATIVE\_INFINITY is zero.

### Question 7. Explain function hoisting in JavaScript?

JavaScript’s default behavior that allows moving declarations to the top is called Hoisting. The 2 ways of creating functions in JavaScript are **Function Declaration** and **Function Expression**. Let’s find out more about these:

#### Function Declaration

A function with the specific parameters is known as function declarations. To create a variable in JavaScript is called declarations.

**e.g:**

hoisted(); // logs "foo"

function hoisted() {

 console.log('foo');

}

### Question 8. What is the use of let & const in JavaScript?

In modern javascript let & const are different ways of creating variables. Earlier in javascript, we use the var keyword for creating variables. let & const keyword is introduced in version [ES6](https://www.onlineinterviewquestions.com/es6-interview-questions/) with the vision of creating two different types of variables in javascript one is immutable and other is mutable.  
**const:** It is used to create an immutable variable. Immutable variables are variables whose value is never changed in the complete life cycle of the program.  
**let:** let is used to create a mutable variable. Mutable variables are normal variables like var that can be changed any number of time.

### Question 9. Explain Arrow functions?

An arrow function is a consise and short way to write function expressions in Es6 or above.A rrow functions cannot be used as constructors and also does not supports this, arguments, super, or new.target keywords. It is best suited for non-method functions. In general an arrow function looks like const function\_name= ()=>{}

const greet=()=>{console.log('hello');}

greet();

### Question 11. What is difference between module.exports and export?

The module is a plain JavaScript object with an exports property. Exports is a plain JavaScript variable that happens to be set to module.exports. At the end of your file, node.js will basically ‘return’ module.exports to the require function. A simplified way to view a JS file in Node could be this:

var module = { exports: {} };

var exports = module.exports;

// your code

return module.exports;

If you set a property on exports, like exports.a = 9;, that will set module.exports.a as well because objects are passed around as references in JavaScript, which means that if you set multiple variables to the same object, they are all the same object; so then exports and module.exports are the same objects.  
But if you set exports to something new, it will no longer be set to module.exports, so exports and module.exports are no longer the same objects.

Source : https://stackoverflow.com/questions/16383795/difference-between-module-exports-and-exports-in-the-commonjs-module-system

### Question 12. How to import all exports of a file as an object.

import \* as object name from ‘./file.js’ is used to import all exported members as an object. You can simply access the exported variables or methods using dot (.) operator of the object.

Example:

objectname.member1;

objectname.member2;

objectname.memberfunc();

### Question 13. Explain “use strict” ?

“use strict” is a javascript directive that is introduced in Es5. The purpose of using “use strict” directive is to enforce the code is executed in strict mode. In strict mode we can’t use a variable without declaring it. “use strict” is ignored by earlier versions of Javascript.

### Question 14. Explain Event bubbling and Event Capturing in JavaScript?

**Event Capture and Bubbling**: In HTML DOM API there are two ways of event propagation and determines the order in which event will be received. The two ways are Event Bubbling and Event Capturing. The first method event bubbling directs the event to its intended target, and the second is called event capture in which the event goes down to the element.

### ****Event Capture****

The capture procedure is rarely used but when it’s used it proves to be very helpful. This process is also called ‘trickling’. In this process, the event is captured first by the outermost element and then propagated to the innermost element. For example:

From the above example, suppose the click event did occur in the ‘li’ element, in that case capturing event it will be first handled ‘div’, then ‘ul’ and at last the target element will be hit that is ‘li’

### ****Event Bubbling****

Bubbling just works like the bubbles, the event gets handled by the innermost element and then propagated to the outer element.

From the above example, suppose the click event did occur in the ‘li’ element in bubbling model the event will be handled first by ‘li’ then by ‘ul’ and at last by ‘div’ element.

### Question 15. In Javascript are calculations with fractional numbers guaranteed to be precise?

NO, calculations with fractional numbers are not guaranteed to be precise in Javascript

### Question 16. List the comparison operators supported by Javascript?

Javascript supports below comparison operators

* > Greater than
* < Less than
* <= Less than or equal to
* >= Greater than or equal to
* == Equal to
* != Not Equal to
* === Equal to with datatype check
* !== Not equal to with datatype check

### Question 17. How do you declare variables in Javascript?

In Javascript variable are declared using the var keyword.A variable must begin with A **letter**, **$** or \_.

**eg.** var myVar=”Online Interview Questions”;

**PS:** All variables in Javascript are Case sensitive.

Also, read [**Advanced JavaScript Interview Questions**](https://www.onlineinterviewquestions.com/advanced-javascript-interview-questions/)

### Question 18. What will happen if an infinite while loop is run in Javascript?

The program will crash the browser.

### Question 19. List HTML DOM mouse events?

HTML DOM mouse events

* onclick
* ondblclick
* mousemove
* mousedown
* mouseover
* mouseout
* mouseup

### Question 20. How to get the last index of a string in Javascript?

**string.length-1** is used to get the last index of a string in Javascript

**Example Usage:-**

var myString="JavascriptQuestions";

console.log(myString.length-1);

### Question 21. How to get the primitive value of a string in Javascript?

In Javascript **valueOf()** method is used to get the primitive value of a string.

**Example Usage:**

var myVar= "Hi!"

console.log(myVar.valueOf())

### Question 22. What are the primitive data types in JavaScript?

A primitive is a basic data type that’s not built out of other data types. It can only represent one single value. All primitives are built-in data types by necessity, (the compiler has to know about them,) but not all built-in data types are primitives.

In JavaScript there are 5 primitive data types are available they are **undefined**, **null**, **boolean**, **string** and **number** are available.Everything else in Javascript is an object.

### Question 23. What does the instanceof operator do?

In Javascript **instanceof** operator checks whether the object is an instance of a class or not:

**Example Usage**

Square.prototype = new Square();

console.log(sq instanceof Square); // true

### Question 24. What is Javascript BOM?

BOM stands for “Browser Object Modal” that allows Javascript to ‘talk’ to the browser, no standards, modern browsers implement similar BOMS – window, screen, location, history, navigator, timing, cookies.

### Question 25. What are different types of Popup boxes available in Javascript?

In Javascript there are 3 types of Popup Boxes are available, they are

* Alert
* Confirm
* Prompt

Read [**80+ Best Angular Js Interview Questions**](https://www.onlineinterviewquestions.com/angular-js-interview-questions/)

### Question 26. How can you create an array in Javascript?

There are 3 different ways to create an array in Javascript. They are

* By array literal  
  **usage:**

var myArray=[value1,value2...valueN];

* By creating instance of Array  
  **usage:**

var myArray=new Array();

* By using an Array constructor  
  **usage:**

var myArray=new Array('value1','value2',...,'valueN');

### Question 27. What is the ‘Strict’ mode in JavaScript and how can it be enabled?

**Strict mode** is a way to introduce better error-checking into your code. When you use strict mode, you cannot, for example, use implicitly declared variables, or assign a value to a read-only property, or add a property to an object that is not extensible.

You can enable strict mode by adding **“use strict”**; at the beginning of a file, a program, or a function. This kind of declaration is known as a directive prologue. The scope of a strict mode declaration depends on its context. If it is declared in a global context (outside the scope of a function), all the code in the program is in strict mode. If it is declared in a function, all the code in the function is in strict mode.

### Question 28. How to calculate Fibonacci numbers in JavaScript?

#### Calculating Fibonacci series in JavaScript

Fibonacci numbers are a sequence of numbers where each value is the sum of the previous two, starting with 0 and 1. The first few values are 0, 1, 1, 2, 3, 5, 8, 13 ,…,

function fib(n) {

var a=0, b=1;

for (var i=0; i < n; i++) {

var temp = a+b;

a = b;

b = temp;

}

return a;

}

### Question 29. What is the difference between the substr() and substring() functions in JavaScript?

#### Difference between the substr() and substring() functions in JavaScript.

The substr() function has the form substr(startIndex,length). It returns the substring from startIndex and returns ‘length’ number of characters.

var s = "hello";

( s.substr(1,4) == "ello" ) // true

The substring() function has the form substring(startIndex,endIndex). It returns the substring from startIndex up to endIndex – 1.

var s = "hello";

( s.substring(1,4) == "ell" ) // true

### Question 30. What are different types of Inheritence? Which Inheritance is followed in Javascript.

There are two types of Inherientence in OOPS Classic and Prototypical Inheritance. Javascript follows Prototypical Inheritance.

### Question 31. What is output of undefined \* 2 in Javascript?

nan is output of undefined \* 2.

### Question 32. How to add/remove properties to object dynamically in Javascript?

You can add a property to an object using object.property\_name =value, delete object.property\_name is used to delete a property.

**Example:**

let user = new Object();

// adding a property

user.name='Anil';

user.age =25;

console.log(user);

delete user.age;

console.log(user);

### Question 33. How to convert Javascript date to ISO standard?

**toISOString()** method is used to convert javascript date to ISO standard. It converts JavaScript Date object into a string, using the ISO standard.

**Usage:**

var date = new Date();

var n = date.toISOString();

console.log(n);

// YYYY-MM-DDTHH:mm:ss.sssZ

### Question 34. How to get inner Html of an element in JavaScript?

InnerHTML property of HTML DOM is used to get inner Html of an element in JavaScript.

**Example Usage:**

This is inner Element

<script type="text/javascript">

var inner= document.getElementById("inner").innerHTML ;

console.log(inner); // This is inner Element

document.getElementById("inner").innerHTML = "Html changed!";

var inner= document.getElementById("inner").innerHTML ;

console.log(inner); // Html changed!

</script>

### Question 35. How to clone an object in Javascript?

Object.assign() method is used for cloning an object in Javascript.Here is sample usage

var x = {myProp: "value"};

var y = Object.assign({}, x);

* )

### Question 37. How to get an element by class in JavaScript ?

**document.getElementsByClassName()** method is used in Javascript to get an element with a class name.

|  |  |
| --- | --- |
| **getElementsByClassName()** | |
| **Method Name** | getElementsByClassName |
| **Syntax** | document.getElementsByClassName('className') |
| **Parameter** | String (name of class) |
| **Output** | Array of HTMLCollection that have inputted className |

### Question 38. Explain Typecasting in Javascript?

In Programming whenever we need to convert a variable from one data type to another Typecasting is used. In Javascript, we can do this via library functions. There are basically 3 typecasts are available in Javascript Programming, they are:

* Boolean(value): Casts the inputted value to a Boolean
* Number(value): Casts the inputted value to an Integer or Floating point Number.
* String(value) : Casts the inputted value value a string

### Question 39. How to encode and decode a URL in JavaScript?

**encodeURI()** function is used to encode an URL in Javascript.It takes a url string as parameter and return encoded string. Note: encodeURI() did not encode characters like **/ ? : @ & = + $ #**, if you have to encode these characters too please use encodeURIComponent(). Usage:

var uri = "my profile.php?name=sammer&occupation=pāntiNG";

var encoded\_uri = encodeURI(uri);

**decodeURI()** function is used to decode an URL in Javascript.It takes a encoded url string as parameter and return decoded string. Usage:

var uri = "my profile.php?name=sammer&occupation=pāntiNG";

var encoded\_uri = encodeURI(uri);

decodeURI(encoded\_uri);

### Question 40. How to you change the title of the page by JavaScript?

You can change the title of a webpage using setting the title property of the document object.

**Example usage**

document.title="My New Title";

### Question 41. What is difference between deep and shallow object coping in JavaScript?

Some differences are:

* Deep copy means copies all values or properties recursively in the new object whereas shallow copy copies only the reference.
* In a deep copy, changes in the new object don't show in original object whereas, in shallow copy, changes in new objects will reflect in the original object.
* In a deep copy, original objects do not share the same properties with new object whereas, in shallow copy, they do.

Javascript is an object-oriented computer programming language commonly used to create interactive effects within web browsers.It is first used by the Netscape browser, that provides access to the HTML document object model (DOM), provides access to the browser object model (BOM). Javascript syntax looks a lot like java, c or c++ syntax.

Below is the list of data types supported by Javascript:-

Undefined

Null

Boolean

String

Symbol

Number

Object

Question 2. What close() does in Javascript?

VDO.AI

In Javascript close() method is used to close the current window. You must write window.close() to ensure that this command is associated with a window object and not some other JavaScript object.

Advanced and Basic JavaScript Interview Questions

Question 3. What is the difference between let and var?

Both var and let are used for variable/ method declaration in javascript but the main difference between let and var is that var is function scoped whereas let is block scoped.

Question 4. Explain Closures in JavaScript?

Closures are the combination of lexical environment and function within which the function was declared. This allows JavaScript programmers to write better, more creative, concise and expressive codes. The closure will consist of all the local variables that were in-scope when the closure was created.

Sure, closures appear to be complex and beyond the scope, but after you read this article, closures will be much more easy to understand and more simple for your everyday JavaScript programming tasks. JavaScript is a very function-oriented language it gives the user freedom to use functions as the wish of the programmer.

Question 5. Explain JavaScript Event Delegation Model?

In JavaScript, there is some cool stuff that makes it the best of all. One of them is Delegation Model. When capturing and bubbling, allow functions to implement one single handler to many elements at one particular time then that is called event delegation. Event delegation allows you to add event listeners to one parent instead of specified nodes. That particular listener analyzes bubbled events to find a match on the child elements. Many people think it to be complicated but in reality, it is very simple if one starts understanding it.

Also, Read Five JavaScript Frameworks to learn in 2018

Any number divided by NEGATIVE\_INFINITY is zero.

Question 8. What is the use of let & const in JavaScript?

In modern javascript let & const are different ways of creating variables. Earlier in javascript, we use the var keyword for creating variables. let & const keyword is introduced in version ES6 with the vision of creating two different types of variables in javascript one is immutable and other is mutable.

const: It is used to create an immutable variable. Immutable variables are variables whose value is never changed in the complete life cycle of the program.

let: let is used to create a mutable variable. Mutable variables are normal variables like var that can be changed any number of time.

Question 9. Explain Arrow functions?

An arrow function is a consise and short way to write function expressions in Es6 or above.A rrow functions cannot be used as constructors and also does not supports this, arguments, super, or new.target keywords. It is best suited for non-method functions. In general an arrow function looks like const function\_name= ()=>{}

const greet=()=>{console.log('hello');}

greet();

Question 10. What are exports and imports?

Imports and exports help us to write modular javascript code. Using Imports and exports we can split our code into multiple files. Imports allow taking only some specific variables or methods of a file. We can import methods or variables that are exported by a module. See the below example for more detail.

//index.js

import name,age from './person';

console.log(name);

console.log(age);

//person.js

let name ='Sharad', occupation='developer', age =26;

export { name, age};

Question 11. What is difference between module.exports and export?

The module is a plain JavaScript object with an exports property. Exports is a plain JavaScript variable that happens to be set to module.exports. At the end of your file, node.js will basically ‘return’ module.exports to the require function. A simplified way to view a JS file in Node could be this:

var module = { exports: {} };

var exports = module.exports;

// your code

return module.exports;

If you set a property on exports, like exports.a = 9;, that will set module.exports.a as well because objects are passed around as references in JavaScript, which means that if you set multiple variables to the same object, they are all the same object; so then exports and module.exports are the same objects.

But if you set exports to something new, it will no longer be set to module.exports, so exports and module.exports are no longer the same objects.

Source : https://stackoverflow.com/questions/16383795/difference-between-module-exports-and-exports-in-the-commonjs-module-system

Question 12. How to import all exports of a file as an object.

import \* as object name from ‘./file.js’ is used to import all exported members as an object. You can simply access the exported variables or methods using dot (.) operator of the object.

Example:

objectname.member1;

objectname.member2;

objectname.memberfunc();

Question 13. Explain “use strict” ?

“use strict” is a javascript directive that is introduced in Es5. The purpose of using “use strict” directive is to enforce the code is executed in strict mode. In strict mode we can’t use a variable without declaring it. “use strict” is ignored by earlier versions of Javascript.

Question 14. Explain Event bubbling and Event Capturing in JavaScript?

Event Capture and Bubbling: In HTML DOM API there are two ways of event propagation and determines the order in which event will be received. The two ways are Event Bubbling and Event Capturing. The first method event bubbling directs the event to its intended target, and the second is called event capture in which the event goes down to the element.

Event Capture

The capture procedure is rarely used but when it’s used it proves to be very helpful. This process is also called ‘trickling’. In this process, the event is captured first by the outermost element and then propagated to the innermost element. For example:

<div>

<ul>

<li></li>

</ul>

</div>

From the above example, suppose the click event did occur in the ‘li’ element, in that case capturing event it will be first handled ‘div’, then ‘ul’ and at last the target element will be hit that is ‘li’

Event Bubbling

Bubbling just works like the bubbles, the event gets handled by the innermost element and then propagated to the outer element.

<div>

<ul>

<li></li>

</ul>

</div>

From the above example, suppose the click event did occur in the ‘li’ element in bubbling model the event will be handled first by ‘li’ then by ‘ul’ and at last by ‘div’ element.

Question 15. In Javascript are calculations with fractional numbers guaranteed to be precise?

NO, calculations with fractional numbers are not guaranteed to be precise in Javascript

Question 16. List the comparison operators supported by Javascript?

Javascript supports below comparison operators

> Greater than

< Less than

<= Less than or equal to

>= Greater than or equal to

== Equal to

!= Not Equal to

=== Equal to with datatype check

!== Not equal to with datatype check

Question 18. What will happen if an infinite while loop is run in Javascript?

The program will crash the browser.

Question 19. List HTML DOM mouse events?

HTML DOM mouse events

onclick

ondblclick

mousemove

mousedown

mouseover

mouseout

mouseup

Question 20. How to get the last index of a string in Javascript?

string.length-1 is used to get the last index of a string in Javascript

Example Usage:-

var myString="JavascriptQuestions";

console.log(myString.length-1);

Question 21. How to get the primitive value of a string in Javascript?

In Javascript valueOf() method is used to get the primitive value of a string.

Example Usage:

var myVar= "Hi!"

console.log(myVar.valueOf())

Question 22. What are the primitive data types in JavaScript?

A primitive is a basic data type that’s not built out of other data types. It can only represent one single value. All primitives are built-in data types by necessity, (the compiler has to know about them,) but not all built-in data types are primitives.

In JavaScript there are 5 primitive data types are available they are undefined, null, boolean, string and number are available.Everything else in Javascript is an object.

Question 23. What does the instanceof operator do?

In Javascript instanceof operator checks whether the object is an instance of a class or not:

Example Usage

Square.prototype = new Square();

console.log(sq instanceof Square); // true

Question 24. What is Javascript BOM?

BOM stands for “Browser Object Modal” that allows Javascript to ‘talk’ to the browser, no standards, modern browsers implement similar BOMS – window, screen, location, history, navigator, timing, cookies.

Question 25. What are different types of Popup boxes available in Javascript?

In Javascript there are 3 types of Popup Boxes are available, they are

Alert

Confirm

Prompt

Read 80+ Best Angular Js Interview Questions

Question 26. How can you create an array in Javascript?

There are 3 different ways to create an array in Javascript. They are

By array literal

usage:

var myArray=[value1,value2...valueN];

By creating instance of Array

usage:

var myArray=new Array();

By using an Array constructor

usage:

var myArray=new Array('value1','value2',...,'valueN');

.

Question 29. What is the difference between the substr() and substring() functions in JavaScript?

Difference between the substr() and substring() functions in JavaScript.

The substr() function has the form substr(startIndex,length). It returns the substring from startIndex and returns ‘length’ number of characters.

var s = "hello";

( s.substr(1,4) == "ello" ) // true

The substring() function has the form substring(startIndex,endIndex). It returns the substring from startIndex up to endIndex – 1.

var s = "hello";

( s.substring(1,4) == "ell" ) // true

Question 30. What are different types of Inheritence? Which Inheritance is followed in Javascript.

There are two types of Inherientence in OOPS Classic and Prototypical Inheritance. Javascript follows Prototypical Inheritance.