JavaScript Interview Questions

What is Javascript

Javascript is a scripting language(supports scripts) for Web pages but it is also used in non-browser environments as well. It is a powerful, lightweight, interpreted, scripting language with first-class functions (i.e. the language supports passing functions as arguments to other functions).

In order to add the dynamic interactivity to web pages Javascript is embedded within Hypertext Markup Language (HTML). Since it runs on the client-side of the web so it can be used to design/program how the web pages behave on the occurrence of a particular event. That's the reason why it is widely used for the behavior of the web pages.

Read Advanced and Basic **Javascript Interview Questions and Answers**. We have listed some Javascript Developer interview questions with their answers that help you to crack javascript interviews

Q1. Explain what is 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

Q2. 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.

Q3. 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.

Q4. 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.

Q5. 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**

Q6. 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.

Q7. 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');
}
```

Function Expression

When a function is created by using an expression it is called function expression.

e.g:

```
notHoisted(); // TypeError: notHoisted is not a function
var notHoisted = function() {
   console.log('bar');
};
```

Q8. 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.

Q9. 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();
```

Q10. 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};
```

Q11. 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

Q12. 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();
```

Q13. 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.

Q14. 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.

Q15. In Javascript are calculations with fractional numbers guaranteed to be precise?

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

Q16. 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

Q17. 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**

Q18. What will happen if an infinite while loop is run in Javascript?

The program will crash the browser.

Q19. List HTML DOM mouse events?

HTML DOM mouse events

- onclick
- ondblclick
- mousemove
- mousedown
- mouseover
- mouseout
- mouseup

Q20. 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);
```

Q21. 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())
```

Q22. 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.

Q23. 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
```

Q24. 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.

Q25. 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**

Q26. 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');
  ```

Q27. 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.

Q28. 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;
}
```

Q29. 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
```

Q30. 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.

Q31. What is output of undefined * 2 in Javascript?

nan is output of undefined * 2.

Q32. 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);
```

Q33. 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
```

Q34. 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>
```

Q35. 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);
```

Q36. List different ways of empty an array in Javascript?

In Javascript, there are many ways to empty an array in Javascript, below we have listed 4 major

- By assigning an empty array.

  ```
  var arr1 =[1,4,5,6];
  arr1=[];
  ```

- By assigning array length to 0.

  ```
  var arr2 =[1,4,5,6];
  arr2.length=0;
  ```

- By poping the elements of the array.

  ```
  var arr2 =[1,4,5,6];
  while(arr.length > 0) {
      arr.pop();
  }
  ```

- By using .splice() .

  ```
  var arr =[1,4,5,6];
  arr.splice(0,arr.length)
  ```

Q37. 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

Q38. 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

Q39. 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=p?ntiNG";
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=p?ntiNG";
var encoded_uri = encodeURI(uri);
decodeURI(encoded_uri);
```

Q40. 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";
```

Q41. 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.

Q42. List some Unit Testing Frameworks JavaScript

Below is the list of few most Popular Javascript Unit Testing Frameworks:

- Unit.js
- Jasmine
- Karma
- Chai
- AVA
- Mocha
- JSUnit
- QUnit
- Jest

Q43. How to add a new property in existing function JavaScript?

It is easy to add a new property in existing function by just giving value to the existing function it. For example, let we have an existing object person, to give new property check the below code:

person.country= “India”;

The new property “country” has added to the object person.

Q44. Explain JavaScript Accessors ?

JavaScript Accessors

Q45. List few difference between primitive and non primitive JavaScript data types?

- The primitive data types are numbers, strings, Boolean, undefined, null and anything other than these data types are known as non-primitive such as objects and functions.
- Primitive data types are immutable while non-primitives are mutable.

- Primitives are known immutable as they can't be changed once they created but non-primitive are changeable, means once an object is created, it can be changed.
- Primitives data types are compared with their values, it means two values are strictly equal if they have the same data type and holds the same value.
- Non-primitives are not compared with values. For example, if two objects have the same properties and values, they are strictly not equal.

Q46. Explain higher-order functions in JavaScript?

Higher order function is the best feature of functional programming available in JavaScript. It is the function which takes a function as an argument and returns a function as a result. Some of the inbuilt higher-order functions are mapping, filtering, reduction, zipping, etc.

Q47. Explain few difference between null, undefined or undeclared JavaScript variable?

Null is a value that can be assigned to a variable or an object.

Undefined means a variable has been declared but no value is assigned to it. This type of variable is declared itself to be undefined.

Undeclared means the variable has declared without any datatype.

Null, Undefined are primitive data types whereas Undeclared is not a primitive data type.

Q48. How host objects are different from native objects in JavaScript?

Host objects: These are those objects which environment gives. It means they are different for different environments. For example, browsers include objects such as windows but Node.js environments give objects such as Node List.

Native Objects: these are built-in objects in JavaScript. They are also known as Global Objects because they will be available to you independent of ay environment if you working in JavaScript.

Q49. What is difference between var x =1; and x=1;?

Q50. Explain spread operator in JavaScript?

The spread operator expands an expression in places where multiple argument/variables/elements are needed to present. It represents with three dots (…).

For example:

var mid = [3, 4];

var newarray = [1, 2, ...mid, 5, 6];

console.log(newarray);

// [1, 2, 3, 4, 5, 6]

In above example, instead of appending mid array, it rather expands in the newarray with the help of spread operator. This is how spread operator works in JavaScript.

Q51. How to remove duplicates from JavaScript Array?

There are many ways to remove duplicates from JavaScript array. These are described below with examples:

1. By using Set: It is the simplest approach to remove duplicates. Set is an inbuilt object to store unique values in an array. Here's how we use set:

```
    function uniquearray(array) {
       let unique_array= Array.from(set(array))
       return unique_array;}
```

As in the above code, you created a set of an array which automatically eliminates the duplicate values.

2. By using Filter: Another approach to remove duplicates from an array is applying filter on an array. To call filter method, it requires three arguments: array, current element, index of current element. Here's how we use filter:

```
function unque_array (arr){
 let unique_array = arr.filter(function(elem, index, self) {
 return index == self.indexOf(elem); }
return unique_array }
 console.log(unique_array(array_with_duplicates));
```

3. By using for loop: In this, we can use for loop to remove duplicates. In this we make an empty array in which those elements will be added from the duplicate array which are not present in this before. Thus, finally we will get an array which has unique elements. Code to implement this:

```
Array dups_names = ['Ron', 'Pal', 'Fred', 'Rongo', 'Ron'];function
 dups_array(dups_names) { let
 unique = {}; names.forEach(functio
n(i) {    If
 (!unique[i]) {      unique[i] = true
;    }
  });return
```

```
 Object.keys(unique);}   // Ron, Pal, Fred, Rongo
Dups_array(names);
```

These are the main three methods used in JavaScript to get a unique array.

Q52. How to call a function in every x seconds in JavaScript?

In JavaScript, we use the function setInterval() to call any function in every x seconds.

Syntax: setInterval(function, milliseconds, param1, param2, ...)

Function: it is a required parameter which includes the function to be execute.

Milliseconds: required parameter which tells how often the function will execute.

Others are an additional parameter.

For example: setInterval(function (){ alert("Hello"); }, 3000);

In the above example, this function calls hello function in very 3 seconds.

Q53. Explain Promise in JavaScript?

A promise is an object in JavaScript which is used to produce a value that may give results in the future. The value can be resolved value or it can be a reason which tells why the value is not resolved.

A promise can be of three states:

- **Fulfilled:** The operation is completed and the promise has a specific value.
- **Rejected:** The operation is failed and promise has a reason which shows why the operation failed.
- **Pending:** Th operation is not fulfilled or rejected, which means it has not completed yet.

Q54. What is difference between Array.splice() and Array.slice() method in JavaScript?

- The array.slice() removes items from the array and then return those removed items as an array whereas array.slice() method is selected items from an array and then those elements as a new array object.
- The splice() method affects the original array whereas slice() method doesn’t affect the original array.
- Splice() method takes n number of arguments whereas slice() can take only two arguments.

Syntax of splice(): array.splice(index, howmany, item1, ..., itemX)

Syntax of slice(): array.slice(start, end)

Q55. Is JavaScript multi-threaded or single-threaded?

JavaScript is single-threaded.

Q56. Explain JavaScript Debounce Function?

Q57. List some Design patterns in JavaScript?

The design pattern is a general reusable solution to a commonly occurring problem in software design. Some of the design patterns are:

1. **Creational design pattern:** These patterns dealt with the mechanism of object creation which optimize object creation with the basic approach.
2. **Structural design pattern:** these patterns deal with different classes and objects to provide new functionality.
3. **Behavioral Patterns:** These patterns are to improve communication between objects and to recognize patterns.
4. **Concurrency design patterns:** These patterns handle with multi-thread programming paradigms.
5. **Architectural design patterns:** These patterns used to deal with architectural designs.

Q58. What is console.time() and console.timeEnd()? What is its syntax, and why is it used?

Q59. What are different types of Scope Chain available in JavaScript?

If we check in the program, every local scope has a connection with one or more scope in their back which forms a chain. This chain goes on until it met with the global scope which is the root of this hierarchy. As global scope doesn't have a parent, so it is on the top of the chain. This is known as scope chain.

The scope chain in JavaScript is basically used to resolve the values of the variable. Without this, it is difficult for a JavaScript to choose a certain value for a variable if there are many variables defined at different scopes.

Q60. How to remove duplicate values from a JavaScript array?

We can use array.indexOf method to check a value exists or not. See below example to remove duplicate values.

let duplicates = ['delhi','kanpur','kanpur','goa','delhi','new york'];

```
function removeDuplicatesValues(arr){
    let unique_array = [];
    for(let i = 0;i < arr.length; i++){
        if(unique_array.indexOf(arr[i]) == -1){
            unique_array.push(arr[i])
        }
    }
    return unique_array
}
console.log(removeDuplicatesValues(duplicates));
```

Q61. How to redirect a page to another page in Javascript?

There are several ways to redirect page to another page in JavaScript. These are:

1. **Using location.href:** It is the first approach to redirect page. In this, we can go back to access the original document. **Syntax**:window.location.href =“https://www.onlineinterviewquestions.com/”
2. **Using location.replace:** Another approach to redirect page. In this, it is not possible to navigate back to the original document by clicking on the back button as it removes the URL of the original document. **Syntax:** window.location.replace(" https://www.onlineinterviewquestions.com/;");

Q62. Is it possible to do 301 redirects in Javascript ?

JavaScript entirely runs on the client machine. 301 is response code that is sent by the server as a response. So it is not possible to do 301 Redirects In JavaScript.

Q63. Write a program to reverse a string in pure JavaScript?

There are many ways to reverse a string in JavaScript. These are:

Using in-built functions: the inbuilt function reverse() reverses the string directly. Here’ how:

```
str="jQuery";
str = str.split(""); //convert 'jQuery' to array
str = str.reverse(); //reverse 'jQuery' order
str = str.join(""); //then combines  the reverse order values.
alert(str);
```

First split the string to an array, then reverse an array and after that join the characters to form a string.

Using a loop: First, count a number of characters in a string, then apply a decrementing loop on an original

string which starts from the last character and prints each character until count becomes zero.

Q64. List few Difference between JAVA and JavaScript?

Q65. Explain MUL function in Javascript?

MUL means simple multiplication of numbers. It is a techique in which you pass a one value as an argument in a function and that function returns another function to which you pass the second value and the process go on. For example: x*y*z can be representing as:

```
function mul (x) {
return function (y) { // anonymous function
return function (z) { // anonymous function
return x * y * z;    };
  };
}
```

Q66. List few advantages of using JavaScript?

Few advantage og Javascript

- Javascript is executed on user's computer, the meaning is that whatever you do in Javascript will not add any processing strain on the server. and that's why it is called as the client-side programming language. And this feature makes your sites responsive for the end user and less expensive for you in terms of server traffic.
- With the help of Javascript, you can create highly responsive interfaces which will improve the user experience and provide dynamic functionality, without waiting for the server to show another page.
- If you want to make online systems available offline and sync automatically once the computer goes online, then Javascript is the best technology you can use. you can do this using the right browser add-ons (Such as Google or Yahoo Browser Plus).
- Content loading and changing it dynamically. Using Ajax in Javascript you can load content into the document if and when the user needs it, without reloading the entire page.
- Using the Principles of unobtrusive JavaScript(defensive Scripting), JavaScript can test for what is possible in your browser and react accordingly.

Q67. What is use of settimeout function in JavaScript?

Q68. What is difference between local and global scope in JavaScript ?

Q69. What are anonymous functions in JavaScript ?

Q70. Please explain equality operators in JavaScript?

Q71. What are the different types of errors available in JavaScript?

There are three types of errors available in JavaScript

- **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 a different operation.

Javascript is an object-oriented programming language that supports the creation of both client and server side applications. It is dynamic, weakly typed, prototype-based and multi-paradigm high-level programming language.

Below are list top 8 trending Javascript Frameworks Interview questions and answers.

Browse Latest Advance Interview Questions on Javascript Frameworks like React js, Vue Js, Angular Js, Ember js, Backbone js, D3 js, Marionette js, knockout js, Node js, Underscore.js, Sails.js, Phantom js, Less.js and many more.

- 

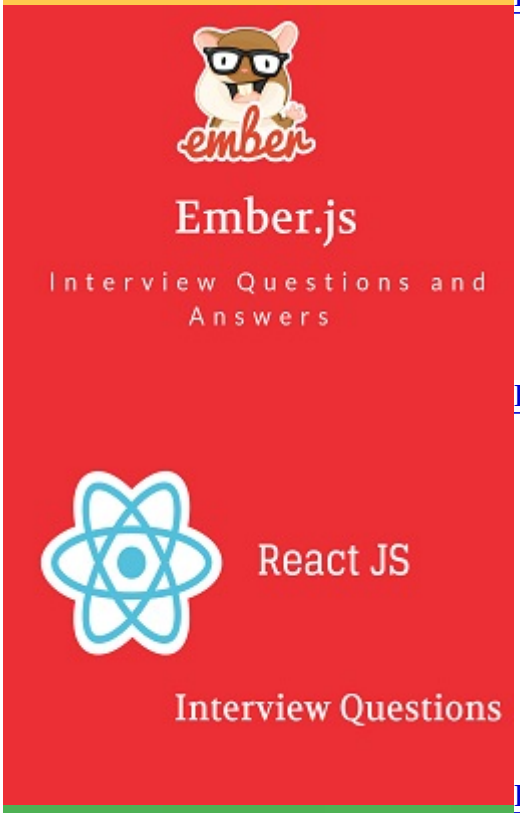
Angular Js Interview Questions
- 

Backbone js interview questions and Answers
- 

D3 js interview questions
- 

Extjs interview questions

- 

Knockout js interview questions
- 

Ember js Interview questions
-

React.js Interview questions
- 

Vue js interview questions

Some Interesting facts about Javascript:

1. JavaScript works on web users computers even when they are offline!
2. JavaScript is the world's most popular programming language.
3. World's most misunderstood programming language.
4. JavaScript will be used as long as people use the internet.
5. JavaScript is also known as Mocha, or LiveScript, or JScript, or ECMAScript.

6. Virtually every personal computer in the world has at least one JavaScript interpreter installed on it and in active use.
7. Javascript was popularly known as a Client Side language, but you can even send Server request with it.
8. With the popularity of Node.js, it can be used as both frontend and backend programming language, JavaScript has become a full-stack language.
9. In JavaScript, NAN (Not a Number) is of type Number. Unbelievable, isn't it?
10. JavaScript is the most-used programming language on Github.