JavaScript Interview Questions

JavaScript interview questions and answers for provides a list of top 20 interview questions. The frequently asked JavaScript interview questions with answers for beginners and professionals are given below.

1) What is JavaScript?

JavaScript is *a scripting language.* It is different from Java language. It is object-based, lightweight, cross-platform translated language. It is widely used for client-side validation. The JavaScript Translator (embedded in the browser) is responsible for translating the JavaScript code for the web browser. More details.

2) List some features of JavaScript.

Some of the features of JavaScript are:

- Lightweight

- Interpreted programming language

- Good for the applications which are network-centric

- Complementary to Java

- Complementary to HTML

- Open source
  Cross-platform

3) Who developed JavaScript, and what was the first name of JavaScript?

JavaScript was developed by Brendan Eich, who was a Netscape programmer. Brendan Eich developed this new scripting language in just ten days in the year September 1995. At the time of its launch, JavaScript was initially called Mocha. After that, it was called Live Script and later known as JavaScript.

4) List some of the advantages of JavaScript.

Some of the advantages of JavaScript are:

- Server interaction is less

- Feedback to the visitors is immediate

- Interactivity is high

- Interfaces are richer

5) List some of the disadvantages of JavaScript.

Some of the disadvantages of JavaScript are:

- No support for multithreading
- No support for multiprocessing
- Reading and writing of files is not allowed
- No support for networking applications.

---

6) Define a named function in JavaScript.

The function which has named at the time of definition is called a named function. For example

```
function msg()
{
  document.writeln("Named Function");
}
msg();
```

---

7) Name the types of functions

The types of function are:

- Named - These type of functions contains name at the time of definition. For Example:

  ```
  function display()
  {
    document.writeln("Named Function");
  }
  display();
  ```

- Anonymous - These type of functions doesn't contain any name. They are declared dynamically at runtime.

  ```
  var display=function()
  {
    document.writeln("Anonymous Function");
  }
  display();
  ```

---

8) Define anonymous function

It is a function that has no name. These functions are declared dynamically at runtime using the function operator instead of the function declaration. The function operator is more flexible than a function declaration. It can be easily used in the place of an expression. For example:

```
var display=function()
{
  alert("Anonymous Function is invoked");
```

```
}
display();
```

9) Can an anonymous function be assigned to a variable?

Yes, you can assign an anonymous function to a variable.

10) In JavaScript what is an argument object?

The variables of JavaScript represent the arguments that are passed to a function.

11) Define closure.

In JavaScript, we need closures when a variable which is defined outside the scope in reference is accessed from some inner scope.

```
var num = 10;
function sum()
{
document.writeln(num+num);
}
sum();
```

12) If we want to return the character from a specific index which method is used?

The JavaScript string charAt() method is used to find out a char value present at the specified index. The index number starts from 0 and goes to n-1, where n is the length of the string. The index value can't be a negative, greater than or equal to the length of the string. For example:

```
var str="Javatpoint";
document.writeln(str.charAt(4));
```

13) What is the difference between JavaScript and JScript?

Netscape provided the JavaScript language. Microsoft changed the name and called it JScript to avoid the trademark issue. In other words, you can say JScript is the same as JavaScript, but Microsoft provides it.

14) How to write a hello world example of JavaScript?

A simple example of JavaScript hello world is given below. You need to place it inside the body tag of HTML.

```
<script type="text/javascript">
document.write("JavaScript Hello World!");
</script>
```

More details.

15) What are the key differences between Java and JavaScript? / How is JavaScript different from Java?

JavaScript is a lightweight programming language (most commonly known as scripting language) developed by Netscape, Inc. It is used to make web pages interactive. It is not a part of the Java platform. Following is a list of some key differences between Java and JavaScript

A list of key differences between Java and JavaScript

Java	JavaScript
Java is a complete and strongly typed programming language used for backend coding. In Java, variables must be declared first to use in the program, and the type of a variable is checked at compile-time.	JavaScript is a weakly typed, lightweight programming language (most commonly known as scripting language) and has more relaxed syntax and rules.
Java is an object-oriented programming (OOPS) language or structured programming languages such as C, C++, or .Net.	JavaScript is a client-side scripting language, and it doesn't fully support the OOPS concept. It resides inside the HTML documents and is used to make web pages interactive (not achievable with simple HTML).
Java creates applications that can run in any virtual machine (JVM) or browser.	JavaScript code can run only in the browser, but it can now run on the server via Node.js.
The Java code needs to be compiled.	The JavaScript code doesn't require to be complied.
Java Objects are class-based. You can't make any program in Java without creating a class.	JavaScript Objects are prototype-based.
Java is a Complete and Standalone language that can be used in backend coding.	JavaScript is assigned within a web page and integrates with its HTML content.
Java programs consume more memory.	JavaScript code is used in HTML web pages and requires less memory.
The file extension of the Java program is written as ".Java" and it translates source code into bytecodes which are then executed by JVM (Java Virtual Machine).	The JavaScript file extension is written as ".js" and it is interpreted but not compiled. Every browser has a JavaScript interpreter to execute the JS code.
Java supports multithreading.	JavaScript doesn't support multithreading.
Java uses a thread-based approach to concurrency.	JavaScript uses an event-based approach to concurrency.

16) How to use external JavaScript file?

I am assuming that js file name is message.js, place the following script tag inside the head tag.

```
<script type="text/javascript" src="message.js"></script>
```

More details.

17) Is JavaScript case sensitive language?

Yes, JavaScript is a case sensitive language. For example:

```
Var msg = "JavaScript is a case-sensitive language"; //Here, var should be used to declare a variable
function display()
{
document.writeln(msg); // It will not display the result.
}
display();
```

18) What is BOM?

BOM stands for *Browser Object Model*. It provides interaction with the browser. The default object of a browser is a window. So, you can call all the functions of the window by specifying the window or directly. The window object provides various properties like document, history, screen, navigator, location, innerHeight, innerWidth,

More Details: Browser Object Model

19) What is DOM? What is the use of document object?

DOM stands for *Document Object Model*. A document object represents the HTML document. It can be used to access and change the content of HTML.

More Details: Document Object Model

20) What is the use of window object?

The window object is created automatically by the browser that represents a window of a browser. It is not an object of JavaScript. It is a browser object.

The window object is used to display the popup dialog box. Let's see with description.

Method	Description
alert()	displays the alert box containing the message with ok button.
confirm()	displays the confirm dialog box containing the message with ok and cancel button.
prompt()	displays a dialog box to get input from the user.
open()	opens the new window.
close()	closes the current window.
setTimeout()	performs the action after specified time like calling function, evaluating expressions.

More details.

21) What is the use of history object?

The history object of a browser can be used to switch to history pages such as back and forward from the current page or another page. There are three methods of history object.

1. history.back() - It loads the previous page.

2. history.forward() - It loads the next page.

3. history.go(number) - The number may be positive for forward, negative for backward. It loads the given page number.

22) How to write a comment in JavaScript?

There are two types of comments in JavaScript.

1. Single Line Comment: It is represented by // (double forward slash)

2. Multi-Line Comment: Slash represents it with asterisk symbol as /* write comment here */

23) How to create a function in JavaScript?

To create a function in JavaScript, follow the following syntax.

```
function function_name(){
//function body
}
```

24) What are the different data types present in JavaScript?

There are two types of data types in JavaScript:

- Primitive data types
- Non- Primitive data types

Primitive data types

The primitive data types are as follows:

String: The string data type represents a sequence of characters. It is written within quotes and can be represented using a single or a double quote.

Example:

```
var str1 = "Hello JavaTpoint"; //using double quotes
var str2 = 'Hello Javatpoint'; //using single quotes
```

Number: The number data type is used to represent numeric values and can be written with or without decimals.

Example:

```
var x = 5; //without decimal
var y = 5.0; //with decimal
```

Boolean: The Boolean data type is used to represent a Boolean value, either false or true. This data type is generally used for conditional testing.

Example:

```
var x = 5;
var y =  6;
var z =  5;
(x == y) // returns false
(x == z) //returns true
```

BigInt: The BigInt data type is used to store numbers beyond the Number data type limitation. This data type can store large integers and is represented by adding "n" to an integer literal.

Example:

```
var bigInteger =  1234567890123456789012345678890;
// This is an example of bigInteger.
```

Undefined: The Undefined data type is used when a variable is declared but not assigned. The value of this data type is undefined, and its type is also undefined.

Example:

```
var x; // value of x is undefined
var y = undefined; // You can also set the value of a variable as undefined.
```

Null: The Null data type is used to represent a non-existent, null, or a invalid value i.e. no value at all.

Example:

```
var  x = null;
```

Symbol: Symbol is a new data type introduced in the ES6 version of JavaScript. It is used to store an anonymous and unique value.

Example:

```
var symbol1 = Symbol('symbol');
```

typeof: The typeof operator is used to determine what type of data a variable or operand contains. It can be used with or without parentheses (typeof(x) or typeof x). This is mainly used in situations when you need to process the values of different types.

Example:

```
typeof 10;  // Returns: "number"
typeof 10.0;  // Returns: "number"
typeof 2.5e-4;  // Returns: "number"
typeof Infinity;  // Returns: "number"
typeof NaN;  // Returns: "number". Despite being "Not-A-Number"
// Strings
typeof '';  // Returns: "string"
typeof 'Welcome to JavaTpoint';  // Returns: "string"
typeof '12';  // Returns: "string". Number within quotes is typeof string
```

```
// Booleans
typeof true;  // Returns: "boolean"
typeof false;  // Returns: "boolean"
// Undefined
typeof undefined;  // Returns: "undefined"
typeof undeclaredVariable; // Returns: "undefined"
// Null
typeof Null;  // Returns: "object"
// Objects
typeof {name: "John", age: 18};  // Returns: "object"
// Arrays
typeof [1, 2, 3];  // Returns: "object"
// Functions
typeof function(){};  // Returns: "function"
```

Non-Primitive data types

In the above examples, we can see that the primitive data types can store only a single value. To store multiple and complex values, we have to use non-primitive data types.

The non-primitive data types are as follows:

Object: The Object is a non-primitive data type. It is used to store collections of data. An object contains properties, defined as a key-value pair. A property key (name) is always a string, but the value can be any data type, such as strings, numbers, Booleans, or complex data types like arrays, functions, and other objects.

Example:

```
// Collection of data in key-value pairs
var obj1 = {
   x:  123,
   y:  "Welcome to JavaTpoint",
   z: function(){
      return this.x;
   }
}
```

Array: The Array data type is used to represent a group of similar values. Every value in an array has a numeric position, called its index, and it may contain data of any data type-numbers, strings, Booleans, functions, objects, and even other arrays. The array index starts from 0 so that the first array element is arr[0], not arr[1].

Example:

```
var colors = ["Red", "Yellow", "Green", "Orange"];
var cities = ["Noida", "Delhi", "Ghaziabad"];
alert(colors[2]);   // Output: Green
alert(cities[1]);   // Output: Delhi
```

More details.

---

25) What is the difference between == and ===?

The == operator checks equality only whereas === checks equality, and data type, i.e., a value must be of the same type.

26) How to write HTML code dynamically using JavaScript?

The innerHTML property is used to write the HTML code using JavaScript dynamically. Let's see a simple example:

```
document.getElementById('mylocation').innerHTML="<h2>This is heading using JavaScript</h2>";
```

More details.

27) How to write normal text code using JavaScript dynamically?

The innerText property is used to write the simple text using JavaScript dynamically. Let's see a simple example:

```
document.getElementById('mylocation').innerText="This is text using JavaScript";
```

More details.

28) How to create objects in JavaScript?

There are 3 ways to create an object in JavaScript.

1. By object literal
2. By creating an instance of Object
3. By Object Constructor

Let's see a simple code to create an object using object literal.

```
emp={id:102,name:"Rahul Kumar",salary:50000}
```

More details.

29) How to create an array in JavaScript?

There are 3 ways to create an array in JavaScript.

1. By array literal
2. By creating an instance of Array
3. By using an Array constructor

Let's see a simple code to create an array using object literal.

```
var emp=["Shyam","Vimal","Ratan"];
```

More details.

30) What does the isNaN() function?

The isNan() function returns true if the variable value is not a number. For example:

```
function number(num) {
  if (isNaN(num)) {
    return "Not a Number";
  }
  return "Number";
}
console.log(number('1000F'));
// expected output: "Not a Number"

console.log(number('1000'));
// expected output: "Number"
```

31) What is the output of 10+20+"30" in JavaScript?

3030 because 10+20 will be 30. If there is numeric value before and after +, it treats as binary + (arithmetic operator).

```
function display()
{
  document.writeln(10+20+"30");
}
display();
```

32) What is the output of "10"+20+30 in JavaScript?

102030 because after a string all the + will be treated as string concatenation operator (not binary +).

```
function display()
{
  document.writeln("10"+20+30);
}
display();
```

33) Difference between Client side JavaScript and Server side JavaScript?

Client-side JavaScript comprises the basic language and predefined objects which are relevant to running JavaScript in a browser. The client-side JavaScript is embedded directly by in the HTML pages. The browser interprets this script at runtime.

Server-side JavaScript also resembles client-side JavaScript. It has a relevant JavaScript which is to run in a server. The server-side JavaScript are deployed only after compilation.

34) In which location cookies are stored on the hard disk?

The storage of cookies on the hard disk depends on the OS and the browser.

The Netscape Navigator on Windows uses a cookies.txt file that contains all the cookies. The path is c:\Program Files\Netscape\Users\username\cookies.txt

The Internet Explorer stores the cookies on a file The path is: c:\Windows\Cookies\

35) What's the difference between event.preventDefault() and event.stopPropagation() methods in JavaScript?

In JavaScript, the event.preventDefault() method is used to prevent the default behavior of an element.

For example: If you use it in a form element, it prevents it from submitting. If used in an anchor element, it prevents it from navigating. If used in a contextmenu, it prevents it from showing or displaying.

On the other hand, the event.stopPropagation() method is used to stop the propagation of an event or stop the event from occurring in the bubbling or capturing phase.

36) What is the real name of JavaScript?

The original name was **Mocha**, a name chosen by Marc Andreessen, founder of Netscape. In September of 1995, the name was changed to LiveScript. In December 1995, after receiving a trademark license from Sun, the name JavaScript was adopted.

37) How can you check if the event.preventDefault() method was used in an element?

When we use the event.defaultPrevent() method in the event object returns a Boolean indicating that the event.preventDefault() was called in a particular element.

38) What is the difference between undefined value and null value?

Undefined value: A value that is not defined and has no keyword is known as undefined value. For example:

```
int number;//Here, a number has an undefined value.
```

Null value: A value that is explicitly specified by the keyword "null" is known as a null value. For example:

```
String str=null;//Here, str has a null value.
```

39) How to set the cursor to wait in JavaScript?

The cursor can be set to wait in JavaScript by using the property "cursor". The following example illustrates the usage:

```
<script>
window.document.body.style.cursor = "wait";
</script>
```

40) What is this [[[]]]?

This is a three-dimensional array.

```
var myArray = [[[]]];
```

41) Are Java and JavaScript same?

No, Java and JavaScript are the two different languages. Java is a robust, secured and object-oriented programming language whereas JavaScript is a client-side scripting language with some limitations.

42) What is negative infinity?

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

```javascript
var num=-5;
function display()
{
  document.writeln(num/0);
}
display();
//expected output: -Infinity
```

43) What is the difference between View state and Session state?

"View state" is specific to a page in a session whereas "Session state" is specific to a user or browser that can be accessed across all pages in the web application.

44) What are the pop-up boxes available in JavaScript?

- Alert Box
- Confirm Box
- Prompt Box

Example of alert() in JavaScript

```html
<script type="text/javascript">
function msg(){
 alert("Hello Alert Box");
}
</script>
<input type="button" value="click" onclick="msg()"/>
```

Example of confirm() in JavaScript

```html
<script type="text/javascript">
function msg(){
var v= confirm("Are u sure?");
if(v==true){
alert("ok");
}
else{
alert("cancel");
}
```

```
}
</script>


<input type="button" value="delete record" onclick="msg()"/>
```

Example of prompt() in JavaScript

```
<script type="text/javascript">
function msg(){
var v= prompt("Who are you?");
alert("I am "+v);


}
</script>


<input type="button" value="click" onclick="msg()"/>
```

45) How can we detect OS of the client machine using JavaScript?

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

46) How to submit a form using JavaScript by clicking a link?

Let's see the JavaScript code to submit the form by clicking the link.

```
<form name="myform" action="index.php">
Search: <input type='text' name='query' />
<a href="javascript: submitform()">Search</a>
</form>
<script type="text/javascript">
function submitform()
{
  document.myform.submit();
}
</script>
```

47) Is JavaScript faster than ASP script?

Yes, because it doesn't require web server's support for execution.

48) How to change the background color of HTML document using JavaScript?

```
<script type="text/javascript">
document.body.bgColor="pink";
</script>
```

49) How to handle exceptions in JavaScript?

By the help of try/catch block, we can handle exceptions in JavaScript. JavaScript supports try, catch, finally and throw keywords for exception handling.

---

50) How to validate a form in JavaScript?

```html
<script>
function validateform(){
var name=document.myform.name.value;
var password=document.myform.password.value;

if (name==null || name==""){
  alert("Name can't be blank");
  return false;
}else if(password.length<6){
  alert("Password must be at least 6 characters long.");
  return false;
  }
}
</script>
<body>
<form name="myform" method="post" action="abc.jsp" onsubmit="return validateform()" >
Name: <input type="text" name="name"><br/>
Password: <input type="password" name="password"><br/>
<input type="submit" value="register">
</form>
```

☑ Test it Now

Visit here: JavaScript form validation.

---

51) How to validate email in JavaScript?

```html
<script>
function validateemail()
{
var x=document.myform.email.value;
var atposition=x.indexOf("@");
var dotposition=x.lastIndexOf(".");
if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length){
  alert("Please enter a valid e-mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);
  return false;
  }
}
</script>
<body>
<form name="myform"  method="post" action="#" onsubmit="return validateemail();">
```

```
Email: <input type="text" name="email"><br/>


<input type="submit" value="register">
</form>
```

Visit here: JavaScript Email validation.

---

52) What is this keyword in JavaScript?

The this keyword is a reference variable that refers to the current object. For example:

```
var address=
{
company:"Javatpoint",
city:"Noida",
state:"UP",
fullAddress:function()
{
return this.company+" "+this.city+" "+this.state;
}
};
var fetch=address.fullAddress();
document.writeln(fetch);
```

---

53) What is the requirement of debugging in JavaScript?

JavaScript didn't show any error message in a browser. However, these mistakes can affect the output. The best practice to find out the error is to debug the code. The code can be debugged easily by using web browsers like Google Chrome, Mozilla Firebox.

To perform debugging, we can use any of the following approaches:

- Using console.log() method
- Using debugger keyword

---

54) What is the use of debugger keyword in JavaScript?

JavaScript debugger keyword sets the breakpoint through the code itself. The debugger stops the execution of the program at the position it is applied. Now, we can start the flow of execution manually. If an exception occurs, the execution will stop again on that particular line.. For example:

```
function display()
{
x = 10;
y = 15;
z = x + y;
debugger;
document.write(z);
```

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

55) What is the role of a strict mode in JavaScript?

The JavaScript strict mode is used to generates silent errors. It provides "use strict"; expression to enable the strict mode. This expression can only be placed as the first statement in a script or a function. For example:

```
"use strict";
x=10;
console.log(x);
```

57) What is the use of Math object in JavaScript?

The JavaScript math object provides several constants and methods to perform a mathematical operation. Unlike date object, it doesn't have constructors. For example:

```
function display()
{
  document.writeln(Math.random());
}
display();
```

58) What is the use of a Date object in JavaScript?

The JavaScript date object can be used to get a year, month and day. You can display a timer on the webpage by the help of JavaScript date object.

```
function display()
{
  var date=new Date();
var day=date.getDate();
var month=date.getMonth()+1;
var year=date.getFullYear();
document.write("<br>Date is: "+day+"/"+month+"/"+year);
}
display();
```

59) What is the use of a Number object in JavaScript?

The JavaScript number object enables you to represent a numeric value. It may be integer or floating-point. JavaScript number object follows the IEEE standard to represent the floating-point numbers.

```
function display()
{
```

```
var x=102;//integer value
var y=102.7;//floating point value
var z=13e4;//exponent value, output: 130000
var n=new Number(16);//integer value by number object
document.write(x+" "+y+" "+z+" "+n);
}
display();
```

60) What is the use of a Boolean object in JavaScript?

The JavaScript Boolean is an object that represents value in two states: true or false. You can create the JavaScript Boolean object by Boolean() constructor.

```
function display()
{
document.writeln(10<20);//true
document.writeln(10<5);//false
}
display();
```

61) What is the use of a TypedArray object in JavaScript?

The JavaScript TypedArray object illustrates an array like a view of an underlying binary data buffer. There is any number of different global properties, whose values are TypedArray constructors for specific element types.

```
function display()
{
var arr1= [1,2,3,4,5,6,7,8,9,10];
    arr1.copyWithin(2) ;
    document.write(arr1);
}
display();
```

62) What is the use of a Set object in JavaScript?

The JavaScript Set object is used to store the elements with unique values. The values can be of any type i.e. whether primitive values or object references. For example:

```
function display()
{
var set = new Set();
set.add("jQuery");
set.add("AngularJS");
set.add("Bootstrap");
for (let elements of set) {
 document.writeln(elements+"<br>");
}
```

```
    }
    display();
```

63) What is the use of a WeakSet object in JavaScript?

The JavaScript WeakSet object is the type of collection that allows us to store weakly held objects. Unlike Set, the WeakSet are the collections of objects only. It doesn't contain the arbitrary values. For example:

```
function display()
{
var ws = new WeakSet();
var obj1={};
var obj2={};
ws.add(obj1);
ws.add(obj2);
//Let's check whether the WeakSet object contains the added object
document.writeln(ws.has(obj1)+"<br>");
document.writeln(ws.has(obj2));
}
display()
```

64) What is the use of a Map object in JavaScript?

The JavaScript Map object is used to map keys to values. It stores each element as key-value pair. It operates the elements such as search, update and delete on the basis of specified key. For example:

```
function display()
{
var map=new Map();
map.set(1,"jQuery");
map.set(2,"AngularJS");
map.set(3,"Bootstrap");
document.writeln(map.get(1)+"<br>");
document.writeln(map.get(2)+"<br>");
document.writeln(map.get(3));
}
display();
```

65) What is the use of a WeakMap object in JavaScript?

The JavaScript WeakMap object is a type of collection which is almost similar to Map. It stores each element as a key-value pair where keys are weakly referenced. Here, the keys are objects and the values are arbitrary values. For example:

```
function display()
{
var wm = new WeakMap();
var obj1 = {};
```

```
var obj2 = {};
var obj3= {};
wm.set(obj1, "jQuery");
wm.set(obj2, "AngularJS");
wm.set(obj3,"Bootstrap");
document.writeln(wm.has(obj2));
}
display();
```

66) What are the falsy values in JavaScript, and how can we check if a value is falsy?

Those values which become false while converting to Boolean are called falsy values.

```
const falsyValues = ['', 0, null, undefined, NaN, false];
```

We can check if a value is falsy by using the Boolean function or the Double NOT operator (!!).

67) What do you understand by hoisting in JavaScript?

Hoisting is the default behavior of JavaScript where all the variable and function declarations are moved on top. In simple words, we can say that Hoisting is a process in which, irrespective of where the variables and functions are declared, they are moved on top of the scope. The scope can be both local and global.

Example 1:

```
hoistedVariable = 12;
console.log(hoistedVariable); // outputs 12 even when the variable is declared after it is initialized
var hoistedVariable;
```

Example2:

```
hoistedFunction();  // Outputs " Welcome to JavaTpoint " even when the function is declared after calling
function hoistedFunction(){
  console.log(" Welcome to JavaTpoint ");
}
Example3:
// Hoisting in a local scope
function doSomething(){
  x = 11;
  console.log(x);
  var x;
}
doSomething(); // Outputs 11 since the local variable "x" is hoisted inside the local scope
```

JavaScript MCQ

1) Which of the following statement is true?

    a. If onKeyDown returns false, the key-up event is canceled.

    b. If onKeyDown returns false, the key-press event is canceled.

    c. If onKeyPress returns false, the key-down event is canceled.

    d. If onKeyPress returns false, the key-up event is canceled.

👁 **Show Answer**　　📝 **Workspace**

2) Which of the following statement shows the primary difference between JavaScript and Java?

    a. There is no difference between JavaScript and Java.

    b. Functions are considered as fields.

    c. Variables are specific.

    d. Functions are values, and there is no such distinction between methods and fields.

👁 **Show Answer**　　📝 **Workspace**

3) Which of the following syntax is valid for creating a RegExp object?

```
1. var txt=new RegExp(pattern,attributes);
2. var txt=/pattern/attributes;
```

    a. Only 1 syntax

    b. Only 2 syntax

    c. Both 1 and 2 syntax

    d. None of the above

👁 **Show Answer**　　📝 **Workspace**

4) Which of the following statement specifies the meaning of Augmenting classes?

    a. Objects inherit prototype properties even in a dynamic state.

    b. Objects inherit prototype properties only in a dynamic state.

    c. Objects inherit prototype properties in the static state.

    d. An object doesn't inherit prototype properties in the static state.

👁 **Show Answer**　　📝 **Workspace**

5) If para1 is the DOM object for a paragraph, which of the following would be the correct syntax to change the text within the paragraph?

    a. para1="New Text"

    b. firstChild.nodeValue= "New Text";

    c. value="New Text";

    d. nodeValue="New Text";

👁 **Show Answer**　　📝 **Workspace**

6) Which of the following is used to define the behavior of the instances present of a class inside a method?

a. Method

b. Classes

c. Interfaces

d. Classes and Interfaces

7) Which of the following statement is true in the case of the property of the JSON() method?

a. A JSON() method can be invoked manually as object.JSON()

b. A JSON() method is invoked automatically by the JSON.stringify() method

c. A JSON() method is automatically invoked by the compiler.

d. A JSON() method cannot be invoked in any form.

8) Which of the following statement is true if class B can extend another class A?

a. A is the superclass, and B is the subclass.

b. B is the superclass, and A is the subclass.

c. Both A and B are the superclass.

d. Both A and B are subclass.

9) Which of the following is the correct syntax of the eval() function?

a. [objectName.]eval(numeric)

b. [EvalName.]eval(string)

c. [EvalName.]eval(numeric)

d. [objectName.]eval(string)

10) What would be the output of the following JavaScript code?

```
const obj1 =
{
   a: 10,
   b: 15,
   c: 18
};
const obj2 = Object.assign({c: 11, d: 12}, obj1);
console.log(obj2.c, obj2.d);
```

a. 10, 12

b. 11, 12

c. Undefined

d. Error

Show Answer   **Workspace**

Other Interview Questions

Core Java Interview Questions

Servlet Interview Questions

JSP Interview Questions

SQL Interview Questions

PL/SQL Interview Questions

Android Interview Questions

You may also like:

- Java Interview Questions
- SQL Interview Questions
- Python Interview Questions
- JavaScript Interview Questions
- Angular Interview Questions
- Selenium Interview Questions
- Spring Boot Interview Questions
- HR Interview Questions
- C Programming Interview Questions
- C++ Interview Questions
- Data Structure Interview Questions
- DBMS Interview Questions
- HTML Interview Questions
- IAS Interview Questions
- Manual Testing Interview Questions
- OOPs Interview Questions
- .Net Interview Questions
- C# Interview Questions
- ReactJS Interview Questions
- Networking Interview Questions
- PHP Interview Questions
- CSS Interview Questions
- Node.js Interview Questions
- Spring Interview Questions
- Hibernate Interview Questions
- AWS Interview Questions
- Accounting Interview Questions

Learn Latest Tutorials

Splunk	SPSS tutorial SPSS	Swagger	T-SQL tutorial Transact-SQL	Tumblr
React tutorial ReactJS	Regex tutorial Regex	Reinforcement Learning	R Programming	RxJS tutorial RxJS
React Native	Python Design Patterns	Python Pillow	Python Turtle	Keras tutorial Keras

Preparation

Aptitude Aptitude	Reasoning	Verbal Ability Verbal Ability	Interview Questions	Company Questions

Trending Technologies

Artificial Intelligence	AWS Tutorial AWS	Selenium	Cloud Computing	Hadoop
ReactJS	Data Science	Angular 7	Blockchain	Git Tutorial Git
Machine Learning	DevOps			

B.Tech / MCA

DBMS tutorial DBMS	Data Structures	DAA tutorial DAA	Operating System	Computer Network

Compiler Design

Computer Organization

Discrete Mathematics

Ethical Hacking

Computer Graphics

Software Engineering

html tutorial

Web Technology

Cyber Security

Automata

C Programming

C++ tutorial

C++

Java tutorial

Java

.Net

Python

Programs

Control System

Data Mining

Data Warehouse