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 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 an 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.
* Q1: What is Coercion in JavaScript?
* Q2: What is Scope in JavaScript?
* Q3: Explain equality in JavaScript
* Q4: What is typeof operator?
* Q5: What is the object type?
* Q6: Explain arrays in JavaScript
* Q7: Explain Values and Types in JavaScript
* Q8: Explain Null and Undefined in JavaScript
* Q9: What is a Polyfill?
* Q10: What is let keyword in JavaScript?
* Q11: Being told that an unsorted array contains (n - 1) of n consecutive numbers (where the bounds are defined), find the missing number in O(n) time
* Q12: Remove duplicates of an array and return an array of only unique elements
* Q1: What is Coercion in JavaScript?
* Q2: What is Scope in JavaScript?
* Q3: Explain equality in JavaScript
* Q4: What is typeof operator?
* Q5: What is the object type?
* Q6: Explain arrays in JavaScript
* Q7: Explain Values and Types in JavaScript
* Q8: Explain Null and Undefined in JavaScript
* Q9: What is a Polyfill?
* Q10: What is let keyword in JavaScript?
* Q11: Being told that an unsorted array contains (n - 1) of n consecutive numbers (where the bounds are defined), find the missing number in O(n) time
* Q12: Remove duplicates of an array and return an array of only unique elements
* Q13: Given a string, reverse each word in the sentence
* Q14: Implement enqueue and dequeue using only two stacks
* Q15: Explain event bubbling and how one may prevent it
* Q16: How to check if an object is an array or not? Provide some code.
* Q17: Write a function that would allow you to do this.
* Q18: How would you check if a number is an integer?
* Q19: What does "use strict" do?
* Q20: Explain what a callback function is and provide a simple example.
* Q21: What are some of the advantages/disadvantages of writing JavaScript code in a language that compiles to JavaScript?
* Q22: Why would you use something like the `load` event? Does this event have disadvantages? Do you know any alternatives, and why would you use those?
* Q23: Make this work
* Q24: Explain the same-origin policy with regards to JavaScript.
* Q25: What is the difference between `==` and `===`?
* Q26: Is there anyway to force using strict mode in Node.js?
* Q27: How to compare two objects in JavaScript?
* Q28: What is the difference between anonymous and named functions?
* Q29: What is the difference between a shim and a polyfill?
* Q30: Could you explain the difference between ES5 and ES6
* Q31: Given an array of integers, find the largest difference between two elements such that the element of lesser value must come before the greater element
* Q32: Find the intersection of two arrays. An intersection would be the common elements that exists within both arrays. In this case, these elements should be unique!
* Q33: Given two strings, return true if they are anagrams of one another
* Q34: Explain the difference between "undefined" and "not defined" in JavaScript
* Q35: What is the drawback of creating true private in JavaScript?
* Q36: What will be the output of the following code?
* Q37: What is IIFEs (Immediately Invoked Function Expressions)?
* Q38: Write a function that would allow you to do this
* Q39: What's the difference between using “let” and “var” to declare a variable in ES6?
* Q40: What are the benefits of using spread syntax in ES6 and how is it different from rest syntax?
* Q41: What is 'Currying'?
* Q42: What are the differences between ES6 class and ES5 function constructors?
* Q43: Explain the differences on the usage of `foo` between `function foo() {}` and `var foo = function() {}`
* Q44: What are the advantages and disadvantages of using "use strict"?
* Q45: Explain `Function.prototype.bind`.
* Q46: What's the difference between `.call` and `.apply`?
* Q47: What's a typical use case for anonymous functions?
* Q48: What do you think of AMD vs CommonJS?
* Q49: Why should we use ES6 classes?
* Q50: Explain the difference between Object.freeze() vs const
* Q51: What is generator in JS?
* Q52: When should we use generators in ES6?
* Q53: What is Hoisting in JavaScript?
* Q54: How does the “this” keyword work? Provide some code examples.
* Q55: What does the term "Transpiling" stand for?
* Q56: Check if a given string is a isomorphic
* Q57: Write a recursive function that performs a binary search
* Q58: Given an integer, determine if it is a power of 2. If so, return that number, else return -1
* Q59: What is “closure” in javascript? Provide an example?
* Q60: What will be the output of the following code?
* Q61: Explain the Prototype Design Pattern
* Q62: How would you add your own method to the Array object so the following code would work?
* Q63: What will the following code output?
* Q64: How would you create a private variable in JavaScript?
* Q65: What are the actual uses of ES6 WeakMap?
* Q66: Can you give an example for destructuring an object or an array in ES6?
* Q67: Explain difference between: `function Person(){}`, `var person = Person()`, and `var person = new Person()`?
* Q68: What's the difference between a variable that is: `null`, `undefined` or undeclared? How would you go about checking for any of these states?
* Q69: Explain why the following doesn't work as an IIFE. What needs to be changed to properly make it an IIFE?
* Q70: Describe the Revealing Module Pattern design pattern
* Q71: Can you give an example of a curry function and why this syntax offers an advantage?
* Q72: How to "deep-freeze" object in JavaScript?
* Q73: What is the difference between the await keyword and the yield keyword?
* Q74: Is it possible to reset an ECMAScript 6 generator to its initial state?
* Q75: Compare Async/Await and Generators usage to achieve same functionality

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.](https://www.javatpoint.com/javascript-tutorial)

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) 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

4) 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.

5) Define a named function in JavaScript.

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

1. function msg()
2. {
3. document.writeln("Named Function");
4. }
5. msg();

6) Name the types of functions

The types of function are:

* Named - These type of functions contains name at the time of definition. For Example:
  1. function display()
  2. {
  3. document.writeln("Named Function");
  4. }
  5. display();
* Anonymous - These types of functions don't contain any name. They are declared dynamically at runtime.
  1. var display=function()
  2. {
  3. document.writeln("Anonymous Function");
  4. }
  5. display();

7) 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:

1. var display=function()
2. {
3. alert("Anonymous Function is invoked");
4. }
5. display();

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

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

9) In JavaScript what is an argument object?

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

10) Define closure.

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

1. var num = 10;
2. function sum()
3. {
4. document.writeln(num+num);
5. }
6. sum();

11) 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:

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

12) 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.

13) 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.

1. **<script** type="text/javascript"**>**
2. document.write("JavaScript Hello World!");
3. **</script>**

[More details.](https://www.javatpoint.com/javascript-example)

14) 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.

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

[More details.](https://www.javatpoint.com/external-javascript-file)

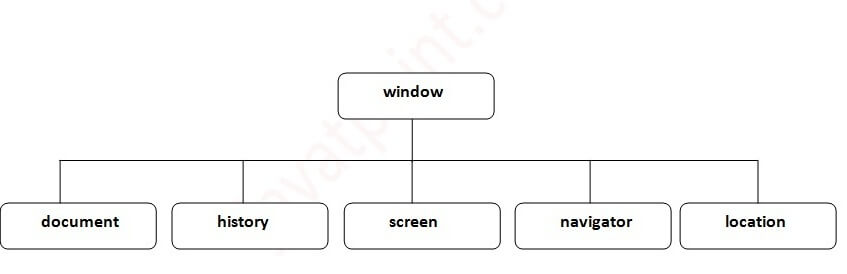
15) Is JavaScript case sensitive language?

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

1. let msg = "JavaScript is a case-sensitive language"; //Here, var should be used to declare a variable
2. function display()
3. {
4. document.writeln(msg); // It will not display the result.
5. }
6. display();

16) 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](https://www.javatpoint.com/browser-object-model)

17) 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](https://www.javatpoint.com/document-object-model)

18) 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.](https://www.javatpoint.com/window-object)

19) 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.

[More details.](https://www.javatpoint.com/javascript-history-object)

20) 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 \*/

[More details.](https://www.javatpoint.com/javascript-comment)

21) How to create a function in JavaScript?

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

1. function function\_name(){
2. //function body
3. }

[More details.](https://www.javatpoint.com/javascript-function)

22) What are the JavaScript data types?

There are two types of data types in JavaScript:

1. Primitive Data Types - The primitive data types are as follows:

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| String | represents a sequence of characters, e.g., "hello" |
| Number | represents numeric values, e.g., 100 |
| Boolean | represents boolean value either false or true |
| Undefined | represents an undefined value |
| Null | represents null, i.e., no value at all |

1. Non-primitive Data Types - The non-primitive data types are as follows:

|  |  |
| --- | --- |
| **Data Type** | **Description** |
| Object | represents an instance through which we can access members |
| Array | represents a group of similar values |
| RegExp | represents regular expression |

[More details.](https://www.javatpoint.com/javascript-data-types)

23) 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.

24) 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:

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

[More details.](https://www.javatpoint.com/javascript-innerHTML)

25) 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:

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

[More details.](https://www.javatpoint.com/javascript-innerText)

26) 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.

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

[More details.](https://www.javatpoint.com/javascript-objects)

27) 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.

1. var emp=["Shyam","Vimal","Ratan"];

[More details.](https://www.javatpoint.com/javascript-array)

28) What does the isNaN() function?

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

1. function number(num) {
2. if (isNaN(num)) {
3. return "Not a Number";
4. }
5. return "Number";
6. }
7. console.log(number('1000F'));
8. // expected output: "Not a Number"
10. console.log(number('1000'));
11. // expected output: "Number"

29) 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).

1. function display()
2. {
3. document.writeln(10+20+"30");
4. }
5. display();

30) 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 +).

1. function display()
2. {
3. document.writeln("10"+20+30);
4. }
5. display();

31) 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.

32) 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 username@website.txt. The path is: c:\Windows\Cookies\username@Website.txt.

33) 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.

34) 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:

1. 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:

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

35) 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:

1. **<script>**
2. window.document.body.style.cursor = "wait";
3. **</script>**

36) What is this [[[]]]?

This is a three-dimensional array.

1. var myArray = [[[]]];

37) 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.

38) What is negative infinity?

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

1. var num=-5;
2. function display()
3. {
4. document.writeln(num/0);
5. }
6. display();
7. //expected output: -Infinity

39) 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.

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

* Alert Box
* Confirm Box
* Prompt Box

Example of alert() in JavaScript

1. **<script** type="text/javascript"**>**
2. function msg(){
3. alert("Hello Alert Box");
4. }
5. **</script>**
6. **<input** type="button" value="click" onclick="msg()"**/>**

Example of confirm() in JavaScript

1. **<script** type="text/javascript"**>**
2. function msg(){
3. var v= confirm("Are u sure?");
4. if(v==true){
5. alert("ok");
6. }
7. else{
8. alert("cancel");
9. }
11. }
12. **</script>**
14. **<input** type="button" value="delete record" onclick="msg()"**/>**

Example of prompt() in JavaScript

1. **<script** type="text/javascript"**>**
2. function msg(){
3. var v= prompt("Who are you?");
4. alert("I am "+v);
6. }
7. **</script>**
9. **<input** type="button" value="click" onclick="msg()"**/>**

41) 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.

42) 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.

1. **<form** name="myform" action="index.php"**>**
2. Search: **<input** type='text' name='query' **/>**
3. **<a** href="javascript: submitform()"**>**Search**</a>**
4. **</form>**
5. **<script** type="text/javascript"**>**
6. function submitform()
7. {
8. document.myform.submit();
9. }
10. **</script>**

43) Is JavaScript faster than ASP script?

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

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

1. **<script** type="text/javascript"**>**
2. document.body.bgColor="pink";
3. **</script>**

45) 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.

46) How to validate a form in JavaScript?

1. **<script>**
2. function validateform(){
3. var name=document.myform.name.value;
4. var password=document.myform.password.value;
6. if (name==null || name==""){
7. alert("Name can't be blank");
8. return false;
9. }else if(password.length**<6**){
10. alert("Password must be at least 6 characters long.");
11. return false;
12. }
13. }
14. **</script>**
15. **<body>**
16. **<form** name="myform" method="post" action="abc.jsp" onsubmit="return validateform()" **>**
17. Name: **<input** type="text" name="name"**><br/>**
18. Password: **<input** type="password" name="password"**><br/>**
19. **<input** type="submit" value="register"**>**
20. **</form>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=jsvalidation1)

Visit here: [JavaScript form validation](https://www.javatpoint.com/javascript-form-validation).

47) How to validate email in JavaScript?

1. **<script>**
2. function validateemail()
3. {
4. var x=document.myform.email.value;
5. var atposition=x.indexOf("@");
6. var dotposition=x.lastIndexOf(".");
7. if (atposition**<1** || dotposition**<atposition**+2 || dotposition+2**>**=x.length){
8. alert("Please enter a valid e-mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);
9. return false;
10. }
11. }
12. **</script>**
13. **<body>**
14. **<form** name="myform"  method="post" action="#" onsubmit="return validateemail();"**>**
15. Email: **<input** type="text" name="email"**><br/>**
17. **<input** type="submit" value="register"**>**
18. **</form>**

[**Test it Now**](http://www.javatpoint.com/oprweb/test.jsp?filename=jsvalidation3)

Visit here: [JavaScript Email validation](https://www.javatpoint.com/javascript-form-validation#email).

48) What is this keyword in JavaScript?

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

1. var address=
2. {
3. company:"Javatpoint",
4. city:"Noida",
5. state:"UP",
6. fullAddress:function()
7. {
8. return this.company+" "+this.city+" "+this.state;
9. }
10. };
11. var fetch=address.fullAddress();
12. document.writeln(fetch);

49) 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

50) 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:

1. function display()
2. {
3. x = 10;
4. y = 15;
5. z = x + y;
6. debugger;
7. document.write(z);
8. document.write(a);
9. }
10. display();

51) 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:

1. "use strict";
2. x=10;
3. console.log(x);

52) 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:

1. function display()
2. {
3. document.writeln(Math.random());
4. }
5. display();

53) 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.

1. function display()
2. {
3. var date=new Date();
4. var day=date.getDate();
5. var month=date.getMonth()+1;
6. var year=date.getFullYear();
7. document.write("**<br>**Date is: "+day+"/"+month+"/"+year);
8. }
9. display();

54) 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.

1. function display()
2. {
3. var x=102;//integer value
4. var y=102.7;//floating point value
5. var z=13e4;//exponent value, output: 130000
6. var n=new Number(16);//integer value by number object
7. document.write(x+" "+y+" "+z+" "+n);
8. }
9. display();

55) 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.

1. function display()
2. {
3. document.writeln(10**<20**);//true
4. document.writeln(10**<5**);//false
5. }
6. display();

56) 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.

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

57) 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:

1. function display()
2. {
3. var set = new Set();
4. set.add("jQuery");
5. set.add("AngularJS");
6. set.add("Bootstrap");
7. for (let elements of set) {
8. document.writeln(elements+"**<br>**");
9. }
10. }
11. display();

58) 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:

1. function display()
2. {
3. var ws = new WeakSet();
4. var obj1={};
5. var obj2={};    “”
6. ws.add(obj1);
7. ws.add(obj2);
8. //Let's check whether the WeakSet object contains the added object
9. document.writeln(ws.has(obj1)+"**<br>**");
10. document.writeln(ws.has(obj2));
11. }
12. display()

59) 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:

1. function display()
2. {
3. var map=new Map();
4. map.set(1,"jQuery");
5. map.set(2,"AngularJS");
6. map.set(3,"Bootstrap");
7. document.writeln(map.get(1)+"**<br>**");
8. document.writeln(map.get(2)+"**<br>**");
9. document.writeln(map.get(3));
10. }
11. display();

60) 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:

1. function display()
2. {
3. var wm = new WeakMap();
4. var obj1 = {};
5. var obj2 = {};
6. var obj3 = {};
7. wm.set(obj1, "jQuery");
8. wm.set(obj2, "AngularJS");
9. wm.set(obj3,"Bootstrap");
10. document.writeln(wm.has(obj2));
11. }
12. display();