JAVA SCRIPT

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

A variable is undeclared when it does not use the var keyword. It gets created on the global object (that is, the window), thus it operates in a different space as the declared variables.

Something is undefined when it hasn’t been defined yet. If you call a variable or function without having actually created it yet the parser will give you an not defined error.

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

A variable that does not use the var keyword. It gets created on the global object (that is, the window), thus it operates in a different space as the declared variables.

They clutter up the global namespace and are slower to look up than local variables.

First of all, having many global variables is always a bad thing because it's easy to forget you declared a variable somewhere and accidentally re-declare it somewhere else. If your first variable was local then you don't have a problem. If it was global, then it just got overwritten. This gets even worse when you get into implied globals (e.g. when you say someVar = someValue without declaring someVar with the var keyword).

Secondly, global variables take longer for Javascript to "find" than local variables. The difference in speed isn't huge, but it does exist.

**3.What do mean by NULL in Javascript?**

Null is where the thing is known to exist, but it's not known what the value is.

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

[window.onload](https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers.onload)

* By default, it is fired when the entire page loads, including its content (images, css, scripts, etc.)
* In some browsers it now takes over the role of document.onload and fires when the DOM is ready as well.

Document.ready (a jQuery event) will fire when all the elements are in place, and they can be referenced in the JavaScript code, but the content is not necessarily loaded. Document.readyexecutes when the HTML document is loaded.

**5.What are global and local variables ?**

The variables declared within a function are known as local variables. Their scope is within the function. The variables declared outside the function are known as global variables. The can accessed globally.

**5.What are the ways to create the objects**

Define and create a single object, using an object literal.

Define and create a single object, with the keyword new.

Define an object constructor, and then create objects of the constructed type.

**6.How many ways we can create the arrays.**

Using an array literal is the easiest way to create a JavaScript Array.

Using the JavaScript Keyword new

**7.What are arguments in javascript functions.**

Function **arguments** are the real **values** passed to (and received by) the function.

**8.What is prototypal inheritance in javascript**

In JavaScript, a **prototype** is a property of functions and of objects that are created by constructor functions. The prototype of a function is an object. Its main use is when a function is used as a constructor. The **prototype** object can be used to derive one object from another. For example, you can use the [Object.create](https://msdn.microsoft.com/en-us/library/ff925952(v=vs.94).aspx) function to derive a new object using the prototype of some other object.

**9.What are enumerators in javascript.**

The enumerator object is useful if you want to iterate through a large collection. Iterating through a large collection with the enumerator object is faster than any other way in javascript. It allows using a walk- through mechanism to iterate through the items. There is a pointer, which designates he currently selected element and can be used to navigate within the enumerator object. In a newly created enumerator object., the currently selected item is the first item of the collection.

**10.Callbacks and closures.**

A callback function, also known as a higher-order function, is a function that is passed to another function (let’s call this other function “otherFunction”) as a parameter, and the callback function is called (or executed) inside the otherFunction. A callback function is essentially a pattern (an established solution to a common problem), and therefore, the use of a callback function is also known as a callback pattern.

A closure is an inner function that has access to the outer (enclosing) function’s variables—scope chain. The closure has three scope chains: it has access to its own scope (variables defined between its curly brackets), it has access to the outer function’s variables, and it has access to the global variables. The inner function has access not only to the outer function’s variables, but also to the outer function’s parameters. Note that the inner function cannot call the outer function’s argumentsobject, however, even though it can call the outer function’s parameters directly. You create a closure by adding a function inside another function.

**11.Module based programming in javascript**.

Modules divide programs into clusters of code that, by some criterion, belong together. This chapter explores some of the benefits that such division provides and shows techniques for building modules in JavaScript. The benefits of organizing a program into several files or modules are similar. Structure helps people who aren’t yet familiar with the code find what they are looking for and makes it easier for the programmer to keep things that are related close together.

**12.What is strict mode in javascript**

The "use strict" directive is new in JavaScript 1.8.5 (ECMAScript version 5).It is not a statement, but a literal expression, ignored by earlier versions of JavaScript.The purpose of "use strict" is to indicate that the code should be executed in "strict mode".With strict mode, you can not, for example, use undeclared variable.

**13.what is the difference between == and ===**

== is just comparing the two values, and if they are of different types, type conversion is done

=== compares the values and well as their types - so no type conversion will be done here.

**14.Ternary operator**.

The **conditional (ternary) operator** is the only JavaScript operator that takes three operands. This operator is frequently used as a shortcut for the [if](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/if...else) statement.

Syntax

Condition ? expr1 : expr2

### **Parameters**

**condition**

An expression that evaluates to true or false.

**expr1, expr2**

Expressions with values of any type.

**15.Difference between public, private and static variables and their use cases.**

A variable declared outside a function, becomes **GLOBAL**.

A global variable has **global scope**: All scripts and functions on a web page can access it.

Private members are made by the constructor. Ordinary vars and parameters of the constructor becomes the private members.

A function is an object. That provides us with a nifty way to create static variables or, in other words, the variables which persist along multiple calls. There are languages which allow to put a static keyword before a variable, and then such variable is not cleared in next calls.

**16. Enumerate the differences between Java and JavaScript?**

Java is a complete programming language. In contrast, JavaScript is a coded program that can be introduced to HTML pages. These two languages are not at all inter-dependent and are designed for the different intent.  Java is an object – oriented programming (OOPS) or structured programming language like C++ or C whereas JavaScript is a client-side scripting language and it is said to be unstructured programming.

**17. What are JavaScript types?**

Following are the JavaScript types:

* Number
* String
* Boolean
* Function
* Object
* Null
* Undefined

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

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

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

‘ViewState’ is specific to a page in a session.

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

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

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

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

If you use innerHTML in JavaScript the disadvantage is

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

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

There are three types of errors:

* **Load time errors**: Errors which come up when loading a web page like improper syntax errors are known as Load time errors and it generates the errors dynamically.
* **Run time errors**: Errors that come due to misuse of the command inside the HTML language.
* **Logical Errors**: These are the errors that occur due to the bad logic performed on a function which is having different operation.