

Java VS Javascript

Java is a programming language, in which the code will get executed sequentially.	In Javascript the program is event driven, the execution of code is dependent on user action
Java is statically type, means the data type of variable is decided at compile time	In javascript type of variables is decided at run time, based on the value you have assigned to the variable
Java is Object oriented.	Javascript is object based.

To declare a variable in javascript, you can use keyword let, var and const

let	var	const
duplicate declaration is not allowed	duplicate declaration is allowed	duplicate declaration is not allowed
let is a blocked scope variable	var is global or function scope variable	const is blocked scope variable
hoisting of variable declaration is not allowed	hoisting of variable declaration is allowed	histing of variable declaration is not allowed
let v;	var v;	const c=34; Initialization of variable is needed, at the time of declaration.

In client side javascript

1. Client side data validation is possible
2. Give dynamic look and feel to the page
3. AJAX(Asynchronous Javascript and XML) requests can be send by javascript, by using XMLHttpRequest(XHR) object

The Javascript code can be executed with the help of HTML, and is called as Client side javascript code.

Or it can be executed with the help of Nodejs, which is server side code.

To display popup boxes in javascript, there are 3 types of boxes, and all these functions are of window objects

Popup boxes	
alert	A box with only one ok button, useful to show messages to user
confirm	A box with 2 buttons ok and cancel, if user clicks on ok button then it returns true, otherwise, if user clicks on cancel button, then it returns false
prompt	A box with a text box, useful to accept one input from user

To display ouput to user in browser window, document.write(“Hello world!!”)

To display messages for testing purpose the use console.log(“Test messages”);

Ways to write functions

<pre>const f3=(x,y=3,...s)=>{ //rest parameter document.write("in f3",x,y,"
") } f1(1,2,3,4) f2(1,2) f3(1,2,3,4)</pre>	function using => operator, you may assign default values to parameters
<pre>function f1(x,...arr){ document.write("in f1",x,"
"); document.write("test:" +arr.length,arguments[0]); }</pre>	... rest parameters and useful for variable number of arguments
<pre>const f2=function(x,y){ document.write("in f2",x,y,"
"); } //var f2=45; const f4=(x,y=45,z=4)=>{ console.log(x,y,z); }</pre>	better to use const keyword while declaring function, so that definition will not get overwritten
<pre>function f1(){ function f2(){ } }</pre>	Closure function → the function which has access to its parent scope, is called as closure function
<pre>(function(x,y){ document.write("in function f1",x,y) })(12,13)</pre>	Self calling function, If you want to execute the function definition only once, then use self calling function

functions in javascript for typecasting

parseInt	to convert the string into integer
parseFloat	To convert the string into float

The tags which has innerHTML property → div, span, p, a,

The tags which are not used to accept data from user via form, use innerHTML property

but the tags which are used in the form to accept data from user has value property

functions on numeric data

isNaN(data)	if the data contains atleast one character, then it returns true, otherwise it returns false
-------------	--

isFinite(data)	if the data contains all digits between 0-9, then it returns true, otherwise it returns false
----------------	---