A Comparison of VbScript and JavaScript: Insight on Selection and Future Enhancements

S. Sri Venkatesh
Undergraduate student, Department of Computer Science & Engineering
Panimalar Engineering College
Chennai, India.

ABSTRACT

Scripting languages such as JavaScript and VBScript are most widely used for developing web pages that are interactive in nature. JavaScript is a programming language that was developed by Netscape Communications Corporations. VBScript is a subset of the Visual Basic programming language by Microsoft that is supported by Microsoft Internet Explorer. These scripts require a Web browser or an interpreter in order to run. In this paper a brief comparison is made between the features of VBScript and JavaScript scripting languages. This paper will provide a acumen into different features of these languages and assist in making an prudent decision and also facilitate the researchers in the development of new and better scripting languages or extending the features of existing scripting languages.

Keywords

Option Explicit, Dim, Var, Rem, Call By Reference, ByVal, ByRef, Classes, Objects, Built-in functions, Cross-platform support.

1. INTRODUCTION

HTML- HyperText Markup Language was designed to display document information on web pages. HTML used a method called, Common Gateway Interface (CGI) scripting in order to enable the Web browsers to send information to the server and allow the server to perform processing of that information. The results from the server after processing were also sent in HTML format. This even though introduced interactivity on the Web also considerably increased the traffic between the browser and the server. The additional disadvantage of this kind of processing was that the CGI scripts were stored and processed on the server which added to the work load of a server. A solution to this problem came in the form of scripts which enabled the browsers on the client side to do most of the processing.

Netscape Communications Corporations in the year 1995 invented the LiveScript which can be run on Netscape browsers. The LiveScript was later changed to JavaScript to capitalize on the success of Java. Soon after the release of JavaScript, Microsoft also introduced its own scripting language known as VBScript in 1996 [1]. The VBScript is a subset of the Visual Basic programming language which is only supported by the Internet Explorer. This paper compares the features of VBScript and JavaScript in various perspectives.

2. LEARNABILITY

The VBScript is syntactically similar to the Visual Basic and Visual Basic for Applications (VBA). VBScript looks less daunting and is also easily understandable because of its resemblance to simple English [2]. The bright side of VBScript is the absence of braces, punctuations such as semicolons which make the language less difficult to debug. The VBScript is the primary scripting language to learn for people who do not have any programming knowledge or for someone who already have knowledge on Visual Basic.

The JavaScript on the other hand share some basic syntax with Java, C, and C++. A common misconception is that JavaScript is similar or closely related to Java. Even though JavaScript follow some syntax of Java it is still a comparatively new language. JavaScript is easier for people who have already learned C or C++. Samples of coding to display “Factorial of 5” are given below for both scripting languages. [4][5][6]

Table 1: Comparison of VBScript and JavaScript through “Factorial of 5” example

VBSCRIPT	JAVASCRIPT

Dim x,f
X=5
f=1
For i=1 To x
  f=f*i
Next
MsgBox "The Factorial of " & x & " is " & f
	Var x=1,f=0;
	for(i=1;i<=n;i++)
	{
	  f=f*i;
	}
	alert(“Factorial of “ + x + “ is ” + f);

3. CASE SENSITIVITY AND VARIABLE DECLARATION

VBScript is not a case-sensitive language, i.e., the names 'counter' and 'Counter' correspond to the same variable, making it a programmer-friendly language [6]. A declaration of variable in VBScript can be either implicit by just using it or explicit through Dim keyword. A powerful statement called Option Explicit is one of the most significant features in VBScript and also helps in creating a bug-free code. By placing Option Explicit at the top of a program the VBScript can force all the variables in the program to be declared. The Option Explicit statement allows the VBScript to behave as a case-sensitive language in terms of variable declaration. Unlike VBScript, JavaScript is a case-sensitive language. In JavaScript it can either be implicitly declared using a Var keyword or explicitly by assigning a value to it. JavaScript does not force the variables to be always declared. VBScript handles typing errors more efficiently than JavaScript. The optional declaration and case-sensitive nature of JavaScript makes it less reliable than VBScript.

4. COMMENT LINES

In JavaScript, a single-line comment should be specified using a pair of forward slashes (//) and a multi-line comment must start with forward slash and asterisk (/*) and end with the same in reverse (*/) [7]. The multi-line comment specification can also be used for a single line comment line. The bright side with JavaScript is that comments are handled exactly the same way as they are in C, C++, and Java. In VBScript, a single-line comment is specified either by using a single quotation mark (') or by using a Rem statement [1]. While placing the Rem statement for comments in a line which already has code in it, the Rem should be preceded by a colon (:). The main disadvantage of VBScript is that there is no specific way to specify a multi-line comment. However, we can specify multi-line comments in VBScript by associating each comment line by a single line comment specification, i.e., (') or Rem. The comment handling feature of JavaScript is better than VBScript.

5. CONSTANT DECLARATION

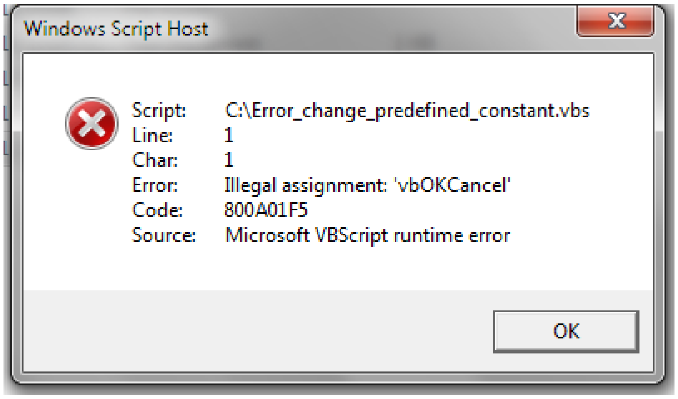
VBScript allows constant values to be declared with the help of a 'Const' statement. A constant is an expression whose value cannot be changed and remains fixed. The VBScript also contains a number of pre-defined constants. The below samples shows a constant declaration for PI (Π) and a pre-defined constant vbOKCancel of MsgBox,

```
Const PI = 3.14
MsgBox "This is a message box", vbOKCancel
```

The above MsgBox statement will display message box with OK and Cancel buttons. On trying to change the value of vbOKCancel an error message is generated. The below example shows the erroneous code that arises due to the attempt to change a predefined value,

```
vbOKCancel  =  6
MsgBox "This is a message box", vbOKCancel
```

The output of such a code will be as shown in below figure,



```
Script:   C:\Error_change_predefined_constant.vbs
Line:     1
Char:     1
Error:    Illegal assignment: 'vbOKCancel'
Code:     800A01F5
Source:   Microsoft VBScript runtime error
```

Fig 1: Error generated by the Windows Script Host when attempting to change the value of an predefined constant.

JavaScript unlike the VBScript script does not have any feature for constant declaration. Even the values of pre-defined constant in a JavaScript can be modified. This is a major drawback for JavaScript. By allowing constant declaration through 'Const' statement and also by guarding against modifications of pre-defined constants VBScript scores over JavaScript.

6. OPERATORS

JavaScript supports a wide variety of operators such as arithmetic operators, comparison operators, string operators, comparison operators, logical operators and bitwise operators. All these set of operators are also supported by VBScript. The difference lies in few specific, but still essential operators. The below table represents the list of operators that are different in both the scripting languages [8][9][10],

Table 2: List of operators supported by both the languages.

OPERATOR	VBSCRIPT	JAVASCRIPT
Increment & Decrement Operator (++ & --)	NO	YES
Conditional/Ternary (?:)	NO	YES
Strict Equality & Strict Inequality (=== & !==)	NO	YES
Shift Operators (<<,>>,>>>)	NO	YES
Shortcut/Compound Assignments	NO	YES
Exponentiation (^)	YES	NO
Equivalent (Eqv)	YES	NO
Logical Implication (Imp)	YES	NO
Integer Division (\)	YES	NO
Is Operator (is)	YES	NO

Special Operators (delete, void, new, in etc.)	NO	YES

Apart from the above set of operators, the operator that distinguishes both the languages is the String Concatenation Operator. JavaScript uses the same operator (+) [11] for Addition and String Concatenation whereas VBScript uses (+) for Addition and (&) for Sting Concatenation. The usage of same operator for both Addition and String Concatenation by JavaScript introduces a new source of error in JavaScript. The absence of Increment/Decrement operator and Shortcut/Compound Assignment Operators makes VBScript less favorable than JavaScript.

7. CALL BY REFERENCE

There are two types of value passing when calling a function, Call By Value and Call By Reference. VBScript supports both Call By Value and Call By Reference through two statements call ByVal and ByRef [12]. The JavaScript does not support Call By Reference and only supports Call By Value similar to Java. The Objects in JavaScript are by default passed by reference. The formal parameter is just a mere copy of actual parameter and when an object is passed as an argument, both the actual and formal parameters are copies of that object and by the property that in JavaScript an object and its copy refer to the same memory location an object by default is passed by reference. The below example shows how a swap program is implemented in both the scripting languages,

Table 3: Swap program implemented in both the languages.

VBSCRIPT	JAVASCRIPT

Option Explicit
Sub Swap(ByRef a, ByRef b)
Dim t
t = a
a = b
b = t
End Sub
Dim a, b
a = 5
b = 6
MsgBox "Before Swapping "& a & ", " & b
call Swap(a, b)
MsgBox "After Swapping "& a & ", " & b
	function swap()
	{
	t = a;
	a = b;
	b = t;
	}
	var a, b;
	a = 3;
	b = 4;
	alert("Before Swapping "+ a + ", " + b);
	swap(a, b);
	alert("After Swapping "+ a + ", " + b);

By having explicit keywords for both Call By Value and Call By Reference VBScript scores over JavaScript.

8. BUILT-IN FUNCTIONS

There are numerous Built-in Functions available to VB Script that is not native to JavaScript. All of these functions are instantly available and can be used in a coding. The VBScript provides built-in functions for date, time, math, array, string, data type conversion etc. JavaScript supports built-in functions for date, math, string, array etc. The difficulty in JavaScript is that nearly every function we wish to use we must first create it. One very good example is the usage of date and time functions. The following code is used to return and display a simple string containing the current date using both scripting languages,

Table 4: Function to display date and time in both the languages

VBSCRIPT	JAVASCRIPT [13]

MsgBox GetDate()
Function GetDate
Dim MyMonthName
MyMonthName = MonthName(Month(Now), False)
GetDate = MyMonthName & " " & Day(Now) & ", " & Year(Now)
End Function
	alert(GetDate());
	function GetDate()
	{
	var MyMonthName, d;
	d = new Date();
	switch(d.getMonth() + 1)
	{
	case 1: MyMonthName = "January ";
	break;
	case 2: MyMonthName = "February ";
	break;
	case 3: MyMonthName = "March ";
	break;
	case 4: MyMonthName = "April ";
	break;
	case 5: MyMonthName = "May ";
	break;
	case 6: MyMonthName = "June ";
	break;
	case 7: MyMonthName = "July ";
	break;
	case 8: MyMonthName = "August ";
	break;
	case 9: MyMonthName = "September ";
	break;
	case 10: MyMonthName = "October ";
	break;
	case 11: MyMonthName = "November ";
	break;
	case 12: MyMonthName = "December ";
	}

	return(MyMonthName + d.getDate() + ", " + d.getFullYear())
	}

It is not hard to notice the length difference between these 2 pieces of code. It is almost impossible to believe that they perform the same function. The VB Script example uses built-in functionality by calling the Month(Now) and Day(Now) functions, while the JavaScript example forces to actually create that functionality before using it. It is clear that VBScript has a rich set of built-in function than the JavaScript.

9. OBJECT CREATION

Objects are the basic amenities of Object Oriented Programming. Objects are collections of related properties and methods. A property is attribute or value and a method is a function over that property. Both VBScript and JavaScript have several built-in objects to perform variety of tasks. In VBScript we can create a class and objects corresponding to that class can also be created [6]. A class is created in VBScript with data members and member functions. This class creation of VBScript resembles class creation in Java. On the other hand, JavaScript does not have the concept of classes. However we are able to create objects in JavaScript. JavaScript involves creating of a constructor with properties and methods and then creation of objects through that constructor using new operator. The object creation exhibits the dynamic nature of JavaScript. It is comparatively hard to debug this kind of dynamic code in JavaScript then VBScript. This makes VBScript more reliable than JavaScript.

10. BROWSERS

JavaScript is the default scripting language for all web browsers. JavaScript also provides cross-platform support from all popular web browsers, this makes the language more favorable one. Whereas VBScript must be specified as the scripting language for browsers and VBScript does not provide cross-platform support, i.e., it is supported only by Microsoft Internet Explorer (MS IE). This is the major drawback of VBScript and the same makes JavaScript most popular and used scripting language. The following survey report depicts the percentage usage of various browsers,

Table 5: Survey Report on usage of various browsers.

Survey by [14]	**Period**	**Internet Explorer**	**Chrome**	**Firefox**
StatCounter	January-2012	37.45%	28.40%	24.78%
	January-2011	46.00%	15.68%	30.68%
W3C [15]	January-2012	20.1 %	37.1 %	35.3 %
	January-2011	26.6 %	42.8 %	23.8 %
W3C	January-2012	20.1 %	37.1 %	35.3 %
	January-2011	40.0%	31.1%	14.2%
StarOwl.com	January-2012	51.81%	18.98%	16.77%
	January-2011	61.28%	19.57	8.80%

It is clear from the table above that, the usage of MS IE is in decline over a year and this act in favour of JavaScript. The JavaScript by providing cross-platform support makes it the most popular scripting language at present [16].

11. CONCLUSION

Any programming language cannot be regarded as a complete failure. It is also difficult to say that one language is better than the other. Even here between JavaScript and VBScript, we cannot say which one is better because each of them have their own merits and demerits. But, what can be done is that we can choose the best fitting script language for our usage. For business applications VBScript is the best scripting language because VBScript constants and built-in functions which reduce code complexity are well suited for business applications. The cross-platform support and better object oriented capabilities regard JavaScript as most appropriate scripting language for commercial applications.

Microsoft is working with the World Wide Web Consortium to develop a standard for ActiveX Scripting and the event-driven model. Since IE is integrated with Windows the most famous Operating System developed by Microsoft, VBScript may take centre stage in Windows development as the browser becomes the primary desktop interface. Work is ongoing at Microsoft to completely unify the desktop environment inside the browser and to create a page view. Implementing page view will radically alter the Windows desktop environment. First and foremost, the concept of a separate browser running on a desktop is eliminated. In fact, the browser will be the desktop. This proves to provide a bright future for the VBScript.

12. REFERENCES

[1] "A Comparison of VBScript,JavaScript and JScript" by Kailash Chandra, Sapana Suhani Chandra, JCSC 19,1 October 2003.

[2] http://www.irt.org/articles/js117/

[3] Nicholas C. Zakas "Professional JavaScript for Web Developers", 2nd ed. Wrox

[4] Alexei White, "JavaScript Programmer's Reference", 1st ed. Wrox..

[5] Jeffrey C. Jackson "Web Techonologies: A Computer Science Perspective", 1st ed., Prentice Hall.

[6] Adrian Kingsley-Hughes, Kathy Kingsley-Hughes, Daniel Read, "VBScript Programmer's Reference" ,3rd ed. Wrox.

[7] https://profiles.google.com/103620921493894370957/buzz/WqbpncfWv2N

[8] http://www.coveryourasp.com/Snippet.asp?snip=23

[9] http://jaliyaudagedara.blogspot.in/2011/03/java-script-vs-jscript-vs-vb-script.html

[10] http://www.harding.edu/fmccown/scriptcompare.html

[11] http://www.itproportal.com/2010/06/28/comparison-javascript-and-vbscript/

[12] http://webcoder.info/reference/vbsvsjs.html

[13] http://www.heferito.com/javascriptVSvbscript.asp

[14] http://en.wikipedia.org/wiki/Usage_share_of_web_browsers

[15] http://www.w3schools.com/browsers/browsers_stats.asp

[16] http://msdn.microsoft.com/en-us/library/ms970435.aspx