

VBSCRIPT - SYNTAX

Your First VBScript

Let us write a VBScript to print out "Hello World".

```
<html>
<body>
<script language="vbscript" type="text/vbscript">
  document.write("Hello World!")
</script>
</body>
</html>
```

In the above example, we called a function *document.write*, which writes a string into the HTML document. This function can be used to write text, HTML or both. So, above code will display following result:

```
Hello World!
```

Whitespace and Line Breaks

VBScript ignores spaces, tabs and newlines that appear within VBScript programs.

Because one can use spaces, tabs and newlines freely within the program so you are free to format and indent your programs in a neat and consistent way that makes the code easy to read and understand.

Formatting

VBScript is based on Microsoft's Visual Basic. Unlike JavaScript, no statement terminators such as semicolon is used to terminate a particular statement.

Single Line Syntax

Colons are used when two or more lines of VBScript ought to be written in a single line. Hence, in VBScript, Colons act as a line separator.

```
<script language="vbscript" type="text/vbscript">
  var1 = 10 : var2 = 20
</script>
```

Multiple Line Syntax

When a statement in VBScript is lengthy and if user wishes to break it into multiple lines, then the user has to use underscore "_".

This improves the readability of the code. The following example illustrates how to work with multiple lines.

```
<script language="vbscript" type="text/vbscript">
  var1 = 10
  var2 = 20
  Sum = var1 + var2
  document.write("The Sum of two numbers"&_
  "var1 and var2 is " & Sum)
</script>
```

Reserved Words

The following list shows the reserved words in VBscripting. These reserved words SHOULD NOT be used as a constant or variable or any other identifier names.

| | | |
|-----------------|------------|---------------|
| Loop | LSet | Me |
| Mod | New | Next |
| Not | Nothing | Null |
| On | Option | Optional |
| Or | ParamArray | Preserve |
| Private | Public | RaiseEvent |
| ReDim | Rem | Resume |
| RSet | Select | Set |
| Shared | Single | Static |
| Stop | Sub | Then |
| To | True | Type |
| And | As | Boolean |
| ByRef | Byte | ByVal |
| Call | Case | Class |
| Const | Currency | Debug |
| Dim | Do | Double |
| Each | Else | ElseIf |
| Empty | End | EndIf |
| Enum | Eqv | Event |
| Exit | False | For |
| Function | Get | GoTo |
| If | Imp | Implements |
| In | Integer | Is |
| Let | Like | Long |
| TypeOf | Until | Variant |
| Wend | While | With |
| Xor | Eval | Execute |
| Msgbox | Erase | ExecuteGlobal |
| Option Explicit | Randomize | SendKeys |

Case Sensitivity

VBScript is a case-insensitive language. This means that language keywords, variables, function names and any other identifiers need NOT be typed with a consistent capitalization of letters.

So identifiers int_counter, INT_Counter and INT_COUNTER have the same meaning within VBScript.

Comments in VBScript

Comments are used to document the program logic and the user information with which other programmers can seamlessly work on the same code in future.

It can include information such as developed by, modified by and it can also include incorporated logic. Comments are ignored by the interpreter while execution.

Comments in VBScript are denoted by two methods.

1. Any statement that starts with a Single Quote ‘ is treated as comment. Following is the example:

```
<script language="vbscript" type="text/vbscript">
<!--
  ' This Script is invoked after successful login
  ' Written by : TutorialsPoint
  ' Return Value : True / False
//-->
</script>
```

2. Any statement that starts with the keyword “REM”. Following is the example:

```
<script language="vbscript" type="text/vbscript">
<!--
  REM This Script is written to Validate the Entered Input
  REM Modified by : Tutorials point/user2
//-->
</script>
```

Loading [MathJax]/jax/output/HTML-CSS/jax.js