



## Unit 9

### Scripting

#### Introduction

In the early age of World Wide Web, Hypertext Markup Language was the language that could be used to create simple web pages. HTML creates static web pages that have fixed contents (text, audio, and graphics). It does not allow user to interact with the web page. Scripting language lets to create dynamic and active web page that can display dynamic information such as display an appropriate greeting like "Good Morning" or "Good Evening" depending on the time, making web page interactive. Lets web page take input from the user and check the data to make sure it is valid or meets certain criteria and much more. VB Script is short for Visual Basic Script. VB Script is a scripting language, a subset of Visual Basic, by Microsoft, intended to be used with Microsoft products

## Lesson 9.1

### VB Script-Basics



#### Outcomes

Upon completion of this unit you will be able to:

*Define* Scripting.

*Write* VBScript.

*Use* VBScript in HTML document.

### Defining Script

In computer programming, 'Script' is a program or sequence of instruction that is interpreted. This means that a program built with a scripting language must be run in the environment that contains the Script Language Interpreter (Interpreter: software program that helps to interpret the script line by line for the computer to understand and get the result).

Broadly scripts are categorized into two types:

- Client-side scripting
- Server-side scripting

#### **Client-side scripting**

Client-side scripting generally refers to the class of computer programs that are executed at client-side, by the user's web browser. JavaScript and VBScript are two widely used client-side scripts.



### Server-side scripting

Server-side scripting is a program that executes at server end. It is usually used to provide interactive web sites that interface to databases or other data stores. Some popular server-side scripting language are: ASP (Active Server Pages), JSP (Java server Pages), PHP, Python etc.

## VB Script

- VBScript is a Microsoft scripting language.
- VBScript is a light version of Microsoft's programming language Visual Basic.
- VBScript is the default scripting language in ASP (Active Server Pages).
- VBScript is inserted into HTML, between standard <script> and </script> tags.
- Use the type attribute in the <script> tag to define the scripting language "vbscript".
- VBScript is supported by **Internet Explorer only**.
- VBScript can play an important role in many ways like validating data, providing interactive responses etc.

## VB Script Output

- When VBScript is used on a web server in an ASP page, the statement **response.write()** produces output.
- When we use Internet Explorer for testing VBScript, we use **document.write()** to produce output.

### Example

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

## VBScript Editor

Internet Explorer supports VBScript. This allows us to experiment with VBScript in the browser, without the need for a web server.

## Writing VBScript

In this section we will write our first VBScript code and run the same in Internet Explorer. We write the VBScript code in an HTML file and save the same with .htm or .html extension. Let us write a VBScript to display "Welcome to my first VBScript".

```
<html>
<head>
<title>VBScript first</title>
</head>
<body>
<script type="text/vbscript">
```



```
document.write("Welcome to my first VBScript")  
</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

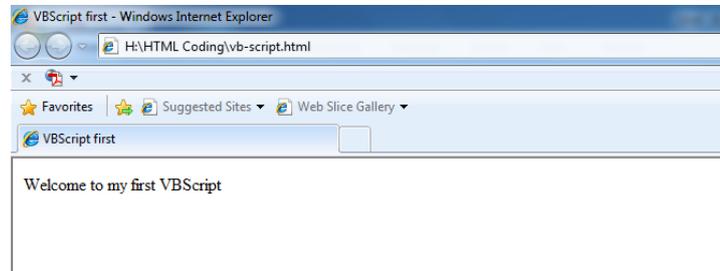


Figure 1.1 VB Script



---

## Assessment



Assessment

### Fill in the blanks

1. In computer programming, 'Script' is a program or \_\_\_\_\_.

State whether the following statements are true or false

1. VBScript is supported by Internet Explorer only.

### Multiple Choice questions

1. VBScript is the default scripting language in

- a) ASP
- b) JSP
- c) PHP
- d) All of above

### Exercises

1. What is VB Script?
2. Differentiate between Client Side Scripting & Server Side Scripting.
3. How can we add VBScript in HTML?
4. Which Tag and Attribute is used to add VB script in html?



## Lesson 9.2

### VB Script-variables, Data Types

Upon completion of this unit you will be able to:



#### Outcomes

*Define* variables.

*Describe* data types.

### Variables

VBScript variables are used to hold values or expressions. A variable can have a short name, like `x`, or a more descriptive name, like `studentname`. Creating variables in VBScript is most often referred to as "declaring" variables. You can declare VBScript variables with the `Dim`, `Public` or the `Private` statement. Like this:

```
Dim x
```

```
Dim studentname
```

You can declare multiple variables by separating each variable name with a comma. For example:

```
Dim name, age, phoneno
```

### Assigning Values to Variables

You can assign a value to a variable like this:

```
name="Mamun"
```

```
age=31
```

The variable name is on the left side of the expression and the value you want to assign to the variable is on the right. Now the variable "name" has the value of "Mamun", and the variable "age" has the value of "31".

### Rules for variable names

- Must begin with an alphabetic character
- Cannot contain a period (.)
- Cannot exceed 255 characters
- Must be unique in the scope in which it is declared.

### Array Variables

Much of the time, you only want to assign a single value to a variable you have declared. Array variables are declared in the same way, except that the declaration of an array variable uses parentheses ( ) following the variable name. In the following example, a single-dimension array containing 11 elements is declared:



Dim A(10)

Although the number shown in the parentheses is 10, all arrays in VBScript are zero-based, so this array actually contains 11 elements.

Arrays aren't limited to a single dimension. You can declare multiple dimensions by separating an array's size numbers in the parentheses with commas. In the following example, the table variable is a two-dimensional array consisting of 4 rows and 6 columns:

Dim table (3, 5)

In a two-dimensional array, the first number is always the number of rows; the second number is the number of columns.

## Data Type

VBScript has only one data type called a Variant. A Variant is a special kind of data type that can contain different kinds of data, depending on how it is used. At its simplest, a Variant can contain either numeric or string data. A Variant behaves as a number when you use it in a numeric context and as a string when you use it in a string context. That is, if you are working with data that looks like numbers, VBScript assumes that it is numbers and does what is most appropriate for numbers. Similarly, if you're working with string data, VBScript treats it as string data. You can always make numbers behave as strings by enclosing them in quotation marks (" ").

### Variant Subtypes

Beyond the simple numeric or string classifications, a Variant can make further distinctions about the specific nature of numeric information. For example, you can have numeric information that represents a date or a time. When used with other date or time data, the result is always expressed as a date or a time. You can also have a rich variety of numeric information ranging in size from Boolean values to huge floating-point numbers. These different categories of information that can be contained in a Variant are called subtypes. Most of the time, you can just put the kind of data you want in a Variant, and the Variant behaves in a way that is most appropriate for the data it contains.



## Assessment



Assessment

### Fill in the blanks

1. VBScript has only one data type called a \_\_\_\_\_.

### State whether the following statements are true or false

1. You can make numbers behave as strings by enclosing them in quotation marks (" ").

### Multiple Choice questions

1. Which one of the following is a correct example multiple dimensions array?
  - a) Dim table 4,2
  - b) Dim table (4,2)
  - c) Dim table (4),(2)
  - d) Dim table {4,2}

### Exercises

1. What is Variant Type? Name the Sub Type of Variant.



## Lesson 9.3

### VBScript-Function, Operators



#### Outcomes

Upon completion of this unit you will be able to:

*Create* function.

*Call* function.

*Describe* operators.

### VBScript Function

VBScript function is a series of statements, enclosed by the Function and End Function keyword which can perform actions and can return a value. VBScript allow us to reuse our code again and again. For example we can write a function that performs the addition of two numbers. After writing the function whenever we need the functionality of the addition of two numbers, we can call our add function.

Here are the steps to write VBScript functions:

1. Mark the start of the function with the Function keyword.
2. Give your function a meaningful name.
3. Follow the function name with a pair of round brackets.
4. If the function is expecting external information to perform its task, add the name(s) for the external information, also called arguments. This is optional.
5. Starting on a new line, write the code that makes up the function.
6. Finish with the End Function keywords.

Here is a simple example that requires two numbers as arguments and sums them both

```
Function add(num1, num2)
add = num1 + num2
End Function
```

### Calling Functions

A function doesn't do anything until it is called. Calling a function in VBScript is very easy. We call a function by writing its name followed by a pair of brackets. Here is a simple example of calling our add function, above:

```
add(10, 5)
```



In the above line of code we call our add function, passing in two values, also called parameters.

**Here is an example**

```
<html>
<head>
<title>Test VBScript</title>
</head>
<body>
<script type="text/vbscript">
Function add(num1, num2)
add = num1 + num2
End Function
Dim total
total = add(7, 5)
document.write("The Total of 7 and 5 is: " & total)
</script>
</body>
</html>
```

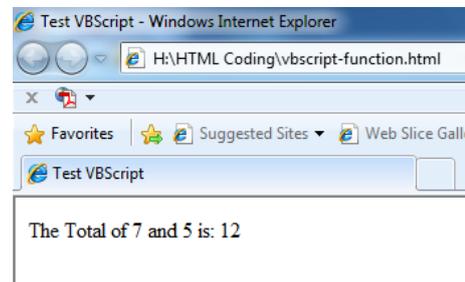


Figure 3.1 Function

## Creating message box

Creating a message box in VBScript is very easy; we only need to supply the MsgBox function with a string. Here is an example that pops a Message Box with the words "okonly":

```
<html>
<head>
<title>VBScript MessageBox</title>
</head>
<body>
<script type="text/vbscript">
MsgBox "okonly"
</script>
</body>
</html>
```

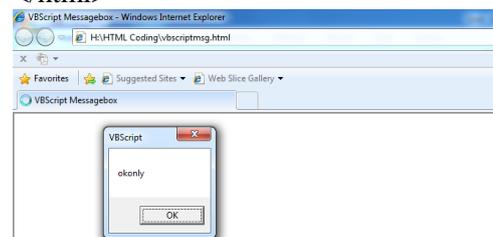


Figure 3.2 Message box



## VBScript Operators

VBScript has a full range of operators, including arithmetic operators, comparison operators, concatenation operators, and logical operators.

Arithmetic		Comparison	
Description	Symbol	Description	Symbol
Equality	=	Exponentiation	^
Inequality	<>	Unary negation	-
Less than	<	Multiplication	*
Greater than	>	Division	/
Less than or equal to	<=	Integer division	\
Greater than or equal to	>=	Modulus arithmetic	Mod
Object equivalence	Is	Addition	+
		Subtraction	-
		String concatenation	&

The string concatenation (&) operator is not an arithmetic operator, but in precedence it falls after all arithmetic operators and before all comparison operators.

When addition and subtraction occur together in an expression, each operation is evaluated in order of appearance from left to right. When multiplication and division occur together in an expression, each operation is evaluated as it occurs from left to right. Likewise,

### Logical

Description	Symbol
Logical negation	Not
Logical conjunction	And
Logical disjunction	Or
Logical exclusion	Xor
Logical equivalence	Eqv
Logical implication	Imp



---

## Assessment



Assessment

### Fill in the blanks

1. VBScript function is a series of statements,

State whether the following statements are true or false

1. When addition and subtraction occur together in an expression, each operation is evaluated in order of appearance from left to right

### Multiple Choice questions

1. Which of the following is not arithmetic operator?

- a) ^
- b) \*
- c) >=
- d) +.

### Exercises

1. Differentiate between Division (/) & Division (\) operator.
2. Explain the use of MOD Operator with Example.



## Lesson 9.4

### VB Script-Conditional statement

Upon completion of this unit you will be able to:



*Explain conditional statements.*

#### Outcomes

#### Conditional statements

Conditional statements are used to perform different actions for different decisions. In VBScript we have

- If statement** - executes a set of code when a condition is true.
- If...Then...Else statement** - select one of two sets of lines to execute.
- If...Then...ElseIf statement** - select one of many sets of lines to execute.
- Select Case statement** - select one of many sets of lines to execute.

#### **If...Then...Else**

The **If...Then...Else** statement is used to evaluate whether a condition is **True** or **False** and, depending on the result, to specify one or more statements to run. If you want to execute only **one** statement when a condition is true, you can write the code on one line

```
If i=10 Then alert("Hello")
```

There is no Else in this syntax. You just tell the code to perform **one action** if a condition is true (in this case If i=10).

Here is an example

```
<html>
<head>
<title>VBScript If</title>
</head>
<body>
<script type="text/vbscript">
Dim a
Dim b
a = 10
```

```
b = 20
If a < b Then
document.write("A is less than B")
End If
</script>
</body>
</html>
```

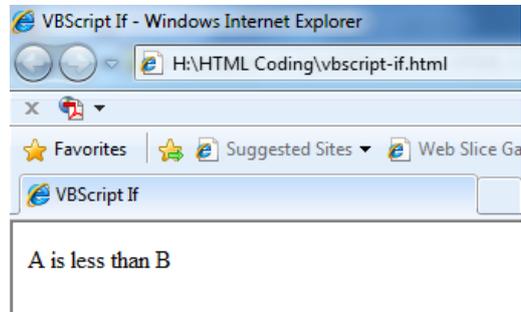


Figure 4.1 If Statement

The If statement in the above code checks if the value of the variable a is less than b and then outputs the text "A is less than B", if the condition is met. However, sometimes we may have two pieces of code and we wish to execute one if the condition is true and the other if the condition is false. In such situations we use the If Else statement. The If Else is an extension to the If statement that allows us to execute a piece of code if the condition is true, in case the condition is false a different block of code gets executed.

Here is an example:

```
<html>
<head>
<title>VBScript If-Else </title>
</head>
<body>
<script type="text/vbscript">
Dim a
Dim b
a = 30
b = 20
If a < b Then
document.write("A is less than B")
Else
document.write("A is greater than B")
End If
</script>
</body>
</html>
```

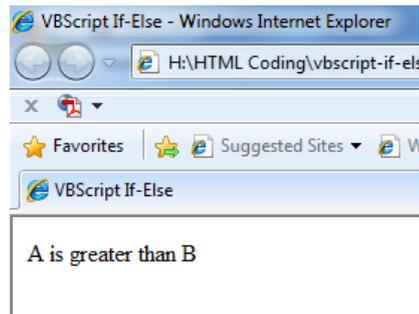


Figure 4.2 If Statement

The result of the If condition above  $a > b$  evaluates to false and therefore the Else part of the code gets executed that generates the above output.

There is another variation of the If statement called Elseif. We use Elseif when we need to check multiple conditions.

## Select Case Statement

In the last two sections we learned VBScript If statement and its different forms. There is another construct available in VBScript called Select Case that is a more elegant way of doing the same thing. A Select Case statement allows us to perform multiple tests, with fewer lines of code.

### Example

```
<html>
<head>
<title>VBScript Select Case</title>
</head>
<body>
<script type="text/vbscript">
d=weekday(date)
Select Case d
Case 1
document.write(" Sunday")
Case 2
document.write("Monday")
Case 3
document.write("Tuesday")
Case 4
document.write("Wednesday")
Case 5
document.write("Thursday")
Case 6
document.write("Friday")
Case else
```



## VB Script-Conditional statement

```
document.write("Saturday")  
End Select  
</script>  
</body>  
</html>  
</script>
```

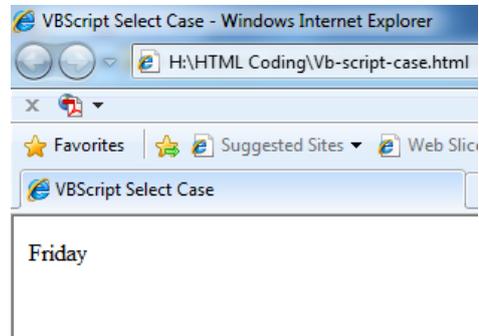


Figure 4.3 Select Case



---

## Assessment



Assessment

### Multiple Choice questions

1. The If...Then...Else statement is used to evaluate whether a condition is
  - a) True
  - b) False
  - c) Both a and b
  - d) None of above.

### Exercises

1. Explain the Use of If.. Else... Endif statement.
2. Give the syntax of Switch Statement.



## Lesson 9.5

### VB Script looping



#### Outcomes

#### Loop

Upon completion of this unit you will be able to:

*Define* loop.

*Use* loop.

Sometimes in our programs we have some code that we need to repeat for a specific number of times. Looping statements are used to run the same block of code a specified number of times. In VBScript we have four looping statements:

- For...Next statement** - runs code a specified number of times
- For Each...Next statement** - runs code for each item in a collection or each element of an array
- Do...Loop statement** - loops while or until a condition is true
- While...Wend statement** - Do not use it - use the Do...Loop statement instead

#### For Next Loop

Use the **For...Next** statement to run a block of code a specified number of times. The **For** statement specifies the counter variable *i*, and its start and end values. The **Next** statement increases the counter variable *i* by one. With the **Step** keyword, you can increase or decrease the counter variable by the value you specify. In the example below, the counter variable *i* is increased by two, each time the loop repeats.

```
For i=2 To 10 Step 2
  some code
Next
```

To decrease the counter variable, you must use a negative **Step** value. You must specify an end value that is less than the start value. In the example below, the counter variable *i* is decreased by two, each time the loop repeats.

```
For i=10 To 2 Step -2
  some code
Next
```

**Example**



```
<html>
<head>
<title>VBScript for loop</title>
</head>
<body>
<script type="text/vbscript">
For i = 2 To 10 step 2
  document.write("The number is " &i& "<br />")
Next
</script>
</body>
</html>
```

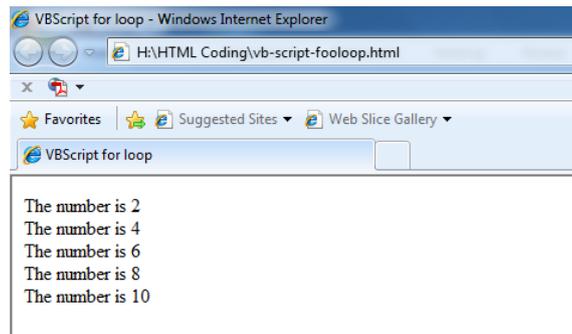


Figure 5.1 For Statement

### Exit a For...Next

You can exit a For Next statement with the Exit For keyword.

```
For i=1 To 10
  If i=5 Then Exit For
  some code
Next
```

## Do...Loop

You can use **Do...Loop** statements to run a block of statements an indefinite number of times. The statements are repeated either while a condition is **true** or until a condition becomes **true**.

The below example uses **Do..While** loop to check the condition at the beginning of the loop. The statements inside the loop are executed only if the condition becomes true.

```
<html>
<head>
<title>VBScript for loop</title>
```



```
</head>
<body>
<script language="vbscript" type="text/vbscript">
  Do While i < 5
    i = i + 1
  Document.write("The value of i is : " &i)
  Document.write("<br></br>")
  Loop
</script>
</body>
</html>
```

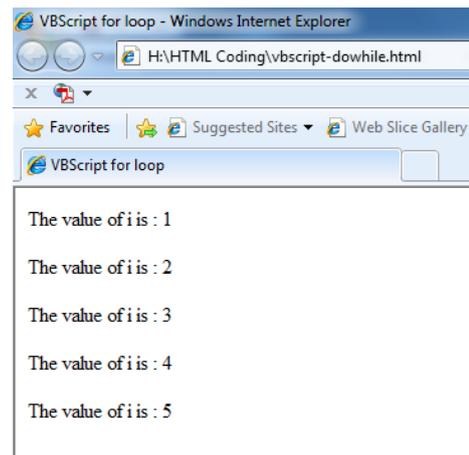


Figure 5.2 Do loop

### Repeat Code Until a Condition Becomes True

Use the Until keyword to check a condition in a Do...Loop statement.

```
<html>
<head>
<title>VBScript for loop</title>
</head>
<body>
<script language="vbscript" type="text/vbscript">
  Do until i=5
    i=i+1
  Document.write("The value of i is : " &i)
  Document.write("<br></br>")
  Loop
</script>
</body>
</html>
```



---

## Assessment



Assessment

### Multiple Choice questions

1. How many looping statements have in VBScript?
  - a) 4
  - b) 2
  - c) 3
  - d) 1.

### Exercises

1. Explain the Use of For loop with example.
2. Differentiate between Do While & Do Until loop.