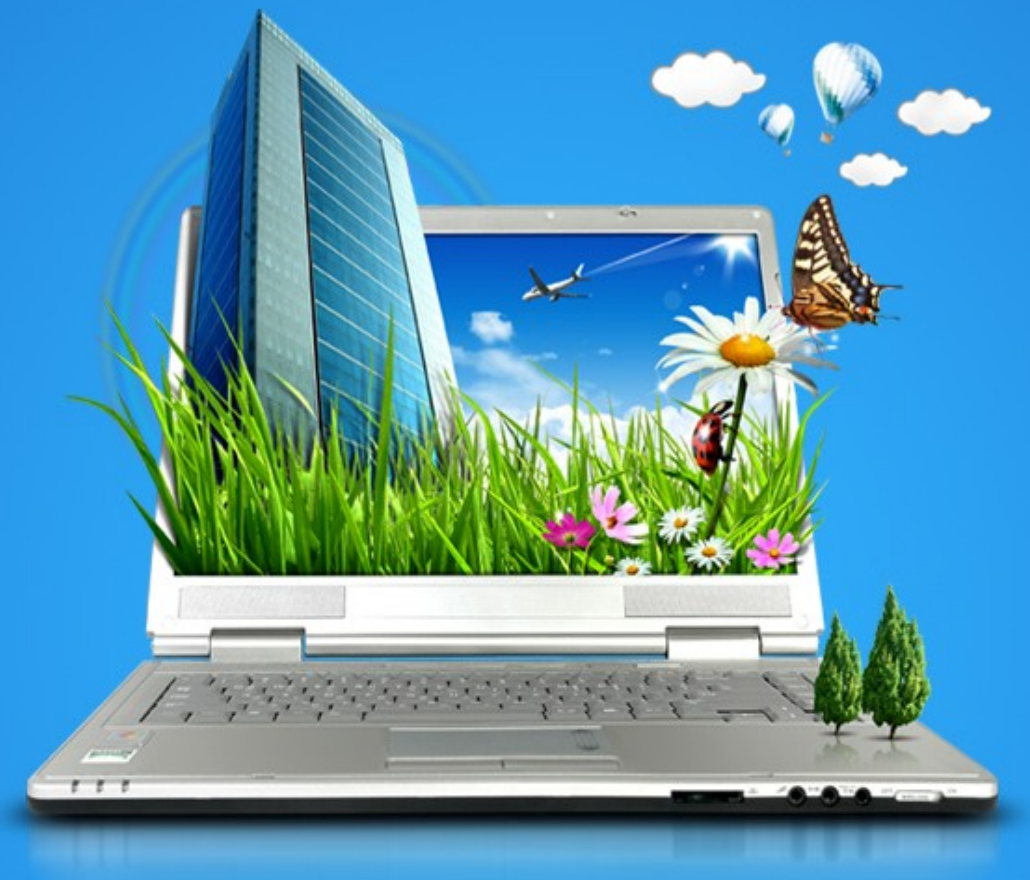

VB Script

What is VBScript?

An interpreted programming language primarily used to write relatively short programs that access operating system and application objects. Such languages are referred to as "scripting" languages. The VBScript commands are a subset of the more complete, compiled Visual Basic language.

How Does it Work?

When a VBScript is inserted into an HTML document, the Internet browser will read the HTML and interpret the VBScript. The VBScript can be executed immediately, or at a later event.

Scripting in the Browser

- An important use of Web browsers in electronic commerce is to enter information.
- Validation of input and calculations on the browser can be carried out via a type of computer programming known as **scripting**.
- Scripting is similar to other types of computer programming in that it uses variables and statements to carry out a desired set of logical operations, but it is also different.
- Instead of being executed as a compiled program, it is executed by another program, in our case the Web browser.
- **Scripting languages** are easier to work with than compiled languages.
- A script takes longer to run than a compiled program since each instruction is being interpreted rather than being executed directly by the processor.

Scripting Languages

- The two most widely used scripting languages for working with Web pages are JavaScript and VBScript.
- **Javascript** uses a C-like syntax and can run on either Netscape or Internet Explorer browsers.
- **VBScript** is based on Visual Basic, but runs only on Internet Explorer.
- Both Javascript and VBScript, when used on the browser, are termed **client-side scripting** since they are running on the Web client.
- VBScript is also widely used in the **Active Server Page (ASP)** approach to directing Microsoft Web server software that runs on Windows NT or 2000 operating systems. This is called **server-side scripting**.

Differences between VBScript and Visual Basic

Visual Basic	VBScript
Uses different types of variables and constants	Uses only one type of variable--the Variant
Can be compiled into an exe file	Is interpreted by the Internet Explorer browser software
Uses event procedures to react to events	Uses event handlers to react to events
Has an easy-to-use integrated development environment (IDE)	Does not have an easy-to-use IDE specifically for VBScript
Runs as stand-alone language	Must be integrated with HTML

Using VBScript

- VBScript has only one type of variable—the **Variant**. It is a special kind of data type, having the capability to store different kinds of information, depending on how it is used. A variant can contain a numeric value or it can also contain string information.

  VBScript programs must be **interpreted** by other software, usually the Internet Explorer browser on the client side and Web server software on the server side.

- In VBScript, we can write code to respond to the events, these code procedures are referred to as **event handlers**.

- Since there is no easy-to-use IDE for VBScript that is readily available, it is convenient to use a **text editor** like Notepad.

- VBScript must be **tightly integrated** with the HTML.

The <Script> Tag

- All VBScript code must be enclosed in HTML script tags.

```
<SCRIPT language="VBScript"> ...  </SCRIPT>
```

- The same type tags are used for JavaScript with a change in the language parameter.
- VBScript code is typically located in two places in the Web page—in the HEAD section and in the BODY section.
- When VBScript is placed in the HEAD section, it is in the form of functions and sub programs that act as event handlers.
- When VBScript code appears in the BODY section of the HTML code, it is executed when the page loads.

Put a VBScript into an HTML Page

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

Declaring a variable in VBScript

```
<script type="text/vbscript">
Dim age
</script>
```

```
<script type="text/vbscript">
Dim age, age2, age3
</script>
```

Assign Values to Variables in VBScript

```
<script type="text/vbscript">
  Dim age
age = 25
document.write(age)
</script>
```

The example below shows how to add HTML tags to the VBScript:

```
<html>
<body>
        <script type="text/vbscript">
                document.write("<h1>Hello World!</h1>")
        </script>
</body>
</html>
```

VBScript 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.

The **document.write** command is a standard VBScript command for writing output to a page.

Scripts in <head>

VBScripts in the head section will be executed when CALLED.

```
<html>
<head>
<script type="text/vbscript">
alert("Hello World!")
</script>
</head>

<body>
</body>
</html>
```

Scripts in <body>

VBScripts in the body section will be executed WHILE the page loads.

```
<html>
<head>
</head>

<body>
<script type="text/vbscript">
    document.write("This message is written by VBScript")
</script>
</body>

</html>
```

Scripts in <head> and <body>

You can place an unlimited number of scripts in your document, so you can have scripts in both the body and the head section.

```
<html>
<head>
        <script type="text/vbscript">
                ....
        </script>
</head>
<body>
        <script type="text/vbscript">
                ....
        </script>
</body>
```

VBScript Procedures

In VBScript, there are two kinds of procedures:

- Sub procedure
- Function procedure

VBScript Sub Procedures

A Sub procedure:

- is a series of statements, enclosed by the Sub and End Sub statements
- can perform actions, but **does not return** a value
- can take arguments
- without arguments, it must include an empty set of parentheses ()

```
Sub mysub ()
  some statements
End Sub

or

Sub mysub (argument1,argument2)
  some statements
End Sub
```

VBScript Function Procedures

A Function procedure:

- is a series of statements, enclosed by the Function and End Function statements
- can perform actions and **can return** a value
- can take arguments that are passed to it by a calling procedure
- without arguments, must include an empty set of parentheses ()
- returns a value by assigning a value to its name

```
Function myfunction ()
  some statements
  myfunction=some value
End Function
```

or

```
Function myfunction (argument1,argument2)
  some statements
  myfunction=some value
End Function
```

VBScript Conditional Statements

Conditional Statements

Conditional statements are used to perform different actions for different decisions.

In VBScript we have four conditional statements:

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

```
<html>
<body>
    <script type="text/vbscript">
     Function greeting()
            i=hour(time)
            If i < 10 Then
                        document.write("Good morning!")
            Else
                        document.write("Have a nice day!")
            End If
      End Function
    </script>
</head>

<body onload="greeting()">
</body>

</html>
```

VBScript Looping

Looping Statements

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 Each...Next Loop

```
<html>
<body>

<script type="text/vbscript">
    Dim cars(2)
    cars(0)="Volvo"
    cars(1)="Saab"
    cars(2)="BMW"

    For Each x In cars
       document.write(x & "<br />")
     Next
</script>

</body>
</html>
```

Line Breaks

```
'******************
' LongString.vbs
'******************
Option Explicit
Dim strMessage
strMessage = "This is a string concatenation" &_
"that is too long to fit on a single line of code"
Wscript.echo(strMessage)
```

Order of Operations

- Please — Parenthesis
- Excuse — Exponents
- My — Multiplication
- Dear — Division
- Aunt — Addition
- Sally — Subtraction

OpOrder.vbs

```
'******************
' OpOrder.vbs
'******************
option explicit
dim X, Y
X = 3 * 5 + 3 * 5 ^ 2
Y = (2 * (5 + 3) * 5) ^ 2
Wscript.echo("The value of X is: " & X)
Wscript.echo("The value of Y is: " & Y)
```

What is the final value of the variables X and Y?
Write and execute this program to find out if you are correct.

String Concatenation

- Strings in VBScript can be concatenated (connected) using either the + or & operators. Most VBScript programmers use & to avoid confusion with the addition operation:

```
<script type="text/vbscript">
Dim h
Dim w
h = "Hello "
w = "World!"
hw = h & w
document.write(hw)
</script>
```

What is the purpose of the two quotes in the middle of the concatenation? Write and execute this program to verify your suspicion.

HTML Code for Input Form (Part 1)

```
<HTML>
<HEAD>
<TITLE>Vintage Videos Online Rental System</TITLE>
</HEAD>
<BODY>
<H1 ALIGN=center>Vintage Videos Online Rental Form</H1>
<FORM NAME=frmInput METHOD=post ACTION=mailto:videosv@negia.net
   ENCTYPE=text/plain>
<H3>Please input your name, telephone number including area code,
   and e-mail address:</H3>
<H3>Name: <INPUT TYPE=text NAME=txtName></H3>
<H3>Telephone Number: <INPUT TYPE=text NAME=txtPhoneNum></H3>
<H3 align=left>E-mail Address: <INPUT TYPE=text
   NAME=txtEmail></H3>
<H3>Now, select a video to rent and have delivered:</H3>
<SELECT NAME=lstVideos>
<OPTION value=0>  </OPTION>
<OPTION value=2>Psycho</OPTION>
<OPTION value=1>Bambi</OPTION>
<OPTION value=2>Ghost</OPTION>
<OPTION value=3>Star Wars</OPTION>
<OPTION value=1>Dumbo</OPTION>
<OPTION value=2>Rain Man</OPTION>
```

HTML Code for Input Form (Part 2)

```
<OPTION value=2>Blazing Saddles</OPTION>
<OPTION value=2>Ben Hur</OPTION>
<OPTION value=3>Spartacus</OPTION>
<OPTION value=2>Tootsie</OPTION>
<OPTION value=3>The Sting</OPTION>
</SELECT>
<H3>The video you have selected is: <INPUT TYPE=text
   NAME=txtVideo>
The price of this video is: <INPUT TYPE=text NAME=txtprice>
The delivery fee and taxes are: <INPUT TYPE=text
   NAME=txtDeliveryFee>
<H3>The total cost is: <INPUT TYPE=text NAME=txtTotalCost>
<H3>If you are satisfied with the results and want the video
delivered, click the Submit button. To start over, click the
   Reset button.</H3>
<INPUT TYPE=submit NAME=cmdSubmit VALUE="Submit Order">
      
<INPUT NAME=cmdReset TYPE=reset VALUE="Clear Entries">
</FORM>
</BODY>
</HTML>
```

Vintage Videos Online Rental Form

Please input your name, telephone number (including area code), and e-mail address:

Name: text box

Telephone Number:

E-mail Address:

Now, select a video to rent and have delivered:

list box

The video you have selected is:

The price of this video is:

The delivery fee and taxes are

The total cost is:

Submit button

Reset button

If you are satisfied with the results and want the video delivered, click the Submit button. To start over, click the Reset button.

The Validation Process

- The validation process on a Web page is similar in many ways to the validation process for Visual Basic forms.
- Typical validation questions are:
  - Is there an appropriate number of digits in a name or telephone number?
  - Is there an @ sign in an e-mail address with a sufficient number of characters?
  - Are there exactly nine digits in a Social Security number with dashes in the correct location?
  - Are there an appropriate number of characters in a credit card number?

Checking for Empty or Short Text Box Entries

```
<SCRIPT LANGUAGE ="VBScript">
Function frmInput_OnSubmit
    Dim strName, strEmail, strPhone, strVideo, intSelIndex
    strName = frmInput.txtName.Value
    strPhone = frmInput.txtPhoneNum.Value
    strEmail = frmInput.txtEmail.Value
    If Len(strName) < 5 Then
            Msgbox "Please input a name at least 5 characters long!"
            frmInput.txtName.Value = ""
            frmInput.txtName.Focus
            frmInput_OnSubmit = False
            Exit Function
    ElseIf Len(strPhone) <> 12 Then
            Msgbox "Please input a phone number with exactly 12 digits!"
            frmInput.txtPhoneNum.Value = ""
            frmInput.txtPhoneNum.Focus
            frmInput_OnSubmit = False
            Exit Function
    ElseIf InStr(strEmail,"@") = 0 Or Len(strEmail) < 5 Then
            Msgbox "Please input an e-mail address with an @ sign" _
         & "and at least 5 characters!"
            frmInput.txtEmail.Value = ""
            frmInput.txtEmail.Focus
            frmInput_OnSubmit = False
            Exit Function
    End If
End Function  </SCRIPT>
```

List Box Validations

```
intSelIndex = frmInput.lstVideos.SelectedIndex
If intSelIndex < 1 Then
    Msgbox "You must select a video!"
    frmInput.lstVideos.Focus
    frmInput_OnSubmit = False
    Exit Function
End If
```

Displaying Video Names in Text Box

```
Sub lstVideos_OnClick
    Dim strVideoName, curVideoPrice, curTaxes
    Dim intIndex, intPrice, curTaxesFees, curTotal
    Const curDeliveryFee = 2.00
    intIndex = frmInput.lstVideos.SelectedIndex
    strVideoName =
    frmInput.lstVideos.Options(intIndex).Text
    frmInput.txtVideo.Value = strVideoName
End Sub
```

Determining and Displaying Price

```
intPrice = frmInput.lstVideos.Value
Select Case intPrice
Case 1
    curVideoprice = .99
Case 2
    curVideoPrice = 1.99
Case 3
    curVideoPrice = 2.99
End Select
frmInput.txtPrice.Value=FormatCurrency(curVideoPrice)
```

Code to Calculate Rental Cost

```
frmInput.txtPrice.Value
FormatCurrency(curVideoPrice)
curTaxes = 0.07 * curVideoPrice
curTaxesFees = curTaxes + curDeliveryFee
frmInput.txtDeliveryFee.Value = FormatCurrency(curTaxesFees)
curTotal = curVideoPrice + curTaxesFees
frmInput.txtTotalCost.Value = FormatCurrency(curTotal)
```