VBScript脚本语言

就业培训教材

http://www.51testing.com

上海博为峰软件技术有限公司

VB家族语言简介

- Visual Basic 6.0
  - 源自于BASIC编程语言。
  - 由微软公司开发的包含协助开发环境的事件驱动编程语言。开发的程序只能运行在Microsoft Windows中，并且在运行时还需要一个1.4M大小的运行库。
  - 功能强大、易学易用。
  - 虽然是面向对象的编程语言，但是不支持继承、多线程等特性。
- VB6 派生的语言
  - Visual Basic for Applications，即VBA，包含在微软的应用程序中（比如Microsoft Office），以及类似WordPerfect Office这样第三方的产品里面。
  - Visual Basic Scripting Edition ，即VBScript，是默认的ASP语言，还可以用在Windows脚本编写和网页编程中。它的语法类似于VB，但不使用VB运行库运行，而是由Windows脚本主机解释执行。
- Visual Basic .NET
  - VB6.0 的继任者Visual Basic .NET，使用了新的核心和特性，需要.NET Framework的支持，是.NET平台的一部分（如使用ADO.NET来访问数据库）。
  - VB.NET编程语言是一种真正的面向对象编程语言，和VB6 并不完全兼容。
  - VB.NET系列包括Visual Basic .NET 2003（VB7.1）、 Visual Basic 2005（VB8.0）、Visual Basic 2008（VB9.0）

VBScript 数据类型

- VBScript 只有一种数据类型，称为 Variant。Variant 是一种特殊的数据类型，根据使用的方式，它可以包含不同类别的信息。因为 Variant 是 VBScript 中唯一的数据类型，所以它也是 VBScript 中所有函数的返回值的数据类型，包括Empty，Null 和 Error 三种特殊类型。
- 最简单的 Variant 可以包含数字或字符串信息。Variant 用于数字上下文中时作为数字处理，用于字符串上下文中时作为字符串处理。这就是说，如果使用看起来象是数字的数据，则 VBScript 会假定其为数字并以适用于数字的方式处理。与此类似，如果使用的数据只可能是字符串，则 VBScript 将按字符串处理。也可以将数字包含在引号 (" ") 中使其成为字符串。

Variant 子类型

- 除简单数字或字符串以外，**Variant** 可以进一步区分数值信息的特定含义。例如使用数值信息表示日期或时间。此类数据在与其他日期或时间数据一起使用时，结果也总是表示为日期或时间。从 **Boolean** 值到浮点数，数值信息是多种多样的。**Variant** 包含的数值信息类型称为子类型。大多数情况下，可将所需的数据放进 **Variant** 中，而 **Variant** 也会按照最适用于其包含的数据的方式进行操作。
- 可以使用转换函数来转换数据的子类型。另外，可使用 **VarType** 函数返回数据的 **Variant** 子类型。

Variant 数据子类型

子类型	描述
Empty	未初始化的 **Variant**。对于数值变量，值为 0；对于字符串变量，值为零长度字符串 ("")。
Null	不包含任何有效数据的 **Variant**。
Boolean	包含 True 或 False。
Byte	包含 0 到 255 之间的整数。
Integer	包含 -32,768 到 32,767 之间的整数。
Currency	货币类型
Long	包含 -2,147,483,648 到 2,147,483,647 之间的整数。
Single	单精度浮点数
Double	双精度浮点数
Date (Time)	包含表示日期的数字，日期范围从公元 100 年 1 月 1 日到公元 9999 年 12 月 31 日。
String	包含变长字符串，最大长度可为 20 亿个字符。
Object	包含对象。
Error	包含错误号。

VBScript 变量

- 声明变量
  - 声明变量的一种方式是使用 **Dim** 语句、**Public** 语句和 **Private** 语句在脚本中显式声明变量。例如：**Dim UserName**
  - 声明多个变量时，使用逗号分隔变量。例如： **Dim Top, Bottom, Left, Right**
  - 另一种方式是通过直接在脚本中使用变量名这一简单方式隐式声明变量。这通常不是一个好习惯，因为这样有时会由于变量名被拼错而导致在运行脚本时出现意外的结果。因此，最好使用 **Option Explicit** 语句显式声明所有变量，并将其作为脚本的第一条语句。
- 命名规则
  - 第一个字符必须是字母。
  - 不能包含嵌入的句点。
  - 长度不能超过 **255** 个字符。
  - 在被声明的作用域内必须唯一。

VBScript 变量

- 变量的作用域与存活期
  - 变量的作用域由声明它的位置决定。如果在过程中声明变量，则只有该过程中的代码可以访问或更改变量值，此时变量具有局部作用域并且是过程级变量。如果在过程之外声明变量，则该变量可以被脚本中所有过程所识别，称为 **Script** 级变量，具有脚本级作用域。
  - 变量存在的时间称为存活期。**Script** 级变量的存活期从被声明的一刻起，直到脚本运行结束。对于过程级变量，其存活期仅是该过程运行的时间，该过程结束后，变量随之消失。在执行过程时，局部变量是理想的临时存储空间。可以在不同过程中使用同名的局部变量，这是因为每个局部变量只被声明它的过程识别。
- 给变量赋值
  - 创建如下形式的表达式给变量赋值：变量在表达式左边，要赋的值在表达式右边。
- 标量变量和数组变量
  - 多数情况下，只需为声明的变量赋一个值。只包含一个值的变量被称为标量变量。有时候，将多个相关值赋给一个变量更为方便，因此可以创建包含一系列值的变量，称为数组变量。声明数组变量时变量名后面带有括号 **( )**，如**Dim A(10)**。在 **VBScript** 中所有数组都是基于 **0** 的，在基于 **0** 的数组中，数组元素的数目总是括号中显示的数目加 **1**。这种数组被称为固定大小的数组。
  - 要使用动态数组，必须随后使用 **ReDim** 确定维数和每一维的大小。使用 **Preserve** 关键字在重新调整大小时保留数组的内容。重新调整动态数组大小的次数是没有任何限制的，尽管将数组的大小调小时，将会丢失被删除元素的数据。

VBScript 常数

- 常数是具有一定含义的名称，用于代替数字或字符串，其值被创建后就不允许再被改变。**VBScript** 定义了许多内部常数。例如：

  MsgBox "提示信息： " & vbCrLf & "操作成功", vbInformation, "Title"

- 创建常数
  - 使用 **Const** 语句在 **VBScript** 中创建用户自定义常数。使用 **Const** 语句可以创建名称具有一定含义的字符串型或数值型常数，并给它们赋原义值。例如：

    Const conUserName = “songfun"

    Const PI = 3.1415926535897

    Const conOlympicDate = #08/08/08#
  - 最好采用一个命名方案以区分常数和变量。这样可以避免在运行脚本时对常数重新赋值。例如，可以使用“**vb**”或“**con**”作常数名的前缀，或将常数名的所有字母大写。将常数和变量区分开可以在开发复杂的脚本时避免混乱。
- 常用常数

vbCr	回车符	**vbLf**	换行符
vbCrLf	回车符与换行符。	**vbNewLine**	新行字符

VBScript 运算符

算术运算符		比较运算符		逻辑运算符	
描述	符号	描述	符号	描述	符号
求幂	^	等于	=	逻辑非	Not
负号	-	不等于	<>	逻辑与	And
乘	*	小于	<	逻辑或	Or
除	/	大于	>	逻辑异或	Xor
整除	\	小于等于	<=	逻辑等价	Eqv
求余	Mod	大于等于	>=	逻辑隐含	Imp
加	+	对象引用比较	Is		
减	-				
字符串连接	&				

VBScript条件语句

- 使用条件语句和循环语句可以控制脚本的流程。使用条件语句可以编写进行判断和重复操作的 **VBScript** 代码。在 **VBScript** 中可使用以下条件语句：
  - **If...Then...Else** 语句

    条件为 **True** 时运行语句，例如：

    ```
    If myDate < Now Then myDate = Now  '单行不需要加End If
    ```

    条件为 **True** 和 **False** 时分别运行某些语句。

    对多个条件进行判断
  - **Select Case** 语句

    ```
    Select Case iValue
        Case 1
            MsgBox "Hello World!"
        Case 2
            MsgBox "Hello Software Testing!"
        Case Else
            MsgBox "Hello 51Testing!"
    End Select
    ```

VBScript循环语句

- 循环用于重复执行一组语句。循环可分为三类：一类在条件变为 **False** 之前重复执行语句，一类在条件变为 **True** 之前重复执行语句，另一类按照指定的次数重复执行语句。
- 在 **VBScript** 中可使用下列循环语句：
- **Do...Loop**： 当（或直到）条件为 **True** 时循环。

```
Do While myNum > 10
    myNum = myNum - 1
Loop
```

```
Do
    myNum = myNum - 1
Loop While myNum > 10
```

- **While...Wend**： 当条件为 **True** 时循环。
- **For...Next**： 指定循环次数，使用计数器重复运行语句。
- **For Each...Next**：对于集合中的每项或数组中的每个元素，重复执行一组语句。

VBScript中的With语句

- 对一个对象执行一系列的语句。
- 语法：

```
With object
    statements
End With
```

例如：

```
With MyLabel
   .Height = 2000
   .Width = 2000
   .Caption = "这是MyLabel"
End With
```

- 当程序一旦进入 **With** 块，***object*** 就不能改变。因此不能用一个 **With** 语句来设置多个不同的对象。
- **With**语句可以嵌套

VBScript 过程

- 在 VBScript 中，过程被分为两类：Sub 过程和 Function 过程。
  - Sub 过程

    Sub 过程是包含在 Sub 和 End Sub 语句之间的一组 VBScript 语句，执行操作但不返回值。Sub 过程可以使用参数（由调用过程传递的常数、变量或表达式）。如果 Sub 过程无任何参数，则 Sub 语句必须包含空括号 ()。
  - Function 过程

    Function 过程是包含在 Function 和 End Function 语句之间的一组 VBScript 语句。Function 过程与 Sub 过程类似，但是 Function 过程可以返回值。Function 过程可以使用参数（由调用过程传递的常数、变量或表达式）。如果 Function 过程无任何参数，则 Function 语句必须包含空括号 ()。Function 过程通过函数名返回一个值，这个值是在过程的语句中赋给函数名的。Function 返回值的数据类型总是 Variant。

VBScript 过程

- **Sub语句**
  - 声明 **Sub** 过程的名称、参数以及构成其主体的代码。

```
[Public [Default] | Private] Sub name [(arglist)]
    [statements]
    [Exit Sub]
    [statements]
End Sub
```

- **Function语句**
  - 声明 **Function** 过程的名称、参数以及构成其主体的代码。

```
[Public [Default]| Private] Function name [( arglist )]
    [statements]
    [name = expression]
    [Exit Function]
    [statements]
    [name = expression]
End Function
```

VBScript 过程

- 过程中的**arglist** 参数
  - **[ByVal | ByRef]** *varname*[( )]

    ByVal 表示该参数按值传递。

    ByRef 表示该参数按引用传递。

    varname 代表参数的变量名称，遵循标准变量命名规则。
- 过程的声明
  - 没有显式地指定使用 **Public** 或 **Private**，则 **Sub/Function** 过程默认为公用，即它们对于脚本中的所有其他过程都是可见的。**Sub/Function** 过程中局部变量的值在调用过程中不被保留。
- 过程的调用
  - 使用 **Call** 语句调用
  - 直接输入过程名调用
  - 支持递归调用

VBScript的编码规范

- 编码约定
  - 对象、变量和过程的命名规范
    - 匈牙利命名法
    - 骆驼命名法
    - 帕斯卡命名法
  - 注释约定
    - 不要在代码行的结尾处使用注释。要将注释放在单独行。
    - 注释文本以大写字母开头。
    - 注释以句点结束。
    - 在注释分隔符 (') 和注释文本之间插入一个空格。
    - 请勿创建已设置格式的将注释包含在内的星号块。
  - 文本格式和缩进指南
    - 关键字空一格，运算符两边不加空格，括号内侧空一格
    - 缩进以4个空格为单位
    - 函数之间空两行
    - 语句块的配对对齐

VBScript变量的命名规则

类型	前缀	示例
Variant	var	varArray
Boolean	bln	blnFound
Byte	byt	bytRasterData
Date (Time)	dtm	dtmStart
Double	dbl	dblTolerance
Error	err	errOrderNum
Integer	int	intQuantity
Long	lng	lngDistance
Object	obj	objCurrent
Single	sng	sngAverage
String	str	strFirstName

VBScript 编码规范（例）

```
'*********************************************************
' 目的：在 UserList 数组中
            ' 定位指定用户的首次出现。
' 输入：strUserList()：要搜索的用户列表。
            ' strTargetUser：要搜索的用户名。
' 返回：索引 strUserList 数组中
            ' strTargetUser 的首次出现。
            ' 如果找不到目标用户，则返回 -1。
'*********************************************************
Function intFindUser (strUserList(), strTargetUser)
    Dim i ' Loop counter.
    Dim blnFound ' 找到目标标志
    intFindUser = -1
    i = 0 ' 初始化循环计数器
    Do While i <= Ubound(strUserList) and Not blnFound
        If strUserList(i) = strTargetUser Then
            blnFound = True ' 将标志设置为 True
            intFindUser = i ' 将返回值设置成循环计数
        End If
        i = i + 1 ' 递增循环计数器
    Loop
End Function
```

VBScript的常用函数（1）

- 字符串函数
  - Len 函数
  - Left 函数
  - Mid 函数
  - Right 函数
  - InStr 函数
  - InStrRev 函数
  - LTrim、RTrim 和 Trim 函数
  - LCase 函数
  - UCase 函数
  - Replace 函数
  - StrComp 函数
  - Split 函数
  - Join 函数

VBScript的常用函数（2）

- 转换函数
  - Asc 函数
  - Chr 函数
  - Str 函数
  - Val 函数
  - CBool 函数
  - CByte 函数
  - CCur 函数
  - CDate 函数
  - CDbl 函数
  - CInt 函数
  - CLng 函数
  - CSng 函数
  - CStr 函数

VBScript的常用函数（3）

- 判断函数
  - IsArray 函数
  - IsDate 函数
  - IsEmpty 函数
  - IsNumeric 函数
  - IsNull 函数
  - IsObject 函数
  - VarType 函数
  - TypeName 函数

VBScript的常用函数（4）

- 时间函数
  - Date 函数
  - Day 函数
  - Hour 函数
  - Minute 函数
  - Month 函数
  - Now 函数
  - Second 函数
  - Time 函数
  - Weekday 函数
  - WeekDayName 函数
  - Year 函数

VBScript的常用函数（5）

- 其他函数
  - Rnd 函数和 Randomize 语句

    公式：Int((upperbound - lowerbound + 1) * Rnd + lowerbound)
  - CreateObject 函数

    Set myObj = CreateObject("WScript.Shell")

    Set myObj = CreateObject("Excel.Application")

    Set myObj = CreateObject("Scripting.FileSystemObject")

    Set myObj = CreateObject("Scripting.Dictionary")

    Set myObj = CreateObject("ADODB.Connection")

    Set myObj = CreateObject("ADODB.Recordset")

    Set myObj = CreateObject("Microsoft.XMLDOM")

    Set myObj = CreateObject(“InternetExplorer.Application”)
  - GetObject 函数
  - Int、Fix 函数
  - LBound、Ubound 函数
  - MsgBox、InputBox 函数

VBScript的对象

- Class 对象
  - Property Get 语句
  - Property Let 语句
  - Property Set 语句
  - Initialize 事件
  - Terminate 事件
- Err 对象
  - 属性
    - Description 属性
    - HelpContext 属性
    - HelpFile 属性
    - Number 属性
    - Source 属性
  - 方法
    - Clear 方法
    - Raise 方法
- RegExp 对象

VBScript中的错误处理

- On Error 语句
  - On Error GoTo *line*
  - On Error Resume Next
  - On Error GoTo 0

VBScript中的文件读写（1）

- 文本文件的读写

代码示例

```
Option Explicit
Const ForReading = 1,ForWriting = 2,ForAppending = 8
Dim fso,file,msg
Set fso = CreateObject("Scripting.FileSystemObject")

Set file = fso.OpenTextFile ("c:\calc.txt",ForReading)
While (Not file.AtEndOfLine )
    msg = msg & file.ReadLine & Chr(13) & Chr(10)
Wend
MsgBox msg
file.Close
Set file = Nothing
Set fso = Nothing
```

- **Excel文件的读写**

代码示例

```
Dim xlApp,xlWorkBook,xlSheet
Dim iRowCount,iLoop,numAdd
Set xlApp = CreateObject("Excel.Application")
xlApp.Visible = True
Set xlWorkBook = xlApp.Workbooks.Open("c:\data.xls")
Set xlSheet = xlWorkBook.Sheets("Sheet1")
iRowCount = xlSheet.usedRange.Rows.Count
For iLoop = 2 To iRowCount
    numAdd = xlSheet.Cells(iLoop,1)
Next
xlWorkBook.Save
xlWorkBook.Close
xlApp.Quit
Set xlSheet = Nothing
Set xlWorkBook = Nothing
Set xlApp = Nothing
```

VBScript中的文件读写（3）

- 数据库文件的读写

代码示例

```
Dim Cnn,Rst,strCnn
strCnn = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=C:\calc.mdb;Persist Security
Info=False“
Set Cnn = CreateObject("ADODB.Connection")
Cnn.Open strCnn
Set Rst = CreateObject("ADODB.Recordset")
Rst.Open "Select * from calc",Cnn
Rst.MoveFirst
Do While Not Rst.EOF
    MsgBox Trim(Rst.Fields("TestResult"))
    Rst.MoveNext
Loop
Rst.Close
Cnn.Close
Set Rst = Nothing
Set Cnn = Nothing
```

VBScript中的文件读写（4）

- **XML文件的读写**

代码示例

```
Dim xmlDoc, xmlRoot, rootChildItem, msg
Set xmlDoc = CreateObject("Microsoft.XMLDOM")
xmlDoc.async = False
xmlDoc.Load "C:\calc.xml"
If xmlDoc.parseError.errorCode <> 0 Then
    MsgBox "XML loaded failed. The reason is :" & xmlDoc.parseError.reason
    Exit Sub
End If
Set xmlRoot = xmlDoc.documentElement
If Not xmlRoot.hasChildNodes Then Exit Sub
For Each rootChildItem In xmlRoot.childNodes
    If rootChildItem.nodeName = "TestCase" Then
        msg = msg & rootChildItem.firstchild.nodeValue & vbNewLine
    End If
Next
MsgBox msg
```

WSH

- 什么是**WSH**
- **Windows** 脚本宿主对象模型
  - **WScript** 对象
  - **WshShell** 对象
- 代码示例

```
Option Explicit

Dim oShell
Set oShell = CreateObject ("WSCript.shell")
oShell.run "cmd /K CD C:\ & java Counter"
Set oShell = Nothing
```

答疑