



# 第13章 VBScript基础

- 教学目标
  - 了解VBScript的基本概念
  - 掌握VBScript的主要语法结构
  - 了解VBScript的编码约定方法
- 重点
  - 掌握VBScript的基本编程方法和技巧

- 13.1 VBScript简介
- 13.13 VBScript语法基础
  - 数据类型
  - VBScript语法基础
  - 常数
  - VBScript运算符
  - 条件语句
  - 循环语句
  - VBScript过程
- 13.3 正则表达式
  - 正则表达式语法
  - 建立正则表达式
- 13.4 VBScript编码约定
  - 常量命名规则
  - 变量命名规则
  - 代码注释
- 13.5 VBScript在项目中的使用
  - 13.5.1 动态对象识别方法
  - 13.5.2 UTF异常处理 ( 场景恢复 )



# 13.1 VBScript简介

- VBScript是Visual Basic Script的简称，是Visual Basic的一个安全子集，设计目的：在多平台上运行并能使用ActiveX接口，适合于Windows管理、网页脚本编程、开发测试脚本。
- VBScript是微软开发的一种解析型的服务端（也支持客户端）脚本语言
- VBScript高效，易学
- HP公司并未针对UFT自动化测试工具提供单独的厂商语言，而是使用VBScript脚本语言。因此，VBScript内置的大部分函数、方法都可以直接在UFT中使用，从而节省了读者学习厂商语言的时间



## 13.2 VBScript语法基础

- Hello Word

- 新建一个记事本，在记事本中输入以下语句：msgbox “Hello World”。
- 保存记事本，将文件类型保存为vbs格式。
- 打开该文件，会看到系统弹出一个消息框上写着 “Hello World” 字样。

注意：MsgBox是[Visual Basic](#)和[VBS](#)中的一个函数，功能是弹出一个[对话框](#)，等待用户单击按钮

MsgBox (***Prompt [, Buttons]***



***helpfile, Context]***)

## 13.2.1 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
- 最简单的 Variant 可以包含数字或字符串信息
- 除了简单的数字或字符串以外，Variant可以进一步区分数据类型  
如：使用数值信息表示日期或时间



## 13.2.1 数据类型

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。嗯，
Currency	-922,337,203,685,477.5808 到 922,337,203,685,477.5807。
Long	包含 -2,147,483,648 到 2,147,483,647 之间的整数。
Single	包含单精度浮点数，负数范围从 -3.402823E38 到 -1.401298E-45，正数范围从 1.401298E-45 到 3.402823E38。
Double	包含双精度浮点数，负数范围从 -1.79769313486232E308 到 -4.94065645841247E-324，正数范围从 4.94065645841247E-324 到 1.79769313486232E308。
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。

- **变量名称的规则**

- 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

- **变量声明**

- 可以使用 Dim、Public 或 Private 语句来声明变量

Dim Name

Dim Top,Buttom,Left,Right

Name=some value

Option Explicit ‘强制显式声明模块中的所有变量

Dim Name

Name=some value





- **为变量赋值**

- 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

```
Name = "UFT"  
T = 300
```

- **变量的作用域和生存期**

- **过程级变量**:过程中声明变量，局部作用域
- **Script级变量**：过程外声明变量，被Script中所有过程所识别
- **本地变量**：退出程序时失效

- **数组变量**

- 有时需要向一个单一的变量赋多个值
- 可以创建一个可包含一系列值的变量，这种变量被称为数组

```
Dim Names (2)  
Names(0) = "George"  
Names(1) = "John"  
Names(2) = "Thomas"
```

## 13.2.3 VBScript变量

- 大家来找茬

```
option explicit  
dim a5,b7,7cc,b7 cc  
na=3  
msgbox "Variant Test "&na&a5&cc
```

错误：

1. Line2 变量 7cc违反变量定义规则必须字符开头
2. Line2 变量b7违反变量唯一
3. Line2 变量cc之前使用的空格，应该为逗号
4. Line3 变量na在Option Explicit模式下没有显式定义

## 13.2.3 常数

常量	值	描述	常量	值	描述
vbOKOnly	0	只显示确定按钮。	vbExclamation	48	显示警告消息图标。
vbOKCancel	1	显示确定和取消按钮。	vbInformation	64	显示提示消息图标。
vbAbortRetry Ignore	2	显示终止、重试和忽略按钮。	vbDefaultButton1	0	第一个按钮是默认按钮。
vbYesNoCancel	3	显示是、否或取消按钮。	vbDefaultButton2	256	第二个按钮是默认按钮。
vbYesNo	4	显示是或否按钮。	vbDefaultButton3	512	第三个按钮是默认按钮。
vbRetryCancel	5	显示重试和取消按钮。	vbDefaultButton4	768	第四个按钮是默认按钮。
vbCritical	16	显示临界消息图标。	vbApplicationModal	0	应用程序模式。用户必须响应消息框，才能继续在当前应用程序中工作。
vbQuestion	32	显示警告询问图标。	vbSystemModal	4096	系统模式。用户响应消息框前，挂起所有应用程序。



# 13.2.4 VBScript运算符

- 运算符优先级

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
乘方	^	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
取模	Mod	大于等于	>=	逻辑隐含	Imp
加	+	对象引用比较	Is		
减	-				
字符串连接	&				

## 13.2.4 VBScript运算符

- 字符串连接符&、 +

```
Dim Str '定义一个字符串  
Str = "QuickTest"  
Str = Str & " Professional" '字符串连接  
MsgBox Str
```

```
Dim Str '定义一个字符串  
Str = "QuickTest"  
Str = Str + " Professional" '字符串连接  
MsgBox Str
```

- 注：“+”运算符可用于数值相加运算，建议使用“&”运算符来连接字符串

## 13.2.5 条件语句

- **if 语句**
  - 假如希望在条件为 true 时执行一系列的代码，可以使用这个语句
- **if...then...else 语句**
  - 假如希望执行两套代码其中之一，可以使用这个语句
- **if...then...elseif 语句**
  - 假如希望选择多套代码之一来执行，可以使用这个语句
- **select case 语句**
  - 假如希望选择多套代码之一来执行，可以使用这个语句

- 使用 If...Then...Else 进行判断
  - 条件为True时运行语句

```
Sub FixDate()  
    Dim myDate  
    myDate = #2/13/95#  
    if myDate < Now Then myDate = Now  
End Sub
```

```
Sub AlertUser(Value)  
    if value = 0 Then  
        AlertLabel.ForeColor = vbRed  
        AlertLabel.Font.Bold = True  
        AlertLabel.Font.Italic = Ture  
    End if  
End Sub
```

- **使用 If...Then...Else 进行判断**
  - 条件为 True 和 False 时分别运行某些语句

```
Sub AlertUser(Value)
    If value = 0 Then
        AlertLabel.ForeColor = vbRed
        AlertLabel.Font.Bold = True
        AlertLabel.Font.Italic = Ture
    Else
        AlertLabel.ForeColor = vbBlack
        AlertLabel.Font.Bold = False
        AlertLabel.Font.Italic = False
    End if
End Sub
```





- 使用 If...Then...Else 进行判断
  - 对多个条件进行判断

```
Sub ReportValue (value)
    If value = 0 Then
        MsgBox value
    ElseIf value = 1 Then
        MsgBox value
    ElseIf value = 2 Then
        MsgBox value
    Else
        MsgBox "数组超出范围！"
    End If
End Sub
```

- 使用 Select Case 进行判断

```
Select Case Document.Form1.CardType.Options(SelectedIndex).Text
    Case "MasterCard"
        DisplayMCLogo
        ValidateMCAccount
    Case "Visa"
        DisplayVisaLogo
        ValidateVisalAccount
    Case "American Express"
        DisplayAMEXCOLogo
        ValidateAMEXCOAccount
    Case Else
        DisplayUnknownImage
        PromptAgain
End Select
```



## 13.2.6 循环语句（ Looping语句 ）

- **For...Next 循环语句**
  - 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
- **Do...Loop 循环语句**
  - 当（或直到）条件为 True 时循环
- **While...Wend 循环语句**
  - 当条件为 True 时循环



## 13.2.6 循环语句（Looping语句）

- 使用For...Next循环语句

**'MyProc** 重复执行50次，For语句指定计数器变量x及其初始值与终止值

**'Next** 语句使计数器每次加1

```
Sub DoMyProc50Times ()
```

```
    Dim x
```

```
    For x = 1 To 50
```

```
        MyProc
```

```
    Next
```

```
End Sub
```

**'Step** 用于指定计数器变量每次增加或减少的值

```
Sub TwosTotal ()
```

```
    Dim j, total
```

```
    For j = 2 To 10 Step 2
```

```
-2
```

```
        total = total + j
```

```
    Next
```

```
    MsgBox "总和为" & total & "。"
```

```
End Sub
```

**'For j = 16 To 2 Step**

## 13.2.6 循环语句（Looping语句）

### • 使用For Each...Next循环语句

```
<HTML>
<HEAD><TITLE>窗体与元素</TITLE></HEAD>
<SCRIPT LANGUAGE="VBScript">
<!--
Sub cmdChange_OnClick
    Dim d                                '创建一个变量
    Set d =
CreateObject("Scripting.Dictionary")
    d.Add "0","Athens"                  '添加键和项
    d.Add "1","Belgrade"
    d.Add "2","Cairo"
    For Each I in d
        Document.frmFrom.Elements(I).Value =
D.Item(I)
    Next
End Sub
</SCRIPT>
```

```
<BODY>
<CENTER>
<Form Name = "frmFrom">
    <Input Type = "Text"><p>
    <Input Type = "Text"><p>
    <Input Type = "Text"><p>
    <Input Type = "Text"><p>
    <Input Type = "Button"
Name=
    "cmdChange" Value= "单击
此处"><p>
</FORM>
</CENTER>
</BODY>
</HTML>
```

注： **For Each...Next**不是将语句运行指定的次数，而是对于数组中的每个元素或对象集合中的每一项重复一组语句。这在不知道集合中元素的数目时非常有用

## 13.2.6 循环语句（Looping语句）

- 使用do...loop循环语句

```
Sub ChkFirstWhile()  
    Dim counter ,myNum  
    counter = 0  
    myNum = 20  
    Do While myNum > 10  
        myNum = myNum-1  
        counter = counter+1  
    Loop  
    MsgBox "循环重复了"& counter & "次。"  
End Sub  
Sub ChkLastWhile()  
    Dim counter ,myNum  
    counter =0  
    myNum = 9  
    Do  
        myNum = myNum - 1  
        counter = counter + 1  
    Loop While myNum >10  
    MsgBox "循环重复了"& counter & "次。"  
End Sub
```

**while**关键字用于检查  
**Do Loop**语句中的条件，  
两种检查方式：进入循  
环之前  
(如：ChkFirstWhile)

或在循环至少执行一次  
后检查条件  
(如：ChkLastWhile)

## 13.2.6 循环语句（Looping语句）

- 使用do...loop循环语句

```
Sub ChkFirstUntil()  
    Dim counter ,myNum  
    counter = 0  
    myNum = 20  
    Do Until myNum = 10  
        myNum = myNum-1  
        counter = counter+1  
    Loop  
    MsgBox "循环重复了"& counter & "次。"  
End Sub  
Sub ChkLastUntil()  
    Dim counter ,myNum  
    counter =0  
    myNum = 1  
    Do  
        myNum = myNum + 1  
        counter = counter + 1  
    Loop Until myNum = 10  
    MsgBox "循环重复了"& counter & "次。"  
End Sub
```

**Until**关键字用于检查  
**Do Loop**语句中的条件，两种检查方式：  
进入循环之前  
(如：ChkFirstUntil)

或在循环至少执行一次后检查条件  
(如：ChkLastUntil)

## 13.2.6 循环语句（Looping语句）

- 使用do...loop循环语句

```
Sub ExitExample()  
    Dim counter ,myNum  
    counter = 0  
    myNum = 9  
    Do Until myNum = 10  
        myNum = myNum-1  
        counter = counter+1  
        If myNum < 10 Then Exit Do  
    Loop  
    MsgBox "循环重复了"& counter & "次。"  
End Sub
```

**Exit Do** 语句用于退出**Do ...Loop** 循环。某些特殊情况下要退出循环（例如要避免死循环）。上面的例子汇总myNum的初始值将导致死循环



## 13.2.6 循环语句（ Looping语句 ）

- **使用 While...Wend**
  - While...Wend 语句是为那些熟悉其用法的用户提供的。但是由于 While...Wend 缺少灵活性，所以建议最好使用 Do...Loop 语句

- **使用过程函数的优势**
  - 1：简化代码，增加可读性
  - 2：覆盖已有的对象方法
  - 3：使脚本中的引用代码可重用或使用恢复场景
- **在VBScript脚本中，过程分为：Sub过程和Function过程**
  - Sub 过程没有返回值
  - Function 过程可以有返回值

- **Sub 过程**

- Sub 过程是包含在 Sub 和 End Sub 语句之间的一组 VBScript 语句
- 可以使用Sub...End Sub语句来创建子过程

```

'=====
' Purpose: Cleans up the given RecordSet and Connection.
' Inputs: objDBConnection: the Connection to be closed/freed.
'         objResults: the RecordSet to be closed/freed.
'=====
Sub CleanUpDBConnection(objDBConnection, objResults)
    If (objResults <> Nothing) Then
        objResults.Close
        Set objResults = Nothing
    End If
    If (objDBConnection <> Nothing) Then
        objDBConnection.Close
        Set objDBConnection = Nothing
    End If
End Sub

```

## • 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）
- 可以使用 Function 和 End Function 语句来创建函数

```
.....  
' Purpose: Run a SQL query and return the RecordSet.  
' Inputs: strQuery: the query to be run.  
'         objDBConnection: the Connection on which to run the query.  
' Result: The RecordSet returned by executing the query.  
.....
```

```
Function ExecuteQuery(strQuery, objDBConnection)  
    If (objDBConnection.State <> adStateOpen) Then  
        objDBConnection.Open 'assumes ConnectionString property is set  
    End If  
    ExecuteQuery = objResults.Execute(strQuery)  
End Function
```

- **创建新过程的步骤**
  - 为过程添加基本的步骤
  - 创建 过程声明
  - 定义过程参数
  - 设定过程返回值
  - 处理过程中的错误

- 创建新过程的步骤
  - 为过程添加基本的步骤

```
Window("Flight Reservation").WinButton("Button").Click  
Window("Flight Reservation").WinButton("Button_2").Click  
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"  
Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set "6"  
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click
```



- 创建新过程的步骤
  - 创建 过程声明

```
Function OpenByOrderNum|
```

```
Window("Flight Reservation").WinButton("Button").Click  
Window("Flight Reservation").WinButton("Button_2").Click  
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"  
Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set "6"  
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click
```

```
End Function
```

## 13.2.7 VBScript过程

- 创建新过程的步骤
  - 定义过程参数

```
Function OpenByOrderNum(ByVal OrderNum)  
Window("Flight Reservation").WinButton("Button").Click  
Window("Flight Reservation").WinButton("Button_2").Click  
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"  
Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set OrderNum  
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click  
  
End Function
```



- 创建新过程的步骤
  - 设定过程返回值

```
Function OpenByOrderNum(ByVal OrderNum)

Window("Flight Reservation").WinButton("Button").Click
Window("Flight Reservation").WinButton("Button_2").Click
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"
Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set OrderNum
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

OpenByOrderNum = True

End Function
```

- 创建新过程的步骤
  - 处理过程中的错误

```
Function OpenByOrderNum(ByVal OrderNum)

Window("Flight Reservation").WinButton("Button").Click
Window("Flight Reservation").WinButton("Button_2").Click
Window("Flight Reservation").Dialog("Open Order").WinCheckBox("Order No.").Set "ON"
Window("Flight Reservation").Dialog("Open Order").WinEdit("Edit").Set OrderNum
Window("Flight Reservation").Dialog("Open Order").WinButton("OK").Click

If Window("Flight Reservation").Dialog("Open Order").Dialog("FlightReservations").Exist Then
    Window("Flight Reservation").Dialog("Open Order").Dialog("FlightReservations").WinButton("OK").Click
    OpenByOrderNum=false
else
    OpenByOrderNum = True
End If

End Function
```

- **过程的数据进出**

- 给过程传递数据的途径是使用参数。参数被作为要传递给过程的数据的占位符

```
Function Celsius(fDegrees)  
Celsius = (fDegrees - 32)*5/9  
End Function
```

要从过程获取数据，必须使用**Function**过程

调用**Function**过程时，函数名必须用在变量赋值语句的右端或表达式中

```
Temp = Celsius(fDegrees)
```

或

```
MsgBox “温度为”& Celsius(fDegrees)& “摄氏度。”
```

调用**sub**过程时，只需输入过程名称及所有参数值。使用**call**时，参数需要包含在括号中

```
Call MyProc (firstarg,secondarg)
```

```
MyProc firstarg,secondarg
```

# 补充：VBScript中常用函数

- 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

- 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
  - CDbI 函数
  - CInt 函数
  - CLng 函数
  - CSng 函数
  - CStr 函数

- 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

- 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
```