

VBScript

网页制作快速入门

施波 杨清 等编著

网页制作快速入门丛书

3.092

新时代出版社

VBScript 网页制作快速入门

施波 杨清 等编著

新 时 代 出 版 社

·北京·

图书在版编目(CIP)数据

VBScript 网页制作快速入门/施波等编著. —北京:
新时代出版社, 2000. 8
(网页制作软件快速入门丛书)
ISBN 7-5042-0497-8

I. V... II. 施... III. 主页-制作
IV. TP393.092

中国版本图书馆 CIP 数据核字(2000)第 28040 号

新时代出版社出版发行

(北京市海淀区紫竹院南路 23 号)

(邮政编码 100044)

北京奥隆印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 印张 16¼ 397 千字

2000 年 8 月第 1 版 2000 年 8 月北京第 1 次印刷

印数:1—4000 册 定价:22.00 元

(本书如有印装错误,我社负责调换)

前 言

近年来,随着计算机网络应用技术的发展,我国的网站数量在不断地以惊人的速度增加,网页制作已经成为热门技术。为了使自己编辑的主页更加吸引人,网页开发者在不断地学习新技术。看到网上五花八门的主页形式,新手也在跃跃欲试。

创建交互式网页内容是许多 Web 开发者的首要目标,这可用多种办法实现。Web 浏览器通过使用视频剪辑、声音和可下载程序能够具有添加动态内容的能力。尽管其他交互式方法也能够提供吸引人的内容,但脚本语言给试图创建交互式网页的开发者提供了非常便利、有效的条件。脚本语言的最大优势在于它们易于使用。由于脚本语言直接包含在 HTML 页面中,是用纯文本创建的,创建和修改都很简单。开发者不必忍受编译和调试过程的折磨。脚本语言可以满足一个 Web 站点所要求的大多数基本交互功能,如数据验证,还可以很容易地提供交互经验,如游戏。当前主要有两种脚本语言:JavaScript 和 VBScript。

VBScript 用于创建 Internet Explorer 浏览器支持的交互式 Web 网页。Microsoft 正致力于使其他浏览器也支持 VBScript,而且现在较流行的 ASP 技术支持 VBScript,可以说 VBScript 方兴未艾。

本书分两篇(基础篇和提高篇),内容既简单易懂又全面、系统。各个章节都提供大量的典型范例。循序渐进,图文并茂的讲解,力图使读者尽快掌握 VBScript 的应用技术,并熟悉它们的全貌。我相信本书会成为 VBScript 初、中级学习者很好的参考书。

网络技术发展之快令人吃惊,网络技术产品不断地推陈出新,脚本语言本身可能会相对稳定,但是新特性会不断地添加到 Microsoft Internet Explorer 浏览器等相关宿主产品中。尽管作者在本书的写作过程中不断地关注网页开发的相关技术动态,还是希望广大读者不断学习,及时掌握新技术,让你的主页与众不同!

本书由施波、杨清编写,战晓苏、汪钟鸣、杨岳湘审校。参加本书编写的还有施磊、徐明、施晗、刘淑苹、张志刚、孙俞、曹军、金博。此外,周立、吴阿明、任鸿、王峰、康建鹏、李跃华、郭伟参加了资料收集和实例编写工作,董朝旭、王英、何建宇、陶涛、周利完成了录排等大量的实际工作。

对于本书的内容和编排作者下了一番苦心,但是由于编写时间紧迫,书中疏漏之处在所难免,恳请读者提出宝贵意见。

编 著 者
2000 年 3 月

第 1 篇 基础篇

第 1 章 VBScript 概述

也许你已经听说过 VBScript,也许 VBScript 对你是一个全新的概念。为了深入学习或者活学活用 VBScript,最好是先搞懂它的概念。本章介绍了 VBScript 脚本语言的概念、功能、特点和运行环境等基本知识,使你对 VBScript 脚本语言及其相关技术知识产生一个总的认识。

1.1 什么是 VBScript?

VBScript 是一种脚本编写语言。那么,什么是脚本编写语言呢?可以把脚本编写语言定义为任何一套可嵌入 HTML 页面并可被浏览器执行的程序指令。这些语言使用简单,直接包含在 HTML 页面中,用纯 ASCII 文本创建,没有一般语言的编译和调试过程。脚本编写语言可以满足一个 Web 站点所要求的大多数基本交互功能。

用脚本语言可以使你的网页达到以下效果:

• 在用户来到或离开你的网页时,在鼠标指针经过网页中的图形或超链接时,VBScript 可以在浏览器的状态栏中显示相关信息。

- 自动显示当前日期。
- 使网页改变显示格式。
- 使网页显示用户的信息。
- 创建旋转、颜色变化的文字和图形。
- 创建定制的信息框、对话框。
- 自动显示文档的目录。
- 提供实时计算。

当前有两种主要脚本编写语言:一个是 JavaScript,另一个就是 VBScript。

VBScript 不同于 Visual Basic,不是用户产品。它是一个能够嵌入到 Web 浏览器中的解释性语言,一个配备了解释器的 Web 浏览器能够运行嵌入到 HTML 页面中的 VBScript 程序。Microsoft Internet Explorer 中包括 VBScript 解释器,并且 Microsoft 正致力于使其他的浏览器也可以容易地使用 VBScript。

VBScript 不是 Java,也不是 ActiveX,更不是 CGI/Perl 的替代品。VBScript 脚本程序用于处理及使用 HTML 页面里的各种对象,如按钮对象、表单对象和 ActiveX 控件等。

Microsoft VBScript 是程序语言 Visual Basic 家族的最新成员,这一交互式脚本编程语言可应用于各种环境,包括 Microsoft Internet Explorer 中的网络客户端和 Microsoft Internet Information Server 的网络服务器。

如果你已经知道 Visual Basic 或 Visual Basic for Applications,你会非常熟悉 VBScript。即使你不懂 Visual Basic,一旦学习了 VBScript,你就是在学习用 Visual Basic 家族的语言进行编程了。你也可以通过几个网页学习使用 VBScript,而不用学习真正的编程。

1.2 VBScript 与 ActiveX Scripting 的关系

为了开放脚本语言,提供一个完全交互的环境,Microsoft 已经为在 HTML 页面中包含任意的脚本语言引入了一个标准,这个标准称作 ActiveX Scripting。ActiveX Scripting 是一个独立于语言的标准,它定义了脚本宿主(Scripting Host)和脚本引擎(Scripting Engine)之间的关系。脚本引擎就是脚本解释器,是符合 ActiveX Scripting 规范的进程内 ActiveX 组件(以前称作 OLE 服务器)。脚本宿主是一个使用脚本引擎的应用程序。Microsoft Internet Explorer 就是一个脚本宿主。

VBScript 通过 ActiveX Scripting 与脚本宿主应用程序通信。有了 ActiveX Scripting,浏览器和其他脚本宿主应用程序无须为每种脚本组件特增专门的综合代码。ActiveX Scripting 启动宿主应用程序编译脚本,获得并调用入口点。有了 ActiveX Scripting,语言开发商可以为脚本编程创建标准的实时语言。Microsoft 将提供对 VBScript 的实时支持。Microsoft 正在与各种 Internet 组织共同定义 ActiveX Scripting 标准,以便脚本编写引擎可以相互转换。ActiveX Scripting 用于 Microsoft Internet Explorer 和 Microsoft Internet Information Server。图 1.1 显示了 ActiveX Scripting 与脚本宿主和脚本引擎的关系。

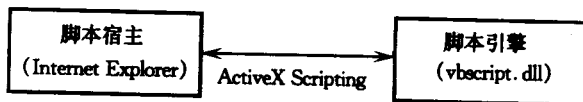


图 1.1 ActiveX Scripting 与脚本宿主和脚本引擎的关系

1.3 VBScript 与 JavaScript 的比较

VBScript 与 JavaScript 是目前最流行的两种脚本语言。二者有许多相似之处,它们的运行机制是一样的。不要将 JavaScript 与 JScript 混为一谈,JScript 是 Microsoft 为了解决 Netscape 浏览器的支持问题而开发的与 JavaScript 相兼容的脚本语言。VBScript 与 JavaScript 的主要区别表现在如下几个方面。

1. 语法

VBScript 与 JavaScript 的最主要区别就是语法。VBScript 用 Function...End Function 定义函数过程,用 Sub...End Sub 定义子程序过程;JavaScript 用大括号定义函数过程,它不支持子程序过程。JavaScript 采用松散的数据类型,变量可以不必定义直接使用;VBScript 采用复合的单一的 Variant(变体)数据类型,提供 Option Explicit 语句强制声明变量,避免变量拼写错误带来的麻烦,变量以 DIM 语句定义后使用。

2. 脚本宿主

Microsoft Internet Explorer 和 Netscape 都是脚本宿主,它们都支持 JavaScript;只有 Microsoft Internet Explorer 支持 VBScript。

3. 学习

VBScript 是 Visual Basic 家族的成员,而且是 Visual Basic for Applications 的一个子集。如果你会 Microsoft Visual Basic 或者 Visual Basic for Applications,你也就学会了 VBScript。JavaScript 和 C++ 有相似之处,但它是一种新的语言。

下面通过实例看看 VBScript 与 JavaScript 的异同。我要分别用这两个脚本实现一个 HTML 页面,该页面在按钮“单击这里”被单击时在一个消息对话框中显示“你好!”。用 Internet Explorer 3.0 浏览并单击按钮的效果如图 1.2 所示。

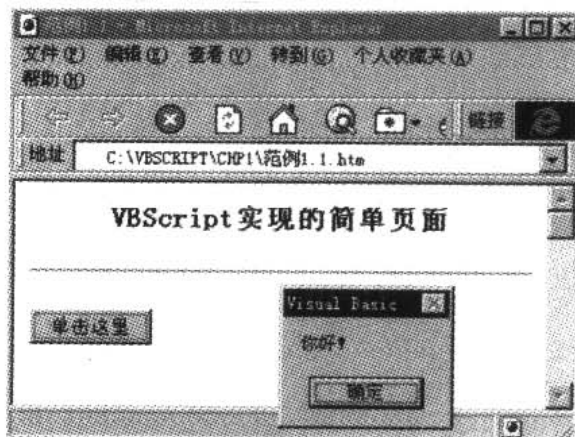


图 1.2 一个 HTML 页面的浏览效果

范例 1.1

使用 VBScript 实现这样的页面:

- (1) 将清单 1.1 保存为文件“范例 1.1.htm”。
- (2) 浏览文件“范例 1.1.htm”,结果与图 1.2 一样。

清单 1.1

```
<HTML>
<HEAD>
<TITLE>范例 1.1</TITLE>

<SCRIPT LANGUAGE="VBScript">
<!--
    Sub Button1_OnClick
        MsgBox "你好!"
    End Sub
-->
</SCRIPT>

</HEAD>
```

```

<BODY>
<CENTER>
<H3>VBScript 实现的简单页面</H3>
<HR>
</CENTER>
<FORM>
<INPUT TYPE="Button" NAME="Button1" VALUE="单击这里">
</FORM>
</BODY>
</HTML>

```

范例 1.2

使用 JavaScript 实现这样的页面：

- (1) 将清单 1.2 保存为文件“范例 1.2.htm”。
- (2) 浏览文件“范例 1.2.htm”，结果与图 1.2 进行比较。

清单 1.2

```

<HTML>
<HEAD>
<TITLE>范例 1.2</TITLE>

<SCRIPT LANGUAGE="JavaScript">
<! --
    var rtn = false
    function click_here() {
        alert("你好!")
        return true
    }
-->
</SCRIPT>

</HEAD>
<BODY>
<CENTER>
<H3>JavaScript 实现的简单页面</H3>
<HR>
</CENTER>
<FORM>
<INPUT NAME="Button1" TYPE="BUTTON" VALUE="单击这里"
OnClick="rtn= click_here()">

```

```
</FORM>
</BODY>
</HTML>
```

从上面两个清单可以看出,VBScript 和 JavaScript 的主要区别在 < SCRIPT > 和 < /SCRIPT > 标记之间的内容,也就是脚本语言的语法。

实际上,在一个 HTML 页面中可以同时使用 VBScript 和 JavaScript。清单 1.3 显示了一个 HTML 页面中同时使用 VBScript 和 JavaScript 的 HTML 代码。图 1.3 是该代码通过 Internet Explorer 4.0 浏览的输出效果。此时读者可能不能完全看懂这些代码,但学习了本书的相关章节后,这些代码对你将是“小菜一碟”。

范例 1.3

在一个网页中同时使用 VBScript 和 JavaScript:

- (1) 将清单 1.3 所示代码保存为文件“范例 1.3.htm”。
- (2) 浏览文件“范例 1.3.htm”,结果如图 1.3 所示。

清单 1.3

```
<HTML>
<HEAD>
<TITLE>范例 1.3 </TITLE>

<SCRIPT LANGUAGE="VBScript">
<! --
    Sub Button1_OnClick
        MsgBox "你启动了 VBScript!"
    End Sub
-->
</SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
<! --
    var rtn = false
    function click_here() {
        alert("你启动了 JavaScript!")
        return true
    }
-->
</SCRIPT>

</HEAD>
```

```

<BODY>
<CENTER>
<H3>在一个网页中同时使用 VBScript 和 JavaScript </H3>
<HR>
</CENTER>

<FORM>
<INPUT TYPE="Button" NAME="Button1" VALUE="单击这里启动 VBScript">
<P>
<INPUT TYPE="Button" NAME="Button2" VALUE="单击这里启动 JavaScript"
OnClick="rtm= click here()">
</FORM>
</BODY>
</HTML>

```



图 1.3 清单 1.3 所示代码的浏览效果

1.4 VBScript 与 HTML

VBScript 的主要任务就是处理网页对象,加强网页的机动性。有关网页对象在第 3 章有专门的介绍。VBScript 必须嵌入 HTML 页面才能够起作用。怎样在 HTML 文档中加入 VBScript 代码呢?

VBScript 代码写在 HTML 的 <SCRIPT> 和 </SCRIPT> 标记对之间。例如,测试发货日期的过程代码如下:

```

<SCRIPT LANGUAGE="VBScript">
<! --
    Function CanDeliver(Dt)
        CanDeliver = (CDate(Dt) - Now()) > 2
    End Function

```

```
-->
</SCRIPT>
```

<SCRIPT>和</SCRIPT> 标记包住了代码。LANGUAGE 属性指明脚本编写语言。因为浏览器可以使用其他脚本语言,你必须指明所使用的脚本语言。注意,CanDeliver 函数包含在注释标记(<!-- 和 -->)之间,这样是为了防止不支持<SCRIPT>标记的浏览器将这一段代码本身显示出来。

由于这一例子中是通用函数,不绑定任何特定的表单控件,你可以将它放在 HTML 页的 HEAD 节。代码如下:

```
<HTML>
<HEAD>
<TITLE>订购单</TITLE>
<SCRIPT LANGUAGE="VBScript">
<!--
    Function CanDeliver(Dt)
        CanDeliver = (CDate(Dt) - Now()) > 2
    End Function
-->
</SCRIPT>
</HEAD>
<BODY>
...
```

可以在 HTML 页中的任何位置使用 SCRIPT 块。你既可以将它们放在 BODY 节,又可以将它们放在 HEAD 节(我试过,将它们放在 HEAD 节与 BODY 节之间也可以)。然而最好将所有的通用脚本代码放在 HEAD 节,以便将所有的代码放在一起。这样可以保证在 BODY 节的任何调用之前浏览器已经读取了所有代码并进行了译码。

这一规则也有一个值得注意的例外:你可能要提供对表单内对象的事件作出反应的行内脚本代码。例如,你可以在一个表单中嵌入对单击按钮事件作出反应的一段代码。请看下例。

范例 1.4

练习使用行内脚本代码:

- (1) 将清单 1.4 保存为文件“范例 1.4.htm”。
- (2) 浏览文件“范例 1.4.htm”,效果同范例 1.1 一样(单击按钮弹出“你好!”对话框)。

清单 1.4

```
<HTML>
<HEAD>
<TITLE>范例 1.4</TITLE>
```

```

</HEAD>
<BODY>

<CENTER>
<H3>练习使用行内脚本代码</H3>
<HR>
</CENTER>

<FORM NAME="Form1">
  <INPUT TYPE="Button" NAME="Button1" VALUE="单击这里">
  <SCRIPT FOR="Button1" EVENT="OnClick" LANGUAGE="VBScript">
    MsgBox "你好!"
  </SCRIPT>
</FORM>
</BODY>
</HTML>

```

大多数情况下都将代码放在 Sub 或 Function 过程中,仅当你所写的代码调用它时才执行。然而,也可以在过程之外写 VBScript 脚本代码,但是一定要在<SCRIPT>块中。这样的代码只在加载 HTML 页时执行一次,它使你能够在加载网页时初始化数据或动态地改变网页的面貌。下例在页面加载时背景变为红色并弹出对话框,用户没有其他办法再次启动这个对话框。

范例 1.5

- (1) 将清单 1.5 保存为文件“范例 1.5.htm”。
- (2) 浏览文件“范例 1.5.htm”,结果如图 1.4 所示。

清单 1.5

```

<HTML>
<HEAD>
<TITLE>范例 1.5</TITLE>
</HEAD>
<BODY>
<CENTER>
<H3>在过程之外写 VBScript 脚本代码</H3>
<HR>
网页的其他内容放在这里
</CENTER>

<SCRIPT LANGUAGE="VBScript">

```

```

Document.bgColor="red" '新的背景颜色为红色
MsgBox "你好!"
</SCRIPT>
</BODY>
</HTML>

```

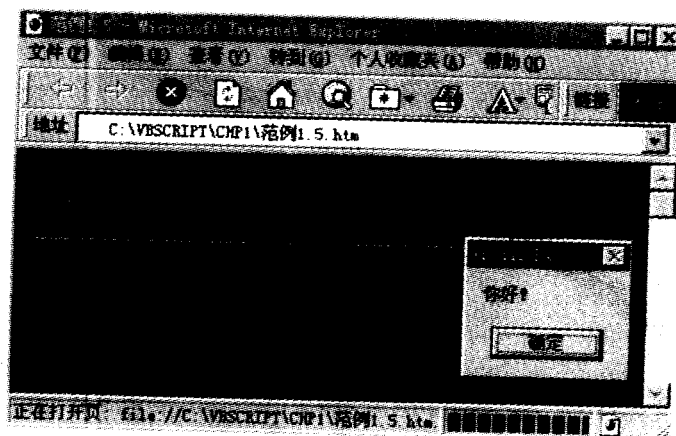


图 1.4 过程外的脚本代码只在网页加载时执行一次

1.5 其他应用程序和浏览器中的 VBScript

作为开发者,你可以免费将 VBScript 源代码应用于你的产品中。Microsoft 为 32 位 Windows API、16 位 Windows API 和 Macintosh 提供了 VBScript 的二进制实现。VBScript 集成于网络浏览器中。VBScript 和 ActiveX Scripting 也可以在其他应用程序中用作通用脚本编写语言。

1.6 VBScript 的版本

现在,VBScript 从 1.0 版到 5.0 版,功能越来越强大,高版本增加了很多内置函数。由于网络浏览技术还在发展之中,VBScript 与宿主应用程序之间存在版本对应问题。特别是在我们的实践中还发现在低版本的宿主应用程序(如 Microsoft Internet Explorer 3.0)运行通过的脚本代码在高版本的宿主应用程序(如 Microsoft Internet Explorer 4.0/5.0)上却不能通过或者变了样子。对此,读者要有清醒的认识。

表 1.1 列出了 VBScript 版本与宿主应用程序的对应关系,供读者参考。

表 1.1 VBScript 版本与宿主应用程序的对应关系

宿主应用程序	VBScript 版本				
	1.0	2.0	3.0	4.0	5.0
Microsoft Internet Explorer 3.0	x				
Microsoft Internet Information Server 3.0		x			

(续)

宿主应用程序	VBScript 版本				
	1.0	2.0	3.0	4.0	5.0
Microsoft Internet Explorer 4.0			x		
Microsoft Internet Information Server 4.0			x		
Microsoft Windows Scripting Host 1.0			x		
Microsoft Outlook 98			x		
Microsoft Visual Studio 6.0				x	
Microsoft Internet Explorer 5.0					x
Microsoft Internet Information Server 5.0					x

第 2 章 VBScript 语言基础

本章是为那些没有语言基础的读者写的,如果你已经熟悉 Visual Basic 或 Visual Basic for Applications,你可以越过此章。如果你有其他语言的编程基础,你会觉得学习本章很轻松。

在介绍 VBScript 的语言基础知识之前,让我们先来看一个例子。

范例 2.1

用 VBScript 进行简单的数学运算。

(1) 将清单 2.1 所示的代码保存为文件“范例 2.1.htm”。

(2) 用 Internet Explorer 浏览文件“范例 2.1.htm”,在出现的前两个文本框中输入任意数字,然后单击“运算”按钮,结果在第三个文本框中显示两个数的乘积,如图 2.1 所示。

清单 2.1

```
<HTML>
<HEAD>
<TITLE>范例 2.1</TITLE>

<SCRIPT LANGUAGE="VBScript">
<! --
    Sub Button1_OnClick
        Dim Num1, Num2, Result
        Num1 = Form1.Text1.Value
        Num2 = Form1.Text2.Value
        Result = Num1 * Num2
        Form1.Text3.Value = Result
    End Sub

    Sub Button1_OnClick
        Form1.Text1.Value = ""
        Form1.Text2.Value = ""
        Form1.Text3.Value = ""
    End Sub
-->
</SCRIPT>

</HEAD>
```

```

<BODY>
<CENTER>
<H3>用 VBScript 进行简单的数学运算</H3>
<HR>

```

```

<FORM NAME="Form1">
<INPUT TYPE="Text" NAME="Text1" SIZE="4" MAXLENGTH="4" VALUE="0">
乘以
<INPUT TYPE="Text" NAME="Text2" SIZE="4" MAXLENGTH="4" VALUE="0">
等于
<INPUT TYPE="Text" NAME="Text3" SIZE="10" MAXLENGTH="10" VALUE="0">
<P>
<INPUT TYPE="Button" NAME="Button1" VALUE="运算">
<INPUT TYPE="Button" NAME="Button2" VALUE="清除">

</FORM>
</CENTER>
</BODY>
</HTML>

```

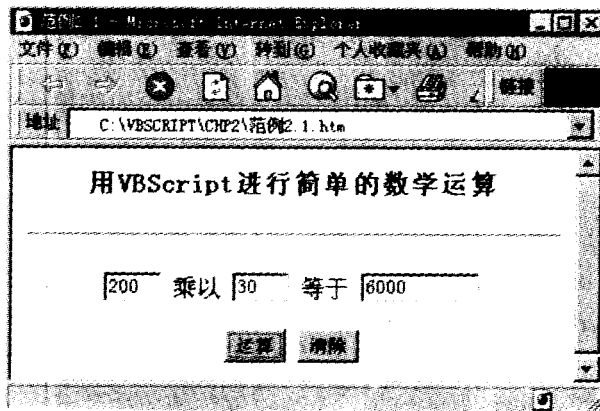


图 2.1 用 VBScript 可以进行数学运算

本例不是一个实际的例子,但是它包含了 VBScript 语言的很多知识。总代码虽然很长,但我们要关注的 VBScript 代码并不多,都在<SCRIPT>、</SCRIPT>标记对中。

本例有两个子程序,Button1_OnClick 和 Button2_OnClick。它们分别处理“Button1”(运算按钮)和 Button2(清除按钮)的单击事件。这里分析一下 Button1_OnClick 过程(为了说明方便,语句之前加上了数字标号):

```

Sub Button1_OnClick
1 Dim Num1, Num2, Result
2 Num1 = Form1.Text1.Value
3 Num2 = Form1.Text2.Value

```

```

4 Result = Num1 * Num2
5 Form1.Text3.Value = Result
End Sub

```

第 1 条语句定义了 3 个变量 Num1、Num2、Result。

第 2 条语句和第 3 条语句分别为变量 Num1 和 Num2 赋值。

第 4 条语句将变量 Num1 与 Num2 相乘后赋给变量 Result。

第 5 条语句将结果赋给第三个文本框。

本例虽然简单,但是也反映了 VBScript 作为一种语言具有计算机语言的基本语法要求。

2.1 VBScript 的数据类型

2.1.1 什么是 VBScript 数据类型?

VBScript 只有一种叫做 Variant(一般翻译为“变体”)的数据类型。Variant 是一种特殊的数据类型,它可以包含不同类型的数据,究竟是什么类型的数据取决于怎样使用这个数据。由于 Variant 是 VBScript 中的唯一一种数据类型,VBScript 函数返回的数据也都是 Variant 类型。

最简单的情况是 Variant 可以包含数字或字符串信息。当用于数字环境时它就是一个数字,用于字符串环境时它就是一个字符串。也就是说,如果你在用看起来像数字的数据工作时,VBScript 就假定这个 Variant 是数字并使其做适合数字的操作或运算。同样地,如果你在用字符串数据工作,VBScript 就以字符串来对待这个 Variant 数据。注意,用引号(“”)引起来的数字是字符串。

2.1.2 VBScript 的数据子类型

除了简单的数字或字符串类型之外,Variant 可以进一步区分数字信息的特性。例如,可以用数字信息代表日期或时间。当与其他日期或时间数据一起使用时,结果总是表示为日期或时间。当然,还有很多种数字信息,从布尔型到浮点数类型。Variant 可以包括的这些数据类型叫做 VBScript 数据子类型。多数情况下只在 Variant 中使用一种数据类型,Variant 会以最适合这种数据类型的方式参加运作。

表 2.1 列出了 Variant 可能包含的数据子类型。

表 2.1 VBScript 的数据子类型

数据子类型	说 明
Empty	Variant 没有初始化,对于数字变量是 0,字符串变量为零长度字符串(“”)
Null	Variant intentionally 包含非法数据
Boolean	包含 True 或 False 值
Byte	包含 0 到 255 的整数值
Integer	包含 - 32768 到 32767 的整数值
Currency	- 922337203685477.5808 到 922337203685477.5807
Long	包含 - 2147483648 到 2147483647 的整数值