

PROGRAMMER TO PROGRAMMER™



VBScript Programmer's Reference Third Edition

# VBScript

# 程序员参考手册

(第3版)

Adrian Kingsley-Hughes  
(美) Kathie Kingsley-Hughes 著  
Daniel Read  
富弘毅 陈钢 译



清华大学出版社

# VBScript 程序员参考手册 (第 3 版)

Adrian Kingsley-Hughes

(美) Kathie Kingsley-Hughes 著

Daniel Read

富弘毅 陈 钢 译

清华大学出版社

北 京

Adrian Kingsley-Hughes, Kathie Kingsley-Hughes, Daniel Read

VBScript Programmer's Reference, Third Edition

EISBN: 978-0-470-16808-0

Copyright © 2007 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2008-2279

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

#### 图书在版编目(CIP)数据

VBScript 程序员参考手册(第3版)/(美) 金斯利-休斯(Kingsley-Hughes, A.), (美) 金斯利-休斯(Kingsley-Hughes, K.), (美) 瑞德(Read, D.) 著; 富弘毅, 陈钢 译. —北京: 清华大学出版社, 2009.2

书名原文: VBScript Programmer's Reference, Third Edition

ISBN 978-7-302-19464-4

I.V… II.①金…②金…③瑞…④富…⑤陈… III. BASIC 语言—程序设计—技术手册 IV.TP312-62

中国版本图书馆 CIP 数据核字(2009)第 010677 号

责任编辑: 王 军 梁卫红

装帧设计: 康 博

责任校对: 胡雁翎

责任印制: 李红英

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 清华大学印刷厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 44.5 字 数: 1083 千字

版 次: 2009 年 2 月第 1 版 印 次: 2009 年 2 月第 1 次印刷

印 数: 1~4000

定 价: 98.00 元

---

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 027691-01

# 译者序

Wrox 红皮书系列一直以来都是技术图书市场上的一个口碑不错的品牌,其中无论是入门书籍还是针对高级开发人员的进阶书籍都诞生了不少经典的作品,在 China-Pub、程序员等网络或传统媒体的书籍排行榜上也经常可以看到该系列书籍的身影。

Basic 语言家族是微软公司长期支持的开发语言,从最早的 Basic 到 Visual Basic 和 Visual Basic .NET,一直发展到最新的 Visual Basic 2008。尽管在这个过程中在微软公司的开发平台上诞生了许多新的开发语言,比如 C#、J#,但 Basic 始终是最重要的语言之一。TIOBE 公司每月发布的编程语言排名上也可以看出该语言家族与日俱增的影响力。同时脚本语言,比如 Ruby、Python、Perl 的兴起无疑是当前开发领域的一大趋势和热点。因此 VBScript 作为该家族的一员,同时也是 Windows 平台上不多的原生脚本语言,自然也值得每一个对微软平台感兴趣的管理和开发人员学习。

本书是国内屈指可数的针对最新的 Windows Script 5.7 中 VBScript 的书籍。最新版的 Windows Script 中最大的特色是提供了很多专门针对 Windows Vista、IIS7 和 Internet Explorer 7 的功能。这不仅延续了 VBScript 语言在 Windows 平台上的使用,还极大地扩展了 VBScript 语言的功能,使其成为值得所有 Windows 平台管理人员和开发人员学习的编程语言。本书的特色在于通过大量有针对性的真实例子,由浅入深地介绍了最新版 VBScript 在包括 Windows Vista、IIS7、IE7 在内的各种脚本宿主上的实际使用,其中包括了 Windows 系统管理、Web 客户端和服务端编程、Vista 桌面小工具等各种引人入胜的应用。无论是编程新手还是经验丰富的程序员都能通过这些例子迅速地掌握最新的 VBScript。

本书由富弘毅和陈钢翻译。Be Flying 工作室负责人肖国尊对本书的翻译和出版做了大量的协调和规范工作,负责本书译员选定、翻译指导、进度和质量控制,在此予以衷心感谢。同时欢迎各位光顾 Be Flying 工作室博客: [http://blog.csdn.net/be\\_flying](http://blog.csdn.net/be_flying) 及其所译图书宣传页面: <http://www.china-pub.com/main/sale/renwu/luminary.asp?id=64>。

陈 钢

于长沙市中南大学计算机楼

# 前 言

我们希望，能够在—个文本编辑器中快速而简便地编写代码，而不用考虑任何复杂的开发环境。

我们希望，不用牵绊于编译代码或发布复杂的安装程序的繁杂工作。

我们希望，能够以多种方式部署代码。

我们还希望，只要学习—种语言就可以完成服务器端 Internet、客户端 Internet 和桌面程序的编码。

实际上这不仅仅是希望，使用 VBScript 能够实现的不仅是这些。

VBScript 是—种绝对高级的语言，甚至可以用它来“说话”。它易于学习，功能强大，灵活性强，而且很便宜。这使得 VBScript 无论对于经验丰富还是初出茅庐的程序员来说，都成为其首选语言。如果您是经验丰富的程序员，就会享受到由于不需要复杂的开发环境和编译的需要而带来的编码乐趣。如果您是初学者，那么只要懂—点文本编辑器的知识，就可以开始学习编程。

VBScript 的知识和经验还将开启很多技术的大门。如果有很好的 VBScript 编程基础，就可以涉足 Internet 开发、网络管理、服务器端编码甚至其他编程语言的使用(Visual Basic 是—条最流行的路线，因为它的语法与 VBScript 非常相似)。使用 VBScript，还可以创建看上去像使用 C++ 这样复杂的编程语言编写的程序。还有一点值得注意的是，对脚本的支持已经嵌入到所有新版本的 Windows 操作系统中——这对于帮助了解 VBScript 的工作机制来说，无疑是一种便利条件。通过在文本编辑器中编写—些简单的脚本，就可以完成很多工作，如复制和移动文件、创建文件夹和文件、修改 Windows 注册表，还有很多很多。—种易于使用的脚本语言居然无所不能。

我们相信，了解如何编写 VBScript 应用程序对很多人来说都将是—种有用且有益的技能，无论他们是 IT 行业用户、SOHO PC 用户、学生还是家庭用户。了解和使用 VBScript 可以节省时间，更重要的是，节省金钱。

## 本书面向的读者

对于所有对 VBScript 学习有兴趣的人来说，这都是一本—站式的书籍。对本书的使用方式取决于读者已有的程序设计和脚本编写的知识与经验：

- 如果您是一个纯粹的初学者，目前只是听说过 VBScript，那么好极了，这本书是最适合您的。对于—个初学者，这将是—个引人入胜的学习过程。我们假设您能够从头到尾通读这本书，这样才能充分地利用它进行学习。
- 如果您已经具有 IT 和程序设计经验，只是想要学习 VBScript(可能是要用于 ASP (Active Server Pages, 活动服务器页面)或 WSH(Windows Scripting Host, Windows

脚本宿主)),那么这本书也适合您。您对程序设计的了解意味着您对书中涉及的大多数术语和技术都比较熟悉。因此对您来说,学习另一种语言是比较简单的。如果了解使用 VBScript 的目的(例如 ASP 和 WSH),那么就请带着这个目的阅读本书,跳过某些章节以节省时间。

- 网络管理员可能会发现本书不仅十分有用,而且还能大幅度节省他们的时间,因为他们经常会使用 VBScript 编写功能强大的登录脚本或自动化那些在 WSH 下遇到的烦琐的、重复性的、耗时巨大的、易于出错的任务。
- 如果您已经使用 VBScript,而只是想填补一些知识空白,或者只是为了跟踪最新技术而购买本书,您也无疑会从本书特定章节中发现新的信息,而对这些章节(比如在改版的附录中关于 Windows Vista 的相关内容)予以特别的关注。

## 本书涵盖的内容

正如您所期望的,关于 VBScript 的书只讨论 VBScript。准确地说,本书只讨论 VBScript 的最新版本(版本 5.7)。但是,VBScript 作为一种工具,可以有多种使用方法,可以用于多种不同的应用程序。因此,在详细讨论 VBScript 的同时,本书还涉及一些与 VBScript 相关的技术。这包括 Windows 脚本宿主(WSH)。同样地,如果您具有 Visual Basic 的知识,那么前三章中绝大部分内容(变量、数据类型、过程、控制流等)对您来说都很熟悉。本书将为您介绍如何深入到 Windows 操作系统内部,通过几行代码实现操作系统环境的改变。

## 本书组织方式

只要快速浏览一下这一版的目录,就会发现本书分为三个部分:

- 首先是关于 VBScript 核心内容的一些章节——基本上是关于 VBScript 作为一种语言的工作机制。
- 接下来着眼于如何在其他技术(如 WSH 或 ASP)内部使用 VBScript。这些章节中分析了一些更高级的 VBScript 脚本代码示例的运行。
- 最后以一系列附录的形式提供了详细而全面的参考信息。这些参考信息可以作为独立的部分来使用,也可以通过它们深入了解前面的章节中介绍的 VBScript 的工作机制。

以怎样的计划阅读本书实际取决于您当前 VBScript 和其他编程语言的技术水平以及您的目的。这是您的书——请使用最适于您的方式来阅读!

如果不能确定阅读本书的最佳方式,那么我们建议从头至尾地通读,这样就能最大程度地受益。不必担心记不住所有读到的内容——那并不是重点。本书是一本参考手册,这意味着您可以反复参考。在阅读的时候请做一些笔记,这将帮助您更好地记住相应内容,并便于发现曾经阅读过的重点部分。

## 使用本书需要的准备工作

VBScript 对于大部分脚本编写和/程序设计需求来说，可能是一种低代价的解决方案。好在您(和您的终端用户)都是使用 Microsoft Windows 操作系统，这样就已经准备好了使用本书的所有工作(您也可以从网上下载)。

所有的代码编写工作都可以使用系统中已安装的 Windows Notepad 应用程序来完成。我们将建议一些其他可用的工具，它们可能会使代码编写工作更加轻松，但实际上使用一个文本编辑器就足够了。

Microsoft Scripting Web 站点包含了与 VBScript 相关的文档，可供下载。您可能也需要下载这些文档以辅助本书的阅读。

如果不是使用 Windows Vista 或 XP，就可能需要下载最新的 VBScript 引擎——请访问 [www.microsoft.com/scripting](http://www.microsoft.com/scripting)。

## 源代码

学习本书的示例时，可以选择手工输入所有代码，也可以使用本书附带的源码文件。本书中使用的所有源代码都可以从 [www.wrox.com](http://www.wrox.com) 下载得到。进入该站点后，可以使用 Search 框找到本书的题目，然后可以选择想下载的内容。

因为很多书籍都有相似的题目，所以按照 ISBN 来搜索可能更容易一些；本书的 ISBN 是 978-0-470-16808-0。

## 勘误

我们将尽全力保证本书的正文和代码中没有错误。但人无完人，错误总是会有有的。如果您发现了本书中的错误，例如拼写错误或代码的错误，那么我们将非常感谢您的反馈。通过发送勘误信息，您将节省其他读者的阅读时间；这样，您的反馈将有助于我们提供更高品质的信息。

要访问本书的勘误页面，请访问 [www.wrox.com](http://www.wrox.com)，并使用 Search 框或书名列表找到本书的书名。然后，在书籍的详细信息页面上，点击 Book Errata 链接。在这个页面上，可以查看关于本书所有已经提交并由 Wrox 编辑公布的勘误信息。包括到每本书的勘误页面的链接的完整书籍列表位于 [www.wrox.com/misc-pages/booklist.shtml](http://www.wrox.com/misc-pages/booklist.shtml)。

如果您有任何意见或建议，或者在本书中发现任何问题，都可以发信到 [wkservice@vip.163.com](mailto:wkservice@vip.163.com)，我们将查看该信息，如果确切的话，将在本书的勘误页面上发布一则消息，并在本书的后续版本中修正这一问题。

## p2p.wrox.com

如果要同作者进行直接的讨论,请加入 p2p.wrox.com 的 P2P 论坛。该论坛是一个基于 Web 的系统,您可以发布关于 Wrox 书籍的消息以及相关的技术信息,并与其他读者和技术用户进行交互。这个论坛提供订阅功能,可以以 E-mail 的方式将您所选择的感兴趣主题的论坛新帖发送给您。Wrox 作者、编辑、其他业界专家以及和您一样的读者都活跃在这个论坛上。

在 p2p.wrox.com,您可以找到很多不同的论坛,不仅可以帮助您阅读本书,还可以帮助您开发应用程序。要加入论坛,可以按照下面的步骤操作:

1. 进入 p2p.wrox.com,单击 Register 链接。
2. 阅读用户协议,单击 Agree 按钮。
3. 填写加入论坛必需的信息以及您想提供的可选信息,单击 Submit 按钮。
4. 您将收到一封电子邮件,其中的信息将描述如何验证您的账户并完成加入过程。

不用加入 P2P 也可以阅读帖子,但要发帖,就必须加入。

加入论坛之后,就可以发新帖并回复其他用户的帖子。任何时候您都可以在 Web 上阅读帖子。如果想将某一论坛中的新帖以电子邮件的方式发送给您,可以单击论坛列表中论坛名称旁边的 Subscribe to this Forum 图标。

关于如何使用 Wrox P2P 的更多信息,请阅读 P2P FAQ,了解论坛软件的工作机制,以及关于 P2P 和 Wrox 书籍的很多一般问题。要阅读 FAQ,请单击 P2P 上任何一个页面中的 FAQ 链接。



# 目 录

第 1 章 编程快速入门	1	第 2 章 VBScript 是什么? 不是什么? ...	31
1.1 变量及数据类型	2	2.1 Windows 脚本	31
1.1.1 变量的使用	2	2.2 版本信息	31
1.1.2 注释的使用	4	2.3 VBScript 是 VB 的子集	32
1.1.3 VBScript 内建函数的使用	4	2.4 VBScript 是一种脚本语言	32
1.1.4 理解语法问题	6	2.5 VBScript 在运行时被解释	33
1.2 流程控制	9	2.5.1 运行时编译的缺点	34
1.2.1 分支	9	2.5.2 运行时编译的优点	35
1.2.2 循环	13	2.6 使用 VBScript 的优势	36
1.3 操作符与操作符优先级	18	2.7 VBScript 适合您吗	36
1.4 代码的组织结构和重用	19	2.8 VBScript 在 Visual Basic	
1.4.1 模块化、黑盒、过程		家族中的地位	37
和子过程	19	2.8.1 Visual Basic	37
1.4.2 编写函数	20	2.8.2 Visual Basic for Applications	38
1.4.3 使用过程的好处	22	2.8.3 VBScript	38
1.5 自顶向下与事件驱动	23	2.8.4 VBScript 是一门“真正”	
1.5.1 理解自顶向下程序设计	23	的编程语言吗	39
1.5.2 理解事件驱动程序设计	23	2.9 VBScript 能做什么	39
1.5.3 自顶向下和事件驱动的		2.9.1 PowerShell	39
结合	24	2.9.2 Windows 脚本宿主	39
1.5.4 事件驱动实例	24	2.9.3 桌面小工具	40
1.6 关于编程的一些忠告	25	2.9.4 Windows 脚本组件	40
1.6.1 要考虑到意外情况	25	2.9.5 客户端 Web 脚本	40
1.6.2 保持代码的可读性	26	2.9.6 服务器端 Web 脚本	41
1.6.3 将代码模块化成模块、		2.9.7 远程脚本	42
类和组件	27	2.9.8 HTML 应用程序	42
1.6.4 匈牙利变量命名法	27	2.9.9 在程序中添加 VBScript	42
1.6.5 不要在多个任务中使用		2.10 VBScript 的开发工具	42
同一个变量	27	2.11 小结	43
1.6.6 时刻都要合理地布局代码	28	第 3 章 数据类型	45
1.6.7 用注释使您的代码清晰		3.1 弱类型脚本语言	45
可读, 但不要滥用	28	3.2 为什么数据类型是如此重要	46
1.7 小结	29		

3.3	Variant——VBScript 中唯一的数据类型	48	4.7.3	使用具名常量的好处	102
3.3.1	子类型的检查和类型转换	49	4.7.4	具名常量的使用原则	103
3.3.2	隐式类型转换	57	4.7.5	VBScript 的内建常量	103
3.3.3	隐式类型转换实战	58	4.8	小结	104
3.3.4	Empty 与 Null	63	<b>第 5 章</b>	<b>流程控制</b>	<b>105</b>
3.3.5	对象(Object)子类型	68	5.1	分支结构	105
3.3.6	Error 子类型	69	5.1.1	If 分支	105
3.4	复杂数据类型——数组	71	5.1.2	Select Case 分支	107
3.4.1	什么是数组	71	5.2	循环结构	110
3.4.2	数组的维度	71	5.2.1	For...Next	110
3.4.3	数组边界和声明数组	72	5.2.2	For Each...Next	114
3.4.4	用下标访问数组	73	5.2.3	Do Loop	116
3.4.5	遍历数组	76	5.2.4	While...Wend	123
3.4.6	清除数组	78	5.3	小结	124
3.4.7	在数组中使用 VarType()	78	<b>第 6 章</b>	<b>错误处理和调试</b>	<b>125</b>
3.5	小结	79	6.1	错误的类型	126
<b>第 4 章</b>	<b>变量与过程</b>	<b>81</b>	6.1.1	语法错误	126
4.1	Option Explicit	81	6.1.2	运行时错误	127
4.2	变量命名	83	6.1.3	逻辑错误	131
4.3	过程和函数	84	6.2	错误的可见性和环境	132
4.3.1	过程的语法	85	6.2.1	Windows 脚本宿主错误	132
4.3.2	函数的语法	86	6.2.2	服务器端 ASP 错误	132
4.3.3	调用过程和函数	89	6.2.3	Internet Explorer 中的客户端 VBScript 错误	133
4.3.4	可选参数	91	6.3	错误处理	134
4.3.5	退出过程或函数	91	6.3.1	Err 对象的使用	134
4.4	变量作用域、声明及生存期	92	6.3.2	On Error 语句的使用	135
4.4.1	理解变量作用域	92	6.3.3	呈现和记录错误	140
4.4.2	理解变量声明	93	6.3.4	显示服务器端 ASP 错误	142
4.4.3	变量生存期	94	6.4	生成自定义错误	146
4.5	脚本和过程的设计策略	95	6.4.1	使用 Err.Raise	146
4.5.1	限制代码读取和修改变量	96	6.4.2	什么时候不该用 Err.Raise	147
4.5.2	将代码分割成过程和函数	96	6.4.3	什么时候该生成自定义错误	148
4.5.3	关于脚本设计的忠告	97	6.5	调试	151
4.6	ByRef 和 ByVal	98	6.5.1	什么是调试器	151
4.7	文字常量和具名常量	100	6.5.2	VBScript 调试概论	152
4.7.1	什么是文字常量	100			
4.7.2	什么是具名常量	101			

6.5.3	用微软脚本调试器调试 WSH 脚本	153	8.2	类语句	202
6.5.4	用微软脚本调试器调试 客户端网页脚本	155	8.3	定义属性	203
6.5.5	用微软脚本调试器 调试 ASP	159	8.3.1	私有属性变量	203
6.5.6	在没有调试器的情况下 调试代码	162	8.3.2	Property Let	204
6.5.7	微软脚本调试器的使用	165	8.3.3	Property Get	204
6.6	小结	172	8.3.4	Property Set	206
<b>第 7 章</b>	<b>脚本运行时对象</b>	<b>175</b>	8.3.5	创建只读属性	207
7.1	什么是运行时对象	175	8.3.6	创建只写属性	208
7.2	对象基础	176	8.3.7	没有属性过程的公共属性	209
7.2.1	创建对象	176	8.4	定义方法	210
7.2.2	属性和方法	177	8.5	类事件	212
7.2.3	With 关键字	177	8.5.1	Class_Initialize 事件	212
7.2.4	对象可以有多种引用	178	8.5.2	Class_Terminate 事件	213
7.2.5	对象生存期和销毁对象	180	8.6	类常量	214
7.3	字典对象	182	8.7	构建和使用 VBScript 类	215
7.3.1	概论	182	8.8	小结	222
7.3.2	三种不同的添加方法	186	<b>第 9 章</b>	<b>正则表达式</b>	<b>225</b>
7.3.3	CompareMode 属性	187	9.1	正则表达式简介	225
7.3.4	Item 属性	187	9.1.1	实战正则表达式	225
7.3.5	Exists 方法	188	9.1.2	从简单的开始	228
7.4	FileSystemObject 库	188	9.2	RegExp 对象	230
7.4.1	为什么要用 FileSystemObject?	189	9.2.1	Global 属性	230
7.4.2	集合的使用	190	9.2.2	IgnoreCase 属性	231
7.4.3	理解 FileSystemObject	190	9.2.3	Pattern 属性	231
7.4.4	创建文件夹	191	9.2.4	正则表达式字符	232
7.4.5	复制文件	192	9.2.5	Execute 方法	240
7.4.6	复制文件夹	193	9.2.6	Replace 方法	240
7.4.7	读取文本文件	194	9.2.7	Backreferencing	241
7.4.8	写文本文件	197	9.2.8	Test 方法	242
7.5	小结	199	9.3	Matches 集合	243
<b>第 8 章</b>	<b>VBScript 中的类 (编写 COM 对象)</b>	<b>201</b>	9.3.1	Matches 的属性	244
8.1	对象、类和组件	201	9.3.2	Match 对象	244
			9.4	一些例子	246
			9.4.1	验证电话号码输入	247
			9.4.2	分解 URI	248
			9.4.3	检查 HTML 元素	248
			9.4.4	匹配空白	249
			9.4.5	匹配 HTML 注释标签	249

9.5	小结	250	12.1.3	Task Scheduler XML 模式	301
<b>第 10 章</b>	<b>客户端 Web 脚本</b>	<b>251</b>	<b>12.2</b>	<b>Task Scheduler 2.0</b>	
10.1	专用工具	251		脚本对象	301
10.2	脚本编程的发展历程	252	12.2.1	Action	301
10.3	各种类型的脚本语言	253	12.2.2	ActionCollection	302
10.3.1	JavaScript、JScript 和 ECMAScript	254	12.2.3	BootTrigger	302
10.3.2	VBScript	254	12.2.4	ComHandlerAction	303
10.4	响应浏览器事件	255	12.2.5	DailyTrigger	303
10.4.1	添加事件句柄	255	12.2.6	EmailAction	304
10.4.2	添加一个带参数的 事件句柄	257	12.2.7	EventTrigger	305
10.4.3	取消事件	258	12.2.8	ExecAction	306
10.4.4	事件的顺序	260	12.2.9	IdleSettings	306
10.4.5	表单验证	263	12.2.10	IdleTrigger	307
10.4.6	验证输入框中的数值	264	12.2.11	LogonTrigger	307
10.4.7	验证单选按钮	265	12.2.12	MonthlyDOWTrigger	308
10.4.8	验证选择控件和日期	267	12.2.13	MonthlyTrigger	309
10.5	实战文档对象模型	270	12.2.14	Networksettings	310
10.5.1	窗口对象	271	12.2.15	Principal	310
10.5.2	集合	273	12.2.16	RegisteredTask	310
10.6	小结	276	12.2.17	RegisteredTask Collection	311
<b>第 11 章</b>	<b>Windows 边栏和小工具</b>	<b>277</b>	12.2.18	RegistrationInfo	312
11.1	小工具基础	278	12.2.19	RegistrationTrigger	312
11.1.1	小工具中的文件	280	12.2.20	RepetitionPattern	313
11.1.2	Manifest 文件	280	12.2.21	RunningTask	313
11.1.3	图标	282	12.2.22	RunningTaskCollection	314
11.2	构建小工具	282	12.2.23	SessionStateChange Trigger	314
11.3	自动刷新小工具	293	12.2.24	ShowMessageAction	315
11.4	打包小工具	295	12.2.25	TaskDefinition	315
11.5	小结	295	12.2.26	TaskFolder	316
<b>第 12 章</b>	<b>Task Scheduler 脚本编写</b>	<b>297</b>	12.2.27	TaskFolderCollection	317
12.1	使用 Task Scheduler	297	12.2.28	TaskNamedValuePair	317
12.1.1	使用 MMC 插件	298	12.2.29	TaskNamedValue Collection	318
12.1.2	在 Task Scheduler 中 定义并创建任务	298	12.2.30	TaskService	318
			12.2.31	TaskSettings	319
			12.2.32	TaskVariables	320

12.2.33	TimeTrigger .....	320	14.3.2	Scriptlet 模型扩展 .....	366
12.2.34	Trigger .....	321	14.4	IE5 中不支持 Scriptlet .....	369
12.2.35	TriggerCollection .....	322	14.4.1	行为机制 .....	369
12.2.36	WeeklyTrigger .....	322	14.4.2	行为机制的实现技术 .....	369
12.3	Task Scheduler 范例脚本 .....	323	14.4.3	将行为应用于 HTML 元素 .....	370
12.4	小结 .....	327	14.5	HTML 组件(HTC) .....	371
<b>第 13 章</b>	<b>PowerShell .....</b>	<b>329</b>	14.6	小结 .....	380
13.1	安装需求 .....	329	<b>第 15 章</b>	<b>Windows 脚本宿主 .....</b>	<b>383</b>
13.2	功能特性 .....	329	15.1	相关工具 .....	384
13.3	为什么说 PowerShell 是 一种新的脚本语言 .....	330	15.2	WSH 的概念 .....	384
13.4	开始使用 PowerShell .....	330	15.3	脚本文件的类型 .....	385
13.4.1	使用 PowerShell .....	332	15.4	使用 Windows 脚本宿主 运行脚本 .....	386
13.4.2	进一步深入 PowerShell .....	335	15.4.1	命令行执行 .....	386
13.5	在 PowerShell 中使用脚本 .....	338	15.4.2	在 Windows 环境中 执行 WSH .....	388
13.5.1	修改 PowerShell 的 执行策略 .....	338	15.5	使用 WSH 文件运行脚本 .....	389
13.5.2	命名脚本 .....	339	15.6	Windows 脚本宿主的 内建对象 .....	390
13.6	创建并调用第一个 PowerShell cmdlet 脚本 .....	339	15.6.1	WScript 对象 .....	390
13.7	VBScript 与 PowerShell 之间的联系 .....	341	15.6.2	WshArguments 对象 .....	397
13.7.1	操作符 .....	341	15.6.3	WshShell 对象 .....	399
13.7.2	函数 .....	343	15.6.4	WshNamed 对象 .....	415
13.7.3	语句 .....	351	15.6.5	WshUnnamed 对象 .....	417
13.8	小结 .....	354	15.6.6	WshNetwork 对象 .....	417
<b>第 14 章</b>	<b>增强的客户端脚本 .....</b>	<b>355</b>	15.6.7	WshEnvironment 对象 .....	422
14.1	使用需求与浏览器安全性 .....	355	15.6.8	WshSpecialFolders 对象 .....	424
14.2	Scriptlet——行为机制的 前身 .....	356	15.6.9	WshShortcut 对象 .....	426
14.2.1	什么是 Scriptlet .....	356	15.6.10	WshUrlShortcut 对象 .....	432
14.2.2	使用 public 前缀暴露 scriptlet 成员 .....	358	15.7	小结 .....	433
14.2.3	在 Scriptlet 中封装代码 以便重用 .....	358	<b>第 16 章</b>	<b>Windows 脚本组件 .....</b>	<b>435</b>
14.3	事件管理 .....	364	16.1	什么是 Windows 脚本组件 .....	435
14.3.1	与事件处理程序之间 的关系 .....	364	16.2	需要的工具 .....	436
			16.3	脚本组件运行时 .....	436
			16.4	脚本组件文件和向导 .....	437
			16.5	暴露属性、方法和事件 .....	443
			16.5.1	属性 .....	443

- 16.5.2 方法 ..... 444
- 16.5.3 事件 ..... 446
- 16.6 注册信息 ..... 447
- 16.7 创建脚本组件类型库 ..... 448
- 16.8 如何引用其他组件 ..... 450
- 16.9 ASP 的脚本组件 ..... 451
- 16.10 编译时错误检查 ..... 452
- 16.11 在 Script 组件中使用  
VBScript 类 ..... 453
  - 16.11.1 VBScript 类的  
使用限制 ..... 453
  - 16.11.2 使用内部类 ..... 453
  - 16.11.3 包含外部源文件 ..... 455
- 16.12 小结 ..... 456
- 第 17 章 脚本编码 ..... 457**
  - 17.1 脚本编码的使用限制 ..... 458
  - 17.2 已编码脚本——什么该  
做什么不该做 ..... 458
  - 17.3 使用 Microsoft 脚本编码器  
进行编码 ..... 459
    - 17.3.1 获取方式和安装过程 ..... 459
    - 17.3.2 使用 Microsoft  
脚本编码器 ..... 459
    - 17.3.3 语法 ..... 460
    - 17.3.4 可编码的文件类型 ..... 462
  - 17.4 脚本的解码 ..... 473
  - 17.5 脚本模糊化的其他方法 ..... 473
  - 17.6 小结 ..... 474
- 第 18 章 远程脚本 ..... 475**
  - 18.1 远程脚本的工作原理 ..... 475
    - 18.1.1 安全性 ..... 476
    - 18.1.2 远程脚本需要的文件 ..... 476
  - 18.2 使用 VBScript 实现  
远程脚本 ..... 476
    - 18.2.1 在服务器上安装  
远程脚本 ..... 477
    - 18.2.2 在服务端启用远程脚本 ..... 477
    - 18.2.3 在客户端启用远程脚本 ..... 478
    - 18.2.4 调用一个远程方法 ..... 478
    - 18.2.5 将 ASP 页面转换成  
VBScript 对象 ..... 480
- 18.3 小结 ..... 481
- 第 19 章 HTML 应用程序 ..... 483**
  - 19.1 HTML 应用程序的优点 ..... 483
  - 19.2 如何创建一个基本的 HTA ..... 484
    - 19.2.1 HTML 文件范例 ..... 485
    - 19.2.2 将一个 HTML 文件  
变为一个 HTML  
应用程序 ..... 487
  - 19.3 HTA:APPLICATION 元素 ..... 488
    - 19.3.1 修改应用程序的外观 ..... 488
    - 19.3.2 从命令行中更改参数 ..... 489
    - 19.3.3 访问其他的  
HTA:APPLICATION  
属性 ..... 491
  - 19.4 HTA 及其安全性 ..... 493
    - 19.4.1 关于框架的安全性问题 ..... 493
    - 19.4.2 使用 APPLICATION  
属性 ..... 493
    - 19.4.3 使用嵌套框架 ..... 495
  - 19.5 HTA 部署模型 ..... 496
    - 19.5.1 Web 模型 ..... 496
    - 19.5.2 包模型 ..... 496
    - 19.5.3 混合模型 ..... 497
  - 19.6 HTA 不支持的特性 ..... 497
    - 19.6.1 windows 对象 ..... 497
    - 19.6.2 默认行为 ..... 498
  - 19.7 小结 ..... 498
- 第 20 章 服务器端 Web 脚本 ..... 499**
  - 20.1 理解 HTTP 协议的  
内部机制 ..... 499
    - 20.1.1 HTTP 服务器 ..... 500
    - 20.1.2 协议基础 ..... 500
  - 20.2 ASP 简介 ..... 503
    - 20.2.1 服务器如何识别 ASP ..... 504
    - 20.2.2 ASP 基础 ..... 504

20.2.3	ASP 的标签	504	21.4	脚本控件参考	535
20.3	使用 ASP 对象模型	507	21.4.1	对象模型	535
20.3.1	集合	507	21.4.2	对象与集合	535
20.3.2	Request 对象的集合	509	21.4.3	常量	549
20.3.3	Response 对象的集合	512	21.5	脚本控件错误处理	549
20.3.4	Response 对象的属性	515	21.6	调试	552
20.3.5	理解 Application 和 Session 对象	516	21.7	使用已编码脚本	553
20.3.6	Application 对象	517	21.8	.NET 工程范例	553
20.3.7	Session 对象	518	21.9	Visual Basic 6 工程范例	554
20.3.8	Server 对象	520	21.10	小结	558
20.3.9	ObjectContext 对象	521	附录 A	VBScript 函数和关键字	559
20.4	高效地使用 ASP	522	附录 B	变量命名约定	621
20.4.1	站点设计	522	附录 C	编码规范	623
20.4.2	创建 global.asa 文件	522	附录 D	VBScript 中支持的 Visual Basic 常量	627
20.4.3	创建主页面	523	附录 E	VBScript 错误代码和 Err 对象	633
20.4.4	ASP/VBScript 部分	526	附录 F	脚本运行时库对象参考	647
20.4.5	HTML 部分	528	附录 G	Windows 脚本宿主 对象模型	659
20.5	小结	529	附录 H	正则表达式	667
第 21 章	在 VB 和 .NET 应用程序中 添加 VBScript 代码	531	附录 I	Variant 的子类型	671
21.1	为什么要在应用程序中 添加脚本	532	附录 J	ActiveX 数据对象	675
21.2	宏和脚本的概念	532			
21.2.1	使用 Scriptlet	533			
21.2.2	使用脚本	533			
21.2.3	选择最佳的使用范围	533			
21.3	在 VB 和 .NET 应用程序中 添加脚本控件	534			

# 第 1 章

## 编程快速入门

鉴于 VBScript 这类语言以及作者所接触到的正在转向这类语言的用户特殊性，用一个涵盖 VBScript 所有基本知识的章节作为本书的开始是非常合适的。在本章中，您将简要地学习一些编程的基本知识。如果是从其他编程语言(Visual Basic、Visual Basic .NET、C、C++、Delphi、C#)转向 VBScript 并熟悉各种专业术语，您可能就不需要阅读本章的内容。若确实如此，请放心地跳过本章直接阅读后面的章节。当然，若您没有任何编程经验，本章将会为您提供牢固的基本知识，使您信心百倍地开始 VBScript 的学习。

若您继续阅读本章，您应该是属于以下三种人：

- 网络管理员或系统管理员，想用 VBScript 和 Windows 脚本宿主或 PowerShell 编写登录脚本或是自动完成管理工作。
- 想提升和拓展自己技术能力的网页设计师，以便能从事某些关于 ASP 的工作。
- 对编程很感兴趣(可能是 Visual Basic 或 Visual Basic .NET)，想在深入学习之前先体验一下。

编程涉及到很多方面。多年以来，在书本中和 Internet 上有无数关于编程的书籍和文章。在本章中，我们将用单独一章的篇幅来介绍一些您之前不熟悉的概念。我们的学习进度会非常快，若您想仔细地阅读这些内容，可以自己试试其中的那些例子，这样可以更好地掌握这些概念。

此外，务必要注意的是，还有很多编程的基础知识是我们在这里没有提及的，例如：

- 软件架构
- 系统设计
- 数据库设计
- 文档化代码
- 高级测试、调试和 beta 版本测试
- 软件部署和支持

本章简要地介绍了构成整个程序设计领域的一些基本概念。显然，我们不可能使您在一夜之间成为编程专家，但是我们希望本章能让您知道如何学习本书中的其他内容。



## 1.1 变量及数据类型

本节中，您将快速地学习一些编程最基本的概念，尤其是：

- 变量的使用
- 注释的使用
- VBScript 的内建函数的使用
- 调解语法问题

### 1.1.1 变量的使用

很简单，变量(variable)就是计算机内存中用户的脚本存放信息或数据的地方。变量中存放的数据可以是任何东西。它可能很简单，譬如一个简单的数字，如 4；也可以稍微复杂些，譬如一个浮点数，如 2.3 或一个更复杂的数字，如 981.12932134。或者根本就不是一个数字，也可以是一个单词或数字和字母的组合。实际上，变量中可以存放任何用户想存放的东西。

实质上，变量是一段预留的用于存储数据的计算机内存。内存是临时性的——存储在内存中的数据并不像存储在硬盘中那样能永久保留。由于内存是一个临时存储区域，而变量又是存放在内存中的，所以变量也是临时的。脚本用变量临时存放数据，并且还需要记住它的地址以便以后使用。如果脚本需要永久地存储数据，那就应该将它存放到计算机硬盘上的文件或数据库中。

为了能使计算机随时方便地管理存储在内存中的大量数据，内存被分为了很多块。每一块内存的大小都是一样的，并且有一个独一无二的地址。不要担心什么是内存的地址或是该如何使用它们，因为使用 VBScript 并不需要用户知道这些，但是要明白变量实际上就是预留的若干块内存，这有助于您接下来的学习。同时，不同类型的变量所需的内存数量也不尽相同。

在 VBScript 程序中，变量在使用前先要声明(或指定大小)，变量的生命周期也是以这个声明开始的。

用户并不需要声明其使用的每一个变量。默认情况下，VBScript 允许您使用未声明的变量。但是，强烈地建议您养成声明脚本中所有变量的好习惯。在使用变量之前先声明有助于提高代码的可读性，方便以后的调试工作。总之，声明总是有益的。

在声明变量的过程中您还得给变量取个名字。下面的例子就是 VBScript 中的变量声明：

```
Dim YourName
```

这样做实际上是向计算机传达了一条指令，要它为您保留一些内存空间，并将这块内存命名为 YourName。从现在开始，计算机(或者更准确地说是 VBScript 引擎)会为您保留