

北京科海培训中心

► 网页设计·网站建设培训系列教材

网页编程入门与应用实例

(学用 VBScript, JavaScript
JavaApplet, ActiveX)

朱 凯 周湘文 编著

清华大学出版社

前 言

目前Internet正在以一种前所未有的冲击力影响着我们的生活。面对制作越来越精美的各类商业和个人网站，越来越多的人都希望拥有自己的个人主页，越来越多的商业网站都希望让自己的站点更具有吸引力和有更多的人来访问。

制作一个网站很容易，任何一个会使用Word的人都可以胜任。然而，事实上想做好一个网站非常困难，仅仅学会一些可视化的网页编辑工具是远远不够的，后台编程开发才是网站制作的核心。

本书分为四个部分，每个部分介绍一种编程语言，各个部分相对独立。如果你只是想学习其中的某一种编程语言也毫无问题。同时，在本书中我们也介绍了将这些编程语言相互配合，发挥各自特点，制作出精美网站的方法。

本书主要有以下几个特点：

- 内容全面。本书结合实例由浅入深地介绍了VBScript，JavaScript，JavaApplet和ActiveX控件的使用，熟练掌握以上开发工具将使你能够轻松地进行网站的后台开发。
- 注重实用性。本书各个部分首先介绍编程语言的基本语法以及初步的使用方法，使读者能够尽快上手，然后通过若干实例的操作，使读者不断加深对软件的了解，最终达到融会贯通的目的。
- 循序渐进。本书中每一部分的实例都由浅入深，首先介绍一些基本的实例，使读者可进行一些基本的实践，然后介绍一到两个大的综合实例来深化读者对语言的认识。

在全书的编写过程中，作者已经做出了最大的努力，唯恐疏漏，但是由于水平所限，疏漏之处在所难免，希望广大专家读者提出批评和建议。

目 录

第1部分 VBScript编程与实例

第1章 VBScript快速入门	1
1.1 VBScript简介	1
1.2 VBScript的数据类型	2
1.2.1 Variant	2
1.2.2 Subtypes	2
1.2.3 识别数据类型	3
1.3 VBScript的变量和常数	4
1.3.1 VBScript的变量	4
1.3.2 VBScript的常数	5
1.4 VBScript的运算符	5
1.5 VBScript的程序流程控制	6
1.5.1 条件语句	6
1.5.2 循环语句	8
1.6 VBScript的过程	12
1.6.1 子程序	12
1.6.2 函数	13
1.7 VBScript的编码约定	13
1.7.1 什么是编码约定	14
1.7.2 常数命名约定	14
1.7.3 变量命名约定	14
1.7.4 变量的作用域	14
1.7.5 描述性变量名和过程名	15
1.7.6 代码注释约定	16
1.7.7 格式化的代码	16
第2章 VBScript初级制作实例	18
2.1 网上书店	18
2.1.1 主要内容	18
2.1.2 实现过程	18
2.1.3 运行效果	23
2.1.4 实例注释和详解	24

2.2 注册页面的实现	25
2.2.1 主要内容	25
2.2.2 实现过程	25
2.2.3 运行效果	29
2.2.4 实例注释和详解	31
2.3 多框架页面的制作	32
2.3.1 主要内容	32
2.3.2 实现过程	33
2.3.3 运行效果	38
2.3.4 实例注释和详解	39
2.4 制作主页计数器	40
2.4.1 主要内容	40
2.4.2 实现过程	40
2.4.3 运行效果	43
2.4.4 实例注释和详解	44
第3章 VBScript高级制作实例	47
3.1 网上图书馆	47
3.1.1 主要内容	47
3.1.2 实现过程	47
3.1.3 运行效果	56
3.1.4 实例注释和详解	59
3.2 网上交谈	61
3.2.1 主要内容	61
3.2.2 实现过程	62
3.2.3 运行效果	66
3.2.4 实例注释和详解	67
3.3 主页留言板	68
3.3.1 主要内容	68
3.3.2 实现过程	68
3.3.3 运行效果	81
3.3.4 实例注释和详解	83
第4章 综合实例：电子布告栏系统	86
4.1 主要内容	86
4.2 实现过程	86
4.2.1 创建数据库bbs.mdb	86
4.2.2 页面制作与程序的编写	87
4.2.3 运行效果	119
4.2.4 实例注释和详解	122

第2部分 JavaScript编程与实例

第5章 JavaScript快速入门	124
5.1 JavaScript 语言特点	124
5.1.1 JavaScript的历史	124
5.1.2 JavaScript的语言特点	124
5.1.3 JavaScript与其他语言的比较	126
5.1.4 JavaScript的运行条件	127
5.1.5 JavaScript的局限性	127
5.2 JavaScript与HTML	128
5.2.1 在HTML中嵌入JavaScript	128
5.2.2 注意事项	130
5.3 JavaScript基本语法	130
5.3.1 数据类型、变量、表达式及运算符	130
5.3.2 JavaScript程序构成	134
5.3.3 事件处理	138
5.3.4 JavaScript对象	141
第6章 JavaScript实例	147
6.1 简单实例	147
6.1.1 创建你的第一个脚本	147
6.1.2 显示对话框	148
6.1.3 装载退出文件显示对话框	149
6.1.4 设定主页用户密码	151
6.1.5 输出用户姓名	152
6.1.6 指数对数转化器	154
6.1.7 通过不同的方法获得彩色文字	156
6.1.8 在主页上显示化学式	158
6.1.9 小时钟	159
6.1.10 情人节快乐	161
6.1.11 用户, 欢迎光临我的主页	163
6.1.12 自定义back和forward按钮	164
6.1.13 显示窗体中元素	165
6.1.14 本节小结	167
6.2 一些小的程序	167
6.2.1 内容随时间而改变	167
6.2.2 制作跑马灯效果	169
6.2.3 随机产生乱数	171
6.2.4 自动为主页加上最后修改时间	172

6.2.5 显示网页停留时间	174
6.2.6 获得美国历届总统名	176
6.2.7 利用JavaScript做动画	178
6.2.8 滚动文本实例	180
6.2.9 随机产生背景颜色	182
第7章 综合应用实例	185
7.1 银行贷款计算器	185
7.2 扑克游戏21点	191
7.3 猜单词	197
7.4 小结	222
 第3部分 JavaApplet编程与实例	
第8章 Java语言快速入门	223
8.1 Java 概述	223
8.1.1 JavaApplet和Java Application	223
8.1.2 Java虚拟机概述	225
8.1.3 JavaApplet的工作原理	225
8.1.4 JavaApplet的生命周期	226
8.2 JavaApplet与HTML	228
8.3 JavaApplet与网络浏览器	229
8.4 Java语言基础	230
8.4.1 JavaApplet的结构特点及主要语言规范说明	230
8.4.2 数据类型	233
8.4.3 变量类型	234
8.4.4 数组	235
8.4.5 程序流程控制	235
8.4.6 Java的类与函数	238
8.5 一个简单的JavaApplet程序设计实例	243
第9章 JavaApplet开发实例	245
9.1 新项目的创建执行过程	245
9.1.1 创建新项目	245
9.1.2 编辑项目	245
9.1.3 编译、连接和运行	248
9.2 嵌在HTML文件内部的Applet	249
9.2.1 创建项目	249
9.2.2 编辑项目	250

9.2.3 编译、连接和运行	255
9.3 在JavaApplet中调用paint()方法	257
9.4 用JavaApplet编写日历	260
9.5 用Java制作漂亮的菜单	265
9.5.1 设计思想	265
9.5.2 在Visual J++ 6.0中的实现步骤	268
9.5.3 源代码程序示例	270
第10章 JavaApplet多媒体编程技术	280
10.1 图形处理	280
10.1.1 绘制直线	280
10.1.2 绘制矩形	281
10.1.3 绘制圆和椭圆	283
10.1.4 绘制圆弧	284
10.1.5 绘制多边形	285
10.2 字型与颜色	286
10.2.1 字型的设置	286
10.2.2 显示方式	286
10.2.3 获取字型的数据	287
10.2.4 调整颜色	289
10.3 播放声音文件	291
10.4 图像处理	292
10.4.1 图像的装载	293
10.4.2 图像的显示	294
10.5 用Java制作图形时钟	296

第4部分 ActiveX控件编程与应用实例

第11章 ActiveX控件基础	310
11.1 OLE技术基本概念	310
11.1.1 OLE 1.0	310
11.1.2 OLE 2.0	311
11.1.3 COM	312
11.2 创建ActiveX控件	312
11.2.1 创建ActiveX容器应用程序	312
11.2.2 创建ActiveX服务器应用程序	316
11.2.3 创建自动化服务器程序并编写相应代码	318
11.2.4 定义服务器程序的属性和方法	325
11.2.5 创建一个客户程序用来测试服务器程序	326

第12章 编写ActiveX控件实例——拼字游戏	331
12.1 创建ActiveX控件框架.....	331
12.2 编写控件用户接口	332
12.3 创建属性和方法	337
12.4 按钮响应事件	338
 第13章 ActiveX控件在Web中的应用	340
13.1 从Web下载对象.....	340
13.2 <Object>标记的使用	341
13.2.1 <OBJECT>符号	342
13.2.2 生成<OBJECT>HTML代码	342
13.3 在HTML代码中使用ActiveX控件.....	343
13.3.1 添加标签对象	344
13.3.2 添加按钮对象	345
13.3.3 添加VBScript程序	346
13.4 利用ActiveX控制板制作Web页	348
13.4.1 下载和安装ActiveX控制板	348
13.4.2 在Web页面中插入多媒体控件	350
13.5 使用HTML布局控件	362
13.5.1 HTML布局控件综述	363
13.5.2 HTML布局控件工具箱	364
13.5.3 为HTML布局控件工具箱添加新控件	365
13.5.4 为布局添加控件	367
13.5.5 访问控件属性	367
13.5.6 对齐布局内的控件	368
13.5.7 管理重叠控件	370
13.5.8 用HTML布局控件创建表单	371
13.5.9 为Web页添加HTML布局表单.....	375
13.5.10 为HTML布局添加VBScript.....	376

第1部分 VBScript编程与实例

第1章 VBScript 快速入门

本章讲解的内容是最常用的动态网页编程技术——VBScript脚本语言。通过本章你能迅速掌握VBScript的一些基本概念,如VBScript的变量、常数、流程控制和过程等一些编程的基本知识。

1.1 VBScript简介

我们已经知道ActiveX技术正得到越来越广泛的使用,这是因为通过Internet可以方便的分发ActiveX控件和文档,为了在Internet上更有效的使用ActiveX控件,必须将这些控件封装到一种脚本语言中去。近些年来有两种脚本语言得到广泛的宣传和使用,这就是JavaScript和VBScript。VBScript是Microsoft公司针对JavaScript而提出的,它是从Visual Basic中引申出来经过提炼和简化的一种脚本语言,它保留了许多Visual Basic的命令(在下一章的几个实例中,你将学到使用VBScript编写HTML脚本语言的基本方法,同时也将学习如何使用VBScript将ActiveX控件放置到Web页面上去)。

Microsoft公司设计VBScript的主要目的是用于开发Web页面。而VBScript区别于VBA (Visual Basic for Application)之处在于VBA提供了在某种应用程序中(如Microsoft Excel)开发实用程序的能力。随着浏览器与操作系统的逐渐融合,VBScript和VBA也将逐渐合二为一。

VBScript与Visual Basic最显著的区别在于VBScript缺乏对文件输入/输出操作的支持。但这对于网络程序设计而言并不是什么缺陷,相反却是提供了更有利的安全保证。没有了文件输入/输出的操作,VBScript代码就不可能访问存储在客户机本地硬盘上的私人信息,因此VBScript应用也不可能将计算机病毒传染到用户磁盘上。由于VBScript和Visual Basic使用了相同的语法和语义,在学习过程中你会发现对VBScript仿佛已经很熟悉了。因此,熟悉Visual Basic的程序员更愿意使用VBScript这种脚本语言。

虽然通过VBScript这种用于HTML程序设计的脚本语言,难以让你进行非常复杂的程序设计,但你可以用VBScript完成以下的一些主要功能:

- 可以将VBScript代码嵌入到HTML页面中来操作客户端的处理过程。
- 使用VBScript,可以访问HTML语言中<INPUT>等标记创建的标准对象的属性和方法,并对这些对象的事件做出反应。
- 使用VBScript,可以访问ActiveX控件的属性和方法,并对这些控件产生的事件做出

反应。

- 可以通过脚本语言支持对象继承而来的Internet Explorer的脚本对象模型，从而访问文档的属性和对象。

注意：现阶段Microsoft Internet Explorer是唯一具有内置VBScript支持功能的浏览器。通过Ncompass实验室开发的可插拔控件ScriptActive，在Netscape Navigator中也可以使用VBScript。随着ActiveX技术的发展和普及，支持VBScript和JavaScript将逐渐成为对浏览器的基本要求。想了解更多的关于VBScript的资料，可以访问以下网站：

<http://www.ncompasslabs.com> 下载ScriptActive插拔件
<http://www.vbscript.com> 关于VBScript的最全面的站点
<http://microsoft.com/vbscript> 由Microsoft提供的VBScript教程和例程。

1.2 VBScript的数据类型

所有编程语言都使用编程变量，所谓变量是计算机内存中已命名的存储单元，其内容可能是数字或字符串等数据，它使用户便于理解脚本操作的名称，为用户提供了一种存储、检索和操作数据的途径。变量由变量名识别，并把数据作为值存在它所占的内存空间中。在VBScript里只有唯一的一种变量数据类型。

1.2.1 Variant

虽然VB支持许多数据类型，但VBScript只支持一种，即：Variant型。这种数据类型是唯一的，因为它代表了所有可在VB中找到的数据类型，如：Integer、String等。Variant数据类型有能根据它存储的信息来改变变量的特点。

1.2.2 Subtypes

Variant数据类型由更小的单位Subtypes构成。Subtypes能识别数据是如何被存于Variant中的，它能够识别String、Numeric、Byte、Boolean、Date、Empty、Null、Error和Object等数据类型，如表1.1所示。这类特殊化的Subtypes为特定数据类型的扩展功能提供了最优函数。而且Variant的Subtype为操纵Subtypes提供了大范围的转换功能，如日期/时间值、字符串表达式和货币值等等，如表1.2所示。

表1.1 Variant包含的数据子类型

子类型	描述
Empty	数字变量为0，字符串变量中的零长度字符串
Null	不包含任何有效数据的
Boolean	真或假（True或false）
Byte	从0到255的整数
Integer	从-32768到32767的整数

续表

子类型	描述
Currency	-922 337 203 685 477.5808到922 337 203 685 477.5807
Long	从-2 147 483 648到2 147 483 647的整数
Single	单精度浮点数, 负数从-3.402823E38到-1.401298E-45, 正数从1.401298E-45到3.402823E38
Double	双精度浮点数, 负数从-1.79769313486232E308到-4.94065645841247E-324, 正数从4.94065645841247E-324到1.79769313486232E308
Date(Time)	日期范围从公元100年1月1日到9999年12月31日
String	可变长度字符串, 最大长度可达20亿个字符
Object	对象
Error	错误数

表1.2 数据类型的转换功能

函数类型	转换类型
Cbyte	Byte
Ccur	Currency
Cdata	Data
Cdbl	Double
Cint	Integer
Clng	Long
Cstr	String
Csng	Single

1.2.3 识别数据类型

由于有着众多的子类型, 你有时也需要判断处理的究竟是什么变量子类型。VarType函数的功能就是识别数据子类型, 此函数返回一个数值, 而每一个数值对应一种子类型。表1.3显示了函数识别后返回的结果。

表1.3 数据类型返回值表

Subtype	返回值	Subtype	返回值
Empty	0	String	8
Null	1	Automation Object	9
Integer	2	Error	10
Long	3	Boolean	11
Single	4	Variant	12
Double	5	Non-Automation Object	13
Currency	6	Byte	17
Date(Time)	7	Array	8192

1.3 VBScript的变量和常数

我们已经简要介绍了Variant数据类型的特点及转换函数。这一节我们将具体讨论变量和常数的创建过程。另外变量有效期有多长是非常关键的，所以还将介绍变量的生存期及使用VBScript变量的限制。

1.3.1 VBScript的变量

1. 什么是变量

变量是一种使用方便的占位符，用于引用计算机的内存地址，该地址可以存储Script运行时的程序信息（可更改的部分），使用变量时并不需要了解它在内存中的地址，只要通过变量名引用就可以查看或更改它的值。在VBScript中所有变量的数据类型都是Variant。

2. 变量的声明

在VBScript中，严格的来讲变量是不用声明的，但应养成在编程时声明变量的良好习惯，因为这样有助于防止错误发生。声明一个变量意味着告诉脚本引擎，存在一个特定名称的变量，这样就可以在脚本中引用该变量。在VBScript中声明变量的一种方式是使用Dim、Public或Private关键字在Script中显式地声明变量。如：

```
Dim myDate  
Public A(10)
```

声明多个变量时，应使用逗号分隔变量，如：

```
Dim Top,Bottom,Left,Right
```

声明变量的另一种方式是直接在Script中使用变量名这一简单方式隐式地声明变量，但这样易导致由于变量名被拼错而在运行Script时出现意外的结果。因此最好使用Option Explicit语句显式地声明所有变量，并将其作为Script的第一条语句。

3. 变量命名规则

变量命名必须遵循VBScript的标准命名规则，变量命名必须要：

- 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。

4. 变量的生命期

变量的生命期即作用域，决定了那些脚本命令可访问变量。在过程内部声明的变量只在该过程内部有效，每执行一次过程，变量就被创建然后消亡，而过程外部的任何命令都不能访问它。在过程外部声明的变量在全局范围内都有效，其值能被页上的任何脚本命令访问和修改。声明变量时，局部变量和全局变量可以有相同的名称；而改变其中一个变量的值并不会改变另一个同名变量的值。如果没有声明变量，则可能会不小心改变一个全局变量的值。所以按其作用域来划分变量有两种：

- Script-level: 变量声明代码在所有的Function或Sub模块之外, 全局变量。
- Procedure-level: 变量声明代码在一个Function或Sub模块之内, 局部变量。

5. 给变量赋值

VBScript使用如下形式给变量赋值: 变量在表达式左边, 要赋的值在表达式右边, 中间用等号进行赋值。如:

```
myDate=20  
A(3)=345
```

1.3.2 VBScript的常数

1. 什么是常数

常数是用来代替一个数字或字符串的字符串, 具有一定的含义, 它在整个脚本中保持不变。VBScript定义了许多固有常数(由应用程序提供的常数)。

2. 常数的创建

您可以使用Const语句在VBScript中创建用户自定义的常数。使用Const语句可以创建其名称具有一定含义的字符串型或数值型常数, 并给它们赋值。例如:

```
Const MyString="这是一个常数"  
Const MyAge=100  
Const MyNum="85"
```

请注意字符串文字包含在两个引号"之间, 这是区分字符串型常数和数值型常数的最明显的方法。日期字符串和时间字符串包含在两个井号##之间, 以定义日期型和时间型常数。例如:

```
Const CutoffDate=#6-1-97#
```

最好采用一个统一的命名方案以区分常数和变量, 这样可以避免在运行Script语句时对常数重新赋值, 例如: 可以使用“vb”或“con”作为常数名的前缀, 或将常数名的所有字母大写, 这样做可以在开发复杂的Script时避免混乱。

1.4 VBScript的运算符

VBScript提供了各种各样的编程运算符, 包括算术运算符、比较运算符、连接运算符和逻辑运算符。当表达式包含多个运算符时, 将按预定顺序计算每一部分, 这个顺序被称为运算符优先级, 可以使用括号越过这种优先级顺序, 强制首先计算表达式的某些部分。运算时, 总是先执行括号中的运算符, 然后再执行括号外的运算符。但是, 括号内的运算仍遵循标准运算符优先级顺序。

当表达式包含多种运算符时, 首先计算算术运算符, 然后计算比较运算符, 最后计算逻辑运算符。所有比较运算符的优先级相同, 即按照从左到右的顺序计算比较运算符, 而算术运算符和逻辑运算符的优先级如表1.4所示(表中运算符从上到下, 优先级从大到小)。

当乘号和除号或当加号与减号同时出现在一个表达式中时，按从左到右的顺序计算运算符。字符串连接（&）运算符不是算术运算符，但是在优先级顺序中，它排在所有算术运算符之后及所有比较运算符之前。对象引用比较（Is）运算符并不比较对象和对象的值，而只是进行检查，判断两个对象引用的是否为同一个对象。

表1.4 算术运算符和逻辑运算符优先级

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
求幂	^	等于	=	非	Not
负号	-	不等于	<>	与	And
乘	*	小于	<	或	Or
除	/	大于	>	异或	Xor
整除	\	小于等于	<=	等价	Eqv
求余	Mod	大于等于	>=	隐含	Imp
加	+	对象引用比较	Is		
减	-				
字符串连接	&				

1.5 VBScript的程序流程控制

本部分将讨论如何控制VBScript中的程序执行流程，并研究如何在VBScript中创建基本逻辑判断以驱动应用程序中的各种控制结果。了解编程语言的朋友一定知道在程序中控制程序流程的语句主要可以分为条件语句和循环语句。条件语句是根据给定条件的测试结果来执行相应的程序代码的判定分支，循环结构是循环处理一段代码直到条件满足的循环分支。

1.5.1 条件语句

使用条件语句可以控制Script的流程，还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
- Select Case语句

1. If...Then...Else 语句

VBScript中最常用来作判断的控制语句就是If...Then...Else语句。这条语句用于计算条件是否为True或False，并且根据计算结果指定要运行的语句。通常，条件是使用比较运算符对值或变量进行比较的表达式，If...Then...Else语句可以按照需要进行嵌套。

(1) If...Then语句

此语句省略了关键字Else，即如果条件为真（True），则Then后紧跟的一段源程序将被执行。例如：

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

如果If...Then语句后要运行多行代码，则必须使用多行语句。该多行语句的最后需包含End If语句，以表示条件满足后，所执行的语句结束的位置。例如：

```
Sub Alert(value)  
    If value=0 Then  
        AlertLable.ForeColor=vbRed  
        AlertLable.Font.Bold=True  
        AlertLable.Font.Italic=True  
    End If  
End Sub
```

（2）If...Then...Else语句

在该语句中，Else条件分支使得在If条件判断下多了一个失败的可能性。它可以使内部语句源程序代码分开，分别执行Then和Else后的两个可执行语句块。条件为True时运行Then后的语句块，条件为False时运行Else后的语句块。例如：

```
Sub Alert(value)  
    If value=0 Then  
        AlertLable.ForeColor=vbRed  
        AlertLable.Font.Bold=True  
        AlertLable.Font.Italic=True  
    Else  
        AlertLable.ForeColor=vbBlack  
        AlertLable.Font.Bold=False  
        AlertLable.Font.Italic=False  
    End If  
End Sub
```

（3）If语句的嵌套

If...Then...Else语句的一种变形允许您从多个条件中选择，即添加Elseif子句以扩充If...Then...Else语句的功能，使您可以控制基于多种可能的程序流程。例如：

```
Sub Report(value)  
    If value=0 Then  
        MsgBox value  
    ElseIf value=1 Then  
        MsgBox value
```

```
ElseIf value=2 Then
    MsgBox value
Else
    MsgBox"数值错误！"
End If
End Sub
```

可以添加任意多个ElseIf子句以提供多种选择。但使用多个ElseIf子句经常会使程序变得很累赘。在多个条件中进行选择的更好方法是使用Select Case语句。

2. Select Case 语句

Select Case结构提供了If...Then...ElseIf结构的一个变通形式，可以从多个语句块中选择执行其中的一个。Select Case语句提供的功能与If...Then...Else语句类似，但是可以使代码更加简炼易读。Select Case结构在其开始处使用一个只计算一次的简单测试表达式，表达式的结果将与结构中每个Case的值比较，如果匹配，则执行与该Case关联的语句块。我们同样可以用Select Case语句来改写上例：

```
Sub Report(value)
    Select Case value
        Case 0
            MsgBox value
        Case 1
            MsgBox value
        Case 2
            MsgBox value
        Case Else
            MsgBox"数值错误！"
    End Select
End Sub
```

注意：Select Case结构只计算开始处的一个表达式，并且只计算一次。而If...Then...ElseIf结构计算每个ElseIf语句的表达式，并且这些表达式可以各不相同。因此仅当每个ElseIf语句计算的表达式都相同时，才可以使用Select Case结构代替If...Then...ElseIf结构。Select Case语句也可以是嵌套的，每一层嵌套的Select Case语句必须有与之相匹配的End Select语句。

1.5.2 循环语句

VBScript中循环语句的作用就是重复执行某段程序代码，并按设置的次数进行或直到条件满足为止。循环可分为三类：一类在条件变为“假”之前重复执行语句，一类在条件变为“真”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：

- For...Next：指定循环次数，使用计数器重复运行语句。

- Do...Loop: 当（或直到）条件为“真”时循环。
- While...Wend: 当条件为“真”时循环。
- 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。

1. For...Next 语句

For...Next语句用于为语句块指定运行的次数，在循环中使用计数器变量，该变量的值随每一次循环增加或减少一个指定的步长（step）。如果知道循环次数或者知道要求重复的次数，这将对你很有帮助。

下例设置了Count的For...Next循环,从65到122，每隔三个打印出ASCII字符：

```
Sub Count()  
    Dim myCounter  
    For myCounter=65 to 122 step 3  
        Response.Writechr(myCounter)  
    Next  
End Sub
```

若省略step设置，则step的值默认为1。要使计数器变量递减，可将Step设为负值。此时计数器变量的终止值必须小于起始值。例如，可改写上面的例子为：

```
Sub Count()  
    Dim myCounter  
    For myCounter=122 to 65 step -3  
        Response.Writechr(myCounter)  
    Next  
End Sub
```

注意：VBScript中Next的运算符与原始VB中的Next运算符不同，原始VB要求在Next语句中有一个确定的counter变量，然而VBScript并不要求一定存在有counter变量。

Exit For语句用于在计数器达到其终止值之前使流程退出For...Next语句。因为通常只是在某些特殊情况下（例如在发生错误时）要退出循环。所以可以在If...Then...Else语句的True语句块中使用Exit For语句。如果条件为False，循环将照常运行。例如：

```
Sub Count()  
    Dim myCounter  
    For myCounter=1 to 10  
        If (myCounter=4) Then  
            Exit For  
        Else  
            Response.Writechr(100+myCounter)  
        End If  
    Next  
End Sub
```