

第 4 章 VBScript 语言



本章内容是 VBScript 脚本语言。主要介绍在网页中使用 VBScript 进行脚本编程的基本知识。其中包括 VBScript 的基本数据类型、常量与变量、运算符与表达式、语句结构、常见内置函数的应用、VBScript 过程、VBScript 对象编程等。

本章引例中各页面都是在 HTML 文档中嵌入了 VBScript 脚本代码，代码是使用 VBScript 的各种基本语法设计出来的。引例涵盖了本章全部内容以及关键知识点。



- (1) 了解 VBScript 的基本概念。
- (2) 熟练掌握 VBScript 语法，包括基本数据类型、常量与变量、运算符与表达式的应用和语句控制方法。
- (3) 熟练掌握 VBScript 的几种常用函数的使用。
- (4) 掌握 VBScript 中部分对象的属性、方法和事件的使用。
- (5) 学会使用 VBScript 编程。

通过前面的学习，我们已掌握了页面设计的基本方法。在实际应用时，经常在网页中加入一些脚本，来控制网页中的元素与对象，增强网页的灵活性、多样性和交互性。

引例

本案例以一个简单的“趣味智力测试”系统为引线，将 VBScript 脚本编程的知识要点有机串连起来，构成一个系统的综合案例。在学习过程中，要注意理解和掌握 VBScript 的语法结构、函数和对象的使用，学会在网页中使用 VBScript 进行脚本编程。要求在学习完本章知识后，能够有效地使用 VBScript 脚本语言实现本案例。

本案例由首页 login.html 和测试主页 test.html 组成，通过对案例功能的逐步剖析，将这两个页面分解成多个简单的页面，逐一讲解 VBScript 有关的语法、函数、对象和事件，最终能够完整实现本案例。

如图 4-1 及图 4-2 所示为系统案例页面的效果图。



图 4-1 系统案例首页



图 4-2 系统案例测试主页

4.1 VBScript 概述

4.1.1 VBScript 基本概念

VBScript 即 Microsoft Visual Basic Scripting Edition, 是一种基于对象的脚本语言。所谓脚本, 是指嵌入到 Web 页中的程序代码, 利用这些代码, 可以控制网页的控件和对象, 增强网页的灵活性和多样性。因此, 在 HTML 文件中可直接嵌入 VBScript 脚本, 从而扩展 HTML, 使其不仅仅是一种页面格式语言, 而且还可以对用户的操作做出反应。

VBScript 是目前最流行的脚本语言之一, 以其简单的语法、完善的功能, 成为既可应用在客户端进行编程, 也可应用在服务器端进行编程的脚本语言。在 ASP 程序设计中, Web 服务器 IIS 默认使用 VBScript 作为脚本语言, 服务器端编程通常采用 VBScript。

在编写 VBScript 脚本时不区分大小写, 可使用任意文本编辑器, 将其嵌入到 HTML 标记中, 并保存为 .htm、.html 或 .asp 等文件格式即可。

4.1.2 在 HTML 网页中使用 VBScript

HTML 网页是在客户端的浏览器上执行的, 在 HTML 网页中使用 VBScript, 其脚本代码必须放入 `<script>...</script>` 标记中, 其格式为:

```
<script language="vbscript">
.....
' vbscript 代码
</script>
```

【例 4-1】 在 HTML 网页中输出文本。

(1) 打开“记事本”, 输入以下代码:

```
<html>
<head>
<script language="vbscript">
    document.write("用户【林小枫】, 你好, 欢迎使用!")
</script>
```



```
</head>
<body></body>
</html>
```

本程序的作用是在 HTML 网页中输出指定的文本内容，其中 document.write()作用是输出参数值到浏览器窗口中。

- (2) 单击“保存”按钮，保存页面名为 welcome_1.html。
- (3) 打开该文件，启动浏览器预览，效果如图 4-3 所示。



图 4-3 在 HTML 网页中使用 VBScript

4.2 VBScript 的数据类型

VBScript 中只有一种数据类型，称为 Variant（变体型）。根据使用方式的不同，Variant 包含不同类型的信息，可以根据用途的需要选择最合适的子类型来存储数据。常见的子类型如表 4-1 所示。

表 4-1 Variant 的子类型

类型	描述
Empty	未初始化 Variant 变量，对数值变量值为 0，对字符串变量值为""
Null	不包含有效数据的变量
Boolean	值只能是 true 或 false
Byte	值为 0~255 之间的整数
Integer	值为-32768~32767 之间的整数
Currency	货币数据类型，值为-922337203685477.5807~922337203685477.5807 之间的数字
Long	值为-2147483648~2147483647 之间的整数
Single	单精度浮点数，负数范围-3.402823E38~-1.401298E-45，正数范围 1.401298E-45~3.402823E38
Double	双精度浮点数，负数范围-1.79769313486232E308~-4.94065645841247E-324，正数范围 4.94065645841247E-324~1.79769313486232E308
Date	日期时间型，日期范围为公元 100 年 1 月 1 日~公元 9999 年 12 月 31 日
String	字符串型，最大字符串长度为 20 亿个字符
Object	对象型
Error	错误编号



4.3 常量、变量与表达式

常量和变量是编程语言最基本的组成单位，通常由常量和变量组成表达式，然后由表达式组成更为复杂的程序。

4.3.1 常量的定义

常量的值，在程序处理过程中是不变的。有时一个常量在程序中需要使用多次，通常可将其定义成一个符号常量，即用一個符号代表一个具体的常量值，其命名规则与变量的命名规则一样。在 VBScript 中定义符号常量使用 Const 语句，语法结构为：

Const 常量名=值

该语句的作用是将值赋给常量名，例如：

Const Day="星期五" '字符串常量，必须用双引号" "括起来

Const Date=#27/11/2009# '日期常量，必须用#括起来

Const PI=3.1415926 '数值常量

Const T=true '逻辑型常量，取值为 true 或 false

4.3.2 变量的声明与赋值

变量是指存储数据的地方。在程序设计中，常利用变量来保存要运算的数、中间结果或最终结果等。在 VBScript 程序中，变量的类型也都是 Variant 类型。

1. 变量的命名规则

变量命名必须遵循 VBScript 的标准命名规则：

- ① 第一个字符必须是字母；
- ② 不能含有除下划线以外的特殊符号；
- ③ 长度不能超过 255 个字符；
- ④ 不能和 VBScript 中的关键词同名；
- ⑤ 在被声明的作用域内必须唯一。

2. 变量的声明

VBScript 对变量的声明可以采用两种处理方式：默认情况下，变量在使用前不要求事先声明，可以直接使用；另一种方式是变量在使用前，必须事先声明，为此，在脚本的开始要加入 Option Explicit 语句。在进行 VBScript 程序设计时，尽量采用后一种方式，这样可减少出错的几率。

在 VBScript 中声明变量要用到 Dim，语法结构为：

Dim 变量名

如：Dim x

x=5

一个 Dim 语句可以声明一个或多个变量，在声明多个变量时，用“,”分隔变量名。

如：Dim x,y '声明 x 和 y 两个变量。

3. 变量的赋值

当变量声明后，就可以将一个具体的值赋给变量，语法结构为：



变量名=值

如：x=100 '将数值 100 赋给变量 x

【例 4-2】变量赋值。

①新建一个 Dreamweaver 文档，选择“代码”视图，在<head>…</head>标记中输入如下脚本代码：

```
<script language="vbscript">  
    Dim name  
    name=prompt("请输入你的姓名")  
    document.write("用户【"&name&"】，你好，欢迎使用！")  
</script>
```

程序中使用 Dim 声明变量 name，使用 Window 对象中的 prompt()方法产生一个输入框，如图 4-4 所示，当用户在输入框中输入姓名后，将被赋予 name 变量。document 对象的 write 方法的作用是将变量 name 的值在页面中输出显示。

②文件保存为 welcome_2.html；

③启动浏览器预览，程序运行结果与图 4-3 相同。

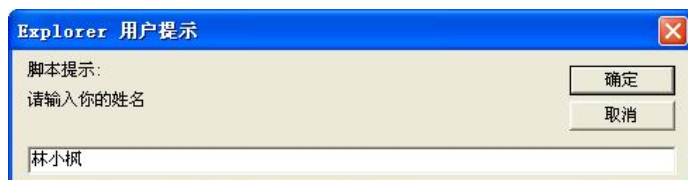


图 4-4 在输入框中输入变量的值

4. 变量的作用域和生存期

变量的作用域是指变量所起作用的范围，变量声明的位置不同，其作用域也不同。如果一个变量在过程或函数内声明，则称其为局部变量或过程级变量，其作用域仅局限于该过程或函数内部，用于临时存放数据。如果一个变量在过程或函数外声明，则称其为全局变量或页面级变量，其作用域为整个网页，用于存放整个程序公用的数据，可供当前网页中的各个脚本程序访问。

变量的生存期指变量的存活期。对于全局变量，其生命周期是从它被声明起一直到整个脚本运行完毕才结束；而对于局部变量，其生命周期是从过程或函数被执行的时刻开始到过程或函数执行完毕就结束。所以，局部变量是用于存取临时数据的。

【例 4-3】变量的作用域。

在<head>…</head>标记中输入如下脚本代码，保存为 score_1.html。

```
<script language="vbscript">  
    dim score  
    score=20  
    sub test_1()  
        num=30  
        score=score+num  
    end sub  
    call test_1  
    document.write(score)
```



```
</script>
```

score 为全局变量，num 为局部变量。程序运行结果如图 4-5 所示。



图 4-5 程序运行结果

4.3.3 数组的定义

数组是一组具有相同名称不同下标的变量，数组中的所有数据称为该数组的元素。数组的定义与变量一样，都需要使用 Dim 语句，只是将元素的个数，即数组的长度包含在数组名之后的括号里。如定义一个名为 a，且包含 10 个元素的一维数组格式如下：

```
Dim a(9)
```

其中，第一个元素为 a(0)，因此，一个含 10 个元素的数组长度值为 9。

如定义一个名为 b 的 3 行 4 列的二维数组格式如下：

```
Dim b(2,3)
```

在脚本运行期间，长度可以变化的数组，叫动态数组。使用 Dim 或 ReDim 定义，定义时不指定数组的长度。如：

```
Dim a()
```

```
ReDim b()
```

ReDim 语句与 Dim 的不同之处是 ReDim 语句定义的数组可以重设数组的维数和大小。

如：

```
ReDim a(9)           '定义了 10 个元素
```

```
.....
```

ReDim preserve a(10) 将 10 个元素改为 11 个元素，原数组的值不变。若改为小于 10 的数值，则会丢失相应的数组元素

4.3.4 运算符与表达式

VBScript 的运算符包括算术、连接、逻辑和比较 4 种运算符。

1. 算术运算符

算术运算符包括加 (+)、减 (-)、乘 (*)、除 (/)、取余 (Mod)、求幂 (^) 及整除 (\) 等。如：3 Mod 2，结果为 1；3^2 结果为 9；5\3 结果为 1。

2. 连接运算符

连接运算符也称字符串运算符，包括 “&” 和 “+” 两种符号，用于将两个字符串连接起来。因为连接符 “+” 与算术运算符中的加号 “+” 相同，当表达式两边都是数值时，会根据运算符的优先级将 “+” 默认为算术运算符中的加号 “+”，表示相加。因此，在使用时，经常



用“&”连接两个字符串。

3. 逻辑运算符

逻辑运算符通常包括与（And）、或（Or）、非（Not）三种，用于判断表达式是否成立，其返回值为：真（true）或假（false）。

- （1）And：只有两个值全为真时，结果才为真。如(3>2)And(3>4)，表达式的值为 false。
- （2）Or：只要有一个值为真时，结果就为真。如(3>2) Or(3>4)，表达式的值为 true。
- （3）Not：非真则为假，非假则为真。如 Not(3>2)，结果为 false；Not(3>4)，结果为 true。

4. 比较运算符

比较运算符通常包括等于（=）、不等于（<>）、小于（<）、大于（>）、小于等于（<=）、大于等于（>=）6 种，用于将表达式两边的值做出比较，其返回值为真（true）或假（false）。

5. 运算符的优先级

当一个表达式中包含有多种运算符时，就必须遵守一个优先级的规则，即首先处理算术运算符，其次处理连接运算符，然后处理比较运算符，最后处理逻辑运算符。各运算符的优先级如表 4-2 所示。

表 4-2 运算符的优先级

算术运算符	求幂	取负	乘法	浮点除	整除	取模	加法	减法
	^	-	*	/	\	Mod	+	-
连接运算符	具有相同的优先顺序，它们按出现的顺序从左到右进行计算							
比较运算符	具有相同的优先顺序，它们按出现的顺序从左到右进行计算							
逻辑运算符	逻辑非	逻辑与	逻辑或					
	Not	And	Or					

4.4 VBScript 常用函数

函数是一段用来表示完成某种特定的运算功能的程序。VB 支持的函数，VBScript 一般也支持。下面介绍几种使用 VBScript 进行网页编程时的常用函数。

4.4.1 数学函数

1. 取整

格式 1: Int(x)

格式 2: Fix(x)

功能：若 x 为正数，则返回 x 的整数部分。若 x 为负数，则 Int 返回小于或等于 x 的第一个负整数，Fix 返回大于或等于 x 的第一个负整数。

如：

Int (4.8) = 4 Fix (4.8) = 4
Int (-4.8) = -5 Fix (-4.8) = -4

2. 取绝对值

格式: Abs(x)



功能：返回 x 的绝对值。如果 x 的值大于等于零，返回的数值就会等于输入的数值；相反小于等于零，则会去掉负号输出，如：

Abs (1) =1

Abs (-1) =1

Abs (0) =0

3. 四舍五入

格式：Round (表达式, [小数位数])

功能：返回表达式按指定的小数位数进行四舍五入后的结果，若缺省小数位数，则四舍五入为整数。如：

Round (5.2345,2) =5.23

Round (5.5123) =6

4. 开平方

格式：Sqr(x)

功能：返回 x 的平方根。要求 x 大于或等于 0，如：

Sqr (361) =19

Sqr (23) =4.79583152331272

5. 表达式计算

格式：Eval(数学表达式)

功能：计算并返回指定数学表达式的值。如：

Eval(15+28/4)=22

6. 随机数

格式：Rnd([x])

功能：返回一个小于 1 但大于或等于 0 的随机数。若 $x < 0$ ，则每次产生的随机数均相同；若 $x > 0$ 或缺省，则产生与上次不同的新随机数；若 $x = 0$ ，则本次产生的随机数与上次产生的随机数相同。如：

Rnd(-1) = .224007

Rnd(1) = 3.584582E-02

【例 4-4】各数学函数应用示例。

新建一个 Dreamweaver 文档，选择“代码”视图，在<head>...</head>标记中输入如下脚本代码：

```
<script language="VBScript">
    document.write("-5 的绝对值"&Abs (-5))
    document.write("<br>-4.8 的取整"&Int (-4.8))
    document.write("<br>-4.8 的取整"&Fix (-4.8))
    document.write("<br>5.6 四舍五入"&CInt (5.6))
    document.write("<br>5.2345 四舍五入"&Round (5.2345,2))
    document.write("<br>23 的平方根"&Sqr (23))
    document.write("<br>Rnd (-1) "&Rnd (-1))
    document.write("<br>Rnd (-1) "&Rnd (-1))
    document.write("<br>Rnd (1) "&Rnd (1))
```



```
document.write("<br>Rnd(1)" & Rnd(1))
document.write("<br>Rnd(0)" & Rnd(0))
</script>
```

文件保存为 `math.html`。运行结果如图 4-6 所示。



图 4-6 各数学函数应用示例

4.4.2 字符串处理函数

1. 字符串长度检测

格式: `Len (字符串)`

功能: 返回字符串中字符的个数。如:

`Len ("VBScript 函数")` '结果为: 10

2. 截取字符串

左截取: `Left (字符串, 截取的个数)`

右截取: `Right (字符串, 截取的个数)`

指定位置截取: `Mid (字符串, 截取开始位置, 截取个数)`

如:

`Left ("VBScript 函数有哪些", 5)` '结果为: VBScr

`Right ("VBScript 函数有哪些", 5)` '结果为: 函数有哪些

`Mid ("VBScript 函数有哪些", 5)` '结果为: ript 函数有哪些

`Mid ("VBScript 函数有哪些", 5, 5)` '结果为: ript 函

3. 删除空格

删除左边空格: `Ltrim(字符串)`

删除右边空格: `Rtrim(字符串)`

删除两端空格: `Trim(字符串)`

如:

`Ltrim (" 你好 ")` '结果为 "你好 "

`Rtrim (" 你好 ")` '结果为 " 你好"

`Trim (" 你好 ")` '结果为 "你好"



4.4.3 转换函数

1. 变体表达式转换成字符串

格式: Cstr(变体表达式)

功能: 将数值表达式的值转换成字符串。

如:

Cstr(123) 结果为: "123"

2. 变体表达式转换成整数

格式: CInt (x)

功能: 将 x 的值的小数部分四舍五入后, 返回一个整数值, 如:

CInt (5.2) =5

CInt ("5.2") =5

CInt (5.6) =6

CInt ("5.6") =5

3. 首字符转换成 ASCII 码

格式: Asc(字符表达式)

功能: 返回字符表达式的首字母的 ASCII 码。如:

Asc ("ABC") '结果为: 65 (即大写字母 A 的 ASCII 码)

4. 字符代码转换成字符

格式: Chr(数值表达式)

功能: 将数值表达式的值当作 ASCII 码, 返回该 ASCII 码对应的字符。如:

Chr (97) '结果为: a

Chr (13) '结果为: 回车符

5. 大小写转换

格式: UCase(字符串)

 LCase(字符串)

功能: UCase(字符串)返回指定字符串的大写; LCase(字符串) 返回指定字符串的小写。

如:

UCase("VBScript") '结果为: VBSCRIPT

LCase("VBScript") '结果为: vbscript

4.4.4 日期时间函数

1. 日期函数

格式: Date()

功能: 返回当前系统的日期。如:

Document.write (Date()) '结果为: 2007-2-14

注意: 日期函数的返回值依据系统上的区域设置来决定日期的显示格式。

2. 时间函数

格式:



（1）Time()

功能：返回当前系统的时间。如：

Document.write (Time()) '结果为：16:27:42

（2）Timer（）

功能：返回从午夜算起已过的秒数。

3. 日期时间函数

格式：Now()

功能：返回当前系统的日期和时间。如：

Document.write (Now()) '结果为：2007-2-14 16:27:42

4. 星期函数

格式：WeekDay(日期表达式)

WeekDayName(星期数)

功能：WeekDay 用于返回当前的星期数，其返回值为 1~7，其中 1 代表星期天，7 代表星期六；WeekDayName 返回制定星期数对应星期的名称。如：

a=#2007-02-14#

document.write(WeekDay(a)) '结果为“4”

document.write(WeekDayName(4)) '结果为“星期四”

5. 日期分量函数

格式：Year(日期表达式)

Month(日期表达式)

Day(日期表达式)

功能：Year、Month、Day 分别用于返回日期表达式中的年、月、日。如：

Year (#2007-02-14#) '结果为 2007

Month (#2007-02-14#) '结果为 2

Day (#2007-02-14#) '结果为 14

6. 时间分量函数

格式：Hour(时间表达式)

Minute(时间表达式)

Second(时间表达式)

功能：Hour、Minute、Second 分别用于返回时间表达式中的时、分、秒。如：

Hour (#16:27:42#) '结果为 16

Minute (#16:27:42#) '结果为 27

Second (#16:27:42#) '结果为 42

【例 4-5】显示系统当前日期和星期。

新建一个 Dreamweaver 文档，在<head>...</head>标记中输入如下脚本代码，保存为 date.html。

```
<script language="vbscript">
today=now()
document.write "今天是："&Year(today)&"年"&Month(today)&"月"&Day(today)&"日"
```



```
week=weekday(now())
select case week
    case "1"
        document.write "星期一"
    case "2"
        document.write "星期二"
    case "3"
        document.write "星期三"
    case "4"
        document.write "星期四"
    case "5"
        document.write "星期五"
    case "6"
        document.write "星期六"
    case "7"
        document.write "星期天"
end select
</script>
```

程序运行结果如图 4-7 所示。



图 4-7 日期星和期函数应用示例

4.4.5 测试函数

在 VBScript 中, 还提供了一些测试函数, 用来测试传入参数的各种形态。

IsNumeric: 返回 Boolean 值指明表达式的值是否为数字。

isArray: 返回 Boolean 值指明某变量是否为数组。

IsDate: 返回 Boolean 值指明某表达式是否可以转换为日期。

IsEmpty: 返回 Boolean 值指明变量是否已初始化。

Null: 返回 Boolean 值, 指明表达式是否不包含任何有效数据 (Null)。

【例 4-6】检测用户输入是否为数字。

新建一个 Dreamweaver 文档, 在<body>...</body>标记中输入如下脚本代码, 保存为 IsNumeric.html。

```
<script language="vbscript">
sub test_2()
dim num
num=form1.textfield1.value
if IsNumeric(num) then
```



```
msgbox("你输入的是") & num
else msgbox("你输入的不是数值数据")
end if
end sub
</script>
<form id="form1" name="form1" method="post" action="">
    题 2: 100 以内偶数的和是多少?
    请输入数值:
        <input name="textfield1" type="text" size="10" />
        <input type="submit" name="Submit" value="确定" onclick="test_2()" />
</form>
```

运行结果如图 4-8 所示:



图 4-8 测试函数应用示例

4.4.6 对话框函数

VBScript 中提供了多种用于产生对话框函数, 它们实际上都是 Window (窗口) 对象的方法函数或 API 函数。

1. 产生输入框

输入框函数 (inputbox) 函数显示一个能接受用户输入的对话框, 并返回用户在对话框中输入的信息。其语法格式为:

格式 1: inputbox (<信息内容>[,<对话框标题>][,<默认内容>])

说明:

① “信息内容” 指定在对话框中出现的文本。在“信息内容”中使用硬回车符 (CHR(13)) 可以使文本换行。对话框的高度和宽度随着“信息内容”的增加而增加, 最多可有 1024 个字符。

② “对话框标题” 指定对话框的标题。

③ “默认内容” 可以指定输入框的文本框中显示的默认文本。如果用户单击“确定”按钮, 文本框中的文本 (字符串) 将返回到变量中, 若用户单击“取消”按钮, 返回的将是一个零长度的字符串。

【例 4-7】输入用户密码并判断。

新建一个 Dreamweaver 文档, 在<head>...</head>标记中输入如下脚本代码, 保存为 password.html。



```
<script language="vbscript">
    password=inputbox("请输入你的密码")
    if(password<>"1234") then
        alert("你的密码不正确，禁止使用！")
    end if
</script>
```

执行效果如图 4-9 和图 4-10 所示。

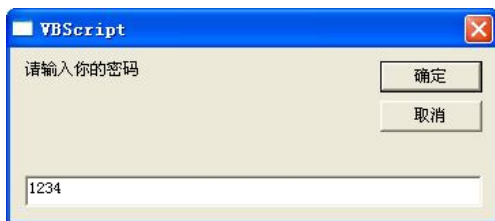


图 4-9 inputbox 输入框



图 4-10 输入密码错误弹出提示框

格式 2: `prompt("提示信息")`

如: `prompt("请输入您的姓名: ")`

弹出的提示框如图 4-11 所示，可以在框内输入信息，默认的信息为 `undefined`。在前面的例 4-2 中使用了 `prompt` 提示框。

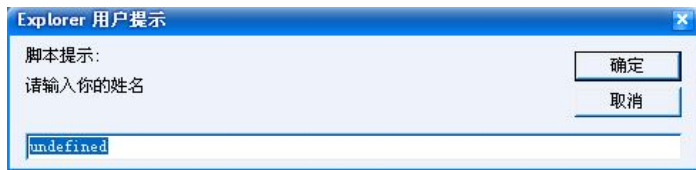


图 4-11 prompt 提示框

2. 产生消息框

用于产生一个弹出式的消息框，其图标为一个警告标识，表示警告、欢迎、提示等。

格式: `alert("消息内容")`

如: `alert("你的密码不正确，禁止使用！")`，效果如图 4-10 所示。

3. 产生选择框

用来产生一个选择框，供用户选择后执行相应的操作。

格式 1: `confirm("提示信息")`

如: `confirm("确定选择做这题吗? ")`

【例 4-8】选择框应用。

新建一个 Dreamweaver 文档，在 `<head>...</head>` 标记中输入如下脚本代码，保存为 `confirm.html`。

```
<script language="vbscript">
    confirm("你确定开始吗? ")
</script>
```

效果如图 4-12 所示。



图 4-12 confirm 选择框

格式 2: 变量 = msgbox (<信息内容>[,<对话框类型>[,<对话框标题>]])

信息框函数 (msgbox) 是常用的输出信息的函数, msgbox 函数在对话框中显示信息, 等待用户单击按钮, 并返回一个整数以标明用户单击了哪个按钮。

说明:

① “信息内容” 指定在对话框中出现的文本, 在信息内容中使用硬回车符 (CHR(13)) 可以使文本换行。对话框的高度和宽度随着信息内容的增加而增加, 最多可有 1024 个字符。

② “对话框类型” 指定对话框中出现的按钮和图标, 一般有三个参数。其取值和含义如表 4-3、表 4-4、表 4-5 所示。

表 4-3 参数 1 —— 出现按钮

值	常量	说明
0	vbOKOnly	确定按钮
1	vbOKCancel	确定和取消按钮
2	vbAbortRetryIgnore	终止、重试和忽略按钮
3	vbYesNoCancel	是、否和取消按钮
4	vbYesNo	是和否按钮
5	vbRetryCancel	重试和取消按钮

表 4-4 参数 2 —— 图标类型

值	常量	说明
16	vbCritical	停止图标
32	vbQuestion	问号 (?) 图标
48	vbExclamation	感叹号 (!) 图标
64	vbInformation	信息图标

表 4-5 参数 3 —— 默认按钮

值	常量	说明
0	vbDefaultButton1	指定默认按钮为第一按钮
256	vbDefaultButton2	指定默认按钮为第二按钮
512	vbDefaultButton3	指定默认按钮为第三按钮

上述三种参数值可以相加以达到所需要的样式。

③ “对话框标题” 指定对话框的标题。



④ `msgbox()` 返回的值指明了在对话框中选择哪一个按钮，如表 4-6 所示。

表 4-6 `msgbox` 函数的返回值

返回值	常量	按钮
1	<code>vbOK</code>	确定按钮
2	<code>vbCancel</code>	取消按钮
3	<code>vbAbort</code>	终止按钮
4	<code>vbRetry</code>	重试按钮
5	<code>vbIgnore</code>	忽略按钮
6	<code>vbYes</code>	是
7	<code>vbNo</code>	否

⑤ 代码中的值可以是数值，也可以是数值常量。

⑥ 如果省略了某些可选项，必须加入相应的逗号分隔符。

⑦ 若不需要返回值，则可以使用 `msgbox` 的命令形式：

`msgbox <信息内容>[,<对话框类型>[,<对话框标题>]]`

在程序运行的过程中，有时需要显示一些简单的信息如警告或错误等，此时可以利用信息对话框来显示这些内容。当用户接收到信息后，可以单击按钮来关闭对话框，并返回单击的按钮值。

【例 4-9】判断 2010 年是否闰年。

新建一个 Dreamweaver 文档，在 `<body>...</body>` 标记中输入如下脚本代码，保存为 `msgbox.html`。

```
<script language="vbscript">
  sub test_1()
    dim a
    if(form1.radiobutton2.checked) then
      a=msgbox("恭喜你答对了!",1)
    else
      a=msgbox("对不起，你答错了!",2)
    end if
  end sub
</script>
<form id="form1" name="form1" method="post" action="">
  题 1: 2010 年是闰年还是非闰年?
  <input type="radio" name="radiobutton1" value="radiobutton" />闰年
  <input type="radio" name="radiobutton2" value="radiobutton" />非闰年
  <input type="submit" name="Submit" value="确定" onclick="test_1()" />
</form>
```

执行效果如图 4-13、图 4-14 所示。



图 4-13 判断是否闰年



图 4-14 不同参数的 msgbox 选择框

4.5 VBScript 语句

4.5.1 语句

1. 程序语句

VBScript 程序中的一行代码称为一条程序语句，简称为语句。语句是执行具体操作的指令，每个语句行以回车（Enter）键结束。一个语句行的长度最多不能超过 1023 个字符。程序语句是 VBScript 关键字、属性、函数、运算符以及能够被浏览器识别的符号的任意组合。

建立程序语句时必须遵从的构造规则称为语法。编写正确程序语句的前提，就是学习语言元素的语法，并在程序中使用这些元素正确地处理数据。

2. 语句的书写规则

在编写程序代码时要遵循一定的规则，这样写出的程序既能被 VBScript 正确地识别，又能增加程序的可读性。

（1）一行中的多条语句。一般情况下，输入程序时要求一行写一个语句。但是也可以使用复合语句行，即把几个语句放在一个语句行中，语句之间用冒号“:”隔开。例如：

```
Text1.Text = "Hello" : Red = 255 : A = Red
```

（2）语句的续行。当一条语句很长时，在“记事本”窗口阅读程序时将不便查看，使用滚动条又比较麻烦。这时，就可以使用续行功能，用续行符“_”将一个较长的语句分为多个程序行。例如：

```
strMyStr="当前用户为：" & _  
strUsername
```

续行符只能出现在行尾。

3. 命令格式中的符号约定

为了便于解释语句、方法和函数，本教程语句、方法和函数格式中的符号采用统一约定。



在各语句、方法、函数的语法格式和功能说明中，以尖括号(<>)、方括号([])、花括号({})、竖线(|)、逗号加省略号(,…)、省略号(...)作为专用符号，这些符号的含义如表 4-7 所示。

表 4-7 约定的符号

符号	含义
<>	为必选参数表示符。尖括号中的中文提示说明，由使用者根据问题的需要提供具体参数。如果缺少必选参数，语句则发生语法错误
[]	为可选参数表示符。方括号中的内容选与不选由用户根据具体情况决定，且都不影响语句本身的功能。如省略，则默认为默认值
	为多中取一表示符，含义为“或者选择”。竖线分隔多个选择项，必须选择其中之一
{ }	包含多中取一的各项
, ...	表示同类项目的重复出现
...	表示省略了在当时叙述中不涉及的部分

注意：这些专用符号和其中的提示，不是语句行或函数的组成部分。在输入具体命令或函数时，上面的符号均不可作为语句中的成分输入计算机，它们只是语句、函数格式的书面表示。例如：

```
if<条件>then [<语句 1>] [ else<语句 2>]
```

4.5.2 条件语句

选择结构是一种可以根据条件实现程序分支的控制结构。其特点是，根据所给定的选择条件为真（即条件成立）或为假，而决定从各分支中执行某一分支的相应操作。

选择结构是通过条件语句来实现的。VBScript 中条件语句的主要形式如下：

1. If 语句

（1）选择分支语句。

```
格式：if<条件表达式> then
    [命令 1]
else
    [命令 2]
end if
```

功能：如果条件成立，执行 then 后的命令；反之，执行 else 后的命令。
如例 4-6 和例 4-9。

（2）多路选择分支语句。

```
格式：if<条件表达式 1> then
    [命令 1]
elseif <条件表达式 2> then
    [命令 2]]
elseif <条件表达式 3> then
    [命令 3]]
...
```



```
[else  
    [命令 n]]  
end if
```

【例 4-10】根据得分进行智力评价。

新建一个 Dreamweaver 文档，在<head>…</head>标记中输入如下脚本代码，保存为 result.html。

```
<script language="vbscript">  
    score=inputbox("请输入得分")  
    if score<30 then  
        alert("无话可说!")  
    elseif score>=30 and score<60 then  
        alert("继续努力!")  
    elseif score>=60 and score<100 then  
        alert("大有前途!")  
    else  
        alert("绝对天才!")  
    end if  
</script>
```

执行效果如图 4-15 及图 4-16 所示。

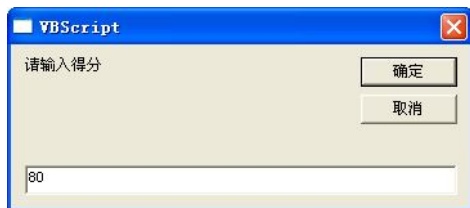


图 4-15 输入得分



图 4-16 显示智力评价结果

2. select case 语句

在上面的例题中，多个 elseif 子句会使程序可读性差，在多个条件中进行选择的更好方法是使用 select case…end select 语句。

格式：select case<表达式>

```
    [case<表达式 1>
```

```
        [命令 1]]
```

```
    [case<表达式 2>
```

```
        [命令 2]]
```

```
    ...
```

```
    [case else
```

```
        [命令 n]]
```

```
end select
```

如例 4-5 中显示当前是星期几。

4.5.3 循环语句

循环结构是一种可以根据条件实现程序循环执行的控制结构，一般有当型循环和直到型



循环两种结构，其他循环结构可以看作这两种结构的变形。

VBScript 常用的循环语句有 do...loop 和 for...next。

1. do...Loop 循环语句

(1) 当型循环结构

格式 1: do while<条件>

[命令 1]

[exit do] '在特定条件下退出循环

[命令 2]

loop

格式 2: do

[命令 1]

[exit do]

[命令 2]

loop while<条件>

两者区别：当第一次执行循环语句条件不成立时，格式 1 不执行循环体，而格式 2 要执行一次循环体。

注：当型循环语句，当条件为真（true）时执行循环体，条件为假（false）时终止循环。

【例 4-11】计算 100 以内偶数之和。

新建一个 Dreamweaver 文档，在<head>...</head>标记中输入如下脚本代码，保存为 do_loop.html。

```
<script language = "vbscript">
    dim i,sum
    i=2
    sum=0
    do while i<100
        sum=sum+i
        i=i+2
    loop
    msgbox ("100 以内偶数的和是"&sum)
</script>
```

效果如图 4-17 所示。



图 4-17 Do 循环语句示例

(2) 直到型循环结构

格式 1: do until<条件>

[命令 1]

[exit do]

[命令 2]

loop

格式 2: do

[命令]

[exit do]

[命令 2]

loop until<条件>

注：直到型循环结构，条件为假时执行循环体，直到条件为真时终止循环。

如：求 $1+2+\dots+100$ ，用 do until 描述。

格式 1: dim sum,i

sum=0

i=1

do until i>100

sum =sum+i

i=i+1

loop

格式 2: dim sum,i

sum=0

i=1

do

sum=sum+i

i=i+1

loop until i>100



2. while...wend 语句

当循环（while...wend）比较简单，其特点是，只要指定的条件为 true，则会重复执行一系列的语句。其语法格式为：

```
while<条件>
```

```
[<语句列>]
```

```
wend
```

说明：

① “条件”为必要参数，可以是关系表达式或布尔表达式。如果“条件”为 null，则“条件”会视为 false。

② 如果“条件”为 true，则所有的“语句列”都会执行，一直执行到 wend 语句。然后再回到 while 语句，并再一次检查“条件”，如果“条件”还是为 true，则重复执行。否则程序会从 wend 语句之后的语句继续执行。

【例 4-12】笼子里有若干只鸡和兔子，共有头 37 个，脚 134 只，问鸡和兔子各为多少？

新建一个 Dreamweaver 文档，在<head>...</head>标记中输入如下脚本代码，保存为 while_wend.html。

```
<script language = "vbscript">
    dim cock,rabbit,foot
    cook = 0
    foot=0
    while foot<>134
        cock=cock+1
        rabbit=37-cock
        foot=cock*2+rabbit*4
    wend
    msgbox("鸡有"& cock &"只；兔子有"& rabbit &"只")
</script>
```

效果如图 4-18 所示。



图 4-18 while...wend 应用示例

3. for...next 循环语句

格式：for<循环变量=初值>to<终值>[step 步长]

[命令]

[exit for] '用在特定条件下退出循环

[命令 2]

next

【例 4-13】计算 100 以内奇数之和。

新建一个 Dreamweaver 文档，在<head>...</head>标记中输入如下脚本代码，保存为 for_next.html。

```
<script language="vbscript">
    dim i,sum
    sum = 0
    for i = 1 to 99 step 2
        sum = sum + i
    next
```



```
next
msgbox ("100 以内奇数之和为"&sum)
</script>
```

效果如图 4-19 所示。



图 4-19 for...next 应用示例

4. for each...next 循环语句

for each...next 循环与 for...next 循环类似。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。其语法格式为：

```
for each<元素>in<名称>
```

```
    [语句列 1]
```

```
    [exit for]
```

```
    [语句列 2]
```

```
next [<元素>]
```

说明：

① “元素”是用来枚举集合或数组中所有元素的变量。对于集合，“元素”可能是 Variant 变量、通用 Object 变量或任意指定的 Automation 对象变量。对于数组，“元素”只能是 Variant 变量。

② “名称”是指对象集合或数组的名称。

【例 4-14】在窗体中有若干文本框，在代码中创建 Dictionary 对象，并赋值。然后将 Dictionary 对象的内容分别放置在多个文本框中。

新建一个 Dreamweaver 文档，在<head>...</head>标记中输入如下脚本代码，保存为 for_each_next.html。

```
<html>
<head><title>控件组</title></head>
<body>
    <Input Type = "Text" name = text1><br>
    <Input Type = "Text" name = text1><br>
    <Input Type = "Text" name = text1><br>
    <Input Type = "Text" name = text1><br>
    <Input Type = "Button" NAME="Button1" VALUE="单击此处"><p>
    <script for="Button1" event="onClick" language="VBScript">
        Dim d                                ' 创建一个变量
        Set d = CreateObject("Scripting.Dictionary")
        d.Add 0, "北京"                        ' 添加键和项目
        d.Add 1, "上海"
```



```
d.Add 2, "天津"
d.Add 3, "重庆"
For each i in d
    text1(i).Value = d.Item(i)
next
</script>
</body>
</html>
```

效果如图 4-20 所示。



图 4-20 for each...next 应用示例

4.6 VBScript 过程与函数

在程序设计中，一个应用程序可以由若干个程序模块组成，每个程序模块又可分为若干个程序段，每个程序段实现一个简单的逻辑功能，这个程序段就是过程。

函数也是一个程序段，通常用于实现某种特殊的计算，有一个返回值。

利用过程或函数，可以简化对程序的引用。在 VBScript 中，过程用 Sub 语句定义，函数用 Function 语句定义。

4.6.1 Sub 过程

定义 Sub 过程的格式为：

```
Sub<过程名>([<形参表>])
    [<语句列>]
    [ Exit Sub ]
    [<语句列>]
```

End Sub

调用过程：Call 过程名([实际参数])

或 过程名 [实际参数]

【例 4-15】Sub 过程应用。

新建一个 Dreamweaver 文档，在<body>...</body>标记中输入如下脚本代码，保存为 sub.html。

```
<script language="vbscript">
    sub test_2()
        dim score
        if(form1.textfield1.value="2450") then
            score=score+30
            msgbox("恭喜你答对了！得分："&score)
        else
            msgbox("对不起，你答错了！")
        end if
    end sub
</script>
<form id="form1" name="form1" method="post" action="">
```



请输入 100 以内偶数之和的值: >

</form>

效果如图 4-21 所示。



图 4-21 Sub 过程应用示例

4.6.2 Function 函数

定义 Function 函数的格式为:

Function<函数名>[(*<形参表>*)]

[*<语句列>*]

[*<函数名>=<表达式>*]

[Exit Sub]

[*<语句列>*]

[*<函数名>=<表达式>*]

End Function

其中“表达式”为 Function 过程的返回值。

调用函数: 函数名([实际参数])

【例 4-16】实现倒计时功能。

新建一个 Dreamweaver 文档, 输入如下脚本代码, 保存为 function.html。

```
<html>
<head>
<script language="VBScript">
    count=60 '倒计时的起始时间
    sub countDown()
        writeNumber(count)
        count=count-1
        If count>=0 then
            window.setTimeout"countDown()",1000 '每隔一秒钟调用本函数
        else
            window.close()
        end If
    end sub
    function writeNumber(count)
```



```
content.innerHTML="倒计时: <font size=4 color=red><b>"&count"&</b></font>
秒! "

    end function
</script>
</head>
<body onload="countDown()">
<div id="content"></div>
</body>
</html>
```

效果如图 4-22 所示。



图 4-22 Function 函数示例

4.7 使用 VBScript 对象编程

VBScript 是基于对象的程序设计语言，但它并不是真正面向对象的程序设计语言，因为它不具有面向对象的程序设计语言的全部特征。在面向对象的程序设计语言中，程序员可以自己定义对象；而在 VBScript 中，对象是系统定义好的，可以直接使用。

VBScript 采用的是面向对象、事件驱动编程机制，因此，必须理解对象以及对象的属性、事件和方法等概念。

4.7.1 对象的属性、事件和方法

VBScript 采用的是面向对象、事件驱动编程机制，因此，对象的概念是 VBScript 最基本的概念之一。

1. 对象的概念

在现实生活中，任何一个实体都可以视为一个对象（Object）。如一只气球是一个对象，一台电脑也是一个对象。一台电脑又可以拆分为主板、CPU、内存、外设等部件，这些部件又都分别是对象，因此电脑对象可以说是由多个“子”对象组成的。

从面向对象程序设计的角度来看，对象是具有属性（数据）和方法（行为方式）的实体。简单地说，属性用于描述对象的一组特征，方法为对象实施一些动作，对象的动作则常常要触发事件，而触发事件又可以修改属性。一个对象建立以后，其操作就通过与该对象有关的属性、事件和方法来描述。

VBScript 所支持的对象有脚本对象（如 Window 对象、Document 对象、Forms 对象等）和 DHTML 内部控件（Button 控件、Text 控件等）。

2. 对象的属性

在 VBScript 中，每一种对象都有一组特定的属性。有许多属性可能为大多数对象所共有，如 Name 属性定义对象的内部名称。还有一些属性只局限于个别对象才有，如只有文本框才有 maxLength 属性，该属性用来确定用户可以在文本框中输入的字符数。

3. 对象的事件

事件（Event）就是对象所发生的事情。在 VBScript 中，事件是预先定义好的、能够被对象识别的动作，如单击（Click）事件、双击（DbClick）事件、装载（Load）事件、鼠标移动（MouseMove）事件等，不同的对象能够识别不同的事件。当事件发生时，VBScript 将检测



两条信息，即发生的是哪种事件和哪个对象接收了事件。

每种对象能识别一组预先定义好的事件，但并非每一种事件都会产生结果，因为 VBScript 只是识别事件的发生。为了使对象能够对某一事件做出响应 (Response)，就必须编写事件过程。

4. 对象的方法

一般来说，方法就是要执行的动作。VBScript 的方法与事件过程类似，可以是函数，也可以是过程，它用于完成某种特定功能而不能响应某个事件。如 Window 对象的关闭 (Close) 方法、打开 (Open) 方法等。每个方法完成某个功能，但其实现步骤和细节，用户既看不到，也不能修改，用户能做的工作就是按照约定直接调用它们。

方法只能在代码中使用，其用法依赖于方法所需的参数的个数以及它是否具有返回值。当方法不需要参数并且也没有返回值时，可用下面的格式调用对象方法：

对象名.方法名 参数 1, 参数 2...

4.7.2 事件过程的调用方式

事件过程按其名称被识别，事件过程的命名规则是：on 事件名，如 Click 事件过程名为 onclick。事件过程的定义与调用有如下 4 种方式。

(1) 通过<script>标记块实现。

为每个事件过程分配一个<script>标记块，然后使用<script>标记的 event 属性调用事件过程，其语法格式为：

```
<script for=对象名 event=on 事件名 language=VBScript>
    事件过程代码
</script>
```

【例 4-17】华氏温度和摄氏温度相互转换的网页。利用输入框输入温度，利用信息框输出转换后的温度，输入一个华氏温度可以得到相应的摄氏温度，而输入一个摄氏温度则可以得到相应的华氏温度。

分析：设 C 为摄氏温度， F 为华氏温度，则有：

$$F = \frac{9}{5} \times C + 32$$

因此可以得到：

$$C = \frac{5}{9}(F - 32)$$

编写如下 HTML 代码：

```
<html>
<head><title>温度转换</title></head>
<body><h3 align = center>华氏温度和摄氏温度相互转换</h3><hr>
    <input type="Button" name="Button1" value="摄氏转华氏">
    <input type="Button" name="Button2" value="华氏转摄氏">
    <script for="Button1" event="onClick" language="VBScript">
        c = InputBox("请输入摄氏温度值：", "摄氏转华氏", 0)
        f = 32 + 9 * c / 5
        MsgBox("摄氏" & c & "度 = 华氏" & f & "度")
    </script>
```



```
<script for="Button2" event="onClick" language="VBScript">
  f = InputBox("请输入华氏温度值:", "华氏转摄氏", 0)
  c = 5 * (f - 32) / 9
  MsgBox "华氏" & f & "度 = 摄氏" & c & "度"
</script>
</body>
</html>
```

(2) 在对象定义标记中设置事件过程。通过在对象定义标记中设置事件过程可以直接调用事件过程代码，如下面代码：

```
<html>
<head><title>调用事件过程</title></head>
<body><h3>调用事件过程</h3><hr>
  <input type=button onclick = 'aa="这是按钮的单击事件": msgbox aa'
    value=确定 language=vbscript>
</body>
</html>
```

(3) 在对象定义标记中调用事件过程。在对象定义标记中设置事件过程属性可以调用命名 Sub 过程。上述代码可以改为：

```
<html>
<head><title>调用事件过程</title>
  <script language=vbscript>
    Sub mysub()
      aa = "这是按钮的单击事件"
      msgbox aa
    End Sub
  </script>
</head>
<body><h3>调用事件过程</h3><hr>
  <input type=button onclick=mysub value=确定>
</body>
</html>
```

(4) 单独定义事件过程。这种方式对事件过程的命名有严格的要求，过程名必须是：对象名_on 事件名

通常在脚本的<head>块中定义事件过程，如上述代码可以改为：

```
<html>
<head><title>调用事件过程</title>
  <script language=vbscript>
    Sub mybtn_onclick
      aa = "这是按钮的单击事件"
      msgbox aa
    End Sub
  </script>
</head>
<body><h3>调用事件过程</h3><hr>
  <input type=button name=mybtn value=确定>
```



```
</body>
</html>
```

4.7.3 Window 对象

Window 对象表示浏览器中一个打开的窗口。使用 Window 对象可以获得当前窗口的状态信息、文档信息、浏览器的信息，还可以响应发生在 IE 中的事件。通常，浏览器在打开 HTML 文档时，创建 Window 对象。

Window 对象包括的属性、方法和事件如表 4-8 所示。

表 4-8 Window 对象的属性、方法和事件

属性	方法	事件
Name	Alert	OnLoad
Parent	Confirm	OnUnload
Self	Prompt	
Top	Open	
Location	Close	
DefaultStatus	SetTimeout	
Status	ClearTimeout	
Document	Navigator	
Frames		
Navigator		
History		

下面说明 Window 对象经常使用的属性、方法和事件。

1. Alert、Prompt 方法

分别产生提示框及输入框。

2. Open、Close 方法

Open 用来打开一个页面。

【例 4-21】使用 open()方法打开 test.html 页面。

新建一个 Dreamweaver 文档，在<head>...</head>标记中输入如下脚本代码，保存为 open.html。

```
<script language="vbscript">
    window.open("test.html")
</script>
```

Close 用来关闭一个页面。

如：关闭窗口

程序运行后在页面上显示“关闭窗口”按钮，当用户单击“关闭窗口”按钮，则自动关闭当前 Web 页面。

3. Status 属性

更改浏览器状态栏的文字。如：



`window.status="欢迎光临本站！"`

结果为：在浏览器的状态栏上显示“欢迎光临本站！”

4. OnLoad、OnUnload 事件

OnLoad 事件是在页面完全传递到浏览器时发生的事件，OnUnload 事件是当离开页面时发生的事件。

如例 4-16 中的语句`<body onload="countdown()">`。

4.7.4 Document 对象

Document 对象代表了当前浏览器窗口中的 HTML 文档，它是脚本对象模型中最为重要的对象。可以通过该对象检测 HTML 文档中各元素的状态（如颜色、字体、发生的事件等），并且可以更改这些元素的状态。在 HTML 文档的任何位置都可以调用该对象。Document 对象的属性和方法如表 4-9 所示。

表 4-9 Document 对象的属性和方法

属性	描述	方法	描述
AlinkColor	激活的链接的颜色	Write	在 HTML 中写入新的代码
VlinkColor	访问过的链接颜色	WriteLn	在 HTML 中写入新的代码(并换行)
LinkColor	链接的颜色	Close	关闭一个 HTML 文档
Title	文档标题	Clear	清除一个 HTML 文档的内容
Anchor	书签	Open	打开一个 HTML 文档
Cookie	在客户机存放的反映客户信息的数据	Settimeout	创建一个定时器
Forms	文档中的表单	ClearTimeout	关闭定时器
Location	文档的 URL		
Referref	返回前一个 URL		
Links	页面中的链接，数组形式		
LastModified	文档的最后修改日期		

下面简单说明 Document 对象经常使用的属性、方法和事件。

1. LastModified 属性

使用 LastModified 属性可以在页面上自动产生最后修改页面的日期。如：

`document. Write(document.lastmodified)`

结果为：2010-1-1 12:21:32（假设当前日期为 2010-1-1 12:21:32）。

2. Write 方法

使用 Write 方法可以动态地在 HTML 文档中写入新的 HTML 代码，如：

`document. Write("欢迎光临我的主页！")`

可以加入修饰的标记代码：

`document. Write("<font color='red'>欢迎光临我的主页！</font>")`

3. Document 对象的事件

Document 对象常见的事件如表 4-10 所示。



表 4-10 Document 对象的事件

事件	描述
onclick	单击鼠标时产生
ondblclick	双击鼠标时产生
onkeydown	按下某个键时产生
onkeypress	按下某个键后又松开时产生
onkeyup	松开某个键时产生
onmousedown	按下鼠标时产生
onmouseup	松开鼠标的键时产生
onmouseover	鼠标移到某个元素上时产生
onmouseout	鼠标从某个元素上移开时产生
onmousemove	移动鼠标时产生

通常情况下，在文档中与 Document 对象相对应的 HTML 标记是<body>，所以，调用 Document 对象的事件过程一般可以通过在<body>标记中添加相应代码来实现。

4.8 案例实现

学习完本章上述内容后，可以来完整地制作本章案例，通过对本章案例各个页面的制作，熟练掌握 VBScript 的基本数据类型、常量与变量、运算符与表达式的使用、语句控制的方法、常见内置函数的应用、VBScript 过程、VBScript 对象编程等基础知识。

下面对案例系统实现过程进行分析。

4.8.1 设计思路

1. 系统功能定位和模块划分

本案例以一个简单的趣味智力测试系统为引线，将 VBScript 脚本编程的知识要点有机连接起来。

案例系统的功能模块主要包括：

- ① 站点首页：显示当前日期与星期，打开测试主页，显示状态栏信息。
- ② 测试主页：输入用户名，检测密码，显示欢迎用户信息；答题，得分，智力评价；倒计时。

2. 站点设计

- ① 在 D 盘根目录下创建本地站点 mysite，并创建相关文件夹；
- ② 根据案例结构，创建各页面文件；主要页面有站点首页 login.html 和测试主页 test.html，根据知识点结构将这两个页面分解成 date.html 等多个功能页面。
- ③ 根据页面设计需求，实现各页面功能。设计完毕案例站点结构如图 4-23 所示；

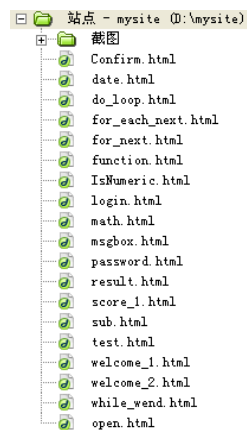


图 4-23 案例站点目录

4.8.2 案例结构

本章案例各页面关系如图 4-24 所示。

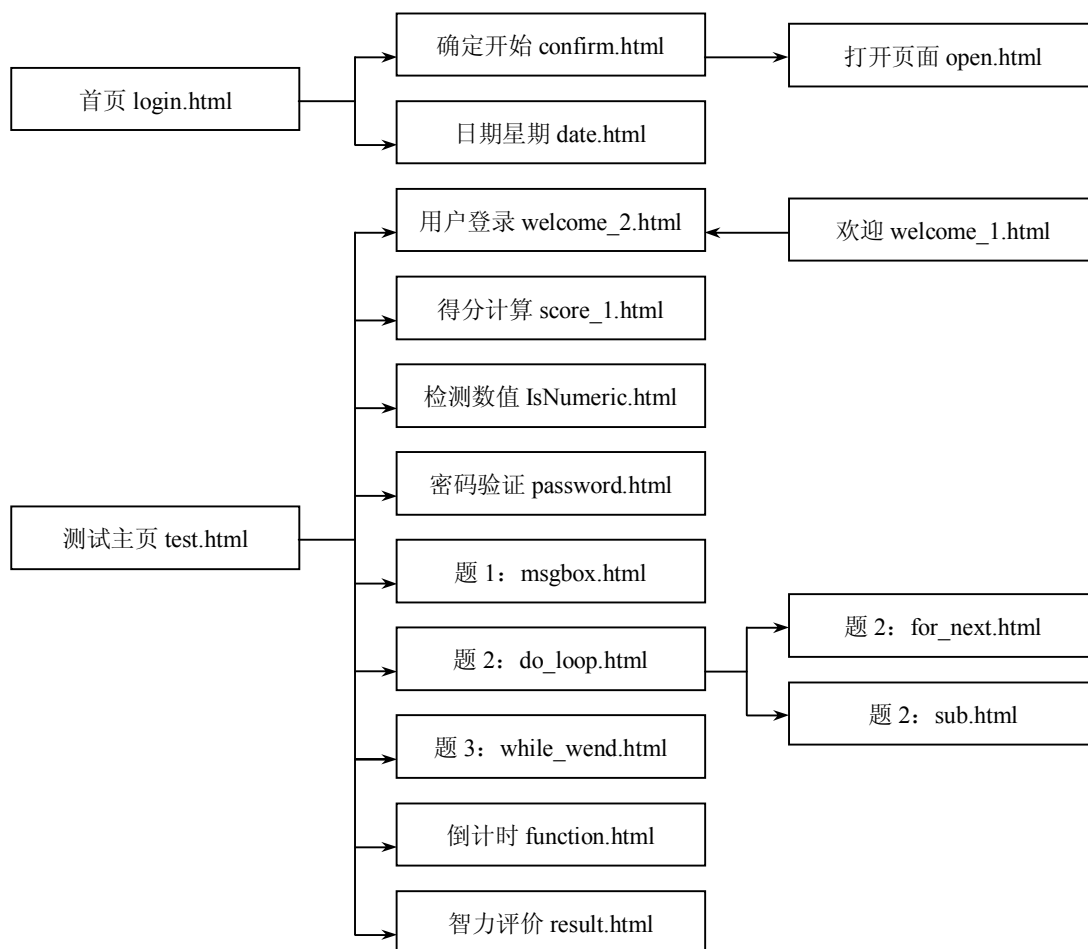


图 4-24 案例结构图



4.8.3 页面解析

本章案例中各页面的功能设计、知识要点解析如表 4-11 所示。

表 4-11 案例各页面解析

页面	实现功能	知识要点	说明
confirm.html	信息确认	confirm()函数	例 4-8
open.html	弹出窗口页面	open()方法	例 4-7
date.html	显示日期与星期	日期函数和星期函数 select case 语句	例 4-5
welcome_1.html	输出文本	document.write 语句	例 4-1
welcome_2.html	输入姓名	变量赋值	例 4-2
score_1.html	计算答题得分	变量作用域	例 4-3
IsNumeric.html	检测数值	IsNumeric 函数	例 4-6
password.html	输入密码	inputbox ()函数	例 4-7
	密码正确提示	alert()函数	
msgbox.html	题 1: 判断是否闰年	Sub 过程、msgbox ()函数	例 4-9
do_loop.html	题 2: 计算 100 以内偶数之和	do...loop 循环语句 msgbox ()函数	例 4-11
for_next.html	题 2 拓展: 计算 100 以内奇数之和	for...next 循环语句 msgbox ()函数	例 4-13
sub.html	题 2: 计算 100 以内偶数之和	Sub 过程、if 循环语句 msgbox ()函数	例 4-15
while_wend.html	题 3: 计算鸡兔多少	while...wend 循环语句 msgbox ()函数	例 4-12
function.html	倒计时	Function 函数、Sub 过程 Close()方法、SetTimeout 方法	例 4-16
result.html	智力测试评价	inputbox ()函数、alert()函数和 If 循环语句	例 4-10

通过创建本案例系统,进一步加深对知识要点的理解和掌握,并能够独立编写各例的脚本代码。



本章主要介绍在网页中使用 VBScript 进行脚本编程的基本知识,是 ASP 程序设计的基础。内容主要包括 VBScript 的基本概念,如何在网页中嵌入 VBScript 脚本,VBScript 的基本数据类型、常量与变量、运算符与表达式的使用、语句控制的方法、常见内置函数的应用、VBScript 过程、VBScript 对象编程等。

VBScript 是一种基于对象的脚本语言,是 VB 的子集。在网页文件中可直接嵌入 VBScript



脚本，以控制网页的控件和对象，增强网页的灵活性和多样性，而且还可以对用户的操作作出反应。

VBScript 可在客户端进行编程，也可作为服务器端编程语言。因此，它可以在 HTML 网页中使用，也可在 ASP 网页中使用，但使用格式有所差异。

VBScript 中只有一种数据类型，称为 Variant（变体型）。根据使用方式的不同，Variant 可以包含不同类型的信息，即这种数据类型可以根据用途的需要选择最合适的子类型来存储数据。

在 VBScript 中，定义常量使用 Const 语句，对变量的定义可以直接使用，也可以在使用前事先定义。变量声明的位置不同，其作用域也不同。

VBScript 的运算符包括算术、字符串（连接）、逻辑和比较 4 种。

在 VBScript 中经常用到的函数有数学运算函数、字符串函数、日期函数、数据类型转换函数和多种用于产生对话框的函数。

VBScript 语句主要包括条件语句与循环语句，条件语句有 if 语句、select case 语句；循环语句有当型和直到型循环语句。

在 VBScript 中，为了简化程序，经常需要使用 Sub 过程和 Function 函数。

可以使用 VBScript 对象模型进行编程，如 Window 对象、Document 对象等。

关于本章内容的进一步学习，读者可以参阅以下网站：

<http://www.duoko.com>

<http://www.adminlm.net/jc/VBscript>

<http://ie.zzu.edu.cn/jiaoxue/asp/index.asp>

<http://www.bianceng.cn/vbscript>

<http://www.176net.com/article/wangzhankaifa/vbscript>



习题 4

一、选择题

1. 以下对 VBScript 描述，不正确的是（ ）。
A. 是基于对象的脚本语言 B. 是 VB 的子集
C. IE 浏览器不支持 VBScript D. 通常用于 ASP 服务器端编程
2. 在 VBScript 中，注释采用（ ）。
A. // B. /*...*/ C. ' D. "
3. 若 VBScript 的变量在使用前必须事先定义，则应使用（ ）语句设置。
A. Option Explicit B. Option Base C. Dim D. ReDim
4. 在 VBScript 中，日期常量必须用（ ）括起来。
A. '' B. "" C. ; D. #
5. 可用于计算一个表达式的值的函数是（ ）。
A. Fix() B. Eval() C. Val() D. Round()
6. 若要获得当前系统日期为星期几，可使用（ ）函数实现。



- A. WeekDay() B. Date() C. Time() D. Now()
7. 退出 do 循环, 应使用语句 ()。
- A. exit B. exit sub C. exit for D. exit do
8. 若要定义一个有 4 个成员的数组 answer, 以下定义方法中, 正确的是 ()。
- A. Dim answer (5) B. ReDim answer(5)
C. Dim answer (4) D. ReDim answer(4)
9. 可以用来产生一个接收用户输入信息的输入框的函数有 ()。
- A. prompt() B. inputbox ()
C. alert() D. prompt()和 Inputbox ()
10. 使用 () 方法可以动态地在 HTML 文档中写入代码。
- A. alert() B. write() C. open() D. colse()

二、填空题

1. ASP 的服务器端脚本编程通常采用_____, 它是 Visual Basic 的子集。
2. Microsoft 公司的_____浏览器支持 VBScript, 也支持 JavaScript。
3. 在 VBScript 中, 字符连接可使用的运算符有_____或_____。
4. 在 VBScript 中, 可用_____函数来获取当前时间。
5. 编写 HTML 静态网页时, 从兼容性考虑, 脚本语言宜选用_____。
6. VBScript 的运算符包括_____, _____、逻辑和比较 4 种运算符。
7. 在 VBScript 中, 为了简化程序, 经常需要使用_____过程和_____函数。
8. 在 VBScript 中, 定义常量使用_____语句。
9. VBScript 是一种基于_____的脚本语言。
10. 使用 Document 对象的_____属性可以在页面上自动产生最后修改页面的日期。

三、程序分析题

1. 编写一个随意确定 N 的数值, 并计算且显示出 $1+2+3+\cdots+N$ 的值的页面。

```
<html>
<body>
<script language="vbscript">
N=_____
Sum=0
For i=1 to N
_____
Next
Document.write("N 的值为: " _____)
</Script>
</body>
</html>
```

2. 使用 VBScript 中的事件: 页面中只显示一个“点击试一下”的按钮; 当用户单击“点击试一下”按钮后, 弹出对话框: 你成功了!

```
<html>
```



```
<body>
<form>
<input type="_____" value="点击试一下" onclick=_____>
</form>
</body>
</html>
```

3. 交换变量内容。

```
Sub Swap(x, y)
    Dim t
    _____ ' 交换两个变量的值
End Sub

Sub Button1_onClick()
    Dim a, b
    a = Form1.Text1.Value
    b = Form1.Text2.Value

    _____
    Form1.Text1.Value = a
    Form1.Text2.Value = b
End Sub
```

四、编程题

1. 编写一个 Web 页面，在状态栏中显示系统当前时间。
2. 编写一个求最大值的 HTML 页面，输入两个整数并求出较大者。
3. 利用 Window 对象的有关方法实现在状态栏显示滚动文本“欢迎光临!”。

实训 4 VBScript 脚本编程

一、实验目的与要求

- (1) 掌握 VBScript 中常用函数的使用。
- (2) 掌握 VBScript 常用程序控制语句的使用。
- (3) 掌握 VBScript 中 Sub 过程与 Function 函数的使用。
- (4) 掌握 Window 对象、Document 对象常用的属性、方法和事件的使用。

二、实验内容

1. 使用日期时间函数 Date()、Time()、Now()，获取系统当前日期与时间。如图 4-25 所示。



图 4-25 日期时间函数应用



2. 使用对话框函数 `inputbox()`、`msgbox()`，输入一个年份数字，并判断是“闰年”或“非闰年”。如图 4-26 和图 4-27 所示。

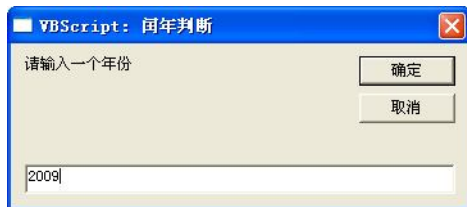


图 4-26 `inputbox()` 函数应用



图 4-27 `msgbox()` 函数应用

3. 使用数组和随机数函数 `Rnd()`，编写一个随机显示信息的页面，当单击“随机字符”按钮时，把获得的随机字符串显示到文本框中。如图 4-28 所示。

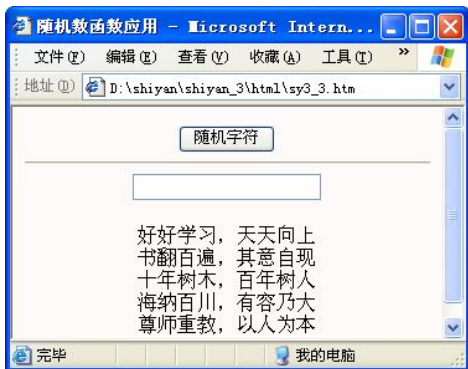


图 4-28 `Rnd()` 函数应用

4. 使用 `if` 语句，编写一个数值比较程序。在表格的三个文本框中任意输入三个数字，单击“排序”按钮，在弹出的消息窗中显示最大数和最小数。如图 4-29 所示。



图 4-29 `if` 语句应用

5. 使用 `if...elseif` 语句，编写一个背景和文字颜色变化程序。当用户选择背景颜色或文字颜色时，页面中的背景颜色或文字颜色将会发生相应的变化。如图 4-30 所示。



图 4-30 if...elseif 语句应用

6. 使用 do while 语句，编写一个动态输出表格的程序。如图 4-31 和图 4-32 所示。



图 4-31 do while 语句应用



图 4-32 do while 语句应用

7. 编写一个倒计时程序。页面加载时，首先出现一个从 10~1 的秒数倒计时，计时结束跳转到 a.htm 页面。如图 4-33 所示。



图 4-33 Sub 过程与 Function 函数应用