

网站建设系列丛书

编著
风火轮小组



ASP

建站编程高手指南

大恒电子出版社

网站建设系列丛书

ASP 建站编程高手指南

编著 风火轮小组

大恒电子出版社

内容提要

在本书的第一章,我们从宏观上向大家介绍了 asp 的相关知识,力争使大家经过第一章的学习能够对 asp 有一个整体上的认识,便于以后的具体内容的学习。在第二章,我们详细为大家讲述了 javascript,众所周知,asp 技术中的各种算法的实现是依赖于脚本语言的。离开脚本,asp 便失去了灵魂。对于初级读者,请认真的学好本章所讲知识,对于以后的实际编程会有很大的帮助。另外,由于本书的例程主要使用了 vbscript,我们在本章也向读者简要介绍了 vbscript 的相关知识。从第三章开始,我们正式开始介绍 asp 的组成。当然只是限于组成方面的知识。Asp 的灵魂——五个内置对象,我们在第四章进行了讲述。第五章讲述的是 asp 组建的相关知识。这是相当重要的一部分,有了它,我们才能随心所欲的使用 asp 实现各种功能。第六至第七章我们讲述了数据库访问语言——SQL 语句,由浅入深的讲述将使读者容易掌握。第八章是数据库的访问方式,这是不可不认真对待的知识,希望读者在学习的过程中能多结合实际编程,那样效果会更好。至此,读者就可以编制完整的 asp 页面了。在第九章,我们为大家准备了很多编程高手的实际编程技巧,在大家学到基本技巧后,若有兴趣,可参看本章,结合自己的编程体会,加以实际应用。最后一部分的编程实例,可供读者参考。

另外,本书有些实例程序未提供实际显示界面,但各程序均已经过实际调试,读者可放心参考使用。

网站建设系列丛书

ASP 建站编程高手指南

著译:风火轮小组

责任编辑:李振华

出版:大恒电子出版社

经销:各地新华书店软件连锁店

CD 生产:北京维宝光盘有限公司

文本印刷:北京北七家印刷厂

开本/规格:787×1092 1/16 开 20.5 印章 230 千字

版次/印次:2002 年 5 月第一版 2002 年 5 月第一次印刷

印数:0001-5000 册

本版号:ISBN 7-900092-76-5/TP.76

定价:50 元(1CD 含配套书)

说明:凡光盘配套图书若有自然损坏、缺页、脱页,本社负责调换。

前 言

如今世界因为 Internet 的蓬勃发展而变得越来越小,人们交换信息的途径也因此发生了翻天覆地的变化,所以网络开发技术变得越来越重要。在各种开发技术中,Microsoft 推出的 asp 技术使得人机间的交互第一次达到了随心所欲的程度。经过几年的发展,asp 已成为一项成熟的技术,广泛的应用于各种网站的建设,应广大读者的要求,我们特地编写了这本书,向大家详细介绍了 asp 的相关知识。

本书编写过程中考虑到了各个阶段读者的需求,从基本的脚本语言讲起,让从未有过网络编程经验的读者经过本书的学习也能达到熟练开发各种网络应用程序的程度。对于以前曾有过应用 asp 编程经验的读者,可跳过前面三章,从第四章学起,本书将为读者进行 asp 高级开发技术的讲解,从而使读者能够更上一层楼。

在当今各种相关书籍中,脚本语言部分大多讲述的是 vbscript,因此,在本书中我们着重讲述了 javascript, vbscript 知识只进行了简略的介绍,而书内的例子大多还是应用 vbscript 作为脚本语言,读者若在此方面有什么困惑,请自行查阅相关书籍。

另外,在书的最后部分我们附上了大量的实例,请读者自行参考调试运行,当然,若想达到熟练编程的程度,少不了读者自己多加练习。

在本书的编著过程中,得到了身边亲友的大力帮助,在此对他们表示衷心的感谢!

最后,祝大家学有所成,家庭幸福。

编者

2002 年 4 月

目 录

第一章 初识 ASP

1.1 初识 ASP	1
1.2 ASP web 动态编程	5

第二章 ASP 脚本语言

2.1 Java Script 语言概况	10
2.1.1 什么是 JavaScript	10
2.1.2 JavaScript 和 Java 的区别	11
2.1.3 编写第一个 JavaScript 程序	12
2.2 JavaScript 基本数据结构	13
2.2.1 JavaScript 代码的加入	13
2.2.2 基本数据类型	14
2.2.3 表达式和运算符	15
2.2.4 范例	16
2.3 JavaScript 程序构成	17
2.3.1 程序控制流	17
2.3.2 函数	18
2.3.3 事件驱动及事件处理	19
2.3.4 范例	20
2.4 基于对象的 JavaScript 语言	23
2.4.1 对象的基础知识	23
2.4.2 常用对象的属性和方法	25
2.4.3 范例	27
2.5 创建新对象	29
2.5.1 对象的定义	29
2.5.2 创建对象实例	30
2.5.3 对象方法的使用	30
2.5.4 JavaScript 中的数组	30
2.5.5 范例	32
2.6 使用内部对象系统	36
2.6.1 浏览器对象层次及其主要作用	36
2.6.2 文档对象功能及其作用	36
2.6.3 范例	38
2.7 窗口及输入输出	39
2.7.1 窗口及输入输出	39
2.7.2 简单的输入、输出例子	41
2.7.3 范例	43
2.8 VBScript 语言概况	43
2.8.1 VBScript 的变量	44
2.8.2 程序控制流语句	45
2.8.3 循环语句的使用	48
2.8.4 VBScript 过程	51
2.9 在 Web 网页上使用 VBScript 和 Jscript	53
2.9.1 VBScript 和 JScript 之间的区别	
2.9.2 什么时候选择 VBScript	
2.9.3 什么时候选择 Jscript	
2.9.4 如何在 Visual InterDev 中明确所用的脚本编写语言	
2.9.5 在 Visual InterDev 1.0 中为 Script Wizard 设置语言	
2.9.6 在 Visual InterDev 6.0 中为脚本生成工具设置一个网页中的语言	
2.9.7 Visual InterDev 6.0 项目中改变脚本生成工具的缺省语言设置	
2.9.8 在同一个网页中同时使用几种语言	
2.9.9 在一个 VBScript 调用中的括号	
2.9.10 服务器脚本的执行顺序	
2.9.11 脚本块的顺序	
2.9.12 区分大小写以及命名习惯	
2.9.13 函数调用中把 JScript 单元的名字作为一个字符串参数来使用	
2.9.14 从 VBScript 往 JScript 中传送数组	
2.9.15 更新脚本语言	

第三章 ASP 基础

3.1 ASP 文件的结构	
3.2 ASP 的基本语法	
3.3 声明脚本语言	
3.4 <Script>标记和<%.....%>标记的区别	

3.5	HTML 文件中应用 ASP 变量	66
3.6	在 ASP 中使用基本脚本语言	68
3.6.1	do...loop 语句	68
3.6.2	For...next 语句	69
3.6.3	格式化日期	70
3.6.4	格式化数字	71
3.6.5	使用数学函数	72
3.6.6	split 函数的使用	73
3.6.7	使用 with 语句	73
3.6.8	调用函数和过程	74
3.7	使用 Server-Side Include (SSI)	76
3.7.1	SSI 的功能	76
3.7.2	#include 指令	76
3.7.3	使用动态的 SSI	76
3.7.4	应用程序的输出	77
3.7.5	包含文件的大小	77
3.7.6	#config	77
3.7.7	#lastmod	78
3.7.8	.inc 文件	78
3.8	ASP 使用小结	79
3.8.1	作为单个表达式使用	79
3.8.2	与脚本语言一起使用	79
3.8.3	嵌在 HTML 语言中使用	79
3.9	测试 Web 服务器	80

第四章 ASP 的五个内置对象

4.1	ASP 内建对象 Request	83
4.2	ASP 内建对象 Response	87
4.2.1	属性	87
4.2.2	方法	88
4.2.3	集合	88
4.3	ASP 内建对象 Application 和 Session	90
4.3.1	Application	90
4.3.2	Session	92
4.4	Global.asa 文件的使用及 Chat 程序	94
4.5	ASP 内建对象 Server	98
4.5.1	语法	98
4.5.2	属性	98
4.5.3	方法	98

第五章 功能强大的内置组件

5.1	ActiveX 组件	101
5.2	常用的 asp activex 组件	104
5.2.1	Database Access 组件	104
5.2.2	Content Linking 组件	106
5.3	其它的 ASP 常用组件	108
5.4	编写自己的组件	112
5.4.1	创建 Java ActiveX 财务组件	112
5.4.2	从脚本调用 Finance 组件	117

章 数据库操作语言——SQL 语言基础

6.1	SQL 简介	119
6.2	SQL 功能介绍	120
6.3	ISQL 执行 SELECT 查询	121
6.4	操作多个表	123
6.5	操作字段	124
6.6	排序查询结果	126
6.7	取出互不相同的记录	128
6.8	创建新表	128
6.9	字段类型	129
6.9.1	字符型数据	129
6.9.2	文本型数据	130
6.9.3	数值型数据	130
6.9.4	存储逻辑值	132
6.9.5	存储日期和时间	132
6.10	字段属性	132
6.10.1	允许和禁止空值	132
6.10.2	缺省值	133

6.10.3 标识字段	133
6.11 使用 SQL 事务管理器创建新表	134
6.12 向表中添加数据	135
6.13 删除和修改表	135

第七章 SQL 语言详解

7.1 建立索引	137
7.2 聚簇索引和非聚簇索引	137
7.3 用 SQL 建立索引	138
7.4 用事务管理器建立索引	140
7.5 SQL 核心语句	140
7.5.1 插入数据	140
7.5.2 删除记录	141
7.5.3 更新记录	142
7.5.4 用 SELECT 创建记录 and 表	143
7.6 集合函数	144
7.6.1 统计字段值的数目	144
7.6.2 计算字段的平均值	145
7.6.3 计算字段值的和	145
7.6.4 返回最大值或最小值	145
7.7 其它常用的 SQL 表达式, 函数, 和过程	146
7.7.1 通过匹配一定范围的值来取出数据	146
7.7.2 转换数据	147
7.7.3 操作字符串数据	148
7.7.4 操作日期和时间	151
7.8 发送邮件	156

第八章 asp 访问数据库的三种方式

8.1 Internet 数据库接口 (IDC)	158
8.2 ActiveX 数据对象 (ADO)	159
8.2.1 创建 ODBC DSN 文件	159
8.2.2 配置 SQL Server 数据库文件 DSN	160
8.2.3 SQL server 连接和安全信息	160
8.2.4 配置 Oracle 数据库文件 DSN	161
8.2.5 连接数据库	161
8.2.6 用 Connection 对象执行查询	161
8.2.7 使用 Recordset 对象处理结果	162
8.2.8 检索记录	163
8.2.9 结合 HTML 表格和数据库访问	165
8.2.10 管理数据库连接	166
8.2.11 共享连接	166
8.2.12 使用跨页连接	166
8.2.13 分页显示	167
8.3 远程数据服务 (RDS)	169
8.4 ASP 访问 SQL Server 内置对象	172
8.5 在 A S P 中用集成批操作数据库	173
8.5.1 H T M L 的集合属性	173
8.5.2 H T M L 的集合属性的应用	174
8.5.4 列出你的所有 Session 变量	176

第九章 如何增强 ASP 程序性能

9.1 概述	178
9.2 ASP 技巧	178

第十章 ASP 编程实例

10.1 不用组件上载文件代码具体例子	192
10.2 个人主页留言簿的制作	197
10.3 用 asp 建立网上论坛	205
10.4 利用 ASP 开发网络聊天室	245
10.5 用 ASP 开发一个在线考试程序	264
10.6 广告播放和跟踪系统的制作	265
10.7 利用 ASP 制作网络传呼机	267
10.8 统计在线人数	278

附录

附录一 基本函数	287
附录二 常见错误调试信息	313

第一章 初识 ASP

Microsoft Active Server Pages 即我们所称的 ASP，其实是一套微软开发的服务器端脚本环境，ASP 内含于 IIS 3.0 和 4.0 之中，通过 ASP 我们可以结合 HTML 网页、ASP 指令和 ActiveX 元件建立动态、交互且高效的 WEB 服务器应用程序。有了 ASP 你就不必担心客户的浏览器是否能运行你所编写的代码，因为所有的程序都将在服务器端执行，包括所有嵌在普通 HTML 中的脚本程序。当程序执行完毕后，服务器仅将执行的结果返回给客户浏览器，这样也就减轻了客户端浏览器的负担，大大提高了交互的速度。以下罗列了 Active Server Pages 所独具的一些特点：

1. 使用 VBScript、JScript 等简单易懂的脚本语言，结合 HTML 代码，即可快速地完成网站的应用程序。

2. 无须 compile 编译，容易编写，可在服务器端直接执行。

3. 使用普通的文本编辑器，如 Windows 的记事本，即可进行编辑设计。

4. 与浏览器无关 (Browser Independence)，用户端只要使用可执行 HTML 码的浏览器，即可浏览 Active Server Pages 所设计的网页内容。Active Server Pages 所使用的脚本语言 (VBScript、Jscript) 均在 WEB 服务器端执行，用户端的浏览器不需要能够执行这些脚本语言。

5. Active Server Pages 能与任何 ActiveX scripting 语言相容。除了可使用 VBScript 或 JScript 语言来设计外，还通过 plug-in 的方式，使用由第三方所提供的其他脚本语言，譬如 REXX、Perl、Tcl 等。脚本引擎是处理脚本程序的 COM(Component Object Model) 物件。

6. Active Server Pages 的源程序，不会被传到客户浏览器，因而可以避免所写的源程序被他人剽窃，也提高了程序的安全性。

7. 可使用服务器端的脚本来产生客户端的脚本。

8. 物件导向 (Object-oriented)。

9. ActiveX Server Components(ActiveX 服务器元件) 具有无限可扩充性。可以使用 Visual Basic、Java、Visual C++、COBOL 等编程语言来编写你所需要的 ActiveX Server Component。

ASP 的奇妙之处真是不胜枚举，下面就请各位系好安全带，我将带领大家进入 ASP 的梦幻世界。

1.1 初识 ASP

首先，让我们来看看运行 ASP 所需的环境：

- Microsoft Internet Information Server version 3.0/4.0 on Windows NT Server
- Microsoft Peer Web Services Version 3.0 on Windows NT Workstation
- Microsoft Personal Web Server on Windows 95/98

正如前文所述，与一般的程序不同，.ASP 程序无须编译，ASP 程序的控制部份，是使用 VBScript、JScript 等脚本语言来设计的，当执行 ASP 程序时，脚本程序将一整套命令发送给脚本解释器（即脚本引擎），由脚本解释器进行翻译并将其转换成服务器所能执行的命令。当然，同其他编程语言一样，ASP 程序的编写也遵循一定的规

则，如果你想使用你所喜爱的脚本语言编写 ASP 程序，那么你的服务器上必须要有能解释这种脚本语言的脚本解释器。当你安装 ASP 时，系统提供了两种脚本语言：VBscript 和 JScript，而 VBScript 则被作为系统默认的脚本语言。你也可以根据自己的喜好改变系统默认的脚本语言，关于如何改变系统默认的脚本语言请看《在 ASP 中使用脚本语言》一文。

ASP 本身并不是一种脚本语言，它只是提供了一种使镶嵌在 HTML 页面中的脚本程序得以运行的环境。但是，要学好 ASP 又必须掌握它的语法规则。现在就让我们开始一步一步地认识并学习 Active Server Pages。

ASP 程序其实是以扩展名为 .ASP 的纯文本形式存在于 WEB 服务器上的，你可以用任何文本编辑器打开它，ASP 程序中可以包含纯文本、HTML 标记以及脚本命令。你只需将 .ASP 程序放在 WEB 服务器的虚拟目录下（该目录必须要有可执行权限），就可以通过 WWW 的方式访问 ASP 程序了。要学好 ASP 程序的设计，必须掌握脚本的编写，那么究竟什么是脚本呢？其实脚本是由一系列的脚本命令所组成的，如同一般的程序，脚本可以将一个值赋给一个变量，可以命令 WEB 服务器发送一个值到客户浏览器，还可以将一系列命令定义成一个过程。要编写脚本，你必须熟悉至少一门脚本语言，如 VBScript。脚本语言是一种介乎于 HTML 和诸如 JAVA、Visual Basic、C++ 等编程语言之间的一种特殊的语言，尽管它更接近后者，但它却不具有编程语言复杂、严谨的语法规则。如前所述 ASP 所提供的脚本运行环境可支持多种脚本语言，譬如：JScript、REXX、PERL 等等，这无疑给 ASP 程序设计者提供了广泛的发挥余地。ASP 的出现使得广大 WEB 设计者不必在为浏览器是否支持而担心，实际上就算你在同一个 .ASP 文件中使用不同的脚本语言，你都无须为此担忧，因为所有的一切都将在服务器端进行，客户浏览器得到的只是一个程序执行的结果，而你也只需在 .ASP 中声明使用不同的脚本语言即可。下面是一个典型的在同一 .ASP 文件中使用两种脚本语言的例子：

```
<HTML>
<BODY>
<TABLE>
<% Call Callme %>
</TABLE>
<% Call ViewDate %>
</BODY>
</HTML>
<SCRIPT LANGUAGE=VBScript RUNAT=Server>
Sub Callme
    Response.Write "<TR><TD>Call</TD><TD>Me</TD></TR>"
End Sub
</SCRIPT>
<SCRIPT LANGUAGE=JScript RUNAT=Server>
function ViewDate()
{
    var x
    x = new Date()
    Response.Write(x.toString())
}
```

```
}
</SCRIPT>
```

这是各位在本文中接触的第一个真正的 ASP 程序，千万不要被 “<% %>” 符号搞糊涂，这其实是标准的 ASP 定界符，而 “<SCRIPT></SCRIPT>” 之间的就是脚本语言。ASP 不同于脚本语言，它有自己的语法，所有的 ASP 命令都必须包含在 <% 和 %> 之内，如：<% test="English" %>，ASP 通过包含在 <% 和 %> 中的表达式将执行结果输出到客户浏览器，如：<% =test %> 就是将前面赋给变量 test 的值 English 发送到客户浏览器中，而当变量 test 的值为 Mathematics 时，以下程序：

```
This weekend we will test <% =test %>.
```

在客户浏览器中则显示为：

```
This weekend we will test Mathematics.
```

学习 ASP 最好的方法莫过于亲自动手编写，为了在最短的时间内使各位掌握 ASP 的程序设计技巧，本书将采取实例分析法，通过一系列的实例让各位在实践中学习 ASP。建立一个 ASP 页面，你所需的只是打开一个文本编辑器，如：Notepad，然后跟我开始动手编写第一个 ASP 程序。下面我们将建立一个自动监测浏览时间并根据不同时段动态显示不同页面内容的 ASP 程序，请将以下代码剪贴到你的文本编辑器中，并保存为 test1.ASP：

```
<html>
<body>
<FONT COLOR="Green">
<% If Time <#12:00:00# And Time >= #00:00:00# Then %>
早上好，今天天气不错！
<% ElseIf Time <#19:00:00# And Time >= #12:00:00# Then %>
下午好！
<% Else %>
Hello! 今晚你有没有去聊天！
<% End If %>
</body>
</html>
```

将 test1.ASP 保存在 WEB 服务器的虚拟目录（如：aspsamp/）下，并在浏览器中用 HTTP 的方式进行浏览，如：http://yourcomputername/aspsamp/test1.ASP，你会新奇地发现，你的页面真的活起来了。虽然，这只是一个非常简单的实例，而且这一功能完全可以通过 JavaScript 完成，但是不难发现使用 ASP 要比 JavaScript 简洁、迅速得多，而且运用此法，你完全可以轻而易举地令你的网页在不同的时段展示不同的风格。此例中的 “Time” 实际上是一个 VBScript 内置的显示系统当前时间的函数，由于系统默认的脚本语言是 VBScript，因此当你在 ASP 命令中调用该函数时，脚本引擎会自动将其转换成当前的系统时间。接下来我们将给 test1.ASP 添加一点色彩，在 <body> 标识中添加 “bgcolor=”<% =bgc %>” 即变为 <body bgcolor=”<% =bgc %>”>，并在 <body> 标记前添加如下语句：

```
<% If Time <#12:00:00# And Time >= #00:00:00# Then
bgc="silver"
```

```
ElseIf Time <#19:00:00# And Time >= #12:00:00# Then
bgc="navy"
Else
bgc="red"
End If
%>
```

如此一来，当用户在不同的时段访问你的页面时，他们将会看到不同的页面背景色。我们可以做的事情还有很多，譬如你想知道在凌晨至十二点之间浏览你页面的客户的姓名，并向他或她问好，那么下面的这段程序将能助你达成心愿。首先你需要在页面中设置表单，将以下 HTML 代码剪贴到 “<% If Time <#12:00:00# And Time >= #00:00:00# Then %>” 之后：

```
欢迎光临我的主页，请填写以下信息：<FORM METHOD="POST" ACTION="test1.ASP">
<P>
First Name: <INPUT NAME="fname" SIZE="48">
<P>
Last Name: <INPUT NAME="lname" SIZE="48">
<P>
Title: <INPUT NAME="title" TYPE=RADIO VALUE="mr">Mr.
<INPUT NAME="title" TYPE=RADIO VALUE="ms">Ms.
<P><INPUT TYPE=SUBMIT><INPUT TYPE=RESET>
</FORM>
```

然后在以上 HTML 代码后面添加如下 ASP 命令：

```
<%
title=request.form("title")
if title="mr" then
%>
欢迎您 Mr.<% =request.form("fname") %> 。
<% elseif title="ms" then %>
欢迎您 Ms.<% =request.form("fname") %> 。
<% else %>
<B><font color=blue>欢迎您 <% =request.form("fname")&" "&request.form("lname")
%> 。 </font></B>
<% end if %>
```

这其实是一个在 Internet 和 Intranet 上常见的功能，即当用户在浏览器端填写完表单后，通过调用一个通用网关程序将用户数据传送到服务器，由服务器进行处理后再将结果返还给客户浏览器。过去为了实现这样的功能必须编写一个独立于 HTML 之外的 CGI 程序，并通过 HTML 进行调用，撇开 CGI 编写复杂等缺点不谈，CGI 的执行效率也是一个大问题，每一个表单 (form) 均须执行一个可执行文档，当多人同时上线使用时，多个文档同时执行，将大大降低 WEB 服务器的执行速度，如今 ASP 提供了与 HTML 完全相融的编程环境，显然要比使用 CGI 便捷得多。

在本节的最后让我们再来看看 ASP 的循环功能，譬如你希望当客户在晚间 7:00 至凌晨 0:00:00 浏览你的页面时，在页面中央显示 6 个笑脸表示问候，那么你只需要将以下命令剪贴到“Good Evening!”后面即可：

```
<% for i=1 to 6 %>
<p><center>

</center></p>
<% next %>
```

这是一个最最基本的循环语句，它将一个笑脸图片重复调用六次并显示在页面上。当然此例的效果完全可以用 HTML 做到，但是不难发现使用 ASP 大大缩短了代码的重复编写，使得程序具有良好的可阅读性。另外，当你制作一个诸如根据用户投票来给所评测的对象评定星级的评测站点时，使用此法，你就完全不需要为每一个星级制作一幅图片。如果一个对象被评为 4 星级，那只需将一颗星的图片循环显示 4 次，以此类推即可。当然，当工作量较小的时候你不会感觉到 ASP 的好处，然而一旦数据量巨增，你将会深刻的体会到 ASP 动态网站给你带来的前所未有的轻、爽感受！

首先，要再次强调 ASP 是一套服务器端的脚本环境，当用户从浏览器向 Web 服务器请求 .ASP 文件时，ASP 脚本开始运行，然后 Web 服务器调用 ASP，ASP 全面读取请求的文件，执行所有脚本命令，并将 Web 页传送给浏览器，这就意味着你不能简单地在现有的操作系统（如 Win95、98）下，用浏览器中的“打开”命令来浏览 .ASP 文件；其次，运行 .ASP 文件需要服务器端的支持，目前除了 WindowsNT 下的 Microsoft Internet Information Server 支持 Active Server 技术外，Active Server 还可用于 MS Windows NT Workstation 和 MS Windows 95/98 中的 Personal Web Server 的构件，因此无论你安装的是 NT 还是 WIN95/98，都必须在原系统上安装支持 ASP 的服务器端环境，才能使你的机器得以运行 .ASP 文件。由于目前绝大多数朋友安装的是 Win95 或是 98，因此就在这里着重介绍如何在 Win95/98 下运行和浏览 .ASP 文件。因为 MS Personal Web Server 支持通用网关接口 (CGI) 应用程序、Internet 数据库接口 (IDC) 应用程序和 IIS 应用程序接口 (ISAPI) 应用程序，并且支持运行和开发 ASP 应用程序，所以使得 Win95 及其以后操作平台具备了近似于 NT 的健全的网站运作和管理功能。你所要做的只是在现有的 Win95/98 系统上安装 MS Personal Web Server，建议安装最新的 4.0 版本。当安装完成后，PWS4.0 会在系统启动时自动加载。打开 PWS，选择左栏中的“高级”按钮（见下图 Asp2a.gif），点取“虚拟目录”下的“/_private”目录，然后点击右边的“编辑属性”按钮，如图所示，在对话框中你可以编辑该虚拟目录的物理路径和目录名，上图中“/_private”的物理路径为“D:\Inetpub\wwwroot”，该目录是系统默认的 WWW 发布根目录，将上一篇中的范例 test1.ASP 文件复制到“D:\Inetpub\wwwroot”目录下。最后，你只要将该虚拟目录的访问权限选择为“执行”，并在浏览器的“地址”栏中输入“http://localhost/test1.ASP”就可以运行并浏览该 ASP 程序了。

知道了如何运行并浏览 ASP 程序，下面我们就继续 ASP 动态网站设计的学习。

1.2 ASP web 动态编程

上一节给大家介绍了几个最基本的 ASP 程序，相信大家已经对 ASP 的应用有了大致的了解，本节将继续介绍一些用 ASP 编写的 WEB 动态功能。由于 WEB 浏览器标准的不

一致从而使得如何能够让自己制作的网站去适应各种不同的浏览器成为了广大网站设计者最为头疼的事，在如今的形势之下，我们不可能抛弃 Netscape 或 IE 中的任何一种客户群，但我们有时候又不得不去考虑客户端浏览器的实际浏览效果，过去我们常用 JavaScript 编写一段程序来辨别客户端使用的不同的浏览器，那么今天就让我们来看看如何使用 ASP 更为便捷且精确地达到这一目的。将以下代码，剪贴到你的 Notebook 中，并保存为 browser.ASP。

```
<%@ LANGUAGE = VBScript %>
<HTML>
<HEAD>
    <TITLE>用 ASP 提供的浏览器性能组件来辨别客户浏览器</TITLE>
</HEAD>
<BODY BGCOLOR="White" topmargin="10" leftmargin="10">
    <font size="4" face="Arial, Helvetica">
    <b>用 ASP 提供的浏览器性能组件来辨<br>别客户浏览器</b></font><br>
    <hr size="1" color="#000000">
    <!-- 定义并创建浏览器性能对象 -->
    <%
        Dim bc
        Set bc = Server.CreateObject("MSWC.BrowserType")
    %>
    <!-- 在表格中描述客户浏览器的信息 -->
    <Table border=1>
        <tr>
            <td>Browser Name</td>
            <td> <%=bc.Browser %> </Td>
        <tr>
            <td>Browser Version</td>
            <td> <%=bc.Version %> </Td>
        <tr>
            <td>Major Version</td>
            <td> <%=bc.Majorver %> </Td>
        <tr>
            <td>Minor Version</td>
            <td> <%=bc.Minorver %> </Td>
        <tr>
            <td>Frame Support</td>
            <td> <%=bc.Frames %> </Td>
        <tr>
            <td>Table Support</td>
            <td> <%=bc.Tables %> </Td>
        <tr>
            <td>Cookie Support</td>
            <td> <%=bc.Cookies %> </Td>
```

```

        <tr>
        <td>Background Sound Support</td>
        <td> <%=bc.BackgroundSounds %> </Td>
        <tr>
        <td>VBScript Support</td>
        <td> <%=bc.VBScript %> </Td>
        <tr>
        <td>JavaScript Support</td>
        <td> <%=bc.JavaScript %> </Td>
    </table>
</BODY>
</HTML>

```

在本例中我们用到了一个 ASP 提供的 ActiveX 组件 --“Browser Capabilities”。ActiveX 组件作为基于 Web 的应用程序部分在 Web 服务器上运行，组件提供了应用程序的主要功能（如访问、修改数据库等），这样 WEB 设计者们就不必创建或重新创建执行这些任务的代码，从而提高了工作的效率，有关 ActiveX 组件将在今后详细讨论。为了使用 ASP 辨别客户浏览器，在这里我们利用“Browser Capabilities”组件创建了一个 Browser Type 对象，该对象提供带有客户端网络浏览器功能说明的用户脚本，当浏览器连接到网络服务器上时，它会自动发送一个 UserAgentHTTP 标题，该标题是一个声明浏览器及其版本的 ASCII 字符串。此 Browser Type 对象将该标题和在 Browscap.ini 文件中的项进行比较（Win98、IIS3、IIS4 的用户分别可以在以下目录“win98\system\inetsrv”、“\winnt\system32\inetsrv\ASP\cmpnts”、“\winnt\system32\inetsrv”中找到该文件）。如果找到匹配的项，则该 Browser Type 对象将认为浏览器列表属性与 UserAgent 标题匹配。若该对象在 browscap.ini 文件中找不到与该标题匹配的项，那么将使用默认的浏览器属性。若该对象既未找到匹配项且 browscap.ini 文件中也未指定默认的浏览器设置，则它将每个属性都设为字符串“UNKNOWN”。我们可以通过更新 browscap.ini 文件将属性或新的浏览器定义添加到该组件中，从而扩大 ASP 识别的浏览器范围和精确度。下面列出了 browscap.ini 中的一部分内容（不包括注释部分）：

```

[IE 4.0] ;;HTTPUserAgentHeader
browser=IE ;;指定该浏览器的名称。
Version=4.0 ;;指定该浏览器的版本号。
majorver=4 ;;指定主版本号
minorver=0 ;;指定副版本号
frames=TRUE ;;指定该浏览器是否支持框架。
tables=TRUE ;;指定该浏览器是否支持表格。
cookies=TRUE ;;指定该浏览器是否支持 cookies。
backgroundsounds=TRUE ;;指定该浏览器是否支持背景音乐。
vbscript=TRUE ;;指定该浏览器是否支持 VBScript。
javascript=TRUE ;;指定该浏览器是否支持 JScript。
javaapplets=TRUE ;;指定该浏览器是否支持 Java 程序。
ActiveXControls=TRUE ;;指定该浏览器是否支持 ActiveX 控件。

```

```
Win16=False ;; 指定该浏览器是否支持 Win16
beta=False ;;指定该浏览器是否测试版。
cdf=True ;;指定该浏览器是否 支持用于 Web 预测的 Channel Definition Format。
;;ie 4.01
[Mozilla/4.0 (compatible; MSIE 4.01*; Windows 95)]
parent=IE 4.0 ;;父标签允许第二个浏览器继承第一个浏览器的定义
version=4.01
minorver=01
platform=Win98
;;Default Browser ;;指定默认的浏览器的设置
[Default Browser Capability Settings]
browser=Default
frames=FALSE
tables=TRUE
cookies=FALSE
backgroundsounds=FALSE
vbscript=FALSE
javascript=FALSE
```

在上面的例子中，父标签允许第二个浏览器继承第一个浏览器的定义，以便 Microsoft Internet Explorer4.01 定义得以继承 Microsoft Internet Explorer4.0 定义全部的属性（例如，frames=TRUE、tables=TRUE 以及 cookies=TRUE）。并通过添加 platform=Win98 行来指定平台，用 version=4.01 重写版本信息。

在先前的 browser.ASP 中我们仅仅是将客户浏览器的属性一一罗列了出来，下面让我们添加一些动态的效果。将以下代码剪贴到文件 browser.ASP 中：

```
<% if (bc.frames = TRUE) then %>
你的浏览器支持框架<br>
<% else %>
你的浏览器不支持框架<br>
<% end if %>
<% if (bc.tables = TRUE) then %>
你的浏览器支持表格<br>
<% else %>
你的浏览器不支持表格<br>
<% end if %>
<% if (bc.BackgroundSounds = TRUE) then %>
有没有听到音乐？<br>
<% else %>
您的浏览器不支持背景音乐<br>
<% end if %>
<% if (bc.vbscript = TRUE) then %>
您的浏览器支持 Vbscript。<br>
<% else %>
```

```
您的浏览器不支持 Vbscript。<br>
<% end if %>
<% if (bc.javascript = TRUE) then %>
您的浏览器支持 Javascript。<br>
<% else %>
您的浏览器不支持 Javascript。<br>
<% end if %>
```

在你的浏览器中刷新 browser.ASP，程序将自动辨别浏览器的属性并动态显示不同的信息。不难发现在整个 browser.ASP 文件中几乎没有用到任何复杂的编程，就轻而易举的达到了动态鉴别客户浏览器并动态产生响应事件的效果。事实上，这个程序的关键就在于前文提到的 ActiveX 组件——“Browser Capabilities”，它的作用类似于一个 Function，只需简单地在程序中调用该组件即能达到你所希望的效果。

通过这两节对 ASP 的介绍，大家已经看到编写 ASP 程序是相当轻松的，要掌握 ASP，无非就是掌握 ASP 内置的五个对象，以及 ASP 提供的 ActiveX 组件的对象、方法和属性，当然扎实的脚本语言编写能力也是必须的。脚本语言将在下一章中为大家介绍

第二章 ASP 脚本语言

2.1 JavaScript 语言概况

怎样把自己的或公司的信息资源加入到 WWW 服务器,是广大用户日益关心的问题。采用超链技术(超文本和超媒体技术)是实现这个目标最简单的、最快速的手段和途径。具体实现这种手段的支持环境,那就是 HTML 超文本标识语言。通过它们可制作所需的 Web 网页。

通过超文本(Hyper Text)和超媒体(Hyper Media)技术结合超链接(Hyper link)的链接功能将各种信息组织成网络结构(web),构成网络文档(Document),实现 Internet 上的“漫游”。通过 HTML 符号的描述就可以实现文字、表格、声音、图像、动画等多媒体信息的检索。

然而采用这种超链技术存在有一定的缺陷,那就是它只能提供一种静态的信息资源,缺少动态的客户端与服务器端的交互。虽然可通过 CGI (Common Gateway Interface)通用网关接口实现一定的交互,但由于该方法编程较为复杂,因而在一段时间妨碍了 Internet 技术的发展。而 JavaScript 的出现,无疑为 Internet 网上用户带来了一线生机。可以这样说,JavaScript 的出现是时代的需求,是当今的信息时代造就了 JavaScript。

JavaScript 的出现,它可以使得信息和用户之间不仅只是一种显示和浏览的关系,而是实现了一种实时的、动态的、可交式的表达能力。从而基于 CGI 静态的 HTML 页面将被可提供动态实时信息,并对客户操作进行反应的 Web 页面的取代。JavaScript 脚本正是满足这种需求而产生的语言。它深受广大用户的喜爱的欢迎。它是众多脚本语言中较为优秀的一种,它与 WWW 的结合有效地实现了网络计算和网络计算机的蓝图。无疑 Java 家族将占领 Internet 网络的主导地位。因此,尽快掌握 JavaScript 脚本语言编程方法是我国广大用户日益关心的。

2.1.1 什么是 JavaScript

JavaScript 是一种基于对象(Object)和事件驱动(Event Driven)并具有安全性能的脚本语言。使用它的目的是与 HTML 超文本标记语言、Java 脚本语言(Java 小程序)一起实现在一个 Web 页面中链接多个对象,与 Web 客户交互作用。从而可以开发客户端的应用程序等。它是通过嵌入或调入在标准的 HTML 语言中实现的。它的出现弥补了 HTML 语言的缺陷,它是 Java 与 HTML 折衷的选择,具有以下几个基本特点:

(1) 是一种脚本编写语言

JavaScript 是一种脚本语言,它采用小程序段的方式实现编程。像其它脚本语言一样,JavaScript 同样已是一种解释性语言,它提供了一个易开发过程。

它的基本结构形式与 C、C++、VB、Delphi 十分类似。但它不像这些语言一样,需要先编译,而是在程序运行过程中被逐行地解释。它与 HTML 标识结合在一起,从而方便用户的使用操作。

(2) 基于对象的语言

JavaScript 是一种基于对象的语言,同时还可以看作一种面向对象的。这意味着它能运用自己已经创建的对象。因此,许多功能可以来自于脚本环境中对象的方法与脚本的相互作用。

(3) 简单性