

计算机网络技术系列教材

ASP网络应用程序设计 (第2版)

高怡新 编著



人民邮电出版社
POSTS & TELECOM PRESS

TP393.092/932

2008

计算机网络技术系列教材

ASP 网络应用程序设计

(第2版)

高怡新 编著

人民邮电出版社

北京

图书在版编目 (CIP) 数据

ASP 网络应用程序设计 / 高怡新编著. —2 版. 北京: 人民邮电出版社, 2008.4
(计算机网络技术系列教材)
ISBN 978-7-115-17528-1

I. A… II. 高… III. 主页制作—程序设计—教材
IV. TP393.092

中国版本图书馆 CIP 数据核字 (2008) 第 009639 号

内 容 提 要

本书从实际应用出发, 系统地介绍了使用 ASP 技术进行网络应用程序设计与网站开发的基本知识与编程实践。除了讲述 HTML、VBScript 语言、ASP 内置对象与组件、ADO 数据库技术之外, 本书还较为详细地介绍了在网页设计中极为有用的 JavaScript 语言以及 CSS 技术和 DOM 技术, 并在讲清各个知识点及其联系的基础上, 提供了相当丰富的范例。读者对照书中的讲解与实例进行演练, 并在其基础上加以创新, 可以开发出各种实用的 Web 应用程序。

本书可作为高等院校、高职高专信息技术和电子商务等专业的教学用书, 也可作为网络技术人员培训教材。

计算机网络技术系列教材

ASP 网络应用程序设计 (第 2 版)

- ◆ 编 著 高怡新
责任编辑 刘雁斌
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
新华书店总店北京发行所经销
- ◆ 开本: 787×1092 1/16
印张: 19
字数: 452 千字 2008 年 4 月第 2 版
印数: 19 001—22 000 册 2008 年 4 月第 1 次印刷

ISBN 978-7-115-17528-1/TP

定价: 28.50 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前 言

ASP 技术无疑是微软公司最为成熟、应用最为广泛的 Web 应用程序开发技术。多年来,有关 ASP 技术及其应用的课程已经成为众多高等院校相关专业的必修课程。

本书自 2005 年 3 月出版以来,受到了许多院校师生的欢迎。作者结合近几年的教学实践和读者的反馈意见,在保留原书特色的基础上,对其进行了全面修订。这次修订的主要工作如下。

- 对本书第一版的多个章节进行了调整与完善,使各章内容的编排更显合理、概念更为准确、叙述更加到位,并新增了不少范例。
- 在原有知识结构的基础上,本书更为系统地介绍了 JavaScript 语言、CSS 技术和 DOM 技术,目的是使读者更为全面地了解和掌握当今有关网页设计和网络应用程序开发的各种知识,并使之更加贴近实际应用的需要。
- 在本书最后添加“ASP 网上书城实例”一章,通过对该实例网站各个主要功能页面设计的详细讲解和探讨,读者能够进一步掌握此类网络应用程序的设计理念与编程技巧,并初步具备开发一个实用商务网站的能力。

修订后,本书从介绍 ASP 技术框架开始,按照 HTML 标记语言、VBScript 和 JavaScript 脚本语言、CSS 技术和 DOM 技术、ASP 内置对象和组件、ADO 对象和 Web 数据库应用的顺序,全面介绍了采用 ASP 技术开发网络应用程序所需的各种知识。书中每个知识点的讲解都配有相应的范例,且所有范例都是可以实际运行的。

本书的最大特点是概念准确、讲述清楚、实例丰富。第 8 章介绍了多个 ASP 应用程序开发实例,包括网上投票系统、在线考试系统、学生成绩查询系统、客户登录系统和网上留言簿等。这些实例中的许多代码都可以直接在实际开发工作中应用。通过对这些实例的讲解剖析,能够使读者进一步消化与贯通前面各章的知识,逐步练就编写各种 Web 应用程序的能力。第 9 章介绍了一个较为完整的商务网站实例——网上书城,详细讲述了该网站各个主要功能页面的开发过程并给出了完整的源代码,其中不乏宝贵的开发经验与技巧。读者若能仔细研读这一章,必然收获良多。

本书可作为高等院校、高职高专信息技术和电子商务等专业的教学用书,也可作为网络技术人员的培训教材或网络应用开发人员的参考书。读者对照书中的讲解与实例进行演练,并在其基础上加以创新,可以开发出各种实用的 Web 应用程序。

本书在编写过程中,得到了申蔚、王之源、高丰、张玮等多位老师的参与和帮助,在此一并表示诚挚的谢意!书中错误与不妥之处,恳请读者批评指正。

作 者
2008 年 1 月

目 录

第1章 ASP 概述	1
1.1 静态网页与动态网页	1
1.1.1 静态 HTML 网页	1
1.1.2 客户端动态网页	2
1.1.3 服务器端动态网页	3
1.2 ASP 简介	4
1.2.1 ASP 的特点	4
1.2.2 ASP 工作过程	5
1.2.3 ASP 组成要素	6
1.3 ASP 运行环境	6
1.3.1 ASP 运行的软硬件环境	7
1.3.2 IIS 简介	7
1.3.3 IIS 的安装	8
1.4 IIS 网站管理	10
1.4.1 IIS 默认网站	10
1.4.2 IIS 的启动与停止	11
1.4.3 IIS 网站属性的设置	12
1.4.4 IIS 虚拟目录的创建	13
1.5 ASP 文档的创建与运行	16
1.5.1 ASP 与脚本语言	16
1.5.2 ASP 文档的创建	16
1.5.3 ASP 文档的运行	17
1.5.4 ASP 文档编写工具	18
习题	19
第2章 HTML 网页设计	20
2.1 HTML 概述	20
2.1.1 什么是 HTML	20
2.1.2 HTML 文档基本架构	21
2.2 HTML 常用标记	22
2.2.1 头部标记	22
2.2.2 主体标记	23
2.2.3 段落标记	25
2.2.4 文字格式标记	27
2.2.5 特殊字符标记	29

2.2.6	列表标记	29
2.2.7	图像标记	31
2.2.8	超链接标记	32
2.3	HTML 表格标记	33
2.3.1	表格定义格式	33
2.3.2	表格应用举例	34
2.4	HTML 框架标记	37
2.4.1	框架定义格式	37
2.4.2	框架应用举例	38
2.4.3	框架嵌套举例	40
2.5	HTML 表单标记	41
2.5.1	表单定义格式	41
2.5.2	常用表单域标记	42
2.5.3	表单应用举例	44
2.6	HTML 其他标记	46
2.6.1	动态文字标记	46
2.6.2	层标记	47
2.6.3	嵌入多媒体信息标记	49
2.7	HTML 文档样式	49
2.7.1	CSS 简介	50
2.7.2	在文档头部定义样式	52
2.7.3	为单个元素定义样式	53
2.7.4	定义与引用外部样式表	55
习题		56
第3章	VBScript 脚本语言	58
3.1	VBScript 概述	58
3.1.1	什么是 VBScript	58
3.1.2	用 VBScript 开发客户端脚本	58
3.1.3	用 VBScript 开发服务器端脚本	59
3.1.4	VBScript 语句书写规则	61
3.2	VBScript 语法基础	62
3.2.1	VBScript 数据类型	62
3.2.2	VBScript 常量	63
3.2.3	VBScript 变量	63
3.2.4	VBScript 运算符	65
3.3	VBScript 输入与输出	66
3.3.1	输入函数 InputBox()	66
3.3.2	输出函数 MsgBox()	67
3.3.3	输出语句 MsgBox	69

3.3.4 利用 Document.Write 输出	70
3.4 VBScript 内置函数	70
3.4.1 日期时间函数	70
3.4.2 数学运算函数	72
3.4.3 字符处理函数	73
3.4.4 数据类型判别函数	74
3.4.5 数据类型转换函数	75
3.5 VBScript 流程控制	76
3.5.1 If...Then...Else 语句	76
3.5.2 Select...Case 语句	79
3.5.3 DO...Loop 循环	80
3.5.4 While...Wend 循环	82
3.5.5 For...Next 循环	83
3.5.6 For Each In...Next 循环	84
3.5.7 循环语句的嵌套	85
3.5.8 用 Exit 退出循环	86
3.6 VBScript 过程	87
3.6.1 Sub 过程	87
3.6.2 Function 过程	88
3.6.3 VBScript 过程的调用	89
3.7 VBScript 与 DOM	91
3.7.1 DOM 概述	91
3.7.2 DOM 事件	92
3.7.3 DOM 事件过程的调用	93
3.7.4 Window 对象	95
3.7.5 Document 对象	96
3.7.6 Form 对象	98
3.7.7 History 对象	99
习题	100
第 4 章 JavaScript 脚本语言	102
4.1 JavaScript 概述	102
4.2 JavaScript 语法概述	103
4.2.1 JavaScript 语法基础	103
4.2.2 JavaScript 变量	104
4.2.3 JavaScript 运算符	105
4.3 JavaScript 流程控制	106
4.3.1 If 分支语句	106
4.3.2 Switch 分支语句	107
4.3.3 For 循环语句	108

4.3.4	While 循环语句	109
4.3.5	Do-While 循环语句	109
4.3.6	break、continue 语句	109
4.4	JavaScript 对象	110
4.4.1	Array 对象	110
4.4.2	Date 对象	113
4.4.3	Math 对象	114
4.5	JavaScript 函数	114
4.6	JavaScript 与 DOM	116
4.6.1	DOM 技术编程示例	116
4.6.2	DOM 与表单验证	119
4.7	JavaScript 与 DHTML	123
4.7.1	DHTML 简介	123
4.7.2	DHTML 举例	124
	习题	127
第 5 章	ASP 内置对象	129
5.1	ASP 内置对象简介	129
5.2	Response 对象	130
5.2.1	Response 对象概述	130
5.2.2	Response.Write 方法	131
5.2.3	Response.Redirect 方法	132
5.2.4	Response 对象的其他方法	134
5.2.5	Response 对象的属性	134
5.2.6	Response.Cookies 集合	136
5.3	Request 对象	137
5.3.1	Request 对象的属性与方法	137
5.3.2	Request.Form 集合	138
5.3.3	Request.QueryString 集合	141
5.3.4	Request.ServerVariables 集合	143
5.3.5	Request.Cookies 集合	145
5.4	Server 对象	146
5.4.1	Server.CreateObject 方法	146
5.4.2	Server.MapPath 方法	147
5.4.3	Server.Execute 方法	147
5.4.4	Server.Transfer 方法	148
5.4.5	Server.HtmlEncode 方法	149
5.4.6	Server.UrlEncode 方法	149
5.5	Application 对象	150
5.5.1	Application 对象概述	150

5.5.2	Application 对象的集合	150
5.5.3	Application 对象的事件	152
5.5.4	Application 对象应用举例	153
5.6	Session 对象	154
5.6.1	Session 对象概述	154
5.6.2	Session 对象的集合	155
5.6.3	Session 对象的事件	156
5.6.4	Session 对象应用举例	157
5.7	Global.asa 文件	159
5.7.1	Global.asa 文件概述	159
5.7.2	Global.asa 文件举例	160
	习题	161
第 6 章	ASP 内置组件	163
6.1	ASP 组件概述	163
6.1.1	内置组件与外置组件	163
6.1.2	ASP 常用内置组件	164
6.2	File Access 组件	164
6.2.1	File Access 组件概述	165
6.2.2	文件夹操作	165
6.2.3	文件的创建、复制与删除	169
6.2.4	文件内容的写入	171
6.2.5	文件内容的读取	172
6.3	Ad Rotator 组件	174
6.3.1	Ad Rotator 组件简介	174
6.3.2	Ad Rotator 组件应用示例	175
6.4	PageCounter 组件	176
6.4.1	PageCounter 组件简介	176
6.4.2	PageCounter 组件应用示例	177
6.5	ContentLinking 组件	178
6.5.1	ContentLinking 组件简介	178
6.5.2	ContentLinking 组件应用示例	179
	习题	180
第 7 章	ADO 与 Web 数据库	181
7.1	Web 数据库概述	181
7.1.1	Web 数据库及其优势	181
7.1.2	访问 Web 数据库的几种技术	182
7.2	ODBC 数据库连接技术	183
7.2.1	ODBC 简介	184
7.2.2	ODBC 数据源管理器	184

7.2.3 DSN 的创建与设置	185
7.3 ADO 数据库访问技术	186
7.3.1 ADO 技术概述	187
7.3.2 ADO 组件对象简介	187
7.3.3 ADO 对象之间的关系	188
7.4 Connection 对象	189
7.4.1 Connection 对象的常用方法	189
7.4.2 Connection 对象的事务操作	190
7.4.3 ConnectionString 连接字符串	191
7.4.4 Connection 对象的属性	192
7.5 Command 对象	193
7.5.1 Command 对象概述	193
7.5.2 Command 对象的属性与方法	194
7.6 Recordset 对象	195
7.6.1 Recordset.Open 方法	195
7.6.2 Recordset 对象的其他方法	196
7.6.3 Recordset 对象的属性	196
7.6.4 Field 对象与 Fields 集合	197
7.7 SQL 查询语言	199
7.7.1 SQL 语言概述	200
7.7.2 数据查询命令	200
7.7.3 插入记录命令	202
7.7.4 更新数据命令	203
7.7.5 删除记录命令	203
7.8 Web 数据库访问实例	204
7.8.1 连接数据库	204
7.8.2 读取并输出数据	206
7.8.3 查询数据	208
7.8.4 添加记录	210
7.8.5 更新数据	212
7.8.6 删除记录	214
习题	215
第8章 ASP 应用程序设计实例	217
8.1 百年日历	217
8.1.1 设计思路	217
8.1.2 百年日历完整代码	219
8.2 网上投票系统	222
8.2.1 设计思路	222
8.2.2 网上投票系统源代码	223

8.2.3 网上投票系统的完善	225
8.2.4 用图形显示投票结果	226
8.3 在线考试系统	228
8.3.1 设计思路	228
8.3.2 数据库设计	229
8.3.3 在线考试系统源代码	230
8.3.4 在线考试系统的完善	233
8.4 成绩查询系统	233
8.4.1 设计思路	233
8.4.2 数据库设计	234
8.4.3 首页框架设计	235
8.4.4 上方页面设计	235
8.4.5 下方页面设计	237
8.5 注册登录系统	239
8.5.1 设计思路	239
8.5.2 数据库设计	239
8.5.3 Global.asa 文件	240
8.5.4 登录页面设计	241
8.5.5 登录信息验证页面设计	242
8.5.6 注册页面设计	243
8.5.7 保存注册信息页面设计	246
8.6 访客留言簿	248
8.6.1 设计思路	248
8.6.2 数据库设计	250
8.6.3 留言簿主页面设计	250
8.6.4 撰写留言页面设计	253
8.6.5 保存留言页面设计	255
习题	256
第9章 ASP 网上书城实例	257
9.1 网上书城总体设计	257
9.1.1 总体设计思路	257
9.1.2 网站数据库设计	258
9.1.3 Global.asa 文件	260
9.2 包含文件设计	261
9.2.1 包含文件的概念	261
9.2.2 head.inc 文件设计	262
9.2.3 foot.inc 文件设计	263
9.2.4 style.css 文件设计	264
9.3 网上书城首页设计	265

9.3.1	首页设计思路	265
9.3.2	图书展示功能设计	266
9.3.3	图书搜索功能设计	267
9.3.4	完整的首页源代码	268
9.3.5	图书详情页面设计	271
9.4	购物车页面设计	274
9.4.1	购物车设计思路	274
9.4.2	放入图书功能设计	275
9.4.3	删除图书功能设计	276
9.4.4	更改数量与计算书款金额	276
9.4.5	完整的购物车页面源代码	278
9.5	订单页面设计	280
9.5.1	订单页面设计思路	280
9.5.2	填写订单页面设计	281
9.5.3	保存订单数据功能设计	284
9.5.4	查看订单信息页面设计	285
9.6	留言板	288
9.7	留言板	290
9.8	留言板	292
9.9	留言板	294
9.10	留言板	296
9.11	留言板	298
9.12	留言板	300
9.13	留言板	302
9.14	留言板	304
9.15	留言板	306
9.16	留言板	308
9.17	留言板	310
9.18	留言板	312
9.19	留言板	314
9.20	留言板	316
9.21	留言板	318
9.22	留言板	320
9.23	留言板	322
9.24	留言板	324
9.25	留言板	326
9.26	留言板	328
9.27	留言板	330
9.28	留言板	332
9.29	留言板	334
9.30	留言板	336
9.31	留言板	338
9.32	留言板	340
9.33	留言板	342
9.34	留言板	344
9.35	留言板	346
9.36	留言板	348
9.37	留言板	350
9.38	留言板	352
9.39	留言板	354
9.40	留言板	356
9.41	留言板	358
9.42	留言板	360
9.43	留言板	362
9.44	留言板	364
9.45	留言板	366
9.46	留言板	368
9.47	留言板	370
9.48	留言板	372
9.49	留言板	374
9.50	留言板	376
9.51	留言板	378
9.52	留言板	380
9.53	留言板	382
9.54	留言板	384
9.55	留言板	386
9.56	留言板	388
9.57	留言板	390
9.58	留言板	392
9.59	留言板	394
9.60	留言板	396
9.61	留言板	398
9.62	留言板	400
9.63	留言板	402
9.64	留言板	404
9.65	留言板	406
9.66	留言板	408
9.67	留言板	410
9.68	留言板	412
9.69	留言板	414
9.70	留言板	416
9.71	留言板	418
9.72	留言板	420
9.73	留言板	422
9.74	留言板	424
9.75	留言板	426
9.76	留言板	428
9.77	留言板	430
9.78	留言板	432
9.79	留言板	434
9.80	留言板	436
9.81	留言板	438
9.82	留言板	440
9.83	留言板	442
9.84	留言板	444
9.85	留言板	446
9.86	留言板	448
9.87	留言板	450
9.88	留言板	452
9.89	留言板	454
9.90	留言板	456
9.91	留言板	458
9.92	留言板	460
9.93	留言板	462
9.94	留言板	464
9.95	留言板	466
9.96	留言板	468
9.97	留言板	470
9.98	留言板	472
9.99	留言板	474
9.100	留言板	476

第 1 章 ASP 概述

ASP 的全称为 Active Server Pages，是用来创建在 Web 服务器端运行的动态网页技术。本章首先介绍静态网页与动态网页的概念，然后说明 ASP 的工作过程及其运行环境的建立，最后给出一个创建与运行简单 ASP 网页的实例。

1.1 静态网页与动态网页

在 Internet 风行的早期，浏览器中显示的网页通常由基本的文字、图片和超链接组成，网页的内容在浏览器中打开后不会发生更多的变化，浏览者可以在网页上阅读信息，但无法进一步地发表意见、查询信息或进行在线交易等活动，这样的网页被称为静态网页。随着 Internet 及其应用的飞速发展，目前多数网站的网页不仅内容丰富多彩，而且可以与用户的操作进行互动、发生动态变化，此种网页被称为动态网页。

事实上，存在着两种不同意义上的动态网页：一种是在客户端浏览器中执行代码而产生动态显示效果的网页；另一种则是在 Web 服务器端执行代码而实现与客户端动态交互的网页。当然，无论哪一种网页都是存放在网站的 Web 服务器中，并最终呈现在客户端的浏览器窗口中。

1.1.1 静态 HTML 网页

静态网页用单纯的 HTML 标记语言进行编写，并以 .htm 或 .html 为文件扩展名进行保存的 HTML 文档。静态网页中可以包含文字、图片，以及能够跳转到其他网页或文档的超链接等。此种网页中的所有内容都用特定的 HTML 标记进行标注，仅用来控制网页内容的布局及各个网页元素的静态显示方式。

【例 1.1】简单静态 HTML 网页示例。

下面的示例说明了一个简单的静态 HTML 网页是如何实现的。启动 Windows 系统的“记事本”程序，在打开的窗口中输入以下代码。

```
<!-- File Name: 世界你好.htm -->
<html>
<head>
  <title>
    简单静态网页示例
  </title>
</head>
<body>
  <p align="center"><font color="blue">Hello World!</font></p>
```

```
<p align="center"><font face="黑体">世界你好! </font></p>
</body>
</html>
```

上述代码中用尖括号括起来的特定字符即是各种 HTML 标记, 其中的 `<html>` 标记表明这是一个 HTML 文档; `<body>` 标记表明这是网页的主体部分; `<font>` 标记则专门用来设定文字的字体、字号与颜色等, 该标记在这里将“Hello World!”文字标注为用蓝色显示、将“世界你好!”文字标注为用黑体显示。有关各种 HTML 标记的格式及其具体应用, 将在下一章详细介绍。

将上述内容以 .htm 或 .html 为文件扩展名保存, 用鼠标双击此文件的图标或文件名, 即可在打开的 IE 浏览器窗口中看到如图 1-1 所示的网页内容。

此种单纯使用 HTML 标记语言编写的网页所显示的内容是不会改变的, 因而是静态的。若在浏览器窗口的“查看”菜单下选择“源文件”命令, 则可在打开的“记事本”窗口中看到与上面所输入的 HTML 代码完全一样的内容。



图 1-1 简单 HTML 静态网页示例

1.1.2 客户端动态网页

目前在网页中除了可以添加 GIF 动画和 Flash 动画产生动态特效外, 大多数网页还广泛采用 DHTML (Dynamic HTML) 技术使网页动起来。DHTML 是一种在网页文档下载到浏览器后仍然能够控制网页中的 HTML 元素, 使其变换表现效果的技术。

例如, 在访问 Microsoft 公司的站点时, 当浏览者将鼠标指针移动到页面的导航条上时, 会动态地弹出一个下拉菜单; 当在该菜单中移动鼠标指针时, 所指向的菜单项会用另一种颜色或形状表示, 如果将鼠标指针移出所显示的菜单范围, 则该菜单会自动隐藏起来。再如, 浏览者在某些网站购物时, 常可见到四处飘零的广告, 以及随着滚动条的移动而始终处于窗口同一位置的购物车图标等。

DHTML 技术的实现并不是孤立的, 它通常需要 HTML 标记与客户端脚本语言、CSS 层叠样式表以及网页的“层”布局技术组合在一起使用。此种网页中的一些 HTML 元素是可以被动态操纵的, 一些元素的 CSS 显示样式以及所在“层”面的显示属性也是能够被动态操纵的, 因此可以获得这些元素的动态显示效果。JavaScript 或 VBScript 等客户端脚本语言用来编写进行控制的脚本程序代码, 以实现动态操纵网页中的 HTML 元素及其 CSS 样式或者所在“层”面的显示属性。

【例 1.2】客户端动态网页示例。

下面的示例说明了一个简单的动态特效网页是如何实现的。启动 Windows 系统的“记事本”程序, 在打开的窗口中输入以下代码。

```
<!-- File Name: hello.htm -->
<html>
<head>
<script language="JavaScript">
```

```

function getname(str) {alert("您好! "+str+"!");}
</script>
</head>
<body>
  请输入您的姓名:
  <form>
    <input type="text" name="name" onBlur="getname(this.value)" value="">
  </form>
</body>
</html>

```

上述代码中除了普通 HTML 标记外,还包括了用 JavaScript 编写的脚本程序代码,这些代码规定了当网页中的文本框失去焦点后,将弹出一个显示相关内容的对话框。有关 JavaScript 脚本语言与 DHTML 技术,将在本书第 4 章展开讨论。

将上述内容以文件名“hello.htm”保存后,用鼠标双击此文件的图标或文件名,在打开的 IE 浏览器窗口中将显示“请输入您的姓名:”提示信息及一个文本框。在该文本框中输入某个姓名后再将鼠标光标移出文本框外单击,即可弹出一个对话框并显示出与输入姓名相关的问候语,其效果如图 1-2 所示。



图 1-2 简单客户端动态网页示例

此种动态特效网页的特点是:操纵网页动态变化的程序代码是在客户端浏览器中执行的,而且在执行期间与站点的 Web 服务器无关。

1.1.3 服务器端动态网页

这里所说的“服务器端动态网页”是指在 Web 服务器端执行,并实现与客户端浏览器进行交互和动态数据传递的网页,服务器端可以实时处理客户端浏览器的请求 (Request),然后再将处理的结果作为对请求的响应 (Response) 传送给客户端浏览器。

目前,几乎所有具有一定规模的网站都采用此种动态交互网页来为用户提供留言、信息查询和电子邮件等实时服务。电子商务和电子政务网站更是离不开此种动态交互网页来进行网上交易、信息检索和在线公务处理等活动。

最早的动态交互网页是采用 CGI (Common Gateway Interface, 公共网关接口) 技术实现的,CGI 的功能强大、技术成熟,但是编程困难且运行效率低下,因而未能得到较好的发展。目前大量采用 ASP、JSP 或 PHP 等动态交互网页技术,这些技术不仅具有良好的可编程性,而且不需编译即可直接运行,极大地提高了 Web 服务器的性能和网络应用程序的运行效率,是目前网站建设的主流技术。

ASP (Active Server Pages) 是微软公司开发的一套动态交互网页技术,同时也是一种服务器端脚本程序开发工具和运行环境,可用来轻松地创建动态、交互、高性能的 Web 应用程序。PHP 是一种跨平台的服务器端嵌入式脚本语言,它大量借用 C、Java 和 Perl 语言的语法,并配合 PHP 自己的特性,使 Web 应用程序开发者能够快速写出可与服务器交互的动态网页。PHP 最大的优势在于:它是完全免费的,有关软件可在 PHP

官方站点 (www.php.net) 自由下载。JSP (Java Server Pages) 是 Sun 公司推出的新一代网站开发语言, JSP 的优势在于可以在 Servlet 和 JavaBean 的支持下创建功能强大的 Web 应用程序。

需要指出的是, 创建动态、交互的 Web 应用程序在许多情况下离不开 Web 数据库的支持。目前的做法是, 使网页与网站后台的 Web 数据库建立一定的联系, 当 Web 数据库中的信息改变时, 用户接收到的信息便会相应地发生变化, 从而使得网页的内容实现动态变化。时至今日, 在创建各种网站时, 实现网页内容的动态变化已极为必要。用户通过浏览器发出请求, 实现与 Web 服务器连接并动态访问网站的 Web 数据库, 不仅可以使网页内容及时得到自动更新, 并且利用这种机制可以开发出电子商务、电子政务、在线娱乐游戏等各种丰富多彩的网络应用。

1.2 ASP 简介

1.2.1 ASP 的特点

ASP (Active Server Pages) 是微软提供的一套 Web 服务器端脚本程序开发工具和运行环境, 可用来创建和运行动态交互的 Web 服务应用程序。有了 ASP 就不必担心客户端的浏览器能否运行所编写的代码, 因为所有的程序都将在服务器端执行。当程序执行完毕后, 服务器仅将所执行的结果返回给客户端的浏览器, 这样就减轻了客户端浏览器的负担, 极大地提高了 Web 应用的交互速度。

使用 ASP 编写服务器端脚本代码, 可以方便地创建复杂、实用的 Web 应用程序。如果希望将访问者在网页表单中填写的信息存储到站点的数据库中, 或者希望根据访问者的选择向客户端浏览器输出不同内容的网页, 或者能够对不同的浏览器使用不同的 HTML 功能, 都将发现 ASP 是一个非常优秀的解决方案。例如, 以前在 Web 服务器上处理访问者的输入时, Web 应用程序开发者必须首先学习用 Perl 或 C 语言等建立传统的 CGI (公共网关接口) 应用程序。而使用 ASP 后, 只需在 HTML 文档中直接嵌入简单的服务器端脚本代码, 便可以收集 HTML 表单信息并传递到站点数据库。

与使用 Visual Basic、C++ 或 Java 等编程语言开发 Web 应用程序相比, ASP 是一个能够灵活快速地创建 Web 应用程序的解决方案。除了通过添加脚本为应用程序创建诱人的 HTML 界面之外, 还可以建立自己的 COM (组件对象模型) 组件, 并且可将应用程序的商业逻辑封装在可重复使用的模块中, 以便在脚本、其他组件或其他应用程序中调用。

ASP 具有如下一些特点。

- 使用 VBScript 或者 JavaScript 等简单易懂的脚本语言, 结合 HTML 代码, 即可快速方便地创建大多数 Web 应用程序。
- 使用简单的文本编辑器 (如 Windows 系统的“记事本”) 即可创建和编辑 ASP 程序, 并且 ASP 程序无需编译, 即可在服务器端直接执行。
- ASP 提供了一些内置的对象, 可用来从客户端的浏览器接收信息或将服务器处理后的响应信息发送给客户端的浏览器。

- ASP 提供了一些标准的 ActiveX 组件，同时允许用户添加或创建属于自己的 ActiveX 组件。这些组件允许依据客户端浏览器的能力进行不同的显示，并可在浏览器端包含计数器等。
- ASP 提供了与后台数据库连接和访问的功能，允许站点访问者通过客户端浏览器对站点的数据库进行访问，并且可以使得动态网页的内容随着相关数据库内容的变更而自动更新。
- ASP 程序的源代码不会被传送到客户端的浏览器，因而可以避免站点设计者所编写的源程序被他人剽窃，同时也提高了程序的安全性。
- ASP 可使用服务器端的脚本程序来自动生成客户端的脚本代码。
- 在 ASP 网页中，可以使用已经装有 COM（组件对象模型）脚本兼容引擎的任何脚本编辑语言。

1.2.2 ASP 工作过程

ASP 页面是一种包含脚本代码的网页，在发送到客户端浏览器之前，站点的 Web 服务器将对其中包含的脚本代码进行处理。

HTML 是最简单、最基本的网页编写语言，使用纯粹的 HTML 标记只能创建静态网页。当一个客户通过浏览器向站点的 Web 服务器请求静态 HTML 网页时，Web 服务器将向客户端浏览器直接发送所请求的 HTML 文档而无需经过任何处理。随后，客户端浏览器处理这个接收到的文档，并按照其中 HTML 代码的规定显示该文档的内容。其工作过程如图 1-3 所示。

而当客户端浏览器向站点的 Web 服务器请求 ASP 网页时，Web 服务器先将 ASP 文档交给自身携带的 ASP 引擎，在此处执行 ASP 文档中的脚本代码并将结果转化为 HTML 代码，然后再将转换后的代码发送到客户端的浏览器。其过程如图 1-4 所示。

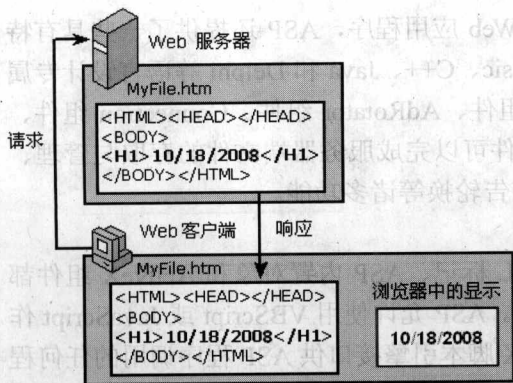


图 1-3 HTML 静态网页工作过程

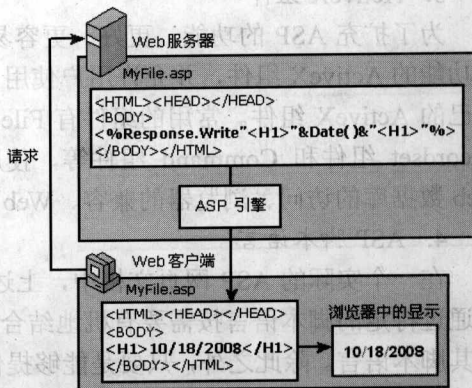


图 1-4 ASP 动态网页工作过程

由此可见，ASP 极大地简化了 Web 应用程序的开发过程。只需几行脚本，就可以在网页中添加数据库连接或一些高级自定义功能。有了 ASP，就可以使用普通的站点脚本语言，如 VBScript、JScript 或 JavaScript 等编写各种网络应用程序。

由于此种应用程序的脚本代码是在服务器端而不是在客户端执行，传送到客户端浏览器的 Web 页面是在 Web 服务器上生成的标准的 HTML 文档，所以不必担心浏览器是否能够处理此种脚本程序代码。此外，由于仅将脚本程序的处理结果发送到客户端的浏览器，所以服