



21世纪高等学校电子信息类专业规划教材

ASP.NET

案例教程

林菲 孙勇 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北京交通大学出版社
<http://press.bjtu.edu.cn>

21 世纪高等学校电子信息类专业规划教材

ASP.NET 案例教程

林 菲 孙 勇 编著

清华大学出版社
北京交通大学出版社

·北京·

内 容 简 介

本书系统地介绍了 ASP.NET 3.5 网站开发技术。全书共分为 13 章, 主要内容包括: Web 应用基础、Visual Studio 2008 集成开发环境、ASP.NET 3.5 技术基础、ASP.NET 服务器控件、Web 应用的状态管理、页面外观设计与布局、站点导航技术、ADO.NET 数据访问技术、ASP.NET 的数据绑定与绑定控件、ASP.NET 安全管理、Web 服务、ASP.NET AJAX、Web 应用程序的部署。此外, 每章均配有上机实训和习题, 有助于读者对每章所学知识的掌握。

本书有配套的实训教材——《ASP.NET 案例教程实训指导》。配套教材以一个大型项目为素材, 围绕本书的章节展开, 提供了项目开发的实际操作, 并补充了教材中没有介绍的内容, 如开发技巧等。

本书结构清晰, 实例丰富, 图文对照, 浅显易懂, 可作为高等院校计算机及相关专业 ASP.NET 开发课程的教材, 还可作为有一定的面向对象编程基础和数据库基础, 想利用 Visual Studio 2008 开发 Web 应用程序的开发人员的参考书籍。

本书的电子教案、示例源代码、习题答案和上机实训源代码可以到北京交通大学出版社网站下载, 网址为 <http://press.bjtu.edu.cn>。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13501256678 13801310933

图书在版编目 (CIP) 数据

ASP.NET 案例教程/林菲, 孙勇编著. —北京: 清华大学出版社; 北京交通大学出版社, 2009. 9

(21 世纪高等学校电子信息类专业规划教材)

ISBN 978-7-81123-694-1

I. A… II. ①林… ②孙… III. 主页制作-程序设计-高等学校-教材 IV. TP393.092

中国版本图书馆 CIP 数据核字 (2009) 第 163840 号

责任编辑: 郭东青

出版发行: 清华大学出版社 邮编: 100084 电话: 010-62776969 <http://www.tup.com.cn>
北京交通大学出版社 邮编: 100044 电话: 010-51686414 <http://press.bjtu.edu.cn>

印刷者: 北京瑞达方舟印务有限公司

经 销: 全国新华书店

开 本: 185×260 印张: 25.75 字数: 640 千字

版 次: 2009 年 10 月第 1 版 2009 年 10 月第 1 次印刷

书 号: ISBN 978-7-81123-694-1/TP·522

印 数: 1~4000 册 定价: 38.00 元

本书如有质量问题, 请向北京交通大学出版社质监组反映。对您的意见和批评, 我们表示欢迎和感谢。

投诉电话: 010-51686043, 51686008; 传真: 010-62225406; E-mail: press@bjtu.edu.cn。

前 言

ASP.NET 技术是目前开发 Web 应用程序最流行和最前沿的技术,它提供了为建立和部署企业级 Web 应用程序所必需的服务。ASP.NET 3.5 是目前开发 ASP.NET 的最新发布版本。使用 Visual Studio 2008 (简称 VS 2008) 和 C# 开发 ASP.NET 应用程序是最佳选择,也深受广大编程人员青睐。Visual Studio 2008 中提供了 Web 应用程序的开发模型,利用这些开发模型,开发者能够实现各种复杂的页面设计和后台代码处理功能。

本书以易学易用为重点,充分考虑实际的开发需求,使用大量实例,引导读者掌握 ASP.NET 页面设计和网站开发的方法和技巧。读者学习本书各章知识点时,可以通过各章所配套的实训和习题巩固所学内容。另外,还编写了与本书配套的实训教材——《ASP.NET 案例教程实训指导》,该实训教材围绕本书讲授知识点的顺序开发一个实用的网络考试系统。读者可以在学习本书的同时,按章节顺序完成该配套教材中的实训内容。掌握网络考试系统应用程序的开发后,读者可以非常方便地设计并实现其他类似的 Web 应用程序。

本书系统地介绍了 ASP.NET 3.5 网站开发技术。全书共分为 13 章,各章内容要点如下。

第 1 章 Web 应用基础。主要介绍 Web 应用相关技术的发展,并对 HTML、XHTML、XML 及 Web 服务器的基本知识做了简单介绍,方便后续章节的学习。

第 2 章 Visual Studio 2008 集成开发环境。主要介绍如何使用 Visual Studio 2008 集成开发环境创建一个简单的 Hello World Web 应用程序,介绍开发 Web 应用程序过程中 Visual Studio 2008 集成开发环境的几个常用窗口。

第 3 章 ASP.NET 3.5 技术基础。主要介绍 ASP.NET 3.5 的基础知识,包括:ASP.NET 应用程序生命周期,ASP.NET 网页,Page 类的内置对象,Web 应用的配置与配置管理工具及 Web 应用的异常处理机制。

第 4 章 ASP.NET 服务器控件。主要介绍 HTML 服务器控件、Web 服务器控件、验证控件。最后,简单介绍了用户控件的创建和使用。

第 5 章 Web 应用的状态管理。主要介绍了客户端状态管理技术和服务器端状态管理技术。其中客户端状态管理技术包括视图状态、查询字符串和 Cookie;服务器端状态管理技术包括会话状态和应用程序状态管理。

第 6 章 页面外观设计与布局。首先介绍了 CSS 样式,以及如何在 ASP.NET 页面中应用 CSS。接下来讨论 ASP.NET 中的主题,以及利用主题为页面提供一致外观的方法。最后介绍母版页的创建与使用方法,以及母版页的多层嵌套技术。

第 7 章 站点导航技术。主要介绍了站点地图的创建及站点导航控件的使用,包括 SiteMapPath 控件、SiteMapDataSource 控件、Menu 控件和 TreeView 控件。

第8章 ADO.NET 数据访问技术。主要介绍如何通过 ADO.NET 技术中的五大对象访问数据库。以连接模式和断开模式两条主线分别介绍,并分析了这两种模式的适用场景。

第9章 ASP.NET 3.5 的数据绑定及绑定控件。主要介绍了数据源控件和数据绑定控件。其中,数据源控件主要介绍了 SqlDataSource、ObjectDataSource 和 LinqDataSource 控件的使用;数据绑定控件主要介绍了 GridView、DetailsView、FormView 和 ListView 控件的使用。

第10章 ASP.NET 安全管理。主要介绍了 ASP.NET 中三种用于安全管理的技术,包括身份验证技术、授权技术和配置文件加密技术。

第11章 Web 服务。主要介绍了微软公司 Web 服务相关的基本概念和特点,以及什么情况下需要使用 Web 服务,并通过具体示例说明了在 ASP.NET 3.5 中创建与使用 Web 服务的方法。

第12章 ASP.NET AJAX。主要介绍了 AJAX 技术及相关概念、ASP.NET AJAX 主要控件的使用方法及 ASP.NET AJAX Control Toolkit 工具集。

第13章 Web 应用程序的部署。主要介绍了部署 Web 应用程序的3种方法:复制网站、发布网站和创建安装包,并分析了3种方法的优缺点,以便能正确地选择合适的方法部署 Web 应用程序。

为了配合教学需要,本书配有电子教案、示例源代码、习题参考答案和上机实训参考源代码。读者可以到北京交通大学出版社网站下载,网址为 <http://press.bjtu.edu.cn>。在学习本书的过程中,建议读者可以在互联网上下载微软的 WebCast 相关课程的视频来帮助学习。另外,要真正精通 ASP.NET 技术的开发,除了学习本书知识,还应该培养独立解决实际问题的能力。

本书由林菲、孙勇编著,林菲、孙勇对全书内容进行了统稿、修改、整理和定稿。其中第1章、第2章、第3章、第4章、第5章、第7章、第8章及第9章由林菲编写;第6章、第10章、第11章、第12章及第13章由孙勇编写。徐海涛负责全书的文字校对、习题参考答案及上机实训参考源代码的整理工作;方绪健负责资料收集及电子教案的制作工作。

在编写本书的过程中参考了相关文献,在此向这些文献的作者深表感谢。由于编者水平有限,书中难免有不妥之处,敬请专家和广大读者批评指正。

编 者

2009年8月

目 录

第1章 Web 应用基础	1
1.1 Web 应用概述	1
1.1.1 B/S 与 C/S 结构模式	1
1.1.2 Web 应用相关技术的发展	1
1.2 Web 应用的相关技术	4
1.2.1 HTML	4
1.2.2 XHTML	7
1.2.3 可扩展标记语言 XML	10
1.2.4 Web 服务器	14
1.3 小结	16
实训1 Web 应用基础	16
习题	17
第2章 Visual Studio 2008 集成开发环境	20
2.1 创建一个简单的 ASP.NET 应用程序	20
2.2 Visual Studio 2008 IDE 集成开发环境介绍	23
2.2.1 服务器资源管理器	24
2.2.2 解决方案资源管理器	24
2.2.3 工具箱	26
2.2.4 Web 页面设计窗口	26
2.2.5 HTML 源代码编辑窗口	27
2.2.6 后台代码编辑窗口	27
2.2.7 属性窗口	28
2.2.8 类视图	29
2.2.9 对象浏览器	29
2.3 小结	30
实训2 Visual Studio 2008 集成开发环境	30
习题	30



第3章 ASP.NET 3.5 技术基础	31
3.1 ASP.NET 应用程序生命周期	31
3.1.1 应用程序生命周期概述	31
3.1.2 应用程序生命周期事件	33
3.2 ASP.NET 网页	34
3.2.1 ASP.NET 网页语法概述	35
3.2.2 ASP.NET 网页代码模型	37
3.2.3 Page 类的属性	41
3.2.4 ASP.NET 网页的生命周期与 Page 类的事件	43
3.2.5 ASP.NET 网页的添加	44
3.3 Page 类的内置对象	45
3.3.1 Response 对象	45
3.3.2 Request 对象	48
3.3.3 Server 对象	53
3.4 Web 应用的配置与配置管理工具	55
3.4.1 web.config 配置文件	55
3.4.2 嵌套配置设置	57
3.4.3 在 web.config 中存储自定义设置	58
3.4.4 ASP.NET Web 站点管理工具 WAT	59
3.5 Web 应用的异常处理	61
3.5.1 为什么要进行异常处理	61
3.5.2 try-catch 异常处理块	62
3.5.3 页面级的 Page_Error 事件异常处理	64
3.5.4 页面级的 ErrorPage 属性异常处理	64
3.5.5 应用程序级的 Application_Error 事件异常处理	65
3.5.6 配置应用程序的 <customErrors> 配置项异常处理	66
3.6 小结	67
实训 3 ASP.NET 3.5 技术基础	67
习题	68
第4章 ASP.NET 服务器控件	70
4.1 服务器控件概述	70
4.2 HTML 服务器控件	71
4.2.1 HTML 服务器控件概述	71
4.2.2 HTML 服务器控件综合示例	76
4.3 Web 服务器控件	78

4.3.1 Web 服务器控件概述	78
4.3.2 常用 Web 服务器控件	87
4.4 验证控件	107
4.4.1 验证控件概述	107
4.4.2 验证控件的使用	109
4.4.3 验证组的使用	119
4.4.4 禁用验证	120
4.4.5 以编程方式测试验证有效性	120
4.5 用户控件	121
4.5.1 用户控件的创建	122
4.5.2 用户控件的使用	125
4.6 小结	126
实训4 ASP. NET 服务器控件	127
习题	127
第5章 Web 应用的状态管理	133
5.1 Web 应用状态管理概述	133
5.2 客户端状态管理	134
5.2.1 视图状态	134
5.2.2 查询字符串	137
5.2.3 Cookie	138
5.3 服务器端状态管理	146
5.3.1 会话状态	146
5.3.2 应用程序状态	152
5.4 小结	155
实训5 Web 应用的状态管理	155
习题	156
第6章 页面外观设计与布局	158
6.1 CSS 样式控制	158
6.1.1 页面中使用 CSS 的三种方法	158
6.1.2 样式规则	164
6.2 主题	167
6.2.1 主题的创建与应用	167
6.2.2 主题中的外观文件	169
6.2.3 主题中的 CSS 样式文件	170



6.2.4 主题的动态应用	171
6.3 母版页	172
6.3.1 创建母版页	173
6.3.2 创建内容页	176
6.3.3 母版页的工作原理	178
6.3.4 母版页和内容页中的事件	178
6.3.5 从内容页访问母版页的内容	179
6.3.6 母版页的嵌套	180
6.4 小结	181
实训6 页面外观设计与布局	182
习题	183
第7章 站点导航技术	185
7.1 ASP.NET 站点导航概述	185
7.2 站点地图	186
7.3 配置多个站点地图	189
7.3.1 从父站点地图链接到子站点地图文件	189
7.3.2 在 web.config 文件中配置多个站点地图	190
7.4 SiteMapPath 控件	191
7.5 SiteMapDataSource 控件	193
7.6 Menu 控件	195
7.6.1 定义 Menu 菜单内容	195
7.6.2 Menu 控件样式	201
7.7 TreeView 控件	202
7.7.1 定义 TreeView 节点内容	202
7.7.2 带复选框的 TreeView 控件	209
7.8 小结	211
实训7 站点导航技术	211
习题	212
第8章 ADO.NET 数据访问技术	214
8.1 ADO.NET 基础	214
8.1.1 ADO.NET 简介	214
8.1.2 ADO.NET 的组件	215
8.1.3 ADO.NET 的数据访问模式	216
8.2 连接模式数据库访问	217

8.2.1 使用 SqlConnection 对象连接数据库	218
8.2.2 使用 SqlCommand 对象执行数据库命令	225
8.2.3 使用 SqlDataReader 读取数据	234
8.2.4 为 SqlCommand 传递参数	239
8.2.5 使用 SqlCommand 执行存储过程	243
8.2.6 使用事务处理	246
8.3 断开模式数据库访问	248
8.3.1 DataSet 数据集	249
8.3.2 使用 SqlDataAdapter 对象执行数据库命令	256
8.4 小结	267
实训 8 ADO.NET 数据访问技术	268
习题	269
第 9 章 ASP.NET 的数据绑定及绑定控件	274
9.1 数据源控件	274
9.1.1 SqlDataSource 数据源控件	275
9.1.2 ObjectDataSource 数据源控件	289
9.1.3 LinqDataSource 数据源控件	293
9.2 数据绑定控件	296
9.2.1 GridView 控件	297
9.2.2 DetailsView 控件	309
9.2.3 FormView 控件	318
9.2.4 ListView 控件和 DataPager 控件	321
9.3 小结	333
实训 9 ASP.NET 的数据绑定及绑定控件	333
习题	334
第 10 章 ASP.NET 安全管理	337
10.1 身份验证	337
10.1.1 验证模式	337
10.1.2 使用 CreateUserWizard 控件注册	338
10.1.3 使用 Login 控件登录	339
10.1.4 其他登录型控件	340
10.2 角色与授权	342
10.2.1 创建角色	342
10.2.2 在 web.config 中授权	343



10.3 通过编程方式实现验证与授权	344
10.3.1 使用成员资格服务类验证	344
10.3.2 使用角色管理类授权	347
10.4 配置文件加密	348
10.5 小结	348
实训 10 ASP.NET 安全管理	349
习题	349
第 11 章 Web 服务	352
11.1 云计算与 Web 服务	352
11.2 Web 服务的相关标准与规范	354
11.3 创建 Web 服务	355
11.3.1 Web 服务的声明	355
11.3.2 Web 方法的定义	356
11.3.3 Web 服务的测试	356
11.3.4 创建 Web 服务示例	357
11.4 使用 Web 服务	359
11.4.1 添加 Web 引用	359
11.4.2 调用 Web 服务	360
11.5 小结	361
实训 11 Web 服务	362
习题	362
第 12 章 ASP.NET AJAX	363
12.1 ASP.NET AJAX 简介	363
12.1.1 AJAX 概述	363
12.1.2 ASP.NET AJAX 技术特点	364
12.1.3 Hello World 示例程序	364
12.2 ScriptManager 控件	365
12.2.1 在页面中添加 ScriptManager 控件	365
12.2.2 ScriptManager 控件的属性与方法	366
12.3 UpdatePanel 控件	366
12.3.1 在页面中添加 UpdatePanel 控件	367
12.3.2 UpdatePanel 控件的属性	368
12.3.3 页面中的多个 UpdatePanel 及更新模式	369
12.3.4 UpdatePanel 更新策略总结	371

12.4 UpdateProgress 控件	372
12.4.1 UpdateProgress 控件的属性	372
12.4.2 UpdateProgress 控件的使用方法	372
12.5 Timer 控件	373
12.5.1 在页面中添加 Timer 控件	373
12.5.2 Timer 控件的属性与事件	374
12.5.3 Timer 控件的使用方法	374
12.6 ASP.NET AJAX Control Toolkit	375
12.6.1 安装 ASP.NET AJAX Control Toolkit	376
12.6.2 ASP.NET AJAX Control Toolkit 的示例站点	377
12.6.3 AlwaysVisibleControlExtender 控件	378
12.6.4 ModalPopupExtender 控件	378
12.6.5 Accordion 控件	380
12.7 小结	381
实训 12 ASP.NET AJAX	382
习题	382
 第 13 章 Web 应用程序的部署	383
13.1 部署的内容	383
13.2 部署准备	383
13.3 部署 Web 应用程序的方法	384
13.3.1 使用复制网站工具部署站点	384
13.3.2 使用发布网站工具部署站点	387
13.3.3 创建安装包部署站点	389
13.4 小结	395
实训 13 Web 应用程序的部署	395
习题	396
 参考文献	397

第 1 章 Web 应用基础

Web 应用和相关技术的飞速发展给人们的工作、学习和生活带来了重大变化，人们可以利用网络处理数据、获取信息，极大地提高了工作效率。Web 应用已经成为目前企业应用最广泛的一种形式。本章将重点介绍 Web 应用的基本概念、发展历程及相关技术。

1.1 Web 应用概述

1.1.1 B/S 与 C/S 结构模式

在企业应用软件中，若按系统部署的体系结构来分，往往可将其分为 B/S (Browser/Server) 和 C/S (Client/Server) 两种结构模式。

C/S 架构是指通过在客户端安装一个软件，通过该软件访问服务端资源的一种结构体系。例如，网络游戏《魔兽世界》，基本就属于 C/S 结构，C 就是通常说的胖客户端。这种结构的好处是很多服务可以不在服务端进行处理，由客户端直接处理。因此受网络的影响较小。但是不足就是对客户端的要求较高，而且需要在客户端安装较大的客户端程序。

B/S 模式是指通过在服务端安装一些应用程序，在客户端只要通过浏览器访问服务器，就可以查看相关内容。例如，新浪、搜狐等网站，就属于 B/S 结构，也就是通常说的瘦客户端。这个结构中几乎所有的服务都在服务端处理。好处就是对客户端要求不高，一般只需要浏览器就可以了，而且便于进行权限验证，安全维护。但缺陷就是处理任何内容，可能都要送到服务器端处理，因此，需要经常刷新页面，受网络条件的影响很大。如果网络不好，刷新速度会很慢。

Web 应用就是指在 B/S 结构体系下的应用软件系统，除前面所提到的网站外，还有很多的电子商务网站、Hotmail、百度、企业应用中的 OA (Office Automation, 办公自动化系统) 等，这些都属于 Web 应用的范畴。

1.1.2 Web 应用相关技术的发展

Web 这个 Internet 上最热门的应用架构是由 Tim Berners-Lee 发明的。Web 的前身是 1980 年 Tim Berners-Lee 负责的 Enquire (Enquire Within Upon Everything 的简称) 项目。1990 年 11 月，第一个 Web 服务器 nxoc01.cern.ch 开始运行，Tim Berners-Lee 在自己编写的图形化 Web 浏览器 “WorldWideWeb” 上看到了最早的 Web 页面。1991 年，CERN (European Particle Physics Laboratory) 正式发布了 Web 技术标准。目前，与 Web 相关的各种技术标准都由著名的 W3C 组织 (World Wide Web Consortium) 管理和维护。



Web 是一个分布式的超媒体 (hypermedia) 信息系统, 它将大量的信息分布于整个 Internet 上。Web 的任务就是向人们提供多媒体网络信息服务。

从技术层面看, Web 技术核心有三点:

- ◆ 超文本传输 (HTTP) 协议, 实现万维网的信息传输;
- ◆ 统一资源定位符 (URL), 实现 Internet 信息的定位统一标识;
- ◆ 超文本标记语言 (HTML), 实现信息的表示与存储。

Web 是一种典型的分布式应用架构。Web 应用中的每一次信息交换都要涉及客户端和服务端两个层面。因此, Web 开发技术大体上也可以被分为客户端技术和服务端技术两大类。下面分别介绍客户端和服务端技术的发展。

1. 客户端技术的发展

Web 客户端的主要任务是展现信息内容, 而 HTML 语言则是信息展现的最有效载体之一。作为一种实用的超文本语言, HTML 的历史最早可以追溯到 20 世纪 40 年代。1945 年, Vannevar Bush 在一篇文章中阐述了文本和文本之间通过超级链接相互关联的思想, 并在文中给出了一种能实现信息关联的计算机 Memex 的设计方案。Doug Engelbart 等人则在 1960 年前后, 对信息关联技术做了最早的实验。与此同时, Ted Nelson 正式将这种信息关联技术命名为超文本 (Hypertext) 技术。1969 年, IBM 的 Charles Goldfarb 发明了可用于描述超文本信息的 GML (Generalized Markup Language) 语言。1978 年到 1986 年间, 在 ANSI 等组织的努力下, GML 语言进一步发展成为著名的 SGML 语言标准。当 Tim Berners-Lee 和他的同事们在 1989 年试图创建一个基于超文本的分布式应用系统时, Tim Berners-Lee 意识到, SGML 是描述超文本信息的一个上佳方案, 但美中不足的是, SGML 过于复杂, 不利于信息的传递和解析。于是, Tim Berners-Lee 对 SGML 语言做了大刀阔斧的简化和完善。1990 年, 第一个图形化的 Web 浏览器 “World Wide Web” 终于可以使用一种为 Web 度身定制的语言——HTML 来展现超文本信息了。

最初的 HTML 语言只能在浏览器中展现静态的文本或图像信息, 这满足不了人们对信息丰富性和多样性的强烈需求——这件事情最终的结果是, 由静态技术向动态技术的转变成为了 Web 客户端技术演进的永恒定律。

能存储、展现二维动画的 GIF 图像格式早在 1989 年就已发展成熟。Web 出现后, GIF 第一次为 HTML 页面引入了动感元素。但更大的变革来源于 1995 年 Java 语言的问世。Java 语言天生就具备的与平台无关的特点, 让人们一下子找到了在浏览器中开发动态应用的捷径。1996 年, 著名的 Netscape 浏览器在其 2.0 版中增加了对 JavaApplets 和 JavaScript 的支持。Microsoft 的 IE 3.0 也在这一年开始支持 Java 技术。现在, 喜欢动画、喜欢交互操作、喜欢客户端应用的开发人员可以用 Java 或 JavaScript 语言制作出丰富多彩的 HTML 页面。为了与 JavaScript 抗衡, Microsoft 还为 1996 年的 IE 3.0 设计了另一种后来也声名显赫的脚本语言——VBScript 语言。

真正让 HTML 页面动感无限的是 CSS (Cascading Style Sheets) 和 DHTML (Dynamic HTML) 技术。1996 年底, W3C 提出了 CSS 的建议标准, 同年, IE 3.0 引入了对 CSS 的支持。CSS 大大提高了开发者对信息展现格式的控制能力。1997 年的 Netscape 4.0 不但支持 CSS, 而且增加了许多 Netscape 公司自定义的动态 HTML 标记, 这些标记在 CSS 的基础上, 让 HTML 页面中的各种要素 “活动” 了起来。1997 年, Microsoft 发布了 IE 4.0, 并将动态 HTML 标记、

CSS 和动态对象模型 (DHTML Object Model) 发展成了一套完整、实用、高效的客户端开发技术体系, Microsoft 称其为 DHTML。同样是实现 HTML 页面的动态效果, DHTML 技术无须启动 Java 虚拟机或其他脚本环境, 可以在浏览器的支持下, 获得更好的展现效果和更高的执行效率。今天, 已经很少有哪个 HTML 页面的开发者还会对 CSS 和 DHTML 技术视而不见了。

为了在 HTML 页面中实现音频、视频等更为复杂的多媒体应用, 1996 年的 Netscape 2.0 成功地引入了对 QuickTime 插件的支持, 插件这种开发方式也迅速风靡了浏览器的世界。在 Windows 平台上, Microsoft 将客户端应用集成的赌注押到了 90 年代中期刚刚问世的 COM 和 ActiveX 身上。1996 年, IE 3.0 正式支持在 HTML 页面中插入 ActiveX 控件的功能, 这为其他厂商扩展 Web 客户端的信息展现方式开辟了一条自由之路。1999 年, Realplayer 插件先后在 Netscape 和 IE 浏览器中取得了成功, 与此同时, Microsoft 自己的媒体播放插件 Media Player 也被预装到了各种 Windows 版本之中。同样值得纪念的还有 Flash 插件的横空出世: 90 年代初期, Jonathan Gay 在 FutureWave 公司开发了一种名为 Future Splash Animator 的二维矢量动画展示工具, 1996 年, Macromedia 公司收购了 FutureWave, 并将 Jonathan Gay 的发明改名为我们熟悉的 Flash。从此, Flash 动画成了 Web 开发者表现自我、展示个性的最佳方式。

2. 服务端技术的成熟与发展

与客户端技术从静态向动态的演进过程类似, Web 服务端的开发技术也是由静态向动态逐渐发展、完善起来的。

最早的 Web 服务器简单地响应浏览器发来的 HTTP 请求, 并将存储在服务器上的 HTML 文件返回给浏览器。一种名为 SSI (Server Side Includes) 的技术可以让 Web 服务器在返回 HTML 文件前, 更新 HTML 文件的某些内容, 但其功能非常有限。第一种真正使服务器能根据运行时的具体情况, 动态生成 HTML 页面的技术是大名鼎鼎的 CGI (Common Gateway Interface) 技术。1993 年, CGI 1.0 的标准草案由 NCSA (National Center for Supercomputing Applications) 提出, 1995 年, NCSA 开始制定 CGI 1.1 标准, 1997 年, CGI 1.2 也被纳入了议事日程。CGI 技术允许服务端的应用程序根据客户端的请求, 动态生成 HTML 页面, 这使客户端和服务端的动态信息交换成为可能。随着 CGI 技术的普及, 聊天室、论坛、电子商务、信息查询、全文检索等各式各样的 Web 应用蓬勃兴起, 人们终于可以享受到信息检索、信息交换、信息处理等更为便捷的信息服务了。

早期的 CGI 程序大多是编译后的可执行程序, 其编程语言可以是 C、C++、Pascal 等任何通用的程序设计语言。为了简化 CGI 程序的修改、编译和发布过程, 人们开始探寻用脚本语言实现 CGI 应用的可行方式。在此方面, 不能不提的是 Larry Wall 于 1987 年发明的 Perl 语言。Perl 结合了 C 语言的高效及 shell、awk 等脚本语言的便捷, 似乎天生就适用于 CGI 程序的编写。1995 年, 第一个用 Perl 写成的 CGI 程序问世。很快, Perl 在 CGI 编程领域的风头就盖过了它的前辈 C 语言。随后, Python 等著名的脚本语言也陆续加入了 CGI 编程语言的行列。

1994 年, Rasmus Lerdorf 发明了专用于 Web 服务端编程的 PHP (Personal Home Page Tools) 语言。与以往的 CGI 程序不同, PHP 语言将 HTML 代码和 PHP 指令合成为完整的服务端动态页面, Web 应用的开发者可以用一种更加简便、快捷的方式实现动态 Web 功能。1996 年, Microsoft 借鉴 PHP 的思想, 在其 Web 服务器 IIS 3.0 中引入了 ASP 技术。ASP 使用的脚本语言是我们熟悉的 VBScript 和 JavaScript。



Web 服务端开发技术的完善使开发复杂的 Web 应用成为可能。随着电子政务、电子商务等大规模 Web 应用的迅速推广,为了适应 Web 应用开发的各种复杂需求,为用户提供更可靠、更完善的信息服 务,两个最重要的企业级开发平台——J2EE 和 .NET 在 2000 年前后分别由 Java 和 Windows 阵营推出。由此引发了在企业级 Web 开发平台领域的激烈竞争,也促使 Web 开发技术以前所未有的速度发展。

J2EE 是纯粹基于 Java 的解决方案。1997 年,Servlet 技术问世,1998 年,JSP 技术诞生。Servlet 和 JSP 的组合让 Java 开发者同时拥有了类似 CGI 程序的集中处理功能和类似 PHP 的 HTML 嵌入功能。1998 年,Sun 公司发布了 EJB 1.0 标准。EJB 为企业级应用中必不可少的数据封装、事务处理、交易控制等功能提供了良好的技术基础。1999 年,Sun 正式发布了 J2EE 的第一个版本,J2EE 平台的三大核心技术是 Servlet、JSP 和 EJB。随后,又出现了多个遵循 J2EE 标准,为企业级应用提供支撑平台的各类应用服务软件。最具代表性的是 IBM 的 WebSphere 和 BEA 的 WebLogic。

2002 年,Microsoft 正式发布 .NET Framework 和 Visual Studio .NET 开发环境。和 J2EE 不同,Microsoft 的 .NET 平台及相关的开发环境为 Web 服务端应用提供了一个支持多种语言的、通用的运行平台。伴随着 .NET 技术的出现,ASP.NET 1.0 也应运而生。ASP.NET 超越了 ASP 的局限,是 ASP 的升级版本,提供了一种以 Microsoft .NET Framework 为基础开发 Web 应用程序的全新编程模式。ASP.NET 作为 Windows 平台上流行的网站开发工具,能够提供各种方便的 Web 开发模型,利用这些模型开发人员能够快速开发出动态网站所需的各种复杂功能,提高了开发效率和网站性能。

2005 年,.NET 框架从 1.0 版升级为 2.0 版,Microsoft 公司发布了 Visual Studio 2005,相应的 ASP.NET 1.0 也升级为 ASP.NET 2.0。它修正了以往版本中的一些 Bug 并在移动应用程序的开发、代码安全及对 Oracle 数据库和 ODBC 的支持等方面都做了很多改进。

2008 年,Visual Studio 2008 问世,ASP.NET 由 2.0 版升级为 3.5 版。该版本支持 AJAX 技术和语言集成查询 (LINQ) 技术,提供了新的服务器控件和新的面向对象的客户端类型库等功能。

1.2 Web 应用的相关技术

Web 应用的相关技术有很多,如 HTML、XHTML、XML、JavaScript、VBScript、PHP、JSP、ASP、ASP.NET 等,本书主要学习用 ASP.NET 3.5 技术开发 Web 应用程序。为了方便后续章节的学习,本节有必要介绍一下 HTML、XHTML 和 XML 技术。

1.2.1 HTML

HTML 是一种用来制作超文本文档的简单标记语言。超文本传输协议规定了浏览器在运行 HTML 文档时所遵循的规则和进行的操作。HTTP 协议的制定使浏览器在运行超文本时有了统一的规则 and 标准。用 HTML 编写的超文本文档称为 HTML 文档,它能独立于各种操作系统平台,自 1990 年以来 HTML 就一直被用做 WWW (World Wide Web,万维网)的信息表示语言,使用 HTML 语言描述的文件,需要通过 Web 浏览器显示出效果。

每一个 HTML 文档都是一种静态的网页文件，这个文件里面包含了 HTML 指令代码，这些指令代码并不是一种程序语言，它只是一种排版网页中资料显示位置的标记结构语言，易学易懂，非常简单。同时，HTML 标准在不断地发展完善，每个版本都会提供一些新的功能，让设计者更加方便、灵活地设计网页，但是它的基本格式没有变化。HTML 文档的内容一般都位于 `<html>` 和 `</html>` 之间，分为首部 and 主体两个部分。文档中的命令一般采用 `<标记>` 和 `</标记>` 的形式配对出现，有些标记也可以单个出现，标记符不区分大小写。基本格式：

```
<html>
  <head>
    <title> </title>
  </head>
  <body>
    HTML 文档的主体部分
  </body>
</html>
```

文档的首部位于标记 `<head>` 和 `</head>` 之间，其中可以加入其他标记，例如，标题标记 `title`、样式标记 `style` 和脚本语言标记 `script` 等。首部的信息不在网页内出现，只用来设置 HTML 文档的标题、作者等信息，因此在 HTML 中可以省略。标记 `<body>` 和 `</body>` 之间的内容构成了 HTML 的主体部分，也是主要设计区域，网页中所有的内容，包括文字、图形、链接及其他网页元素都包含在该区域内。

在由“`<`”和“`>`”包含的标记中，通过设置属性，能使页面产生不同的效果。`<标记>` 中可以包含一个或多个属性，若有多个属性，则各属性之间必须用空格隔开。属性名和属性值之间用等号隔开，等号左边是属性名称，右边是属性值。一般格式为：

```
<标记 属性1 = "值1" 属性2 = "值2"...>
```

在 HTML 中，等号右边的属性值可以带双引号，也可以省略双引号。例如：

```
<body bgcolor="silver" text="blue" link="red">
```

标记 `<body>` 中设置了 3 个属性，其中，`bgcolor` 属性用于设置网页的背景颜色；`text` 属性用于设置网页中文字信息的颜色；`link` 属性用于设置网页中超链接的颜色。在此，分别设置为银色、蓝色和红色。

除上面所提到的一些主要结构标记外，HTML 还提供了许多其他标记，表 1-1 中列出了部分常用的 HTML 标记符及其功能。

表 1-1 常用 HTML 标记符及其功能

标记格式	功 能	标记格式	功 能
<code><head>...</head></code>	首部标记符，不包含网页的内容，仅提供一些与网页相关的信息	<code><div>...</div></code>	定义一个块，块内可以有文本、图像等，目的是为了控制该块的样式
<code><body>...</body></code>	主体标记符，包含网页内的所有内容：文字、图片及超链接等	<code>
</code>	强行中断当前行，多个 <code>
</code> 标记可以创建多个空行