

计算机基础课程系列教材

ASP.NET程序 设计教程

第2版

本书为教师配有
完备的教学资源

郑阿奇 主编
顾韵华 等编著



机械工业出版社
China Machine Press

计算机基础课程系列教材

ASP.NET程序 设计教程

第2版

郑阿奇 主编
顾韵华 等编著



机械工业出版社
China Machine Press

本书第2版在继承第1版优点的前提下，将全代码方式改为开发平台方式，从而更突出主要内容。本书内容包括Web基础知识、ASP.NET应用程序基础、Visual Basic .NET语言、服务器控件、ASP.NET内建对象、文件操作、网站设计、ADO.NET数据库、数据库编程和高级应用。

本书第1~10章是教程部分，讲解ASP.NET开发的基础知识，并包含大量实例；然后是上机实验指导，帮助读者通过上机操作巩固所学知识；最后通过一个综合应用实例（BBS系统）教会读者活学活用。

本书可作为大学本、专科有关课程的教材，也可作为各类ASP.NET培训和广大用户自学与参考资料。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问 北京市展达律师事务所

图书在版编目（CIP）数据

ASP.NET程序设计教程 / 郑阿奇主编. —2版. —北京：机械工业出版社，2011.4
(计算机基础课程系列教材)

ISBN 978-7-111-33647-1

I. A… II. 郑… III. 网页制作工具—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（2011）第035083号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：刘立卿

北京市荣盛彩色印刷有限公司印刷

2011年5月第2版第1次印刷

185mm×260mm · 22.25印张

标准书号：ISBN 978-7-111-33647-1

定价：39.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991；88361066

购书热线：(010) 68326294；88379649；68995259

投稿热线：(010) 88379604

读者信箱：hzjsj@hzbook.com

前 言

ASP.NET开发B/S平台应用系统是市场的主流，ASP.NET程序设计已经成为普通高等学校许多专业必开的课程。2006年，我们根据当时的情况和应用实践，采用全代码方式编写了《ASP.NET程序设计教程》，受到市场的广泛好评。如今，Visual Studio 2008已经相当流行，通过该系统提供的平台进行ASP.NET开发非常方便，其脚本语言是C#和Visual Basic .NET。目前，市场上介绍以C#作为脚本语言开发ASP.NET的教程比较多，但介绍以Visual Basic .NET作为脚本语言开发ASP.NET的教程比较少。在第2版中，我们根据近年来采用Visual Basic .NET脚本语言开发ASP.NET的实践，在继承第1版优点的前提下，将全代码方式改为开发平台方式，这样更能突出主要内容。

本书分成教程、实验、综合应用实例和附录几个部分。教程部分包括Web基础知识、ASP.NET应用程序基础、Visual Basic .NET语言、服务器控件、ASP.NET内建对象、文件操作、网站设计、ADO.NET数据库、数据库编程和高级应用。

本书在多年ASP.NET教学和应用项目开发经验总结的基础上编写完成，充分吸取了此前广受市场和师生欢迎的实用教程编写的成功经验。与现有的ASP.NET教材相比，本教程具有明显的应用特色，只要阅读本书，结合上机操作指导进行练习，就能在较短的时间内基本掌握ASP.NET及其应用技术。

本教程为教师免费提供教学课件、实例文件、综合应用编程源文件，需要者请登录华章网站www.hzbook.com下载。

本书主要由南京信息工程大学顾韵华编写，南京师范大学郑阿奇统编、定稿。参加本教程编写的还有梁敬东、朱毅华、时跃华、赵青松、丁有和、王燕平、彭作民、崔海源、徐卫军、刘毅等。

编 者

2011年3月

目 录

前言

第1章 Web基础知识	1
1.1 TCP/IP协议	1
1.1.1 TCP/IP协议模型	1
1.1.2 IP地址和DNS域名	1
1.2 Web概述	2
1.2.1 HTTP协议	2
1.2.2 统一资源定位符URL	3
1.3 超文本标记语言HTML	3
1.3.1 HTML文档的基本构成	4
1.3.2 HTML基本标记	5
1.4 可扩展标记语言XML	21
1.4.1 XML概述	21
1.4.2 XML文档的编写	22
1.4.3 XML文档的显示	23
1.5 XHTML标记语言	24
1.5.1 XHTML语言概述	24
1.5.2 XHTML的格式	24
1.5.3 XHTML与HTML的区别	24
1.6 Web服务器	26
1.6.1 Web服务器概述	26
1.6.2 IIS的安装与网页的发布	26
习题	27
第2章 ASP.NET应用程序基础	28
2.1 .NET框架	28
2.1.1 .NET框架介绍	28
2.1.2 建立ASP.NET运行环境	29
2.1.3 测试ASP.NET运行环境	30
2.2 Visual Studio 2008开发环境	31
2.2.1 创建Web应用程序的一般步骤	31
2.2.2 编译和运行Web应用程序	32
2.2.3 部署Web应用程序	32
2.2.4 一个简单的ASP.NET实例	33
2.2.5 ASP.NET与XHTML	38
2.3 ASP.NET应用程序结构	39

2.3.1 应用程序目录结构	39
2.3.2 应用程序文件	39
2.4 ASP.NET页面框架	39
2.4.1 aspx页面代码模式	39
2.4.2 aspx页面元素	40
2.4.3 ASP.NET页面布局	42
2.4.4 页面指令	44
2.4.5 页面生命周期	44
2.4.6 页面事件	45
习题	46
第3章 Visual Basic .NET语言	47
3.1 程序结构	47
3.2 数据类型、常量和变量	49
3.2.1 数据类型	49
3.2.2 常量	50
3.2.3 变量	51
3.3 运算符和表达式	51
3.3.1 运算符	51
3.3.2 表达式	53
3.4 流程控制	53
3.4.1 分支结构	53
3.4.2 循环结构	57
3.5 数组	61
3.5.1 数组的声明及使用	61
3.5.2 动态数组	62
3.5.3 与数组相关的系统函数	62
3.6 过程	63
3.6.1 Sub子程序	63
3.6.2 参数传递	64
3.6.3 函数	67
3.7 类和对象基本概念	71
3.7.1 类和对象	71
3.7.2 命名空间	72
3.7.3 访问控制	72
3.7.4 继承	72
习题	73

第4章 服务器控件	74	4.4.5 MultiView和View控件	124
4.1 概述	74	4.4.6 Wizard控件	127
4.1.1 服务器控件分类	74	习题	131
4.1.2 服务器控件属性、事件和方法	74	第5章 ASP.NET内建对象	133
4.1.3 服务器控件层次结构	75	5.1 概述	133
4.2 HTML服务器控件	75	5.1.1 ASP.NET页面处理过程	133
4.2.1 HTML服务器控件概述	75	5.1.2 ASP.NET对象概述	134
4.2.2 HtmlAnchor控件	78	5.2 Page对象	135
4.2.3 HtmlForm控件	79	5.2.1 Page对象的属性	135
4.2.4 HtmlButton控件	79	5.2.2 Page对象的方法	136
4.2.5 HtmlInputButton控件	80	5.2.3 Page对象的事件	136
4.2.6 HtmlInputImage控件	81	5.3 Response对象	138
4.2.7 HtmlInputText控件	82	5.3.1 向浏览器发送信息	139
4.2.8 HtmlTextArea控件	82	5.3.2 重定向	139
4.2.9 HtmlImage控件	82	5.3.3 输出文本文件	140
4.2.10 HtmlInputCheckBox控件	84	5.3.4 使用缓冲区	140
4.2.11 HtmlInputRadioButton控件	84	5.3.5 检查浏览器联机状态	141
4.2.12 HtmlSelect控件	85	5.3.6 自定义日志记录	141
4.2.13 HtmlInputFile控件	87	5.3.7 在指定时间段显示网页	141
4.3 WEB服务器控件	90	5.4 Request对象	142
4.3.1 Web服务器控件概述	90	5.4.1 获取表单数据	143
4.3.2 Label控件	93	5.4.2 获取客户端浏览器的信息	145
4.3.3 Button控件	93	5.4.3 获取服务器端环境变量	146
4.3.4 HyperLink控件	94	5.4.4 获取当前浏览网页的路径	147
4.3.5 LinkButton控件	95	5.5 Server对象	148
4.3.6 ImageButton控件	96	5.5.1 HTML编码和解码	148
4.3.7 TextBox控件	97	5.5.2 URL编码和解码	149
4.3.8 CheckBox控件	99	5.5.3 执行指定程序	149
4.3.9 CheckBoxList控件	100	5.5.4 路径转换	150
4.3.10 RadioButton控件和RadioButtonList 控件	102	5.5.5 建立新对象	150
4.3.11 Image控件和ImageMap控件	105	5.6 Application对象	151
4.3.12 DropDownList控件	107	5.6.1 存取Application对象变量值	151
4.3.13 ListBox控件	109	5.6.2 锁定Application对象	152
4.3.14 Panel控件	111	5.6.3 Application对象的事件	153
4.3.15 Table控件	111	5.7 Session对象	154
4.4 增强控件	117	5.7.1 设定Session对象变量的有效期限	155
4.4.1 Literal控件	117	5.7.2 在页面之间传递数据	155
4.4.2 Placeholder控件	117	5.7.3 页面访问控制	156
4.4.3 AdRotator控件	118	5.7.4 Session对象事件的应用	156
4.4.4 Calendar控件	120	5.8 Cookie对象	157
		5.8.1 自定义Cookie对象	159

5.8.2 设定Cookie变量的生命周期	160	8.3.1 数据库查询	209
习题	160	8.3.2 数据库操作	213
第6章 文件操作	162	8.4 数据库存储过程	214
6.1 FileSystemObject对象	162	8.5 ADO.NET模型	216
6.2 System.IO命名空间	164	8.6 ADO.NET的结构	217
6.2.1 目录和路径操作	164	8.6.1 ADO.NET数据提供程序	217
6.2.2 文件读写操作	165	8.6.2 ADO.NET对象结构	218
6.3 文件上传	171	8.7 ADO.NET数据库操作	219
习题	172	8.7.1 数据库应用程序结构	219
第7章 网站设计	173	8.7.2 数据库操作过程	219
7.1 母版页	173	8.8 ASP.NET数据库应用程序简例	220
7.1.1 母版页和内容页概述	173	习题	222
7.1.2 创建母版页	174	第9章 数据库编程	223
7.1.3 创建内容页	176	9.1 连接数据库	223
7.1.4 母版页和内容页的运行机制	177	9.2 执行SQL命令	224
7.1.5 访问母版页的控件和属性	179	9.2.1 Command对象	224
7.1.6 动态加载母版页	180	9.2.2 DataAdapter对象	225
7.1.7 母版页应用范围	180	9.3 数据查询	227
7.1.8 缓存母版页	181	9.3.1 通过DataReader对象查询数据	227
7.2 样式表	181	9.3.2 通过DataSet查询数据	232
7.2.1 样式选择符	182	9.4 数据更新	243
7.2.2 样式生成器	185	9.4.1 通过Command对象更新数据库	243
7.3 主题和皮肤	192	9.4.2 通过DataAdapter对象更新数据库	245
7.3.1 主题概述	193	9.4.3 执行存储过程更新数据库	249
7.3.2 创建主题	195	9.5 数据源控件	250
7.3.3 应用主题	196	9.5.1 SqlDataSource控件	251
7.3.4 动态加载主题	198	9.5.2 AccessDataSource控件	253
7.4 网站导航	199	9.5.3 XmlDataSource控件	253
7.4.1 网站地图	199	9.5.4 SiteMapDataSource控件	253
7.4.2 用SiteMapPath控件导航	200	9.5.5 ObjectDataSource控件	254
7.4.3 用Menu控件导航	201	9.5.6 LinqDataSource控件	255
7.4.4 用TreeView控件导航	203	9.6 数据绑定控件	256
习题	204	9.6.1 GridView控件	256
第8章 ADO.NET数据库	205	9.6.2 DetailsView控件	257
8.1 数据库简介	205	9.6.3 FormView控件	258
8.1.1 数据库结构	205	9.6.4 ListView控件	259
8.1.2 数据库管理系统	206	9.6.5 Repeater控件	260
8.2 常用数据库管理系统	206	9.6.6 其他数据绑定控件	262
8.2.1 Access 2007	206	9.6.7 内部数据绑定语法	262
8.2.2 SQL Server 2005	208	习题	263
8.3 SQL语言及其数据库操作	209		

第10章 高级应用	265	10.6.2 配置文件结构	290
10.1 验证控件	265	10.6.3 常用设置	291
10.1.1 客户端验证和服务器端验证	266	10.7 性能优化	292
10.1.2 RequiredFieldValidator控件	266	10.7.1 输出缓存	292
10.1.3 CompareValidator控件	268	10.7.2 数据缓存	293
10.1.4 RangeValidator控件	269	10.8 程序调试与异常处理	294
10.1.5 RegularExpressionValidator控件	270	10.8.1 常见错误类型	294
10.1.6 CustomValidator控件	272	10.8.2 异常处理	294
10.1.7 ValidationSummary控件	273	10.9 ASP.NET AJAX	296
10.1.8 关闭客户端验证功能	274	10.9.1 ASP.NET AJAX概述	296
10.2 用JavaScript处理页面和服务器控件	274	10.9.2 建立ASP.NET AJAX应用程序	297
10.2.1 在控件上直接应用JavaScript	275	10.9.3 ASP.NET AJAX服务器端控件	298
10.2.2 使用Page.ClientScript属性	275	习题	302
10.3 用户控件	277	上机实验指导	303
10.3.1 创建用户控件	277	实验1 网页制作	303
10.3.2 将用户控件添加到ASP.NET网页	278	实验2 VB.NET基本程序设计	305
10.4 XML的应用	278	实验3 VB.NET过程和函数	308
10.4.1 读取XML文档	278	实验4 Web服务器控件的使用	311
10.4.2 写入XML文档	279	实验5 内建对象的应用	314
10.4.3 基于XML的新闻系统	280	实验6 网站设计	316
10.5 Web Service	284	实验7 数据库基本操作	318
10.5.1 创建Web Service	285	实验8 AJAX应用	321
10.5.2 发布Web Service	286	综合应用实例：BBS系统	322
10.5.3 使用Web Service	287	附录A Visual Basic .NET常用函数	339
10.5.4 通过Web Service发布新闻	288	附录B XSCJ数据库各表的样本数据	343
10.6 ASP.NET配置	289	附录C 程序调试	345
10.6.1 配置系统的层次结构	289		

第1章 Web基础知识

因特网 (Internet) 已经深入人心, 它改变了我们的工作、学习和生活方式。而这一切改变应主要归功于WWW。WWW (World Wide Web) 简称Web, 中文名为万维网, 它是Internet最基本、应用最广泛的服务。Web用超链接的方式使用户能非常方便地从Internet的一个站点访问另一个站点, 从而获取丰富的信息。各级政府、企事业单位乃至个人都已经或准备建立自己的Web站点。在这样的背景下, Web站点设计技术越来越受到人们的重视, 技术发展日新月异。Web设计技术分为三个方面: 做网页、建网站和Web编程, 三者又是相互联系的, 前两者是后者的基础。本书编著的主要宗旨是介绍微软Web编程的最新技术——ASP.NET。在讲述ASP.NET编程技术之前, 本章和下一章将概要地介绍Web基础知识和ASP.NET编程环境, 主要讲解做网页和建网站的基本知识, 为学习ASP.NET编程打下基础。

1.1 TCP/IP协议

网络协议是网络中各节点进行通信的规范和准则。Internet使用的网络协议是TCP/IP协议, 凡是连入Internet的计算机都必须安装和运行TCP/IP协议软件。

1.1.1 TCP/IP协议模型

TCP/IP代表一个协议集, 其中最重要的是TCP协议和IP协议。TCP/IP协议模型如图1-1所示, 它包含四个层次: 应用层、传输层、网络层和物理接口层。

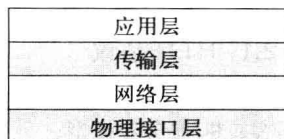


图1-1 TCP/IP参考模型

1) 应用层。应用层是TCP/IP参考模型的最高层, 它向用户提供应用服务。应用层协议主要有: 网络终端协议TELNET, 用于实现互联网中的远程登录; 文件传输协议FTP, 用于实现文件传输; 简单电子邮件协议SMTP, 实现电子邮件收发; 域名服务系统DNS, 用于实现域名到IP地址的映射; 超文本传输协议HTTP, 用于Web浏览器和Web服务器之间传输Web文档。

2) 传输层。也叫运输层, 其主要功能是负责应用进程之间的端到端通信。传输层定义了两种协议: 传输控制协议TCP与用户数据报协议UDP。

3) 网络层。也叫IP层, 负责处理互联网中计算机之间的通信, 向传输层提供统一的数据包。它的主要功能有以下三个方面: 处理来自传输层的分组发送请求, 处理接收的数据包, 处理互连的路径。

4) 物理接口层。它的主要功能是接收IP层的IP数据报, 并通过网络向外发送; 或接收处理从网络上来的物理帧, 抽出IP数据报, 向IP发送。该层是主机与网络的实际连接层。

1.1.2 IP地址和DNS域名

1. IP地址

IP地址是识别Internet网络中的主机及网络设备的唯一标识。每个IP地址通常分为网络地址和主机地址两部分, 长度为4个字节, 由4个用“.”分隔的十进制数组成, 每个数都在0~255之间, 如202.119.106.253。

IP地址分成A~E五大类, 其中常用的是A~C三类。

A类: A类地址用于规模很大、主机数目非常多的网络。A类地址的最高位为0, 接下来的7位为网络地址, 其余24位为主机地址。

B类: B类地址用于中型到大型的网络。B类地址最高两位为10, 接下来14位为网络地址, 其余16位为主机地址。

C类: C类地址用于小型网络。C类地址最高三位为110, 接下来的21位为网络地址, 其余8位为主机

地址。

2. DNS域名

IP地址是连入网络的计算机的地址标识，但对大多数人来说，记住很多计算机的IP地址并不是件容易的事情。所以TCP/IP协议中提供了域名服务系统（DNS），它允许为主机分配字符名称，即域名。在网络通信时由DNS自动实现域名与IP地址的转换。

1.2 Web概述

Web出现于1989年3月，由欧洲粒子物理实验室（European Organization for Nuclear Research, CERN）科学家Tim Berners-Lee发明。1990年11月，第一个Web服务器nxc01.cern.ch开始运行，Tim Berners-Lee在自己编写的Web浏览器上看到了最早的Web页面。1991年，CERN正式发布了Web技术标准。1993年，第一个图形界面的浏览器Mosaic开发成功。1995年，著名的Netscape Navigator浏览器问世。随后，微软推出了著名的浏览器软件IE（Internet Explorer）。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因特网上。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：

- 1) 超文本传输（HTTP）协议，实现万维网的信息传输；
- 2) 统一资源定位符（URL），实现互联网信息的统一定位标识；
- 3) 超文本标记语言（HTML），实现信息的表示与存储。

1.2.1 HTTP协议

超文本传输协议（Hyper Text Transfer Protocol, HTTP）是专门为Web设计的一种应用层协议，采用客户机/服务器模型。

1. Web的客户机/服务器模型

Web以客户机/服务器方式工作。前文所提及的浏览器就是Web客户端程序，它的任务是向服务器发出文档访问请求；Web服务器则存储信息，响应客户端请求，返回客户所要求的Web文档。Web客户机和Web服务器之间的信息传递使用HTTP协议。该模型如图1-2所示。

常用的Web服务器软件有Apache、IIS Web等，Web浏览器软件包括IE、Netscape等。

用户浏览Web页面的方法主要有两种。一种方法是在浏览器的地址栏中键入URL；另一种方法是点击当前访问的Web页面中的超链接。无论采用哪种方法，浏览器都向目标主机（即Web服务器）发送一个HTTP请求。HTTP定义的信息交互处理由以下四步组成：

- 1) 浏览器与Web服务器建立TCP连接；
- 2) 浏览器向Web服务器提出请求；
- 3) 如果请求被接受，则服务器送回响应，在响应中包括状态码和所需的文件；
- 4) 浏览器和Web服务器断开连接。

2. 超文本和超媒体

在浏览器主窗口中显示的Web文档称为Web页面（page），它是一种超媒体（Hypermedia）信息。超媒体信息的基础是超文本（Hypertext）信息。超文本指的是一种电子文档，其中包含有可以链接到其他段落或文档的超文本链接，即超链接（Hyperlink）。利用超链接可以找到连接在因特网上的任何超文本系统，而这些超文本系统分布于因特网上的众多主机上，形成庞大的分布式信息系统。

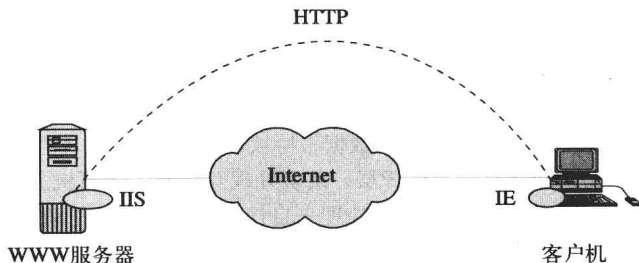


图1-2 Web系统的客户机/服务器模型

超媒体与超文本的区别在于文档内容不同。超文本文档仅包含文本信息，而超媒体文档中除了文本信息外，还包含图形、图像、声音、动画及视频等多媒体信息，其内容更为丰富多彩，表现手段更为多样。

1.2.2 统一资源定位符URL

Web信息分布于全球，要找到所需信息就必须有一种说明该信息存放在哪台计算机的哪个路径下的定位信息。统一资源定位符（Uniform Resource Locator, URL）就是用来标识Web文档的，任何一个文档在因特网范围内具有唯一的URL。

实际上，URL不仅用于标识Web文档，它还用于标识因特网其他类别的文档资源，如FTP、电子邮件文档等，这也是其名称中“统一资源”所表达的含义。URL通过定义资源位置的抽象标识来定位网络资源，格式如下：

<信息访问方式>://<主机>:<端口号>/<文件路径>

其中，<信息访问方式>是访问信息采用的TCP/IP应用协议，最常用的有三种，即HTTP（超文本传输服务）、FTP（文件传输服务）和NEWS（网络新闻服务）。

<主机>是网络主机的域名或IP地址，它指出信息存放的主机。

<端口号>是信息服务采用的软件端口。

<文件路径>是所访问信息的存储路径（通常为虚拟路径而非存储文件的实际路径）。

在URL各部分中，只有<主机>部分是必需的，其余3项均可以省略。若省略<信息访问方式>，则默认采用HTTP协议；若省略<端口号>，则采用TCP/IP标准的保留端口号（如HTTP协议保留端口号是TCP的80）；若省略<文件路径>，则访问该主机的默认文档（如IIS Web默认访问文档为default.htm，或default.html等）。以下是一些URL的例子：

```
http://www.njnet.edu.cn/
http://www.njnet.edu.cn/info/library/calis.htm
ftp://ftp.pku.edu.cn/pub/rfc/rfc2091.txt
news:comp.sys.novell
```

1.3 超文本标记语言HTML

这一节学习编写Web页面的基本语言——HTML语言。

先看一个Web页面：“学生成绩管理系统”主页面，如图1-3所示。

图1-3所示的“学生成绩管理系统”主页面，以表格作为页面的总布局方式，页面设计中使用了常用的HTML标记，包括表格、表单、文字显示控制、加入图片、超链接、水平线、换行、分段及设置页面背景图片等。用记事本打开该页面对应的HTML文档，其内容显示如下：

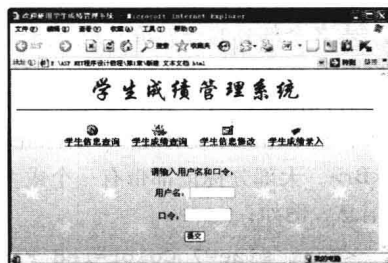
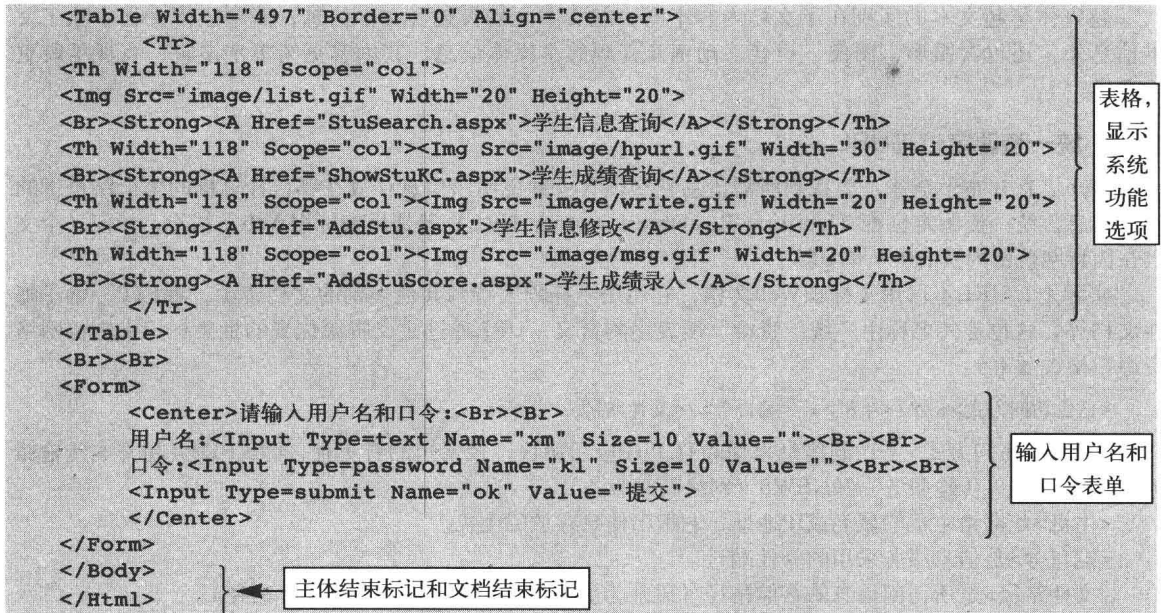


图1-3 “学生成绩管理系统”主页面

```
<Html> ← 文档起始标记
<Head>
  <Meta Http-equiv="Content-Type" Content="text/html; Charset=gb2312">
  <Title>欢迎使用学生成绩管理系统</Title>
</Head>
<Body Background="image/fallb.jpg"> ← 主体标记起始，并设置页面背景图片
  <Div Align="center">
    <Font size=7 Color=red Face="华文行楷">学生成绩管理系统</Font>
  </Div>
  <Hr Color=blue Border="2"><Br> ← 加入水平线，换行
```

图1-3所示的HTML代码片段包含以下标注：

- 文档起始标记：指向 `<Html>`。
- 文档头部标记：指向 `<Head>` 部分。
- 主体标记起始，并设置页面背景图片：指向 `<Body Background="image/fallb.jpg">`。
- 分段，设置标题文字格式：指向 ``。
- 加入水平线，换行：指向 `<Hr Color=blue Border="2">
`。



从这个例子我们看到, HTML文档是一个文本文件, 其中包含HTML标记(tag)和属性形式的指令。HTML文件与平台无关, 可用任何文本编辑器(如记事本)进行编辑。文件的扩展名为.htm或.html。双击HTML文件名即可在浏览器中显示页面内容。

HTML(Hypertext Markup Language, 超文本标记语言)是在万维网上建立超文本文件的语言。它源于SGML(Standard Generalize Markup Language, 标准通用标记语言)的设计概念, SGML是ISO(国际标准化组织)指定的定义标记语言的国际标准。SGML的设计目的是为了使网络上的文档格式统一, 易于交流。SGML的标记, 即英文的tag, 就是在文档需要的地方, 插入特定记号, 来控制文档内容的显示, 这就是文档格式定义。HTML采用SGML的“文档格式定义”的概念, 通过标记和属性对文本的语义进行描述, 并提供由一个文件到另一个文件或在一个文件内部不同部分之间的链接。HTML标记是区分文本各个部分的分界符, 用于将HTML文档划分成不同的逻辑部分(如段落、标题等), 它描述了文档的结构, 与属性一起向浏览器提供该文档的格式化信息以决定Web页面的显示特征。

HTML标记用一对中间包含若干字符的<>表示, 通常成对出现, 前一个是起始标签, 后一个为结束标签, 如<Html>...</Html>、<Head>...</Head>等。但也有部分标记非成对出现, 如上例中出现的换行标记
。大部分标记都带有一个或多个属性, 属性(如果有的话)为浏览器提供执行标记命令所需的附加信息。例如:

```
<Font size=7 Color=red Face="华文行楷">学生成绩管理系统</Font>
```

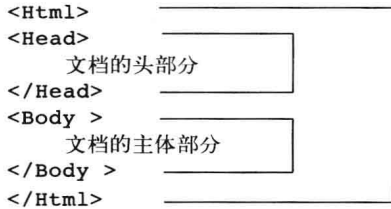
上例中, Font是标记名, 告诉浏览器设置由和与其对应的所界定的文字显示属性, 而Color和Face为属性, 用于设置文字的颜色和字体。

HTML标记和属性不区分大小。例如, 上例也可写成, 等。

有些标记(如上例的Body)还有一些事件。通过设置事件代码, 当该事件产生时, 事件代码便得以执行。事件代码用脚本语言编写。目前, 常用的脚本语言有VBScript和JavaScript。脚本语言编写的程序用Script标签括起, Language属性告知浏览器Script标签括起的脚本是用什么脚本语言编写的。如果用VBScript脚本语言, 则Language="VBScript"或Language="VBS"; 如果用JavaScript脚本语言, 则Language="JavaScript"或Language="JScript"。

1.3.1 HTML文档的基本构成

HTML文档的基本结构如下:



HTML页面以<Html>标记开始，以</Html>结束。在它们之间，就是Head和Body。Head部分用<Head>...</Head>标记界定，一般包含网页标题和文档属性参数等不在页面上显示的网页元素。Body部分是网页的主体，其中的内容均会反映在页面上，用<Body>....</Body>标记来界定。页面的内容主要包括文字、图片、动画及超链接等。

1.3.2 HTML基本标记

1. 文档标记

(1) HTML标记<Html>...</Html>

Html标记表示文档内容的开始和结束，<Html>是开始标记，</ Html >是结束标记，其他所有HTML代码都位于这两个标记之间。浏览器将该标记中的内容看做一个Web文档，按照HTML语言规则对文档内的标记进行解释。< Html >...</ Html>标记是可选的，但最好不要省略这两个标记，以保持Web文档结构的完整性。

(2) 头部标记<Head>...</Head>

在头部标记中提供与Web页有关的各种信息。在头部标记内部，一般使用下列标记：

- <Title></Title>，指定网页的标题；
- <Style></Style>，定义文档内容样式；
- <Script></Script>，插入脚本语言；
- <Meta>，描述网页信息；
- <!-- -->，注释内容。

这些信息首先向浏览器提供，但不作为文档内容提交。

(3) 标题栏标记<Title>...</Title>

在浏览器标题栏中显示的文本。通常，Web搜索工具用它作为索引。

(4) 描述标记<Meta属性=值...>

该标记描述文档属性参数，其属性有以下几种：

- Name，Meta名字；
- Http-Equiv，说明Content属性内容的类别。Http-Equiv="refresh"，则Content中是页面刷新的时间；Http-Equiv="content-language"，则Content中是页面语言；Http-Equiv="PICS-Label"，则Content中是页面内容的等级；Http-Equiv="expires"，则content中是页面过期的日期。
- Content，定义页面内容，一些特定内容要借助Http-Equiv属性定义。例如：

```
<Meta Name="description" Content="Books,Sale,Price">
<Meta Http-Equiv="expires" Content="Dec,30,2005 00:00:00 GMT">
<Meta Http-Equiv="content-language" Content="en">
```

(5) 主体标记<Body>...</Body>

该标记表明HTML文件主体的开始和结束。格式为：

```
<Body 属性=值... 事件=执行的程序...>...</Body>
```

<Body>标记中包含了文档的内容。其常用属性如下：

- BackGround，文档背景图像的URL地址。
- BgColor，文档的背景颜色。
- Text，文档中文本的颜色。
- Link，文档中链接的颜色。

- VLink, 文档中已被访问过的链接的颜色。
- ALink, 文档中正被选中的链接的颜色。

例如:

```
<Body Background="images/back05.gif" Text="#ff0000" Link="yellow" ALink="#00ff00">
```

在HTML中,许多标记都要用到颜色属性,这里,表示颜色属性的值有3种方法:

- 1) 使用颜色名称来表示。例如,红色、绿色和蓝色分别用red、green和blue表示。
- 2) 使用十六进制格式数值#RRGGBB来表示,RR、GG和BB分别表示红、绿、蓝三基色的两位十六进制数据。例如,红色、绿色和蓝色分别用#FF0000、#00FF00和#0000FF表示。表1-1列出了16种标准颜色。

表1-1 标准颜色的名称和十六进制值

颜色	名称	十六进制值	颜色	名称	十六进制值
淡蓝	aqua (cyan)	#00FFFF	海蓝	navy	#000080
黑	black	#000000	橄榄色	oliver	#080000
蓝	blue	#0000FF	紫	purple	#800080
紫红	fuchsia (magenta)	#FF00FF	红	red	#FF0000
灰	gray	#808080	银色	silver	#C0C0C0
绿	green	#008000	淡青	teal	#008080
橙	lime	#00FF00	白	white	#FFFFFF
褐红	maroon	#800000	黄	yellow	#FFFF00

- 3) 使用RGB (r, g, b) 函数表示。r、g、b的数值范围为0~255或者0%~100%。

例如, RGB (0, 0, 9)、RGB (100%, 0%, 100%)。

其常用事件如下:

- OnLoad, 文档首次加载时调用的事件处理程序。
- OnUnload, 文档卸载时调用的事件处理程序。

2. 设置文本格式

文本是HTML网页中的重要内容之一。在编写HTML文档时,可以将文本放置在标记之间来设置文本的格式。文本的格式包括分段与换行、段落对齐方式、字体、字号、文本颜色及字符样式等。

(1) 分段标记<P属性=值...></P>

段落是文档的基本信息单位。当使用分段标记时,文档中原有的回车和换行均被忽略,分段标记定义一个新的段落。

如果单独使用标记<P>会空一行,且后续内容隔一行显示。如果同时使用<P>和</P>,则段落被包围起来,表示一个分段的块。其常用属性如下:

- Align, 段落的水平对齐方式。

其值如下:

- Left, 左对齐 (默认值)
- Center, 居中对齐
- Right, 右对齐
- Justify, 两边对齐

如果省略该属性,则取默认值。下同。

(2) 换行标记

换行标记强制规定当前行的中断,使后续内容在下一行显示。

(3) 标题标记

- <H1属性=值...>...</H1>
- <H2属性=值...>...</H2>
- <H3属性=值...>...</H3>
- <H4属性=值...>...</H4>

<H5属性=值...>...</H5>

<H6属性=值...>...</H6>

标题标记用于设置文档中的标题和副标题，其中，<H1>...</H1>标记表示字体最大的标题，<H6>...</H6>标记表示字体最小的标题。

(4) 居中标记<Center>...</Center>

标记中间的内容全部居中。

(5) 块标记<Div属性=值...>...</Div>

该标记定义文档块。

(6) 水平线标记<Hr属性=值...>

该标记在文档中添加一条水平线，用来分隔文档。常用属性如下：

- Color, 线的颜色。
- NoShade, 显示一条无阴影的实线。
- Size, 线的长度。
- Width, 线的宽度[像素或百分比(占页面宽度的百分数)]。

(7) 字体标记...

该标记用来设置文本的字符格式，主要包括字体、字号和颜色等。常用属性如下：

- Face, 字体名表(字体名之间用","分隔)。

浏览器首先使用字体名表中的第1种字体显示标记内的文本。如果在运行浏览器的计算机上没有安装第1种字体，则会尝试字体名表中的第2种字体，直至找到匹配字体。如果到达列表末尾，仍然找不到匹配字体，浏览器将使用默认字体。

- Size, 字号值。

Size属性指定字体的大小(字号)，其值为1~7，默认值为3。Size值越大，显示的字号就越大。也可以使用+或-号来指定相对字号，表示相对于基本字体(BASEFONT)的大小。

(8) 字体标记

...表示粗体；<Big>...</Big>表示大字体；<TT>...</TT>表示固定宽度字体；<I>...</I>表示斜体；<Small>...</Small>表示小字体。

(9) 样式标记

^{...}表示上标；<U>...</U>表示下划线；<S>...</S>表示删除线；_{...}表示下标；<Strike>...</Strike>表示删除线。

(10) 原样显示标记<Pre>...</Pre>

该标记将包含在其中的内容(包括回车和空格)原样显示出来。

文档中文本之间经常会有一些空格，有时在输入文本时按了多次空格键，但在浏览器中打开网页时却只能看到一个空格。另外，可能要插入一些特殊符号，如版权符号©和注册符号®等。在网页中，这些符号不能直接输入，可以使用字符实体名称或数字表示方式。例如，若要在网页中输入一个空格，可以输入" "或" "，等等。表1-2列出了常用特殊符号的实体名称或数字表示。

表1-2 常用特殊符号的实体名称或数字表示

字符	说明	字符实体名	数字表示	字符	说明	字符实体名	数字表示
	无断行空格	 	 	¥	元符号	¥	¥
¢	美分符号	¢	¢	§	节符号	§	§
£	英镑符号	£	£	©	版权符号	©	©
®	注册符号	®	®	&	"and"符号	&	&
°	度	°	°	<	小于符号	<	<
²	平方符号	²	²	>	大于符号	>	>
³	立方符号	³	³	€	欧元符号	€	€

如果文档中某一段的文本特殊符号较多，就可以放在<Pre>与</Pre>标记中间，这样浏览器就会原样


```
<Ul>
  <Li>列表项1
  <Li>列表项2
  .....
  <Li>列表项n
</Ul>
```

Ul标记控制列表项前面显示的项目符号。常用属性如下：

- Type, 列表项前面显示的项目符号。

其取值如下：

- disc, 使用实心圆作为项目符号（默认值）
- circle, 使用空心圆作为项目符号
- square, 使用方块作为项目符号

注意：

在IE浏览器中，Type属性的值是区分大小写的。

无序列表也是可以嵌套的。若要显示嵌套的无序列表，将相关的列表标记嵌套使用即可。

(3) 描述性列表

<Dl>...</Dl> 是描述性列表标记，它与其子标记<Dt>和<Dd>配合使用，用来控制数据项目的显示，类似于术语及其定义的书写格式。通常，<Dt>标记给出需定义的术语，<Dd>则给出相应的描述。

例如：

```
<Dl>
  <Dt>局域网
  <Dd>局域网是指将小范围内的数据设备经过通信系统连接起来的计算机网络
</Dl>
<Dl>
  <Dt>列表描述项
  <Dd>列表项
  <Dd>列表项
  .....
  <Dt>列表描述项
  .....
</Dl>
```

【例1.2】 创建如图1-5所示的含有有序列表、无序列表和描述性列表的页面。

源代码如下 (Ex1-2.htm)：

```
<Html>
<Head><Title>各种列表示例</Title></Head>
<Body><Center>
<P><Font Color=Red>用数字表示的有序列表:</Font>
<Ol Type="A" Start="1">
  <Li>电子管
  <Li>晶体管
  <Li>小规模集成电路
  <Li>中规模集成电路
  <Li>大规模集成电路
</Ol></P>
<P><Font Color=Red>无序列表:</Font>
<Ul Type = "disc">
  <Li>计算机导论
  <Li>操作系统
  <Li>计算机原理
  <Li>数据结构
</Ul></P>
<P><Font Color=Red>描述性列表:</Font>
```