

21世纪高等学校计算机教育实用规划教材

ASP.NET Web数据库开发 实践教程

饶俊 赵富强 编著



清华大学出版社

013071061

TP393.092
2505

21世纪高等学校计算机教育实用规划教材

ASP.NET Web数据库开发 实践教程

饶俊 赵富强 编著



北航

C1680223

清华大学出版社
北京

TP393.092

2505

130170310

计算机数据库开发实践教程/饶俊,赵富强编著. —北京:清华大学出版社,2013

内 容 简 介

本书是关于 ASP.NET 数据库开发的实践教程。全书共分 10 章。第 1 章介绍了 Web 开发的基本知识,包括 HTML 语言、JavaScript 脚本、基于软件工程的网站开发流程等知识。第 2 章介绍了需求建模过程中常用的一些工具,如数据流图、用例图、类图等,并分析了一个实际项目的需求模型。第 3 章介绍了数据库的设计方法,包括运用 SQL 语言进行数据的查询和更新、E-R 图的绘制方法等。第 4~9 章以微软的 Visual Studio 2010 开发工具为背景,介绍了开发过程中用到的各种具体技术,包括 IIS 的安装与配置、C# 语言、Web 窗体的基本控件、ADO.NET 组件和 Web 窗体的数据控件等必备的 ASP.NET 技术。最后,第 10 章对书中案例的实现进行了完整和详细的说明。

本书设计思路简明、实用,文档详尽,实验步骤明晰,不仅可以作为计算机相关专业本科生和专科生的 ASP.NET 课程实践教材,也可以作为相关开发人员和爱好者的参考用书。

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 13701121933

图书在版编目(CIP)数据

ASP.NET Web 数据库开发实践教程/饶俊,赵富强编著. —北京:清华大学出版社,2013

ISBN 978-7-302-33082-0

I. ①A… II. ①饶… ②赵… III. ①网页制作工具—程序设计—教材 IV. ①TP393.092

中国版本图书馆 CIP 数据核字(2013)第 150988 号

责任编辑:黄 芝 王冰飞

封面设计:常雪影

责任校对:白 蕾

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京市人民文学印刷厂

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:19.75

字 数:484 千字

版 次:2013 年 9 月第 1 版

印 次:2013 年 9 月第 1 次印刷

印 数:1~2000

定 价:35.00 元

产品编号:051909-01

出版说明

随着我国高等教育规模的扩大以及产业结构调整的进一步完善,社会对高层次应用型人才的需求将更加迫切。各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,合理调整和配置教育资源,在改革和改造传统学科专业的基础上,加强工程型和应用型学科专业建设,积极设置主要面向地方支柱产业、高新技术产业、服务业的工程型和应用型学科专业,积极为地方经济建设输送各类应用型人才。各高校加大了使用信息科学等现代科学技术提升、改造传统学科专业的力度,从而实现传统学科专业向工程型和应用型学科专业的发展与转变。在发挥传统学科专业师资力量强、办学经验丰富、教学资源充裕等优势的同时,不断更新教学内容、改革课程体系,使工程型和应用型学科专业教育与经济建设相适应。计算机课程教学在从传统学科向工程型和应用型学科转变中起着至关重要的作用,工程型和应用型学科专业中的计算机课程设置、内容体系和教学手段及方法等也具有不同于传统学科的鲜明特点。

为了配合高校工程型和应用型学科专业的建设和发展,急需出版一批内容新、体系新、方法新、手段新的高水平计算机课程教材。目前,工程型和应用型学科专业计算机课程教材的建设工作仍滞后于教学改革的实践,如现有的计算机教材中有不少内容陈旧(依然用传统专业计算机教材代替工程型和应用型学科专业教材),重理论、轻实践,不能满足新的教学计划、课程设置的需要;一些课程的教材可供选择的品种太少;一些基础课的教材虽然品种较多,但低水平重复严重;有些教材内容庞杂,书越编越厚;专业课教材、教学辅助教材及教学参考书短缺,等等,都不利于学生能力的提高和素质的培养。为此,在教育部相关教学指导委员会专家的指导和建议下,清华大学出版社组织出版本系列教材,以满足工程型和应用型学科专业计算机课程教学的需要。本系列教材在规划过程中体现了如下一些基本原则和特点。

(1) 面向工程型与应用型学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映基本理论和原理的综合应用,强调实践和应用环节。

(2) 反映教学需要,促进教学发展。教材规划以新的工程型和应用型专业目录为依据。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材建设仍然把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现工程型和应用型专业教学内容和课程体系改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材要配套,同一门课程可以有多个具有不同内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材,教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主编。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平的以老带新的教材编写队伍才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机教育实用规划教材编委会

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前言

随着 Internet 的广泛应用,基于 Web 应用的电子商务、电子政务蓬勃发展起来。目前,Web 应用开发的两大主流体系是 Microsoft 公司的 .NET 体系和 Oracle 公司的 J2EE 体系(收购自 Sun 公司)。其中,ASP.NET 是 .NET 体系中 Web 应用开发的主要技术。

ASP.NET 的前身是非常受大家欢迎的 ASP 语言,但 ASP.NET 相比 ASP 在语法和应用上有许多重大的改进。例如,ASP.NET 具有 Web 开发的三层结构。

- 表示层:该层负责处理用户的输入和向用户输出。
- 业务逻辑层:该层是连接上、下两层的“纽带”,封装了用户对数据库的访问请求。
- 数据访问层:该层负责实现连接数据库和对底层数据的查询、更新等细节操作。

与此同时,ASP.NET 保持并发扬了 Microsoft 公司的可视化开发工具所见即所得的特点,将页面设计与后台开发代码分离,大大降低了设计开发网站的复杂度,非常适合 Web 应用开发领域的初学者。

不过,Web 应用开发综合了软件工程、网页设计、数据库原理等多方面的知识,开发者仅掌握某种开发工具无法胜任这一任务。鉴于此,本书首先介绍了 Web 开发的基本知识,包括 HTML 语言、JavaScript 脚本、基于软件工程的网站开发流程、需求建模过程中常用的数据流图、用例图、类图、E-R 图等,以及设计 Web 网站数据库的方法,然后以微软的 Visual Studio 2010 开发工具为背景,通过一个贯穿本书的项目案例介绍了开发过程中用到的各种具体技术,包括 ISS 的安装与配置、C# 语言、Web 窗体的基本控件、ADO.NET 组件和 Web 窗体的数据控件等必备的 ASP.NET 技术。

另外,为了方便教学,在清华大学出版社网站上配有教学课件和应用实例源代码。

本书具备一定的理论性和较强的实用性,通过实例带动知识点的讲解,可以帮助读者深入理解和掌握 Web 开发所需的理论和方法,真正做到学以致用。本书不仅可以作为计算机相关专业本科生和专科生的 ASP.NET 课程实践教材,也可以作为相关开发人员和爱好者的参考用书。

本书具有以下特色:

- (1) 将软件工程理论与开发实践相结合。
- (2) 案例驱动教学。
- (3) 采用较新开发工具 Visual Studio 2010。
- (4) 贴近实际,部分模块甚至可以直接用于实际开发中。

本书第1、2、3、7、10章由饶俊编写,第4、5、6、8、9章由赵富强编写,肖晓青和旷昕同学参与了第6章部分内容的编写,在此对他们表于感谢。

本书编者是教学一线的教师,长期从事相关课程的教学工作,但因水平有限加之时间仓促,书中难免存在不足和错误之处,敬请广大读者批评指正。

编 者

2013年6月

目 录

第 1 章 Web 开发概述	1
1.1 Web 开发基础	1
1.1.1 HTML 语言	1
1.1.2 HTTP 协议	4
1.1.3 CSS 层叠样式表	6
1.1.4 JavaScript 脚本	9
1.1.5 用户界面设计	13
1.2 软件工程基础	16
1.2.1 软件工程的框架	16
1.2.2 软件工程的基本原理	17
1.2.3 软件生命周期	18
1.3 网站开发流程	21
习题 1	22
第 2 章 需求建模	24
2.1 需求问题	24
2.1.1 一个糟糕的案例	24
2.1.2 需求问题产生的原因	26
2.2 需求过程	27
2.2.1 需求层次	27
2.2.2 需求获取	28
2.2.3 需求分析	29
2.2.4 需求说明	30
2.2.5 需求验证	31
2.3 需求模型	32
2.3.1 数据流图	33
2.3.2 状态转换图	35
2.3.3 用例图	36
2.3.4 类图	37
2.3.5 顺序图	40

2.3.6	活动图	41
2.3.7	判定表和判定树	43
2.4	案例分析	44
2.4.1	用户需求	44
2.4.2	需求模型的运用	45
习题 2		49
第 3 章	数据库基础	51
3.1	数据库概述	51
3.1.1	数据库管理系统	51
3.1.2	数据库系统	52
3.2	创建数据库和表	54
3.2.1	创建数据库	54
3.2.2	创建、删除和修改基本表	56
3.3	数据的查询	61
3.3.1	单表查询	61
3.3.2	连接查询	65
3.3.3	嵌套查询	67
3.4	数据的更新	70
3.4.1	插入数据	70
3.4.2	修改数据	71
3.4.3	删除数据	72
3.5	视图	73
3.5.1	定义和删除视图	73
3.5.2	查询视图	75
3.5.3	更新视图	76
3.6	存储过程	77
3.7	数据库的设计	78
3.7.1	设计步骤	78
3.7.2	采用 E-R 方法的数据库概念设计	79
3.7.3	将 E-R 图转换为数据模型	80
3.8	案例分析	82
习题 3		84
第 4 章	ASP.NET 简介	87
4.1	ASP.NET 概述	87
4.1.1	ASP.NET 的发展历史与特性	87
4.1.2	.NET Framework 简介	88
4.1.3	ASP.NET 的文件类型	89

4.2	Visual Studio 2010 集成开发环境	90
4.2.1	安装 Visual Studio 2010	90
4.2.2	主窗口	92
4.2.3	文档窗口	93
4.2.4	工具箱窗口	93
4.2.5	解决方案资源管理器窗口	94
4.2.6	属性窗口	95
4.2.7	错误列表窗口	95
4.2.8	安装 SQL Server 2008 MSE	96
4.3	IIS 的安装与配置	99
4.3.1	IIS 的安装	99
4.3.2	IIS 的配置	101
4.4	案例分析	103
4.4.1	学习目标	103
4.4.2	案例描述	103
4.4.3	案例实施	103
	习题 4	108
第 5 章	C# 语言	109
5.1	C# 语言概述	109
5.1.1	C# 语言的特点	109
5.1.2	创建控制台应用程序	110
5.2	变量和常量	113
5.2.1	变量	113
5.2.2	常量	114
5.3	数据类型	115
5.3.1	值类型	115
5.3.2	引用类型	117
5.4	基本语句	120
5.4.1	条件语句	120
5.4.2	循环语句	123
5.4.3	跳转语句	125
5.5	异常处理	126
5.6	面向对象特性	130
5.6.1	类的定义	130
5.6.2	类的成员	131
5.6.3	构造函数和析构函数	133
5.7	案例分析	134
5.7.1	学习目标	134

5.7.2	案例描述	134
5.7.3	案例实施	134
习题 5		137
第 6 章	Web 窗体的基本控件	139
6.1	标准控件	139
6.1.1	标签控件	140
6.1.2	文本框控件	141
6.1.3	按钮控件	142
6.1.4	超链接控件	144
6.1.5	图像控件	145
6.1.6	单选控件和单选组控件	147
6.1.7	复选框控件和复选组控件	151
6.1.8	列表控件	153
6.1.9	面板控件	156
6.1.10	文件上传控件	157
6.1.11	日历控件	158
6.2	验证控件	160
6.2.1	范围验证控件	161
6.2.2	表单验证控件	161
6.2.3	比较验证控件	162
6.2.4	正则验证控件	163
6.2.5	自定义逻辑验证控件	164
6.2.6	验证组控件	166
6.3	导航控件	166
6.3.1	SiteMapPath 控件	167
6.3.2	TreeView 控件	168
6.3.3	Menu 控件	170
6.4	案例分析	171
6.4.1	学习目标	171
6.4.2	案例描述	171
6.4.3	案例实施	171
习题 6		173
第 7 章	ADO.NET 基础	176
7.1	ADD.NET 概述	176
7.1.1	ADO.NET 组件	176
7.1.2	.NET Framework 数据提供程序	176
7.1.3	DataSet 对象模型	178

7.2	连接数据库	178
7.2.1	连接字符串	179
7.2.2	连接数据库示例	184
7.3	获取数据	184
7.3.1	Command 对象	184
7.3.2	从数据库中获取单个值	187
7.4	检索和填充数据	188
7.4.1	DataReader 类	188
7.4.2	DataAdapter 类	189
7.4.3	DataSet 类	194
7.5	案例分析	198
7.5.1	学习目标	198
7.5.2	案例描述	198
7.5.3	案例实施	198
	习题 7	211
第 8 章	Web 窗体的数据控件	213
8.1	数据源控件	213
8.1.1	SQL 数据源控件	213
8.1.2	Access 数据源控件	219
8.1.3	目标数据源控件	223
8.1.4	LINQ 数据源控件	225
8.1.5	XML 数据源控件	227
8.1.6	站点导航数据源控件	230
8.1.7	实体数据源控件	231
8.2	数据列表控件	236
8.2.1	Repeater 控件	236
8.2.2	DataList 控件	238
8.2.3	GridView 控件	241
8.3	数据绑定控件	245
8.3.1	FormView 控件	245
8.3.2	DetailsView 控件	247
8.3.3	ListView 控件	249
8.3.4	DataPager 控件	253
8.4	案例分析	256
8.4.1	学习目标	256
8.4.2	案例描述	256
8.4.3	案例实施	256
	习题 8	260

第 9 章 其他 ASP.NET 技术	262
9.1 ASP.NET 内置对象	262
9.1.1 Request 对象	262
9.1.2 Response 对象	263
9.1.3 Application 对象	265
9.1.4 Session 对象	266
9.1.5 Server 对象	268
9.1.6 Cookie 对象	269
9.1.7 Cache 对象	270
9.2 Global.asax 配置	271
9.3 ASP.NET 应用程序配置	274
9.4 XML 和 Web Service 简介	276
9.4.1 XML 简介	276
9.4.2 Web Service 简介	277
9.5 AJAX 基础	281
9.5.1 AJAX 简介	281
9.5.2 AJAX 控件及其使用	282
9.6 MVC 基础	284
9.6.1 MVC 简介	284
9.6.2 创建 ASP.NET MVC 应用程序	284
习题 9	290
第 10 章 综合案例的实现	292
10.1 引言	292
10.2 需求概述	292
10.3 系统模块	293
10.4 系统的实现	294
参考文献	304

第 1 章

Web 开发概述

本章主要介绍 Web 开发的一些必备知识,包括超文本标记语言 HTML、超文本传输协议 HTTP、层叠样式表 CSS、脚本语言 JavaScript、用户界面设计原则、软件工程的基础知识以及网站开发的一般流程等。通过本章的学习,能够帮助读者对 Web 开发有一个整体上的了解。

1.1 Web 开发基础

1.1.1 HTML 语言

超文本标记语言(HyperText Markup Language, HTML)是为“网页创建和其他可在网页浏览器中看到的信息”设计的一种标记语言。HTML 被用来结构化信息,例如标题、段落和列表等,也可用来在一定程度上描述文档的外观和语义。1982 年由蒂姆·伯纳斯-李(Tim Berners-Lee)创建,由互联网工程工作小组 IETF 用简化的 SGML(标准通用标记语言)语法进一步发展的 HTML,后来成为国际标准,由万维网联盟(W3C)维护。

HTML 文档最常用的扩展名是 .html,但是像 DOS 这样的旧操作系统限制扩展名为最多 3 个字符,所以 .htm 扩展名也允许使用。现在,.htm 扩展名使用的比较少了,但是仍然受到支持。用户可以用任何文本编辑器或所见即所得的 HTML 编辑器来编辑 HTML 文件。

早期的 HTML 语法规则定义较为松散,有助于不熟悉网络出版的人采用。网页浏览器接受了这个事实,可以显示语法不严格的网页。随着时间的流逝,官方标准渐渐趋于严格的语法,但是浏览器继续显示一些不合乎标准的 HTML。使用 XML 的严格规则的 XHTML(可扩展超文本标记语言)是 W3C 计划中的 HTML 的接替者。虽然很多人认为它已经成为当前的 HTML 标准,但是它实际上是一个独立的和 HTML 平行发展的标准。W3C 目前建议使用 XHTML 1.1、XHTML 1.0 或者 HTML 4.01 标准编写网页,但也有不少网页转用较新的 HTML 5.0 编写(如 Google)。

1. HTML 的发展历史

最早设计出来的 HTML 以纯文字格式为基础,可以使用任何文本编辑器处理,最初仅有少量标记(TAG),因此易于掌握运用。随着 HTML 使用率的增加,人们不再满足在网页上只能看到文字。1993 年,还是大学生的马克·安德生在他的 Mosaic 浏览器中加入 标记,从此可以在 Web 页面上浏览图片。但人们认为仅有图片还是不够的,希望可以将任何形式的媒体加到网页上,因此,HTML 不断地被扩充和发展。

下面介绍 HTML 版本的变迁历史。

- 超文本标记语言(第一版): 在 1993 年 6 月作为互联网工程工作小组(IETF)工作草案发布(并非标准)。
- HTML 2.0: 在 1993 年 6 月作为互联网工程工作小组(IETF)工作草案发布(并非标准)。

HTML 没有 1.0 版本是因为当时有很多不同的版本。有些人认为,蒂姆·伯纳斯-李的版本应该算初版,但这个版本没有 IMG 元素。当时被称为 HTML+ 的后续版开发工作于 1993 年开始,最初被设计成为“HTML 的一个超集”。第一个正式规范为了和当时的各种 HTML 标准区分开来,使用了 2.0 作为其版本号。HTML+ 继续发展下去,但是它从未成为标准。

- HTML 3.0: 1995 年 11 月作为 RFC 1866 发布,在 RFC 2854 于 2000 年 6 月发布之后被宣布已经过时。
- HTML 3.2: 1997 年 1 月 14 日发布,W3C 的推荐标准。

HTML 3.0 规范由当时刚成立的 W3C 于 1995 年 3 月提出,它提供了很多新的特性,例如表格、文字绕排和复杂数学元素的显示。虽然它是被设计用来兼容 2.0 版本的,但是实现这个标准的工作在当时过于复杂,在草案于 1995 年 9 月过期时,标准开发也因为缺乏浏览器支持而中止了。3.1 版从未被正式提出,而下一个被提出的版本是开发代号为 Wilbur 的 HTML 3.2,它去掉了大部分 3.0 中的新特性,但是加入了很多特定浏览器(例如 Netscape 和 Mosaic)的元素和属性。HTML 对数学公式的支持最后成为另外一个标准——MathML。

- HTML 4.0: 1997 年 12 月 18 日发布,W3C 的推荐标准。

HTML 4.0 同样加入了很多特定浏览器的元素和属性,但是同一时间有不少的过时元素和属性标准被“清理”掉,建议不再使用它们。HTML 在未来与 CSS 结合会更好。

- HTML 4.01(微小改进): 1999 年 12 月 24 日发布,W3C 的推荐标准。
- ISO/IEC 15445: 2000(“ISO HTML”): 2000 年 5 月 15 日发布,基于严格的 HTML 4.01 语法,是国际标准化组织和国际电工委员会(IEEE)的标准。
- HTML 5.0: 2008 年 1 月 22 日公布,HTML 5.0 草案的前身为 Web Applications 1.0,于 2004 年被网页超文本技术工作小组(Web Hypertext Application Technology Working Group, WHATWG)提出,于 2007 年被 W3C 接纳,目前仍在不断完善当中。

2. HTML 的工作原理

HTML 网页在浏览器上实际运行时,通常是从上到下逐步刷新整个页面,以至于大多数人都以为网页是在线运行的,其实这是个错误的认识。发明 HTML 的理由之一是加速在线运行的效率与提高数据交换的便利。如果用户想看到一个 HTML 网页,实际上需要经过以下几个步骤:

- (1) 用户在计算机的网页浏览器中输入网址(URL),该网址可以是因特网(Internet)、内部网(Intranet)或本地计算机的一个位置。
- (2) 若使用者输入的网址为 Internet,则浏览器会先将该网页全部下载至用户本地计算机的存储器上(通常存于硬盘的暂存区)。微软的 Internet Explorer 还提供了互联网暂存空

间,使用户查看网页时,与网页相关的文件及图片存储至计算机上的互联网暂存盘文件夹中。

(3) 网页浏览器打开下载的网页文件,依据 HTML 内的描述找到该 URL 内的各项网络资源,然后依次将资源下载到本地计算机。

(4) 网页浏览器依据 HTML 的描述,将已经下载的各种网络资源(如图形、文字)排列成 HTML 网页设计者当初建立该网页时的样式。

如果使用者想要长期存储该网页及该网页的网络资源,可以利用浏览器提供的存储网页的功能进行,该功能运行时,其实用户根本不需要再次从网上下载任何数据,浏览器只是将原先存放于暂存区的网页及已经下载好的网络资源转移至用户指定的长期存储区(可以是硬盘、光盘或闪存盘)。

在网络上,网页的扩展名通常为 .html 或 .htm,但并不是只有这种扩展名的文件才是网页,目前有许多动态网页或交互网页其扩展名就不是 .html 或 .htm,例如 Java 系统的 .jsp、微软系统的 .asp 等。其实这些文件的内容与格式确实与 HTML 不同,但是这种动态网页在浏览器阅读时,会自动产生纯正的 HTML 文档给浏览器,所以浏览器阅读到的是经过服务器转换过的标准 HTML,所以能够顺利展现 HTML 所描述的样式。

3. HTML 的语法

通过几个简单的步骤就可以生成一个 HTML 文件:

第一步:启动记事本。

第二步:输入以下代码。

```
<html>
  <head>
    <title> Title of page </title>
  </head>
  <body>
    This is my first homepage. <b> This text is bold </b>
  </body>
</html>
```

第三步:将这个文件存为“mypage.htm”。

第四步:启动浏览器,在浏览器的“文件”菜单中选择“打开”或者“打开页面”命令,这时会弹出一个对话框。单击“浏览”或者“选择文件”按钮,找到刚才创建的文件“mypage.htm”,选定它然后打开,会看到对话框中有一行地址,例如“C:\Documents and Settings\Administrator\桌面\mypage.htm”。单击“确定”按钮,浏览器就会显示这个页面,如图 1-1 所示。

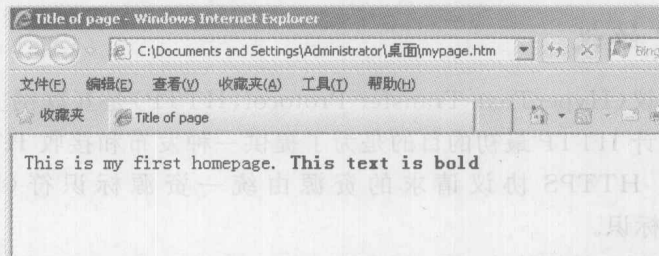


图 1-1 浏览器显示结果

HTML 文档是由 HTML 元素构成的文本文件。HTML 元素是通过使用 HTML 标签定义的,位于起始标签和终止标签之间的文本是元素的内容。

HTML 标签被“<”和“>”符号包围。HTML 标签是成对出现的,例如和。HTML 标签对大小写不敏感,例如和的作用是相同的。

下面就是一个 HTML 元素:

```
<b>此文本是粗体的。</b>
```

HTML 文件“mypage. htm”中的第一个标签是<html>,这个标签告诉浏览器该 HTML 文件的开始点。文件中的最后一个标签是</html>,这个标签告诉浏览器这是 HTML 文件的结束点。在 HTML 文件中,位于<head>标签和</head>标签之间的文本是头信息,头信息不会显示在浏览器窗口中;<title>标签中的文本是文件的标题,标题会显示在浏览器的标题栏中;<body>标签中的文本是将被浏览器显示出来的文本;和标签中的文本将以粗体显示。常用的 HTML 标签见表 1-1。

表 1-1 常用的 HTML 标签

标 签	描 述	标 签	描 述
<a>	定义锚	<label>	定义针对表单控件的标签
	定义粗体字		定义列表的项目
<body>	定义 body 元素	<meta>	定义元信息
 	插入一个回车(折行)		定义有序列表
<div>	定义文档中的节	<option>	定义下拉列表中的选项
<form>	定义表单	<p>	定义段落
<h1>to<h6>	定义标题 1 到标题 6	<script>	定义脚本
<head>	定义关于文档的信息		定义强调文本
<hr>	定义水平线	<style>	定义样式的定义
<html>	定义 HTML 文档	<table>	定义表格
<i>	定义斜体字	<textarea>	定义文本区域
	定义图像	<title>	定义文档的标题
<input>	定义输入域		定义无序列表

HTML 标签拥有属性,属性为 HTML 元素提供附加信息。属性总是以名称/值对的形式出现,形如 name="value"。另外,属性总是在 HTML 元素的开始标签中规定,例如<body bgcolor="yellow">,拥有关于背景颜色的附加信息。

1.1.2 HTTP 协议

超文本传输协议(HyperText Transfer Protocol,HTTP)是互联网上应用最为广泛的一种网络协议。设计 HTTP 最初的目的是为了提供一种发布和接收 HTML 页面的方法。通过 HTTP 或者 HTTPS 协议请求的资源由统一资源标识符(Uniform Resource Identifier,URI)来标识。

HTTP 的发展是万维网协会(World Wide Web Consortium,W3C)和互联网工程工作小组(Internet Engineering Task Force,IETF)合作的结果,他们最终发布了一系列的 RFC,