

高等学校计算机应用人才培养系列教材

HTML网页设计

贾素玲 王强 主编
罗昌 姜浩 许珂 编著

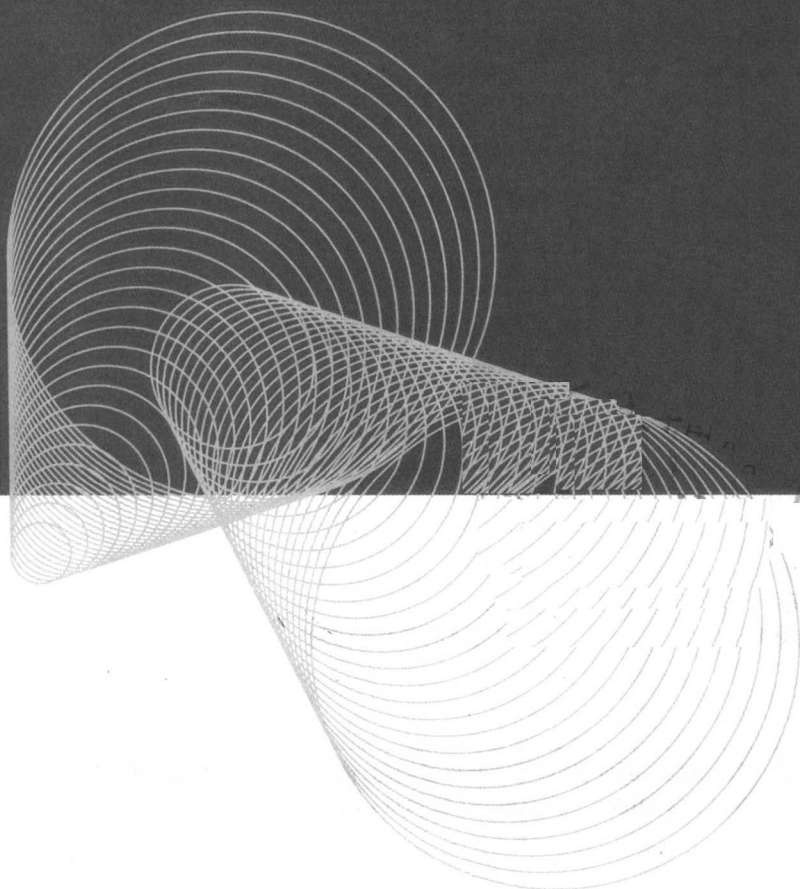


清华大学出版社

高等学校计算机应用人才培养系列教材

贾素玲 王强 主编
罗昌 姜浩 许珂 编著

HTML网页设计



清华大学出版社
北京

内 容 简 介

本书共9章,将从HTML的基本知识开始,力图深入浅出地向读者介绍什么是HTML,以及如何运用HTML来设计网页。主要内容包括:HTML基础知识、创建一个HTML文件、文档格式化、使用图像和背景、表格布局、框架布局、使用表单收集数据、样式表和HTML高级主题。本书每章末皆有小结和习题,可以帮助读者能较快地掌握各章的知识。书中的文字叙述通俗易懂、切入主题、并配有程序代码,能够让读者迅速地掌握HTML网页设计的方法。

本书不仅可作为高等院校计算机专业教材,还适合作为广大网页设计人员和爱好者的参考资料。

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

HTML 网页设计/贾素玲,王强主编;罗昌等编著. —北京:清华大学出版社,2007.5
(高等学校计算机应用人才培养系列教材)

ISBN 978-7-302-14691-9

I. H… II. ①贾… ②王… ③罗… III. 超文本标记语言,HTML、DHTML—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 021296 号

责任编辑:索 梅 林晴佳

责任校对:白 蕾

责任印制:王秀菊

出版发行:清华大学出版社 地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn> 邮 编:100084

c-service@tup.tsinghua.edu.cn

社 总 机:010-62770175 邮购热线:010-62786544

投稿咨询:010-62772015 客户服务:010-62776969

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:14 字 数:338 千字

版 次:2007 年 5 月第 1 版 印 次:2007 年 5 月第 1 次印刷

印 数:1~4000

定 价:21.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:022478-01

打开本套丛书的朋友,如果你抱有获得 IT 应用成功的愿望或想尽快加入令人羡慕的 IT 行业发展的期盼,相信这套丛书将使你豁然开朗。

信息技术是当代人类社会中发展最快、渗透性最强、应用面最广的先导技术,这些日新月异的高新技术不仅改变了世界,也改变了人们的生活。掌握计算机基础知识,提高计算机应用能力,不再是计算机专业人士的特长,并且已经成为许多行业必备的基本技能。计算机技术不仅仅是一门独立的学科,而且日益成为其他学科飞速发展的助推器。对于当代的大学生,或者想进入五彩缤纷的 IT 应用领域的朋友,要跟上信息时代的步伐,就必须掌握现代科学技术,调整自己的知识结构,使自己具备开拓创新的意识和能力,以适应当前社会发展。

信息时代离不开信息化人才,掌握计算机基础知识和提高应用能力,是信息化人才培养的一个重要环节。我们知道,印度是当今的计算机软件出口王国,软件产业的发展成就令人赞叹。之所以如此,除了政府长期实施的一系列扶持政策外,主要还是得益于持之以恒、行之有效的系列化的优秀教材和教育培训。

本套系列教材的形成,一是根据教育部人才培养的指导方针,以培养 IT 应用人才为目标,在引进推广印度 IT 培训教材的同时,借鉴国内外的计算机专业、信息管理专业人才培养的经验,力求课程的设置重点突出、循序渐进,将知识学习与能力培养相结合,使理论与实践完美融合;二是以企业对信息化人才的需求为依据,把面向对象、数据库、软件体系结构、软件工程的思想融入教材体系中,将基本技能的培养与主流应用技术相结合,培养具有扎实的实用型人才;三是在多年从事信息化人才培养和信息系统项目开发的经验基础上,充分理解企业人才需求层次和大学传统人才培养模式的错位,把课程体系的理论知识学习成功转变为应用能力的掌握,使大学真正成为企业的人才资源库。这就是我们开发这套系列教材的最终目标。

计算机基础和语言类的教材,可以说是多如牛毛,那么本套系列教材又凭什么在市场竞争中获得优势呢?

一是知识系列化。本套系列教材以“面向对象、数据库、软件工程、信息系统开发、……、项目管理”的思想为主线,以 Java、XML 为主流技术,形成系列化的能力培养阶梯,使得学生能从一个初学入门者,逐渐成长为合格的 IT 应用技术开发人才。

二是理论实践化。本套系列教材从应用实践的需要入手,合理组织每门课程的结构和内容,在总体框架下,通过大量的案例训练使学生掌握程序设计语言的核心技术和应用技巧,使理论知识在实践中得到升华,在不知不觉中能力得到积累和提高。

三是案例实战化。本套系列教材的编著者既是教师又是软件工程师,具有丰富的教学经验和软件项目开发经验,善于把握计算机技术与学生能力需求之间的尺度,按照循序渐进、突出重点的原则,从多个承担的科研项目中精心抽取和设计教材中引入的案例,使案例

与技术更接近实战的要求。

四是重点内容课件化。除学生用书外,本套系列教材还配有完善的课件,既可以直接为教师上课服务,也能为学生提供本课程的学习要点,引导学生深刻理解每一章节的主体内容,轻松完成相关知识的学习和案例设计。

总之,本套系列教材的指导思想是力求内容新颖、概念清晰、结合实践需要,突出应用能力的培养,使学生在循序渐进的学习中,达到软件项目开发的技术能力要求,成为满足企业信息化需要的人才。

本套系列教材在编写过程中得到了多方专业人士的指导、支持和帮助,在此表示由衷的感谢。尽管我们在教材编著时力求准确,但难免存在不当之处,恳请各位同仁和读者批评指正。

编 者

2007年3月10日



前言

20 世纪是新技术革命的时代,技术的核心是信息的收集、处理和发布。进入 21 世纪,信息技术的发展仍然保持着强劲的势头。其中,万维网起了至关重要的作用。万维网是因特网中提供文字、图片和一些多媒体信息页面的部分,在万维网中建立一个网站就可以将企业的业务展现给全世界的客户,人们可以通过这个媒介进行广泛的交流。

从 20 世纪后半叶开始,人们就在寻找一种有效的万维网信息交流载体,直到 20 世纪 90 年代超文本标记语言(HTML)才成为万维网信息发布的公认载体。经过十多年的发展,HTML 仍然是当今发布在万维网中的所有网页的核心。

HTML 具有完整而规范的定义和标准,同时这些标准比较繁琐和复杂。本书将从 HTML 的基本知识开始,力图深入浅出地向读者介绍什么是 HTML,以及如何运用 HTML 来设计网页。主要包括:

- HTML 基本知识
- 如何创建 HTML 网页
- HTML 文档格式化
- 使用图像美化 HTML 网页
- HTML 中的页面布局
- HTML 中的数据收集
- 利用样式表增强网页表现力
- HTML 高级主题

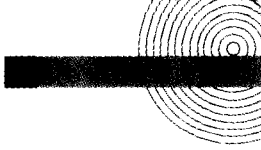
如何应用好 HTML 的各种标记及其属性是每一个网页设计者需要思考的问题。本书力图用最简洁的语言来说明这些问题,并且试图帮助读者掌握 HTML。读者在学习本书之前可以没有任何编程经验,但最好对万维网和因特网已经有了一个感性的认识,并且已经掌握 Windows 的基本操作,以及会使用一些主流的 Web 浏览器,如 Internet Explorer 和 Netscape Navigator。



目 录

第 1 章 HTML 基础知识	1
1.1 HTML 简介	1
1.1.1 HTML 的发展历史	1
1.1.2 HTML 的未来发展	2
1.2 查看和编辑 HTML	4
1.2.1 HTML 编辑器	4
1.2.2 Web 浏览器	4
1.3 HTML 基本术语	7
1.3.1 HTML 标记	7
1.3.2 HTML 元素	7
1.3.3 HTML 属性	8
1.4 HTML 数据类型	8
1.4.1 长度值	9
1.4.2 颜色值	9
1.4.3 URI 地址规范	11
本章小结	11
习题 1	11
第 2 章 创建一个 HTML 文件	12
2.1 使用记事本新建一个 HTML 文件	12
2.2 HTML 的组织规范	15
2.3 HTML 基本标记	16
2.3.1 结构标记<html>	16
2.3.2 头标记<head>	17
2.3.3 文档标题标记<title>	18
2.3.4 内容标记<body>	18
2.3.5 注释标记	20
2.3.6 综合示例	20
本章小结	21
习题 2	21

第 3 章 文档格式化	22
3.1 文字格式化	22
3.1.1 添加标准文字	22
3.1.2 改变文字属性	22
3.1.3 格式化文字	24
3.1.4 特殊字符	26
3.2 超链接和锚站	28
3.2.1 什么是链接	28
3.2.2 锚站元素	28
3.2.3 E-mail 链接	32
3.3 段落格式化	33
3.3.1 段落标记	33
3.3.2 预格式化段落标记	35
3.3.3 标题和对齐	36
3.3.4 换行与空行	38
3.3.5 使用水平线	38
3.4 使用列表	39
3.4.1 有序列表	40
3.4.2 无序列表	44
3.4.3 定义列表	46
本章小结	47
习题 3	48
第 4 章 使用图像和背景	49
4.1 常见网络图像格式	49
4.1.1 GIF 图像	49
4.1.2 JPEG 图像	49
4.1.3 PNG 图像	50
4.2 使用图像	50
4.2.1 在 Web 页中添加图像	50
4.2.2 设定图片高度和宽度	53
4.2.3 设置图片位置	54
4.2.4 设置图片周围空白大小	56
4.2.5 设置边框	57
4.2.6 创建图片链接	58
4.2.7 创建缩略图	59
4.2.8 设置背景图片	61
4.3 创建图像地图	62



- 4.3.1 创建客户端的图像地图 63
 - 4.3.2 创建服务器端的图像地图 66
 - 4.3.3 实际应用中的例子 67
- 本章小结 68
- 习题 4 69

第 5 章 表格布局 70

- 5.1 创建表格 70
 - 5.1.1 基本表格标记 70
 - 5.1.2 设置表格行 70
 - 5.1.3 设置表格列 71
- 5.2 设置表格格式 73
 - 5.2.1 调整表格尺寸和对齐方式 73
 - 5.2.2 设置表格背景颜色 74
 - 5.2.3 在表格中插入图片 75
 - 5.2.4 调整单元格大小 77
 - 5.2.5 设置单元格内容的对齐方式 78
 - 5.2.6 设置单元格间距和边距 79
 - 5.2.7 扩展行和列 80
- 5.3 高级表格标记 84
 - 5.3.1 添加标题 84
 - 5.3.2 行编组 84
 - 5.3.3 列编组 86
 - 5.3.4 表格边框设置 87
- 本章小结 88
- 习题 5 89

第 6 章 框架布局 90

- 6.1 框架的概念 90
- 6.2 创建含有框架的 HTML 文档 90
 - 6.2.1 简单框架示例 90
 - 6.2.2 基本框架标记 93
 - 6.2.3 框架集嵌套 99
 - 6.2.4 框架目标 102
- 6.3 框架格式化 104
 - 6.3.1 边框设置 104
 - 6.3.2 页白设置 105
 - 6.3.3 大小设置 106
 - 6.3.4 滚动设置 107

6.4 浮动框架	108
本章小结	109
习题 6	110

第 7 章 使用表单收集数据 111

7.1 表单初步	111
7.2 表单元素	111
7.2.1 <form> 标记	111
7.2.2 <input> 标记	112
7.2.3 文本域	112
7.2.4 密码域	114
7.2.5 单选按钮	114
7.2.6 复选框	115
7.2.7 下拉菜单	116
7.2.8 多行文本框	119
7.2.9 文件	120
7.2.10 提交和重置按钮	121
7.2.11 图像按钮	122
7.2.12 隐藏域	124
7.2.13 button 值和 button 标记	125
7.2.14 标签标记	125
7.2.15 表单区域 fieldset 和 legend	126
7.3 表单标记访问控制	128
7.3.1 控制 Tab 键的次序	128
7.3.2 快捷键	128
7.3.3 disable 属性和 readonly 属性	129
7.3.4 工具提示与表单域	130
7.4 处理表单	130
7.4.1 method 属性: get 和 post	131
7.4.2 action 属性	132
7.4.3 enctype 属性	132
7.5 设计方法	132
本章小结	133
习题 7	133

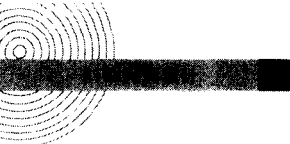
第 8 章 样式表 134

8.1 样式表简介	134
8.1.1 早期 HTML 版本的缺陷	134
8.1.2 样式表概念	134

8.2	使用样式表	136
8.2.1	内部样式表	136
8.2.2	外部样式表	139
8.3	样式表创建规则	141
8.3.1	id 规则	141
8.3.2	class 规则	143
8.3.3	伪类和伪元素	144
8.3.4	继承性	146
8.3.5	情景选择	147
8.4	层叠样式表	148
8.5	样式表属性	149
8.5.1	文本	149
8.5.2	颜色和背景属性	153
8.5.3	字体	158
8.5.4	容器模型	160
8.5.5	列表	164
8.5.6	混合属性	166
8.6	外部和内联样式表示例	169
8.6.1	外部样式表	169
8.6.2	内联样式表	171
	本章小结	171
	习题 8	172

第 9 章 HTML 高级主题 173

9.1	使用多媒体	173
9.1.1	Web 中的多媒体	173
9.1.2	添加活动的 GIF	174
9.1.3	添加音频	174
9.1.4	添加视频	175
9.1.5	添加 Flash 动画	177
9.1.6	<object> 标记	179
9.2	使用 JavaScript 脚本	180
9.2.1	JavaScript 使用目的	180
9.2.2	在 HTML 中包含 JavaScript	180
9.2.3	JavaScript 语言概览	182
9.2.4	常用 JavaScript	186
9.3	区域组件布局	191
9.3.1	<div> 标记的使用	191
9.3.2	 标记的使用	194



9.3.3	<layer>标记的使用	195
9.3.4	三种区域组件的差异	196
9.3.5	区域标记位置设定	197
9.3.6	区域立体空间建立	201
9.4	XML 初步	204
9.4.1	XML 概述	204
9.4.2	XML 基础	205
9.4.3	XHTML: 用 XML 改写 HTML	210
	本章小结	210
	习题 9	211
	参考文献	212

第 1 章 HTML 基础知识

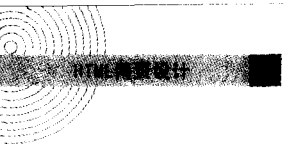
当今的社会是一个信息社会,万维网(World Wide Web, WWW)在其中起到了至关重要的作用。万维网是因特网(Internet)中提供文字、图片和一些多媒体信息页面的部分,人们可通过世界上任何一台接入网络的计算机连接到这些信息页面,也可以通过一个信息页面链接到其他页面,而 HTML 则是这些页面展现信息的最有效载体之一。

1.1 HTML 简介

1.1.1 HTML 的发展历史

HTML(HyperText Markup Language)的中文称为超文本标记语言。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。Ted Nelson 在 1960 年前后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,通用标记语言)。1978—1986 年,在 ANSI 等组织的努力下,GML 语言进一步发展成为著名的 SGML(Standard Generalized Markup Language,标准通用标记语言)标准。当 Tim Berners-Lee 和他的同事在 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 做了大刀阔斧的简化和完善,创建了最初的 HTML,并用它在自己编写的图形化 Web 浏览器上看到了最早的 Web 页面。

在 20 世纪 90 年代初,各个浏览器厂商支持的 HTML 标准各不相同,限制了 HTML 的发展和应用。此后 W3C(World Wide Web Consortium,万维网联盟)网络标准化组织联合主流浏览器开发商一同定义 HTML 标准。HTML 1 于 1992 年发布到 Internet 上,紧接着是 HTML 2、HTML 3、HTML 3.2 和 HTML 4。HTML 发展过程中不断产生新型、功能强大的标记形式,W3C 标准也尽量包容和收纳当前各方的一些实验成果。1997 年推出的 HTML 3.2 极大地丰富了 HTML 的功能,而同年 12 月的 HTML 4.0 将这种语言推向了如



下方向：文档结构和显示样式的分离、更广泛的文档兼容性。现在最新的 HTML 版本是 W3C 在 1999 年 12 月推出的 HTML 4.01，它是 HTML 4.0 的改进版，一直沿用至今。

1.1.2 HTML 的未来发展

1. 模块分离

现在 HTML 语言的一个问题是：结构化 HTML 标记、表现形式和用户事件逻辑都混杂在一起。在许多页面中，HTML 主要用于创建格式，许多 HTML 元素的 style 属性直接被赋予 CSS (Cascading Style Sheets, 层叠样式表，用于为特定系列网页元素创建表现样式的一种规范，详见本书第 8 章) 规则，另外还有很多客户端或服务器端脚本，例如用于装饰网页和提供网页动态特征的 JavaScript 脚本 (详见本书第 9 章)。不过，随着 Web 技术的进步，上述问题都有所改善。比较下面两段代码：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv = "Content-Type" content = "text/html; charset = gb2312">
<title>模块未分离</title>
</head>

<body style = "font-family: Arial,Helvetica,sans-serif;
font-size: 12px;
background-color: #006699;
background-position: center center; " onBlur = 'alert("别离开")'//还有其他代码省略'>

</body>
</html>
```

这种将 CSS 规则直接赋给某一 HTML 元素，并且直接在元素中指定用户事件的做法，使得维护这些页面，或者更新其中某项内容非常困难。必须逐项查找需要修改的地方，如果有多处需要修改，还必须逐一修改。上面这种形式的代码逻辑不清晰，内容不明确。在 HTML 网页中，可以将内容分成几个独立的链接文件，然后对它们进行引用，如下：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<meta http-equiv = "Content-Type" content = "text/html; charset = gb2312">
<title>模块分离</title>
<link rel = "stylesheet" href = "style.css">
<script language = "JavaScript" src = "body.js"></script>
</head>
<body onBlur = "leave()">
...
</body>
```

</html>

上面代码中的 style.css 文件除了可以设置 body 的外观之外,还可以设置文件中其他元素的表现外观:

```
body {
    font-family: Arial,Helvetica,sans-serif;
    font-size: 12px;
    background-color: #006699;
    background-position: center center;
}

textarea {
    font-family: Arial,Helvetica,sans-serif;
    font-size: 12px;
}

select {
    font-family: Arial,Helvetica,sans-serif;
    font-size: 12px;
}
```

链接的 JavaScript 脚本文件 body.js 内容如下:

```
// JavaScript Document
function leave()
{
    alert("别离开");
    //还有其他代码,省略
}
```

通过将 Web 页面的各组成部分分开,更新复杂的内容变得十分容易,这种分离的做法称为模块化设计应用。

2. DHTML、XML 和 XHTML 等的出现和应用

网页上要显示各种动态行为特征,例如网页内容的移动、显示和隐藏,颜色、大小等属性的动态改变等。随着这些需求的出现,产生了 DHTML(Dynamic HTML,动态 HTML)。在某种意义上,DHTML 可以看做 HTML、CSS、JavaScript 和 DOM 模型的综合。DOM(Document Object Model,文档对象模型)是 W3C 提出的一种标准模型,旨在解决不同浏览器支持的脚本的兼容性问题。现在通过 DOM 模型,用户可以控制到 HTML 元素的任何部分和任何属性。不过有些浏览器支持的访问控制方法并没有包括在 DOM 标准中,使用时需要特别注意。

XML(eXtensible Markup Language,可扩展标记语言,详见本书第 9 章)是一种在网络上传输结构化数据的新标准。它也是对 SGML 的改编,允许设计者定义自己的元素,它的最初应用是对 Web 文档的扩展。尽管 XML 现在更主要应用于数据存储介质和程序间消息传递介质,但将来的某一天,人们可能会使用由 XML 定义的新型 HTML——XHTML,甚

至在浏览器中直接使用自己定义的 XML 语言。

2000 年 1 月, XHTML 规范正式发布, 它把 HTML 语言分成各个模块, 并把所有的 XML 规则应用于 HTML 元素。由于将现存大量的 HTML 网页转移到 XHTML 非常麻烦, 并且它的很多规则还没有被主流浏览器厂商所支持, 现在 XHTML 并没有被人们广泛接受。然而, 可以预测的是, XHTML 所提供的结构将成为未来发展的坚实基础和方向。

1.2 查看和编辑 HTML

在学习 HTML 的过程中, 首先要会使用 HTML 编辑器新建或编辑 HTML 文件, 保存之后, 使用 HTML 浏览器查看其显示效果。同样, 也可以使用编辑器打开和编辑已有的网页。

1.2.1 HTML 编辑器

可以使用任一款文字编辑软件来充当 HTML 编辑器, 例如 Windows 操作系统自带的记事本、Microsoft Office Word 等文字编辑工具, 也有专门的 HTML 编辑器, 例如 Macromedia 公司的 Dreamweaver、Microsoft 公司的 FrontPage。专门的 HTML 编辑器比文字编辑工具更专业, 可以用来完成复杂的任务, 并且使用更加方便。一般在专门工具上修改 HTML 代码后, 可以通过它们直接看到显示效果的变化, 这称为“所见即所得”(即 WYSIWYG, What You See Is What You Get)。为了帮助读者更好地学习和理解 HTML 自身的结构和内容, 本书使用 Windows 记事本作为基本编辑器。

1.2.2 Web 浏览器

用于浏览、显示 HTML 网页效果的软件称为浏览器。随着万维网的发展, 出现了多种浏览器。当今使用得最广泛的两种浏览器是微软公司的 Internet Explorer 和 Netscape 公司的 Netscape 浏览器。

Internet Explorer(IE)浏览器以其简便易用性、私密性、灵活性、可靠性而著称, 是使用最广泛的浏览器之一。其界面如图 1-1 所示, 本书所有的 HTML 代码及其效果显示都基于 Microsoft IE 6.0 浏览器。

现在, 微软公司推出了最新的浏览器版本 Internet Explorer 7, IE 7 在易用性、安全性和开发平台方面做了不少改进。例如, 增加了选项卡式浏览、高级打印、快速查找、阅读和订阅 RSS 源等功能。图 1-2 和图 1-3 是 Internet Explorer 7 的界面及其增加的快速选项卡预览功能界面。

Netscape 浏览器是 Netscape 通信公司开发的网络客户器。它虽然是一个商业软件, 但也提供了可在 UNIX、VMS、Macs 和 Microsoft Windows 等操作系统上运行的免费版本。作为成熟浏览器最早的前驱者, 它在 UNIX 用户群中普及率极高。现在 Netscape 公司已经推出了最新的 Netscape Browser 8.1 浏览器。图 1-4 是 Netscape 7.0 浏览器的界面。

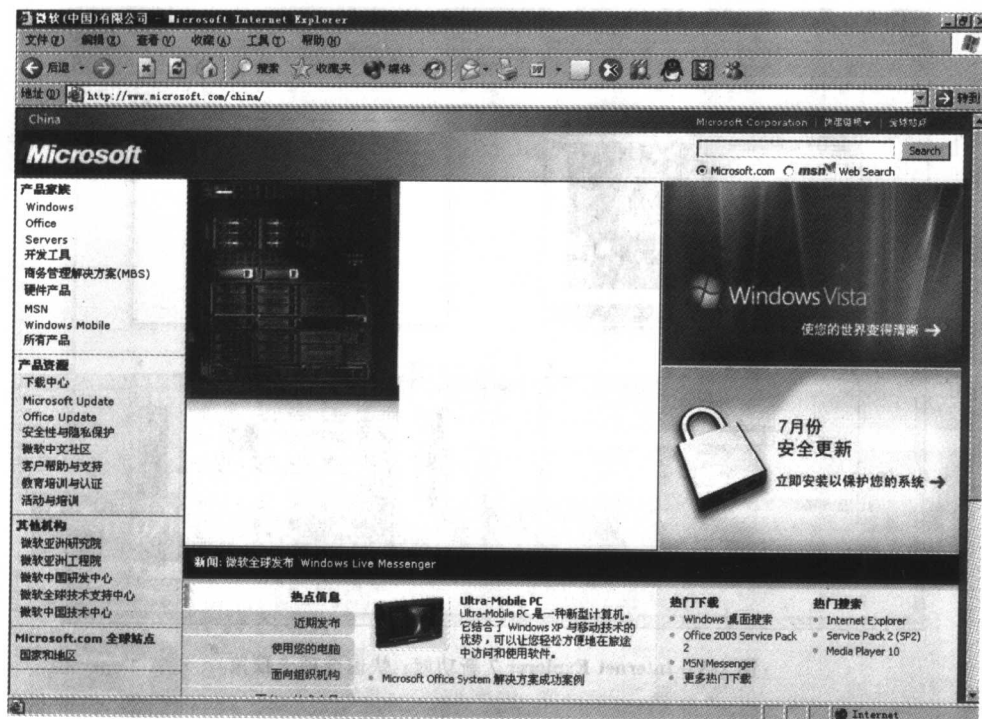


图 1-1 Internet Explorer 浏览器



图 1-2 Internet Explorer 7 浏览器