北大青鸟 APTECH

ACCP FUTURE PROOFED

ACCP 6.0

ACCP软件开发初级程序员

使用HTML语言和CSS开发商业站点

北京阿博泰克北大青鸟信息技术有限公司　职业教育研究院　编著

第一学年 [第一学期]

Accp6.0

科学技术文献出版社

WE CHANGE LIVES

ACCP 软件开发初级程序员

使用 HTML 语言和 CSS 开发商业站点

北京阿博泰克北大青鸟信息技术有限公司　职业教育研究院　编著

第一学年［第一学期］

科学技术文献出版社

Scientific and Technical Documents Publishing House

图书在版编目（CIP）数据

ACCP 软件开发初级程序员学生用书 / 北京阿博泰克北大青鸟信息技术有限公司职业教育研究院编著. —北京：科学技术文献出版社，2011.1

ISBN 978-7-5023-6800-5

Ⅰ. ①A…　Ⅱ. ①北…　Ⅲ. ①软件开发—教材　Ⅳ. ①TP311.52

中国版本图书馆 CIP 数据核字（2010）第 239396 号

书　　名：ACCP 软件开发初级程序员学生用书
策　　划：杨　明
顾　　问：谭浩强
编著单位：北京阿博泰克北大青鸟信息技术有限公司　职业教育研究院
责任编辑：科　文
标准书号：ISBN 978-7-5023-6800-5
地　　址：北京市海淀区成府路207号北大青鸟楼
邮　　编：100871
电　　话：010-62768866
网　　站：http://www.jb-aptech.com.cn
出版发行：科学技术文献出版社
版次/印次：2011年1月第1版　　2011年1月第1次印刷
规　　格：16开（889×1194）　　20印张　480千字
定　　价：500.00元（共5册）

前 言

欢迎来到北大青鸟学习，我们致力于为 IT 企业提供掌握实用技术、能解决实际问题的软件工程师，同时也致力于通过 ACCP 软件工程师职业教育课程，帮助你改善生活现状，实现心中的理想。

当前，人才是困扰 IT 企业发展的第一大问题，企业之间的竞争正在变成人才的竞争。同时，我们也看到，很多计算机专业毕业的本科生却找不到理想的工作。问题出在哪里？针对这种现象，业内专家称之为人才的“结构性过剩”。究其原因，就是因为社会上虽然有大量的“人材”，但是企业需要的却是“人才”，而“人材”并不等于“人才”。

大多数企业在招聘的时候，并不在乎你有多么高深的理论、多么尖端的技术、多么耀眼的学位，而更在乎你能否用最实用的技能解决企业中最实际的问题、做出实际的软件产品和项目，最终能够为企业创造更多的实际价值，这些都是企业在招聘面试时考察的重点。只有能为企业创造出价值的“人材”，才是他们最需要的“人才”。我们的目标就是让更多的“人材”，通过 ACCP 课程的学习和实践，变成炙手可热的软件“人才”，从“我要找工作”，变成“好工作，来找我”！

为此，ACCP 6.0 课程遵循了以下几大原则。

一、以就业为导向——ACCP 6.0 是什么样的课程

以就业为导向是 ACCP 课程最根本、最核心的特征。ACCP 是职业教育，不是学历教育，而职业教育的本质就是就业教育，学习的目标就是为了就业，有助于就业的我们就学，无助于就业的我们就不学。

为了能够满足就业的需求，我们做的第一件事情就是去软件开发企业了解用人状况和技术需求。经过一年多的努力，我们收集了 1147 家企业的招聘信息、对 33 个城市的 214 家企业进行了重点跟踪、与 78 家企业进行了实地接触和深入访谈，通过对掌握的一手数据进行详细分析和多次专家研讨，我们比较全面地掌握了企业用人需求。

时至今日，软件开发企业用人主要有以下几个特征。

- 外包开发行业快速发展，对“人才”在代码和文档方面的规范性、技能和工具的熟练程度要求越来越高。

- Java 和.NET 技术在市场上平分秋色，都有大量的岗位需求，同时值得庆幸的是两者在应用层面上的技术差异越来越小。
- 软件开发企业对开发人员的基本技术素养强调得越来越多，例如：面向对象的程序设计思想和代码组织方法、数据库访问技术、HTML/CSS/JavaScript 客户端技术。
- 为了保证质量和工期，企业中大量使用各种框架技术，要求开发人员至少熟悉一种框架技术。
- MIS、OA、ERP、CRM、系统集成、物流、进销存、电子政务、网站建设这一类 B/S 系统，成为软件工程师需求最大的业务领域。

二、以职业素质为突破点——ACCP 6.0 的训练重心是什么

软件行业的发展使得对软件开发人员的要求也在不断提升，越来越多的企业在注重开发人员技术水平的同时，也更加关注员工在职业素质方面的表现。作为一名职业人，守时、踏实、耐心的习惯，以及适应能力、学习能力、抗压能力、规范意识、合作意识、沟通意识，都是不可或缺的要素。

为此，ACCP 6.0 把职业修养和实战项目训练课程(COT)作为课程的重点来实施，也希望在整个学习过程中，每位学员能够不断反省自己，养成良好的生活和工作习惯，顺利完成从“学校人”到“准职业人”和“企业人”的飞跃。

三、以实用技能为核心——ACCP 6.0 如何选择课程内容

我们知道，软件开发领域内所涉及的技术内容是非常多的，不同的行业会使用不同的技术，不同的项目也会使用不同的技术，在技术选择上我们遵循了实用原则和 80/20 原则，以及更加灵活的授课理念。

- 实用原则：所选择的技术一定是能够解决实际工作中遇到的实际问题的技术。打个比方，假设我们是在学习修车，关于汽车原理、发动机原理、轮胎橡胶特征我们就不会花费精力去学习，因为这些内容在修车过程中很少会用到，而如何使用各种检测设备和螺丝刀等工具以及不同品牌汽车的修理和保养的经验，则是汽车修理人员最迫切需要掌握的技术。因此 ACCP 课程的主要内容也集中在如何解决软件开发项目中所涉及的技术工具、技术框架、开发流程和编码调试经验等方面。请记住：“不要去背诵对象有哪些方法、属性，而是要使用这个对象去解决实际问题”。
- 80/20 原则：80/20 原则包含两层意思，第一是我们重点讲解企业 80%的时间在使用的 20%的核心技术，而那些 80%不常用的非核心技术我们弱化讲解，或者只作为介绍或者拓展内容；第二是花费 80%的精力才能够学会的 20%内容我们不讲解。
- 逐层深入，而非一步到位：我们会对核心技能进行拆分，分散到多门课程或多个学期讲解。
- 随用随讲，而非一次铺开：对于部分技能，第一次讲解时只介绍基本的、常用的技术细节，其他技术细节将在案例、项目中需要时再讲解。

总之，以用为本、学以致用、不用不学、学了就用、综合运用是 ACCP 6.0 选择课程内容的最终标准。

四、以案例为主线——ACCP 6.0 如何组织课程内容

与传统的教材编排方式不同，ACCP 6.0 内容的安排是基于案例和项目。在每一门课程中，都采用了一个实际的开发案例来组织技术内容，理论课上老师会通过一个案例或小项目的讲解来引出技术内容，而上机课则要求学员自己逐步完成一个类似的项目。

在案例的选择上，我们选择与企业应用相近、实用性更强的内容，同时也尽可能地提高案例的趣味性，并加强与日常生活中遇到的问题和现象的联系，从而帮助学员理解案例内容。

通过这种案例教学的方式，学员不会迷失在浩如烟海的知识中，同时会具备更多的行业知识和项目经验。

五、以边讲边练为训练思路——ACCP 6.0 如何训练技能

ACCP 6.0 与 ACCP 5.0 一个显著的区别就在于，技能讲解与技能练习的衔接更加合理。在授课过程中根据技能点进行划分，讲解后即刻进行练习，从而加深印象，提高学员的学习效果。

通过在有限的授课时间内，合理地将技能点的讲解与练习融合到一起，充分利用课时，在边讲边练的过程中，对每一个技能点做到深刻理解，并能灵活运用。

六、以动手能力为合格目标，以项目经验为学习目标——怎样才能学好 ACCP 课程

ACCP 6.0 课程最终的学习目标是："能够运用 Java 或者.NET 技术，在团队合作的基础上，参与到整个软件项目开发过程，熟练、规范地完成企业级 B/S 系统模块的编码、调试和文档编写，成为企业紧缺的技能型应用人才。"这长长的一句话再浓缩成四个字，就是"项目经验"，学习完 ACCP 6.0，相当于具备一年的实际软件项目开发经验，这也是目前绝大多数软件企业对应聘者的最低要求。

学习的目标改变了，学习的内容也改变了，必然要求学习方法也相应调整和改变。以往的死记硬背不再适合于 ACCP 6.0 的学习，ACCP 6.0 需要的是通过"动手"来训练你的技能，不但要快，还要规范。企业需要的不是夸夸其谈的纸上谈兵，而是能够实实在在地解决问题的实干者。为此，在进行 ACCP 6.0 课程学习时，要敢于编码、乐于编码、大量编码才能够达到熟练的程度。

现代企业要求合作，代码规范是团队开发的基础，没有人能看得懂的代码，不管自己觉得写得如何漂亮，终将被称之为"垃圾代码"。在以后的学习中，要记住："代码运行通过了，不算代码写完了，代码规范了才算。"

只有通过大量的、反复的动手实践，才能具备熟练、规范的编码和调试能力，具备真正实用的技能，才有可能被称之为应聘者中"有项目开发经验的人"，成为一名企业真正需要的"人才"，从而在日益激烈的职业竞争中拔得头筹。

作为 IT 职业教育领域内的先行者，依托北京大学雄厚的师资和文化底蕴，依托北大青鸟、APTECH 最优秀的职业教育专家，联合 IT 行业内顶级的技术专家和人力资源专家，在总结了之前五个版本、十余年成熟经验的基础上，我们终于完成了 ACCP 6.0 的

产品研发和教材编写，在这里对支持过我们的中心、院校、行业和教育专家们表示深深的感谢。

我们清楚地知道，我们正在做一件很有价值的事，那就是“教育改变生活”(We Change Lives)。我们也时刻提醒自己：责任重于泰山、细节决定成败。我们衷心希望，通过一年多勤奋的学习，你能借助青鸟的翅膀，翱翔于 IT 的天空，成就心中理想，实现人生价值，回报所有爱你和你爱的人。

北京阿博泰克北大青鸟信息技术有限公司　职业教育研究院

课程说明

随着 Internet 网络的盛行，网站逐渐成为政府办公、企业宣传、信息资讯等方面的主流媒体。好的网站设计，给浏览者的感觉是在和网站进行愉快的交流。因此，如何吸引客户眼球、增加客户用户体验自然成为客户端技术的关键。这包括如何设计网站的整体架构、如何进行页面布局、如何制作网页特效、如何增加用户体验等技术。

Web 客户端技术在整个 ACCP 6.0 课程中的分布如下。

第一学期的 HTML 网页设计侧重于静态商业网站的制作，第二学期将讲解 JavaScript 脚本实现网页高级编程，为后续的第二学期 JSP 课程以及第二学年的企业级应用开发课程打基础。整个课程系列将使用 HTML 进行页面设计、使用 JavaScript 编写脚本，最终制作出界面美观大方、具备动态效果、面向企业应用的商业级网站，第二学年将讲解 Ajax 无刷新技术，实现从服务器端快速更新和提取数据的高级功能。

本课程分为两个阶段：手写 HTML 和使用 Dreamweaver 工具制作网页，总体思路如下。

第一部分（第 1～第 8 章）：手写 HTML，讲解常用的 HTML 标签和基本的 CSS 样式，训练企业要求的手写 HTML、读懂并修改 CSS 的能力。

第二部分（第 9～第 12 章）：使用 Dreamweaver 工具制作网页，以贵美购物商城为实战案例，完成一个完整网站的设计、制作、测试及发布。

第三部分（第 13 章）：项目案例，对 HTML 和 CSS 进行综合训练。

本课程采用边讲边练的方式，你将在学习过程中完成贵美商城网站的制作，最后在项目实战中综合运用所学技能完成六人行网站的设计制作。课程学习中你还有机会参加 COT 活动——网页设计大赛，亲身体验完整网站开发的流程，积累网站设计和开发的经验。

本课程的重点是手写 HTML 标签、CSS 基本样式，这已成为企业面试的必考题。唯一的学习技巧就是多练、多看其他网站，在工具自动生成的代码基础上多修改。学习完成后，相信你也会自行设计、制作个人的 QQ 空间。

目 录

第 1 章

HTML 的基本标签（一）

◈本章工作任务

- 制作图文并茂的商品介绍页

◈本章技能目标

- 使用 HTML 的基本结构创建网页
- 使用行级和块级标签组织页面内容
- 使用图像标签实现图文并茂的页面

本章单词

请在预习时学会下列单词的含义和发音，并填写在横线处。

1. head: _____
2. title: _____
3. body: _____
4. input: _____
5. table: _____
6. form: _____
7. span: _____
8. strict: _____
9. transitional: _____
10. frameset: _____

就业技能结构图

对一些企业所做调研显示，企业承接的 B/S（Browser/Server，浏览器/服务器）架构的项目较多，这些软件项目的表现形式为网站，并且企业、最终用户都很看重 Web 页面能否吸引用户眼球或界面是否方便用户使用。用户看到的 Web 页面主要涉及客户端技术，客户端方向常用的技能如图 1.1 所示，展开的节点部分为本课程训练的主要技能。客户端方面相关课程的安排为：第一学年的第一学期学习 HTML+CSS 课程，侧重于静态商业网站的设计和制作；第二学期学习 JavaScript 课程，侧重于网页交互特效和表单验证。第二学年开设 JQuery、AJAX 等框架技术课程，侧重于高级交互特效的制作。本门课程将边学边完成贵美商城的网站设计和制作，学完本课程后学员还将独立完成一个项目——六人行交友网站。

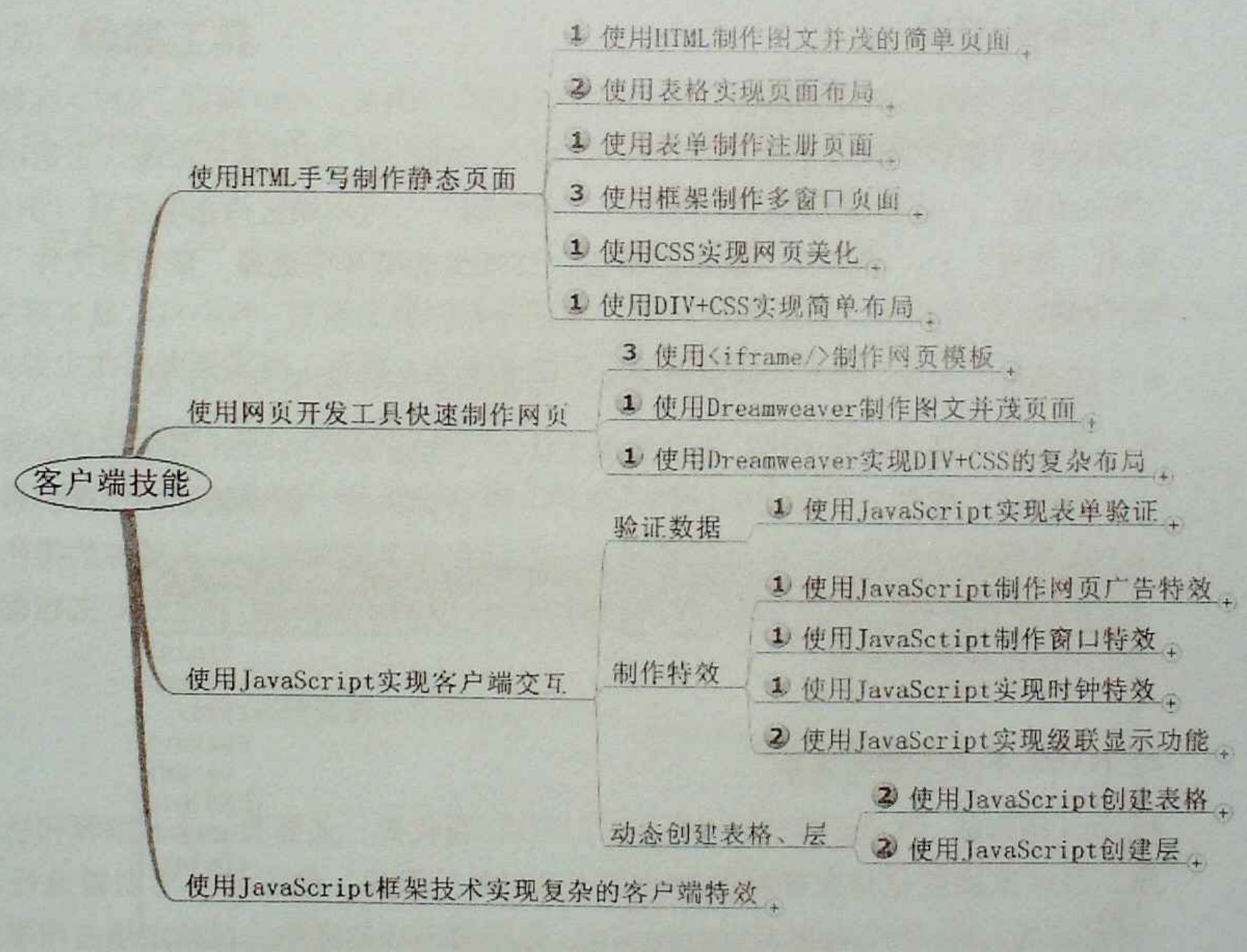


图 1.1　HTML+CSS 就业技能结构图

本章简介

在当今社会中，Web 页已成为网络信息共享和发布的主要形式，而 HTML（Hyper Text Markup Language，超文本标签语言，有的教材也翻译为超文本标记语言）是创建 Web 页的基础。本章将介绍 HTML 文件的基本结构、组成 HTML 文档的各类常用标签以及相关标准。本章重点是各类标签的基本语法，多练习是记住这些标签及其语法的最好方法。

1.1 HTML 文件的基本结构

要深入学习 HTML，首先我们来了解一下什么是 HTML 以及 HTML 和浏览器的关系。

1. 什么是 HTML

HTML 被称为超文本标签语言，它包括很多标签（例如：<p>段落、<h1>标题 1），告诉浏览器（例如微软公司的 Internet Explorer，简称 IE）如何显示页面，它是网页制作的“核心语言”。如希望查看某个页面的“源代码”，可以通过点击浏览器（例如：IE）的菜单“查看”→“源文件”或右键单击，在弹出的菜单中选择“查看源文件”。它具备如下特点。

- 简易性：各类 HTML 标签简单易学，方便网站制作者学习、开发。
- 平台无关性：这是 HTML 语言的最大优点，也是当今 Internet 网络盛行的原因之一。它包括“硬件”平台无关性和“软件”平台无关性，即不管你的计算机是普通的个人电脑机，还是用于专业的苹果机（专门用于制图领域）；不管你的操作系统是常见的 Windows，还是高端 UNIX 或 Linux（一般用于服务器），HTML 文档都可以得到广泛的应用和传输。

2. HTML 和浏览器的关系

和 Java 语言的 JDK 运行环境一样，有了 HTML 源代码，还需要一个“解释和执行”的工具，而浏览器就是用来解释并执行显示 HTML “源码”的工具。早期曾流行 NetScape（网景公司）的 NetScapte Communicator，但随着市场的竞争，目前市场占用率占绝对优势的是微软公司的 IE，以及 Mozilla 公司的 Firefox（也称火狐），本教材使用 IE 8.0 和 Firefox 3.5 进行介绍，关于浏览器的其他介绍学员可以上网获取更多资料。

下面将介绍 HTML 文档的基本结构和各类标签的用法。

1.1.1 基本结构

HTML 的基本结构分两部分，如图 1.2 所示，整个 HTML 包括头部（Head）和主体（Body）两部分，头部包括网页标题（Title）等基本信息，主体包括网页的内容信息（如图片、文字等）。注意标签都以“< >”开始、“</ >”结束，要求成对出现，并且标签之间要有缩进，体现层次感，以便阅读和修改。

```
<html>
<head>
  <title>我的第一个网页 </title>
</head>
<body>
      Hello World!
</body>
</html>
```

头部部分
HTML
主体部分

图 1.2 HTML 代码结构

1.1.2 编辑工具

了解了 HTML 文档的基本结构后，下面介绍常用的 HTML 代码编辑工具。

1. 记事本

记事本是 Windows 自带安装的编辑附件，使用简单方便，实际项目开发中也常用于代码较少的编辑或维护。使用记事本编辑 HTML 文档的步骤如下。

(1) 在 Windows 中打开记事本程序。

(2) 在记事本中输入 HTML 代码，如图 1.3 所示。

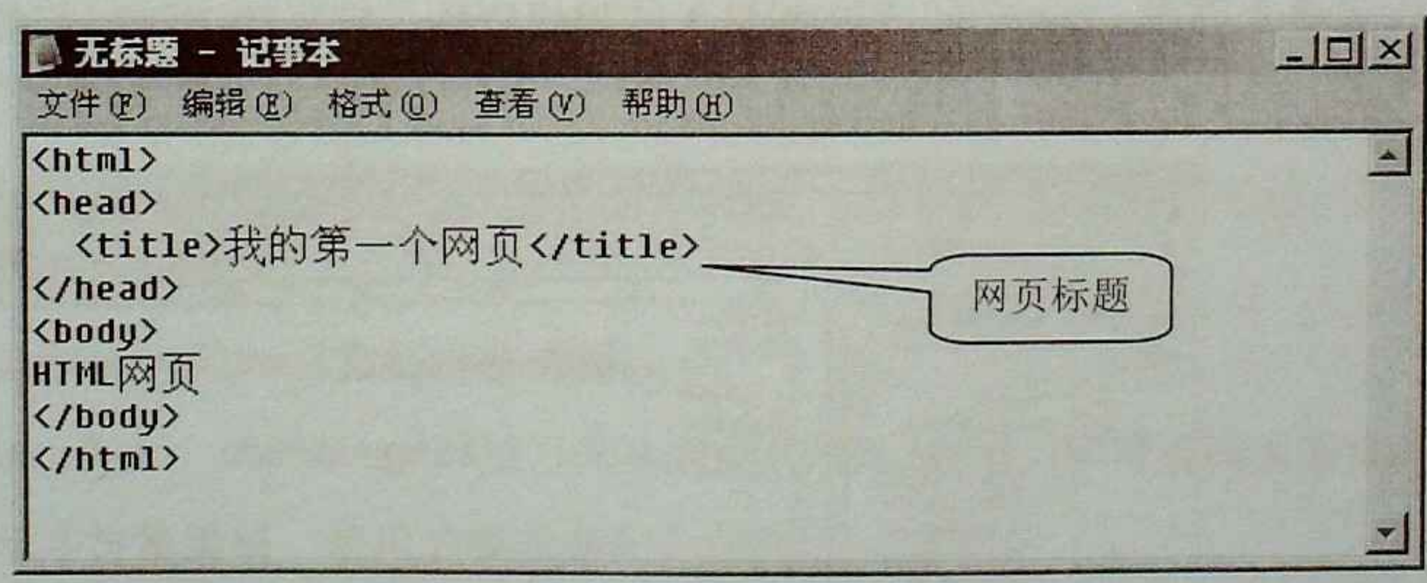


图 1.3 在记事本里编辑 HTML

(3) 单击菜单“文件”→“保存”，弹出“另存为”对话框，如图 1.4 所示，将上述文档保存为后缀为*.htm 或*.html 的 HTML 文档，如 my_firstPage.html。需要注意，因记事本默认保存的文件后缀名为“*.txt”，所以需要用英文的双引号（“”）将文件名括起来。

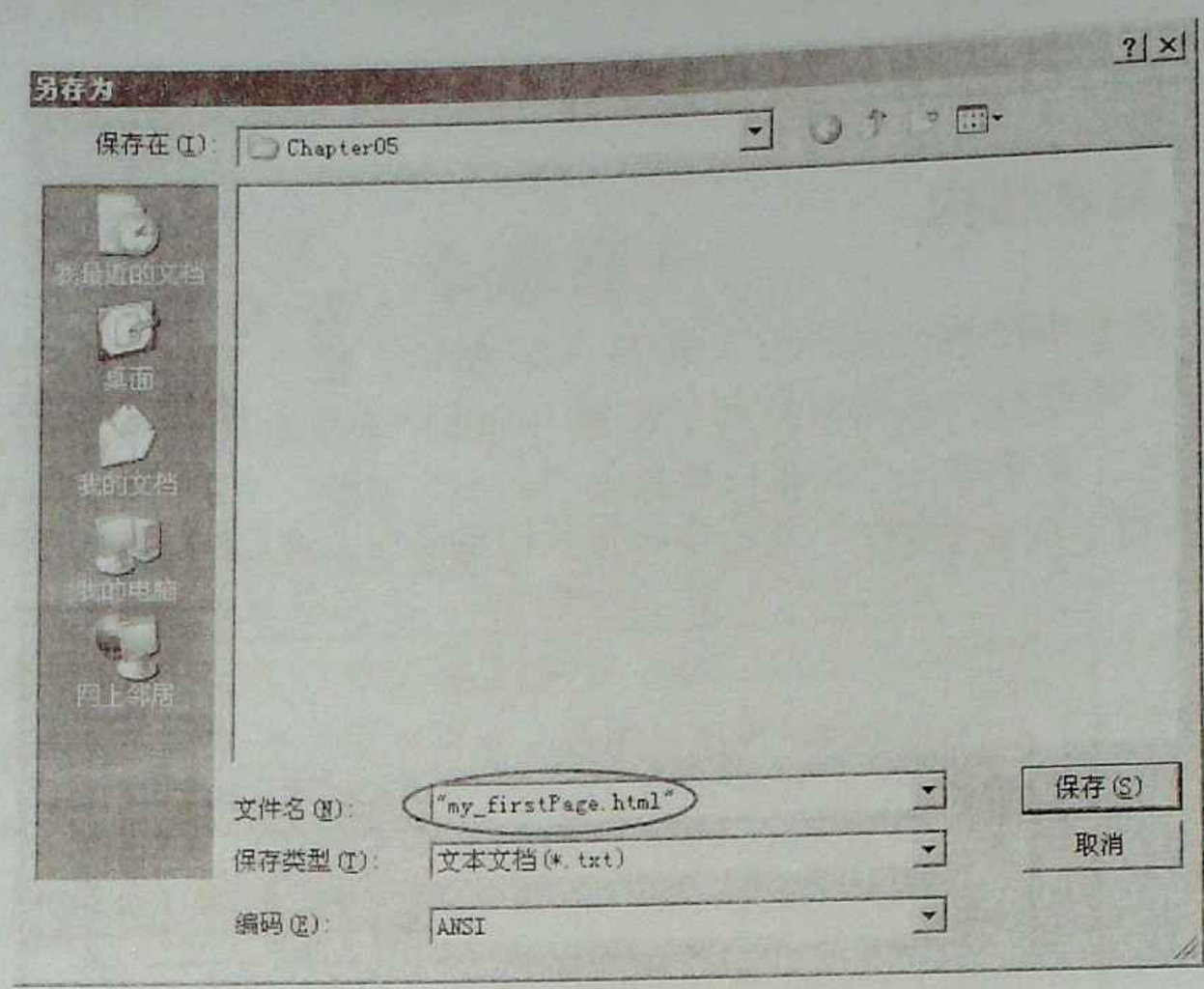


图 1.4 “另存为”对话框

(4) 双击保存的 HTML 文档，Windows 将自动调用浏览器软件（如 IE）打开 HTML 文档，如图 1.5 所示。

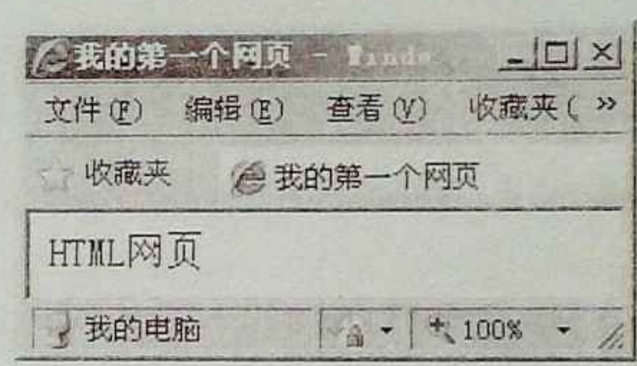


图 1.5 我的第一个网页

2. UltraEdit

相比记事本而言，UltraEdit 是功能强大的编辑软件，它支持 HTML 标签的颜色标识、代码缩进、搜索等功能，如图 1.6 所示。

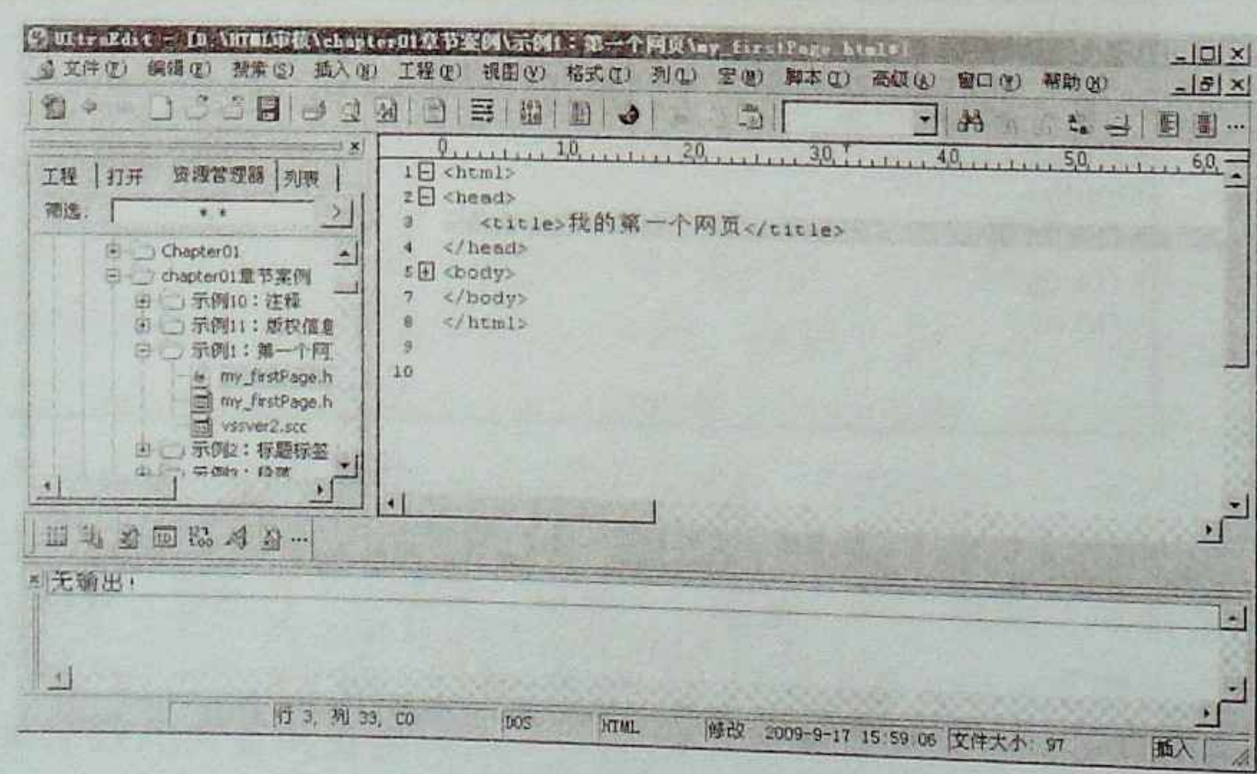


图 1.6 手写利器 UltraEdit

其编辑 HTML 文档的步骤和记事本相同。为了方便学习 HTML 的各类标签，从本章到第 4 章，我们都将采用 UltraEdit 作为 HTML 文档的编辑工具。

1.1.3 网页的摘要信息

网页一般包含大量的文字及图片等信息内容，和报纸或论文一样，它需要一个简短的摘要信息，方便用户浏览和查找。尤其是在当今网络信息“爆炸”的时代，如果希望自己发布的网页能被百度、Google 等搜索引擎搜索，或提高在搜索结果中的排名，那么在制作网页时更需要注意编写网页的摘要信息。

网页的摘要信息一般放在 HTML 文档的头部（Head）区域内容，主要通过如下两个标签进行描述。

1. <title>标签

使用该标签描述网页的标题，类似一篇文章的标题，一般为一个简洁的主题，并能吸引读者有兴趣读下去。例如：sohu 网站的主页，对应的网页标题为：

```
<title>搜狐-中国最大的门户网站</title>
```

打开网页后，将在浏览器窗口的标题栏显示网页标题。

2. <meta>标签

使用该标签描述网页的具体摘要信息，包括文档内容类型、字符编码信息、搜索关键字、网站提供的功能和服务的详细描述等。<meta>标签描述的内容并不显示，其目的是方便浏览器解析或利于搜索引擎搜索，它采用“名称/值”对的方式描述摘要信息。

1）文档内容类型、字符编码信息

对应的 HTML 代码为：

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
```

其中属性“http-equiv”提供“名称/值”中的名称，“content”提供“名称/值”中的值，所以上述 HTML 代码的含义如下。

- 名称：Content-Type（文档内容类型）
- 值：text/html；charset=gb2312（文本类别的 html 类型，字符编码为简体中文）

charset 表示字符集编码，常用的编码有如下四种。

- gb2312：简体中文，一般用于包含中文和英文的页面。
- ISO-885901：纯英文，一般用于只包含英文的页面。
- big5：繁体，一般用于带有繁体字的页面。
- utf-8：国际通用的字符编码，同样适用于中文和英文的页面。和“gb2312”编码相比，utf-8 的国际通用性更好，但字符编码的压缩比较低，对网页性能有一定的影响。

这种字符编码的设置效果，就类似于在 IE 浏览器中，单击菜单“查看”→“编码”给 HTML 文档设置不同的字符编码。需要注意，不正确的编码设置，将带来网页乱码。

2）搜索关键字和内容描述信息

对应的 HTML 代码如下：

```
<meta name="keywords" content="北大青鸟,IT 培训" />
<meta name="description" content="北大青鸟是国内最大的 IT 教育集团,致力于为中国培养优秀的 IT 技术人才。" />
```

实现的方式仍然为“名称/值”对的形式，其中 keywords 表示搜索关键字，“description”表示网站内容的具体描述。通过提供搜索关键字和内容描述信息，方便搜索引擎的搜索。

经验总结

在团队开发中，需要注意养成良好的习惯。书写顺序应该成对书写，从外层写到内层，结构清晰，利于阅读和调试排错。例如编写：

```
<html>
  <head>
    <title>标题</title>
  </head>
</html>
```

推荐顺序如下。

（1）<html></html>。

（2）写入 head 再排版缩进（默认缩进为 2 个空格或 4 个空格，各个团队的具体规定不一样）。

```
<html>
  <head></head>
</html>
```

（3）title，先写标签，再写里面的内容。

```
<html>
  <head>
    <title></title>
  </head>
</html>
```

（4）同理再编写<body>中的其他标签。

1.2 块级标签

介绍了头部（Head）的常用标签后，下面再介绍主体（Body）内常见的各类标签，如图 1.7 所示，从页面布局和显示外观的角度看，一个页面的布局就类似一篇报纸的排版，需要

分为多个区块，大的区块再细分为小区块。块内为多行逐一排列的文字、图片、超链接等内容，这些区块一般称为块级元素；而区块内的文字、图片或超链接等一般称为行级元素。页面这种布局结构，其本质上是由各种 HTML 标签组织完成的。因此，本文将 HTML 标签分为相应的块级标签和行级标签（有的教材也称为块级元素和行级元素），以方便理解后续章节讲解的页面布局。

图 1.7　页面中的块级元素和行级元素

顾名思义，块级标签显示的外观按“块”显示，具有一定的高度和宽度，例如 块标签、<p>段落标签等；行级元素显示的外观为按“行”显示，类似文本的显示，例如<img>图片标签、<a>超链接标签等。和行级标签相比，块级标签具有如下特点。

- 前后断行显示，如图 1.8 所示，块级标签比较“霸道”，默认状态下占据一整行。
- 具有一定的宽度和高度，可以通过设置块级标签的 width、height 属性来控制。
- 块级标签常用作容器，即可以再“容纳”其他块级标签或行级标签，而行级标签一般用于组织内容，即只能用于“容纳”文字、图片或其他行级标签。

从页面布局的角度，块级标签又可细分为基本块级标签和常用于页面布局的块级标签。

1.2.1　基本的块级标签

基本的块级标签包括标题标签、段落标签及水平线标签。

1. 标题标签 <h1>～<h6>

标题标签表示一段文字的标题（主题），并且支持多层次的内容结构。例如，一级标题

采用<h1>，如还有二级标题则采用<h2>，其他以此类推。HTML 共提供了六级标题，并赋予了标题一定的外观：所有标题字体加粗，<h1>字号最大，<h6>字号最小。例如，如图 1.8 所示的各级标题对应的 HTML 代码如示例 1 所示。

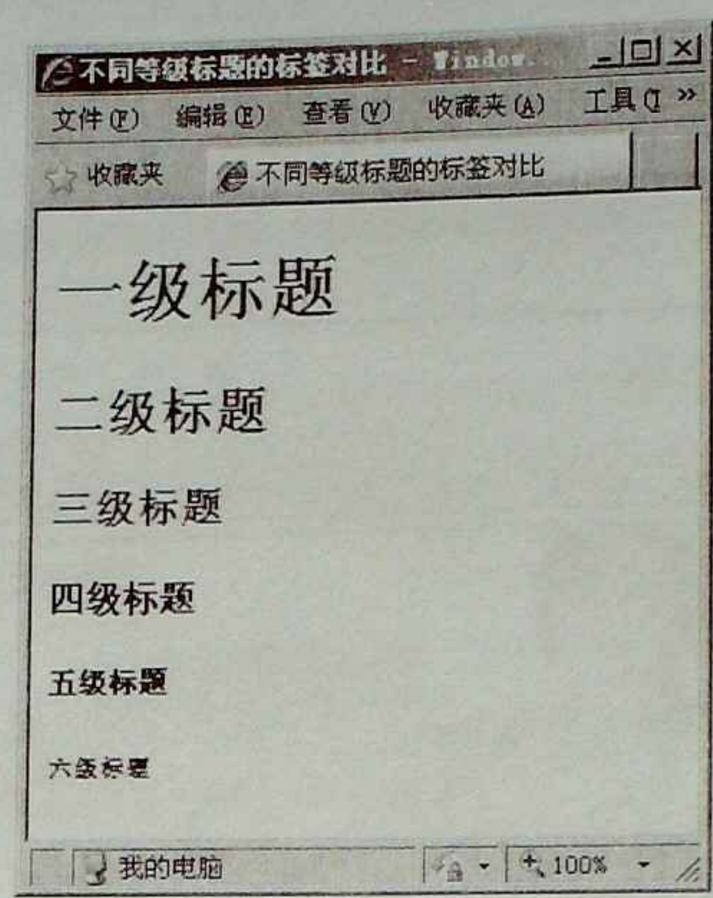


图 1.8 不同级别的标题输出的结果

示例 1

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <title>不同等级标题的标签对比</title>
</head>
<body>
  <h1>一级标题</h1>
  <h2>二级标题</h2>
  <h3>三级标题</h3>
  <h4>四级标题</h4>
  <h5>五级标题</h5>
  <h6>六级标题</h6>
</body>
</html>
```

示例 1 在浏览器中的预览效果如图 1.8 所示。

2. 段落标签<p>

顾名思义，段落标签表示一段文字的内容。例如，希望描述“北京欢迎你”这首歌，如图 1.9 所示，包括歌名（标题）和歌词（段落），则对应的 HTML 代码如示例 2 所示。需要注意，本例的段落只包含一行文字，实际上，一个段落中可以包含多行文字，文字内容将随浏览器窗口的大小自动换行。

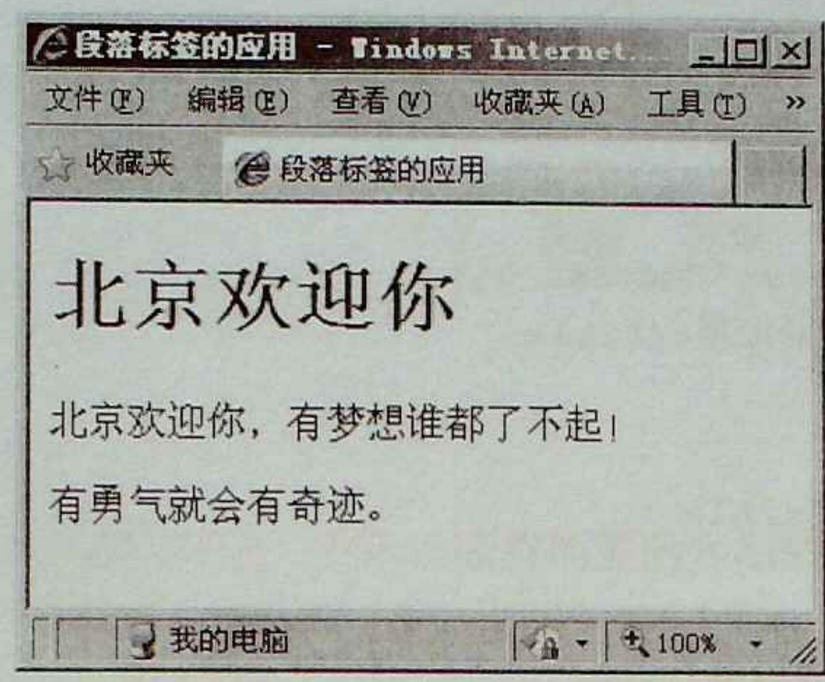


图 1.9　段落标签的应用

示例 2

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>段落标签的应用</title>
</head>
<body>
  <h1>北京欢迎你</h1>
  <p>北京欢迎你,有梦想谁都了不起!</p>
  <p>有勇气就会有奇迹。</p>
</body>
</html>
```

3. 水平线标签 <hr />

顾名思义，水平线标签表示一条水平线，注意该标签比较特殊，没有结束标签，直接使用“<hr/>”表示标签的开始和结束。例如，为了让版面更加清晰直观，可以在歌名和歌词间加一条水平分隔线（图 1.10），对应的 HTML 代码如示例 3 所示。

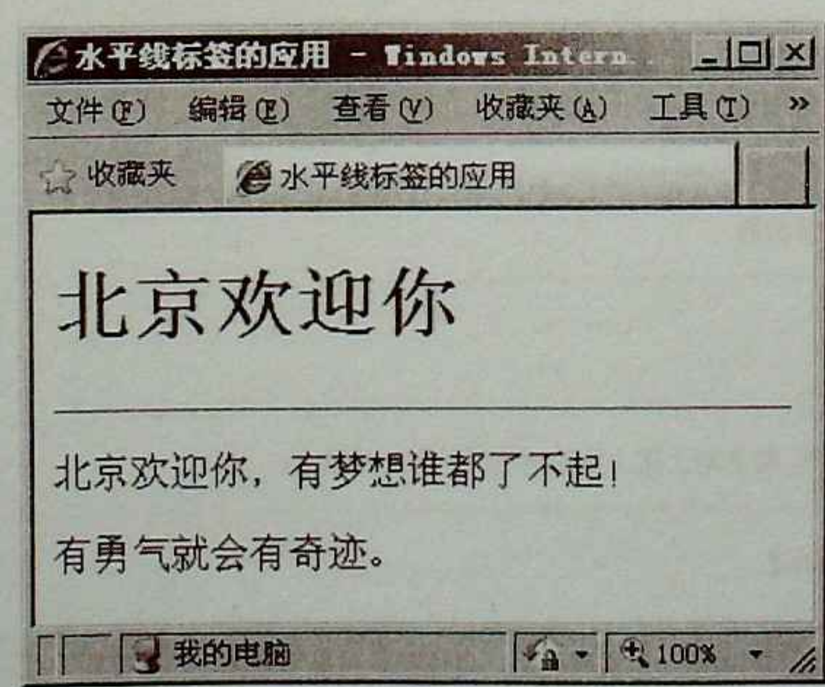


图 1.10　水平线标签的应用

示例 3

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>段落标签的应用</title>
</head>
<body>
  <h1>北京欢迎你</h1>
  <hr />
  <p>北京欢迎你,有梦想谁都了不起!</p>
  <p>有勇气就会有奇迹。</p>
</body>
</html>
```

1.2.2 上机练习

上机练习 1

练习——基本块级标签

需求说明

使用 UltraEdit 编辑工具，编写 HTML 代码，实现如图 1.11 所示的页面效果，注意内容间的层次结构。

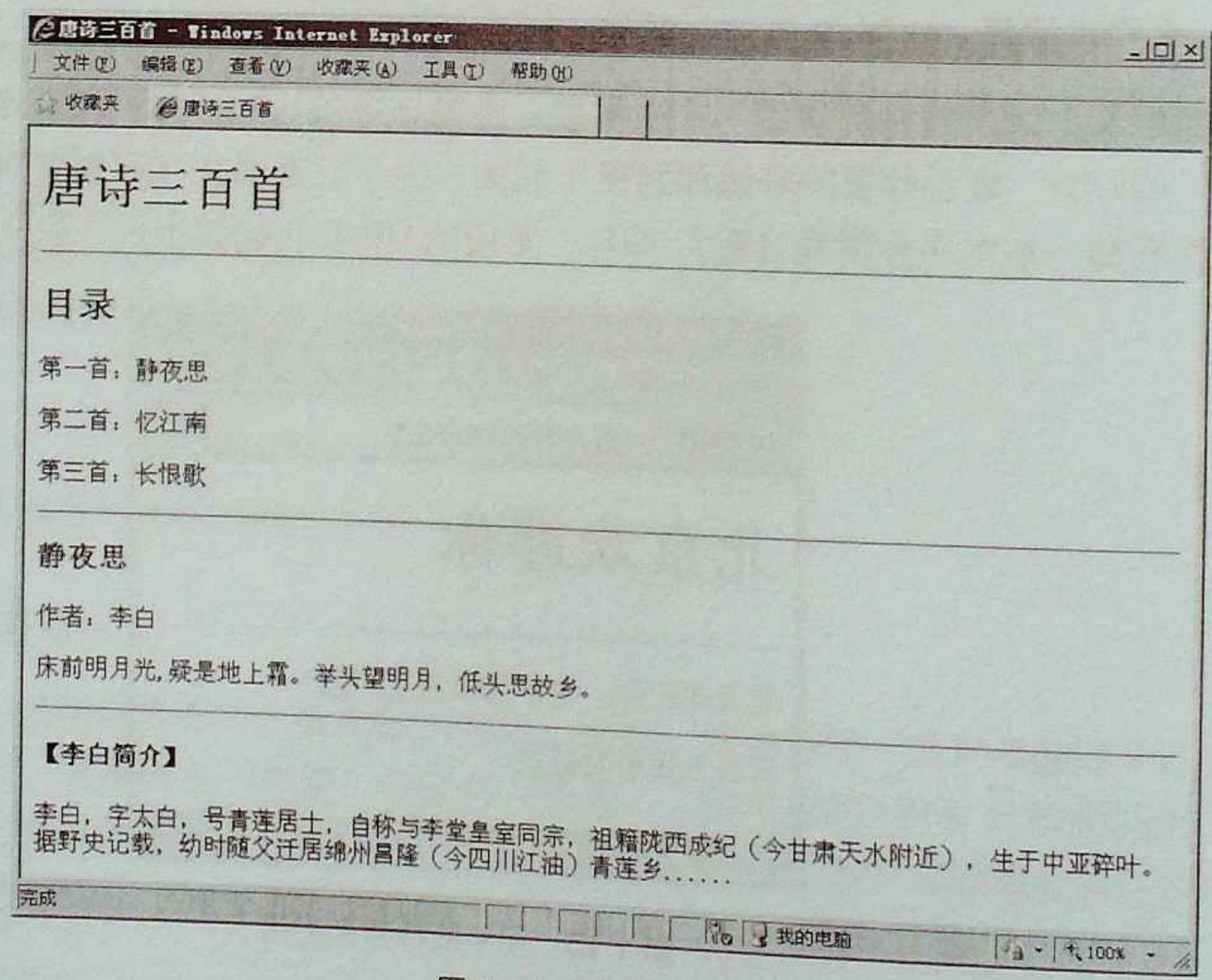


图 1.11 静夜思练习

1.2.3 常用于布局的块级标签

这类标签包括有序列表、无序列表、定义列表、表格、表单、分区标签（ ），它们常用于布局网页，组织 HTML 的内容结构。

1. 有序列表标签

有序列表标签表示多个并列的列表项，它们之间有明显的先后顺序，使用
、
表示有序列表，
、
表示各列表项。例如，描述某个网站的注册步骤（图 1.12），对应的 HTML 代码如示例 4 所示。

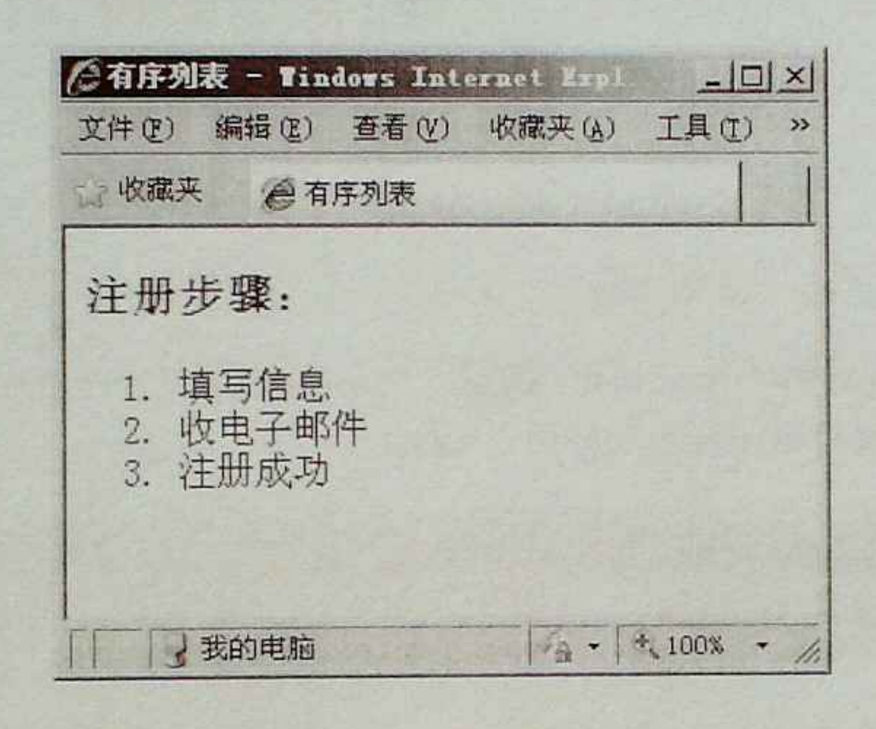


图 1.12 有序列表

示例 4

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>有序列表</title>
</head>
<body>
<h3>注册步骤:</h3>
<ol>
  <li>填写信息</li>
  <li>收电子邮件</li>
  <li>注册成功</li>
</ol>
</body>
</html>
```

2. 无序列表标签

无序列表和有序列表类似，但多个并列的列表项之间没有先后顺序，使用
、
表示有序列表，
、
表示各列表项。例如，描述某个网站的使用帮助（图 1.13），对应的 HTML 代码如示例 5 所示。

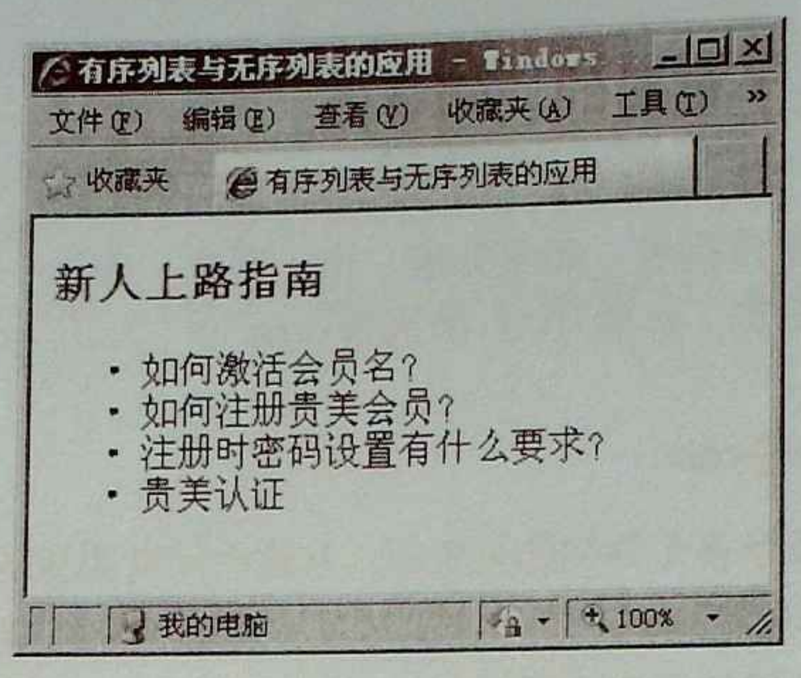


图 1.13　无序列表

示例 5

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>有序列表与无序列表的应用</title>
</head>
<body>
<h3>新人上路指南 </h3>
<ul>
  <li>如何激活会员名?</li>
  <li>如何注册贵美会员?</li>
  <li>注册时密码设置有什么要求?</li>
  <li>贵美认证</li>
</ul>
</body>
</html>
```

3. 定义列表标签<dl>

顾名思义，定义列表标签用于描述某个术语或产品的定义或解释，例如计算机、Java 语言、MP4 电子产品的定义等。它使用<dl>、</dl>表示定义列表，<dt>、</dt>表示术语，<dd>、</dd>表示术语的具体描述。例如，咖啡的定义（图 1.14）所对应的 HTML 代码如示例 6 所示。

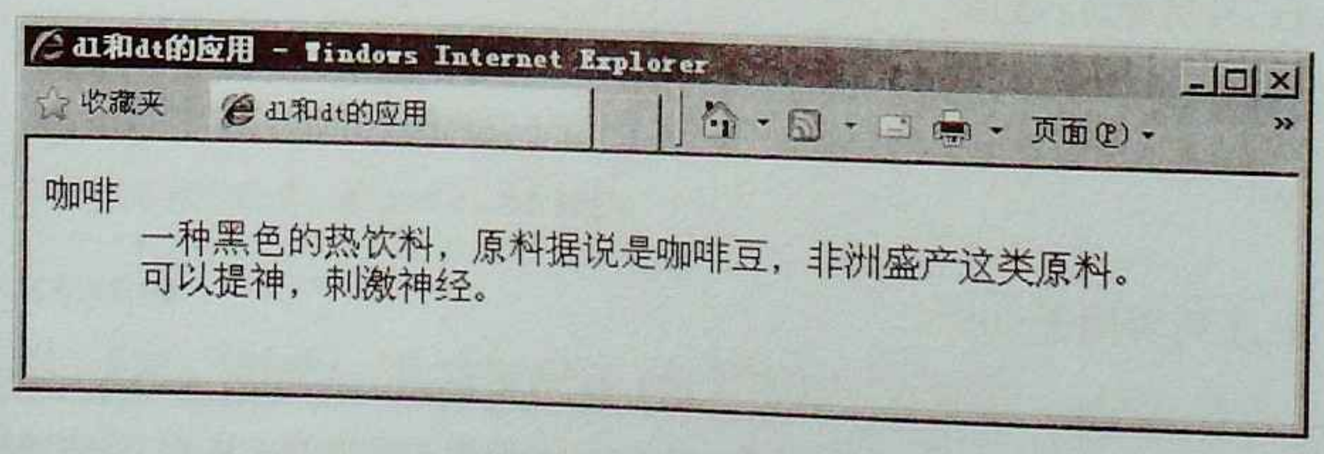


图 1.14　dl-dt-dd 的应用

示例 6

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>dl、dt 和 dd 的应用</title>
</head>
<body>
  <dl>
    <dt>咖啡</dt>
    <dd>一种黑色的热饮料,原料据说是咖啡豆,非洲盛产这类原料。</dd>
    <dd>可以提神,刺激神经。</dd>
  <dl>
</body>
</html>
```

在实际应用中，定义列表还被扩展应用到图文混编的场合，如图 1.15 所示。将产品图片作为术语标题<dt>，文字内容作为术语描述<dd>。这种局部布局结构将在后续章节进行详细讲解。

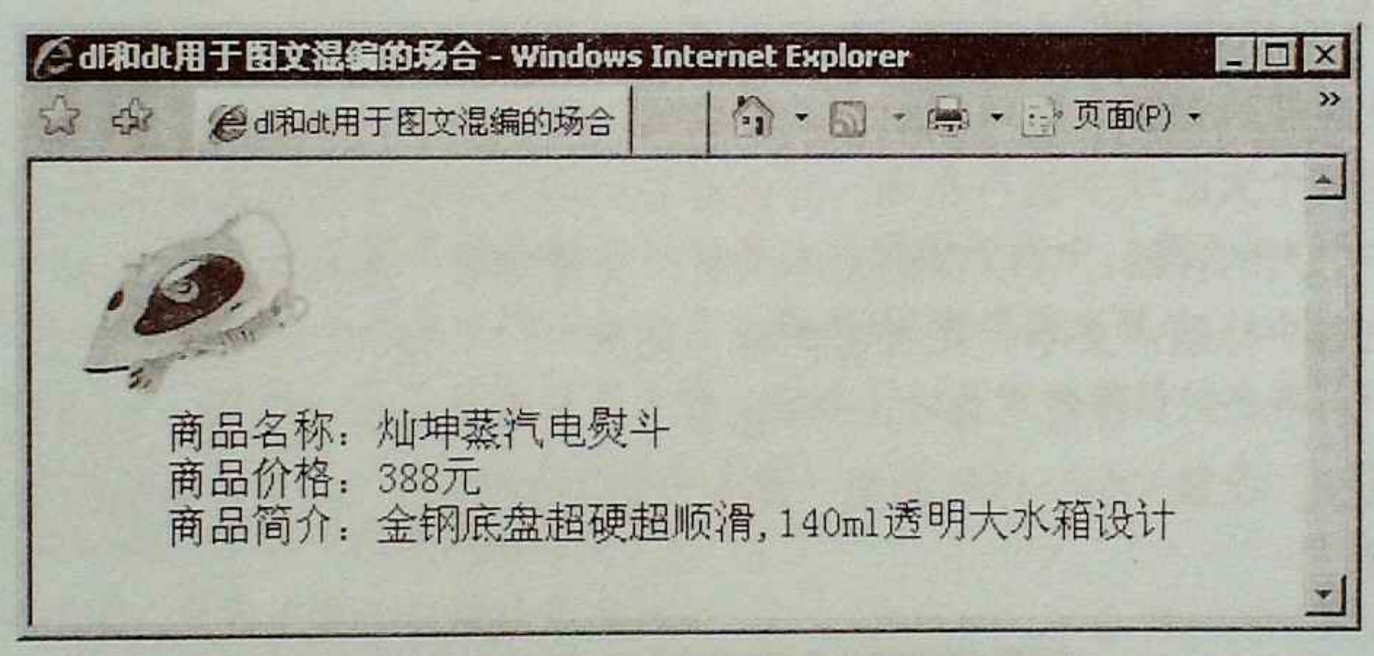


图 1.15　定义列表标签用于图文混编的场合

4．表格标签<table>

顾名思义，表格标签用于描述一个表格，它使用规整的数据显示，如图 1.16 所示。它使用<table>、</table>表示表格，<tr>、</tr>表示行，<td>、</td>表示列。表格的用法将在第 3 章进行详细介绍。

图 1.16　表格标签的应用

5. 表单标签<form>

表单标签用于描述需要用户输入的页面内容，例如注册页面（图 1.17）、登录页面等。它使用<form>、</form>表示表单，<input/>表示输入内容项。表单的具体用法将在第 2 章进行详细介绍。

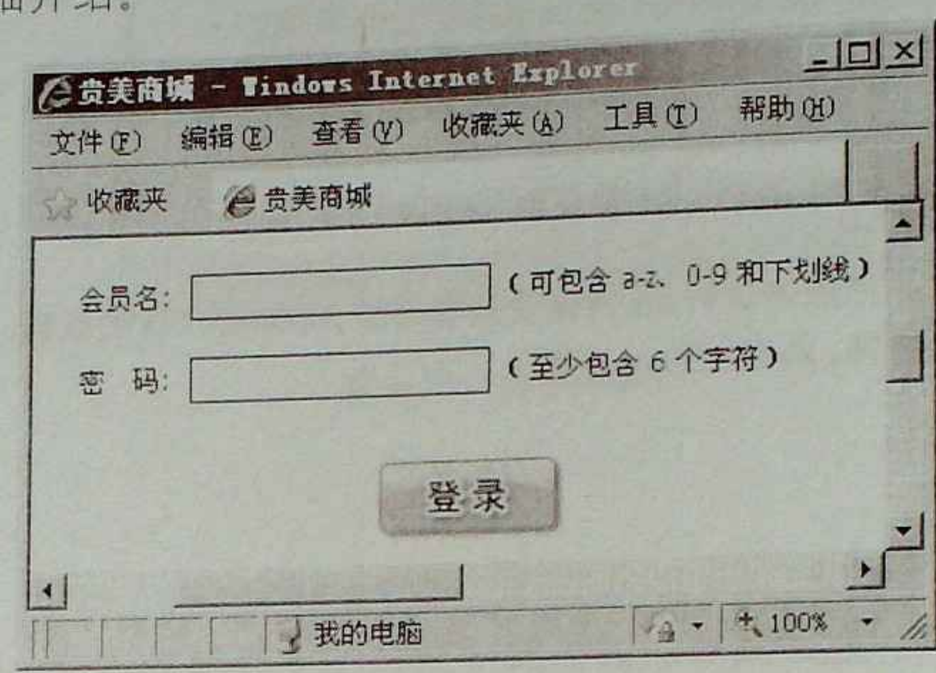


图 1.17 表单标签的应用

6. 分区标签

前几类块级标签一般用于组织小区块的内容。为了方便管理，数目众多的小区块还需要放到一个大区块中进行布局。分区标签 常用于页面布局时对区块的划分，它相当于一个大的容器，可以容纳无序列表、有序列表、表格等块级标签，同时也可容纳普通的段落、标题、文字、图片等内容，如图 1.18 所示，对应的 HTML 代码如示例 7 所示。由于 标签不像<h1>等标签，没有明显的外观效果，所以特意添加“style”样式属性，设置 标签的宽、高尺寸以及背景色。样式方面的用法将在后续章节详细介绍。

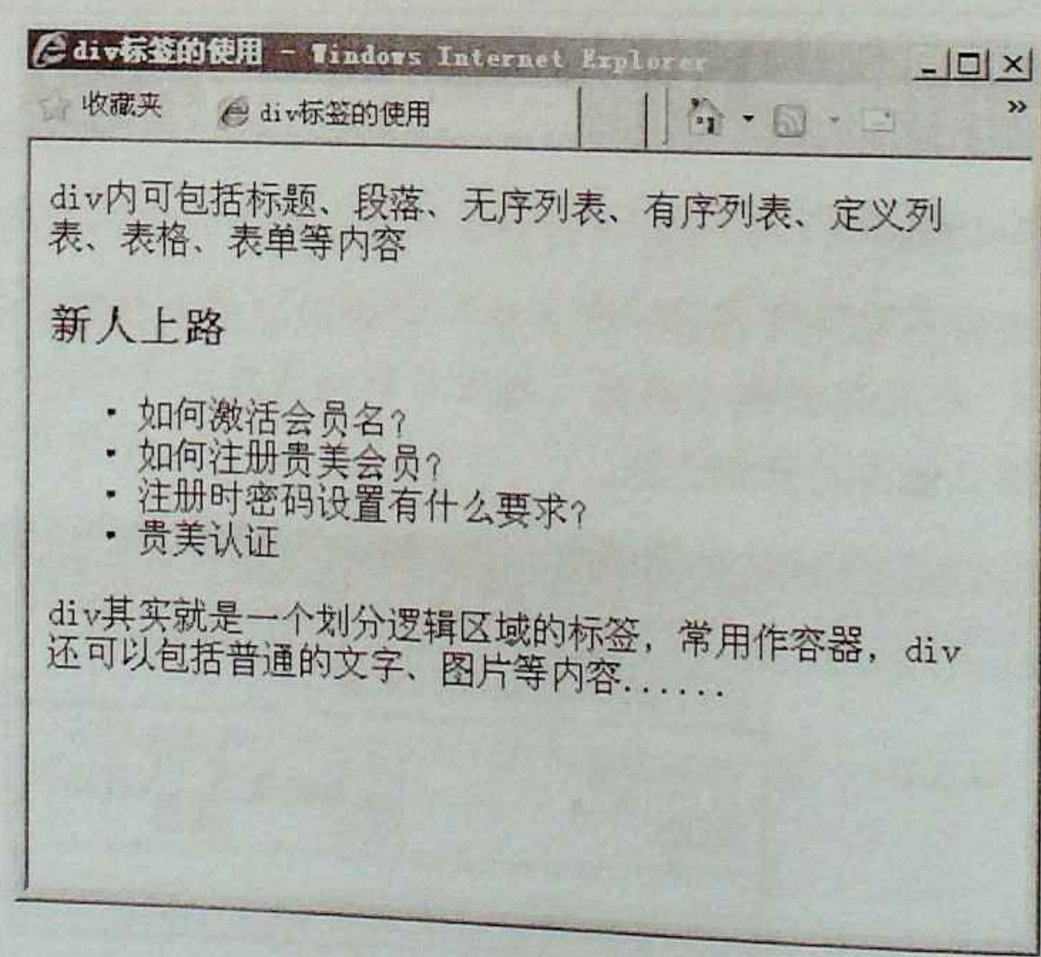


图 1.18 div 分区标签的应用

示例 7

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>dl、dt 和 dd 的应用</title>
</head>
<body>
  <div style="width:400px; height:300px; background:#9FF">
    <p>div 内可包括标题、段落、无序列表、有序列表、定义列表、表格、表单等内容</p>
    <h3>新人上路</h3>
    <ul>
      <li>如何激活会员名?</li>
      <li>如何注册贵美会员?</li>
      <li>注册时密码设置有什么要求?</li>
      <li>贵美认证</li>
    </ul>
    div 其实就是一个划分逻辑区域的标签,常用作容器,div 还可以包括普通的文字、图片等内容
  </div>
</body>
</html>
```

经验总结

到目前为止，我们学习了常用的各种块级标签。在实际开发中，常使用 进行分区，ul-li 或 ol-li 实现无序或有序的项目列表，dl-dt-dd 实现图文混编，table-tr-td 实现规整数据的显示。由此，在页面局部布局中，形成了如下四种常用的块状结构。

- div-ul(ol)-li：常用于分类导航或菜单等场合。
- div-dl-dt-dd：常用于图文混编场合。
- table-tr-td：常用于规整数据的显示场合。
- form-table-tr-td：常用于表单布局的场合。

这四种块状结构非常实用，它们的具体应用还将在后续章节进行深入讲解和训练。

1.2.4 上机练习

上机练习 2

练习——常用于布局的块级标签

需求说明

使用 UltraEdit 编辑工具，编写 HTML 代码，实现如图 1.19 所示的页面效果。

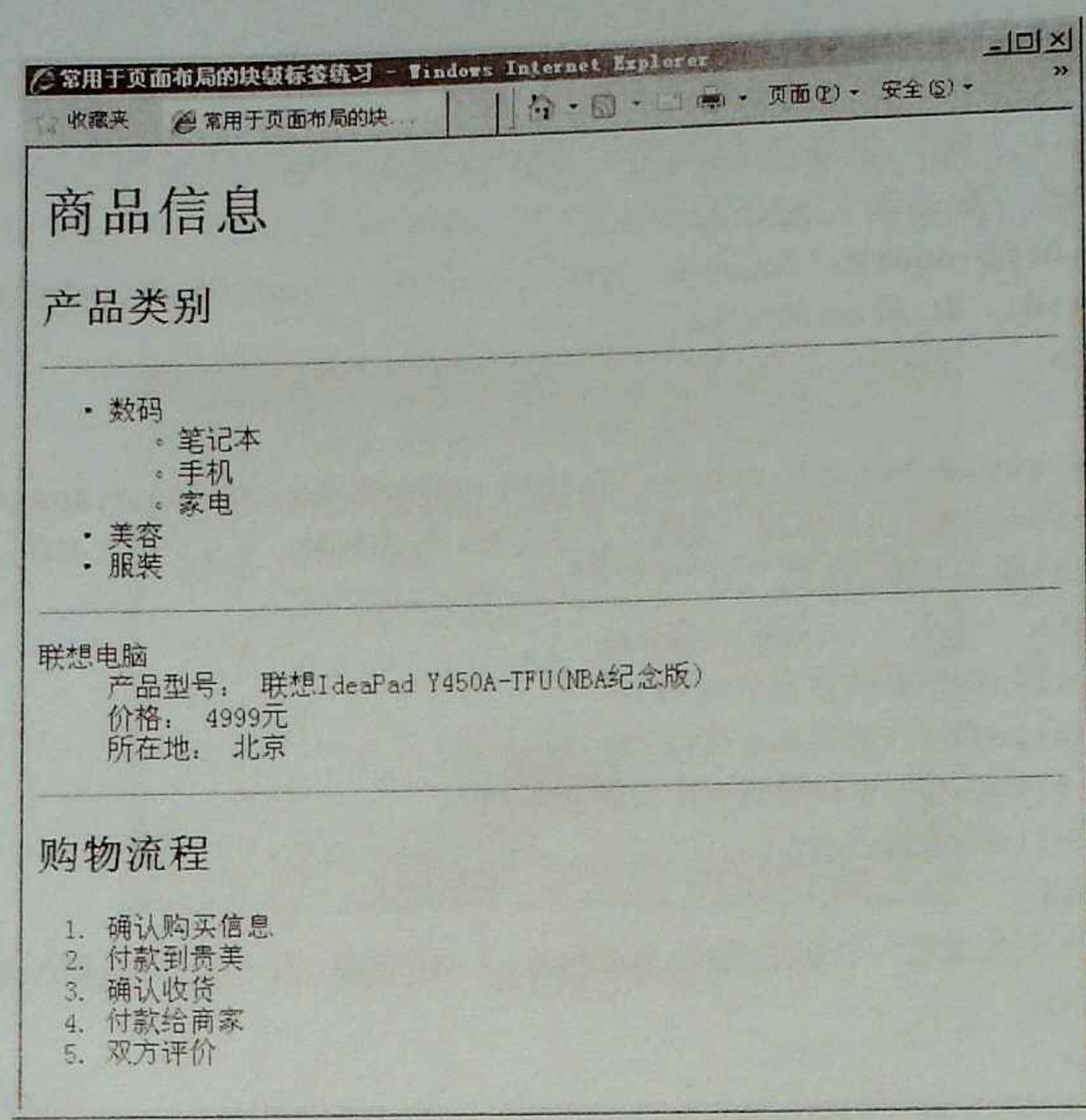


图 1.19　简单的商品购买页

提示

- 使用 分区标签作为整个页面内容的容器，然后放置标题、无序列表、有序列表等。
- 块级标签都支持嵌套。例如，无序列表可以再嵌套无序列表。

1.3　行级标签

行级标签也称内联标签、行内标签。使用块级标签为页面“搭建完成组织结构”后，往每个小区块添加内容时，就需要用到行级标签。

行级标签类似文本的显示，按“行”逐一显示。常用的行级标签包括图像标签<img />、超链接标签<a>、范围标签、换行标签
以及和表单相关的输入框标签<input />、文本域标签<textarea>等，表单涉及的行级标签将在第 2 章详细介绍。

1.3.1　图像标签<img/>

1. 常见的图片格式

在日常生活中，使用比较多的图像格式有四种，即 JPG、GIF、BMP、PNG。在网页中使

用比较多的是 JPG、GIF 和 PNG，大多数浏览器都可以显示这些图像。下面我们就来分别介绍这五种常用的图像。

1）JPG

JPG（JPEG）格式图像是在 Internet 上被广泛支持的图像格式，它是联合图像专家组文件格式（Joint Photographic Experts Group）的英文缩写。JPG 格式采用的是有损压缩，会造成图像画面的失真，不过压缩之后的体积很小，而且还比较清晰，所以比较适合在网页中应用。

通常可以通过压缩 JPG 文件在图像品质和文件大小之间达到良好的平衡。

制作 JPG 文件的软件很多，比较常见的有 Photoshop、HyperSnap 等。

2）GIF

GIF 图像是网页中使用最广泛、最普遍的一种图像格式，它是图像交换格式（Graphics Interchange Format）的英文缩写。GIF 文件的众多特点恰恰适应了 Internet 的需要，于是它成了 Internet 上最流行的图像格式，它的出现为 Internet 注入了一股新鲜的活力。GIF 格式图像采用的是无损压缩。由于只支持 256 色，所以 GIF 格式图像体积也很小，支持透明色，这使得 GIF 在网页的背景和一些多层特效的显示上用得非常多。它还支持帧动画，这是它最突出的一个特点。要设计 GIF 动画，首先要用图像处理软件做好 GIF 动画中的每一幅单帧画面，然后再用专门的 GIF 动画软件把这些静止的画面连在一起，再定好帧与帧之间的时间间隔，最后保存成 GIF 格式就可以了。

制作 GIF 文件的软件也很多，比较常见的有 Photoshop、GIF Construction Set 等。

3）BMP

BMP 图像在 Windows 操作系统中使用得比较多，它是“位图”的英文缩写。BMP 图像可以用任何颜色深度（从黑白到 24 位颜色）存储单个光栅图像。BMP 图像文件格式与其他 Microsoft Windows 程序兼容。它不支持文件压缩，也不适用于 Web 页。从总体上看，BMP 图像格式的缺点超过了它的优点。为了保证照片图像的质量，一般使用 JPG 文件。BMP 文件适用于 Windows 中的墙纸。BMP 的优点是支持 1 位到 24 位颜色深度，并且与现有 Windows 程序广泛兼容。BMP 的缺点是不支持压缩，这会使文件非常大。

制作 BMP 文件的软件也很多，比较常见的有 Photoshop、Windows 自带的“画图”软件等。

4）PNG

PNG 是 20 世纪 90 年代中期开始开发的图像文件存储格式，它兼有 GIF 和 JPG 的优势，同时具备 GIF 文件格式不具备的特性。流式网络图形格式（Portable Network Graphic Format，PNG）名称来源于非官方的“PNG's Not GIF”，读成“ping”。PNG 用来存储灰度图像时，灰度图像的深度多达 16 位；存储彩色图像时，彩色图像的深度多达 48 位。唯一遗憾的是，PNG 是一种新兴的 Web 图片格式，部分旧版本浏览器（如 IE 6.0）对 PNG 的支持不是很好。

制作图片时，使用 Photoshop 将图片另存为 PNG 格式即可。

2. 语法

```
<img src="图片地址" alt="图像的替代文字" title="鼠标悬停提示文字" />
```

其中，alt 属性指定的替代文本，表示图像无法显示时（例如，图片路径错误或网速太慢等），替代显示的文本。这样，即使图像无法显示，用户还是可以看到网页丢失的信息内容（图 1.20），所以在制作网页时一般推荐和“src”配合使用。其次，使用“title”属性，还可以提供额外的提示或帮助信息（图 1.21），方便用户使用。

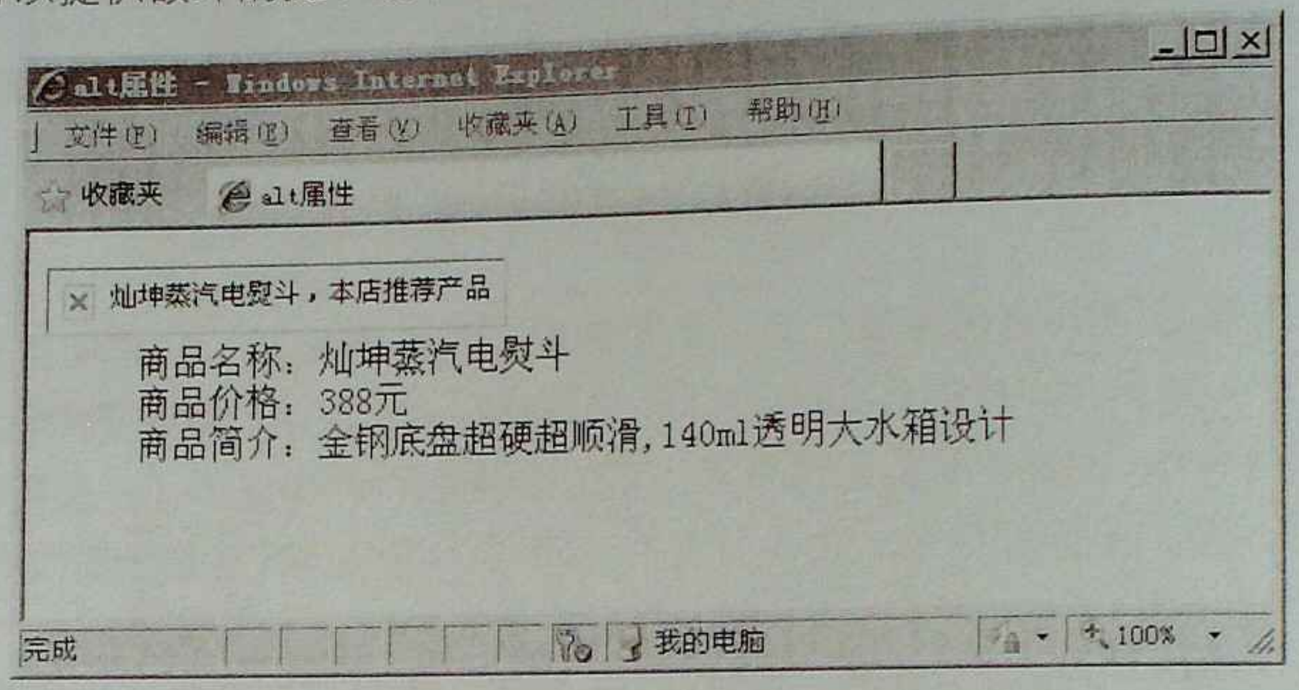


图 1.20　图片标签效果

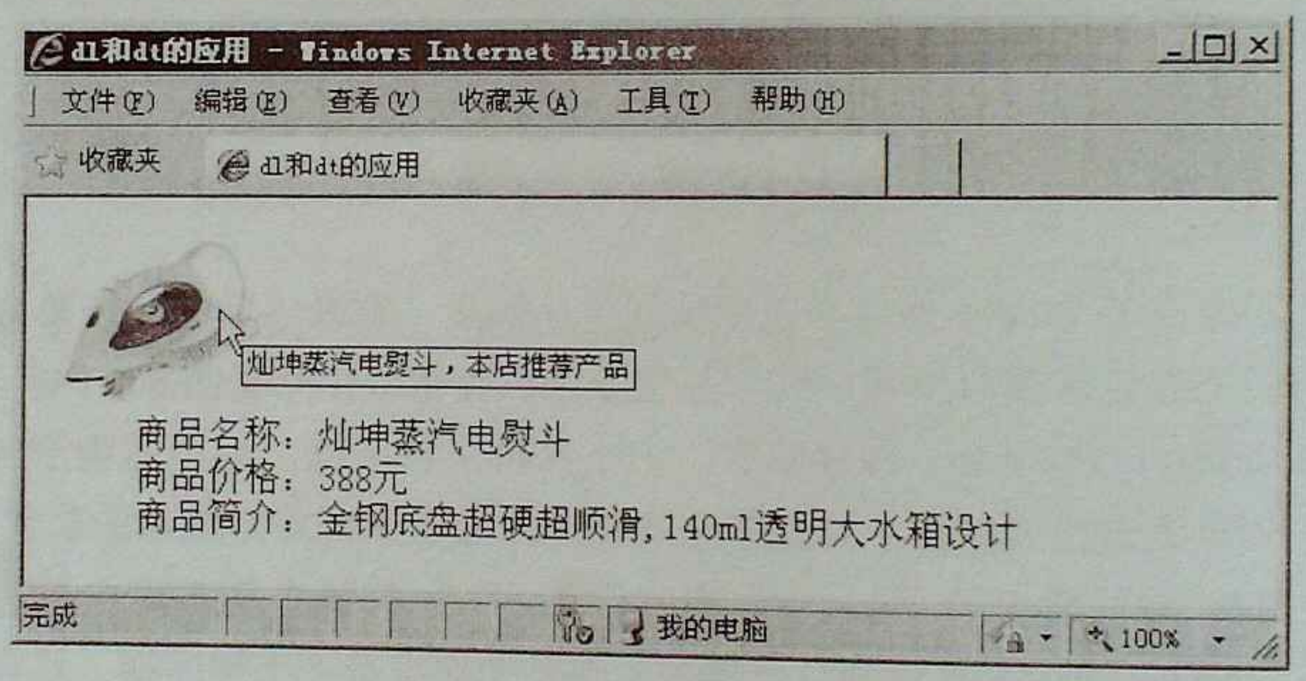


图 1.21　图片标签效果

图 1.21 对应的 HTML 代码如示例 8 所示，此处的图片作为定义列表的术语标题，实现了图文混编的效果。

示例 8

```
<html>
   <head>
     <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
     <title>dl、dt 和 dd 的应用</title>
   </head>
   <body>
```

```
    <dl>
      <dt><img src="images/iron.jpg" alt="灿坤蒸汽电熨斗,本店推荐商品" title="
     灿坤蒸气电熨斗,本店推荐商品"  /></dt>
      <dd>商品名称:灿坤蒸汽电熨斗</dd>
      <dd>商品价格:388 元</dd>
      <dd>商品简介:金钢底盘超硬超顺滑,140ml 透明大水箱设计</dd>
    </dl>
  </body>
</html>
```

经验总结

设计和制作网页时，需从方便用户使用的角度考虑问题。用户体验已越来越成为 Web 设计和开发需要考虑的重要因素，用户体验的原则之一就是以用户为中心，并体现在细微之处。例如，使用<img>标签时，强烈推荐同时使用“alt”和“title”属性，避免因网速太慢或路径错误带来的“一片空白”或“错误”提示；同时，增加的鼠标提示信息也方便用户的使用。

1.3.2 范围标签

范围标签用于标识行内的某个范围。例如，实现行内某个部分的特殊设置以区分其他内容；又如为了用低价吸引用户购买商品，特意将价格一行中的“8”设置为红色、大字体显示，如图 1.22 所示。

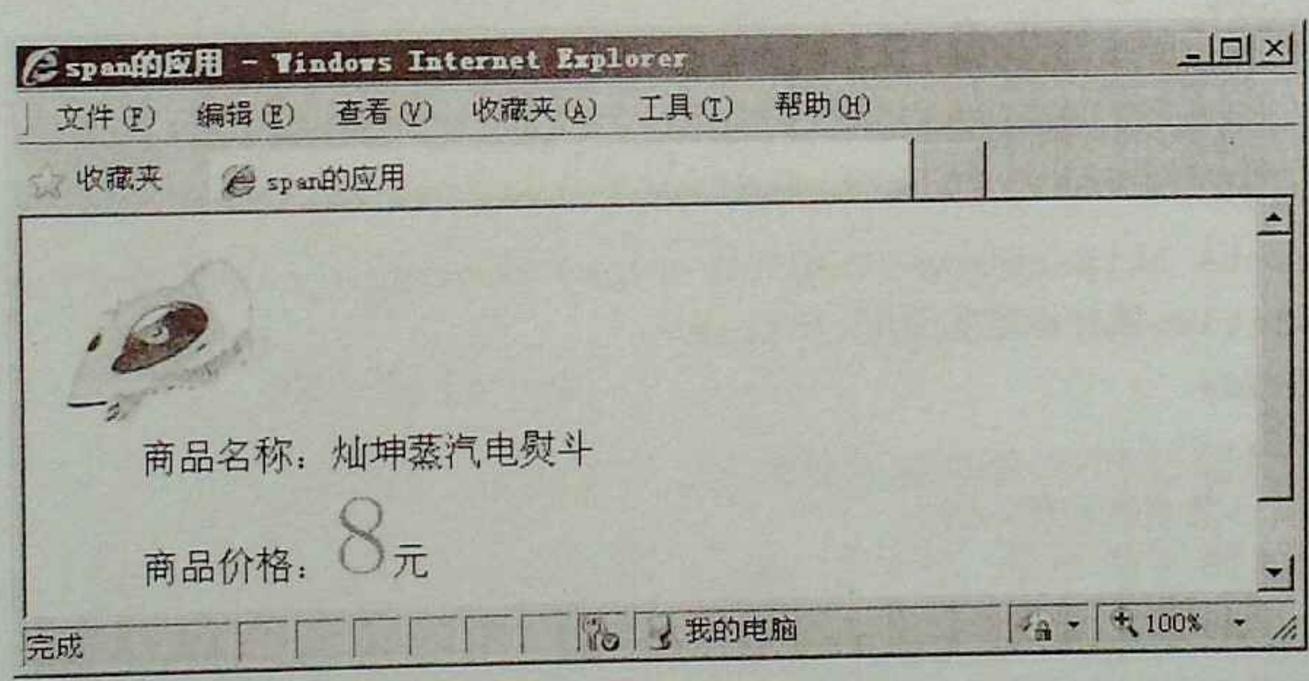


图 1.22 span 标签的应用

只需将示例 8 的相关代码进行如下修改。

```
<dd>商品价格:<span style="color:red;font-size:50px;">8</span>元</dd>
```

其中、标签限定某个范围，style 属性添加突出显示的样式（红色、字体大小为 50 像素）。

1.3.3 换行标签

换行标签
表示强制换行显示，该标签和<hr/>标签一样，没有结束标签。例如，希望“北京欢迎你”的歌词紧凑显示，每句间要求换行，如图 1.23 所示，其对应的 HTML 代码如示例 9 所示。

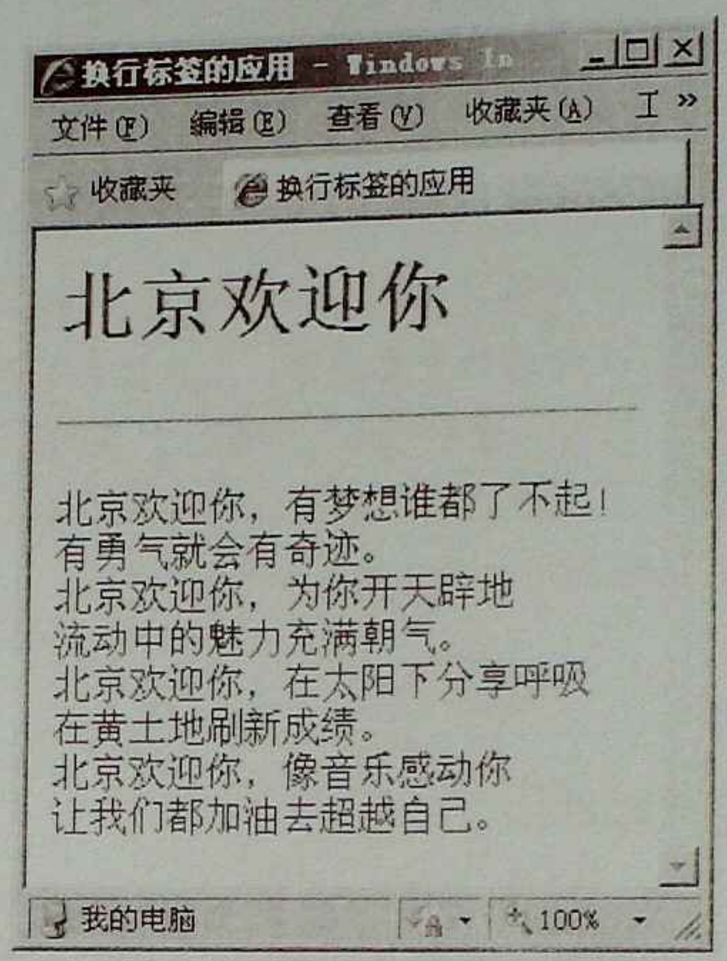


图 1.23
标签的应用

示例 9

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>换行标签的应用</title>
  </head>
  <body>
    <h1>北京欢迎你</h1>
    <p>
    北京欢迎你,有梦想谁都了不起!<br />
    有勇气就会有奇迹。<br />
    北京欢迎你,为你开天辟地<br />
    流动中的魅力充满朝气。<br />
    北京欢迎你,在太阳下分享呼吸<br />
    在黄土地刷新成绩。<br />
    北京欢迎你,像音乐感动你<br />
    让我们都加油去超越自己。<br />
    </p>
  </body>
</html>
```

经验总结

- W3C 标准规定单标签，必须有结尾，正确写法为：
。
-
和使用块级元素产生的分行存在语义区别。

1.4 W3C 标准

正如前面所述，发明 HTML 的初衷是为了信息资料的网络传播和共享，希望 HTML 文档具有平台无关性，即同一 HTML 文档在不同平台上（包括使用不同的浏览器）将看到同样的页面内容和效果。但遗憾的是，随着浏览器市场的激烈竞争，各大浏览器厂商，例如早期的网景公司（Netscape 浏览器）以及后来居上的微软（IE 浏览器）等，为了吸引用户，都在早期 HTML 版本的基础上进行各类标签的扩展。但各浏览器之间互不兼容，导致 HTML 编码规则混乱，违背了 HTML 发明的初衷，因此需要一个组织来制定和维护统一的 Web 开发标准。W3C 正是这样的一个组织，下面介绍为什么是 W3C 标准以及 Web 开发方面的基本规范。

1.4.1 什么是 W3C 标准

W3C 是指万维网联盟（World Wide Web Consortium），创建于 1994 年 10 月，它是一个会员组织，主要职责是负责 Web 标准的制定和维护，Web 开发方面常涉及的 W3C 标准如下。

- HTML 内容方面——XHTML
- 样式美化方面——CSS
- 结构文档访问方面——DOM
- 页面交互方面——ECMAScript

其中 DOM 和 ECMAScript 将在第二学期学习，本课程主要涉及 XHTML 和 CSS 两类标准。

HTML 方面目前比较常用的版本是 XHTML 1.0，XHTML 表示可扩展超文本标签语言（eXtensible HyperText Language），它是更严格更纯净的 HTML 版本，且是一个 W3C 标准，它规定了 HTML 编写的具体规范，所有主流浏览器都支持。

1.4.2 W3C 提倡的 Web 页结构

Web 页的开发为什么需要 XHTML、CSS 两套标准？W3C 提倡应如何编写 Web 页？不妨先看一个糟糕的 HTML 文档示例，代码如示例 10 所示。

示例 10

```
<html>
```

```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <title>不规范的示例</title>
</head>
<body>
  <font size="7">一级主题</Font><br/>
  一级主题阐述文字 <br /><Br />
  <font size="5">二级主题</font><br />
  二级主题相关文字
  <P>项目列表 1
  <p>项目列表 2
  <p>项目列表 3
  </body>
</html>
```

这里使用了 HTML 早期标签<font>表示字体大小，标签大小写不统一，段落<p>标签没有配对，但在浏览器中还能正常显示（这是因为浏览器为了“讨好”而“纵容”用户的原因），如图 1.24 所示。

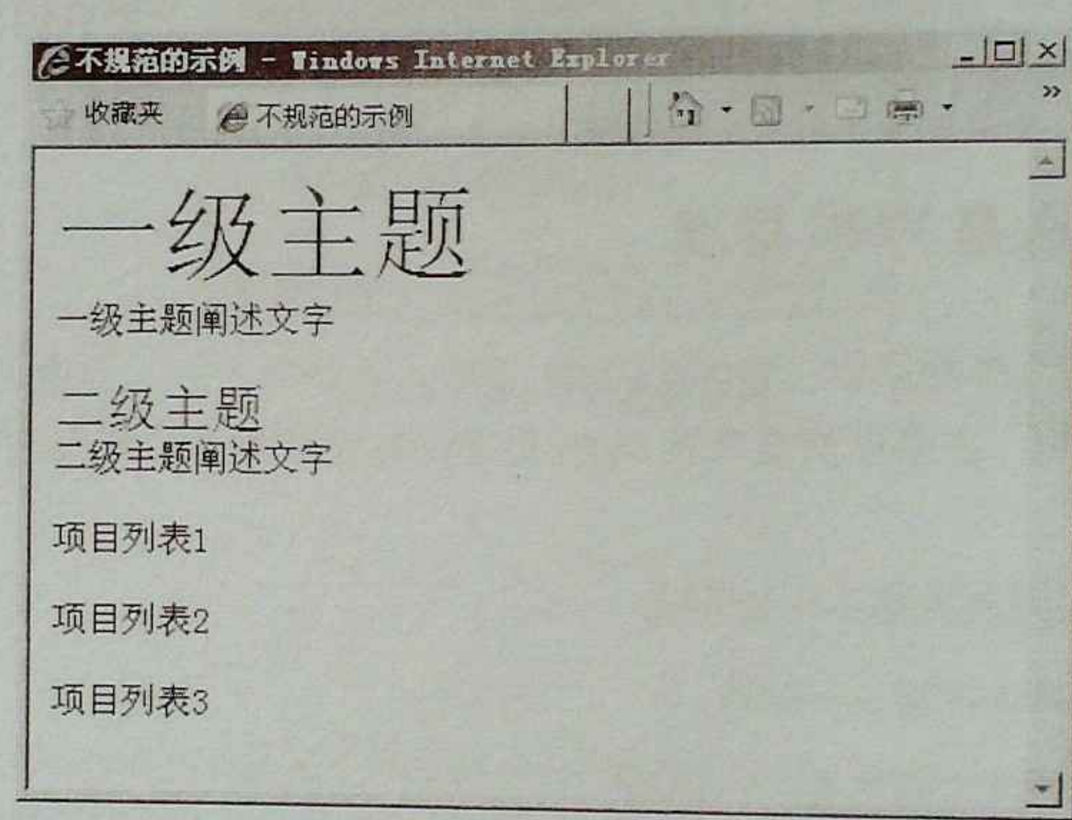


图 1.24　不规范的 HTML 文档示例

这样编写有什么问题？仔细查看 HTML 结构和表述的页面内容，不难发现存在如下弊端。

- 内容和表现没分离，后期很难维护和修改。编写的 HTML 代码既表示字体大小等样式，又包含内容，如网站升级改版时需要改变字体大小等样式，则需要逐行修改 HTML 代码，非常繁琐。
- HTML 代码不能表示页面的内容语义，不利于搜索引擎搜索。即从 HTML 代码不能看出页面内容的关系，很难判断哪些内容是主题，哪些内容是相关的阐述文字，很难看出各列表项的内容之间的关系。而搜索引擎的爬虫在搜索页面时只识别含有语义化的标签（例如<h1>标题，<p>段落等），而不识别表示样式的标签（例如<font>字体、<b>加粗等）。

因此，W3C 提倡的 Web 结构如下。

1. 内容（结构）和表现（样式）分离

即 XHTML 只负责网页的内容结构，CSS（Cascading Style Sheets，层叠样式表）负责表现样式（例如字体大小、颜色、背景图、显示位置等），方便网站的修改和维护。打个比方，如把页面看作一幢房子，XHTML 就好比是房子骨架（即钢架混泥土部分）；而 CSS 就好比是房子的装饰材料（油漆、木地板、大理石等）。如需要给房子重新装饰，则只需要更换装饰材料即可。

2. HTML 内容结构要求语义化

即要求能根据 HTML 代码能看出这部分内容是什么，代表什么含义。例如，是标题、段落还是项目列表等。这样做的好处：一是前面提及的方便搜索引擎搜索；二是方便在各种平台传递，除普通的计算机外，还包括手机、PDA、MP4 等。因为手机、PDA 等轻量级显示终端可能不具备普通计算机上浏览器的渲染能力，它将按照 HTML 结构的语义，使用自身设备的渲染能力显示页面内容。因此，HTML 结构语义化越来越成为一种主流趋势。

具体如何编写具有语义化的 HTML 结构，要点就是根据页面文档表达的含义，选用对应含义的 HTML 标签。例如，修改上述不规范的示例，具有语义化的 HTML 结构如示例 11 所示，页面效果如图 1.25 所示。

示例 11

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>规范的示例</title>
  </head>
  <body>
    <h1>一级主题 1</h1>
    <p>一级主题阐述文字</p>
    <h2>二级主题</h2>
    <p>二级主题阐述文字</p>
    <ul>
      <li>项目列表 1</li>
      <li>项目列表 2</li>
      <li>项目列表 3</li>
    </ul>
  </body>
</html>
```

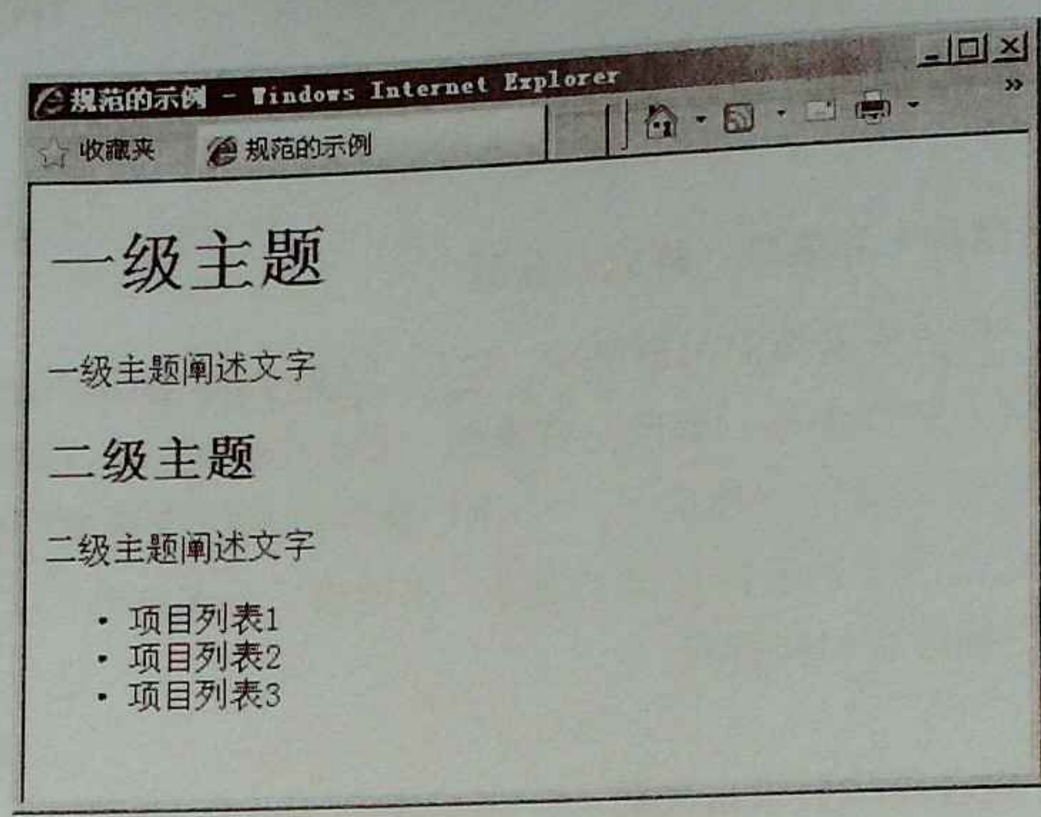


图 1.25　规范的 HTML 文档示例

1.4.3　XHTML 1.0 的基本规范

了解了 W3C 提倡的 Web 结构后，下面介绍 XHTML 的基本规范。

1. 标签名和属性名称必须小写

例如，规范的写法为<html>，不规范的写法为<HTML>、<hTML>等。

2. HTML 标签必须关闭

即 HTML 标签要求配对使用。例如，规范的写法为 …… ，不规范的写法为 ……。

其中换行和水平线比较特殊，使用
和<hr/>表示关闭。

3. 属性值必须用引号括起来

即必须使用单引号或双引号将属性值括起来，示例如下。

规范的写法：

```
<meta name="keywords" content="北大青鸟,IT 培训" />
```

不规范的写法：

```
<meta name=keywords content=北大青鸟,IT 培训/>
```

4. 标签必须正确嵌套

示例如下。

正确的写法：

```
<head>
   <title>……</title>
```

```
</head>
```

不正确的写法：

```
<head>
   <title>……</head>
</title>
```

5. 必须添加文档类型声明

即必须使用<!DOCTYPE>标签添加文档类型声明，声明 HTML 文档遵守 W3C XHTML 哪个级别的规范。

需要注意以下两点。

- 该声明必须位于 HTML 文档的第一行。
- XHTML 1.0 规定了三种级别的声明。
  - Strict（严格类型）：这种声明完全符合 W3C 的标准，但要求比较严格。对应的声明如下。

    ```
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
    www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
    ```

  - Transitional（过度类型）：也称松散(Loose)声明。相比 Strict 而言，要求相对宽松些。对应的声明如下。

    ```
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
    ```

  - Frameset（框架类型）：Strict 严格标准中不允许使用框架，当页面中需要使用框架时，则使用该声明。框架页将在后续章节讲解，对应的声明如下。

    ```
    <!DOCTYPE html PUBLIC"-//W3C//DTD XHTML 1.0 Frameset/EN" "http://
    www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
    ```

一个符合 XHTML 基本规范的 HTML 文档如示例 12 所示，此网页的显示效果如图 1.5 所示。

示例 12

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/
TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>我的第一个网页</title>
  </head>
  <body>
  HTML 网页
```

```
  </body>
</html>
```

1.4.4 上机练习

上机练习 3

练习——行级元素

需求说明

使用 UltraEdit 编辑工具，编写 HTML 代码，实现如图 1.26 所示。要求遵守 XHTML 1.0 Strict 标准，HTML 结构尽量语义化。

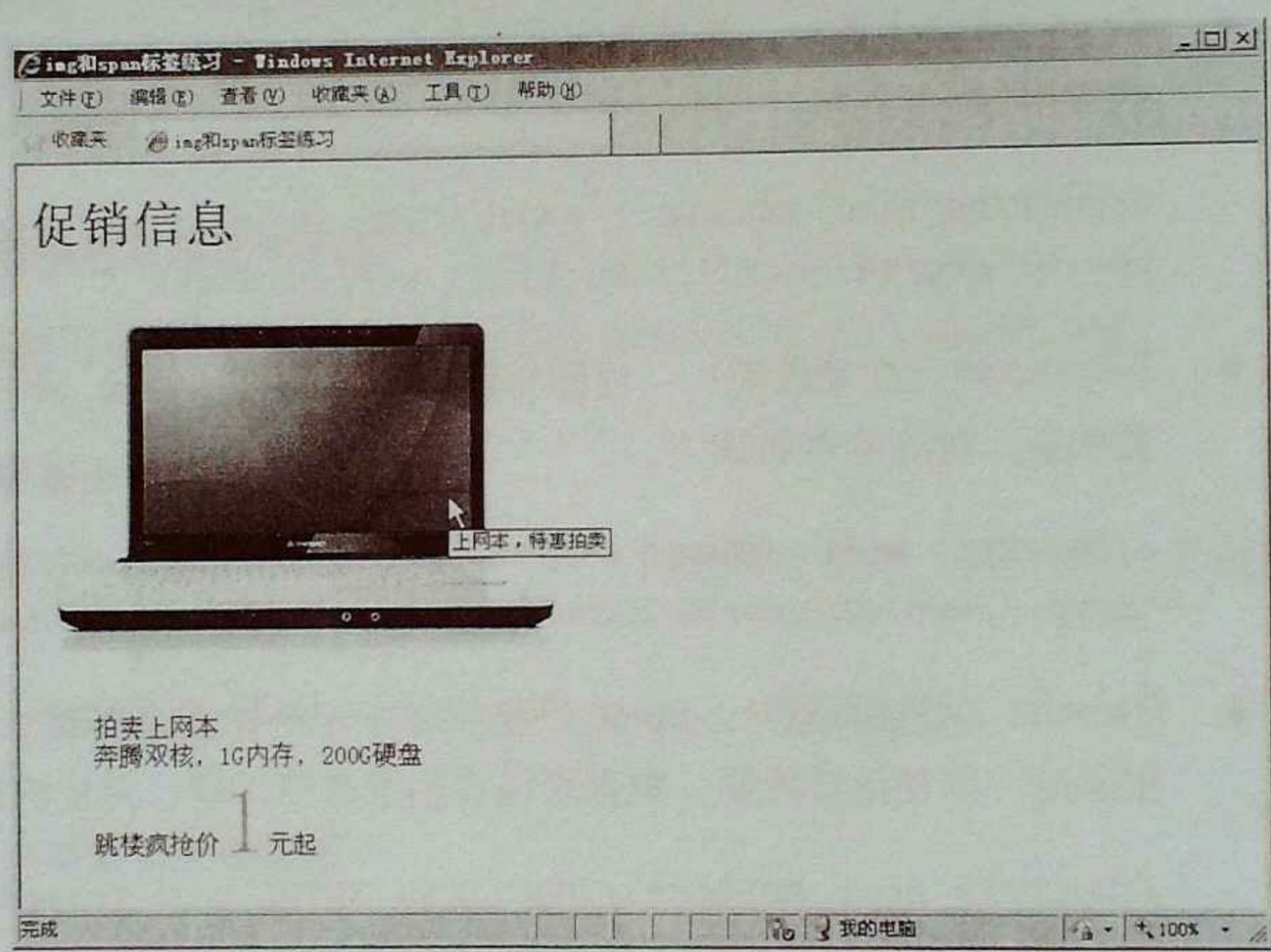


图 1.26 行级标签

本章总结

- HTML 标签分为块级和行级标签。块级标签按“块”显示，行级标签按“行”逐一显示。
- 基本的块级标签包括段落标签<p>、标题标签<h1>~<h6>、水平线标签<hr/>等。
- 常用于布局的块级标签包括无序列表标签<ul>、有序列表标签<ol>、定义列表标签<dl>、分区标签 等。
- 实际应用中，常使用如下四种块状结构。
  - div-ul(ol)-li：常用于分类导航或菜单等场合。
  - div-dl-dt-dd：常用于图文混编场合。
  - table-tr-td：常用于规整数据的显示。
  - form-table-tr-td：常用于表单布局的场合。
- 行级标签包括图片标签<img />、范围标签、换行标签
等。插入图片时，要求“src”和“alt”属性必选，“title”和“alt”属性推荐同时使用。
- 编写 HTML 文档注意遵守 W3C 相关标准，W3C 提倡内容和结构分离，HTML 结构具有语义化。

本章作业

一、选择题

1. HTML 的基本结构是（　　）。

A. <html><body></body><head></head></html>

B. <html><head></head><body></body></html>

C. <html><head></head><foot></foot></html>

D. <html><head><tittle></title></head></html>

2. 在 HTML 中，图片显示与其悬停提示文字显示分别用（　　）。

A. img 标签和 alt 属性

B. img 标签和 title 属性

C. img 属性和 alt 标签

D. img 属性和 title 标签

3. 下面说法正确的是（　　）。

A. HTML 页面元素的组织顺序是先从上到下，而后从左往右

B. 所有常用的图片格式都可以是透明背景

C. HTML 中的元素有两类：块级元素和行级元素

D. <h1>标签属于块级元素，但并不会自动换行

4. 以下（　　）是 HTML 常用的块状结构。

A. div-dl-dt-dd

B. div-ul-li

C. div-ol-li

D. table-tr-td

5. 下面关于多目标块状元素，正确的是（　　）。

A. 既可以终端形式出现，又可以结构化形式出现

B. 因为比较灵活，可以嵌套，所以很适合在块结构最内层或最外层

C. 规范用法中，多目标块状元素可以任意组合、放置在任何地方

D. 、<li>、<dd>、<dl>都是多目标块状元素

二、简答题

1. HTML 内容的组织顺序是什么？请举出一个你熟悉的网站，简要描述它的 HTML 页面内容的组织顺序。

2. 请写出 HTML 常用的四种块结构。

3. 用 div-dl-dt-dd 块状结构实现图文混编效果，如图 1.27 所示。

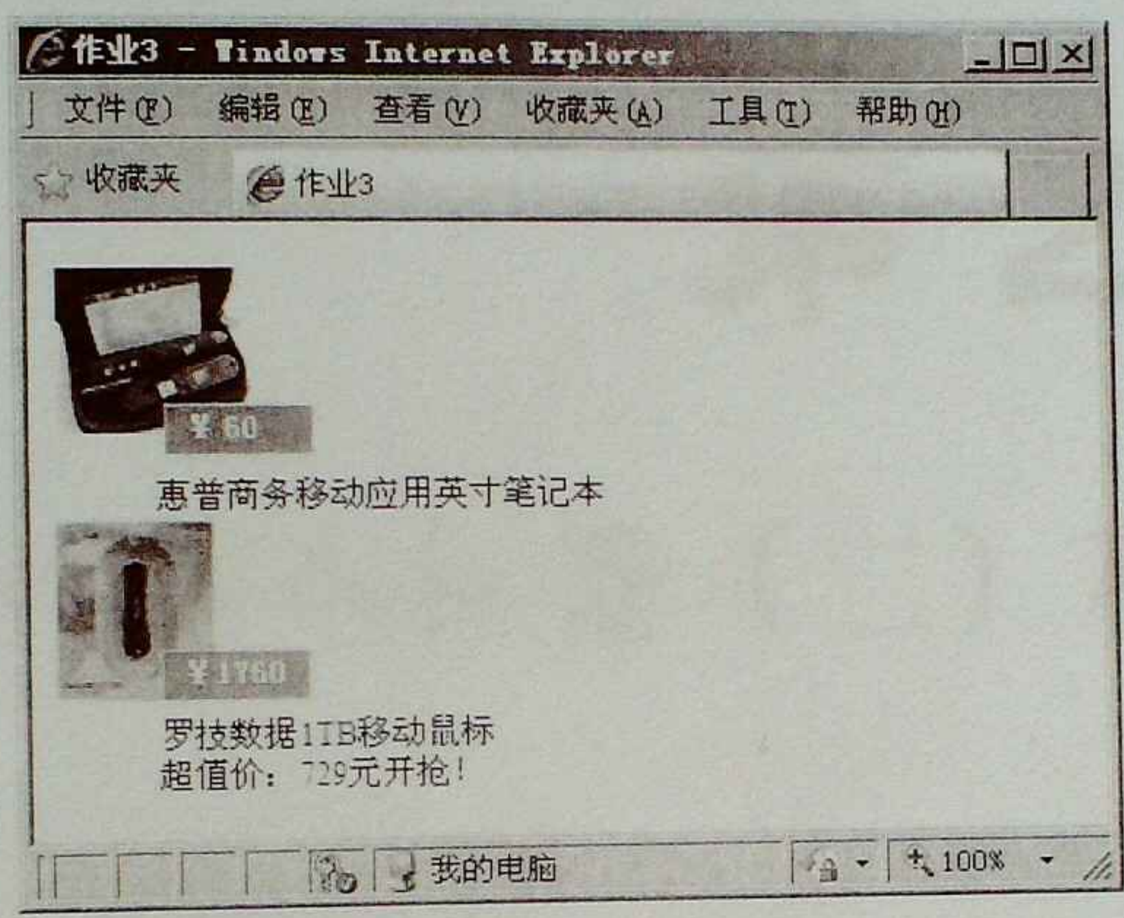


图 1.27　图文混编效果

4. 用 div-ul-li 结构实现图 1.27 的效果。（行前出现黑圆点，不用处理）

5. 不用块状结构，而使用<img />和
标签实现如图 1.27 所示的效果，并说明多种实现方法之间的语义区别。

第2章

HTML 基本标签（二）和表单

◈本章工作任务

- 制作“注册”表单页面

◈本章技能目标

- 使用超链接实现页面间导航
- 使用表单实现注册页面

本章单词

请在预习时学会下列单词的含义和发音，并填写在横线处。

1. form：_____
2. action：_____
3. input：_____
4. textarea：_____
5. method：_____
6. target：_____
7. checkbox：_____
8. radio：_____
9. hidden：_____

本章简介

上一章已介绍过 HTML 的基本结构、块级标签以及部分行级标签。本章将继续介绍另一个应用广泛的行级标签：超链接<a>标签，同时还将介绍特殊符号和表单<form>。其中超链接和表单是本章的重点。

2.1 超链接<a>标签

超链接<a>标签极为常用，常用来设置到其他页面的导航链接。超链接包含如下两部分内容。

- 链接地址：即链接的目标，可以是某个网址或文件的路径，对应为<a>标签的 href 属性。
- 链接文本或图像：单击该文本或图像，将跳转到 href 属性指定的链接地址，对应为<a>标签中的文字或图片。

2.1.1 超链接的基本用法

1. 基本语法

超链接的基本语法如下。

```
<a href="链接地址" target="目标窗口位置">链接文本或图像</a>
```

属性说明如下。

- href：表示链接地址的路径。
- target：指定链接在哪个窗口打开，常用的取值有_self（自身窗口）、_blank（新建窗口）。

例如，在新窗口中打开链接的程序如下。

```
<a  href="span.html"  target="_blank">灿坤蒸汽电熨斗</a>
```

效果如图 2.1 所示。

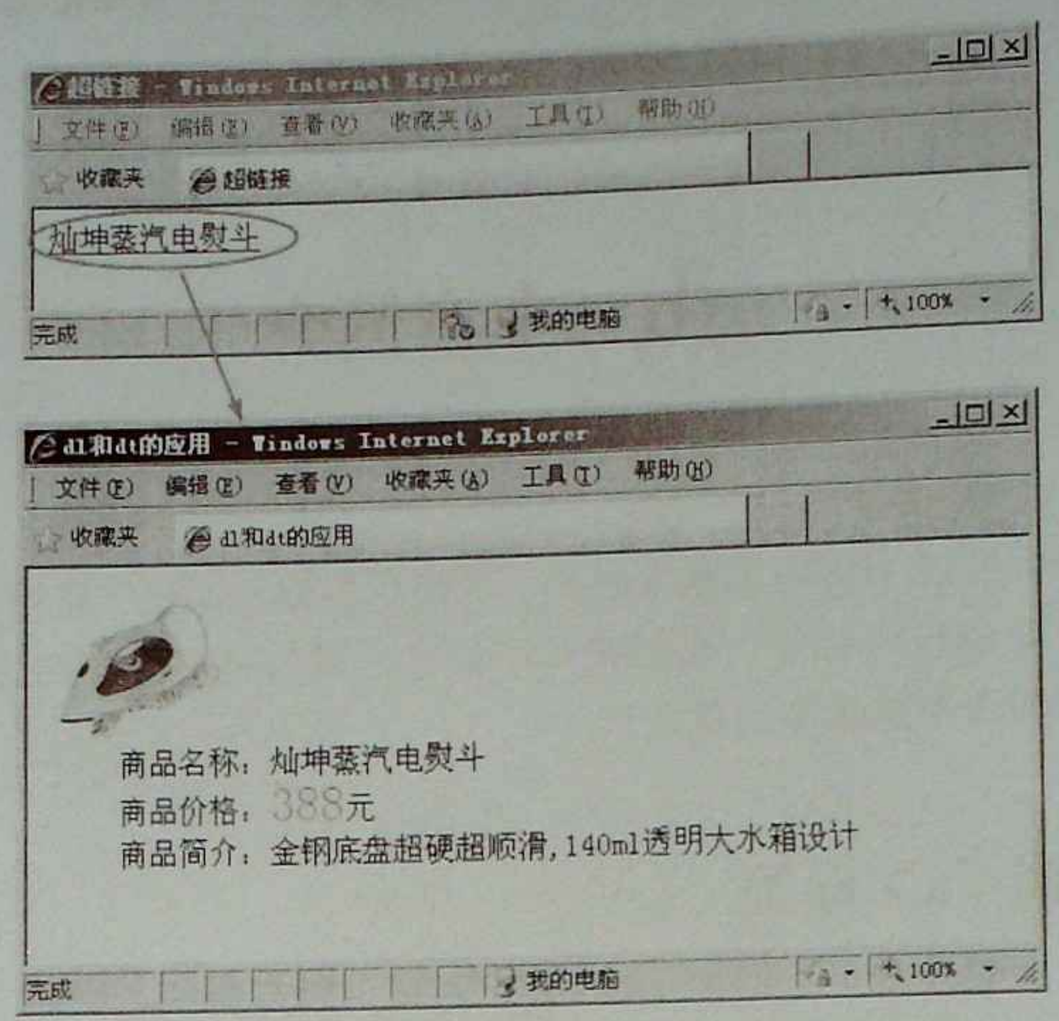


图 2.1　超链接语法示意图

2. 链接路径

当单击某个链接时，将指向万维网上的文档，万维网使用称为 URL（Uniform ResourceLocation，统一资源定位器）的方式来定义一个链接地址。例如，一个完整的链接地址的常见形式为 http://www.sohu.com:80/reg/register.html，URL 地址的统一格式为 Scheme://host.domain:port//path/filename。

对上面的格式解释如下。

- Scheme：表示各类通讯协议，例如，常用的是 http（超文本传输协议）、ftp（文件传输协议）。
- domain：定义因特网域名，以方便访问，例如 sohu.com 等。
- host：定义域中的主机名，如果被省略，http 协议默认的主机是 www。
- post：定义主机的端口，一般默认为 80 端口，可以省略。
- path：定义服务器上的路径，例如 reg 目录。
- filename：定义文档的名称，例如 register.html。

根据链接的地址是指向站内文件还是站外，链接地址又分为相对路径和绝对路径。

- 绝对路径：指向目标地址的完整描述，一般指向本站点外的文件。

  例如搜狐。
- 相对路径：相对于当前页面的路径，一般指向本站点内的文件，所以一般不需要一个完整的 URL 地址的形式。

  例如，登录表示链接地址为：当前页面所在路径的“login”目录下的“login.htm”页面。假定当前页面所在的目录为“D:\root”，则链接地址对应的页面为“D:\root\login\login.htm”。

另外，站内使用相对路径时常用到两个特殊符号："../"表示前目录的上级目录，"../../"表示前目录的上上级目录。假定当前页面中包含两个超链接，分别指向上级目录的 web1.htm 及上上级目录的 web2.htm，如图 2.2 所示。

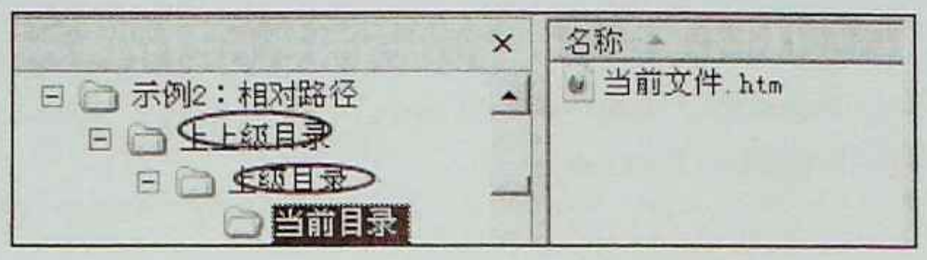


图 2.2　相对路径的上级目录与上上级目录

对应的 HTML 代码如下。

```
<a href="../web1.htm">上级目录</a>
<a href="../../web2.htm">上上级目录</a>
```

2.1.2　超链接的三类应用场合

根据超链接的应用场合，可以把链接分为如下三类。

- 页面间链接：A 页到 B 页，最常用，用于网站导航。
- 锚链接：A 页甲位置到 A 页（本页）的乙位置或 A 页甲位置到 B 页的乙位置，称为锚链接。
- 功能性链接：在页面中调用其他程序功能，例如电子邮件、QQ、MSN 等。

1. 页面间链接

如图 2.3 显示了三个页面间的链接，当前页有两个链接分别指向注册页和登录页。

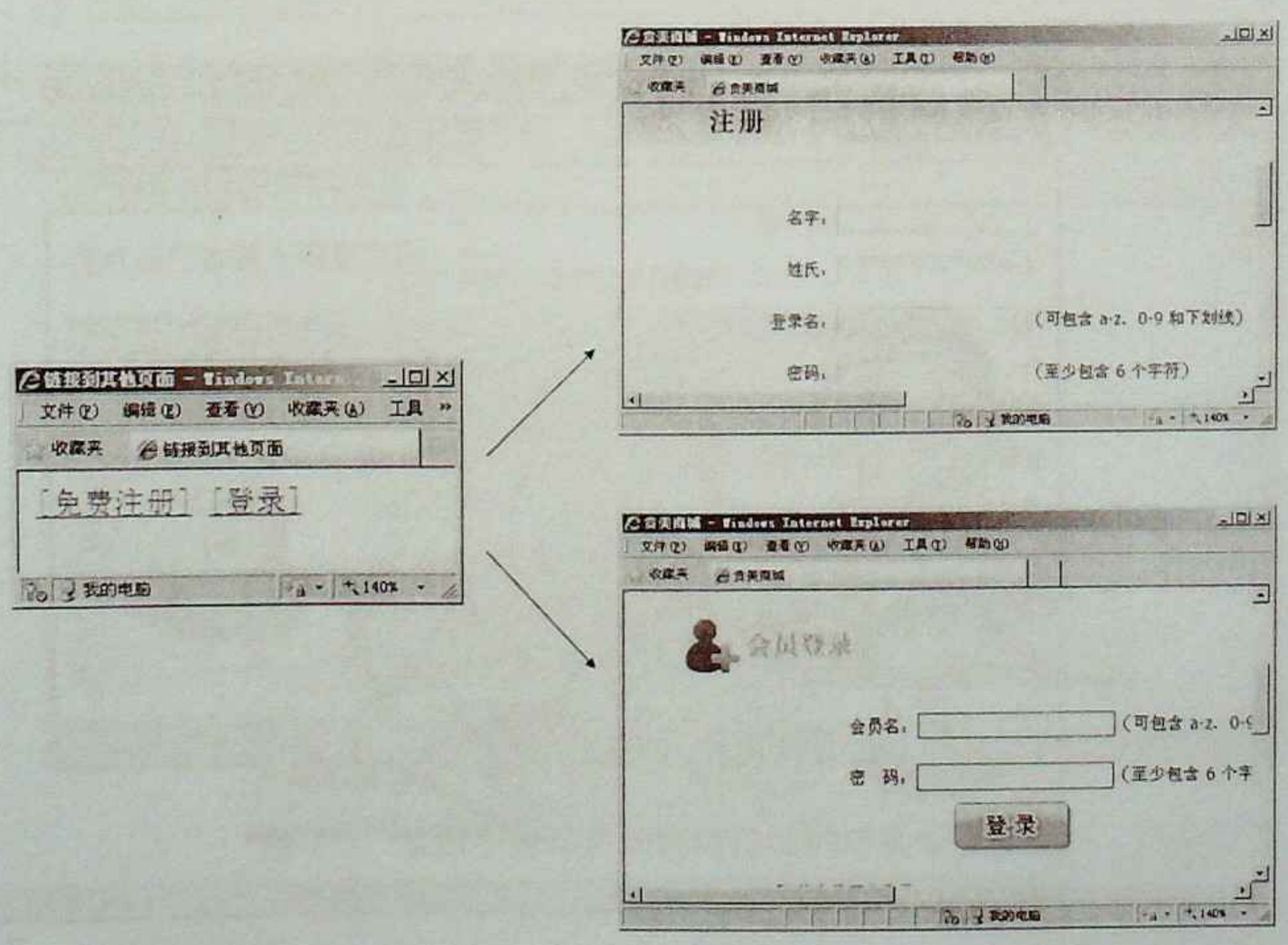


图 2.3　页面间链接

页面间链接对应的 HTML 代码如示例 1 所示，设置 href="其他页地址"即可。

示例 1

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>页面间链接</title>
  </head>
  <body>
    <a href="register/register.html">[免费注册]</a>
    <a href="login/login.htm">[登录]</a>
  </body>
</html>
```

2. 锚链接

常用于目标页内容很多，需定位到目标页内容中的某个具体位置时。例如网上常见的某个学习教程，当单击教程的章节目录时，将跳转到对应章节的内容介绍处。

如何实现从 A 页面的甲位置跳转到本页中的乙位置？实现步骤如下。

(1) 在页面的乙位置设置标记：目标位置乙，“name”为<a>标签的属性，“marker”为标记名，其功能类似古时用于固定船的锚（或钩），所以也称为锚名。

(2) 在甲位置设置 href 属性值为“#标记名”，语法如下。

```
<a href="#marker">当前位置甲</a>
```

我们先来学习如何实现同页面的锚链接，如图 2.4 所示，当单击“明星专区”超链接时，将跳转到页面下方的相关位置处。

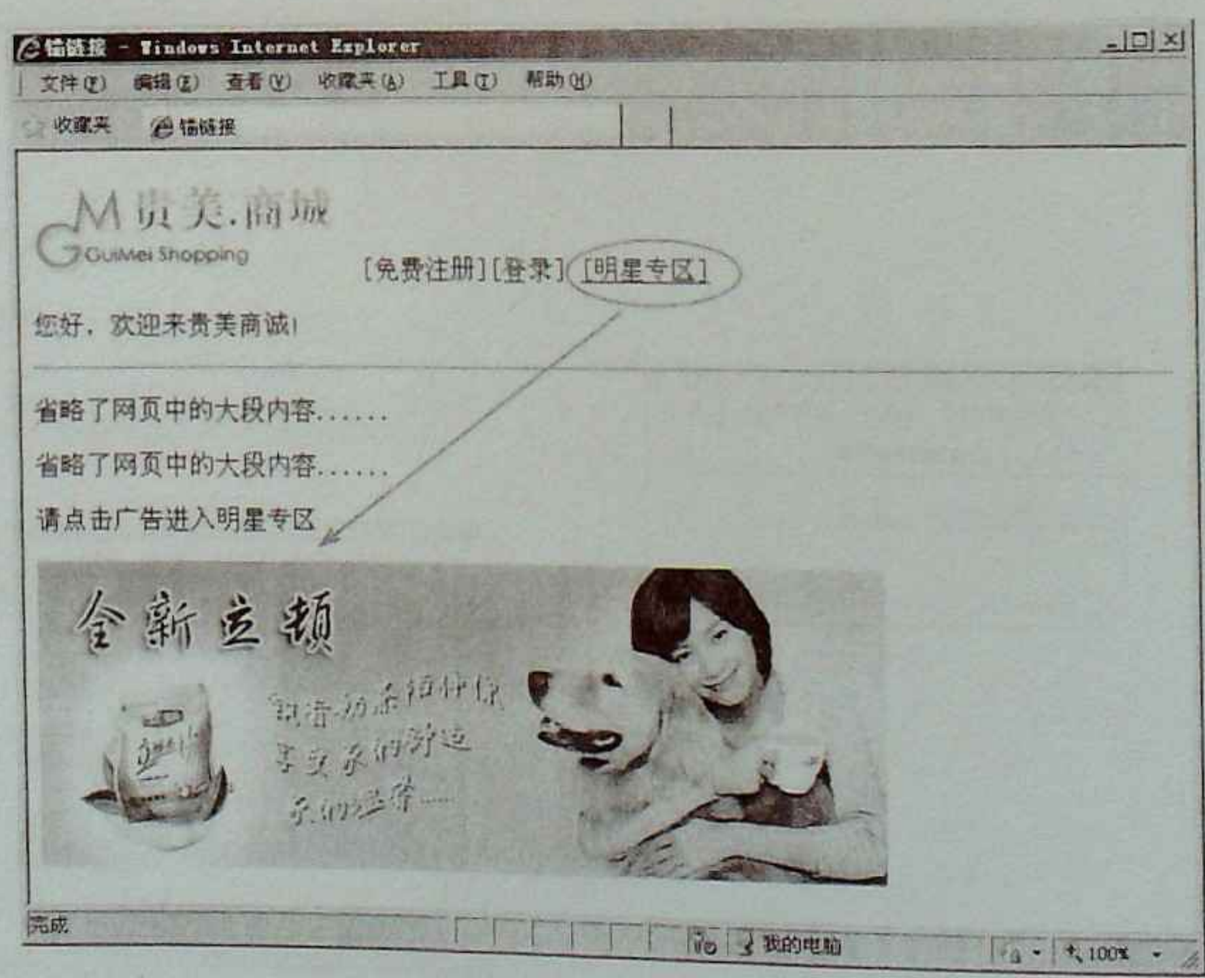


图 2.4 同页 anchor.html 锚链接

对应的 HTML 代码如示例 2 所示。

示例 2

```
<!-省略部分 HTML 代码-->
<a href="#star">[明星专区]</a>
<!-省略部分 HTML 代码,包含大量的页面内容-->
<a name="star"><img src="images/adv_2.jpg" alt="明星专区" title="明星专区"
/></a>
<!-省略部分 HTML 代码-->
```

这是同页间的锚链接，那么，如何实现不同页间的锚链接，即从 A 页甲位置跳到 B 页的乙位置，例如，单击“明星专区”链接，将跳转到另一个页面的对应内容处，如图 2.5 所示。

实现步骤和示例 2 很类似。

(1) 在 B 页乙位置设置锚标记的语法如下。

```
<a name="star">目标位置乙</a>
```

(2) 在 A 页甲位置设置链接的语法如下。

```
<a href="anchor.htm#star">明星专区</a>
```

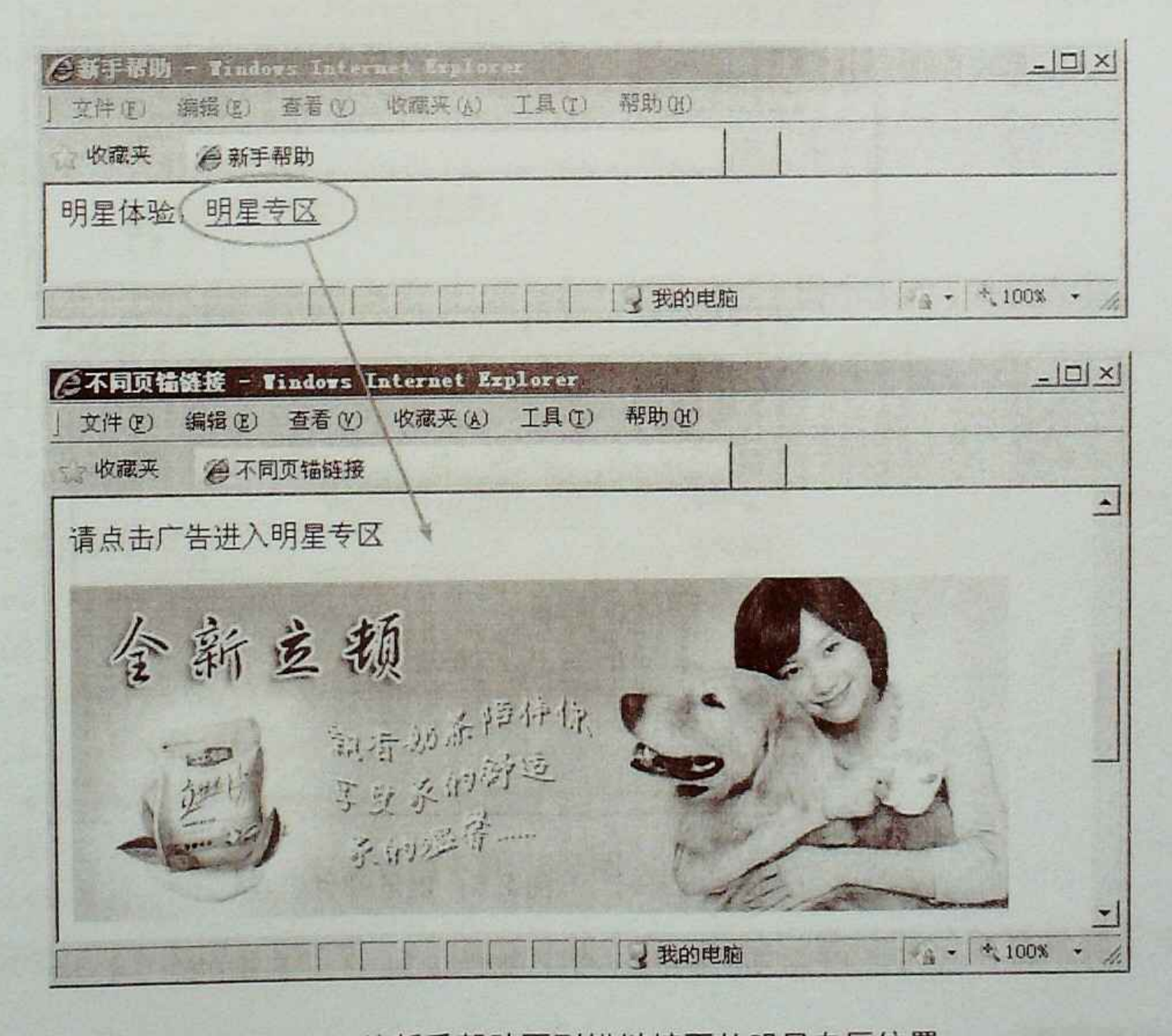


图 2.5　从新手帮助页到锚链接页的明星专区位置

两个页面对应的 HTML 代码如示例 3 所示。

示例 3

新手帮助页：help.html

```
<!-省略部分 HTML 代码-->
明星体验:<a href="anchor.html#star">明星专区</a>
<!-省略部分 HTML 代码-->
```

锚链接页：anchor.html

```
<!-省略部分 HTML 代码,包含大量的页面内容-->
<a name="star"><img src="images/adv_2.jpg" alt="明星专区" title="明星专区"
/></a>
<!-省略部分 HTML 代码-->
```

3. 功能性链接

功能性链接比较特殊，当单击链接时不是打开某个网页，而是启动本机自带的某个应用程序，例如网上常见的电子邮件、QQ、MSN 等链接。接下来以最常用的电子邮件链接为例，当单击“站长信箱”邮件链接，将打开用户的电子邮件程序，并自动填写“收件人”文本框中的电子邮件地址，如图 2.6 所示。

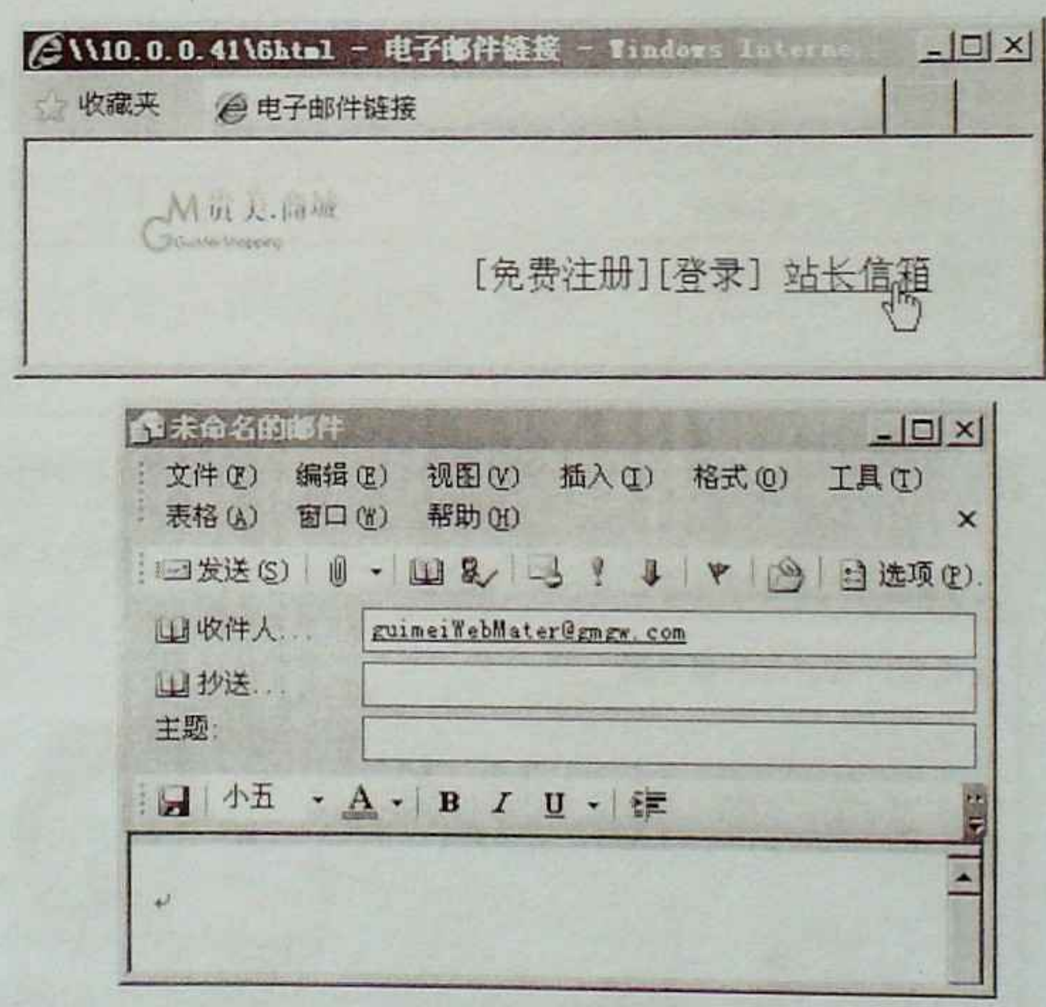


图 2.6 电子邮件链接

电子邮件链接的用法是“maito:电子邮件地址”，完整的 HTML 代码如示例 4 所示。

示例 4

```
<html>
  <head>
```

```
    <title>贵美网</title>
  </head>
  <body>
    <img src="images/logo.jpg" alt="贵美网 Logo" />[免费注册]登录]
    <a href="mailto:guimeiWebMater@gmgw.com">站长信箱</a>
  </body>
</html>
```

创建电子邮件链接

2.2 注释和特殊符号

2.2.1 注释

HTML 中的注释和 Java 中的注释类似，其作用是方便阅读和调试代码。当浏览器遇到注释时会自动忽略注释内容。HTML 的注释格式与 Java 不同，多行与单行注释都用一种格式，如下：

```
<!-- 注释内容 -->
```

和 Java 类似，当页面的 HTML 结构复杂或内容较多时，需要添加必要的注释方便阅读和维护代码。同时，有时为了调试，需要暂时注释掉一些不必要的 HTML 代码。示例 5 演示了如何添加注释和如何注释掉不需要的代码行。

示例 5

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title></title>
  </head>
  <body>
    <!--项目列表部分-->
    <div>
      <ul>
        <!-- <li>被注释掉的行将不显示</li> -->
        <li>正常显示行 1</li>
        <li>正常显示行 2</li>
      <ul>
    </div>
  </body>
</html>
```

示例 5 中加粗的“<li>”代码行已被注释掉，浏览器在解析时遇到注释将自动忽略，所以并不显示，最后的页面效果如图 2.7 所示。

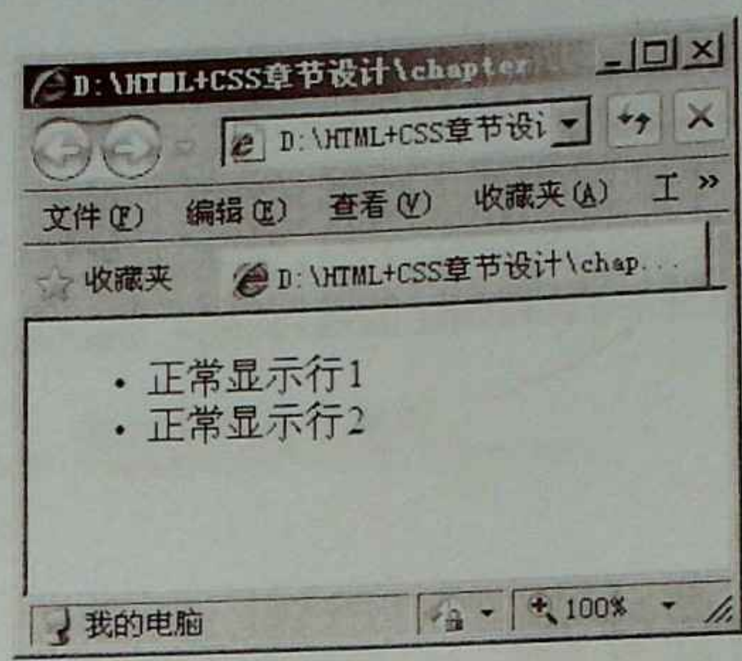


图 2.7 注释后浏览器不显示

代码规范

需要注意，XHTML 1.0 标准规定不要在注释中再出现双横线“--”。

- 不规范的注释：<!------项目列表部分--------->。
- 规范的注释：<!--#######项目列表部分##########-->。

2.2.2 特殊符号

因大于号（“>”）、小于号（“<”）等已作为 HTML 的语法符号，所以，如果要在页面中显示这些特殊符号，就必须使用相应的 HTML 代码表示，这些特殊符号对应的 HTML 代码被称为字符实体。

常用的特殊符号及对应的字符实体如表 2-1 所示，这些实体符号都以“&”开头，以“;”结束。

表 2-1 常用的特殊符号及对应的字符实体

特殊符号	字符实体	示例
空格		百度 | Google
大于号(>)	>	如果时间>晚 6 点，贾君鹏他妈就喊他回家吃饭
小于号(<)	<	如果时间<晚 6 点，贾君鹏他妈就得润喉化妆等待出场
引号(")	"	W3C 规范中，HTML 的属性值必须用成对的"号括起来
版权符号(©)	©	COPYRIGHT © 2003-2010 贵美商城 ALL RIGHT RESERVED

2.2.3 贵美商城网站的版权案例

学完注释和特殊符号后，我们来制作网站中常见的底部版权页，页面效果如图 2.8 所示。

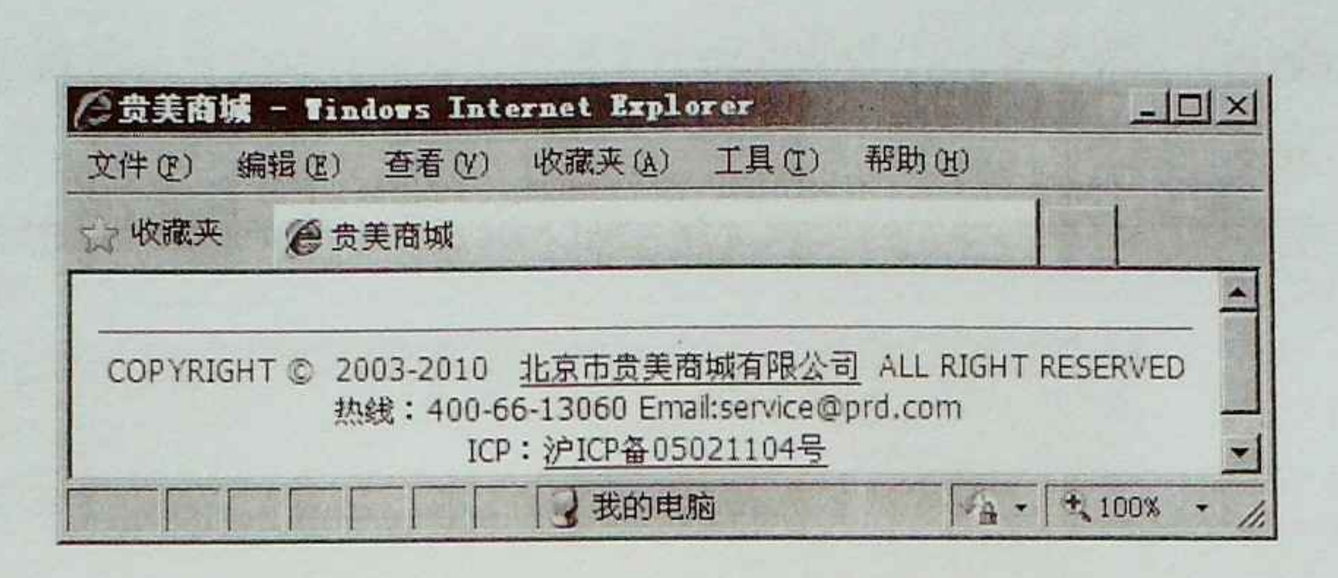


图 2.8 贵美商城网站的底部版权信息

显然，页面中包含特殊符号：版权符号（©）、空格。对应的 HTML 代码如示例 6 所示。

示例 6

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>贵美商城</title>
  </head>
  <body>
  <div style="font:12px Tahoma;margin:0px auto;text-align:center;">
        <div>
        <hr/>
        COPYRIGHT  &copy;  2003-2010    <a  href="index.htm">
北京市贵美商城有限公司</a> ALL RIGHT RESERVED<br />
        热线:400-66-13060 Email:service@prd.com<br />
        ICP:<a href="#">沪 ICP 备 05021104 号</a><br />
        </div><!—copyright    end-->
  </div><!--footer   end-->
  </body>
</html>
```

2.2.4 上机练习

上机练习 1

练习——超链接与特殊符号

需求说明

编写 HTML 代码实现导航菜单的链接，如图 2.9 所示，要求实现以下功能。

- 单击“登录”，将跳转到登录页（login.html）。在登录页中有一“返回”链接，单击返回到当前页。

- 单击“联系我们”，将自动打开本机的电子邮件程序，邮件地址自行设置。
- 单击“MOTO”，将跳转到本页面中“MOTO E2 音乐手机”的位置。
- 在底部，实现版权信息的内容。

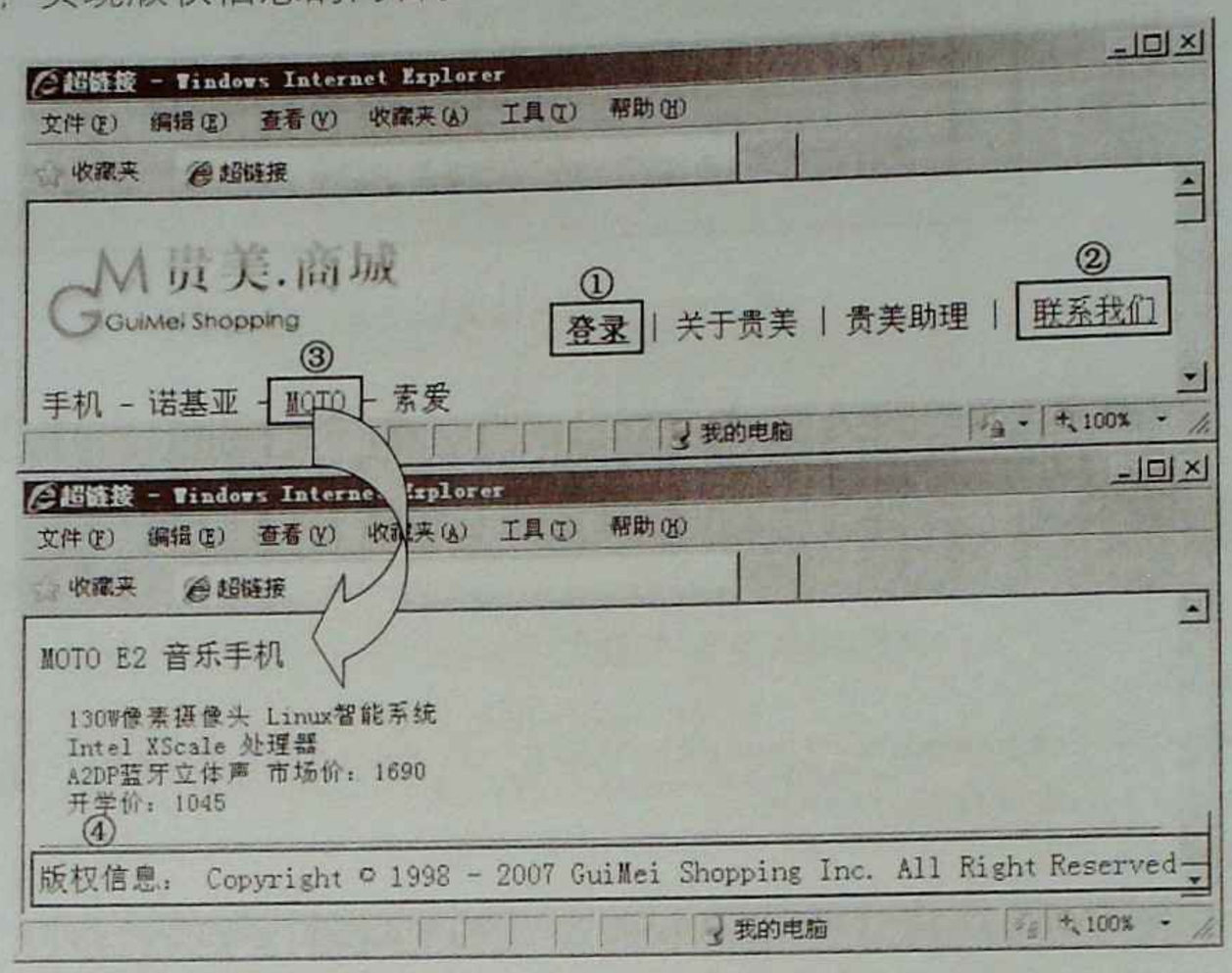


图 2.9 超链接与特殊字符

2.3 表单

在上一章，我们简单介绍了表单<form>块级标签，本小节将深入讲解表单的应用场景和具体用法。表单的应用比较常见，典型的应用场景有以下几种。

- 网站的登录、注册：登录时填写用户名，密码；注册时填写姓名、电话等个人信息。
- 网上订单：在网上购买商品，一般要求填写姓名、联系方式、付款方式等信息。
- 调查问卷：回答对某些问题的看法，以便形成统计数据，方便分析。
- 网上搜索：输入关键字，搜索想要的可用信息。

如图 2.10 所示为一个典型的注册表单，注册表单中常包含如下表单元素。

- 单行文本框：常用于输入姓名、用户名、电子邮件等。
- 密码框：为了安全起见，输入的密码显示为星号（“*”）。
- 单选按钮：在多个项目中只选一个，例如，性别的选择。
- 复选框：在多个项目中选 1 个或多个，例如，兴趣爱好的选择。
- 下拉列表：在多个项目中选 1 个，较直观，例如，月份、所在省市的选择。
- 多行文本域：用于显示或填写大段文字内容。例如，显示服务协议或填写评论。

- “提交”按钮：单击后，将用户填写的信息提交到网站服务器，网站服务器负责进一步处理。

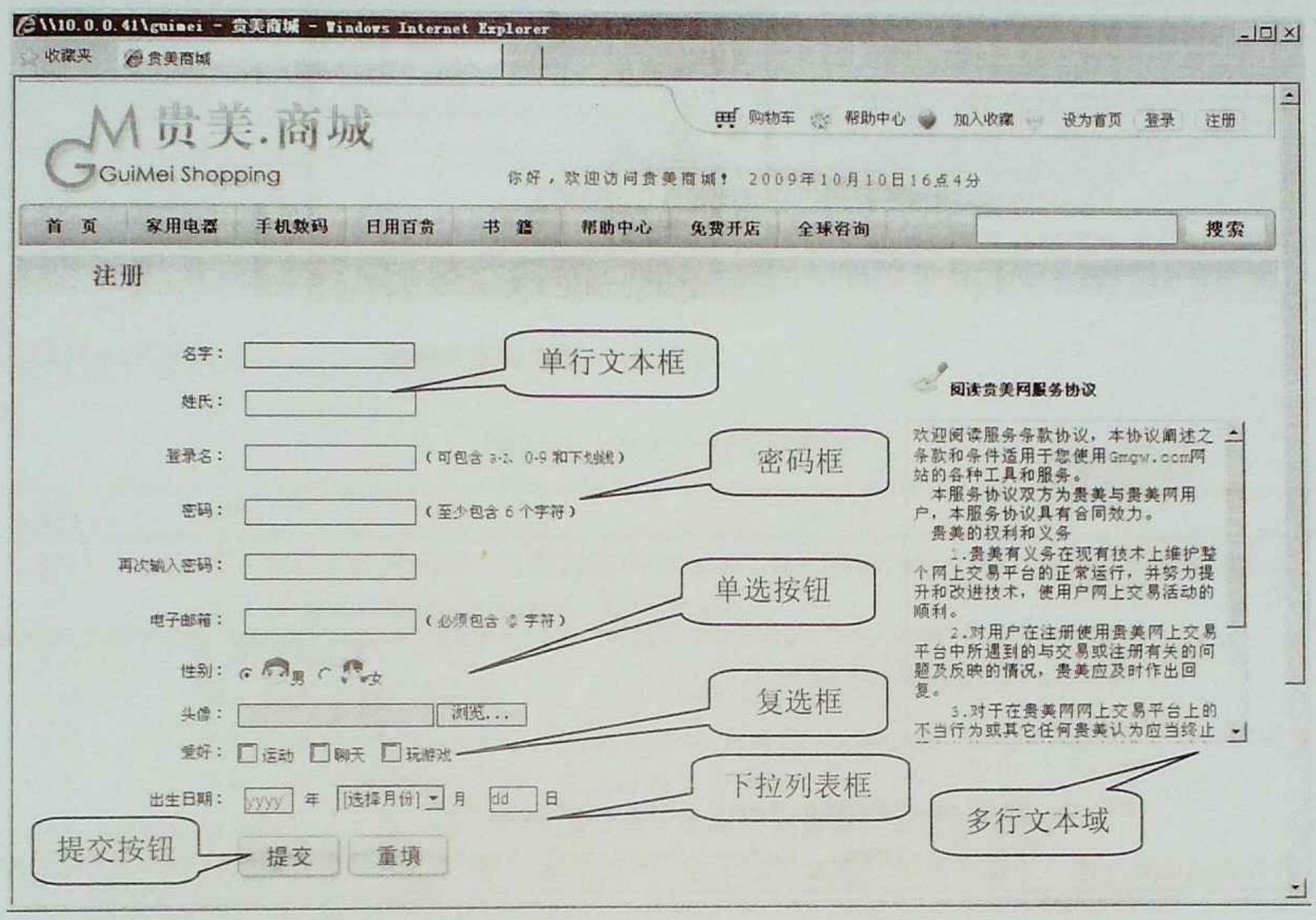


图 2.10　常见的注册表单

2.3.1　表单的基本语法

1. 基本语法

表单结构的基本语法如下。

```
<form  action="表单提交地址"  method="提交方法">
   <!--文本框、按钮等表单元素-->
</form>
```

对其属性解释如下。

- action 属性：规定当提交表单信息时，向何处发送表单数据，即处理表单数据的目标地址。如不填，默认为当前页面。
- method 属性：规定提交方式，取值为“get”或“post”。“get”方式一般适用于安全性要求不高的场合，而 post 一般适用于安全性较高的场合。两者的具体应用，将在第二学期的相关课程进行深入讲解。

下面以常见的登录表单（图 2.11）为例，介绍表单的基本结构和语法。对应的 HTML 代码如示例 7 所示。

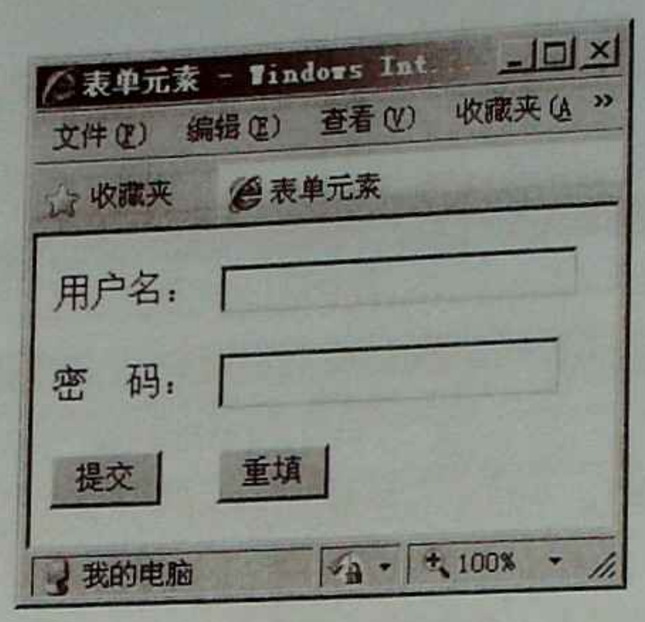


图 2.11 表单的基本结构

示例 7

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>表单元素</title>
  </head>
  <body>
    <form action="login.jsp" method="post">
      <p>用户名:<input name="username" type="text" size="20" maxlength="10"
    /></p>
      <p>密  码:<input  name="pwd" type="password" size="20"
    /></p>
      <p><input type="submit" name="btn" value=" 提 交 " />  
    <input name="reset" type="reset" value="重填" /></p>
    </form>
  </body>
</html>
```

2. 表单的执行原理

示例 7 是登录验证的表单。对比银行自动柜员机的存取款过程如下。

(1) 用户填写账号信息，提交给服务器。

(2) 服务器验证提交的账号，如不正确，则提示账号错误；如正确，则进入取钱或存钱等操作，再次让用户填写一张新的业务表单。

简单地说，表单的执行过程就是完成网站服务器器和客户端（用户计算机）之间的交互，提供双方需要的信息的过程。更直接地说，表单的执行原理类似两人之间沟通。例如，两人约定出去旅游，得先彼此问什么时候空闲，去什么地方等等，然后再决定是否去。使用表单描述沟通过程如下。

(1) A 问 B，相当于 A 给 B 一张表单。

(2) B 回答空闲时间、想去什么地方等，相当于填完这张表单，提交给 A。

(3) 然后 A 根据 B 的回答说出自己的想法。

3. 表单元素的基本格式

学习了表单的基本语法后，下面介绍表单中各表单元素的具体用法，包括文本框、按钮、下拉列表等。

除下拉列表框、多行文本域等少数表单元素外，大部分表单元素都使用<input>标签，只是它们的属性设置不同，它们的统一用法如下。

```
<input name="表单元素名称" type="类型" value="值" size="显示宽度" maxlength="
能输入的最大字符长度" checked="是否选中" />
```

各属性的具体含义如表 2-2 所示。

表 2-2 <input>元素的属性

属性	说明
type	指定表单元素的类型，可用的选项有 text、password、checkbox、radio、submit、reset、file、hidden、image 和 button，默认为 text
name	指定表单元素的名称
value	指定表单元素的初始值
size	指定表单元素的初始宽度。如果 type 为 text 或 password，则表单元素的大小以字符为单位；对于其他输入类型，宽度以像素为单位
maxlength	指定可在 text 或 password 元素中输入的最大字符数，默认不做限制
checked	此属性只有一个值，为“checked”，表示选中状态，如果不选中，则不需添加此属性

下面将逐一介绍表单中常用到的表单元素。鉴于表单元素较多，为了便于边学边练，接下来按照由易到难的顺序，分为多个部分逐一介绍。

2.3.2 表单元素的介绍（一）

1. 文本框

表单中最常用的输入元素就是文本框(Text)，它提供给用户输入单行文本信息。例如，用户名的输入框。文本框的用法很简单，只需将<input>标签的“type”属性设为“text”即可。

```
<form method="post" action="">
  <p>用户名:
    <input  type="text"  value="zhangsan"  size="20"  name="userName" />
  </p>
</form>
```

文本框　文本框初始值　文本框名字为 name

2. 密码框

将表单元素的 type 属性设为 password 就可以创建一个密码框，输入的字符全都以“*”显示，提高数据安全。

```
<p>密码:
  <input  type="password"  size="10"  name="pass" />
</p>
```

3. 重置、提交与普通按钮

根据按钮的功能，分为提交按钮、重置按钮和普通按钮。提交按钮用于提交表单数据；重置按钮用于清空现有表单数据；普通按钮一般用于调用 JavaScript 脚本（第二学期讲解）。在用法上，设置“type”属性对应的类型即可。

```
<p>
  <input  type="submit"  value="提交按钮"  name="button" />
  <input  type="reset"   value="重置按钮"  name="reset" />
  <input  type="button"  value="普通按钮"  name="cancel" />
</p>
```

示例 8 中列举了几种类型的表单按钮，显示效果如图 2.12 所示。

示例 8

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>文本框,密码框和按钮</title>
  </head>
  <body>
    <form action="" method="post">
      <p>用户名:
          <input name=" userName" type="text" size="21" />
      </p>
      <p>密  码:
          <input name="pass" type="password" size="22" />
      </p>
      <p>
          <input type="reset" name="reset" value=" 重填 " />
          <input type="submit" name="button" value="同意以下服务条款,提交注册
      信息" />
      </p>
      <p>
          <input  type="button" name="confirm" value="点播音乐" />
          <input  type="button" name="cancel" value="取消" />
      </p>
    </form>
  </body>
</html>
```

文本框　重置按钮　密码框　提交按钮　普通按钮

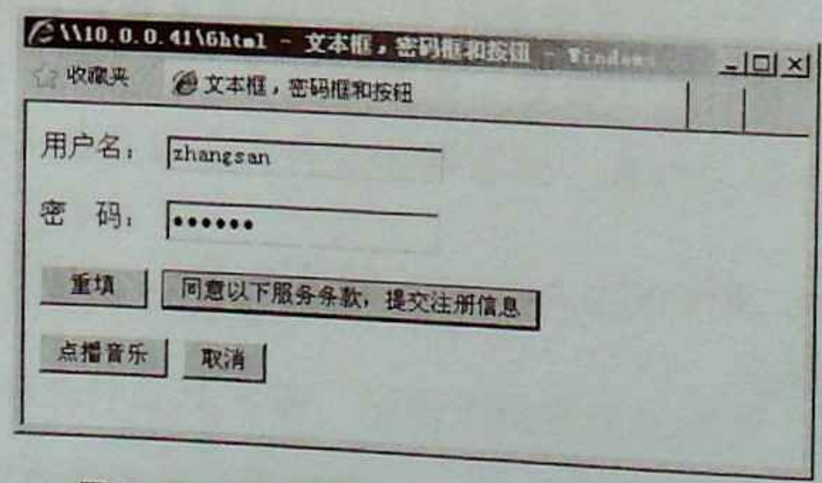


图 2.12　文本框、密码框和按钮的效果

上图的提交按钮比较难看，实际应用中，经常用图片按钮来代替，如图 2.13 所示。实现图片按钮的效果有多种方法，比较简单的方法就是配合使用“type”和“src”属性。例如：

```
<input  type="image"  src="images/login.gif" />
```

需要注意：这种方式实现的图片按钮比较特殊，虽然“type”属性没有设置为“submit”，但仍然具备提交功能。

4. 单选按钮

单选按钮用于一组相互排斥的选项，组中的每个选项应具有相同的名称（Name），以确保用户只能选择一个选项，单选按钮对应的“type”属性为“radio”。例如，实现性别的选择（图 2.14），对应的 HTML 关键代码如示例 9 所示。

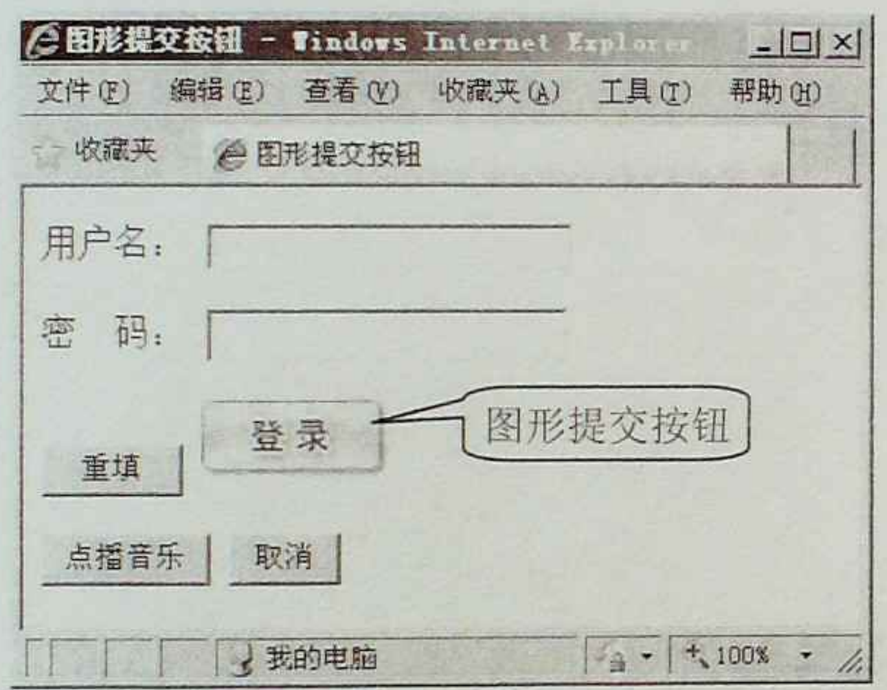


图 2.13　图形提交按钮

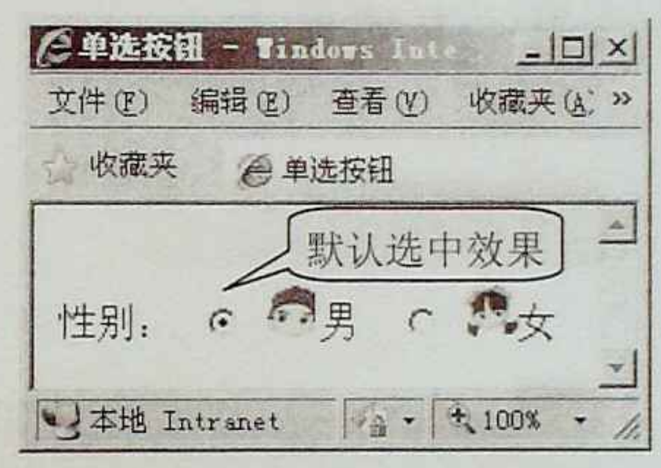


图 2.14　单选按钮的效果

示例 9

```
< !—省略部分 HTML 代码-->
<form method="post" action="">
    性别：
    <input name="gen" type="radio" class="input" value="男"
      checked="checked">  ←——使用 checked 选中属性设置初始状态是否被选中
    <img src="images/male.gif" alt="女" />男 
    <input name="gen" type="radio" value="女" class="input">
    <img src="images/female.gif" alt="女" />女
</form>
< !—省略部分 HTML 代码-->
```

5. 复选框

复选框用于选择多个选项，将 input 的 type 属性设为“checkbox”就可以创建一个复选框。例如，爱好方面的选择（图 2.15），对应的 HTML 代码如示例 10 所示。

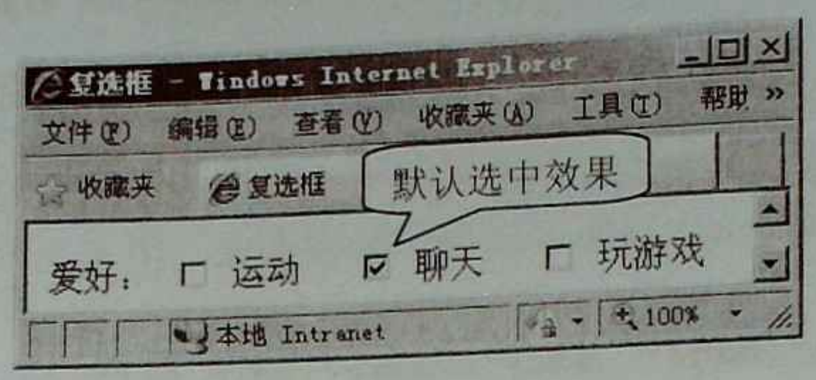


图 2.15 复选框的效果

示例 10

```
< !—省略部分 HTML 代码-->
<form method="post" action="">
  爱好:
  <input type="checkbox" name="hobby" value="sports" />运动  
  <input type="checkbox" name="hobby" value="talk" checked="checked" />
        聊天  
  <input type="checkbox" name="hobby" value="play" /> 玩游戏
</form>
< !—省略部分 HTML 代码-->
```

2.3.3 上机练习

上机练习 2

练习——贵美商城网站登录页

需求说明

使用所学的表单元素相关知识，制作贵美商城网站登录页，如图 2.16 所示。

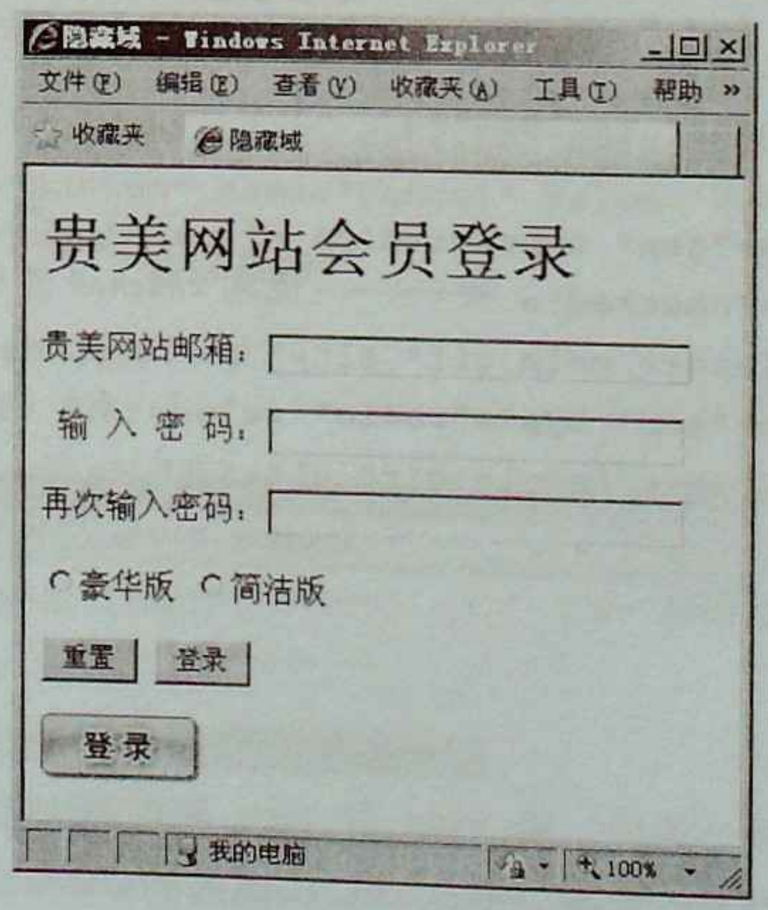


图 2.16 贵美商城网站登录页

2.3.4 表单元素的介绍（二）

1. 文件域

文件域用于上传文件（图 2.17），设置时只需把 type 属性设为“file”即可。

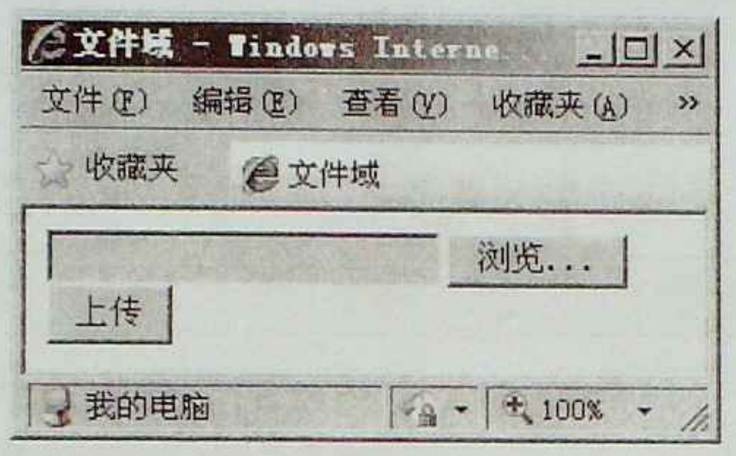


图 2.17 文件域的应用

如图 2.18 所示，文件域会创建一个不能输入内容的地址文本框和一个“浏览”按钮。单击“浏览...”按钮，将会弹出“选择要加载的文件”窗口，选择文件后，路径将显示在地址文本框中。对应的 HTML 关键代码如示例 11 所示。

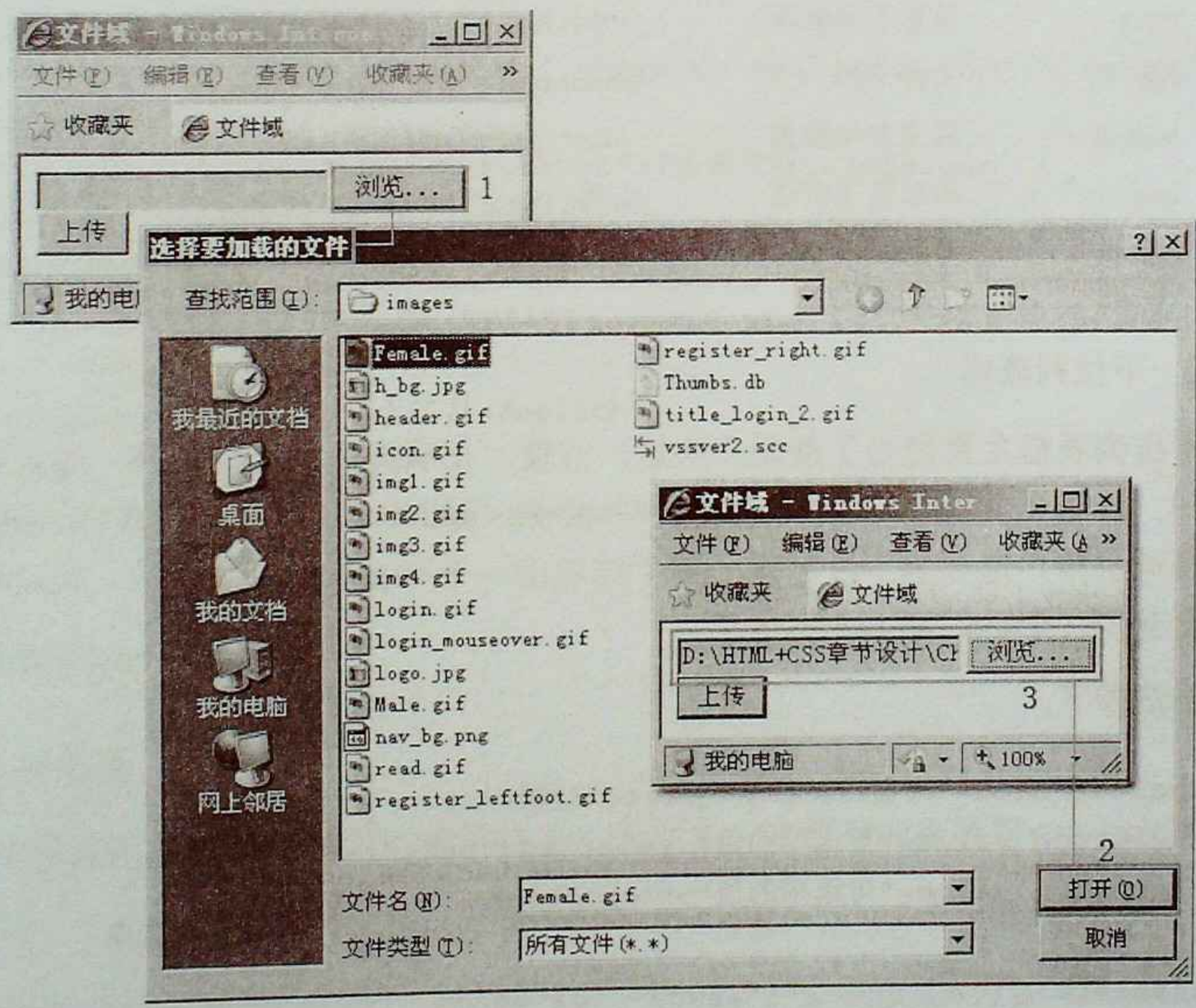


图 2.18 文件域与上传操作

示例 11

```
<!—省略部分 HTML 代码-->
<form action="" method="post" enctype="multipart/form-data">
```

```
  <p><input type="file" name="files" /><br />
  <input type="submit" name="upload" value="上传" /></p>
</form>
<!—省略部分 HTML 代码-->
```

需要注意：包含文件域的表单，由于提交的表单数据包括普通的表单数据、文件数据等多部分内容，所以必须设置表单的“enctype”编码属性为“multipart/form-data”，表示将表单数据分为多部分提交。

通过上面的学习，我们再来小结表单元素<input>的用法，如表 2-3 所示。

表 2-3　表单元素<input>的用法

类型	功能	例子
text	单行文本输入	<input **type="text"** name="username" />
password	密码	<input **type="password"** name="password" />
radio	单选	<input **type="radio" name="sex"** value="男" />男
		<input **type="radio" name="sex"** value="女" />女
checkbox	多选	<input type=**"checkbox"** name="hobby" value="书" />书
		<input type=**"checkbox"** name="hobby" value="画" />画
		<input type=**"checkbox"** name="hobby" value="琴" />琴
reset	重置表单数据	<input **type="reset"** value="重置" />
file	文件上传	<input **type="file"** name="files" />
submit	提交表单数据	<input **type="submit"** value="提交" />
image	图形提交按钮	<input **type="image"** src="images/button.gif" />
button	普通按钮	<input **type="button"** value="播放音乐" />

2. 下拉列表框

下拉列表框主要是为了使用户快速、方便、正确地选择一些选项，而且还能节省页面空间，如图 2.19 所示。它是通过<select>和<option>标签来实现的，<select>标签用于显示可供用户选择的下拉列表，每个选项由一个<option>标签表示，<select>标签必须包含至少一个<option>标签。

语法如下。

```
<select name="指定列表名称" size="行数">
  <option value="可选项的值" selected="selected">…</option>
  <option value="可选项的值">…</option>
  ……
</select>
```

默认选中

其中，在有多种选项可供用户滚动查看时，“size”属性确定列表中可同时看到的行数；“selected”属性表示该选项在默认情况下是被选中的，而且一个列表框中只能有一个列表项默认被选中，如同单选按钮组那样。

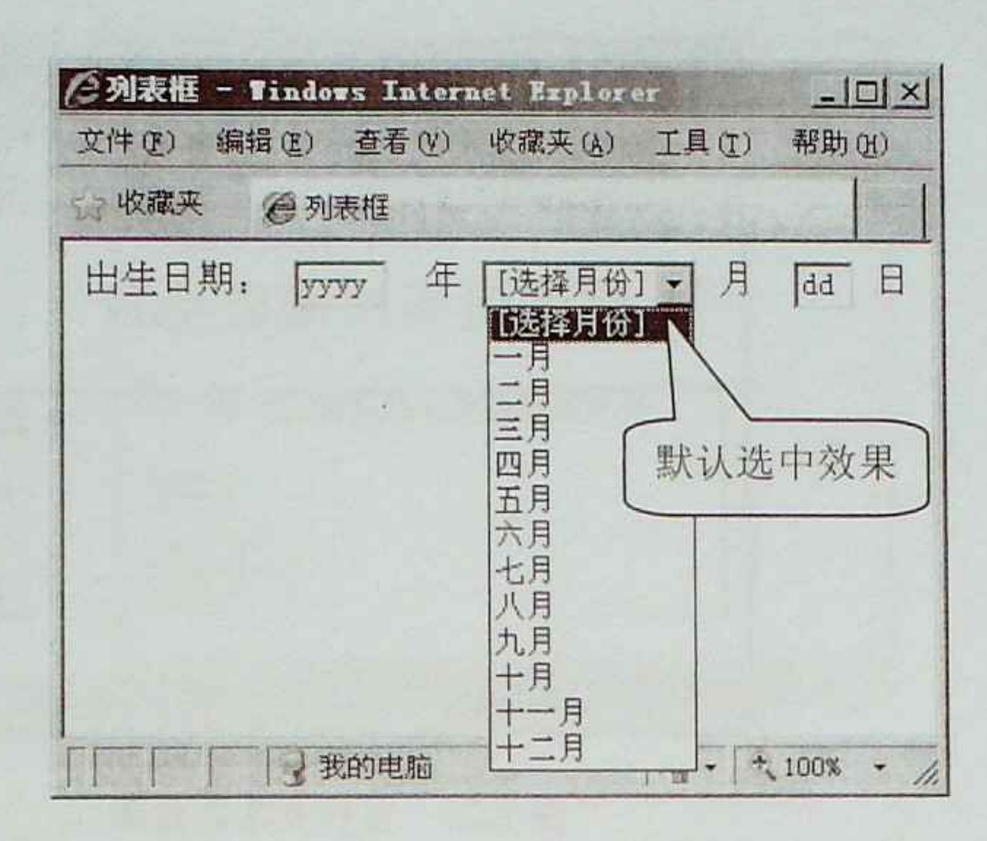


图 2.19　下拉列表框效果

图 2.19 对应的 HTML 关键代码如示例 12 所示。

示例 12

```
<!--省略部分 HTML 代码-->
<form method="post" action="">
<p>出生日期:
  <input name="byear" value="yyyy" size="4" maxlength="4" />  年
  <select name="bmon">
  <option value="" selected="selected">[选择月份]</option>
      <option value="0">一月</option>
      <option value="1">二月</option>
      <option value="2">三月</option>
      <!--省略其他月份的选项-->
      <option value="11">十二月</option>
  </select> 月 
  <input name="bday" value="dd" size="2" maxlength="2" /> 日
</p>
</form>
< !--省略部分 HTML 代码-->
```

默认选中

3. 多行文本域

多行文本域用于显示或输入两行或两行以上的文本，它使用的标签是<textarea>。

语法如下：

```
<textarea  name="textarea"  cols="40"  rows="6"> 初始文本内容 </textarea>
```

其中，cols 用来指定多行文本域的列数，rows 用来指定多行文本域的行数。在<textarea>...</textarea>标签对中不能使用 value 属性来赋初始值。

例如，实现如图 2.20 所示的服务协议显示，对应的 HTML 关键代码如示例 13 所示。

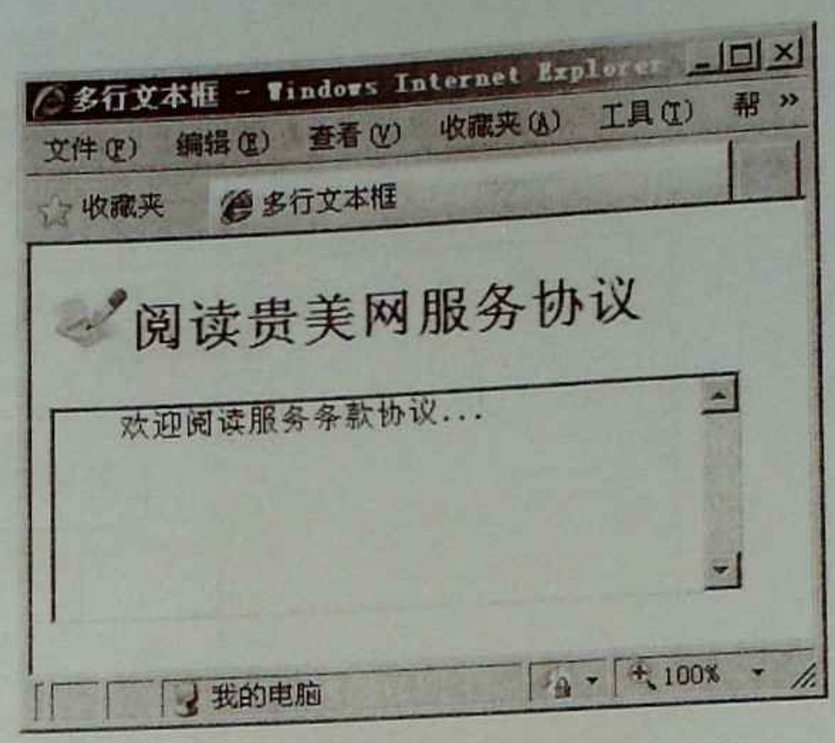


图 2.20 多行文本域效果

示例 13

```
<!--省略部分 HTML 代码-->
<form method="post" action="">
  <h4>
    <img  src="images/read.gif" alt="阅读协议" title="阅读协议" />
    阅读贵美网服务协议
  </h4>
  <p>
    <textarea  name="textarea"  cols="40"  rows="6"> 欢迎阅读服务条款协议...
    </textarea>
</p>
</form>
<!--省略部分 HTML 代码-->
```

6 行 40 个字符宽度的多行文本框

2.3.5 表单的高级用法

1. 隐藏域

网站服务器端发送到客户端（用户计算机）的信息，除用户直观看到的页面内容外，可能还包含一些“隐藏”信息。例如用户登录后的用户名、用于区别不同用户的用户 id 等。这些信息对于用户可能没用，但对网站服务器有用。所以一般“隐藏”起来，而不在页面中显示。

将“type”属性设置为“hidden”隐藏类型即可创建一个隐藏域。例如，在登录页中隐藏用户 id 信息“666”：

```
<input type="hidden" name="userid" value="666" />
```

页面显示的结果如图 2.21 所示，我们将发现隐藏信息并不显示，但能通过页面的 HTML 代码查看到。

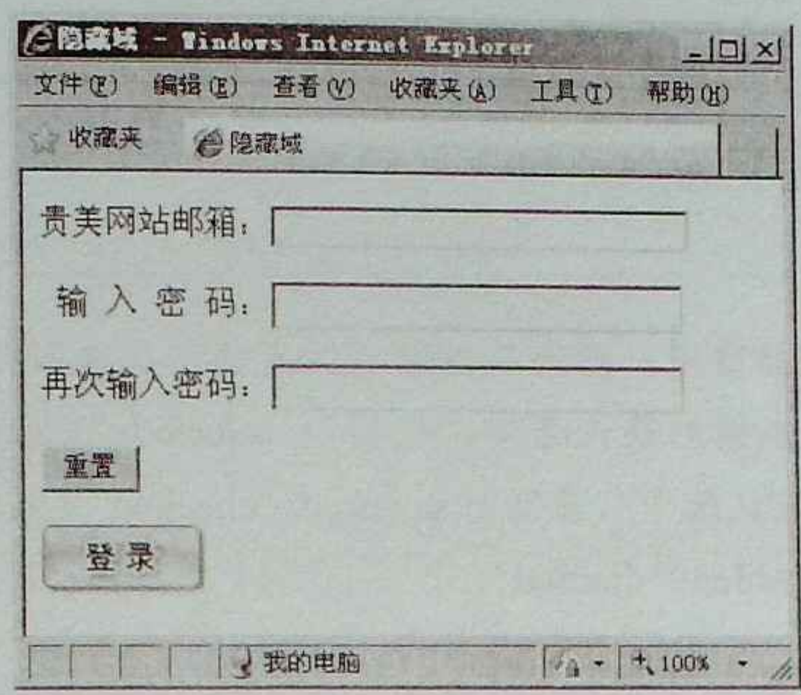


图 2.21　隐藏域并不显示在页面中

2. 只读和禁用属性

在某些情况下，我们需要对表单元素进行限制，设置表单元素为只读或禁用，如图 2.22 所示。它们常见的应用场景如下。

- 只读场景：网站服务器方不希望用户修改的数据，这些数据在表单元素中显示。例如：注册或交易协议、商品价格等。
- 禁用场景：只有满足某个条件后，才能选用某项功能。例如，只有用户同意注册协议后，才允许单击“注册”按钮；播放器控件在播放状态时，不能再单击“播放”按钮等。

只读和禁用效果分别通过设置“readonly”和“disabled”属性。例如，要实现协议只读和禁用属性注册按钮的效果，如图 2.22 所示，对应的 HTML 代码如示例 14 所示。

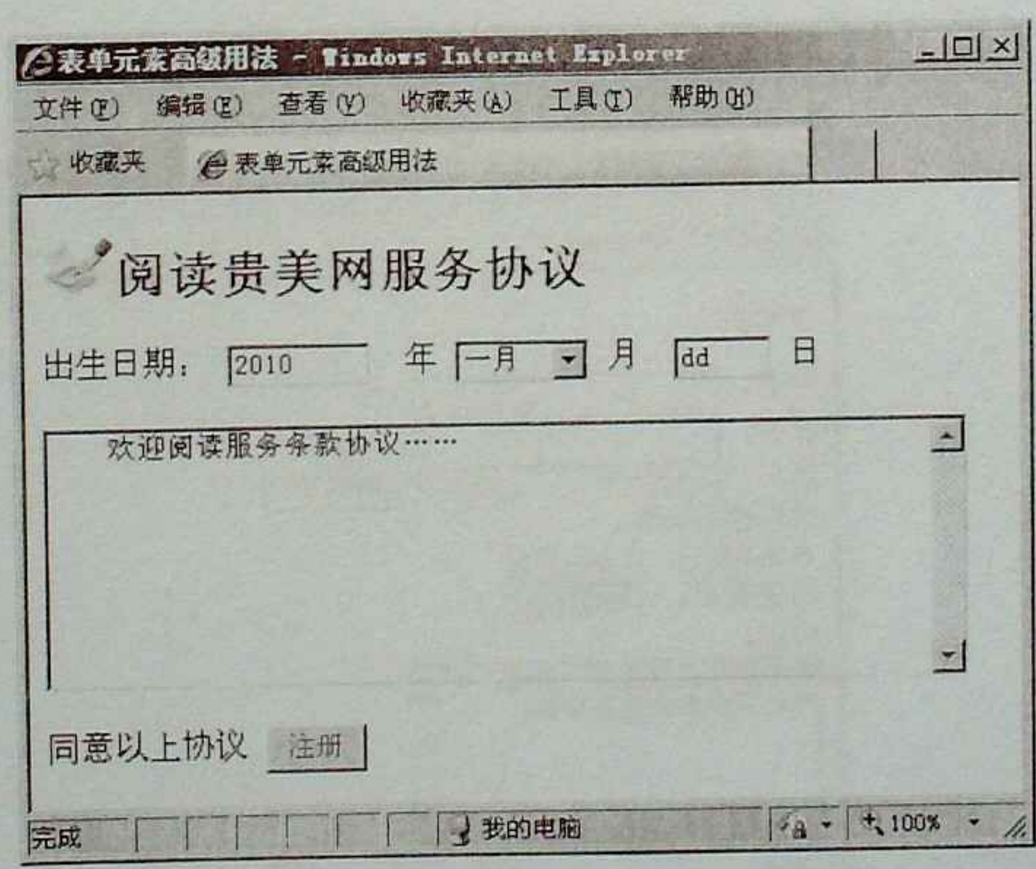


图 2.22　只读和禁用属性

示例 14

```
<!--省略部分 HTML 代码-->
 <textarea name="content" cols="60" rows="8" readonly="readonly">
    欢迎阅读服务条款协议……
```

```
    </textarea><br /><br />
    <input name="btn" type="submit" value="注册" disabled="disabled" />
<!--省略部分 HTML 代码-->
```

代码规范

W3C XHTML 标准中，规定属性值不能省略，要求写为属性="属性值"的形式。例如，下拉列表框的默认选中，应写为 selected="selected"，而不能仅写 selected。同理，复选框的默认选中，应写为 checked="checked"，只读应写为 readonly="readonly"，禁用应写为 disabled="diabled"。

2.3.6 上机练习

上机练习 3

练习——简易求职表

需求说明

制作简易求职表，如图 2.23 所示，具体要求如下。

- 实现如图 2.23 所示的常见功能。
- "协议"只读，"提交"按钮禁用。
- 用隐藏域提交求职者的用户名，例如"zhangsan_007"。

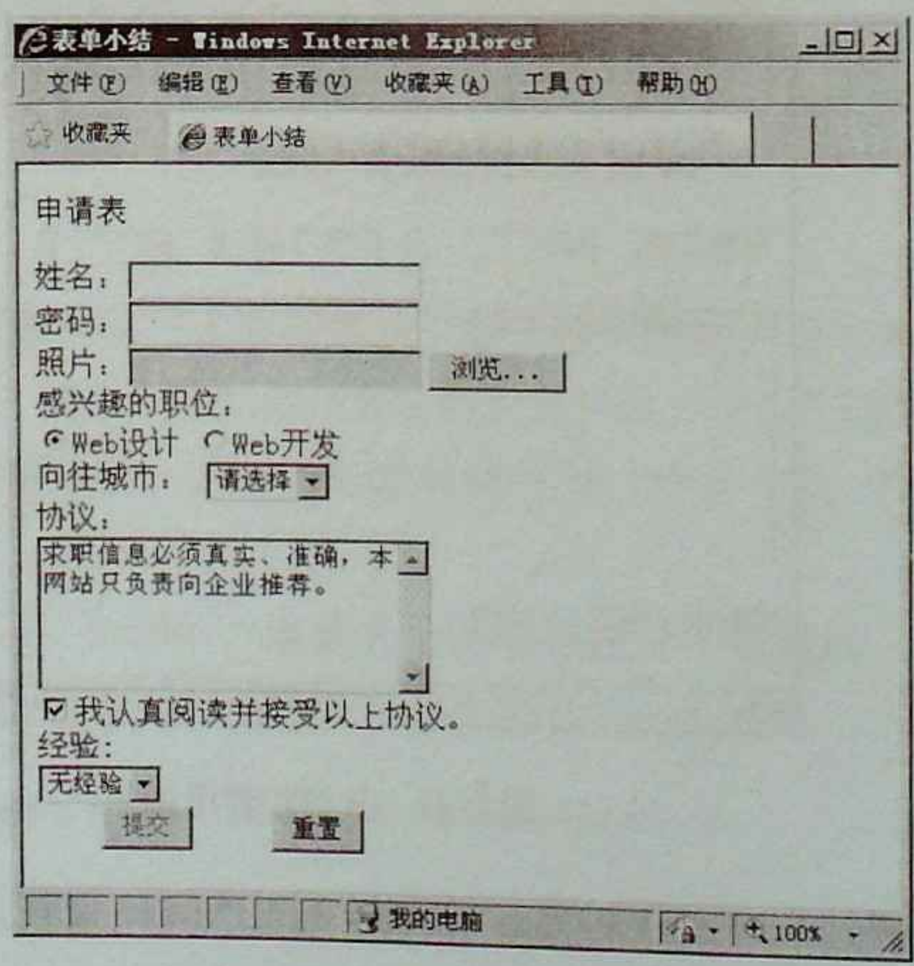


图 2.23 简易求职表

本章总结

- 超链接标签<a>用于建立页面间的导航链接，链接可分为页面间链接、锚链接、功能性链接。
- 注释用于提高 HTML 代码的可读性，字符实体用于表示“>”、“<”等特殊符号。
- 表单<form>的常用属性包括 action 和 method。
- 大部分的表单元素使用<input>标签表示，通过设置“type”值来分别实现如下几项。
  - 文本框、密码框、隐藏域
  - 提交按钮、重置按钮、普通按钮、图片提交按钮
  - 单选按钮、复选框
- 其他表单元素还包括下拉列表框、多行文本域、文件域等。
- 表单的高级用法是设置表单元素的只读、禁用、隐藏状态。

本章作业

一、选择题

1. 在 HTML 中有效、规范的注释声明是（　　）。

A. //这是注释

B. <!-- 这是--注释 -->

C. /*这是注释*/

D. <!-- 这是注释 -->

2. 下列说法正确的是（　　）。

A. 密码框需要设置 input 标签 type="password"

B. 图形提交按钮也需要设置 type="submit"

C. 提交方法 post 比 get 更安全

D. value 属性表示初始值，可能会随着用户的操作而改变，以提交时为准

3. 下面说法正确的是（　　）。

A. 文件域所传的内容较复杂，编码需要设为 enctype="multipart-form-data"

B. 复选框各个选项的 name 属性必须设置为相同

C. readonly 属性表示是否禁用

D. 为了实现性别判断用两个单选按钮，name 分别设置为“男”和“女”

4. 列表框的默认选择属性符合规范的正确写法为（　　）。

A. selected="selected"

B. selected

C. checked="checked"

D. selected="true"

5. 关于 form 表单传值，说法正确的是（　　）。

A. name 属性是表单传值的标识，网站服务器据此来判断是哪个表单元素的值

B. action 是<form>标签的必须属性，表示提交给网站服务器的具体处理程序地址

C. 通过用户填写表单提交到服务器，服务器再做出反馈，从而实现客户端-服务器之间的交互

D. 隐藏域包含的数据不会在页面中显示，通过查看网页源码也无法看到

二、简答题

1. 超链接的类型有几种？它们的区别是什么？

2. 根据你的理解，请画图说明表单的执行原理，描述客户端（用户计算机）和服务器的交互过程。

3. 用超链接模仿实现网上学习教程或网上小说阅读（类似图 2.24 的效果），题材可以根据自己的爱好任选，具体要求如下。

- 首页分两部分：前一部分是章节目录和广告图片，后一部分是每章的具体内容。
- 使用锚链接实现：单击章节目录，将跳转到本页面的对应章节内容。
- 使用超链接实现：单击广告图片，在新开的窗口中打开广告页面。

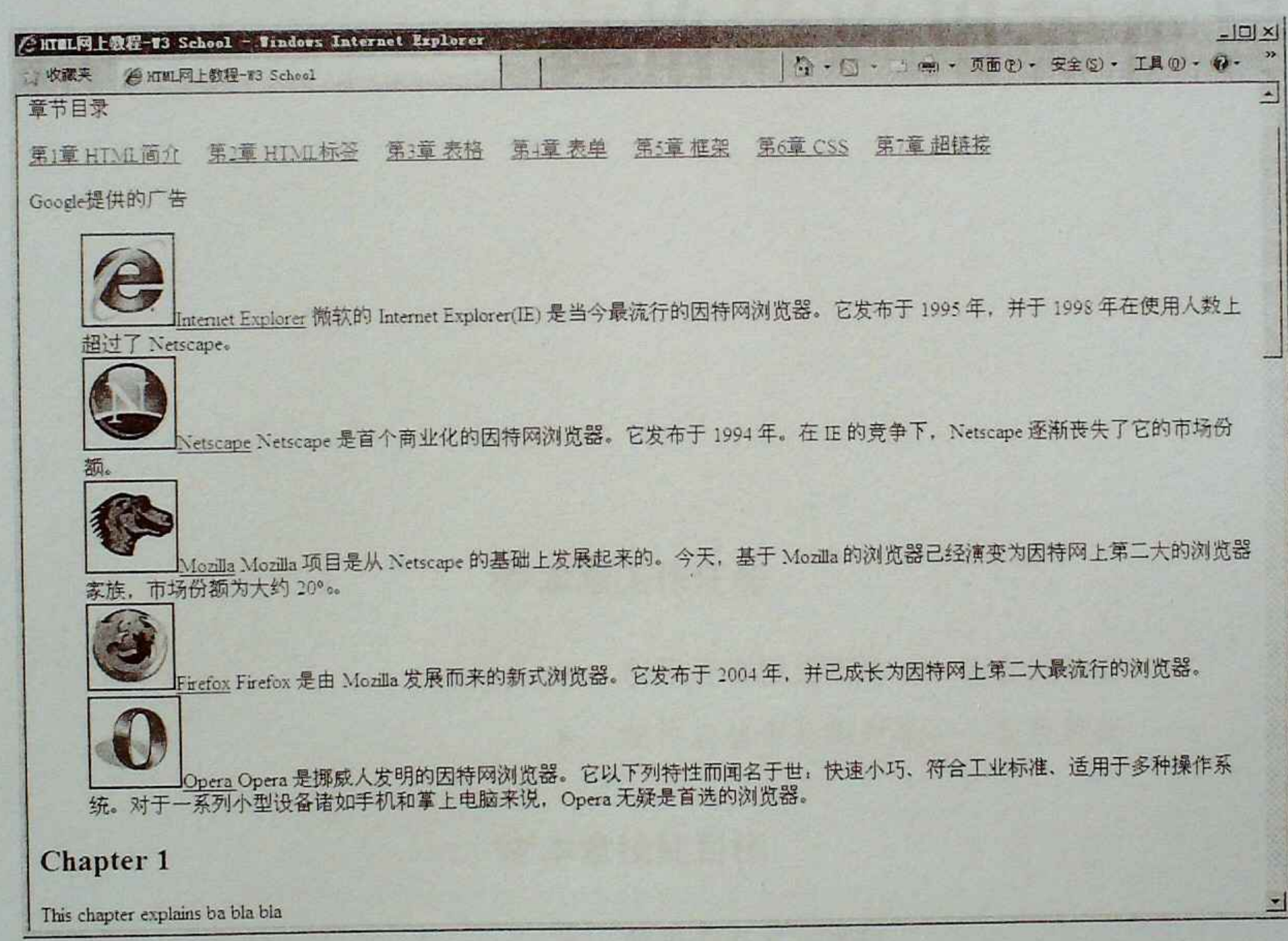


图 2.24　学习教程

4. 手写一个注册页面，包含用户名、密码、性别、照片、爱好、email、所在省市和协议，并使用隐藏域记录用户的来源（假设为电视广告），如图 2.25 所示。

5. 表单元素旁边的提示文字（例如姓名输入框旁边的提示文字“姓名”），XHTML 标准推荐使用<label>标签实现，这样的 HTML 结构更具有语义化。请上网查阅相关资料，了解使用<label>标签的好处，并用<label>标签重新实现本章示例 7 的登录页面。

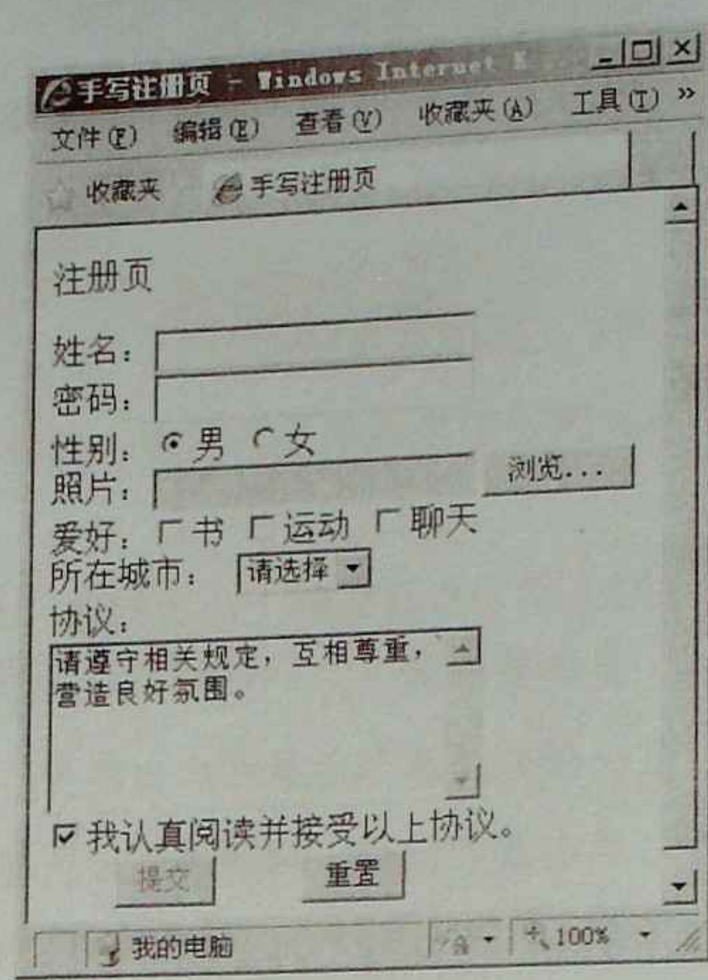

手写注册页 - Windows Internet E
文件(F) 编辑(E) 查看(V) 收藏夹(A) 工具(T)
收藏夹
手写注册页
注册页
姓名：
密码：
性别： 男 女
照片：
浏览...
爱好： 书 运动 聊天
所在城市： 请选择
协议：
请遵守相关规定，互相尊重，
营造良好氛围。
我认真阅读并接受以上协议。
提交
重置
我的电脑
100%

图 2.25　注册页的效果

第 3 章

表格的应用和布局

◈本章工作任务

- 制作表格图文布局——公告栏页面
- 制作表格布局的表单——登录页面

◈本章技能目标

- 使用表格显示数据
- 使用表格实现页面布局

本章单词

请在预习时学会下列单词的含义和发音，并填写在横线处。

1. row：_____
2. column：_____
3. cell：_____
4. spacing：_____
5. border：_____
6. width：_____
7. height：_____
8. padding：_____

本章简介

前两章我们学习了 HTML 的基本标签及表单，本章将学习另一个块级标签——表格<table>。本章将介绍表格的基本用法，并使用表格实现图文内容和表单的布局。其中重点是表格的基本结构和表格的布局思路。

3.1 表格基础

表格是块状元素，发明该标签的初衷是用于显示表格数据。例如学校中常见的考试成绩单、选修课表、企业中常见的工资账单等。

3.1.1 为什么要使用表格

1. 简单通用

由于表格行列的简单结构，且在生活中的广泛使用，对它的理解和编写都很方便。

2. 结构稳定

表格每行的列数通常一致，同行单元格高度一致且水平对齐，同列单元格宽度一致且垂直对齐。这种严格的约束形成了一个不易变形的长方形盒子结构，堆叠排列起来结构很稳定。

基于以上两点，表格在网页技术发展的初中期就被广泛应用于论坛（图 3.1）、政务网站（图 3.2）和咨讯网站（图 3.3）中。

图 3.1 论坛页面

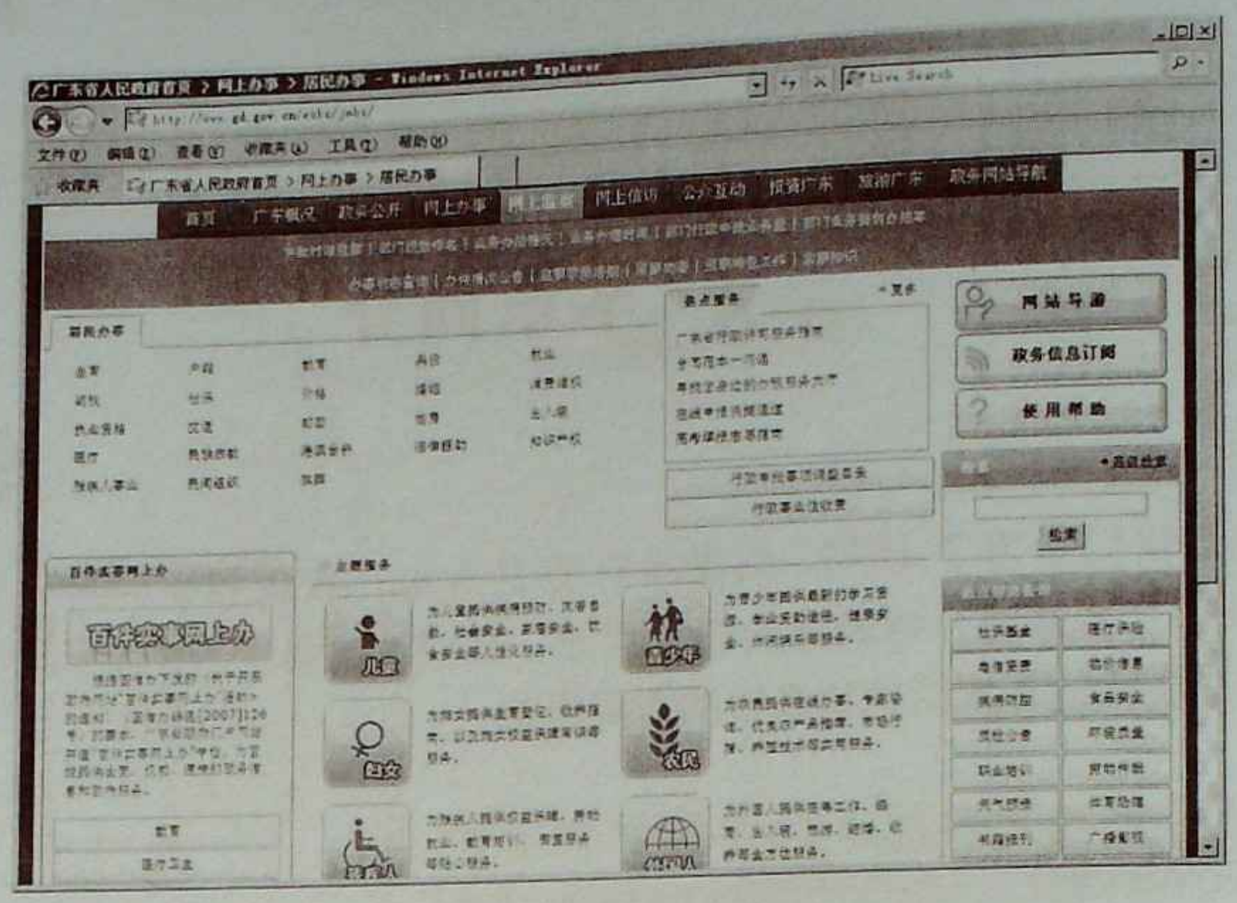


图 3.2　早期的政务网站页面（目前已改版升级）

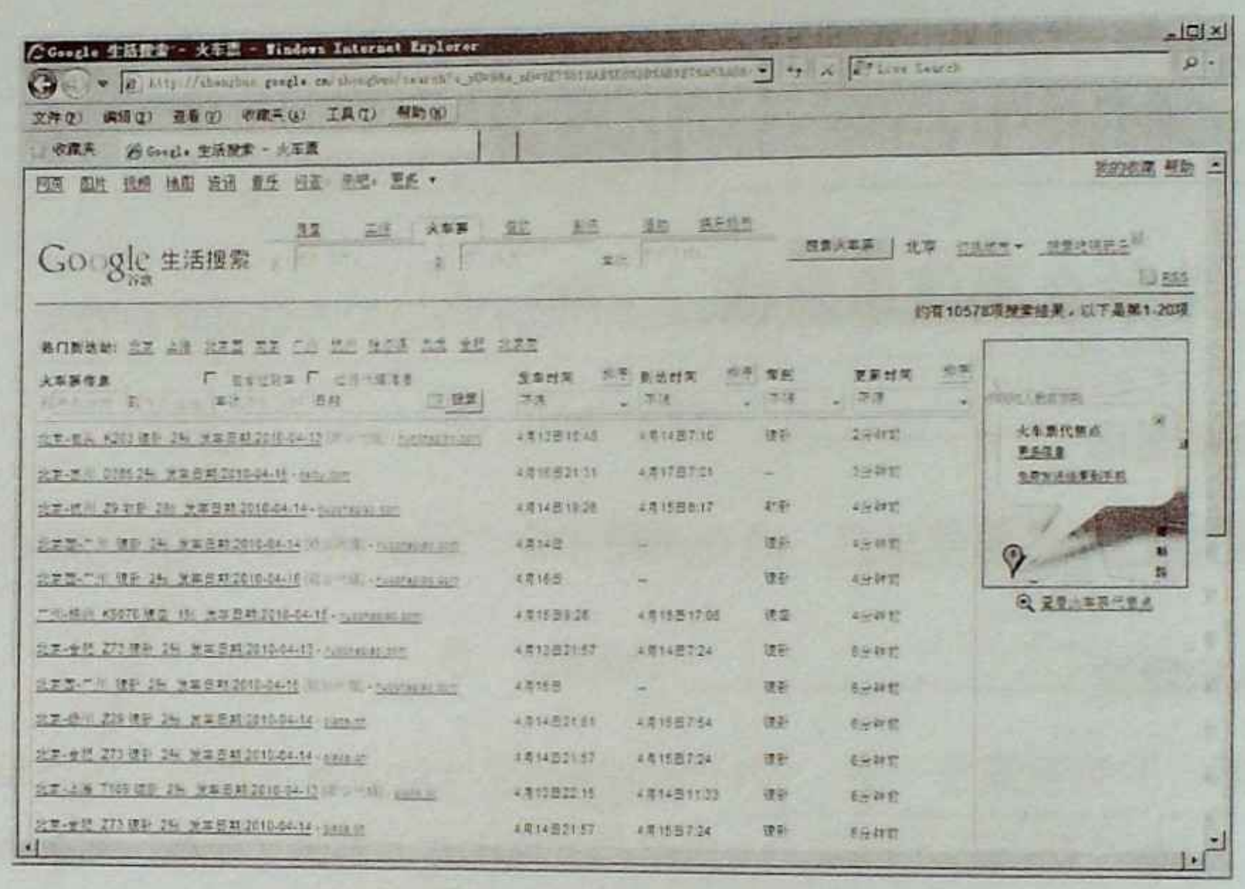


图 3.3　咨讯网站页面

3.1.2　表格的基本结构

先看一看表格的基本结构，表格是由指定数目的行和列组成的，如图 3.4 所示，其中的文字或图片按照相应的列或行进行分类和显示。

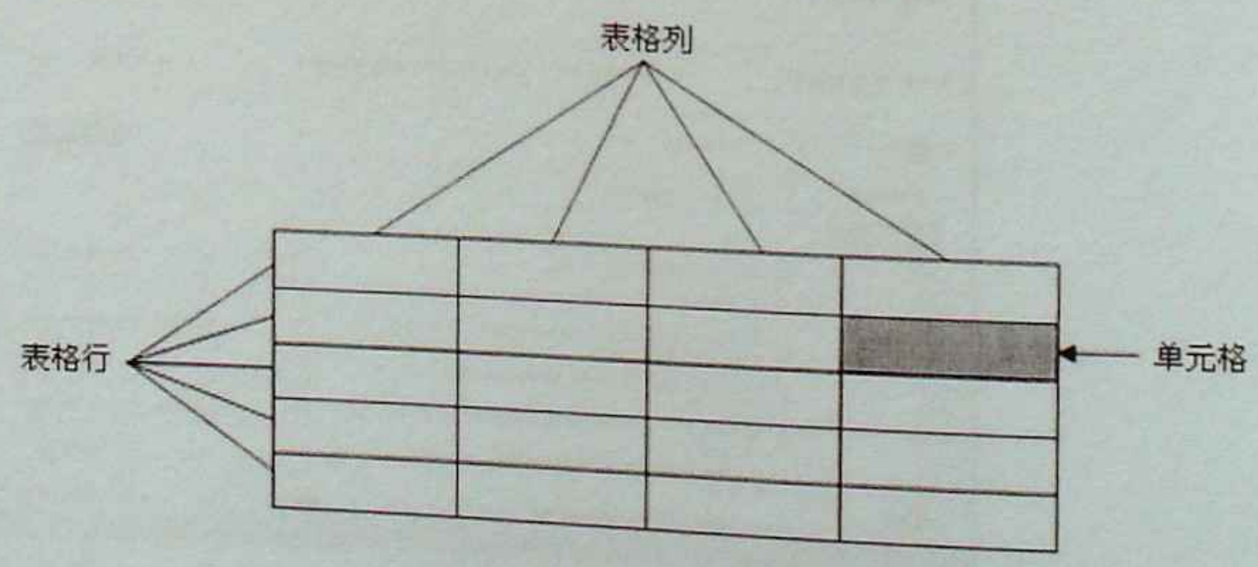


图 3.4　表格的基本结构

1. 单元格

表格的最小单位，一个或多个单元格纵横排列组成了表格。

2. 行

一个或多个单元格横向堆叠形成了行。

3. 列

由于表格单元格的宽度必须一致，所以单元格纵向划分形成了列。

3.1.3 表格的基本语法

创建表格的基本语法如下。

```
<table>
    <tr>
        <td>第 1 个单元格的内容</td>
        <td>第 2 个单元格的内容</td>
        ……
    </tr>
    <tr>
        <td>第 1 个单元格的内容</td>
        <td>第 2 个单元格的内容</td>
        ……
    </tr>
</table>
```

第一行

第二行

创建表格一般分为下面三步。

第一步：创建表格标签<table>……</table>。

第二步：在表格标签<table>……</table>里创建行标签<tr>……</tr>，可以有多行。

第三步：在行标签<tr>……</tr>里创建单元格标签<td>……</td>，可以有多个单元格。

为了显示表格的轮廓，一般还需要设置<table>标签的“border”边框属性，指定边框的宽度。

例如，在页面中添加一个 2 行 3 列的表格，对应的 HTML 代码如示例 1 所示。

示例 1

```
<html>
  <head>
      <title>基本表格</title>
  </head>
  <body>
      <table border="2">
```

<table>代表表格的开始，border="2"表示边框宽度为 2

```
    <tr>
      <td>1 行 1 列的单元格</td>
      <td>1 行 2 列的单元格</td>
      <td>1 行 3 列的单元格</td>
    </tr>
    <tr>
      <td>2 行 1 列的单元格</td>
      <td>2 行 2 列的单元格</td>
      <td>2 行 3 列的单元格</td>
    </tr>
  </table>
 </body>
</html>
```

表格的第一行，有 3 条数据，<tr>……</tr>代表行

表格的第二行，有 3 条数据

运行示例 1 的代码，其效果如图 3.5 所示。

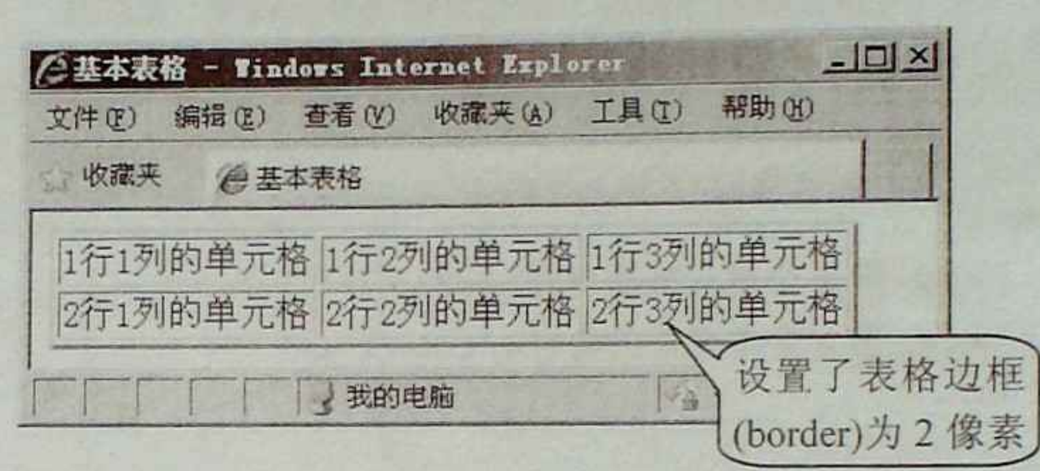


图 3.5 创建基本表格

3.2 跨行跨列的表格

上面我们介绍了简单表格的创建，而现实中往往需要较复杂的表格，有时就需要把多个单元格合并为一个单元格，也就是要用到表格的跨行跨列功能。

3.2.1 跨列

跨列是指单元格的横向合并，语法如下。

```
<table>
  <tr>
     <td colspan="所跨的列数">单元格内容</td>
  </tr>
</table>
```

col 为 column（列）的缩写，span 为跨度，所以 colspan 意思为跨列。

下面通过示例 2 来说明 colspan 属性的用法，对应的页面效果如图 3.6 所示。

示例 2

```
<html>
  <head>
    <title>跨多列的表格</title>
  </head>
  <body>
    <table width="200" border="1">
      <tr>
        <td colspan="2">学生成绩</td>
      </tr>
      <tr>
        <td>语文</td>
        <td>98</td>
      </tr>
      <tr>
        <td>数学</td>
        <td>95</td>
      </tr>
    </table>
  </body>
</html>
```

设置单元格水平跨 2 列，2 是单元格所跨的列的个数

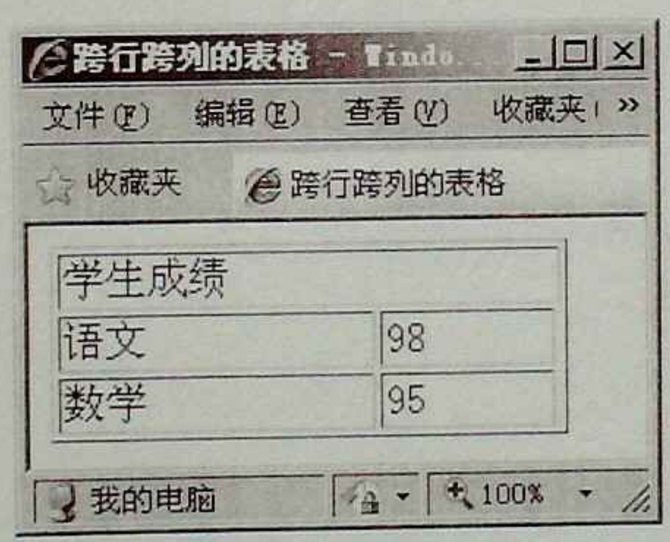


图 3.6 跨列的表格

3.2.2 跨行

跨行是指单元格在垂直方向上合并，语法如下。

```
<table>
  <tr>
      <td rowspan="所跨的行数">单元格内容</td >
  </tr>
</table>
```

row 为行的意思，rowspan 即跨行。

下面通过示例 3 来说明 rowspan 属性的用法，页面对应的效果如图 3.7 所示。

示例 3

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>跨行表格</title>
  </head>
  <body>
    <table width="500" border="1">
      <tr>
        <td rowspan="2">张三</td>
        <td>语文</td>
        <td>98</td>
      </tr>
      <tr>
        <td>数学</td>
        <td>95</td>
      </tr>
      <tr>
        <td rowspan="2">李四</td>
        <td>语文</td>
        <td>88</td>
      </tr>
      <tr>
        <td>数学</td>
        <td>91</td>
      </tr>
    </table>
  </body>
</html>
```

设置单元格垂直跨 2 行，2 是单元格所跨的行的个数

图 3.7 跨行的表格

经验总结

一般而言，跨行或跨列操作时，需要以下两个步骤。

(1) 在需合并的第一个单元格，设置跨列或跨行属性，如 colspan="3"。

(2) 删除被合并的其他单元格，即把某个单元格看成多个单元格合并后的单元格。

3.2.3 跨行、跨列

有时表格中既有跨行又有跨列的情况，如图 3.8 所示。修改示例 3 和示例 2，完整的 HTML 代码如示例 4 所示。

示例 4

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>跨行跨列的表格</title>
  </head>
  <body>
    <table width="200" border="1">
      <tr>
        <td colspan="3">学生成绩</td>
      </tr>                    跨 3 列
      <tr>
        <td rowspan="2">张三</td>
        <td>语文</td>          跨 2 行
        <td>98</td>
      </tr>
      <tr>
        <td>数学</td>
        <td>95</td>
      </tr>
      <tr>
        <td rowspan="2">李四</td>
        <td>语文</td>          跨 2 行
        <td>88</td>
      </tr>
      <tr>
        <td>数学</td>
        <td>91</td>
      </tr>
    </table>
  </body>
</html>
```

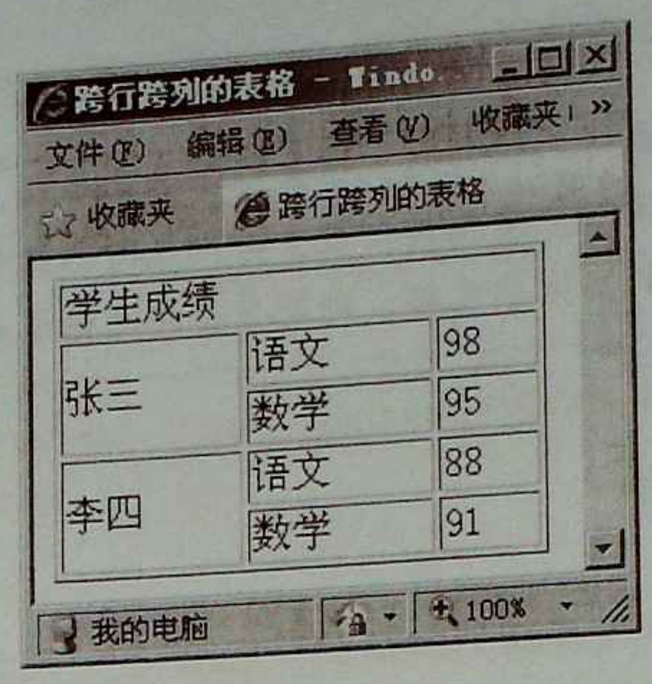


图 3.8 跨行、列的综合应用

经验总结

跨行和跨列以后，并不改变表格的特点，同行的总高度一致，同列的总宽度一致。因此，表格中各单元格的宽度或高度互相影响，结构相对稳定，但缺点是不能灵活地进行布局控制。

3.3 表格的高级用法

除了设置表格跨行和跨列外，还可以为整个表格添加标题（Caption）、对表格数据分组等，从而实现企业中常见的年度统计报表等复杂表格。如图 3.11 所示为一张简化了的企业收入报表表格。

3.3.1 表格的高级标签

如何实现上述报表效果？HTML 提供了如下表格标签，对应的含义如图 3.9 所示。

1. 表格标题<caption>

用于描述整个表格的标题。

2. 表格表头<th>

用于定义表格的表头，一般是表格的第一行数据，以粗体、居中的方式显示。

3. 表格数据的分组标签<thead>、<tbody>、<tfooter>

这三个标签一般配合使用，主要是对报表数据进行逻辑分组，<thead>标签对应报表的页眉，即表格的表头部分；<tbody>对应报表的数据主体，即详细的数据描述部分；<tfooter>对应报表的页脚，即对各分组数据进行汇总的部分。各分组标签内由多行<tr>组成。

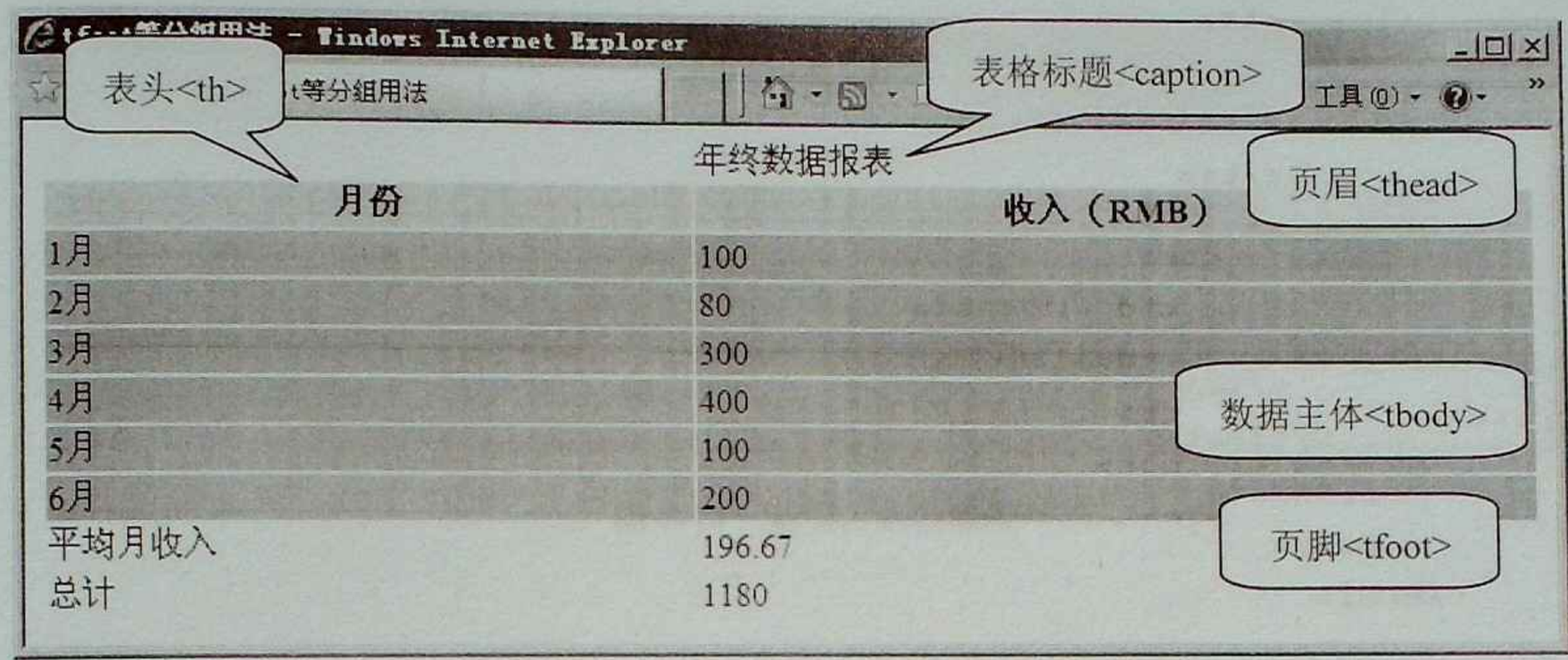


图 3.9　跨行跨列的商品分类表格

为了区分各部分的数据，可以利用“style”样式属性分别为<thead>、<tbody>、<tfooter>设置背景颜色。同时，为了使整个表格的宽度充满浏览器窗口的整行，可以利用“width”属性设置表格宽度为“100%”。

图 3.9 对应的 HTML 代码如示例 5 所示，需要说明的是，宽度、背景色等样式一般应放在专门的 CSS 文件中（后续章节讲解），此处仅是为了演示页面效果。

示例 5

```
<html >
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>tfoot 等分组用法</title>
  </head>
  <body>
    <table width="100%">
      <caption>年终数据报表</caption>
      <thead style="background: #0FF">
        <tr>
          <th>月份</th>
          <th>收入(RMB)</th>
        </tr>
      </thead>
      <tbody style=" background: #9CC">
        <tr>
          <td>1 月</td>
          <td>100</td>
        </tr>
        <tr>
          <td>2 月</td>
          <td>80</td>
        </tr>
        <!—省略 3-6 月的类似 HTML 代码-->
      </tbody>
      <tfoot style="background: #FF0"ss>
```

```
    <tr>
      <td>平均月收入</td>
      <td>196.67</td>
    </tr>
    <tr>
      <td>总计</td>
      <td>1180</td>
    </tr>
  </tfoot>
  </table>
 </body>
</html>
```

3.3.2 上机练习

上机练习 1

练习——跨行跨列表格

需求说明

实现一个跨行跨列的商品类目表格，效果如图 3.10 所示。

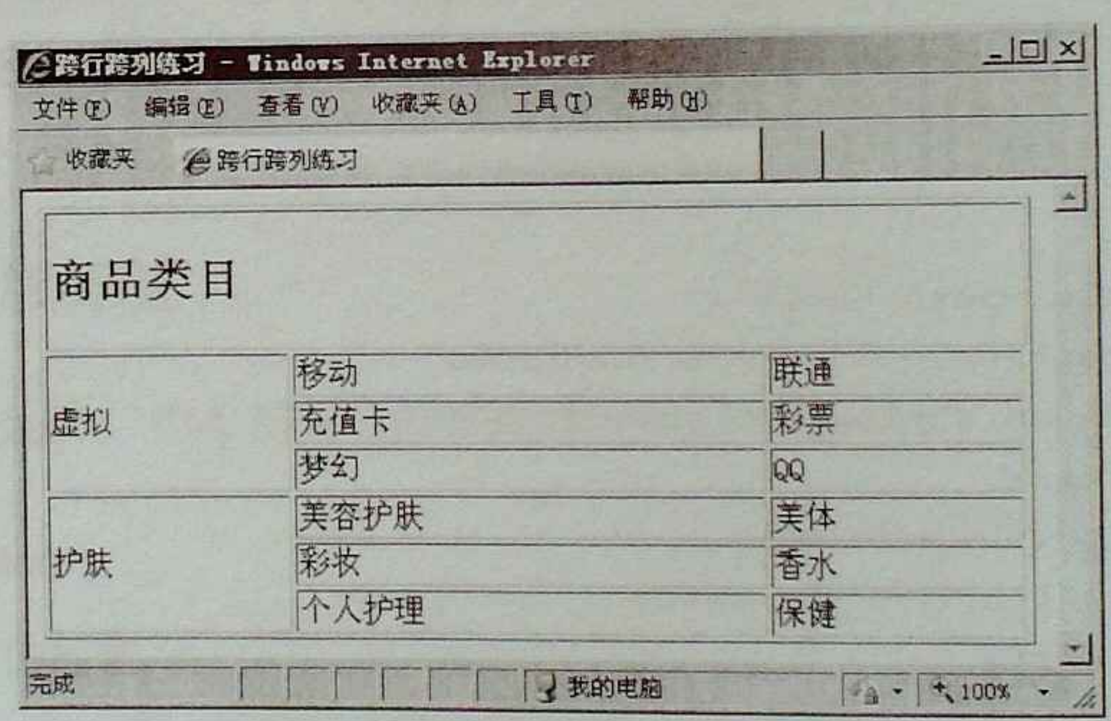


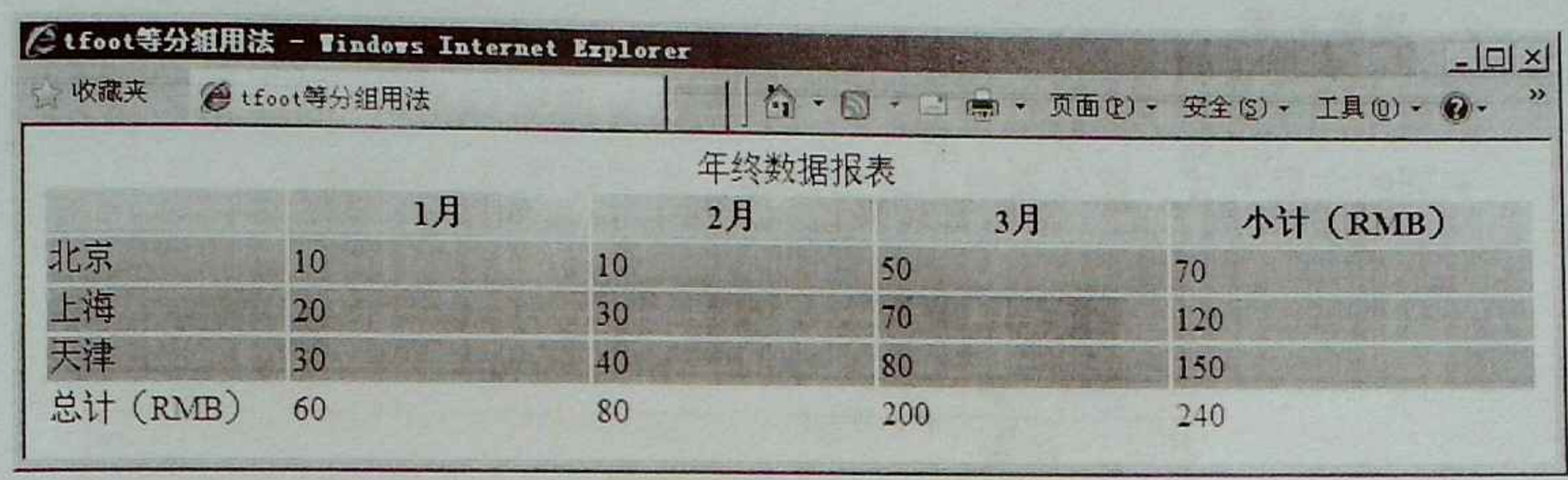
图 3.10 跨行跨列的商品分类表格

上机练习 2

练习——表格分组标签的使用

需求说明

实现一个按地点和时间汇总的年终报表，效果如图 3.11 所示。



年终数据报表

	1月	2月	3月	小计（RMB）
北京	10	10	50	70
上海	20	30	70	120
天津	30	40	80	150
总计（RMB）	60	80	200	240

图 3.11　数据分组的表格

3.4　表格布局

随着表格应用的深入，表格除用来显示数据外，还用于搭建网页的结构，也就是通常所说的网页布局。下面介绍如何使用表格实现页面布局。

3.4.1　应用场景

表格布局最典型的应用有两种，图文布局及表单布局。

1．图文布局

表格的图文布局是将图像和文本都看成单元格的组成内容，然后设置它们所占的行数或列数，这种布局方式的效果如图 3.12 所示。

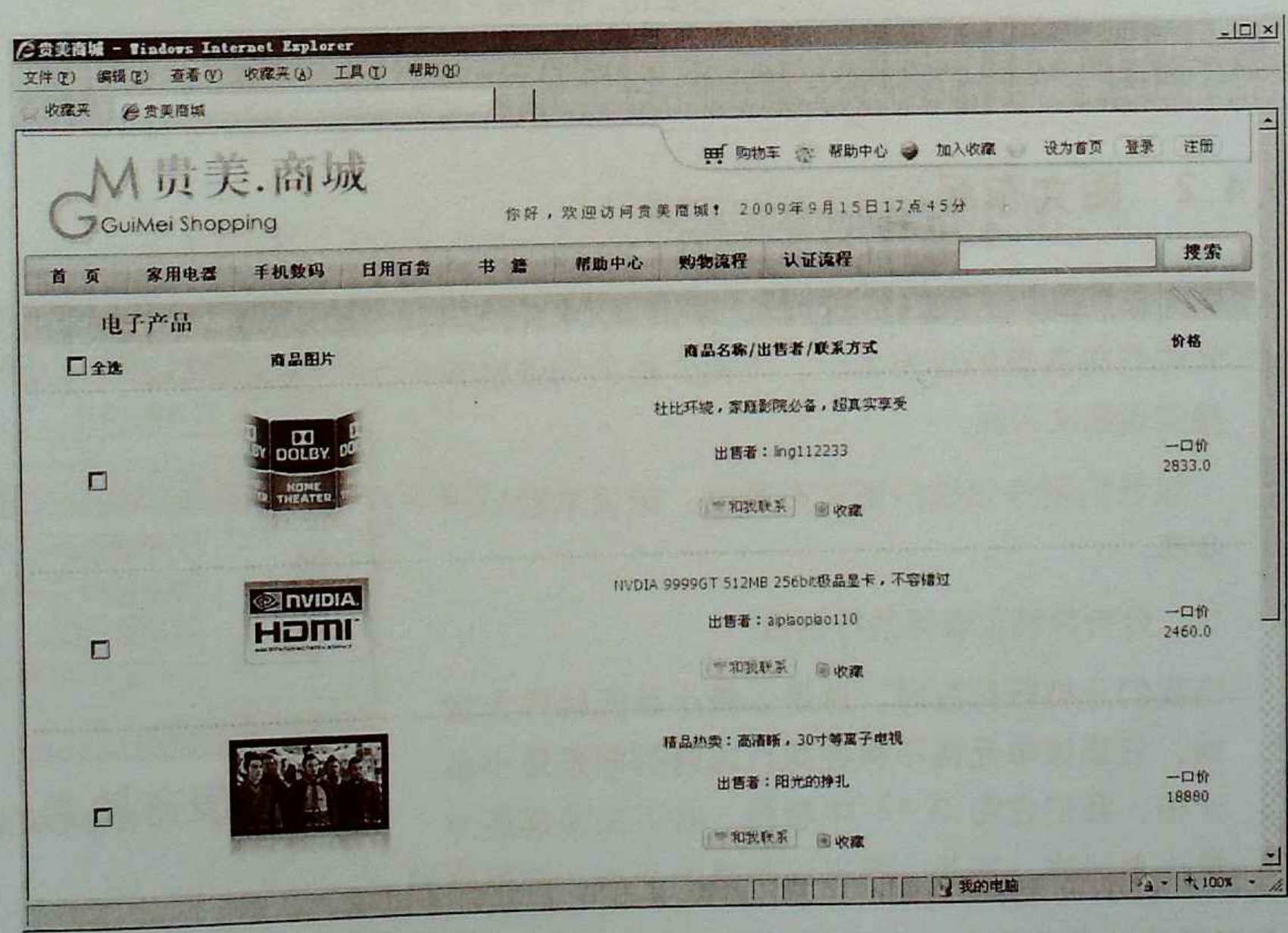


图 3.12　表格用于图文布局

2. 表单布局

表单布局是把注册的各项看成一行，每项的标题显示在同一列，而所填信息也显示在同一列的布局方式。整体看起来较为规整，效果如图 3.13 所示。

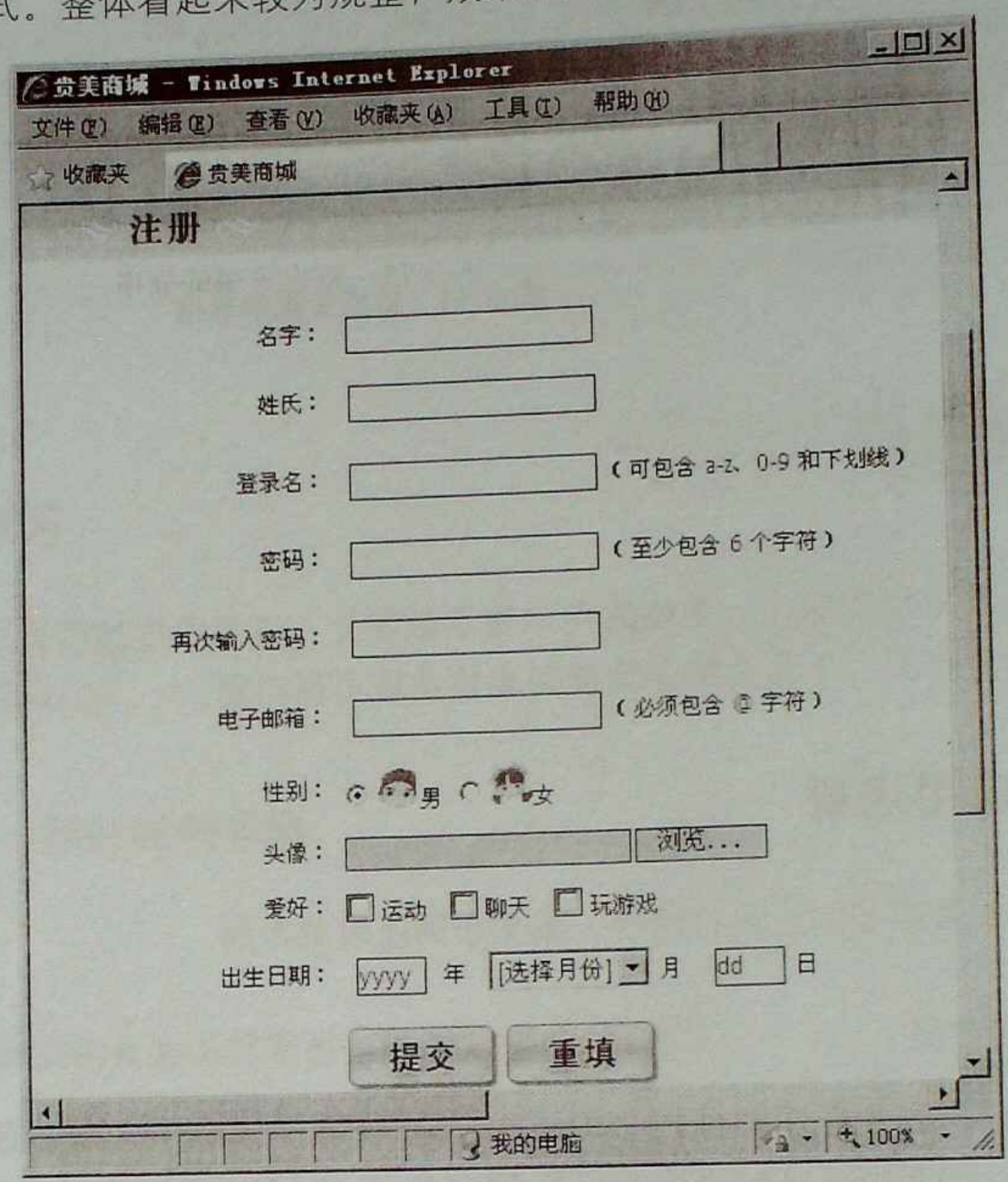


图 3.13　表格用于表单布局

下面逐一介绍它们的具体实现过程。

3.4.2　图文布局

图文布局是最常用的局部布局，如图 3.14 所示的公告栏是典型的应用之一。下面介绍如何使用表格实现图文布局。

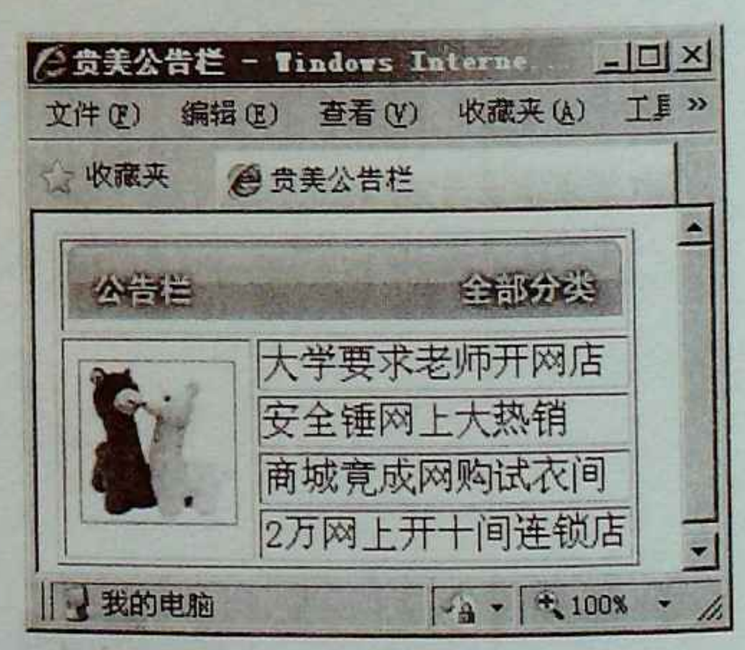


图 3.14　公告栏的图文布局

一般我们刚开始进行图文布局时，建议采取以下步骤。

（1）分析并确定表格的行列数。

当我们分析行列数时，总是以最小单元格作为依据。只要该单元格不存在跨行或跨列即是最小单元格。我们在图 3.14 中看到，右下文字描述为最小单元格。据此，得出该表格为 5 行 2 列的表格。

(2) 写出一个 5 行 2 列的表格。

为显示效果，设置 border="2"，并在<td></td>里加个空格（" "）。

(3) 确定合并单元格位于几行几列并跨了几行几列。

公告栏标题图片位于一行一列跨了两列；左侧图片位于两行一列跨了四行。

(4) 增加 colspan 及 rowspan 属性。

设置跨行列属性后要删除多余单元格达到合并效果。

公告栏标题图片跨两列，即横向合并两个单元格，在一行一列单元格<td>里加入跨列属性 colspan="2"，然后再删除右边的一个单元格。左侧图片跨四行，则在两行一列单元格里加入跨行属性 rowspan="4"，再删除下方的三个单元格。

布局完以后，我们再来考虑诸如边框 border 及总宽度 width 的修饰设置。

完整的 HTML 代码如示例 6 所示。

示例 6

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>贵美公告栏</title>
  </head>
  <body>
    <table border="1px">
      <tr>
        <td colspan="2"><img src="images/a_title.jpg" alt="公告栏" /></td>
      </tr>
      <tr>
        <td rowspan="4"><img src="images/a_left.jpg" alt="公告左侧图" /></td>
        <td>大学要求老师开网店</td>
      </tr>
      <tr>
        <td>安全锤网上大热销</td>
      </tr>
      <tr>
        <td>商城竟成网购试衣间</td>
      </tr>
      <tr>
        <td>2 万网上开十间连锁店</td>
      </tr>
    </table>
  </body>
</html>
```

3.4.3 表单布局

上一章我们学习了表单主要用于收集用户信息，实现与服务器交互目的。由于表单元素和相应的提示标题一一对应，因此我们可以把标题和表单输入元素各归入相邻的两列

中，再根据信息数决定行数，以实现使用表格对表单的基本布局。

我们先考虑简单的表单布局，如图 3.15 所示。和图文布局类似，使用表格布局表单只需关注以下三点。

- 需要多少列。结合表单的数据信息，只需标题及输入框两列。
- 各列的宽度是多少。标题“会员名”、“密码”的宽度够容纳四五个汉字即可。
- 特殊元素的跨行跨列数。“登录”按钮需要跨两列，登录页面的标题图片也跨两列。

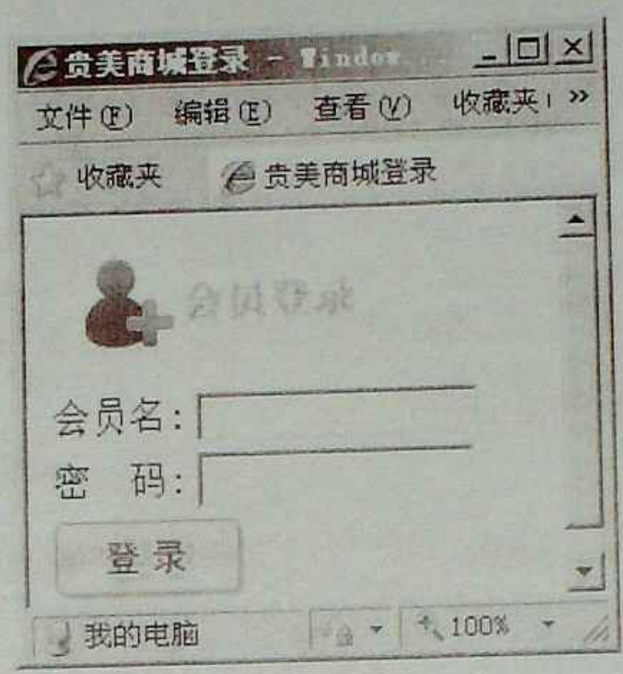


图 3.15 登录的简单结构设计

这个结构虽然简单，却很实用，大家上网看到的很多登录框基本与此类似。但这个结构所占空间较小，如果放在一个实际的登录页中就会显得不协调。例如，放到贵美商城的登录页中，页面效果如图 3.16 所示，显然，因登录框所占空间过小，导致页面中间出现了大片“空白”。

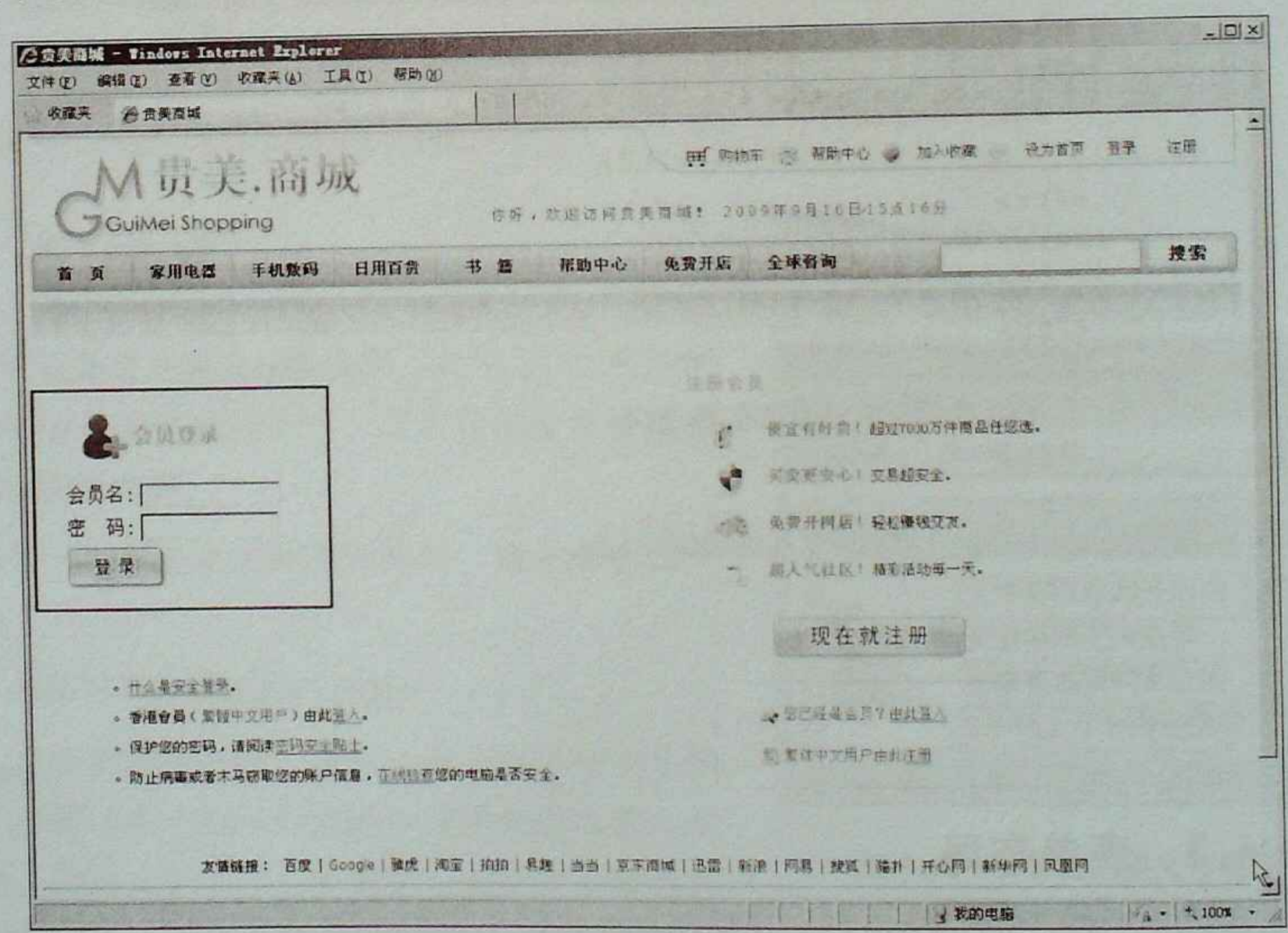


图 3.16 登录页中的简单结构

解决的思路还是一样。

- 扩展为几列。针对登录框所占空间小的问题，我们调整为三列，“会员登录”标题图片下的两个单元格内容为空，即整个登录框采用四行三列的表格进行布局，如图 3.17 所示。
- 各列宽度。“输入框”所在列的宽度增加，增加输入框的宽度，同时添加输入提示文字。
- 特殊元素的跨行或跨列，仍是“标题图片”和“登录按钮”存在跨多列的情况。

根据思路分析，对应的 HTML 代码如示例 7 所示。

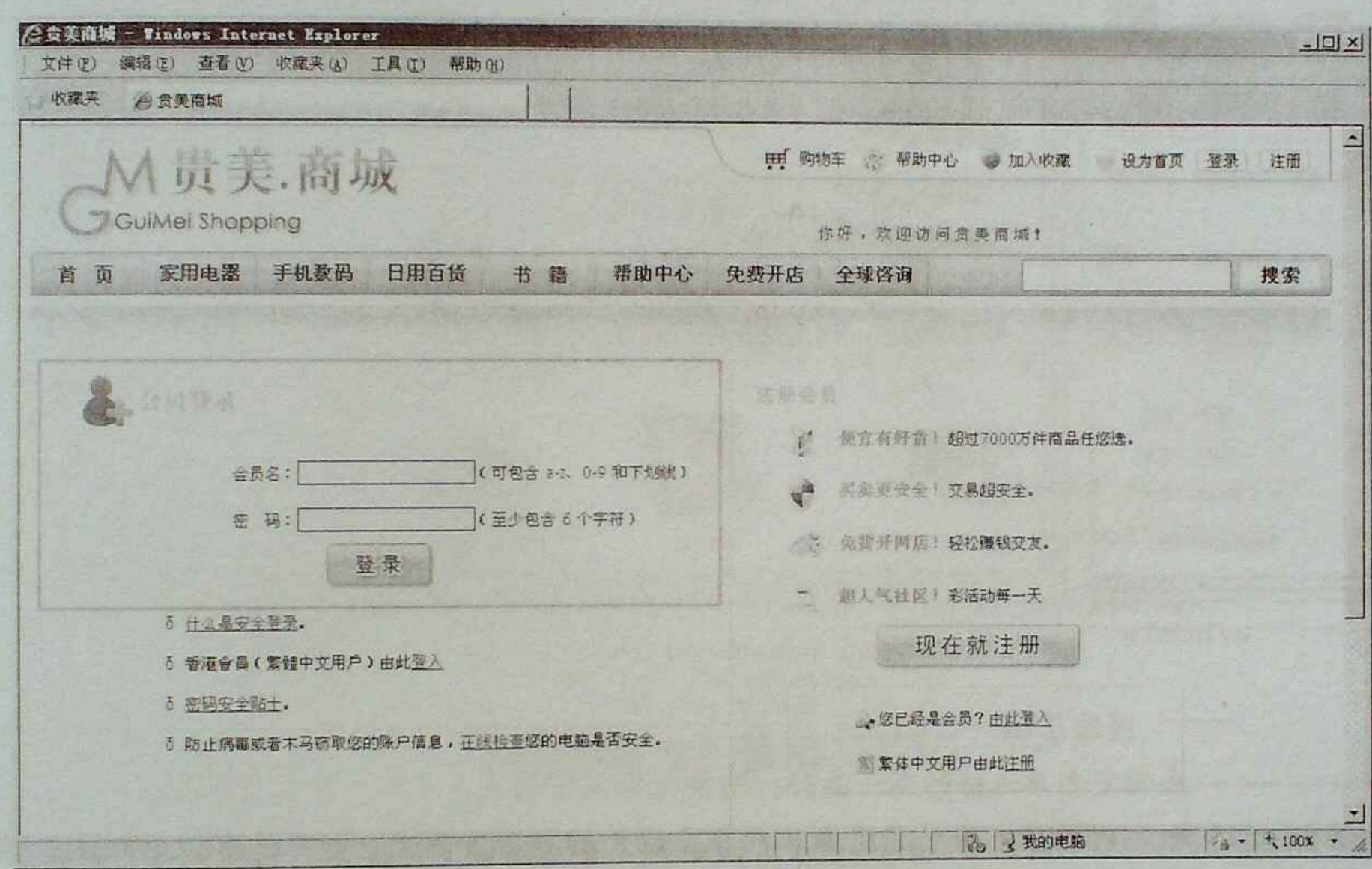


图 3.17　块结构布局实现登录页

示例 7

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>贵美商城登录</title>
  </head>
  <body>
    <form method="post" action="login_success.htm">
        <table>
          <tr>
            <td><img src="images/title_login_2.png" alt="alt" /></td>
            <td colspan="2"> </td>
          </tr>
```

```
<tr>
  <td></td>
  <td>会员名:</td>
  <td>
     <input name="sname" type="text" size="15" />(可包含 a-z、
     0-9 和下划线)</td>
</tr>
<tr>
  <td></td>
  <td>密   码:</td>
  <td><input name="pass" type="password" size="15" />(至少包含
  6 个字符)</td>
</tr>
<tr>
  <td></td>
  <td colspan="2">
     <input  type="image"  style="border:0px;"  name="Button"
     src="images/login.gif" />
    </td>
</tr>
  </table>
 </form>
</body>
</html>
```

经验总结

在确定各单元格的宽、高时，需要注意以下两点。

- 同列单元格的宽度由该列宽度最大的单元格决定。如果没有，则默认与该列第一行单元格宽度一致。在布局时必须注意内容过长撑开单元格的情况，合理设置好各列列宽。
- 同行单元格的高度类似，但是请注意换行带来的高度变化。

3.4.4 表格的嵌套布局

既然表格能用于页面布局，那么我们尝试使用表格来实现如图 3.18 所示的贵美商城的首页布局。显然，整个页面可以划分为上、中、下三行一列的表格，其中，中间部分比较复杂，可以再分为三列，各列很明显又嵌套多个表格，这些表格的完整结构都类似图 3.18 左上角的“家用电器”版块。

通过上述页面结构的划分，可以看到如使用表格实现页面的整体布局，需要使用表格的多层嵌套，结构非常复杂。下面再看看表格嵌套对应的 HTML 代码。例如，实现一个简单的表格嵌套，如图 3.19 所示。

图 3.18　块结构布局实现登录页

图 3.19　表格遇到了参差结构

从整体看是一张一行两列的大表格，左右两列又各包含三张一行六列的表格。对应的 HTML 代码如示例 8 所示（这些代码不要求掌握，仅用于展示嵌套表格的代码复杂性）。

示例 8

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <title>贵美商城登录</title>
  </head>
  <body>
    <table border="10">
      <tr>
       <td>
         <table border="2">
         <tr>
           <td width="50px"> </td>
           <td width="50px"> </td>
           <td width="50px"> </td>
           <td width="50px"> </td>
           <td width="50px"> 
           </td><td width="25px"> </td>
         </tr>
         </table>
         <table border="2">
         <tr>
           <td width="25px"> </td>
           <td width="50px"> </td>
           <td width="50px"> </td>
           <td width="50px"> </td>
           <td width="50px"> </td>
         </tr>
         </table>
         <table border="2">
         <tr>
           <td width="50px"> </td>
           <td width="50px"> </td>
           <td width="50px"> </td>
           <td width="50px"> </td>
           <td width="50px"> </td>
           <td width="25px"> </td>
         </tr>
         </table>
         </td>
         <td>
         <table border="2">
         <tr>
           <td width="50px"> </td>
           <td width="50px"> </td>
           <td width="50px"> </td>
```

```
    <td width="50px"> </td>
    <td width="50px"> </td>
    <td width="25px"> </td>
  </tr>
  </table>
  <table border="2">
  <tr>
    <td width="25px"> </td>
    <td width="50px"> </td>
    <td width="50px"> </td>
    <td width="50px"> </td>
    <td width="50px"> </td>
    <td width="50px"> </td>
    <td width="50px"> </td>
  </tr>
  </table>
  <table border="2">
  <tr>
    <td width="50px"> </td>
    <td width="50px"> </td>
    <td width="50px"> </td>
    <td width="50px"> </td>
    <td width="50px"> </td>
    <td width="25px"> </td>
  </tr>
  </table>
</td>
</tr>
</table>
</body>
</html>
```

可以看出，如使用嵌套表格布局页面，HTML 层次结构复杂，代码量非常大，并且 HTML 结构的语义化差，但表格布局又具有结构相对稳定、简单通用的优点，所以表格布局仅适用于页面中数据规整的局部布局，而页面的整体布局一般采用主流的 DIV+CSS 布局，DIV+CSS 布局将在后续章节进行详细讲解。

3.4.5 上机练习

上机练习 3

练习——贵美商品分类页

需求说明

实现贵美商品分类页，效果如图 3.20 所示。

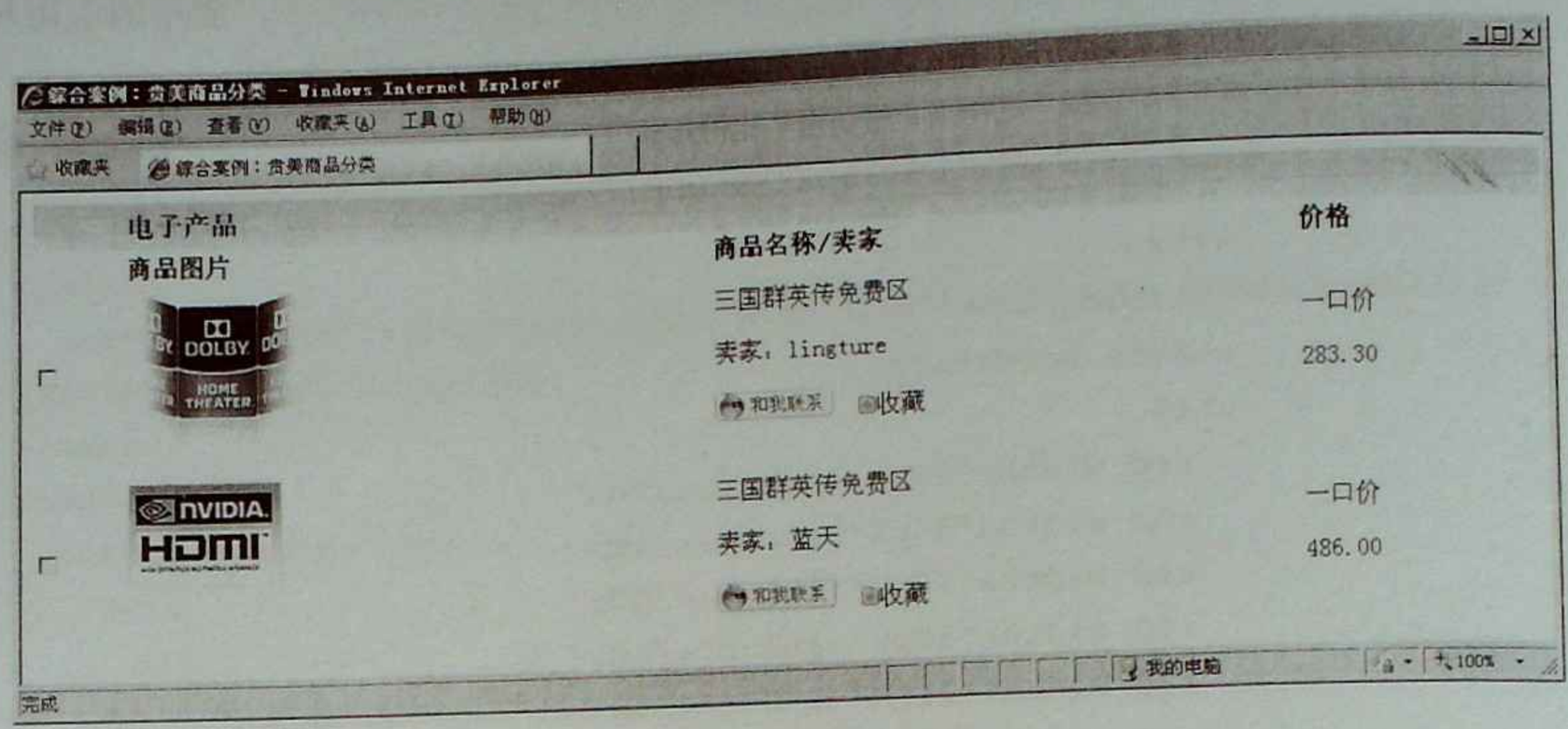


图 3.20　贵美商品分类页

上机练习 4

练习——贵美注册页

需求说明

实现贵美商城注册页，效果如图 3.21 所示，具体要求如下。

- 使用表格布局。
- 包含文本框、密码框、单选框、复选框及列表框。
- 包含隐藏域、文件域用法。
- 包含“提交”及“重置”按钮，“提交”按钮失效。
- 包含多行文本框，内容设置为只读。

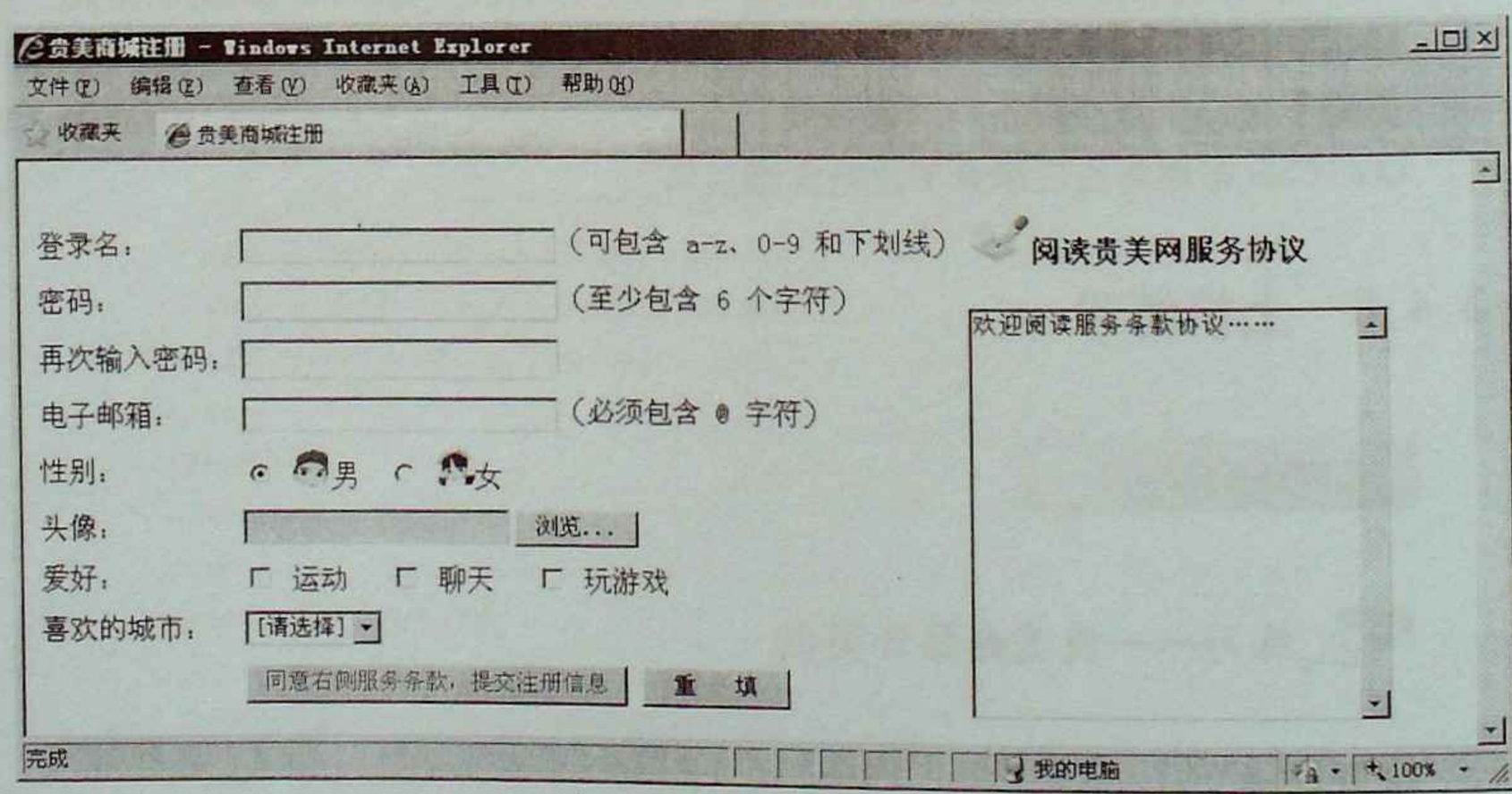


图 3.21　贵美商城注册页

本章总结

- 掌握表格的基本使用方法。
  - 使用<table>、<tr>、<td>创建表格。
  - 使用“border”属性设置边框，“width”属性设置表格宽度。
- 制作跨行跨列的表格。
  - 跨列：colspan="横向跨的单元格数"。
  - 跨行：rowspan="纵向跨的单元格数"。
- 表格的高级用法包括：使用<caption>标签添加表格标题，使用数据分组标签<thead>、<tbody>、<tfoot>实现表格数据的逻辑分组。
- 表格还可以用于图文和表单的布局。
- 表格布局适用于数据规整的局部布局，页面的整体布局推荐使用流行的 DIV+CSS 布局。

本章作业

一、选择题

1. 表格的基本语法结构是（　　）。

A. <table><td><tr></tr></td></table>

B. <table><td></tr><tr></td></table>

C. <tr><table><td></td></table></tr>

D. <table><tr><td></td></tr></table>

2. 表格各行宽度或高度的特点是（　　）。

A. 各列宽度一致，各行高度也一致

B. 各行宽度一致，各列单元格高度也一致

C. 同列单元格宽度一致，且垂直对齐

D. 同行单元格高度一致，且水平对齐

3. 下面说法正确的是（　　）。

A. 表格布局适用于结构化数据显示和表单

B. 表格布局也可以实现参差结构，但必须用到嵌套

C. 同列单元格宽度由该列宽度最大的单元格决定

D. 同行单元格高度由该行第 1 列的单元格决定

4. 关于表格的数据分组标签，说法正确的是（　　）。

A. <thead>、<tbody>、<tfoot>分别表示页眉、数据主体、页脚

B. <thead>必须包含<th>表头标签

C. <thead>和<tfoot>标签只能包含一行数据

D. <thead>、<tbody>、<tfoot>由行组成，所以一般包含<tr>标签

5. 关于表格布局的描述中，正确的是（　　）。

A. 先确定行列数并写出表格，然后再实现跨行跨列的单元格合并

B. 表格布局用于数据的规则显示。例如，成绩显示、表单布局等

C. 用表格能实现所有布局，因此不需要学习其他块状结构布局

D. 从技术实现角度，表格布局不支持表格的多层嵌套

二、简答题

1. 表格各行、各列的宽高有什么特点？这决定了表格应适用于哪些场景？
2. 表格的跨行跨列分别是什么属性？要实现一个跨三行两列的单元格要哪几个步骤？
3. 表格用于表单布局时，各列的列宽应如何决定？在使用中应该注意什么问题？
4. 编写 HTML 代码，实现如图 3.22 所示的页面效果，背景色自行设置。

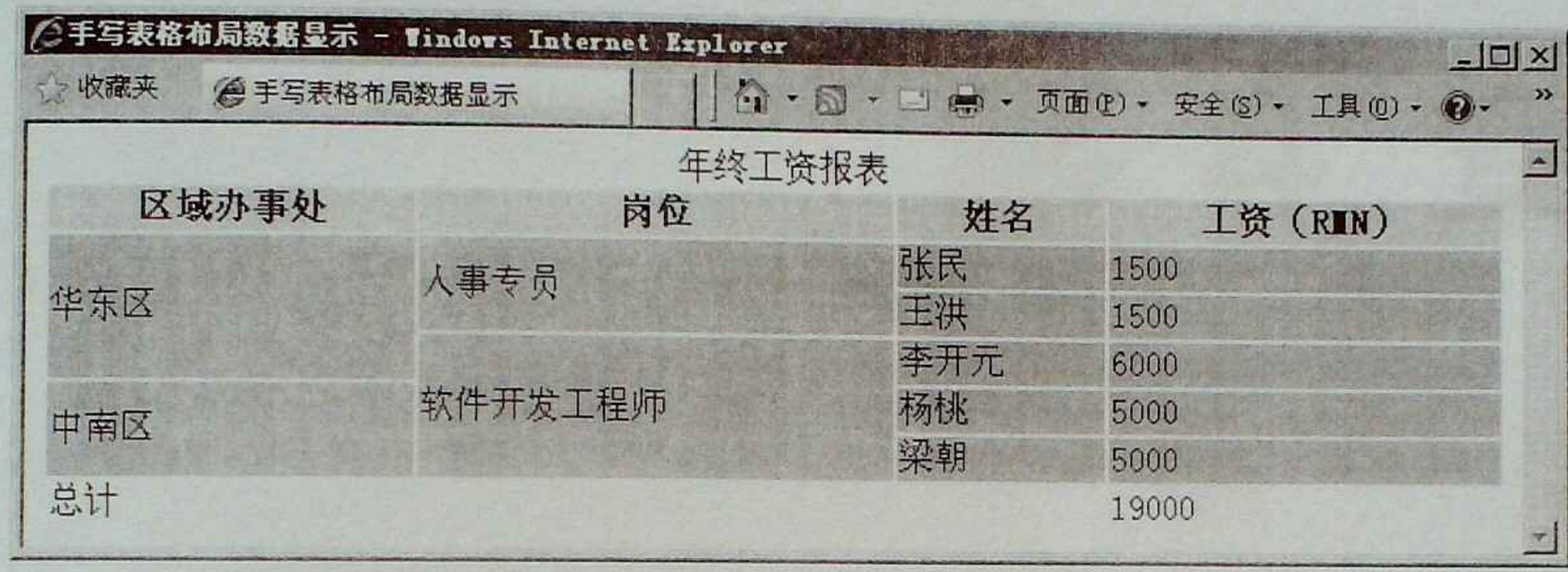


图 3.22　贵美邮箱登录

5. 编写 HTML 代码，利用表格实现表单布局，如图 3.23 所示。

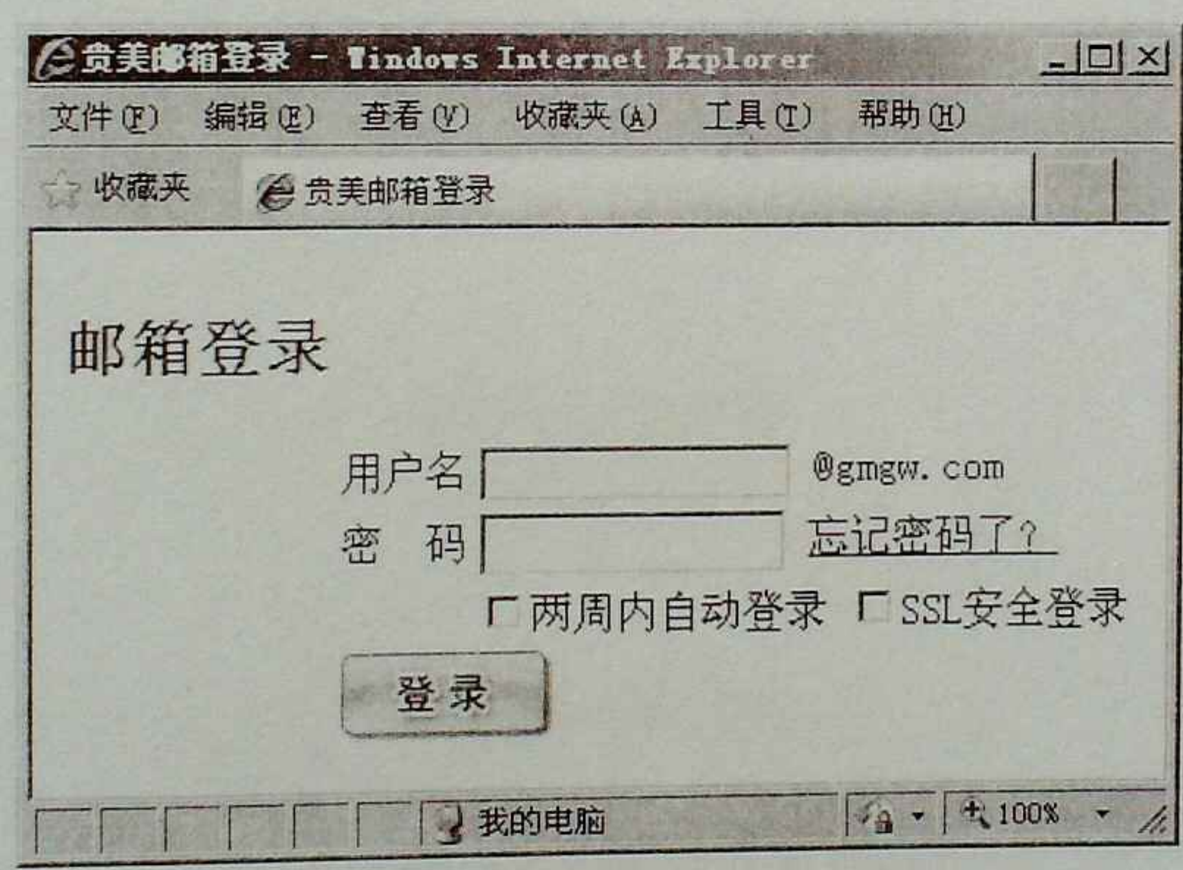


图 3.23　贵美邮箱登录

第4章

框　架

◈本章工作任务

- 制作“帮助中心”页面

◈本章技能目标

- 使用框架结构实现多窗口页面
- 使用<iframe>内嵌重用页面

本章单词

请在预习时学会下列单词的含义和发音，并填写在横线处。

1. frame：_____
2. register：_____
3. blank：_____
4. parent：_____
5. resize：_____
6. scrolling：_____
7. target：_____

本章简介

上几章，我们学习了如何使用标签组织页面内容，有时候我们需要在一个浏览器窗口中显示多个页面内容，这时候就需要使用框架技术。本章重点讲解框架的两种实现技术，内容包括基本语法及优劣势比较等。

4.1 为什么使用框架

框架是 HTML 早期的应用技术，但目前还有部分网站在使用。如图 4.1 所示，粗线标识的每部分就代表一个“框架”，每个框架对应一个页面。使用框架技术具有以下好处。

- 在同一个浏览器窗口显示多个页面。使用框架能有机地把多个页面组合在一起，但各个页面间相互独立。
- 可以实现页面复用。例如，为了保证统一的网站风格，网站每个页面的底部和顶部一般都相同。因此，可以利用框架技术，将网站的顶部或底部单独作为一个页面，方便其他页面复用。
- 实现典型的“目录结构”，即左侧目录、右侧内容，当用户单击左侧窗口的目录时，在右侧窗口中显示具体内容，如网上在线学习教程、论坛、后台管理、产品介绍等网页都是这样的页面结构。当然，这种结构除能用框架技术实现外，也可采用其他技术实现。

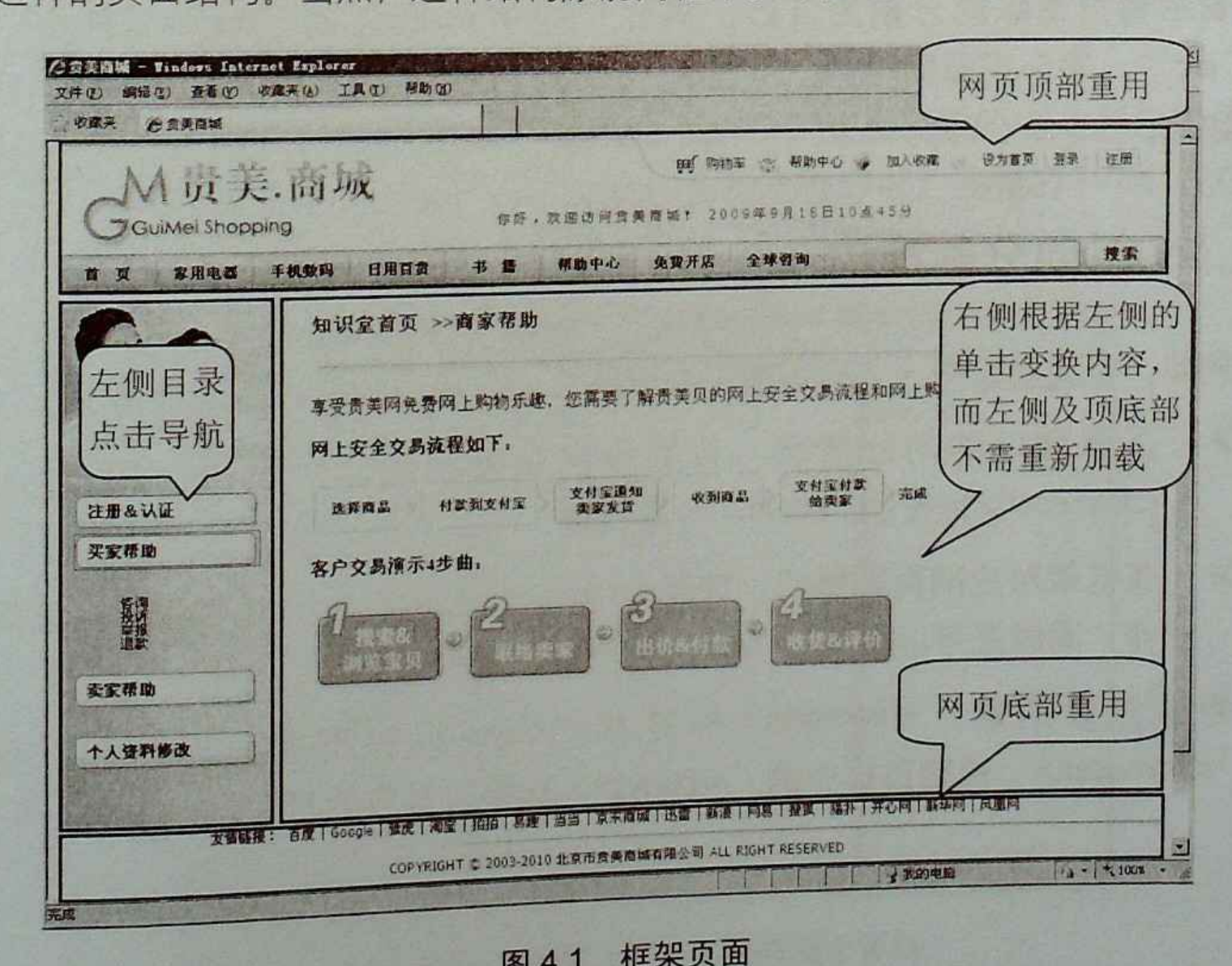


图 4.1　框架页面

常用的框架技术有以下两种。

- 框架集（<frameset>）：这是早期的框架技术，页面各窗口全部用框架（<iframe>）实现，形成一个框架集。这种结构非常清晰，适用于整个页面都用框架实现的场合。
- 内嵌框架（<iframe >）：页面中的部分内容用框架实现，一般用于在页面中引用站外的页面内容，使用比较方便、灵活。

考虑到框架集的结构清晰，我们先用它来讲解框架集的基本结构。

4.2 <frameset>框架集

框架集包含<frameset>和<frame>两个标签，其中<frameset>描述窗口的分割，<frame>定义放置在每个框架中的 HTML 页面。基本语法如下。

分割成左中右三部分　　分割成上下两部分　　框架边框

```
<frameset cols="25%,50%,*"  rows ="50%,*" border="5">
  <frame src="the_first.html ">
<frame src="the_second.html ">
  ……
</frameset>
```

其中，<frameset >标签的 cols 属性表示将页面横向分割为几列，例如，cols="25%,50%,*"表示将页面分割为三列，第一列占浏览器窗口总宽度的 25%，第二列占 50%，第三列占剩余部分，各列的宽度值也可使用具体数值（单位像素为 px）。同理，"rows"属性表示将页面纵向分割为几行。另外，<frame>标签的"src"属性类似<img>标签的"src"，表示页面的路径。下面逐一讲解框架的纵向、横向以及横纵向同时分割的实现方法。

4.2.1 纵向分割窗口

例如，要实现如图 4.2 所示的纵向分割页面效果，则有如下两种情况。

- 页面窗口的划分情况：页面纵向被分割为三个窗口，即三行，显然应使用<frameset>标签的"rows"行数属性。
- 各框架对应的页面情况：使用<frame>标签的"src"属性引用各框架对应的页面文件，同时还可以使用"name"属性标识各框架窗口。

需要注意的是：<frameset>标签和<body>标签不能同时使用，所以需要使用"<frameset>"代替页面中的"<body>"标签。对应的 HTML 代码如示例 1 所示。

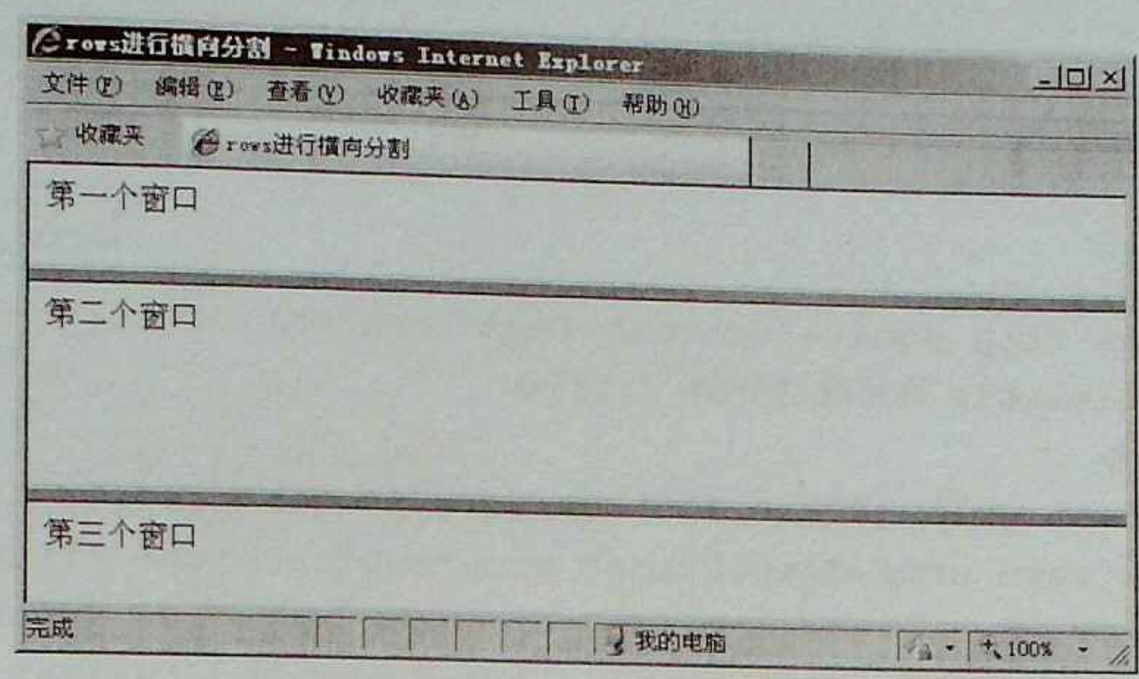


图 4.2　纵向分割为上中下三个窗口

示例 1

```
<html>
  <head>
    <title>rows 纵向分割为 3 个窗口</title>
  </head>
  <frameset  bordercolor="red"  rows="25%,50%,*"  border="5">
       <frame src="subframe/the_first.html" name="top" />
       <frame src="subframe/the_second.html" name="middle" />
       <frame src="subframe/the_third.html" name="bottom" />
  </frameset>
</html>
```

其中，为了突出显示各框架，加了宽度为 5 的红色边框。另外，由于框架网页包含多个页面，为了分清框架结构页以及各框架窗口对应的子页面，特意将各子页面单独放到文件夹“subframe”中。

4.2.2　横向分割窗口

横向分割窗口的思路与纵向分割窗口很类似，例如，要实现如图 4.3 所示的横向分割的页面效果，只需要设置<frameset>的“cols”列数属性即可，完整的 HTML 代码如示例 2 所示。

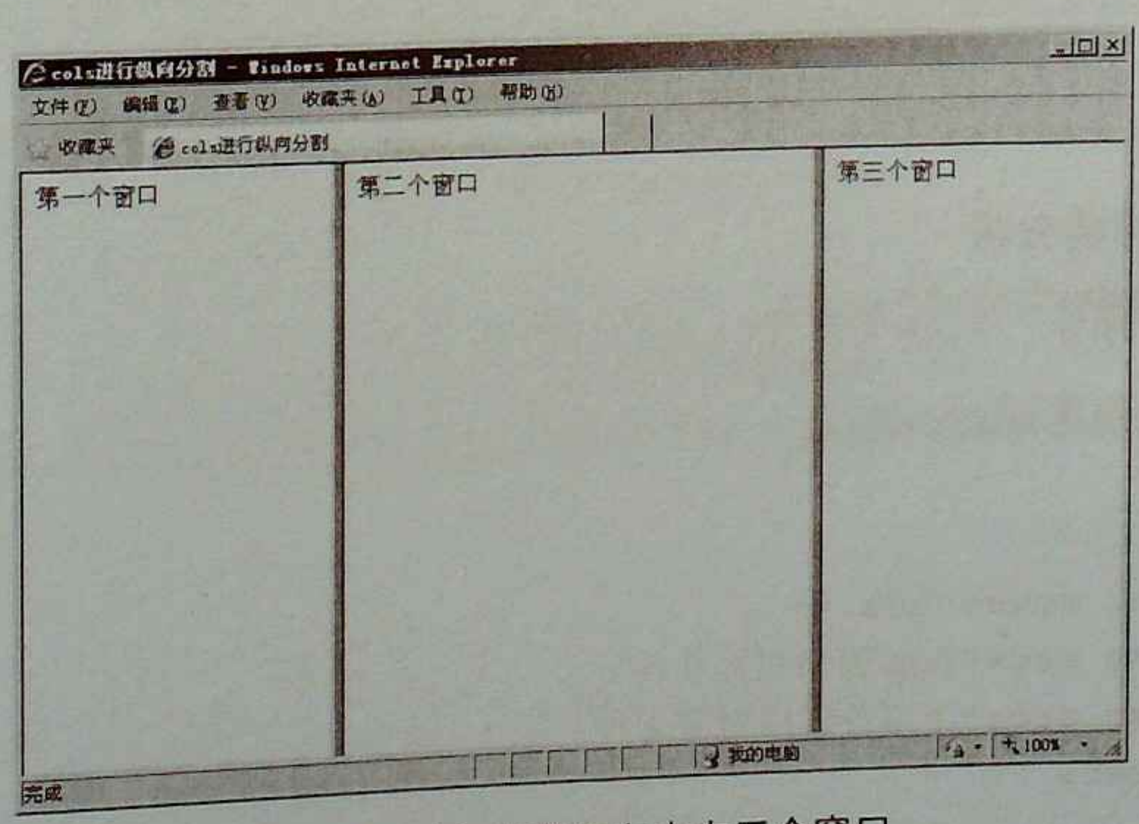


图 4.3　横向分割为左中右三个窗口

示例 2

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>cols 进行纵向分割</title>
  </head>
  <frameset cols="200,*,200" border="5" bordercolor="#FF0000">
        <frame name="leftFrame" src="subframe/the_first.html" />
        <frame name="mainFrame" src="subframe/the_second.html" />
        <frame name="rightFrame" src="subframe/the_third.html" />
  </frameset>
</html>
```

4.2.3 横向和纵向同时分割窗口

接下来以图 4.4 中典型的两行两列结构为例，我们来分析其实现思路。

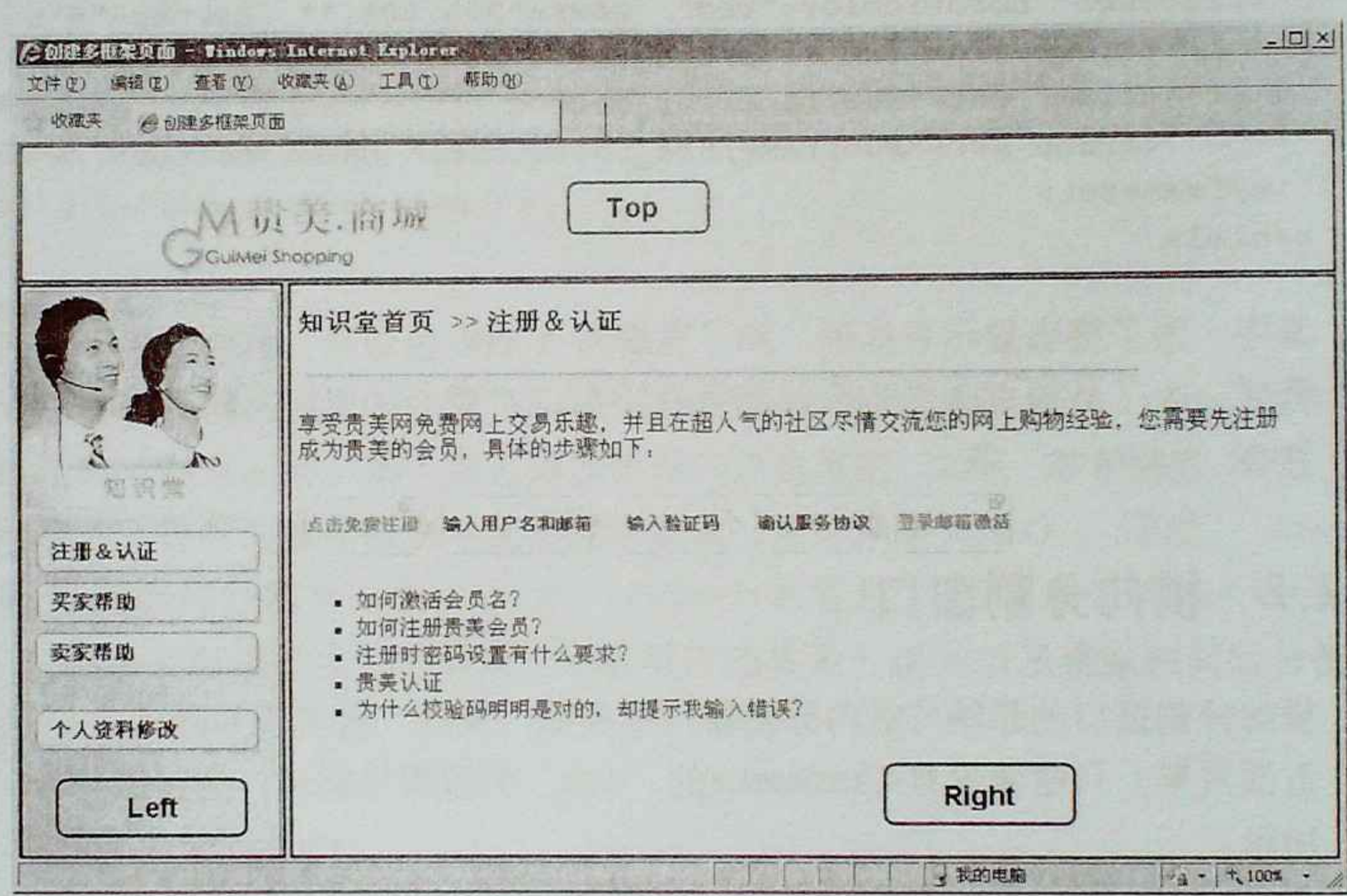


图 4.4 典型的 2 行 2 列结构示意图

1. 页面结构分析

如图 4.4 所示，对整个页面结构的分析如下。

(1) 整个页面纵向分割为上、下两部分，高度分别为窗口的 20%和 80%，对应的关键代码如下。

```
<frameset rows="20%,*" >
   <frame src="Top 窗口对应文件"/>
   <frame src="下部分窗口对应文件"/>
</frameset>
```

(2) 下部分再次横向分割为左、右两部分，宽度分别为窗口的 20%和 80%。即需要把上述第二个<frame>改为<frameset>实现，对应的关键代码如下。

```
<frameset rows="20%,*" >
   <frame src="Top 窗口对应文件"/>
   <frameset cols="20%,*">
     <frame src="Left 窗口对应文件"/>
     <frame src="Left 窗口对应文件"/>
   </frameset>
</frameset>
```


2. 框架修饰分析

要实现上述框架效果，除边框外，还需用到框架的其他修饰属性。例如，是否允许调整各框架窗口的大小，则使用“noresize”属性设置；当框架内的页面内容较多时，是否需要显示滚动条，则使用“scrolling”属性设置。框架<frame>常用的属性如表 4-1 所示。

表 4-1　框架<frame>的常用属性

属性	作用	举例
frameborder	是否显示框架周围的边框	frameborder="1"
name	框架标识名	name="mainFrame"
scrolling	是否显示滚动条	scrolling="no"
noresize	是否允许调整框架窗口大小	noresize="noresize"

图 4.4 对应的 HTML 代码如示例 3 所示，使用“name”属性标识各窗口的名称，用于后续建立框架窗口间的关联。

示例 3

```
<html>
  <head>
    <title>创建多框架页面</title>
  </head>
  <frameset rows="20%,*" frameborder="0">
      <frame src="subframe/top.html" name="topframe"
            scrolling="no" noresize="noresize" />
      <frameset cols="20%,*">
            <frame src="subframe/left.html"  name="leftframe"
                scrolling="no" noresize="noresize" />
            <frame src="subframe/right.html" name="rightframe" />
       </frameset>
  </frameset>
</html>
```

4.2.4 如何实现窗口间的关联

学习了如何使用框架创建多个窗口后，下面介绍如何建立窗口间的关联。例如，单击如图 4.5 所示，左侧窗口中的导航栏链接，在右侧窗口将显示对应的内容。

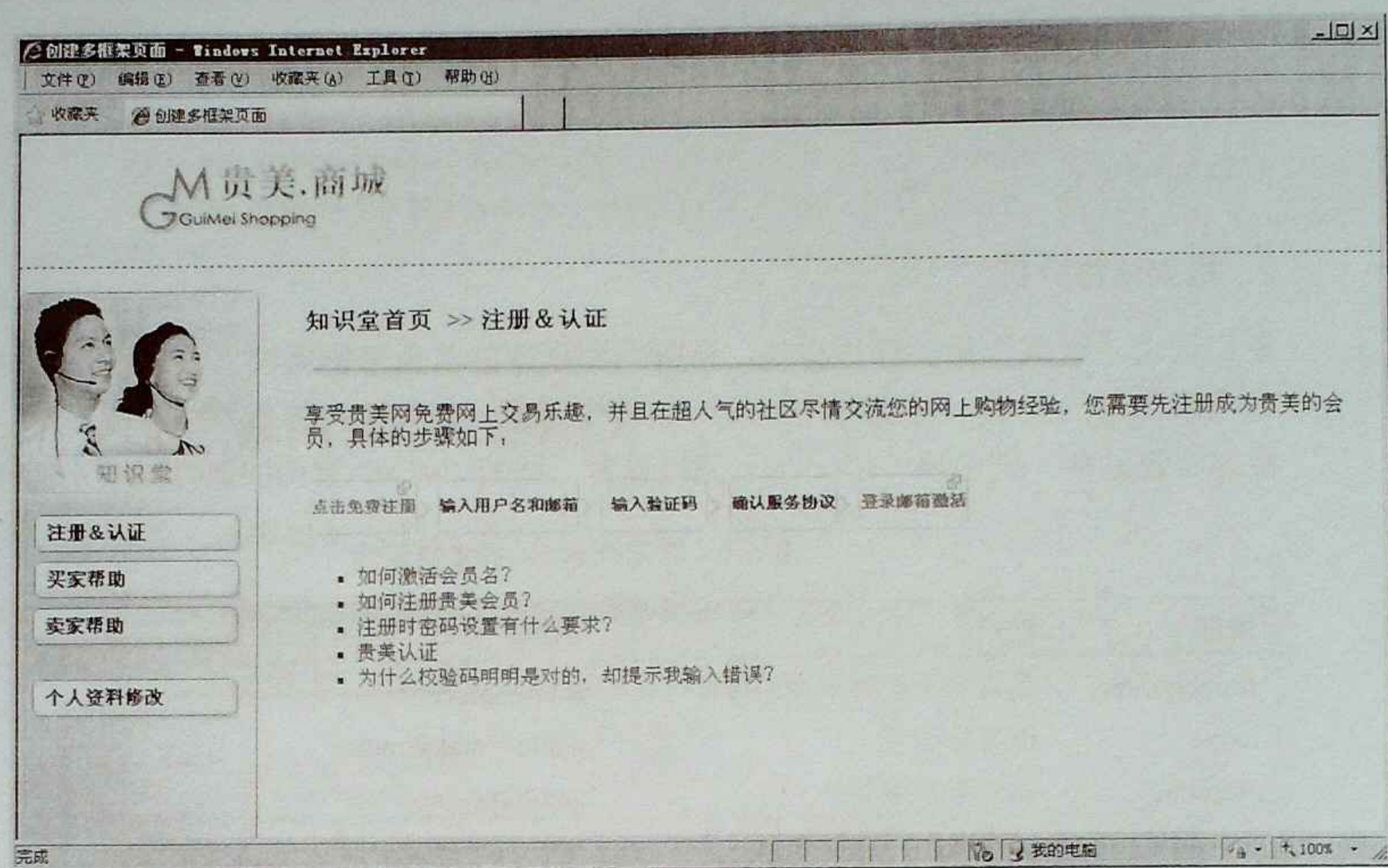


图 4.5 框架窗口间的关联

如何实现？关键在于设置超链接的“target”目标窗口属性，具体的实现思路如下。

（1）在框架页面中，为右侧框架窗口添加“name”名称标识，例如 rightframe。这一步在上例已实现，关键代码如下。

```
<frame src="subframe/right.html" name="rightframe" />
```

（2）在左侧窗口对应的页面中，设置超链接“target”目标窗口属性为希望显示的框架窗口名，在右侧窗口显示即为：

```
<a href="right.html"  target="rightFrame">……</a>
```

整个框架页的代码如示例 3 所示，关于左侧窗口对应的页面（left.html），其关键代码如示例 4 所示。

示例 4

```
<!--省略部分 HTML 代码-->
 <span>
    <a  href="right.html"  target="rightFrame"><img  src="images/reg.jpg"
alt="注册" /></a>
```

```
</span>
<!—省略了其他类似链接的 HTML 代码-->
```

在第 2 章，我们曾学习过<a>链接标签的 target 属性，我们小结一下其用法，如表 4-2 所示。

表 4-2　target 属性的取值

属性值	含义
_blank	在新窗口中打开链接
_self	在链接所在页面的自身窗口中打开链接
框架窗口名	在指定的框架窗口中打开链接
_parent	在父框架集中打开链接，如果不是框架网页，则含义同“_self”
_top	在顶级窗口即整个浏览器窗口中打开链接

除_parent 和_top 两个属性值外，其他属性值都学过，下面演示_parent 和_top 两个属性值的用法。修改本例左侧窗口对应的 left.html 页面，分别设置链接打开的目标窗口为“_parent”和“_top”，对应的 HTML 代码分别如下。

```
<a href="buy.html" target="_parent">……</a>
<a href="buy.html" target="_top">……</a>
```

如图 4.6 所示，本例中两者的效果一样，都在整个框架页的浏览器窗口中显示。究其原因，是因为本例只有两个层次的二级框架，所以框架页所在的浏览器窗口是左侧窗口的父窗口，同时也是框架集页面的顶级窗口。

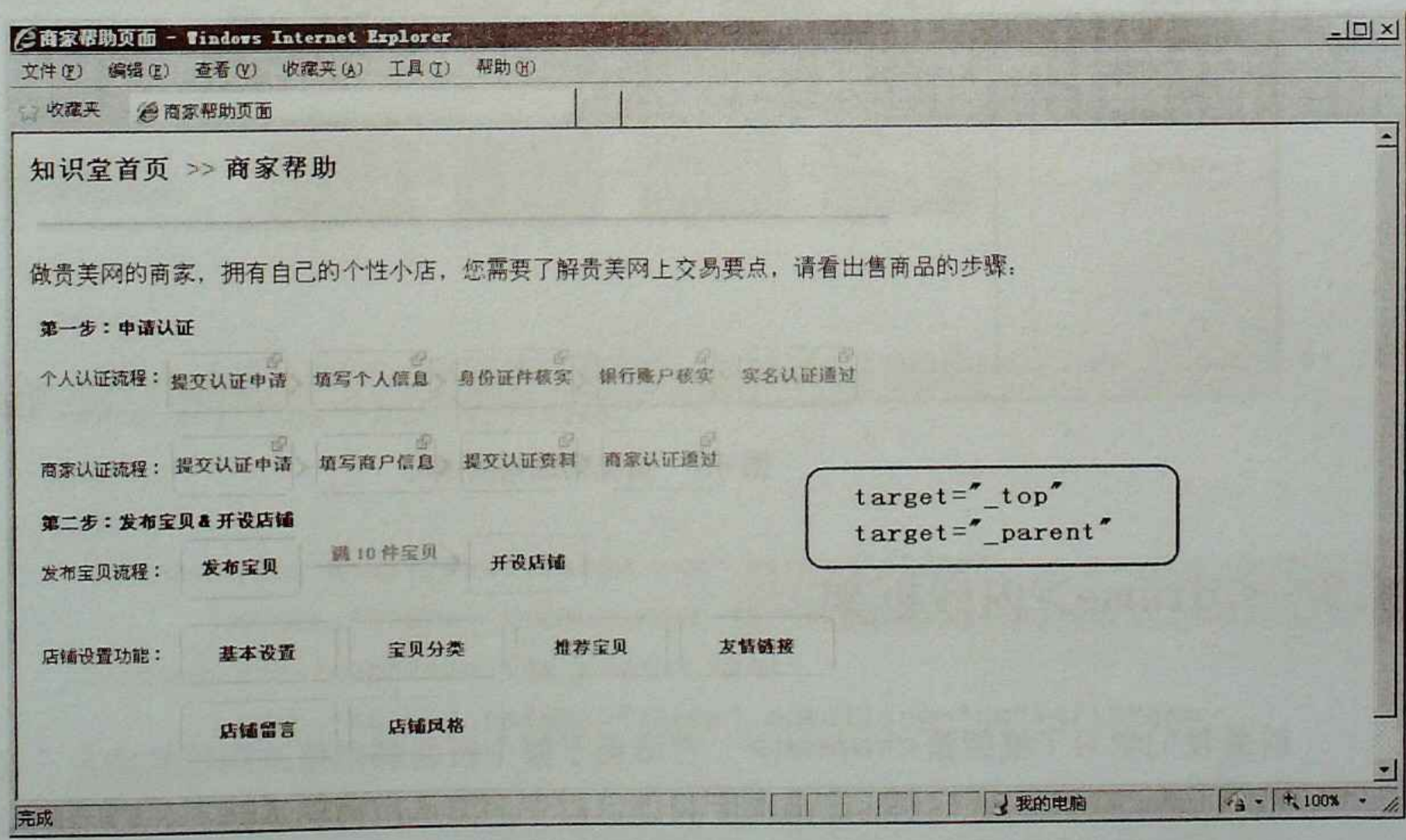


图 4.6　target 属性为_top 和_parent

4.2.5 上机练习

上机练习 1

练习——用框架分割多个窗口

需求说明

用提供的素材实现典型的两行两列框架结构，边框设置为宽度 5，白色，效果如图 4.7 所示。

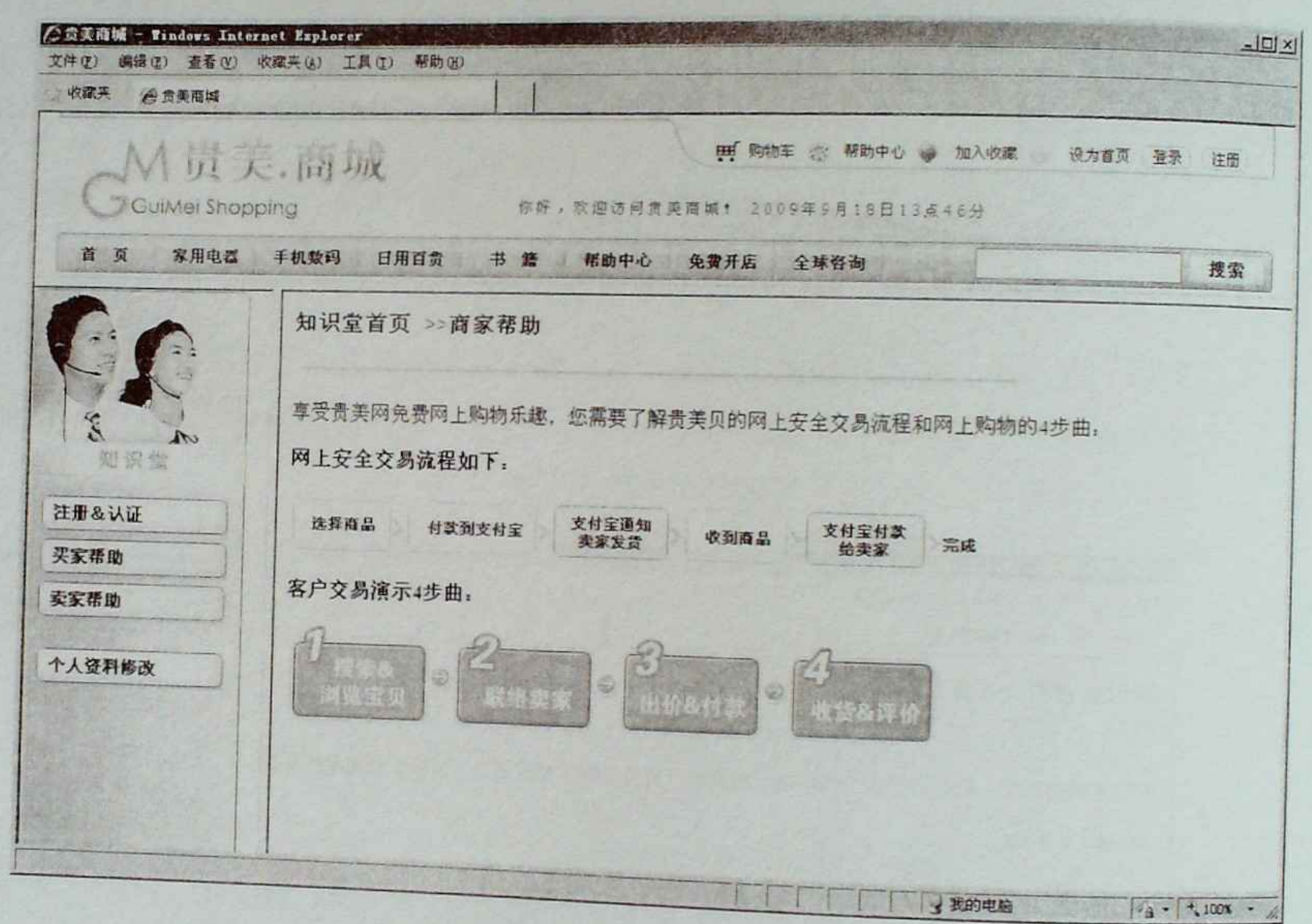


图 4.7 贵美商城帮助中心

4.3 <iframe>内嵌框架

前面我们学习了框架集<frameset>，它适用于整个页面都用框架实现的场合，这节我们将学习<iframe>内嵌框架，它适用于将部分框架内嵌入页面的场合，一般用于引用其他网站的页面。例如，在自己制作的网页中引用搜狐网页的新闻页面等。

4.3.1 <iframe>的用法

1. <iframe>语法

<iframe >用法和<frame >比较类似，其语法如下。

```
<iframe src="引用页面地址" name="框架标识名" frameborder="边框" scrolling="是否出现滚动条" ></iframe>
```

例如，引用站外的百度搜索页和站内的一个页面，如图 4.8 所示，对应的 HTML 代码如示例 5 所示。

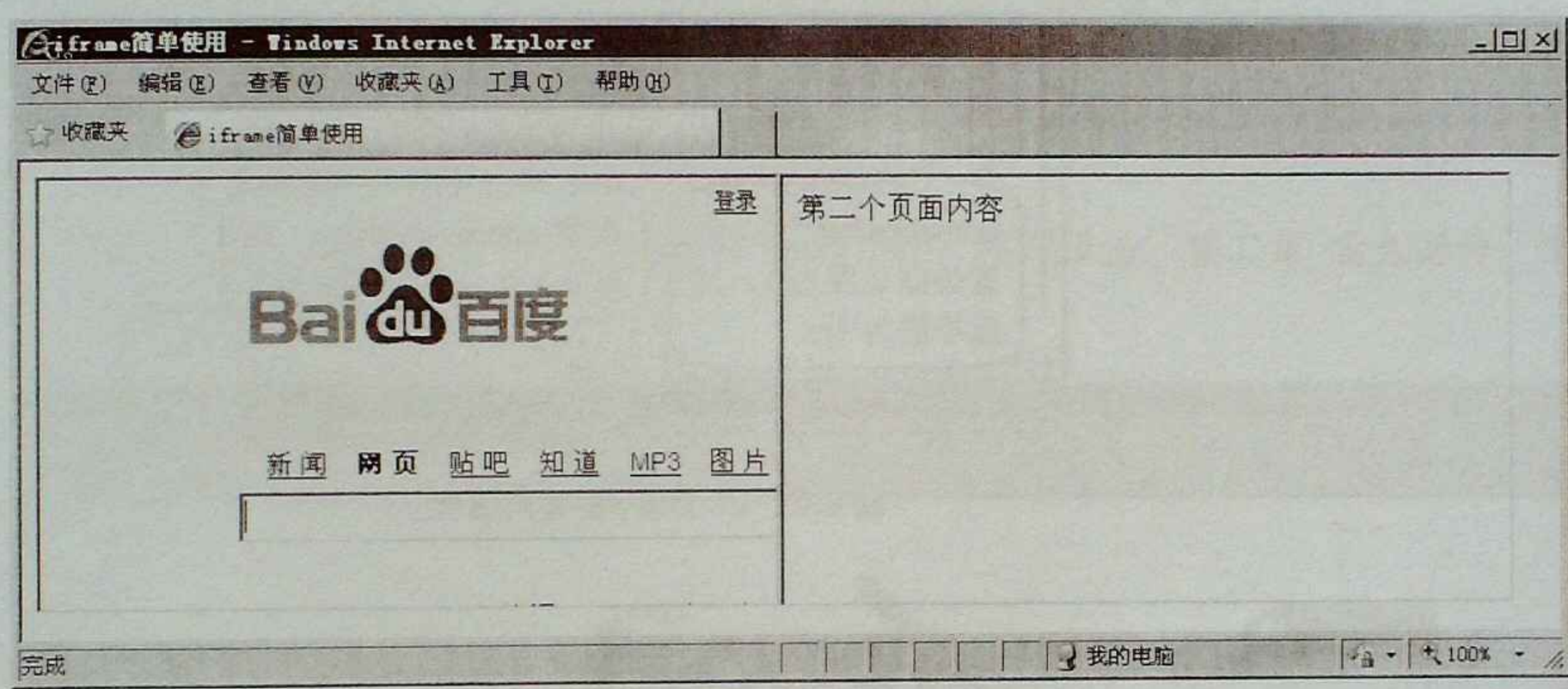


图 4.8　<iframe >的简单使用

示例 5

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>iframe 简单使用</title>
  </head>
  <body>
    <iframe src="http://www.baidu.com" width="400px"
          height="236px" frameborder="1" scrolling="no"></iframe>
    <iframe src="subframe/the_second.html"
          width="400px" height="236px" scrolling="no"></iframe>
  </body>
</html>
```

2. 常用属性

类似前面学习的<frameset>框架，<iframe>内嵌框架的常用属性包括 name、scrolling、

noresize 和 frameborder。其中 name、noresize 和 scrolling 与表 4-1 所列<iframe>属性的作用一样。例如，实现链接在<iframe>内嵌框架中打开的效果，如图 4.9 所示，对应的 HTML 代码如示例 6 所示。

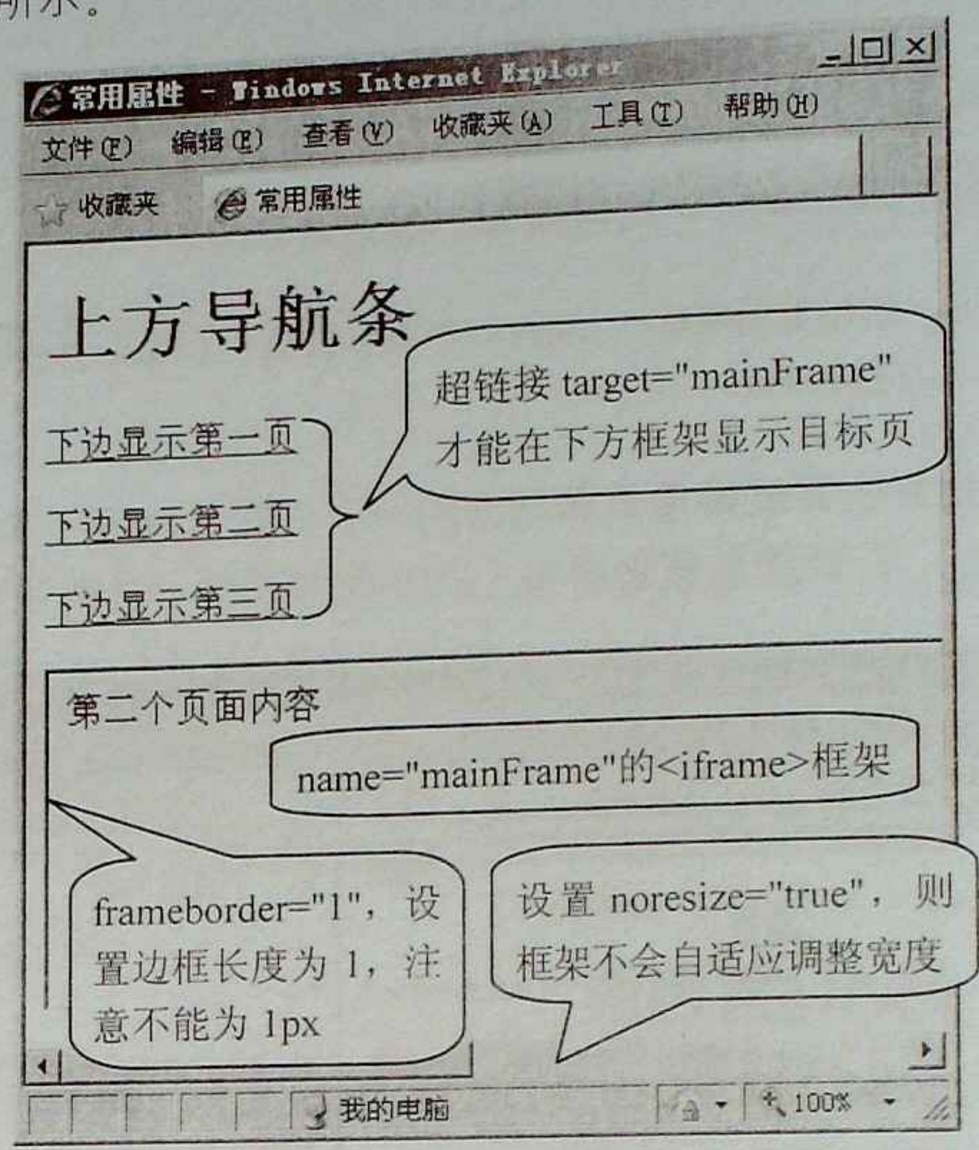


图 4.9 <iframe >的常用属性

示例 6

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
   <title>&lt;iframe&gt;常用属性</title>
  </head>
  <body>
    <h1>上方导航条</h1>
    <p><a href="subframe/the_first.html" target="mainFrame">
          下边显示第一页</a><br /><br />
    <a href="subframe/the_second.html" target="mainFrame">
          下边显示第二页</a><br /><br />
    <a href="subframe/the_third.html" target="mainFrame">
          下边显示第三页</a><br />
    </p>
    <iframe name="mainFrame" width="800px" height="150px" scrolling="yes"
          noresize="noresize" src="subframe/the_second.html" />
  </body>
</html>
```

4.3.2 上机练习

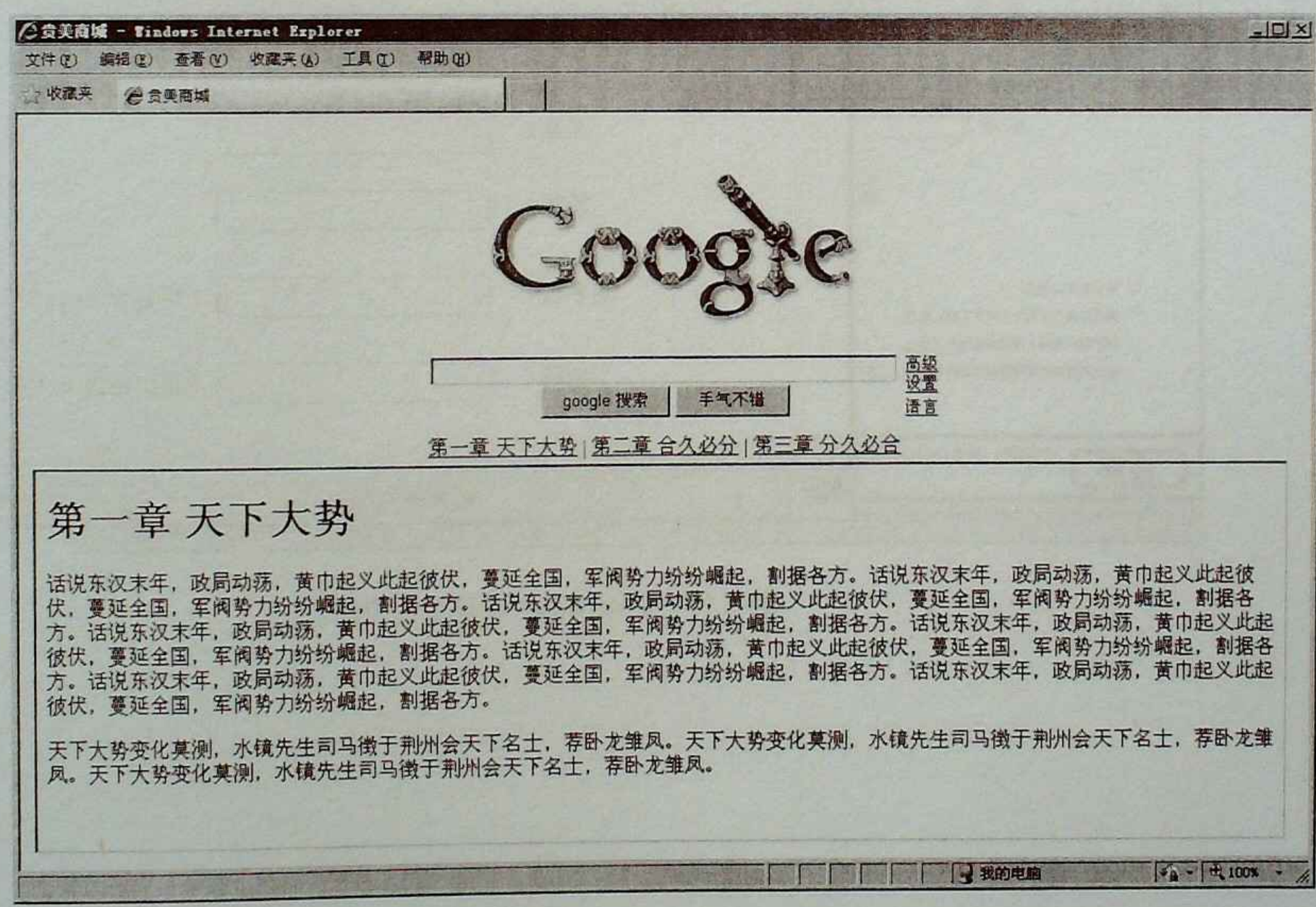
上机练习 2

练习——引用 Google 主页

需求说明

用提供的素材实现 Google 页引用，效果如图 4.10 所示，具体要求如下。

- 上方为 Google 引用。
- 下方为小说章节内容。
- 单击“第一章 天下大势”下方显示 text1.htm，同理单击“第二章 合久必分”下方显示 text2.htm。



第一章 天下大势 | 第二章 合久必分 | 第三章 分久必合

第一章 天下大势

话说东汉末年，政局动荡，黄巾起义此起彼伏，蔓延全国，军阀势力纷纷崛起，割据各方。话说东汉末年，政局动荡，黄巾起义此起彼伏，蔓延全国，军阀势力纷纷崛起，割据各方。话说东汉末年，政局动荡，黄巾起义此起彼伏，蔓延全国，军阀势力纷纷崛起，割据各方。话说东汉末年，政局动荡，黄巾起义此起彼伏，蔓延全国，军阀势力纷纷崛起，割据各方。话说东汉末年，政局动荡，黄巾起义此起彼伏，蔓延全国，军阀势力纷纷崛起，割据各方。话说东汉末年，政局动荡，黄巾起义此起彼伏，蔓延全国，军阀势力纷纷崛起，割据各方。

天下大势变化莫测，水镜先生司马徽于荆州会天下名士，荐卧龙雏凤。天下大势变化莫测，水镜先生司马徽于荆州会天下名士，荐卧龙雏凤。天下大势变化莫测，水镜先生司马徽于荆州会天下名士，荐卧龙雏凤。

图 4.10 Google 页的引用

上机练习 3

练习——使用<iframe>实现页面重用

需求说明

用提供的素材实现内容重用，效果如图 4.11 所示，具体要求如下。

- 在给出的登录和注册页中重用贵美页面顶部和底部。
- 只需找到 login.htm 和 register.htm 页面中的<iframe>代码行，并增加属性即可。

图 4.11 用<iframe>实现页面重用

本章总结

- 常用的框架技术包括<frameset>框架集和<iframe>内嵌框架。
- <frameset>框架集结构非常清晰，适用于整个页面都用框架实现的场合；<iframe>内嵌框架使用比较方便、灵活，一般用于在页面中引用站外的页面内容时。
- <frameset>框架集使用<frameset>和<frame>标签实现，<frameset>的 rows 和 cols 属性实现窗口的横向和纵向分割。
- 配合使用<a>标签的 target 属性以及<frame>标签的 name 属性，可以实现窗口间的关联。
- <a>标签的 target 属性除设置为具体的框架名外，还包括_blank、_self、_top、_parent 四种特殊窗口。
- 和<frameset>框架相比，<iframe>内嵌框架比较灵活，只需要一句代码即可引用站外或站内的某个网页。

本章作业

一、选择题

1. 框架集的标签名是（　　）。

A. frame

B. setframe

C. frameset

D. iframe

2. 下列常用属性设置中，（　　）适用于<iframe >标签。

A. border="0"

B. frameborder="1"

C. scrolling="no"

D. src="head.htm"

3. 实现如图 4.4 所示的 2 行 2 列典型结构，代码（　　）是正确的。

A.
```
<frameset rows="20%,*">
    <frame />
    <frameset cols="30%,*"><frame /><frame /></frameset>
</frameset>
```

B.
```
<frame cols="20%,*">
    <frameset />
    <frame cols="30%,*"><frameset /><frameset /></frame>
</frame>
```

C.
```
<frameset cols="20%,30%,*">
    <frame />
    <frameset cols="30%,*"><frame /><frame /></frameset>
</frameset>
```

D.
```
<frameset cols="20%,*">
    <frame />
    <frameset rows="30%,40%,*"><frame /><frame /></frameset>
</frameset>
```

4. 要在新窗口中打开链接，<a>中需要选用属性（　　）。

A. target="_top"

B. target="_parent"

C. target="_blank"

D. target="_self"

5. 关于页面的重用，说法正确的是（　　）。

A. 可以减少不必要的重复工作

B. 可以提高重用内容的维护效率

C. 可以提高网站的访问速度、下载速度

D. 页面重用是网站常用的技术之一

二、简答题

1. 根据你的理解，请简述框架的功能和应用场景。

2. 和<frameset>相比，<iframe>有哪些优点？

3. 请描述一个二行二列的典型框架集的实现过程。

4. 分别阐述如何用<frameset>和<iframe>实现常见的两列窗口关联效果。例如，单击左边的链接，在右边窗口显示链接目标。

5. 根据给出的素材，用<iframe>实现对顶部及底部的重用。效果如图 4.12 所示。

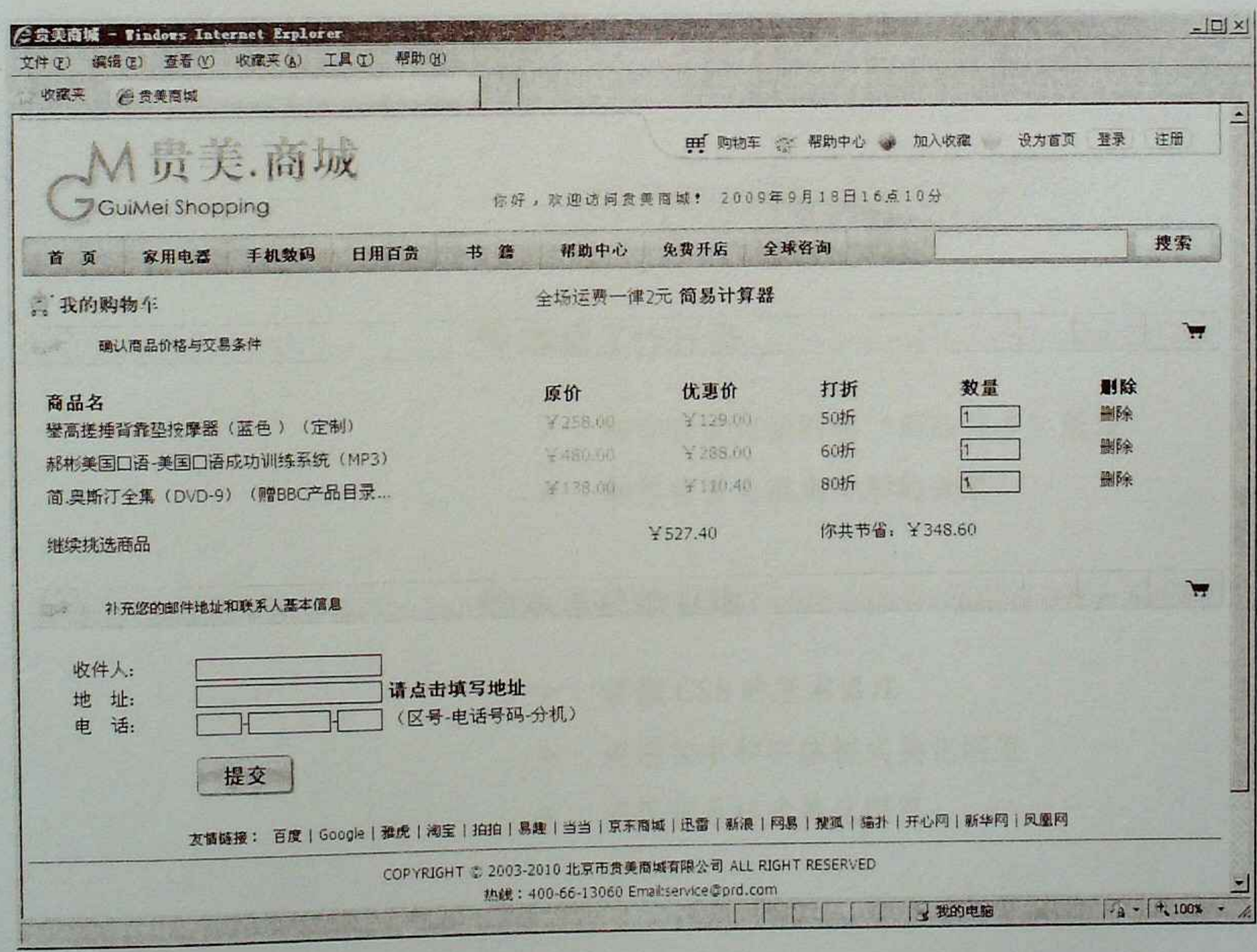


图 4.12　贵美购物车页

第5章

CSS 样式表（一）

◈本章工作任务

- 制作贵美商城网站“商品分类”版块
- 制作背景为圆角矩形的效果

◈本章技能目标

- 掌握 CSS 的基本语法
- 使用文本和字体样式美化网页
- 使用背景样式美化网页

本章单词

请在预习时学会下列单词的含义和发音，并填写在横线处。

1. style：_____
2. family：_____
3. background：_____
4. repeat：_____
5. float：_____
6. decoration：_____
7. space：_____
8. position：_____

本章简介

在第 1 章曾提及 W3C 提倡的 Web 页结构是内容和样式分离，其中 XHTML 负责组织内容结构，CSS 负责表现样式。通过前面章节的学习，我们学会了如何使用 HTML 标签组织内容结构，并要求内容结构具有语义化。从本章开始，我们将学习负责表现样式的 CSS 部分。

本章将介绍使用 CSS 的好处、基本语法、文本、背景等常见的样式修饰，重点是理解内容和样式分离的思想、CSS 的基本语法和各类常用的修饰。

除教材外，学员还可以在课下学习教员提供的电子资料“DIV+CSS 布局大全”，多看多练是学好 CSS 的关键。

5.1 为什么使用 CSS

CSS 是 Cascading Style Sheets 的缩写，一般翻译为层叠样式表。W3C 的构想是 HTML 标签只表示内容结构，即只表示“这是一个段落”、“这是一个标题”、“这是一个项目列表”等含义，而不具备任何样式；而这些“段落”、“标题”等内容的字体类型、字号大小、演示、显示位置等样式，完全由 CSS 指定，从而实现结构和样式的分离。所以，W3C 制定了相应的 XHTML 和 CSS 两套标准，目前应用较多的版本分别是 XHTML 1.0 和 CSS 2.0。

早期曾使用 HTML 标签表示样式，和使用 CSS 表示样式相比，使用 CSS 具有如下突出优势。

- 实现内容和样式的分离，利于团队开发。

  样式美化可以由美工人员负责，而软件开发人员主要负责页面内容的开发。

- 实现样式复用，提高开发效率。

  同一网站的多个页面可以共用同一个样式表，提高了网站的开发效率，同时也方便对网站的更新和维护。如需要更新网站外观，则更新网站的样式表文件即可。

- 实现页面的精确控制。

  和带有样式的 HTML 标签相比，CSS 具有强大的样式控制能力和排版能力。CSS 包含文本（含字体）、背景、列表、超链接、外边距等丰富的各类样式，可以实现各种复杂、精美的页面效果。

- 更利于搜索引擎的搜索。

  和早期同时包含内容和样式的 Web 页相比，内容和样式分离后，减少了原 Web 页

的代码量，使得 Web 页的内容结构更加突出，利于搜索引擎更加有效地搜索 Web 页面。因此，W3C 不推荐使用带样式的 HTML 标签，HTML 标签中也尽量少用带样式的属性。

CSS 的作用是定义外观及布局，图 5.1 为应用 CSS 样式前后的对比效果。

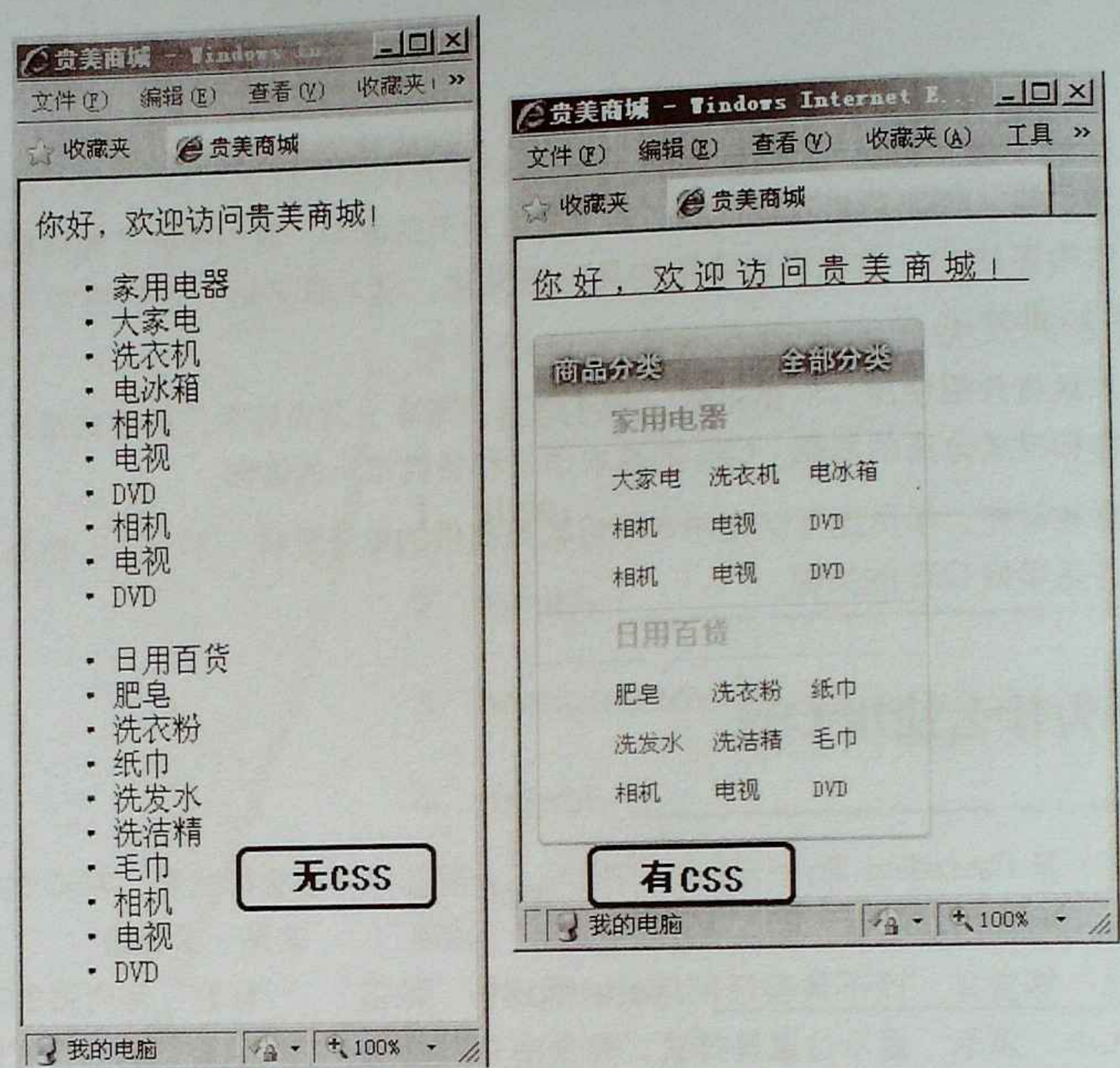


图 5.1 应用 CSS 样式的前后对比

CSS 样式常用的两大用途是页面内容（元素）修饰和页面布局。下面先介绍 CSS 的基本布局，再介绍页面内容的修饰，页面布局放在下一章讲解。

5.2 CSS 基本语法

样式表由样式规则组成，这些规则告诉浏览器如何显示文档。一个样式(Style)的基本语法由三部分构成：选择器、属性和属性值。我们先来了解其基本结构及其出现的相关概念。

5.2.1 基本结构

层叠样式表一般用<style>标签来声明样式规则，即告诉浏览器页面中各类内容或页面元素应如何显示，其基本结构如下。

```
<style type="text/css">
     选择器{
          对象的属性 1:属性值 1;
          对象的属性 2:属性值 2;
    }
</style>
```

其中选择器表示被修饰的对象，例如页面中被修饰的段落<p>、列表<li>等；属性是希望改变的样式，例如颜色 color、字体大小 font-size 等。属性和属性值用冒号（：）隔开。例如，设页面中所有的 标签的文字颜色为红色，字体大小为 30px，字体类型为隶书。对应的样式规则则为：

```
<style type="text/css">
   li { color:red ;
         font-size:30px ;
         font-family:隶书 ;
  }
</style>
```

代码规范

- 虽然 CSS 代码不区分大小写，但推荐全用小写。
- 每条样式规则用分号（；）隔开，一般写为多行，简单的规则也可以合并为一行。
- 当 CSS 代码较多时，可以使用“/*.....*/”添加必要的注释，增加代码的可读性。

5.2.2 选择器的分类

根据选择器表示所修饰的内容类别，选择器又分为标签选择器（有的教材也称元素选择器）、类选择器、ID 选择器。

1. 标签选择器

当需要对页面内某类标签的内容进行修饰时，则采用标签选择器，这些标签可以是前面章节学过的所有 HTML 标签。其用法如下。

```
标签名{属性名 1：属性值 1 ；
       属性名 2：属性值 2 ；
       ......
       }
```

例如，希望修饰页面中的所有项目列表项（<li>）的样式为：字体大小为 28px、红色、隶书，如图 5.2 所示。对应的 CSS 代码如示例 1 所示。

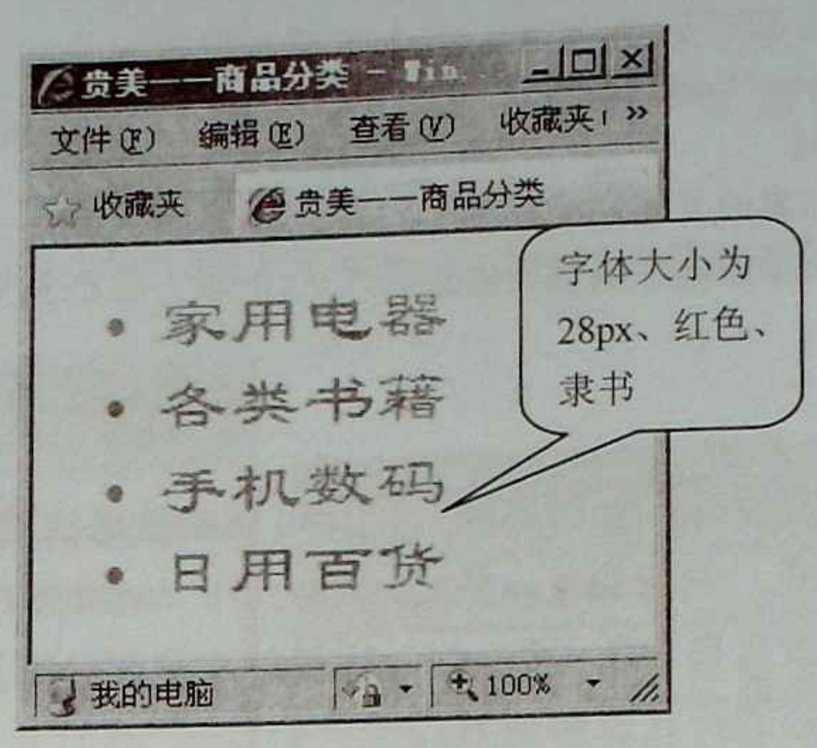


图 5.2 标签选择器

示例 1

```
<html>
  <head>
     <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
     <title>左侧菜单</title>
     <style type="text/css">
        li{
        color:red;
        font-size:28px;
        font-family:隶书;
        }
      </style>
    </head>
    <body>
      <div>
        <ul>
            <li>家用电器</li>
            <li>各类书籍</li>
            <li>手机数码</li>
            <li>日用百货</li>
        </ul>
      </div>
    </body>
</html>
```

需要注意，本例为了讲解方便，将样式修饰的 CSS 代码单独放到<head>区域内，实现了页面和内容的分离。但 W3C 推荐采用更彻底的分离方式，即将 CSS 文件单独放在另一个样式文件中，后续章节将会深入讲解。

2. 类选择器

从示例 1 可以看出，使用标签选择器修饰的范围比较广，如果希望设置个别<li>元素的

样式和其他<li>元素不同，该如何实现？答案就是使用类选择器。类选择器类似 Java 中学习的 class（类），使用步骤分为以下两步。

（1）定义样式，语法如下。

```
.类名{属性名 1:属性值 1;
      属性名 2: 属性值 2 ;
      ……
      }
```

（2）应用样式，使用标签的“class”属性引用类样式，即<标签名 class="类名">标签内容</标签名> 。

需要注意：定义时类名前有个点号（.），应用样式时则不需要点号。

对于上述提出的问题，对应的实现思路为：先定义表示某类样式的类选择器，再设置与众不同的<li>元素应用这类样式即可。

例如，修改示例 1，希望列表项“家用电器”、“手机数码”和其他列表项目的颜色不同，显示为蓝色，如图 5.3 所示，对应的 CSS 代码如示例 2 所示。

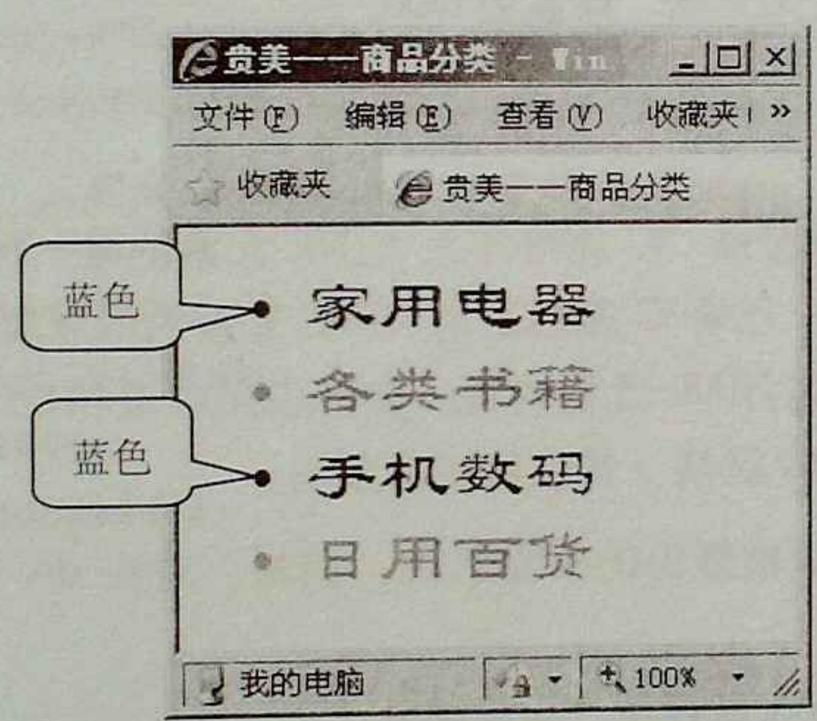


图 5.3 类选择器的用法

示例 2

```
<html>
  <head>
     <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>贵美——商品分类</title>
   <style>
     li{
         color:red;
         font-size:28px;
         font-family:隶书;
     }
     .blue{color:blue;}
   </style>
```

定义时类名前有个点号

```
</head>
<body>
 <div>
     <ul>
          <li class="blue">家用电器</li>
          <li>各类书籍</li>
          <li class="blue">手机数码</li>
          <li>日用百货</li>
     </ul>
 </div>
</body>
</html>
```

应用类样式，应用时不需要点号

定义类选择的好处是任何标签都可以应用该类样式，从而实现样式的共享和代码复用。需要注意，样式是叠加和继承的。例如，对于列表项“家用电器”和“手机数码”，由于<li>标签选择器先规定了页面内所有列表项的字体类型、字体大小、颜色，所以这两个列表项的字体类型和大小和其他列表项一样；至于颜色方面，由于这两个列表项还应用了 blue 类样式，即产生了样式的叠加，CSS 规定后定义的样式覆盖前面定义的样式，所以颜色以最后的定义为准，即蓝色 blue。

3．ID 选择器

ID 属性类似我们的身份证，ID 属性作为 HTML 元素的唯一标识，要求页面内不能有重复的 ID 标识属性，W3C 已将 ID 属性定为 HTML 元素的标准属性。对应的 ID 选择器一般用于修饰对应 ID 标识的 HTML 元素内容，实际应用中常和 标签配合使用，表示修饰对应 ID 标识的某个 div 区块，其使用步骤如下。

（1）使用 ID 属性标识被修饰的页面元素。例如，对应 div 块，则为 。

（2）定义相应的 ID 选择器样式，语法如下。

```
#ID 标识名{属性名 1:属性值 1;
          属性名 2: 属性值 2 ;
          ……
         }
```

需要注意，定义 ID 选择器时有个井号（#），但给 HTML 标签设置 ID 属性时不需要。

很明显，ID 选择器和类选择器的用途刚好相反：ID 选择器用于修饰某个指定的页面元素或区块，这些样式是对应 ID 标识的 HTML 元素所独占的；而类选择器是定义某类样式让多个 HTML 元素共享，这些样式是可以共享和代码复用的。

例如，修改示例 2，如图 5.4 所示，将整个列表项看作一个带 ID 标识的 div 块，现希望只对该 div 块进行字体、宽度、背景色的修饰，则实现思路如下。

（1）将所有列表项内容放到一个 div 块中，并设置唯一的 ID 标识属性（例如“menu”）。

（2）根据上述 ID 标识，定义对应的 ID 选择器。

对应的 CSS 代码如示例 3 所示。

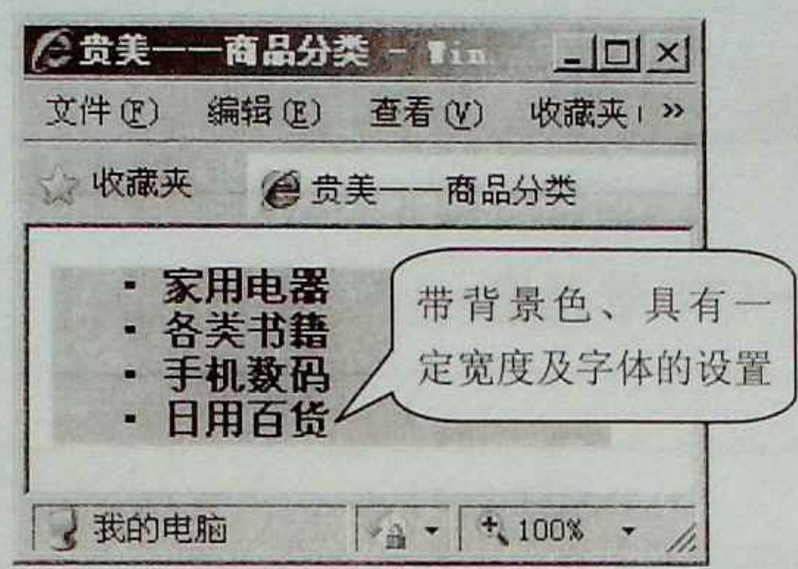


图 5.4 ID 选择器

示例 3

```
<html>
  <head>
     <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>贵美——商品分类</title>
     <style>
       #menu{
            font-size:14px;
            font-family: 宋体;
            width:200px;
            background:#ccc;
    }
     </style>
  </head>
  <body>
     <div id="menu">
       <ul>
            <li>家用电器</li>
            <li>各类书籍</li>
            <li>手机数码</li>
            <li>日用百货</li>
       </ul>
     </div>
   </body>
</html>
```

定义 ID 选择器，对应标识为 menu

标识 HTML 元素的 id 属性

5.2.3 上机练习

上机练习 1

练习——CSS 选择器

需求说明

利用 div-ul-li 结构组织及各类选择器制作如图 5.5 所示的页面效果，具体要求如下。

- 整个 总宽度 200px，背景灰色（#ccc）。
- 列表项“家用电器”和“日用百货”字体为橘色（#ff7300），大小为 14px。
- 其余列表项的字体大小为 12px。

提示

HTML 页面结构采用 内套两个 ul-li 结构，然后再根据样式要求，分别考虑用哪类选择器实现。

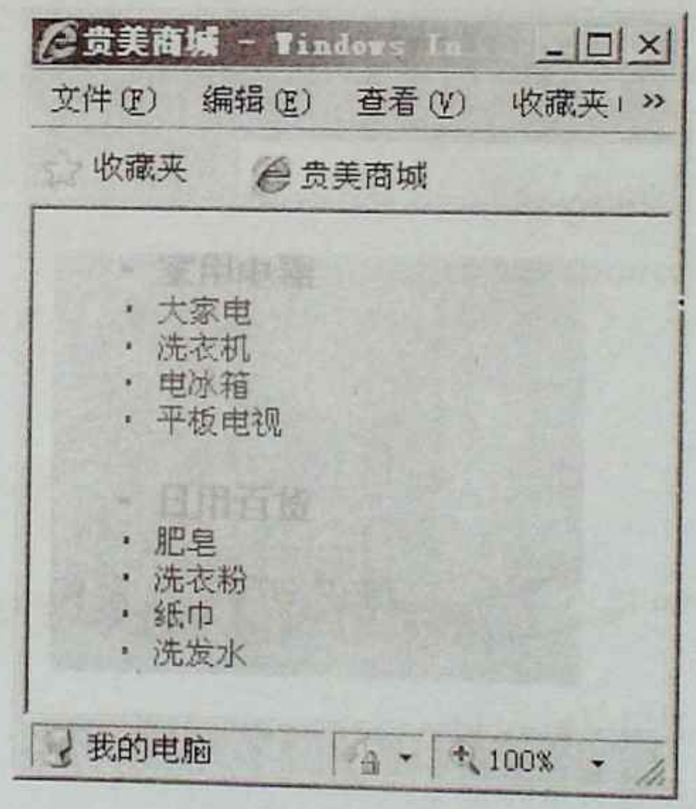


图 5.5 贵美商品分类

5.3 常用的样式修饰

网页元素可以修饰的样式属性很多，常用的样式分为：文本及字体、背景、列表几个方面，下面逐一介绍。

5.3.1 文本及字体属性

文本属性用于定义文本的外观，包括文本颜色、行高、对齐方式、字符间距等，常用的文本属性如表 5-1 所示。

表 5-1 常用的文本属性

属性名	含义	举例	应用场景
line-height	设置行高（即行间距），常用取值为 25px、28px	line-height="25px"	布局多行文本
text-align	设置对齐方式，常用取值为 left、right、center	text-align="center"	各种元素对齐
letter-spacing	设置字符间距，常用取值为 3px、8px	letter-spacing="5px"	加大字符间间隔
text-decoration	设置文本修饰，常用取值 underline（下划线）、none	text-decoration="underline"	加中划线、下划线等
white-space	规定如何处理空白，例如规定是否换行等，常用取值为 nowrap（不换行）	white-space="nowrap"	文本溢出时不断行

字体属性用于定义字体类型、字号大小、字体是否加粗等，常用的字体样式如表 5-2 所示。

表 5-2 常用的字体属性

属性名	含义	举例	应用场景
font	在一个声明中设置字体的所有样式属性	font：bold 12px 宋体	常用于字体样式的缩写
font-family	定义字体类型	font-family：宋体	定义字体样式
font-size	定义字体大小	font-size：12px	定义字体样式
font-weight	定义字体的粗细	font-weight：bold	定义字体样式

其中，“font-family”、“font-size”等是“font”属性的子属性，所以一般常使用字体的缩写形式，即利用“font”属性一次设置字体的所有样式属性，例如“font:bold 12px 宋体;”，但需注意三种格式的顺序依次为粗细→字体大小→字体类型。

例如，假定要实现如图 5.6 所示的文本、字体修饰效果，我们可以根据样式要求，先定义适用于所有列表项<li>的标签选择器样式，然后再分别定义适用于个别<li>的独特类选择器样式，对应的 CSS 代码如示例 4 所示。

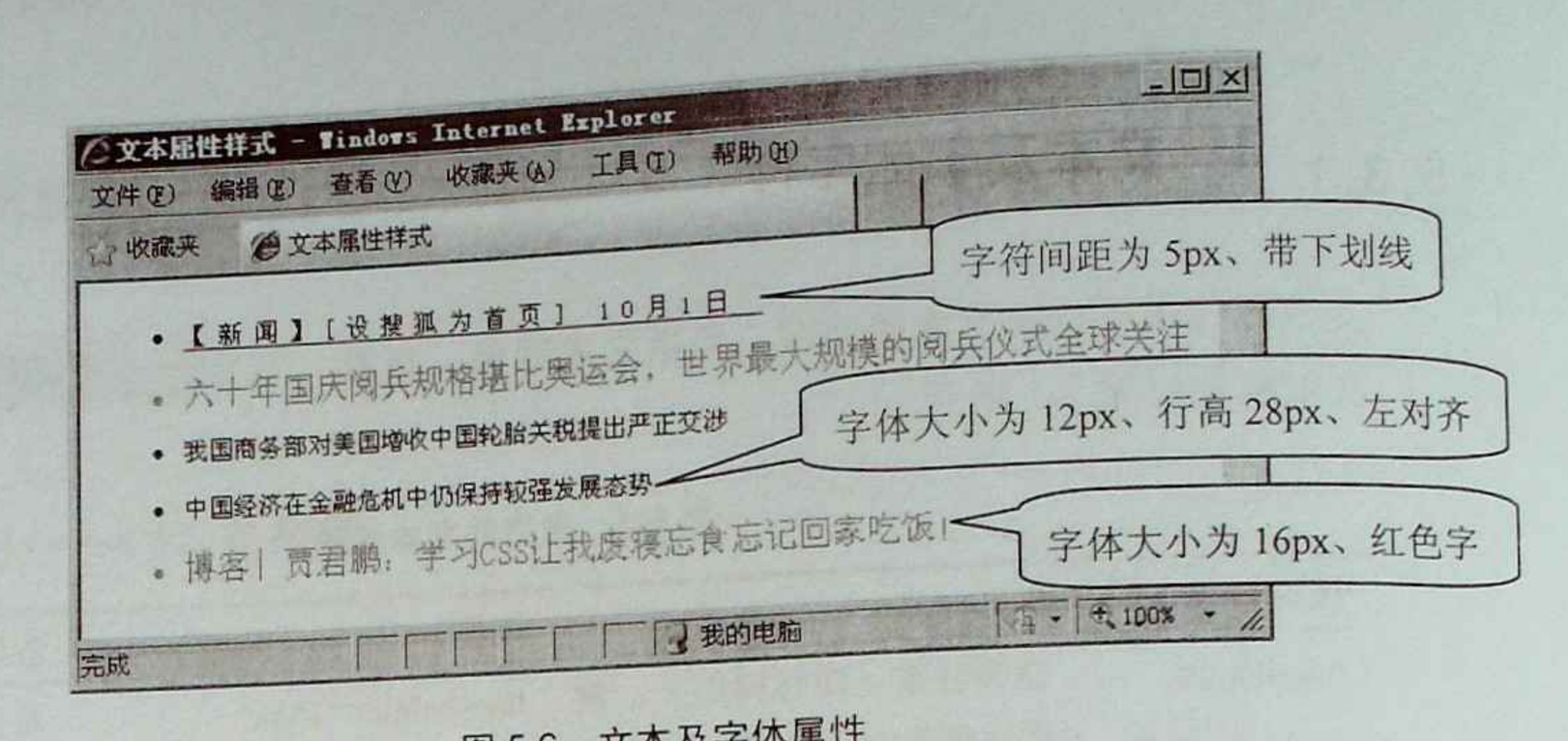


图 5.6 文本及字体属性

示例 4

```
<html>
  <head>
    <meta httli-equiv="Content-Tylie" content="text/html; charset=gb2312" />
    <title>文本属性样式</title>
    <style tylie="text/css">
    /* 所有列表项都适用的字体样式*/
    li{                          12px 宋体
      font:12px 宋体;
      text-align:left;           行高为 28px
      line-height:28px;
    }
    /* 新闻标题的类样式*/
    .title{                      字符间距为 5px
      letter-spacing:5px;
      white-space:nowrap;        实际的新闻标题可能从数据库读出，不清楚
      text-decoration:underline; 标题有多长，所以设置内容溢出不断行，确
    }                            保标题在一行显示
    /* 大字体的类样式*/
    .bigFont {                   带下划线
      font-size:16px;
      color:red;
    }
    </style>
  </head>
  <body>
    <div>
      <ul>
        <li class="title">【新闻】[设搜狐为首页] 10 月 1 日 </li>
        <li class="bigFont">六十年国庆阅兵规格堪比奥运会，世界最大规模的阅兵仪式全
      球关注</li>
        <li>我国商务部对美国增收中国轮胎关税提出严正交涉</li>
```

```
        <li>中国经济在金融危机中仍保持较强发展态势</li>
        <li class="bigFont">博客| 贾君鹏:学习 CSS 让我废寝忘食忘记回家吃饭！ </li>
      </ul>
    </div>
  </body>
</html>
```

示例 4 在浏览器的预览效果如图 5.8 所示，注意观察各种样式设置对应的具体效果。

经验总结

示例 4 中的 li>line-height 属性，可以加进 font 的缩写，其格式为“font:粗细大小/行高 字体类型;”，要求至少需要两个属性（字体类型、大小）。示例 4 的相关代码可合并为：“font:12px/28px 宋体;”。

5.3.2 背景属性

背景属性用于定义页面元素的背景色或背景图片，同时还可以精确控制背景出现的位置、平铺方向等，常用的背景属性如表 5-3 所示。

表 5-3 常用的背景属性

属性	含义	举例
background	在一个声明中设置所有的背景属性	background: #ccc url(images/bg.png) repeat-x 20px -100px
background-color	设置背景颜色	background-color: #ccc
background-image	设置背景图片	background-image: url(images/bg.png)
background-repeat	设置背景的平铺方式	background-repeat: no-repeat
background-position	设置背景出现的初始位置	background-position: 20px -100px（分别表示横向和纵向的位置坐标）

属性 background-repeat 表示背景图的重复方式，即如何平铺背景。类似计算机的桌面背景设置，又细分为横向、纵向的平铺，对应的取值如表 5-4 所示。

表 5-4 background-repeat 属性对应的取值

平铺方式	含义	应用场景
repeat	横纵向都平铺，不填时的默认值	小方块图平铺构建整体背景
repeat-x	横向平铺	细长小图实现渐变效果
repeat-y	纵向平铺	小图背景实现特殊边框
no-repeat	背景图不重复平铺	大图做背景或使用偏移量控制

表 5-4 中的四种平铺效果如图 5.7 所示。

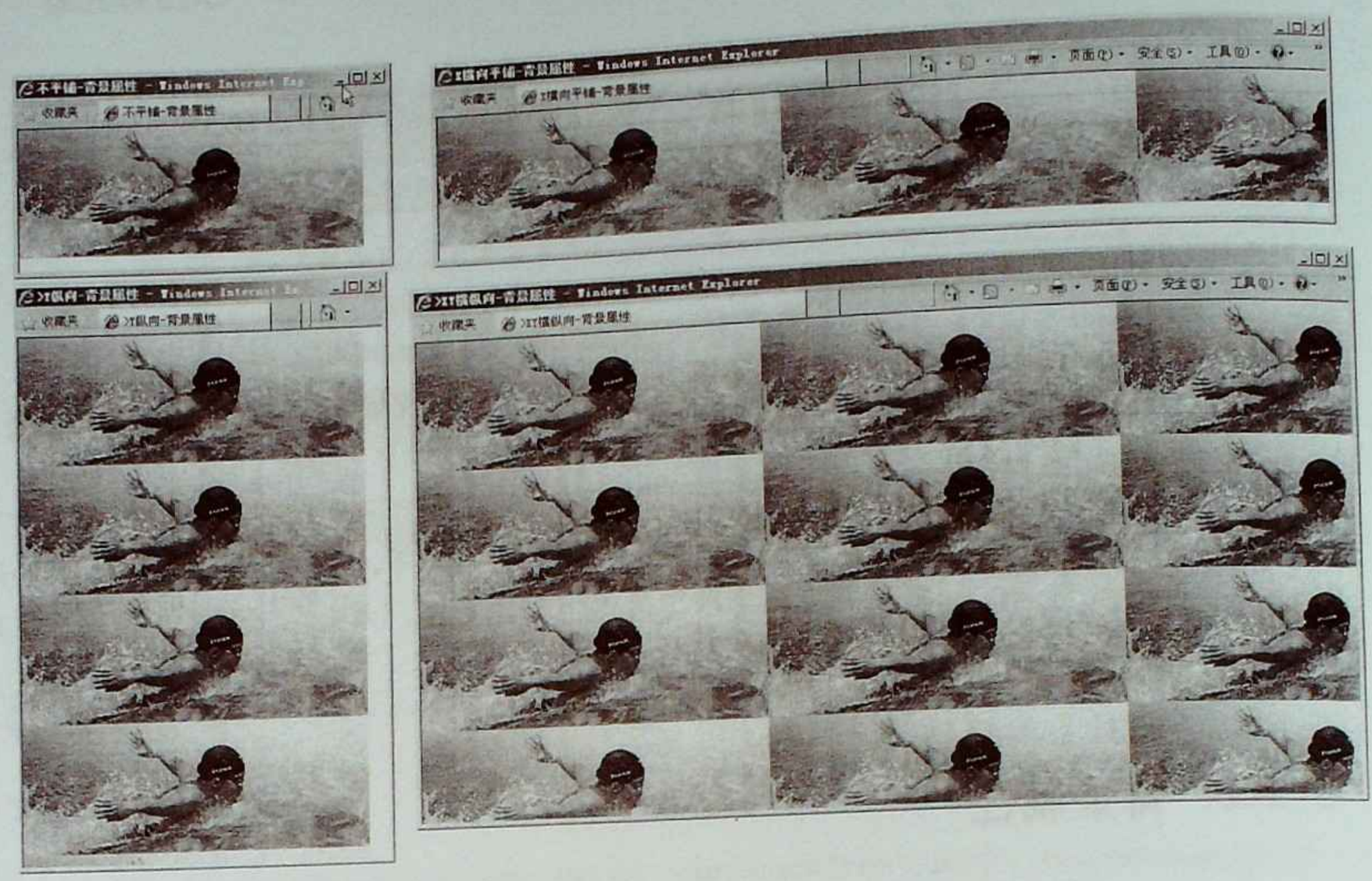


图 5.7　背景图的各种平铺方式

背景图默认从被修饰元素的左上角开始显示图像，但也可以使用 background-position 属性设置背景图出现的位置，即背景出现一定的偏移量，如图 5.8 所示。它可以使用具体数值、百分比、关键词三种方式表示水平和垂直方向的偏移量，如表 5-5 所示。

表 5-5　background-position 属性对应的取值

背景图出现的初始位置	含义	示例
Xpos Ypos	使用具体的像素值表示背景出现的位置，第一个数值表示水平位置，第二个表示垂直位置	➢ 0px 0px（默认，表示从左上角出现背景，无偏移） ➢ 30px 40px（正向偏移，图片向下和向左移动） ➢ -50px -60px（反向偏移，图片向上和向右移动）
X% Y%	使用百分比表示背景出现的位置	30% 50%（垂直方向居中，水平方向偏移 30%）
X 方向关键词 Y 方向关键词	使用关键词表示背景出现的位置，水平方向的关键词有 left、center、right，垂直方向的关键词有 top、center、bottom	可以使用水平和垂直方向的关键词进行自由组合，如省略，则默认为 center。例如： ➢ right top（右上角出现） ➢ left bottom（左下角出现） ➢ top（上方水平居中位置出现）

背景图默认以被修饰元素的左上角作为原点坐标（0px，0px），以此为基础设置背景图出现的水平方向和垂直方向的坐标。如某个方向的坐标为正，即正偏移，则背景图向右或向下偏移；相反，则出现负偏移，背景图向左或向上偏移，各种偏移效果如图 5.8 所示。

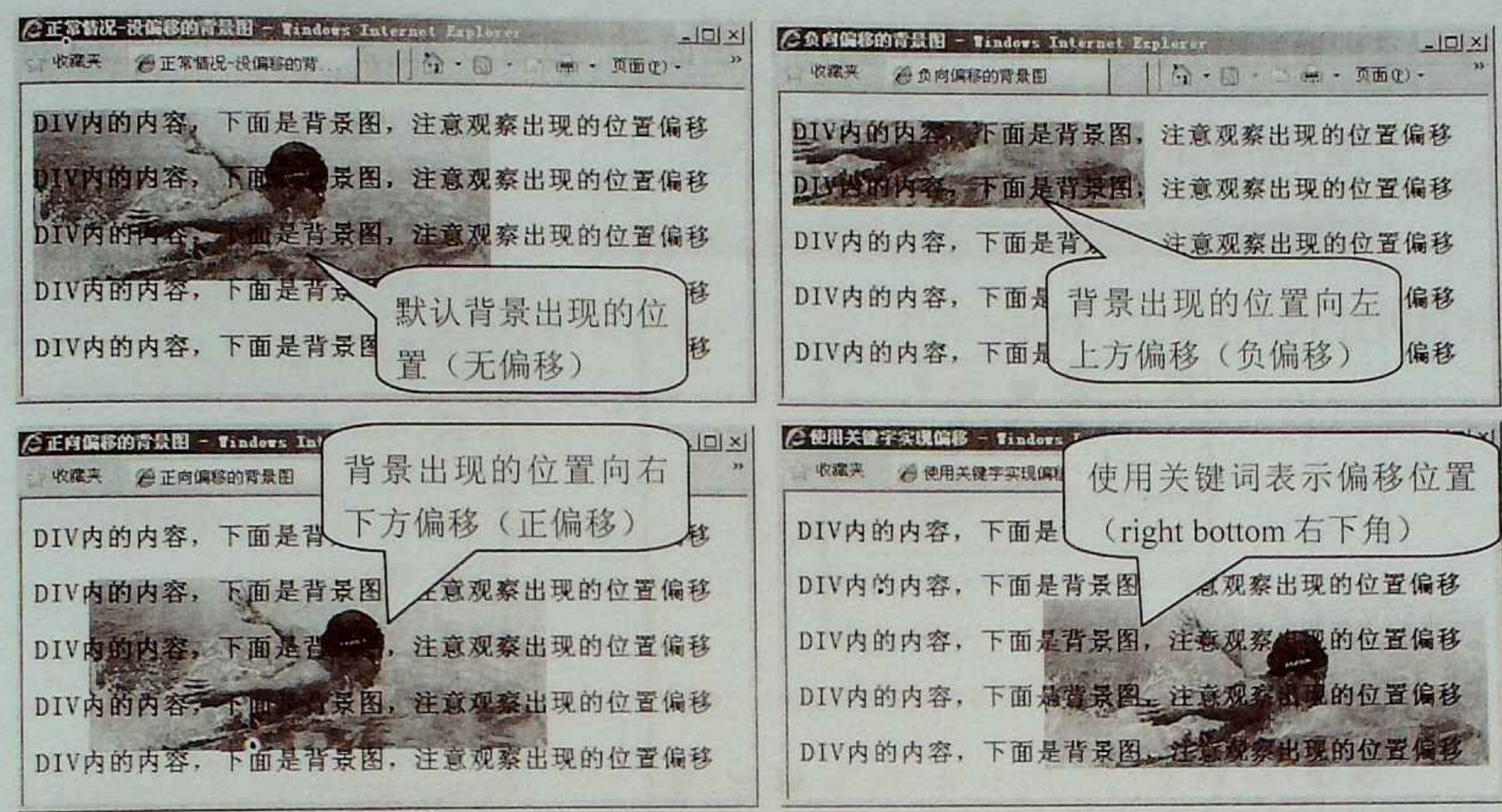


图 5.8　各类偏移效果

例如，使用数值实现图 5.8 中的负偏移效果，对应的 HTML 代码如示例 5 所示。

示例 5

```
<html>
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
   <title>正向偏移的背景图</title>
   <style type="text/css">
        /*设置 div 元素的背景图*/
       div{
        background:url(images/bg.jpg) no-repeat; /*设置背景图片、不重复平铺*/
        background-position: -70px -60px /*背景出现的水平和垂直位置坐标*/
      }
   </style>
  </head>
  <body>
    <div>
     <h3>DIV 内的内容,下面是背景图,注意观察出现的位置偏移</h3>
     <h3>DIV 内的内容,下面是背景图,注意观察出现的位置偏移</h3>
     <h3>DIV 内的内容,下面是背景图,注意观察出现的位置偏移</h3>
     <h3>DIV 内的内容,下面是背景图,注意观察出现的位置偏移</h3>
     <h3>DIV 内的内容,下面是背景图,注意观察出现的位置偏移</h3>
    </div>
   </body>
  </html>
```

网站开发中常见的应用是利用背景坐标的偏移，截取一张背景图中某部分内容。例如，为了减少客户端从服务器端下载图片的次数，提高服务器的性能，比较流行的做法是将多张图片拼合为一张大图片（如图 5.9 所示），然后再利用 background-position 属性截

取其中的各个小图，最后再显示在页面中，如图 5.10 所示。常见的应用是菜单图标的拼合和截取，这种技术网上也称为 CSS Sprite 技术，即图片的拼合和截取技术，此处我们重点讲解如何从大图中截取出各个小图。

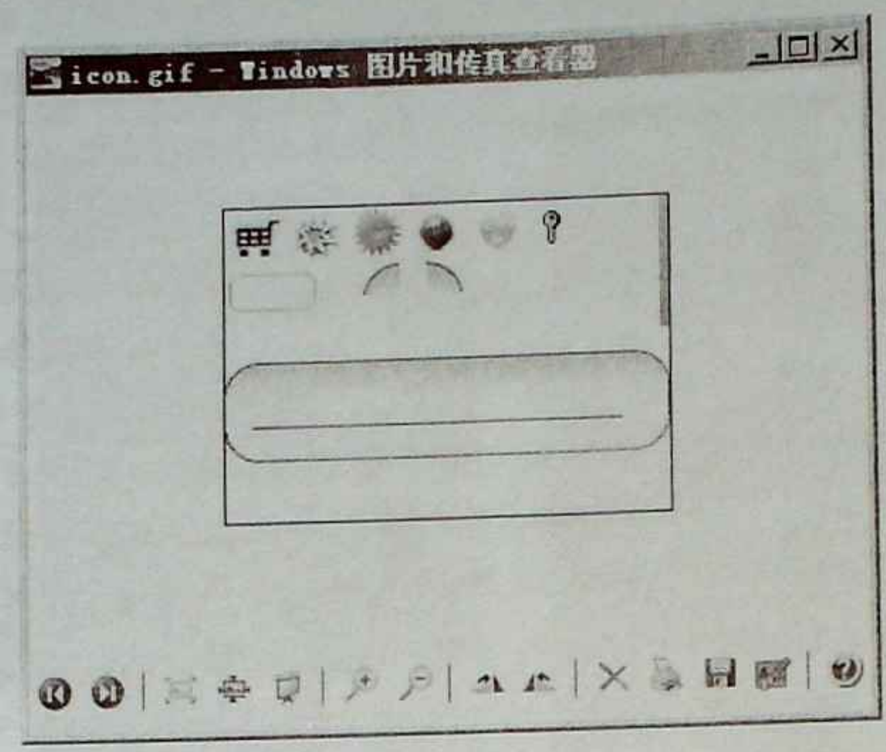


图 5.9　背景图

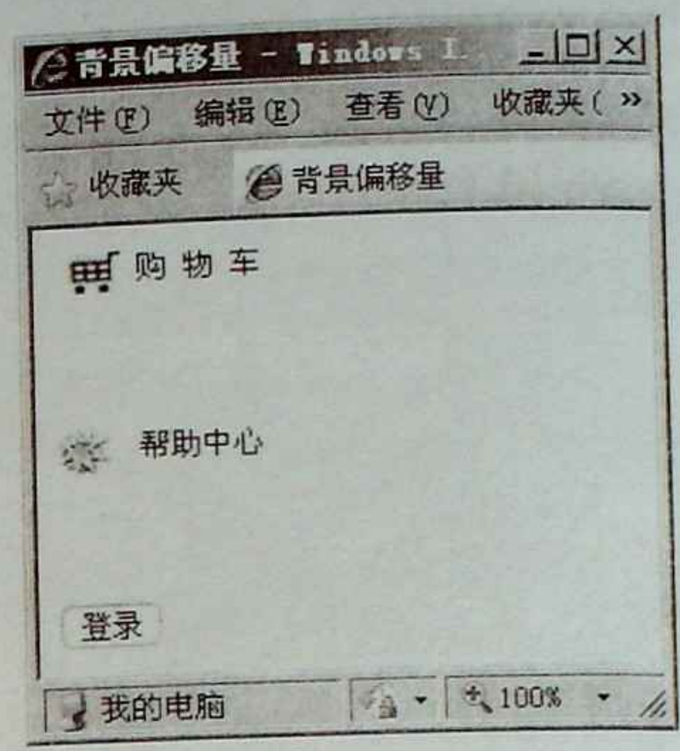


图 5.10　从同一张背景图截取的菜单图标

从一张背景图中截取各个小图标，就类似我们用放大镜看地图一样，通过左右或上下方向移动放大镜，即可获取到需要的位置，如图 5.11。而放大镜的移动，就类似我们改变背景图片出现的位置坐标一样，所以图片截取的关键是设置 background-position 的位置属性。

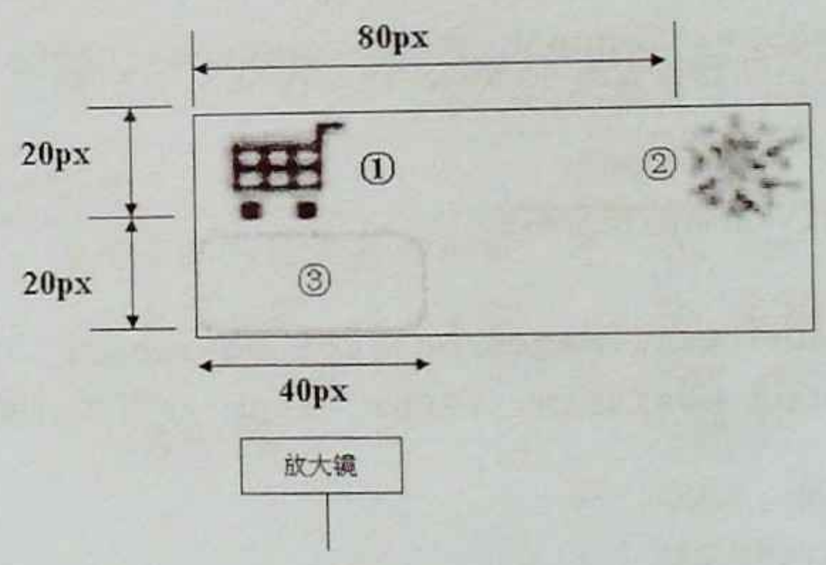


图 5.11　类似于用方形放大镜看小字体的书或地图

如何实现图 5.10 所示的页面效果？显然，实现方法是将文字内容分别放入三个 或<p>等块级标签中，然后设置该标签的背景图片，HTML 组织结构如下。

```
<div>购物车</div>
<div>帮助中心</div>
<div >登录</div>
```

但背景图片中有很多小图标，如何截取图 5.11 中的各个小图标？实现思路如下。

(1) 设置上述三个标签为同一背景："background:url(images/icon.gif) no-repeat;}"。

(2) 截取小图 1（购物车图标）作为第一个标签的背景图标。如图 5.11 所示，该图标位于图片的左上方，无偏移量。

(3) 放置小图 2（帮助中心图标）。如图 5.11 所示，纵向无偏移，横向需向左移动，所以为-80px，对应的 CSS 设置为："background-position:-80px 0px;"。

(4) 放置小图 3（登录图标）。如图 5.11 所示，横向无偏移，纵向需向上移动，所以为-20px，对应的 CSS 设置为："background-position:0px -20px;"。

另外，为了美观起见，还需设置 div 宽度、行高、对齐方式等，完整的 CSS 代码如示例 6 所示。

示例 6

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
    <title>背景偏移量</title>
    <style>
      /*设置三个 div 标签共同的样式属性*/
      div{
      width:80px;  /*设定宽度很重要,决定了截取图标范围的大小*/
      font:12px/20px 宋体;  /*设置行高,方便垂直对齐,div 占据一定高度*/
      text-align:right;
      background:url(images/icon.gif) no-repeat; /*设置同一背景*/
      }

     /*利用类样式添加"帮助中心"图标 */
      .help{
        background-position:-80px 0px; /*此处负值向左偏移*/
        }

     /*利用类样式添加"登录"图标 */
      .login{
        width:40px;
        background-position:0px -20px;  /* 此处负值表示向上偏移*/
         text-align:center;
    }
    </style>
  </head>
  <body>
      <div>购 物 车</div>
      <p> </p>
      <div class="help">帮助中心</div>
      <p> </p>
      <div class="login">登录</div>
    </body>
  </html>
```

5.3.3 列表的常用属性

常见的各类商品分类列表或导航菜单一般都使用 ul-li 结构实现，如图 5.12 所示。和实际应用的导航菜单（图 5.13）相比，样式方面比较难看，应去掉列表项（<li>）默认的圆点符号，并且需要将排列方式改为横向排列。如何实现这样的样式效果呢？答案就是使用列表<li>的两个常用属性：list-style 和 float 属性。

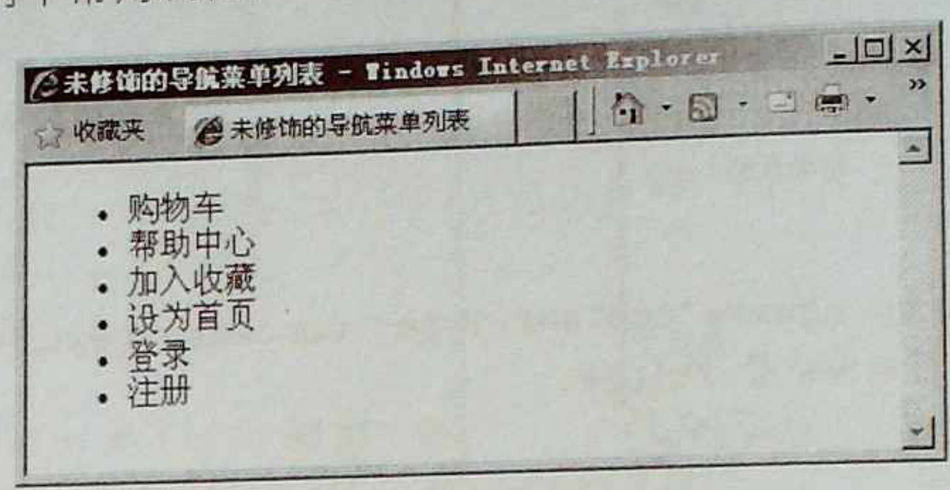


图 5.12 未修饰的导航菜单

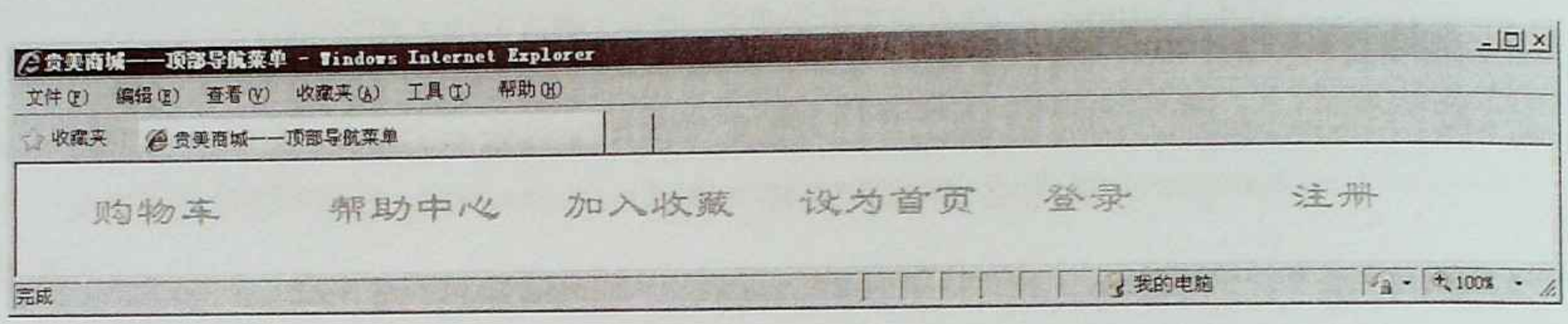


图 5.13 修饰后的导航菜单

1. list-style

list-style 属性用于定义列表项的各类风格，例如，定义列表的类型图标等，如表 5-5 所示。虽然可以利用表 5-5 定义各种类型的图标，但最常用的还是不使用任何图标，将"list-style"设为"none"。

表 5-5 常用的列表类型图标

属性值	方式	语法实现（常用缩写）	示例
none	去掉修饰符号	list-style:none	刷牙 洗脸
disc	实心圆（ 默认类型）	list-style:disc	● 刷牙 ● 洗脸
circle	空心圆	list-style:circle	○ 刷牙 ○ 洗脸
square	实心正方形	list-style:square	■ 刷牙 ■ 洗脸
decimal	数字（ 默认类型）	list-style:decimal	1. 刷牙 2. 洗脸

2. float

float 属性用于定义元素的浮动方向，它实际上不是列表<li>具有的独特属性，而是所有元素都支持的 CSS 属性，它可以改变块级元素默认的换行显示方式，即显示时不再“换行”。此处仅用于将纵向排列的列表项改为横向排列，对应的 CSS 样式应设置为左浮动“float：left”，表示列表项都向左浮动，从而实现横向排列的效果。其具体用法和含义将在后续章节详细讲解。

除此外，确保各列表项之间的间隔，还需使用 width 属性设置宽度，和 float 属性一样，它实际上不是<li>具有的特殊属性，而是所有元素支持的 CSS 属性，后续章节将会讲解。图 5.13 对应的完整 CSS 代码如示例 7 所示。

示例 7

```
<html>
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=gb2312"  />
   <title>贵美商城——顶部导航菜单</title>
   <style>
    /*设置菜单列表项<li>的样式*/
     li{
       width:150px;  /*设置宽度确保各列表项之间的间距*?
       color:red;
       font:28px 隶书;
       list-style:none;   /*无列表类型图标*/
       float:left;  /*设置左浮动,是列表改为横向排列*/
     }
   </style>
  </head>
  <body>
    <div>
     <ul>
          <li>购物车</li>
          <li>帮助中心</li>
          <li>加入收藏</li>
          <li>设为首页</li>
          <li>登录</li>
          <li>注册</li>
       </ul>
     </div>
   </body>
</html>
```

5.3.4 上机练习

上机练习 2

练习——修饰贵美商城网站商品分类页

需求说明

在上机练习 1 的基础上，修饰商品分类页，效果如图 5.14 所示，具体要求如下。

- 欢迎词的字符间距为 5px 并带下划线。
- 商品分类用提供的背景，家用电器、日用百货为 14px 字体醒目显示，表示商品分类。
- 各商品类别下的商品名称用 12px 字体显示，各列表名称间有一定的间隔。

提示

- 整个 div 设置背景样式，定义一定的宽度和高度，参考背景图大小进行调整。
- 顶边距（“商品分类”背景处）可以添加一个空<ul>并设置高度为 15px。
- 设置商品类别下的各列表项左浮动、无列表符号，行高 25px，定义适当宽度以分隔各列表项。

图 5.14 贵美商城商品分类

上机练习 3

练习——实现圆角矩形

需求说明

使用提供的背景素材（图 5.15）实现如图 5.16 所示的圆角矩形效果。

提示

- 先分顶部、中部和底部，对应为三种类样式，如图 5.15 所示，底部使用空的<ul>占位。
- 参考图 5.16 的尺寸标注，从背景素材中截取相应的部分作为背景，注意调整行高以适应布局。

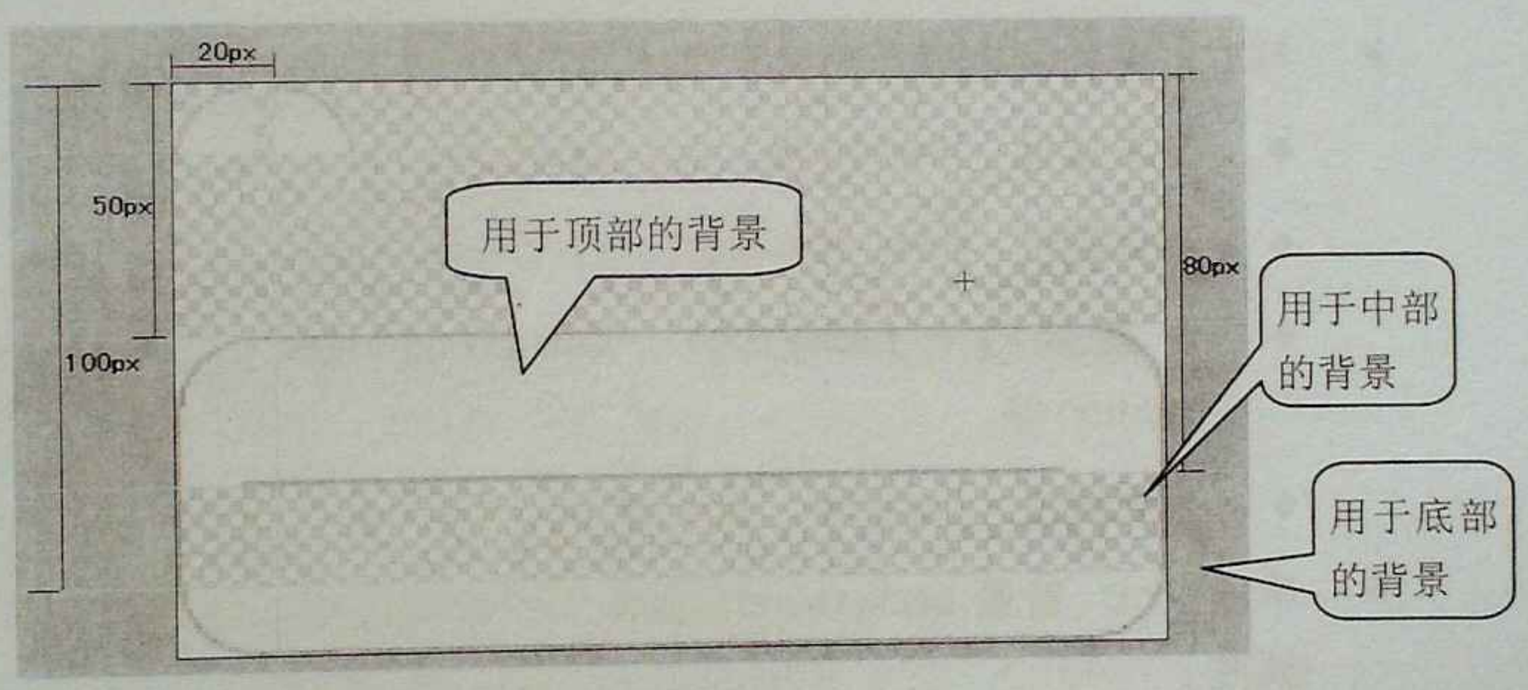


图 5.15　提供的背景图片

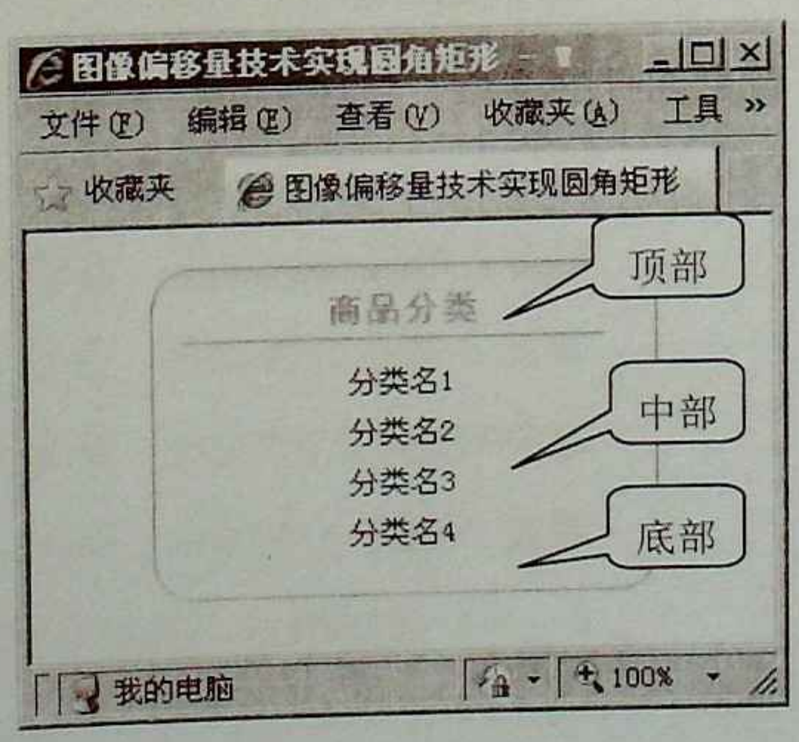


图 5.16　图像偏移量技术的简单应用

本章总结

- 使用 CSS 可以实现 W3C 提倡的结构和样式分离的思想，这样做的好处包括：利于团队开发，精确控制页面，提高开发效率，利用搜索引擎搜索等。
- CSS 样式规则采用选择器、属性、属性值进行描述。
- 选择器即为被修饰的对象，常用的三类选择器有以下三类。
  - 标签选择器，直接用标签名方式定义。
  - 类选择器，先为标签设置属性 class="类别名"，后用.（英文句号）+类别名方式定义。
  - ID 选择器，先为标签设置属性 id="id 名"，后用#+类别名方式定义。
- 样式的两大用途是页面元素修饰和布局，页面元素修饰的 CSS 属性包括如下四类。
  - 文本属性：line-height、text-decoration、text-align、color、letter-spacing 等。
  - 字体属性：font、font-size、font-weight、font-size 等。
  - 背景属性：background、background-image、background-repeat、background-position 等。
  - 列表属性：list-style、float 等。
- 字体 font、背景 background、列表 list 都可以采用缩写形式，依次声明完所有相关属性。
- 网上流行的 CSS Sprite 图片拼合和截取技术，实现的关键是利用 background-position 属性实现背景图标的截取。该属性可以采用数值、百分比、关键词设置图片出现的位置。

本章作业

一、选择题

1. 以下是文本属性的是（　　）

A. font-size

B. line-height

C. background

D. text-align

2. 要实现无列表符号、横向排列的导航菜单，应使用（　　）CSS 属性。

A. list-style

B. lineheight

C. float

D. background-repeat

3. 实现背景横向平铺的效果，对应的 CSS 为（　　）。

A. div{background-image:url(images/bg.gif);}

B. div{background:url(images/bg.gif) repeat-x;}

C. div{background:url(images/bg.gif) repeat-y;}

D. div{background:url(images/bg.gif) no-repeat;}

4. 关于 CSS 代码："background:url(images/bg.gif) no-repeat -5px 6px;"，说法正确的是（　　）。

A. 背景图向下偏移 5 像素，同时向左偏移 6 像素

B. 背景图向上偏移 5 像素，同时向右偏移 6 像素

C. 背景图向左偏移 5 像素，同时向下偏移 6 像素

D. 背景图向左偏移 5 像素，同时向上偏移 6 像素

5. 以下关于 CSS 属性的缩写形式，正确的是（　　）。

A. font:12px/28px 宋体

B. font:12px 宋体 28px bold

C. background: url(images/bg.png) repeat-x 20px -100px

D. background: repeat-x 20px -100px

二、简答题

1. 请简述 CSS 的特点及优势。

2. 选择器的主要作用是什么？都有哪些选择器，分别适用于什么场景？

3. 实现表格的背景、字体、颜色等方面的修饰，要求效果如图 5.17 所示。

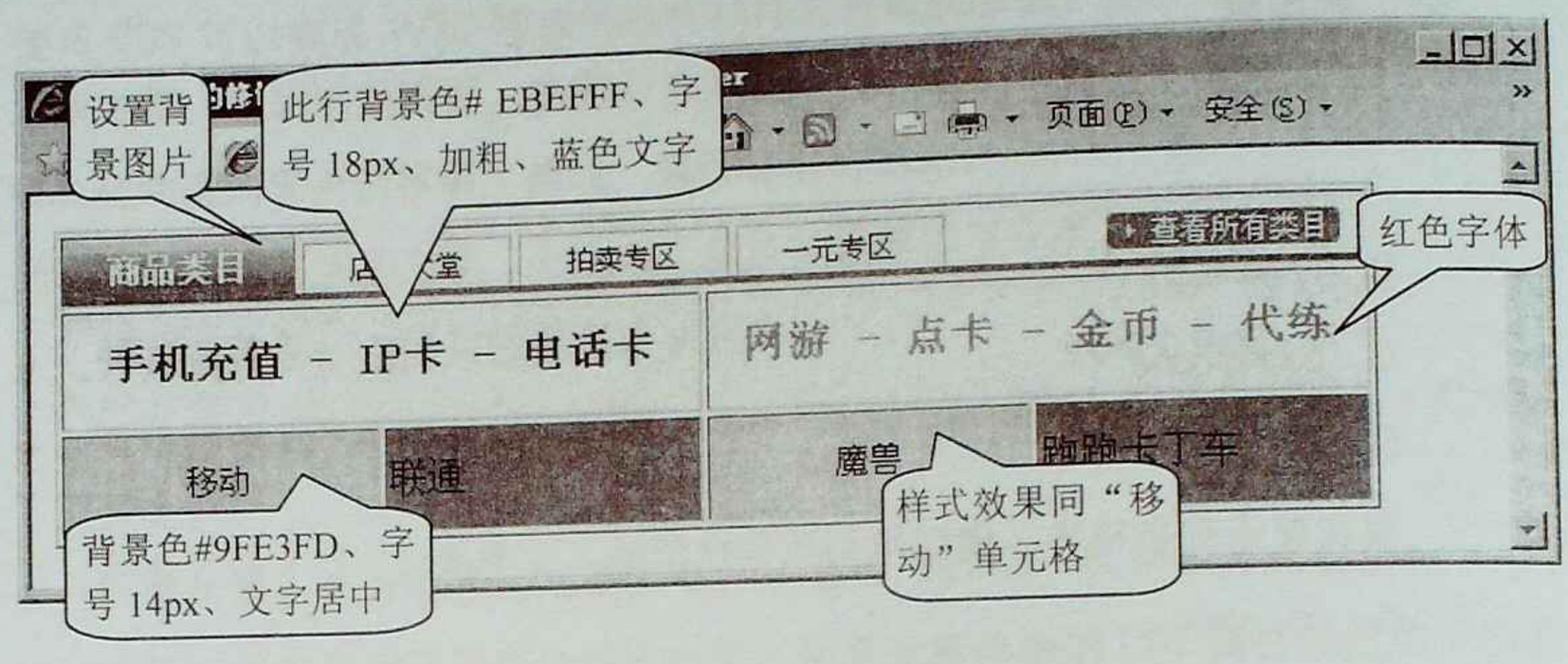


图 5.17　表格修饰

4. 利用背景图的截取功能，实现 Google 首页下方的列表图效果，如图 5.18 所示，背景素材如图 5.19 所示。

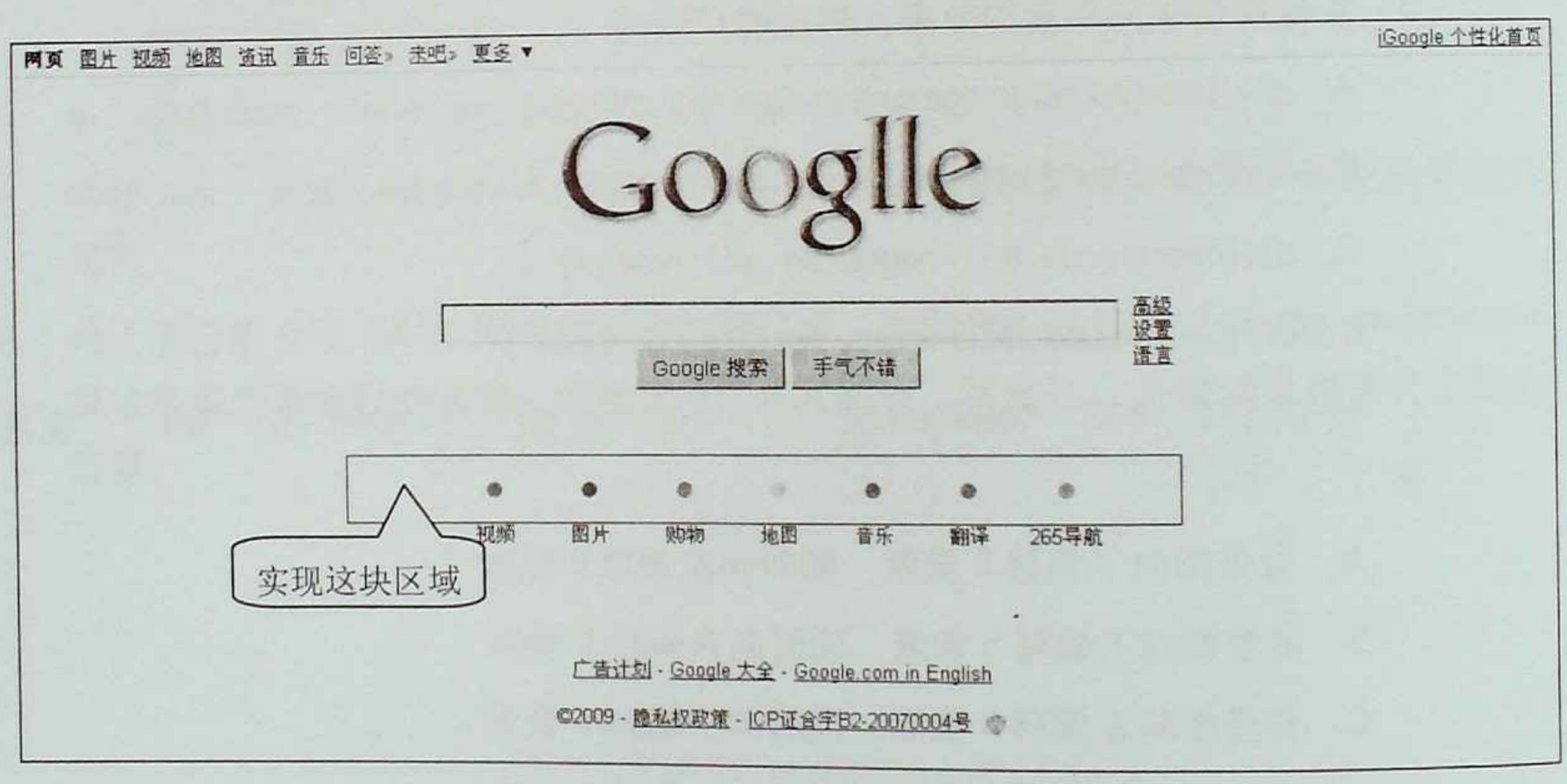


图 5.18　Google 首页

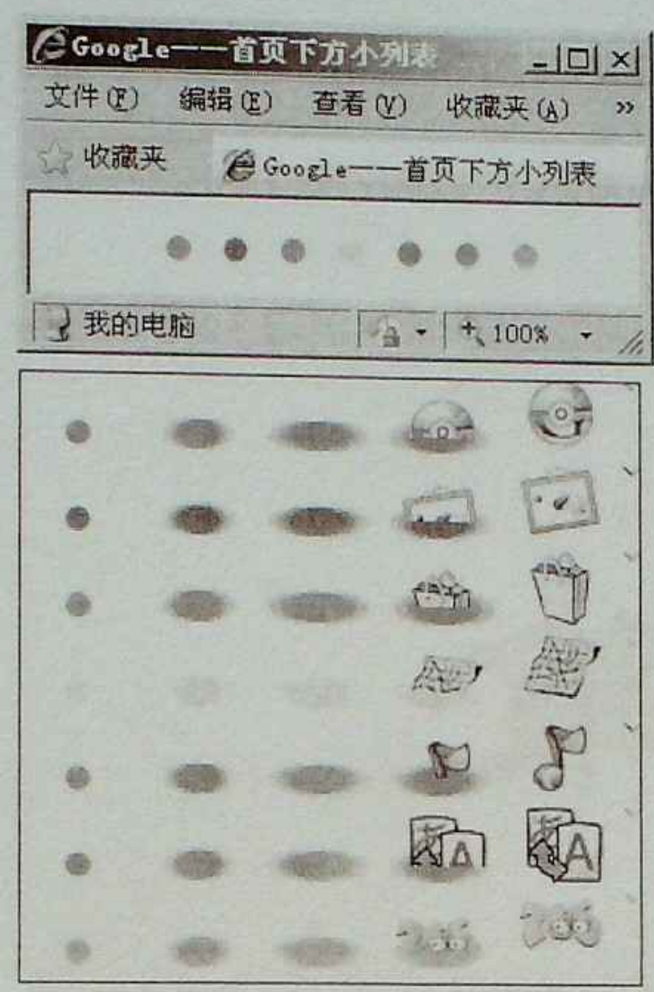


图 5.19 实现效果及背景素材

5. 自学教员提供的“学习参考资料”，利用背景图的截取功能，实现圆角矩形效果，如图 5.20 所示。

图 5.20 圆角矩形效果

第6章

CSS 样式表（二）

◈本章工作任务

- 完成贵美商城网站登录页的布局
- 实现贵美商城网站登录页顶部的局部布局

◈本章技能目标

- 使用盒子模型的相关属性实现页面布局
- 掌握普通文档流、浮动与清除属性的含义

本章单词

请在预习时学会下列单词的含义和发音，并填写在横线处。

1. solid：_____
2. margin：_____
3. padding：_____
4. header：_____
5. content：_____
6. footer：_____
7. main：_____
8. index：_____
9. container：_____
10. auto：_____

本章简介

在上一章，我们学习了 CSS 语法以及常见的 CSS 样式属性。前面曾提到过，CSS 样式的两大用途分别是页面元素修饰和页面布局，从本章开始将讲解如何实现页面布局。本章将围绕页面布局，依次介绍盒子模型及相关属性、布局应用、浮动属性和清除属性。本章的重点是理解盒子模型及相关的盒子属性，因为它是页面布局的基础。

6.1 盒子模型及相关属性

盒子模型（Box Model）是实现页面布局的基础，学习页面的布局必须先理解盒子模型。

6.1.1 盒子模型

盒子的概念在我们生活中并不陌生，例如礼品的包装盒，如图 6.1 所示，礼品是最终运输的物品，四周一般会添加用于抗震的填充材料，再外面是包装纸壳。

图 6.1　生活中的盒子模型

CSS 中盒子模型的概念与此类似，CSS 将网页中所有元素都看成一个个盒子。例如，网页中显示的一幅图片如图 6.2 所示，背后实际对应着一个盒子模型结构，它包括如下属性。

- 边框（border）：对应包装盒的纸壳，它一般具有一定的厚度。
- 内边距（padding）：位于边框内部，是内容与边框的距离，对应包装盒的填充部分，所以有的教材也称其为“填充”。
- 外边距（margin）：位于边框外部，是边框外面周围的间隙，所以有的教材也称其为“边界”。

图 6.2　网页元素对应的盒子模型

注意

- 因为盒子是矩形结构，所以边框、内边距、外边距这些属性都分别对应上（top）、下（bottom）、左（left）、右（right）四个边，这四个边的设置可以不同。图 6.2 只给出了外边距的完整标注，边框、内边距同理。
- 页面元素图片实际占据的空间，即实际占位是图 6.2 中的虚线区域。

除边框、内边距、外边距之外，还应包括元素内容本身，所以完整的盒子模型的平面结构图如图 6.3 所示。

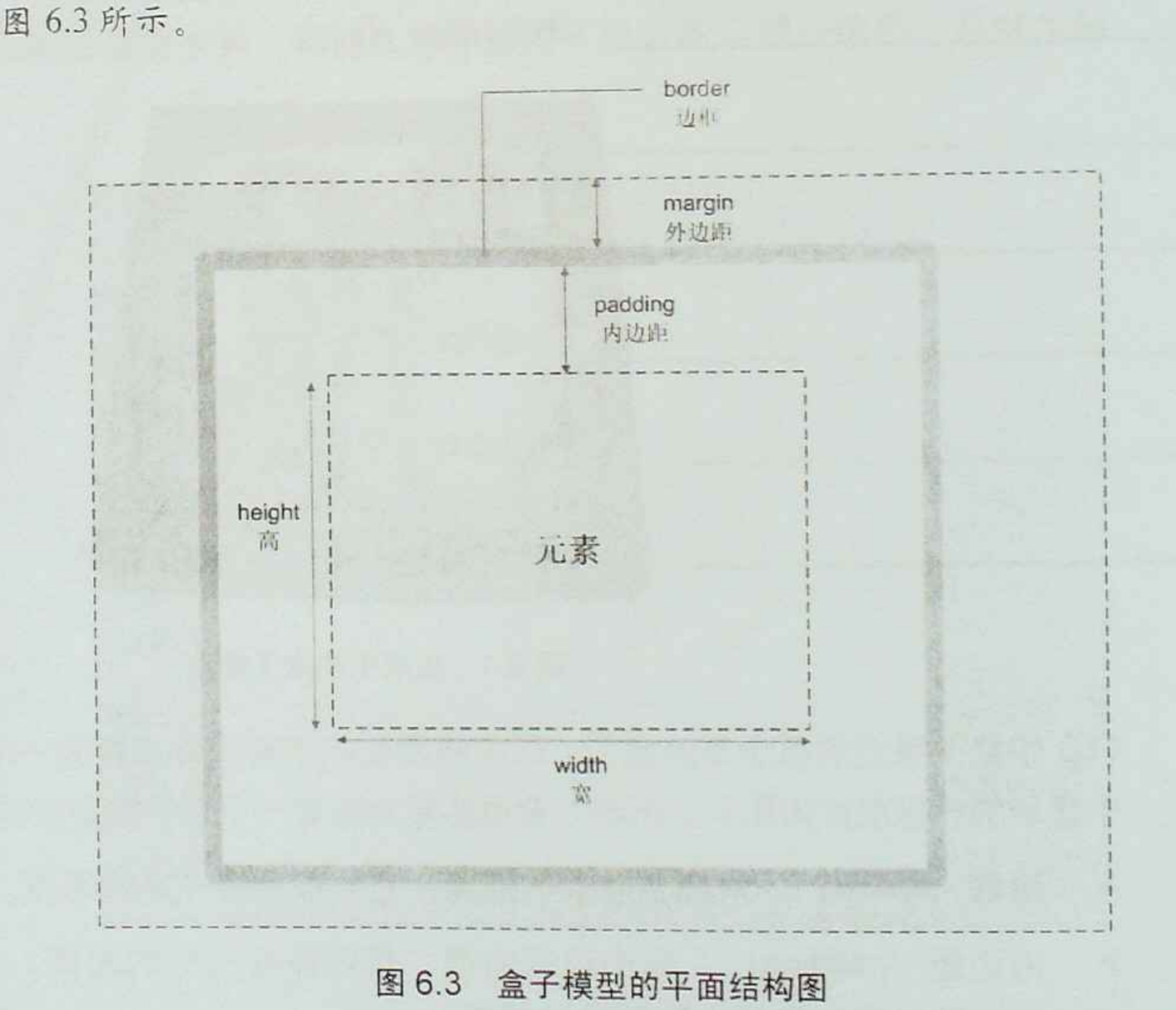


图 6.3　盒子模型的平面结构图

除平面结构图外，盒子模型还包括三维的立体结构图，如图 6.4 所示，从上往下看，它表示的层次关系依次如下。

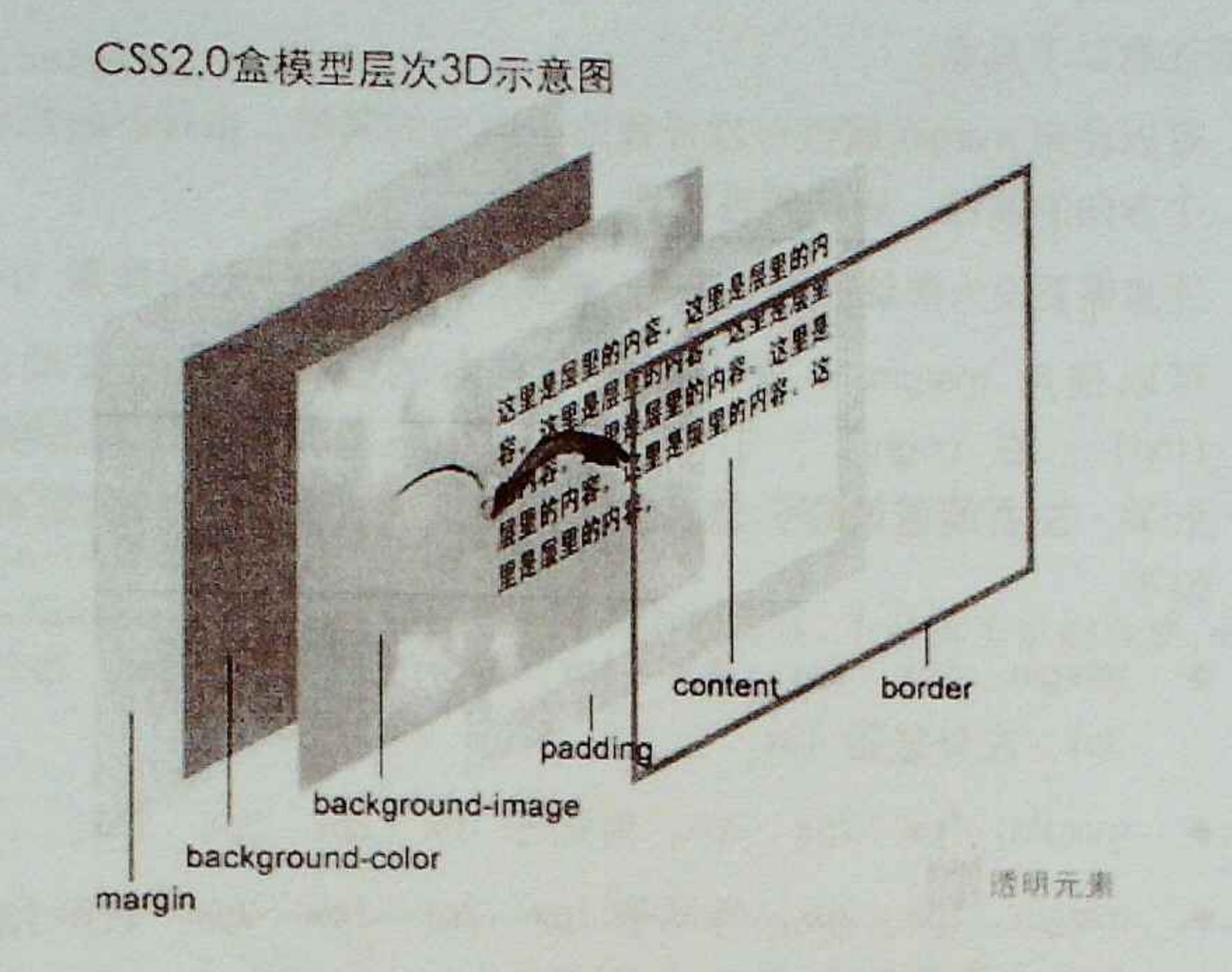


图 6.4　盒子模型的三维立体层次结构图

- 首先是盒子的主要标识：边框（border），位于盒子的第一层。
- 其次是元素内容（content）、外边距（padding），两者同位于第二层。
- 再次是上章着重讲解的背景图（background-image），位于第三层。
- 背景色（background-color）位于第四层。
- 最后是整个盒子的外边距（margin）。

我们在网页中看到的页面内容，即为图 6.4 多层叠加的最终效果。从这里可以看出，若对某个页面元素同时设置背景图和背景色，则背景图将在背景色的上方显示。

6.1.2　盒子属性

盒子模型中主要包括外边距、边框、内边距以及元素内容的宽高，前三个属性一般称为盒子属性，下面具体介绍。

1．margin（外边距）

margin（外边距）位于盒子边框外，指与其他盒子之间的距离。外边距根据上、下、左、右四个方向，可细分为上外边距、下外边距、左外边距、右外边距，具体属性如表 6-1 所示。

表 6-1　外边距的属性

类别	属性	含义	举例
四个方向	margin-top	上外边距	margin-top: 1 px
	margin-right	右外边距	margin-right : 2 px
	margin-bottom	下外边距	margin-bottom : 2 px
	margin-left	左外边距	margin-left : 1 px
缩写形式	margin	在一个声明中统一设置四个方向的外边距	1px 2px 3px 4px

需要注意以下几点。

- 可以使用 margin 属性一次设置四个方向的属性，也可以分别设置上、下、左、右四个方向的属性，后续属性同理。
- 注意需要设为带单位的长度值，常用的长度单位一般是像素（px），后续属性同理。
- 可以使用 margin 一次设置四个方向的值，但必须按顺时针方向依次代表上（top）、右（right）、下（bottom）、左（left）四个方向的属性值，如省略则按上下、左右同值处理，这些规则同样适用于后续讲解的边框、内边距，举例说明如下。
  - margin：1px 2px 3px 4px，表示上外边距 1px、右外边距 2px、下外边距 3px、左外边距 4px。
  - margin：1px 2px 3px，等同于 1px 2px 3px 2px。
  - margin：1px 2px，等同于 1px 2px 1px 2px，表示上下外边距各为 1px，左右外边距各为 2px。
  - margin：1px，等同于 1px 1px 1px 1px，表示四个边都为 1px。
  - 特殊设置：可以设置水平为 auto，表示让浏览器计算外边距，一般表现为水平居中效果。例如 margin：0px auto 表示在父级元素容器中水平居中（上、下外边距为 0px，左、右外边距自动计算）。

例如，利用 margin 属性实现某个段落的缩进以及位置的居中，如图 6.5 所示，对应的 CSS 代码如示例 1 所示。

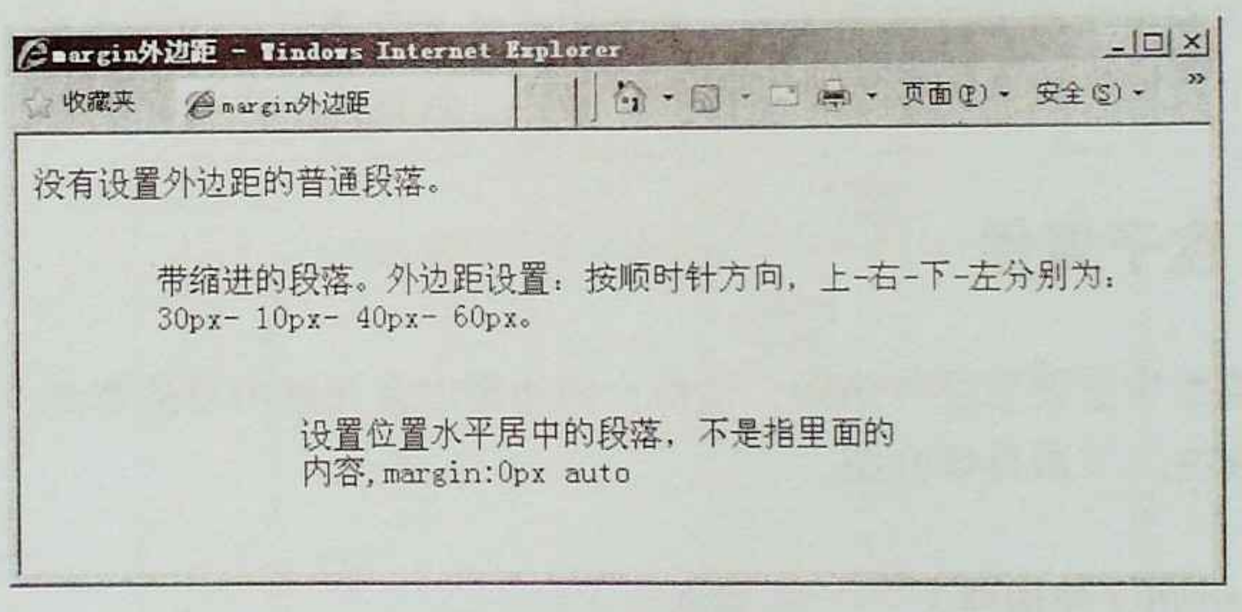


图 6.5 带缩进的段落

示例 1

```
<html >
  <head>
    <title>margin 外边距</title>
    <style type="text/css">
      .margin {
        width:500px;
        margin:30px 10px 40px 60px
        }
      .automargin {
```

```
      width:300px;
      margin:0px auto
     }
  </style>
</head>
<body>
  <p>没有设置外边距的普通段落。</p>
  <p class="margin">带缩进的段落。外边距设置：按顺时针方向，上-右-下-左分别为：
  30px- 10px- 40px- 60px。</p>
  <p class="automargin">设置位置水平居中的段落，不是指里面的内容，margin:0px
  auto</p>
 </body>
</html>
```

经验总结

水平居中包括块级元素的位置居中以及块级元素内的文字等内容居中。

- 块级元素的位置居中：margin 作为盒子间的距离，在布局中，扮演着较为重要的角色。典型的应用为布局中块级元素的水平居中，如“margin:0px auto;”。
- 文字内容的水平居中：例如，段落<p>内的文字对齐等设置，则设置块级元素的“text-align:center;”属性实现水平居中。

对于垂直居中，常见的是单行文字的垂直居中问题，可以设置文字所在行的高度 height 与文字行高属性 line-height 一致。例如，要实现图 6.6 所示的文字垂直居中效果，对应的 CSS 关键代码如下。

```
div{ width:400px; height:400px; line-height:400px;……}
```

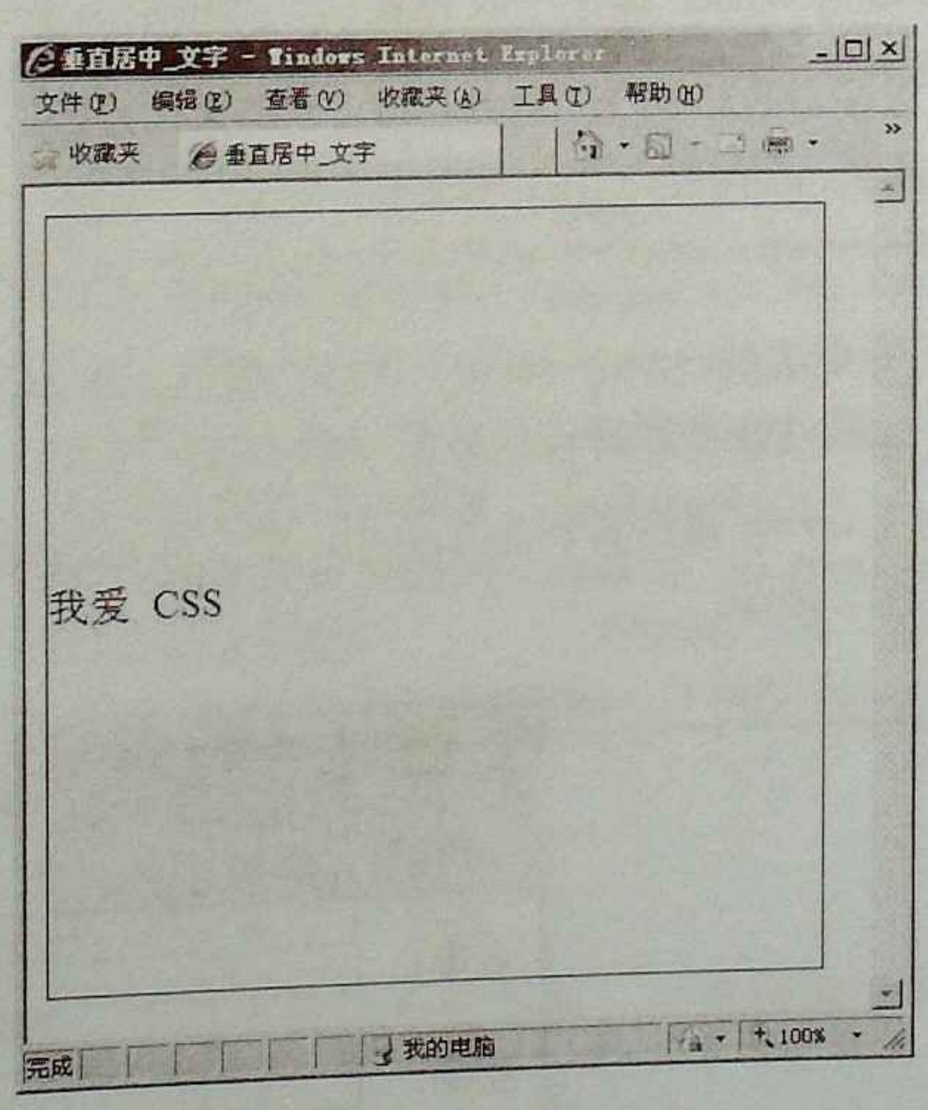


图 6.6　文字的垂直居中

2．border（边框）

边框比外边界稍复杂些，除四个边外，边框又细分为边框颜色、边框宽度、边框样式（实线、虚线等）三方面的属性，完整的属性列表如表 6-2 所示。

表 6-2　边框属性

类别	属性	含义	举例
修饰属性	border-color	边框颜色	border-color：#FF00FF
	border-width	边框宽度	border-width：2px
	border-style	边框样式，常用的取值如下。 ➢ none；默认值，无边框 ➢ solid；实线 ➢ dashed；虚线	border-style：solid
四个方向	border-top	上边框	border-top：5px
	border-right	右边框	border-right：5px solid blue
	border-bottom	下边框	border-bottom：5px dotted
	border-left	左边框	border-left：5px
缩写形式	border	在一个声明中统一设置四个方向的边框属性	border：1px solid red（设置四个方向的边框均为 1 像素、红色、实线）
	border-left 等	在一个声明中设置左边框的属性	border-left：1px solid red（设置左边框为 1 像素、红色、实线）

需要注意以下两点。

➢ 可以使用 border 属性一次设置四个边的所有属性，包括 style、width、color，也可以使用 border-top、border-right 等单独设置各边的属性，缩写时应注意格式及顺序。

```
border: border-width border-style color  /*一次设置 4 个边,注意顺序*/
border-top: border-width border-style color   /*设置上边框,其他边同理*/
```

➢ 对于每个边的 style、color、width 属性，可以一次性设置，也可以按四边分别设置，但要注意按顺时针的方向（即上-右-下-左），规则同 margin。

例如，利用 border 属性实现表单中 1px 的实线细边框效果，如图 6.7 所示，对应的 CSS 代码如示例 2 所示。

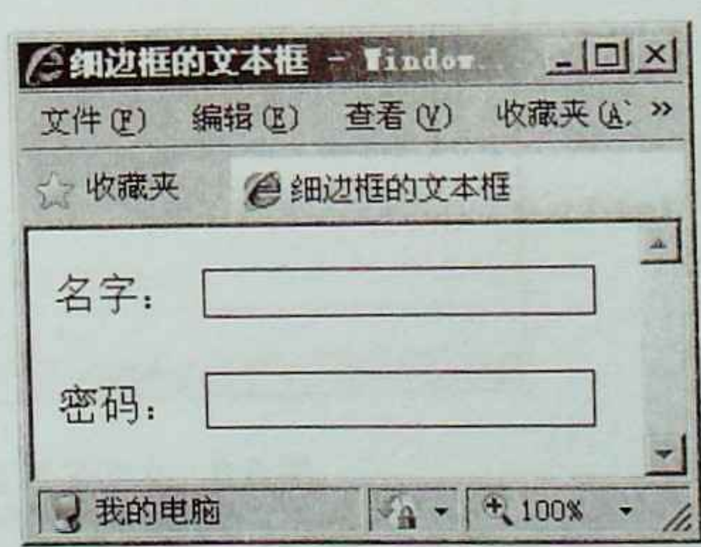


图 6.7　表单细边框样式

示例 2

```
<html>
  <head>
    <title>细边框的文本框</title>
    <style type="text/css">
     .textBorder{
      border-width:1px;
      border-style:solid;
    }
    </style>
  </head>
  <body>
    <form action="" method="post">
      <p>名字:<input name="fname" type="text" class="textBorder" /></p>
      <p>密码:<input name="pass" type="password" class="textBorder" size="21"
        /></p>
    </form>
  </body>
</html>
```


3. padding（内边距）

边框确定以后，一般需设置边框与内容之间的距离，以便精确控制内容在盒子中的位置。与 border 最明显的区别在于：padding 并非实体，而是透明留白，所以并没有修饰属性。表 6-3 列出了内边距的常用属性。

表 6-3　内边距属性

类别	属性	含义	举例
四个方向	padding-top	内容与上边框之间的距离	padding-top：5px
	padding-right	内容与右边框之间的距离	padding-right：5px
	padding-bottom	内容与下边框之间的距离	padding-bottom：5px
	padding-left	内容与左边框之间的距离	padding-left：5px
缩写形式	padding	在一个声明中统一设置四个方向的填充属性	padding：5px 10px 20px 40px（按顺时针方向，上填充 5px，右填充 10px，下填充 20px，左填充 40px）

需要注意以下几点。

- padding 值不允许为负值。
- padding 值可以一次设置，也可以四边分别设置，注意按顺时针方向，规则同 margin。

例如，利用 padding 实现表格中常见的填充效果，如图 6.8 所示，确保单元格内容和边框有一定的间隙，对应的 CSS 代码如示例 3 所示。

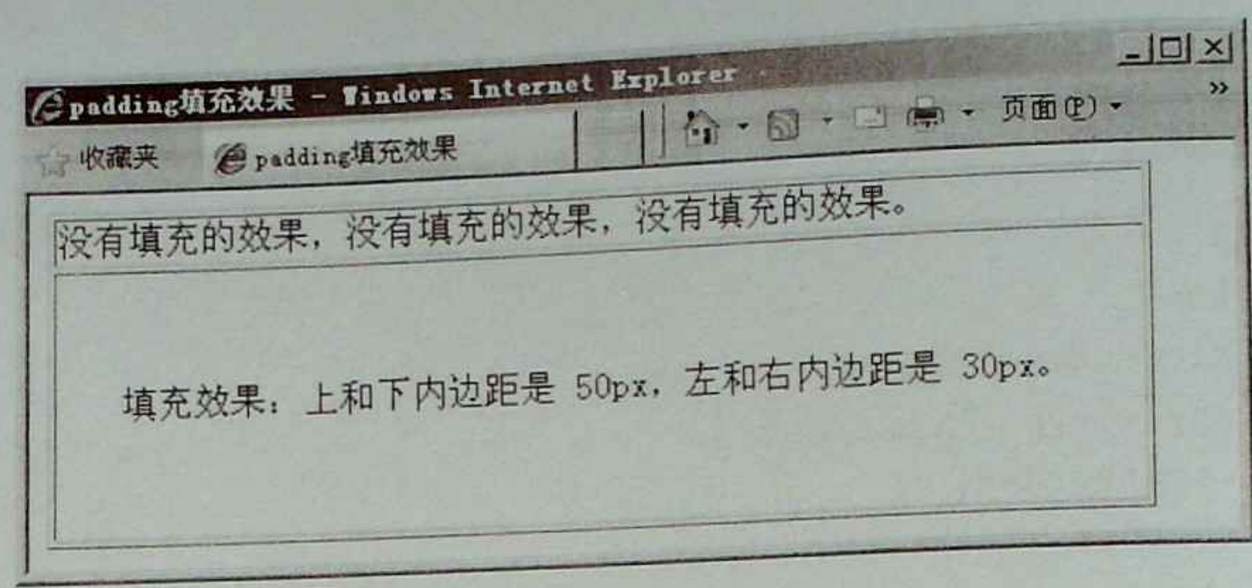


图 6.8　表格的填充效果

示例 3

```
<html >
  <head>
    <title>padding 填充效果</title>
    <style type="text/css">
     .padding{padding: 50px 30px}
    </style>
  </head>
  <body>
    <table border="1">
      <tr>
        <td>没有填充的效果,没有填充的效果,没有填充的效果。</td>
      </tr>
      <tr>
        <td class="padding">填充效果:上和下内边距是 50px,左和右内边距是 30px。
      </td>
      </tr>
    </table>
  </body>
</html>
```

学习了盒子模型后，综合盒子属性实现图 6.2 所示的效果，对应的代码如示例 4 所示。

示例 4

```
<html>
  <head>
     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
     <title>盒子模型</title>
   <style>
     #logo{
      width:128px;
      border:15px solid #999;  /*边框宽度 15px、实线、边框色浅灰色*/
      padding:20px 20px 20px 20px;   /*内边距:上、右、下、左均为 20px,可缩写为
                                       20px*/
      margin:40px auto;  /*水平居中,上下外边距各为 40px*/
      }
```

```
    </style>
  </head>
  <body>
      logo 图片上方的内容,图片上方的内容。
      <div id="logo"><img src="images/face.png" alt="logo" title="logo"/>
      </div>
      logo 图片下方的内容,图片下方的内容。
  </body>
</html>
```

6.1.3 宽高及实际占位

我们学习标签的常用属性时，例如对表格<table>标签，曾提及其宽度 width、高度 height 属性，那么页面元素实际占据的尺寸是否和设置的宽度、高度一致？结合图 6.2 和图 6.3 的盒子模型示意图可看出，显然，图片实际占据的空间大小应为图 6.2 中最外面的虚线区域。因此，对于某个页面元素，可以得出以下结论：

元素的实际占位尺寸 = 元素尺寸 + 填充 + 边框

这个结论按横纵向可分为以下两种情况。

- 元素实际占位的高度 = height 属性 + 上下填充高度 + 上下边框高度
- 元素实际占位的宽度 = width 属性 + 左右填充宽度 + 左右边框宽度

理解页面元素的宽高和实际占位的关系，这对于后续页面布局中尺寸大小的计算很关键。

6.2 盒子模型用于页面布局

页面的布局类似于报纸的排版，通过将报纸的版面规划为几个版块，最终达到美观、大方的目的。结合 W3C 提倡的结构和样式分离的思想，所以页面布局的思路是：先对页面进行版块划分并使用 XHTML 描述内容结构，然后使用 CSS 样式描述各版块的位置、尺寸等样式。CSS 具体的描述方法是：将各版块看作一个个盒子，利用盒子属性描述各版块的尺寸、外边距、内边距等 CSS 样式，而位置方面一般由浏览器自动控制，不用设置。根据 W3C 提倡的语义化思想，即要求尽量使用表示相关含义的 HTML 标签，因此各版块应采用表示“块”、“分区”含义的 标签进行描述，因此，国内很多教材也将此称为“DIV+CSS 布局”或“层布局”，但其实质是遵守 W3C 结构和样式分离以及语义化的思想。

6.2.1 简单的页面布局示例

下面以贵美商城网站中结构简单的登录页（图 6.9）为例，介绍如何利用盒子属性实现流行的 DIV+CSS 布局。

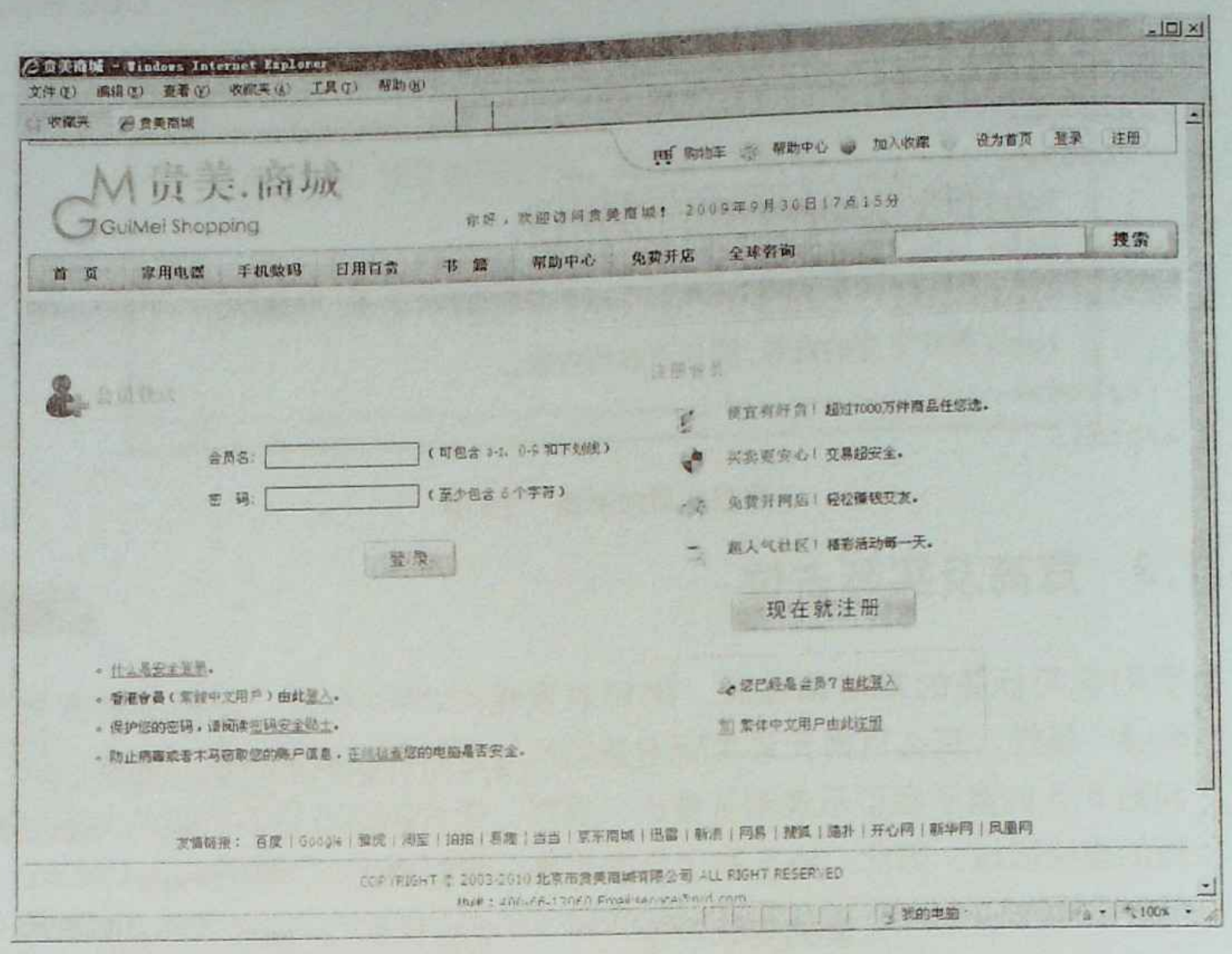


图 6.9　贵美商城网站登录页

假定美工人员已设计完毕图 6.9 所示的页面效果图，现在需要将此转换为 HTML 页面。制作 HTML 页面的第一步是规划页面的整体布局，也就是前面提及的版块的大致划分。分析如下。

（1）先纵向划分：可以明显分为顶部、主体和底部三大块，如图 6.10 所示。

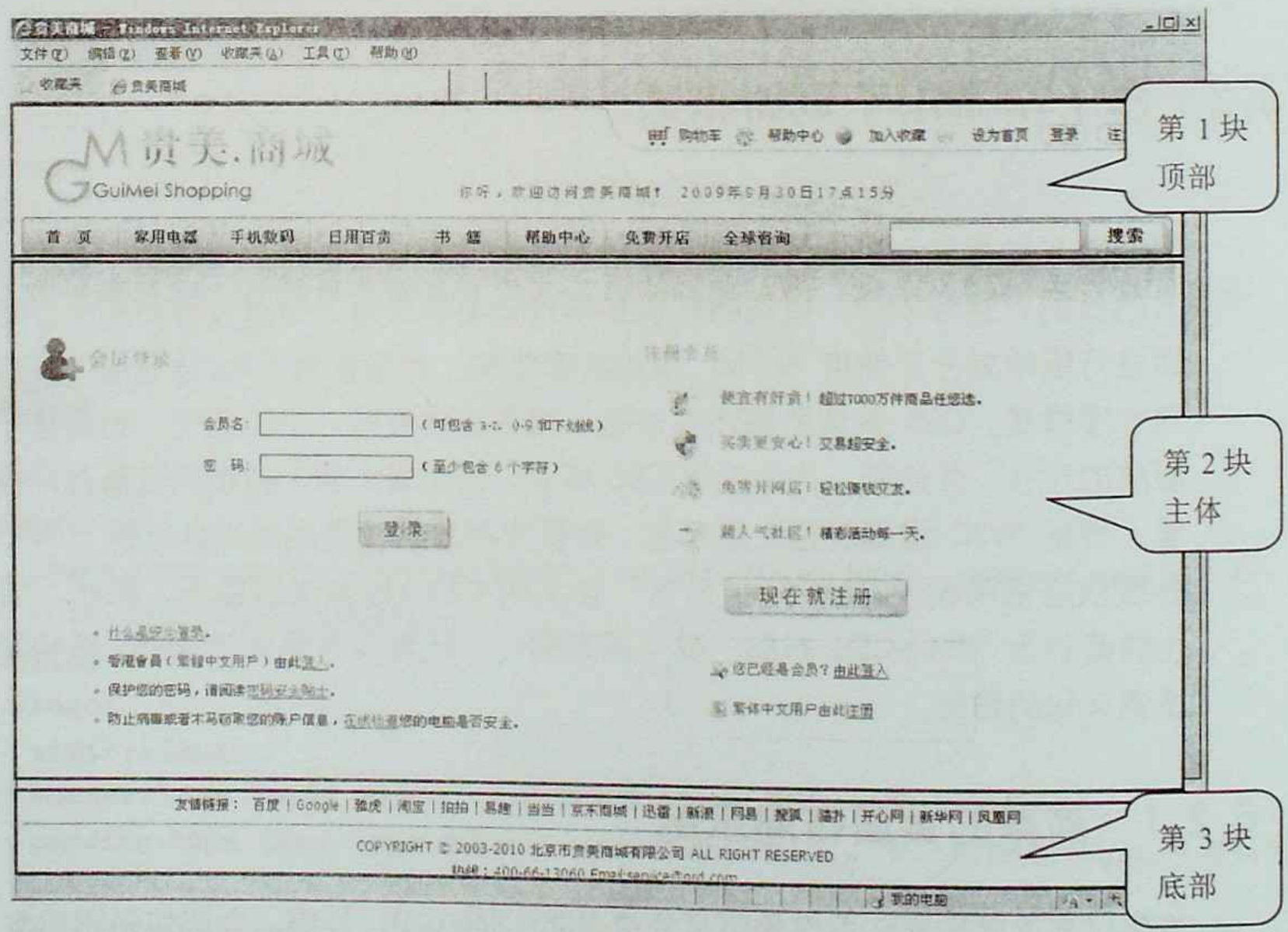


图 6.10　页面的整体划分

从图 6.10 可以看出，三块的宽度一致（均为 980px）。可以用表示“块”的标签 表示，并使用“id”属性给予标识，业界习惯的命名为 header（顶部）、main（主题）、footer（底部）。

```
<div id="header"></div>
<div id="main"></div>
<div id="footer"></div>
```

而整个页面是水平居中，所以在外围增加一个等宽的上级容器（Container），其他块设为 100%宽即可。完整的 HTML 内容结构和 CSS 样式分别如下。

HTML 内容结构：

```
<div id="container">
  <div id="header">
  <div id="main">
  <div id="footer">
</div>
```

CSS 样式：

```
#container{width:980px;margin:0px auto;}  /*整个容器水平居中*/
#header,#main,#footer{width:100%;}   /*三个 div 块的宽度均为 100%*/
```

（2）确定各块的高度，具体语法如下。

```
#header{height:136px;}
#main{ height:400px;}
#footer{height:100px;}
```

（3）根据上述分析，我们分离出布局来，具体效果如图 6.11 所示。

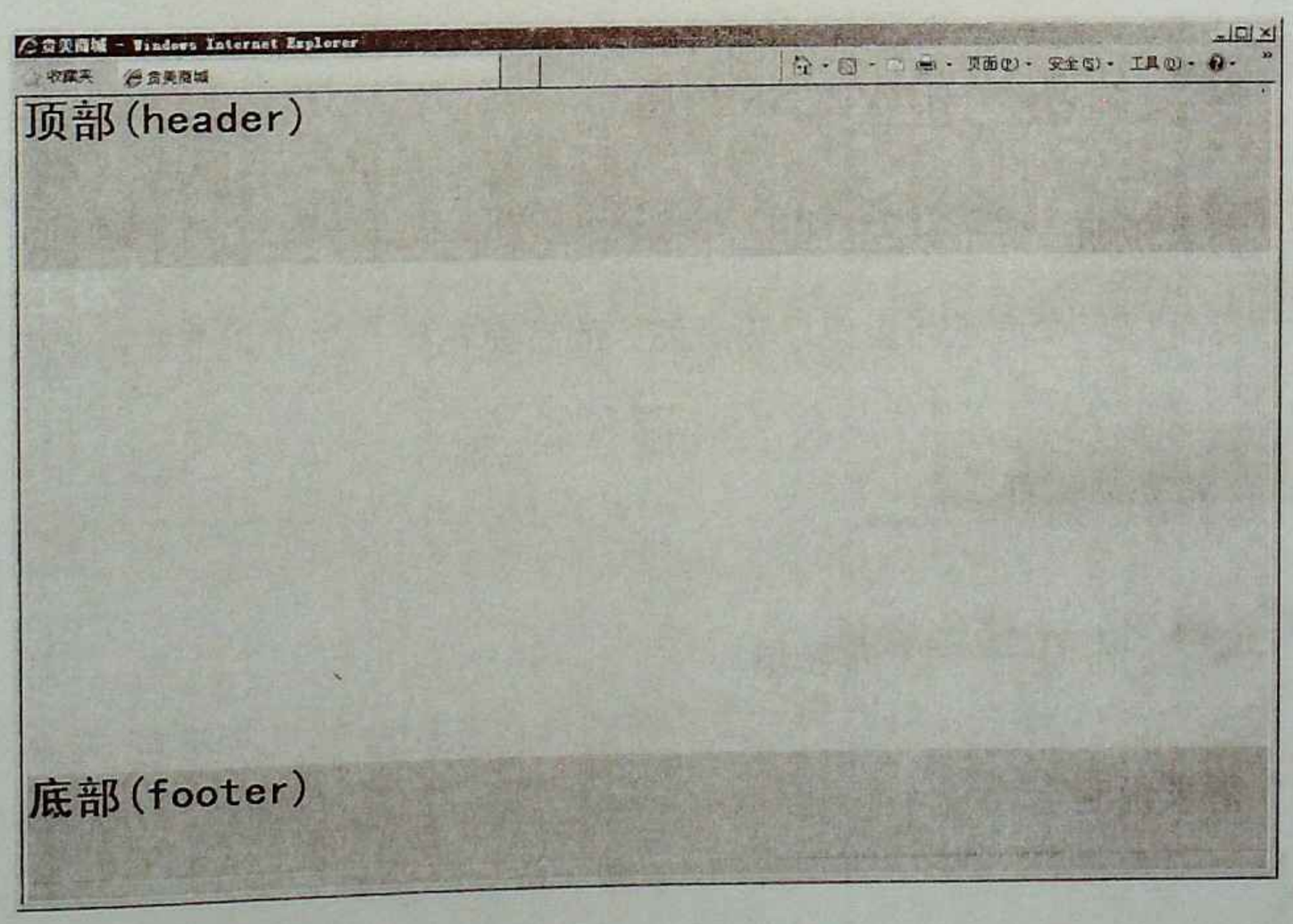


图 6.11　具体的布局效果

另外，为了标识各块，还需设置背景色，完整的布局代码如示例 5 所示。

示例 5

```
<html>
  <head>
     <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
     <title>贵美商城</title>
     <style type="text/css">
     body{margin:0px;padding:0px;} /*修改浏览器对 body 的默认设置,清除为 0*/
     div{font:bold 36px 黑体;color:#000;} /*设置各 div 块的字体、颜色*/
     #container{width:980px;margin:0px auto;} /*提倡有包含关系的各块之间有缩进*/
         #header {width:100%;height:136px; background:#ccc;}
         #main{width:100%;height:400px; background:blue; color:#fff;}
         #footer{width:100%;height:100px; background:#ccc;}
     </style>
  </head>
  <body>
     <div id="container">
       <div id="header">顶部(header)</div>
       <div id="main">主体</div>
       <div id="footer">底部(footer)</div>
     </div> <!--container end-->
  </body>
</html>
```

6.2.2 上机练习

上机练习 1

练习——DIV+CSS 布局

需求说明

使用 DIV+CSS 重新实现“登录页”的页面布局，如 6.2.1 节中的图 6.11 所示。

上机练习 2

练习——修饰表单

需求说明

用给出的素材实现如图 6.12 所示的效果，具体要求如下。

- 输入框边框修饰为 1px 颜色#ccc 的实线，并设置高为 16px、宽为 150px。

- 设置文字的大小为 12px、宋体并且适当设置行高。
- 整个表单左填充 15px，“注册页”标题居中。
- 提示文本与输入框之间的外边距为 10px，如“姓名：”与后面输入框的距离。

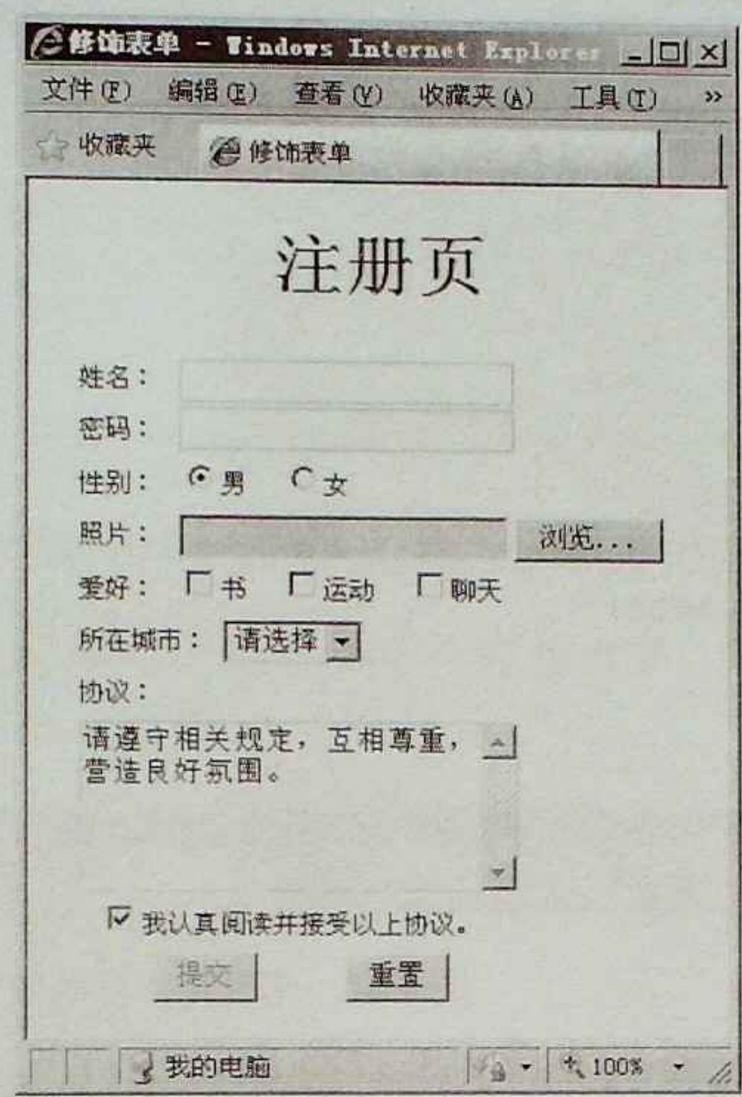


图 6.12　盒子属性修饰表单

6.3　浮动属性

前面章节曾提及使用 float 浮动属性，可以将纵向排列的菜单项改为横向排列。本小节将围绕页面布局，深入讲解浮动的含义及其在布局中的应用。

6.3.1　为什么需要浮动

我们还是结合图 6.11 所示的页面布局来分析。对于 main 块部分，根据图 6.11 的效果图，显然还需进一步分为左、右两块，对应的 HTML 代码如下。

```
<div id="main">
  <div class="login">登录(login)</div>
  <div class="login_right">登录说明(login_right)</div>
</div>
```

根据左右两块各占一半宽度的特点，设置对应的 CSS 为：

```
.login,.login_right{width:50%;height:100%;}
```

这样能实现希望的效果吗？很遗憾的是最终的显示结果如图 6.13 所示，“登录”块与“登录说明”块将独占一行，并没实现如图 6.14 所希望的效果。究其原因，是由于

属于块级标签，具有“换行”显示的特点。那么，如何实现如图 6.14 所示的两个 div 块并排显示的效果？答案就是使用浮动，在这两个块的 CSS 样式中添加“float:left”即可完成。

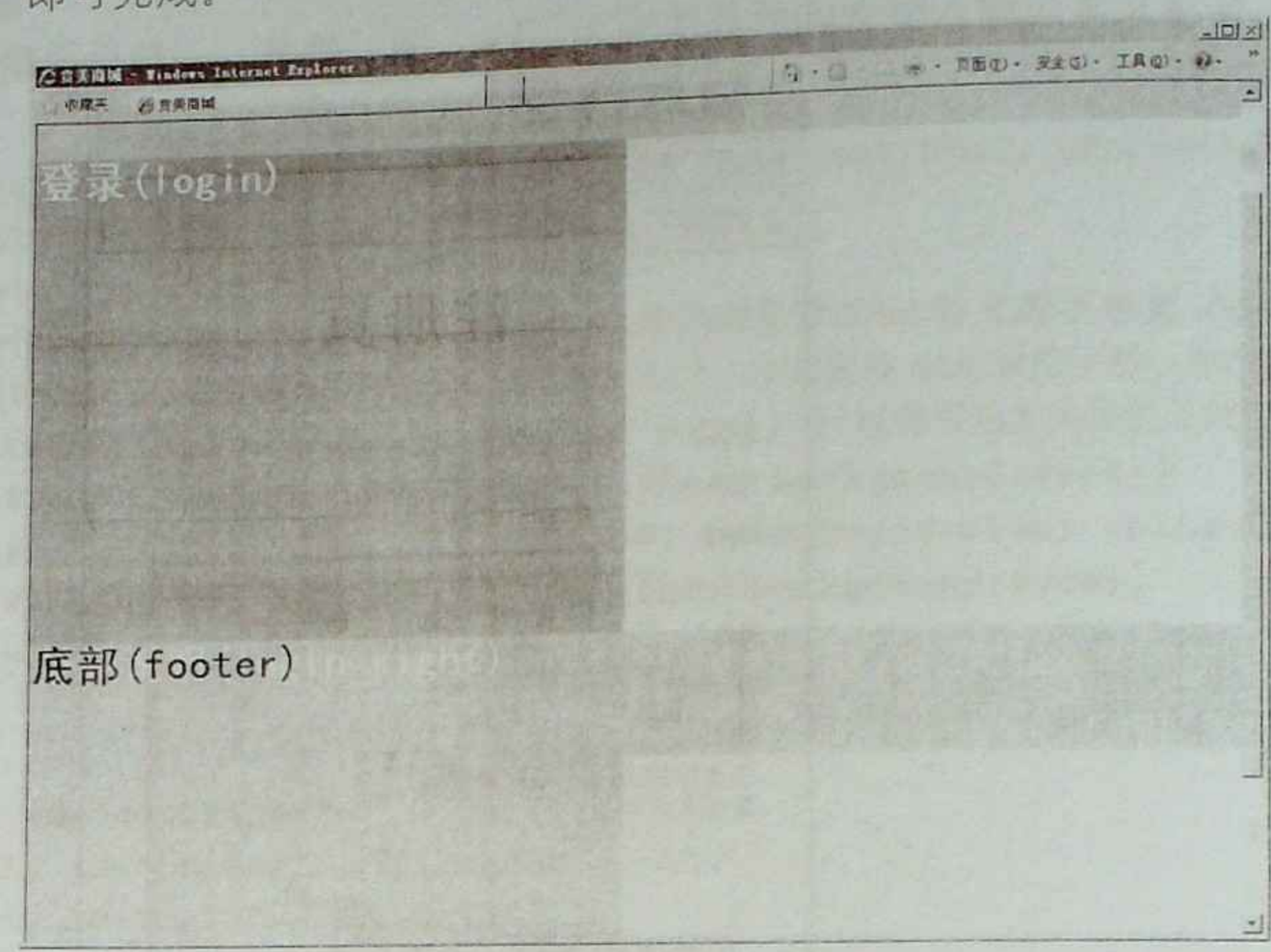


图 6.13　未应用浮动的效果

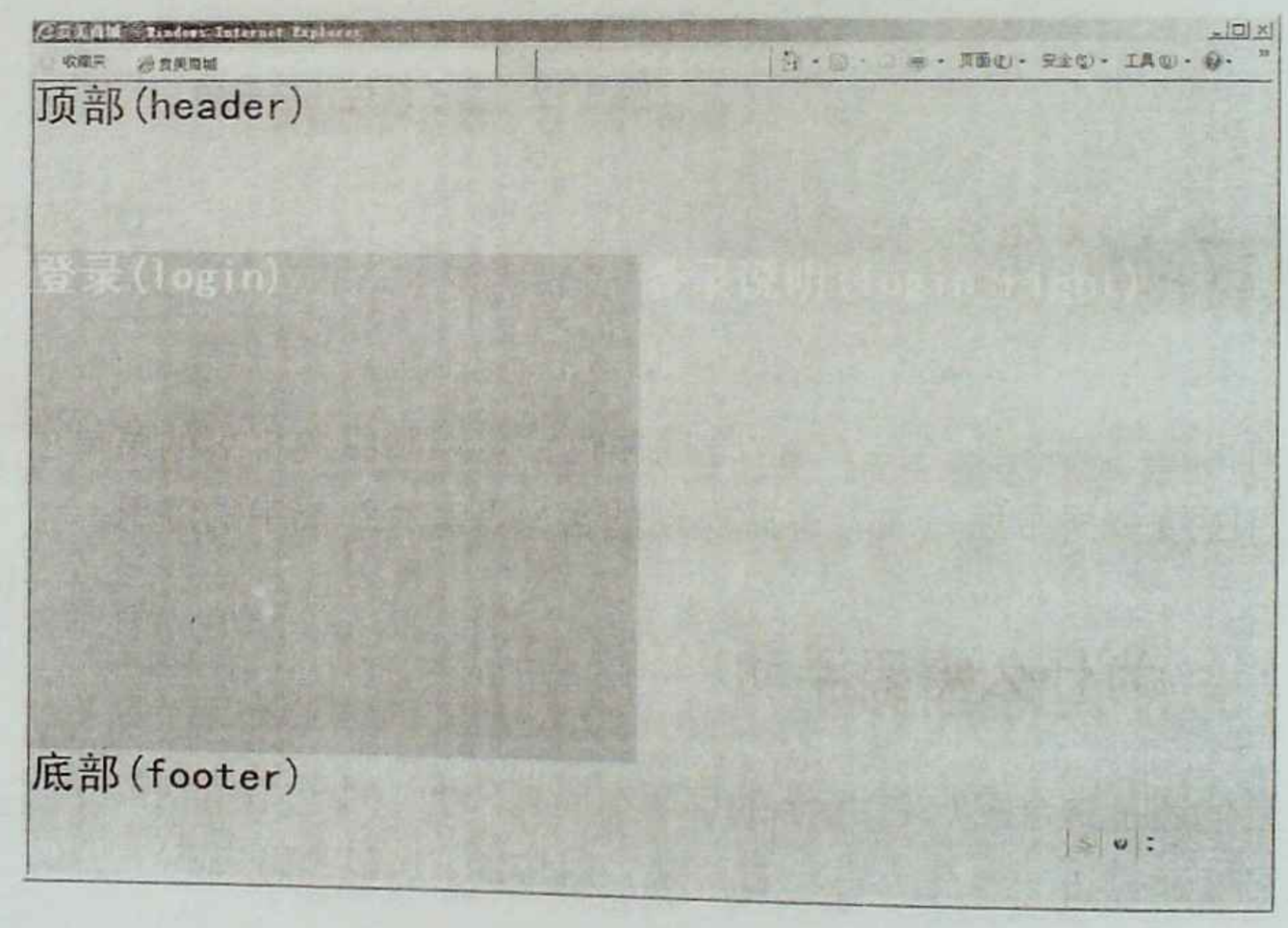


图 6.14　应用浮动的效果

6.3.2　什么是浮动

前面曾提及，页面布局时，对于各块的位置不需要额外设置，即不需要通过设置各块的横向或纵向坐标来确定各块的位置，原因是 Web 页面默认采用常规文档流的方式自动排列各元素，即根据各元素在 HTML 文档中出现的先后顺序，采用从上往下、从左往右的方式自动排列；对于块级元素则换行。对于行级元素，则按行逐一显示，这种定位机制的好处是：Web 页面的内容将跟随浏览器窗口的大小自动调整。

如表 6-4 所示，可以使用“float:浮动方向”设置某个元素的浮动行为，该元素将脱离常规文档流，向左或向右移动，直到它的外边距碰到父元素（容器）的边框或另一个浮动元素的边框为止。关于浮动的理论比较复杂，在实际应用中，掌握浮动的一些基本特点即可。

表 6-4　浮动的设置

常见的浮动设置	含义
float:left	设置元素左浮动
float:right	设置元素右浮动
float: none	默认值，不浮动

1. 块级元素（如 ）设置浮动后将失去“独占一行”的特征

我们不妨做如下试验，假定页面中有三个 块级元素，按常规文档流的方式，将默认排列为如图 6.15 所示的页面效果。

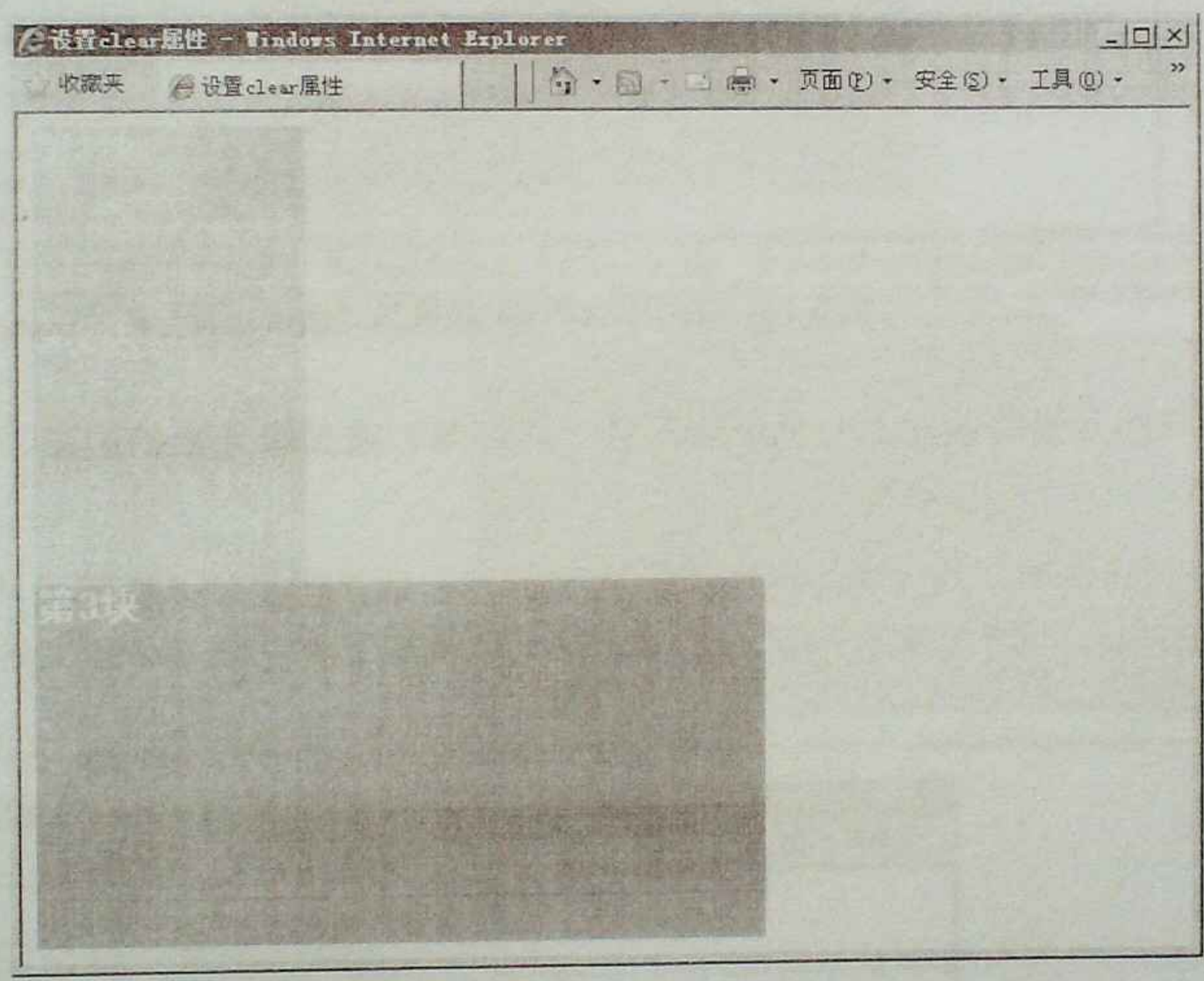


图 6.15　按常规文档流排列的 div 块

对应的 HTML 页面的关键代码如下。

```
<style type="text/css">
  div{color:#fff;font:bold 22px 黑体;}
  #div1{width:150px;height:100px;background:#ccc;}
  #div2{width:150px;height:150px;background:#33FF33; }
  #div3{width:400px;height:200px;background:#3cc;}
</style>
</head>
<body>
  <div id="div1">第 1 块</div>
  <div id="div2">第 2 块</div>
```

```
  <div id="div3">第 3 块</div>
</body>
```

现在在三个 div 块的样式中都加上“float：right”，三个 div 块的 CSS 代码如下。

```
#div1{width:150px;height:100px;background:#ccc;float:right;}
#div2{width:150px;height:150px;background:#33FF33; float:right;}
#div3{width:400px;height:200px;background:#3cc;float:right;}
```

我们将会发现如图 6.16 所示的页面效果。究其原因，是因为三个块设置浮动属性后失去了“独占一行”的特征，然后脱离常规文档流向右浮动，第一块直到父级容器（即页面窗口<body>）才停止；后两块直到前一个浮动元素才停止，最后出现如图 6.16 所示的效果。

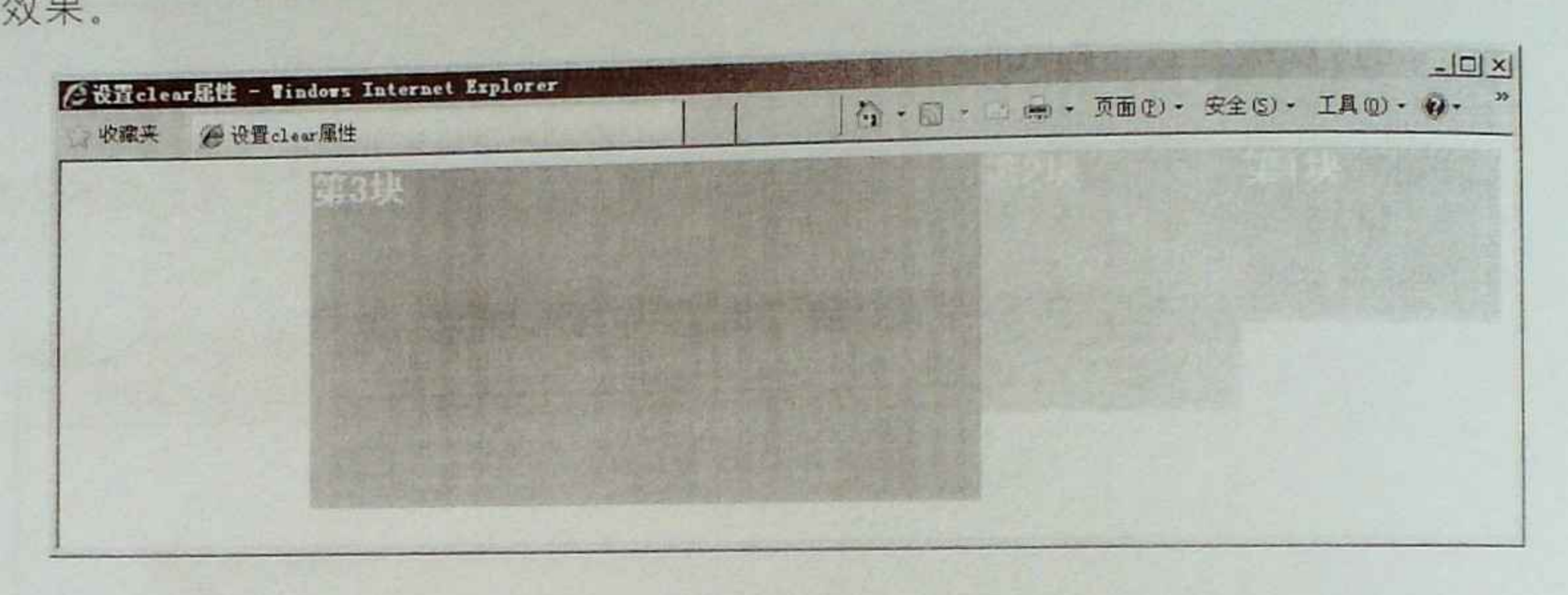


图 6.16　设为浮动的 div 块失去“独占一行”的特征

2．浮动元素将紧贴上一个浮动元素（同方向）或父级元素的边框，如宽度不够将换行显示

缩小浏览器窗口的大小，还将发现第三个 div 块换行显示，如图 6.17 所示。这是因为第二个 div 块左边剩余的空间宽度不够，容纳不了第三个 div 块，所以第三个 div 块换行显示。

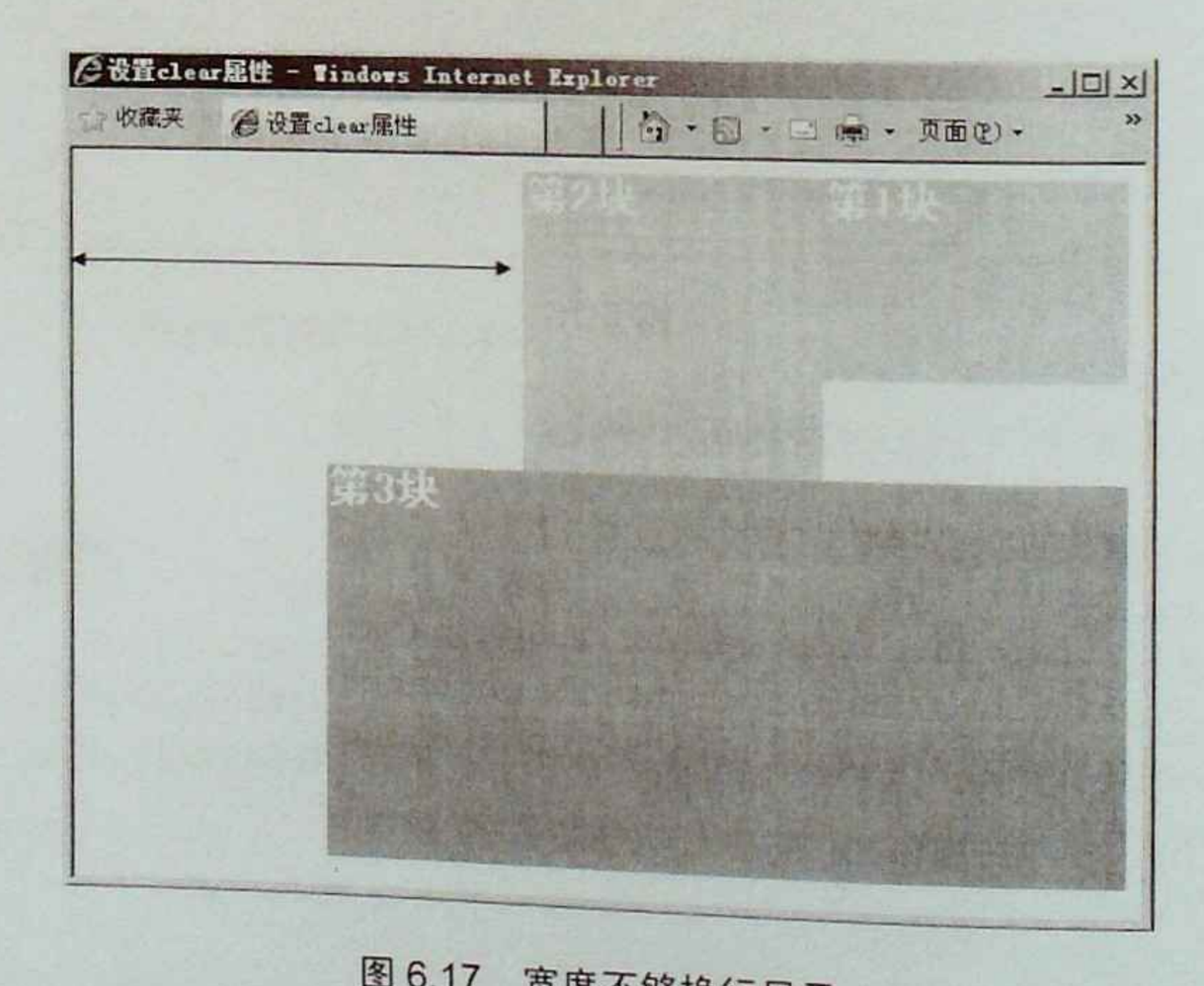


图 6.17　宽度不够换行显示

3．浮动元素将占据行内元素的空间

我们继续做试验：修改代码，在第二个 div 块后加一些文本，对应的 HTML 代码如下。

```
<div id="div2">第 2 块</div>
```

其他内容充满 DIV 剩余空间，div 块失去原换行显示的效果，变成类似行内元素。

```
……<!—省略了大段文本行内元素-->
<div id="div3">第 3 块</div>
```

然后保存查看网页，将发现如图 6.18 所示的“文本围绕”效果。究其原因，是因为浮动元素变成特殊的“行内元素”后，将占据行内元素的空间，行内元素只能在剩余空间进行排列显示，所以出现了“文本围绕”效果。

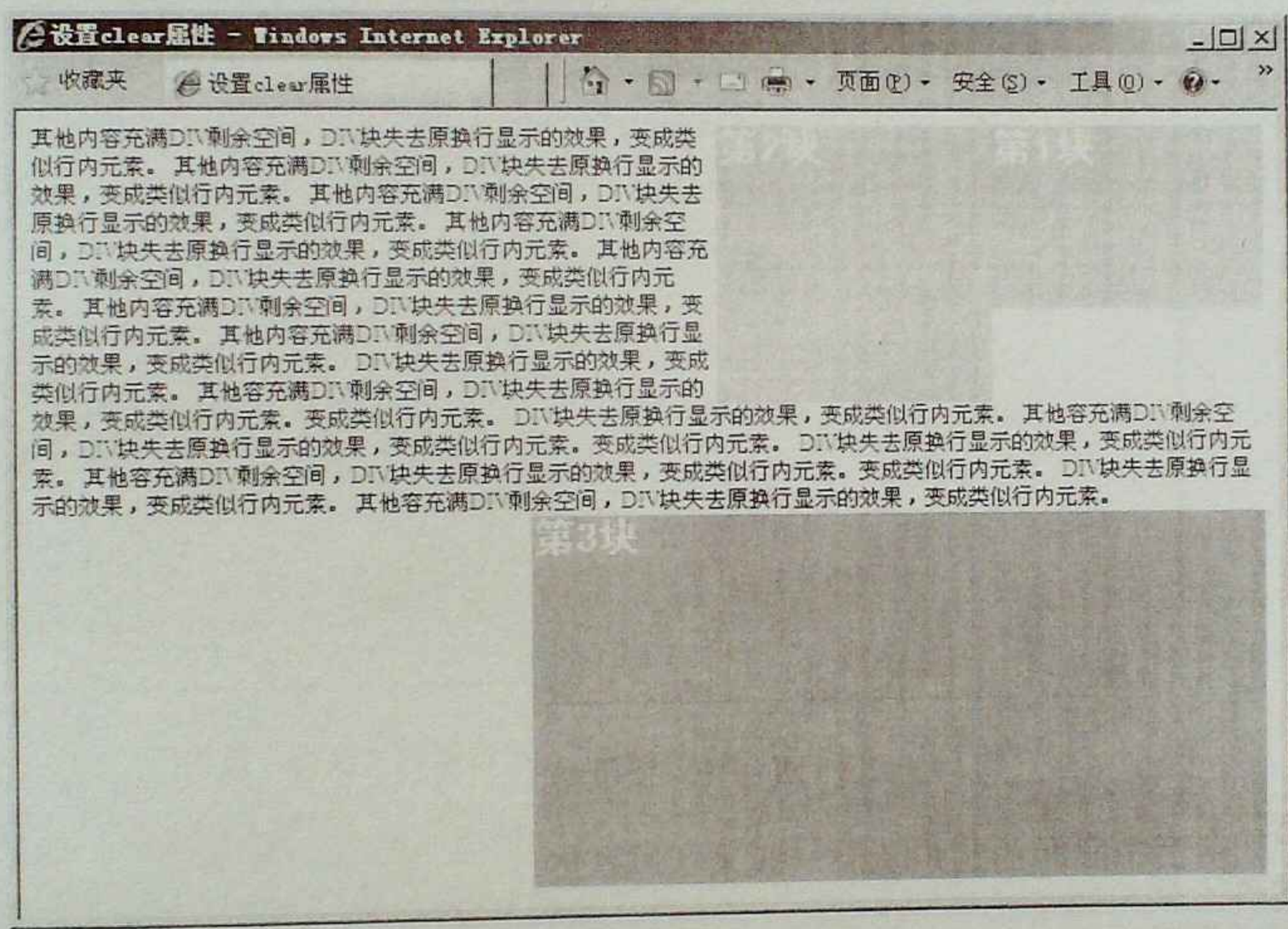


图 6.18　浮动元素占据了行内元素的空间

6.3.3　使用 clear 清除浮动

正如图 6.17 所示，浮动元素将根据浏览器窗口剩余的宽度，决定是否换行显示。在实际的页面布局应用中，有时需要强制换行显示，避免影响页面效果，这时就需要使用 clear 清除属性。

“clear”一般都翻译为“清除”，但其实际含义表示和前一个浮动元素换行区隔开，只对块级元素有效。其用法如表 6-5 所示。根据前一个浮动元素向左或向右浮动，使用相应的“clear:left”或“clear:right”进行区隔，从而实现“强制换行”的效果。例如，对于上述提及的“强制换行”问题，只需在第三块 div 样式中加入“clear：right”，代码如下。

```
#div3{width:400px;height:200px;background:#3cc; float:right;clear:right;}
```

这样就实现了我们希望得到的“换行效果”，如图 6.19 所示。实际开发中有时不知道上一个浮动元素的浮动方向，所以常用“clear：both”来替代，表示不管前一个浮动元素是左浮动还是右浮动，都进行换行区隔显示，这种用法更通用。

表 6-5　清除的设置

常见的清除设置	含义
clear:left	在左侧不允许有浮动元素
clear:right	在右侧不允许有浮动元素
clear:both	在左右两侧均不允许有浮动元素
clear: none	默认值，允许两侧都有浮动元素

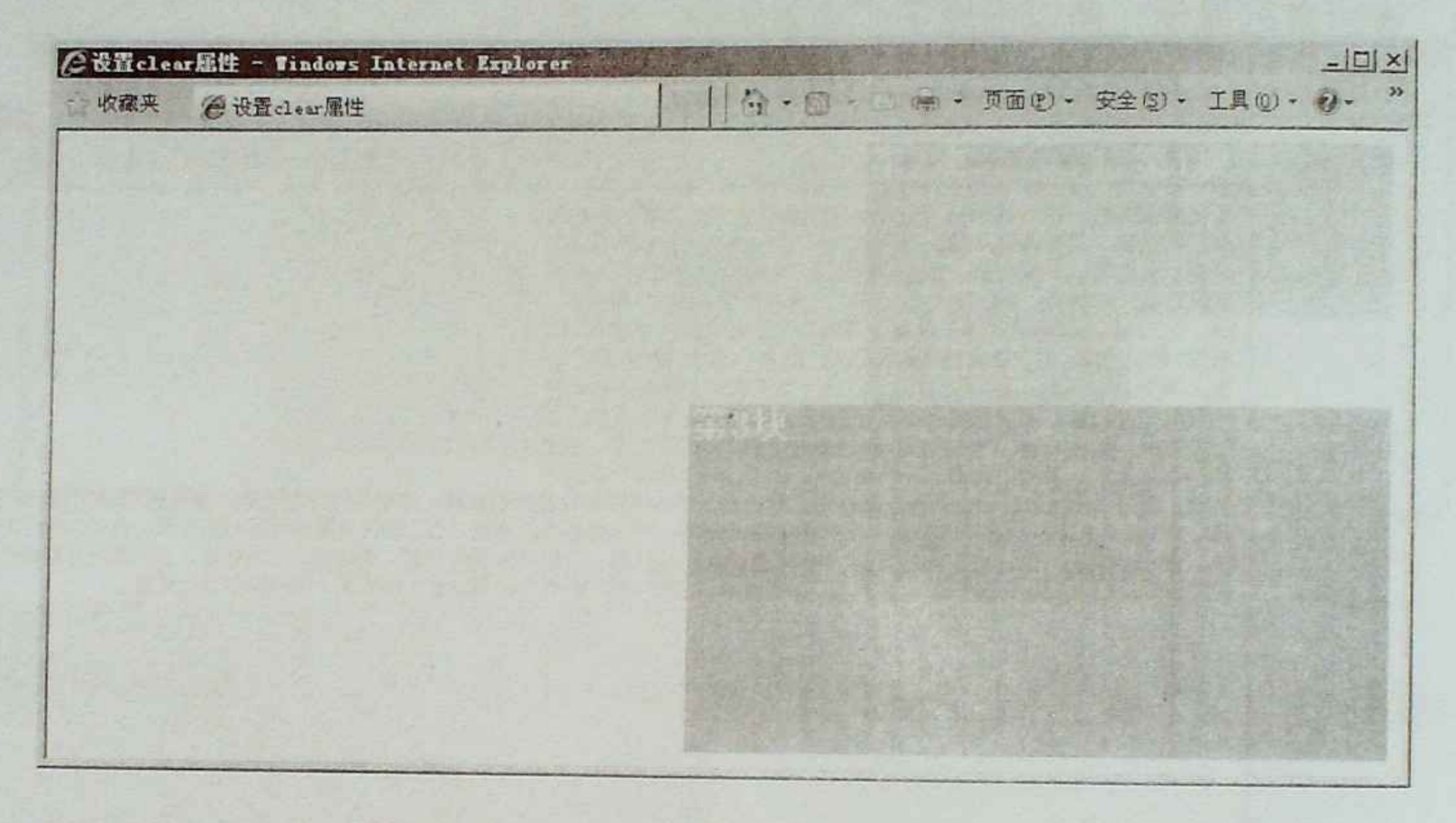


图 6.19　使用 clear 属性达到的“换行”效果

页面布局时的定位机制一般采用默认的普通文档流方式，除浮动外，偶尔还会使用绝对定位、相对定位的定位机制，它们都是偏移默认位置的定位方式，请在课下学习第 11 章的在线培训内容，并完成相应的训练任务。

6.3.4　上机练习

上机练习 3

练习——实现贵美商城网站登录页的完整布局

需求说明

在上一个练习的基础上实现布局，将中间 main 块分为左右两块，如图 6.20 所示。

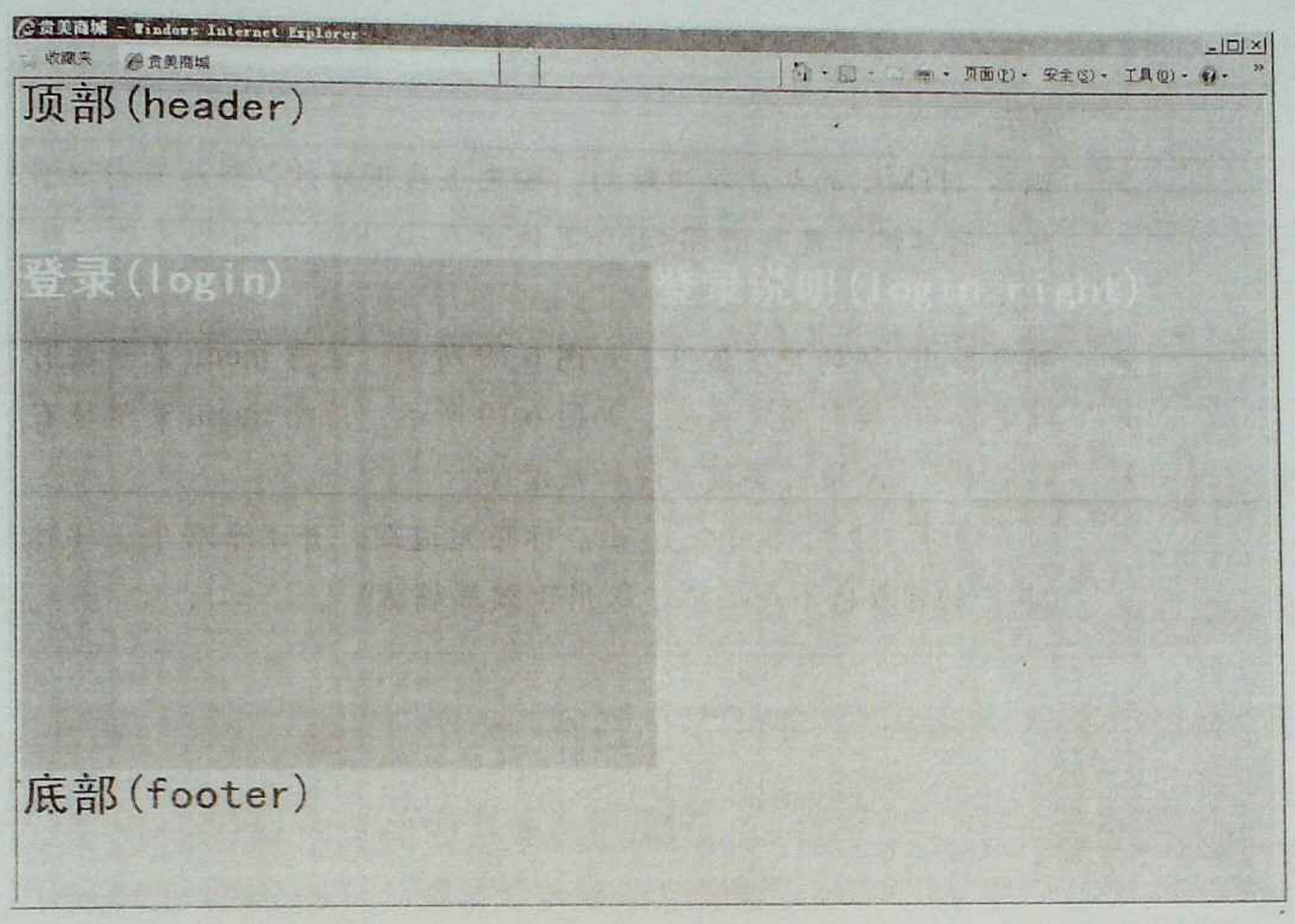


图 6.20　贵美商城网站登录页的整体布局

上机练习 4

练习——实现贵美商城网站顶部的局部布局

需求说明

根据图 6.21 所示的顶部页面效果图，分区块实现局部布局，如图 6.22 所示。

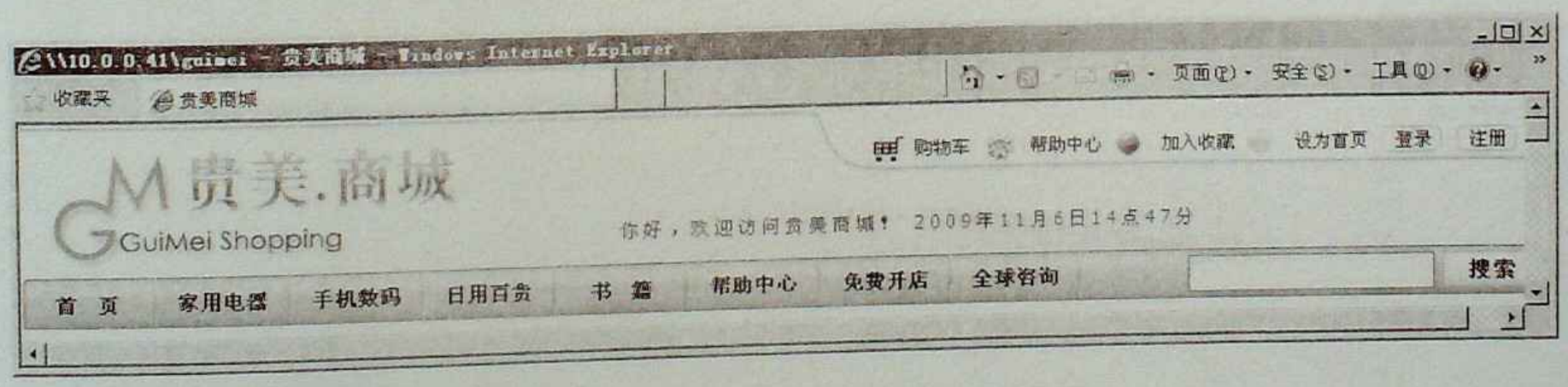


图 6.21　贵美商城网站顶部的页面效果

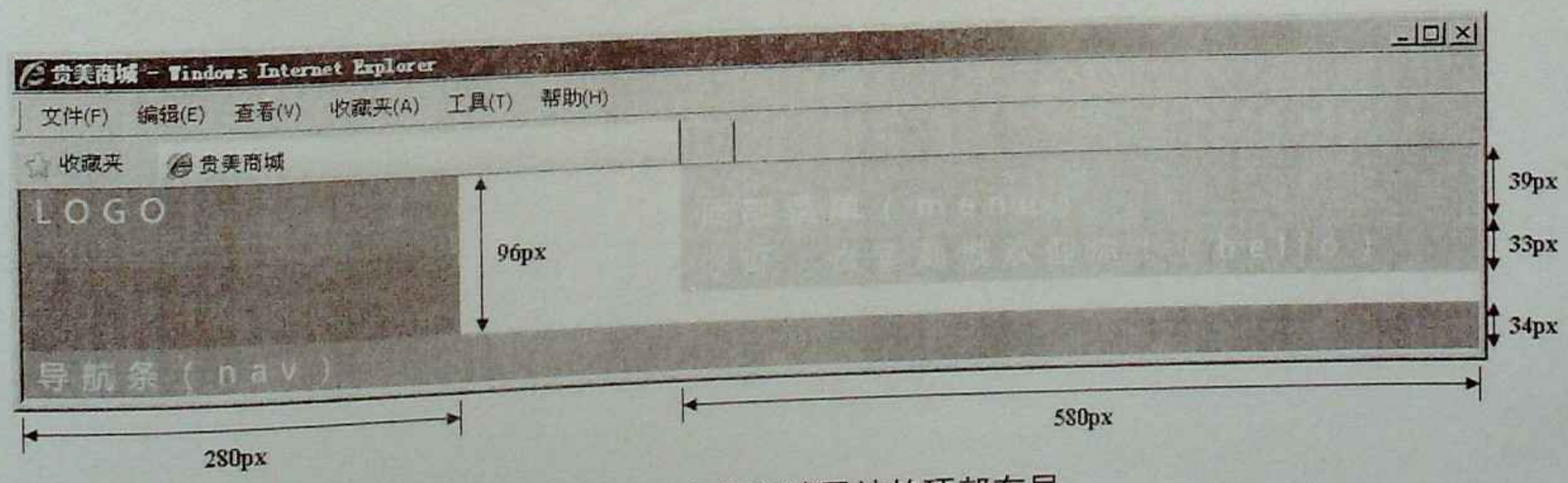


图 6.22　贵美商城网站的顶部布局

提示

- 编写 HTML 的层次组织结构。按先上后下分行，然后每行从左到右的文档流顺序。简单讲，就是头部 里依次包含 logo、顶部菜单、欢迎和导航条四个 。
- 确定各 div 块的宽度属性，如图 6.19 所示，注意 menu 菜单块有左填充 20px。
- 确定各 div 块的高度属性，如图 6.19 所示，注意 menu 菜单块有上填充 15px。
- 确定哪些 div 块需要设置浮动及清除。
- 细节修饰，设置<body>、 标签无边距，并水平居中。字体样式及各背景色（其中的背景色不是必须，仅用于效果调试）。

本章总结

- 盒子模型是页面布局的基础，它包括外边距、边框、内边距以及元素的宽高等属性。
- 流行的 DIV+CSS 布局采用语义化的 标签组织 XHTML 结构，使用 CSS 中的盒子模型描述宽、高、内边距等 CSS 样式。
- 盒子具有如下三个属性。
  - border（边框）：盒子外壳本身的宽度。
  - padding（内边距）：内容与边框间的距离。
  - margin（外边距）：盒子与其他盒子之间的距离。
- float（浮动）的属性如下。
  - float 通常用于让块级元素横向紧挨排列或实现特殊排序，而非独占一行。
  - float 属性常用的三个取值为 left、right、none。
- clear（清除）的属性如下。
  - clear 紧跟在 float 元素之后，用换行方式区隔开浮动元素。
  - clear 属性常用的取值有 left、right、both、none，注意和前一个浮动元素的浮动方向必须一致。

本章作业

一、选择题

1. 盒子模型的边框、内边距和外边距属性依次是（　　）。

A. margin、padding、border

B. margin、border、padding

C. border、padding、margin

D. border、margin、padding

2. 盒子模型的层次结构依次是（　　）。

A. 内容及填充>>边框>>背景图>>背景色>>外边距

B. 内容及填充>>边框>>背景色>>背景图>>外边距

C. 边框>>内容及填充>>背景色>>背景图>>外边距

D. 边框>>内容及填充>>背景图>>背景色>>外边距

3. 关于盒子模型实际宽高的说法，正确的有（　　）。

A. 盒子模型的实际占位=填充+边框+元素宽/高（即 padding+border+width/height）

B. 盒子模型的实际高度=上下填充+上下边框+标签宽度

C. 盒子模型的实际宽度=左右填充+左右边框+标签宽度

D. 盒子模型的实际宽高就是标签的 width/height 的属性值

4. 关于 float 和 clear 属性，说法错误的是（　　）。

A. 元素的左浮动效果和文本左对齐效果，其本质完全一样

B. 浮动总是依次填充常规文档流所在行的最左侧或最右侧

C. clear 的作用是清除浮动，或者说是不允许元素左右侧出现浮动元素

D. "clear:left;" 具体含义是如果前一元素存在左浮动，并且为块元素，则分行区隔

5. 关于各类元素的居中，下面说法正确的是（　　）。

A. 块级元素的水平居中，使用 "margin:0px auto;"

B. 文本内容的水平居中，使用 "text-align:center;"

C. 单行文字的垂直居中，需要设置“行高属性 line-height 等于文本所在行的高度 height”

D. 文字的垂直居中，需要设置“vertical-align:middle”

二、简答题

1. 请描述什么是盒子模型，并画出盒子模型的平面结构图和立体结构图。

2. 请描述 float 浮动的特点。

3. 自学老师提供的“学习参考资料”，学习 CSS 的定位知识，制作常见的对联广告，如图 6.23 所示。

图 6.23　对联广告

4. 请结合效果图 6.24，使用 DIV+CSS 实现页面的整体布局。

5. 根据图 6.25 的效果，合理使用 float 与 clear 规划实现局部布局结构，注意合理设置 float 与 clear。

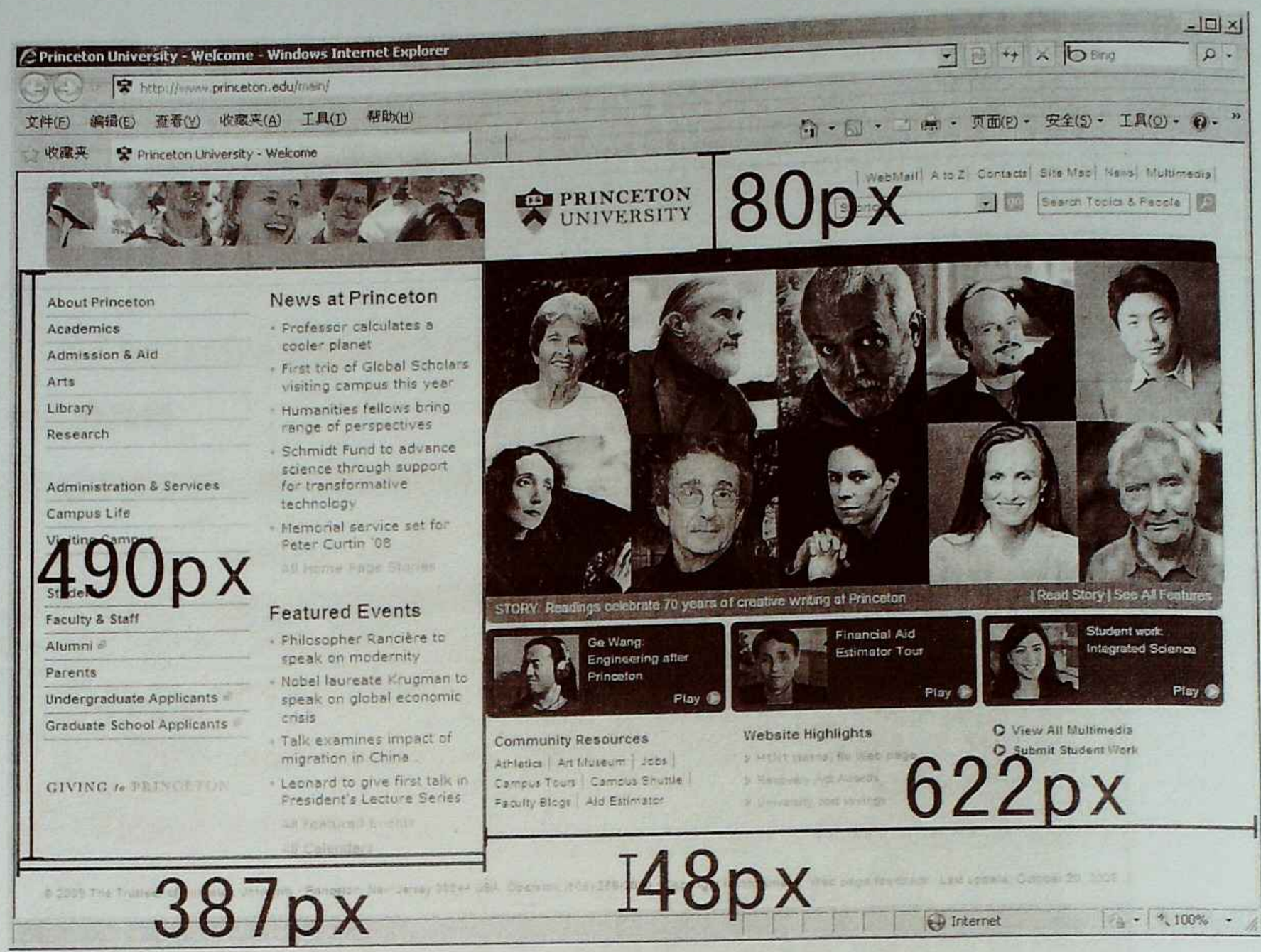


图 6.24　普林斯顿大学首页布局

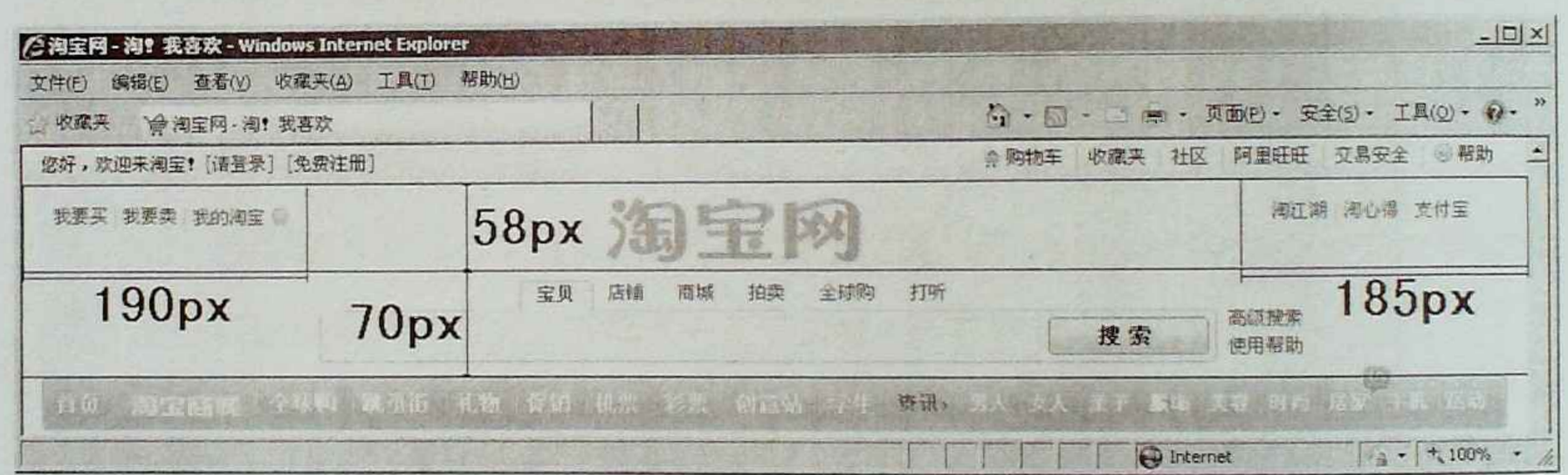


图 6.25　局部布局实现

第 7 章

CSS 样式表（三）

◈本章工作任务

- 制作贵美商城网站的横向主导航条
- 制作贵美商城网站的顶部菜单
- 制作贵美商城网站的新品上架版块

◈本章技能目标

- 使用 div-ul-li 实现局部布局
- 使用 div-dl-dt-dd 实现图文混编
- 使用伪类样式控制超链接样式

本章单词

请在预习时学会下列单词的含义和发音，并填写在横线处。

1. hover：_____
2. link：_____
3. visited：_____
4. active：_____
5. navigation：_____
6. position：_____
7. sheet：_____
8. layout：_____
9. vertical：_____

本章简介

上章学习了 CSS 的布局，本章我们将结合 CSS 的实际使用经验，讲解超链接的特殊性及其伪类、典型的局部应用、多选择器复杂表示及代码复用技巧。

7.1 超链接伪类样式

1. 超链接的特殊性

在第 2 章时，我们学习了超链接的用法。作为 HTML 中常用的标签，同时也是 HTML 区别其他标识语言的最重要特点，超链接的样式有其显著的特殊性：当为某文本或图片设置超链接时，文本或图片标签将继承超链接的默认样式，标签的原默认样式将失效，如图 7.1 所示。文字加超链接后将加上下划线，图片加超链接后将添加边框。单击链接前为蓝色，单击后为紫色。

图 7.1 超链接的特殊性

经验总结

为防止图片加入链接后出现 2px 边框，一般会在 CSS 文件开头加入“img {border:0px;}”。

2. 超链接伪类

上述提及链接单击前和单击后的样式变化，其实是超链接的默认伪类样式。所谓伪类，就是不根据名字、属性、内容而根据标签处于某种行为或状态时的特征来修饰样式。伪类可以对用户与文档交互时的行为作出响应。伪类样式的基本语法为“标签名:伪类名{属性:属性值;}”。

最常用的伪类是超链接伪类，如表 7-1 所示。

表 7-1 超链接伪类

伪类	示例	含义	应用场景
a:link	a:link{color:#999;}	未单击访问时的超链接样式	常用，超链接主样式
a:visited	a:visited {color:#333;}	单击访问后的超链接样式	需区分是否已被访问
a:hover	a:hover{color:#ff7300;}	鼠标悬浮其上的超链接样式	常用，实现动态效果
a:active	a:active {color:#999;}	鼠标单击未释放的超链接样式	少用，一般与 link 一致

例如，假定要实现页面中导航菜单区域内的超链接样式，希望当鼠标移动到超链接上时改变字体大小，给予用户醒目显示，如图 7.2 所示。

图 7.2 超链接伪类

分析思路如下。

- 需要分别定义未访问（a:link）和鼠标悬浮（a:hover）两种状态的样式，而对于鼠标单击时和访问后的样式保持原样即可。
- 在实际应用中，一个页面中有很多链接，现希望只改变导航菜单部分的链接样式，所以需要限定超链接样式的范围。可以使用类或 ID 样式来实现，但实际应用中更流行的做法是采用“父级元素+空格+子元素”，表示区域限制的选择器。例如，对于以下导航菜单结构：

```
<div class="nav">
  <ul>
    <li><a href="#">家用电器</a></li>
    <li><a href="#">手机数码</a></li>
    <li><a href="#">日用百货</a></li>
  </ul>
</div>
```

假定希望只修饰上述 块内的<li>标签，则选择器为 div li{样式代码……}；为了进一步限制是类样式为“nav”的 块，则选择器为 .nav li {样式代码……}，再如：描述上述结构中的链接样式，则选择器为 .nav li a {样式代码……}。这样做的好处是可读性强，减少了不必要的类选择器命名。

图 7.2 的效果对应的 CSS 代码如示例 1 所示。

示例 1

```
<html>
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=gb2312"  />
   <title>超链接伪类</title>
    <style>
      .nav{width:100%;}
      .nav ul{list-style:none;}
      .nav li {
        width:120px;
        float:left;
        text-align:center;
        font-weight:bold;
        }
      .nav li a{padding:8px 15px;}
      .nav li a:link{
        color:#333;
        font-size:13px;
        }
      .nav li a:hover{
        color:#ff7300;
        font-size:20px;
        }
      .nav li a:active{
        color:#333;
        font-size:13px;
        }
      .nav li a:visited{
        color:#333;
        font-size:13px;
        }
    </style>
  </head>
  <body>
    <div class="nav">
      <ul>
        <li><a href="#">家用电器</a></li>
        <li><a href="#">手机数码</a></li>
        <li><a href="#">日用百货</a></li>
      </ul>
    </div> <!—nav end-->
```

为增加代码可读性，同时也为了限制样式修饰的范围和优先级的考虑，专业的做法是用：“父元素+空格+子元素”限定范围。结合 HTML 代码，此处表示类名为 nav 的 div 块中的 ul 元素，后续同理

未访问

鼠标悬浮其上

点击时

点击后

```
  </body>
</html>
```

在实际应用中，可以利用 CSS 样式的继承特点，先定义四种状态统一的样式，然后再根据需要定义个别状态的样式，关键代码即为：

```
a{color:#333;}/*4 个伪类采用同一样式(含 link)*/
a:hover{color:#ff0;} /*再单独为鼠标悬浮定义特殊样式*/
/*如还有需要,则可以再写 a:visited 和 a:active*/
```

7.2 如何应用样式

7.2.1 三种应用方式

之前所学 CSS 样式代码，几乎都是在同文件的<head>标签中加入 CSS 代码，即内部样式表，但这并非是唯一方法。在 CSS 中，应用样式有三类方式：内部样式表、外部样式表及行内样式。下面我们来依次学习各种应用方式的优缺点及应用场景。

1. 内部样式表

正如之前所有的示例一样，我们把 CSS 代码写在<head>的<style>标签中，与 HTML 内容位于同一文件，这就是内部样式表。这种方式方便在同页面中修改样式，但不利于在多页面间共享复用代码及维护，对内容与样式的分离也不够彻底。实际开发时，会在页面开发结束后，将这些样式代码剪切到单独的 CSS 文件中，将样式和内容彻底分离开，即下面介绍的外部样式表。

2. 外部样式表

把 CSS 代码单独写在另外一个或多个 CSS 文件中，需要用时在<head>中通过<link/>标签引用，这种方式就是应用外部样式表文件的方式。它的好处是实现了样式和结构的彻底分离，同时方便网站的其他页面复用该样式，利于保持网站的统一样式和网站维护。其语法如下。

```
<link rel="stylesheet" type="text/css" href="css 文件地址"  />
```

接下来以示例 1 作例子，讲解如何转为外部样式表文件的方式。

(1) 新建一个文本文件，另存到 css 目录下，并把文件名修改为 layout.css。

(2) 把示例 1 中<style>标签中间的所有代码，剪切到 layout.css 中。

(3) 删除示例 1 中的<style></style>标签，在<head>标签中加入一行<link/>标签语句，并把 href 文件的地址属性设置为第一步另存为的“css/layout.css”，即：

```
<link rel="stylesheet" type="text/css" href="css/layout.css"  />
```

下面通过一个示例来演示这种外部样式表文件的共享复用。我们可以在另外一个页面加入上述代码实现相同的样式，如图 7.3 所示。

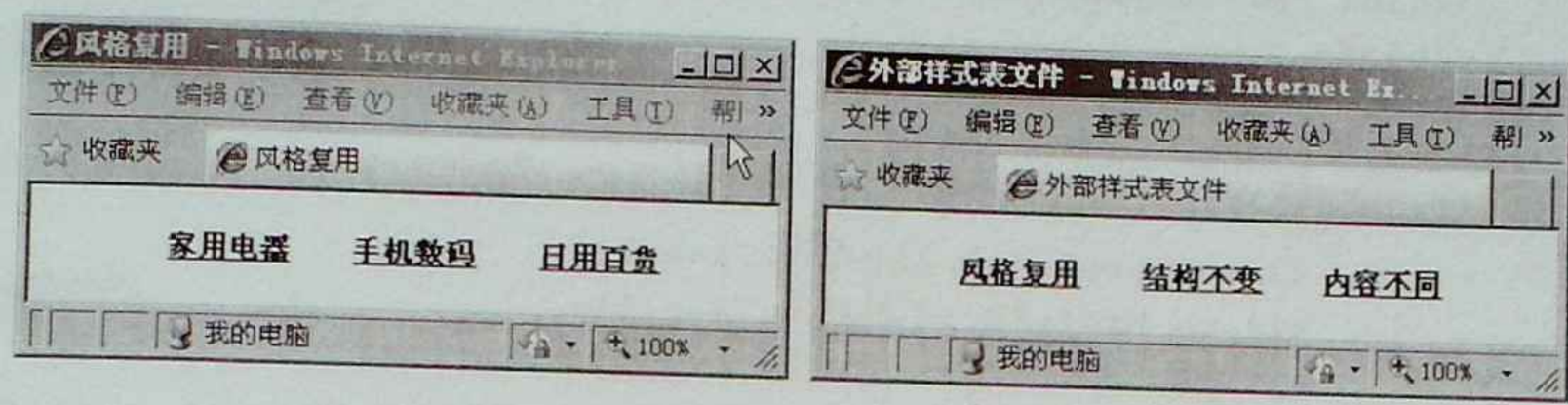


图 7.3　采用外部样式表文件的方式有利于复用及维护

图 7.3 的实现方法就像示例 2 那样，在两个页面<head>标签中间都加上上面这条<link/>标签语句。而且对于 css 目录下 layout.css 文件的任何修改，将在所有引用页面中生效。

示例 2

layout.css 文件：

```
.nav{width:100%;}
.nav ul{list-style:none;}
.nav li {
  width:84px;
  float:left;
  text-align:center;
  font-weight:bold;
  }
.nav li a{padding:8px 10px;}
.nav li a:link{
  color:#333;
  font-size:13px;
  }
.nav li a:active{
  color:#333;
  font-size:13px;
  }
.nav li a:visited{
  color:#333;
  font-size:13px;
  }
.nav li a:hover{
  color:#ff7300;
  font-size:13px;
  }
```

firstPage.html 文件：

```
<html>
  <head>
```

```
 <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
 <title>样式复用</title>
 <link rel="stylesheet" href="css/layout.css" type="text/css"  />
</head>
<body>
  <div class="nav">
     <ul>
       <li><a href="#">家用电器</a></li>
       <li><a href="#">手机数码</a></li>
       <li><a href="#">日用百货</a></li>
     </ul>
    </div> <!—nav end-->
  </body>
</html>
```

secondPage.html 文件：

```
<html>
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
   <title>外部样式表文件</title>
   <link rel="stylesheet" href="css/layout.css" type="text/css"  />
  </head>
  <body>
     <div class="nav">
       <ul>
         <li><a href="#">样式复用</a></li>
         <li><a href="#">结构不变</a></li>
         <li><a href="#">内容不同</a></li>
       </ul>
     </div> <!—nav end-->
  </body>
</html>
```

因为外部样式表文件方式的优点众多，因此被广泛应用。以后的示例中，都将采用此方式。

3. 行内样式

某些情况下，我们需要对特定某个标签进行单独设置，最直观的方法就是在标签的属性内直接设置。如我们修改上例，实现图 7.4 中把“日用百货”样式设置为红色大小 10px。

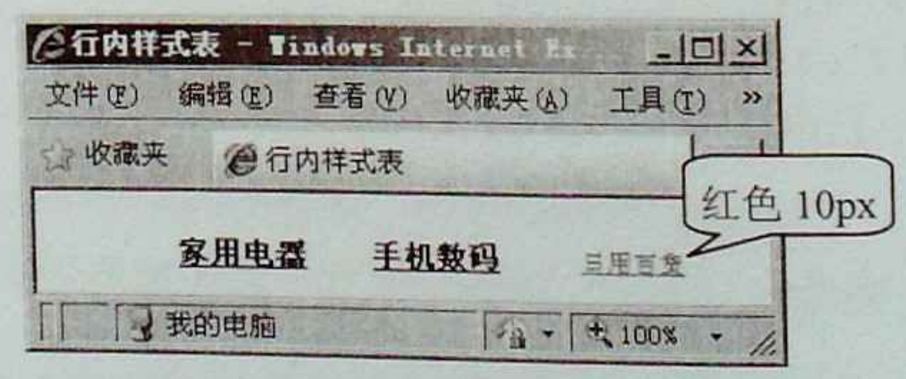


图 7.4 外部样式表文件利于复用及维护

其用法是在所需修饰的标签内加 style 属性，后续为多条样式规则，多条样式规则用分号区分开：

```
style="color:red;font-size:10px;"
```

示例 3 中演示了此操作的实现，我们称这种方式为行内样式。

示例 3

```
<html>
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
   <title>行内样式表</title>
   <link rel="stylesheet" href="css/layout.css" type="text/css"  />
  </head>
  <body>
     <div class="nav">
       <ul>
         <li><a href="#">家用电器</a></li>
         <li><a href="#">手机数码</a></li>
         <li><a href="#" style="color:red;font-size:10px;">日用百货</a></li>
       </ul>
     </div>   <!--nav end-->
  </body>
</html>
```

这种方法虽然直观，但却应尽量少用或不用，因为内容与样式混写在一起，失去了 CSS 的最大优点。可以用外部样式表文件的方式来代替。

譬如上例中，我们为该<li>标签设一个类名 redli，删除 " style="color:red;font-size:10px;" " 并在 layout.css 的最后行加入代码 ".redli{color:red;font-size:10px;}"，完全可以实现相同的效果。

7.2.2 样式优先级

前面曾提及，CSS 的全称为"层叠样式表"，因此，对于页面中的某个元素，它允许同时应用多类样式（即叠加），页面元素最终的样式即为多类样式的叠加效果。但这存在一个问题：当同时应用上述的三类样式时，页面元素将同时继承这些样式，但样式之间如有冲突，应继承哪种样式？即存在样式优先级的问题。同理，从选择器角度，当某个元素同时应用标签选择器、ID 选择器、类选择器定义的样式时，也存在样式优先级的问题。CSS 中规定的优先级规则为：

- 行内样式表>内部样式表>外部样式表。
- ID 选择器>类选择器>标签选择器。

行内样式表>内部样式表>外部样式表，即"就近原则"。例如示例 4，对"日用百货"链接，同时应用了外部样式表、内部样式表和行内样式，这三类样式在字体颜色

(color)、字号(font-size)方面定义的规则有冲突，因为行内样式离被修饰对象<a>最近，所以最终的样式以行内样式定义的为准，即红色、10px，最终的样式效果如图 7.5 所示。

示例 4

```
<html>
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=utf-8"  />
   <title>样式优先级</title>
     <style type="text/css"">
       .nav ul li a:link{color:blue;}  /*内部样式表*/
     </style>
   <link rel="stylesheet" href="css/layout.css" type="text/css" /> /*外部样式表*/
  </head>
  <body>
    <div class="nav">
      <ul>
        <li><a href=""#">家用电器</a></li>
        <li><a href=""#">手机数码</a></li> <!--下面为行内样式-->
        <li><a href=""#" style="color:red;font-size:10px;">日用百货</a></li>
      </ul>
    </div>
  </body>
</html>
```

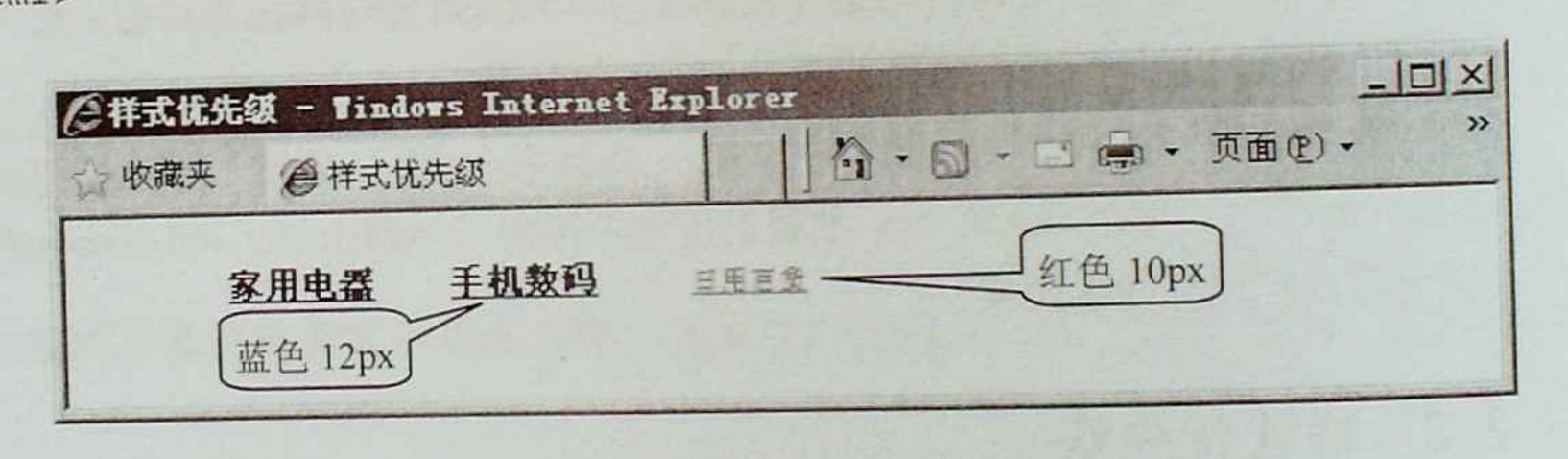


图 7.5　各类样式表的优先级

下面再举例说明 ID 选择器、类选择器、标签选择器的优先级问题。例如，对应示例 5 中的 元素，同时用了 ID 选择器（#nav_id）、类选择器（.nav）、标签选择器（div），仔细比较各类选择器定义的样式规则，发现背景色定义有冲突，所以最终的背景色以 ID 选择器定义的#ccc（灰色）为准，而对于其他不冲突的样式规则，则将全部应用到被修饰对象 标签上，最终的样式效果如图 7.6 所示。

示例 5

```
<html>
  <head>
     <style>
       #nav_id{width:300px;  background: #ccc;} /*ID 选择器*/
```

```
      .nav{height:100px; background: red;} /*类选择器*/
      div{border:5px solid green; background: blue;} /*标签选择器*/
    </style>
  </head>
  <body>
    <div class="nav" id="nav_id">  <!-同时应用 ID、class、标签三类选择器-->
      <ul>
        <li><a href="#">购物车</a></li>
      </ul>
    </div>
  </body>
</html>
```

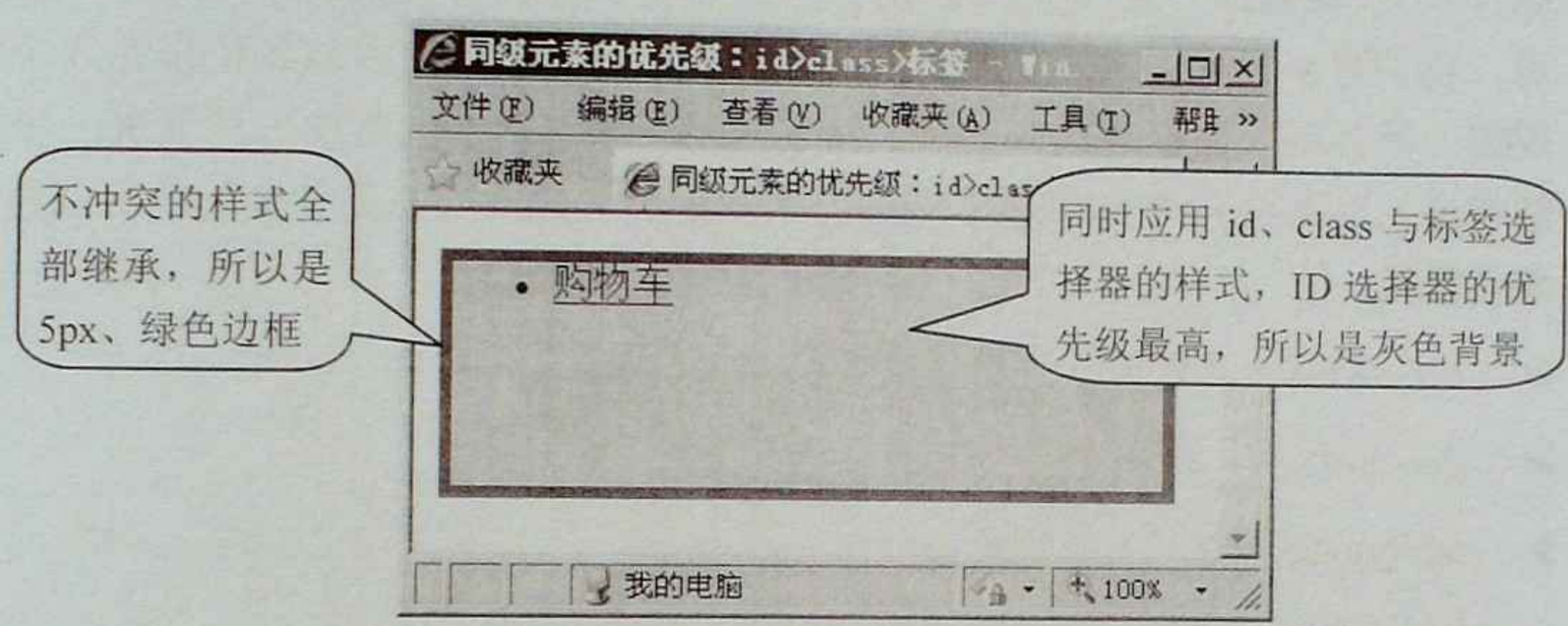


图 7.6　各类选择器的优先级

7.2.3　上机练习

上机练习 1

练习——使用超链接伪类实现导航条

需求说明

用提供的图片素材实现如图 7.7 所示的效果，具体要求为：使用超链接伪类，实现鼠标悬浮时显示投影效果（背景图变为 nav_bg.png），要求采用外部样式文件的方式实现。

提示

- div-ul-li 布局并设左浮动，同时为<a>标签设置相应的等宽（84px）。
- 合理设置宽高属性及填充属性，实现所需效果。
- 利用 a:hover 以及 background-image 属性，实现鼠标悬停效果。

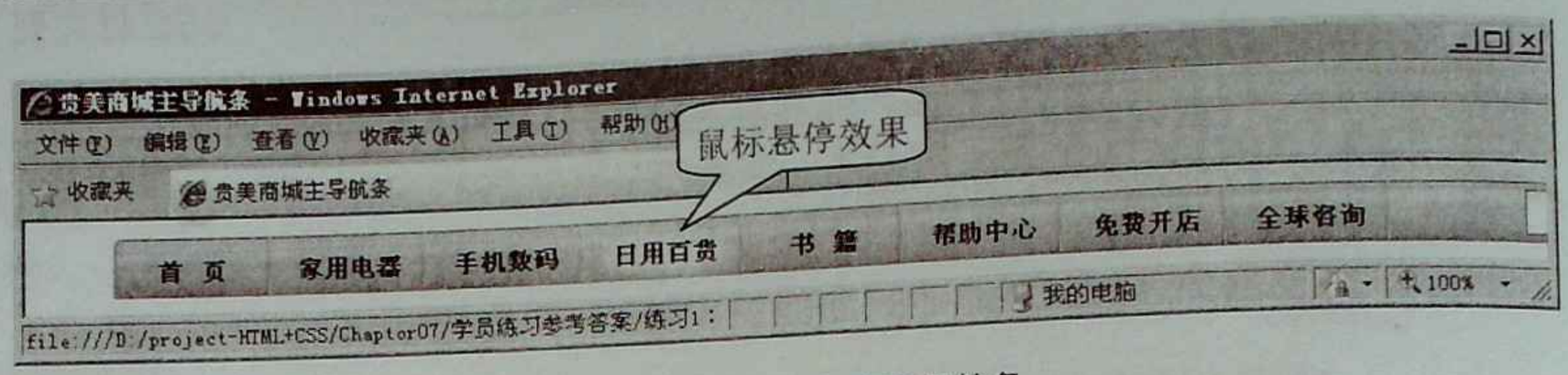


图 7.7 超链接伪类实现导航条

7.3 典型的局部布局

到目前为止，CCS 方面已学习了样式的基本语法、三类样式的应用方式及优先级、常用的 CSS 页面修饰、盒模型以及使用 DIV+CSS 实现页面布局。通过对这些技能的综合应用，我们可以实现页面的整体布局。类似于报纸的排版，目前我们只完成了版面的整体规划，但各版块内部的具体结构，特别是对于某些复杂的版块，还需进一步规划，即局部布局。下面将介绍如何使用 DIV+CSS 实现页面的局部布局，典型的局部结构曾在第 1 章就提及。

- div-ul(ol)-li：常用于分类导航或菜单等场合。
- div-dl-dt-dd：常用于图文混编场合。
- table-tr-td：常用于规整数据的显示场合。
- form-table-tr-td：常用于表单布局的场合。

其中前两类比较常用，也相对复杂，下面就重点讲解这两类局部结构的实现方法。

7.3.1 div-ul-li 局部布局

1. 页面效果

例如，假定要实现贵美商城网站首页顶部（Header）中的导航菜单局部效果，如图 7.8 所示，页面效果要求如下。

- 整个导航菜单横向排列，向右靠齐。
- 菜单文字和图标有一定的空白间隙。
- 菜单文字在菜单项中居中对齐。

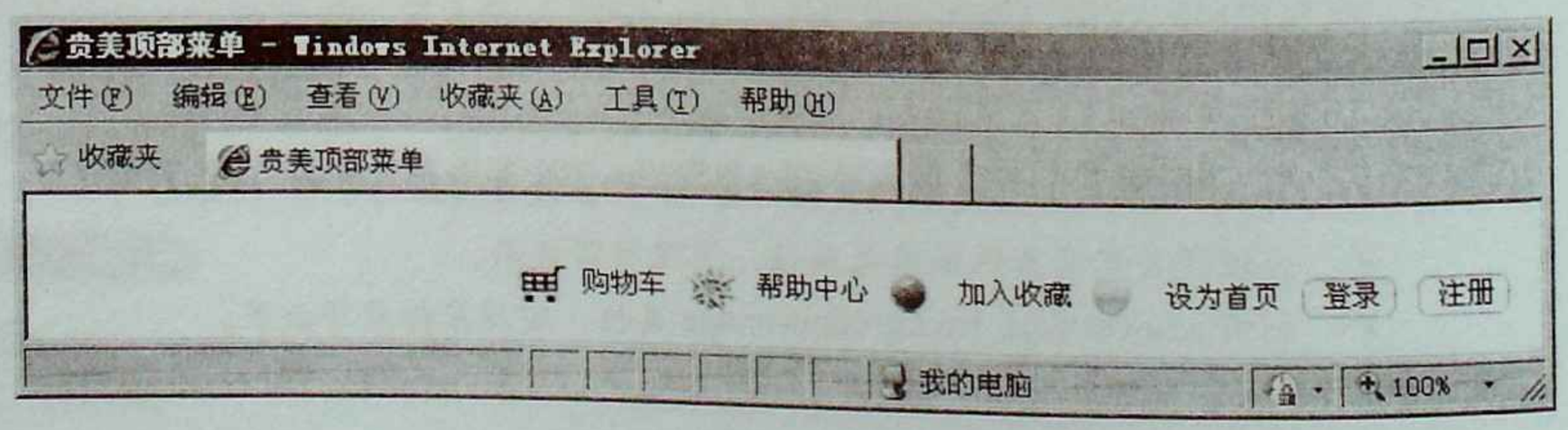


图 7.8 div-ul-li 实现顶部菜单

2. 思路分析

对 HTML 内容结构分析如下。

- 多项菜单并列显示，且不存在父子或包含关系，从语义角度应采用 div-ul-li 实现。
- 图标和文字要求有一定的间隔，图标仅为修饰作用，从语义角度应将图标作为背景修饰而不是内容，另外，文字和图标各占一个<li>（图标的<li>内容为空），方便设置间隔。

对 CSS 样式修饰分析如下。

- 多个小图标，使用背景图的偏移技术（background-position 属性）。
- 浮动：整体居右，则 容器右浮动；图标及菜单文字左浮动。
- 调整宽高及边框属性实现实际的效果。

实现步骤如下。

（1）先建立标签组织结构，为各标签增加类名以区分。

```
<div class="top_menu">
  <ul>
   <li class="pic"></li><li class="text">购物车</li>
   <li class="pic"></li><li class="text">帮助中心</li>
   <li class="pic"></li><li class="text">加入收藏</li>
   <li class="pic"></li><li class="text">设为首页</li>
   <li class="pic">登录</li>
   <li class="pic">注册</li>
  </ul>
</div>
```

因各小图的背景位置偏移量不同，且最后两个图标是按钮形式，所以各小图再按类细分。

```
<div class="top_menu">
  <ul>
   <li class="pic1"></li><li class="text">购物车</li>
   <li class="pic2"></li><li class="text">帮助中心</li>
   <li class="pic3"></li><li class="text">加入收藏</li>
   <li class="pic4"></li><li class="text">设为首页</li>
   <li class="btn">登录</li>
   <li class="btn">注册</li>
  </ul>
</div>
```

（2）为各菜单添加超链接（此步实现简单，这里略过不写代码）。

（3）设置 CSS 样式代码。

- 容器设置右浮动，<li>左浮动，并取消列表样式。

```
.top_menu{float:right;}
.top_menu ul{list-style:none;}
.top_menu ul li{float:left;}
```

- 据图 7.9 确定布局各块大小，统一高度 26px，小图标宽 28px，登录注册宽度 38px。

```
.pic1{width:28px;height:26px;background:url(images/bg.gif) no-repeat;}
.pic2{width:28px;height:26px;background:url(../images/icon.gif)
      no-repeat -28px 0px;}
/* .pic2 至.pic4 省略 */
.btn{width:38px;height:26px; background:url(images/bg.gif) no-repeat;}
```

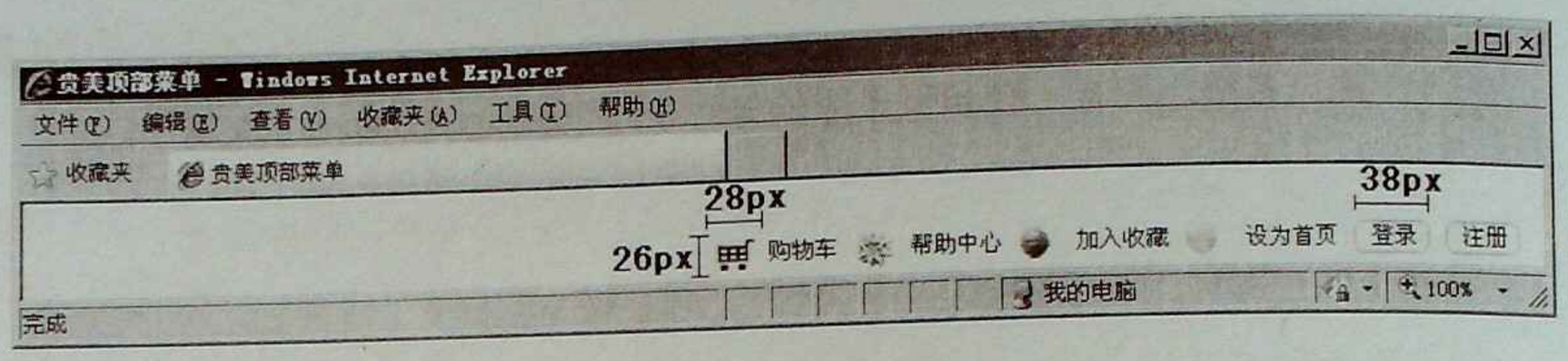


图 7.9 布局中的宽高确定

- 调整背景偏移量，实现小图标正常显示。
- 设置文字大小及菜单文字间填充，菜单文字居中对齐方式，左右填充 5px。

```
.top_menu ul li a{font:12px/26px 宋体;}
.text{padding:0px 5px;text-align:center;}
.btn{padding:0px 5px;text-align:center;}
```

理清思路后，完成的 HTML 文件和 CSS 文件如示例 6 所示。

示例 6

top_menu.html 文件：

```
<html>
  <head>
      <meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
      <title>贵美顶部菜单</title>
      <link rel="stylesheet" type="text/css" href="css/layout.css"/>
  </head>
  <body>
     <div class="top_menu">
       <ul>
         <li class="pic1"></li>
         <li class="text"><a href="#">购物车</a></li>
         <li class="pic2"></li>
         <li class="text"><a href="#">帮助中心</a></li>
         <li class="pic3"></li>
         <li class="text"><a href="#">加入收藏</a></li>
         <li class="pic4"></li>
         <li class="text"><a href="#">设为首页</a></li>
         <li class="btn"><a href="#">登录</a></li>
```

```
        <li class="btn"><a href="#">注册</a></li>
      </ul>
    </div>
  </body>
</html>
```

layout.css 文件（关键代码）：

```
/*逐一定义顶部菜单中的浮动、列表样式,选择器采用区域性的写法,增加代码可读性*/
.top_menu{float:right;}
.top_menu ul{list-style:none;}
.top_menu li{float:left;}
/*定义顶部菜单中的链接样式,代码可以进一步优化*/
.top_menu ul li a{font:12px/26px 宋体;}
.top_menu ul li a:link  {color:#333333;text-decoration: none;}
.top_menu ul li a:visited {color:#333333;text-decoration: none;}
.top_menu ul li a:active  {color:#333333;text-decoration: none;}
.top_menu ul li a:hover   {color:#ff7300;}
  /*定义链接各菜单项的具体样式,代码缩进表示隶属关系,增加代码可读性*/
  .pic1{width:28px;height:26px;background:url(../images/icon.gif) no-repeat;}
  .pic2{width:28px;height:26px;background:url(../images/icon.gif)  no-repeat
   -28px 0px;}
  .pic3{width:28px;height:26px;background:url(../images/icon.gif)  no-repeat
   -84px 0px;}
  .pic4{width:28px;height:26px;background:url(../images/icon.gif)  no-repeat
   -112px 0px;}
  .text{padding:0px 5px;text-align:center;}
  .btn {padding:0px 5px;text-align:center;
  width:38px;height:26px;background:url(../images/icon.gif)  no-repeat  0px
   -25px;}
```

在 div-ul-li 的结构中，<ul>存在默认左右外边距，<li>标签当 list-style 属性有值时存在默认的缩进。应在布局前在 CSS 顶端先设置 ul,ol{margin:0px;padding:0px;}及 li{list-style:none;}。这点在布局中非常重要，有利于写出同时兼容 IE 6.0、IE 7.0、IE 8.0 和 Firefox 的样式代码而不需针对某个浏览器进行修订。

3. 代码复用

从示例 6 中，我们可以很明显地看到很多相似的样式代码，例如，前 4 个图标的样式修饰 pic1 至 pic4，除了偏移量以外，其他完全一致。我们可以把这些共同特征单独提取出来作为一个类，例如 pic，然后再具体设置其他图标的独特样式，这样可以提高代码的复用性并方便维护。

对应的 CSS 代码修改为：

```
/* pic 类样式,用于定义各图标共性的样式*/
.pic{width:28px;height:26px;background:url(../images/icon.gif) no-repeat;}
/* 定义各图标独特的样式,第一个图标无偏移,不需定义 pic1 的独特样式;}*/
```

```
.pic2{background-position:-28px 0px;}
.pic3{background-position:-84px 0px;}
.pic4{background-position:-112px 0px;}
```

在应用样式时，需要同时应用两种类样式，对应的 HTML 代码修改为：

```
<li class="pic pic1"></li>
<li class="pic pic2"></li>
<li class="pic pic3"></li>
<li class="pic pic4"></li>
```

其中“"pic pic1"”表示同时应用两个类样式，注意中间用空格隔开。

而“登录”和“注册”图标也类似，只是宽度不一样，修改思路同前 4 个图标。

对应的 CSS 代码修改：

```
.btn {width:38px;background-position:0px -25px;}
```

对应的 HTML 代码修改：

```
<li class="pic btn">登录</li>
<li class="pic btn">注册</li>
```

而重复的超链接及其伪类的 6 行代码则合并为两行：

```
.top_menu ul li a{font:12px/26px 宋体;color:#333333;text-decoration: none;}
.top_menu ul li a:hover{color:#ff7300;}
```

精简后的代码如示例 7 所示。

示例 7

top_menu.html 文件：

```
<html>
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
   <title>贵美顶部菜单</title>
   <link rel="stylesheet" type="text/css" href="css/layout.css"/>
  </head>
  <body>
    <div class="top_menu">
      <ul>
        <li class="pic1"></li>
        <li class="text"><a href="#">购物车</a></li>
        <li class="pic2"></li>
        <li class="text"><a href="#">帮助中心</a></li>
        <li class="pic3"></li>
        <li class="text"><a href="#">加入收藏</a></li>
```

```
        <li class="pic4"></li>
        <li class="text"><a href="#">设为首页</a></li>
        <li class="btn"><a href="#">登录</a></li>
        <li class="btn"><a href="#">注册</a></li>
      </ul>
    </div>
  </body>
</html>
```

layout.css 文件（关键代码）：

```
.top_menu{float:right;}
.top_menu ul{list-style:none;}
.top_menu ul li{float:left;}
.top_menu ul li a{font:12px/26px 宋体;color:#333333;text-decoration: none;}
.top_menu ul li a:hover{color:#ff7300;}
  .pic{width:28px;height:26px;background:url(../images/icon.gif) no-repeat;}
  .pic2{background-position:-28px 0px;}
  .pic3{background-position:-84px 0px;}
  .pic4{background-position:-112px 0px;}
  .text{padding:0px 5px;text-align:center;}
  .btn {width:38px;background-position:0px -25px;}
```

7.3.2 上机练习

上机练习 2

练习——div-ul-li 局部布局

需求说明

重新实现贵美网站的导航菜单，如图 7.10 所示。

7.3.3 div-dl-dt-dd 局部布局

1. 页面效果

对于网上常见的图文混编结构，如图 7.10 所示，图片和文字显然存在父子或包含关系，文字显然是对商品图片的具体说明，即可以把图片看作“标题”，将后续的多行文字看作“具体的描述”。因此，从语义化的角度，应采用 div-dl-dt-dd 结构进行描述。类似的结构还有带多层次的二级或三级菜单等。

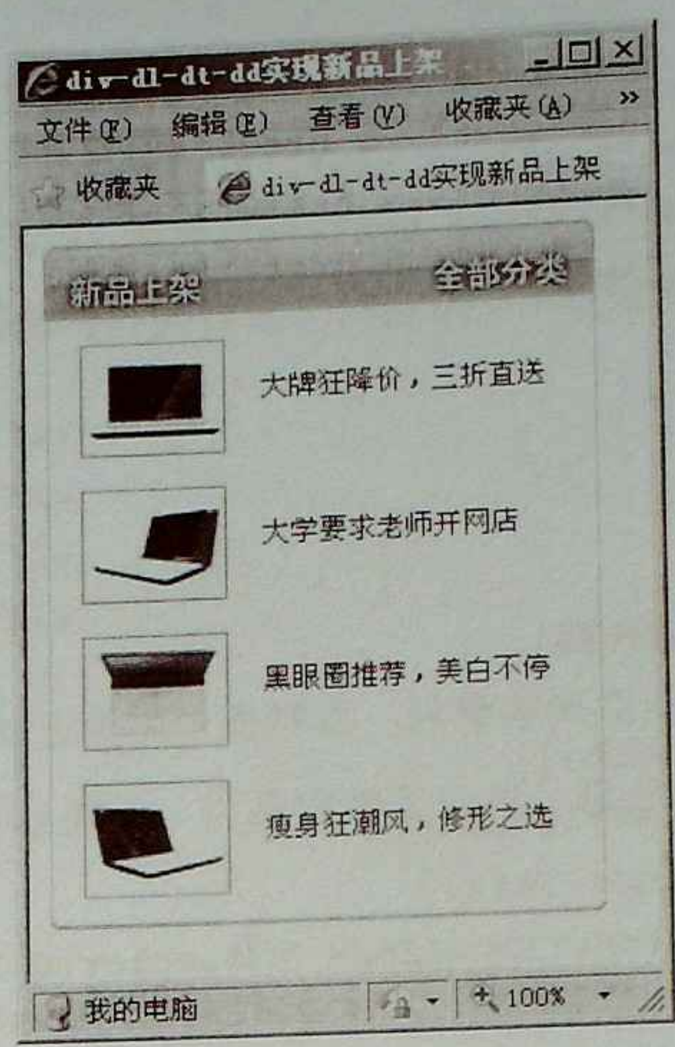


图 7.10　div-dl-dt-dd 实现图文混编

2．思路分析

对 HTML 内容结构分析如下。

- 本例的图文混编结构，图片和文字关系密切，采用 div-dl-dt-dd 结构描述。
- 每行的图文结构都对应一个 dl-dt-dd 结构，易于扩展。
- 根据图片和文字的关系，本例<dt>放图片，<dd>放文字，<dl>作为结构容器。

对 CSS 样式修饰分析如下。

- 浮动：<dd>内的文字和<dt>内的图片排列在同一行，所以应设置<dt>左浮动。
- 调整<dd>宽高与行高实现文字垂直居中，用盒子属性修饰出实际效果。

实现步骤如下。

(1) 编写 HTML 内容结构，即每行一个<dl>，各行又包括左图<dt>和右文<dd>。

```
<div id="right">
  <dl>
    <dt><img src="images/show1.jpg" alt="alt" /></dt>
    <dd><a href="#">大牌狂降价,三折直送</a></dd>
  </dl>
  <dl>
    <dt><img src="images/show2.jpg" alt="alt" /></dt>
    <dd><a href="#">大学要求老师开网店</a></dd>
  </dl>
  <dl>
    <dt><img src="images/show5.jpg" alt="alt" /></dt>
    <dd><a href="#">黑眼圈推荐,美白不停</a></dd>
  </dl>
  <dl>
```

```
    <dt><img src="images/show4.jpg" alt="alt" /></dt>
    <dd><a href="#">瘦身狂潮风,修形之选</a></dd>
  </dl>
</div>
```

(2) 编写 CSS，参考图 7.11 规划 块（#right）的宽高以及<dt>的浮动，并且设置<dt>的高度和<dd>的行高一致，以实现单行文字的垂直居中。

```
#right{width:250px;height:270px;padding-top:32px;}/* div 块的宽高及填充*/
#right dl dt{float:left;width:80px;height:60px; }/*设置 dt 浮动和宽高*/
#right dl dd{width:190px;line-height:60px;} /*设置 dd 浮动、宽度和行高*/
```

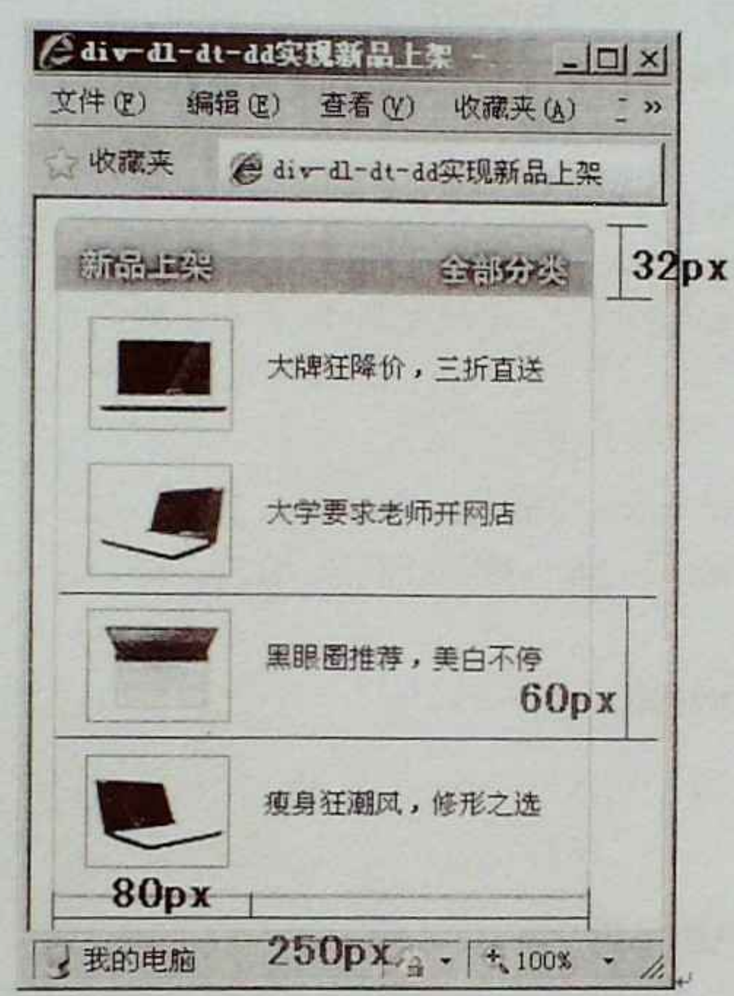


图 7.11　div-dl-dt-dd 实现图文混编

为左边图片设置宽高并修饰边框，图片水平及垂直居中，文字垂直居中。

```
#right dl dt{text-align:center; padding:2px 0px;}/*文字居中对齐、上下少量填充*/
#right dl dt img{
  width:60px;
  height:47px;
  border:1px solid #9ea0a2;  /*设置图片的外边框*/
    vertical-align:middle;  /*设置图片和文字垂直方向居中对齐*/

}
```

完成的 HTML 和 CSS 代码如示例 8 所示。

示例 8

new.htm 文件：

```
<html>
  <head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>div-dl-dt-dd 实现新品上架</title>
<link rel="stylesheet" type="text/css" href="css/layout.css" />
</head>
<body>
  <div id="right">
    <dl>
      <dt><img src="images/show1.jpg" alt="alt" /></dt>
      <dd><a href="#">大牌狂降价,三折直送</a></dd>
    </dl>
    <dl>
      <dt><img src="images/show2.jpg" alt="alt" /></dt>
      <dd><a href="#">大学要求老师开网店</a></dd>
    </dl>
    <dl>
      <dt><img src="images/show5.jpg" alt="alt" /></dt>
      <dd><a href="#">黑眼圈推荐,美白不停</a></dd>
    </dl>
    <dl>
      <dt><img src="images/show4.jpg" alt="alt" /></dt>
      <dd><a href="#">瘦身狂潮风,修形之选</a></dd>
    </dl>
  </div> <!--right end-->
</body>
</html>
```

layout.css 文件（关键代码）：

```
#right{
    width:250px;
    height:270px;
    padding-top:32px;
    background:url(../images/bg.gif) no-repeat;
}
 #right dl dt{
    float:left;
    margin:0px;
    width:80px;
    padding:2px 0px;
    text-align:center;
}
 #right dl dt img{
    width:60px;
    height:47px;
    border:1px solid #9ea0a2;
    veritical-align:middle;
}
```

```
 #right dl dd a{
    font:12px/47px 宋体;
    color:#333;
    text-decoration: none;
}
#right dl dd a:hover{color:#ff7300;}
```

在 div-dl-dt-dd 的结构中，<dl>存在默认上下外边距，<dd>标签存在默认的左外边距。应在布局前在 CSS 顶端先设置 dl,dd{margin:0px;padding:0px;}。这点在布局中非常重要，有利于写出同时兼容 IE 6.0、IE 7.0、IE 8.0 和 Firefox 的样式代码，而不需针对某个浏览器进行修订。

小结

通过前面各示例的学习，用到了各种 CSS 选择器相关的常用符号，如“.”为类名标识符，“#”表示 id 标识符，“,”区隔多个选择器共用一套样式，“ ”空格区分上下级元素，“:”为伪类符号等，在网上还可以看到更复杂的组合。为了方便阅读网上的 CSS 代码，请参考如下 HTML 代码，理解其他复杂组合方式的含义。多选择器的常用符号及组合如表 7-2 所示。

```
<div id="nav">
  <ul>
    <li class="pic pic1"></li>
    <li class="text">购物车</li>
    <li class="pic pic2"></li>
    <li class="text">帮助中心</li>
  </ul>
</div>
```

表 7-2　多选择器的常用符号及组合

	符号	中文	示例	含义
基		空格	div ul{list-style:none;}	内的 元素样式
本	,	逗号	div,ul{text-align:center;}	和 元素采用相同样式
符	#	id 标识符	#nav{width:100%;}	id 为“nav”的元素样式
号	.	类标识符	.pic{background:url(bg.gif);}	类名为 pic 的元素样式
	:	冒号	a:hover{#ff0;}	<a>标签的 hover 伪类样式
组	li .	标签+类	li.pic{width:28px;}	类名为 pic 的 标签样式
合	div #	标签+id	div#nav{text-align:center;}	id 为“nav”的 标签样式
	# .	id+空格+类	#nav .pic{border:1px;}	id 为“nav”元素内的 pic 类样式
	# . ,	id+空格+类+逗号	#nav .pic,#nav .text{height:26px;}	id 为“nav”元素内的 pic 和 text 类，都采用相同样式

7.3.4 上机练习

上机练习 3

练习——div-dl-dt-dd 局部布局

需求说明

重新实现贵美网站的新品上架，如图 7.11 所示。

本章总结

- 超链接伪类有以下几种。
  - a:link
  - a:hover
  - a:visited
  - a:actived
- 样式有三种应用方式：外部样式表、内部样式表、内部样式，其中外部样式实现了内容和样式的彻底分离，是主流的应用方式。
- 各类样式的优先级如下。
  - 行内样式表>内部样式表>外部样式表。
  - ID 选择器>类选择器>标签选择器。
- 典型的局部结构包括如下几种。
  - div-ul(ol)-li：常用于分类导航或菜单等场合。
  - div-dl-dt-dd：常用于图文混编场合。
  - table-tr-td：常用于规整数据的显示场合。
  - form-table-tr-td：常用于表单布局的场合。
- div-ul-li 局部结构的注意要点如下。
  - 先确定页面内容的组织结构。
  - 利用宽高、浮动、方框属性实现布局细节。
  - 用途广泛，常见于并列或一图一文参差并列结构。
- div-dl-dt-dd 局部结构的注意要点如下。
  - 先分析适用场景，确定<dt>、<dd>放置什么内容。<dt>可以用于放置标题或图片，<dd>可以用于放置普通文本。
  - 设置<dt>浮动，保持<dd>行高与<dt>图片高度一致，合理设置<dt>填充属性。
  - 常用于图文混编及二级导航菜单等场合。
- 选择器涉及如下常用符号。
  - 空格，表示上下级元素，即“某个标签（对象）内的某个元素的样式”。
  - 逗号，表示多个选择器共用一套样式，即“多个元素（对象）都采用这种样式”。
  - “.”为类名标识符，“#”表示 id 标识符。

本章作业

一、选择题

1. 对超链接伪类的理解，说法正确的是（　　）。

A. a:link 表示未单击状态

B. a:hover 表示单击后的状态

C. a:visited 表示已访问状态

D. a:active 表示单击后的状态

2. 关于各类应用样式的方式，说法正确的是（　　）。

A. 外部样式表具有最高的可复用性及便于维护，因此，具有最高优先级

B. 行内样式应该尽量不用，因为违背样式与内容分离的原则

C. 样式优先级的主要根据是内容在文档结构中的先后顺序

D. 内部样式表完全实现了多页面的代码与样式的分离

3. 以下对 div-ul-li 局部布局的理解，正确的是（　　）。

A. 比较常用，适用于无层次关系的并列结构。例如，并列的多个产品类别

B. div-ul-li 与现实中常用的典型业务逻辑非常吻合。例如，二级导航条与图文混编

C. 实现中通常先分析特点，规划区域，总结样式类别，最后确定宽高、浮动等

D. 在网页布局中，完全不需要其他结构，全部用 div-ul-li 实现

4. 关于 div-dl-dt-dd 默认的样式，说法错误的是（　　）。

A. <dl>有默认的上下外边距

B. <dd>没有默认左外边距

C. div-dl-dt-dd 的布局适合于一图多文混编或者二级导航菜单

D. 在样式书写前，预先设置 dl,dd{margin:0px;}可以清除相关标签的默认设置

5. 关于优先级的说法，说法正确的是（　　）。

A. 对于各类选择器，优先级最低的是标签选择器

B. 对于各类应用方式，优先级最高的外部样式

C. 当外部样式表、内部样式表、行内样式叠加到同一个页面元素时，页面元素的最终样式是行内样式设置的样式，和其他两类定义的样式毫无关系

D. 当多个样式叠加多同一个页面元素时，有冲突的样式取决于样式的优先级规则，但不冲突的样式将全部应用到页面元素

二、简答题

1. 请描述超链接的特殊性，及其各种状态下的默认样式及相关伪类。

2. 简述局部布局中的常用结构、应用场合及布局时的注意要点。

3. 用 div-dl-dt-dd 经典布局和所提供的素材，实现贵美商城中的“疯狂抢购”页面，效果如图 7.12 所示。提示：

- 用 div-dl-dt-dd 实现布局，注意先设置<dl>和<dd>外边距及填充为 0px。
- <dl>设置左浮动，并调节各<dl>宽度，在总大小 517px 中每行放置三个。
- 使用背景位置（background-position）的偏移截取 bg.gif 的相应区域，并为 设置适当填充量。
- 设置图片和文字水平居中。

图 7.12　div-dl-dt-dd 实现疯狂抢购页面

4. form-table-tr-td 布局结构比较简单，请尝试使用这种结构实现注册页，并进行样式修饰，如图 7.13 所示，具体要求如下。

- 所有的单行输入框和多行文本域都要求实现细边框的效果，单行输入框高 16px 宽 180px。
- 提示文字，如“登录名”、“密码”等右对齐，并设右填充 10px。

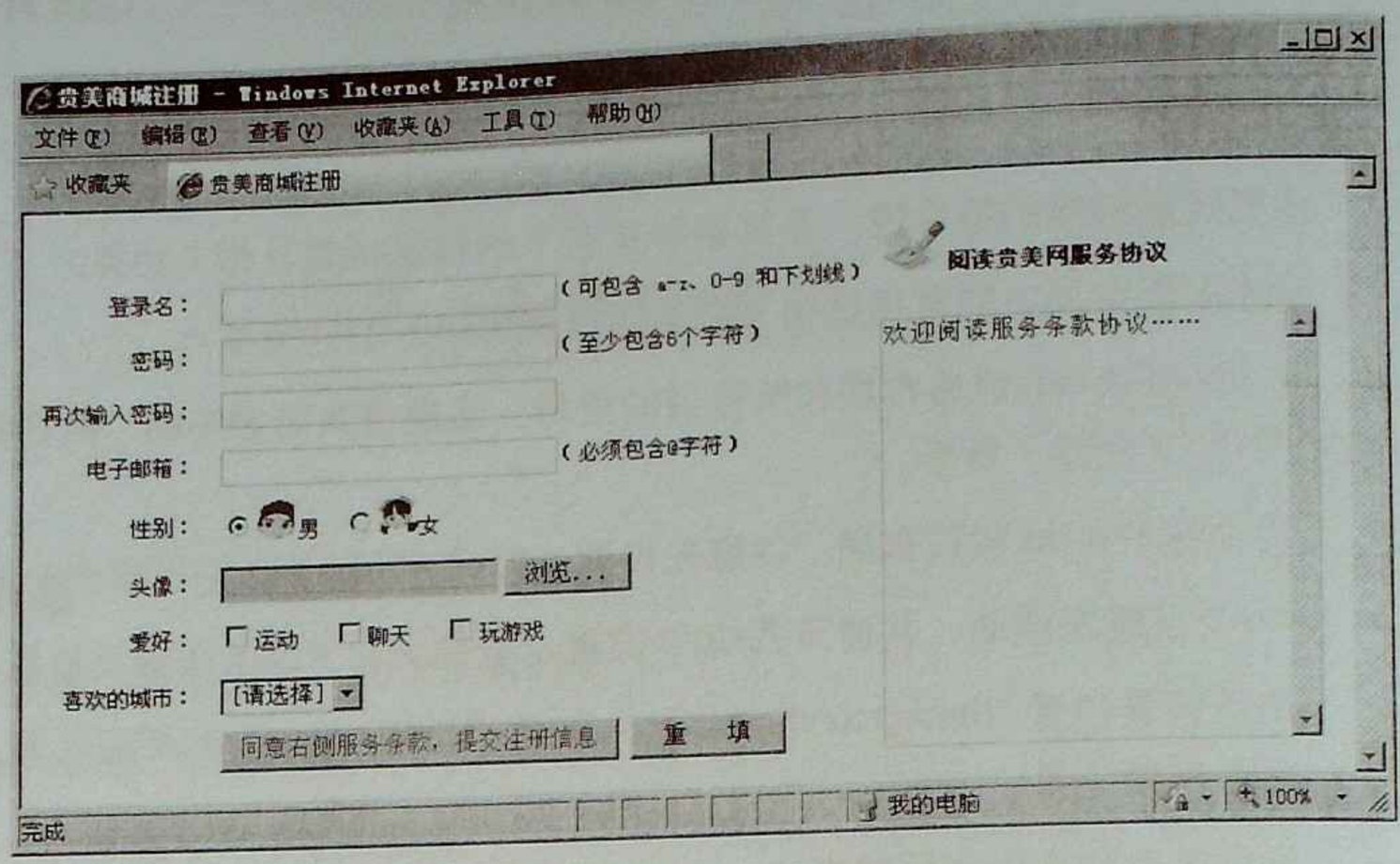


图 7.13　form-table-tr-td 实现注册页

5. 实现如图 7.14 所示的效果，注意多选择器的组合应用及其语法，具体效果如下。

- 定义三个<p>标签。
- 第一个<p>标签含三个<h2>，设第一个<h2>颜色为 green。
- 第二个<p>标签含五个<h3>，设所有<h3>为 green。其中设第一个<h3>ID 为"myH3"，样式是 20px 大小，背景色为#ff00ff。
- 第三个<p>标签含两个<h1>标签，颜色为 red。

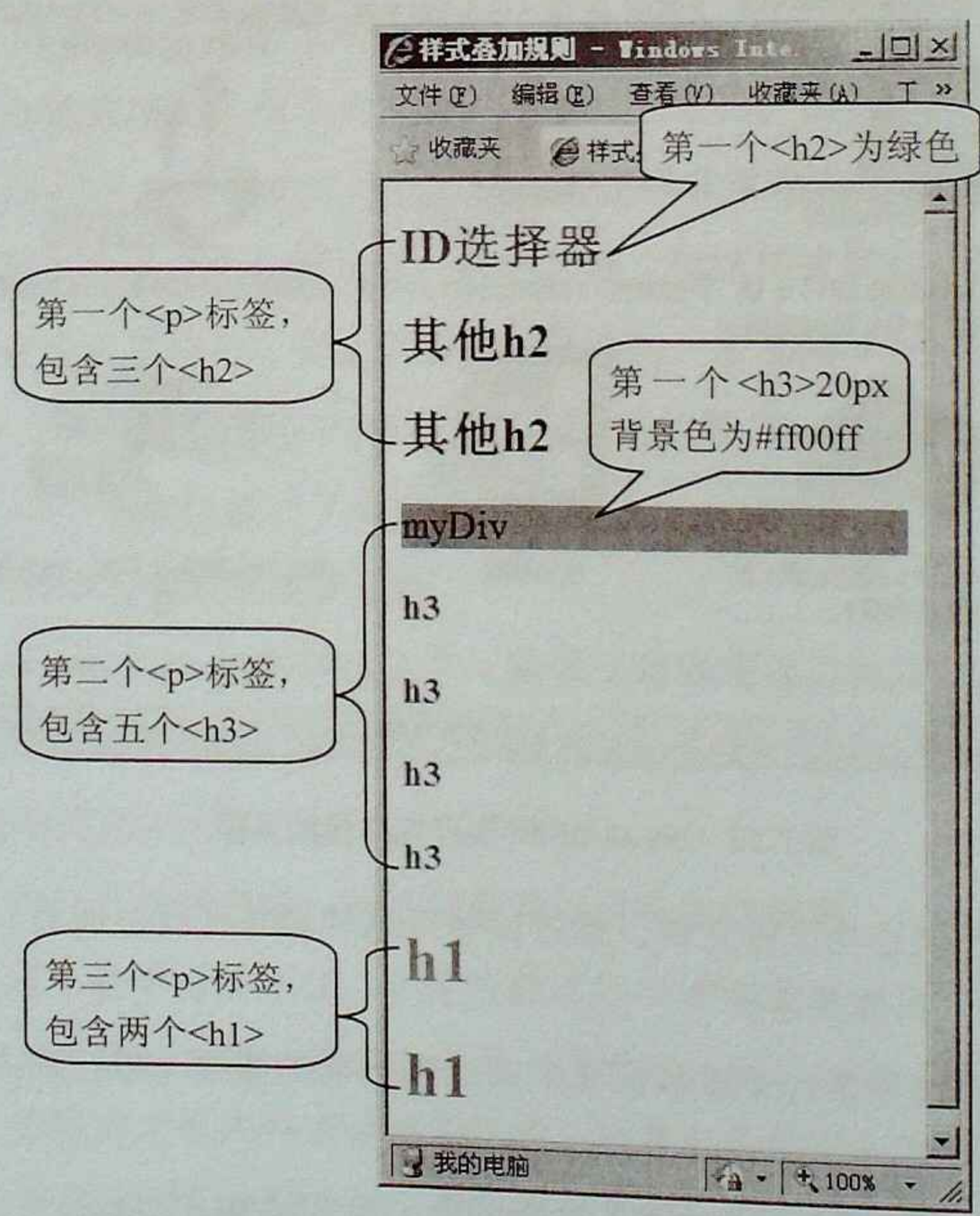


图 7.14　多选择器常用符号

第 8 章

指导学习：活动页的局部布局

◈本章工作任务

- 制作六人行网站活动页的主要版块

◈本章技能目标

- 使用 DIV+CSS 实现页面的局部布局
- 使用绝对定位实现层的叠加效果

8.1 复习串讲

8.1.1 难点突破

表 8-1 中列出了本学习阶段的难点，这些技能你都掌握了吗？

如果还存在疑惑，请写下你对这个技能感到疑惑的地方。我们可以通过教材复习，从网上查找资料学习，和同学探讨，或者向老师请教等方法突破这些难点。直到这个技能已经全部掌握，并在“是否掌握”一栏中画上“√”。这些技能是学习后面技能的基础，一定要在继续学习前全部突破。

表 8-1 难点记录表

难点	感到疑惑的地方	突破方法	是否掌握
HTML 的各类标签			
XHTML 1.0 的基本规范			
CSS 的基本语法			
CSS 常用属性			
CSS 背景的偏移技术			
盒子模型			
DIV+CSS 布局			

如果在学习中遇到其他难点，也请填写在表中。

8.1.2 知识梳理

1. 知识梳理：HTML 部分

HTML 主要负责页面内容的组织结构，HTML 部分的知识体系如图 8.1 所示，图中展示了完整的 HTML 知识体系，我们可以借助这个图理清头脑中的知识体系架构。

2. 知识梳理：CSS 部分

CSS 主要负责页面的表现（显示），CSS 知识体系如图 8.2 所示。CSS 部分是前几章内容的难点，特别需掌握 CSS 常用属性和 DIV+CSS 布局。

在知识体系图中，✔图标表示我们学过该知识；?图标表示我们部分学习过该知识或者学习过相关知识，但需要巩固和补齐；图标表示我们要通过自学学习这个知识。

DIV+CSS 布局已成为主流的页面布局方式，根据企业调研结果，B/S 结构的项目开发越来越重视客户端技术的应用，HTML+CSS 已成为基本的应用技能，我们必须通过多练多实践，扎实掌握相关技能。

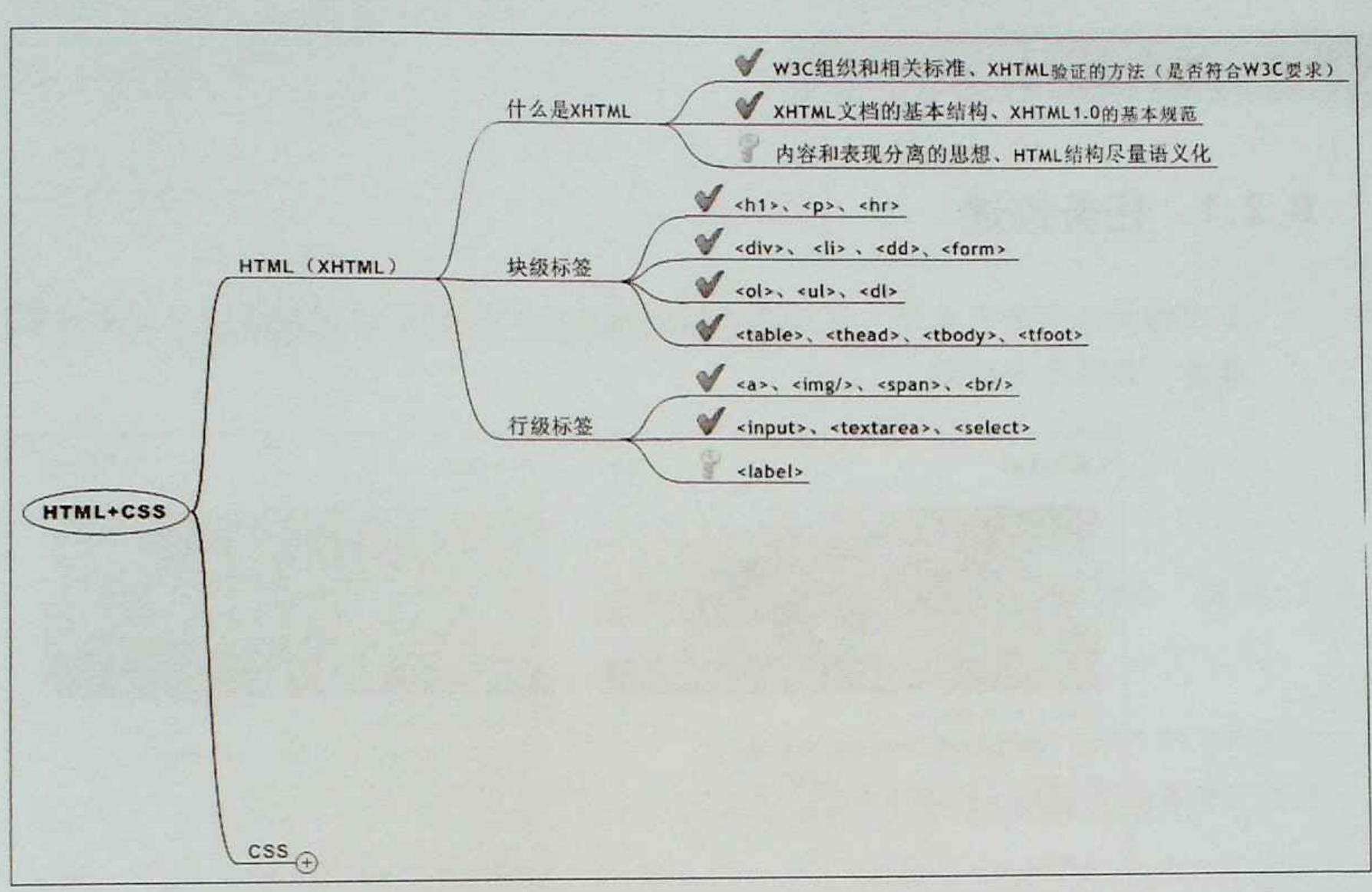


图 8.1　HTML 部分知识体系

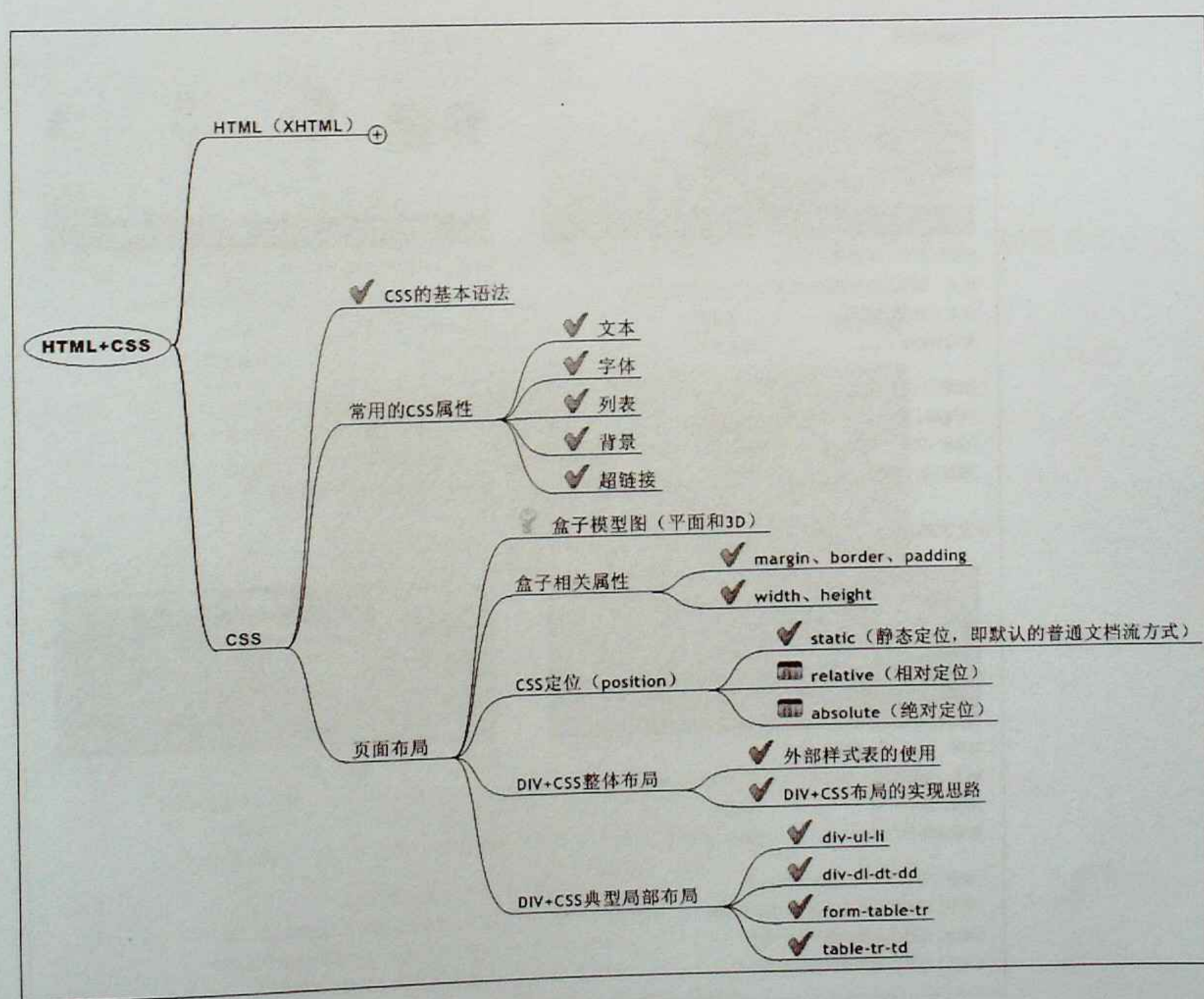


图 8.2　CSS 部分知识体系

8.2 综合练习

8.2.1 任务描述

本次综合练习的任务是：完成六人行网站中活动页的活动宣传版块，为后续项目案例做准备，如图 8.3 所示。

图 8.3 六人行活动页的活动宣传版块

8.2.2 练习

分阶段完成练习。

阶段 1：练习——实现页面的局部布局

需求说明

- 使用 DIV+CSS 布局活动页面版块。
- 页面一共分为六部分内容，分别是都市生活、户外•旅行、运动健身、理财•英语•学习、文艺演出、公益•其他。

提示

- 六个部分的内容，每两个部分在同一行内，可以使用 div-ul-li 布局实现，也可以使用 div-dl-dt/dd 布局实现。
- 通过设置 float 浮动的方式来实现页面布局。

阶段 2：指导——制作圆角矩形

需求说明

页面的六个部分中，除了标题外，其他内容均显示在一个圆角矩形里面，如图 8.4 所示。

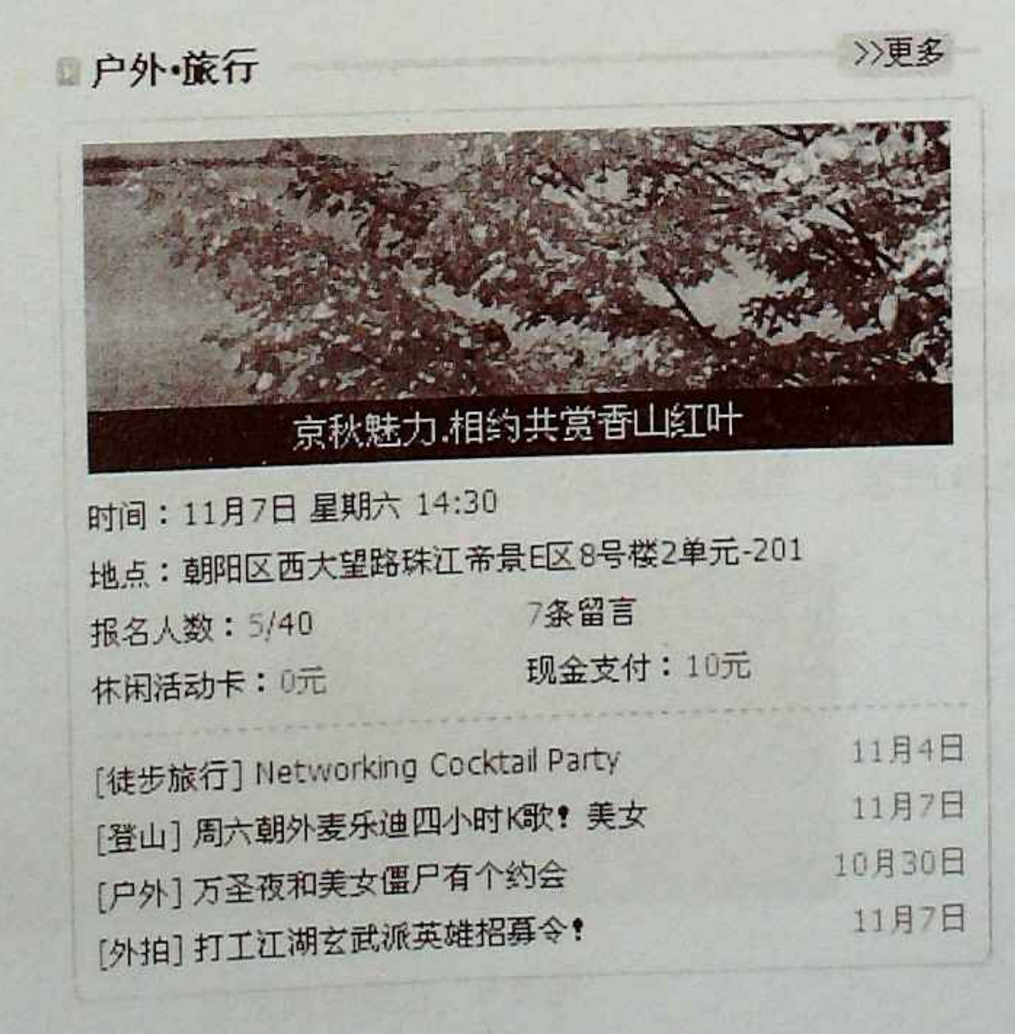


图 8.4　户外・旅行

实现思路及关键代码

(1) 从页面上可以看到六个部分的圆角矩形的宽度均一致，因此可以写一个宽度一样的圆角矩形样式，高度自己设置。

(2) 使用背景样式制作圆角矩形时，使用两个背景图片（rounded-left.gif、rounded-right.gif），分别表示左侧两个圆角和左边线、右侧两个圆角以及上下和右边线，设置圆角矩形的宽度，高度自己设置就可以了。

阶段 3：练习——编写页面内容

需求说明

- 根据提供的页面效果图及图片素材，把页面中的内容添加到圆角矩形中。
- 编写样式美化页面内容，超链接文本颜色为蓝色，当鼠标悬停到超链接上时，文本颜色为橙色。
- 标题后的日期、报名人数、留言条数、休闲活动卡、现金支付等文本颜色均显示为橙色。
- 在每一版块的分类标题信息与上面的现金支付等信息内容之间使用一条虚线隔开，如图 8.4 所示。

阶段 4：指导——制作背景透明效果

需求说明

自学老师提供的“学习参考资料”，实现如下滤镜效果。

- 每一版块都有一个图片，图片的下方均有一个文字说明，文字说明的背景使用滤镜制作透明效果，如图 8.5 所示。
- 文本颜色为白色，当鼠标悬停到文本上时颜色为橙色。

图 8.5 背景透前效果

实现思路及关键代码

(1) 文本背景是一个层，通过设置层的绝对定位及上一级层的相对定位来实现层在图片的下方显示。绝对定位在第 6 章的作业曾要求自学过。

(2) 设置背景层的颜色为黑色，通过滤镜代码“filter:alpha(opacity=50);”设置背景为半透明，由于此代码只适用于 IE 浏览器，因此还需通过代码“opacity:0.5;”设置背景的半透明来兼容其他的浏览器。

第 9 章

贵美商城网站（一）

◈本章工作任务

- 制作贵美商城网站首页的布局

◈本章技能目标

- 使用 DIV+CSS 制作简单的页面布局
- 根据网站开发流程制作网站

本章单词

请在预习时学会下列单词的含义和发音，并填写在横线处。

1. category：_____
2. sidebar：_____
3. upright：_____
4. commodity：_____
5. show：_____
6. automation：_____
7. relationship：_____
8. management：_____
9. promote：_____
10. enterprise：_____

本章简介

通过前面章节的介绍，我们系统学习了 HTML 和 CSS 的基本知识和技能，理解了盒子模型并初步尝试实现页面的简单布局。从本章开始，将用 3 章的篇幅介绍如何综合应用所学知识和技能，完成一个实际的网站项目开发——贵美商城网站，它包括从拿到美工人员设计的效果图开始，到最后发布并通过各种浏览器的测试。本章重点介绍网站开发流程、如何利用 Dreamweaver 搭建站点、实现贵美商城网站首页的布局及各版块的页面效果。本章以及后续两章的关键都是多练、多动手，只有通过反复实践，才能深入理解所学知识，灵活应用各种技能。

9.1 网站开发流程

创建一个商业网站，需要了解网站开发的整个流程。开发流程主要包括需求分析、网站制作、网站发布以及测试。需求分析的目的是了解客户“想做什么”；网站制作主要是完成“如何做”；网站发布以及测试主要是将站点发布到局域网（Intranet）或广域网（Internet）上，供企业内部或全球所有的用户访问。

9.1.1 需求分析

简单地说，需求分析就是通过多次沟通、访谈、参观等方式，知道客户“需要什么”。如果投入大量的人力、物力、财力，最终开发出的网站却不符合客户的要求，那么项目所有的投入都是徒劳的。因此，彻底、深入的需求分析非常有必要，并且在后续的网站制作、开发过程中也必须牢记客户的需求。

需求分析的方法和技巧很多，关键是理清如下几个问题。

1. Why——为什么需要建站？即明确组建网站的目的

常见的建站目的如表 9-1 所示，要么是增加利润，要么是传播信息或观点，另外还有一种特殊的目的是作为应用程序的用户界面。应用程序的界面一般包括两类：一种是桌面窗口形式（例如 Windows 中的记事本、Office 等），一般也称为 C/S 应用（Client/Server，客户端/服务器模式）；另一种是采用网页的形式（例如表 9-1 例举的企业内部信息系统），一般也称为 B/S 应用（Browser/Server，浏览器/服务器模式），这两种应用在后续的课程中都将深入讲解。

表 9-1　常见的建站目的

目的	举例	说明
增加利润	电子商务网站	通过网络销售，降低客户服务成本，增进品牌意识
传播信息或观点	企业产品宣传网站、政府宣传网站	不是进行销售，而是宣传
作为应用程序的用户界面	企业内部信息系统 OA、CRM、ERP 等，后续学期将涉及这些业务	OA（Office Automation），即办公自动化系统；CRM（Customer Relationship Management），即客户关系管理系统；ERP（Enterprise Resource Planning），即企业资源计划系统

显然，创建贵美商城网站的目的是第一种：增加利润。贵美商城的业务背景是：随着网上交易安全性方面的逐渐完善，网上购物已逐渐成为年轻人消费的时尚。同时，通过网上在线销售，可以扩展企业的销售渠道，提高公司品牌的知名度，降低企业的销售成本。

2. Who——谁来访问？即确定网站的目标受众

一般需要分析目标受众的年龄、兴趣爱好、经济状况等方面的问题。例如，贵美商城网站主要针对喜欢时尚购物的年轻人，年龄一般以 18～30 岁为主，针对这个年龄段的特点，网站提供的功能和服务需符合现代、时尚、便捷的特点。另外，美工人员在设计网站界面草图时，也需考虑时尚、明快的设计样式，包括整个网站的色彩、Logo、图标设计等。

3. What——内容是什么？即确定网站的内容

内容决定一切，内容价值决定了访问者的去离。我们需要结合公司的业务背景，设计相关内容的页面，充分展示网站的价值，让访问者尽快获取到自己需要的内容。例如，根据贵美商城的业务背景，设计的主要页面有以下几个。

- 首页（index.htm）：包括网站导航、商品广告、商品分类、版权声明等内容。
- 某类商品展示页（catlist.htm）：展示某类商品下的具体商品。
- 具体商品的详细介绍页（info.htm）：包括商品的名称、性能描述等。
- 购买商品页（buy.htm）：在线购买的商品情况，即购物车。
- 注册页（register.htm）：注册为网站会员。
- 登录页（login.htm）：使用账号登录网站。
- 帮助页（help.htm）：客户服务方面的帮助信息。

明确用户的需求后，为了进一步确认客户的需求，一般还需要美工人员使用 Photoshop 等工具，设计相应的页面效果图并提交给客户确认，有的教材也称为网站原型。这中间一般需要一个反复确认并完善的过程。例如，图 9.1～图 9.7 是美工人员设计的效果图。

图 9.1　首页效果图

图 9.2　商品分类页效果图

图 9.3 商品详细介绍页效果图

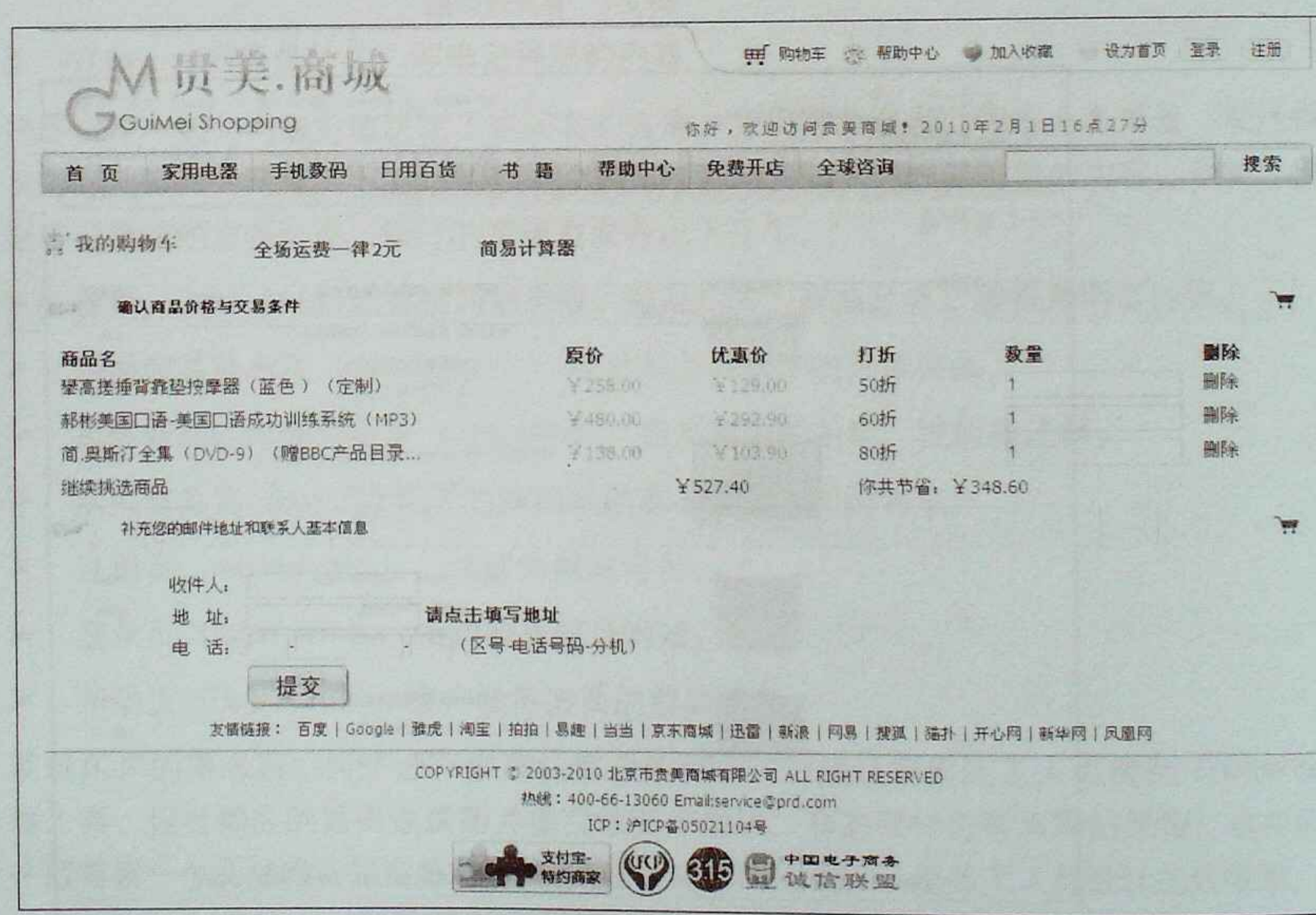


图 9.4 购买页效果图

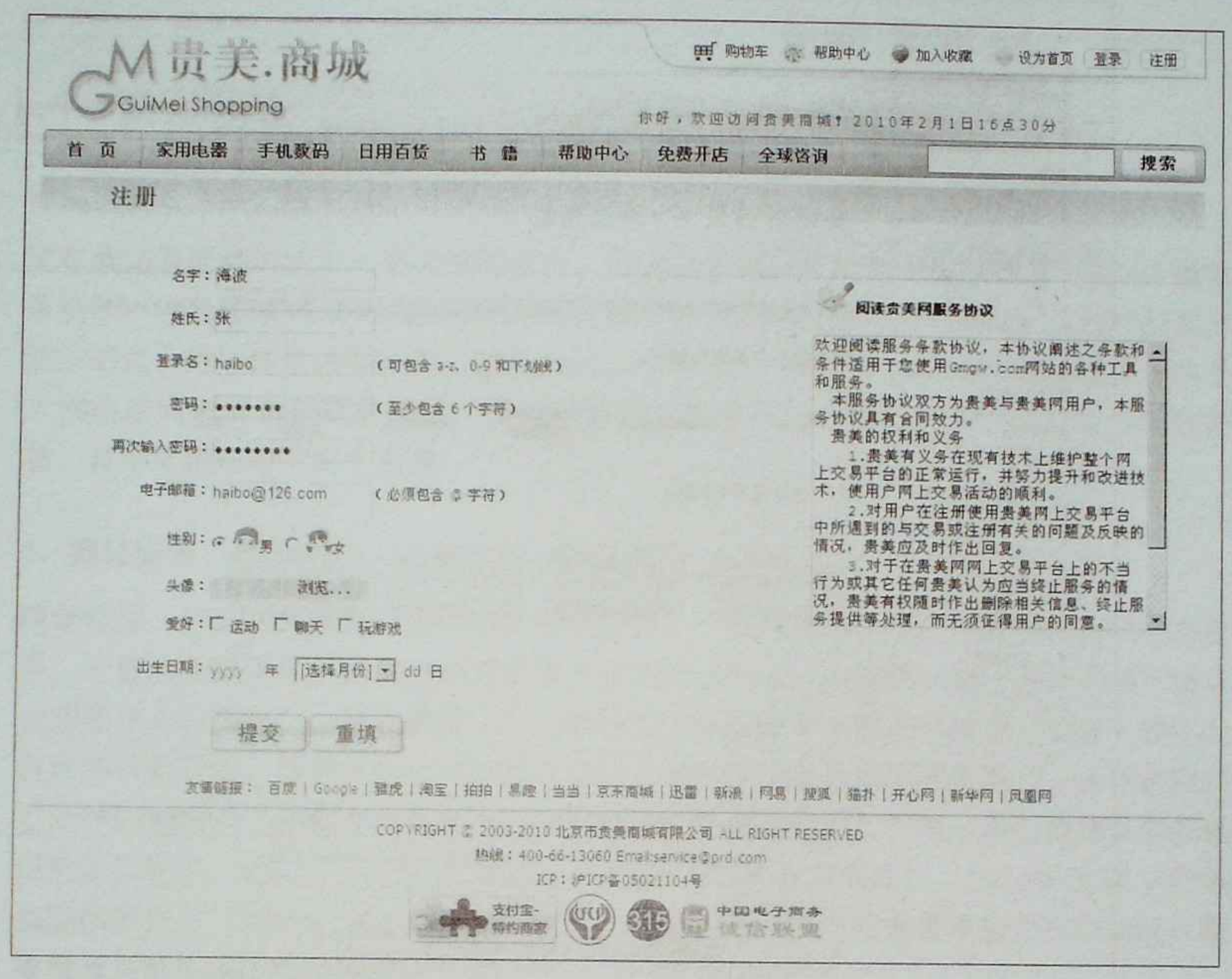


图 9.5　注册页效果图

图 9.6　登录页效果图

图 9.7　帮助中心页效果图

9.1.2　网站制作

完整的网站制作包括以下两个过程。

- 前台页面制作：拿到美工的效果图以后，编写 HTML、CSS，将效果图转换为*.html 网页，其中包括图片收集、页面布局规划等工作。
- 后台程序开发：实现网站和数据库的连接、内容动态添加或修改等后台功能。

本课程重点讲解前台页面的制作，后台页面的开发将在后续的 JSP 或 ASP.NET 相关课程学习。

9.1.3　测试网页

测试网页包括测试以下内容。

- 测试网页是否符合需求。
- 测试并修复网页可能出现的漏洞（Bug）。
- 根据客户使用的浏览器的类型和版本，测试能否同时兼容。一般要求能同时兼容 IE 6.0、IE 7.0、IE 8.0 及 Firefox 3.5，并符合 W3C XHTML 1.0 标准。

9.1.4 发布网站

网站经测试之后，就可以放在服务器上发布，这样，能让更多的人知道你的网站。

发布网站有两种方式，一种是本地发布，即通过本地计算机来完成，在 Windows 操作系统中，一般通过 IIS 来构建本地 Web 发布平台，这种发布方式只能让局域网中的用户访问站点；另一种是远程发布，即登录到 Internet 上，然后利用 Internet 服务商提供的个人网络空间来发布创建的网站。不过，利用这种发布方式要先申请一个网站空间和域名，其中常用到如下两个名词。

1. 网站空间

简单地说，网站空间就是存放网站内容的空间，一般对应为网站服务器的某个文件夹目录。一般企业可以自己购买一台服务器来提供网站空间，但费用太高（具体市场行情可以查阅中关村在线），并且需要 24 小时开机，并配备专人维护。所以，企业一般不会自己购买服务器，而是将自己网站的内容放在网站空间提供商的服务器上，这样的网站空间一般也称为“虚拟主机空间”。为了支持不同类型的网页，网站服务器需要安装相应的应用程序，例如：IIS（支持 ASP 及 ASP.NET 等类型的网页）、Tomcat(支持 JSP 等类型的网页)、Apache（支持 PHP 等类型的网页）等，所以在申请虚拟主机空间时，需要留意它能支持的网页类型。

2. 域名

将本地的网站内容通过 FTP 等方式上传到虚拟主机空间后，还需要一个可供他人访问的 URL 地址（即网址）。可以使用 IP 地址的形式来表示，例如 http://202.106.118.20/index.html，但这种地址很难记忆，不方便用户访问，所以需要类似 http://www.sohu.com/index.html 的访问网址，即域名。域名同样需要申请，网上有很多网站空间和域名申请的服务提供商，例如万网等。

9.2 制作首页布局

熟悉了网站的开发流程后，下面可以开始制作首页了。制作首页前，我们还需要利用 Dreamweaver 创建站点，即搭建整个网站的大致结构。

9.2.1 使用 Dreamweaver 创建站点

在实际的网站开发中，我们常用 Dreamweaver 工具辅助开发。这个软件提供代码智能提示、视图预览、项目管理、站点管理等强大功能。下面我们就以贵美商城为例，演示如何在 Dreamweaver 中创建网站，采用的版本是目前较流行的 Dreamweaver CS4，其主界面如图 9.8 所示。

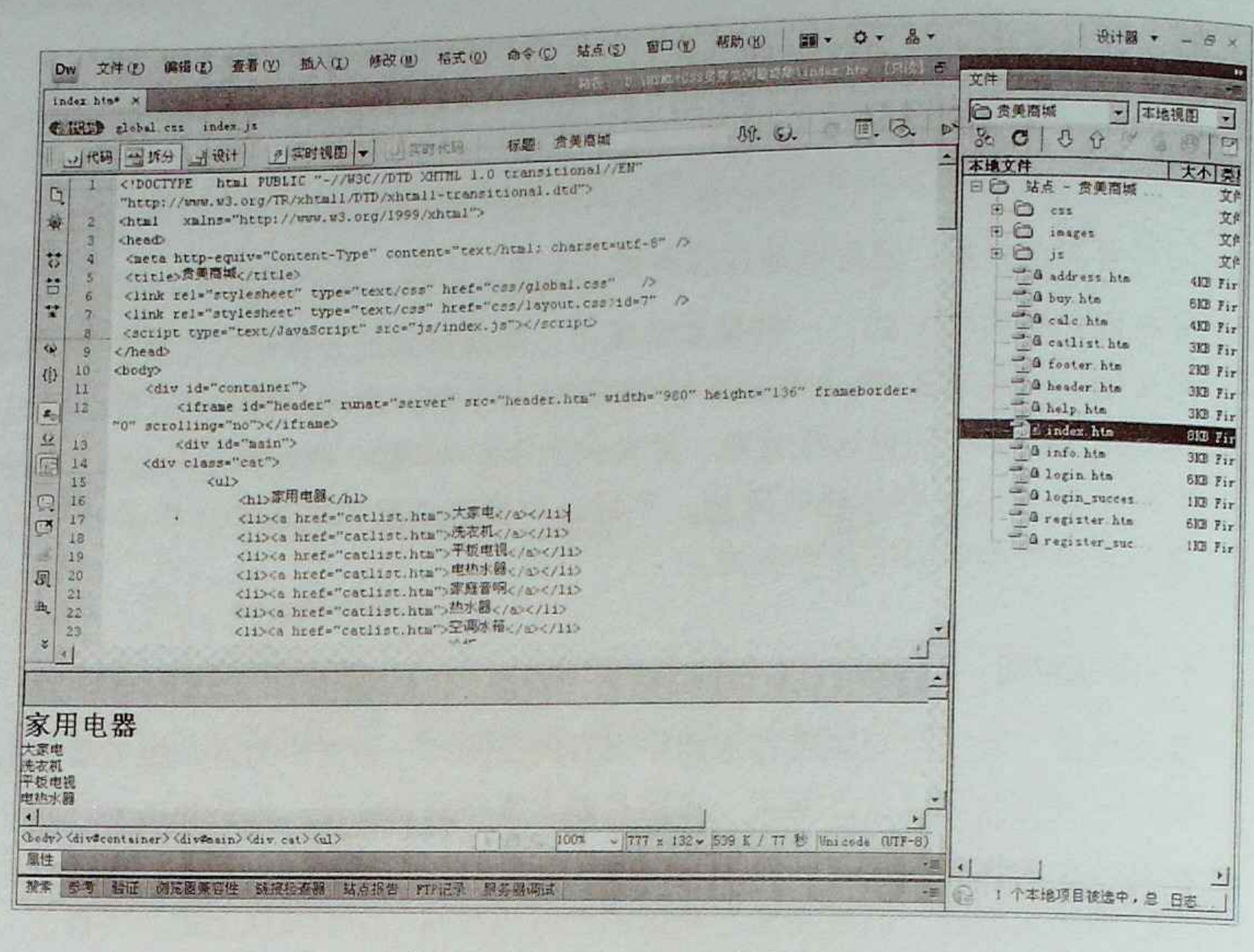


图 9.8 Dreamweaver 主界面

1. 建立站点及相关的目录结构

1）建立站点

具体步骤如下。

（1）启动 Dreamweaver CS4，依次单击“站点(S)”→“新建站点(N)”，输入站点名称及网址，如图 9.9 所示。

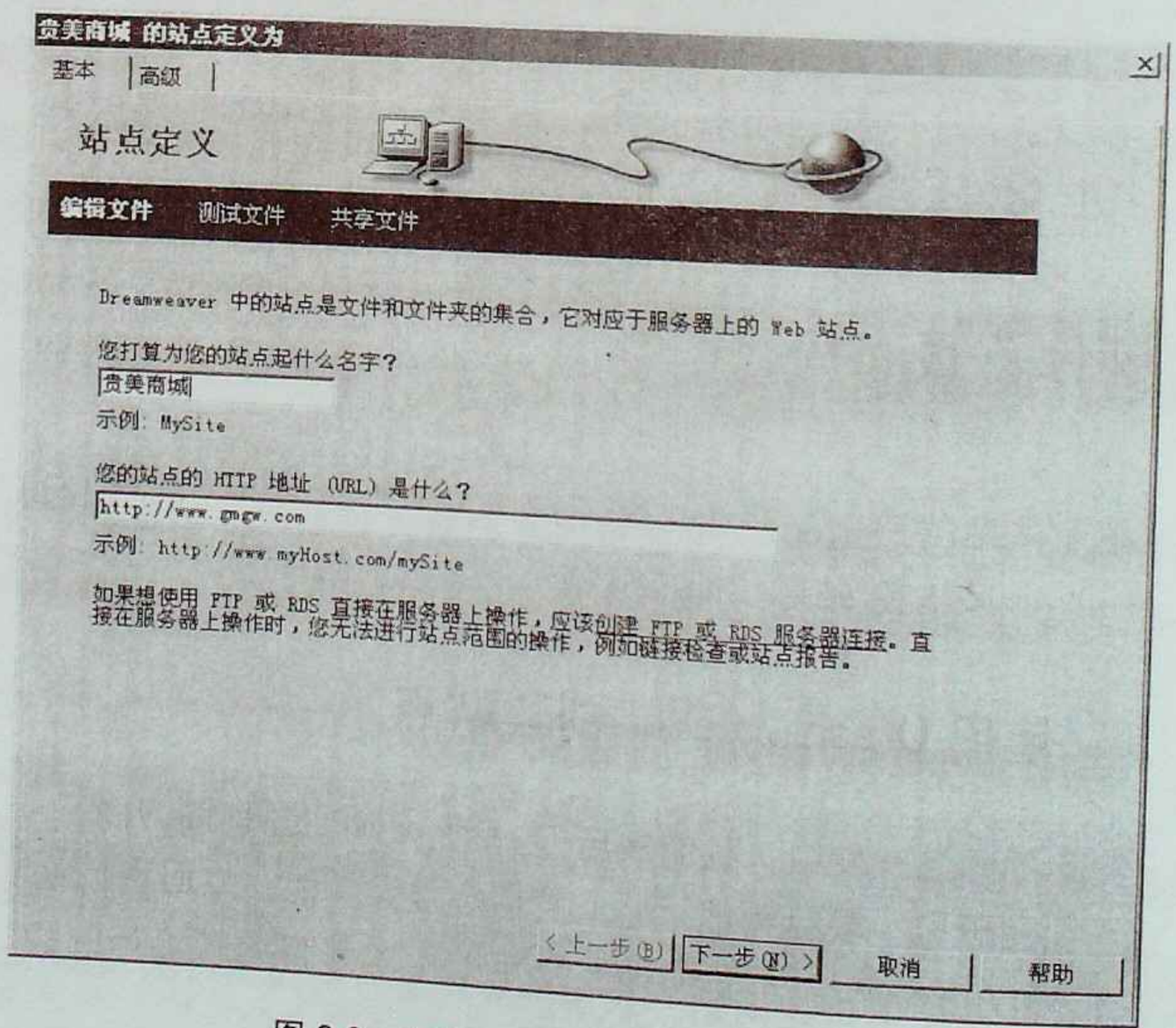


图 9.9 在 Dreamweaver 中建立站点

(2) 单击“下一步(N)”，选择程序采用的服务器技术，如图 9.10 所示。因后续课程才介绍服务器端的技术，所以本课程暂时选择“无”。

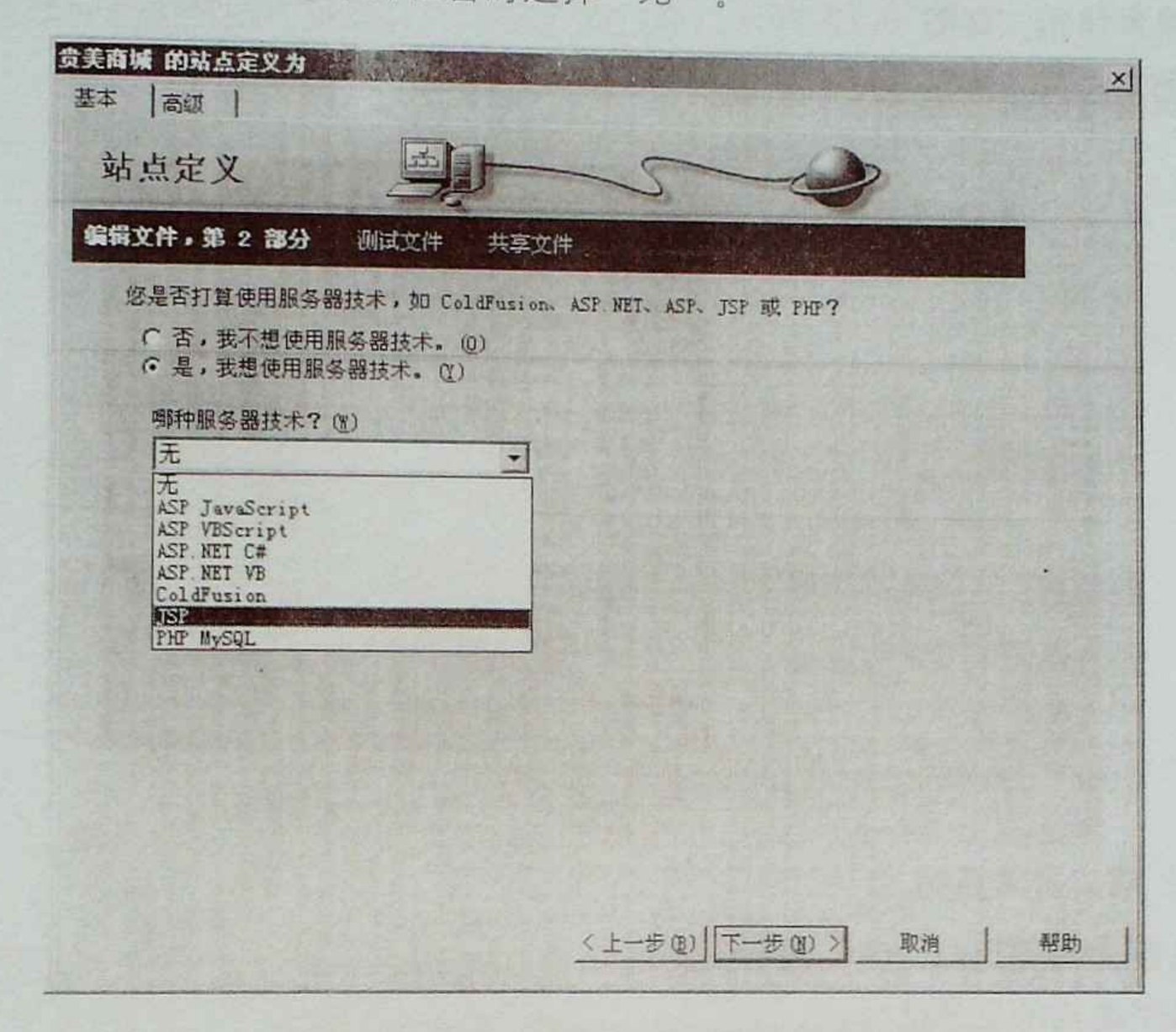


图 9.10　服务器端技术的选择

(3) 单击“下一步(N)”，单击右下方目录图标，在弹出的“选择站点 贵美商城 的本地根文件夹”窗口中单击对应的目录，假设为“d:/guimei/”，单击“选择(S)”，如图 9.11 所示。

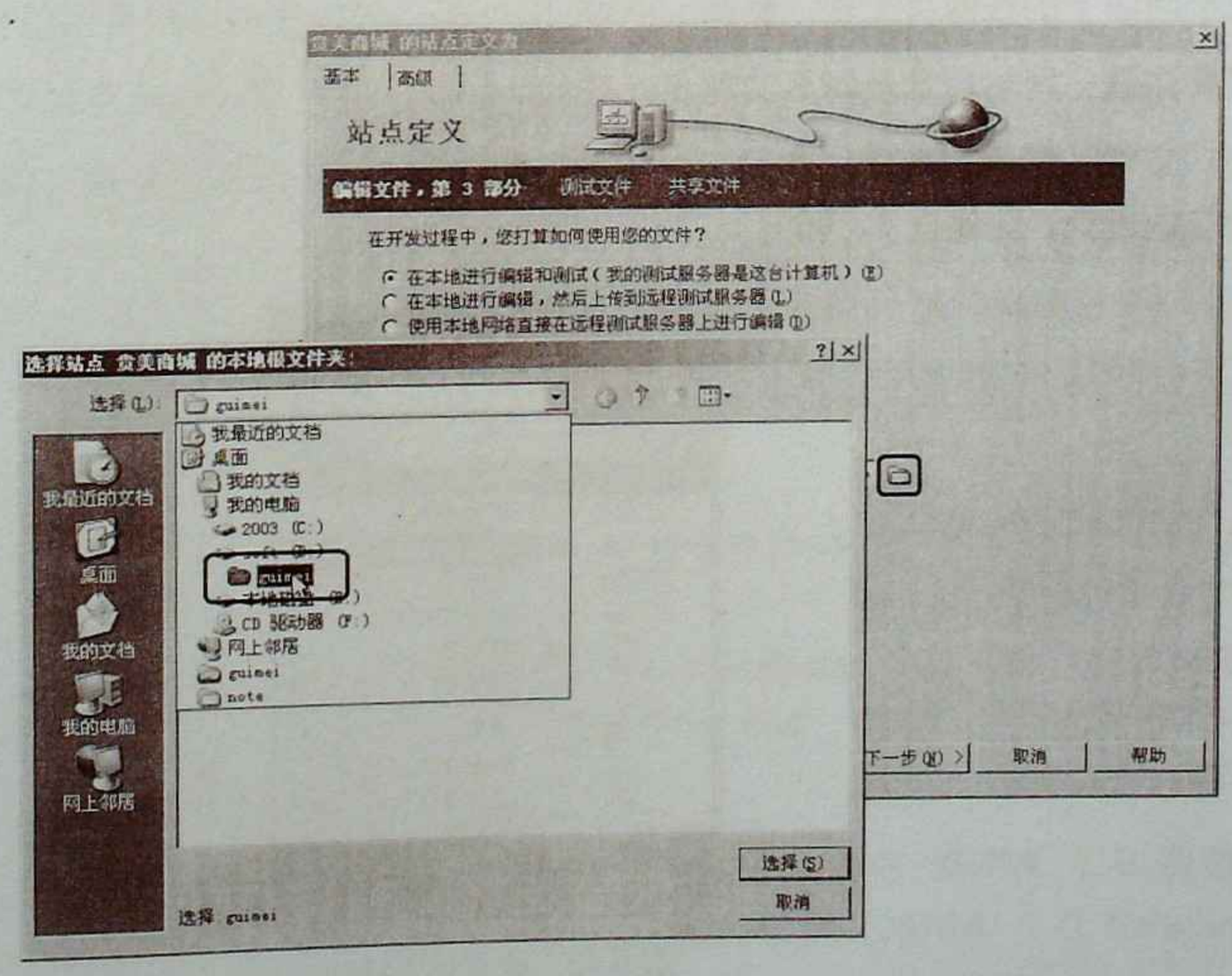


图 9.11　选择本地根目录

(4) 单击“下一步(N)”，选择“我将在以后完成此设置。”，单击“下一步(N)”→“完成”，即创建完毕。建完站点后，可以单击菜单“窗口”→“文件”，查看网站的目录结构，如图 9.12 所示。通过这个窗口，可以很方便地进行网站文件的各种操作，例如添加、删除、打开等。

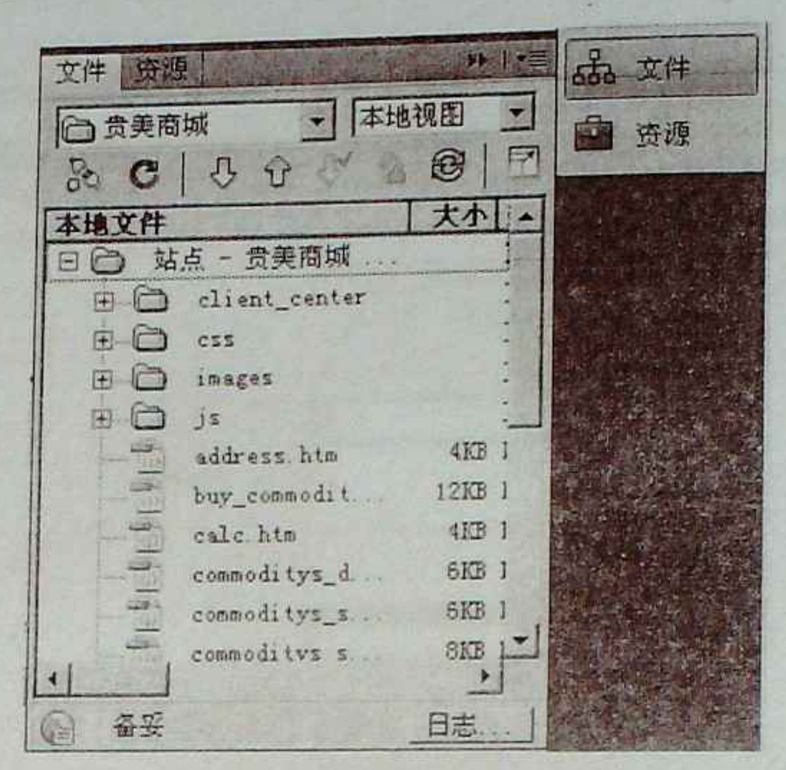


图 9.12 站点文件及资源管理

2）建立目录结构

在制作各网页前，我们需要先确定整个网站的目录结构。对于中小型网站，一般会创建如下通用的目录结构。

- images 目录：存放网站的所有图片。
- js 目录：存放 JavaScript 脚本文件，后缀名为*.js，JavaScript 将在后续课程中介绍。
- css 目录：存放 CSS 文件，即前面曾提及的外部 CSS 文件，实现内容和样式的分离。

对于网站下的各网页文件，例如，index.html 等一般放在网站根目录下。

需要注意，网站目录及网页文件名一般都为小写，并采用代表一定含义的英文命名。

在“文件”管理视图中，单击“站点 - 贵美商城”，选择“新建文件夹”，依次添加相应的目录，如图 9.13 所示。

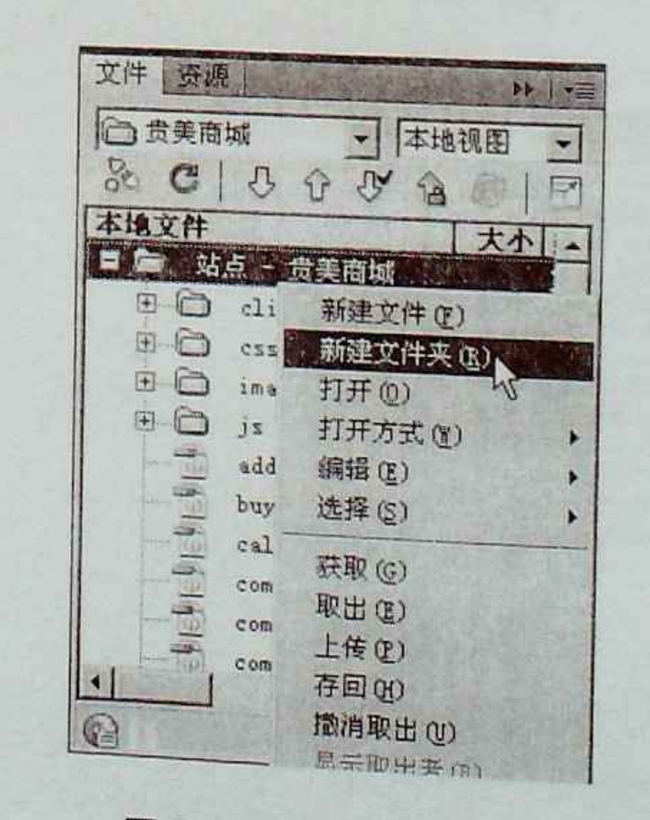


图 9.13 建立目录结构

9.2.2 分析页面布局

毫无疑问，除非特殊应用外，DIV+CSS 布局绝对是首选。我们分析整个贵美商城网站的布局，可以看出该网站是个典型的三行三列的布局结构，如图 9.14 所示。

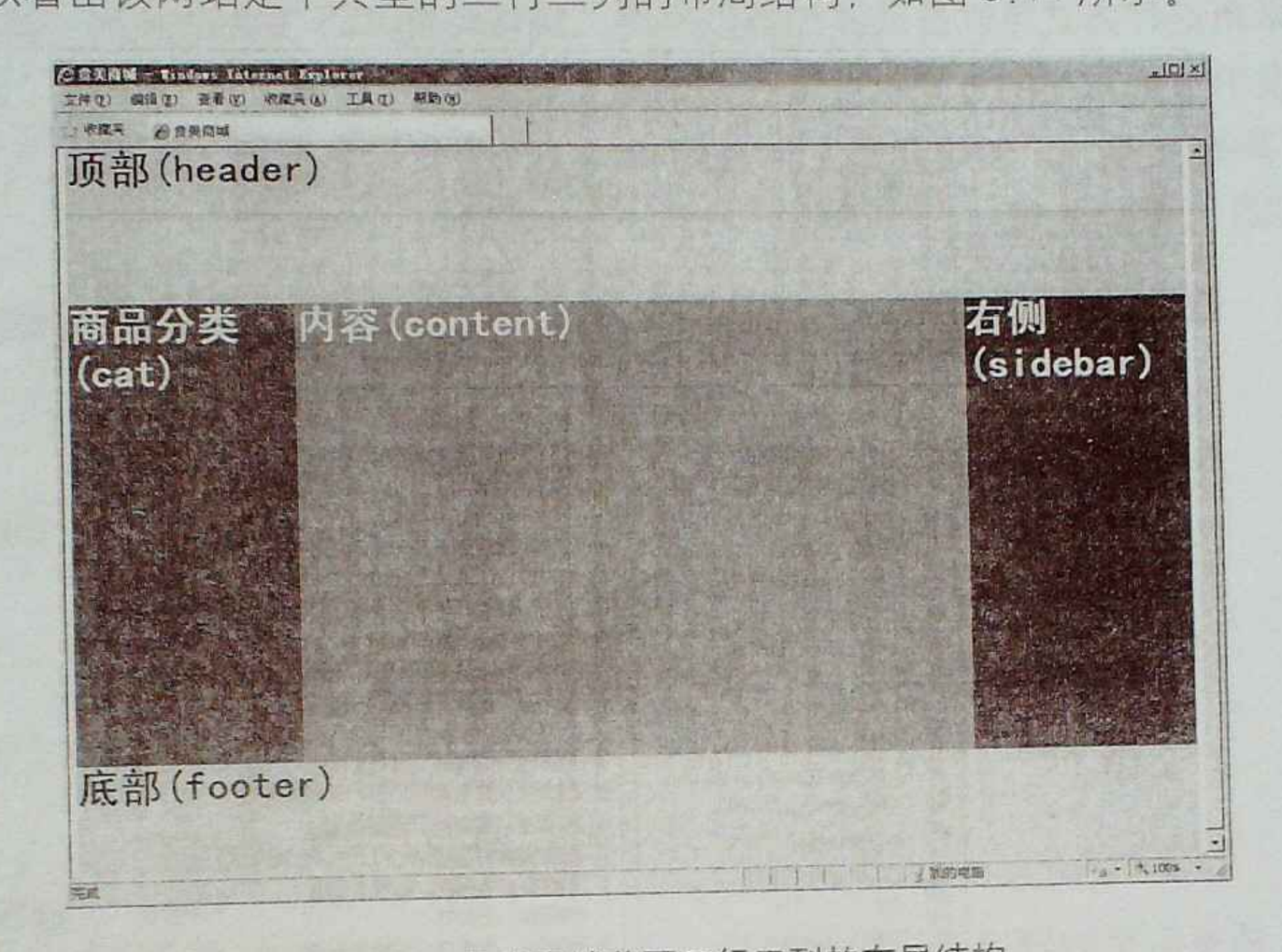


图 9.14　贵美商城首页三行三列的布局结构

命名规范

- Web 页面的开发目前还没有类似 Java 或 C#方面的开发规范，但对于以上各 div 块，习惯采用如下命名规则（这些命名甚至已列入 W3C 的 HTML 5 草案的规范中）。

  container：容器
  header：顶部
  footer：底部
  main：页面主体
  logo：企业标志图片
  copyright：版权信息
  content：页面主体内容
  sidebar：侧边栏

- 各 div 块的样式可以采用 ID 或 class，一般大的 div 区块（黑体标识的 4 个）用 ID。

下面将使用 Dreamweaver 创建首页（index.htm）及相关的 CSS 文件，操作步骤为：单击菜单“文件（F）”→“新建（N）”，如图 9.15 所示。其中，“页面类型”和“布局”选择默认值，需要注意以下三点。

- 可以通过选择右下方的“文档类型”，确定页面遵守 XHTML 1.0 规范的级别，一般选择“XHTML 1.0 Strict”严格级别或默认的“XHTML 1.0 Transitional”过渡级别。

- 单击“附加 CSS 文件”右侧的链接图标，按提示设置链接的 CSS 文件。设置为 css 目录下的“layout.css”，“layout.css”文件可以提前在网站目录结构中创建。
- 将文件保存在“d:\guimei”目录下，命名为“index.htm”。

采用这种方式可以自动添加类型声明，链接外部 CSS 文件的代码，除此之外，在实际开发中，更常用的方式是直接在网站目录结构（图 9.13）中，单击“站点-贵美商城”→“新建文件”，创建网页文件（“index.htm”），然后再人工修改文档类型，添加链接 CSS 文件的代码。

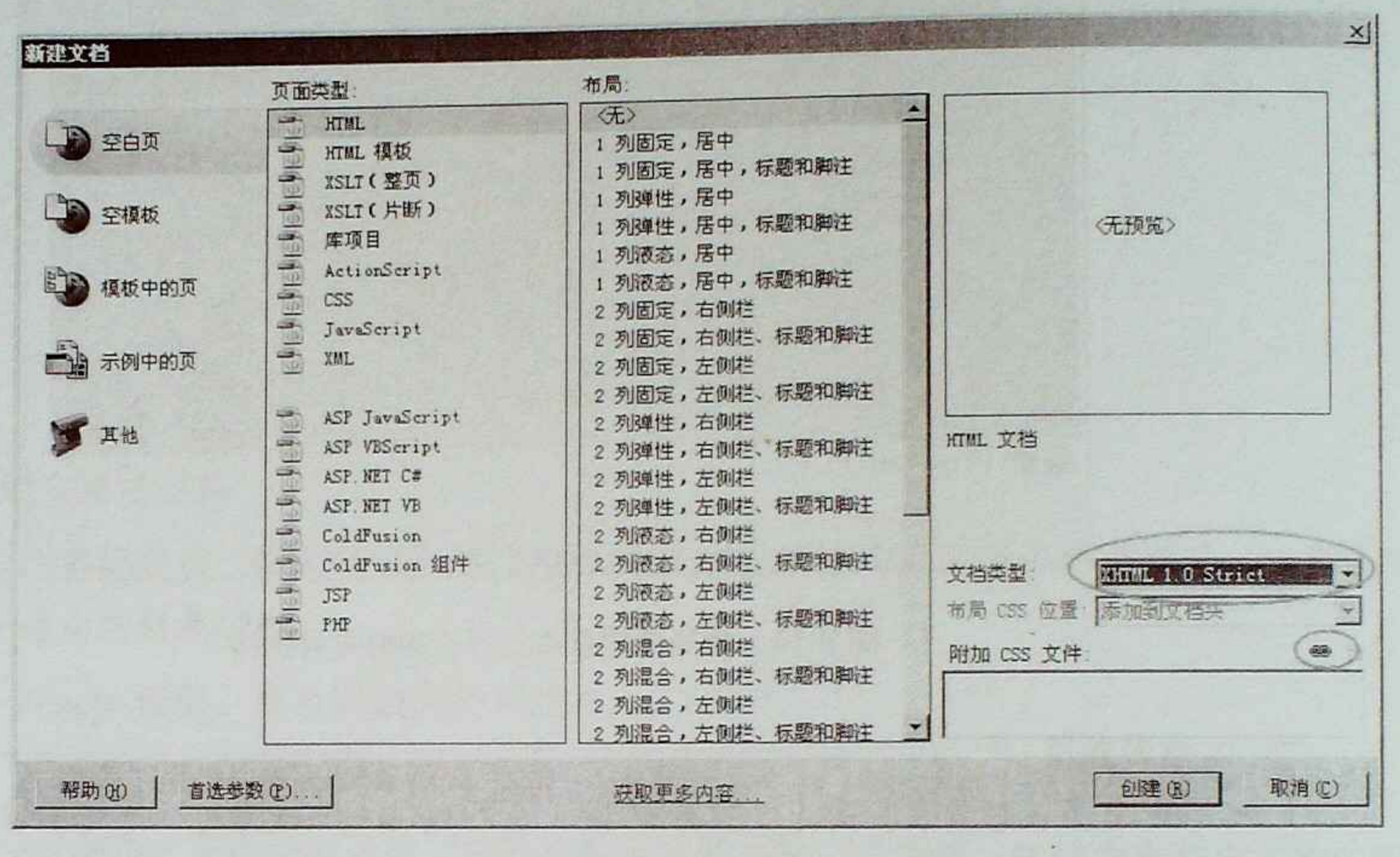


图 9.15　Dreamweaver 新建页面与 CSS 文件

9.2.3　实现布局

首先我们分析图 9.14 的页面特点：三行三列，纵向三行，中间行里又分为左、中、右三列。

采用如下技术。

- 用 标签分区并设置 ID 标识。
- 浮动：中间三块都左浮动。
- 调整宽高及盒子属性实现实际效果。

实现思路

（1）先建立 HTML 组织结构。第一层为三个 ，分三行，中间行的 又包含三个并列的三列 。

```
<div id="header">顶部(header)</div>
<div id="main">
```

```
    <div class="cat">商品分类(cat)</div>
    <div class="content">内容(content)</div>
    <div class="sidebar">右侧(sidebar)</div>
</div>
<div id="footer">底部(footer)</div>
```

(2) 在外部加上顶级容器，以便进行统一设置及整体加载。

```
<div id="container">
  <div id="header">顶部(header)</div>
  <div id="main">
    <div class="cat">商品分类(cat)</div>
    <div class="content">内容(content)</div>
    <div class="sidebar">右侧(sidebar)</div>
  </div>
  <div id="footer">底部(footer)</div>
</div> <!--container end-->
```

(3) 添加 CSS 样式代码（编辑 css/layout.css 文件）。

① 设置中间三列左浮动。

```
.cat,.content,.sidebar{float:left;}
```

② 因为是大块布局只需确定各块的大致大小，具体数值可在实现时灵活更改。例如：

纵向三行宽度 100%，高度分别为 150px、400px、100px；中间三列高度 100%，宽度各为 20%、60%、20%。

顶级容器，设置宽度 980px，水平居中，并设置<body>标签的默认内外边距为零。

```
body{margin:0px;padding:0px;}
#container{width:980px;margin:0px auto;}
 #header {width:100%;height:150px;}  /*注意缩进,表示各块之间的包含关系*/
 #main{width:100%;height:400px;}
   .cat,.content,.sidebar{float:left;width:20%;height:100%;}
   .content{width:60%;}
 #footer{width:100%;height:100px;}
```

③ 设置字体、大小样式及背景色，利于识别。

对于布局中的各块，可以使用边框或背景色标识，因为边框占据一定的宽度，为了精确起见，本文采用背景色标识，添加页面内容后，再取消各块的背景色。

在 Dreamweaver 中，HTML 页面中引用的 CSS 文件会出现在文件标签的下方。直接单击即可访问和编辑，比较方便。如图 9.16 所示，单击“源代码”和“layout.css”即可在两者之间切换。

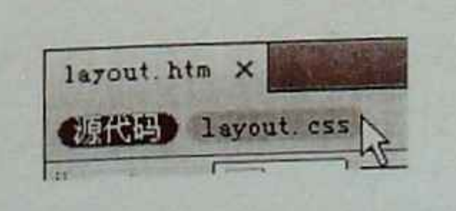


图 9.16 Dreamweaver 中 HTML 与 CSS 文件切换编辑方便

根据上述思路，完成的 HTML 和 CSS 代码如示例 1 所示，对应的页面效果如图 9.14 所示。

示例 1

index.htm 代码：

```
<!DOCTYPE  html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
   <title>贵美商城</title>
   <link rel="stylesheet"  type="text/css" href="css/layout.css"  />
</head>
<body>
  <div id="container">
    <div id="header">顶部(header)</div>
    <div id="main">
      <div class="cat">商品分类(cat)</div>
      <div class="content">内容(content)</div>
      <div class="sidebar">右侧(sidebar)</div>
    </div>
    <div id="footer">底部(footer)</div>
  </div> <!--container end-->
</body>
</html>
```

layout.css 文件的代码：

```
body{margin:0px;padding:0px;}
div{font:bold 36px 黑体;color:#fff;}
#container{width:980px;margin:0px auto;}
 #header {width:100%;height:136px;background:#ccc;color:#000;}
 #main{width:100%;height:400px;}
   .cat,.content,.sidebar{float:left;width:20%;height:100%;}
   .cat{background:#666;}
   .sidebar{background:blue;}
   .content{width:60%;background:red;}
 #footer{width:100%;height:100px;background:#ccc;color:#000;}
```

9.3 贵美网站首页各块的实现

实现了首页的整体布局后，具体看下各块的局部布局。在布局的过程中，需要根据实际内容调整宽高属性，以便和美工设计的页面效果一致。

9.3.1 顶部、Logo 及导航

顶部的实现是最开始也是较关键的一环，往往其结构比较随意，各块间的空隙也较大，因此经常使用 float 与 clear 搭配来实现。根据美工设计的页面效果图，规划首页顶部的思路，原图的结构过程如图 9.17 所示。

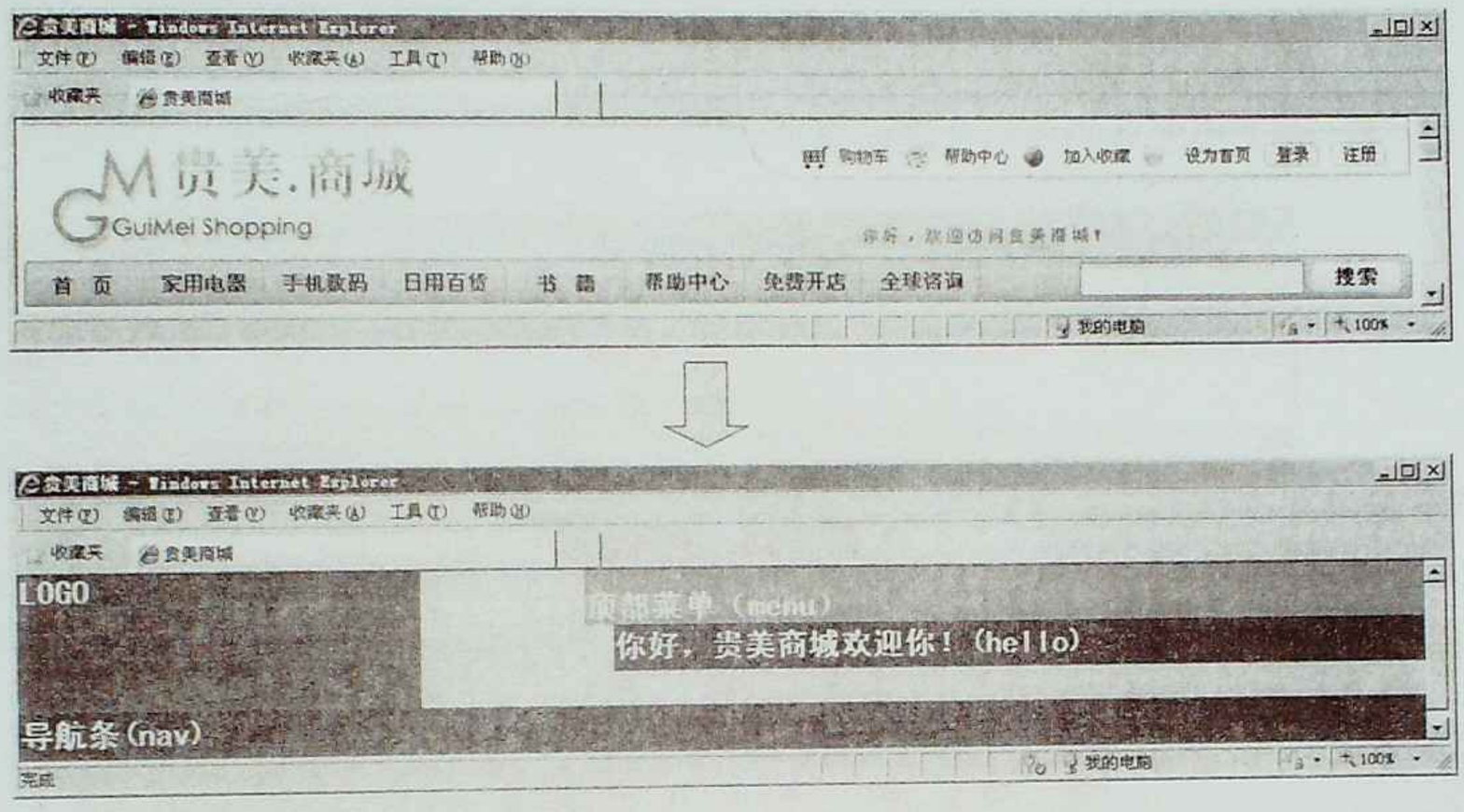


图 9.17　顶部的局部布局

1. 确定组织结构及宽高实现布局

我们在示例 1 的基础上，把顶部的组织结构写入“ ”中。

此部分的布局曾在第 6 章的上机练习 4 中实现过，可以参考相关代码。这里我们结合图 9.18 回顾一下实现思路。

(1) 顶部依次有四个 div 块：Logo、顶部菜单、欢迎词及导航条。

(2) 四个 div 块的浮动：“Logo”左浮动，“顶部菜单”和“欢迎词”右浮动，“导航条”用 clear 清除浮动。

(3) 设置各块的宽高，并设置背景图。

(4) 布局之前，必须先设置<body>、<ul>标签的默认外边距与内边距为零，<ul>无列表样式。

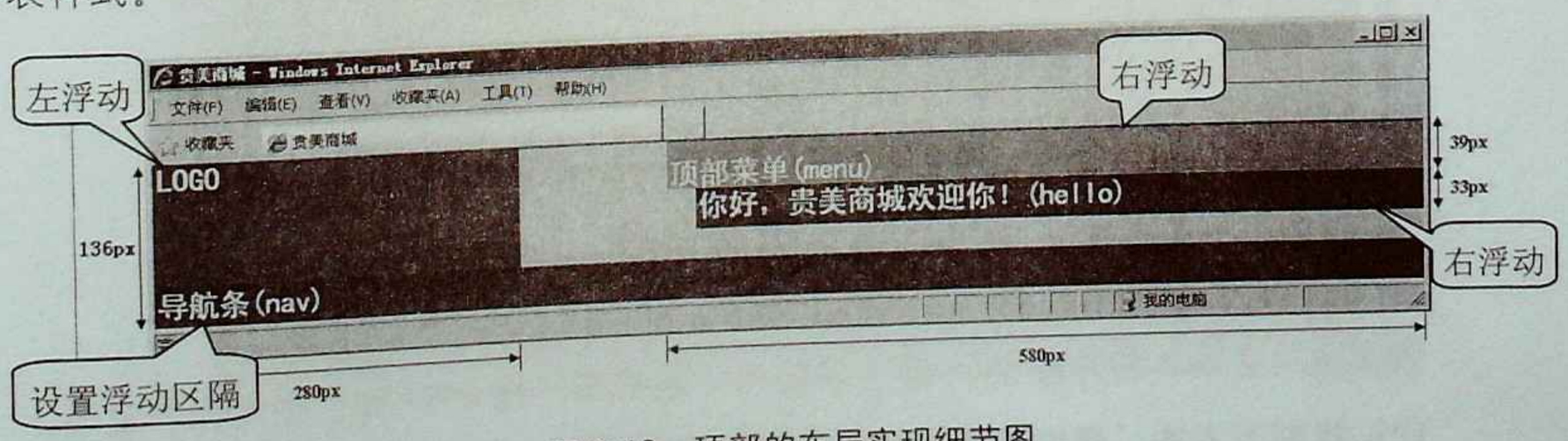


图 9.18　顶部的布局实现细节图

2．实现 Logo

设置 Logo 块的宽高，并用背景图偏移量技术把背景图的 Logo 部分截取出来。

实现以后，取消 Logo 块背景色对应的代码。

3．实现顶部菜单，如图 9.19 所示

（1）使用 div-ul-li，并为小图标与文字设置不同的 CSS 类，<li>标签左浮动。

（2）使用背景图偏移量技术实现同图截取不同的小图标。

（3）为顶部菜单的各项合理设置内边距。

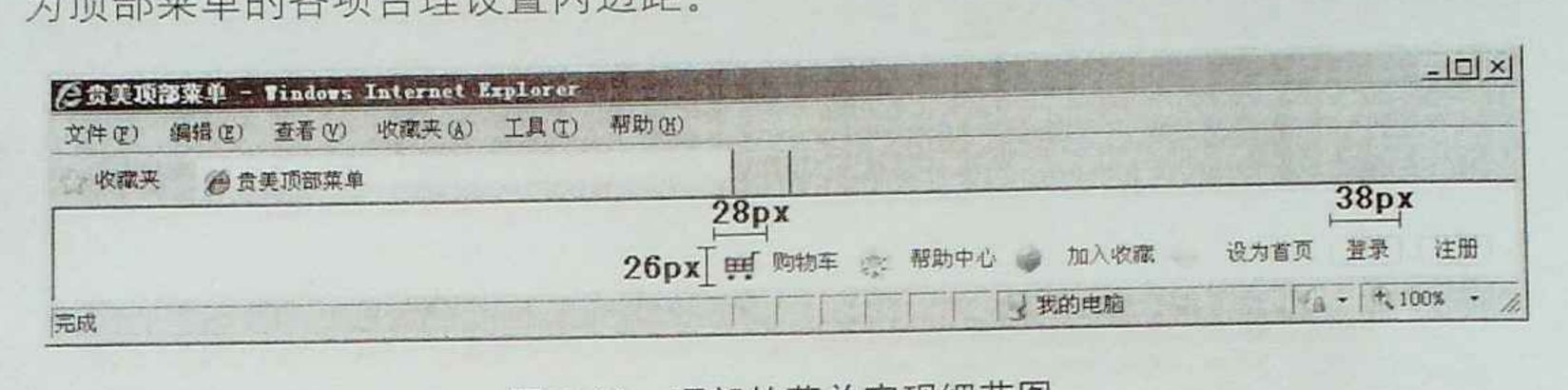


图 9.19　顶部的菜单实现细节图

制作顶部菜单时，可以借助 Dreamweaver 工具简化页面内容的输入。例如菜单列表项的输入，对应的操作过程如图 9.20 所示。

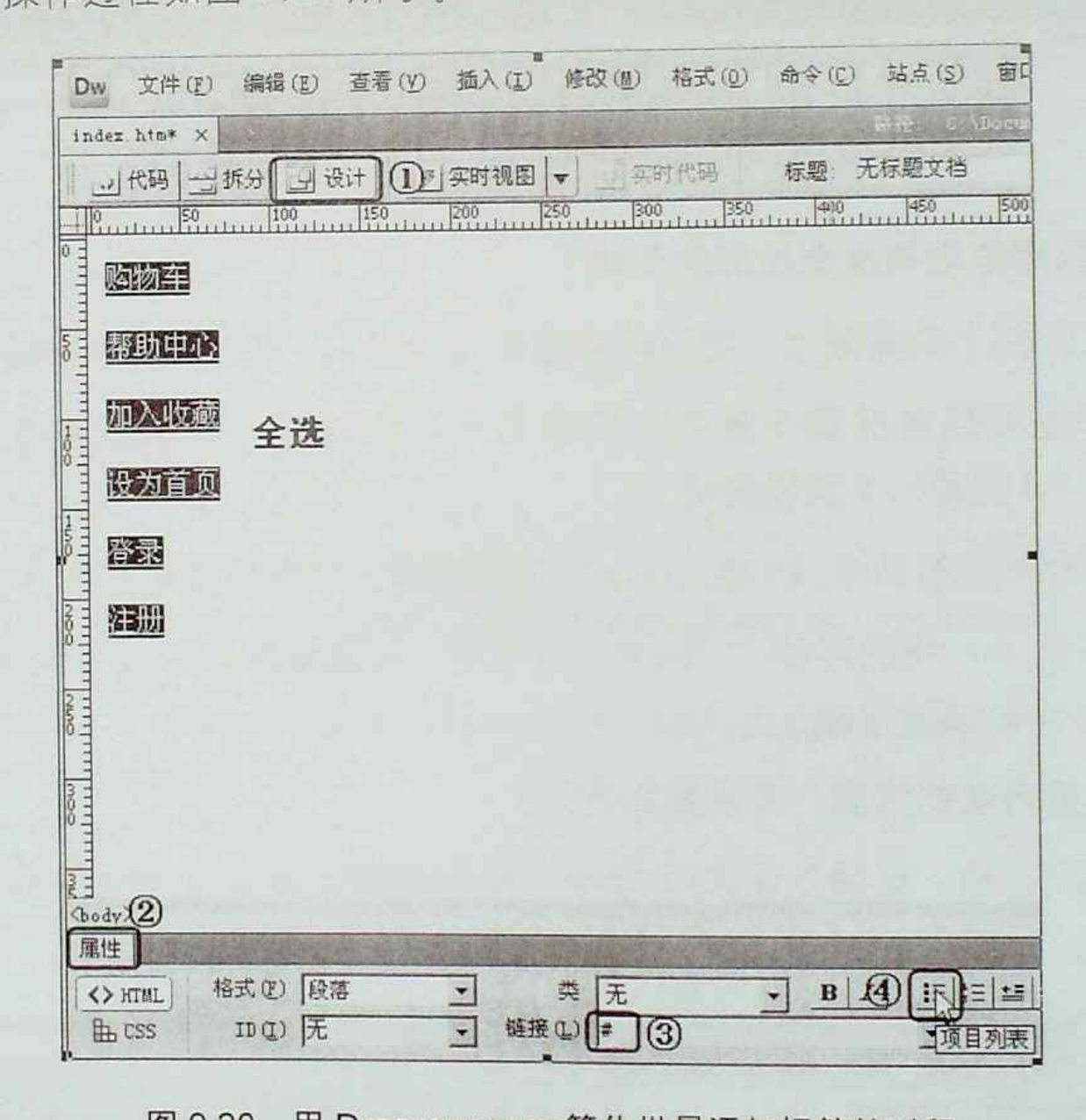


图 9.20　用 Dreamweaver 简化批量添加标签的过程

具体步骤如下。

（1）首先单击上方的“设计”标签进入设计模式。

依次输入各菜单名并按 Enter 键，如输入“购物车”按 Enter 键，输入完后全选。

（2）找到下方的“属性”栏。

（3）在“链接”框中输入“#”号，按 Enter 键，批量创建超链接。

（4）最后在“属性”栏中的“链接”右上方找到“项目列表”按钮并单击，即可完成输入。

需要注意的是，由于 Dreamweaver 自动生成的代码可能包括很多冗余代码，而且对于开发人员，企业更要求 HTML 代码的手写能力，所以本课程只将 Dreamweaver 用于编写 HTML。

4. 实现欢迎词

只需设置宽度、上内边距、字符间距和字色即可。

5. 实现导航条

如图 9.21 所示，此部分曾在第 7 章的上机练习 1 中实现过，实现思路如下。

（1）用 div-ul-li 实现导航条的组织结构，<li>标签左浮动。

（2）设置<li>标签宽度 84px，并设置<a>标签的内边距及行高(32px)与整个按钮的宽高相等。

（3）用超链接伪类 a:hover 实现鼠标悬浮显示特效，需要细微调节自然融入背景图。

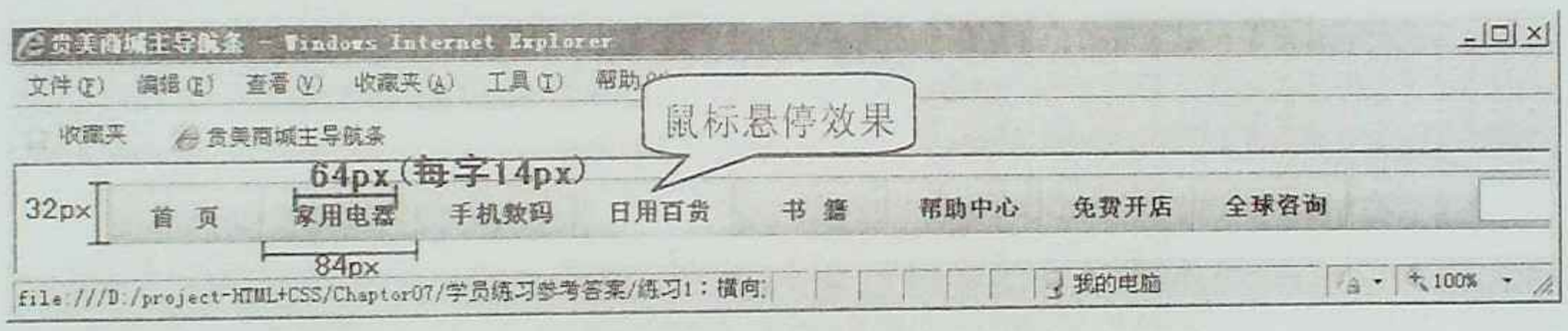


图 9.21　导航菜单实现细节图

到这里，顶部各块都实现成功，效果如图 9.22 所示。

图 9.22　顶部：Logo 及导航

我们来回顾整个实现过程，并结合图 9.22 与示例 2 的关键代码，理清整个顶部实现的流程。

示例 2

head.htm 的代码：

```
<head>
<!--省略部分 HTML 代码 -->
 <link rel="stylesheet"  type="text/css" href="css/layout.css"  />
</head>
<body>
    <div id="container">
        <div id="header">
            <div class="logo"></div>
             <div class="upright_menu">
                 <ul>
                      <li class="pic pic1"></li>
                      <li class="h_text"><a href="buy.htm" target="_parent">
                      购物车</a></li>
        <!-- 后面的帮助中心、加入收藏、设为首页分别设置为pic2-pic4,结构类似,略-->
                      <li  class="pic  btn  h_text"><a  href="login.htm"
                      target="_parent">登录</a></li>
                      <!-- 注册按钮实现同理 -->
                  </ul>
             </div> <!--upright_menu end-->
             <div class="upright_bottom">
                  <ul>
                      <li class="h_text  hello">你好,欢迎访问贵美商城!</li>
                  </ul>
             </div> <!--upright_bottom  end-->
        <div class="nav" id="nav">
             <ul>
                  <li><a  href="index.htm"  target="_parent"> 首 
                    页 </a></li>
                  <li><a href="commoditys_show.htm" target="_parent">家用电
                  器</a></li>
                  <!--与上面类似,手机数码、日用百货、书籍、帮助中心、免费开店与全球
                  咨询略--> </ul>
        </div> <!--nav end-->
        </div> <!--header end-->
      <div id="main">
        <div class="cat">商品分类(cat)</div>
        <div class="content">内容(content)</div>
        <div class="sidebar">右侧(sidebar)</div>
      </div>
      <div id="footer">底部(footer)</div>
        </div> <!--container end-->
  </body>
```

head.htm 中引用的 layout.css 文件：

```
/*设置 body、dl、ul、ol、h1、a、img 等的通用默认样式*/
body{font:12px Tahoma;    }
body,dl,ul,ol,h1{margin:0px;padding:0px;}
a {color:#333333;text-decoration: none;}
a:hover {color:#ff7300;}
a img{border:0px;}

/*布局用的页面层容器 container*/
#container{
    width:980px;
    margin:0 auto;
    float:none;
    }

/*头部从 logo 到服务热线*/
#header {
    width:100%;
    height:136px;
    background:url(../images/h_bg.jpg) no-repeat -22px 0px;
    }

    .logo {     /*缩进表示 logo 属于#header 区块,增加可读性,后续同理*/
      width:280px;
      height:100px;
      float:left;
      background:url(../images/h_bg.jpg) no-repeat -22px 0px;
        }

  /*右上方菜单、购物车、帮助中心、加入收藏、设为首页等*/
.upright_menu {width:45%;height:28px;float:right;padding:13px 0px 0px 0px;}
  .upright_menu ul li{float:left;font:12px/28px 宋体;}

      /* .pic1.pic4 分别为菜单前的小图标*/
      .pic{width:28px;height:28px;background:url(../images/icon.gif) no-
      repeat;}
      .pic2{background-position:-28px 0px;}
      .pic3{background-position:-84px 0px;}
      .pic4{background-position:-112px 0px;}
      /*btn 为注册或登录按钮,h_text 为菜单文本*/
      .h_text,.btn {padding:0px 5px;text-align:center;}
      .btn {width:38px;background-position:0px -25px;}
.upright_bottom{width:62%;float:right;}

    /*贵美欢迎你*/
    .hello {padding-top:36px;color:#636362;letter-spacing:3px;}

/*导航条*/
```

```
.nav{clear:both;width:100%;}
.nav li {
  width:84px;
  float:left;
  text-align:center;
  font-weight:bold;
  }
.nav li a{
  padding:8px 10px;line-height:32px;
  color:#333;
  font-size:14px;
  }
.nav ul li a:hover{
  background:url(../images/nav_bg.gif) no-repeat;
  }

/*顶部以外的其他区域样式*/
#main,#footer,.cat{font:bold 36px 黑体;color:#fff;}

 #main{width:100%;height:643px;}
  .cat,.content,.sidebar {float:left;height:100%;}
  .cat{width:204px;background:#666;}
  .cat_li{font:12px/24px 宋体;color:#636362;width:60px;float:left;}
.sidebar   {background:blue;width:185px;}
.content   {width:60%;background:red;}
#footer{clear:both;width:100%;height:100px;
   background:#ccc;color:#000;}
```

上机练习 1

练习——实现贵美商城网站首页的整体布局

需求说明

根据电子效果图，重新实现贵美商城网站首页的整体布局（顶部不要求，可以在课下完成），如图 9.14 所示。

9.3.2 左部：商品分类

顶部实现以后，接下来就是页面左部的“商品分类”。相对于顶部而言，左侧的结构相对简单。本块实现的重点在于对多行文本的布局能力。本部分在上例的基础上进行，请复制 head.htm 并重命名为 cat.htm。

前面章节已讲解过部分内容的实现，这里主要介绍整体的实现思路。

1. 实现组织结构

参考图 9.1 所示的效果图，左部“商品分类”版块仍然采用 div-ul-li 结构（也可以采用 div-dl-dt-dd 结构），但仔细分析还是有些区别。

首先，由于类目的重要性，使用<h1>标签有利于搜索引擎优化；其次，各类的结构几乎完全一样，用多个<ul>来表示即可。因此，总体组织结构可以归纳如下。

```
<div class="cat">
  <ul>
    <h1></h1>
    <li></li>
    ……<!-- 多个列表项 -->
    <li></li>
  </ul>
  <ul>
    <h1></h1>
    <li></li>
    ……<!-- 多个列表项 -->
    <li></li>
  </ul>
  ……<!-- 多个ul -->
</div>
```

批量添加技巧为：用 Dreamweaver 批量添加功能添加第一块，修改<ul>标签第一项为<h1>标签，然后转到“代码”视图。复制整段<ul>并拷贝四次。修改各段<ul>标签。

2. 确定各块宽高、内边距及<ul>外边距和内边距清零

- “cat”类的 （即“商品分类”版块的 div 容器），背景图设置为 bg.gif。
- “cat”类的 ，上内边距 40px，左内边距 15px。
- “cat”类的 ，宽 189x，高 603px（实际宽度 189+15=204，实际高度 603+40=643）。标签的宽高（width、height 属性）与实际占位大小在布局中很容易混淆，建议尽量不用 margin，少用 padding，用 padding 时必须注意调整标签宽高的属性。
- 对<ul>方框属性清零“margin:0px;padding:0px;”。

3. 确定行高及列宽

- 标题行：行高 27px。
- 列表 5 行：各行的行高 24px。

用确切行高来实现文字垂直居中及整个内容的高度，这点非常关键，因为在多浏览器中兼容性最佳。

➢ 各项宽度 60px。

因为 cat 总实际宽度为 204px，除去左内边距的 15px，cat 的标签内容宽度为 189px，每行 3 项，除以 3 是 63px，略留空白则为 60px。

用宽度 = 总宽度 / 每行个数与浮动配合的方法来实现多行列表最佳，代码简洁，扩展性和兼容性良好。

整个划分如图 9.23 所示。

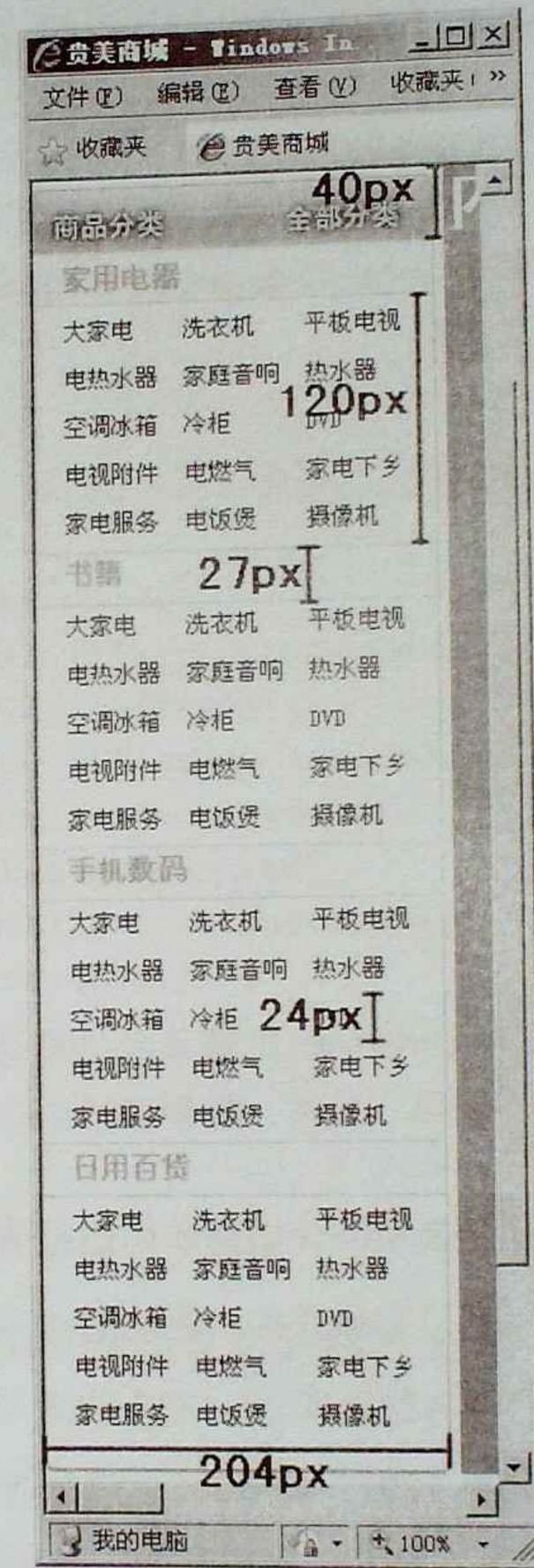


图 9.23 左部：商品分类

实现思路分析完毕后，我们结合示例 3 的具体代码，仔细体会具体的实现过程。

示例 3

cat.htm 的代码：

```
<head>
<!--省略部分 HTML 代码 -->
 <link rel="stylesheet"  type="text/css" href="css/layout.css"  />
```

```
</head>
<body>
  <div id="container">
    <div id="header">……<!--头部代码略--></div>
    <div id="main">
      <div class="cat">
            <ul class="cat_ul">
                <h1>家用电器</h1>
                <li class="cat_li"><a href="#">大家电</a></li>
                ……<!--其余 li 同上,略-->
            </ul>
            <ul class="cat_ul">
                <h1>书籍</h1>
                <li class="cat_li"><a href="#">大家电</a></li>
                  ……<!--其余 li 同上,略-->
            </ul>
            <ul  class="cat_ul">
                <h1>手机数码</h1>
                <li class="cat_li"><a href="#">大家电</a></li>
                  ……<!--其余 li 同上,略-->
            </ul>
            <ul class="cat_ul">
                <h1>日用百货</h1>
                <li class="cat_li"><a href="#">大家电</a></li>
                  ……<!--其余 li 同上,略-->
            </ul>
      </div> <!--cat end-->
      <div class="content">内容(content)</div>
      <div class="sidebar">右侧(sidebar)</div>
    </div> <!--main end-->
    <div id="footer">底部(footer)</div>
  </div> <!--container end-->
</body>
```

cat.htm 中引用的 layout.css 代码：

```
……<!--示例 2 已出现并不影响本部分,略-->
dl,ul,ol,h1{list-style:none;margin:0px;padding:0px;}
……<!--示例 2 已出现并不影响本部分,略-->
<!--以下为本部分实现的关键代码-->
  #main,#footer{font:bold 36px 黑体;color:#fff;}
  #main{width:100%;height:643px;}
    .cat,.content,.sidebar {float:left;height:100%;}
    .cat{
      height:603px;
      width:189px;
      padding:40px 0px 0px 15px;
      background:url(../images/bg.gif) no-repeat;
   }
      .cat h1{
```

```
        font:bold 14px 宋体;
        color:#ff7300;
        line-height:27px;
      }
      .cat_li{width:60px;float:left;font:12px/24px 宋体;color:#636362; }
<!--     以下为其余非关键代码     -->
  .sidebar{width:185px;background:blue;}
  .content{width:60%;background:red;}
 #footer{clear:both;width:100%;height:100px;background:#ccc;color:#000;}
```

示例 3 实现后，整个页面在浏览器中的预览效果如图 9.24 所示。

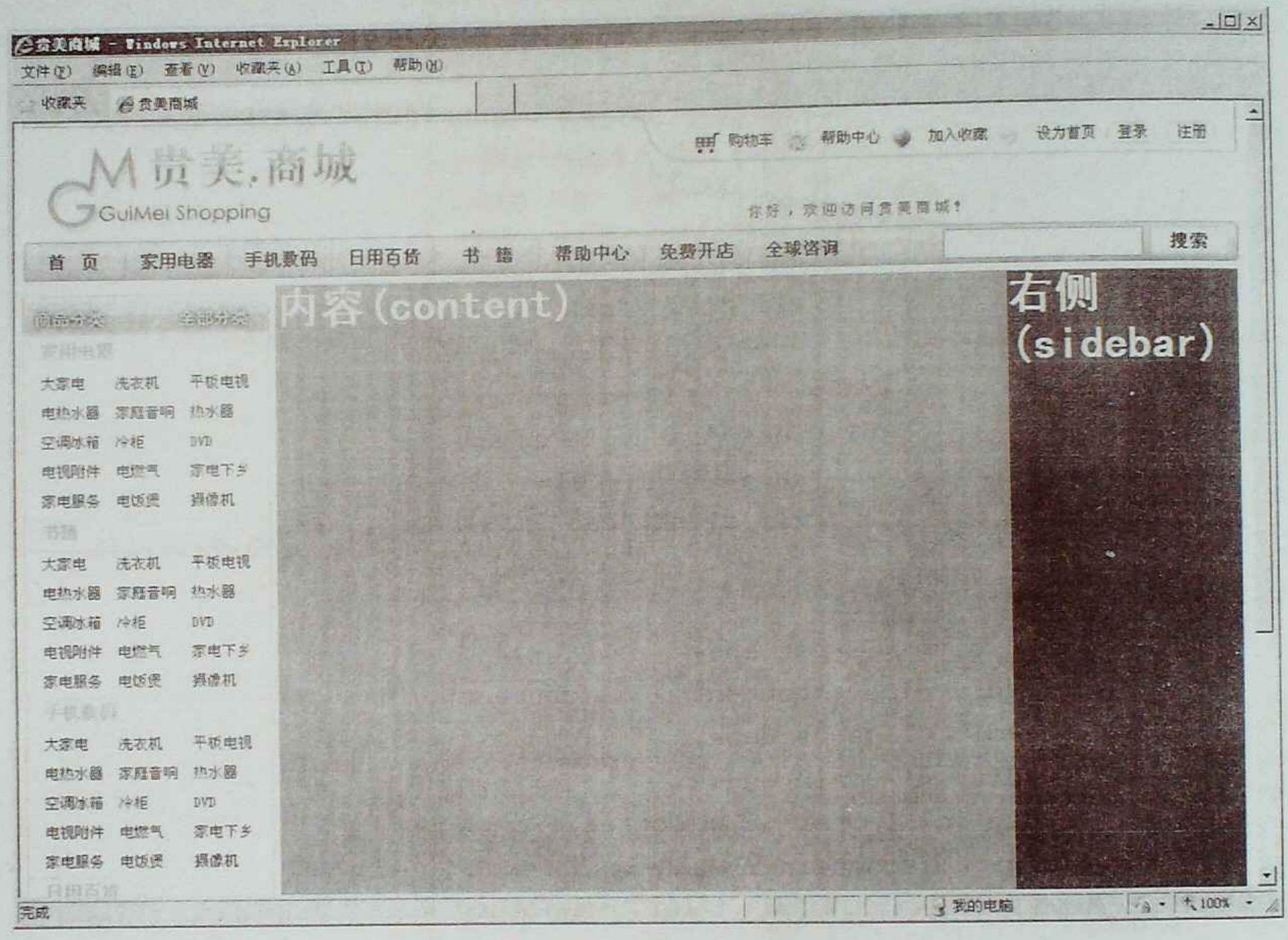


图 9.24　左部商品分类实现后的整体效果图

上机练习 2

练习——实现左部商品分类版块

需求说明

在上一练习的基础上，重新实现商品分类版块。

9.3.3 中部：主体

左部商品分类完成以后，完成中部主体时同样也是在上例基础上进行扩展的，复制 cat.htm 并重命名为 content.htm。

这时涉及到一个问题：如何实现左部与中部之间的空白间隙问题。如果不用 margin、padding，在哪个区块实现比较好？建议的方法是在大区块的右边或下方来实现。对于本例，也即为“cat” 增加右内边距 9px。只需把“padding:40px 0px 0px 15px;”改为“padding:40px 9px 0px 15px;”即可实现。这样做的好处是每个区块开始写的时候，默认的左边和顶边的外边距、内边距为零。另外，在用内边距实现布局时，切忌在内层标签如<ul>、<li>标签中设置边距来实现，应在外层如“cat”类的 标签中实现。这样既简洁又高效，而且利于扩展和修改。

根据上述思路，下面介绍具体实现的步骤。

1. 实现中部局部布局

分上下两块，如图 9.25 所示。

- 上方实际高度为 203px，其中容器高度为 190px，同时设置下方的内边距为 13px，作为和“疯狂购物”版块的空白间隙。
- 下方实际高度为 440px(643px-203px)，其中设置上方的内边距为 37px，作为背景图和下方内容的空白间隙，下方容器标签 的高度设为 403px。
- 上下方的宽度都是 524px。

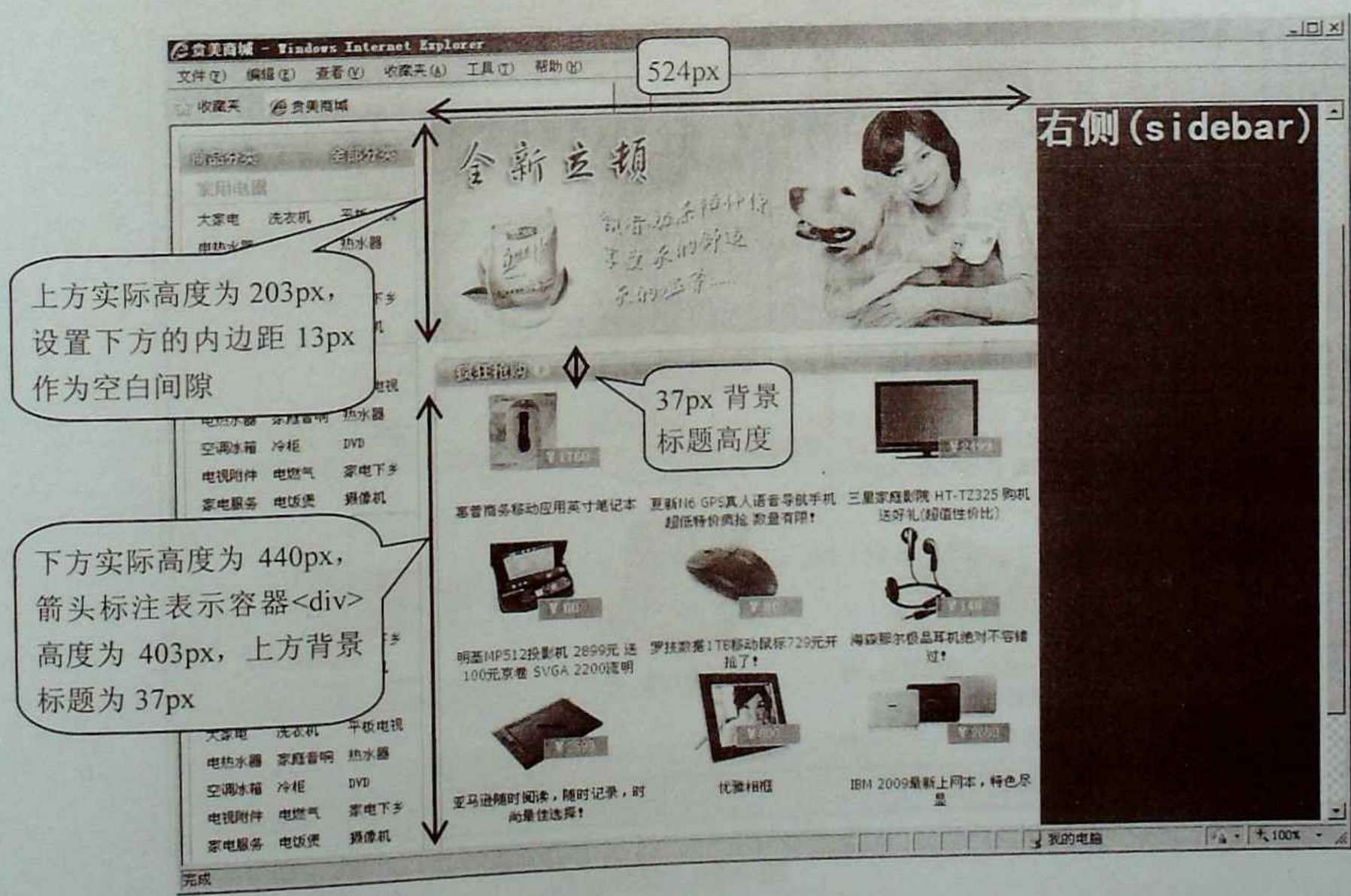


图 9.25 中部版块

2. 实现上方的广告图

在标签中放置一张图即可，如“”。注意图片大小最好等于容器标签 宽高，如不一致，需设置<img />的宽高与之一致。

3. 实现下方的“疯狂抢购”

1）实现组织结构

分析特点：整体可等分为多块，每个块又分为有对应关系的一图一文。简单描述为“整体-块-图-文”的结构，与 div-dl-dt-dd 的结构非常吻合。如果用 div-ul-li 也可以实现，但结构语义不符，图文之间并非并列关系，文是对图的扩展描述，具体实现语句如下。

```
<div>
  <dl>
    <dt><img /></dt>
    <dd>对上图的扩展描述</dd>
  </dl>
  ……<!-- 多个 dl -->
</div>
```

2）确定各块宽高、内边距及浮动、<dl><dd>外边距和内边距清零

我们结合图 9.26 来分析实现思路。

图 9.26 疯狂抢购

（1）从图 9.26 中可看出：共三行，每行包括三个<dl>块。那么每个<dl>块宽 174px(524px/3)，高 134px(403px/3)。

（2）各<dl>块设置左浮动，并且设置容器 的子元素水平居中，<dl><dd>外边距、内边距均为零。

（3）各块中一图一文，图片高度不超过 91px，文字两三行，默认行高不会超过 43px。

（4）设置<dt>的行高（lineheight）=内容高度（height），并且让图片垂直居中，设置图片有 1px 的灰色边框。

理清思路后，我们再对照示例 4 的具体代码实现，体会 div-dl-dt-dd 布局的特点。

示例 4

content.htm 的代码：

```
<!-- 顶部左部部分,同上例,略 -->
<div class="content">
  <div class="content_top"><img src="images/ad-01.jpg" alt="9 月新品" /></div>
     <div class="content_list">
       <dl>
         <dt><img src="images/promote-1.jpg" alt="alt" /></dt>
         <dd><a href="#">惠普商务移动应用英寸笔记本</a></dd>
       </dl>
     <!-- 15 个 dl,结构与上面类似,略 -->
     </div> <!--content_list end-->
</div> <!--content end-->
<!-- 剩余部分,同上例,略 -->
```

content.htm 中引用 layout.css 的代码：

```
<!-- 顶部左部部分,同上例,略-->
     .content,.content_top,.content_list{width:524px;}
     .content{padding-right:10px;}
     .content_top{height:190px;padding-bottom:13px;}
     .content_list{height:403px;padding:37px 0px 0px 0px;text-align:center;}
     .content_list dl{float:left;width:174px;height:134px;}
     .content_list dt{height:91px;line-height:91px;}
     .content_list dd{height:43px;}
     .content_list dt img{vertical-align:middle;border:1px solid #ccc;}
<!-- 剩余部分,同上例,略 -->
```

此步实现完的整体效果如图 9.27 所示。

图 9.27　中部实现完的整体效果图

上机练习 3

练习——实现中部版块

需求说明

利用提供的素材，重新制作贵美商城网站首页中部的版块。

本章总结

- 网站开发流程一般包括需求分析、网站制作、测试网页、发布网站几个环节，其中前期的需求分析非常重要，一般还需要美工人员设计界面草图及和客户进行反复确认。
- 制作网页前一般需要使用 Dreamweaver 工具搭建站点的目录结构，除此之外，还可以使用 Dreamweaver 的智能提示功能，辅助编辑 HTML 代码。
- 首页整体布局的实现思路采用的是内容和样式分离的思想。
  - 根据效果图规划并编写 HTML 组织结构，并尽量使用语义化的 HTML 标签。
  - 在单独的 CSS 文件中设置相关的盒子属性，需要注意各 div 块的规范命名。
- 首页顶部实现的关键点有以下四点。
  - 合理使用左右浮动，实现零散布局。
  - 一图一文混编，并且在同一行，可以用 div-dl-dt-dd 分两类来实现布局。
  - 偏移量技术的典型应用。
  - 用 a:hover 伪类实现鼠标悬浮的特效。
- 首页左部实现的关键点有以下三点。
  - 只用行高布局来实现多行文字布局的浏览器兼容。
  - 区域衔接原则：在右侧或底部添加内边距来实现。
  - 重要标题使用<h1>，以利于搜索引擎优化。
- 首页中部实现的关键点有以下三点。
  - 利用 div-dt-dl-dd 的特点实现一图一文混编并需要多行布局的应用场景。
  - <dt>和<dd>存在默认内边距与外边距，布局时应先清零。
  - 把图片和文字分别放入不同的容器<dt>和<dd>，以便对它们进行排版。

本章作业

一、选择题

1. 以下（　　）不属于开发流程。

A. 需求分析与伪界面设计

B. 测试网站与发布网站

C. HTML 页面制作和 CSS 书写

D. 制定网站营销策略

2. 关于常见的页面布局特点的说法，正确的是（　　）。

A. 表格布局属于传统布局，存在很多不规范的地方，因此，应该完全不用表格布局

B. DIV+CSS 布局这个概念说明布局过程中全部用 标签实现

C. DIV+CSS 主要是体现了 W3C 的内容和样式分离的思想

D. DIV+CSS 布局具有简洁高效、内容样式分离并且利于改版等特点

3. 在中部“疯狂购物”布局中，以下描述正确的是（　　）。

A. 各<dl>的宽高属性，可以分别用总宽度除以每行个数、总高度除以行数来确定

B. 必须为<dl>和<dd>标签进行内边距和外边框的清零设置，以便更好地控制布局

C. 实现图片垂直居中，需要设置图片容器的高度和行高一致

D. 可以采用 div-ul-li 结构代替 div-dl-dt-dd 布局，这样的结构语义更好

4. 在需求分析阶段，不需要重点关注的方面是（　　）

A. 为什么需要建站，即建站目的

B. 谁来访问，即目标受众

C. 内容是什么，即确定网站的内容

D. 访问者的场所及访问频率

5. 在左部商品分类的布局思路中，描述正确的是（　　）。

A. 如采用 div-ul-li 结构实现局部布局，则只能出现一个<ul>块结构

B. 多行文字布局中，采用行高属性进行布局，并合理使用浮动和宽度来实现所需的效果

C. 当在<li>标签内设置 padding 或者 margin，或者没有为<ul>方框属性清零，肯定会出现浏览器兼容问题（即同一页面在不同浏览器中显示效果不一致）

D．尽量使用 margin 和 padding 属性，不会导致浏览器兼容布局的问题

二、简答题

1．简述网站开发的流程以及需求分析阶段的重点关注点。

2．回顾并简述整个首页布局及各块的局部实现思路。

3．对于网页中各版块之间的空白间隙，其实现思路是什么？

4．在示例 4 的基础上实现贵美商城的右侧边栏，如图 9.28 所示。具体要求如下。

- 只要求实现右边第二栏的新品上架，其余留出设置宽高的 即可。
- 要求使用 div-dl-dt-dd 来实现，并且图文垂直居中。

图 9.28　右部实现后的完整体效果图

5．在第 4 题完成的基础上，实现贵美商城网站的底部信息，并最终完成贵美商城网站的首页，如图 9.29 所示。具体要求如下。

- 除了底部 、
、<hr />、<a>和<img />标签外，尽量不用其他多余标签。
- 用居中或者浮动实现所需效果，特殊字符的使用要符合规范。

图 9.29　贵美商城网站的完整效果图

第 10 章

贵美商城网站（二）

◈本章工作任务

- 制作贵美商城商品分类列表页
- 制作贵美商城商品具体介绍页

◈本章技能目标

- 使用<iframe/>制作网页模板
- 使用 Dreamweaver 工具实现表格布局
- 使用 div-ul-li 实现局部布局

本章单词

请在预习时学会下列单词的含义和发音，并填写在横线处。

1. vertical: _____
2. percent: _____
3. detail: _____
4. buy: _____
5. online: _____
6. pay: _____
7. share: _____
8. favorite: _____
9. button: _____
10. operation: _____

本章简介

上一章，我们实现了贵美商城网站的首页，也是工作量最大的一个页面。本章我们将在上章的基础上，分别实现网站的其他页面，包括商品分类列表页和具体介绍页。

首页完成以后，其他页面在制作时就有章可循。顶部和底部可以复用，相同的页面样式可以复用，所以实现其他页面的实际工作量会大大小于首页制作。

10.1 使用<iframe>实现首页顶部和底部的复用

1. 分离首页顶部为单独的页面文件

顶部已经实现过，只需将如下顶部的相关代码分离出来，形成一个新的页面，然后另存为 header.htm 文件即可。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
 <meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
 <title>贵美商城顶部</title>
 <link rel="stylesheet" type="text/css" href="css/layout.css"   />
</head>
<body>
  <div id="header"><!-- 顶部的代码,略 --></div>
</body>
</html>
```

2. 分离底部为单独的页面文件

底部类似，只是<body>之内的标签换成已实现的底部代码，然后另存为 footer.htm 文件即可。

3. 使用<iframe >复用顶部和底部

使用前面章节学习过的<iframe>标签，在网站其他页面的顶部、底部位置，引用上述分离的 header.htm 和 footer.htm 文件即可，对应的代码如示例 1，页面效果如图 10.1 所示。后续网站的其他页面，将以此为网站模板，添加其他内容。

示例 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
   <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
   <title>贵美商城顶部</title>
   <link rel="stylesheet" type="text/css" href="css/layout.css" />
  </head>
  <body>
      <iframe id="head" src="header.htm" width="980" height="136"
      frameborder="0" scrolling="no"></iframe>
      <div id="main"></div>
      <iframe id="foot" src="footer.htm" width="980" height="150"
      frameborder="0" scrolling="no"></iframe>
  </body>
</html>
```

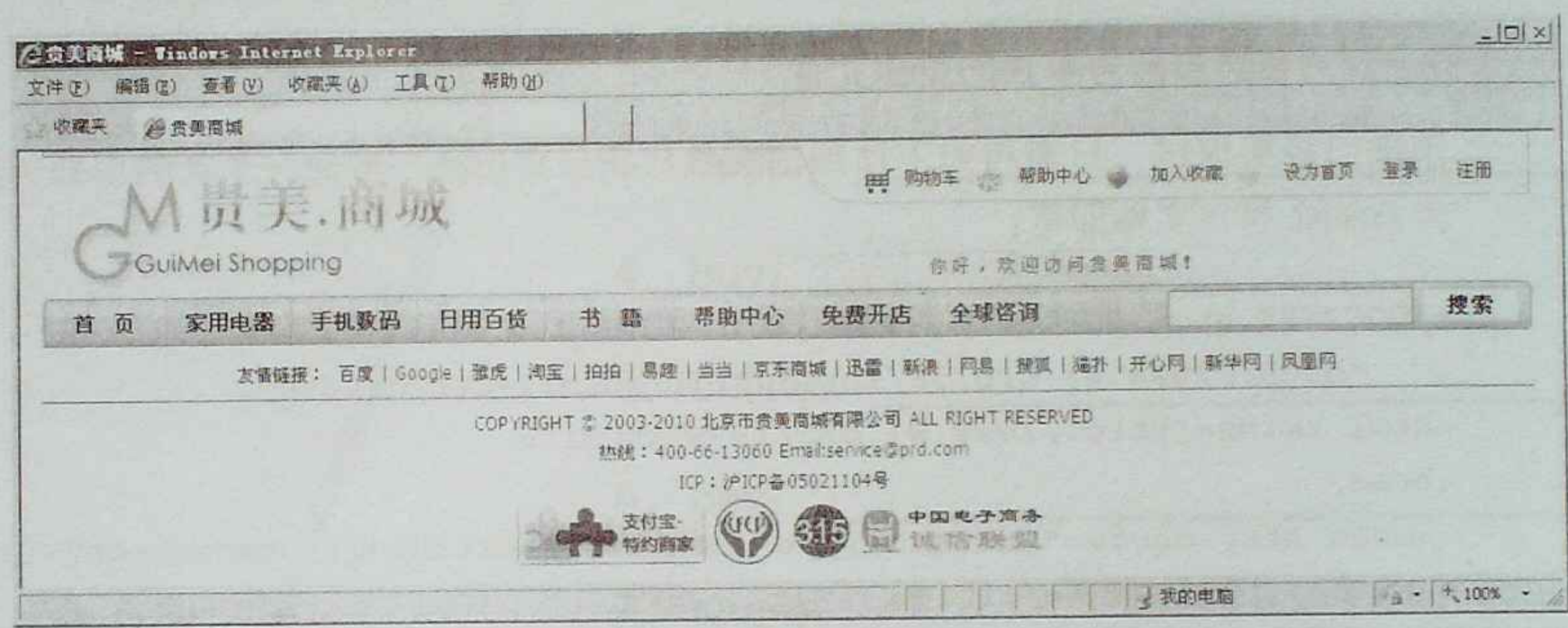


图 10.1　顶部和底部的复用实现

10.2　制作商品分类列表页

在示例 1 完成的网页模板基础上，现制作商品分类列表页，如图 10.2 所示。

显然，因为页面的顶部和底部可以复用，所以只需实现中间部分。这部分的页面结构我们在第 3 章曾讲解过，此处介绍如何利用 Dreamweaver，并结合 CSS 快速实现表格局部布局。实现先前复制上个例子的页面，并另存为 cat.htm。下面具体分析实现思路和步骤。

1. 思路分析

(1) 如图 10.3 所示，整体为 5 行 4 列，特点是第一行为标题行，其他各行基本类似。第 1 列（“全选”列）、第 4 列（“价格”列）宽度 10%，第 2 列（“商品图”列）宽度 20%，第 3 列（“商品简介”列）宽度 50%。

图 10.2　商品分类列表页效果图

（2）设置背景图为 catlist_bg.gif，上方的内边距 65px 作为内容上方的空白间隙，如图 10.3 所示。

（3）特殊修饰：全部居中，标题为粗体，文字行高 30px，如图 10.3 所示。

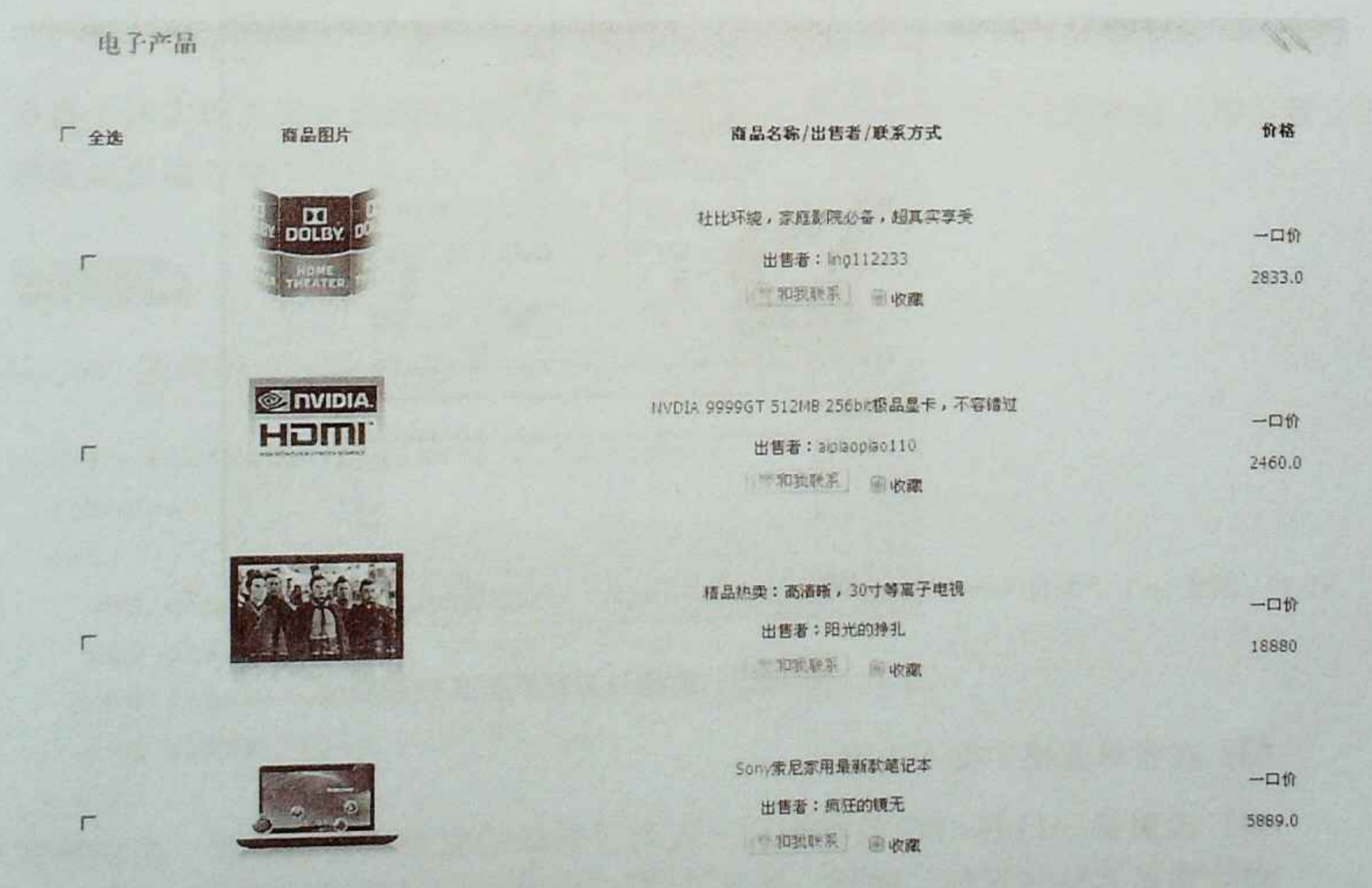


图 10.3　商品分类列表页思路细解

2. 用 Dreamweaver 辅助实现表格布局

理清思路后，我们用 Dreamweaver 来简化制作。

(1) 插入表格。在 Dreamweaver 中打开 cat.htm 文件，光标定位在 id 为 main 的 里，单击顶部菜单“插入(I)”→“表格(T)”，如图 10.4 所示。

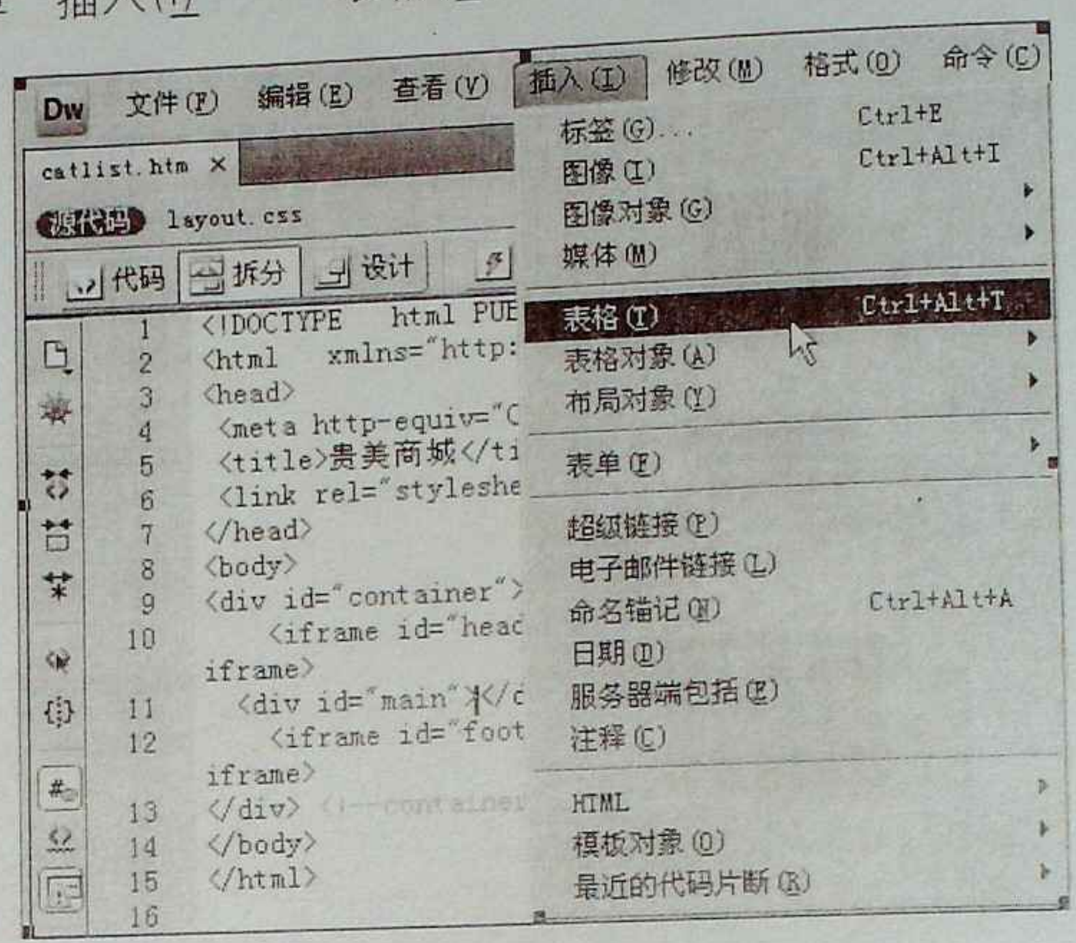


图 10.4 Dreamweaver 表格操作

(2) 设置表格属性。5 行 4 列，边框 0，标题选无，宽度设 100%，如图 10.5 所示。

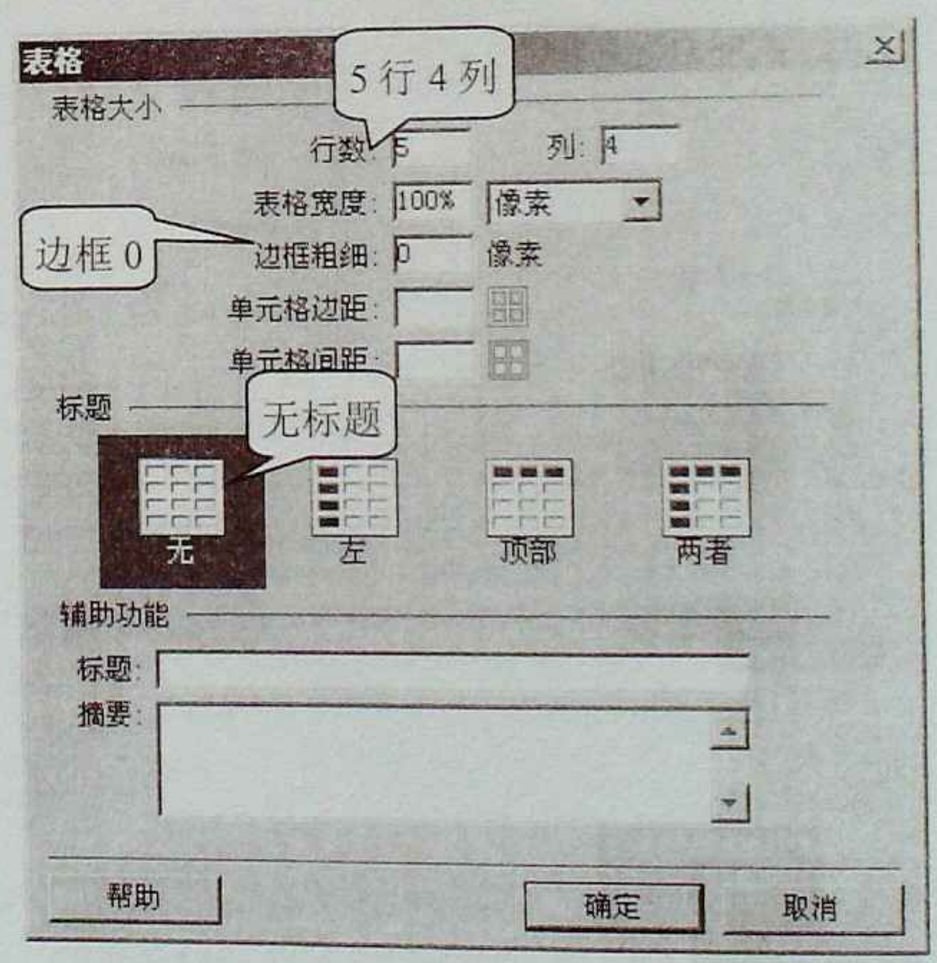


图 10.5 商品分类列表页表格的属性

(3) 在各单元格中输入内容。

(4) 设置第一行各<td>的样式。一般为了后续的复用和直观修改，可以把宽高的属性样式与类名统一规划。例如，类名“w10”代表宽 10px，而“w10p”则代表宽 10%，其中 p 是单词百分比 percent 的缩写。类名“b”代表字体加粗。HTML 及 CSS 代码如下。

第一行的 HTML 代码：

```
<tr>
  <td class="w10p b"><input type="checkbox" value="全选" /> 全选</td>
  <td class="w20p b">商品图片</td>
  <td class="w50p b">商品名称/出售者/联系方式</td>
  <td class="w10p b">价格</td>
</tr>
```

对应的 CSS 代码：

```
/* 常用细节修饰:宽高浮动对齐粗细等 */
    .w10p{width:10%;}
    .w20p{width:20%;}
    .w50p{width:50%;}

    .b{font-weight:bold;}
```

本课程将此类样式统一放置在 CSS 文件末尾，并按大小顺序排列，便于修改。

（5）设置表格的样式。增加类名“catlist”，在 CSS 中设背景图为“../images/catlist_bg.gif”，上方内边距 65px。另外，还需设置所有单元格<td>的行高为 30px，下边框为“1px #ddd dashed”，从而实现每行间的虚线分隔线，具体代码如下。

```
.catlist{
  background:url(../images/catlist_bg.gif) no-repeat;
  padding-top:65px;
  }
  .catlist td{border-bottom:1px #ddd dashed;line-height:30px;}
```

根据上述实现思路，对应的 HTML 和 CSS 代码如示例 2 所示。注意体会：用尽量少的代码实现页面效果。

示例 2

cat.htm 的代码：

```
<table id="main" class="a_c catlist">
  <tbody>
  <tr>
    <td class="w10p b">><input type="checkbox" value="全选" /> 全选</td>
    <td class="w20p b">商品图片</td>
    <td class="w50p b">商品名称/出售者/联系方式</td>
    <td class="w10p b">价格</td>
  </tr>
  <tr>
    <td> <input type="checkbox" value="1" name="auction_id" /></td>
    <td><a href="#" title="商品详细情况">
    <img src="images/list0.jpg" alt="alt" /></a></td>
    <td>
```

```
        <a href="#">杜比环绕,家庭影院必备,超真实享受</a><br />
        出售者:<a href="#">ling112233</a><br />
        <a href="#"><img src="images/contactme.gif" alt="alt" /></a>
          
        <a href="#"><img src="images/addfav.gif" alt="alt" /> 收藏</a>
      </td>
      <td>一口价<br /> 2833.0 <br /></td>
    </tr>
    <!-- 与上述粗体大体类似,略过 -->
    </tbody>
  </table>
```

cat.htm 引用的 layout.css 样式代码：

```
  /* 商品分类列表页 */
.catlist{background:url('../images/catlist_bg.gif') no-repeat;
      padding-top:65px;
      }
  .catlist td{border-bottom:1px #ddd dashed;line-height:30px;}

  /* 常用细节修饰:w 宽度,h 高度,p 百分比*/
  .w10p{width:10%;}
  .w20p{width:20%;}
  .w50p{width:50%;}
  .b {font-weight:bold;}
```

10.3 上机练习

上机练习 1

练习——利用模板制作商品分类列表页

需求说明

根据提供的素材，利用模板重新实现商品分类列表页，如图 10.2 所示。

10.4 制作贵美商城商品具体介绍页

如图 10.6 所示，该页面和制作商品分类页一样，页面顶部和底部复用，只需实现中间部分。中间部分的整体结构类似于 T 字型，可简单看成三块：上方标题及左右两列。

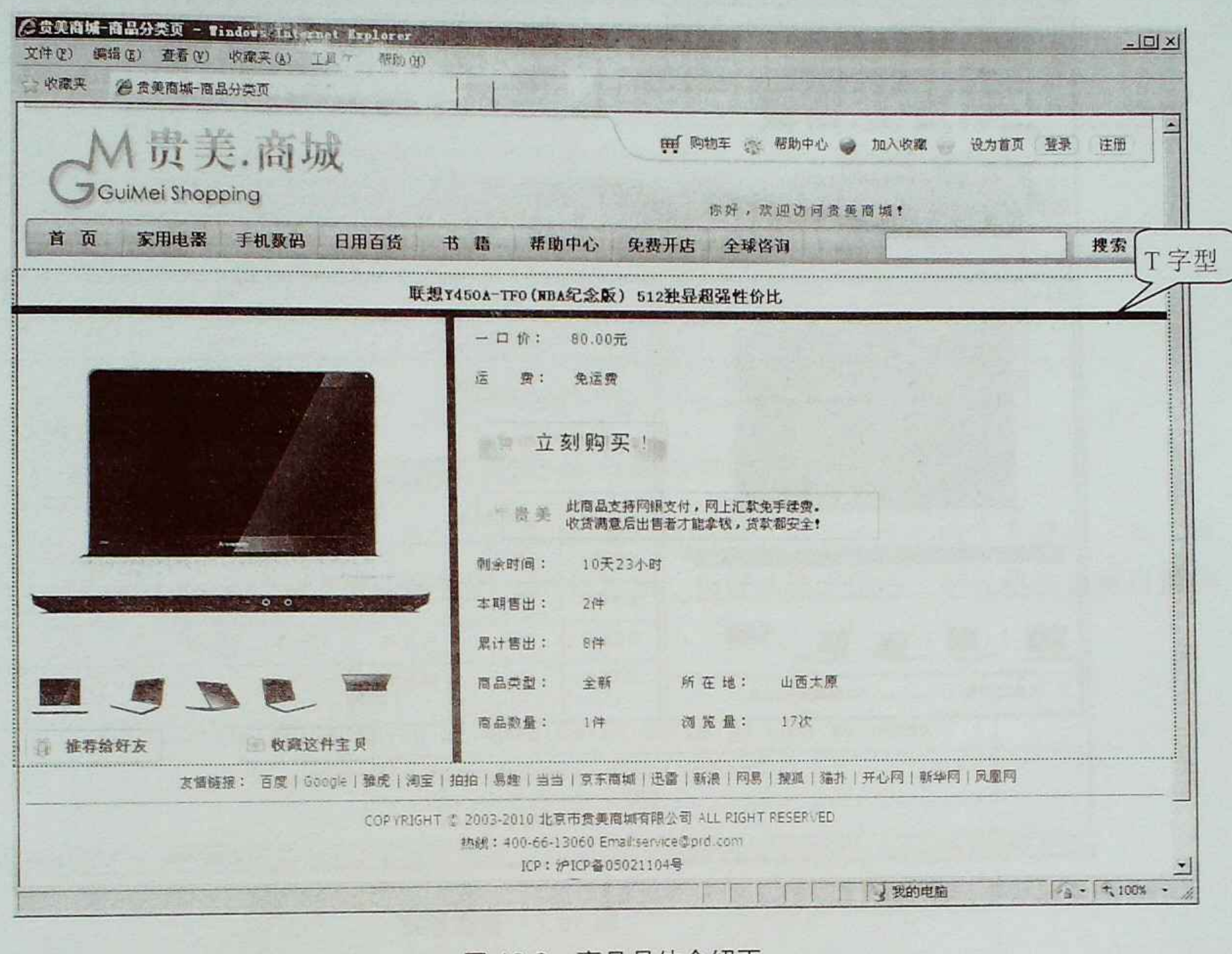


图 10.6 商品具体介绍页

和商品分类页的制作一样，先把示例 1 中实现的页面模板拷贝过来，并另存为 gooddetail.html，css 文件则拷贝示例 2 的 layout.css 文件。

如何实现中间部分的页面布局？我们先用常规的普通思路实现，分析存在的问题，然后再用简洁思路进行优化，下面逐一介绍如下。

10.4.1 用普通思路实现

1. 页面布局分析

如图 10.7 所示，根据页面效果，对中间部分的页面布局分析如下。

- 三块设置三个块级元素：标题<h1>、左列 与右列<table>。左、右两 均设为左浮动。
- 左列 分 3 个<ul>：第一个<ul>包含一个<li>，用于放置笔记本大图；第 2 个<ul>包含五个<li>，用于放置各个缩略图；第 3 个<ul>含两个<li>，用于放置“推荐给好友”、“收藏这件宝贝”这两个链接图标。
- 右列数据：数据规则用 9 行 4 列的表格实现，可以借助 Dreamweaver 辅助完成。

图 10.7 普通思路

- 9 行 4 列的表格，宽度 350px，为操作方便，先设边框为 1，如图 10.8 所示。
- 用 Dreamweaver 设置跨行跨列。

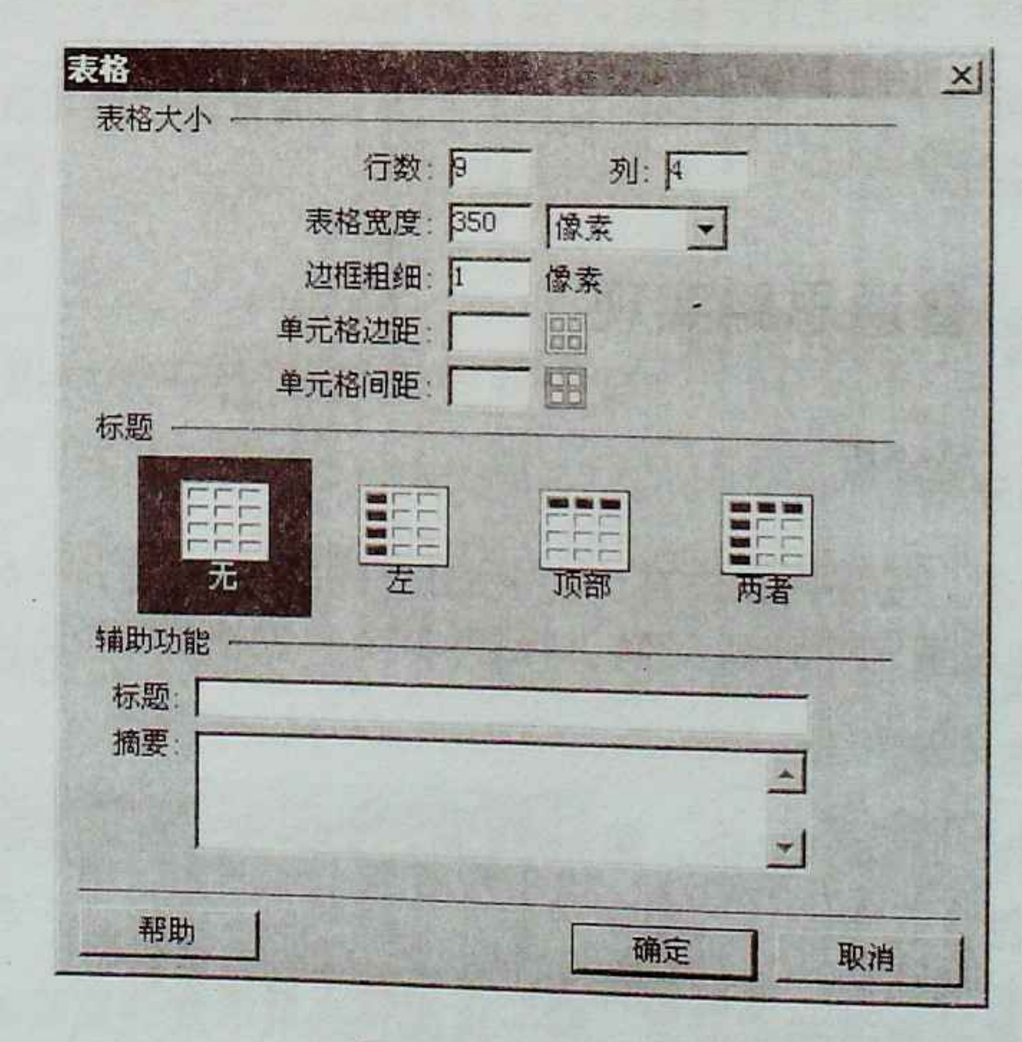


图 10.8 右列表格

切换到设计视图，选中第 3 行的四个单元格，单击右键，选择“表格(T)”→“合并单元格(M)”，同样的对第 4 行进行相同的操作，如图 10.9 所示。

图 10.9　Dreamweaver 合并单元格

另外第 1、2、5、6、7 行第 3 列都跨两列。跨列合并的操作完成后，在“设计”视图中具体效果如图 10.10 所示。保证结果无误后，把表格的边框设置为无边框。

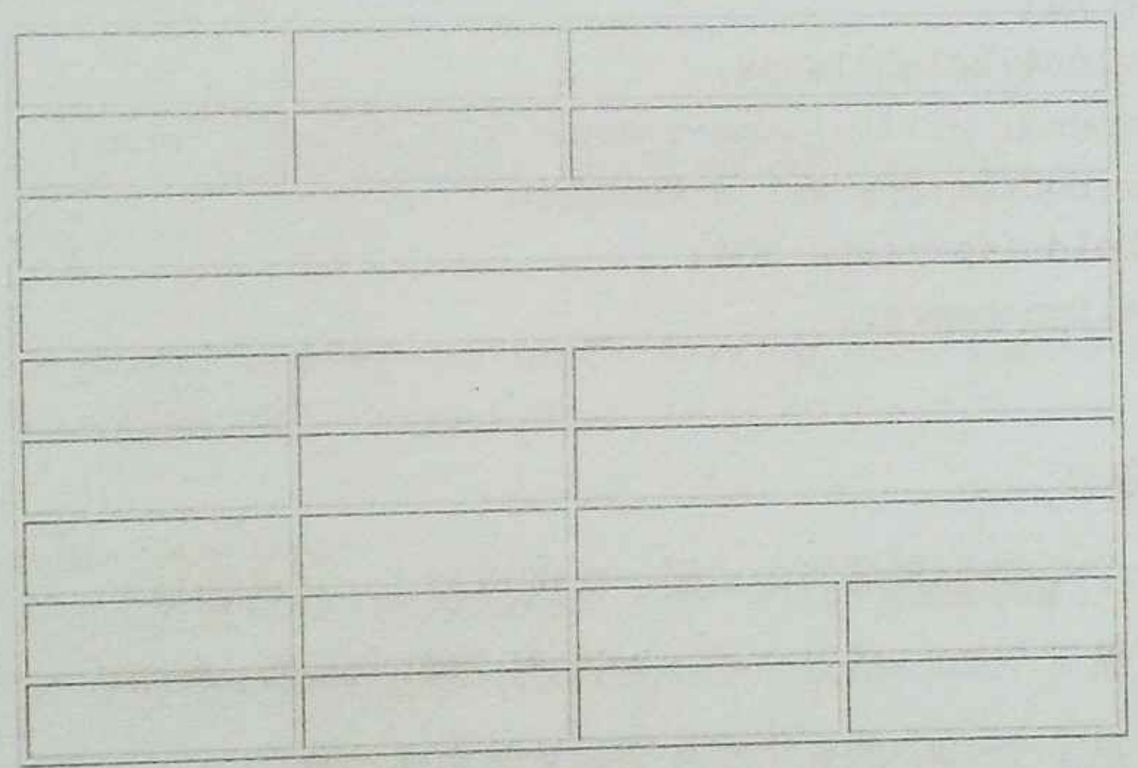


图 10.10　Dreamweaver 设计视图中实现后的布局

- 在各单元格中填入相应的文字及图片内容。

对应的 HTML 组织结构如下所示。

```
<div id="main">
  <h1>联想 Y450A-TFO(NBA 纪念版) 512 独显超强性价比</h1>
  <div class="f_l">  /* "f_l"表示左浮动的类样式,对应的 CSS 在后续的 CSS 文件中*/
    <ul><li class="bigimg"><img src="images/show1_big.jpg" alt="alt" />
  </li></ul>
  <ul class="smallimg">
    <li><img src="images/show1.jpg" alt="alt" /></li>
    <li><img src="images/show2.jpg" alt="alt" /></li>
    <li><img src="images/show3.jpg" alt="alt" /></li>
    <li><img src="images/show4.jpg" alt="alt" /></li>
    <li><img src="images/show5.jpg" alt="alt" /></li>
```

```
      </ul>
      <ul>
        <li class="op"><img src="images/share.gif" alt="alt" /></li>
        <li class="op"><img src="images/addfav.gif" alt="alt" /></li>
      </ul>
    </div>
    <table class="goodinfo f_l">
      <!-- 9 行 4 列表格,代码略 -->
    </table>
  </div>
```

2. CSS 样式实现

1）标题部分

设置字体大小 14px，行高 40px，粗体，居中对齐，并有 1px 灰色实线底边框。

设置背景图及偏移，如下所示。

```
div#main h1{
  width:100%;height:40px;
  background: url(../images/goodtitle_bg.gif) repeat-x 0px -3px;
  border-bottom:1px solid #dddddd;
  font:bold 14px/40px 宋体;
  text-align:center;
  }
```

2）左列部分

- 第一个<ul>设置适当内边距，大图设置 1px 灰色边框。
- 设置第 2 个<ul>的五个<li>左浮动，第三个<ul>的两个<li>左浮动。
- 修饰第 2 个<ul>的五个图片 2px 灰色实线边框。
- 设置第 3 个<ul>的两个<li>宽度为 180px。

```
.bigimg{padding:15px 35px 15px 0px;}
.smallimg li{float:left;width:65px;}
.smallimg img{border:2px #ddd solid;}
.op{width:180px;}
```

3）右列部分

- 表格宽 350px，行高 26px，字符间距 0.5px，并设置单元格 1px 灰色虚线下边框。

```
 .goodinfo{
    width:350px;
    line-height:26px;
    letter-spacing:0.5px;
 }
 .goodinfo td{border-bottom:1px #ddd dashed;}
```

- “立即购买”单元格高 74px，无边框，图片上下内边距 17px，垂直居中。

```
td.buynow{height:74px;line-height:74px;border:0px;}
```

- 贵美商城支持网银，设置背景图及背景色，然后设置边框和内边距如下。

```
td.onlinepay {
background:#FFFFED url(../images/onlinepay.gif) no-repeat 5px 8px;
border:1px solid #DDCF8F;
line-height:16px;
padding:5px 0px 3px 76px;
}
```

根据上述思路，对应的完整的 HTML 和 CSS 代码如示例 3 所示，注意思考普通思路存在的优缺点。

示例 3

普通思路 gooddetail.htm 代码：

```
<div id="main" class="good">
    <h1>联想 Y450A-TFO(NBA 纪念版) 512 独显超强性价比</h1>
    <div class="f_l">
    <ul>
        <li class="bigimg"><a href="#"><img src="images/show1_big.jpg"
        alt="笔记本大图" /></a></li>
    </ul>
    <ul class="smallimg">
        <li><img src="images/show1.jpg" alt="笔记本小图 1" /></li>
        <!-- 笔记本小图 2-5,类似,略 -->
    </ul>
    <ul class="op">
      <li><a href="#"><img src="images/share.gif" alt="分享" /></a></li>
      <li><a href="#"><img src="images/favthis.gif" alt="藏" /></a></li>

    </ul>
    </div>
    <table class="goodinfo f_l">
        <tr>
          <td>一 口 价:</td><td>80.00 元 </td>
          <td colspan="2"> </td>
        </tr>
        <tr>
          <td>运    费:</td>
          <td>免运费</td><td colspan="2"> </td>
        </tr>
        <tr><td colspan="4" class="h74 buynow"><a href="#"><img src=
        "images/btn-buynow.gif" alt="alt" /></a></td></tr>
```

```
<tr><td colspan="4" class="onlinepay">此商品支持<a href="#">网银
支付</a>,网上汇款免手续费。<br />收货满意后出售者才能拿钱,货款都安全!
</td></tr>
<tr>
  <td>剩余时间:</td>
  <td>10 天 23 小时</td>
  <td colspan="2"> </td>
</tr>
<tr>
  <td>本期售出:</td>
  <td>2 件</td>
  <td colspan="2"> </td>
</tr>
<tr>
  <td>累计售出:</td>
  <td>8 件</td>
  <td colspan="2"> </td>
</tr>
<tr>
  <td>商品类型:</td><td>全新</td>
  <td>所 在 地:</td><td>山西太原</td>
</tr>
<tr>
  <td>商品数量:</td><td>1 件</td>
  <td>浏 览 量:</td><td>17 次</td>
</tr>
    </table>
  </div>
```

普通思路 gooddetail.htm 引用的 layout.css 样式代码:

```
/*商品信息页*/
div#main{width:100%;}
div#main h1{
  width:100%;height:40px;
  background:url(../images/goodtitle_bg.gif) repeat-x 0px -3px;
  border-bottom:1px solid #dddddd;
  font:bold 14px/40px 宋体;
  text-align:center;
}
    .bigimg{padding:15px 35px 15px 0px;}
    .smallimg li{float:left;width:65px;}
    .smallimg img{border:2px #ddd solid;}
    .op{clear:both;}
    .op li{float:left;width:180px;}
    .goodinfo{
```

```
        width:350px;
        line-height:26px;
        letter-spacing:0.5px;
        }
    .goodinfo td{border-bottom:1px #ddd dashed;}
      td.buynow{height:74px;line-height:74px;border:0px;}
      .buynow img{padding:17px 0px;vertical-align:middle;}
td.onlinepay{background:#FFFFED url(../images/onlinepay.gif) no-repeat
5px 8px;
border:1px solid #DDCF8F;line-height:16px;
      padding:5px 0px 3px 76px;
     }
/* 常用样式*/
.clear{clear:both;}
.w175{width:175px;}
.h74{height:74px;}
```

10.4.2 用简洁思路实现

学习了使用常规的普通思路实现后，现在改变实现思路，根据右边数据并列关系的特点，改为 ul-li 结构实现，分析能否精简代码。

1. 页面布局分析

整体采用 div-ul-li 的结构，但右列采用<ul>结构，并且各<ul>块内的内容均用<li>实现，其组织结构如下。

```
<div id="main">
  <h1></h1>
  <ul class="f_l">  /*需设置左浮动*/
    <li></li>
    <!-- 多个 li,略-->
  </ul>
  <ul class="f_l">  /*需设置左浮动*/
    <li></li>
    <!-- 多个 li,略-->
  </ul>
  </ul>
</div>
```

显然，层次结构只有三级，即 div-ul(h1)-li，标题外其他各项都用<li>表示。我们可以通过合理设置各块的宽高与浮动来实现布局，最大程度地减少了冗余标签，如图 10.11 所示。

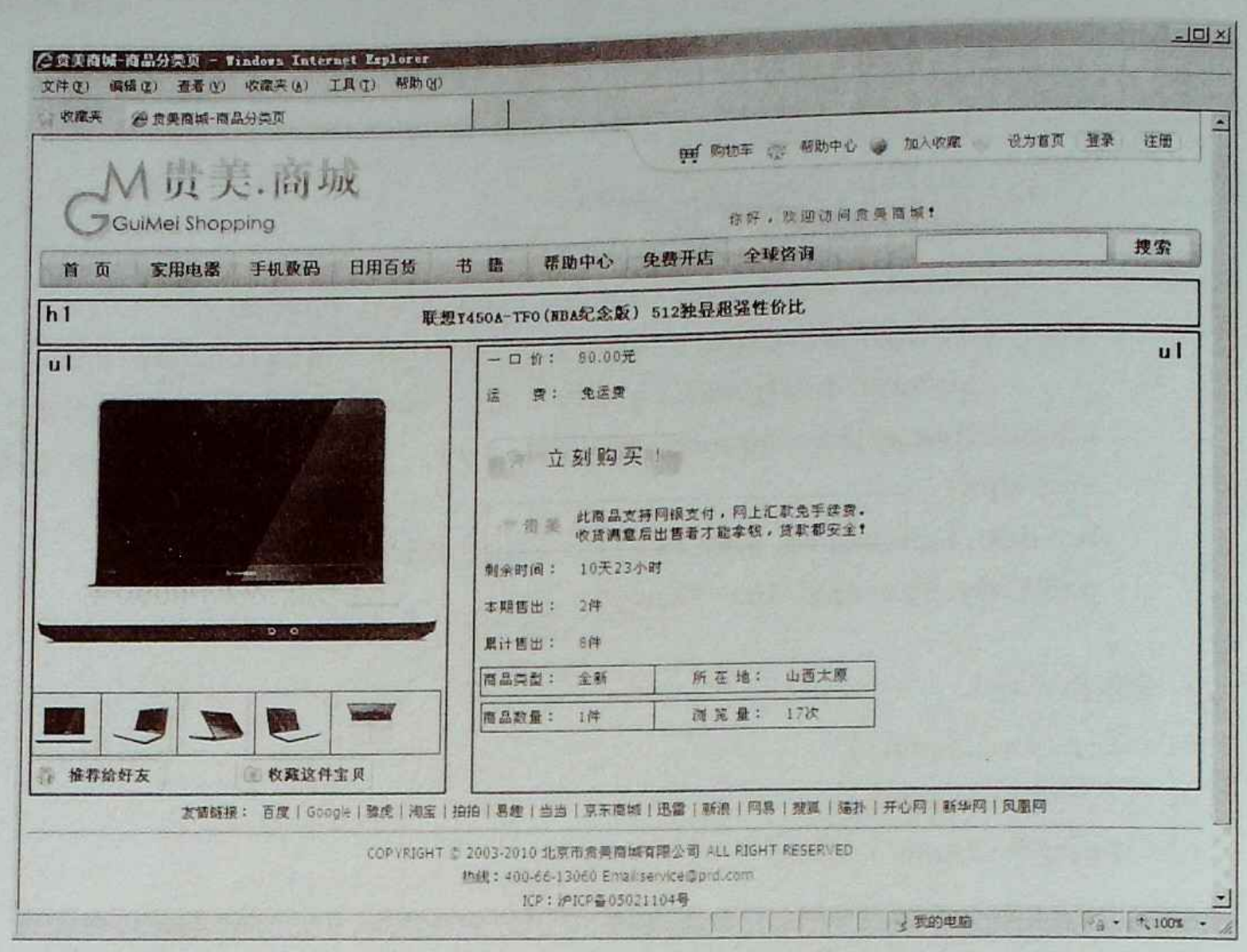


图 10.11　简洁思路的实现示意图

2. CSS 样式的实现

1）标题部分

与普通实现一样，参见相关样式的设置。

2）左侧部分

- 总宽 350px，右内边距 35px。<li>分三类：大图、小图和推荐收藏图。
- 大图<li>上下内边距 15px，图片 1px 灰色实线边框。
- 小图<li>左浮动，宽度 65px，图片 2px 灰色实线边框。
- 推荐收藏图<li>区隔前面小图的浮动，左浮动，宽 175px。

```
/*左侧大小图*/
.good ul{width:350px;padding-right:35px;}
  .bigimg{padding:15px 0px;}
  .smallimg{float:left;width:65px;}
  .smallimg img{border:2px #ddd solid;}
  .op{width:180px;}
  .goodinfo{
      width:350px;letter-spacing:0.5px;
      line-height:30px;
  }
```

3）右侧

- 总宽 350px，行高 30px，总共 11 个<li>，每个<li>都有 1px 灰色虚线下边框。
- 最后四个<li>宽度为 175px，左浮动。其中第 10 个<li>区隔浮动。

- “立即购买”与“贵美支持网银”参见普通思路的实现。
- 为冒号后文字增加标签并设左内边距 15px，以便排版。

```
  /*右侧商品文字信息*/
 .goodinfo{
  width:350px;letter-spacing:0.5px;
  line-height:30px;
 }
 .goodinfo li{border-bottom:1px #ddd dashed;}
 li.buynow{height:74px;line-height:74px;border:0px;}
 li.buynow img{padding:17px 0px;vertical-align:middle;}
  li.onlinepay{
 background:#FFFFED url(../images/onlinepay.gif) no-repeat 5px 8px;
   border:1px solid #DDCF8F;
 line-height:16px;
   padding:5px 0px 3px 76px;
  }
 .goodinfo li span{padding-left:15px;}
/* 常用样式 */
.clear{clear:both;}
.w175{width:175px;}
.h74{height:74px;}
```

根据上述思路，对应的完整的 HTML 和 CSS 代码如示例 4 所示，注意仔细体会简洁思路与普通思路的区别。可以看出，简洁思路的实现步骤比普通思路相对简洁，组织结构简单。但这种思路需要不断地思考总结，在深刻理解各块级元素与行级元素的特征的基础上，用较少的样式代码优化页面的实现。

示例 4

gooddetail.htm 代码：

```
<div id="main" class="good">
  <h1>联想 Y450A-TFO(NBA 纪念版) 512 独显超强性价比</h1>
  <ul class="f_l">
    <li class="bigimg"><a href="#"><img src="images/show1_big.jpg" alt="笔
  记本大图" /></a></li>
  <li class="smallimg"><img src="images/show1.jpg" alt="小图 1" /></li>
  <li class="smallimg"><img src="images/show2.jpg" alt="小图 2" /></li>
  <li class="smallimg"><img src="images/show3.jpg" alt="小图 3" /></li>
  <li class="smallimg"><img src="images/show4.jpg" alt="小图 4" /></li>
  <li class="smallimg"><img src="images/show5.jpg" alt="小图 5" /></li>
  <li class="op clear f_l"><a href="#"><img src="images/share.gif" alt="
  分享" /></a></li>
  <li class="op f_l"><a href="#"><img src="images/favthis.gif" alt="藏"
  /></a></li>
</ul>
<ul class="goodinfo f_l">
```

```
<li>一 口 价:<span>80.00 元 </span></li>
<li>运    费:<span>免运费</span></li>
<li  class="h74  buynow"><a  href="#"><img  src="images/btn-buynow.gif"
alt="alt" /></a></li>
<li class="onlinepay">此商品支持<a href="#">网银支付</a>,网上汇款免手续费。
<br />收货满意后出售者才能拿钱,货款都安全!</li>
<li>剩余时间:<span>10 天 23 小时</span></li>
<li>本期售出:<span>2 件</span></li>
<li>累计售出:<span>8 件</span></li>
<li class="f_l w175">商品类型:<span>全新</span></li>
<li class="f_l w175">所 在 地:<span>山西太原</span></li>
<li class="f_l clear w175">商品数量:<span>1 件</span></li>
<li class="f_l w175">浏 览 量:<span>17 次</span></li>
  </ul>
</div>
```

gooddetail.htm 对应的 CSS 样式代码:

```
/*商品信息页*/
.good{width:100%;}

/*商品标题*/
.good h1{
  width:100%;height:40px;
  background:url(../images/goodtitle_bg.gif) repeat-x 0px -3px;
  border-bottom:1px solid #dddddd;
  font:bold 14px/40px 宋体;
  text-align:center;
  }

/*左侧大小图*/
.good ul{width:350px;padding-right:35px;}
  .bigimg{padding:15px 0px;}
  .smallimg{float:left;width:65px;}
  .smallimg img{border:2px #ddd solid;}
  .op{width:180px;}

  /*右侧商品文字信息*/
  .goodinfo{
      width:350px;letter-spacing:0.5px;
      line-height:30px;
    }
.goodinfo li{border-bottom:1px #ddd dashed;}
  li.buynow{height:74px;line-height:74px;border:0px;}
  li.buynow img{padding:17px 0px;vertical-align:middle;}
  li.onlinepay{
   background:#FFFFED url(../images/onlinepay.gif) no-repeat 5px 8px;
   border:1px solid #DDCF8F;
   line-height:16px;
```

```
        padding:5px 0px 3px 76px;
      }
    .goodinfo li span{padding-left:15px;}
/* 常用样式 */
.clear{clear:both;}
.w175{width:175px;}
.h74{height:74px;}
```

10.4.3 两种思路的对比分析

通过上述分析及具体案例的实现，下面对比分析它们的区别，可以看出简洁思路具有如下优点。

- HTML 组织结构精简，语义性较好。

  例如，右侧商品信息的实现，相对普通思路采用表格实现而言，简洁思路只用了一个结构层次，一个<ul>中包含 11 个<li>。这种方式比 table-tr-td 表格方式语义化更好，结构层次较少。

  这样的结构方便搜索引擎快速搜索，更符合目前流行的 SEO（Search Engine Optimization，搜索引擎优化）思想，提高网站的搜索排名等。

- 样式代码复用性高。

  例如，右侧商品信息中的 11 个<li>，简洁思路采用先实现其大多数<li>的通用样式，包括宽度、下边框、行高等，然后再对特殊的最后四个<li>和两个图片<li>进行设置。

10.4.4 上机练习

上机练习 2

练习——利用模板页实现贵美商城商品具体介绍页

需求说明

使用简洁思路实现，如图 10.11 所示。

本章总结

- 用<iframe>实现页面顶部和底部的复用，实现步骤如下。

  (1) 分离顶部和底部作为一个单独的页面文件。

  (2) 在网站的页面中使用<iframe>引用分离的页面文件，这样方便保持网站风格的统一。
- 商品分类列表页实现的要点如下。
  - 采用模板页，中间部分内容采用表格布局。
  - 编写单独的 CSS 样式，利用盒子属性，设置表格宽度、内边距等属性。
- 实现具体介绍页时有以下两种思路。
  - 普通思路：按常规思路，对整个结构进行明晰的划分，部分采用局部表格实现。
  - 简洁思路：采用 div-ul-li 实现精简结构，去除冗余标签，使用浮动来实现灵活布局。

本章作业

一、选择题

1. 复用页面的思路是（　　）。

A. 使用 CSS 样式引用复用页面

B. 将待复用页面的 HTML 代码拷贝到新页面中

C. 使用<frameset>框架页，设置某个窗口的 src 为待复用文件

D. 使用<iframe>标签，设置 src 引用待复用文件

2. 关于示例 1 中商品分类列表页的布局思路的说法，正确的是（　　）。

A. 局部采用表格布局，并用 Dreamweaver 能较快地实现所需的效果

B. 整个页面都采用表格布局

C. 整个页面采用的是 div-ul-li 布局

D. 应该尽量减少使用表格，因为表格很慢

3. 贵美商城商品具体介绍页采用普通思路的缺点是（　　）。

A. 可维护性、可扩展性、可复用性差

B. 相对于简洁思路，HTML 代码量会少很多

C. 相对于简洁思路，CSS 代码量会少很多

D. 更利于 SEO（搜索引擎优化）

4. 贵美商品具体介绍页所采用了简洁思路，其说法正确的是（　　）。

A. 具有很强的结构语义

B. 可以对内容进行最彻底的修饰

C. 简洁思路总是以简洁的布局为基础

D. 可定义性和扩展性差

5. 贵美商城商品具体介绍页所采用了简洁思路，其实现技巧是（　　）。

A. 灵活使用<li>的各种属性，包括内边距、背景、边框、浮动、清除等进行控制

B. 先提取各类似元素的共性样式，再单独编写各元素的独特样式

C. 简单的结构是实现的基础，因此，HTML 的组织层次越少越好，结构越简单越佳

D. 只考虑结构的简洁，不考虑结构语义

二、简答题

1. 简述顶部、底部复用的具体步骤及注意点。

2. 结合贵美商城商品具体介绍页，说明布局采用简洁思路的好处。

3. 总结常用的样式细节，如常用宽高、对齐浮动、字体粗细、边框置零等，专门整理成一个 CSS 文件，以便日后使用，并说明常用样式细节应放在 CSS 代码中的什么位置。

4. 用简洁思路重新实现商品分类列表页，如图 10.12 所示，具体要求如下。

- 尝试不用表格实现，采用纵向方式组织结构，如图 10.12 中部方框所示。
- 结构层次在三级以内（含 main）。
- 尽量无冗余标签，无冗余命名。
- 页面显示效果完全一致。

图 10.12　商品分类列表页效果图

5. 尝试用简洁思路重新实现贵美商城网站登录页，如图 10.13 所示，具体要求如下。

- 不用表格实现。

- 结构层次在三级以内（含 main）。
- 尽量无冗余标签，无冗余命名。
- 页面显示效果完全一致。

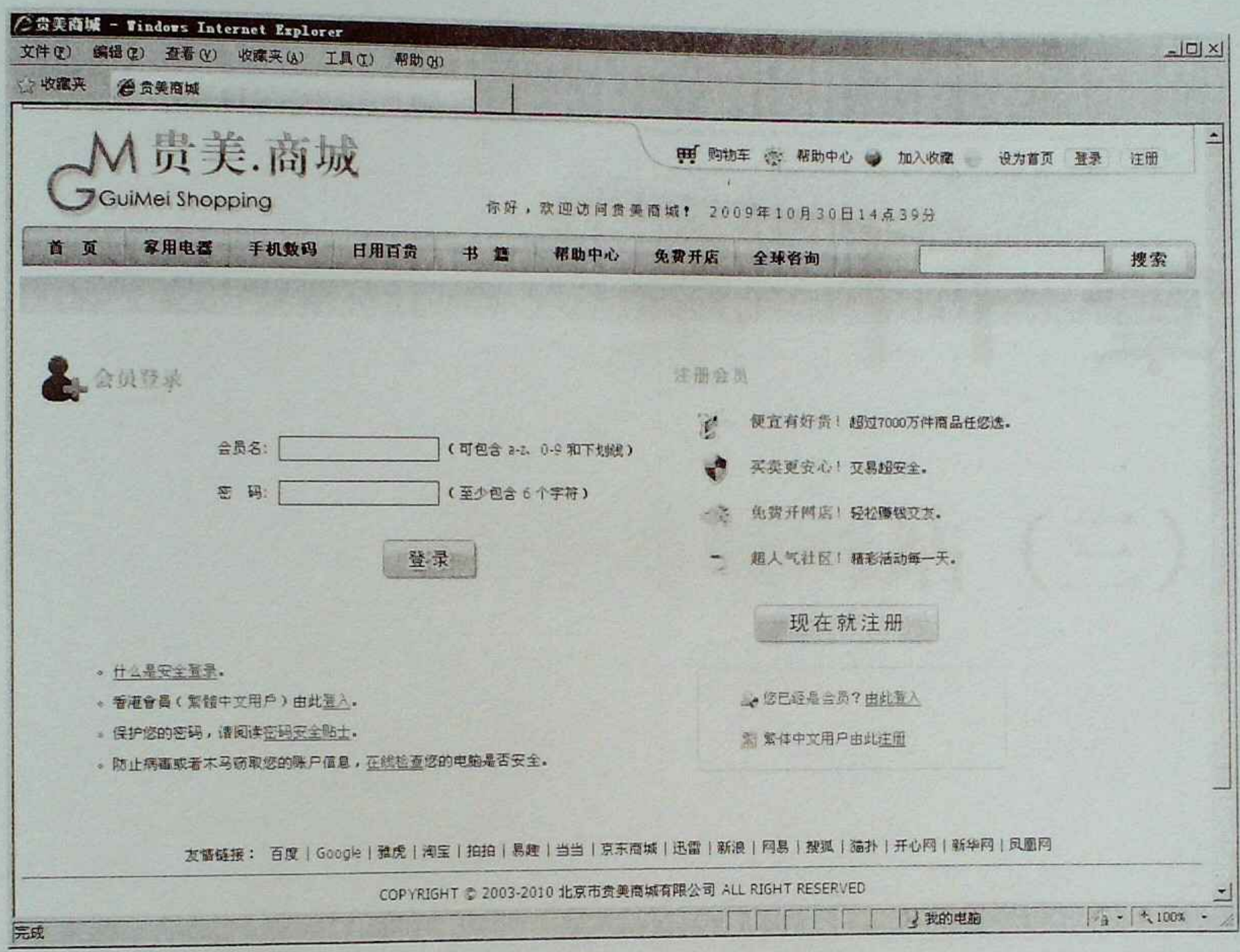


图 10.13　登录页效果图

第 11 章

贵美商城网站（三）

◈本章工作任务

- 制作贵美商城网站注册页
- 制作贵美商城网站客户中心页
- 发布并测试贵美商城网站

◈本章技能目标

- 使用 DIV+CSS 制作表单页面
- 使用 DIV+CSS 制作框架页效果
- 掌握测试和解决网站兼容性的常用办法

本章单词

请在请在预习时学会下列单词的含义和发音，并填写在横线处。

1. success：_____
2. block：_____
3. hack：_____
4. multiple：_____
5. collection：_____
6. validate：_____
7. bug：_____
8. inline：_____
9. if：_____
10. end：_____

本章简介

上章我们用 Dreamweaver 作为辅助工具，制作了除首页以外的两个主要页面：商品分类列表页、商品具体介绍页。本章将重点介绍典型的注册页及客户中心页的制作，其他页面可以在课下自行练习。页面制作完毕后，再在页面间添加链接实现网站导航，最后进行网站的兼容性测试。

11.1 制作贵美商城网站注册页

我们一起分析如何制作如图 11.1 所示的贵美商城网站注册页，该页面的部分内容在第 2 章曾实现过，此处介绍整个页面的实现思路。

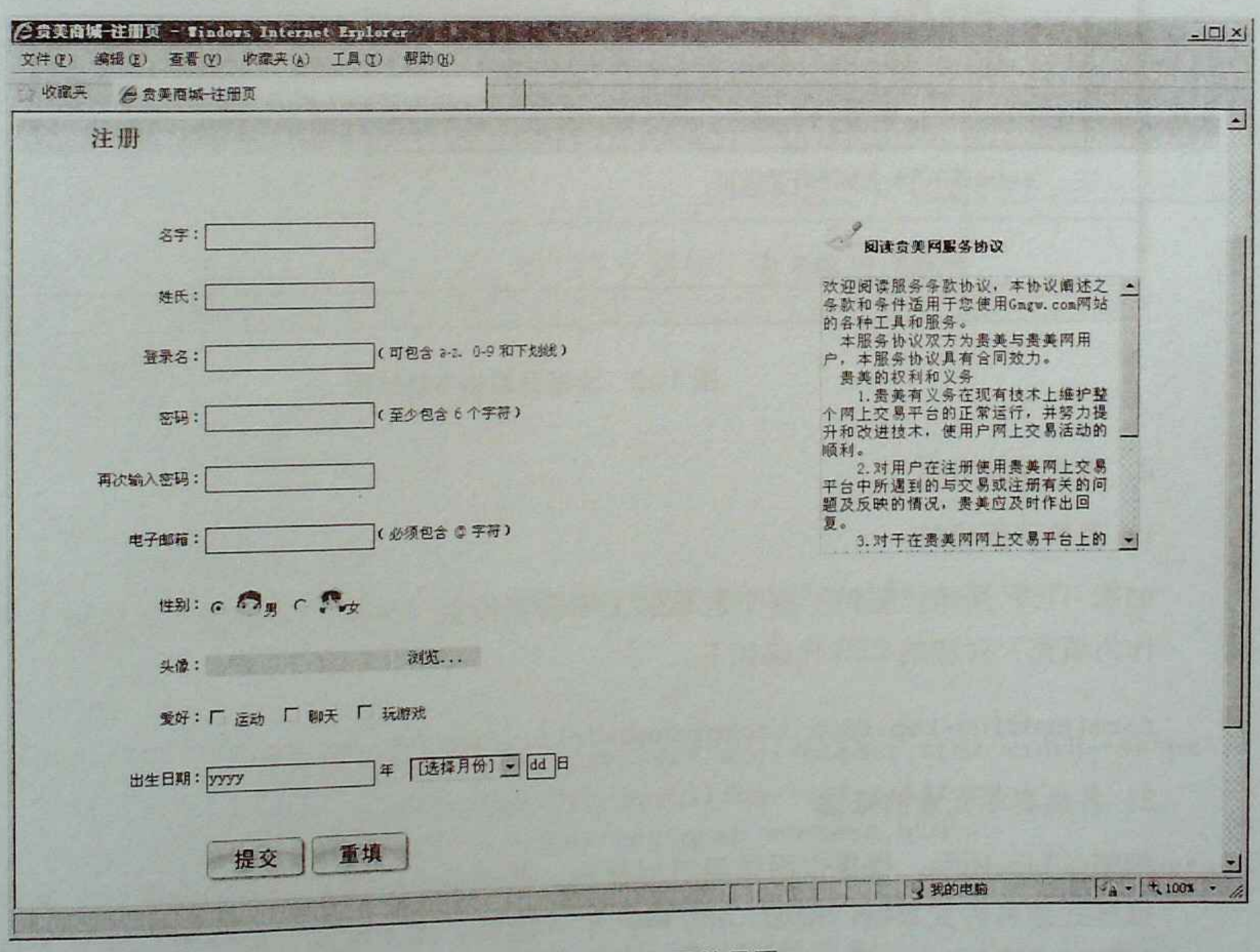


图 11.1 注册页效果图

11.1.1 制作思路

1. 页面结构分析

页面结构很简单，3 列 11 行的表格，右边阅读协议跨 11 行，如图 11.2 所示。

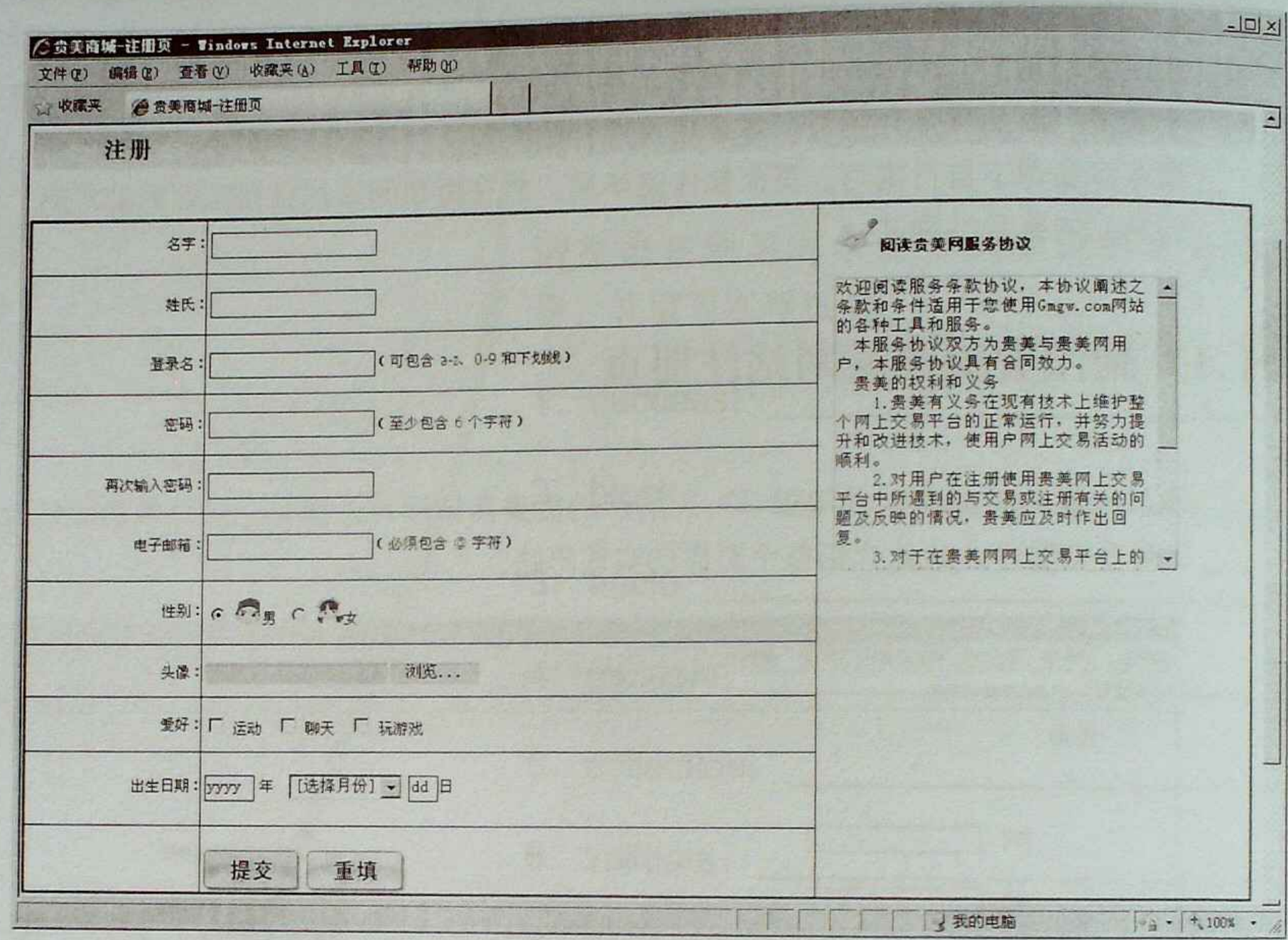


图 11.2 注册页表格布局结构

2. 样式修饰分析

1）表单背景修饰

如图 11.2 所示，显然，整个表单的上方需要设置“注册”背景图，并设置上方内边距作为填充，对应的 CSS 代码如下。

```
form{padding-top:65px;background:url(../images/reg_bg.gif) no-repeat;}
```

2）各类表单元素的修饰

如图 11.2 所示，根据页面效果可以看出，表单输入元素可以分为两类：一类是带细边框样式的各种文本框，例如“用户名”、“密码”输入框等；另一类是非文本框，例如“提交”按钮、复选框等，它们采用默认样式，但宽度设置等还略有不同。

如何实现？传统的方法是创建类（class）选择器，然后各个文本框引用类（class）样式，这里推荐另一种较为简洁的方法。

对于各类文本框，因为它们对应的 HTML 标签都是<input>标签，所以可以在页面顶部定

义所有<input>标签的通用样式：

```
input{width:131px;height:17px;border:1px #333 solid;}
```

然后，再为非文本框定义单独的类样式：

```
.b0{width:auto;height:auto;border:0px; }  //属性 auto 表示自动调节为默认大小
```

另外，对于“服务协议”的文本域（<textarea>标签）需要设置细边框并垂直居中，左列提示文字右对齐，对应的 CSS 代码如下。

```
/*采用区域限制的写法，表示类名为 reg 表格内的文本域样式*/
.reg textarea{border:1px #ccc solid;}
/*各输入框后的提示文字右对齐 */
.a_r{text-align:right;}
```

小结

- 使用表格布局实现注册页中间部分的局部布局。
- 先写通用风格再写特殊风格。
  较好的风格为“border:1px #ccc solid;width:131px;height:17px;”，随后对非文本框进行修饰重置，注意使用 auto 属性。
- 表格单元格可以使用 vertical-align 垂直对齐属性，进行修饰调整。
  其中垂直下对齐为 vertical-align:bottom，垂直上对齐为 vertical-align:top，垂直居中为 vertical-align:middle。

结合上述思路分析，对应的 HTML 和 CSS 代码如示例 1 所示。

示例 1

register.htm 的代码：

```
<head>
  <!--省略部分 HTML 代码 -->
  <link rel="stylesheet" type="text/css" href="css/layout.css"  />
</head>
<body>
  <div id="container">
      <iframe id="header" runat="server" src="header.htm" width="980px"
    height="136px" frameborder="0" scrolling="no"></iframe>
      <form method="post" action="register_success.htm">
      <table id="main" class="reg" cellspacing="0px" cellpadding="0px">
        <tbody>
        <tr>
          <td class="a_r w15p"><label for="fname">名字:</label></td>
        <td class="w500"><input name="fname" id="fname" type="text" /></td>
          <td class="protocol" rowspan="11">
        <h4><img src="images/read.gif" alt="alt" />阅读贵美网服务协议 </h4>
            <textarea cols="35" rows="15">欢迎阅读服务条款协议……</textarea>
```

同时应用两个类，分别实现居中对齐、宽 15%的效果

```
</td>
 </tr>
 <tr>
   <td class="a_r"><label for="lname">姓氏:</label></td>
   <td><input name="lname" id="lname" type="text" /></td>
 </tr>
 <!-- 下面分别为登录名、密码、重输密码、电子邮件,结构类似,略 -->
 <tr>
   <td class="a_r"><label for="gender">性别:</label></td>
<td><input name="gender" id="gender" class="b0" type="radio" value=
"男" checked="checked" />
    <img src="images/male.gif" alt="alt" />男 
    <!-- 选项"女"类似,省略 -->
    </td>
 </tr>
 <!-- "头像"对应的文件域省略,还有"爱好"多选框 -->
 <tr>
   <td class="a_r"><label for="nYear">出生日期:</label></td>
<td>
   <input name="nYear" id="nYear" class="w40" value="yyyy" maxlength=
   "4" /> 年  
  <select name="nMonth">
    <option value="" selected="selected">[选择月份]</option>
    <option value="0">一月</option>
    <!-- 2-12 月类似,略 -->
   </select>
   <input name="nDay" value="dd" class="w20" maxlength="2" />日
  </td>
   </tr>
   <!-- 省略提交和重置按钮-->
   </tbody>
  </table>
  </form>
 <iframe  id="footer"  runat="server"  src="footer.htm"  width="980px"
 height="150px" frameborder="0" scrolling="no"></iframe>
 </div><!--container  end-->
</body>
```

无边框

register.htm 引用的 layout.css 相关代码：

```
/* 注册页 */
form{padding-top:65px;background:url(../images/reg_bg.gif) no-repeat;}
  input{width:131px;height:17px;border:1px #333 solid;}
  .reg textarea{border:1px #ccc solid;}
  .reg .protocol{vertical-align:top;}
/* 常用细节通用风格 */
#main img.hand{cursor:pointer;}
.b0{border:0px;width:auto;height:auto;}
.a_r{text-align:right;}
.w40{width:40px;}
```

```
.w81{width:81px;}
.w500{width:500px;}
```

11.1.2 上机练习

上机练习 1

练习——制作贵美商城网站注册页

需求说明

重新实现贵美商城网站注册页，HTML 页面内容可用 Dreamweaver 辅助输入，但 CSS 代码要求手写。

11.2 制作贵美商城网站客户中心页

贵美客户中心页如图 11.3 所示，我们曾在第 4 章中使用框架技术实现过，现改为用更流行的 DIV+CSS 布局实现，要求实现类似框架窗口间的关联效果。同时，样式方面还增加了动态的 CSS 特效：当鼠标移动到左侧菜单时，将动态显示或隐藏二级子菜单，如图 11.4 所示。

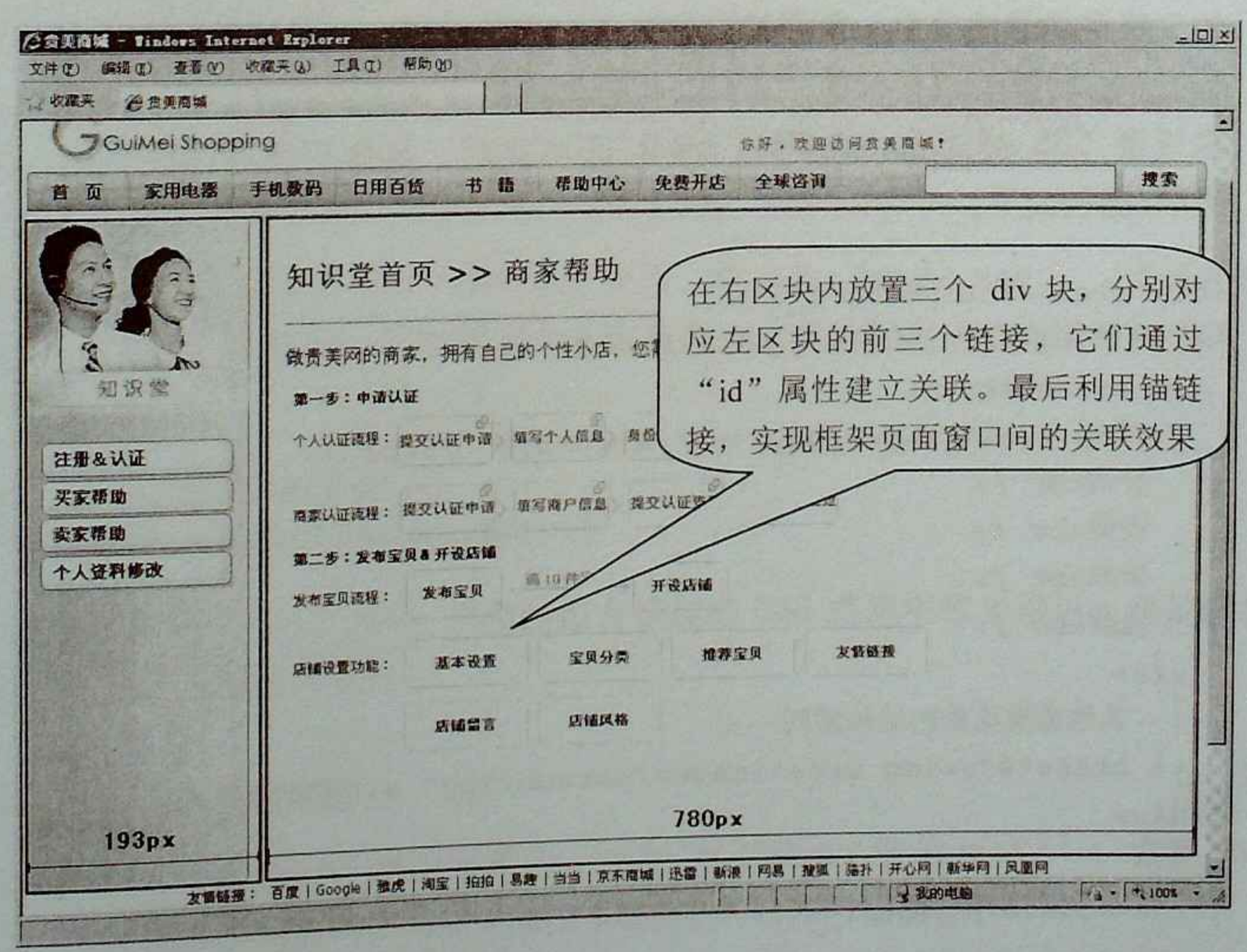


图 11.3 帮助中心分两块

11.2.1 制作思路

1. 结构分析

1）页面的整体组织结构

页面上方仍然采用模板页，中间部分整体高 550px，分左右两个区块，两个区块均设置左浮动。左区块宽 193px，右区块宽 780px。单击左区块内的链接，将在右区块显示相应内容，要实现此功能需要完成两点：一是在右区块内放置链接对应的内容，并给各块标注 id 标识属性，二是左区块内容通过锚链接链接到右区块的各块内容，例如 a href="#buy"，其中“#buy”刚好表示 id 属性为“buy”的 div 块。

本例在右区块内放置三个 块，分别对应左区块前三个链接的三块内容（单击链接时，每次只显示对应块，其他块隐藏），并使用 id 作为标识，以便作为左区块锚链接的锚点。对应的 HTML 结构如下。

```
<div class="helpmenu f_l"></div>
<div class="help f_l">
  <div id="reg"></div>
  <div id="buy"></div>
  <div id="sale"></div>
</div>
```

2）左区块的组织结构

确定了页面的整体结构后，下面再深入分析左区块内部的组织结构。为了实现图 11.4 的二级菜单伸缩效果，需要将父级菜单对应的链接<a>标签设置为块状显示，即将行级标签<a>转换为块级标签，将<a>标签作为二级菜单的容器。行级标签转为块状标签只需设置 CSS 样式为：display：block 即可实现，具体后续再详细介绍。左区块对应的 HTML 结构如下。

```
<div class="helpmenu f_l">
 <!-- "买家帮助"二级菜单对应的 HTML 结构-->
  <a href="#buy">
    <img src="images/buy.jpg" alt="alt" /><br />
    咨询<br />
    投诉<br />
    举报<br />
    退款<br />
  </a>
 <!--其他链接菜单的结构雷同-->
  <a href="#"><img src="images/person.jpg" alt="alt" /></a>
</div>
```

3）右区块的组织结构

右侧内容并不重要，拷贝素材显示即可。为实现页面一致性，修饰 1px 边框，填充 15px。

2. 二级菜单显示/隐藏

1）实现一级菜单，但不显示二级菜单的效果

首先设置<a>标签为块状元素（display:block;），宽高与图片一致，并设内容溢出隐藏（overflow:hidden;），即如果内容超过了容器标签的高度，则不显示超出的内容。

这是因为按默认情况，容器高度将随包含的内容自动调整，如本例的<a>标签包含的子菜单，将自动显示全部内容，如图 11.4 左图所示为其效果。但通过设置<a>标签容器的高度和链接图片（如“买家帮助”）一致，同时设置内容溢出隐藏的样式，则可以达到隐藏子菜单的效果，如图 11.4 右图所示为其效果。

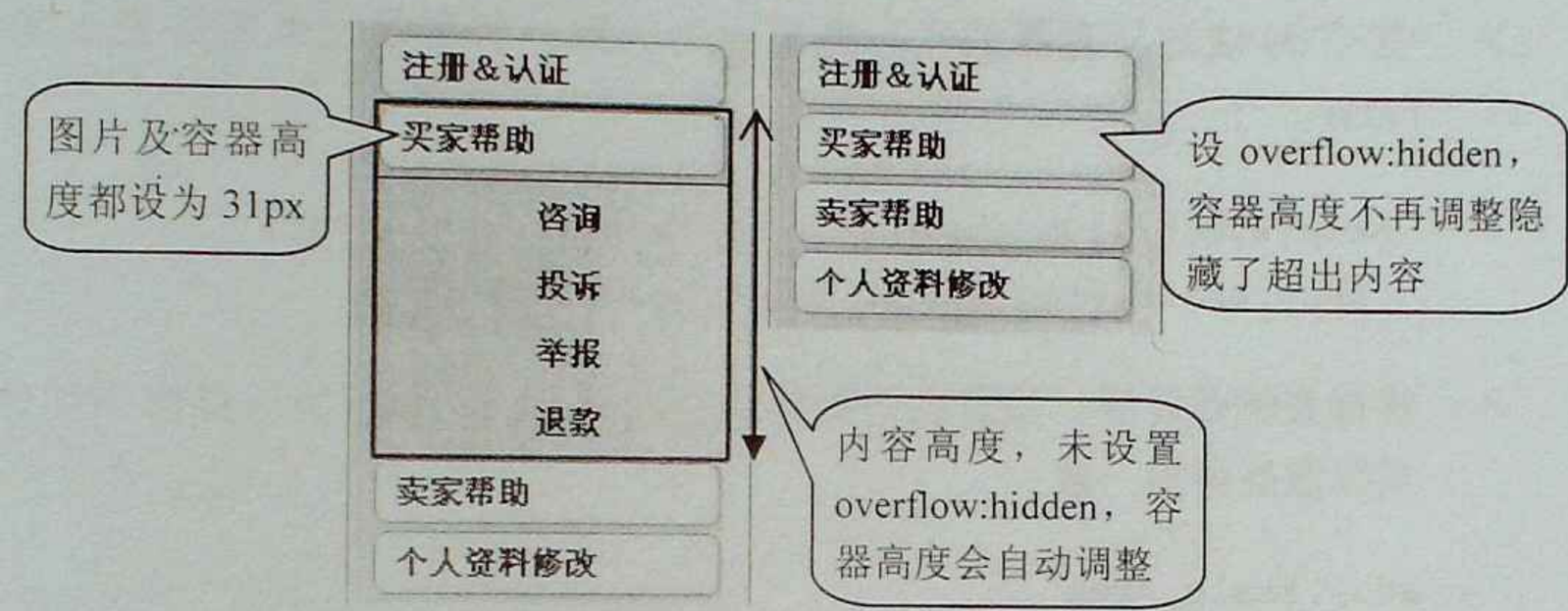


图 11.4　设置固定宽高来隐藏溢出内容

对应的 CSS 代码如下。

```
/*设置<a>默认不显示二级菜单的效果*/
 .helpmenu  a{
       margin:0px auto;width:158px;  /*水平居中*/
       height:31px;display:block;overflow:hidden;  /*内容溢出不显示*/
  }
```

2）实现鼠标移到一级菜单，显示二级菜单的效果

设置链接的鼠标移上状态（a：hover）的样式，样式为“height:auto;”，表示鼠标移上时，容器<a>标签的高度会自动调整为内容高度，从而达到显示二级菜单的效果，对应的 CSS 代码为：

```
/*鼠标移上显示二级菜单的效果*/
    .helpmenu a:hover{height:auto;}
```

需要注意：overflow 与 a：hover 伪类的组合使用在 CSS 应用中较为常见，是实现各种纯 CSS 动态效果的重要技术之一。

3. 实现左右区块间的关联

➢ 左侧各<a>标签的 href 地址设置为“href="#右侧 div 块的 id 名"”，右侧 div 块的 id 名与左侧<a>标签 href 地址对应，即此时 id 名作为锚链接。

```
……<!-- 左侧 -->
<div class="helpmenu">
```

```
  <a href="#reg">……<!-- 非重点,略 --></a>
  <a href="#buy">……<!-- 非重点,略 --></a>
  <a href="#sale">……<!-- 非重点,略 --></a>
</div>
……<!-- 右侧 -->
<div class="help">
  <div id="reg">……<!-- 非重点,略 --></div>
  <div id="buy">……<!-- 非重点,略 --></div>
  <div id="sale">……<!-- 非重点,略 --></div>
</div>
```

- 三个 div 块与父容器的实际宽高设为一致，父容器设置内容溢出隐藏（"overflow: hidden;"）。

```
.help{width:780px;height:550px;overflow:hidden;}
  #buy,#sale,#reg{width:780px;height:550px;}
```

- 修饰右侧各区块，设置内边距作为空白填充，边框 1px，并修饰字体大小，注意和实际宽高保持一致。

```
#buy,#sale,#reg
    {font-size:16px;height:518px;width:748px;padding:15px;border:1px
    #ce6f77 solid;}
```

需要注意：这是另外一个组合 CSS 技术，即使用锚链接和 div 块的 id 名建立关联，可用于在同一区域显示不同的内容，实现类似框架窗口的关联效果。

结合上述思路，完整的 HTML 和 CSS 代码如示例 2 所示。

示例 2

help.htm 的代码：

```
<head>
<!--省略部分 HTML 代码 -->
 <link rel="stylesheet"  type="text/css" href="css/layout.css"  />
</head>
<body>
    <div id="container">
        <iframe id="header" runat="server" src="header.htm" width="980px"
        height="136px" frameborder="0" scrolling="no"></iframe>
        <div class="helpmenu f_l">
         <a href="#reg">
            <img src="images/reg.jpg" alt="alt" /><br />
            点此注册<br />
            认证中心<br />
         </a>
         <a href="#buy">
            <img src="images/buy.jpg" alt="alt" /><br />
```

```
        咨询<br />
        投诉<br />
        举报<br />
        退款<br />
    </a>
    <a href="#sale">
        <!-- 与上面类似,略 -->
    </a>
     <a href="#"><img src="images/person.jpg" alt="alt" /></a>
    </div>
    <div class="help f_l">
        <div id="reg">
            <!-- 非重点,略 -->
        </div><!-- reg end -->
        <div id="buy">
            <!-- 非重点,略 -->
        </div><!-- buy end -->
        <div id="sale">
            <!-- 非重点,略 -->
        </div><!-- sale end -->
    </div>
    <iframe id="footer" runat="server" src="footer.htm" width="980px"
        height="150px" frameborder="0" scrolling="no"></iframe>
    </div><!--container end-->
</body>
```

help.htm 引用的 layout.css 相关风格代码：

```
/* 客户中心 */
.helpmenu{width:180px;height:370px;padding:180px 13px 0px 0px;
    background:url('../images/customer.jpg') no-repeat;}
/*设为块状元素,高度和链接图片一致、内容溢出隐藏,从而实现"隐藏子菜单"的效果*/
 .helpmenu  a{
        margin:0px auto;width:158px;
        height:31px;display:block;overflow:hidden;
  }
/*鼠标移上则高度自适应,即显示子菜单*/
    .helpmenu a:hover{height:auto;}
  .helpmenu li{font:bold 14px 宋体;color:#333;
        line-height:30px;text-align:center;}
  .help{width:780px;height:550px;overflow:hidden;}
  #buy,#sale,#reg {height:518px;width:748px; font-size:16px;
        padding:15px;border:1px #ce6f77 solid;}
  #reg li{background:url('../images/list_icon.gif') no-repeat;
        padding-left:15px;}
```

11.2.2 上机练习

上机练习 2

练习——制作贵美商城网站客户中心页

需求说明

用超链接伪类、锚链接与 overflow 的组合技术，重新实现贵美商城网站客户中心页，如图 11.3 所示。

11.3 建立页面间的链接

一个网站包含多个页面，多个页面间需要通过链接实现网站的导航，方便浏览者访问。创建页面间的链接包括两个方面：一是分析页面间的链接关系，二是建立链接并测试链接是否可用，因为浏览者是最讨厌无任何内容的“空链接”。

11.3.1 分析页面间的链接关系

根据需求分析结果，贵美商城网站页面间的链接关系如表 11-1 所示。

表 11-1 贵美网站页面及其具体含义

页面	含义	链接目标
header.htm	顶部导航	由于是<iframe>的引用页面，所以 target 必须设置为_top 或_blank，即在整个页面窗口或新建窗口中打开
footer.htm	底部版权	
index.htm	首页整站导航	三块：商品分类->catlist.htm；疯狂抢购->info.htm；商品上架->info.htm
buy.htm	购物车	只有特定页面链接，如计算器、提交页和返回商品分类页等
catlist.htm	商品分类列表	一块：列表->info.htm
info.htm	商品具体介绍	主要是商品购买页->buy.htm
help.htm	帮助中心	页面与框架间的跳转
login.htm	登录页	登录信息处理页
register.htm	注册页	注册信息处理页

11.3.2 链接页面及测试修正

1. 链接页面

从上述页面含义和分块中，我们可以很容易地填写相应页面的链接，在静态实现时只需要把相同类信息链接到同一个页面即可。我们以首页建立页面间链接为例，直观展示这个过程。先看图 11.5 中的主页区块及链接。

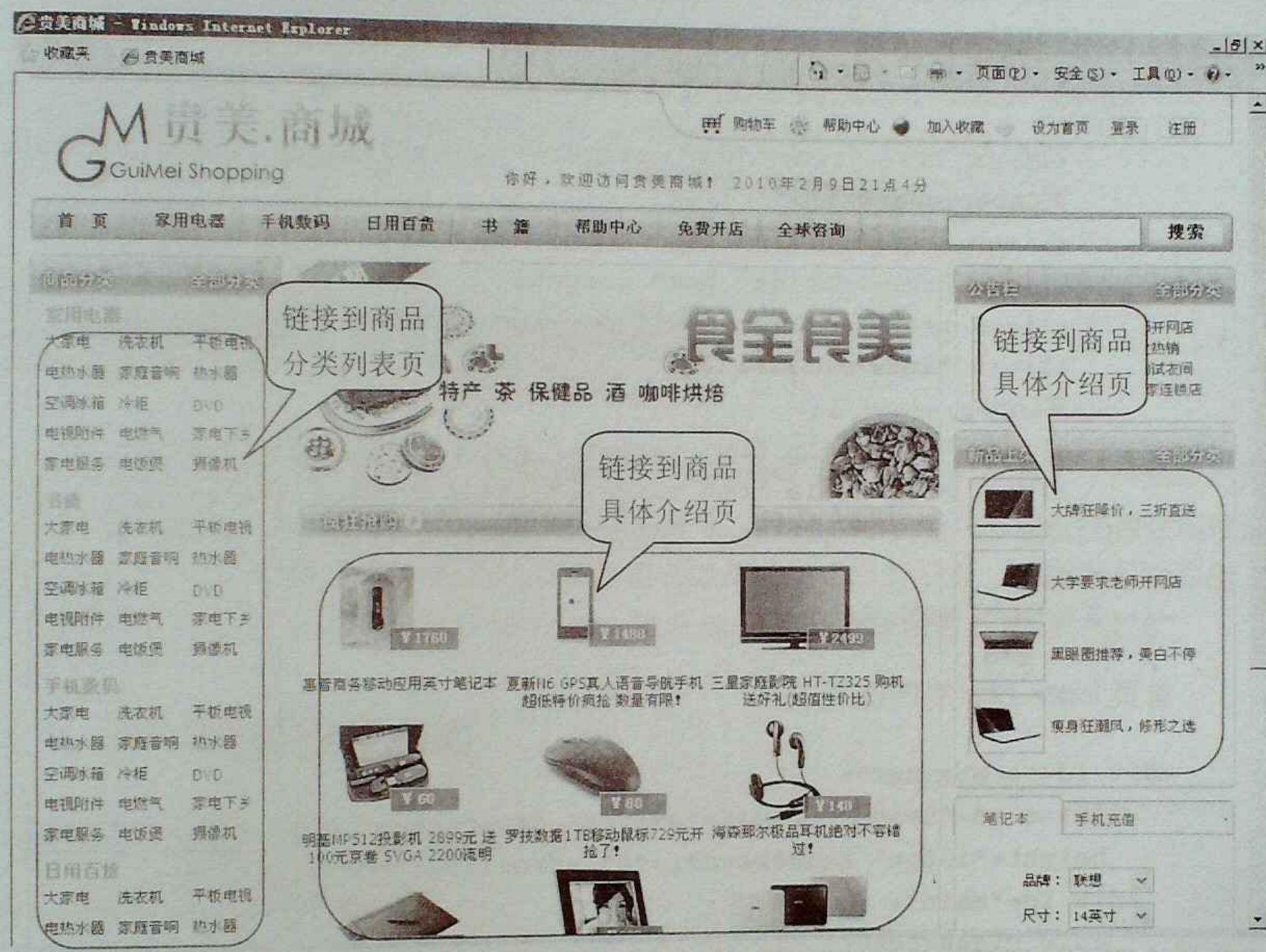


图 11.5 首页各区块中的超链接目标

根据图 11.5，我们设置超链接的 href 属性即可，例如商品分类列表页(catlist.htm)等。一般在制作网页时需先搭建网站结构和基本确定各页面的名字，如有调整可用批量替换更改。

2. 超链接检测与修正

建立完链接后，要点击每个超链接，以便保证符合如下条件。

- 没有无效链接。
- 从用户角度考虑，链接名与链接内容是否良好对应，链接关系是否显而易见，方便用户使用。

根据上述思路，对应的 HTML 代码如示例 3 所示。

示例 3

顶部页 header.htm 的代码：

```
<!-- 外层结构不含超链接,省略 -->
  <li class="h_text"><a href="buy.htm" target="_parent">购物车</a></li>
      <li class="pic pic2"></li>
      <li class="h_text"><a href="help.htm" target="_parent">帮助中心
    </a></li>
      <!-- 菜单 3-5 及登录注册按钮省略 -->
    <ul>
      <li><a  href="index.htm"  target="_parent"> 首   
      页 </a></li>
      <li><a href="catlist.htm" target="_parent">家用电器</a></li>
      <li><a href="catlist.htm" target="_parent">手机数码</a></li>
      <li><a href="catlist.htm" target="_parent">日用百货</a></li>
      <li><a  href="catlist.htm"  target="_parent"> 书   
      籍 </a></li>
      <li><a href="help.htm" target="_parent">帮助中心</a></li>
      <li><a href="help.htm" target="_parent">免费开店</a></li>
      <li><a href="help.htm" target="_parent">全球咨询</a></li>
    </ul>
<!--省略号部分 HTML 代码-->
```

首页 index.htm 的代码：

```
<div id="container">
  <iframe id="header" runat="server" src="header.htm" width="980px"
    height="136px" frameborder="0" scrolling="no"></iframe>
  <div id="main">
    <div class="cat">
      <ul>
       <h1>家用电器</h1>
       <li><a href="catlist.htm">大家电</a></li>
       ……<!-- 多个 li,略 -->
      </ul>
      ……<!-- 多个 ul-li 结构,略 -->
    </div> <!--cat end-->
    <div id="content">
      <div class="content_top"></div>
      <div class="content_list">
        <dl>
          <dt><img src="images/promote-1.jpg" alt="alt" /></dt>
         <dd><a href="info.htm">惠普商务移动应用英寸笔记本</a></dd>
       </dl>
       ……<!-- 多个 dl-dt-dd 结构,略 -->
      </div><!-- content_list end -->
    </div><!-- content end -->
```

```
<div class="sidebar">
  <div class="r1"></div>
  <div class="r2">
    <dl>
      <dt><img src="images/show1.jpg" alt="alt" /></dt>
      <dd><a href="info.htm">大牌狂降价,三折直送</a></dd>
    </dl>
    ……<!-- 多个dl-dt-dd结构,略 -->
  </div> <!--r2  end-->
  </div> <!--sidebar end-->
</div> <!--main end-->
<iframe id="footer" runat="server" src="footer.htm" width="980px"
  height="150px" frameborder="0" scrolling="no"></iframe>
</div><!-- container end -->
```

11.4 网站的兼容性测试及解决办法

根据前期客户方面的需求分析，根据网站用户使用的浏览器的类别和版本，对网站的所有页面进行测试。国内主流的浏览器毋庸置疑是 IE，其次，部分用户可能会使用性能较好的 Firefox。为了确保所有用户看到的页面内容和效果相同或尽量一致，网站制作完毕后，一般都要求对目前主流浏览器 IE 和 Firefox 进行测试，并根据测试结果修订页面。

11.4.1 网页的兼容性测试

测试网站页面的兼容性一般包括如下两种。

- 检查浏览器之间的兼容性。

  根据客户使用的各种浏览器产品及其版本，测试网页的显示效果是否一致。根据国内情况，一般只要求保证 IE 和 Firefox（火狐）兼容即可。目前 Firefox 一般为 Firefox 3.5，IE 一般要求从 IE 6.0 到 IE 8.0 即可。这意味着需要同时安装 IE 的多个版本，可以从网上下载 IE Collection 等软件，实现 IE 多个版本的共存。

  安装好 Firefox 3.5 和 IE 的多个版本后，就可以测试同一网站在不同浏览器的显示效果了。

- 验证是否符合 W3C 相关标准。

  可以使用以下任一方法进行检测，也可以同时进行检测。

  - 制作网页时使用 Dreamweaver 进行测试，Dreamweaver 将给出可能存在的兼容性提示。
  - 网页制作完成后使用 Firefox 插件进行检测。
  - 发送到 validator.w3.org 官方网站进行在线验证。

下面逐一详细介绍如下。

1. 使用 Dreamweaver 测试是否符合 W3C 标准

1）单页测试

打开贵美商城网站的某个网页，依次点击“文件(F)”→“验证”→“标记(M)”，稍后即会在底部验证里出现提示，如图 11.6 所示。

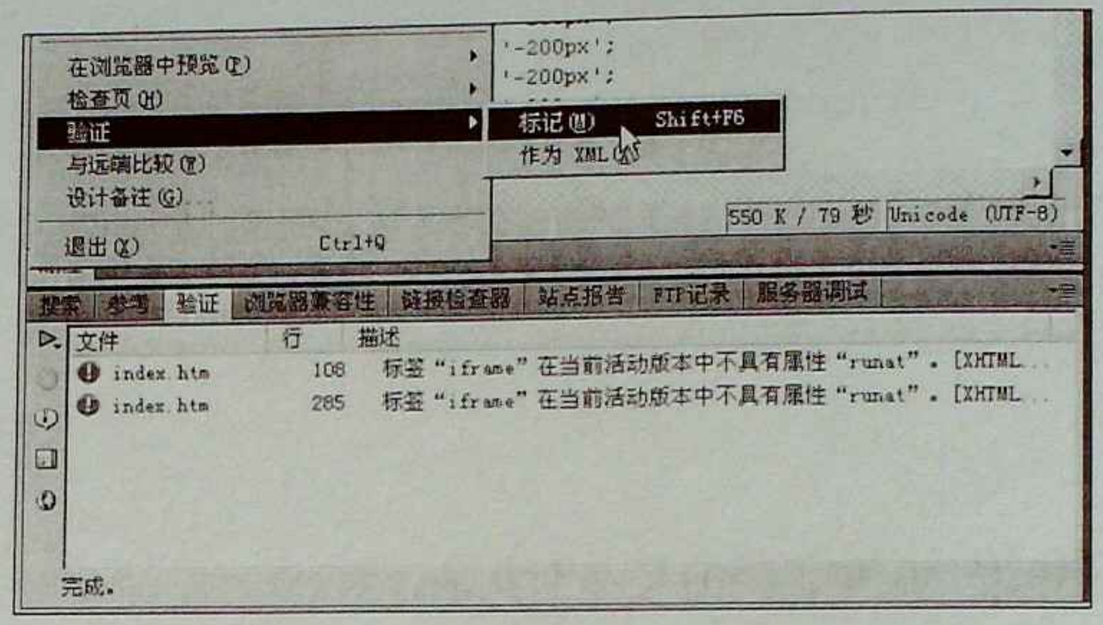


图 11.6　Dreamweaver 中单页标签验证

2）整站测试

在文件窗口中，选中“站点 - 贵美商城”，然后依次点击下方属性窗口中的“验证”标签→绿色三角箭头→“验证整个当前本地站点”，稍后即会在右侧窗口出现提示，如图 11.7 所示。

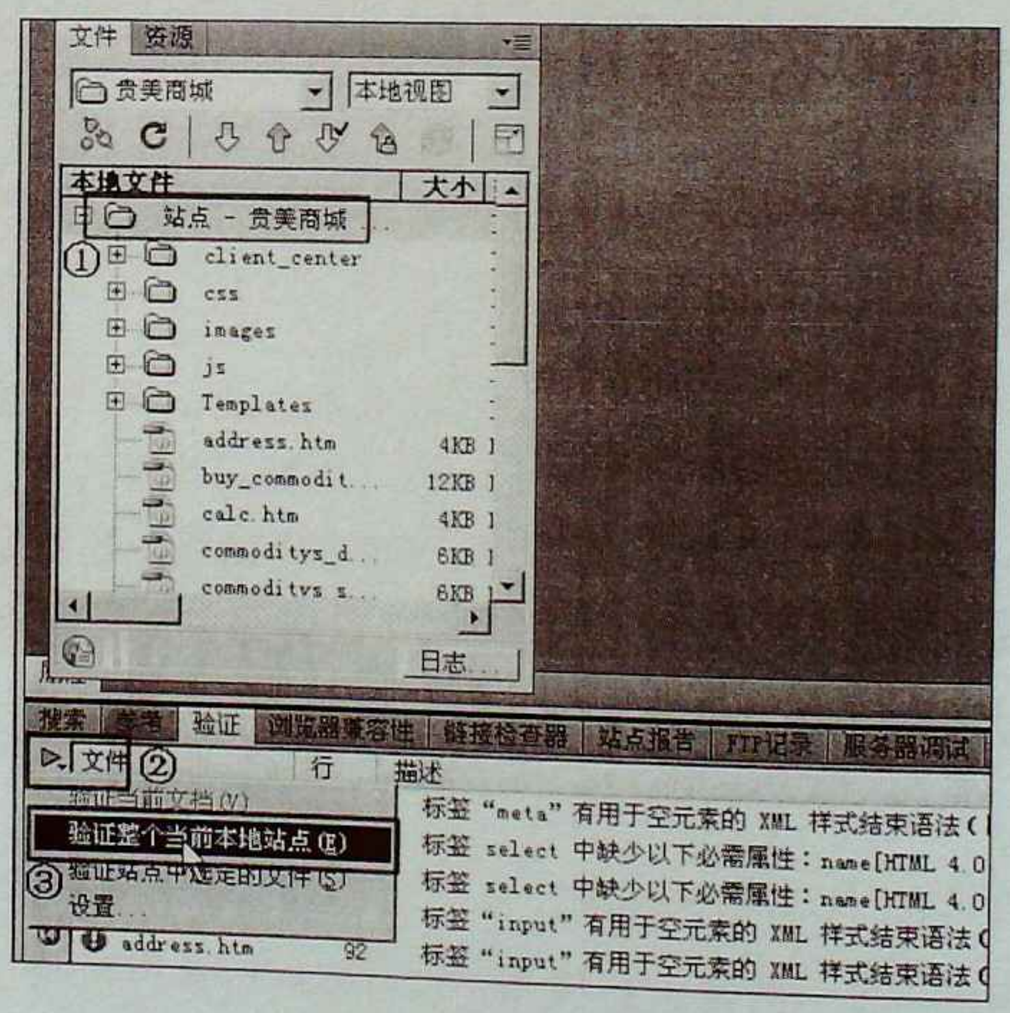


图 11.7　Dreamweaver 中整站多页面标签验证

2. 用 Firefox 插件进行兼容性检测

先安装 Firefox 3.5，然后从网上下载 Firefox 的两个插件：W3C 验证插件（“tidy_firefox_win_0858.xpi”）、firebug 脚本调试插件（“firebug-1.4.2-fx.xpi”），

然后选择使用“Firefox”打开进行安装（注意不能采用传统的双击打开进行安装），如图 11.8 所示。

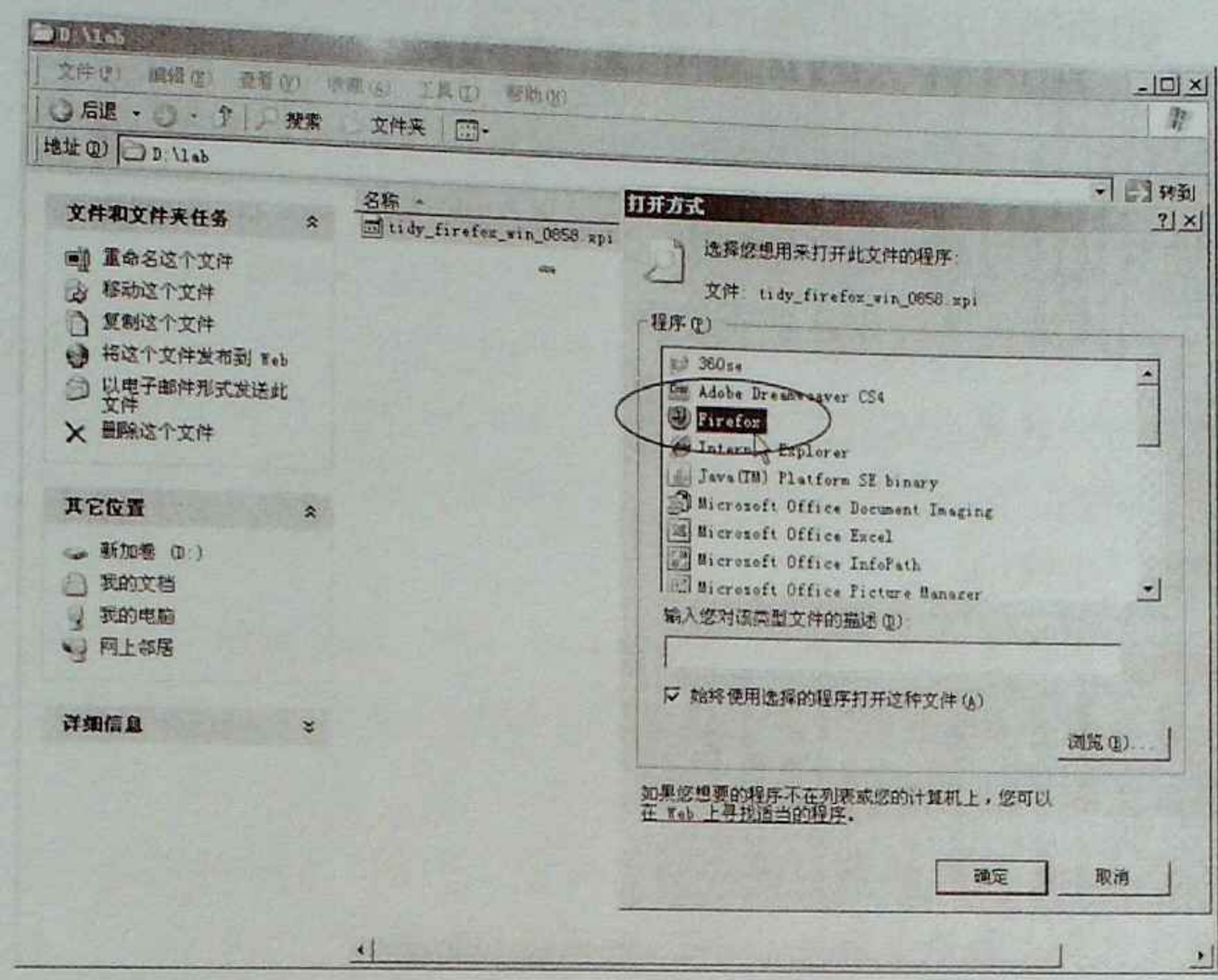


图 11.8　HTML Validator 使用界面

安装完成后，打开某个 HTML 文件，在页面中右击鼠标，选择“查看源代码”，将出现如图 11.9 所示的代码不规范的提示，它能帮助我们迅速定位可能出现兼容问题的地方。一般来说，错误标识的问题必须修改，警告标识的问题需要根据具体问题而定，不要求全部修订。

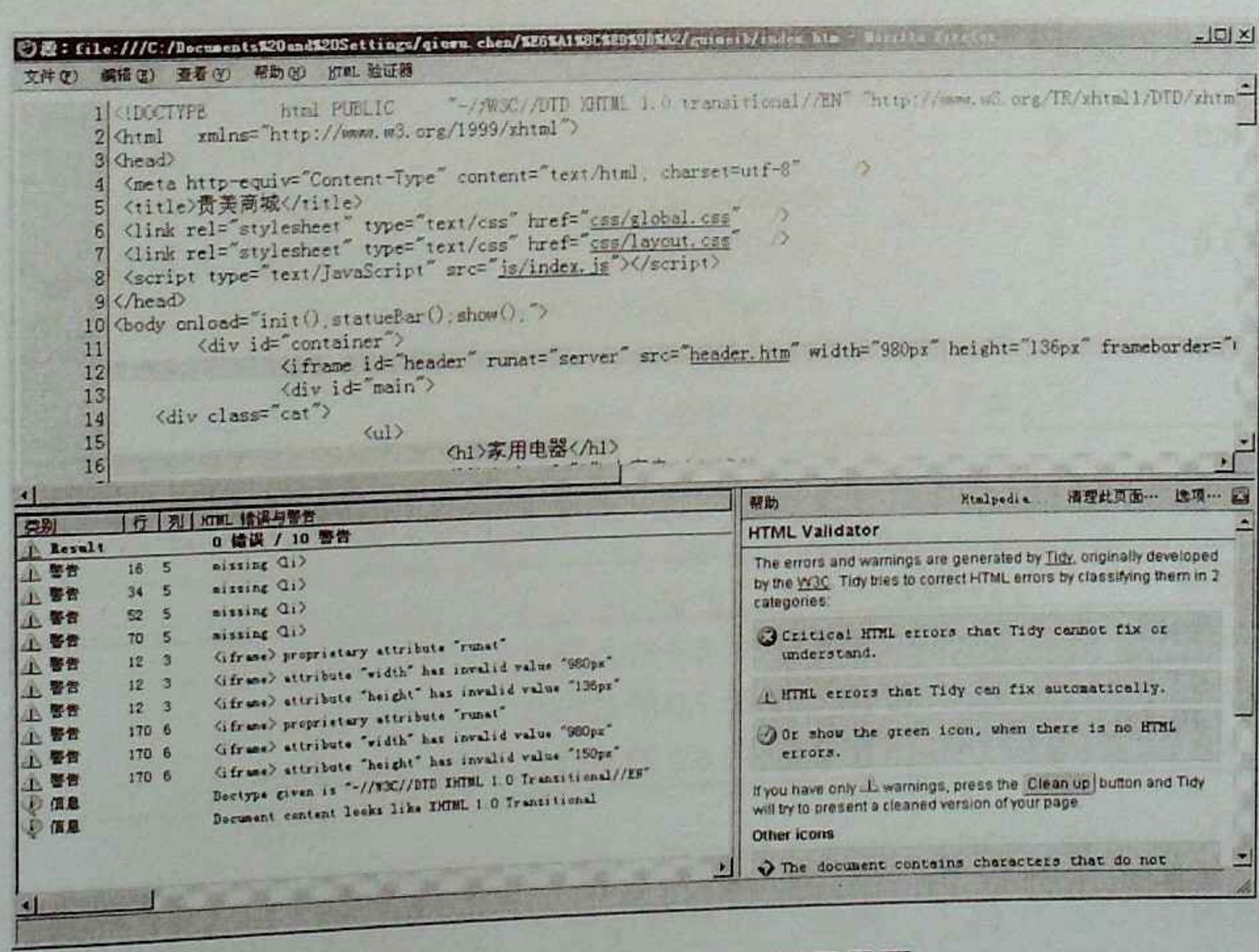


图 11.9　HTML Validator 使用界面

3．通过 W3C 官方网站进行验证

如能上网，还可以将网址发送到 W3C 的官方验证网站（http://validator.w3.org）进行

在线验证，根据页面提示进行相应操作即可，这里不再详细介绍。

11.4.2 如何解决浏览器的兼容问题

同一个网页，在不同浏览器中的显示效果不一样。究其原因，是因为不同浏览器对同一网页代码的解释不同，特别是 CSS 样式方面。例如，页面内容和浏览器窗口的空白间距，每类浏览器的默认值不一致。再如，对于盒子模型 margin 外边距等属性的设置，Firefox 和 IE 浏览器的解释不一致，相差 2px。因此，为了各浏览器显示同一效果，解决的办法是针对各类浏览器存在的“bug”编写相应的 CSS 样式代码，让各浏览器只解释自己能“识别”的 CSS 样式代码。这些样式代码也称为 CSS hack，即解决浏览器兼容性问题的“小窍门”。

相对而言，各类主流浏览器中，Firefox 对 W3C 相关规范支持最好，有“标准浏览器”之称，其次是 IE 8.0，最容易出现兼容性问题的是 IE 6.0，IE 6.0 以下的过时版本用户很少使用，一般不用考虑。

编写 CSS hack 的思路是根据各浏览器能“识别”的特殊符号，编写针对各浏览器的 CSS 样式代码，各浏览器能识别的特殊符号如表 11-2 所示，IE 各版本还支持“if”语句，如表 11-3 所示。

表 11-2 主流浏览器的 CSS hack 方法

浏览器	特殊符号	符号位置	样式书写顺序
Firefox	（无）	（无）	最先 说明： 实际上 Firefox 支持特殊符号“!important”，但一般先写 Firefox 的 CSS，所以一般很少使用
IE 8.0	\0	语句后、分号前，如 padding:5px;padding:0px\0;	Firefox 与 IE 6.0 中间，注意 font 属性的缩写不能用
IE 7.0	+	语句或标签前，如 p{width:55px};+p{width:60px;}	中间（此符号 IE 6.0 也能识别，因此写中间，常用于 IE 7.0 hack）
IE 6.0	_（下划线） +（加号）	语句或标签前，如 width:80px;_width:60px;	最后

表 11-3 IE 各版本支持的 if 语句

浏览器	语句
IE 6.0	<!--[if IE 6.0]>IE 6.0 的 HTML 代码<![endif]-->
IE 7.0	<!--[if IE 7.0]>IE 7.0 的 HTML 代码<![endif]-->
IE 8.0	<!--[if IE 8.0]>IE 8.0 的 HTML 代码<![endif]-->

了解各浏览器“识别”的特殊符号后，我们就可以利用 CSS 代码的先后顺序编写兼容各浏览器的 CSS 代码了。例如，假定要使某个 div 的样式效果兼容 Firefox、IE 8.0、IE 7.0、IE 6.0，则对应的 CSS 代码可能为：

```
.#mydiv{
    padding: 5px ;  /*Firefox*/
```

```
    padding: 10px\0 ;  /*IE8.0*/
    +padding: 15px ;  /*IE7.0*/
    _padding: 20px ;  /*IE6.0*/
  }
```

以上 CSS 代码，Firefox 只“识别”第一行，所以 CSS 效果为“padding：5px”；对于 IE 8.0，能“识别”前两行，但对于同一 CSS 属性的重复定义，以最后一行为准，所以 CSS 效果为“padding：10px”；同理，IE 7.0、IE 6.0 对应的 CSS 效果分别是“padding：15px”、“padding：20px”。需要注意，IE 6.0 除“识别”“_”号外，还“识别”IE 7.0 支持的“+”号，所以编写顺序一般为先“Firefox”，然后“IE 8.0”或“IE 7.0”，“IE 6.0”必须放在最后。

注意

- 发现浏览器的兼容性问题，除编写 CSS hack 外，还需要先检查网页文档自身是否符合 W3C 相关规范，目前大多数主流浏览器都支持 W3C 相关规范，并且支持的程度越来越好。
- 编写 IE 7.0/IE 8.0 的 CSS hack 方法，只有在文档声明为 XHTML Transitional 或 Strict 时才有效。

下面利用上述 CSS hack 方法，编写如下 CSS 和 HTML 代码，如示例 4 所示。请结合图 11.10 的效果图，仔细体会 CSS hack 的编写方法。

示例 4

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
            "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
    <head>
      <title>hack</title>
    <style>
       body{margin:0px;padding:0px;}  /*修改各浏览器默认预留的页面内容边距*/
       div {
            font:bold 20px 宋体;      /* Firefox*/
            font-size:12px\0;          /* IE 8.0 在末尾用"\0"*/
           +font:normal 24px 黑体;  /* IE 7.0 在开始用"+"*/
           _font:normal 16px 宋体; /* IE 6.0 在开始用"_"*/
            color:red;
            color:green\0;
            +color:purple;
            _color:blue;
            }
    </style>
  </head>
  <body>
    <div>不同浏览器效果。
     <!--[if IE 8.0]>IE8.0<![endif]-->
     <!--[if IE 7.0]>IE7.0<![endif]-->
```

```
        <!--[if IE 6.0]>IE6.0<![endif]-->
        </div>
    </body>
</html>
```

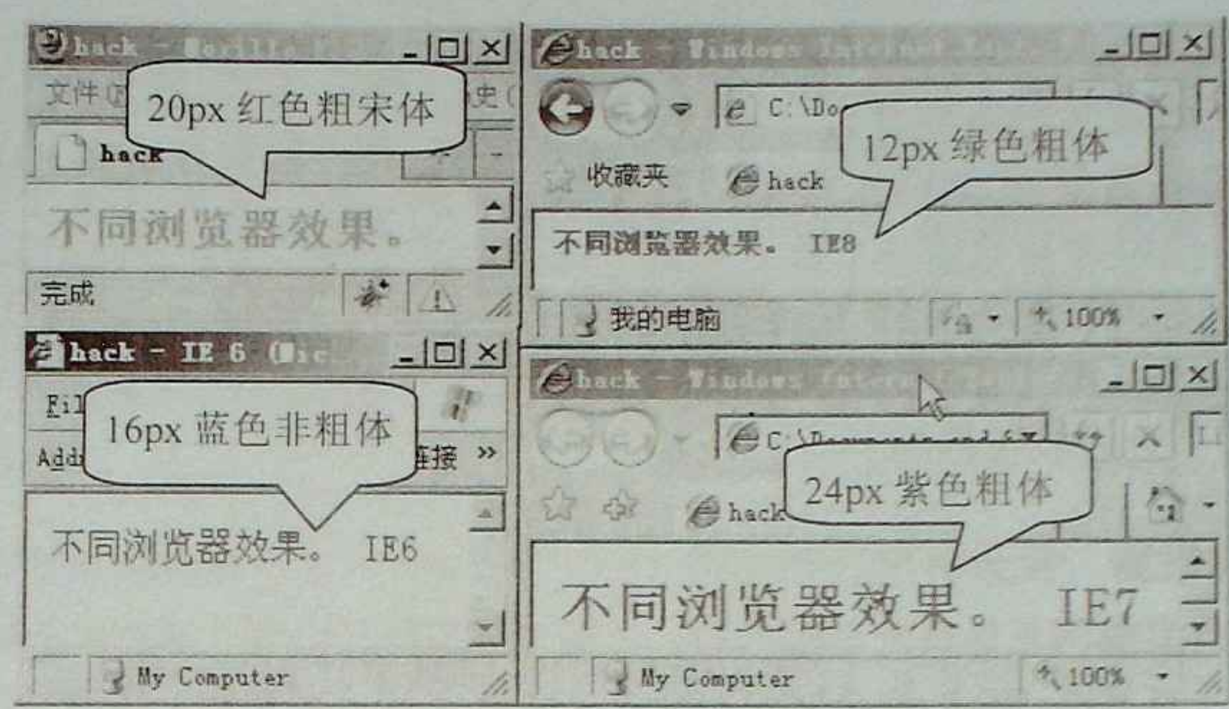


图 11.10　hack 实现各浏览器不同的效果图

除上述常用的方法，对于 IE 的各个版本，还可以根据表 11-3 中的“if”语句，编写不同浏览器对应的外部样式表文件，然后在<head>标签中引用不同的 CSS 文件。

学习 CSS hack 后，我们可以为浏览器兼容问题专门编写对应浏览器的 hack。下面列出常见的兼容性问题及解决方法，其中部分问题需要用到 CSS hack 方法。

1. 三像素文本慢移问题

➢ 出现场景：IE 6.0 中，如在浮动（float）元素后紧跟着一个行内元素，两者之间会有 3 像素的间隙，如图 11.11 所示（IE 6.0~IE 8.0 使用 IE Collection 软件打开），俗称“三像素文本慢移”问题，其他浏览器不存在这个问题。

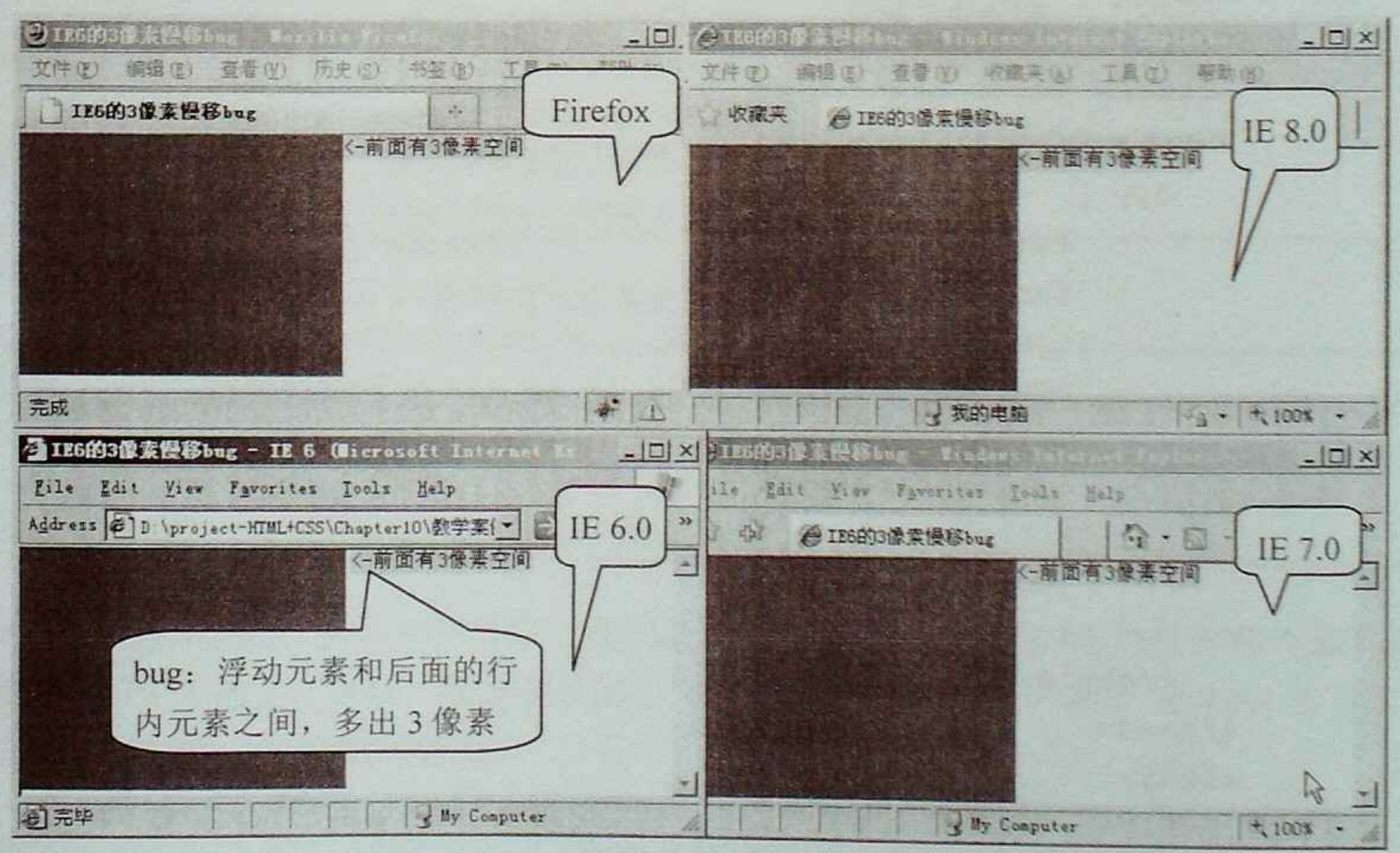


图 11.11　IE 6.0 中的三像素文本慢移 bug

➢ 解决办法：使用 CSS hack，单独为 IE 6.0 设置行内元素为 "_margin-left:-3px;"，关键代码如下。

```
<!--省略部分 HTML 代码-->
<head>
  <style>
    <!--省略部分 CSS 代码-->
    #testdiv{……;float:left;}  /*省略了浮动元素的部分 CSS 代码*/
    span{background:#ccc; _margin-left:-3px;} /*修复 IE6.0 bug*/
 </style>
</head>
<body>
  <div id="container">
       <div id="testdiv"></div><span id="sd">-前面无 3 像素空间</span>
  </div>
</body>
<!--省略部分 HTML 代码-->
```

修复后的效果如图 11.12 所示。

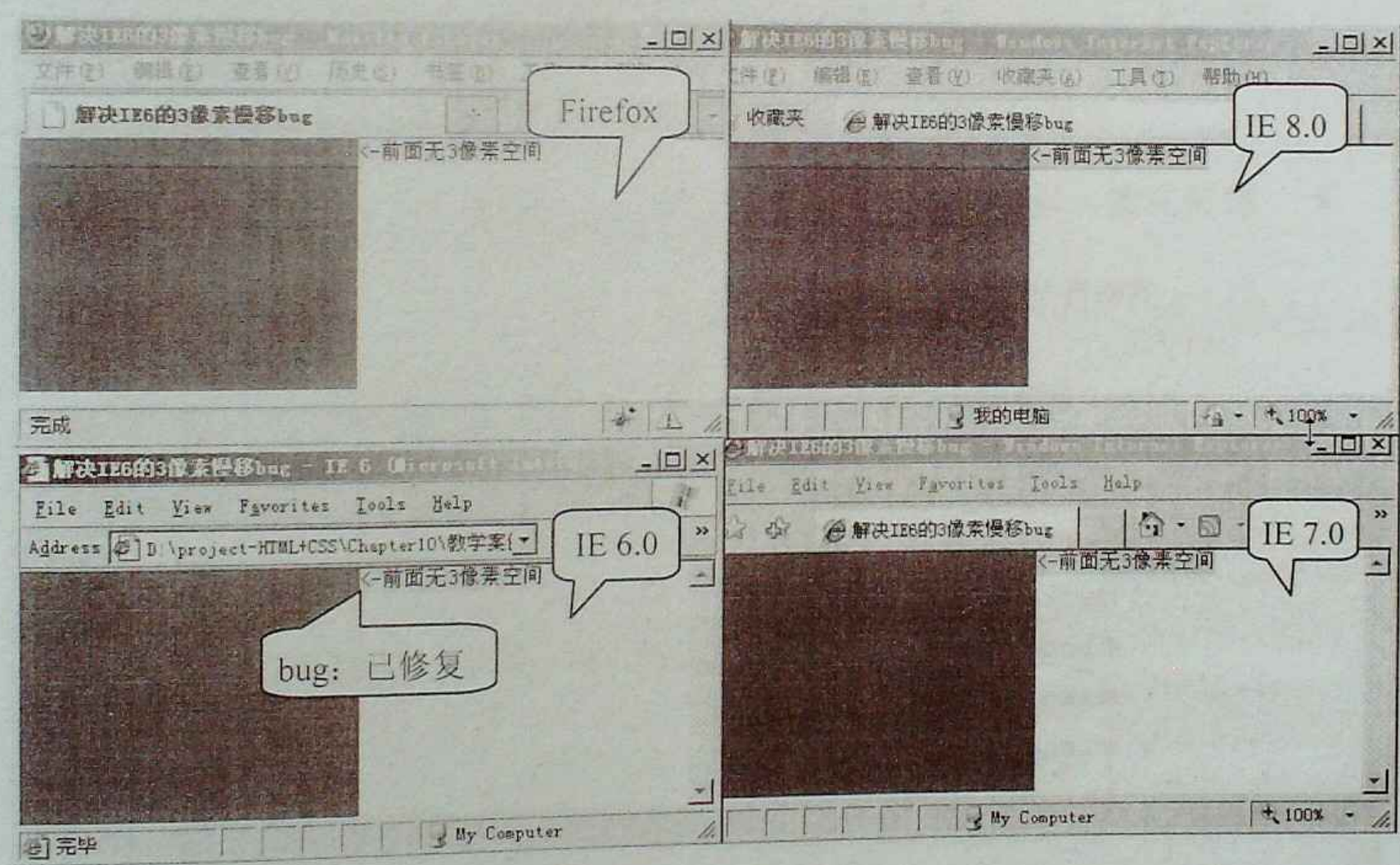


图 11.12　解决 IE 6.0 中的三像素文本慢移 bug

➢ 推荐做法：不要在浮动元素后面出现行内元素。

需要注意：如果未声明文档类型为 XHTML Transitional 或 Strict，则所有 IE 版本都会有此 bug。

2. 双倍边距问题

➢ 出现场景：IE 6.0 中，如果元素的浮动方向与 margin 一致，例如，左浮动元素设置左外边距（"margin-left:50px;"），则外边距会加倍（显示的左外边距为 100px），如图 11.13 所示，俗称"双倍边距"问题，其他浏览器不存在这个问题。

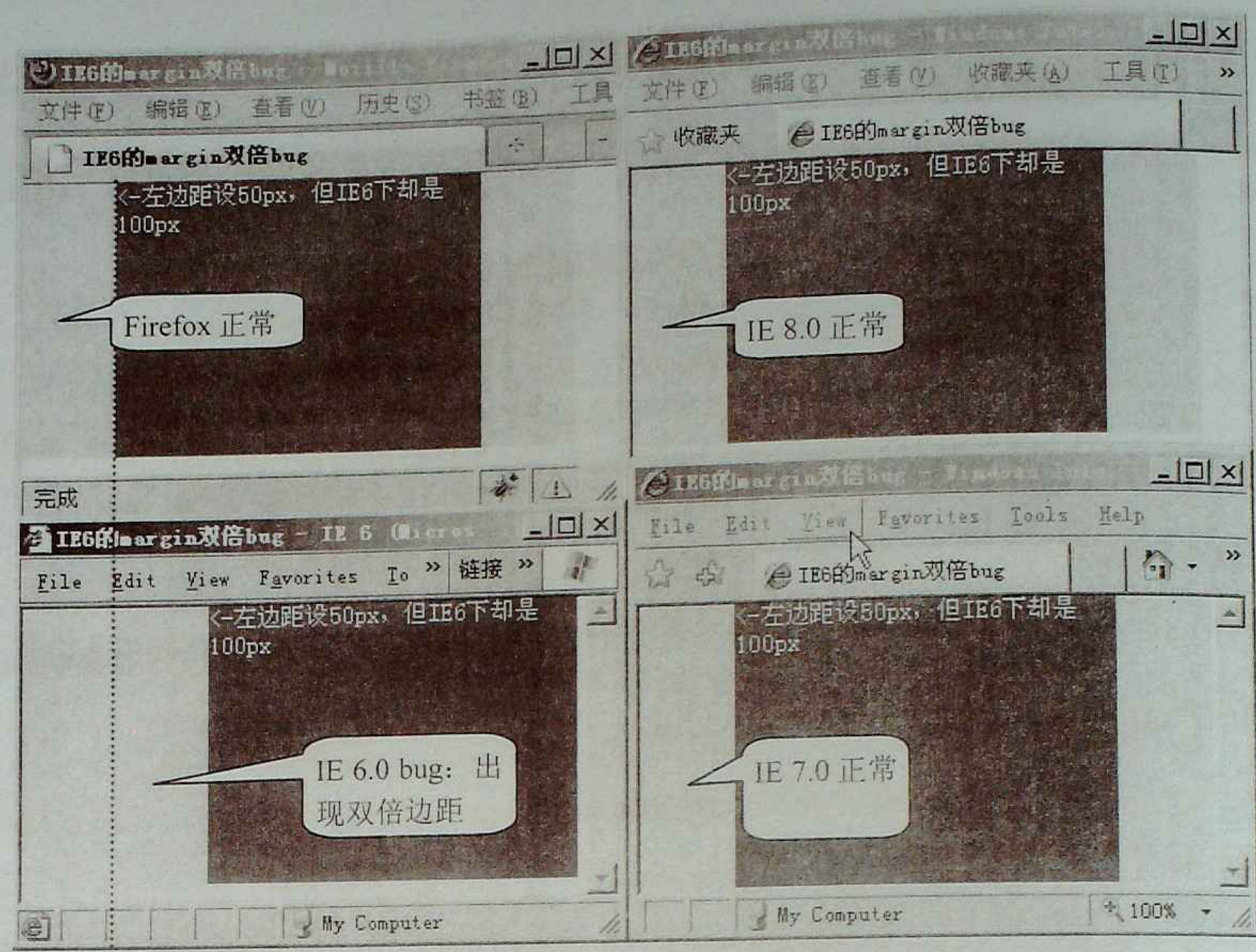


图 11.13 IE 6.0 中的浮动元素左右外边距双倍 bug

➢ 解决方案：设置浮动元素为内联元素（"display:inline;"），具体的修复代码如下。

```
<!-- 省略部分 HTML 代码-->
 <style>
    <!-- 省略部分 CSS 代码-->
    #testdiv{
        _display:inline;  /*修复 IE 6.0 的 bug*/
        color:#fff;
        background:#666;
        float:left;   /*浮动和边距方向一致导致双倍边距问题*/
        margin-left:50px;
        width:200px;
        height:100%;
    }
    span{background:#ccc;}
 </style>
</head>
<body>
  <div id="container">
      <div id="testdiv">&lt;-设置 display:inline;则解决此问题</div>
  </div>
</body>
<!--省略部分 HTML 代码-->
```

修复后的效果如图 11.14 所示。

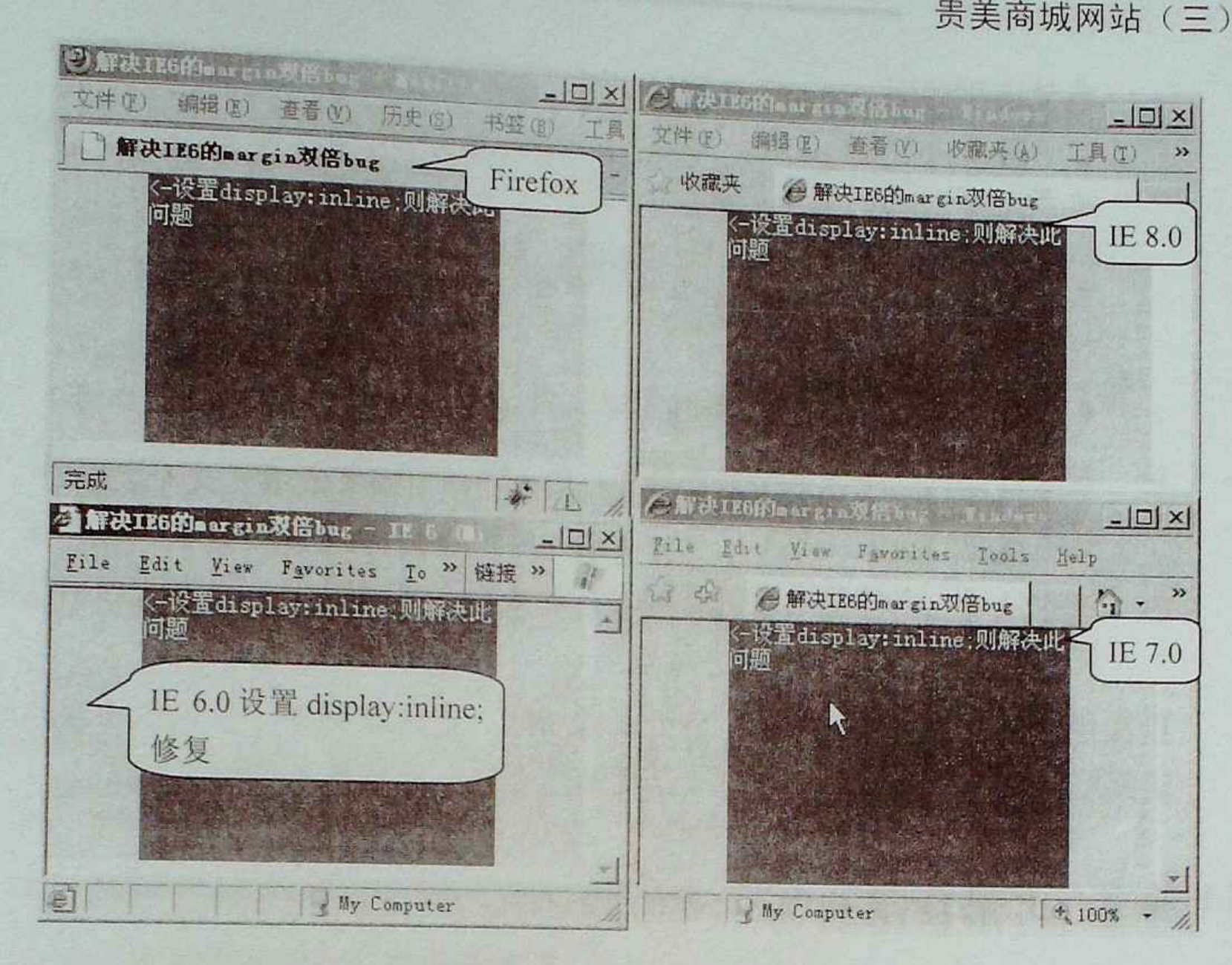


图 11.14　IE 6.0 中的浮动元素左右外边距双倍 bug

➢ 推荐做法：尽量不使用 margin，而用 padding 实现相同效果。

3. 不同浏览器默认的内外边距不一致问题

➢ 出现场景：ol、ul、dl、dd、form 等标签，IE 与 Firefox 的外边距、内边距以及列表 ul，与其他浏览器的默认设置不一致，如图 11.15 所示。

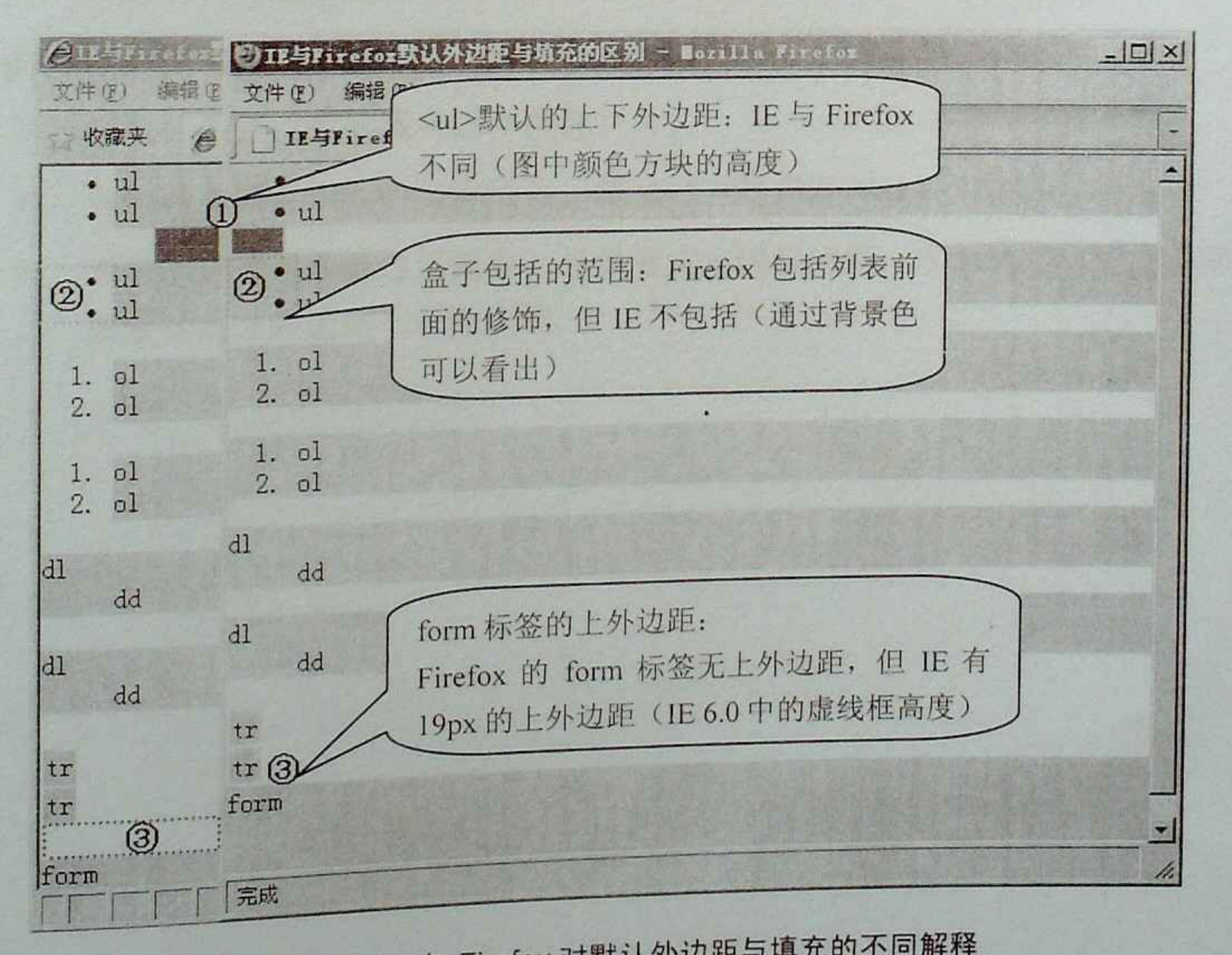


图 11.15　IE 与 Firefox 对默认外边距与填充的不同解释

具体区别如下。

- 在 IE 中，两个<ul>、<ol>、<dl>之间的外边距即 margin 是 19px，而在 Firefox 中是 16px。
- 在 IE 中，列表的盒子范围不包括列表前面的修饰，而在 Firefox 中包括前面的列表修饰。
- 在 IE 中，<form>元素有默认上外边距 19px，而 Firefox 中没有。

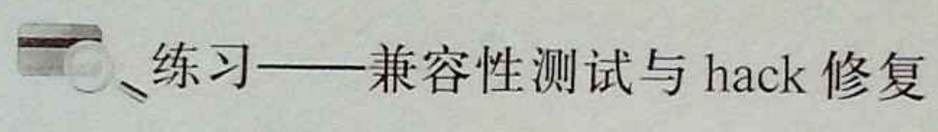
➢ 解决办法：在全局风格里设置 ul、ol、dl、form 的填充及外边距为零，并把 list-style 设置为 none。

➢ 推荐做法：在解决方案的基础上，在多行文本布局中，不用外边距和填充，而纯用 line-height 来实现文本的垂直居中。

现实中还存在很多其他 bug，由于篇幅所限，这里仅列举了常见的兼容性问题。在网站的实际开发中还需反复实践，通过查阅网上资料积累更多的解决办法。

11.4.3 上机练习

上机练习 3



练习——兼容性测试与 hack 修复

需求说明

测试一下自己所写的贵美商城网站的代码，发现其中的兼容性问题并尝试解决。

本章总结

- 制作贵美商城注册页的要点如下。
  - 采用表格实现局部布局。
  - 各表单元素的修饰：文本框统一修饰，然后为非文本框单独修饰。
- 制作贵美商城客户中心页的要点如下。
  - overflow 与 a：hover 伪类组合，实现二级菜单的显示或隐藏。
  - overflow 与<a>标签的锚点组合，实现类似框架窗口的关联效果。
  - 采用“overflow：hidden”隐藏溢出容器尺寸的内容，隐藏后不占页面空间。
- 创建页面间的链接导航的步骤如下。

  (1) 首先理清网站页面含义及彼此的链接关系。

  (2) 测试超链接并修正。
- 测试兼容性的方法有以下几种。
  - 使用 Dreamweaver 验证标签是否符合 W3C 相关规范以及浏览器兼容性问题。
  - 使用 Firefox 验证是否符合 W3C 相关规范。
  - 发送网址到 W3C 官方网站验证是否符合 W3C 相关规范。
- 解决浏览器的兼容性问题有以下两种方法。
  - 根据各浏览器识别的特殊符号，编写对应的 CSS hack。
  - 常见的浏览器兼容性问题有：三像素文本慢移问题、双倍边距问题、不同浏览器默认的内外边距不一致问题。

本章作业

一、选择题

1. 规整的结构化数据显示，应采用（　　）布局方式。

A. table-tr-td

B. div-ul-li

C. dl-dt-dd

D. form-table-tr-td

2. 属性值 auto 的应用场景与含义是（　　）。

A. 相对于实现父容器水平居中，即“margin:0px auto;”，这里的 auto 实现左右外边距对半分，是因为左外边距和右外边距值一样都是 auto

B. 如父容器的高度未设置，设置为“auto”表示自身内容高度将随内容的变化而变化

C. auto 值的意思是自动平均分配宽度或高度

D. auto 不代表任何特殊的含义

3. 关于浏览器兼容性的说法，正确的是（　　）。

A. 如果确定网站的访问者都是 IE 8.0 或允许在用户的计算机上安装 IE 8.0，则不需要考虑和 Firefox 等浏览器的兼容性问题

B. 网站的兼容性只能依靠 CSS hack 方法来补救，没有其他避免的办法

C. 编写 CSS hack 的思路是根据浏览器识别的特殊符号以及 CSS 代码的先后顺序，设置各浏览器相应的 CSS 样式

D. IE 7.0/IE 8.0 下编写的 CSS hack 和有无文档声明无关

4. 在编写 CSS hack 时，关于 Firefox 及 IE 各版本的书写顺序，正确的是（　　）。

A. Firefox 最先写，随后是将“_”添加到 IE 6.0 风格语句的句首，再将“+”添加到 IE 7.0 风格语句的句首，最后将“\0”添加到 IE 8.0 风格语句的属性值与分号之间

B. Firefox 最先写，随后是将“\0”添加到 IE 8.0 风格语句的属性值与分号之间，再将“+”添加到 IE 7.0 风格语句的句首，最后将“_”添加到 IE 6.0 风格语句的句首

C. Firefox 最先写，随后是将“\0”添加到 IE 7.0 风格语句的句首，再将“_”添加到 IE 8.0 风格语句的属性值与分号之间，最后将“+”添加到 IE 6.0 风格语句的句首

D. Firefox 最先写，随后是将“+”添加到 IE 6.0 风格语句的句首，再将“\0”添加到 IE 7.0 风格语句的句首，最后将“_”添加到 IE 8.0 风格语句的属性值与分号之间

5. overflow 与锚点组合实现类似框架窗口的关联效果，关键在于（　　）。

A. 保证超链接锚点位置正确

B. 利用锚链接可定位到 id 属性，保证右侧相同区域的各块大小与父容器一致，并且设置父容器为“overflow:hidden;”

C. 使用 auto 属性值，让右侧内容根据左侧菜单的点击自适应

D. 设置 hover 伪类的特殊效果，显示右侧对应的内容

二、简答题

1. 简述实现类似框架窗口关联效果的实现思路。

2. 简述实现二级菜单显示/隐藏的实现思路。

3. 简述解决浏览器兼容性问题的办法。

4. 利用 overflow 实现中文字符部分截取，如图 11.16 所示。

提示

- 在 div-ul-li 中写入字符串“字符串长度超出，只取 100px 长度的字符”。
- 限定<li>宽高，用 overflow 限定宽度。

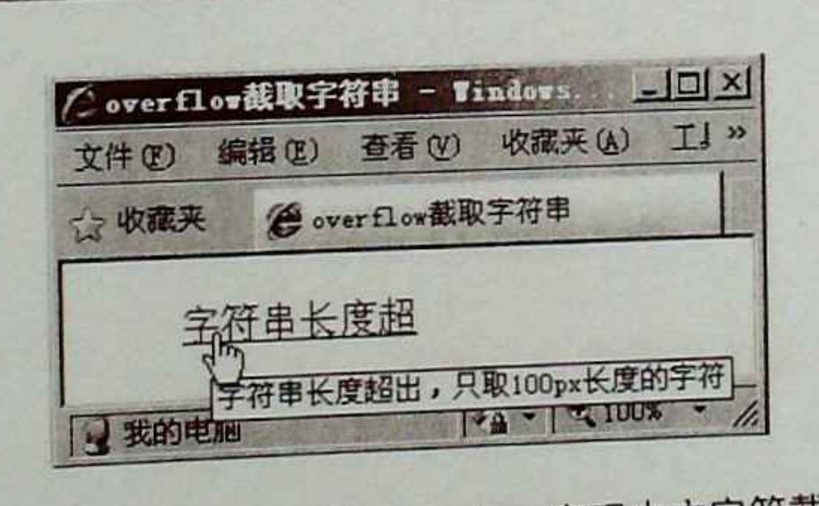


图 11.16　纯 CSS 使用 overflow 实现中文字符截取

5. 总结实现过的页面，创建常用的通用样式时，具体包括如下修饰项。

- 对齐方式，浮动方式，浮动区隔。
- 在网站各页面中出现过的宽高属性。
- 粗体，常用颜色。
- 无边框，宽高自调节。

第 12 章

指导学习：课程总复习

◈本章工作任务

- 实现六人行网站的首页布局

◈本章技能目标

- 使用 DIV+CSS 实现网页的整体布局
- 使用 DIV+CSS 实现网页的局部布局

12.1 复习串讲

12.1.1 核心技能目标

学习后能制作界面美观大方、面向企业应用的静态商业网站，不含动态脚本功能，具体的技能目标如下。

- 掌握符合 W3C XHTML 1.0 规范的 HTML 标签，要求能熟练手写 HTML 代码。
- 能够使用 DIV+CSS 实现页面的整体布局，能正确划分页面结构并编写 HTML 和 CSS 代码。
- 能够使用 DIV+CSS 实现页面的局部布局，熟悉四种典型的局部结构，并灵活应用。
- 能够使用常用的 CSS 属性，实现常见的网页样式，具体包括以下内容。
  - 细边框样式、无下划线链接、文字或图文对齐等。
  - 导航菜单、产品分类目录、图文混编效果等。
  - 背景设置、小图标截取等。
- 能读懂 CSS 代码，并能在后期开发中根据需要修改。

我们可以自我检查，看看是否达到了这些技能目标，如果还有疏漏和不足，就赶快查漏补缺吧。

12.1.2 知识梳理

HTML+CSS 知识体系由两部分组成，分别为 HTML（XHTML）和 CSS，如图 12.1 所示。

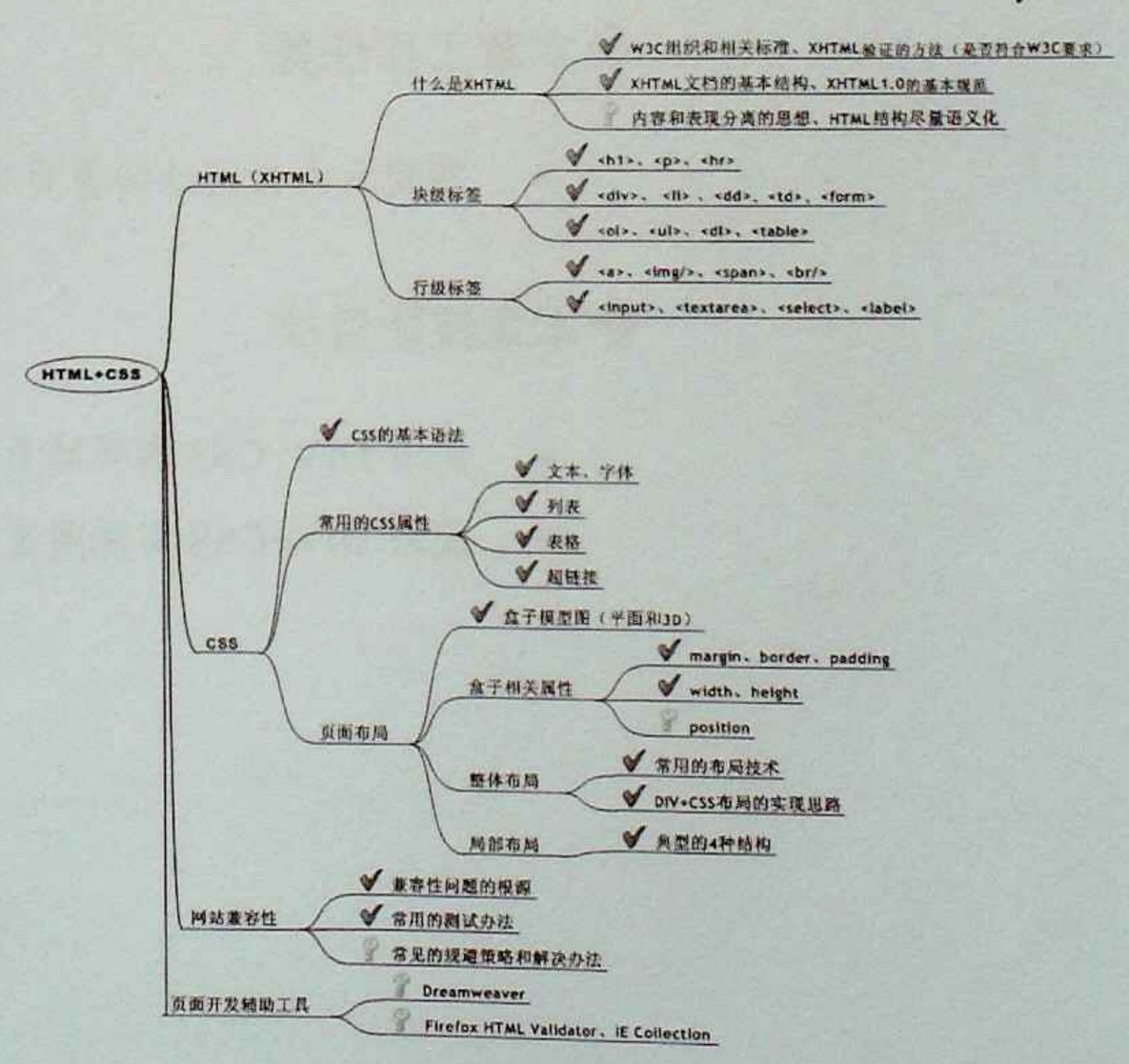


图 12.1 HTML+CSS 知识体系图

在知识体系图中，✔图标表示学过该知识；?图标表示部分学过该知识或者学习过相关知识，但需要巩固和补齐。

12.2 综合练习

12.2.1 任务描述

本次综合练习的任务是，根据美工提供的页面效果图（图 12.2），分析并制作六人行网站的首页布局，为后续第 13 章的项目案例做好准备。

图 12.2 首页效果图

12.2.2 练习

阶段 1：练习——实现页面的整体布局

需求说明

- 根据图 12.2 所示的效果图，划分页面结构，采用 DIV+CSS 布局技术实现页面的整体布局。

- 要求内容和表现分离，HTML 页面结构要尽量符合结构或内容语义。

提示

- 根据页面想表达的内容来划分区块，功能和目的相同的内容归为一个区块。
- 因使用 border 容易带来默认的边框尺寸，所以推荐各块使用背景色和文字进行标识。
- 注意分析各块是否有内边距（padding）。练习素材中提供的电子版效果图，可以使用 Hypersnape 工具获取各块的大致尺寸，不要求精确。

阶段 2：练习——实现各块的局部结构

需求说明

- 实现各块的局部结构，要求使用背景色和文字标识各块。
- 在符合结构或内容语义的基础上，要求局部结构尽量简洁。

提示

- “雪人”图片下方有两个明显的标题，需要再分为左右两块。
- 局部组织结构可以考虑使用学过的四种典型结构，但应注意灵活使用。
- 注意分析各块是否有内边距（padding）、外边距（margin）。
- 注意代码的规范，CSS 代码尽量体现层次结构，选择器命名加上区域标识，方便阅读，例如#header form 表示 header 区域的表单样式。

第 13 章

项目案例：六人行网站

◈本章工作任务

- 制作六人行网站的首页
- 制作六人行网站的活动页
- 制作六人行网站的据点页
- 制作六人行网站的注册页

◈本章技能目标

- 使用 DIV+CSS 实现页面的整体布局
- 使用 DIV+CSS 实现页面的局部布局
- 使用 CSS 美化页面

本章单词

请在预习时学会下列单词的含义和发音，并填写在横线处。

1. spot：_____
2. activity：_____
3. banner：_____
4. flag：_____
5. neighbor：_____
6. focus：_____
7. round：_____
8. current：_____

本章简介

通过前面 12 章的学习，我们掌握了 HTML+CSS 的基本语法，之后深入学习了 DIV+CSS 布局及各种常用技巧，随后制作了贵美商城网站，并尝试解决页面布局过程中遇到的各类问题。从本章开始我们将利用 12 个学时完成一个典型的交友网站——六人行网站。

13.1 案例分析

13.1.1 需求概述

六人行网站（http://www.ifindu.com）是目前人气较旺的网络交友平台，查看 HTML 源码中<meta>标签的“description”属性，可以发现该网站的功能描述为：“北京最大的青年白领写字楼社区网站。网络圈地比赛，创建有意思的据点，发掘身边的打折优惠；和同写字楼的邻居进行各种有趣的交互；通过参与夜店派对、户外旅行、喜剧演出、聚餐聚会等丰富多彩的线下活动结识兴趣相投的朋友”。根据前面学过的 DIV+CSS 相关技能，实战模拟制作如下四个典型网页。

- 首页：网站的窗户，展示网站的主要功能和服务，同时提供热门活动、据点、招贴等导航和登录功能，如图 13.1 所示。

图 13.1 首页

➢ 活动页：提供各类活动列表、各类活动的图片展示宣传，吸引用户参与或发布活动，如图 13.2 所示。

图 13.2 活动页

- 据点页：提供各类好玩、好吃的据点（地方）列表，最新的据点宣传，吸引用户参与或创建新据点，如图 13.3 所示。

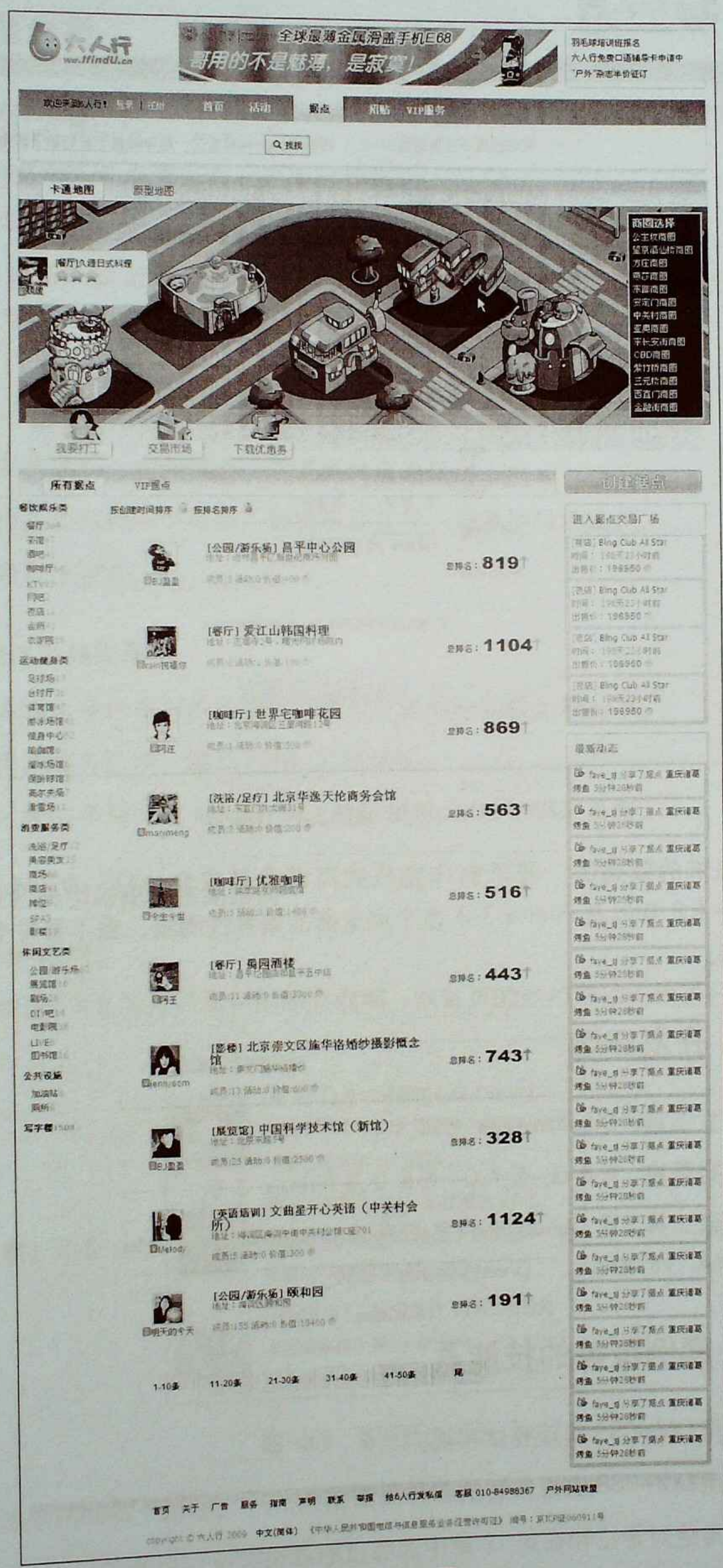


图 13.3　据点页

- 注册页：提供注册功能，方便用户享受会员服务及 VIP 服务，如图 13.4 所示。

图 13.4　注册页

参考提供的效果图，要求制作完成的页面尽量符合 W3C XHTML 1.0 标准，并经过 IE 6.0～IE 8.0 以及 Firefox 3.5 多个浏览器的兼容性测试，但不要求全部解决存在的兼容性问题。

13.1.2　开发环境

- 开发工具：Dreamweaver CS4。
- 运行环境：IE 6.0、IE 7.0、IE 8.0 及 Firefox 3.5。
- 测试工具：IE Collection（多 IE 版本共存软件）、Firefox 3.5（需安装插件 Firebug 和 HTML Validator）、Dreamweaver CS4。

13.1.3　案例覆盖的技能点

- 使用 DIV+CSS 实现整体布局。
- 使用 DIV+CSS 实现典型的四种局部结构。
- 使用绝对定位和在第 12 章中训练过的滤镜实现 CSS 特效。

➢ 使用 CSS 实现页面的美化。

13.1.4 问题分析

1. 网站的制作步骤

(1) 使用 Dreamweaver 创建站点、公用文件夹。

(2) 制作首页。

(3) 创建网站所有页面共用的全局样式 global.css 并绑定页面，例如字体、外边距(margin) 设为 0，制作过程中可以随时增加。

(4) 利用第 11 章作业总结归纳的细节风格 detail.css 并用<link />引用，风格具体包括长宽、浮动与对齐、常用边框等，制作过程中可随时修改或增删。

(5) 创建页面的顶部、底部模板，以便其他页面使用。

(6) 应用模板实现其他三个页面，包括布局实现、涉及的样式实现。

(7) 页面的兼容性测试。

2. 页面的整体布局实现

四个页面基本均为多行多列的布局，页面整体布局的实现步骤如下。

(1) 根据效果图划分区块，编写 HTML 页面结构。

(2) 创建各区块的布局样式 layout.css，然后使用<link/>绑定页面。

3. 网站涉及的典型的局部布局结构

1) div-dl-dt-dd 结构

例如，对于如图 13.5 所示页面中的图文混编，以首页的热门活动为例进行介绍。

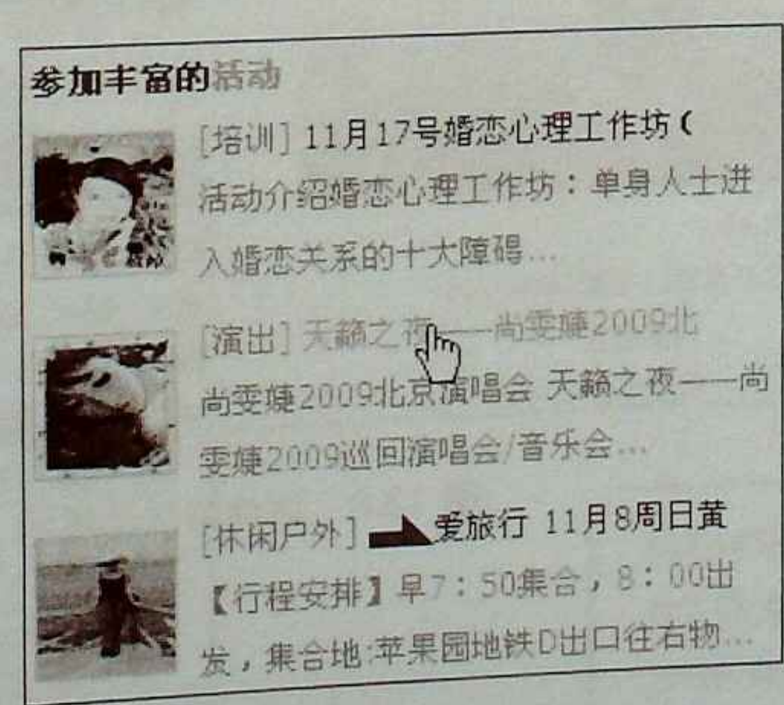


图 13.5 首页热门活动

推荐采用的组织结构如下所示。

```
<div>
  <h1>参加丰富的<span>活动</span></h1>
```

```
    <dl>
      <dt><img src="images/act1.gif" alt="活动 1" /></dt>
      <dd>[培训] <a href="#">11 月 17 号婚恋心理工作坊(</a></dd>
      <dd><p>活动介绍婚恋心理工作坊:单身人士进入婚恋关系的十大障碍...</p></dd>
    </dl>
    ……<!-- 重复上面的 dl-dt-dd 结构,略 -->
    ……<!-- 热门据点块,重复上面的 h1 与 dl-dt-dd 的结构,略 -->
  </div>
```

此区域的布局主要在于浮动与填充属性的使用。

2）div-ul-li 结构

活动页的活动展示如图 13.6 所示。因都市生活为一个可重复的相同结构，包含：

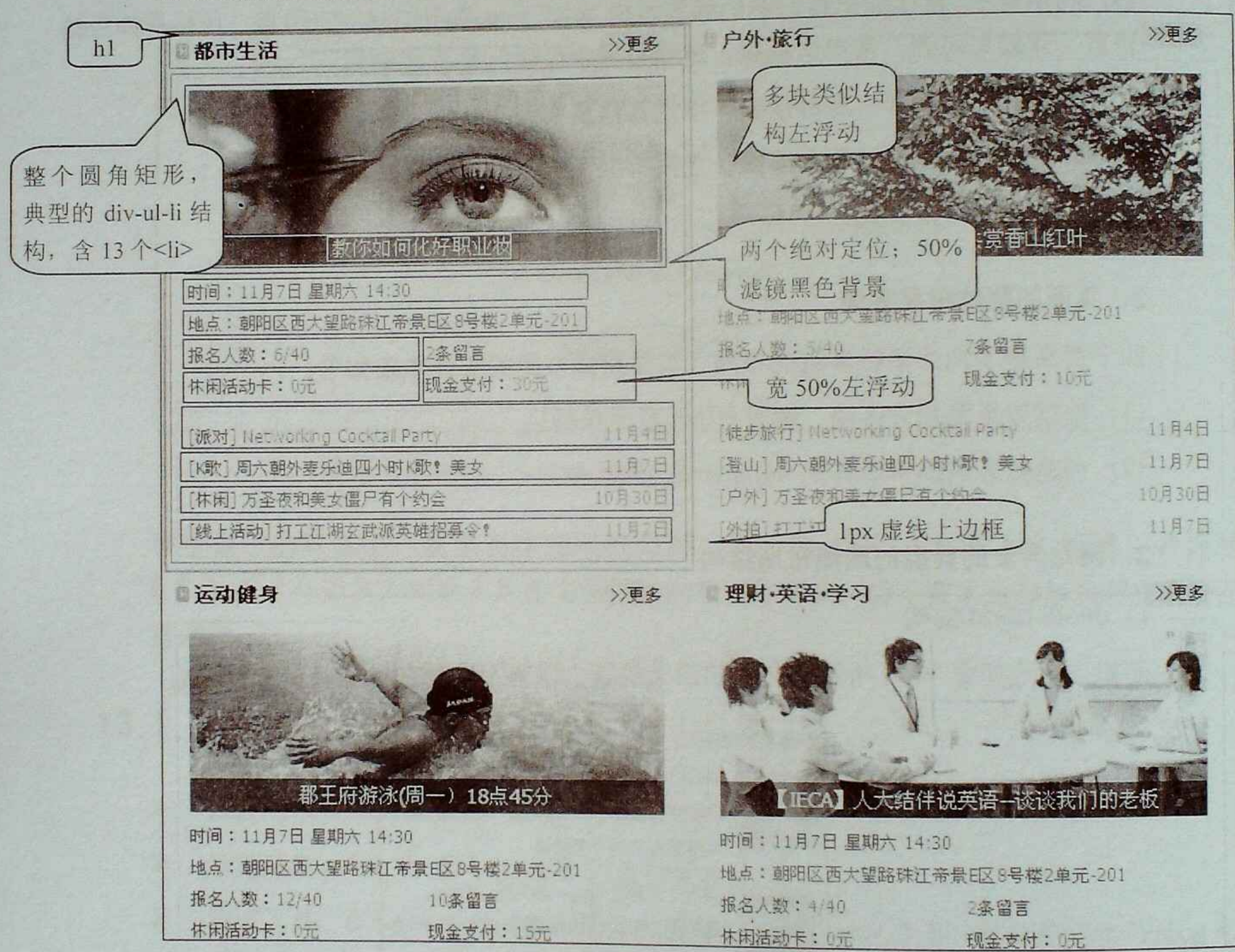


图 13.6　活动页活动展示

外部结构：

```
<div>
    <h1>都市生活</h1>
    <div><!-- 圆角矩形 div --></div>
</div>
```

圆角矩形结构，参考后续的“背景修饰技巧”中的自适应圆角矩形：

```
<div class="round">
    <ul class="list"><!-- 主内容,含 13 个 li -->
      <li><img src="" alt="活动大图" /></li>
      <li><!-- 50%滤镜,绝对定位叠加在大图上的黑色背景--></li>
      <li><!-- 绝对定位在黑色背景上的白色文字--></li>
      <!-- 6 个 li,虚线上方的时间、地点、报名人数、留言、活动卡与现金支付 -->
      <!-- 4 个 li,虚线下方的 4 条信息,第一个 li 有 1px 虚线上边框、填充和外边距-->
    </ul>
    <ul class="end"><p> </p></ul><!-- 圆角矩形下方 -->
</div>
```

再如首页顶部区域，如图 13.7 所示。

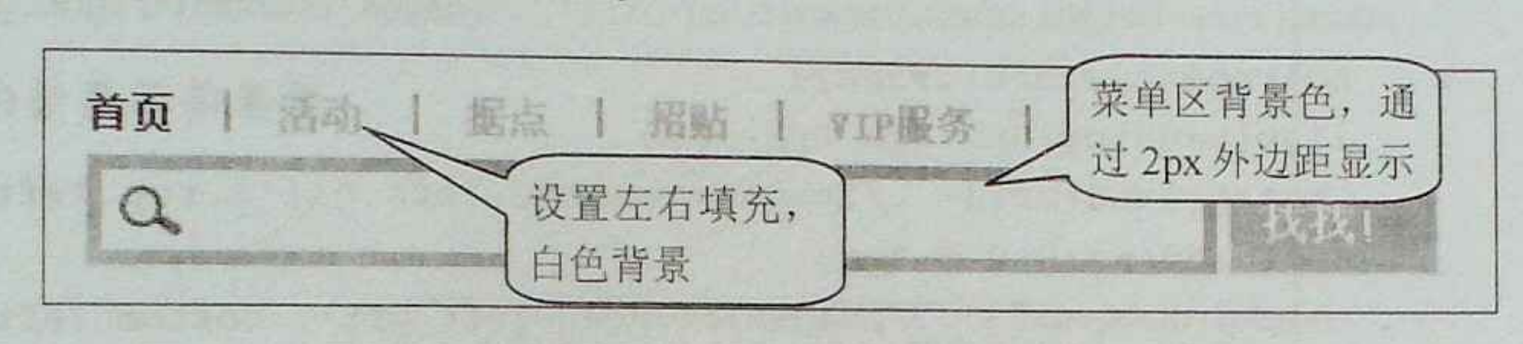


图 13.7　首页顶部区域

- 此区域的菜单：需设置左浮动，左右填充与 2px 右外边距，自身为白色背景，然后把菜单区背景色设置为灰色，则会显示在 2px 右外边距，实现竖线效果。注意行高为字体高度。
- 搜索文本框：需设置背景图、5 像素的边框，并设置左内边距。

3）table-tr-td 结构

例如活动页的活动日期，如图 13.8 所示。

4. 背景修饰技巧

1）自适应高度的圆角矩形

例如，活动页顶部的圆角矩形如图 13.9 所示。

图 13.8　活动页的活动日期

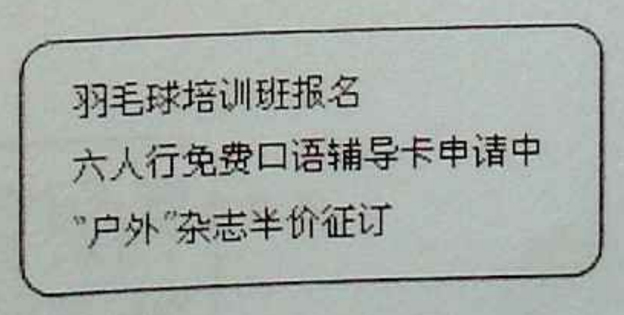


图 13.9　活动页顶部的圆角矩形

圆角矩形的应用范围非常广，也很常见。这里的实现采用通用方法，即利用背景图的偏移，用两张图分别显示圆角矩形的左半与右半边，这种方法能自动适应内容的大小。

➢ 组织结构的代码如下。

```
<div class="round">
    <ul class="list">
        <li><a href="#">羽毛球培训班报名</a></li>
        <li><a href="#">六人行免费口语辅导卡申请中</a></li>
        <li><a href="#">"户外"杂志半价征订</a></li>
    </ul>
    <ul class="end"><p> </p></ul>
</div>
```

➢ CSS 代码如下。

```
.round{line-height:21px;background:url('../images/rounded-right.gif') top right
  no-repeat;width:195px;}
  .list{padding:9px 9px 0px 9px;
   background:url('../images/rounded-left.gif') top left no-repeat;}
  .end{padding:0 0 0 9px;
   background:url('../images/rounded-left.gif') bottom left no-repeat;}
   .end p{line-height:9px;height:9px;margin-top:-9px;padding:0 9px 9px 0;
      background:url('../images/rounded-right.gif') bottom right no-repeat;}
```

背景偏移属性值"右上"

➢ 整体思路如下如下。

四层容器，外边距都为零。最外层容器显示右上角。

- 类 list 容器为内容容器，显示左上角。
- 类 end 为底部容器，显示左下角，里面的<p>容器显示右下角。填充数为圆角的大小 9px。这种圆角矩形的布局方式适用于风格相同但大小不一样的情境，复用性很强。

我们参照 CSS 代码与图 13.10、图 13.11 解释这种思路。

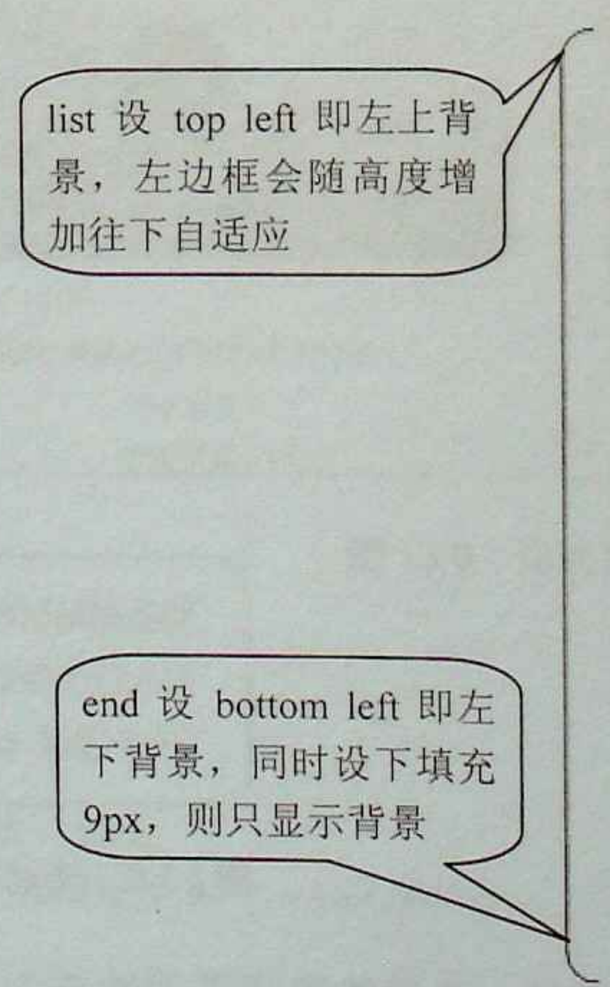


图 13.10 圆角矩形素材左半边 rounded-left.gif

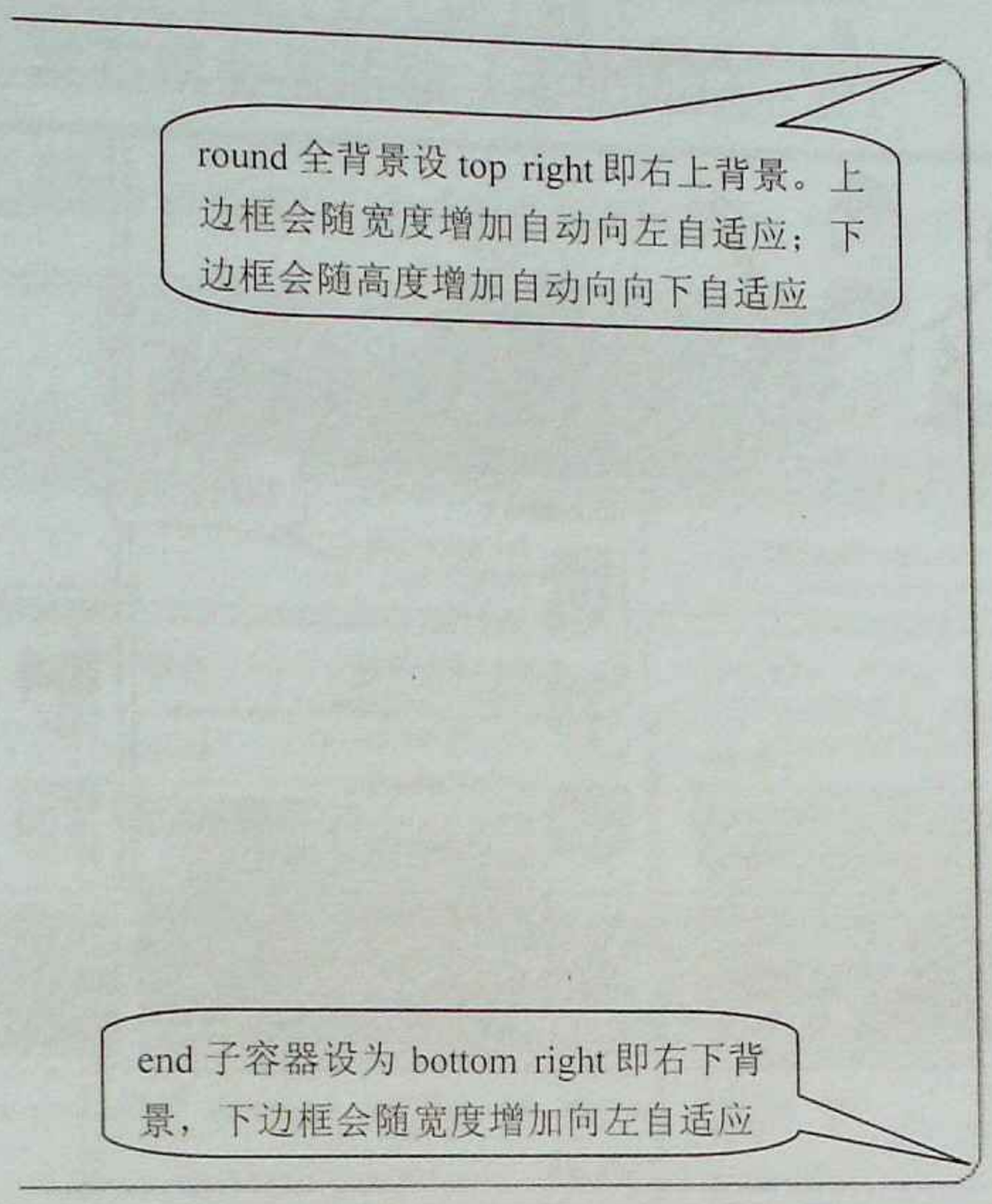


图 13.11　圆角矩形素材右半边 rounded-right.gif

2）背景偏移量技术

如图 13.12 是这个项目中用到的几个背景图标。在使用时，设置对应的偏移数值即可（可使用 Hypersnape 等工具测量），实现方法参考第 5 章。

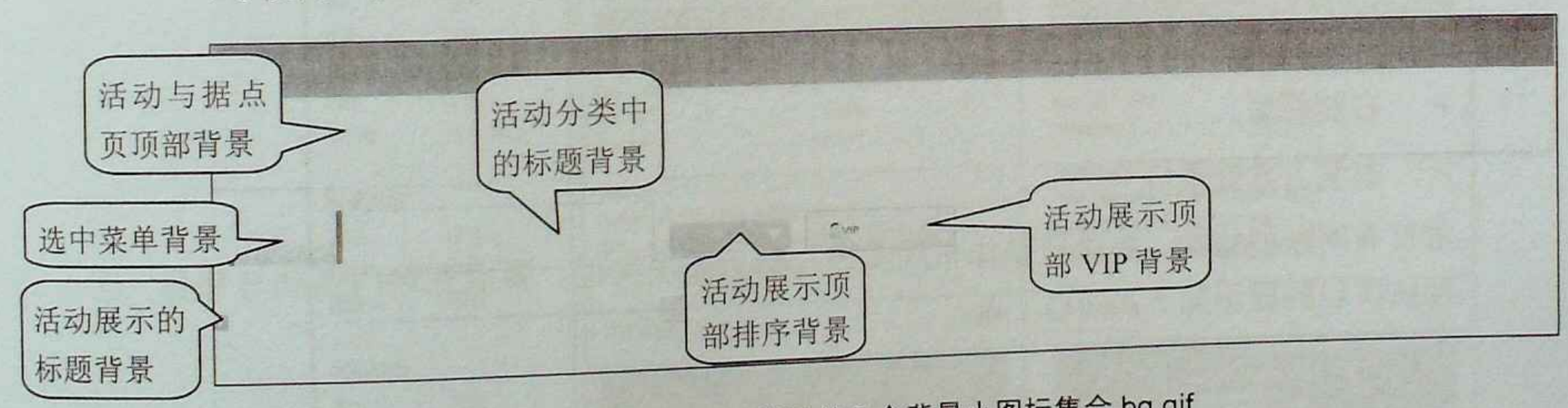


图 13.12　常用的几个背景小图标集合 bg.gif

13.2　项目需求

六人行的项目主要包含四个页面，现采用用例方式，具体描述如下。

13.2.1　用例 1：登录网站（首页）

首页的结构相对简单，结构非常清晰，如图 13.13 所示。

图 13.13　六人行首页结构图

主要的实现功能块包括以下几项。

- 顶部 Logo、菜单及搜索条。
- 底部版权。
- 左侧热门活动与热门据点。
- 右侧邻居。
- 右侧上方登录口

首页有明晰的结构，先进行整体布局实现，然后逐块实现。要点在热门活动与邻居，这两块实现的提示如下。

提示

- 热门活动使用 dl-dt-dd 来实现，组织结构如下：

```
<div>
  <h1>参加丰富的<span>活动</span></h1>
  <dl>
      <dt><img src="images/act1.gif" alt="活动 1" /></dt>
      <dd>[培训] <a href="#">11 月 17 号婚恋心理工作坊(</a></dd>
      <dd><p>活动介绍婚恋心理工作坊：单身人士进入婚恋关系的十大障
      碍...</p></dd>
  </dl>
</div>
```

- "邻居"版块的结构也类似。

13.2.2 用例 2：发布或参与活动（活动页）

活动页较为复杂，利用第 8 章指导学习完成的页面进行完善，结构分析如图 13.14 所示。

图 13.14　六人行活动页

- 顶部放置 Logo、横幅广告和右侧热门信息。

  顶部最右侧热门信息有圆角矩形效果，请用提供的素材实现。
- 顶部下方的导航条包括欢迎词、注册登录链接、主菜单与搜索功能。
  - 菜单的效果，要求当前栏目出现白色渐变背景图效果，如活动页，则“活动”菜单变为白色渐变背景。
  - 搜索功能，相对美观。
- 活动分类。要求各分类的高度随内容自适应，只需在 ul 父容器中不设高度，其高度即会随着子容器内容的高度自动调节。
- 活动日期用表格或 ul-li 实现，设置<a>标签，并实现相应的效果。
- 活动展示，具体内容如下。
  - 实现标题区，为“按时间检索”、“VIP 活动”和“发布新活动”加入超链接。
  - 图片为最热门活动的效果图，图片上方的标题带滤镜效果。
  - 内容分上下两部分，中间用虚线分隔。上部包括最热门活动的时间、地点、报名人数、留言数及支付信息。下部包括其他热门活动的类别、名字及时间。
- 底部版权直接复用首页底部。

提示

热门活动展示：

- 使用圆角矩形实现，其基本组织结构如下。

```
<div class="content_list">
  <h1><div>英语&bull;学习</div><a href="#" class="f_r">&gt;&gt;更多
  </a></h1>
  <div class="round">
  <ul class="list">
       <li></li>
       ……<!-- 12 个<li> 略-->
  </ul>
  <ul class="end"><p> </p></ul>
  </div>
</div>
```

- 图片上方的效果，通过设置相对定位与绝对定位实现。设两个绝对定位层，一个层次为背景色透明度 50%。具体 CSS 代码为“background:#000;filter:alpha(opacity=50); opacity:0.5;”。一个层为白色文字，文字层叠在背景图上方，且两个层宽高及位置一致。
- 报名、留言、支付等一行两段文字的实现，可以通过设置前一个<li>占 50%并左浮动实现。
- 虚线下方的同行左右对齐，通过为两段文字分别设置左浮动和右浮动实现。

13.2.3 用例 3：参与或发布据点（据点页）

完成活动页后，据点页就可以复用顶部与菜单导航，实现相对简单，如图 13.15 所示。

图 13.15　六人行据点页

- 顶部与菜单导航与活动页类似，复用即可。
- 地图用大图片展示效果即可。
- 下方左侧据点分类列表，结构很简单，根据语义，推荐采用 ul-li 实现，数字用定义特殊风格。包括据点一、二级分类的详细列表及二级分类中的项目总数，设置超链接。
- 下方中部热门据点信息是关键块。
  - 每条信息包括左边的创建据点的用户头像及左下方的性别和昵称。
  - 信息的中间是据点的分类、名称、地址、成员、活动及价值信息。
  - 信息的右侧是总排名。
  - 信息列表最下方是页码导航。
- 右侧包含一个据点创建按钮、据点交易广场及据点动态。
  - 据点交易广场每条信息包含据点类、名称、交易发布时间及其出售价格。
  - 据点动态，实时显示据点动态信息，如发布、共享、话题、优惠券等。

提示

据点展示列表的结构实现，可以先为外层容器设置四个方向填充，然后分人物形象图与名字、据点信息及其排行三块，左浮动实现，如图 13.16 所示。然后设置下边框，中间区域直接按 Enter 键换行即可。

图 13.16　六人行据点页据点展示

13.2.4　用例 4：注册会员（注册页）

完成前三个页面后，注册页则较为简单，基本结构如图 13.17 所示。

- 在顶部展示 Logo 及下方右侧提供“返回首页”链接。
- 中间区域为关键区域。
  - 标语实现，由于图文混排，设置行高等于文字行高，通过上下外边距实现垂直居中。
  - 声明信息，浅黄背景色，橙色 1px 边框，提示信息为绿色，所有出现的“*”号和关键提示用橙色。
  - 含手机号、电子邮件、昵称、性别、所在写字楼的必填项，及邀请人手机号可选项。

- 协议链接及提交按钮。

提示

表单排版如用表格则很简单，可以尝试采用 div-ul-li 的布局结构，提示如下。

- 每行用一个<ul>，并设置高度与行高。
- 中的<input />元素的容器高度与自身高度一致，<label>也是，以实现垂直居中。

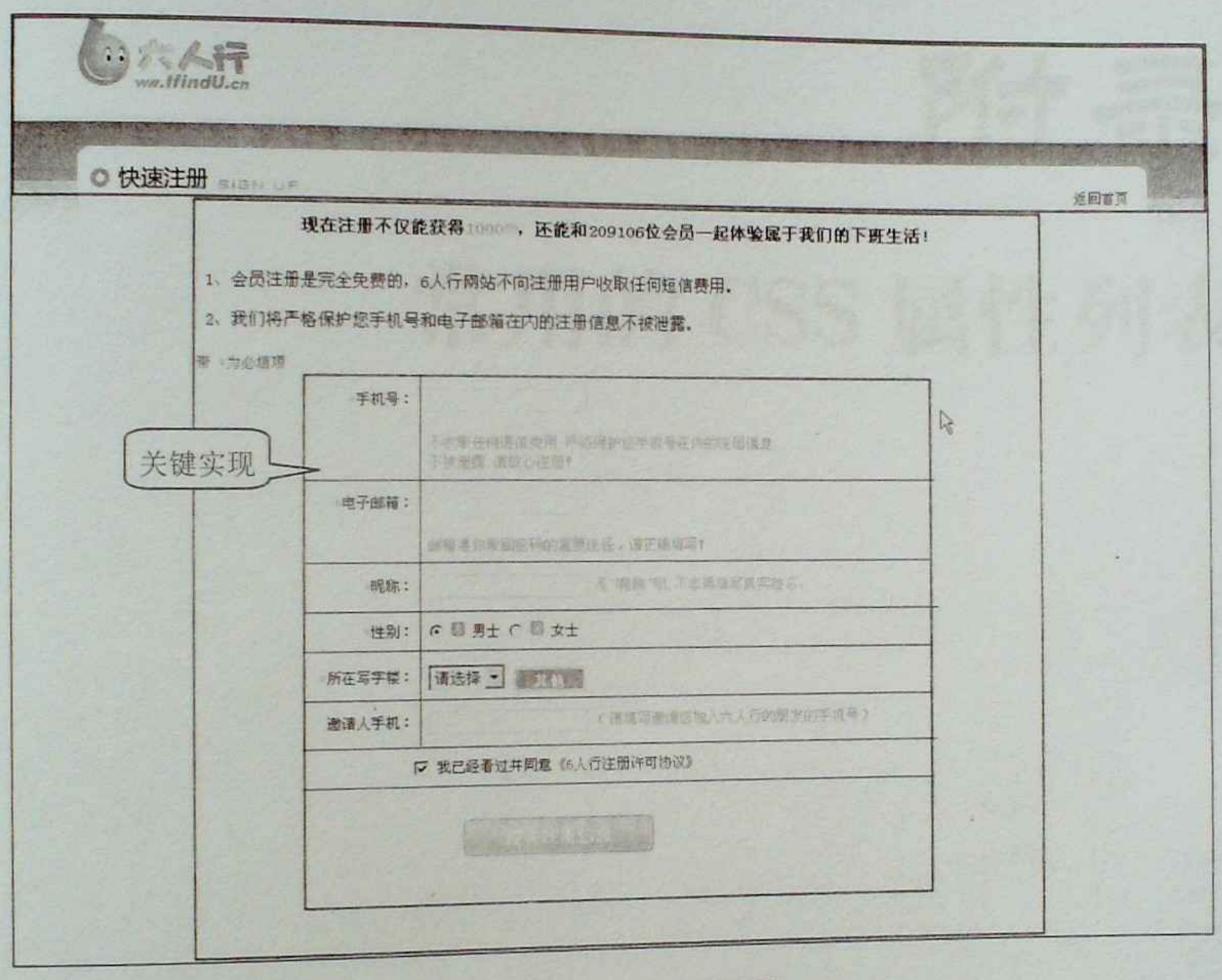


图 13.17　六人行注册页

13.3　进度记录

请填写表 13-1。

表 13-1　开发进度记录表

用例	开发完成时间	测试通过时间	备注
用例 1：登录网站（首页）	3 小时		
用例 2：发布或参与活动（活动页）	1 小时（在第 8 章：指导学习基础上完善）		
用例 3：参与或发布据点（据点页）	4 小时		
用例 4：注册会员（注册页）	3 小时		

附录

常用的 CSS 属性列表

摘选自：http://www.w3school.com.cn/

表格最右列的数字标识支持的 CSS 最低版本。

1．CSS 背景属性（Background）

属性	描述	CSS
background	在一个声明中设置所有的背景属性	1
background-attachment	设置背景图像是否固定或者随着页面的其余部分滚动	1
background-color	设置元素的背景颜色	1
background-image	设置元素的背景图像	1
background-position	设置背景图像的开始位置	1
background-repeat	设置是否及如何重复背景图像	1

2．CSS 边框属性（Border 和 Outline）

属性	描述	CSS
border	在一个声明中设置所有的边框属性	1
border-bottom	在一个声明中设置所有的下边框属性	1
border-bottom-color	设置下边框的颜色	2
border-bottom-style	设置下边框的样式	2
border-bottom-width	设置下边框的宽度	1
border-color	设置四条边框的颜色	1
border-left	在一个声明中设置所有的左边框属性	1
border-left-color	设置左边框的颜色	2
border-left-style	设置左边框的样式	2
border-left-width	设置左边框的宽度	1
border-right	在一个声明中设置所有的右边框属性	1
border-right-color	设置右边框的颜色	2
border-right-style	设置右边框的样式	2
border-right-width	设置右边框的宽度	1
border-style	设置四条边框的样式	1
border-top	在一个声明中设置所有的上边框属性	1
border-top-color	设置上边框的颜色	2
border-top-style	设置上边框的样式	2
border-top-width	设置上边框的宽度	1
border-width	设置四条边框的宽度	1
outline	在一个声明中设置所有的轮廓属性	2
outline-color	设置轮廓的颜色	2
outline-style	设置轮廓的样式	2
outline-width	设置轮廓的宽度	2

3. CSS 文本属性（Text）

属性	描述	CSS
color	设置文本的颜色	1
direction	规定文本的方向/书写方向	2
letter-spacing	设置字符间距	1
line-height	设置行高	1
text-align	规定文本的水平对齐方式	1
text-decoration	规定添加到文本的装饰效果	1
text-indent	规定文本块首行的缩进	1
text-shadow	规定添加到文本的阴影效果	2
text-transform	控制文本的大小写	1
unicode-bidi	设置文本方向	2
white-space	规定如何处理元素中的空白	1
word-spacing	设置单词间距	1

4. CSS 字体属性（Font）

属性	描述	CSS
font	在一个声明中设置所有字体属性	1
font-family	规定文本的字体系列	1
font-size	规定文本的字体尺寸	1
font-size-adjust	为元素规定 aspect 值	2
font-stretch	收缩或拉伸当前的字体系列	2
font-style	规定文本的字体样式	1
font-variant	规定文本的字体样式	1
font-weight	规定字体的粗细	1

5. CSS 外边距属性（Margin）

属性	描述	CSS
margin	在一个声明中设置所有的外边距属性	1
margin-bottom	设置元素的下外边距	1
margin-left	设置元素的左外边距	1
margin-right	设置元素的右外边距	1
margin-top	设置元素的上外边距	1

6. CSS 内边距属性（Padding）

属性	描述	CSS
padding	在一个声明中设置所有的内边距属性	1
padding-bottom	设置元素的下内边距	1
padding-left	设置元素的左内边距	1
padding-right	设置元素的右内边距	1
padding-top	设置元素的上内边距	1

7. CSS 列表属性（List）

属性	描述	CSS
list-style	在一个声明中设置所有的列表属性	1
list-style-image	将图像设置为列表项标记	1
list-style-position	设置列表项标记的放置位置	1
list-style-type	设置列表项标记的类型	1

8. CSS 尺寸属性（Dimension）

属性	描述	CSS
height	设置元素高度	1
max-height	设置元素的最大高度	2
max-width	设置元素的最大宽度	2
min-height	设置元素的最小高度	2
min-width	设置元素的最小宽度	2
width	设置元素的宽度	1

9. CSS 定位属性（Positioning）

属性	描述	CSS
bottom	设置定位元素下外边距边界与其包含块下边界之间的偏移	2
clear	规定元素的哪一侧不允许其他浮动元素	1
clip	剪裁绝对定位元素	2
cursor	规定要显示的光标类型（形状）	2
display	规定元素应该生成的框的类型	1
float	规定框是否应该浮动	1
left	设置定位元素左外边距边界与其包含块左边界之间的偏移	2
overflow	规定当内容溢出元素框时发生的事情	2
position	规定元素的定位类型	2
right	设置定位元素右外边距边界与其包含块右边界之间的偏移	2
top	设置定位元素的上外边距边界与其包含块上边界之间的偏移	2
vertical-align	设置元素的垂直对齐方式	1
visibility	规定元素是否可见	2
z-index	设置元素的堆叠顺序	2

10. CSS 表格属性（Table）

属性	描述	CSS
border-collapse	规定是否合并表格边框	2
border-spacing	规定相邻单元格边框之间的距离	2
caption-side	规定表格标题的位置	2
empty-cells	规定是否显示表格中的空单元格上的边框和背景	2
table-layout	设置用于表格的布局算法	2

第一学年[第一学期]

ACCP软件开发初级程序员

使用Java理解程序逻辑

使用HTML语言和CSS开发商业站点

C#语言和数据库技术基础

使用C#语言开发数据库应用系统

职业修养和基础项目训练

北大青鸟 APTECH ® WE CHANGE LIVES

北京阿博泰克北大青鸟信息技术有限公司

BEIJING APTECH BEIDA JADE BIRD INFORMATION TECHNOLOGY CO.,LTD

http://www.jb-aptech.com.cn

ISBN978-7-5023-5837-2

9 787502 358372 >

定价:500.00(共5册)