

SAMS  
**Teach  
Yourself**

- 全球销量逾百万册的系列图书
- 连续十余年打造的经典品牌
- 直观、循序渐进的学习教程
- 掌握关键知识的最佳起点
- 秉承Read Less, Do More (精读多练)的教学理念
- 以示例引导读者完成最常见的任务

每章内容针对初学者精心设计, **1**小时轻松阅读学习,  
**24**小时彻底掌握关键知识

涵盖  
HTML5  
知识

每章**案例与练习题**助你轻松完成常见任务,  
通过**实践**提高应用技能, 巩固所学知识

# HTML与CSS

**入门经典** (第8版)

[美] Julie C. Meloni Michael Morrison 著  
姚军 译

人民邮电出版社  
POSTS & TELECOM PRESS

# HTML与CSS 入门经典 (第8版)

[美] Julie C. Meloni Michael Morrison 著  
姚军 译



人民邮电出版社  
北京

## 内容提要

本书是畅销图书《HTML 与 CSS 入门经典》的最新版，和过去的版本相同，本书仍然采用直观、循序渐进的方法来为读者讲解使用 HTML 与 CSS 创建并发布网页的过程，以方便读者掌握。本书总共分为 24 章和 1 个附录，其涵盖的内容有理解 Web 的工作方式，如何发布 Web 内容，理解 HTML 与 XHTML 的关系，理解层叠样式单 (CSS)，使用文本块和列表，使用字体，使用表格显示信息，使用外部和内部链接，使用颜色，创建用于 Web 上的对象，在网站中使用对象和多媒体，使用框架，使用边距、填充、对齐和浮动，理解 CSS 盒子模型和定位，使用 CSS 进一步控制列表和控制导航，使用鼠标操作修改文本显示，创建固定或者流动的布局，创建打印友好的网页，理解动态网站，使用基于 Web 的表单，组织和管理网站，以及帮助人们找到你的网站等内容。附录还提供了完整的 XHTML1.1 与 CSS2 快速参考。

本书语言简洁、内容丰富，适合 Web 设计开发人员，以及大中专院校相关专业的学生阅读，对 Web 开发和发布感兴趣的读者也可以从中获益。



## 关于作者

---

**Julie C. Meloni** 是位于 Los Altos 的多媒体公司 i2i Interactive 的技术总监，同时还是一位在数字人文领域工作的学者。她曾经编著了许多关于基于 Web 的编程语言及数据库主题的图书和文章，其中包括畅销的 Sams 《Teach yourself PHP, MySQL, and Apache All in one》。

**Michael Morrison** 是一位作家、开发人员、玩具发明家，同时还是各种计算机技术图书以及基于 Web 的交互课程的作者。除了作家和自由职业者这个主业以外，Michael 还与他的妻子 Masheed 一起创立了 Stalefish Labs 娱乐公司。





# 前 言

2009 年，全世界估计有 15 亿人访问互联网，在美国就有 2.2 亿。加上 3.38 亿的中国用户，5500 万德国用户，4800 万英国用户，3800 万俄罗斯用户，以及 6700 万巴西用户，你可以看到 World Wide Web 中“World”一词的含义。许多互联网用户还为 Web 创建了内容——你可能是其中之一！虽然精确地度量网页的数量是很困难的，但是 Google 的最新数据指出，2008 年中期，索引的页面突破了 1 万亿。

在接下来的 24 小时中，互联网的可访问区域将会出现数千万个新的网页。至少有许多页面会出现在私有的 Intranet，它们将为通过局域网连接的业务人员所查看。每个页面和其他已经在网上的 1 万亿个页面相似，都使用超文本标记语言（HTML）。

随着你完成本书中 24 章内容，你的网页将和那些网页一起出现在互联网上。这些课程将帮助你获得当今世界上最宝贵的技能：精通 HTML。

你真的能够自己学会创建顶级的网页，而不需要任何专用的软件，甚至比花在计划和等待高价的 HTML 高手的时间还要少？这本相对简短、容易理解的书能教你学会如何制作高水平的网页。

是的，实际上，通过本书头两章的学习，完全没有 HTML 经验的人也能够拥有一个可以发布在 Web 上的页面。

如何能快速地学会 Web 的语言呢？依靠示例。本书将 HTML 组织为简单的步骤，然后为你精确地说明每个步骤的处理方法。每个 HTML 代码示例在其生成的网页图像之前直接列出。你会看到 HTML 代码是如何运行的，并能获悉其运行背后的原理，然后就可以将它们应用在自己的页面中。

经过 24 章内容的学习，你将对出现在互联网上属于你自己的酷炫页面感到惊讶。

## 超越 HTML

本书不仅包括 HTML，因为 HTML 不是创建现代的 Web 内容所唯一需要了解的内容。

本书的目标是在 24 个简单而又短小的章节中，教会你创建现代的、与标准兼容的网站所需要的所有技能。本书介绍了如下关键技能和技术。

- XHTML（可扩展的超文本标记语言）是网页创建的当前标准。本书的每个示例都是完全兼容 XHTML 的。在合适的地方，还会讲到 HTML 5。
- 本书的所有示例已经用当今的主流 Web 浏览器的最新版本进行过兼容性测试。这些浏览器包括 Apple Safari、Google Chrome、Microsoft Internet Explorer、Mozilla Firefox 和 Opera。
- 有许多包含层叠样式单（CSS）的内容，CSS 使你能够很好地控制布局、字体以及网页格式的各个方面，包括文本和图像。在创建令人瞩目的网页时，CSS 比传统的 HTML 页面自身能做的多得多。例如，你知道 CSS 除了可以正常地查看 Web 外，还可以对具体信息进行处理，以便于打印吗？
- 第 10 章～第 12 章为你介绍多媒体应用程序及其使用，其中包含了可以免费下载并免费试用业界标准的软件的地址。
- 仅讲解技术内容是不够的，所以本书还包含了在建立网站以实现目标时所需要的建议。关键的细节——设计有效的页面布局，通过 FTP 软件将你的页面张贴到互联网上，组织和管理多个页面，让你的页面出现在主流互联网搜索网站查询列表的前列——都进行了足够深度的介绍，这可使你避免常常使初学者沮丧的困难。

对以上关键技术的讲解是使本书前 7 个版本成为畅销书的原因，而这个更新的版本——2005 年以来本书的第一个版本——也没有差别。所有的示例都做了更新，而且大部分内容都重新修订过，以与新的示例和新技术相匹配。

## 直观的示例

本书的每个示例都分为两个部分。

- 先给出构成 HTML 页面的文本内容，其中所有的 HTML 和 CSS 代码被突出显示。
- 随后给出生成的页面，而且其显示效果与在所有主流浏览器中看到的相同。

你往往可以将示例改编为自己的网页，而完全不需要阅读跟随的文本。

本书中的所有示例都是与标准兼容的，可以用于 Apple Safari、Google Chrome、Microsoft Internet Explorer、Mozilla Firefox 和 Opera。虽然所有的屏幕截图取自于 Firefox，但是所有代码都已经在其他浏览器中测试过。

## 特殊元素

当你完成每章内容后，旁注可帮助你刚学习的内容应用到自己的网页上。

**By the  
Way**

**注意：**注释框提供关于所讨论主题的附加信息。

***Did you  
Know?***

**提示：**节约你宝贵时间的提示和技巧被放置在提示框中，这样你可以快速地发现它们。

***Watch  
Out!***

**警告：**当有你需要注意的事项时，你将在警告框中看到。

## 问答、测验和练习

每一章都以一个简短的问答对话结束，这个对话提出了每个人希望自己敢提出来的“傻问题”。一个简短而完整的测验让你能够测试自己，以确认理解了该章介绍的所有知识。最后，一两个可选的练习让你有机会在继续之前实践你的新技能。



# 目 录

第 1 章 理解 Web 的工作方式..... 1	
1.1 HTML 和万维网简史.....1	
1.2 创建 Web 内容.....2	
1.3 理解 Web 内容交付.....3	
1.4 选择 Web 托管提供商.....4	
1.5 用多种 Web 浏览器测试.....6	
1.6 总结.....6	
1.7 问与答.....6	
1.8 作业.....7	
1.8.1 测验.....7	
1.8.2 测验答案.....7	
1.9 练习.....8	
第 2 章 发布 Web 内容..... 9	
2.1 创建用于本章的示例文件.....9	
2.2 使用 FTP 传送文件.....10	
2.2.1 选择一个 FTP 客户端.....10	
2.2.2 使用一个 FTP 客户端.....11	
2.3 理解 Web 服务器上放置文件的 位置.....13	
2.3.1 基本文件管理.....13	
2.3.2 使用一个索引页面.....14	
2.4 在没有 Web 服务器情况下 分发内容.....15	
2.4.1 本地发布内容.....16	
2.4.2 在博客上发布内容.....16	
2.5 测试 Web 内容.....16	
2.6 总结.....17	
2.7 问与答.....17	
2.8 作业.....18	
2.8.1 测验.....18	
2.8.2 测验答案.....18	
2.9 练习.....18	
第 3 章 理解 HTML 和 XHTML 的 关系..... 19	
3.1 从一个简单的网页开始.....20	
3.2 每个 XHTML 网页必须 有的 HTML 标签.....23	
3.3 使用分段和换行来组织页面.....24	
3.4 用标题组织内容.....26	
学习他人的网页.....28	
3.5 验证你的 Web 内容.....28	
3.6 关于 HTML、XML、XHTML 和 HTML 5 的最新消息.....29	
3.7 总结.....30	
3.8 问与答.....31	
3.9 作业.....31	
3.9.1 测验.....32	
3.9.2 测验答案.....32	
3.10 练习.....32	



<b>第 4 章 理解层叠样式单</b> ..... 33	<b>第 7 章 使用表格显示信息</b> ..... 70
4.1 CSS 工作原理.....34	7.1 创建简单表格.....70
4.2 一个基本的样式单.....34	7.2 控制表格大小.....73
4.3 CSS 样式入门.....38	7.3 表格中的对齐和跨越.....75
4.3.1 布局属性.....38	7.4 使用表格的页面布局.....77
4.3.2 格式化属性.....39	7.5 总结.....78
4.4 使用样式类.....41	7.6 问与答.....79
4.5 使用样式 ID.....43	7.7 作业.....79
4.6 内部样式单和内联样式.....44	7.7.1 测验.....79
验证你的样式单.....45	7.7.2 测验问题.....80
4.7 总结.....45	7.8 练习.....80
4.8 问与答.....46	<b>第 8 章 使用外部和内部链接</b> ..... 81
4.9 作业.....46	8.1 使用 Web 地址.....81
4.9.1 测验.....47	8.2 使用链接锚在页面中链接.....83
4.9.2 测验答案.....47	8.2.1 用链接锚指定页面中的
4.10 练习.....47	位置.....83
<b>第 5 章 使用文本块和列表</b> ..... 48	8.2.2 链接到锚位置.....84
5.1 在页面上对齐文本.....49	8.3 在你的 Web 内容之间链接.....86
5.1.1 使用属性.....49	8.4 链接到外部 Web 内容.....87
5.1.2 对齐块级别元素.....49	8.5 链接到一个 E-mail 地址.....88
5.2 三种 HTML 列表.....51	8.6 在新浏览器窗口里打开链接.....89
5.3 将列表放在列表中.....52	8.7 使用 CSS 设置超链接样式.....90
5.4 总结.....56	8.8 总结.....92
5.5 问与答.....57	8.9 问与答.....93
5.6 作业.....57	8.10 作业.....93
5.6.1 测验.....57	8.10.1 测验.....94
5.6.2 测验答案.....57	8.10.2 测验答案.....94
5.7 练习.....58	8.11 练习.....94
<b>第 6 章 使用字体</b> ..... 59	<b>第 9 章 使用颜色</b> ..... 95
6.1 粗体、斜体和特殊文本格式.....60	9.1 选择颜色的最佳方法.....95
6.2 调整字体.....62	9.2 理解 Web 颜色.....96
6.3 使用特殊字符.....65	9.3 使用十六进制颜色值.....98
6.4 总结.....67	9.4 使用 CSS 设置背景、文本和
6.5 问与答.....68	边框颜色.....98
6.6 作业.....68	9.5 总结.....100
6.6.1 测验.....69	9.6 问与答.....100
6.6.2 测验答案.....69	9.7 作业.....101
6.7 练习.....69	9.7.1 测验.....101

9.7.2 测验答案.....	101	11.10.1 测验.....	129
9.8 练习.....	101	11.10.2 测验答案.....	130
<b>第 10 章 创建用于 Web 上的图像.....</b>	<b>102</b>	11.11 练习.....	130
10.1 选择图像软件.....	102	<b>第 12 章 在网页中使用多媒体.....</b>	<b>131</b>
10.2 必须知道的图像基础 知识.....	104	12.1 链接到多媒体文件.....	132
10.3 准备照片图像.....	104	12.2 嵌入多媒体文件.....	134
10.3.1 裁剪图像.....	105	12.3 使用多媒体的其他技巧.....	136
10.3.2 改变图像大小.....	106	12.4 总结.....	137
10.3.3 调整图像颜色.....	107	12.5 问与答.....	138
10.3.4 控制 JPEG 压缩.....	107	12.6 作业.....	139
10.4 创建横幅和按钮.....	108	12.6.1 测验.....	139
10.5 减少图像中的颜色数量.....	109	12.6.2 测验答案.....	139
10.6 使用透明的图像.....	110	12.7 练习.....	139
10.7 创建平铺背景.....	110	<b>第 13 章 使用框架.....</b>	<b>140</b>
10.8 创建 Web 动画.....	111	13.1 什么是框架?.....	140
10.9 总结.....	112	13.2 建立框架集.....	141
10.10 问与答.....	112	13.2.1 创建一个框架集文档.....	142
10.11 作业.....	112	13.2.2 添加单独的框架.....	144
10.11.1 测验.....	112	13.3 在框架和窗口之间链接.....	144
10.11.2 测验答案.....	113	13.4 使用内联框架.....	145
10.12 练习.....	113	13.5 总结.....	147
<b>第 11 章 在网页中使用图像.....</b>	<b>114</b>	13.6 问与答.....	147
11.1 在网页上放置图像.....	115	13.7 作业.....	148
11.2 用文本描述图像.....	116	13.7.1 测验.....	148
11.3 指定图像高度和宽度.....	117	13.7.2 测验答案.....	148
11.4 对齐图像.....	118	13.8 练习.....	149
11.4.1 水平对齐图像.....	118	<b>第 14 章 使用边距、填充、对齐和 浮动.....</b>	<b>150</b>
11.4.2 垂直对齐图像.....	119	14.1 使用边距.....	151
11.5 将图像变为链接.....	121	14.2 填充元素.....	157
11.6 使用背景图像.....	123	14.3 保持对齐.....	159
11.7 使用图像映射.....	124	14.4 理解 Float 属性.....	159
11.7.1 为什么图像映射不总是 必要的?.....	124	14.5 总结.....	162
11.7.2 映射图像中的区域.....	125	14.6 问与答.....	162
11.7.3 创建用于图像映射的 HTML.....	126	14.7 作业.....	162
11.8 总结.....	128	14.7.1 测验.....	162
11.9 问与答.....	129	14.7.2 测验答案.....	162
11.10 作业.....	129	14.8 练习.....	162

<b>第 15 章 理解 CSS 盒子模型和定位</b> ..... 163	17.7 练习..... 193
15.1 CSS 盒子模型..... 163	<b>第 18 章 使用鼠标操作修改文本显示</b> ..... 194
15.2 定位详解..... 166	18.1 用 CSS 创建工具提示..... 194
15.3 控制元素的堆叠顺序..... 169	18.2 用 CSS 显示附加的翻转文本..... 197
15.4 管理文本流..... 171	18.3 访问事件..... 198
15.5 总结..... 171	18.4 使用 onclick 改变<div>外观..... 199
15.6 问与答..... 171	18.5 总结..... 203
15.7 作业..... 172	18.6 问与答..... 203
15.7.1 测验..... 172	18.7 作业..... 203
15.7.2 测验答案..... 172	18.7.1 测验..... 204
15.8 练习..... 172	18.7.2 测验答案..... 204
<b>第 16 章 使用 CSS 进一步控制列表</b> ..... 173	18.8 练习..... 204
16.1 复习 HTML 列表..... 173	<b>第 19 章 创建固定或者流动的布局</b> ..... 205
16.2 CSS 盒子模型对列表的影响..... 174	19.1 理解固定布局..... 206
16.3 放置列表项目标志..... 176	19.2 理解流动布局..... 207
16.4 用列表项目和 CSS 创建图像映射..... 177	19.3 创建固定/流动混合布局..... 209
16.5 总结..... 180	19.3.1 从基本的布局结构开始..... 209
16.6 问与答..... 180	19.3.2 在固定/流动混合布局中定义两个栏目..... 210
16.7 作业..... 181	19.3.3 设置布局的最小宽度..... 211
16.7.1 测验..... 181	19.3.4 处理固定/流动混合布局中的栏目高度..... 212
16.7.2 测验答案..... 181	19.4 总结..... 216
16.8 练习..... 181	19.5 问与答..... 216
<b>第 17 章 使用 CSS 设计导航</b> ..... 182	19.6 作业..... 217
17.1 导航列表与常规列表的不同..... 182	19.6.1 测验..... 217
17.2 用 CSS 创建垂直导航..... 183	19.6.2 测验答案..... 217
17.2.1 设置单级垂直导航样式..... 185	19.7 练习..... 217
17.2.2 设置多级垂直导航的样式..... 186	<b>第 20 章 创建打印友好的网页</b> ..... 218
17.3 用 CSS 创建水平导航..... 190	20.1 如何成为打印友好的网页..... 219
17.4 总结..... 192	20.2 应用与媒体相关的样式单..... 220
17.5 问与答..... 192	20.3 设计用于打印页面的样式单..... 222
17.6 作业..... 193	20.4 预览网页的打印结果..... 224
17.6.1 测验..... 193	
17.6.2 测验答案..... 193	

20.5	总结	225	23.2	组织简单的网站	252
20.6	问与答	226	23.3	组织较大的网站	253
20.7	作业	226	23.4	编写易于维护的 HTML	
20.7.1	测验	226		代码	255
20.7.2	测验答案	226	23.4.1	使用注释说明代码	256
20.8	练习	226	23.4.2	缩进使代码更清晰	257
<b>第 21 章</b>	<b>理解动态网站</b>	228	23.5	总结	257
21.1	理解不同类型的脚本	228	23.6	问与答	258
21.2	在 HTML 中包含		23.7	作业	258
	JavaScript	229	23.7.1	测验	258
21.3	显示随机内容	231	23.7.2	测验答案	258
21.4	理解文档对象模型	233	23.8	练习	259
21.5	根据用户交互改变图像	234	<b>第 24 章</b>	<b>帮助人们找到你的网站</b>	260
21.6	总结	235	24.1	宣传网站	260
21.7	问与答	235	24.2	在主要的搜索网站列出你的	
21.8	作业	236		网站	261
21.8.1	测验	236	24.3	为搜索引擎提供提示	262
21.8.2	测验答案	236	24.4	更多搜索引擎优化技巧	267
21.9	练习	237	24.5	总结	268
<b>第 22 章</b>	<b>使用基于 Web 的表单</b>	238	24.6	问与答	268
22.1	HTML 表单的工作原理	238	24.7	作业	269
22.2	创建表单	239	24.7.1	测验	269
22.3	接受文本输入	242	24.7.2	测验答案	269
22.4	标识每项表单数据	243	24.8	练习	269
22.5	在表单中包含隐藏数据	243	<b>附录 A</b>	<b>完整的 XHTML 1.1 和 CSS 2</b>	
22.6	研究表单输入控件	243		快速参考	270
22.6.1	复选框	243	A.1	XHTML 结构	271
22.6.2	单选按钮	245	A.2	XHTML 文本短语和段落	273
22.6.3	选择列表	245	A.3	XHTML 文本格式化元素	275
22.6.4	文本区域	246	A.4	XHTML 列表	276
22.7	提交表单数据	246	A.5	XHTML 链接	277
22.8	总结	246	A.6	XHTML 表格	278
22.9	问与答	247	A.7	XHTML 嵌入内容	283
22.10	作业	248	A.8	XHTML 样式	285
22.10.1	测验	248	A.9	XHTML 表单	285
22.10.2	测验答案	248	A.10	XHTML 脚本	288
22.11	练习	249	A.11	XHTML 常用属性	289
<b>第 23 章</b>	<b>组织和管理网站</b>	250	A.12	CSS 尺寸样式属性	289
23.1	一个网页就足够时	251			



A.13	CSS 文本和字体样式 属性.....	290	A.18	CSS 布局和显示样式 属性.....	295
A.14	CSS 背景样式属性.....	292	A.19	CSS 列表和项目符号样式 属性.....	297
A.15	CSS 边框样式属性.....	292	A.20	CSS 表格样式属性.....	298
A.16	CSS 边距样式属性.....	294			
A.17	CSS 填充样式属性.....	295			



# 第 1 章

## 理解 Web 的工作方式

---

本章中你将学到：

- 万维网（WWW）的简史
- “网页（Web Page）”的含义，以及该术语不能反映所涉及的所有内容的原因
- 如何从你的个人计算机进入别人的浏览器
- 选择 Web 托管提供商的方法
- 不同的 Web 浏览器和设备类型对网面内容的影响

在学习复杂的 HTML 和 CSS 之前，先对这些技术有坚实的理解是很重要的，这些技术可以将这些普通文本文件转化为通过计算机或者手持设备浏览网页时看到的丰富的多媒体显示。

如果不使用 Web 浏览器查看，包含 HTML 和 CSS 的文件则看不出效果，除非有一个 Web 服务器，否则除了你自己以外，没有人能看到你的内容。Web 服务器使你的内容可用于其他人，他们使用自己的 Web 浏览器导航到一个地址，并且等待服务器发送信息给他们。你将深入这个过程，因为你必须首先创建文件，然后将它们放到服务器上使其可用，并且必须确保你的内容将按照你的设想出现在最终用户那里。

### 1.1 HTML 和万维网简史

很久以前，在人类还没有登上月球之前，一些富有远见的人决定看看自己能不能将几个主要的计算机网络连接在一起。我不再赘述这些人的名字和故事（到处都可以找到），但是最终的结果是“所有网络之母”，也就是我们对互联网的称呼。

直到 1990 年，通过互联网访问信息还是一个相当技术性的事情。这非常困难，甚至连拥有博士学位的物理学家在尝试交换数据时都会感到沮丧。一位现在非常有名（并且被授予爵位）

的物理学家 Tim Berners-Lee 爵士，发明了很容易地通过“超文本”链接交叉引用互联网上的文本的方法。

这不是一个新的思路，但是他的简单的超文本标记语言（HTML）能够兴旺起来，而更野心勃勃的超文本计划却失败了。超文本原意是指具有页面之间交叉引用的，以电子形式存储的文本。现在这个术语含义更广，指任何可以与其他对象链接的对象（文本、图像、文件等）。超文本标记语言是描述文本、图形和包含其他信息的文件组织和链接在一起的一种方式的语言。

到 1993 年，世界上只有 100 多台计算机能够提供 HTML 页面。这些互相链接的页面被称为万维网（WWW），许多 Web 浏览器程序被编写出来，使用户能够查看网页。因为 Web 的流行，一些编程人员很快就写出能够和文本一起查看图形的浏览器。从那时开始，Web 浏览器软件的持续发展和 HTML 和 XHTML 的标准化将我们带入了今天所生活的世界，现在有超过 1.1 亿台 Web 服务器提供对超过 250 亿个文本和多媒体文件的请求的服务。

**By the Way**

**注意：**关于万维网历史的更多信息，参见这个主题的 Wikipedia 文章：  
[http://en.wikipedia.org/wiki/History\\_of\\_the\\_Web](http://en.wikipedia.org/wiki/History_of_the_Web)。

这几段话确实是这段不寻常的时期的简单历史。现在学校里的新人根本不知道没有万维网的时代，信息“永远在线”的理念和无处不在的计算将推动我们的生活的各个方面不断前进。Web 内容的创建和管理不再只是少数技术方面的人士（你可以称呼他们“高手”）所拥有的技能，在本书的最后，你将看到这些技能是每个人都能掌握的，而不是高手们的特权。

## 1.2 创建 Web 内容

你可能已经注意到“Web 内容”（而不是“网页”）这个术语的使用——这么说内涵更丰富。尽管我们说“访问一个网页”，但真实的意思是“在我们的计算机上查看所有某个地址上的文本和图像”。我们阅读的文本，看到的图像由 Web 浏览器根据在单独的文件上找到的一些指令显示。

这些文件包含由 HTML 代码标示或者包围的文本，告诉浏览器如何显示文本——作为标题、作为段落、红色的字体等。一些 HTML 标示告诉浏览器显示一个图像或者视频文件而不是普通文本，这就把我带回到那个论点——不同类型的内容被发送到你的 Web 浏览器，所以简单地说“网页”不能涵盖。这里我们使用“Web 内容”这个词来代替，以包含文本、图像、音频、视频和其他在线媒体。

在后面，你将学习链接到或者创建各种网站上找到的多媒体 Web 内容的基础知识。现在你需要记住的是你控制着用户访问你的网站时所看到的。从包含所显示的文本行或者告诉服务器传递一个图形到用户的 Web 浏览器的代码的文件开始，你必须计划、设计并且实现最终组成你的 Web 存在的所有部分。随着你在本书中的学习，只要你理解所有的小步骤，理解这个过程就不困难。

Web 内容的最基本形式从简单的包含 HTML 和 XHTML 标记的文本文件开始。XHTML 是另一种味道的 HTML，“X”代表 eXtensible（可扩展），随着你持续地学习这些课程，将会许多到更多关于它的知识。从一开始就应该知道的重要的事情是本书中的所有示例都是兼容

HTML 4 和 XHTML 的，这意味着它们在现在和将来任何新一代的浏览器上的显示都是近似的。这是编写兼容标准的代码的好处之一，你不需要担心未来因为代码“不能正常工作”而必须进行修改。只要 Web 浏览器坚持标准（有希望在长时间内），你的代码可能会始终“正常工作”。

## 1.3 理解 Web 内容交付

在许多不同的地方发生了多个过程，最终产生了你看到的 Web 内容。这些过程发生得很快——在几毫秒之内——而且是在幕后发生的。换句话说，当我们认为所作的一切是打开一个 Web 浏览器，输入 Web 地址，并且立即看到请求的内容时，幕后的技术正在为我们辛勤地工作。图 1.1 展示了浏览器和服务器之间的基本交互。

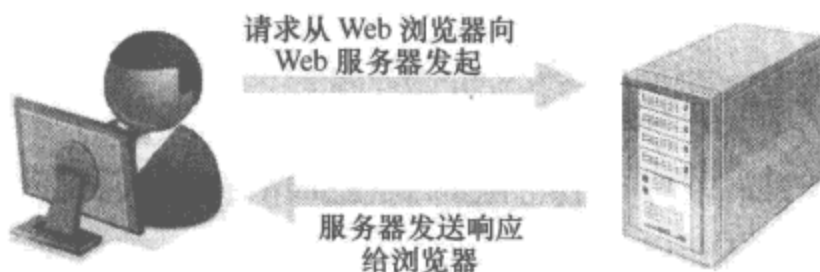


图 1.1

浏览器请求，服务器响应

但是，这个过程中有许多步骤，在你看到整个请求网站的内容之前，可能在浏览器和服务器中有多次往返。

假定你想要进行一次 Google 搜索，所以你认真地在地址栏输入 <http://www.google.com> 或者从书签列表中选择 Google 书签。你的浏览器几乎立即显示出如图 1.2 所示的画面。

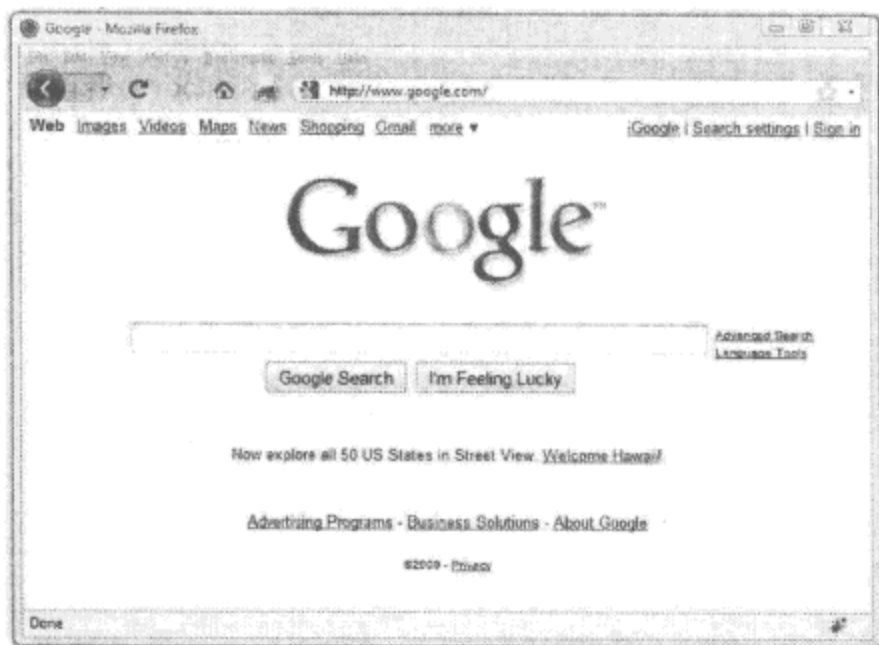


图 1.2

访问 [www.google.com](http://www.google.com)

图 1.2 显示了一个包含文本和一个图形（Google 标志）的网站。从 Web 服务器读取这些文本和图像并且在你的屏幕上显示所发生的简单过程如下。

1. 你的 Web 浏览器发出对位于 <http://www.google.com/> 地址的 `index.html` 文件的一个请求。`index.html` 文件不必是你地址栏上输入的地址的一部分；你将在第 2 章中学习到更多关于 `index.html` 文件的知识。

2. 在接收到对一个具体文件的请求后，Web 服务器进程在它的目录中查找具体的文件，并且将该文件的内容发送回你的浏览器。



3. Web 浏览器接受 `index.html` 文件的内容，这是由 HTML 代码标记的文本，并且根据 HTML 代码显示内容。在显示内容的同时，浏览器发现用于 Google 标识的 HTML 代码，这个标识在图 1.2 中可以看到。HTML 代码类似以下代码：

```

```

这个代码标记提供了告诉浏览器显示该标识所需要的文件源位置 (`src`)、宽度 (`width`)、高度 (`height`)、边框类型 (`border`) 以及辅助文本 (`alt`)。你将通过稍后的课程学习更多关于特性的知识。

4. 浏览器查看 `<img/>` 标记中的 `src` 特性以查找源位置。在这个例子中，图像 `logo.gif` 可以在读取 HTML 文件的相同 Web 地址 (`www.google.com`) 中找到。

5. 浏览器请求 `http://www.google.com/logos/logo.gif` 上的文件。

6. Web 服务器解释该请求，寻找该文件，并且将该文件的内容发送给请求它的浏览器。

7. Web 浏览器在你的显示器上显示该图像。

正如你在 Web 内容交付过程的描述中看到的，Web 浏览器所作的只是像你可以看到内容的一个像框。浏览器组合 Web 内容部件并且按照文件中的 HTML 命令安排这些部分。

你还可以在“本地”或者你的硬盘驱动器上查看 Web 内容，而不需要一个 Web 服务器。内容读取和显示的过程和前面列出的浏览器查找并且解释 HTML 文件的代码和内容的过程相同，但是旅程较短，浏览器在你自己的计算机硬盘上而不是远程机器上寻找文件。解释嵌入在文件中的基于服务器的编程语言需要一个 Web 服务器，但是这超出了本书的范围。实际上，你可以在没有 Web 服务器的情况下完成本书的所有课程，但是在你之外没有人能看到你的杰作。

## 1.4 选择 Web 托管提供商

尽管你可以在没有 Web 服务器的情况下完成本书的所有课程，但是我们建议你使用 Web 服务器。不要担心——得到一个托管提供商一般是个快速、轻松而相对经济的过程。实际上，你可以以稍高于本书的价格得到自己的域名和一年的 Web 托管服务。

如果你在搜索引擎当中输入 `Web hosting provider` (Web 托管提供商)，你将得到上百万个结果和无穷的赞助搜索结果 (也被称为广告)。世界上没有这么多 Web 托管提供商，但是看上去好像有。即使你关注托管提供商的一个受控列表，它也可能太长——特别是如果你寻找的是你自己或者你的公司或者组织放置一个简单的网站的地方时。

你将会希望在寻找一个提供商时缩小范围并且选择最符合你的需求的提供商。Web 托管提供商的一些选择标准如下。

- 可靠性/服务器“正常运行时间”——如果你有一个网站，你希望确保人们能够始终到达它。
- 客户服务——寻求多种联络客户服务的方法 (电话、E-mail、聊天工具) 以及常见问题的在线文档。
- 服务器空间——托管套件是否包含足够的服务器空间来存储所有你计划包含在网站

(如果有)的多媒体文件(图像、音频、视频)。

- 带宽——托管套件是否包含足够的带宽，使所有访问你的网站并且下载文件的人可以放心地进行，并且你不需要付出额外的钱？
- 域名购买和管理——套件是否包含一个自定义的域名，或者你在托管的账目之外需要必须单独购买和维护你的域名？
- 价格——不要为托管付过多的钱。你将会看到很宽的价格范围并且应该立即对“差异是什么？”感到奇怪。差异往往和服务质量无关而取决于公司的开销和公司认为能侥幸成功的要价。一个经验法则是，如果你租用的基本托管和域名套餐每年花费超过 75 美元，你就可能付得太多了。

这里有 3 家可靠的 Web 托管提供商，它们的基本套件在相对低的价格上包含了丰富的服务器空间和带宽（还有域名和额外的好处）。如果你不了解这些 Web 托管提供商，至少你可以将它们的基本套件描述作为你求购的指南。

- A Small Orange (<http://www.asmallorange.com>)——它们的“极小”和“小”主机套件是新的 Web 内容发布者最好的起点。
- DailyRazor (<http://www.dailyrazor.com>)——它们的 RazorLIMIT 和 RazorSTARTER 托管套件是全功能而可靠的。
- LunarPages (<http://www.lunarpages.com>)——基本托管套件适合于所有个人和小企业网站。

好的托管提供商的一个特性是它们为你提供“控制面板”，用来管理你的账户的各个方面。如图 1.3 所示，显示了我自己在 Daily Razor 的 RazorPRO 主机账户的控制面板。许多 Web 托管提供商提供自己的特殊控制面板软件，或者一些设计相似的控制面板——清晰标示的图标使你能够进行配置和管理账户。



图 1.3

一个控制面板实例

你可能永远不需要使用控制面板，但是它可以使你简化数据库和其他软件的安装，查看 Web 统计，以及 E-mail 地址的添加（还有许多其他功能）。如果你能够按照说明书进行，就

能管理自己的 Web 服务器——不需要特殊的培训。

## 1.5 用多种 Web 浏览器测试

刚刚讨论了 Web 内容交付的过程和 Web 服务器的获得,回到使用多种 Web 浏览器测试你的网站看上去有点奇怪。但是,在你离开并且学习所有关于使用 HTML 和 CSS 创建网站的知识之前,要记住这个非常重要的事实:你的网站的每个访问者可能使用和你不同的硬件和软件配置。他们的设备类型(台式机、笔记本、智能电话、iPhone)、屏幕分辨率、浏览器类型、浏览器窗口大小、连接速度——记住,你不能控制访问者查看你的网站时的任何方面。

尽管所有 Web 浏览器以总体相同的方式处理信息,但是它们之间有些特殊的差异导致在不同的浏览器中显示的结果不总是相同。即使使用相同的浏览器的相同版本也可能由于选择不同的显示选项和/或改变视窗大小而改变页面的外观。所有主要的浏览器都允许用户用自己的选择覆盖网页创作者指定的背景和字体。屏幕分辨率、窗口大小和可选的工具栏也可能改变页面第一次出现在屏幕上时所显示的部分。你只能确保你编写了兼容标准的 HTML 和 CSS。

在任何情况下,都不要花费时间去设计在自己的计算机上看上去“完美”的东西——除非你愿意在朋友的计算机上、街上的咖啡店里的计算机上或者你的 iPhone 上查看时感到失望。

你应该始终用尽可能多的 Web 浏览器来测试网站,如下所示。

- Apple Safari (<http://www.apple.com/safari/>) for Mac and Windows
- Google Chrome (<http://www.google.com/chrome>) for Windows
- Mozilla Firefox (<http://www.mozilla.com/firefox>) for Mac, Windows and Linux
- Microsoft Internet Explorer (<http://www.microsoft.com/ie>) for Windows
- Opera (<http://www.opera.com/>) for Mac, Windows and Linux/UNIX

## 1.6 总结

本章为你介绍了使用 HTML 标记文本文件以生成 Web 内容的概念。你还学到了 Web 内容有比“页面”更多的含义——Web 内容还包含图像、音频和视频文件。所有这些内容存在于一个 Web 服务器上——一台通常远离你的计算机的远程机器。在你的计算机或者其他设备上,你使用 Web 浏览器请求、读取,并且最终在你的屏幕上显示 Web 内容。

你学习到在确定 Web 托管提供商是否适合你的需求时所应该考虑的标准。你还学习了在你的作品放置到 Web 服务器上时用多种浏览器测试的重要性。编写正确的、与标准兼容的 HTML 和 CSS 将帮助确保你的网站对于所有访问者来说外观很相似,但是你仍然不应该在没有接受开发团队之外的潜在用户的意见情况下进行设计——当你是一个个人的“设计团队”时得到别人的意见更加重要!

## 1.7 问与答

问: 你用“Web 内容”代替了“网页”,但是我听到人们说“网页”。它们的意思是什么?

这些术语和“主页”、“网站”有什么不同？

答：将万维网比喻为图书馆易于让人们理解。在这个图书馆中，单独的网站就是书。网站上的单独文件就是“页”。一个“网站”由一起创建并且内容相关的一页或者多页组成。“主页”通常指的是人们访问网站时看到的第一个网页。但是问题出现了，当人们在实际上意为“来我的网站”时说“访问我的网页”——网站充满了网页。将一个 Web 内容集合称作页面而不是网站说明你还没有真正理解 Web 的工作方式——这是因为你不理解 Web 内容在一起创建网站的方式，或者因为你已经计划，设计并且实现了一个内容都在一个页面上的网站！

问：我已经看到了 Internet 上一些网页的“源代码”，它看起来难以学习。我必须像编程人员那样思考才能学习这些内容吗？

答：尽管复杂的 HTML 页面确实看起来令人生畏，但是学习 HTML 比学习真正的软件编程语言（如 C++ 或者 Java）要容易得多。HTML 是一种标记语言而不是编程语言，标记文本可以被浏览器以特定的方式呈现。这是与开发计算机程序完全不同的思想过程。你真的不需要任何编程人员的经验或者技能，就能成为成功的 Web 内容创作者。

许多商业网站背后的 HTML 看上去很复杂的原因之一是它可能由可视化的 Web 设计工具创建——“所见即所得”或者“WYSIWYG”编辑器将使用软件开发者告诉它在特定情况下使用的标记——而在手工编码中，你完全控制所产生的标记。在本书中，你从头开始学习基础的编码，这一般产生清晰易读的源代码。可视化 Web 设计工具有让代码难以理解的倾向，还会生成复杂而不兼容标准的代码。

## 1.8 作业

作业包括测验和帮助你巩固对包含的素材理解的联系。在看后面的答案之前，尝试回答所有问题。

### 1.8.1 测验

1. 定义术语“Web 内容”。
2. 产生一个带有一些文本和两个图像的网页需要在 Web 服务器上存储几个文件？
3. 寻找一个 Web 托管供应商时，需要注意的一些特性是什么？

### 1.8.2 测验答案

1. Web 内容是描述从 Web 服务器传递到 Web 浏览器的文本、图像、音频、视频和其他媒体文件的全称。
2. 你需要 3 个文件：一个用于网页本身，其中包含了文本和 HTML 标记；两个图像各为一个文件。
3. 寻求可靠性、客户服务、Web 空间和带宽、域名服务、网站管理附加功能以及价格。



## 1.9 练习

- 安排你的 Web 托管——你打算在你自己的计算机上本地查看文件来完成本书的课程，还是打算使用一个 Web 托管提供商？注意，大部分 Web 托管提供商将使你在购买托管计划的同一天就能够工作。



## 第 2 章

# 发布 Web 内容

---

本章中你将学到：

- 使用文本编辑器创建一个基本的 HTML 文件的方法
- 使用 FTP 将文件传送到你的 Web 服务器的方法
- 文件在 Web 服务器上应该存储的位置
- 在没有 Web 服务器的情况下分发 Web 内容的方法
- 使用其他发布方法（如博客）

在前一章中，你学习了通过 Web 浏览器请求 Web 内容的过程以及 Web 服务器对这些请求的响应方式，在本章中，将学到你作为内容创建者，使 Web 内容在网上可用的入手点——你必须在一个 Web 服务器上发布这些内容以便其他人能找到它。

### 2.1 创建用于本章的示例文件

在我们开始之前，看看程序清单 2.1，这个程序清单表现了一些简单的 Web 内容——以大的加重字体，在浏览器窗口居中的两行打印“Hello World! Welcom to My Web Server。”（你好，世界！欢迎来到我的 Web 服务器。）的 HTML 代码行。

程序清单 2.1 我们的 HTML 文件示例

```
<html>
<head>
<title>Hello World!</title>
</head>
<body>
<h1 align="center">Hello World!<br/>Welcome to My Web Server.</h1>
</body>
</html>
```

---

为了使用这些内容,打开你选择的一个文本编辑器,如记事本(Windows 上)或者 TextEdit (在 Mac 上)。不要使用写字板、Microsoft Word 或者其他全功能的字处理软件,因为这些软件创建的文件不同于我们用于 Web 内容的普通文本文件。

### By the Way

**注意:** 你将在第 3 章学到更多关于文本编辑器的内容。现在,我们只希望你有一个可以放到 Web 服务器上的示例文件!

输入你在程序清单 2.1 中看到的内容并以 `sample.html` 文件名保存。`.html` 扩展名告诉 Web 服务器你的文件确实充满了 HTML。当文件内容被发送到请求它的 Web 浏览器,浏览器也将指导它是 HTML,并且正确地呈现它。

现在你有了一个 HTML 文件示例——并且幸运地有一个地方放置它,比如 Web 托管账户——让我们来发布你的 Web 内容。

## 2.2 使用 FTP 传送文件

正如你目前所学习到的,你应该将 Web 内容放置到一个 Web 服务器上使其可为其他人所访问。这个过程一般通过使用文件传输协议(FTP)来完成。为了使用 FTP,你需要一个 FTP 客户端——一个用于从你的计算机向 Web 服务器传送文件的程序。

FTP 客户端需要 3 部分的信息以便连接到你的 Web 服务器,这些信息将在你的托管提供商建立你的账户后发送给你。

- 你所连接的主机名或者地址
- 你的账户用户名
- 你的账户密码

一旦你有了这些信息,你就可以使用一个 FTP 客户端来将内容传递到你的 Web 服务器。

### 2.2.1 选择一个 FTP 客户端

不管你使用什么 FTP 客户端,FTP 客户端一般使用相同类型的界面。如图 2.1 所示,显示了一个 FireFTP 的示例,它是用于 Firefox 浏览器的 FTP 客户端。本地计算机的目录列表出现在屏幕的左边,远程机器(Web 服务器)的目录列表出现在右边。一般你将看到右箭头和左箭头按钮(见图 2.1)。右箭头将所选文件从你的计算机发送到 Web 服务器,左箭头从 Web 服务器发送文件到你的计算机。许多 FTP 客户端还允许你选择文件并且拖放到目标机器。

你有许多免费的 FTP 客户端可用,但是你也可以通过基于 Web 的文件管理器工具来传递文件,这可能是你的 Web 服务器控制面板的一部分。但是,这种文件传输方法一般引入更多的步骤,不像在你的计算机上安装一个 FTP 客户端那么简单。

下面是一些流行的免费 FTP 客户端。

- Classic FTP (<http://www.ncsoftware.com/classic/>) for Mac and Windows
- Cyberduck (<http://cyberduck.ch/>) for Mac

- Fetch (<http://fetchsoftworks.com/>) for Mac
- FileZilla (<http://filezilla-project.org/>) 用于所有平台
- FireFTP (<http://fireftp.mozdev.org/>) 用于所有平台的 Firefox 扩展

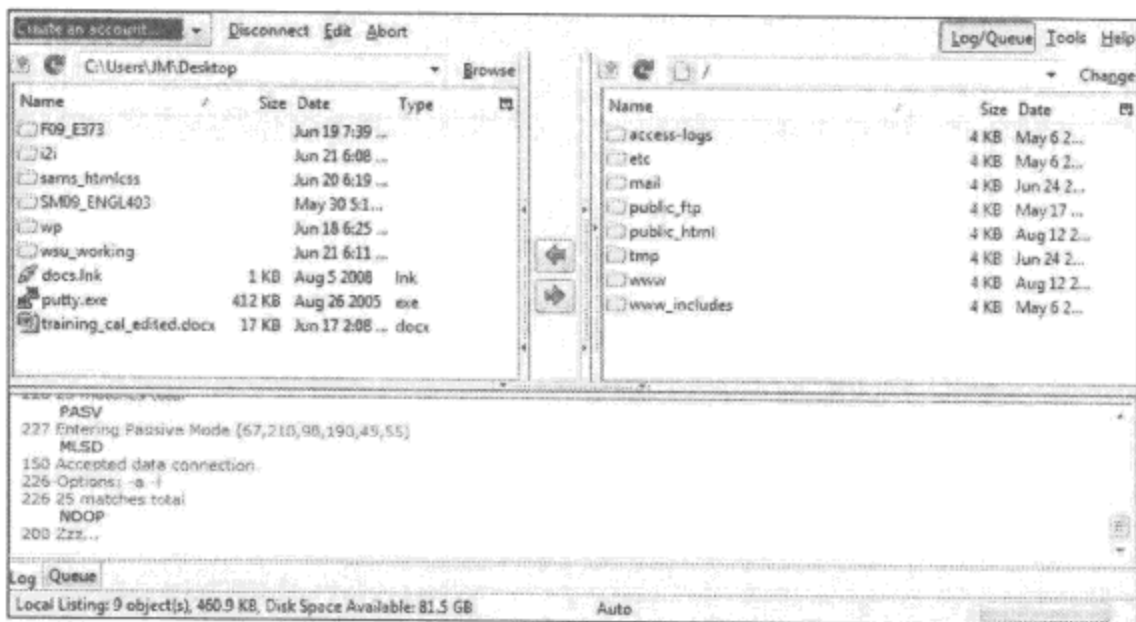


图 2.1  
FireFTP 界面

一旦你选择了一个 FTP 客户端并且安装在你的计算机上，你就可以上传和从你的 Web 服务器下载文件。在下一小节，你将看到这一过程，使用本章开始创建的示例文件。

### 2.2.2 使用一个 FTP 客户端

接下来的步骤说明了使用 Classic FTP 连接到你的 Web 服务器并且传送一个文件的方法。但是，所有 FTP 客户端使用大致相似的界面。如果你理解了下面的步骤，你应该能够使用任何 FTP 客户端。

记住，你首先需要主机名、账户用户名和账户密码。

1. 启动 Classic FTP 程序，单击 Connect (连接) 按钮。你将被提示输入希望连接的网站的信息，如图 2.2 所示。

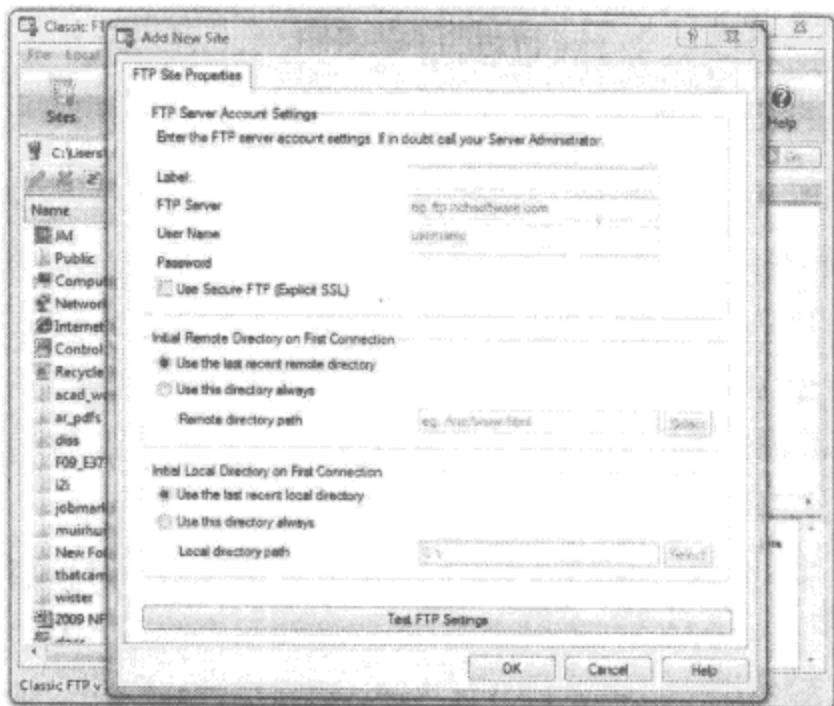


图 2.2  
在 Classic FTP 中  
连接到新的网站

2. 输入图 2.2 中所示的所有项目。

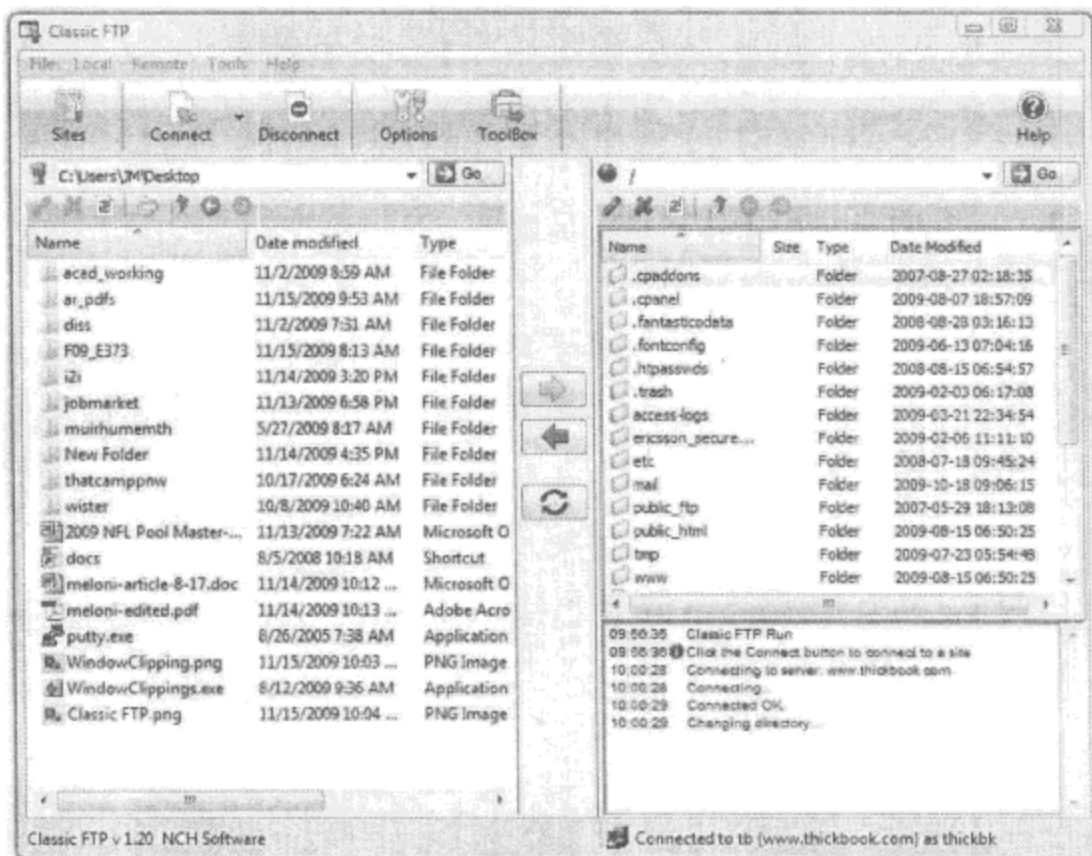
- Site Label (网站标签) 是你用于引用自己的网站的名称。其他人不会看到这一名称, 所以输入你希望的任何名称。
- FTP Server (FTP 服务器) 是你需要发送网页的 Web 服务器的 FTP 地址。这个地址将由你的 Web 托管提供商给出。它可能是 `yourdomain.com`, 但是要检查签署服务时收到的信息。
- User Name (用户名) 和 Password (密码) 也应该使用你的托管提供商给出的信息填写。
- 在你习惯使用客户端并且确定了工作流之前, 不要改动 Initial Remote Directory on First Connection (第一次连接时的初始远程目录) 和 Initial Local Directory on First Connection (第一次连接时的初始本地目录)。

3. 完成了这些设置之后, 单击 OK 保存设置并且建立与 Web 服务器的连接。

4. 你将会看到一个对话框指示 Classic FTP 尝试连接到 Web 服务器。在成功的连接时, 你将看到如图 2.3 所示的界面, 在左边显示本地目录内容, 右边显示 Web 服务器的内容。

图 2.3

通过 Classic FTP 的一个成功的远程 Web 服务器连接



5. 你现在几乎已经为向 Web 服务器传送文件做好了准备。剩下的工作就是将目录修改为你的 Web 服务器的称为文档根的目录。你的 Web 服务器的文档根目录是被设计为你的 Web 内容最高级目录的目录——目录结构的起始点, 你将在本章中学到更多关于它的内容。通常, 这个目录被命名为 `public_html` (如图 2.3 所示)、`www` (在图 2.3 中也能看到, 因为 `www` 被创建为 `public_html` 的别名) 或者 `htdocs`。这不是你必须创建的目录, 因为你的 Web 托管提供商已经为你创建了。

6. 双击文档根目录打开它。FTP 客户端界面的右边将显示为该目录的内容 (这时候它可能为空, 除非你的 Web 托管提供商已经为你在这个目录中放入了占位文件)。

7. 目标是将你前面创建的 `sample.html` 文件从你的计算机传送到 Web 服务器。在 FTP 客户端界面的左边的目录列表里寻找文件并且单击以加亮文件名。



8. 单击客户端界面中间的右箭头按钮将文件发送到 Web 服务器。一旦文件传送完成，客户端界面的右边将刷新，让你看到文件已经到达其目的地。

9. 单击 Disconnect (断开) 按钮关闭连接，然后退出 Classic FTP 程序。

每当你要通过 FTP 传送文件到你的 Web 服务器，概念上都与这些步骤相似。你还可以使用你的 FTP 客户端创建远程 Web 服务器上的子目录。为了使用 Classic FTP 创建子目录，单击 Remote (远程) 菜单，并且单击 New Folder (新建文件夹)。不同的 FTP 客户端有达到相同目标的不同界面选项。

## 2.3 理解 Web 服务器上放置文件的位置

维护 Web 内容的一个重要方面是确定组织内容的方法——不仅是为了用户寻找，也为了你在服务器上的维护。将文件放到指定目录中将帮助你管理这些文件。

在你的 Web 服务器上命名和管理目录，并且开发文件维护规则，这完全取决于你。但是，维护一个精心组织的服务器使你在长期中更加有效地管理它的内容。

### 2.3.1 基本文件管理

在你浏览 Web 时，你可能已经注意到 URL 随着你在网站之间浏览而变化。例如，如果你查看一个公司的网站并且单击图形化导航进入公司的产品或者服务，URL 可能从

`http://www.companyname.com/`

变成

`http://www.companyname.com/products/`

或者

`http://www.companyname.com/services`

在前一小节中，我使用术语“文档根目录”而未作真正的解释。Web 服务器的文档根目录实际上是完整的 URL 中后缀的“斜杠”。例如，如果你的域名是 `yourdomain.com`，你的 URL 为 `http://www.yourdomain.com/`，那么文档根目录是由后续的斜杠 (/) 代表的目录。文档根是在你的 Web 服务器上创建的目录结构的起始点，这时 Web 服务器开始查找 Web 浏览器请求的文件的开始位置。

如果你将 `sample.html` 按照刚才的指示放在文档根目录，那么你将能够通过 Web 浏览器在如下 URL 中访问它：

`http://www.yourdomain.com/sample.html`

如果你在 Web 浏览器中输入这个 URL，你将看到如图 2.4 所示的 `sample.html`。

但是，如果你在文档根目录中创建了一个新的目录 (`newdirectory`) 并且将 `sample.html` 放在那个目录，那么文件可能在如下 URL 中访问：

`http://www.yourdomain.com/newdirectory/sample.html`

如果你将 `sample.html` 放在你连接服务器时原来看到的目录中——也就是说，你不改变目

录并将文件放在文档根目录——那么 `sample.html` 文件将无法以任何 URL 在你的 Web 服务器上访问到。文件将仍然存在于你的 Web 服务器上，但是因为文件不在文档根目录——服务器软件所知的开始寻找文件的位置——它将永远不能被通过 Web 浏览器访问的人读取到。

图 2.4

通过一个 Web 浏览器访问的 `sample.html` 文件



根本的原则是什么？始终导航到你的 Web 服务器的文档根目录，然后才开始传送文件。

对于图形和其他多媒体文件尤其如此。Web 服务器上常用的一个目录称为“images”，你可以想象，这是所有图像资源放置的位置。其他常见的目录包括用于样式单文件的“CSS”（如果你使用超过一个样式单）以及用于外部 JavaScript 文件的“JS”。或者，如果你知道你将在网站上安排一个部分，访问者可以从这里下载许多其他不同类型的文件，你可能称这个目录为“downloads”。

不管是包含你的艺术相集的 ZIP 文件还是具有销售数字的 Excel 試算表，在互联网上发布网页之外的文件都是有帮助的。为了使非 HTML 文件在 Web 上可用，只要按照本章前面的上载说明，将文件如同 HTML 文件一样上传到 Web 服务器。当文件上传到 Web 服务器之后，你可以创建一个指向它的链接（你将在后面的章节中学习）。换句话说，你的 Web 服务器能够提供远远超过 HTML 的服务。

本章中你将学习到一个 HTML 代码样板。如下的代码将被用于一个名为 `artfolio.zip` 文件，该文件位于你的网站的 `downloads` 目录，链接文本称为“Download my art portfolio!”：

```
<a href="/downloads/artfolio.zip">Download my art portfolio!</a>
```

### 2.3.2 使用一个索引页面

当你想到一个索引的时候，你可能想到的是书籍最后告诉你到哪里寻找各种关键词和主题的部分。Web 服务器目录中的索引文件可以服务于这一目的——如果你这么设计它。实际上，这就是这个名字的来源。

`index.html`（或者只称之为索引文件，正如一般提到的那样）是你为希望人们在导航到你的网站的特定目录时看到的默认文件所取的名字。如果你创建该页的时候牢记可用性，你的用户将从索引页面中得到该部分的所有内容。

例如，如图 2.5 所示，显示了包含指向 3 个页面的链接的下拉式导航和左边导航：`Solutions Overview`（解决方案概述，索引页本身所在的部分）、`Connection Management`（连接管理）和 `Cost Management`（成本管理）。页面本身的内容——称为 `index.html`，位于 `solutions` 目录

中——还链接到 solutions 部分的其他两个页面。当用户到达这个特殊网站的 solutions 部分时，他们能够到达该部分的任意其他页面（并且可以用 3 种不同的途径）。

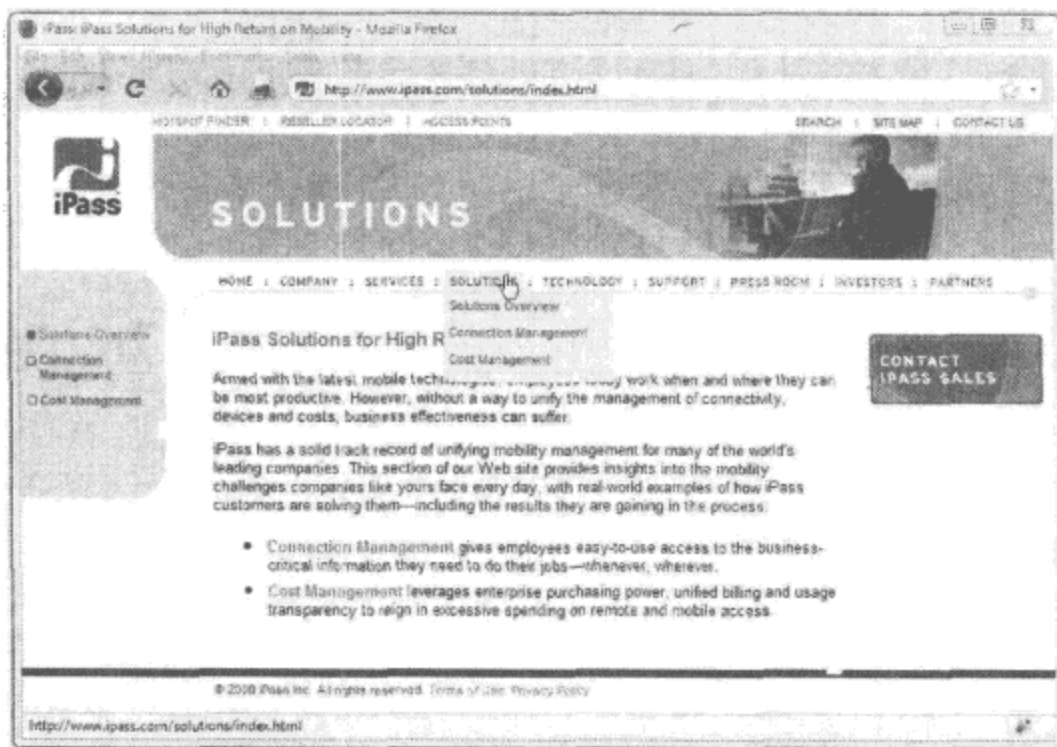


图 2.5

展示一个好的区域索引页面

索引页面的另一个功能是当用户访问你的网站上具有索引页面的一个目录，但是没有指定该页时，他们将会登录到你的网站该部分（或者网站本身）的主页面上。

例如，在前一个示例中，用户可以输入下面的某个 URL 并且登录到网站 solutions 部分的主页面：

<http://www.ipass.com/solutions/>

<http://www.ipass.com/solutions/index.html>

如果 solutions 目录下没有 index.html 页面，结果将取决于 Web 服务器的配置。如果该服务器被配置为不允许目录浏览，用户将在尝试访问没有具体的页面名称的 URL 时看到“Directory Listing Denied”（目录列表被拒绝）信息。但是，如果服务器被配置为允许目录浏览，用户将会看到该目录的文件。

这些服务器配置选项将已经由你的主机提供商为你确定。如果你的主机提供商允许你通过控制面板修改服务器配置，那么你可以修改这些设置，这样你的服务器根据你的需求对请求进行响应。

索引文件不仅用于子目录，它也用在网站的顶级目录（或者文档根目录）。你的网站的第一个页面——或者称为主页（home page），或者其他用来称呼你希望用户在访问你的域时首先看到的 Web 内容的词——应该被命名为 index.html 并且放在 Web 服务器的文档根目录中。这将确保用户在 Web 浏览器中输入 <http://www.yourdomain.com/> 时，服务器将以你打算让他们看到的内容响应（而不是“目录列表被拒绝”或者其他不希望的后果）。

## 2.4 在没有 Web 服务器情况下分发内容

在网上发布 HTML 和多媒体文件是学习 HTML 并且创建 Web 内容的主要原因。但是在有些情况下却不是这样。例如，你可能希望在一个商业展览会上分发带有被设计为 Web 内容

的营销材料的 CD-ROM、DVD-ROM 或者 U 盘——也就是说，通过 Web 浏览器能够查看的超链接文本，但是不需要有 Web 服务器。你还可能希望将基于 HTML 的说明手册包含在培训班的学生的可移动媒体上。这只是 HTML 页面可以被用在不涉及互联网的发布方案的两个示例。

这个过程也被称为创建本地网站。即使没有包含 Web 服务器，这些超文本内容仍然被称为网站。使用本地这个术语是因为你的文件在本地而不是远程（通过 Web 服务器）访问。

### 2.4.1 本地发布内容

我们假定你需要创建一个希望分发到 U 盘上的本地网站。现在即使最便宜的 U 盘也能包含这么多的数据——而基本的超文本文件相当小——你可以在一个小 U 盘上分发整个网站和一个全功能的 Web 浏览器。

#### By the Way

**注意：**发布一个 Web 浏览器在创建一个本地网站时不是必须的，但是这是很好的做法。你可以合理地假定用户有自己的 Web 浏览器并且将打开目录中的 index.html 开始浏览超链接内容。但是，如果你希望在 U 盘上分发一个 Web 浏览器，到 <http://www.portableapps.com/> 寻找 Portable Firefox。

只要把你的 U 盘的目录结构看作 Web 服务器的目录结构。U 盘的顶级目录可以是你的文档根目录。如果你和内容一起分发一个浏览器，你可能有两个目录——例如，一个名为 browser，另一个名为 content。在这种情况下，content 目录将是你的文档根目录。在文档根目录中，你可以有更多的子目录，用来放置内容和其他多媒体文件。

保持好的组织对本地网站和远程网站同样重要，这样你避免了 HTML 文件中失败的链接。你将在后面的章节中学习更多关于链接在一起的文件的知识。

### 2.4.2 在博客上发布内容

你可能有一个第三方主办（如 Blogger 或者 WordPress）的博客，因此在没有专用的 Web 服务器甚至不了解任何 HTML 的情况下已经发布了内容。这些服务除了源代码编辑器之外还提供可视化编辑器，这意味着你可以输入你的文字并且添加视觉格式如加粗、斜体或者字体颜色，而不需要了解这些操作的 HTML。但是这些内容在你单击这些编辑器的 Publish（发布）按钮时仍然变成 HTML。

但是，使用你在本书中将会获得的知识，你的博客将因为你能够为博客帖子内容和博客模板使用源代码编辑器而改进，从而给你对这些内容的观感更多的控制。这些操作和你学到的创建一个 HTML 文件和通过 FTP 上传到专用的 Web 服务器的过程不同，但是我如果不指出博客实际上是 Web 发布的一种形式，那就是我的疏忽。

## 2.5 测试 Web 内容

当你传送文件到 Web 服务器或者将它们放置到可移动媒体用于本地浏览，你应该立即完



全测试所有页面。下面的检查列表将帮助确保你的 Web 内容的表现符合你的预期。注意，某些术语现在对你来说有点不熟悉，但是随着你进一步学习并且创建更大的项目，可以再回到这一列表。

- 在传送文件之前，在你的计算机上测试以确保链接有效，内容反映你的视觉设计。在传送页面到 Web 服务器或者可移动设备之后，再次测试。
- 在可能的情况下用尽量多的浏览器进行这些测试——Chrome、Firefox、Internet Explorer、Opera 和 Safari——并在 Mac 和 Windows 平台上都进行测试。如果可能，以低分辨率（800 像素×600 像素）和高分辨率（1600 像素×1200 像素）分别测试。
- 在开始测试之前关闭 Web 浏览器的自动图像装入，这样你可以看到每个页面没有图像时的样子。检查 alt 标签文本然后开启图像装入，装入图形并且重新小心查看页面。
- 使用浏览器字体大小设置，以各种字体大小来查看每个页面，确保如果用户用自己的字体规格覆盖你的，你的布局不会破坏。
- 等待每个页面完全结束装入，然后滚动到底确定所有图像出现在应该出现的位置。
- 观察每个页面装入花费的时间。装入是否花费超过几秒钟？如果这样，这个页面上的信息是否宝贵到足以阻止用户在页面装入结束前转向别处？诚然，宽带连接很普遍，但是这并不意味着你可以在你的页面上载入 1MB 大小的图像。

如果你的页面通过了所有这些测试，你可以轻松一下了，你的网站已经为公众的查看做好了准备。

## 2.6 总结

你以创建一个非常简单的 HTML 文件开始了本章，将其用作传送文件到 Web 服务器的过程的测试文件。你学习了文件传送过程的工作方式以及进行这些传送所需要的软件类型（FTP 客户端）。你还学习了一些关于 Web 服务器目录结构和文件管理，以及给定的 Web 服务器目录中 index.html 文件的重要作用的知识。你还学到，可以在可移动媒体上分发内容，以及着手构造文件和目录，以达到不使用远程 Web 服务器查看内容的目标的方法。最后，你学习到了在向公众发布你的网站之前，测试文件的方法。

## 2.7 问与答

**问：**你建议的所有测试将比创建我的页面花的时间还多！我不能测试得少一些吗？

**答：**如果你的页面不是用来赚钱或者提供重要的服务，那么对一些用户来说它们看上去很滑稽，或者偶尔出点错误可能不是大事。在那种情况下，只要用两种不同的浏览器测试各个页面就可以收工了。但是，如果你需要建立一种专业的形象，严密的测试是不可替代的。

**问：**说真的，谁关心我如何组织 Web 内容呢？

**答：**不管你信不信，Web 内容的组织对搜索引擎和网站的潜在访问者是很重要的——你将在第 24 章中学到更多这方面的知识。但是总的来说，拥有一个有组织的 Web 服务器目录结构将帮助你跟踪可能经常更新的内容。例如，如果你有一个图像或者多媒体的专用目录，



你将精确地知道到哪里查找希望更新的文件——不需要找遍包含其他内容的目录。

## 2.8 作业

作业包括测验和帮助你巩固对包含的素材理解的联系。在看后面的答案之前，尝试回答所有问题。

### 2.8.1 测验

1. 通过 FTP 连接到你的 Web 服务器，需要哪三部分信息？
2. `index.html` 文件的用途是什么？
3. 你的网站是否必须包含一个目录结构？

### 2.8.2 测验答案

1. 主机名，你的账户用户名，和你的账户密码。
2. `index.html` 文件一般是 Web 服务器中一个目录的默认文件。它允许用户访问 `http://www.yourdomain.com/somedirectory/` 而不需要使用一个文件名后缀并得到正确的位置。
3. 不。使用目录结构组织文件完全取决于你，但是强烈建议使用，因为它简化了内容维护。

## 2.9 练习

- 使用你的 FTP 客户端，在你的网站的文档根目录中创建一个子目录。将 `sample.html` 文件的内容粘贴到另一个名为 `index.html` 的文件中，将 `<title>` 和 `</title>` 标签之间的文本改成新的内容，并将 `<h1>` 和 `</h1>` 标签之间的文本改成新的内容。保存文件并将其上传到新的子目录。使用你的浏览器导航到 Web 服务器上的新目录查看出现的 `index.html` 文件内容。然后，使用你的 FTP 客户端，删除远程子目录的 `index.html`。用你的浏览器返回那个 URL，重新装入页面，看看没有 `index.html` 的情况下服务器的响应。
- 使用上面的练习中创建的不同文件组，将这些文件放到可移动媒体设备——例如一个 CD-ROM 或者一个 U 盘。使用你的浏览器导航你的样板网站的这个本地版本，并且思考一下，你必须与这个可移动媒体一起分发的说明，以使其他人能够使用它。

## 第 3 章

# 理解 HTML 和 XHTML 的关系

---

本章中你将学到：

- 以 HTML 创建一个简单网页的方法
- 包含每个网页必须有的所有 HTML 标签的方法
- 用段落和换行组织页面的方法
- 用标题组织内容的方法
- HTML、XML、XHTML 和 HTML5 之间的差别

前两章给了你创建 Web 内容以及在线或者本地查看它（如果你还没有一个 Web 主机提供商）背后的过程的基本概念。在本章中，我们将开始认真地说明一个 HTML 文件中必须出现的各种元素。

到本章结束，你将学习 HTML 与 XHTML 之间的差别，以及为同一件事情（创建 Web 内容）而设计两种不同的语言的原因。总体上，本章提供 HTML 和 XHTML 基础知识的一个快速总结并且为你作为页面创作者和发布者提供一些实用的技巧。但是，这不都是理论，你将实实在在地看到真正的网页和背后的 HTML。

这是你应准备好使用本书其余部分之前所需要的检查。

1. 有一台计算机。我使用一台带有 Windows Vista 的计算机来测试样板 Web 内容并且捕捉本书的图像，但是你可以使用任何 Windows、Macintosh 或者 Linux/Unix 机器来创建和查看你的 Web 内容。

2. 获得到互联网的连接。不管你拥有的是拨号、无线或者宽带连接，对创建和查看你的 Web 内容来说都不重要，但是连接越快，总体体验就越好。互联网服务提供商（ISP）、学校或者提供给你互联网连接的公司能够帮助你正确地建立它。另外，许多公共空间，如咖啡店、书店和图书馆提供免费的无线互联网服务，如果你有具备 Wi-Fi 网络支持的笔记本电脑，就可以使用它。

3. 得到 Web 浏览器软件。这是你的计算机读取和显示 Web 内容所需要的软件。正如你在第 1 章学到的, 最流行的浏览器软件(以字母排序)是 Apple Safari、Google Chrome、Mozilla Firefox、Microsoft Internet Explorer 和 Opera。安装多种浏览器以便试验并且确定你的内容看起来一致是个好主意, 你不能假设其他人使用的浏览器。

### By the Way

**注意:** 尽管所有浏览器以相同的通用方式处理信息, 但是它们之间有些特殊的差别, 导致不同的浏览器所看到的网页不总是相同的。一定要在多个浏览器中检查你的网页以确保它们的外观有合理的一致性。

4. 探索! 使用浏览器在互联网上到处寻找内容或者外观与你希望创建的相似的网站。注意一些页面让你失望的地方, 吸引你和以及使你离开的, 以及使你一次又一次回到某些网页的原因。如果有些特殊的主题使你感兴趣, 考虑使用流行的搜索引擎如 Google (<http://www.google.com/>) 或者 Bing (<http://www.bing.com/>) 搜索它。

### By the Way

**注意:** 正如第 1 章所讨论的, 如果你计划将你的 Web 内容放在互联网上(与在 CD-ROM 或者本地 Intranet 上发布相反), 你将需要将其传送到一天 24 小时都连接到互联网的计算机上。互联网访问的公司或者学校也可能提供 Web 空间。如果没有, 你可能需要向一个 Web 托管提供商购买服务。

## 3.1 从一个简单的网页开始

在第一章中, 你学习到了“网页”只是一个由 HTML 代码“标示”(或者包围)的文本文件, 这些代码告诉浏览器显示文本的方式。为了创建这些文本文件, 使用像记事本(在 Windows 上)或者 TextEdit(在 Mac 上)这样的文本编辑器——不要使用写字板、Microsoft Word 或者其他全功能的字处理软件, 因为那些软件创建与用于 Web 内容的普通文本文件不同类的文件。

在你开始工作之前, 你应该从希望放置在网页上的一些文本开始。

1. 寻找(或者编写)一些关于你自己、你的家庭、你的公司、你的棒球队或者感兴趣的其他主题的文本段落。

2. 将这些文件保存为普通的标准 ASCII 文本。记事本和大部分简单的文本编辑器始终保存纯文本文件, 但是如果你使用其他程序, 你可能需要选择文件类型(选择文件→另存为命令)。

随着本章的进行, 你将添加 HTML 标识(称之为标签)到文本文件, 从而使其成为 Web 内容。

### Watch Out!

**警告:** 不要用 Microsoft Word 或者任何其他兼容 HTML 的字处理软件创建你的第一个 HTML 文件, 大部分这种程序试图以奇怪的方式重写 HTML, 可能使你完全迷惑。另外, 我建议你不要使用图形化的所见即所得(WYSIWYG)编辑器, 如 Microsoft FrontPage 或者 Adobe Dreamweaver。

你可能会发现，在学习 HTML 时使用一个简单的文本编辑器开始会更容易并且更有教益。你可以在更好地理解了基本原理之后，再转向可视化工具（如 FrontPage 和 Dreamweaver）。

当你保存了包含 HTML 标签的文件，始终取一个以.html 结尾的名称，这是很重要的。如果你在保存时忘记在文件名后输入.html，大部分文本编辑器将给它其他的扩展名（如.txt）。如果这样，你在使用 Web 浏览器查看文件时就可能找不到它。如果你发现了它，它肯定也不能正常显示。换句话说，Web 浏览器预期网页文件有一个扩展名.html。

**注意：**如果你使用 Macintosh 计算机上的 TextEdit，创建一个 HTML 文件的步骤和使用 Windows 上的记事本有所不同——它们都是流行的文本编辑器，但是后者你必须首先单击 Format（格式）菜单并且选择 Make Plain Text（建立纯文本），然后在 Saving Header（保存文件头信息）下不选择 Append '.txt' Extension to Plain Text Files（为纯文本文件附加.txt 扩展名）复选框。而且，默认首选项被设置为将 html 文件显示为浏览器中的样子，这不允许你编辑。为了修复这个错误，在 Rich Text Processing header（富文本处理文件头）下选中 Ignore Rich Text（忽略富文本）命令。

**By the Way**

你可能还会遇到带有文件扩展名.htm 的网页，这也是可以接受的。你可能发现在 Web 上的其他文件扩展名，如.jsp（Java 服务器页面）、.asp（Microsoft 活动服务器页面）或者.php（PHP：超文本预处理器），但是这些文件类型使用超出 HTML 范围的服务器端技术。

程序清单 3.1 展示了一个你可以输入并且保存以创建简单的网页的文本示例。如果你用 Firefox 打开这个文件，你将看到如图 3.1 所示的页面。你创建的每个网页必须包含<html></html>，<head></head>，<title></title>及<body></body>标签对。

### 程序清单 3.1 <html>，<head>，<title>和<body>标签

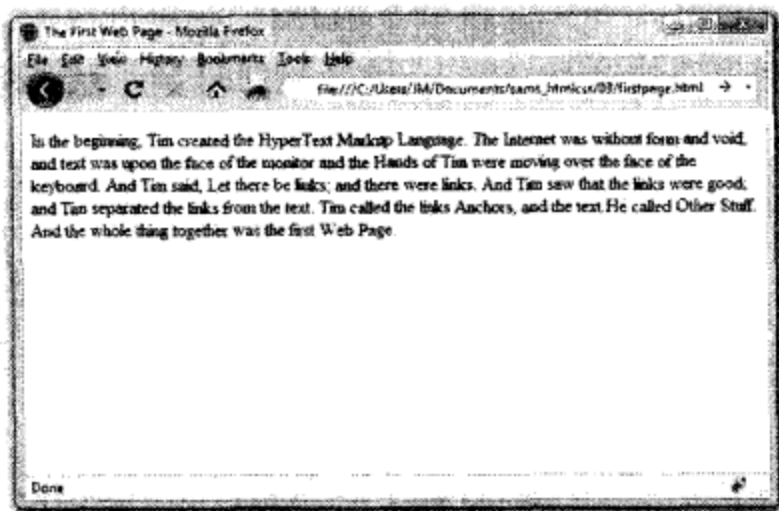
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>The First Web Page</title>
  </head>

  <body>
    <p>
      In the beginning, Tim created the HyperText Markup Language. The Internet
      was without form and void, and text was upon the face of the monitor and
      the Hands of Tim were moving over the face of the keyboard. And Tim said,
      Let there be links; and there were links. And Tim saw that the links were
      good; and Tim separated the links from the text. Tim called the links
      Anchors, and the text He called Other Stuff. And the whole thing together
      was the first Web Page.
    </p>
  </body>
</html>
```

图 3.1

当你保存程序清单 3.1 的文本为一个 HTML 文件并且用 web 浏览器查看它，仅显示标题和正文



在程序清单 3.1 中，和每个 HTML 页面一样，以“<”开始，以“>”结尾的单词实际上是命令编码。这些命令编码被称为 HTML 标签，是因为它们“标记”了文本部分并且告诉 Web 浏览器这些文本的类型。这使浏览器正确地显示这些文本。

网页的头几行代码作为标准的样板编码，将会包含在所有你的网页中。这些代码实际上标识页面为一个 XHTML1.1 文档，这意味着技术上这个网页是 XHTML1.1 页面。在本书开发的所有页面都是 XHTML1.1 页面。因为 XHTML 是更加结构化的 HTML 版本，所以一般来说称本书中的所有页面为 HTML 页面也是对的。通过在你的代码中针对 XHTML1.1，你将开发坚持最新 Web 标准的网页。这是件好事！

### By the Way

**注意：**从技术上说，HTML5 将是下一个 Web 标准，但是它还没有被完全采用。按照当前的估计，HTML 的完全采用将在 2011 年中。但是，随着你在本书中学到 HTML 和 XHTML 的重要特性，我将包含关于 HTML5 特性可能差异的注释。

### TRY IT YOURSELF ▼

**自己动手尝试一下：创建并且查看一个基本的网页**

在你学习程序清单 3.1 中使用的 HTML 标签的含义之前，你可能想要了解我究竟是如何创建和查看文本本身的。按照如下步骤。

1. 在 Windows 记事本中输入程序清单 3.1 中的所有文本，包括 HTML 标签（或者使用 Macintosh TextEdit 或者其他你选择的文本编辑器）。
2. 选择 File（文件）→ Save As（另存为）命令。一定要选择纯文本（或者 ASCII 文本）文件类型。
3. 命名文件为 firstpage.html。
4. 在你的硬盘上选择希望保存网页的文件夹——并且记住你选择的文件夹！单击 Save（保存）或者 OK 按钮来保存文件。
5. 现在启动你所喜欢的 Web 浏览器。（保持记事本运行，这样你可以简单地在查看和编辑页面之间切换）。

在 Internet Explorer 中，选择 File（文件）→ Open（打开）并且单击 Browse（浏览）。如果你使用 Firefox，选择 File（文件）→ Open File（打开文件）。找到正确的文件夹并且选择 firstpage.html 文件。有些浏览器和操作系统还允许你拖放 firstpage.html 文件到浏览器窗口以便查看。

瞧！你应该看到图 3.1 中所示的页面。



如果你已经得到了一个 Web 主机账户，你可以在这时使用 FTP 将 `firstpage.html` 文件发送到 Web 服务器。实际上，从本章以后，本教程中将假定你有了一个主机提供商并且能够轻松地通过 FTP 传递文件。如果情况不是这样，请在继续之前复习前两章。或者，如果你有意识地在本地使用文件（没有 Web 托管），准备好调整操作方法来适应你的特殊情况（比如省略“传送文件”和“输入 URL”）。

**注意：**你不需要连接到互联网来查看自己的计算机上保存的网页。默认的情况下你的浏览器在每次启动时试图连接到互联网，这在大部分时候有意义。如果你在硬盘上（离线）本地开发页面，这将是麻烦，你会不断得到关于页面没有找到的错误。如果你有一个通过 LAN、电缆 Modem 或者 DSL 的全时 Web 连接，这是没有意义的，因为浏览器将永远不会抱怨离线。否则，相应的惩罚性操作将取决于你的浏览器种类。检查浏览器“Tools”（工具）菜单下的选项。

**By the Way**

## 3.2 每个 XHTML 网页必须有的 HTML 标签

向你展现 HTML 标签的秘密语言的时候到了。当你理解了 this 预言，你将拥有比其他人多得多的创造力。不要告诉其他人，但是这其实很简单。

在你进入 HTML 标签之前，让我们首先来解决程序清单 3.1 开头的杂乱代码。第一行指出这个 HTML 文档实际上是个 XML 文档：

```
<?xml version="1.0" encoding="UTF-8"?>
```

XML 的版本被设置为 1.0，这是相当标准的，字符编码的类型（UTF-8）也一样。

程序清单 3.1 的第二和第三行代码看上去更复杂：

```
<!DOCTYPE html PUBLIC "-//W3C/DTD XHTML 1.1//EN"  
http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd>
```

**注意：**这时，像字符编码这样的概念的理解还不是非常重要。重要的是你在页面中包含了相应的样板代码，这样页面将坚持最新的 Web 标准。在本书写作的时候，XHTML 1.1 是 Web 标准。HTML 5 还不是 Web 标准，但如果你创建了 HTML 5 文档，在你的 HTML 文件开始的这些行将不必要。

**By the Way**

同样，只要你记住在页面开始包含这些代码，这些代码的细节也不重要。这些代码标识该文档为 XHTML 1.1，然后允许 Web 浏览器确保这些代码符合所有 XHTML 的需求。

**注意：**XML/XHTML 样板代码对于创建网页不是严格必需的。你可以删除示例中开始的几行而从 `<html>` 标签开始，它仍然能在 Web 浏览器中正常打开。额外的代码用于确保你的页面符合最新的 Web 标准。此外，额外的代码使你能验证你的网页的准确性，你将在本课稍后学习到一些这种验证的做法。

**By the Way**

大部分 HTML 标签有两个部分：一个开始（opening）标签，指出文本块的开始，和一个结束（closing）标签，指出文本块的结束。结束标签以一个紧接着“<”符号的“/”（前向斜杠）开始。另一种类型的标签是空标签，只要它不包含一对相互匹配的开始和结束标签。

作为替代，空标签包含一个以“<”开始，以“>”符号之前加上一个“/”结束。下面是这3种标签的简短总结，只是为了确保你理解每种标签的角色。

- 开始标签是指出 HTML 命令开始的一个 HTML 标签，命令影响的文本出现在开始标签之后。开始标签始终以<开始，以>结束，如<html>。
- 结束标签指出了 HTML 命令的结束，命令影响的文本出现在开始标签之前。结束标签始终以</开始，以>结束，如</html>。
- 空标签是一个发出没有包含页面中的任何文本的 HTML 命令的标签。空标签始终以<开始，以/>结束，如<br/>和<img>。

例如，程序清单 3.1 中的<body>标签告诉 Web 浏览器这是页面的实际正文的开始，</body>指出文本的结束。<body>和</body>标签之间的所有东西将出现在 Web 浏览器窗口的主显示区域中，如图 3.1 所示。

浏览器窗口的最上方（参见图 3.1）显示了标题文本，这是位于<title>和</title>之间的所有文本。标题文本也用于标识在浏览器的书签或者收藏夹菜单上的页面，这取决于你使用的浏览器。为你的页面提供标题是很重要的，这样页面的访问者能够正常地将它们作为将来引用的书签。

你将在创建的每个 HTML 页面中使用<body>和<title>标签对，因为每个网页都需要一个标题和正文。你还将使用<html>和<head>，这是在程序清单 3.1 中看到的另两个标签。将<html>放在文档的最前面表示文档是一个网页。</html>放在最后表示网页结束。

### By the Way

**注意：**你无疑会注意到程序清单 3.1 中有些与<html>标签相关的代码。这些代码由两个属性（xmlns 和 xml:lang）组成，这些属性用于指定与标签相关的附加信息。这两个属性是所有 XHTML 网页的标准要求，前者定义 XML 命名空间，而后者定义内容的语言。在整本书中定义了一个标准命名空间，使用英语。如果你以不同语言编写，用相关的语言标识符替换“en”（英语）。

在页面中有一个标题部分和正文部分，分别由标签<head>和<body>标识。页面标题部分中的消息对页面进行描述，但并不显示在网页浏览器中。放在正文中的信息才被网页浏览器显示。<head>标签的位置总是靠近页面的 HTML 代码的开头，在<html>标签之后。

用于指定页面标题的<title>标签显示在页面的标题部分，也就是说它位于开始标签<head>之后、结束标签</head>之前（接下来将介绍其他一些可放在<head>和</head>之间的高级标题信息，如用于格式化网页的样式单（style sheet）规则）。

### Did you Know?

**提示：**创建并保存与程序清单 3.1 所示文档类似，但只包含<html>、<head>、<title>和<body>的框架页面很方便。这样，需要创建新网页时，可以打开该文档开始编辑，从而省去每次都要输入必需标签的麻烦。

程序清单 3.1 中使用的<p>标签包含一段文本。你应该尽可能将你的文本块包含在合适的容器标签中。

## 3.3 使用分段和换行来组织页面

网页浏览器显示 HTML 页面时，并不会考虑行尾和文本间的空格。例如，如图 3.2 所示，

在页面开头的诗中，每个单词间都只有一个空格，而不管它在程序清单 3.2 是如何输入的。这是因为在 HTML 代码中多余的空格都自动地减为一个。另外，当文本到了浏览器窗口的边缘时，它自动地滚动到下一行，而不管原来的 HTML 文件中的换行在哪里。

### 程序清单 3.2 包含段落和换行的 HTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>The Advertising Agency Song</title>
  </head>

  <body>
    <p>
      When your client's hopping mad,
      put his picture in the ad.

      If he still should prove refractory,
      add a picture of his factory.
    </p>

    <hr />

    <p>
      When your client's hopping mad,<br />
      put his picture in the ad.
    </p>
    <p>
      If he still should prove refractory,<br />
      add a picture of his factory.
    </p>
  </body>
</html>
```

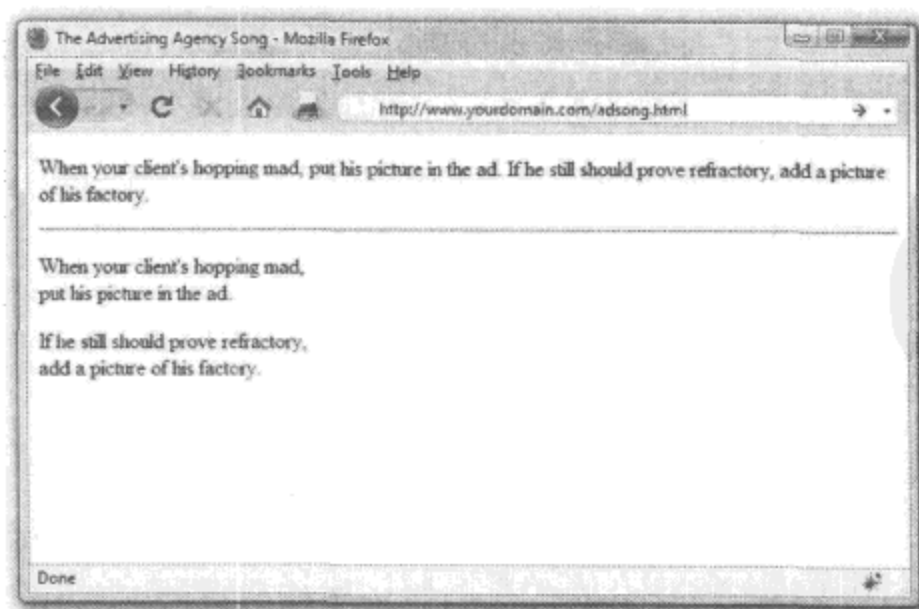


图 3.2

浏览 HTML 代码如程序清单 3.2 所示的页面时，只有包含 `<br/>` 和 `<p>` 标签的地方才出现分行和分段

如果你希望控制分段和换行实际出现的地方，必须使用 HTML 标签。当文本包含在 `<p></p>` 容器标签中时，结束标签之后将有一个虚构的换行。在后面的章节中，你将学习使用 CSS 控制换行的高度。`<br />` 标签在段落中强制一个换行。和你目前看到的标签不同，

`<br />` 标签不需要结束标签 `</br>`——这是前面讨论过的空标签的一个示例。尽管 HTML 4 在空标签中不要求 `/`，但是 XHTML 要求，未来的标准也是如此，所以坚持最新的标准并且创建正确编码的网页是很重要的。始终使空标签以 `/>` 结束。

程序清单 3.2 和图 3.2 中的诗显示了标签 `<br />` 和 `<p>` 用于分开押韵的广告代理歌的行和段。你可能还注意到了程序清单中的 `<hr />` 标签，它使页面包含一条水平的标尺线（见图 2.2）。使用 `<hr />` 标签插入水平标尺线也会引起分行，即使并没有使用 `<br />` 标签。要使水平标尺线上下有更多的空间，可以在 `<hr />` 标签前后分别添加标签 `<p>` 和 `</p>`。这相当于将水平标尺线单独放在一个段落中。与 `<br />` 一样，水平标尺线标签 `<hr />` 也是一个空标签，因此不需要结束标签 `</hr>`。

### Watch Out!

**警告：**你可能发现许多 Web 内容包含 `<br>` 而不是 `<br />`。你还可能发现不包含 `</p>` 结束标签。只要记住互联网上还有许多陈旧的 Web 内容，使用中的并不意味着是正确的。坚持本书中学到的标准，可以免去将来许多的工作和失望。养成清晰的 HTML 编码习惯是成为成功的 Web 设计人员非常重要的一步。

### TRY IT YOURSELF ▼

**自己动手尝试一下：将文本格式化为适当的 HTML**

1. 将 `<html><head><title>My Title</title></head><body>` 添加到文本的开头（使用自己的网页标题代替“My Title”。同时在页面标题中包含满足 XHTML 要求的代码示例）。
2. 在文本的末尾添加 `</body></html>`。
3. 在每段的开头添加 `<p>` 标签，结尾添加 `</p>` 标签。
4. 在需要分行的地方使用 `<br />` 标签。
5. 使用 `<hr />` 标签绘制水平标尺线，将文本的主要部分分开，或在页面需要线条的地方使用。
6. 将文件保存为 `mypage.html`（使用自己的文件名代替“mypage”）。
7. 在 Web 浏览器中打开文件来浏览网页。
8. 如果什么地方出错，回到文本编辑器中进行修改，然后再次保存文件。单击浏览器中的“刷新”可以看到对网页的修改结果。

### By the Way

**注意：**如果使用字处理软件来创建网页，要确保将 HTML 文件保存为纯文本格式或 ASCII 格式。

## 3.4 用标题组织内容

在 Internet 上浏览网页时，将注意到许多网页开头都有一个字体比其他文本更大更粗的标题。程序清单 3.3 是一个简单的示例，这个网页包含了一个标题以及一些正常的段落文本作为比较。在标签 `<h1>` 和 `</h1>` 之间的文本将作为大标题文本显示。另外，`<h2>` 和 `<h3>` 表示字体较小的标题，依此类推，直到 `<h6>`。

### 程序清单 3.3 标题标签

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>My Widgets</title>
  </head>

  <body>
    <h1>My Widgets</h1>
    <p>My widgets are the best in the land. Continue reading to
    learn more about my widgets.</p>

    <h2>Widget Features</h2>
    <p>If I had any features to discuss, you can bet I'd do
    it here.</p>

    <h3>Pricing</h3>
    <p>Here, I would talk about my widget pricing.</p>

    <h3>Comparisons</h3>
    <p>Here, I would talk about how my widgets compare to my
    competitor's widgets.</p>

  </body>
</html>
```

**注意：**现在你可能注意到了，HTML 代码被缩进以表示 HTML 文档中不同部分之间的关系。这种缩进完全是可选的，可以将所有标签都放在一起，之间没有空格或分行，网页在浏览器中仍可正常显示。这里的缩进旨在让你很快理解充满代码的网页以及它们之间的关系。代码缩进是一种良好的 Web 开发习惯，最终使网页更易于维护。

**By the  
Way**

如图 3.3 所示，创建标题的 HTML 简单得不能再简单。在这个示例中，“My Widgets”这个词使用<h1>标签突显在页面中。要创建最大的一级标题，只要在文本前面使用<h1>标签，在文本后使用</h1>标签。至于较小的二级标题（用于重要性较低的标题），使用<h2>和</h2>。对于应该比二级标题更不突出的内容，使用<h3>和</h3>。你的标题应该遵循内容层次结构，仅使用一个一级标题，一级标题以后有一个或者多个二级标题，在二级标题后紧接着使用三级标题，依此类推。

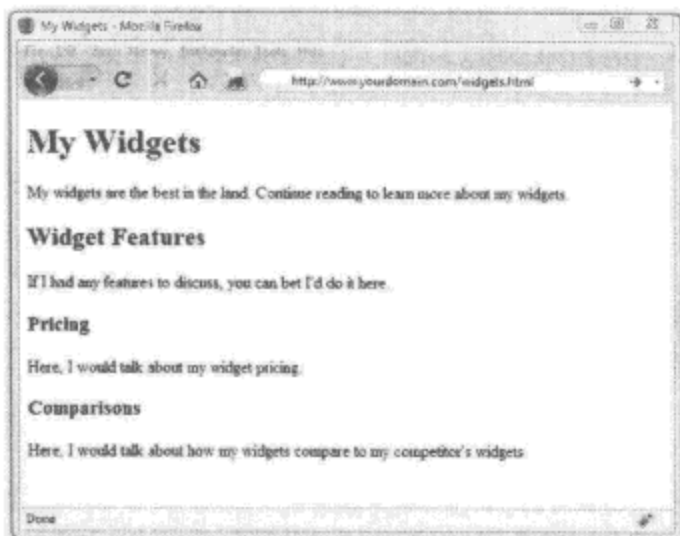


图 3.3

三级标题的使用展示了简单的产品页面上的内容层次结构



理论上,你还可以使用<h4>、<h5>和<h6>来建立更不重要的标题,但是不常这么做。Web 浏览器很少显示出这些标题与<h3>的明显不同,内容一般不会需要 6 个级别的标题来显示其层次结构。

### By the Way

**注意:** 在现在的许多网页中,通常用包含艺术字体和图标的图形图像来取代本章讨论的普通标题文本。但是,使用文本标题是在第 24 章中讨论的许多搜索引擎优化(SEO)技巧之一。搜索引擎查看标题标签来了解内容的组织,它们给予你指定的较重要的内容(例如,一级标题)比较不重要的内容(低级别的标题)更高的优先级。

要记住 title 与 heading 之间的区别。这两个单词在日常的英语中是可以通用的,在 HTML 中,<title>指定整个网页的标题,但并不显示在网页上,而是显示在浏览器窗口的标题栏中。而 heading 标签使一些文本突显在网页上。一个网页只有一个<title>,且只能出现在<head>和</head>标签之间,而<h1>、<h2>和<h3>可以有任何多个,顺序也可以根据需要任意安排。但是,正如我前面提到的,你应该使用标题标签来保持对内容层次结构的紧密控制,不要使用标题作为完成特殊“外观”的方法,那是 CSS 的目的。

你将在本书的第二部分学习完全控制网页上文本外观的方法。标题提供了最简单和最流行的将注意力吸引到重要文本的方法,但是缺乏对大小、字体和颜色的精确控制。

## 学习他人的网页

看到许多流行的网页所呈现的视觉和声音效果,你可能认识到这一章介绍的简单网页只是 HTML 的冰山一角。由于你已经掌握了基础知识,看看 Internet 上别人的网页可能让你有意想不到的收获。只要在网页浏览器中右击然后选择“查看源文件”,就能看到任何网页的 HTML 源代码。

如果还看不懂有些 HTML 标签或不知道如何使用,不要担心。在下面几章将学到这些东西。然而,现在提前看看这些标签,将能够知道已掌握的知识,了解将来能够在网页中实现的功能。

## 3.5 验证你的 Web 内容

在第 2 章中,我讨论了测试你的网页的方法,一个非常重要的测试页面方法是验证。可以这样考虑:设计和绘制一组漂亮的房屋计划是一回事,建筑师将其变成一所适合建设的安全的结构完全是另一回事。验证网页是相似的过程,但在这种情况下,建筑师是一个应用程序——而不是一个人。

简而言之,验证是用特殊的能够查找错误并且确定你的页面遵循严格的 XHTML 标准的应用程序来测试你的页面的过程。验证很简单。实际上,负责开发 Web 标准的组织——万维网联盟(W3C)提供了可用的一个在线验证工具。为了验证一个页面,转到这个 URL:<http://validator.w3.org/>。W3C 标示验证服务如图 3.4 所示。

如果已经在网上发布网页,可以使用 Validate by URL 选项卡进行验证。使用 Validate by File Upload 选项卡验证保存在你本地计算机文件系统上的文件。Validate by Direct Input 选项卡让你

粘贴文本编辑器中的文件内容。如果一切正常，你的页面将得到一个通过报告，如图 3.5 所示。

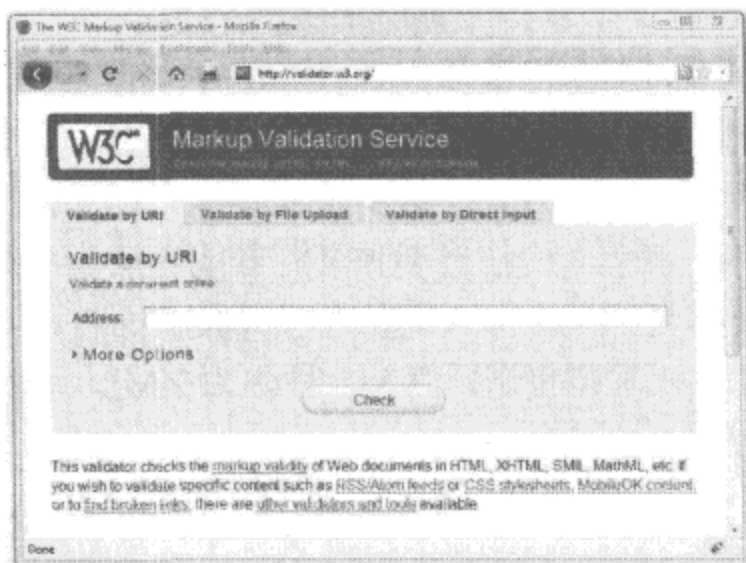


图 3.4

W3C 的标记验证服务让你能够验证 HTML (或 XHTML) 文档，保证文档编码正确

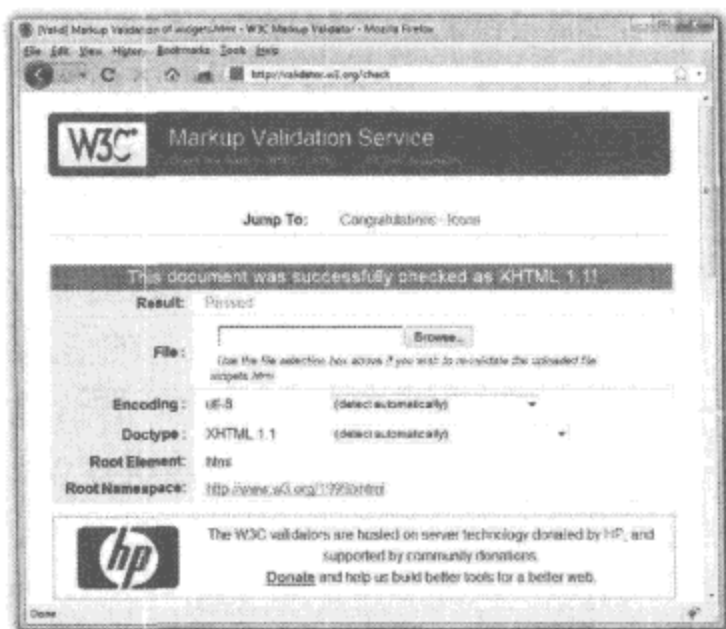


图 3.5

如果网页通过了 W3C 的标记验证服务，就表明这个网页已经准备好了

如果 W3C 的标记验证服务在网页上发现错误，它将提供一些具体的细节，包括出错代码的行号。这不失为跟踪问题、调试网页的好办法。因此，文档验证不仅让你能够知道网页是否正确，还能帮助你在网页发布之前发现问题、解决问题。

**提示:** 有些 Web 开发工具包含内置的验证功能，可以代替 W3C 的标记验证。实例包括浏览器扩展如 Firebug (<http://getfirebug.com/>) 和 HTML Validator (<http://users.skynet.be/mgueury/mozilla/>)，但是许多其他程序提供相似的功能——查看你的用户文档。

**Did you Know?**

## 3.6 关于 HTML、XML、XHTML 和 HTML 5 的最新消息

早期，HTML 是很了不起的，因为它使科学家能够通过互联网，以高效和相对结构化的风格共享数据。不久就出现了图形化的 Web 浏览器，HTML 开始用于科学文章之外的内容。HTML 很快从一种用于研究人员的简洁的小型语言变成一种在线出版语言。在 HTML 能够活跃于图形化浏览器之后，Web 浏览器的创造者疯狂地为这种语言添加时髦的特性。尽管这些新特性开始时很简练，但是它们损害了 HTML 的原始设计并且在浏览器显示网页的方法上引入了不一致，新特性仅仅能在一两种浏览器上正常工作，如果你恰好运行的是错误的浏览器，就没有那么幸运了。HTML 开始变得像一个拙劣的重新模型化的工作——由太多承包人完成

的没有正确计划的工作。结果是，这段时间内创建的一些针对具体浏览器的特性现在被作为标准采用，而其他的完全被抛弃了。

和大部分变革一样，Web 的诞生是非常混乱的，对 HTML 的修改反映了这种乱象。经过多年，为了解决 HTML 的不一致性并且恢复该语言的秩序，人们付出了重大的努力。HTML 的混乱问题导致浏览器必须猜测页面的显示方式，这不是件好事。理想情况下，网页设计人员应该能够精确定义页面的外观并且不管人们使用哪种浏览器或者操作系统，都能有相同的外观。更好的是，设计人员应该能够精确地定义页面的含义并使页面在不同的浏览器和平台上看上去一致。这种理想的实现仍然要等待将来，但是一种称为 XML（可扩展标示语言）的标识语言开始承担了带领我们前进的重要角色。

XML 是一种用于创建具体语言（如 HTML）的通用语言。这听上去有些奇怪，真正的含义是 XML 提供了任何标示语言必须坚持的基本结构和规则。使用 XML，你可以创建一种独特的描述任何类型信息（包括网页）的标示语言。了解 XML 是一种创建其他标识语言的语言，你就能够用 XML 创建自己的 HTML 版本。你甚至可以创建被称为 BCCML（瓶盖收藏标记语言）的标示语言，你可以用它来创建和管理你的大量稀有瓶盖的收藏。重点是 XML 为以一致的风格组织信息打下了基础，而信息可以是来自网页到瓶盖的任何东西。

你可能会想，瓶盖和 Web 毫不相干，为什么提到它？原因是 XML 不是完全关于网页的。XML 的用途不仅限于 Web，它可以在任何计算机上表现任何类型的信息。如果你能想象全球的计算机、移动电话、手持计算机、电视和广播所快速处理的所有信息，就能理解 XML 的用途比仅仅清理网页要广阔得多。但是，XML 的第一个应用是恢复 Web 的一些秩序，这就是 XML 与学习 HTML 相关的原因。

如果 XML 比 HTML 能更好地描述数据，是不是意味着 XML 可以取代 HTML 作为 Web 标识语言的选择？不。XML 不是 HTML 的替代品，它甚至不是 HTML 的竞争对手。XML 对 HTML 的影响是清理 HTML。HTML 是一种得益于 XML 规则的相对无结构的语言。两种技术的自然结合使得 HTML 坚持 XML 的规则和结构。为了完成这种结合，建立了一种遵循更严格的 XML 规则的 HTML 新版本。这种新的 XML 兼容版本的 HTML 被称为 XHTML。很幸运的是，你通过本书实际上学习的是 XHTML，它是更清晰的 HTML 版本。

你可能听说过 HTML 5，这被吹捧为下一个 Web 标准。它将会是，但是在几年内还不会成为标准。当它成为 Web 标准，不会使 XHTML 失去意义——HTML 5 不是 XHTML 的替代品，而是 HTML 4 的重要修订。换句话说，XHTML 和 HTML 5 将并存于 Web，目前支持 XHTML 的浏览器有一天将也会支持 HTML 5。

本书的目标是引导你使用 XHTML 和 CSS 作为页面的核心语言，了解 web 出版的基础知识。但是，只要有可能，我将说明在 HTML 5 中不存在的语言元素，你应该希望设计更具持续性的内容。如果你对 Web 出版和 CSS 与页面的总体标示语言（XHTML 或者 HTML 5）有了坚实的理解，几年以后，再决定从 XHTML 转移到 HTML 5 时你将处于主动的位置。

## 3.7 总结

本章介绍了网页的基础概念和工作方式。你学习到 HTML 命令编码包含在一个文本文件中，自己输入 HTML 文本要好于使用图形化编辑器创建 HTML 命令——特别是当你学习

HTML 时。我们介绍了最基本和重要的 HTML 标签。通过添加这些命令编码到任何纯文本文档，你可以很快地将它们转换为真正的网页。创建网页的第一步是在文档开头和结尾加入一些必要的标签，包括页面的标题。接下来可以根据需要标记行和段落结束的位置，或添加水平标尺线和文本标题。表 3.1 总结了本章介绍的所有标签。

表 3.1 第 2 章介绍的标签

标签	功能
<code>&lt;html&gt;...&lt;/html&gt;</code>	包含整个 HTML 文档
<code>&lt;head&gt;...&lt;/head&gt;</code>	包含 HTML 文档的标题
<code>&lt;title&gt;...&lt;/title&gt;</code>	指明文档的标题，用于<head>内
<code>&lt;body&gt;...&lt;/body&gt;</code>	包含 HTML 文档的正文
<code>&lt;p&gt;...&lt;/p&gt;</code>	包含一个段落；段落间隔一行
<code>&lt;br /&gt;</code>	分行
<code>&lt;hr /&gt;</code>	水平标尺线
<code>&lt;h1&gt;...&lt;/h1&gt;</code>	一级标题
<code>&lt;h2&gt;...&lt;/h2&gt;</code>	二级标题
<code>&lt;h3&gt;...&lt;/h3&gt;</code>	三级标题
<code>&lt;h4&gt;...&lt;/h4&gt;</code>	四级标题（很少使用）
<code>&lt;h5&gt;...&lt;/h5&gt;</code>	五级标题（很少使用）
<code>&lt;h6&gt;...&lt;/h6&gt;</code>	六级标题（很少使用）

最后，你学习了关于 XML 和 XHTML，以及它们与 HTML 的关系的知识，还有 HTML5 与你所学习的知识的关系。

## 3.8 问与答

问：我创建了一个网页，但在 Web 浏览器中打开它时，看到了所有文本，包括 HTML 标签。有时候甚至看到一些奇怪的字符出现在页面开头！什么地方做错了？

答：可能没有将文件保存为纯文本文件。试着重新保存文件，一定要将它保存为纯文本文件或 ASCII 文件。如果不太清楚如何使用字处理器将文件保存为纯文本文件，不要勉强。使用记事本或文本编辑器。另外，还要保证网页的文件名以 .html 或 .htm 为扩展名。

问：Internet 上的有些网页开头并没有使用<html>标签，而你说过网页总是以<html>开始。这是怎么回事？

答：如果网页中没有<html>标签，许多 Web 浏览器也会正常地显示。但在网页中包含这个标签是一个好习惯，因为有些软件需要这个标签来识别有效的 HTML 页面。另外，网页也应该是真正的 XHTML 页面才能符合最新的 Web 标准。

## 3.9 作业

作业包含一些测试题和练习，帮助巩固对这一章介绍的内容的理解。试着回答所有问题然后再看后面的“答案”小节。

### 3.9.1 测验

1. 所有 HTML 页面都必须包含哪 4 个标签?
2. 为生成如下 Web 内容, 你应该使用哪些 HTML 标记和文本?
  - 包含 “We are Proud to Present” 的小标题。
  - 横穿页面的水平标尺线。
  - 包含单词 “Orbit” 的大标题。
  - 包含 “The Geometric Juggler” 的中等标题。
  - 另一条水平标尺线。
3. 编写一个完整的网页, 标题为 “Foo Bar”, 页面开头有一个 “Happy Hour at the Foo Bar” 标题, 下面是正常字体的 “Come on down!”。

### 3.9.2 测验答案

1. `<html>`、`<head>`、`<title>`和`<body>` (及其对应的结束标签`</html>`、`</head>`、`</title>`和`</body>`)。

2.

```
<h3>We are Proud to Present</h3>
<hr />
<h1>Orbit</h1>
<h2>The Geometric Juggler</h2>
<hr />
```

3.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Foo Bar</title>
  </head>

  <body>
    <h1>Happy Hour at the Foo Bar</h1>
    <p>Come on Down!</p>
  </body>
</html>
```

### 3.10 练习

- 即使阅读本书旨在创建商业网页, 可能也希望创建个人网页作为练习。输入几段自我介绍, 并使用本章学到的 HTML 标签使它成为一个网页。
- 通过本书, 你将学习到代码示例并且建立自己的网页。现在花点时间建立一个包含 XML 声明, 文档类型声明以及用于核心 HTML 文档结构的标签的基本文档模板。这样, 你可以在任何需要的时候复制和粘贴这些信息。



## 第 4 章

# 理解层叠样式单

---

本章中你将学到：

- 创建基本样式单的方法
- 使用样式类的方法
- 使用样式 ID 的方法
- 构建内部样式单和嵌入样式的方法

前一章中，你学习了 HTML 和 XHTML 的基础知识，包括了建立用于所有 Web 内容的 HTML 框架模板的方法。本章中，你将学习使用层叠样式单（CSS）微调 Web 内容显示的方法。样式单背后的概念很简单：创建一个独立的样式单文档，以指定字体、颜色、背景和其他确定网站外观的特征。然后链接需要使用该样式单的所有网页，而不是在每个文档中都重复指定这些样式。因此，当你决定修改公司的正式字体或颜色主题时，只要修改一两个样式单而不需要修改所有静态 Web 文件，就可以一次性修改所有网页。也就是说，样式单是一组格式化指令，能够同时控制众多 HTML 页面的外观。

样式单能够设置大量的格式化特征，这包括精确的字体控制、字间距和行间距、页面边距和边框，这还只是少数几个例子。样式单还能够以人们熟悉的英寸、毫米、点和 picas 等来指定大小和其他度量值。你还可以以具体的坐标或者与其他项目的相对位置，使用样式单正确地控制图像和文本在网页上的位置。

简而言之，样式单为 Web 带来了高级的显示。而它们使用了样式来做到这一点。

**注意：**如果有 3 个或更多的网页使用类似的格式和字体，你可能想在阅读本章时为它们创建一个样式单。即使不想创建完整的样式单，将样式直接应用于网页中的 HTML 元素也很有帮助。

**By the  
Way**

PDG

## 4.1 CSS 工作原理

样式单背后的技术被称为 CSS，这是层叠样式单的缩写。CSS 是一种定义样式结构如字体、颜色、位置等的语言，被用于描述网页上的信息格式化和现实的方式。CSS 样式可以直接存储于 HTML 网页或者单独的样式单文件。无论哪一种方式，样式单包含将样式应用到指定类型的元素的规则。外部使用时，样式单规则被放置在一个带有文件扩展名.css 的外部样式单文档中。

样式规则是可应用于网页中元素，如文本段落或链接的格式化指令。样式规则由一个或多个样式属性及其值组成。内部样式单直接放在网页中，外部样式单保存在独立的文档中，网页通过一个特殊标签链接外部样式单——稍后将介绍这个标签。

名称 CSS 中的“层叠 (cascading)”表示样式单规则应用于 HTML 文档元素的方式。具体地说，CSS 样式单中的样式形成一个层次结构，更具体的样式覆盖通用样式。样式规则的优先级由 CSS 根据这个层次结构决定，从而实现级联效果。如果上述解释还不是很清楚，可将 CSS 的级联机制看成一种类似继承的关系，在这种继承关系中，父亲的特征传递给子女，但子女有更特殊的特征，基本样式规则适用于整个样式单，但可被更具体的样式规则覆盖。

用一个简单例子可能更能够解释清楚。看下面的代码，你是否能够说出文本的颜色：

```
<div style="color:green">
  This text is green.
  <p style="color:blue">This text is blue.</p>
  <p>This text is still green.</p>
</div>
```

在这个例子中，颜色 green 通过样式属性 color 应用于 <div> 标签。因此 <div> 标签中的文本是绿色的。由于 <p> 标签是 <div> 标签的子标签，因此绿色的文本样式传递给 <p> 中的文本。然而，第一个 <p> 标签覆盖了 color 样式，将它改为蓝色。最后的结果是，第一段文本是蓝色的，第二段文本仍为传递而来的绿色。

### By the Way

**注意：**你可能注意到了，我在本章中大量使用了术语“元素”（在本书其余部分当中也将如此使用）。元素就是网页中的信息（内容），如图像、段落或链接。标签用于对元素进行编码，可以将元素看成标签及其内部的描述信息（如属性、文本、图像等）。

与许多 Web 技术一样，CSS 已经发展很多年了。CSS 最初的版本称为 Cascading Style Sheets Level 1 (CSS1)，创建于 1996 年。后来的 CSS2 标准创建于 1998 年，虽然这似乎是很久以前了，但直到最近 CSS2 仍得到所有网页浏览器的支持。由于 CSS 的强大功能，浏览器的支持始终是一个问题。幸运的是，浏览器最后支持了大部分功能，现在你可以安全地使用 CSS2 样式单，不用担心太多事情。因此，当我在本书中谈到 CSS 时，指的是 CSS2。

在 <http://www.w3.org/Style/CSS/> 上可以找到 CSS 的完整参考。本章的余下部分说明 CSS 的使用方法。

## 4.2 一个基本的样式单

虽然样式单的功能很强大，但要创建它们也可以很简单。请看如图 4.1 和图 4.2 所示的

网页。这两个网页的一些视觉属性是相同的，可将其放在一个样式单中。

- 它们在标题上使用大的，加重的 Verdana 字体，正文使用普通大小的 Verdana 字体。
- 它们使用浮动于内容之中，位于页面右边的图像 logo.gif。
- 除了子标题为紫色之外，所有文本为黑色。
- 它们在左边和上边有边距。
- 在文本行之间有垂直的空白。
- 脚注居中，字体较小。



图 4.1

本页使用一个样式单来调整文本和图像的外观和间距

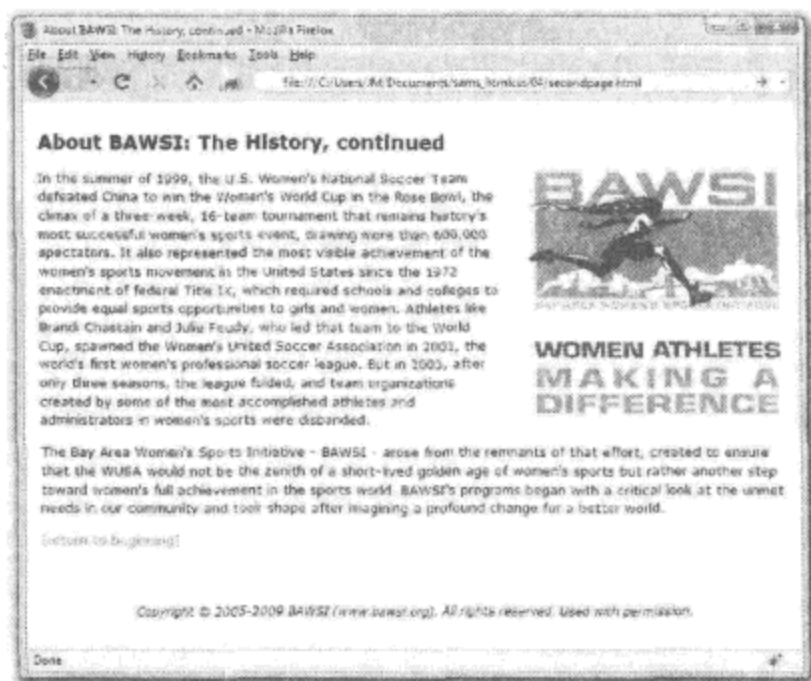


图 4.2

本页使用与图 4.1 中相同的样式单，所以保持一致的观感

程序清单 4.1 所示为指定了这些属性的样式单。

一开始你可能觉得这些代码很多，然而，如果仔细看，将看到每行代码其实没有多少信息。将每个样式规则放在一行上，是相当标准的做法，使样式单更易于阅读。谈到代码可读性，可能你看到这段样式单代码的第一反应是，它并不像普通 HTML 代码。CSS 使用自己的语言来指定样式单。

当然，样式单中也有一些熟悉的 HTML 标签。你可能猜到了，这个样式单中的 `body`、`p`、`h1`、`img`、`a` 和 `div` 指的是将应用该样式单的 HTML 文档中对应的标签。每个标签名后的大括

号内是对如何显示标签中内容的说明。

#### 程序清单 4.1 一个外部样式单

---

```
body {
  font-size: 10pt;
  font-family: Verdana, Geneva, Arial, Helvetica,sans-serif;
  color: black;
  line-height: 14pt;
  padding-left: 5pt;
  padding-right: 5pt;
  padding-top: 5pt;
}

h1 {
  font: 14pt Verdana, Geneva, Arial, Helvetica,sans-serif;
  font-weight: bold;
  line-height: 20pt;
}

p.subheader {
  font-weight: bold;
  color: #593d87;
}

img {
  padding: 3pt;
  float: right;
}

a {
  text-decoration: none;
}

a:link, a:visited {
  color: #8094d6;
}

a:hover, a:active {
  color: #FF9933;
}

div.footer {
  font-size: 9pt;
  font-style: italic;
  line-height: 12pt;
  text-align: center;
  padding-top: 30pt;
}
```

---

在这个例子中，所有 `body` 文本应该显示为 10 点的黑色 `Verdana` 字体（如果可能），行间距为 14 点。如果用户没有安装 `Verdana` 字体，样式单中的字体列表表示了浏览器寻找所用字体的顺序：`Geneva`，然后是 `Arial`，再往后是 `Helvetica`。如果用户没有这些字体，浏览器将使用可用的默认 `sans-serif` 字体。此外，该页面在左、右、上边各有 5 点的边距。

`<h1>` 标签中的所有文本应该显示为 14 点的加重 `Verdana` 字体。继续下去，所有仅使用 `<p>` 标签的段落将继承正文元素所指定的所有样式。但是，如果 `<p>` 标签使用特殊的类名 `subheader`，该文本将显示为加重和颜色 `#593d87`（紫色）。

在程序清单 4.1 中，每个数字后的 `pt` 表示 `point`（点，1 英寸相当于 72 点）。也可以用英

寸 (in)、厘米 (cm)、像素 (px) 或字母 m 的宽度 (em) 等作为样式单中的度量单位。

你可能注意到了, 程序清单中的每个样式规则都以分号 (;) 结束。分号用于分隔样式规则。在每条样式规则后加上分号已成为一种惯例, 这样便于在样式规则后加上另一条规则。

**注意:** 使用样式单可以将字体大小指定为任意值, 但有些显示设备和打印机可能不能正确处理超过 200 点的字体。

**By the Way**

要在 HTML 文档中链接样式单, 可在文档的 <head> 部分包含一个 <link /> 标签。程序清单 4.2 是图 4.1 所示页面的 HTML 代码, 它包含下面的 <link /> 标签:

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

这行代码假定样式单存储在该 HTML 文档所在文件夹下的文件 styles.css 中。只要 Web 浏览器支持样式单——现在所有浏览器都支持, 样式单指定的属性就将应用于网页的内容, 而不需要特殊的 HTML 格式化代码。这与 XHTML 的最终目标相符, XHTML 旨在将网页内容与显示这些内容使用的具体格式分离。

#### 程序清单 4.2 用于图 4.1 所示页面的 HTML 代码

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>About BAWSI</title>
    <link rel="stylesheet" type="text/css" href="styles.css" />
  </head>
  <body>
    <h1>About BAWSI</h1>
    <p>The Bay Area Women's
    Sports Initiative (BAWSI) is a public benefit, nonprofit
    corporation with a mission to create programs and partnerships
    through which women athletes bring health, hope and wholeness to
    our community. Founded in 2005 by Olympic and World Cup soccer
    stars Brandi Chastain and Julie Foudy and Marlene Bjornsrud,
    former general manager of the San Jose CyberRays women's
    professional soccer team, BAWSI provides a meaningful path for
    women athletes to become a more visible and valued part of the
    Bay Area sports culture.</p>
    <p class="subheader">BAWSI's History</p>
    <p>The concept of BAWSI was inspired by one of the most
    spectacular achievements in women's sports history and born out
    of one its biggest disappointments... </p>
    <p><a href="secondpage.html">[continue reading]</a></p>
    <div class="footer">Copyright &copy; 2005-2009 BAWSI
    (www.bawsi.org). All rights reserved. Used with permission.</div>
  </body>
</html>
```

程序清单 4.2 中的代码很有趣, 它没有包含任何格式化代码。换句话说, 没有 HTML 代码用于指示文本和图像应如何显示——没有颜色、字体, 什么格式都没有指定。然而, 由于链接了外部样式单 style.css, 网页是经过仔细格式化的。这种方式的优点在于, 当网站包含许多网页且要求网页的风格一致时, 使用这种方式将很容易。另外还有一个优点, 就是可以



将页面的样式分离出来，存储在一个独立的文档（样式单）中，这样在一个地方修改将影响所有页面。

### Did you Know?

**提示：**在大多数 Web 浏览器中，可以打开 .css 文件，并选择记事本或其他文本编辑器来查看该文件，从而查看样式单中的样式规则。要确定 .css 文件的文件名，可查看链接该文件的 HTML 源代码。要编辑自己的样式单，只要使用文本编辑器就可以了。

### By the Way

**注意：**虽然 CSS 在现代 Web 浏览器中得到广泛支持，但在以前的浏览器中并非如此。另外，并不是每个浏览器对 CSS 的支持都没有缺陷。要比较主要浏览器在支持 CSS 方面的差别，可参考网站 [http://www.westciv.com/style\\_master/academy/browser\\_support/](http://www.westciv.com/style_master/academy/browser_support/)。

### TRY IT YOURSELF ▼

**自己动手尝试一下：创建一个自己的样式单**

从头开始，创建一个新的文本文档，名为 `mystyles.css`，添加用于如下基本 HTML 标签的样式规则：`<body>`、`<p>`、`<h1>`和`<h2>`。创建了样式单之后，建立包含这些基本标签的新 HTML 文件。尝试不同的样式规则，自己看看在样式单文件中做一个小的修改来修改段落中的整块文本有多么简单。

## 4.3 CSS 样式入门

现在你已经具备了 CSS 样式单的基础知识，知道样式单如何根据样式规则来描述网页信息的外观。接下来的几个小节将介绍一些最重要的样式属性，并教你开始在自己的样式单中使用 CSS。

CSS 包括各种样式属性，用于控制字体、颜色、对齐方式和边距，这些只是网页样式的一些方面。CSS 中的样式属性分为两大类。

- 布局属性，由影响网页上元素位置的属性（如边距、填充、对齐等等）组成。
- 格式化属性，由影响网站中元素的视觉显示的属性（如字体类型、大小、颜色等）组成。

### 4.3.1 布局属性

CSS 布局属性用于决定内容在网页上如何排列。最重要的布局属性之一是 `display`，它描述元素相对于其他元素如何显示。`display` 属性有 4 个可能的取值。

- `block`：元素显示在下一行，作为一个新段落。
- `list-item`：元素显示在下一行，前面有列表元素标记（项目符号）。
- `inline`：元素显示在当前段落内。
- `none`：元素隐藏不显示。

将元素在网页上的显示想象成占据一块矩形区域，就更容易理解 `display` 属性：`display`

属性控制该矩形区域如何显示。例如，`block` 值使元素显示在下一行中，而 `inline` 使元素显示在之前的内容后。`display` 是能够在大多数样式规则中使用的几个样式属性之一。下面的例子演示了如何设置 `display` 属性：

```
display:block;
```

**注意：** `display` 属性依赖于相对位置的概念，相对位置指元素相对于网页中其他元素进行定位。CSS 也支持绝对位置，将元素放在网页中特定的位置，而不考虑其他元素。你将在第三部分“使用 CSS 进行高级 Web 设计”中学到这两种定位。

**By the Way**

`width` 和 `height` 属性控制元素矩形区域的大小。同许多与大小相关的 CSS 属性一样，`width` 和 `height` 属性的值可使用几种不同的度量单位：

- `in`（英寸）
- `cm`（厘米）
- `mm`（毫米）
- `px`（像素）
- `pt`（点）

在同一个样式单中，可以使用不同的度量单位，但类似的样式属性最好使用一致的单位。例如，字体大小属性的度量单位统一用点，尺寸统一用像素。下面是一个以像素为单位设置元素宽度的例子：

```
width:200px;
```

### 4.3.2 格式化属性

CSS 格式化属性用于控制网页内容的外观，而不是物理位置。最常用的格式化属性之一是 `border`，它用于在元素四周建立一个可见的方框或部分边框。下列 `border` 属性提供了描述元素边框的方法。

- `border-width`：边框的宽度。
- `border-color`：边框的颜色。
- `border-style`：边框的样式。
- `border-left`：边框的左边。
- `border-right`：边框的右边。
- `border-top`：边框的顶部。
- `border-bottom`：边框的底部。
- `border`：边框的所有边。

`border-width` 属性用于指定边框的宽度，通常以像素为单位，如下列代码所示：

```
border-width:5px;
```



不出意料，属性 `border-color` 和 `border-style` 用于设置边框的颜色和样式。下面是一个如何设置这两个属性的例子：

```
border-color:blue;
border-style:dotted;
```

`border-style` 属性可设置为下列值之一。

- `solid`: 单实线。
- `double`: 双实线。
- `dashed`: 虚线。
- `dotted`: 点线。
- `groove`: 沟线 (groove)。
- `ridge`: 脊线 (ridge)。
- `inset`: 内陷边框。
- `outset`: 外凸边框。
- `none`: 无边框。

`border-style` 属性的默认值为 `none`，这就是为什么除非将 `border` 属性设置为其他类型，否则元素没有边框的原因。最常见的 `border` 样式是 `solid` 和 `double`。

### By the Way

**注意：**`border-style` 属性的默认值为 `none`，这一规则有个例外，就是当图像放在 `<a>` 标签内作为链接图像时。在这种情况下，默认自动设置实线边框。这就是经常见到链接的图像使用样式 `boder-style:none`，将自动设置的边框去掉的原因。

属性 `border-left`、`border-right`、`border-top` 和 `border-bottom` 让你能够单独为元素的每一边设置边框。如果要使四周的边框都显示一致，只要使用 `border` 属性就可以了，后面接着下列样式：`border-width`、`border-style` 和 `border-color`。下面的例子使用 `border` 属性将边框设置为两条红线，宽度总共为 10 像素：

```
border:10px double red;
```

虽然元素边框的颜色由 `border-color` 属性设置，但元素内部区域的颜色由属性 `color` 和 `background-color` 设置。`color` 属性设置元素中文本的颜色（前景色），`background-color` 属性设置文本后面的背景色。下面的例子将这两个属性设置为预定义的颜色：

```
color:black;
background-color:orange;
```

也可以像在 HTML 中一样，通过指定十六进制（十六进制颜色在第 9 章“使用颜色”中详细介绍）或 RGB（红、绿、蓝）十进制值，为这些属性指定自定义的颜色：

```
background-color:#999999;
color:rgb(0,0,255);
```

控制网页内容的对齐和缩进也不麻烦，这是通过 `text-align` 和 `text-indent` 属性来实现的，如下列代码所示：

```
text-align:center;
text-indent:12px;
```

将元素对齐并缩进后，可能还要设置其字体。下列字体属性可用于设置与字体相关的各种参数。

- **font-family**: 字体集。
- **font-size**: 字体大小。
- **font-style**: 字体样式（正常或斜体等）。
- **font-weight**: 字体的粗细（细、中等、粗等）。

**font-family** 属性指定一个有优先级的字体集名称列表。使用有优先级的列表而不是一个值，旨在防止特定系统中没有安装指定的字体。**font-size** 属性使用度量单位（通常是点）来指定字体大小。最后，**font-style** 属性设置字体样式，**font-weight** 设置字体的粗细。下面是一个使用这些字体属性的例子：

```
font-family: Arial, sans-serif;
font-size: 36pt;
font-style: italic;
font-weight: medium;
```

知道许多关于样式属性及如何使用样式属性的知识后，再回过头去看一下程序清单 4.1，看看是否能够更好地理解这些代码。下面再复习一下该样式单用到的样式属性，帮助读者更好地理解样式属性。

- **font**: 让你能够一次指定许多字体属性。可以指定一个字体名称列表，名称之间用逗号分开。如果不能使用第一种字体，将尝试第二种，依此类推。也可以包含 **bold** 或 **italic** 及字体大小。每个字体属性都可以用单独的 **font-family**、**font-size**、**font-weight:bold** 和 **font-style:italic** 指定。
- **line-height**: 在出版界也称为 **leading**。它设置文本行的高度，通常以点为单位。
- **color**: 设置文本颜色，使用标准的颜色名称或十六进制颜色编码（详情请参见第 9 章）。
- **text-decoration**: 可用于删除链接的下划线——只要将该属性设置为 **none**。另外还支持 **underline**、**italic** 和 **line-through**。将样式应用于链接将在下一章更详细地介绍。
- **text-align**: 将文本向左 (**left**)、向右 (**right**) 或居中 (**center**) 对齐，同时使用一个 **justify** 值对每行文本进行排版。
- **padding**——在元素的左、右、上、下添加填充（内边距）；这一填充的单位可以为页宽的百分比。如果想要单独地在元素的左右添加填充，使用 **padding-left** 和 **padding-right**。使用 **padding-top** 或者 **padding-bottom** 相应地添加元素上下的填充。你将在第 14 章和第 15 章学习到这些样式属性。

## 4.4 使用样式类

这是一本“自学”书，因此读者不必参加课程来学习如何创建优秀的网页样式，但需要学习什么是样式类。需要使网页中的有些文本与其他文本不同时，可以创建自定义 HTML 标

签。定义的特殊格式文本称为样式类。样式类是一组自定义的格式化说明，可用于网页中的任何元素。

在演示样式类前，先解释一些 CSS 术语。首先，CSS 样式属性是一个可赋值的具体样式，如 `color` 或 `font-size`。通过选择符 (selector) 将样式规则及其值同网页中的元素关联起来。选择符用于标识网页中要应用样式的标签。下面是一个例子，在这个简单的样式规则中包含选择符、属性和值。

```
h1 { font:36pt Courier; }
```

在这行代码中，`h1` 是选择符，`font` 是样式属性，而 `36pt Courier` 是值。选择符很重要，因为它意味着该字体设置将应用网页中所有 `h1` 元素。但你可能想挑出一些 `h1` 元素，这时该怎么办？答案是使用样式类。

假设要在文档中使用两种不同的 `<h1>` 标签。可以在样式单中添加下列 CSS 代码，为每种标签创建一个样式类：

```
h1.silly { font:36pt Comic Sans; }  
h1.serious { font:36pt Arial; }
```

注意，这些选择符在 `h1` 之后包含一个点号 (.)，然后是一个描述性类名。在 HTML 页面中需要选择这两个样式类之一时，可使用 `class` 属性，如下所示：

```
<h1 class="silly">Marvin's Munchies Inc.</h1>  
<p> Text about Marvin's Munchies goes here.</p>
```

或者你可以这样使用：

```
<h1 class="serious">MMI Investor Information</h1>  
<p> Text for business investors goes here.</p>
```

在 HTML 代码中引用样式类时，只要在元素的 `class` 属性中指定类名。在这个例子中，如果在网页开头包含一个指向样式单的 `<link />` 标签，且用户安装了 `Comic Sans` 字体，`Marvin's Munchies Inc.` 将显示为 36 点的 `Comic Sans` 字体；而 `MMI Investor Information` 将显示为 36 点的 `Arial` 字体。你可以在程序清单 4.2 中看到类的另一个示例：寻找 `subheader<p>` 类和 `footer<div>` 类。

如果要创建可应用于任何元素的样式单，而不仅仅应用于标题或其他特定标签，该怎么办呢？可以像程序清单 4.2 中那样，将样式类与 `<div>` 标签相关联，该标签将文本放在类似于文本段落的块中。

使用 `div` 选择符加上点号以及样式类名和样式说明，可创建自定义的 HTML 标签。这种标签可同时控制任意数量的字体、间距和边距设置。要在网页中应用自定义标签，可使用 `<div>` 标签和 `class="创建的类名"`。

例如，程序清单 4.1 中的样式单包括下面的样式类说明：

```
div.footer {  
  font-size:9pt;  
  font-style: italic;  
  line-height:12pt;  
  text-align: center;  
  padding-top: 30pt;  
}
```



**提示：**你可能注意到了，样式规则包含多个属性时，样式编码将稍有不同。对于只包含一个属性的样式规则，通常属性与规则放在同一行，如下所示：

```
div.footer{ font-size:9pt; }
```

然而，样式规则包含多个样式属性时，如果每个属性都单独一行，代码将更易于阅读和理解，如下所示：

```
div.footer {  
  font-size: 9pt;  
  font-style: italic;  
  line-height: 12pt;  
  text-align: center;  
  padding-top: 30pt;  
}
```

在程序清单 4.2 中，使用下面的标签应用该样式单：

```
<div class="footer">
```

在程序清单 4.2 中，<div>标签和结束标签</div>之间的文本内容都显示为 9 点并居中对齐，斜体，行间距为 12 点并且在元素上方有 30 点的填充。

使用样式类的优点在于，它们将样式代码同网页分离，从而将 HTML 代码用于网页内容，而不是这些内容在屏幕上如何显示。然后，可以集中精力来调整样式单，控制内容如何显示。你可能会惊讶于很少的样式单代码就能对整个网站产生很大的影响。这使得网页更易于维护和操作。

#### 自己动手尝试一下：为你的样式单添加类

使用本章前面创建的样式单，添加一些样式类。为了看到你的劳动成果，将这些类应用到你创建的 HTML 页面。为你的<h1>和<p>标签使用类来感受一下。

TRY IT YOURSELF ▼

## 4.5 使用样式 ID

当你创建自定义样式单时，你可以使用这些类任意多次——它们不是唯一的。但是，有时候你希望因为布局或者格式化的目的（或者两者都有）对独特的元素进行精确的控制。在这种情况下，用 ID 替代类。

样式 ID 是一组自定义的格式化规范，可以只应用到网页中的一个元素。你可以在一组页面中使用 ID，但是每页只使用一次。

例如，假定你在所有页面的正文中有一个标题。每个页面只有一个标题，但是所有页面都包含标题的一个实例。下面的选择符示例指出了 ID，加上一个属性和值：

```
p#title{font: 24pt Verdana, Geneva, Arial, sans-serif}
```

注意，这个选择符在 p 后包含一个 hash 符号(#)，跟着是一个描述性的 ID 名称。在 HTML 代码中引用样式 ID 时，只要在元素的 id 属性中指定 ID 名称就可以了，如：

```
<p id="title">Some Title Goes Here</p>
```

开始和结束<p>标签之间的所有文本将显示为 24 点的 Verdana 字体——但是在任何指定

的页面中只出现一次。你将会经常看到样式 ID 用于布局的目的，定义页面的特殊部分，比如标题区域、页脚区域、正文区域等。页面中的这些区域在一页中只出现一次，所以使用 ID 比类更合适。

## 4.6 内部样式单和内联样式

在有些情况下，你可能希望指定只用于一个网页的样式，在这种情况下，可将样式单放在标签 `<style>` 和 `</style>` 内，直接包含在 HTML 文档中。以这种方式使用的样式单必须出现在 HTML 文档的 `head` 中。不需要 `<link />` 标签，在其他网页中不能引用该样式单（除非将其复制到该文档的开始部分）。这种样式单称为内部样式单。

程序清单 4.3 是一个如何指定内部样式单的例子。

程序清单 4.3 一个具有内部样式单的网页

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Some Page</title>

    <style type="text/css">
      div.footer {
        font-size: 9pt;
        line-height: 12pt;
        text-align: center;
      }
    </style>
  </head>
  <body>
    ...
    <div class="footer">
      Copyright 2009 Acme Products, Inc.
    </div>
  </body>
</html>
```

在这段代码中，`div.footer` 样式类是在一个内部样式单中指定的，后者位于页面的开头。这样，该样式类可用于这个页面的正文。实际上，在页面正文中，它用于格式化版权标记。

创建只在一个页面中使用多次的样式规则时，内部样式单很方便。然而，在有些情况下，可能需要将样式应用于特定的元素。这称为内联（inline）样式规则，让你能够为页面的一小部分，如单个元素指定样式。例如，可以在 `<p>`、`<div>` 或 `<span>` 标签内使用 `style` 属性创建并应用样式规则。这种样式称为内联样式，因为是在 HTML 代码中指定的。

**By the  
Way**

**注意：**`<span>`和`</span>`实际上是伪标签（dummy tag），本身并没有什么功能，只是指定添加的 `style` 属性将应用的范围。标签 `<div>` 和 `<span>` 之间唯一的区别是，`<div>` 是块元素，包含分行，而 `<span>` 不包含。因此，要修改不分行的句子或段落中的文本样式，必须使用 `<span>`。

下面是一个 `style` 属性的例子：

```
<p style="color:green">
  This text is green, but <span style="color:red">this text is
  red.</span>
  Back to green again, but...
</p>
<p>
  ...now the green is over, and we're back to the default color
  for this page.
</p>
```

这段代码使用 `<span>` 标签来说明如何在内联样式规则中应用颜色样式属性。实际上，这个例子中的 `<p>` 标签和 `<span>` 标签都将颜色属性用作内联样式。很重要的一点是，`color:red` 样式属性将覆盖标签 `<span>` 和 `</span>` 之间的文本的 `color:green` 样式。在第二段，不使用这两个 `color` 属性，因为这是一个新段落，使用的是整个页面的默认颜色。

## 验证你的样式单

和验证 HTML 或者 XHTML 标示一样，对样式单的验证也很重要。可以在 <http://jigsaw.w3.org/css-validator/> 中找到一个特殊的验证工具。和第 3 章中讨论过的验证工具一样，你可以将这个工具指向地址，上传文件，或者粘贴内容到表单字段中。终极目标是如图 4.3 所示的结果。

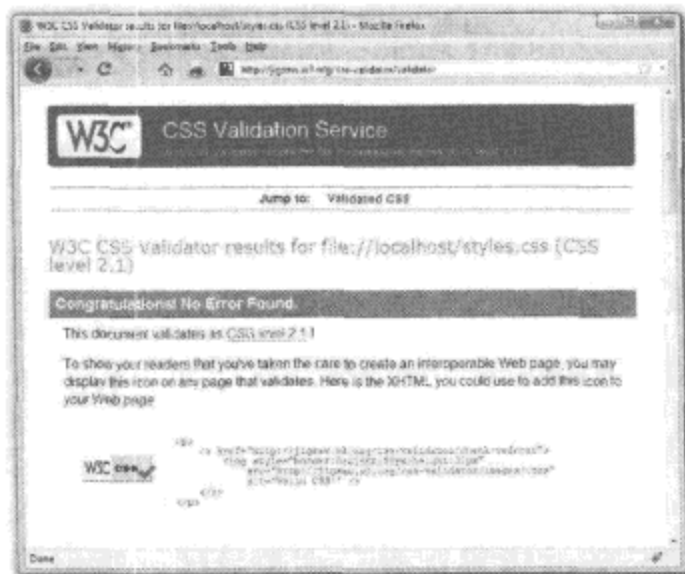


图 4.3

W3C CSS 验证工具显示程序清单 4.1 中的样式单内容没有错误

## 4.7 总结

本章中你学习了样式单可同时控制多个 HTML 网页的外观，它还能够极其精确地控制 HTML 元素的字体、间距和位置。你还了解到，几乎所有 HTML 标签都可以加入 `style` 属性，从而控制 HTML 网页的任何部分，而不需要引用独立的样式单文档。

你学习了 3 种在网站中包含样式单的主要方法：在文档的 `<head>` 部分链接到带有 `.css` 的单独样式单文件，放在文档标题部分 `<style>` 标签中的一组样式规则，以及通过 `style` 属性直接放在 HTML 标签中的规则。

如表 4.1 所示，总结了本章讨论的标签。关于可放在 `<style>` 标签或 `style` 属性后的选项的具体细节，请参阅 [www.w3c.org](http://www.w3c.org) 中的 CSS2 样式标准。

表 4.1 第4章介绍的 HTML 标签与属性

标签/属性	功能
<code>&lt;style&gt;...&lt;/style&gt;</code>	使得可以在文档中包含内部样式单，用于 <code>&lt;head&gt;</code> 和 <code>&lt;/head&gt;</code> 之间
属性	
<code>type="内容类型"</code>	Internet 内容类型，对于 CSS 样式单，总是 <code>text/css</code>
<code>&lt;link /&gt;</code>	链接外部样式单（或其他文档类型），用于文档的 <code>&lt;head&gt;</code> 部分
属性	
<code>href="url"</code>	样式单的地址
<code>type="内容类型"</code>	Internet 内容类型，对于 CSS 样式单，总是 <code>text/css</code>
<code>rel="样式单"</code>	链接类型，对于样式单，总是 <code>stylesheet</code>
<code>&lt;span&gt;...&lt;/span&gt;</code>	除提供一个放置样式或其他属性的位置外，不起什么作用。类似于 <code>&lt;div&gt;...&lt;/div&gt;</code> ，但不产生分行
属性	
<code>style="style"</code>	包含内联样式说明，可用于 <code>&lt;span&gt;</code> 、 <code>&lt;div&gt;</code> 、 <code>&lt;body&gt;</code> 和其他大部分 HTML 标签中

## 4.8 问与答

问：如果网页链接了一个样式单，后者指定所有文本为蓝色，但在网页的某处有一个`<span style="font-color:red">`标签，则这个标签内的文本将是蓝色还是红色？

答：红色。局部内联样式总是优先于外部样式单。放在网页开头的`<style>`与`</style>`之间的样式也优先于外部样式单（但不优先于网页后面的内联样式）。这是我在本章开头提到过的级联效果。可以将级联效果看为首先从外部样式单开始，然后被内部样式单覆盖，再被内联样式单覆盖。

问：一个网页可以链接多个样式单吗？

答：当然可以。例如，可将一个样式单用于格式化（如文本、字体、颜色等），另一个样式单用于布局（如边距、填充、对齐等）——只要分别用一个`<link />`包含这两个样式单。从技术上说，CSS 标准要求网页浏览器允许用户在多个`<link />`标签提供的多个样式单中进行选择。然而，实际上所有主流网络浏览器只是包含每一个样式单。实际上，链接多个样式单更常使用`@import`命令。下面是一个通过`@import`导入多个样式单的例子：

```
@import url(styles1.css);
@import url(styles2.css);
```

与`<link />`标签一样，`@import`命令必须放在网页的开头。你将在第 15 章进一步学习这个方便的小命令，那时你将学习如何创建专门用于打印网页的样式单。

## 4.9 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

## 4.9.1 测验

1. 创建一个样式单，指定标题使用 30 点的 Arial 字体，颜色为蓝色；其他所有文本使用 10 点的 Times Roman 字体（或默认的浏览器字体），颜色为蓝色，行间距为两倍行间距。
2. 假设在问题 1 中创建的样式单为 corporate.css，如何将它应用于名为 intro.html 的网页？
3. 有几种方法能确保样式规则应用到你的内容？

## 4.9.2 测验答案

1.

```
h1 { font:30pt blue Arial; }  
body { font: 10pt blue; }
```

2. 将如下的标签放在 intro.html 文档的<head>与</head>标签之间：

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

3. 3 种：外部、内部和内联。

## 4.10 练习

- 为你的网站开发一个标准样式单，并在所有网页中链接该样式单（对于不需要使用这个样式单的网页，使用内部样式单或内联样式）。如果为某家公司工作，很可能为打印材料指定字体和样式，按这些规范设置公司的网页。
- 研究 <http://www.w3.org/Style/CSS/> 上关于样式单的官方规范，尝试使用一些本章没有提及的高级样式属性。





## 第 5 章

# 使用文本块和列表

---

本章中你将学到：

- 在页面上对齐文本的方法
- 三种 HTML 列表的使用方法
- 在列表中放置列表的方法

在早期的 Web 中，所有文本都以相同的字体和大小显示。但是，现在 HTML 和 CSS 的组合能够控制文本的外观以及在网页上的对齐和显示方式。本章将为你展示文本对齐的基础知识，并且指导你一些高级的技巧和窍门，比如列表的使用。因为列表很常见，HTML 提供了标签，自动地缩进文本，并且在每个列表项的前面添加数字、项目符号或者其他符号。你将学习到不同类型的列表的格式化，这是在你的网站中显示内容的多种方法之一。

### TRY IT YOURSELF ▼

#### 自己动手尝试一下：准备示例文本

如果有文本需要缩进、居中或进行其他视觉处理以便更容易阅读，将可充分利用本章介绍的知识。

- 任何类型的大纲、演示文稿中的项目列表、编号的步骤、词汇表或数据库中的文本信息列表，都可作为工作对象。
- 使用任何文本都可以，但尽量使用要放到网页中的文本。公司宣传册或个人简历中的文本是不错的选择。
- 如果使用的文本来自字处理软件或数据库程序，将它存储为纯文本文件或 ASCII 文件。然后随着本章课程的学习，在文本中添加合适的 HTML 标签和样式属性，对文本进行格式化。
- 添加前面的章节中介绍的标签 `<html>`、`<head>`、`<title>` 和 `<body>`，然后使用本章介绍的代码对正文文本进行格式化。

## 5.1 在页面上对齐文本

在 Web 上浏览信息时，很容易理所当然地认为段落都自动地左对齐。然而，有些时候可能选择将文本向右或居中对齐。HTML 可以选择对齐一个 HTML 块级元素，如包含在 `<p></p>` 或者 `<div></div>` 标签对中的文本。在我们进入对齐块元素的细节之前，先简单地说明一下属性的工作方式。

### 5.1.1 使用属性

属性 (attribute) 用于提供与 HTML 标签相关的附加信息。属性是用于在 HTML 标签内部的特殊代码，精确地控制标签的行为。它们即使在最简单的 Web 内容中也是很重要的，所以你能轻松地使用它们也是很重要的。

属性引起应用到特殊标签的样式，类或者 ID 的使用。如果你在一个样式单中定义了特殊的类——正如在第 4 章中你所学习到的——那么你可以在标签中使用 `class="类名称"` 或者 `id="ID 名称"` 来调用类或者 ID。当浏览器呈现这些内容时，它将查看样式单，根据样式定义来确定内容的准确显示方式。相似地，你可以使用 `style` 属性为特色保护的元素包含样式信息，无需将该元素连接到实际的样式单。例如，当你以 `<p>` 标签开始一个段落，你可以设置 `style` 属性指定特殊段落中的文本应该左对齐、右对齐还是居中。如果你希望将特殊的段落与现有的类或者 ID 关联，设置 `class` 或者 `id` 属性。

下面的例子中，每个段落可以是左对齐：

```
<p style="text-align: left;">Text goes here.</p>
<p class="leftAlignStyle">Text goes here.</p>
<p id="firstLeftAlign">Text goes here.</p>
```

在第一个段落中，样式直接出现在样式属性中。在第二个段落中，如果 `leftAlignStyle` 类的样式单条目包含 `text-align` 语句，段落将为左对齐。相似地，如果 `firstLeftAlignStyle` 类的样式单条目包含 `text-align` 语句，第三个段落将为左对齐。

在前一个示例和前面的章节中的示例中，你可能注意到了，标签、属性和样式使用小写字母。精确的 XHTML 标准要求标签和属性必须小写，XHTML 标准还要求使用引号将属性值括起。

例如，下面的代码从技术上说是大多数主流网页浏览器都能接受的：

```
<P STYLE=TEXT-ALIGN:CENTER>
```

但这行代码不符合网页设计的最新标准，因为它是大写的，且 `style` 属性值 `text-align:cernter` 没有用引号括起。要与将来的标准和软件保持兼容，应使用下面的代码：

```
<p style="text-align:center">
```

### 5.1.2 对齐块级别元素

为了使块级元素如 `<p>` 右对齐，而不在样式单中创建单独的类或者 ID，只要在段落开始

的<p>标签内放置 `style="text-align:right"`。相似地，为了居中该元素，使用<p `style="text-align:center"`>。为了左对齐段落，使用<p `style="text-align:left"`>。

Style 属性的 `text-align` 部分被称为样式规则，这意味着它设置 HTML 元素的特定样式特征。你可以使用许多样式规则来小心地控制 Web 内容的格式。

`text-align` 样式规则不仅为<p>标签所保留。实际上，你可以将 `text-align` 样式规则用于任何块级元素，这包括<h1>、<h2>及其他标题标签，以及<div>标签等。<div>标签特别方便，因为它可以包含其他块级元素，从而使你能一次性控制大部分 Web 内容的对齐。<div>标签中的 `div` 指的是 `division`（分组）。

程序清单 5.1 示范了<p>和<div>标签的 `style` 属性和 `text-align` 样式规则。结果如图 5.1 所示。你将在第三部分学习<div>标签的更多更高级的用法。

### By the Way

**注意：**如果没有设置其值，HTML 中的每种属性和样式规则都有默认值。在<p>标签的 `text-align` 样式规则中，默认值为 `left`，所以使用不带修饰的<p>标签和使用<p `style="text-align:left"`>效果相同。学习常用样式规则的默认值是成为好的网页开发人员的重要部分。

### 程序清单 5.1 style 属性使用的 text-align 样式规则

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Bohemia</title>
  </head>

  <body>
    <div style="text-align:center">
      <h1>Bohemia</h1>
      <h2>by Dorothy Parker</h2>
    </div>
    <p style="text-align:left">
      Authors and actors and artists and such<br />
      Never know nothing, and never know much.<br />
      Sculptors and singers and those of their kidney<br />
      Tell their affairs from Seattle to Sydney.
    </p>
    <p style="text-align:center">
      Playwrights and poets and such horses' necks<br />
      Start off from anywhere, end up at sex.<br />
      Diarists, critics, and similar roe<br />
      Never say nothing, and never say no.
    </p>
    <p style="text-align:right">
      People Who Do Things exceed my endurance;<br />
      God, for a man that solicits insurance!
    </p>
  </body>
</html>
```

<div style="text-align:center">的使用确保了内容区域（包括两个标题）居中。但是，<div>

中的单独段落的文本对齐覆盖了该设置并且确保了第一段的文本为左对齐，第二段为居中，第三段为右对齐。

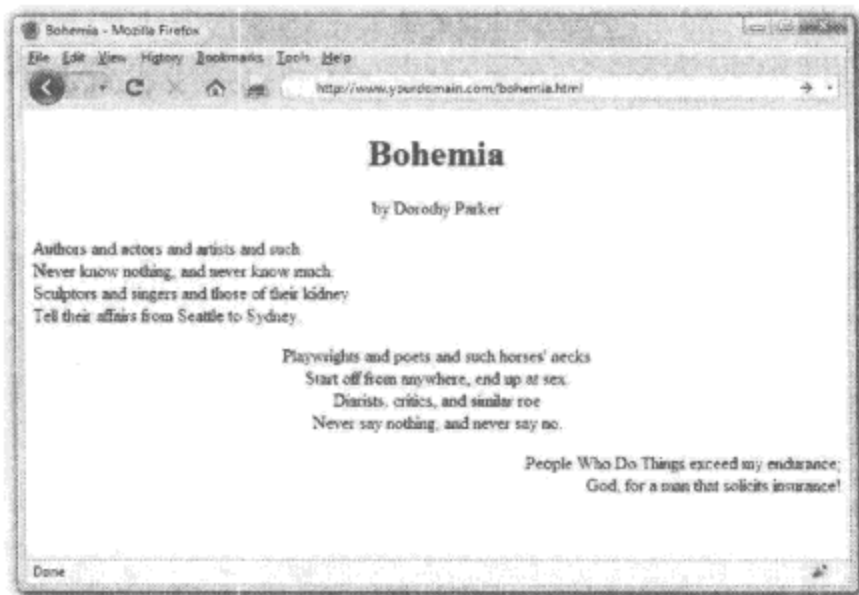


图 5.1

使用程序清单 5.1 中的文本对齐的结果

## 5.2 三种 HTML 列表

为了清晰，常常将网页上的信息表现为列表项目。有 3 种基本的 HTML 列表。如图 5.2 所示，显示了所有列表类型，程序清单 5.2 揭示了构建它们的 HTML。

- 有序列表——在每个列表项目前有数字或者字母的缩进列表。有序列表以<ol>标签开始，</ol>标签结束。列表项包含在<li></li>标签对中，每个开始<li>标签处自动出现一个换行。整个列表都缩进。
- 无序列表——每个列表项目前有一个项目符号或者其他符号的缩进列表。无序列表以<ul>标签开始，</ul>标签结束。和有序列表类似，列表项包含在<li></li>标签对中，每个开始<li>标签处自动出现一个换行和符号，整个列表都缩进。
- ➤ 定义列表——项目及其含义的列表。这种类型的列表在每个项目之前没有特殊的数字、字母或者符号，以<dl>开始，</dl>结束。<dt></dt>标签对包含每个项目，<dd></dd>标签对包含每个定义。换行和缩进自动出现。

### 程序清单 5.2 无序列表、有序列表和定义列表

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>How to Be Proper</title>
  </head>

  <body>
    <h1>How to Be Proper</h1>
    <h2>Basic Etiquette for a Gentlemen Greeting a Lady Acquaintance</h2>
    <ul>
      <li>Wait for her acknowledging bow before tipping your hat.</li>
      <li>Use the hand farthest from her to raise the hat.</li>
    </ul>
  </body>
</html>
```



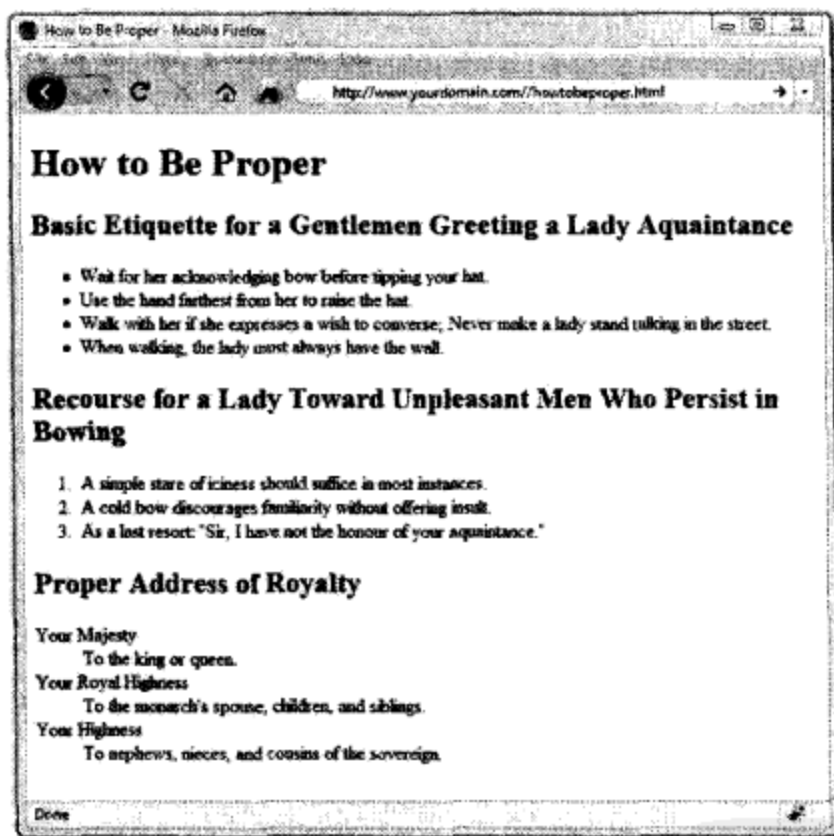
```

<li>Walk with her if she expresses a wish to converse; Never
make a lady stand talking in the street.</li>
<li>When walking, the lady must always have the wall.</li>
</ul>
<h2>Recourse for a Lady Toward Unpleasant Men Who Persist in Bowing</h2>
<ol>
<li>A simple stare of iciness should suffice in most instances.</li>
<li>A cold bow discourages familiarity without offering insult.</li>
<li>As a last resort: "Sir, I have not the honour of your
acquaintance."</li>
</ol>
<h2>Proper Address of Royalty</h2>
<dl>
<dt>Your Majesty</dt>
<dd>To the king or queen.</dd>
<dt>Your Royal Highness</dt>
<dd>To the monarch's spouse, children, and siblings.</dd>
<dt>Your Highness</dt>
<dd>To nephews, nieces, and cousins of the sovereign.</dd>
</dl>
</body>
</html>

```

图 5.2

三种基本 HTML 列表类型



**By the Way**

**注意：**记住，不同的 Web 浏览器显示的 Web 内容可能有所不同。HTML 标准没有精确地规定 Web 浏览器格式化列表的方式，所以使用旧的 Web 浏览器的用户可能不能完全看到和你相同的缩进。你可以使用 CSS 得到对列表项目的精确控制，这些内容你将在本章稍后学习到。

## 5.3 将列表放在列表中

虽然设计定义列表的本意是用于定义术语，但许多网页制作者需要在需要缩进的任何地方都使用定义列表。实际上，可以在需要缩进的文本前面加上 `<dl><dd>`，在文本结束处加上 `</dl></dd>`。然而，还有一种更好的方法，就是使用 `<blockquote></blockquote>` 标签对，这缩



进内容，而不假定为一个定义并且允许更清晰的样式。使用一组属性，你可以设置元素区域的宽度、高度、背景颜色、边框类型和颜色，以及其他视觉效果。

因为使用 CSS 所拥有的对项目显示的控制，没有必要使用嵌套列表来达到所进的视觉外观。将嵌套列表留作内容要求时使用。换句话说，使用嵌套列表来说明信息的层次化结构，如程序清单 5.3。

**注意：**嵌套指的是一个标签完全出现在另一个标签之中。包含其他标签的标签称为父标签，被嵌套在内的标签称为子标签。通常（但不是必须）嵌套的标签要缩进，这样容易看出它们与父标签之间的关系。

**By the  
Way**

有序列表和无序列表也可以互相嵌套，嵌套的级别可以任意深。在程序清单 5.3 中，几个无序列表组成了一个复杂的缩进大纲。如图 5.3 所示，Firefox 自动为前 3 个级别的缩进使用不同类型的项目符号，使列表更具可读性。这在现代浏览器中很常见。

### 程序清单 5.3 使用列表建立大纲

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Vertebrates</title>
  </head>

  <body>
    <h1>Vertebrates</h1>
    <ul>
      <li><span style="font-weight:bold">Fish</span>
        <ul>
          <li>Barramundi</li>
          <li>Kissing Gourami</li>
          <li>Mummichog</li>
        </ul>
      </li>
      <li><span style="font-weight:bold">Amphibians</span>
        <ul>
          <li>Anura
            <ul>
              <li>Goliath Frog</li>
              <li>Poison Dart Frog</li>
              <li>Purple Frog</li>
            </ul>
          </li>
          <li>Caudata
            <ul>
              <li>Hellbender</li>
              <li>Mudpuppy</li>
            </ul>
          </li>
        </ul>
      </li>
      <li><span style="font-weight:bold">Reptiles</span>
        <ul>
          <li>Nile Crocodile</li>
```



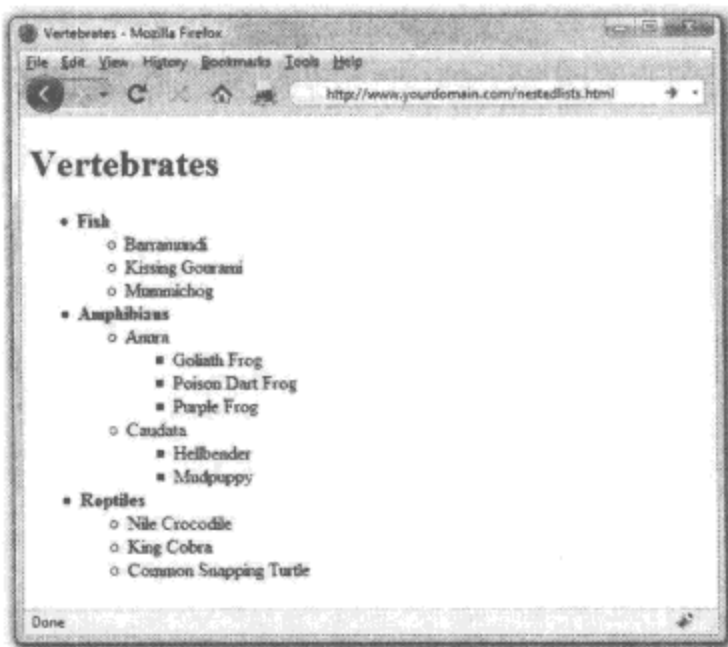
```

        <li>King Cobra</li>
        <li>Common Snapping Turtle</li>
    </ul>
</li>
</ul>
</body>
</html>

```

图 5.3

在 Firefox 中,多级无序列列表整齐地缩进并且具有项目符号,增加了可读性



对于图 5.3 所示的定义列表,浏览器通常都使用实心圆圈作为第一级项目符号,第二级项目符号使用空心圆圈,更深的级别将使用实心方框。然而,可以使用 `<ul style="list-style-type:disc">`、`<ul style="list-style-type:circle">`或`<ul style="list-style-type:square">`来代替`<ul>`,从而显式地指定每个级别要使用的项目符号类型。

甚至可以在无序列列表的`<li>`标签内使用 `list-style-type` 样式规则,为单个列表项修改项目符号。例如,下列代码将在 `extra` 和 `super` 之前显示空心圆圈,而在 `special` 之前显示实心方框:

```

<ul style="list-style-type:circle">
  <li>extra</li>
  <li>super</li>
  <li style="list-style-type:square">special</li>
</ul>

```

`list-style-type` 样式规则也可用于有序列表,但在有序列表中,不是选择项目符号类型,而是选择每个列表项之前的编号或字母类型。程序清单 5.4 显示了如何在多级列表中使用罗马数字 (`list-style-type:upper-roman`)、大写字母 (`list-style-type:upper-alpha`)、小写字母 (`list-style-type:lower-alpha`) 和普通数字。如图 5.4 所示为大纲的格式化结果。

#### 程序清单 5.4 在多级列表内使用样式属性的 `list-style-type` 样式规则

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Advice from the Golf Guru</title>
  </head>

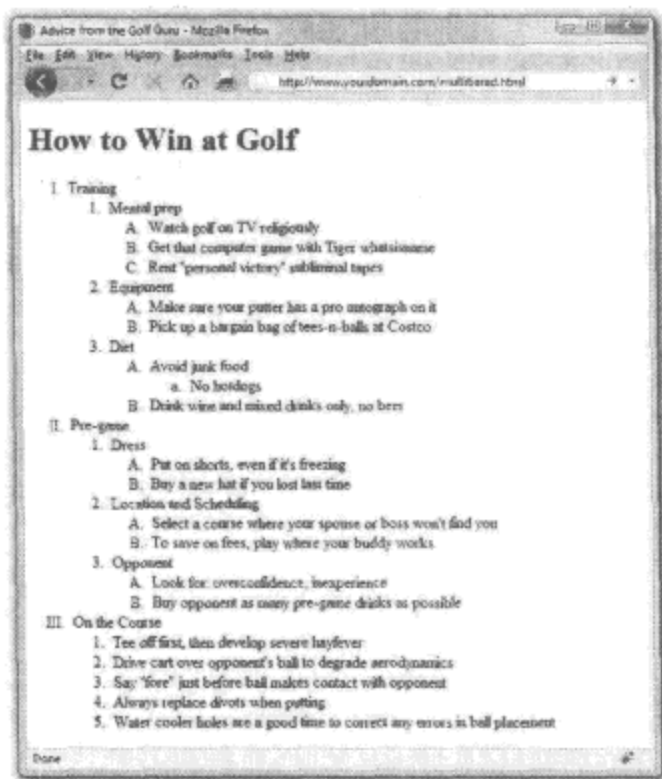
  <body>
    <h1>How to Win at Golf</h1>

```

```
<ol style="list-style-type:upper-roman">
  <li>Training
    <ol>
      <li>Mental prep
        <ol style="list-style-type:upper-alpha">
          <li>Watch golf on TV religiously</li>
          <li>Get that computer game with Tiger whatsisname</li>
          <li>Rent "personal victory" subliminal tapes</li>
        </ol>
      </li>
      <li>Equipment
        <ol style="list-style-type:upper-alpha">
          <li>Make sure your putter has a pro autograph on it</li>
          <li>Pick up a bargain bag of tees-n-balls at Costco</li>
        </ol>
      </li>
      <li>Diet
        <ol style="list-style-type:upper-alpha">
          <li>Avoid junk food
            <ol style="list-style-type:lower-alpha">
              <li>No hotdogs</li>
            </ol>
          </li>
          <li>Drink wine and mixed drinks only, no beer</li>
        </ol>
      </li>
    </ol>
  </li>
  <li>Pre-game
    <ol>
      <li>Dress
        <ol style="list-style-type:upper-alpha">
          <li>Put on shorts, even if it's freezing</li>
          <li>Buy a new hat if you lost last time</li>
        </ol>
      </li>
      <li>Location and Scheduling
        <ol style="list-style-type:upper-alpha">
          <li>Select a course where your spouse or boss won't find you</li>
          <li>To save on fees, play where your buddy works</li>
        </ol>
      </li>
      <li>Opponent
        <ol style="list-style-type:upper-alpha">
          <li>Look for: overconfidence, inexperience</li>
          <li>Buy opponent as many pre-game drinks as possible</li>
        </ol>
      </li>
    </ol>
  </li>
  <li>On the Course
    <ol>
      <li>Tee off first, then develop severe hayfever</li>
      <li>Drive cart over opponent's ball to degrade aerodynamics</li>
      <li>Say "fore" just before ball makes contact with opponent</li>
      <li>Always replace divots when putting</li>
      <li>Water cooler holes are a good time to correct any errors in ball placement</li>
    </ol>
  </li>
</ol>
</body>
</html>
```

图 5.4

格式良好的大纲使  
计划锦上添花



## 5.4 总结

在本章中，你学习了用于指定许多 HTML 标签的选项和特殊行为的属性，还学习了创建和组合 3 种基本 HTML 类型（有序列表、无序列表、定义列表）的方法。列表可以被放在其他列表之中以创建大纲和其他复杂的文本排列。

如表 5.1 所示，总结了本章讨论的标签和属性。顺便说一句，不要觉得一定要记住所有标签！这是你拥有这本书的原因，你可以在需要的时候查找这些标签。所有的 HTML 标签在附录 A 中列出。

表 5.1

第 5 章中介绍的 HTML 标签和属性

标签/属性	功能
<code>&lt;div&gt;...&lt;/div&gt;</code>	要格式化的文本区域
<code>&lt;dl&gt;...&lt;/dl&gt;</code>	定义列表
<code>&lt;dt&gt;...&lt;/dt&gt;</code>	定义项，作为定义列表的一部分
<code>&lt;dd&gt;...&lt;/dd&gt;</code>	与定义项对应的定义，也是定义列表的一部分
<code>&lt;ol&gt;...&lt;/ol&gt;</code>	有序（带编号的）列表
<code>&lt;ul&gt;...&lt;/ul&gt;</code>	无序（带项目符号的）列表
<code>&lt;li&gt;...&lt;/li&gt;</code>	用于<ol>或<ul>的列表项
<b>属性</b>	
<code>style="text-align:alignment"</code>	将文本居中对齐、左对齐或右对齐（也可用于<p>、<h1>、<h2>和<h3>等）
<code>style="list-style-type:numtype"</code>	用于标记列表的编号类型。可能的值包括：decimal（十进制数字）、lower-roman（小写罗马数字）、upper-roman（大写罗马数字）、lower-alpha（小写 alpha 数字）、upper-alpha（大写 alpha 数字）以及 none（不使用数字）
<code>style="list-style-type:bullettype"</code>	用于标记列表项的项目符号类型。可能的值包括：disc、circle、square 和 none
<code>style="list-style-type:type"</code>	标记列表项的编号或项目符号。可能的值包括：disc、circle、square、decimal、lower-roman、upper-roman、lower-alpha、upper-alpha 和 none

## 5.5 问与答

问：我见过 Internet 上的网页使用三维小球或其他特殊图像作为项目符号。这是怎么实现的？

答：这种技巧超出了本章的范围，在第 11 章你将学会如何实现。

问：我如何“完全对齐”文本，使左右边距相同

答：你可以在样式声明中使用 `text-align:justify`。

## 5.6 作业

作业包含一些测试题及练习，帮助巩固对这一章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 5.6.1 测验

1. 如何使网页中的所有内容都居中对齐？
2. 如何缩进一个单词并在它的前面加上方框项目符号？
3. 使用定义列表显示：单词 `glunch` 的含义是 “a look of disdain, anger, or displeasure”，单词 `glumpy` 的含义是 “sullen, morose, or sulky”。

### 5.6.2 测验答案

1. 如果你考虑将 `<div style="text-align:center">` 放在页面开头的 `<body>` 标签后，将 `</div>` 放在页面末尾的 `</body>` 之前，这是正确的。然而，`<body>` 标签也支持 `text-align` 样式，也就是说可以不用 `<div>` 标签，直接将样式 `style="text-align:center"` 放在 `<body>` 标签中，这样网页就都居中了。

2.

```
<ul style="list-style-type:square">  
  <li>supercalifragilisticexpealidocious</li>  
</ul>
```

将 `style="list-style-type:square"` 放在 `<li>` 标签中也能得到相同的结果，因为这个列表中只有一项。

3.

```
<dl>  
<dt>glunch</dt><dd>a look of disdain, anger, or displeasure</dd>  
<dt>glumpy</dt><dd>sullen, morose, or sulky</dd>  
</dl>
```



## 5.7 练习

- 使用文本对齐样式属性将文本块放在你网页中的各种位置。尝试嵌套你的段落和分组（<p>和<div>）以感觉样式在内容层次中层叠的方式。
- 试着为要放在网页中的信息创建一个有序列表大纲。这将让你练习格式化 HTML 列表，还让你提前考虑后面的章节将介绍的问题。



## 第 6 章

# 使用字体

---

本章中你将学到：

- 粗体、斜体和特殊文本格式的使用
- 字体的调整方法
- 特殊字符的使用方法

在前一章中，你学习了创建文本块并将其放入列表的基础知识。在本章中，你将更详细地了解文本本身，并且学习改变字体视觉效果的方法——例如字体集、大小、粗细。你将学习到在页面中组合粗体、斜体、上标、下标和删除线的方法。你还将学到改变字体和字体大小的方法。

**注意：**在查看其他设计人员的 Web 内容时，你可能会注意到标记文本的方法与本书所述的不同。格式化文本的“老方法”包括使用 `<b></b>` 标签对表示应该加重的词语，`<i></i>` 标签对用于表示应该为斜体的词语，以及使用 `<font></font>` 标签对指定字体集、大小和其他属性。但是，没有理由学习这种方法，因为它已经逐渐被 HTML 淘汰，CSS 更加强大。

**By the  
Way**

**自己动手尝试一下：示例文本上的更多工作**

正像你在上一章中创建一个示例文件一样，为本章创建一个示例，这样你可以用它在本章中学习为文本加上样式的方法。

因为本章中的信息必须使用字体级别样式，所以使用哪种文本类型没有什么关系。有许多可以尝试的可能的样式，以至于它们无法同时出现在一个页面上（除非你希望把访问者逼疯）。抓住这个机会，感受文本级别变化对内容外观的影响。

TRY IT YOURSELF ▼

PDG

## 6.1 粗体、斜体和特殊文本格式

在打字机时代，通常只使用纯文本，偶尔使用下划线突出内容。现在，粗体和斜体在书面通信中很普遍。通常也可以在网页上添加粗体和斜体文本。有几个标签和样式规则可用于文本格式化。

添加粗体和斜体文本格式的“守旧”方法是包含**<b></b>**和**<i></i>**标签对。对于粗体文本，在文本的开头放置**<b>**，在结束放置**</b>**。相似地，你可以通过将文本包含在**<i>**和**</i>**之间来使其变成斜体。虽然这种方法在浏览器中仍然正常工作并且为 XHTML 所支持，但是它不像用于文本格式的 CSS 样式规则那么灵活和强大。

虽然你在第三部分将学到更多 CSS 样式规则，但现在应提前学习一点，这样才能理解文本格式化的样式规则。font-weight 样式规则让你能够使用样式规则来设置字体的粗细。标准设置包括 normal、bold、bolder 和 lighter，其中 normal 是默认设置。斜体文本通过 font-style 规则来控制，该规则可设置为 normal、italic 或 oblique。如果要应用多种格式，可指定多个样式规则，如下面的示例所示：

```
<p style="font-weight:bold; font-style:italic">This paragraph is bold and italic!</p>
```

在这个示例中，在<p>标签的 style 属性中指定了两个样式规则。使用多个样式规则的关键在于它们必须使用分号分开。

文本样式不仅可用于段落。下列代码展示了如何将带项目符号的列表中的文本格式化为斜体：

```
<ul>
  <li style="font-style:italic">Important Stuff</li>
  <li style="font-style:italic">Critical Information</li>
  <li style="font-style:italic">Highly Sensitive Material</li>
  <li>Nothing All That Useful</li>
</ul>
```

也可以在标题中使用<b>标签和 font-weight 样式规则，但由于标题默认已经是粗体，因此更重的字体并不会产生什么新效果。

### By the Way

**注意：**格式化粗体和斜体时，还有一种方法可以代替样式规则，那就是<strong>标签和<em>标签。在大多数浏览器中，<strong>标签的作用与<b>标签相同，而<em>标签的作用与<i>标签相同。有人认为，标签<strong>和<em>是对标签<b>和<i>的改进，因为前两者只指出文本应突出，而没有指定突出效果如何实现。换句话说，浏览器不一定将<strong>解释为粗体，将<em>解释为斜体。这使得<strong>和<em>更适合于XHTML，因为它们使文本增加了额外的意义，并影响文本的显示。所有这4个标记仍然是HTML 5的一部分，但是它们的使用稍有差别。

虽然使用 CSS 使你能够应用更丰富的格式，但是当你不需要像 CSS 所允许的那么特殊的格式时，还有其他几个 HTML 标签能够用于添加特殊的格式。下面是一些这种标签。如图 6.1 所示，程序清单 6.1 演示了每个标签的运行效果。

- `<small></small>`——小字体。
- `<big></big>`——大字体；在 HTML 5 中不存在，因为文本大小由 CSS 更好地控制。
- `<sup></sup>`——上标文本。
- `<sub></sub>`——下标文本。
- `<em>`或`<i>`——强调的（斜体）文本。
- `<strong>`或`<b>`——体。
- `<tt></tt>`——等宽字体（打字机字体）。在 HTML 5 中不存在，因为字体外观由 CSS 更好地控制。
- `<pre></pre>`——等宽字体，保留空格和分行。

**警告：**原来还有`<u>`标签，用于定义带下划线的文本，但现在由于两个原因不再使用：首先，人们会以为带下划线的文本是链接，如果遇到不是链接的下划线字体将对用户造成困惑；第二，`<u>`标签已被摒弃，也就是说 HTML/XHTML 语言中已不再使用它。Web 浏览器现在还支持`<u>`标签以及用于定义带删除线文本的`<strike>`标签，在未来的一段时间内也还会支持，但是定义带下划线或删除线的文本最好使用样式单。在 HTML 5 中，删除的文本可以由`<strike></strike>`标签对包含，这将使文本带有删除线。

**Watch  
Out!**

### 程序清单 6.1 特殊格式标签

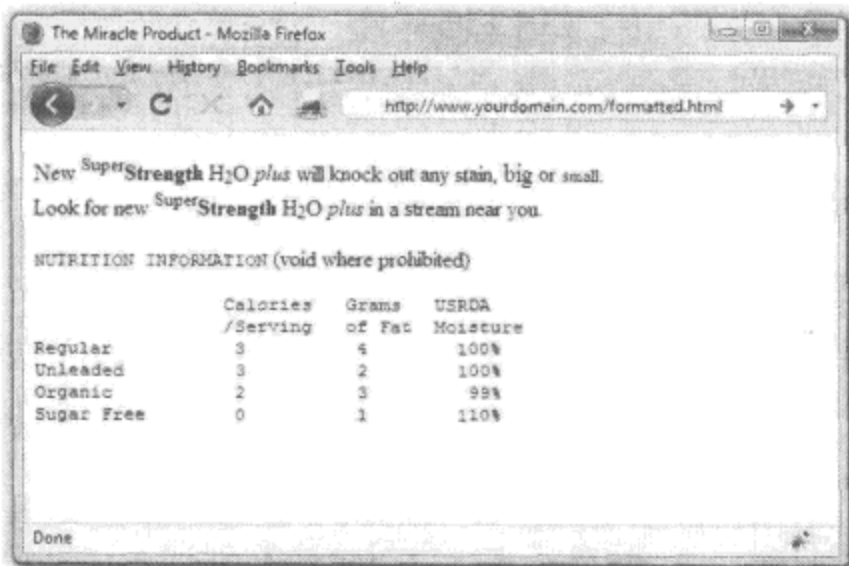
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>The Miracle Product</title>
  </head>

  <body>
    <p>
      New <sup>Super</sup><strong>Strength</strong> H<sub>2</sub>O
      <em>plus</em> will knock out any stain, <big>big</big> or
      <small>small</small>.<br /> Look for new
      <sup>Super</sup><b>Strength</b> H<sub>2</sub>O <i>plus</i>
      in a stream near you.
    </p>
    <p>
      <tt>NUTRITION INFORMATION</tt> (void where prohibited)
    </p>
    <pre>
      Calories   Grams   USRDA
      /Serving  of Fat  Moisture
Regular        3         4      100%
Unleaded       3         2      100%
Organic        2         3       99%
Sugar Free     0         1      110%
    </pre>
  </body>
</html>
```

图 6.1

程序清单 6.1 的字符格式



`<tt>` 标签通常将字体修改为 Courier New，这是一种等宽字体（等宽表示所有字母和空格的宽度都是一样的）。然而，网页浏览器允许用户将等宽的 `<tt>` 字体修改为他们选择的字体（看看你的浏览器的选项菜单）。虽然大多数人将使用浏览器设置的标准字体，但等宽字体对于有些用户来说甚至可能不一定是等宽的。

`<pre>` 将文本格式化为等宽字体，但它还有其他一些独特而有用的功能。你在第 3 章中学习过，HTML 文件中的多个空格符和换行通常被忽略，但 `<pre>` 使空格符和分行被保留。例如，如果没有 `<pre>`，图 6.1 中最后的文本将如下所示：

```
calories grams usrda /serving of fat moisture regular
3 4 100% unleaded 3 2 100% organic 2 3 99% sugar free 0 1 110%
```

即使在每行最后加上 `<br />` 标签，各列也不会对齐。然而，如果在前面加上 `<pre>`、在最后加上 `</pre>`，则各列将整齐排列，因为原来的空格被保留——不需要 `<br />` 标签。`<pre>` 标签是一种简便的方法，让你能够毫不费力地保留希望转化为网页的等宽文本文件中的对齐排列。

CSS 提供更健壮的对齐文本方法（实际上是对文本格式的所有方面），你将在第三部分学到。

## 6.2 调整字体

`<big>`、`<small>` 和 `<tt>` 标签让你能够对网页上文本的大小和外观进行一些基本控制。然而，有时候可能需要对文本的大小和外观进行更多的控制。在介绍对 XHTML 代码中的文本进行格式化的方法前，我们来简单地看看在 CSS 之前是如何完成这些工作的，因为你可能在查看其他网站的源代码时还能看到这种方法的示例。记住，旧的方法仍然在使用，这并不意味着你应该跟着做。

在样式单出现之前，现在已经逐步淘汰的 `<font>` 标签被用来控制网页文本的字体。例如，下面的 HTML 将修改网页上一些文本的大小和颜色：

```
<font size="5" color="purple">this text will be big and purple.</font>
```

可以看到，`<font>` 标签的属性 `size` 和 `color` 可以轻松地修改文本的字体。虽然这种方法可行，但它被一种更好的文本格式化方法取代，这就是 CSS 样式规则。下面是一些用于控制字体的主要样式规则。



- font-family: 设置字体。
- font-size: 设置字号。
- color: 设置字体颜色。

**注意:** 你将在第 9 章“使用颜色”中学习更多关于文本颜色的控制。第 9 章还将介绍如何创建自定义的颜色以及控制文本链接的颜色。

**By the Way**

font-family 样式规则让你能够设置用于显示文本的字体。通常，这个样式可以也应该指定多个值（用逗号分开），这样，如果用户系统中没有第一种字体，浏览器可以使用其他字体。你已经在前面的课程中看到了这点。提供备用字体集是很重要的，因为除了核心的常用基本字体（Arial、Times New Roman 等）外，每个用户可能安装不同的字体集。提供多种字体供选择，当理想的字体找不到时，还可以退而求其次，使用已知的字体。下面是一个使用 font-family 设置文本段落字体的示例：

```
<p style="font-family:arial, sans-serif, 'times roman'">
```

这个示例中有几个值得注意的地方。首先，arial 是指定的首要字体。指定字体时，大小写并不重要，因此，arial 与 Arial 或 ARIAL 相同。第二个地方是字体名 times roman 使用了单引号，因为该名称中包含空格。但是，因为“times roman”出现在通用的“sans serif”之后，所以它不太可能被使用。因为 sans-serif 在第二位，它对浏览器说“如果在这台机器上没有 Arial 字体，使用默认的 sans-serif 字体。”

样式规则 font-size 和 color 也常用于控制字号和颜色。font-size 可设置为预定义的大小，如 small、medium 或 large，也可设置为具体的点数，如 12pt 或 14pt。color 样式可设置为预定义的颜色，如 white、black、blue、red 或 green，也可设置为具体的十六进制颜色，如 #FFB499。下面的示例使用了前一个段落示例，指定了 font-size 和 color：

```
<p style="font-family:arial, sans-serif, 'times roman'; font-size:14pt;color:green">
```

**注意:** 第 9 章将介绍什么是十六进制颜色。现在，只需知道 color 样式规则可用于指定精确的颜色而不只是 green、blue 或 orange 等。

**By the Way**

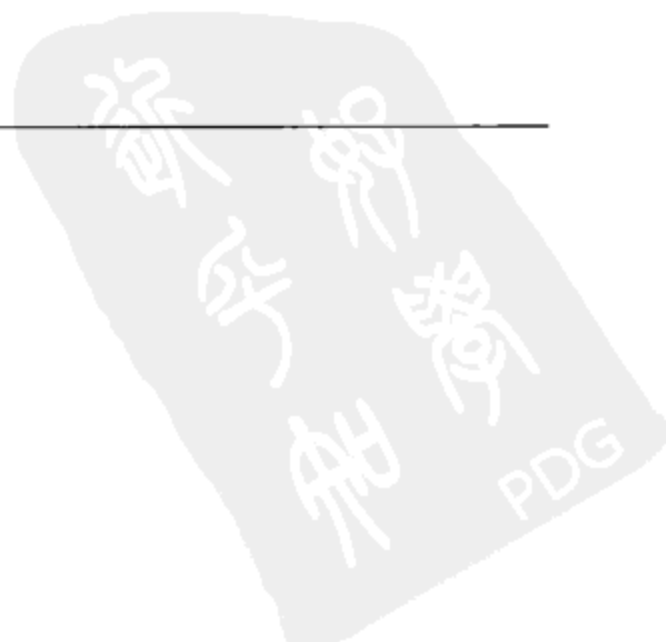
如图 6.2 所示，程序清单 6.2 中的 Web 内容使用一些字体样式规则来创建一个基本的在线简历的开头。

#### 程序清单 6.2 使用字体样式规则创建一个基本的简历

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Résumé; for Jane Doe</title>

    <style type="text/css">
      body {
        font-family: Verdana, sans-serif;
        font-size: 12px;
      }
    </style>
  </head>
</html>
```



```
h1 {
  font-family:Georgia, serif;
  font-size:28px;
  text-align:center;
}

p.contactinfo {
  font-size:14px;
  text-align:center;
}

p.categorylabel {
  font-size:12px;
  font-weight:bold;
  text-transform:uppercase;
}

div.indented {
  margin-left: 25px;
}
</style>
</head>
<body>
  <h1>Jane Doe</h1>
  <p class="contactinfo">1234 Main Street, Sometown,
  CA 93829<br/>
  tel: 555-555-1212, e-mail: jane@doe.com</p>

  <p class="categorylabel">Summary of Qualifications</p>
  <ul>
  <li>Highly skilled and dedicated professional offering a
  solid background in whatever it is you need.</li>
  <li>Provide comprehensive direction for whatever it is
  that will get me a job.</li>
  <li>Computer proficient in a wide range of industry-related
  computer programs and equipment. Any industry.</li>
  </ul>

  <p class="categorylabel">Professional Experience</p>
  <div class="indented">
    <p><span style="font-weight:bold;">Operations Manager,
    Super Awesome Company, Some City, CA [Sept 2002 -
    present]</span></p>
    <ul>
    <li>Direct all departmental operations</li>
    <li>Coordinate work with internal and external
    resources</li>
    <li>Generally in charge of everything</li>
    </ul>
    <p><span style="font-weight:bold;">Project Manager,
    Less Awesome Company, Some City, CA [May 2000 - Sept
    2002]</span></p>
    <ul>
    <li>Direct all departmental operations</li>
    <li>Coordinate work with internal and external
    resources</li>
    <li>Generally in charge of everything</li>
    </ul>
  </div>

  <p class="categorylabel">Education</p>
```



```

<ul>
<li>MBA, MyState University, May 2002</li>
<li>B.A, Business Administration, MyState University,
May 2000</li>
</ul>

<p class="categorylabel">References</p>
<ul>
<li>Available upon request.</li>
</ul>
</body>
</html>

```

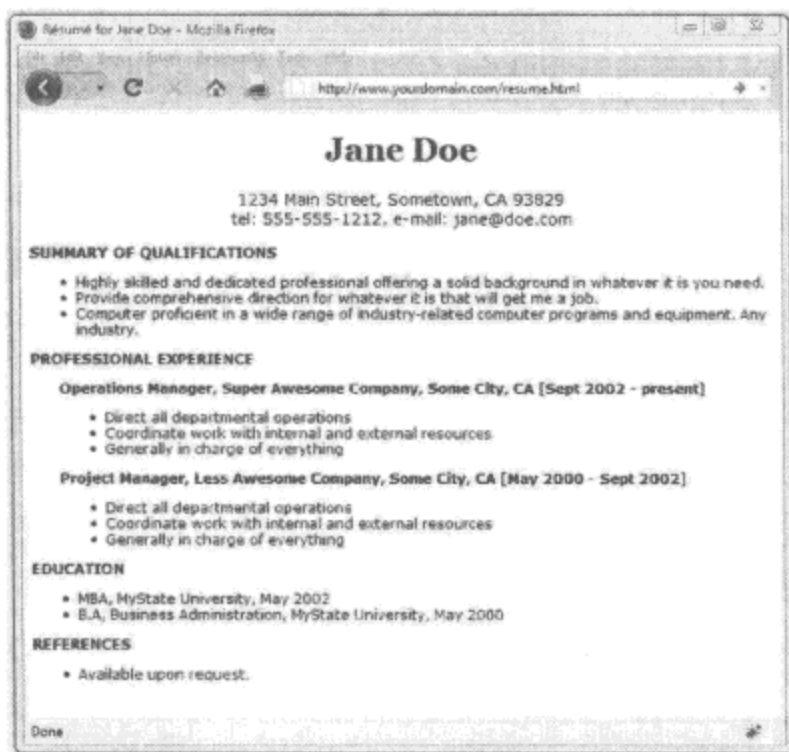


图 6.2

程序清单 6.2 代码的效果

使用 CSS，将规则集组织为类——正如你在第 4 章所学习到的——你可以看到文本格式是如何应用到内容的不同区域的。如果你仔细观察 `div.indented` 类的定义，你将会看到 `margin-left` 样式的使用。这个样式，你将在第二部分中更详细地学习，它在元素左边应用了一些空白（在这个例子中是 25 个像素）。这些空白考虑了图 6.2 中所示的缩进。

## 6.3 使用特殊字符

现在大多数字体都包括欧洲语言的特殊字符，如 Café 中的重音符号 é。还有一些数学符号和特殊的标点符号，如项目符号 •。

如表 6.1 所示，可以使用相应代码在 HTML 文档中任意处插入这些特殊字符。更完整的多字符集列表可参见 [http://www.Webstandards.org/learn/reference/named\\_entities.html](http://www.Webstandards.org/learn/reference/named_entities.html)。

例如，要插入 café，应使用如下代码：

```
caf&eacute;
caf&#233;
```

虽然可用数字来指定字符实体，但每个符号都有一个代记名称，通常更容易记。

HTML/XHTML 使用特殊代码（称为字符实体）来表示特殊字符，如 © 和 ®。字符实体总是以 “&” 开始、以 “;” 结束。表 6.1 列出了一些常用的字符实体，HTML 支持的字符实体远不止这些。

表 6.1 常用的英语特殊字符

字符	数字代码	代码名称	说明
"	&#34;	&quot;	引号
&	&#38;	&amp;	和号
<	&#60;	&lt;	小于号
>	&#62;	&gt;	大于号
¢	&#162;	&cent;	分号 (货币)
£	&#163;	&pound;	英镑符号
	&#166;	&brvbar; 或 &brkbar;	断开的竖直线
§	&#167;	&sect;	节号
©	&#169;	&copy;	版权
®	&#174;	&reg;	注册商标
°	&#176;	&deg;	度数符号
±	&#177;	&plusmn;	加或减
²	&#178;	&sup2;	上标 2
³	&#179;	&sup3;	上标 3
•	&#183;	&middot;	居中的点
¹	&#185;	&sup1;	上标 1
¼	&#188;	&frac14;	分数四分之一
½	&#189;	&frac12;	分数二分之一
¾	&#190;	&frac34;	分数四分之三
Æ	&#198;	&AElig;	大写的 AE 组合
æ	&#230;	&aelig;	小写的 ae 组合
É	&#201;	&Eacute;	重音大写 E
é	&#233;	&eacute;	重音小写 e
×	&#215;	&times;	乘号
÷	&#247;	&divide;	除号

**Did you Know?**

**提示：**想知道版权符号©和注册商标符号®的代码？这两个符号的代码分别是&copy;和&reg;。要创建没有注册的商标符号™，使用&trade;。

如图 6.3 所示，程序清单 6.3 显示了表 6.1 中一些符号的用法。

**程序清单 6.3 特殊字符代码**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Punctuation Lines</title>
  </head>
```

```

<body>
  <p>
    Q: What should you do when a British banker picks a fight
    with you?<br />
    A: &pound; some &cent;&cent; into him.
    <hr />
    Q: What do you call it when a judge takes part of a law
    off the books?<br />
    A: &sect; violence.
    <hr />
    Q: What did the football coach get from the locker room
    vending machine in the middle of the game?<br />
    A: A &frac14; back at &frac12; time.
    <hr />
    Q: How hot did it get when the police detective interrogated
    the mathematician?<br />
    A: x&sup3;&deg;
    <hr />
    Q: What does a punctilious plagiarist do?<br />
    A: &copy;
    <hr />
  </p>
</body>
</html>

```

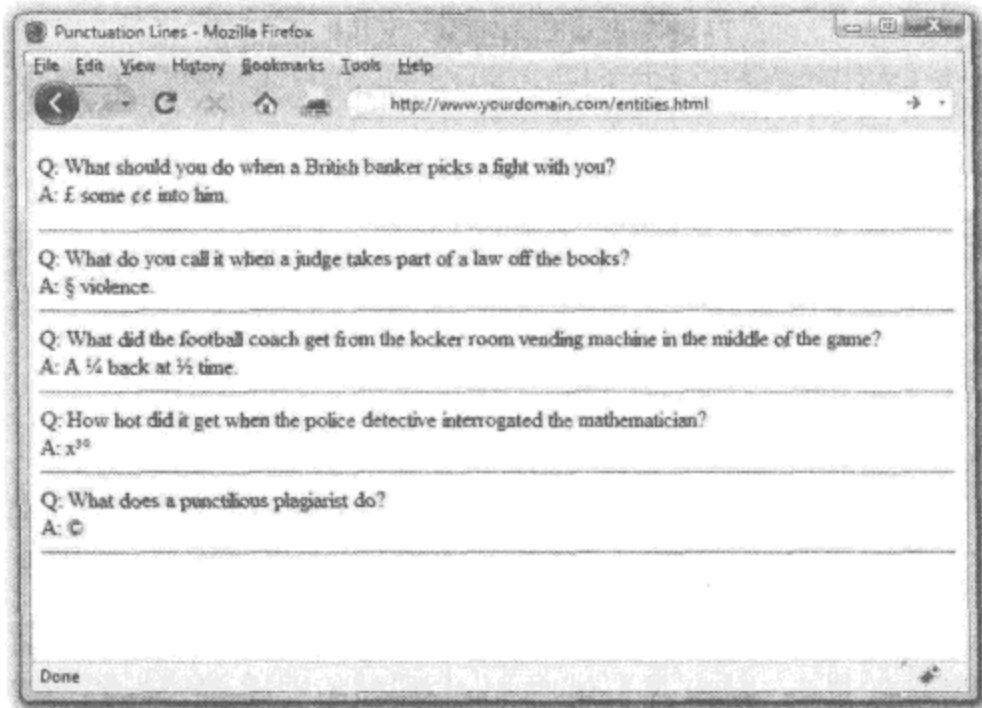


图 6.3

这是程序清单 6.3 中的 HTML 在大部分浏览器中的效果

## 6.4 总结

本章你学习了使文本为粗体或者斜体，以及上标、下标、特殊符号和重音符号的编码方法。你了解了适当地排列预格式化的等宽字体文本的段落、控制网页中文本的大小、颜色和字体的方法。

如表 6.2 所示，对本章讨论的标签和属性进行了总结。记住，所有 HTML 标记在附录 A 中列出。



表 6.2

第 6 章介绍的标签和属性

标签/属性	功能
<code>&lt;em&gt;...&lt;/em&gt;</code>	强调（通常是斜体）
<code>&lt;strong&gt;...&lt;/strong&gt;</code>	更重的强调（通常是粗体）
<code>&lt;b&gt;...&lt;/b&gt;</code>	粗体文本
<code>&lt;i&gt;...&lt;/i&gt;</code>	斜体文本
<code>&lt;tt&gt;...&lt;/tt&gt;</code>	打字机（等宽）字体
<code>&lt;pre&gt;...&lt;/pre&gt;</code>	预格式化的文本（原来的分行和空格都将被保留——这些分行和空格通常以等宽字体显示）
<code>&lt;big&gt;...&lt;/big&gt;</code>	比正常字体稍大的文本
<code>&lt;small&gt;...&lt;/small&gt;</code>	比正常字体稍小的文本
<code>&lt;sub&gt;...&lt;/sub&gt;</code>	下标
<code>&lt;sup&gt;...&lt;/sup&gt;</code>	上标
<b>属性</b>	
<code>style="font-family:typface"</code>	文本的字体名称，如 Arial（也可用于<p>、<h1>、<h2>和<h3>等）
<code>style="font-size:size"</code>	字号，可设置为 small、medium、large、x-small 或 x-large 等。也可设置为具体的点数，如 12pt
<code>style="color:color"</code>	修改文本的颜色

## 6.5 问与答

**问：**如何找到安装在计算机上的字体的准确名称？

**答：**在 Windows 或 Macintosh 计算机上，打开“控制面板”，然后单击“字体”文件夹——这里列出了系统中安装的所有字体。在 font-family 样式规则中要使用准确的字体名，但大小写无所谓。

**问：**如何在网页中使用日语、阿拉伯语、汉语以及其他非欧洲语言字符？

**答：**首先，要浏览网页的用户计算机上必须安装相应语言的字体。用户必须在网页浏览器中选择语言字符集和字体。你可以使用 Windows 中的字符映射程序（或其他操作系统中类似的程序）获得任何语言中字符的数字编码，单击“开始/所有程序/附件”，然后选择“系统工具”。如果要用的字符的编码是 214，则在网页中使用“&#214;”来表示这个字符。如果你找不到字符映射程序，使用你的操作系统的内建帮助功能找到具体的位置。

要在网页中包含亚洲语言的简短信息（如“我们讲泰米尔语——请来电！”）的最好方法是，将它作为图像包含在网页中。这样所有人都能看到它，即使用户使用英语作为浏览网页的主要语言。但是即使在图形中使用语言字体，你也很可能必须下载操作系统的特殊语言包。同样，检查系统的帮助功能来获得具体的操作说明。

## 6.6 作业

作业包含一些测试题及练习，帮助巩固对这一章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 6.6.1 测验

1. 如何使用**<b>**和**<strong>**以外的样式创建一个段落，前3个单词是粗体？
2. 如何表现水的化学方程式？
3. 如何在网页上显示“© 2009, Webwoks Inc.”？

### 6.6.2 测验答案

1. 你可以使用：

```
<p><span style="font-weight: bold">First three words</span> are  
bold.</p>
```

2. 你可以使用  $H_2O$ ：

```
<p><span style="font-weight: bold">First three words</span> are  
bold.</p>
```

3. 你可以使用以下任意一行代码：

```
&copy; 2004, Webwoks Inc.  
&#169; 2004, Webwoks Inc.
```

## 6.7 练习

- 应用本章中你学习到的字体级别样式到各种块级元素，如<p>、<div>、<ul>和<li>项目。尝试嵌套你的元素以感受样式在内容层次中的传递。



## 第 7 章

# 使用表格显示信息

本章中你将学到：

- 创建简单表格的方法
- 控制表格大小的方法
- 对齐内容及在表格中跨越行和列的方法

在本章中，你将学习建立可用于控制 Web 内容中表格化数据的间隔，布局和外观的 HTML 表格的方法。尽管你可以使用 CSS 产生相似的结果，但是有时候表格绝对是表现信息的最佳方式，你将发现表格对将信息排列到行和列中是很有用的。我还将解释设计人员在过去如何将表格用于布局，这不总是最好的思路。在我们开始之前，要记住表格是在竖向列和横向行中内容的有序排列。

### TRY IT YOURSELF ▼

#### 自己动手尝试一下：将你的内容放入表格

阅读本章时，考虑将文本排列为表格对 Web 内容的好处。阅读本章课程时，考虑将文本组织为表格对改进网页布局有帮助。请记住以下几点。

- 表格最明显的用途是用于组织表格信息，如由名称和数字组成的多列表。
- 需要将文本排成多列时，可使用表格。

你可以独立地尝试按照本章的示例添加表格。本章最后的“练习”小节提供了这方面的具体建议。

## 7.1 创建简单表格

表格由多行信息组成，每行包含单元格。要创建表格，必须以<table>标签开始。当然，表格以</table>标签结束。如果要显示表格边框，使用 border 属性来指定边框的宽度，单位为像素。边框宽度为 0 或 none（或省略 border 属性）将隐藏边框，这在使用表格作为网页布局工具时很方便。

**提示：**有些样式属性可用于更详细地控制表格边框。例如，可设置边框宽度（border-width）、样式（border-style）和颜色（border-color）。这些属性是管用的，但必须将其应用于每个表格元素，因此比较麻烦，即使你为表格行或者单元格元素使用类也一样。

**Did you  
Know?**

指定<table>标签后，接下来需要使用<tr>标签。<tr>标签创建表格的一行，包含一个或多个单元格，最后以</tr>标签结束。要创建单元格，可使用<td>标签（<td>表示表格数据—table data）。表格信息放在标签<td>和</td>之间。单元格（cell）是一个矩形区域，可容纳任何文本、图像和HTML标签。表格中的每行至少包含一个单元格。行中的单元格组成列。

创建表格时，涉及一个基本标签：<th>，它与<td>标签类似，但表示单元格是表头的一部分。大多数Web浏览器将<th>单元格中的文本居中对齐并显示为粗体。

表格中的行可以包含任意数量的单元格，但每行应包含相等数量的列。程序清单 7.1 中的HTML代码使用前面提到的4个表格标签，创建了一个简单表格。图 7.1 所示为该页面在Web浏览器中的显示结果。

#### 程序清单 7.1 使用<table>、<tr>、<td>和<th>标签创建表格

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Baseball Standings</title>
  </head>

  <body>
    <h1>Baseball Standings</h1>
    <table>
      <tr>
        <th>Team</th>
        <th>W</th>
        <th>L</th>
        <th>GB</th>
      </tr>
      <tr>
        <td>Los Angeles Dodgers</td>
        <td>62</td>
        <td>38</td>
        <td>—</td>
      </tr>
      <tr>
        <td>San Francisco Giants</td>
        <td>54</td>
        <td>46</td>
        <td>8.0</td>
      </tr>
      <tr>
        <td>Colorado Rockies</td>
        <td>54</td>
        <td>46</td>
        <td>8.0</td>
      </tr>
    </table>
  </body>
</html>
```



```

        <td>Arizona Diamondbacks</td>
        <td>43</td>
        <td>58</td>
        <td>19.5</td>
    </tr>
    <tr>
        <td>San Diego Padres</td>
        <td>39</td>
        <td>62</td>
        <td>23.5</td>
    </tr>
</table>
</body>
</html>

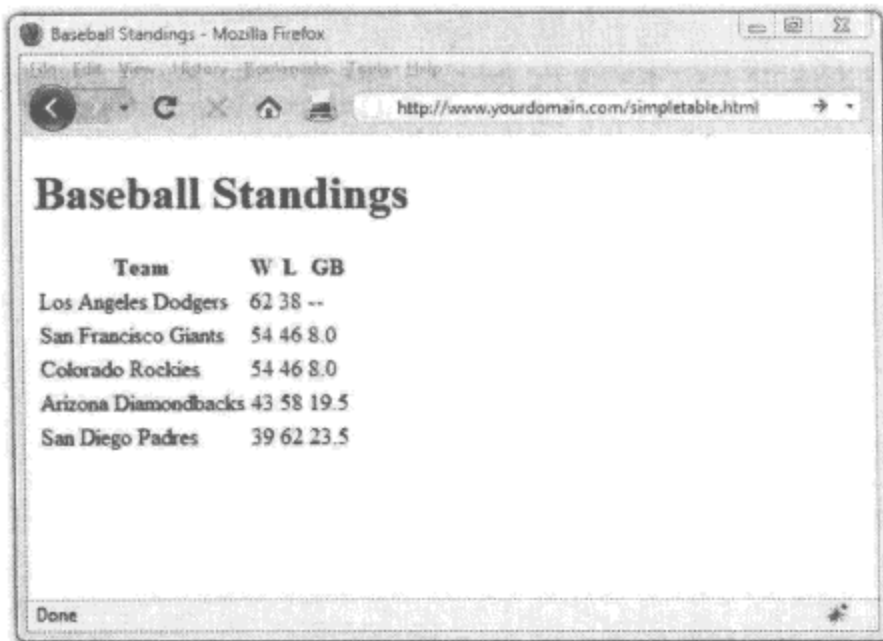
```

**Did you Know?**

**提示：**HTML 忽略单词与标签之间的空格。然而，如果像程序清单 7.1 那样使用空格来缩进 `<tr><td>` 标签，HTML 表格将更易于阅读（也不容易出现浪费时间的错误）。

图 7.1

程序清单 7.1 中的 HTML 代码创建一个四行六列的表格



Team	W	L	GB
Los Angeles Dodgers	62	38	--
San Francisco Giants	54	46	8.0
Colorado Rockies	54	46	8.0
Arizona Diamondbacks	43	58	19.5
San Diego Padres	39	62	23.5

这个表格包含棒球比赛排名，这很适合排列在行和列中——如果不是略有点简单的话。你将在本章中学习到如何使表格变得活泼。表头显示了球队、胜场数（W）、负场数（L），以及胜场差（GB）的排名。

虽然我们没有对程序清单 7.1 中的 HTML 应用任何样式，但是你可以在表格单元中使用任何文本样式。然而，一个单元格中的标签不会影响其他单元格，表格外的标签也不会影响表格中的信息。例如，请看下面的表格：

```

<p style="font-weight:bold">
  <table>
    <tr>
      <td style="font-style:italic">hello</td>
      <td>there</td>
    </tr>
  </table>
</p>

```

在这个例子中，`<div>` 标签用于包含表格，以演示表格如何能不受外部标签的影响。单词 `there` 既不是粗体也不是斜体，因为表格外的 `font-weight:bold` 和前一个单元格中的 `font-`

`style:italic` 样式都不会对它产生影响。然而，这个例子中的单词 `hello` 是斜体的。

要使单词 `hello` 和 `there` 都变成粗体，需要将表格的代码改成：

```
<table style="font-weight:bold">
  <tr>
    <td style="font-style:italic">hello</td>
    <td>there</td>
  </tr>
</table>
```

在这个例子中，两个单词都是粗体，且单词 `hello` 还是斜体。当然，可以不将样式应用于整个表格。如果需要的话，`font-weight:bold` 也可应用于单个单元格，你可以在每个单元中重复 `style="font-weight:bold"` 或者在样式单中创建一个类，并且在每个单元中使用 `class="类名"`——这是你的选择。

## 7.2 控制表格大小

表格宽度未指定时，表格及其单元格的大小会自动扩展，以适应其中包含的数据。然而，可以在 `<table>` 标签中指定 `width` 和 `height` 样式来控制整个表格的大小，也可以在每个 `<td>` 标签中指定 `width` 和 `height` 样式来控制每个单元格的大小。`width` 和 `height` 样式可用像素或百分比来指定。例如，下面的代码创建了一个宽 500 像素、高 400 像素的表格：

```
<table style="width:500px; height:400px">
```

**注意：**在 XHTML 中，不提倡使用属性 `width` 和 `height`，你可能在查看其他设计人员的代码时仍然看到它们。这两个属性在网页浏览器中仍可用，但应尽量使用样式属性 `width` 和 `height`，这是 XHTML 中的正确方法。

**By the  
Way**

要使第一个单元格占据整个表格宽度的 20%，第二个单元格占据整个表格宽度的 80%，应使用如下所示的代码：

```
<table style="width:100%">
  <tr>
    <td style="width:20%">skinny cell</td>
    <td style="width:80%">fat cell</td>
  </tr>
</table>
```

注意，表格的大小指定为 100%，这将使表格与浏览器窗口等宽。使用百分比而不是像素时，表格将自动调整大小以适应浏览器窗口的大小，同时保持你所追求的美学平衡。在这个例子中，表格中的两个单元格将分别自动调整为表格宽度的 20% 和 80%。

程序清单 7.2 中，扩展了程序清单 7.1 中的简单表格，展示了对表格单元宽度的控制。

程序清单 7.1 和 7.2 中使用的代码有两处不同。首先，程序清单 7.2 中多添加一列，这一列没有表头，但是 `<th></th>` 标签对仍然出现在第一行。在第二到第六行中，这多出来的一列包含一个图像 (`<img>` 标签)。第二个不同是，程序清单 7.2 为第一行的每个 `<th>` 元素添加了一个具体的宽度样式。第一列被定义为 36 像素宽，第二列 175 像素宽，第三、四、五列均为 25 像素宽。



## 程序清单 7.2 指定表格单元宽度

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Baseball Standings</title>
  </head>

  <body>
    <h1>Baseball Standings</h1>
    <table>
      <tr>
        <th style="width:35px;"></th>
        <th style="width:175px;">Team</th>
        <th style="width:25px;">W</th>
        <th style="width:25px;">L</th>
        <th style="width:25px;">GB</th>
      </tr>
      <tr>
        <td></td>
        <td>Los Angeles Dodgers</td>
        <td>62</td>
        <td>38</td>
        <td>-</td>
      </tr>
      <tr>
        <td></td>
        <td>San Francisco Giants</td>
        <td>54</td>
        <td>46</td>
        <td>8.0</td>
      </tr>
      <tr>
        <td></td>
        <td>Colorado Rockies</td>
        <td>54</td>
        <td>46</td>
        <td>8.0</td>
      </tr>
      <tr>
        <td></td>
        <td>Arizona Diamondbacks</td>
        <td>43</td>
        <td>58</td>
        <td>19.5</td>
      </tr>
      <tr>
        <td></td>
        <td>San Diego Padres</td>
        <td>39</td>
        <td>62</td>
        <td>23.5</td>
      </tr>
    </table>
  </body>
</html>
```



```

    </tr>
  </table>
</body>
</html>

```

Team	W	L	GB
Los Angeles Dodgers	62	38	--
San Francisco Giants	54	46	8.0
Colorado Rockies	54	46	8.0
Arizona Diamondbacks	43	58	19.5
San Diego Padres	39	62	23.5

图 7.2

程序清单 7.2 中的 HTML 代码创建的 6 行 5 列的表格, 每行有特定的宽度

还要注意, 这些宽度在后续行的<td>元素中没有重复。从技术上说, 你只需要定义第一行的宽度, 其余的行将按此进行, 因为它们都是相同表格的一部分。但是, 如果你使用另一个格式样式(比如改变字体大小或者颜色), 该样式必须对每个具有这些显示属性的元素重复。

## 7.3 表格中的对齐和跨越

默认情况下, 放在表格单元格中的内容将水平左对齐, 并垂直居中。图 7.1 和图 7.2 所示为这种默认的对齐。但是, 你可用样式属性 `text-align` 和 `vertical-align` 来控制表格单元格中的内容在水平方向和垂直方向的对齐方式。

这些属性可用于任何<tr>、<td>或<th>标签。应用于<tr>标签的对齐属性将应用于该行所有的单元格。如果表格很大, 应将这些属性应用于<tr>而不是每个<td>或<th>标签, 这样可以节省很多时间和精力。

程序清单 7.3 中的 HTML 代码使用一组文本对齐样式来将默认的对齐应用到一行, 但是覆盖几个单元格。图 7.3 所示为程序清单 7.3 的结果。

### 程序清单 7.3 表格中的对齐、单元格间距、边框和背景颜色

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Things to Fear</title>
  </head>

  <body>
    <h1>Things to Fear</h1>
    <table border="2" cellpadding="4" cellspacing="2"
      width="100%">
      <tr style="background-color:red;color:white">

```

```

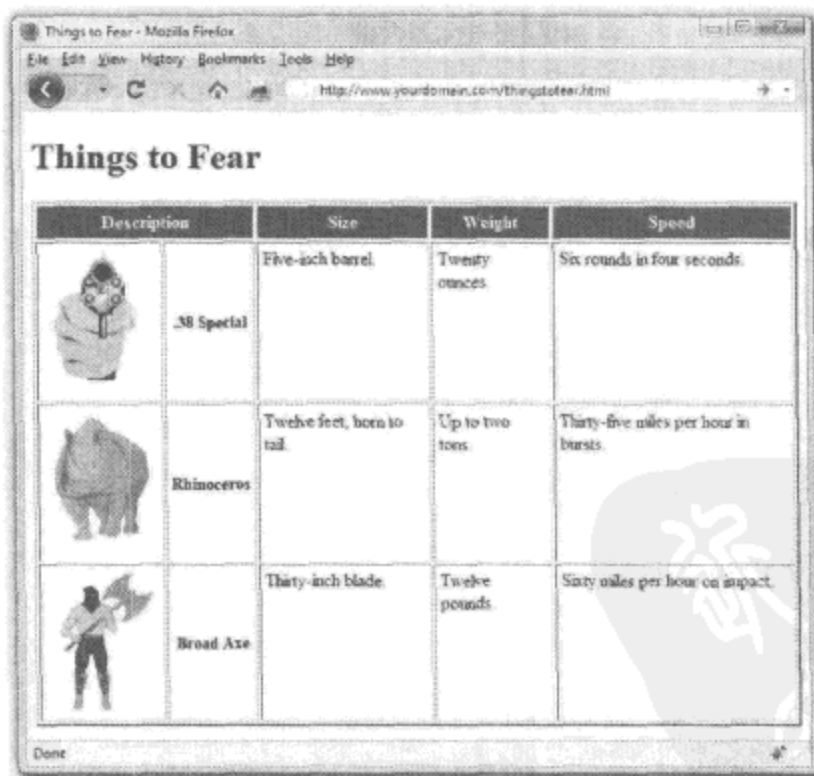
        <th colspan="2">Description</th>
        <th>Size</th>
        <th>Weight</th>
        <th>Speed</th>
    </tr>
    <tr style="vertical-align:top">
        <td></td>

        <td style="font-size: 14px;font-weight:bold;
        vertical-align:middle;text-align:center">.38 Special</td>
        <td>Five-inch barrel.</td>
        <td>Twenty ounces.</td>
        <td>Six rounds in four seconds.</td>
    </tr>
    <tr style="vertical-align:top">
        <td></td>
        <td style="font-size: 14px;font-weight:bold;
        vertical-align:middle;text-align:center">Rhinoceros</td>
        <td>Twelve feet, horn to tail.</td>
        <td>Up to two tons.</td>
        <td>Thirty-five miles per hour in bursts.</td>
    </tr>
    <tr style="vertical-align:top">
        <td></td>
        <td style="font-size: 14px;font-weight:bold;
        vertical-align:middle;text-align:center">Broad Axe</td>
        <td>Thirty-inch blade.</td>
        <td>Twelve pounds.</td>
        <td>Sixty miles per hour on impact.</td>
    </tr>
</table>
</body>
</html>

```

图 7.3

程序清单 7.3 中的代码说明了 colspan 属性和对齐样式的用法



较常用的几种 vertical-align 样式属性值为：top、middle、bottom、text-top、text-bottom 和 baseline（用于文本）。这些属性值为你带来丰富的表格数据垂直对齐方式的灵活性。

在图 11.2 的顶部，一个单元格（Description）跨越两列。这是因为该单元格的 <th> 标签指定了属性 colspan="2"。也可用 rowspan 属性来创建跨越多行的单元格。

跨越 (spanning) 是使单元格扩展到占据表格中多行或多列的过程。Colspan 属性使单元格跨越多列; rowspan 使单元格跨越多行。

**提示:** 创建含有跨越多行或多列的单元格的表格时, 记住行和列的组织结构可能是最困难的事。通常一个很小的错误就能使整个表格排列错误。编写表格的 HTML 代码前, 先在纸上画出表格, 可以节省很多时间, 避免造成困惑。

**Did you Know?**

而且, Description 列中的第二个单元格中使用的文本样式创建了水平和垂直居中的粗体文本。

程序清单 7.3 中还有一些技巧我没有介绍。可以指定整个表格——以及表格中的每行或每个单元格——的背景, 该背景可能与网页本身使用的背景不同。为此, 可在标签 <table>、<tr>、<td> 和 <th> 中指定样式 background-color 或 background-image, 就像在 <body> 标签中一样 (参见第 9 章)。例如, 要使整个表格都使用黄色背景, 可用 <table style="background-color:yellow"> 或等价的 <table style="background-color:#FFFF00">。

程序清单 7.3 中, 只有第一行有背景颜色, 使用 <tr style="background-color:red; color:white"> 在该行的各个单元格上应用红色背景。此外, color 样式确保该行的文本为白色。

**注意:** 你将常常看到表格中各行的颜色交替。例如, 某一行可能有灰色的背景, 下一行可能有白色的背景。交替的行颜色帮助用户更清晰地辨认表格内容, 特别是在表格很大时。

**By the Way**

与 background-color 类似的是 background-image 属性, 后者用于设置一个图像作为表格背景。要将图像 leaves.gif 设置为表格背景, 可用 <table style="background-image:url (leaves.gif)">。注意, 图像文件名放在括号中, 前面加上前缀 url, 表明描述的是图像文件位置。

对表格的设置不仅可以使使用 style 属性。如程序清单 7.3 所示, 可用 cellpadding 和 cellspacing 属性来控制表格边框的间距。cellspacing 属性设置表格边框之间和表格单元格之间的间距 (以像素为单位); cellpadding 属性设置单元格中的信息四周的间距 (也是以像素为单位)。如果将 cellpadding 属性设置为 0, 将使表格中的所有信息尽量接近表格边框, 甚至接触到边框。cellpadding 和 cellspacing 属性让你能够全面控制表格的外观。

**注意:** 虽然在 XHTML 中仍允许使用 cellpadding 和 cellspacing 属性, 但 CSS 中存在与这两个属性等价的样式属性 padding 和 border-spacing。附录 A 中有信息和示例。

**By the Way**

## 7.4 使用表格的页面布局

在本章的开始, 我指出了设计人员曾经在显示表格化数据之外, 还将表格用于页面布局。如果你查看其他设计人员的源代码, 将仍然会找到许多基于表格的布局的示例。这种设计方法源于旧浏览器 (1990 年代中期到 2000 年代早期) 对 CSS 支持的不一致。所有浏览器基本以相同的方式支持表格, 所以 Web 设计人员利用了基于表格的内容创建方法来完成所有浏览器中相同的视觉效果。但是, 现在所有主流浏览器对 CSS 的支持相对接近, 设计人员可以遵循长期存在的基于标准建议——不要将表格用于页面布局。

了解 Web 未来的标准组织——万维网联盟 (W3C) 提倡将样式单作为设计页面的正确方法 (而不是使用表格)。样式单远远比表格强大, 这就是有大量书籍传授使用样式单设计页面的方法的原因。

避免将表格用于布局的主要原因包括以下几点。

- 将内容与表现混合——CSS 和兼容标准的 Web 设计的目标之一是将表现层与内容层分离。
- 毫无必要地造成重新设计的困难——为了修改基于表格的布局, 你可能必须修改网站的每一页上的基于表格的布局 (除非它是一个复杂的动态驱动网站的一部分, 在这种情况下, 你必须撤销所有动态部分并且重新制作它们)。
- 可访问性问题——读屏软件在表格上查找内容, 所以常常将布局表格当作内容表格读取。
- 在移动设备上的呈现——表格布局通常不够灵活, 不能缩小以适应小的屏幕 (参见第 19 章)。

这只是基于表格的 Web 设计的一些问题。为了更仔细地了解这些问题, 可以参看流行的演示“Why Tables for Layout is Stupid”(为什么将表格用于布局是愚蠢的, <http://www.hotdesign.com/seibold/everything.html>)。

## 7.5 总结

本章读者学习了如何将文本和图像排列成由行和列组成的表格。学习了用于创建表格的 3 个基本标签以及用于控制表格的对齐方式、间距与外观的众多可选属性和样式。还看到多个表格可一起使用, 一个表格可嵌套到另一个表格中, 从而实现更多样化的布局。

表 7.1 总结了本章介绍的标签与属性。

表 7.1 第 7 章介绍的 HTML 标签与属性

标签/属性	功能
<code>&lt;table&gt;...&lt;/table&gt;</code>	创建表格, 可以包含任意数量的行 ( <code>&lt;tr&gt;</code> 标签)
<b>属性</b>	
<code>border="宽度 "</code>	指定表格边框宽度, 单位为像素, 如果 <code>border="0"</code> 或省略该属性, 边框将不显示
<code>cellspacing="间距 "</code>	表格单元格之间的间距, 单位为像素
<code>cellpadding="填充距离 "</code>	表格单元格边框同其内容的间距, 单位为像素
<code>style="width:宽度 "</code>	表格在网页中的宽度, 可以是像素数或网页宽度的百分比
<code>style="height:高度 "</code>	表格在网页中的高度, 可以是像素数或网页高度的百分比
<code>style="background-color:颜色 "</code>	表格与没有背景颜色的单元格的背景颜色
<code>style="backgroundimage:url (图像 url) "</code>	表格与没有背景图像的单元格的背景图像 (如果也指定了背景颜色, 背景颜色将透过图像的透明区域显示出来)
<code>&lt;tr&gt;...&lt;/tr&gt;</code>	定义表格的一行, 可以包含一个或多个单元格 ( <code>&lt;td&gt;</code> 标签)
<code>style="text-align:对齐方式 "</code>	该行中单元格内容的水平对齐方式, 可能的值为 <code>left</code> 、 <code>right</code> 和 <code>center</code>
<code>style="vertical-align:对齐方式 "</code>	该行中单元格内容的垂直对齐方式, 可能的值为 <code>top</code> 、 <code>middle</code> 和 <code>bottom</code>

续表

属性	
style="background-color:颜色 "	该行中所有没有背景图像的单元格的背景颜色
style="backgroundimage:url (图像 url) "	该行中所有没有背景图像的单元格的背景图像
<td>...</td>	定义表格中的一个数据单元格
<th>...</th>	定义表头单元格 (可接受的属性与样式同<td>一样)
style="text-align:对齐方式 "	该单元格内容在水平方向上的对齐方式, 可能的值为 left、right 和 center
style="vertical-align:对齐方式 "	该单元格内容在垂直方向上的对齐方式, 可能的值为 top、middle 和 bottom
rowspan="行数 "	该单元格跨越的行数
colspan="列数 "	该单元格跨越的列数
style="width:宽度 "	该列单元格的宽度, 可以是像素数或表格宽度的百分比
style="height:高度 "	该行单元格的宽度, 可以是精确的像素或表格高度的百分比
style="background-color:颜色 "	该单元格的背景颜色
style="backgroundimage:url (图像 url) "	在这个单元格中显示的背景图像

## 7.6 问与答

问: 我制作了一个大表格, 加载这个网页时, 很长时间都没有任何显示, 为什么会这样?

答: 因为网络浏览器需要计算出表格中每部分的大小才能将它显示出来, 所以复杂的表格需要一段时间才能显示在屏幕上。对表格中的每个图像指定 width 和 height 可稍微提高速度。在<table>和<td>标签中使用 width 属性也会有所帮助。

问: 我可以在表格中放置一个表格吗?

答: 是的, 你可以在其他表格单元中嵌套表格。但是, 嵌套的表格——特别是大的表格——需要花费较多的时间装入和正常显示。在创建嵌套表格之前, 考虑你放在页面上的内容并且问自己是否可以使用 CSS 来显示。可能到本书结束之前你还不知道所有答案, 但是有个提示: 在大部分情况下, 答案是“是”。

## 7.7 作业

作业包含一些测试题及练习, 帮助巩固对本章介绍的内容的理解。试着回答所有问题, 然后再看后面的答案。

### 7.7.1 测验

1. 如何创建一个简单的两行两列具有标准边框的表格?
2. 在问题 1 上扩展, 如何在表格边框和单元格之间加入 30 个像素的间距?
3. 继续问题 1 和 2 建立的表格, 如何使左上角的单元格为绿色, 右上角的单元格为红色, 左下角的单元格为黄色, 右下角单元格为蓝色?



## 7.7.2 测验问题

1. 使用如下 HTML。

```
<table border="1">
  <tr>
    <td>Top left...</td>
    <td>Top right...</td>
  </tr>
  <tr>
    <td>Bottom left...</td>
    <td>Bottom right...</td>
  </tr>
</table>
```

2. 为<table>标签添加 `cellspacing="30"`。

3. 为左上<td>标签添加 `style="background-color:green"`，为右上<td>标签添加 `style="background-color:red"`，为左下<td>标签添加 `style="background-color:yellow"`，为左上<td>标签添加 `style="background-color:blue"`。

## 7.8 练习

- 你有没有访问者有兴趣以列表或者表格形式查看的页面？使用表格表现一些表格化信息。确定每列有其表头（或者自己的图形）。尝试在本章中学到的各种对齐和间距。
- 你经常看到表格中的行交替颜色，其中一行有灰色的背景，而下一行为白色背景。表格行交替颜色的目标是快速查看充满数据的表格时能够容易分清各行。创建一个具有交替行颜色和文本颜色（如果必要）的表格。虽然第9章才学习关于颜色的课程，但是在本课中你已经有足够的信息尝试这一过程。



## 第 8 章

# 使用外部和内部链接

---

本章中你将学到：

- 链接锚的使用方法
- 在自己的网站上的页面之间链接的方法
- 链接到外部内容的方法
- 链接到一个 E-mail 地址的方法
- 在新浏览器窗口中查看链接的方法
- 用 CSS 为链接添加样式的方法

到目前为止，你已经学习到使用 HTML 标签创建一些基本页面的方法。然而，这时候的网页还是个孤岛，没有与其他信息连接（尽管在第 4 章中我偷偷地在示例中放了一些页面链接）。要使其成为“真正”的 Web 内容，需要将它与万维网连接——或者至少要与自己的个人网页或公司网页连接。

本章为你展示创建链接到自己的文档或者其他外部文档的超链接中的内容的方法。此外，你将学到为超链接添加样式，使其以彩色及你所想要的装饰显示的方法——不一定是默认蓝色下划线。

### 8.1 使用 Web 地址

要存储网站的 Web 内容，最简单的方法是将所有文件放在同一文件夹下。当文件以这种方式存储时，可以只在 <a> 标签的 href 属性中指定文件名来链接它们。

**By the  
Way**

**注意：**在我们开始之前，你可能希望复习一下你的服务器上存放文件的位置和管理一组目录中的文件的基础知识。这些信息在创建 Web 内容中的链接时很重要。参见第 2 章中的内容，特别是“理解 Web 服务器上放置文件的位置”小节。

属性 (attribute) 是与标签相关的额外信息, 进一步提供了关于标签的细节。例如, `<a>` 标签的 `href` 属性指定了你所要链接的页面的地址。

如果你有许多文件, 或者开始为网站中的内容建立一个组织结构, 就应该将文件放到目录 (或者“文件夹”) 中, 目录的名称反映其中的内容。例如, 你的所有图像应该在“image”目录中, 公司信息可以在“about”目录中等。

不管你在自己的 Web 服务器上如何组织文档, 都可以使用相对地址, 它仅包含从一个页面找到另一个页面所需要的足够信息。

相对地址描述了从一个网页到另一个网页的路径, 而不是完整 (或绝对) 的互联网地址。

回忆一下第 2 章, 你的 Web 服务器的文档根目录是 Web 内容的最高级目录。在 Web 地址中, 文档根目录由前向斜杠 (/) 来表现。所有下级目录由相同类型的斜杠来分隔。例如:

```
/directory/subdirectory/subsubdirectory/
```

### Watch Out!

**警告:** HTML 总是使用斜杠 “/” 来分隔目录。不要使用反斜杠 “\” (这通常用在 Windows) 来分隔目录。记住, Web 的一切都是向前发展的, 所以使用前向斜杠。

例如, 假设要在硬盘的文档根目录下创建一个 `zoo.html` 页面。这个文件包含两个链接, 指向位于 `Webpages` 下的 `elephants` 子文件夹中的 `african.html` 页面和 `asian.html` 页面。这些链接如下所示:

```
<a href="/elephants/african.html">Learn About african elephants.</a>  
<a href="/elephants/asian.html">Learn About asian elephants.</a>
```

这些特殊的地址被称为根相对地址, 它们是缺少完整域名的相对地址, 但是它们明确地相对于斜杠指定的文档根目录。

使用常规的相对地址, 你可以跳过开头的斜杠。这种类型的地址使链接相对于所在的任何目录——可以是根目录, 也可以是文档根以下一级或者多级的目录:

```
<a href="elephants/african.html">Learn About african elephants.</a>  
<a href="elephants/asian.html">Learn About asian elephants.</a>
```

`elephants` 子目录中的 `african.html` 和 `asian.html` 文档可以用以下的任一种方式链接回主页 `zoo.html`:

```
<a href="http://www.yourdomain.com/zoo.html">Return to the zoo.</a>  
<a href="/zoo.html">Return to the zoo.</a>  
<a href="../zoo.html">Return to the zoo.</a>
```

第一个链接是绝对链接。使用绝对链接, 链接的去向毫无疑问, 因为提供了完整的 URL——包括域名。

第二个链接是根相对链接。它相对于你当前浏览的域从而不需要协议类型 (例如 `http://`) 和域名 (如 `www.yourdomain.com`), 但是开头的前向斜杠说明地址开始于文档根目录。

在第三个链接中, 双点号 (..) 是指出包含当前文件夹的文件夹的命令——换句话说, 就是上一级目录。看到双点号时, 只要认为自己移到目录结构的上一级就对了。

如果在网页中统一使用相对地址, 将页面移到另一个文件夹、硬盘或 Web 服务器时, 就不必修改链接。

**提示：**关于使用相对地址（`elephants/african.html`）还是绝对地址（`http://www.takeme2thezoo.com/elephants/african.html`），有一个通用规则，那就是在链接存储在一起的文件时应用相对地址，链接到其他地方——其他计算机、其他硬盘或 Internet 的其他网站上的文件时，应使用绝对地址。

**Did you Know?**

必要时相对地址也可以用于跨越复杂的目录结构，第 23 章提供了关于组织和链接大量网页的更多建议。

### 自己动手试一试：

你现在可能已经随着课程的进行创建了一两个自己的网页。按照如下的步骤添加一些页面并且互相链接。

1. 使用一个主页作为进入其他网页的入口和枢纽。如果已经创建了关于自己或公司的网页，将它作为主页。也可以现在创建一个新网页，将其作为主页。
2. 在主页上，建立一些指向已经建立的 HTML 文件的链接（或者用于你计划很快建立的 HTML 文件的占位符）。在每个链接中一定要准确地拼写文件名，包括大小写。
3. 在除了主页的其他页面上，在底部（或者顶部）包含一个回到主页的链接。这使网站上的导航变得简便。
4. 你可能还想要在主页或者独立的链接页面上包含一些指向相关或者感兴趣的网站的链接。人们通常在自己的主页上包含一些朋友的个人主页。而公司对此应该小心，以免使潜在客户很快地转到其他网站——他们不一定能记住在你自己的页面中使用相对地址，在链接到其他网站时使用绝对地址。

TRY IT YOURSELF ▼

## 8.2 使用链接锚在页面中链接

`<a>` 标签是负责 Web 上的超链接的标签，它的名称来自于“anchor”（锚）一词，意思是链接是网页上的一个特定的位置。在本书目前为止的示例中，你已经学习到了用 `<a>` 标签来链接到别处的方法，但是这只是它的用处的一半。让我们从链接到相同页面中的内容开始。

### 8.2.1 用链接锚指定页面中的位置

`<a>` 标签可以用来将页面上的某个位置标记为锚，是你可以创建一个指向具体位置的链接。本章稍后介绍的程序清单 8.1 演示指向页面中一个锚的链接。为了看到这样的链接的制作方法，让我们先来看看程序清单中的第一个 `<a>` 标签：

```
<a id="top"></a>
```

`<a>` 标签一般使用 `href` 属性来指定超链接的目标，`<a href>` 是你所点击的，`<a id>` 是点击后所转向的位置。在这个示例中，`<a>` 标签仍然指定一个目标，但是没有创建实际的链接。这个 `<a>` 标签给该标签出现的页面位置取一个名字。必须包含 `<a>` 标签并为 `id` 属性赋一个唯一的名称，但是在 `<a>` 和 `</a>` 之间没有必要有任何文字。

**By the Way**

**注意:** 旧版本的 HTML 使用 name 而不是 id。较新版本的 HTML 和 XHTML 废除了 name 而代之以 id。

## 8.2.2 链接到锚位置

程序清单 8.1 显示了一个页面上放置了各种锚接点的网站。以最后一个<a>标签为例:

```
<a href="#top">Return to Index. </a>
```

#符号意味着 top 指向当前文档中的命名锚接点而不是单独的页面。当用户单击 Return to Index, Web 浏览器显示以<a id="top">标签开头的部分。

### 程序清单 8.1 使用<a>标签和 id 属性设置锚接点

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Alphabetical Shakespeare</title>
  </head>

  <body>
    <h1><a id="top"></a>First Lines of Shakespearean Sonnets</h1>
    <p>Don't you just hate when you go a-courting, and you're down
      on one knee about to rattle off a totally romantic Shakespearean
      sonnet, and zap! You space it. <em>"Um... It was, uh... I think it
      started with a B..."</em></p>
    <p>Well, appearest thou no longer the dork. Simply refer to this page,
      click on the first letter of the sonnet you want, and get an instant
      reminder of the first line to get you started. <em>"Beshrew that
      heart that makes my heart to groan..."</em></p>
    <h2 style="text-align:center">Alphabetical Index</h2>
    <h3 style="text-align:center">
      <a href="#A">A</a> <a href="#B">B</a> <a href="#C">C</a>
      <a href="#D">D</a> <a href="#E">E</a> <a href="#F">F</a>
      <a href="#G">G</a> <a href="#H">H</a> <a href="#I">I</a>
      <a href="#J">J</a> <a href="#K">K</a> <a href="#L">L</a>
      <a href="#M">M</a> <a href="#N">N</a> <a href="#O">O</a>
      <a href="#P">P</a> <a href="#Q">Q</a> <a href="#R">R</a>
      <a href="#S">S</a> <a href="#T">T</a> <a href="#U">U</a>
      <a href="#V">V</a> <a href="#W">W</a> <a href="#X">X</a>
      <a href="#Y">Y</a> <a href="#Z">Z</a>
    </h3>
    <hr />
    <h3><a id="A"></a>A</h3>
    <ul>
      <li>A woman's face with nature's own hand painted,</li>
      <li>Accuse me thus, that I have scanted all, </li>
      <li>Against my love shall be as I am now</li>
      <li>Against that time (if ever that time come) </li>
      <li>Ah wherefore with infection should he live, </li>
      <li>Alack what poverty my muse brings forth, </li>
      <li>Alas 'tis true, I have gone here and there, </li>
      <li>As a decrepit father takes delight, </li>
      <li>As an unperfect actor on the stage, </li>
      <li>As fast as thou shalt wane so fast thou grow'st, </li>
```



```

</ul>
<p><a href="#top"><em>Return to Index.</em></a></p>
<hr />
<!-- continue with the alphabet -->
<h3><a id="Z"></a>Z</h3>
<p>(No sonnets start with Z.)</p>
<p><a href="#top"><em>Return to Index.</em></a></p>
</body>
</html>

```

**注意：**靠近程序清单 8.1 的结尾，你将看到这样一行：

```

<!-- continue with the alphabet -->

```

这段文本（一个 HTML 注释）将出现在你的源代码中，但是不会被浏览器显是。你可以在第 23 章中学习到更多关于代码注释的内容。

**By the Way**

**警告：**通过<a>标签的 id 属性指定的锚名称必须以数字或者字母开头。所以如果你想要为锚的 ID 编号，一定要以文本开始（如 photo1、photo2 等）而不要只是使用 1、2。纯数字的 ID 在浏览器中能正常工作但是不能被验证为有效的 XHTML 代码。

**Watch Out!**

如图 8.1 所示，程序清单 8.1 中的每个<a href>链接建立一个下划线链接，对应一个<a id>锚——如果我填写所有的文本。这个示例中只有 A 和 Z 能够正常工作，因为只有 A 和 Z 链接具有对应的文本，但是你可以自己填写其余部分！例如，单击图 8.1 中字母索引中的字母 Z，你将到达如图 8.2 所示的页面部分。

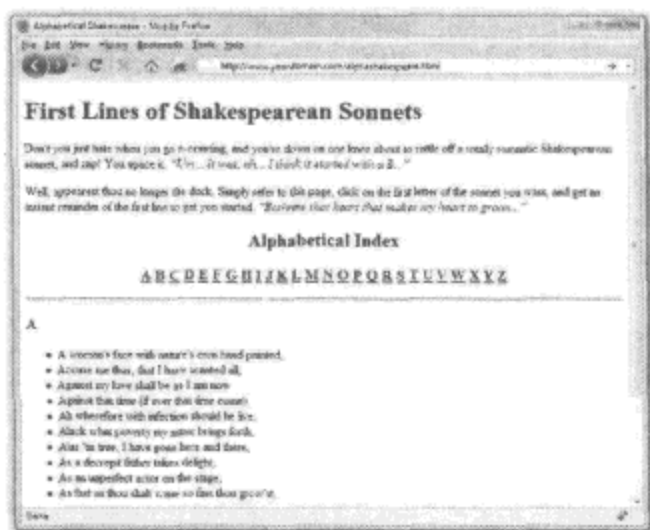


图 8.1

程序清单 8.1 中的 <a id> 标签没有完全出现在网页上。<a href> 标签以下划线链接的形式出现

掌握了链接到页面中各部分文本的概念之后，你现在可以学习链接到其他部分的 Web 内容。

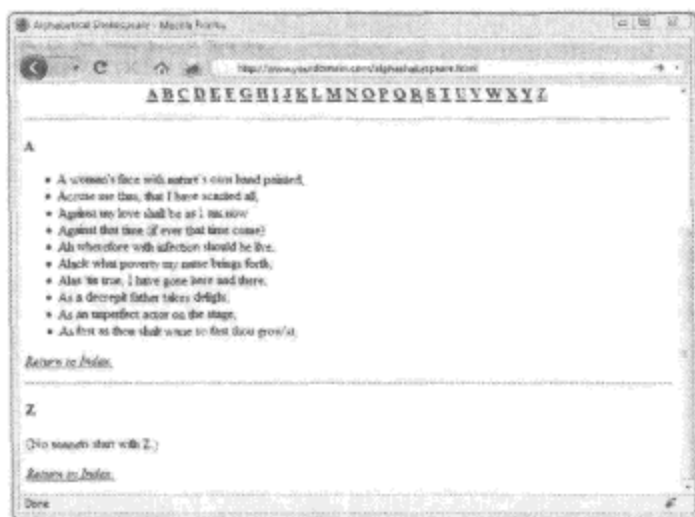
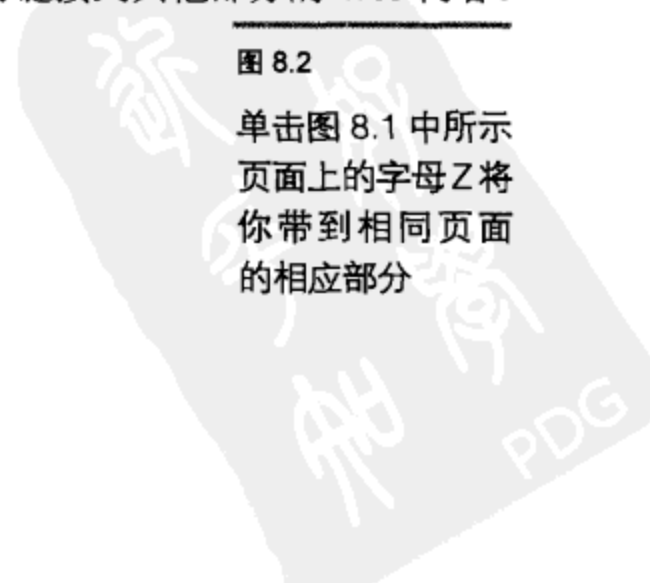


图 8.2

单击图 8.1 中所示页面上的字母 Z 将你带到相同页面的相应部分





## 8.3 在你的 Web 内容之间链接

本章前面已经学到，当你链接到你的域之中的内容（或者在本地查看网站时的同一台计算机上）时，`href` 属性指定的每个地址之前不一定要有 `http://`。当你创建从一个文件指向相同域或者相同计算机中另一个文件的链接时，你不需要指定完整的互联网地址。实际上，如果两个文件在相同的文件夹中，你可以只使用 HTML 文件名：

```
<a href="pagetwo.html"> Go to page 2.</a>
```

举个例子，如图 8.3 所示，程序清单 8.2 是一个测试页面，包含到程序清单 8.3 和如图 8.4 所示的答案页面的链接。答案页面包含一个链接返回测试页面。由于程序清单 8.2 中的页面链接到同一目录下的其他页面，因此可使用文件名而不必使用完整的地址。

### 程序清单 8.2 historyanswers.html 文件

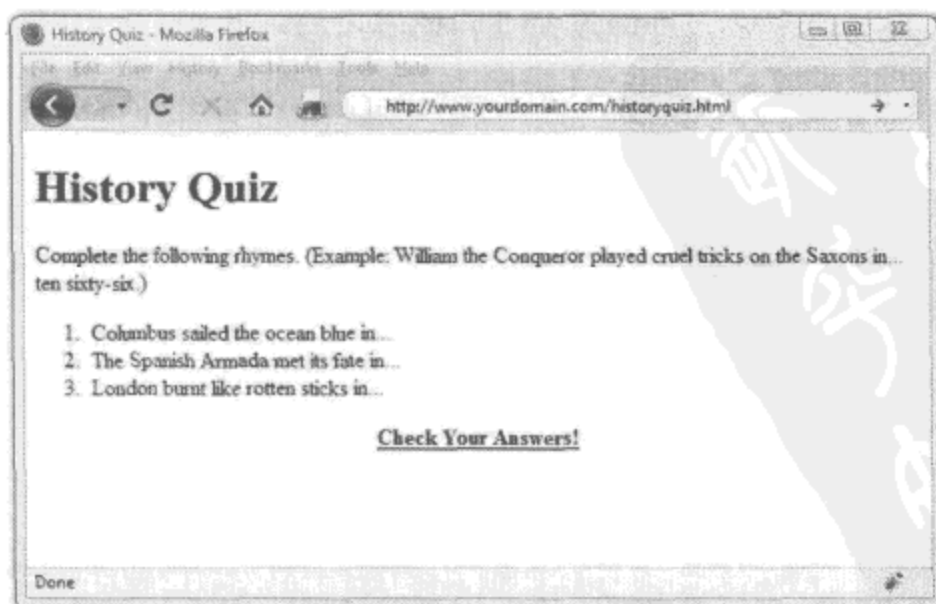
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>History Quiz</title>
  </head>

  <body>
    <h1>History Quiz</h1>
    <p>Complete the following rhymes. (Example: William the Conqueror
    Played cruel tricks on the Saxons in... ten sixty-six.)</p>
    <ol>
      <li>Columbus sailed the ocean blue in...</li>
      <li>The Spanish Armada met its fate in...</li>
      <li>London burnt like rotten sticks in...</li>
    </ol>
    <p style="text-align: center;font-weight: bold;">
      <a href="historyanswers.html">Check Your Answers!</a>
    </p>
  </body>
</html>
```

图 8.3

这是程序清单 8.2 中列出的 `historyquiz.html` 文件，被清单 8.3 中的链接所引用



### 程序清单 8.3 historyquiz.html 文件指向的 historyanswers.html 文件

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>History Quiz Answers</title>
  </head>

  <body>
    <h1>History Quiz Answers</h1>
    <ol>
      <li>...fourteen hundred and ninety-two.</li>
      <li>...fifteen hundred and eighty eight.</li>
      <li>...sixteen hundred and sixty-six.</li>
    </ol>
    <p style="text-align: center;font-weight: bold;">
      <a href="historyquiz.html">Return to the Questions</a>
    </p>
  </body>
</html>
```

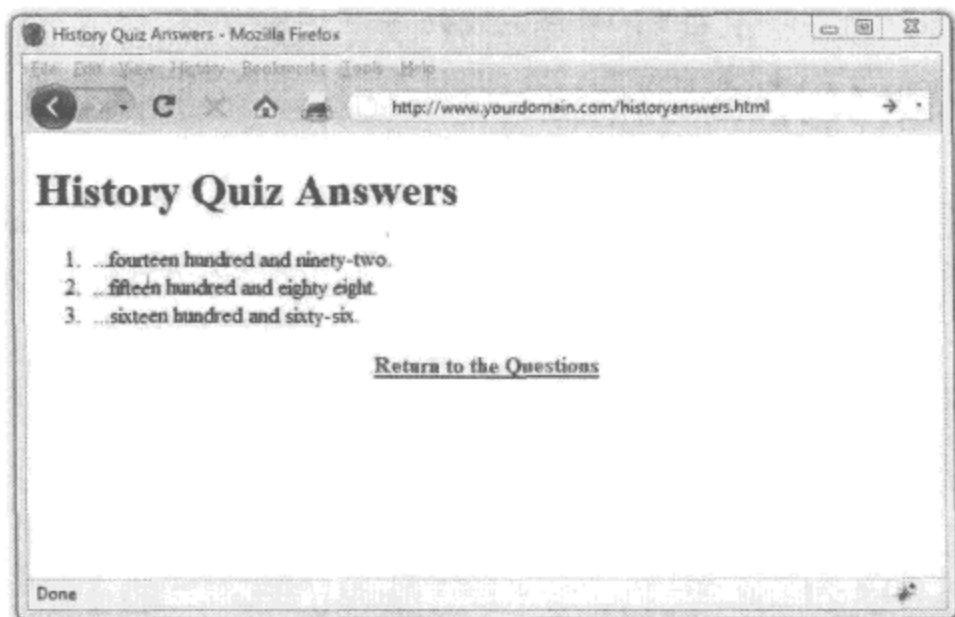


图 8.4

图 8.3 中的 Check Your Answers! 链接将你带到这个答案页面。Return to the Questions 链接将你带回到图 8.3 所示的画面

使用文件名而不是完整的 Internet 地址可以节省很多打字时间。更重要的是，网页之间的链接不管整个页面集合如何存储都可以正常工作。文件还保存在计算机硬盘上时，可以测试这些链接。你可以将这些文件存储到 Internet、CD-ROM、DVD 或存储卡上，所有链接仍能正常工作。这种识别网页的简单方法并没有什么神奇的地方——它仍与网页地址有关，正如你已经学到的。

## 8.4 链接到外部 Web 内容

链接到自己的网站中的页面和链接到外部 Web 内容的唯一区别是，链接到网站之外时，你必须包含内容的完整地址。完整的地址包括域名之前的 `http://`，然后是文件的完整路径名（如一个 HTML 文件、图像文件等）。

例如，为了从你的网页中链接到 Google，你应该在 `<a>` 链接中使用这样的绝对地址：

```
<a href="http://www.google.com/">Go to Google</a>
```

**Watch Out!**

**警告:** 你可能知道, 在大部分 Web 浏览器中, 你可以省略地址前面的 `http://`。但是, 你不能在网页上的 `<a href>` 链接中输入互联网地址时省略这个部分。

你可以应用前一小节中学到的知识来创建指向其他页面上的命名锚的链接。链接锚不仅限于相同的页面。你可以通过包含地址或者文件名, 加上 # 和锚名称链接到其他页面上的命名锚。例如, 下面的链接将把你带到域名 `www.takeme2thetoo.com` 上 `elephants` 目录中的 `african.html` 页面中的 `photos` 锚。

```
<a href="http://www.takeme2thetoo.com/elephants/african.html#photos">Check out the African Elephant Photos!</a>
```

如果你从已经在 `www.takeme2thetoo.com` 域上的另一个页面链接 (因为你实际上是网站维护者), 那么你的链接可以只是:

```
<a href="/elephants/african.html#photos">Check out the African Elephant Photos!</a>
```

你已经学过, `http://` 和域名在这个例子中不必要。

**Watch Out!**

**警告:** 仅在 `<a href>` 链接标签中使用 # 符号。不要将 # 号放在 `<a id>` 标签中, 在这种情况下, 指向那个名字的链接不能正常工作。

## 8.5 链接到一个 E-mail 地址

除了在页面之间和页面各部分之间的链接以外, `<a>` 标签允许你链接到 E-mail 地址。这是使你的网页访问者能够“顶嘴”的最简单方式。当然, 你可以提供给访问者你的 E-mail 地址并且信任他们将它输入到所用的 E-mail 程序中, 但是这增加了错误的可能性。通过提供可点击的 E-mail 地址链接, 你能够使它们毫不费力地发送信息给你并且消除打字错误的可能。

到 E-mail 地址的 HTML 链接如:

```
<a href="mailto:yourusername@yourdomain.com">Send me an e-mail message.</a>
```

`Send me an E-mail message` 将会出现, 就像其他 `<a>` 链接一样。

如果你希望人们看到你的真实 E-mail 地址 (这样他们可以记住它而用不同的 E-mail 程序来发送信息), 在 `href` 属性和 `<a>` 及 `</a>` 标签之间都包含它, 如:

```
<a href="mailto:yourusername@yourdomain.com" yourusername@yourdomain.com</a>
```

在大部分 Web 浏览器中, 单击该链接时, 会得到一个窗口, 用于输入立即发送给你的信息——自动使用人们用来发送和接收 E-mail 的程序。你可以在链接中提供更多的信息, 这样信息的主题和正文也可以有默认值。你通过在 `mailto` 链接中添加 `subject` 和 `body` 变量来完成这一工作。使用问号来分隔 E-mail 地址和变量, 变量的值使用等号, 然后用 & 符号分隔每个变量/值对。现在不需要理解变量/值术语。下面是为前一个 E-mail 示例指定主题和正文的示例:

```
<a href="mailto:author@somedomain.com?subject=Book Question&body=When is the next edition coming out?"> author@somedomain.com</a>
```

当用户单击这个链接, 创建一个以 `author@somedomain.com` 为收件人, `Book Question` 为主题, `When is the next edition coming out?` 为信息正文的 E-mail。

**提示:** 如果你希望仅仅指定 E-mail 信息主体而不是正文, 你可以省略 & 符号和 body 变量、等号和取值文本串, 如:

```
<a href="mailto:author@somedomain.com?subject=Book Question">
author@somedomain.com.</a>
```

**Did you  
Know?**

在你开始在所有页面上放置你的 E-mail 地址之前, 我必须给你一个小小的警告, 然后告诉你一个方便的技巧。你毫无疑问很熟悉垃圾邮件制造者, 它们建立 E-mail 地址数据库, 然后用垃圾邮件炸弹攻击这些地址。通过使用自动搜索网页上的 mailto 链接是垃圾邮件制造者“收获” E-mail 地址的一种方法。

幸运的是, 有一个小技巧可以挫败大部分垃圾邮件制造者。这个技巧包含了使用字符实体来编码 E-mail 地址, 这使企图“收割”你的网页上的 E-mail 地址的“铲土机”程序迷惑。以 E-mail 地址 jcmeloni@gmail.com 为例, 如果你用等价的字符实体代替地址中的字符, 就能摆脱大部分 E-mail 收割程序。小写的 ASCII 字符实体以 &#97 开始, 对字母来说按照字典顺序递增。例如, 字符 j 为 &#106, c 为 &#99, 以此类推。用 ASCII 属性替换所有字符产生如下字符串:

```
<a
href="mailto:&#106;&#099;&#109;&#101;&#108;&#111;&#110;&#105;&#064;&#103;
&#109;&#097;&#105;&#108;&#046;&#099;&#111;&#109;">Send me an email
message.</a>
```

因为浏览器将这些字符编码解释为字符, 所以从浏览器的角度看结果相同。但是, 自动化的 E-mail 收获程序搜索页面的 HTML 原始代码, 在这种情况下显示相当混乱的 E-mail 地址。如果你不想自己计算地址的字符编码, 只要在搜索引擎中输入 E-mail address encoder, 就会找到一些生成编码字符串的在线服务。

**提示:** 习惯上将网页作者的 E-mail 链接放在每个网页的底部。这不仅使其他人容易联络你, 而且给他们一个途径, 告诉你页面测试中没有发现的问题。不要忘记使用 E-mail 地址字符实体技巧, 这样你的地址可以躲过垃圾邮件制造者。

**Did you  
Know?**

## 8.6 在新浏览器窗口里打开链接

现在你已经知道如何创建链接地址——内部（在你的网站中）和外部（指向其他网站），还有一种链接方法：强迫用户在新的窗口中打开链接。

你肯定听说过弹出窗口，它是一个不经过用户同意就打开和显示的新浏览器窗口——一般是产品或者服务的广告。但是，打开另一个窗口或者指向其他位置的概念在有些情况下确实有用。例如，你可能希望在较小的辅助浏览器窗口里提供信息，但是允许用户仍然看到主窗口中的信息。这种情况往往发生在单击一个动画演示、电影片段或者其他多媒体元素时。你还可能在链接到网站外的内容时选择新的浏览器窗口。

但是，为你的用户打开新的浏览器窗口——特别是一个全尺寸的新窗口——和一些可用性和可访问性准则相悖。当用户打开新窗口时，一般使用 <a> 标签的 target 属性完成。target 属性在严格的 XHTML1.1 规范中已经被删除。

有一些方法能够达到同样的效果，同时仍然坚持可用性和可访问性准则，但是这些方法需要一些 JavaScript 和其他高级技术。你将在第 18 章中学习到这些方法，第 18 章还将介绍兼容标准和可访问的调用打开外部链接的新窗口的方法。

## 8.7 使用 CSS 设置超链接样式

网页上基于文本的超链接的默认显示是带下划线的蓝色文本。你可能已经注意到，以前访问过的链接显示为带下划线的紫色文本——这个颜色也是默认的。如果你曾经在网上打发过时间，还将会注意到并不是所有链接都是蓝色或者紫色的——为此，我们都深感欣慰。使用一点 CSS 和各种用于链接的伪类的知识，你可以使链接成为所希望的样子。

### By the Way

**注意：**你可以在开始标签和结束标签之间放置标签，使用图形（代替文本）作为链接。你将在第 11 章中学习到图形作为链接的方法。

伪类（pseudoclass）是一种描述应用到某些情况的元素样式的类，比如用户与该元素的各种交互状态。

例如，标签常用的伪类是 link、visited、hover 和 active。你可以使用助记符“Love-Hate”（爱恨）——LVHA 来记住它们。

- a:link 描述未被访问过的超链接的样式。
- a:visited 描述访问过并且存在于浏览器内存的超链接样式。
- a:hover 描述用户的鼠标悬停（单击之前）下的超链接的样式。
- a:active 描述被点击但未被释放时超链接的样式。

例如，我们假定你想要以如下样式建立一个链接：

- 字体为加粗的 Verdana（没有下划线，意味着没有文本修饰）。
- 基本颜色为浅蓝色。
- 用户悬停或者正在点击它时为红色。
- 用户访问过后为灰色。

你的样式单条目如下：

```
a {
    font-family: Verdana, sans-serif;
    font-weight: bold;
    text-decoration: none;
}
a:link {
    color: #6479A0;
}
a:visited {
    color: #CCCCCC;
}
a:hover {
    color: #E03A3E;
}
a:active {
    color: #E03A3E;
}
```



**注意：**这个例子中的颜色由 16 进制数值表示，第 9 章中将学习到它。

**By the  
Way**

因为这个链接示例不管状态如何都将是加粗的 Verdana 字体（没有下划线），这 3 个属性/值对可以存在于 a 选择符的规则中。但是，因为每个伪类都必须有一个具体的关联颜色，我们在代码示例中为每个伪类使用一条规则。除非明确地覆盖该规则，否则伪类继承上级规则的样式。换句话说，前一个示例中的所有伪类都是加粗的 Verdana 字体（没有下划线）。但是，如果我们将如下的规则用于 hover 伪类，文本在用户悬停时将显示为 Comic Sans 字体（如果用户实际安装了 Comic Sans 字体）：

```
a:hover {
    font-family: "Comic Sans MS";
    color: #E03A3E;
}
```

此外，因为 active 和 hover 伪类使用相同的字体颜色，你可以合并样式：

```
a:hover, a:active {
    color: #E03A3E;
}
```

程序清单 8.4 将这些代码片段放在一起，生成使用样式化伪类的页面，这段代码的结果如图 8.5 所示。

#### 程序清单 8.4 使用样式显示链接伪类

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

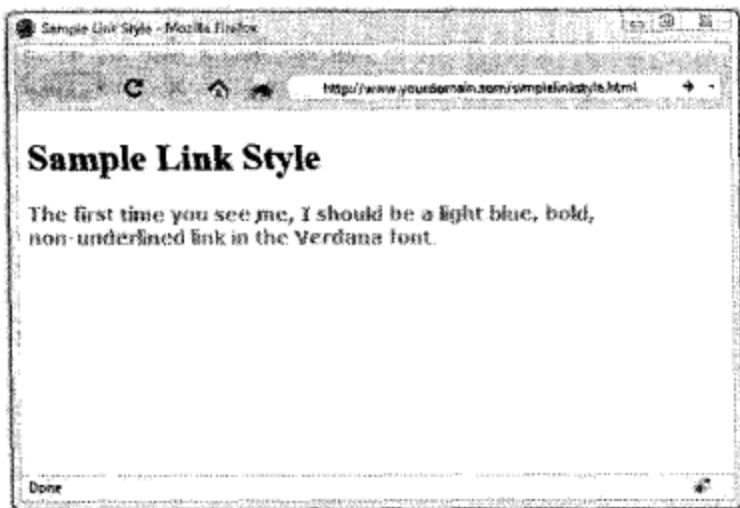
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Sample Link Style</title>

    <style type="text/css">
      a {
        font-family: Verdana, sans-serif;
        font-weight: bold;
        text-decoration: none;
      }
      a:link {
        color: #6479A0;
      }
      a:visited {
        color: #CCCCCC;
      }
      a:hover, a:active {
        color: #FF0000;
      }
    </style>
  </head>
  <body>
    <h1>Sample Link Style</h1>
    <p><a href="simplelinkstyle.html">The first time you see me, I should
      be a light blue, bold, non-underlined link in the Verdana font</a>.</p>
  </body>
</html>
```



图 8.5

链接可以使用特殊的样式来控制视觉效果



如果你在 Web 浏览器中查看该示例，链接确实应该是浅蓝色、粗体、无下划线的 Verdana 字体。如果你的鼠标在该链接上悬停，或者单击而未释放，链接应该变成红色。如果你单击并且释放链接，页面将重新装入，因为该链接指向同名的文件。但是，这时候该链接应该在浏览器的存储之中，因而被显示为访问过的链接——它将显示为灰色而不是蓝色。

你可以使用 CSS 为链接应用大范围的文本相关修改。你可以修改字体、大小、深浅、修饰等。有时候你可能希望在样式单中有许多不同的链接样式集。在这种情况下，你可以创建类，<a>标签不仅限于使用一个样式集。如下的示例是一组 footerlink 类的样式单规则，用于放在我的网页的页脚区域的链接：

```
a.footerlink {
    font-family: Verdana, sans-serif;
    font-weight: bold;
    font-size: 75%;
    text-decoration: none;
}
a.footerlink:link, a.footerlink:visited {
    color: #6479A0;
}
a.footerlink:hover, a.footerlink:active {
    color: #E03A3E;
}
```

你在下面这个示例中可以看到，类名（footerlink）出现在选择符（a）之后，由句点分隔，在伪类名（hover）之前，用冒号分隔：

```
selector.class:pseudoclass
a.footerlink:hover
```

花点时间阅读附录 A，可以了解能够应用到链接上的样式。

## 8.8 总结

<a>标签实现了“超文本”。使用这个标签，可以创建可单击的页面间链接，以及指向页面上特定锚点的链接。本章集中介绍了使用相对路径或绝对路径来标识网页，创建到其他页面的简单链接。

创建指向其他人的网页的链接时，应在<a href>标签中包含每个网页的完整 Internet 地址。对于指向自己网页的链接，只要包含文件名或足够的目录信息。

表 8.1 总结了本章介绍的<a>标签。

表 8.1

第 8 章介绍的 HTML 标签及其属性

标签/属性	功能
<code>&lt;a&gt;...&lt;/a&gt;</code>	使用 href 属性可创建指向另一个文档或锚点的链接；使用 id 属性，创建一个可被链接的锚点
属性	
href="address"	要链接的文档或锚点的地址
Id="name"	在文档中指向该锚点的名称

## 8.9 问与答

问：如果链接到 Internet 上的一个网页，然后该网页的拥有者将这个网页移走或删除了，将发生什么情况？

答：这取决于外部网页的维护者设置 Web 服务器的方式。通常，你将看到网页无法找到的信息或者其他效果。你仍然可以单击返回按钮回到上一页。作为网站维护者，你可以定期运行链接检查程序来确定内部和外部链接有效。一个示例是 Link Checker 服务(<http://validator.w3.org/checklink>)。

问：链接在我的计算机上工作得好好的，但当将网页放到网上后，链接就不能正常使用了。这是为什么？

答：下面是最有可能的一些原因。

- 大小写错误。在 Windows 计算机上，使用 `<a href="myfile.html">` 链接文件名为 N、MyFile.html 的页面是可行的。在大多数 Web 服务器上，链接必须为 `<a href="MyFile.html">`（或将文件名改为 `myfile.html`）。更糟糕的是，有些文本编辑器和文件转换程序在用户不知情的情况下修改了文件名的大小写。最好的解决方案是所有网页的文件名都采用小写。
- 文件名包含空格。大多数 Web 服务器不允许文件名包含空格。例如，不应将网页命名为 `my page.html`，而应命名为 `mypage.html` 或 `my_page.html`。
- 本地绝对地址。如果在链接中使用了本地绝对地址，如 `C:\myWebsite\news.html`，将网页放到 Internet 上时，链接将无效。不应使用本地绝对地址，本地绝对地址通常是因创建临时链接来测试网页的一部分而产生的。因此，将网页发布到 Web 前，务必删除这样的测试链接。

问：我可以在同一个 `<a>` 标签中使用 href 和 id 吗？这有什么理由吗？

答：可以，如果你有一个命名的锚点，紧接着有一个链接，这能节省你的输入时间。但是分别使用 `<a href>` 和 `<a id>` 一般更好，可以避免混乱，因为它们在 HTML 文档中扮演非常不同的角色。

问：如果我不小心错误拼写锚的名称或者忘记在前面放上 #，会发生什么？

答：如果你链接到页面中不存在的一个锚或者错误拼写了锚名称，链接将到达页面的顶部。

## 8.10 作业

作业包含一些测试题及练习，帮助巩固对这一章介绍的内容的理解。试着回答所有问题，

然后再看后面的“答案”小节。

### 8.10.1 测验

1. 你小学时最好的朋友通过 Internet 找到了你，并告诉你他想要和你交换主页链接。你如何在网页上添加一条链接，指向他的网页 [www.supercheapsuits.com/~billybob/](http://www.supercheapsuits.com/~billybob/)?
2. 使用什么样的 HTML 能够使其他人单击页面顶端的“About the Authors”时，会跳转到页面其他地方的一个致谢列表？
3. 如果你的 E-mail 地址为 [bon@soir.com](mailto:bon@soir.com)，你如何建立文本链接“goodnight greeting”，单击它可以编写并且发送给你一个 E-mail？

### 8.10.2 测验答案

1. 将下列代码加入网页：

```
<a href="http://www.supercheapsuits.com/~billybob/">  
Billy Bob's site </a>
```

2. 在页面顶端输入：

```
<a href="#credits"> About the Authors</a>
```

在致谢部分的开头输入：

```
<a id=" credits"></a>
```

3. 在网页上输入：

```
Send me a <a href="mailto: bon@soir.com">goodnight greeting</a>!
```

## 8.11 练习

- 创建一个网页，网页包含你喜欢的网站列表。你可能已经将这些网站收藏在网页浏览器中，如果是这样，可以访问这些网站，在浏览器的地址栏中找到这些网站的地址。
- 如果你已经为网站创建了网页，通读它们并且考虑文本中是否有让人们容易联络到你的地方。在那些位置包含一个指向你的 E-mail 地址的链接。让人们联络到你并且告诉你他们的需求及对你的产品的想法的机会，多多益善——特别在你管理一个公司的时候。

# 第 9 章

## 使用颜色

---

本章中你将学到：

- 为网站选择颜色的方法
- 颜色在 Web 上的工作方式
- 使用十六进制颜色值的方法
- 使用 CSS 设置背景、文本和边框颜色的方法

目前为止本书中的所有样板页面都使用标准的白色背景和黑色文本。尽管你在网上最常看到的颜色组合是深色的文本和浅色的背景，但是这并不是必需的。在简短概述颜色选择的一些最佳方法之后，你将学习使用颜色的时机，选择颜色的方法以及在创建各种网站元素时指定颜色的方法。

### 9.1 选择颜色的最佳方法

我不能精确地告诉你在网站上使用什么颜色，但是可以帮助你理解选择颜色时应该考虑的因素。你使用的颜色可能大大地影响访问者，如果你运作的是一个电子商务网站，将会希望使用诱惑用户查看产品目录并且最终购买的颜色。最终，你希望确保明智而审慎地使用颜色，并且尊重他人。你可能会奇怪，为什么在颜色中需要“尊重”，要记住万维网是一个国际社区并且有不同的解读，例如，在日本粉红是很流行的颜色，但是在东欧国家却很不常见。相似地，在美国绿色是“钱的颜色”，但是大部分其他国家“钱的颜色”是多种颜色的，因而这种隐喻对他们来说没有价值。

除了使用文化敏感的颜色以外，其他最佳方法包括以下几点。

- 使用自然的调色板。这不意味着你应该使用泥土的色调，而是指使用在城里漫步时不经意之中自然看到的颜色——避免导致视力疲劳的过亮颜色。

- 使用小的调色板。你不需要使用 15 种不同的颜色来达到目标。实际上，如果你的页面包含 15 种不同颜色的文本和图像，你可能要重新评估试图发送的信息。最多关注三四种主要颜色和少数的补色。
- 考虑人口统计数字。你可能没办法控制人口统计数字，因而必须找到一种为所有人接受的中庸颜色。年轻人喜欢的颜色不一定为老人所欣赏，就像男人和女人以及来自不同地区和文化的人对颜色的看法也有偏差。

有了这些提示，颜色的选择似乎就有限了。事情并非如此——这只是指你应该在做出决定之前深思。在你选择的搜索引擎中搜索“色彩理论”能给你更多的思路，就像使用色环。

色环是显示颜色组织的圆形图表。显示的方法是尝试帮助你理解主色、合成色和补色之间的关系。配色方案由色环开发而来，理解配色方案能够帮助你确定网站中一致使用的调色板。例如，知道颜色关系的一些知识将使你避免在浅蓝色背景上使用橙色文本或者褐色背景上使用蓝色文本。

Web 设计中的一些常用的配色方案如下。

- 类比——色环上相邻的颜色，比如黄和绿。一种颜色占据主导地位，相似的颜色用于丰富显示效果。
- 补色——色环上相对的颜色，比如暖色（红）和冷色（绿）。
- 三元——色环上间隔相等的 3 种颜色。三元方案提供平衡的同时，允许使用丰富的颜色。

专门用于理解色彩理论的书籍和教程很多，所以在本书中继续讨论实在离题了。但是，如果你打算进行 Web 设计和开发工作，对色彩理论基础知识的理解将很有帮助。花费时间阅读一下相关的内容——网上的搜索将提供很多信息。

此外，在色环上花点时间。<http://colorshemesdesigner.com/>上的 Color Scheme Generator 软件允许你从一个基色开始，生成单色、补色、三元、四元、类比和加重类比赛色方案。

## 9.2 理解 Web 颜色

为网页指定白色以外的背景颜色比想象的还简单。例如，指定页面背景为蓝色，在<body> 标签或者用于 body 元素的样式单规则中放入 `style="background-color:blue"`。当然，你可以使用许多蓝色以外的颜色。实际上，W3C 标准中列出 16 种颜色：aqua（浅绿）、black（黑）、blue（蓝）、fuchsia（紫红）、gray（灰）、green（绿）、lime（青柠色）、maroon（栗子色）、navy（藏青）、olive（橄榄绿）、purple（紫）、red（红）、silver（银灰）、teal（鳊蓝）、white（白）和 yellow（黄）。

很明显，Web 上在这 16 种颜色之外还有许多种颜色显示。实际上，你可以使用 140 种在所有浏览器上显示效果相似的颜色名称。下面是这 140 种描述性颜色名称的一部分：azure（天蓝）、bisque（橘黄）、cornflowerblue（矢车菊蓝）、darksalmon（橙红）、firebrick（砖红）、honeydew（蜜色）、lemonchiffon（柠檬黄）、papayawhip（番木色）、peachpuff（桃红）、saddlebrown（重褐）、thistle（薊色）、tomato（茄红）、wheat（小麦色）和 whitesmoke（烟白）。



**注意：**这 140 种描述性颜色名称的完整列表，以及十六进制码和浏览器显示的颜色示例，参见 [http://www.w3schools.com/HTML/html\\_colornames.asp](http://www.w3schools.com/HTML/html_colornames.asp)。

**By the Way**

但是名称是主观的——例如，如果你查看 140 种跨浏览器的颜色名称图表，你可能无法区分 fuchsia（紫红）和 magenta（品红）。然后你将发现这两个词相关的十六进制颜色值完全相同：#FF00FF。你将在下一小节中学习关于十六进制颜色值的内容，现在，如果你希望和标准兼容并且使用超过 W3C 标准描述的 16 种颜色，应该尽可能使用十六进制值。

**提示：**应该指出，颜色名称不是大小写敏感的。所以，Black、black 和 BLACK 都是黑色，但是大部分 Web 设计人员坚持小写或者混合大小写（如果他们使用颜色名称的话，大部分设计人员将使用十六进制符号来更精细地表示颜色）。

**Did you Know?**

实际上，使用十六进制值可以表示 1600 万种颜色。但是，大部分现代计算机仅能显示 16384 种，不过这仍然比 140 或者 16 多得多。

你应该知道不是所有计算机显示器都能以相同的色调显示颜色。在你的显示器上显示出来的漂亮的浅蓝色背景在其他用户的现实其上可能更偏紫色一些。模糊的泥土颜色（例如中灰色、褐色和象牙色）可能在许多计算机显示器上产生更难以预测的结果。这些颜色根据不同的房间和时间的光线条件，甚至可能像换了一个颜色。

除了将页面的背景改成白色之外的颜色，你还可以改变文本链接的颜色，包括各种链接属性（如用户鼠标悬停于链接之上和点击链接时的颜色——正如你在上一章中学到的）。你还可以设置容器元素（如段落、div、块引用和表格单元）的颜色并且使用颜色指定元素的边框。你将在本章稍后看到颜色和容器元素示例。

有很多很糟糕的网站，其中有一些是由认真的人建立的，这没有任何嘲讽的意思。但是，如图 9.1 所示的“世界上最糟糕的网站”是特意创建的，用于说明网站设计的一些极其愚蠢的做法，特别是颜色的使用。屏幕截图还不能证明——访问并且体验网站 <http://www.angelfire.com/super/badwebs/main.htm>。

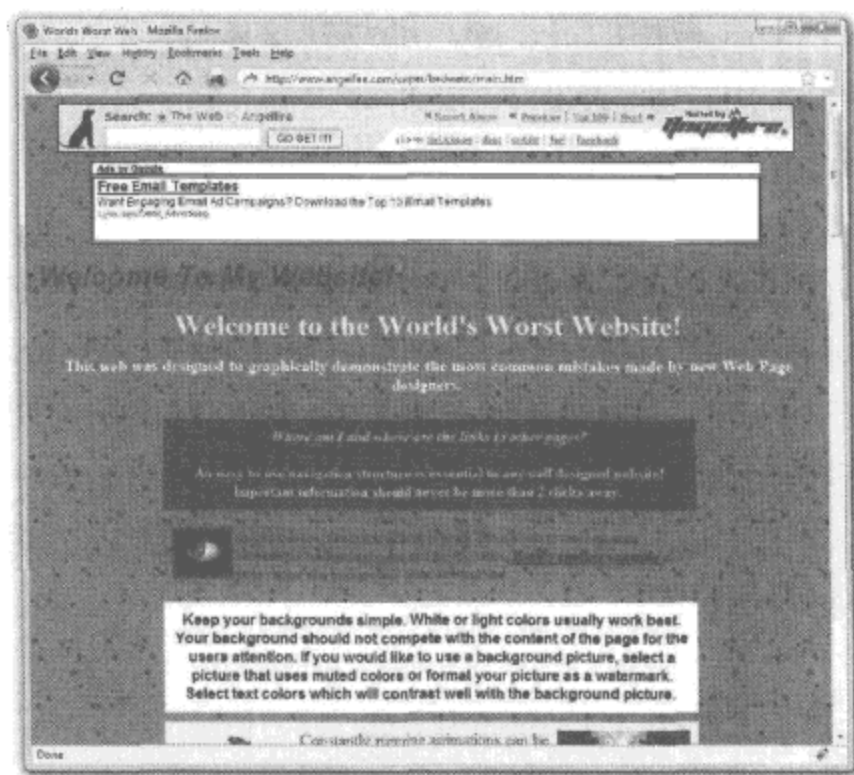


图 9.1

“世界上最糟糕的网站”的部分截图

如果你在搜索引擎中搜索“坏网站的示例”，你将找到许多收集不好的设计的网站，以



及对这些网站应该进入羞耻馆而不是荣誉馆的原因的解释。许多网站被看作“坏网站”是因为它们的显示效果，而这种效果从颜色选择开始。因此，理解颜色，以及规范和使用的微妙之处是创建好的网站的关键步骤。

### 9.3 使用十六进制颜色值

为了保持与标准的兼容，并且保持对网站颜色的精确控制，你可以通过十六进制值引用颜色。颜色的十六进制值是红、蓝、绿色光在该颜色中所占比例的表示。这有点像儿童彩色塑泥——混合所需要的红、蓝和绿色来得到合适的颜色。

十六进制颜色格式为#rrggbb，其中 rr、gg 和 bb 是两位十六进制数值，分别表示颜色的红、绿、蓝部分。如果你不熟悉十六进制数字，不要着急。只要记住 FF 最大，00 最小。在每个部分使用如下代码。

- FF 意味着最大亮度。
- CC 意味着 80%的亮度。
- 99 意味着 60%的亮度。
- 66 意味着 40%的亮度。
- 33 意味着 20%的亮度。
- 00 意味着没有这个颜色成分。

例如，大红为#FF0000，深绿色为#003300，蓝紫色为#660099，中灰色为#999999。为了建立具有红色背景和深绿色文本的页面，HTML 代码为：

```
<body style="background-color:#FF0000;color:#003300">
```

尽管这里只展示了六个两位十六进制示例，但是实际上有 256 个两位十六进制数组合——每一位从 0 到 9，A 到 F。例如，F0 是一个可能的取值（十进制为 240），62 是一个可能的取指（十进制值为 98）等。

前面已经讲过，#rrggbb 十六进制值中的 rr，gg，bb 表示的是颜色中的红，绿和蓝色成分。每个成分取值范围为 0（无此颜色）到 255（全色）。

所以白色(#FFFFFF)转换为红色值 255，绿色值 255 和蓝色值 255。相似地，黑色(#000000)转换为红色值 0，绿色值 0 和蓝色值 0。纯红色为#FF0000（全红、无绿、无蓝），纯绿色为#00FF00（无红、全绿、无蓝），纯蓝色为#0000FF（无红、无绿、全蓝）。所有其他十六进制符号转换为三种颜色各 255 种可能的取值。跨浏览器兼容的颜色名称 CornflowerBlue 与十六进制符号#6495ED 相关——红色值 100、绿色值 149 和蓝色值 237（几乎所有可用的蓝色值）。

在选择颜色时，通过图形程序或者网上你喜欢的某个图形，你可以看到十六进制或者十进制的颜色符号。如果你在搜索引擎中输入十六进制颜色转换器，将找到许多可以帮助你颜色值转换为可用于样式单的符号的程序。

### 9.4 使用 CSS 设置背景、文本和边框颜色

使用 CSS 时，可在 3 种实例中使用颜色：指定背景颜色、文本颜色或者边框颜色。前面

的章节包含了指定颜色的示例而没有详细地说明颜色符号或者色彩理论。例如，在第 8 章中，你学习到了为各种链接状态使用颜色。在第 7 章的测验中询问了如何用颜色填充表格单元。

图 9.2 所示为一个很容易进入耻辱馆的颜色使用示例。我不能想象在一个重要的网站中使用这种颜色组合和样式，但是它在这里提供了一个颜色样式可以应用到各种元素的示例。

程序清单 9.1 展示了用于生成图 9.2 的 XHTML 和 CSS 样式。

### 程序清单 9.1 使用样式生成背景，文本和边框颜色

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Background, Text, and Border Colors</title>
  </head>

  <body>
    <h1>Background, Text, and Border Colors</h1>

    <p style="background-color:#CCCCCC;
border:1px solid #000000; color:#FF0000">
Grey paragraph, black border, red text with a
<span style="color:#FFA500">orange span</span>.</p>

    <div style="width:300px; height:75px; margin-bottom: 12px;
background-color:#000000; border:2px dashed #FF0000;color:
  #FFFFFF">
Black div, red border, white text. </div>

    <table border="1">
      <tr>
        <td style="background-color: #00FF00">Green Table Cell</td>
        <td style="background-color: #FF0000">Red Table Cell</td>
      </tr>
      <tr>
        <td style="background-color: #FFFF00">Blue Table Cell</td>
        <td style="background-color: #0000FF">Yellow Table Cell</td>
      </tr>
    </table>

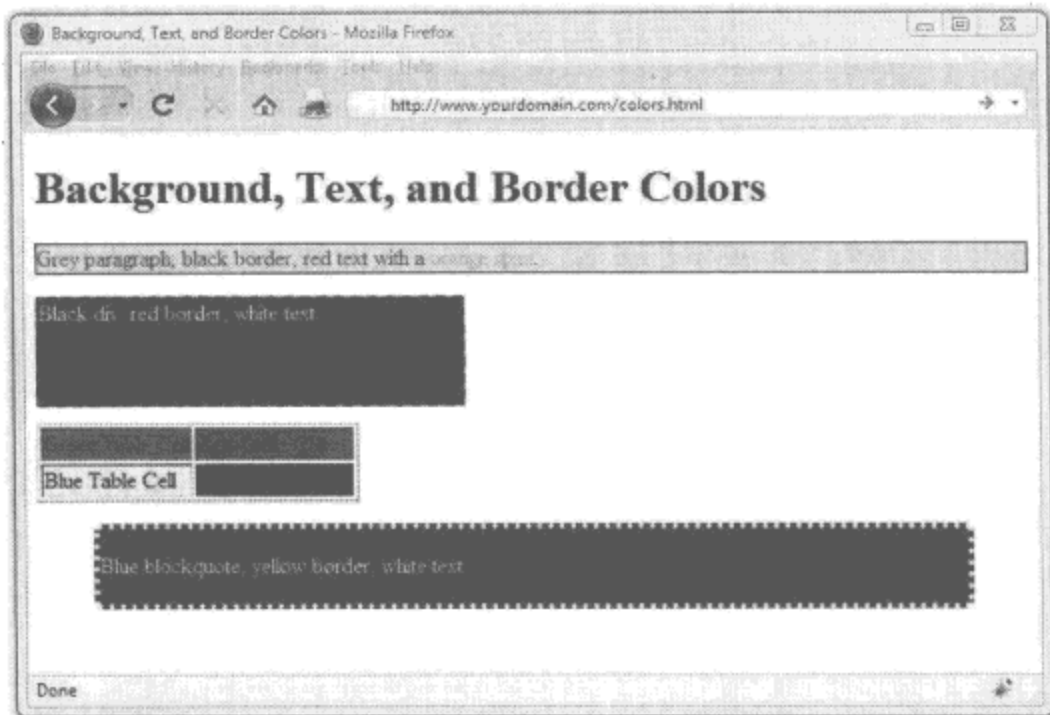
    <blockquote style="background-color:#0000FF;
border:4px dotted #FFFF00; color:#FFFFFF"><p>Blue blockquote,
yellow border, white text.</p></blockquote>
  </body>
</html>
```

查看程序清单 9.1 中使用的样式，你应该能够领会几乎所有的内容，除了一些边框样式之外。在 CSS 中，边框不能在没有任何宽度和类型的情况下定义一个颜色，在程序清单 9.1 中所示的第一个示例中，边框宽度为 2 个像素，边框类型为虚线。在程序清单中所示的第四个示例中，边框宽度为 4 个像素，边框类型为点线。

**注意：**你可以设置边框的许多特征，为元素的 4 个边框（上、下、左、右）指定不同的宽度、类型和颜色。更多信息参见附录 B “XHTML 1.1 和 CSS 2 快速参考”。

图 9.2

背景, 文本和边框颜色都由 CSS 设置



为网站选择颜色时, 要记住点滴就能成就很多——如果你真的喜欢鲜艳和显眼的颜色, 将它作为一种重点部位的颜色, 而不是所有主要设计元素的颜色。至于可读性, 要记住浅色背景的深色文本比深色背景的浅色文本容易辨认。

最后, 考虑你的一部分目标用户群可能是色盲。为了可访问性, 你可能要考虑使用 <http://colorfilter.wickline.org/> 上的 Colorblind Web Page Filter 来看看你的网站在色盲人士看来是什么样子。

## 9.5 总结

在本章中, 你学习了一些颜色使用中的最佳方法, 以及使用色环帮助查找文本补充颜色的方法。此外, 你学习了颜色的十六进制标记法——所有颜色表现为与红、绿和蓝色量的符号——十六进制符号使你能为元素应用精细的颜色。更重要的是, 你学习了 3 种与颜色相关的样式属性, 可以用于在 CSS 中应用颜色到容器背景、边框和文本。

表 9.1 总结了这些样式属性。

表 9.1

第 9 章中介绍的样式属性

标签/属性/样式	功能
<code>style="background-color:color"</code>	设置元素 (如 <code>&lt;body&gt;</code> 、 <code>&lt;p&gt;</code> 、 <code>&lt;div&gt;</code> 、 <code>&lt;blockquote&gt;</code> 和其他容器) 的背景颜色
<code>style="color:color"</code>	设置元素中文本的颜色
<code>style="border:size type color"</code>	设置元素 4 个边框的颜色。边框颜色不能在指定宽度和类型时使用

## 9.6 问与答

问: Web 浏览器不能让人们选择自己的背景和文本颜色吗?

答: 可以, Web 浏览器允许用户覆盖网页创作者指定的颜色。有些人可能将你的蓝底白字页面用白底绿字或者他们喜欢的颜色代替, 但是很少人实际使用这个选项。人们看到的通常是 `<body>` 标签中指定的颜色。

问：我听说可以使用 231 种“浏览器安全颜色”而不应该使用其它的颜色。真的吗？

答：真相是：对于以 256 色（8 位）视频模式操作计算机的用户来说，有 231 种颜色的显示较不“模糊”。一些网页创作者尝试坚持这些颜色。但是，真彩色或者高彩色计算机显示器现在已经是标准配置，它们以同样的清晰度显示所有颜色。所以，如果你的图形程序提供给你十六进制格式的颜色值，尽管将该值插入样式单来显示自定义配色方案。

## 9.7 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 9.7.1 测验

1. 如何将网页的背景设置为黑色，所有文本显示为亮绿色？
2. 为网站选择配色方案时，下列颜色关系类型中哪一个给你更多颜色选择：类比、补色还是三元？
3. 如果你在样式单中为正文元素指定 `background-color #FFFFFF`，然后在页面上第一个 `<div>` 使用 `background-color #FF0000`，`<div>` 的背景将是红色还是白色？

### 9.7.2 测验答案

1. 将如下代码放在网页开头：  

```
<body style="background-color:#000000; color:#00FF00">
```
2. 三元。它使用色环上间隔相同的 3 种颜色。
3. `<div>` 颜色将为红色，因为容器元素的背景颜色指定值将覆盖 `<body>` 标签中或者样式单中正文条目的指定值。

## 9.8 练习

- 选择你喜欢的基色——可能是可爱的蓝色或者泥土的颜色——并使用 <http://colorshemesdesigner.com/> 上的 Color Scheme Generator 来提出一组用于网站的颜色。我建议 使用四元或者加重的类比方案类型。
- 一旦你有了一组颜色——或者颜色集的一些选择——创建一个带有一个 `<h1>` 元素，一段文本，和一些列表项目的基本网页。使用本章中学习到的颜色相关样式修改页面和各种块级元素的背景颜色，以便了解这些颜色集的情况。了解它们之间的相互作用并且决定最适合于容器和普通文本、标题文本和链接文本的颜色。

## 第 10 章

# 创建用于 Web 上的图像

---

本章中你将学到：

- 选择图像软件的方法
- 准备用于网上的照片的方法
- 创建标题和按钮的方法
- 减少图像中颜色数量的方法
- 创建透明图像的方法
- 创建平铺背景的方法
- 创建 Web 动画的方法

虽然注意配色方案和生成在视觉上很有吸引力的网站是重要的，但是你没有必要成为一名专业的艺术家，在网页上放置高冲击力的图形。更重要的是，你不需要花费几十万美元在软件上。本章将帮助你开始创建可用于网站上的一些图像。尽管本章的样板图像使用 Windows、Mac 和 Linux 上的一个流行的免费图像软件（GNU 图像操纵程序，GIMP），但是你可以将本章中学到的知识应用到任何主要的 Windows 或者 Macintosh 图像应用程序——只是菜单和选项可能略有不同。

本章中，你只要学习创建图像本身，包括不同类型的图像用法。在第 11 章中，你将学习使用 HTML 和 CSS 集成图像的方法。

### 10.1 选择图像软件

可以使用任何计算机图像软件来创建网页图像，从最简单的随操作系统一起发布的免费画图程序，到昂贵的专业软件，如 Adobe Photoshop。如果计算机连接了数码相机或扫描仪，

这些仪器可能自带了图像软件，可用于创建网页图像。还有许多可免费下载的图像编辑器——甚至是 Web 应用程序——它们可以处理图像元素的操纵。

**注意：**Adobe Photoshop 无疑是图像编辑软件的翘楚。然而，它很昂贵，而且如果没有计算机图形的经验，它对你来说也非常复杂。关于 Adobe 的产品，可参见 Adobe 的网站 <http://www.adobe.com/>。如果你打算购买它们的产品，可以从网站上下载评估版本。

**By the Way**

如果觉得某个软件用于创建 Web 图像不错，可使用它来完成本章的课程。如果它无法执行这里介绍的某些功能，它可能就不是好的 Web 图像工具。在这种情况下，可以从 <http://www.gimp.org> 下载和安装 GIMP。这个全功能的图像软件是完全免费的。

1. 访问 <http://www.gimp.org/> 并且单击 Downloads（下载）链接。

2. 你应该看到指向用于你的操作系统的下载链接。你还可以单击 Show other downloads（显示其他下载）链接，它将引导你到所有可用的选项。一旦你看到可用于你的操作系统的软件，单击该链接开始下载。

3. 下载完成后，双击安装程序安装该软件。

本章中的示例使用 GIMP 演示许多必须知道的关键 Web 图像技术。当然，有许多生成图像的方法，可以使用许多不同类型的软件，我甚至无法开始解释所有的软件和技术。但是，所有主要的图像创建软件都包含详细的帮助功能、文档以及联机用户教程。使用搜索引擎来学习在一个课时中难以解释的内容。

如果 GIMP 不适合你，考虑下载 Adobe Photoshop 或者 Corel DRAW 的评估版本。仅仅对于照片处理，有许多免费的选择，它们都有很有用的特性。Google 的 Picasa 是其中之一，可在 <http://picasa.google.com/> 上免费下载。另一个是 Picnik (<http://www.picnik.com>)。这两个软件都适合于编辑图像而不是从头开始创建，Picasa 面向数字照片收藏的组织。这些软件不一定能够帮助你设计网站的横幅或者按钮图像。但是，这些软件能够帮助你处理一些辅助的图像，它们的能力值得注意。

## 使用其他网站的素材

节省创建用于网页的图像和媒体文件时间的最佳方法当然是完全不创建它们。从网页上截取图像只要简单地右键单击（在 Macintosh 中是单击并且按住鼠标按钮）并且选择 Save Image as 或者 Save Picture as（“图像另存为”，取决于浏览器）。从页面提取背景图像也同样简单：右键单击并且选择“背景另存为”。

但是，你不应该在所有者的明确授权情况下使用图像，询问所有者或者查找创意共享组织的许可。采用没有明确授权的图像是违反版权法的（也是令人反感的）。更多关于版权法的知识，访问 <http://www.utsystem.edu/OGC/IntellectualProperty/cprtindx.htm>。

你还可能考虑无版权的剪贴画，这不需要得到版权许可。好的网上剪贴画来源之一是 Microsoft Office 联机剪贴和媒体网站，位于 <http://office.microsoft.com/clipart>。另一个流行的剪贴画网站是 Barry's Clipart Server (<http://www.barrysclipart.com>)。



## 10.2 必须知道的图像基础知识

将图像和多媒体放到 Internet 上时，总有两股力量会互相冲突。用户的眼睛和耳朵希望一切都尽可能地详细和精确，又希望文件尽可能小。精致的、颜色丰富的图像意味着大文件，这即使在快速网络上也将增加传输时间。如何使多媒体的质量最大化同时又使文件最小？为进行选择，需要了解颜色和分辨率是如何营造出高质量的主观感觉的。

图像的分辨率是组成图像的点或像素（组成数字图像的点）的数量。高分辨率的大型图像通常比低分辨率的小型图像需要更长的传输和显示时间。分辨率通常以图像的宽乘以高表示，单位是像素，例如，300×200 的图像宽 300 像素、高 200 像素。

### By the Way

**注意：**有多种图像分辨率，包括像素、空间、瞬时、辐射。学习这些类型可能要花上好几个小时。如果你上过图像设计课，只管做就是了。但是现在，你需要记住的是大的图像花费更长的下载时间并且使用更大的现实空间。显示大小和存储及下载的大小是设计网站时应该考虑的因素。

你可能会觉得惊讶，分辨率并不是图像文件的大小（和传输时间）最主要的决定因素。这是因为网页上使用的图像总是以压缩格式存储和传输。图像压缩是一种数学运算，将图像中的重复模式都删除。图像压缩的数学运算很复杂，基本原理是重复模式或相同颜色的大块区域在图像存储到磁盘上时可以删除。这使图像文件变得小得多，在 Internet 上传输也更快。然后网页浏览器在显示图像时再恢复图像原来的样子。

在本章的余下部分，你将学习创建具有很大的视觉冲击力而又有小的文件尺寸的图像的方法。你将使用的技术取决于每个图像的内容和用途。Web 图像的用途和网页一样多，但是以下 4 种图像最常见。

- 人、产品或地方的照片。
- 图形横幅或徽标。
- 用于链接和操作的按钮或图标。
- 网页的背景纹理或墙纸。

## 10.3 准备照片图像

为了将照片放在网页上，你需要将印刷的照片转化为数码图像或者使用数码相机创建数字照片。你可能需要使用扫描仪或者相机自带的软件将照片存储到硬盘上，或者从照相机上直接拖放文件到硬盘上。如果你使用扫描仪创建印刷照片的数字版本，可以直接从选用的图像程序中使用任何扫描仪——详细信息参见软件文档。

### Did you Know?

**提示：**如果你没有扫描仪或者数码相机，几乎所有胶片开发商都提供将 35mm 胶片转换到 CD-ROM 或者 DVD-ROM 的服务，收费也合理。你可以从 CD-ROM 或者 DVD-ROM 将文件复制到硬盘，然后使用图像软件打开和修改图像文件。

将数码图像文件传送到计算机上后，你可以使用图像软件裁剪、改变大小，更正颜色以及压缩，使其适合于网站。

### 10.3.1 裁剪图像

网页图像应尽可能简洁，因此通常需要裁剪或缩小数码照片。裁剪照片时，选择希望显示的区域，并将其他部分裁剪掉。

#### 自己动手尝试一下：在 GIMP 中裁剪

GIMP 工具箱提供裁剪工具及可能的属性的快速访问。寻找一个图像文件——可以是来自相机并且保存在硬盘上的数码图像，或者网上找到的图像。在 GIMP 中打开图像，执行如下步骤进行裁剪。

1. 如图 10.1 所示，在 GIMP 工具箱中，单击裁剪工具，根据选择的工具，可能有额外的属性。例如，图 10.1 显示了裁剪工具的属性（如长宽比、位置、大小等）。
2. 在希望裁剪的图像中，单击希望保留的图像部分的左上角并按住左键拖曳到右下角，画出一个选择框，如图 10.2 所示。
3. 单击选择区域的一角应用裁剪。

你的图像软件可能有不同的方法，但是概念相同：选择保留的区域然后裁剪其余部分。

TRY IT YOURSELF ▼

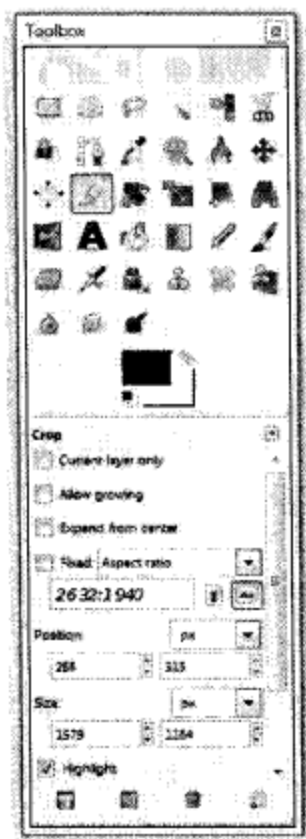


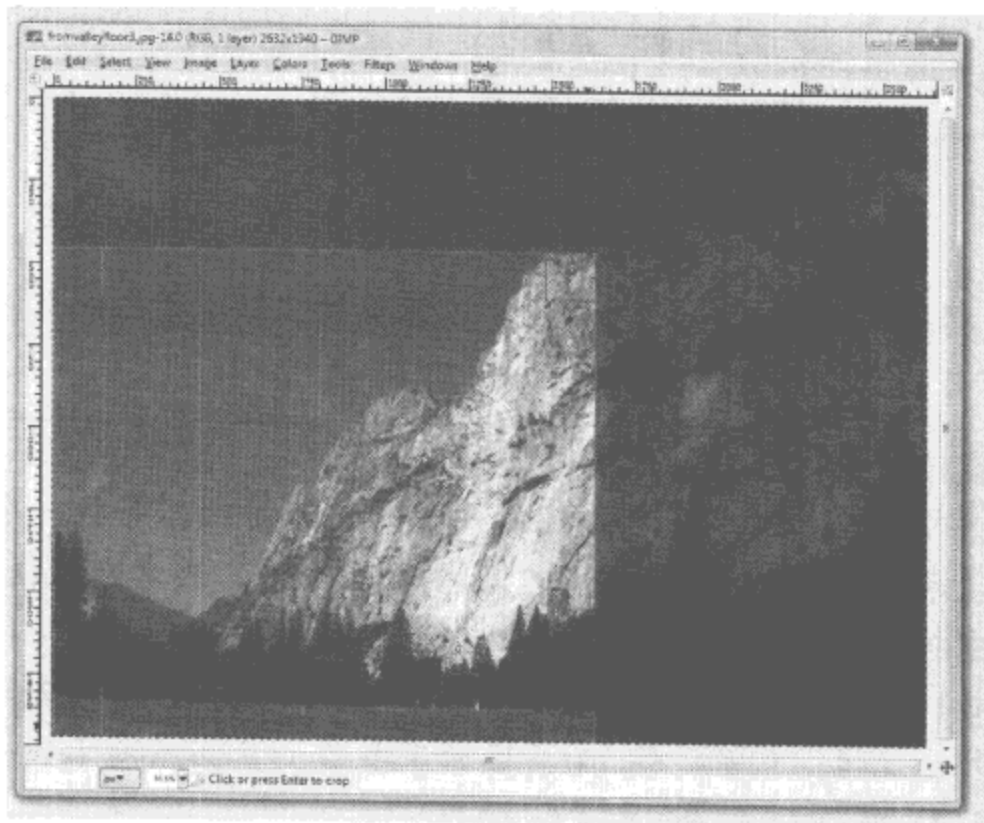
图 10.1

从工具箱中选择  
裁剪工具

即使图像经过裁剪后，它可能还是比用于网页所需的尺寸大。根据网页设计，可能需要将大图像限制在 800 像素×600 像素（如果它出现在自身的页面上，如产品目录）、640 像素×480 像素以内或更小。与文本一起显示时，图像的宽度可能在 250 像素~350 像素之间，这样才有足够的空间显示文本。在有些情况下，可能还要为图像创建一个缩略图链接，它指向大图像，缩略图的较长边可能要在 100 像素以内。

图 10.2

选择希望显示的  
图像部分



### Did you Know?

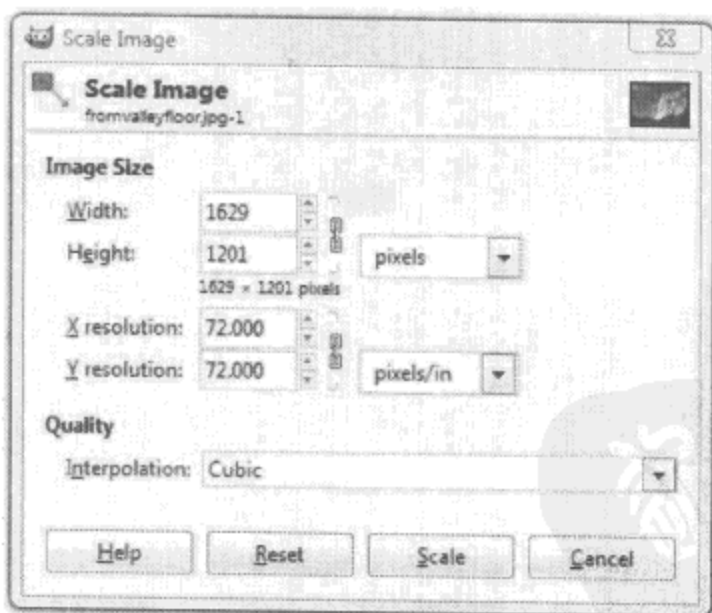
**提示：**你的图像软件在图像窗口中的某处可能有一个尺寸的显示。在 GIMP 中，当前图像尺寸可以在窗口顶端看到。其他软件可能在右下或者左下角显示。你还可能在窗口中看到放大率，以及改变放大率的方法（放大和缩小）。

## 10.3.2 改变图像大小

改变图像大小所需的工具取决于使用的图像软件。在 GIMP 中，在 Image（图像）菜单中单击 Scale Image（缩放图像）打开缩放图像对话框，如图 10.3 所示。

图 10.3

使用缩放图像对话框  
改变图像大小



你几乎总是希望使用原有长宽比来改变大小，这意味着输入图像宽度时，高度将自动计算（反之亦然），使图像不会受到挤压。在 GIMP 中，默认情况下锁定长宽比，这由图 10.3 中 Width（宽度）和 Height（高度）选项旁的链条指示。单击该链条将解锁，这时可以选择自己的像素宽度和高度——可以自己决定是否挤压。

在大部分（不是所有）图像软件中，你还可以根据百分比而不是具体的像素大小改变图像尺寸。例如，如果图像起始分辨率为 1629 像素×1487 像素，我不希望计算使其宽度缩小一

半的数值,可以选择百分比(从图 10.3 中所示像素旁边的下拉式菜单中)并且将默认设置(100)修改为 50。图像宽度会变成 815 像素,高度为 744——不需要做任何计算。

**注意:** 和 GIMP 中的许多功能一样,缩放图像对话框出现在包含被缩放图像的窗口的前面。这样可以在对话框中改变,应用并且立刻看到结果。

By the  
Way

### 10.3.3 调整图像颜色

如果你编辑照片而不是创建自己的图像,可能需要使用一些颜色校正工具来得到合适的照片。和许多图像编辑软件一样,GIMP 提供多种调整图像亮度、对比度、颜色平衡的选项,以及减少红眼的滤镜。要使用 GIMP 消除红眼,选择 Filters (滤镜) → 单击 Enhance (增强),然后单击 Red Eye Removal (红眼消除)。

大部分选项都相当直观,如果希望图像更亮,调整亮度。如果希望图像更红,调整颜色平衡。在 GIMP 中,Colors (颜色) 菜单可以访问许多工具。和前一小节中描述的缩放图像对话框一样,这些工具在工作区前景显示一个对话框。随着颜色的调整,图像反映这些更改。这种预览功能包含在大部分图像编辑软件中。

图 10.4 所示为调整色调/亮度/饱和度工具,这是颜色菜单中提供的工具之一。如图中所示,许多与颜色相关的更改通过使用对话框中的各种滑块的调整来进行。预览功能使你能够看到正在进行的工作。Reset Color (重置颜色) 按钮将图像返回到未作修改的原始状态。

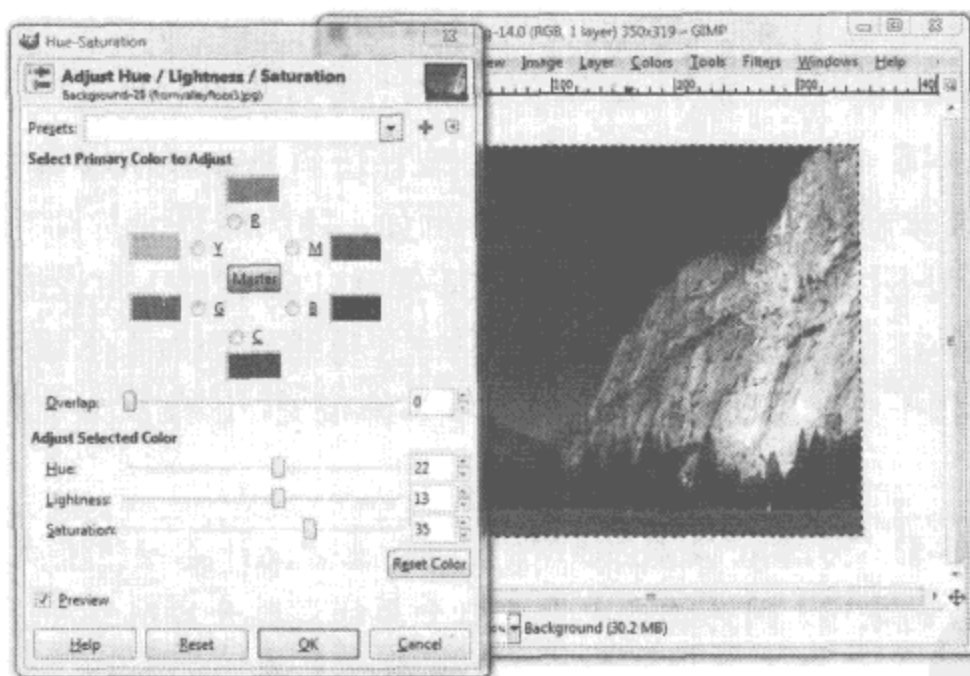


图 10.4

调整色调/亮度/饱和度工具是 GIMP 中基于滑块的颜色修改工具之一

因为可以使用多种工具,以及各种工具的预览功能,轻松的试验是了解各种工具作用的最佳方式。

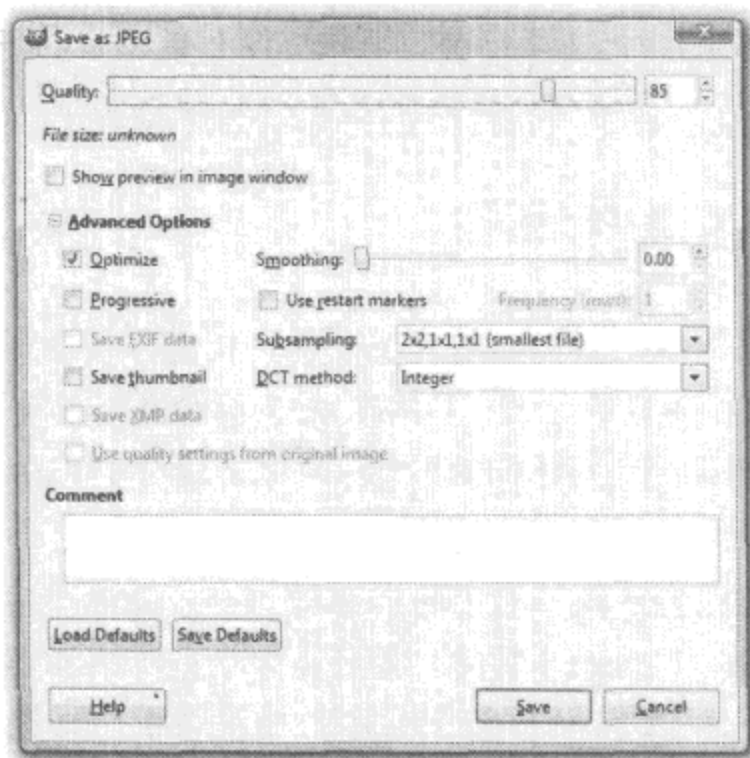
### 10.3.4 控制 JPEG 压缩

Web 上最好的照片图像存储方式是 JPEG 文件而不是 GIF, JPG 使你能够保留文件中的颜色数量,同时仍然保持文件大小在可管理的级别上。当你结束了照片大小和外观的调整之后,选择 File (文件) → Save As (另存为) 并且选择 JPEG 文件类型。你的图像软件可能提供另一个对话框,可以控制各种 JPEG 选项,如压缩。

图 10.5 所示为 GIMP 中保存 JPEG 的对话框。你可以到，这里可以通过调整 Quality（质量）滑块控制保存文件时的压缩率，数字从 1（低质量、小文件）到 100（高质量、大文件）。

图 10.5

GIMP 允许你将文件保存为 JPEG 格式，减小文件并且仍然保持图像质量



你可能希望试验一下，看看各种 JPEG 压缩水平对图像质量的影响，85%质量（或者 15% 压缩）对大部分照片来说一般是文件尺寸（也就是下载速度）和质量之间一个较好的妥协。

## 10.4 创建横幅和按钮

从头开始创建的图像，如横幅和按钮时，要考虑的因素与照片很不一样。

创建横幅或按钮时要做的第一个决定是图像应多大。现在访问 Web 的大多数用户的屏幕分辨率至少是 1024 像素×768 像素，有的可能要大得多，例如，我的屏幕现在设置为 1440 像素×900 像素。应规划好图像大小以便在较小的屏幕（1024 像素×768 像素）中也能正常显示，还能留下空间给滚动条和边距。重要的大小约束是页面的水平宽度，因为水平滚动页面争议很大，其常常使 Web 用户感到不便。竖直滚动页面就比较容易接受，因此页面的高度可以比屏幕的最小高度还大。

### **Did you Know?**

**提示：**许多年来，为 800 像素×600 像素的屏幕做设计是一种常规。仍然要记住这一较低的数字，因为许多人没有以全屏模式打开应用程序。但是，为 1024 像素×768 像素的屏幕分辨率设计不是个坏主意。

假设最小目标分辨率是 800 像素×600 像素，这意味着横幅和标题图像的宽度不应超过 770 像素，高度不应超过 430 像素，这是除去滚动条、工具栏和浏览器窗口的其他部分剩下的最大可见区域。在页面中，普通的照片和艺术画的每边应在 100 像素~300 像素之间，按钮和图标应更小，在 20 像素~100 像素之间。显然，如果你为 1024 像素×768 像素的屏幕分辨率设计，就有了更多的屏幕“空间”，但是用于横幅，按钮和其他辅助图像的尺寸准则仍然有效。

单击 File（文件）菜单并选择 New（新建）在 GIMP 中创建新文件。显示新建图像对话框，如图 10.6 所示。如果你不确定图像的大小，接受默认的 640 像素×480 像素。你也可以在 Template（模板）下拉菜单中选择其他预定义的尺寸，如常用 Web 横幅（468 像素×60 像



素) 或者大型 Web 横幅 (728 像素×90 像素)。这两种设置确实考虑了“常用”和“大型”网站横幅。否则, 输入新图像的宽度和高度。

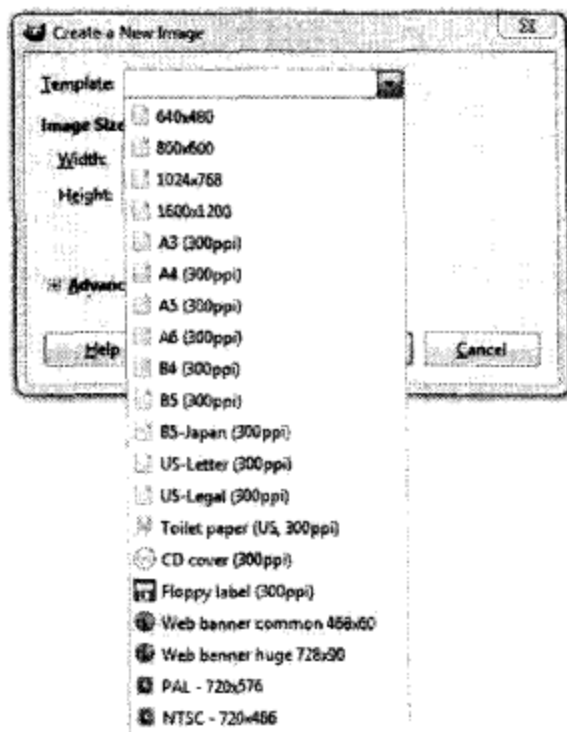


图 10.6

你需要在开始建立图像之前决定其尺寸

至于图像的背景颜色, 通常应选择白色, 以便和大多数网页浏览器用于显示网页时的背景颜色相匹配 (你在上一章中学到了, 背景颜色可以改变)。如果页面的背景颜色不是白色, 可以选择其他背景颜色。实际上, 你可能希望创建没有背景的图像, 在这种情况下可以选择 Transparent (透明) 背景颜色。如果图像的背景是透明的, 图像下面的网页将能够透过透明区域显示出来。在 GIMP 中, 在新建图像对话框中打开 Advanced Options (高级工具) 来选择背景颜色。

输入图像的宽度和高度并单击 OK 后, 将显示一块空白画布——这有点可怕, 如果你和大多数人一样有艺术恐惧症的话。但是, 有许多可以指导你的图像创建教程 (不是指完整的书籍), 我可以放心地让你自己去发挥创意。本章主要是向你介绍创建网站中使用的图像时需要记住的事情, 没有必要告诉你怎么做, 因为轻松地对待你选择的工具是掌握它的第一步。

## 10.5 减少图像中的颜色数量

减少图像大小, 从而减少下载时间的最有效方法之一是减少图像中使用的颜色。这可能会大大降低一些照片的质量, 但是对于大部分横幅, 按钮和其他图标效果很好。

你将会很高兴地知道有一种用于有限颜色数量的图像格式, 它叫做图像交换格式 (GIF)。当图像保存为 GIF 时, 你可能得到一个提示, 是使用灰度还是转换为减少颜色的索引图像, 这是 GIF 的要求, 如图 10.7 所示。对话框要求你确定保存进程所作的改变——这时候不要求理解这些选项, 但是阅读关于层次和索引颜色的软件帮助文件可以完全地理解这些概念。

**提示:** 抖动 (Dithering) 技术是图像编辑软件用于模拟不在调色板中的颜色的一种方法, 它用两种相似颜色的像素来替换。例如, 一个抖动的粉红色将包含红色和白色的替换像素, 这将给出粉红的大致印象。抖动有时候使图像显得更美观, 但是通常应该在网页图像中避免它。为什么? 它本质上增加了图像的信息复杂度, 这通常导致大得多的文件尺寸和较慢的下载。

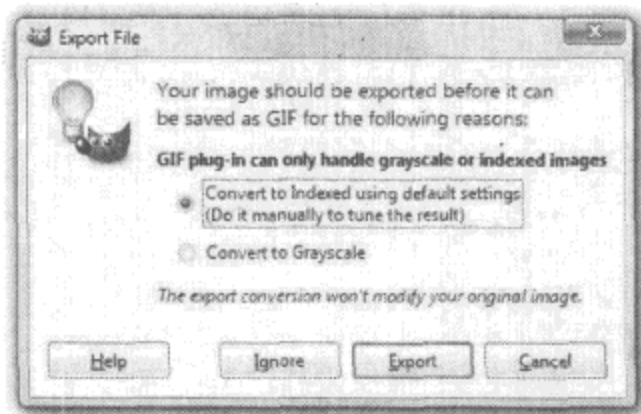
**Did you know?**

PDG



图 10.7

保存 GIF 图像时，  
你可能得到转换为  
索引调色板的提示



记住，GIF 图像格式设计用于包含固定颜色区域的图像，比如网页标题和其他演示图形，GIF 格式对于照片来说不理想。

PNG（发音为“ping”）是另一种所有主要浏览器都支持的文件格式。GIF 允许你指定一种透明的颜色，这意味着网页的背景在图像的这些区域将会显露出来，PNG 格式更进一步，使你可以指定不同的透明度。

## 10.6 使用透明的图像

你可能见过一些网站在容器元素中使用背景颜色或者图像，但是允许背景在一部分前景图像中显露出来。在这种情况下，前景中的图像有透明的部分，这样这些图像本身——总是在一个矩形的画布中——不显示没有出现这种设计的画布区域。你经常希望使用这种部分透明的图像，使图像在所有背景色或者背景图像上的效果更好。

要使图像部分透明，图像必须是 PNG 或 GIF 文件格式。前面已经提到，大多数支持 GIF 格式的图像软件让你能够指定一种透明颜色，而 PNG 图像则可以指定透明范围。PNG 优于 GIF，很大程度上就是因为这一点。最新的 Web 浏览器都支持 PNG 图像。关于 PNG 图像格式的详情，请参阅 <http://www.libpng.org/pub/png/pngintro.html>。

创建透明图像的过程取决于所创建图像的类型（GIF 或者 PNG）和用于创建的图像软件。查看图像软件的帮助文件或者在搜索引擎中输入“使用[你的软件名称]创建透明图像”。

## 10.7 创建平铺背景

任何 GIF 或 JPEG 图像都可用作背景图片。但是，在你开始创建平铺的背景（特别是图案组成的平铺背景）之前，问问自己这个平铺背景对网站的观感有何帮助，更重要的是，问问自己网站的文本在图案之上是否容易辨认？

考虑一下你每天光顾的网站，事实是网站很少在整个页面上使用平铺的图案背景。如果你将浏览限制在公司、产品、体育团队或者其他信息（一般是文本）优先的网站上，具有平铺的图案背景的网站就更少了。虽然 Web 给每个人进行独特设计的权利，但是如果你创建一个用于业务的网站，可能要避免图案背景和对比强烈的彩色文本。

如果你为整个网站使用平铺的图案背景，记住平铺的图像在你看不出它是平铺的时候最美观。换句话说，背景图像的顶部与底部、左边与右边能够无缝衔接时最好。

图 10.8 和图 10.9 所示为在用的背景平铺，两者都有无缝的背景，但是效果不同。

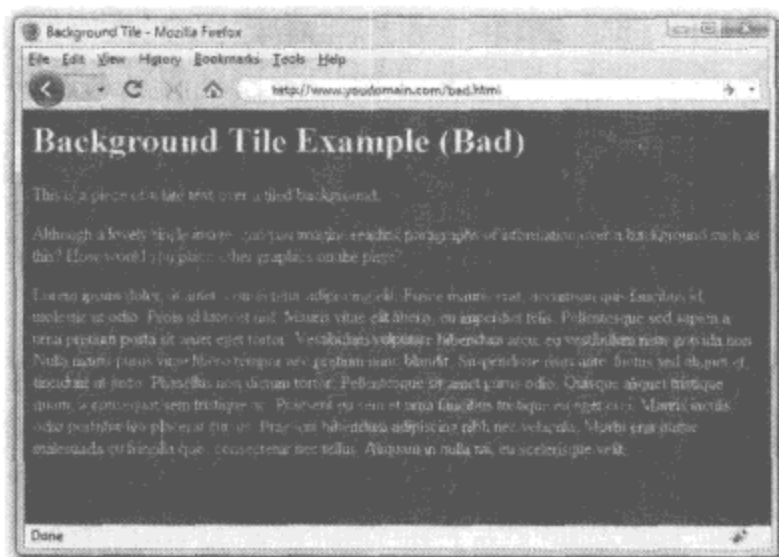


图 10.8

这是一个无缝的背景图像示例，你可以看出背景是平铺的，因为可以看到6个相同的形状

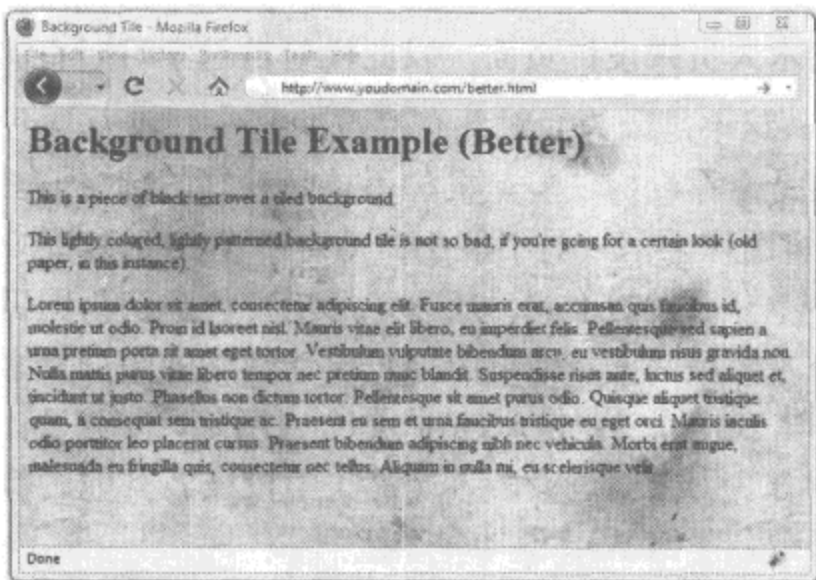


图 10.9

这也是一个无缝的背景图像示例，但是无法看出是平铺的

在第 11 章中，你将学习在容器元素中放置背景图像的方法。尽管我已经警告过，但是有时候背景图像是强有力的武器——但不是用作整个页面的背景。

**提示：**如果你确实希望使用背景平铺但是没有想要的图案，可以查看互联网上几百个提供免费或者价格低廉但设计专业的无版权背景图像的网站。

**Did you know?**

## 10.8 创建 Web 动画

GIF 图像格式使你能创建动画，这可以用于为网页添加一些生动的姿态。GIF 动画比大部分效果类似的视频或者多媒体文件的传输要快得多。使用 GIMP，你能够在图像中创建多个图层，然后再保存修改 GIF 动画选项，以此来创建 GIF 动画。此外，如果你有一系列希望用作动画的图像，可以使用免费的基于 Web 的 GIF 动画服务 Gickr (<http://www.gickr.com/>)。

创建一个 GIF 动画的第一步是创建一系列先后显示的图像——或者一系列图层，这取决于你的软件。每个图像被称为帧 (Frame)。顺便说一句，这里使用的 frame 一词和第 13 章中的 frame 没有任何关系，想想电影或者动画片中的帧——你在屏幕上看到的图像由许多单独的外观上稍有不同的帧组成。有了这些帧，将它们联系在一起的过程就很简单了——计划阶段是最困难的。花点时间在情节图板上草拟这些帧，特别是你计划使用较多的帧的时候。在你知道了帧组合的方式之后，使用本节前面提到的 Gickr 服务，或者阅读图像软件的文档来学习将这些帧组合在一起的过程。

## 10.9 总结

本章中，你学习了准备用于网页的图像的方法。如果没有意外，你已经知道这是很复杂的主题，并且在本章中学习到感兴趣的一些知识。本章中的示例使用流行（并且免费！）的 GIMP 软件包，但是可以尽管使用最适合你的需求的图像软件。

你学习了图像的裁剪、改变大小和调整颜色，并且学习了不同的文件格式。在网站上包含图像时需要考虑许多因素，包括图像尺寸、分辨率以及透明度、GIF 动画和平铺背景的使用。

## 10.10 问与答

**问：**我是否应该雇佣图形艺术人员来设计页面，以此代替自己学习所有的知识？

**答：**这是个很难回答的问题，这不是因为我对此有利益冲突——我为一个 Web 开发和设计机构工作，所以我的最大利益是推荐机构。但是机构不总是最好的解决方案。雇用图像设计师花费时间和金钱。此外，许多图像艺术人员不能制作适合于 Web 的作品——他们是基于印刷的艺术人员，印刷界和网络界有很大的不同。还有，雇用一位自认为图像设计师的人来创建网站可能无法发挥他的长处。换句话说，他可能擅长设计网站的图形元素，但是可能不擅长内容架构或者使用 HTML 和 CSS。如果你的网站只是个人网站，专业设计可能不是你想投入资金的地方。但是如果你的网站用来促进业务、产品、学校或者其他图像与成功相关的地方，那就值得花精力（和金钱）去咨询 Web 设计专家。

**问：**我已经建立用于纸面印刷的图像。网页图像有何不同？

**答：**是的，实际上许多印刷图像上的规则在 Web 上是相反的。网页图像必须是低分辨率的，而印刷图像的分辨率应该尽可能高。计算机屏幕上去掉黑色，而黑色却渗入纸上的白色。而且，有的人可能在图像下载一般的时候停止网页装入。如果你做过许多印刷图像设计，要避免陷入旧的习惯。

**问：**我有一个 Windows AVI 视频片段。我能把它转化为 GIF 动画吗？

**答：**可以。只要用 Animation Shop 之类的软件打开 AVI 文件就可以将其转换为 GIF 动画。该软件提供减少帧数的选项，一般每 3 帧采样一次，以便将文件尺寸降低到合理的比例。你也可以将 AVI 文件直接嵌入网页，这将在第 12 章中讨论。

## 10.11 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 10.11.1 测验

1. 有一幅扫描后的马的图片，你打算将它放到网页上。应该把它的尺寸控制为多大？保存为什么格式？

2. 你的公司标志是黑色的字母 Z，背后有一个红圈。你应该将它绘制或者扫描为多大？为了在网页上使用，应该将它保存为什么格式？

3. 你的业务伙伴带来了一个细节丰富的森林植物背景图像，希望你将它用作公司网站的背景——文字为加重的白色。你该怎么做？

### 10.11.2 测验答案

1. 根据图像在网页上的重要性，你应该将它做成小的 100 像素×40 像素或者大的 300 像素×120 像素的图像。格式为 JPEG，压缩率 85% 为最好。当然，你也可以提供一个指向较大的图像的缩略图链接。你将在下一章中学习到将图像作为链接的方法。

2. 正方形的标志的尺寸一般为 100 像素×100 像素左右，但是这样简单的图像将很容易压缩。你可以使其大到 300 像素×300 像素或者更大（在你的模板中要留有合适的位置——这是相当大的正方形）。将其保存为索引的 GIF 文件，因为它包含的颜色很少。

3. 你告诉他“不”，并且拒绝他对未来的设计作出任何建议。

## 10.12 练习

- 如果你有公司（或者个人）照片印刷的存档，仔细查看并且找出一些可能改进网站的照片。扫描它们（或者送交其他机构扫描），这样你将拥有一个可用于网页的数码照片库。如果你有取自数码照相机的数字文件，很显然可以跳过扫描这一步并且直接将其用于网站。
- 在你开始设计重要业务网站的图像之前，尝试为自己的个人主页增添情趣。这将给你一个学习 GIMP（或者其他图像软件）的机会，你将了解在工作时处理这一任务的方法。



## 第 11 章

# 在网页中使用图像

---

本章中你将学到：

- 在网页上放置图像的方法
- 用文本描述图像的方法
- 指定图像高度和宽度的方法
- 对齐图像的方法
- 将图像转换为链接的方法
- 使用背景图像的方法
- 使用图像映射的方法

在第 10 章中，你学习了寻找和创建用于网站中的数码图像的方法。本章将向你说明，将这些图像放置在网站中有多么容易。本章中，你将学习放置和描述网站中图像的 HTML，对齐图像以及将图像用作链接或者“映射”到其他内容的方法。

### TRY IT YOURSELF ▼

#### 自己动手尝试一下：为网站准备图像

现在你应该已经准备好两三幅图像，可以在本章的学习中将它们放在网页中。如果你有 GIF、PNG 或 JPEG 格式的图像文件（文件扩展名分别为 .gif、.png 和 .jpg），可以使用这些文件。也可以使用在前一章创建的其他图像。

搜索引擎（如 Google）也能成为图像的金矿，可以找到与所需主题相关的图像网站。它们还能帮助你找到许多专门提供免费或价格低廉的可复用媒体的网站。另外，不要忘记 Microsoft Office 在线剪贴画和媒体网站有大量的剪贴画库，其网址为 <http://office.microsoft.com/clipart/>。其他有价值的资源包括 Google Images（<http://images.google.com/>）和 Flickr（<http://www.flickr.com>）——寻找使用创意共享组织许可的图像，这允许图像免费使用。

## 11.1 在网页上放置图像

要将图像放在网页上，首先将图像文件放在 HTML 文本文件所在的文件夹下，或者为了更容易组织放在名为 Images 的目录下。

将下列 HTML 标签插入到要显示图像的地方。用实际图像文件替换 myimage.gif:

```

```

如果图像文件在文档根目录下的图像目录中，你可以使用如下 HTML:

```

```

**注意:** 对 Web 服务器，Web 浏览器或者最终用户来说你将图像放在哪里并不重要，只要你知道它们的位置并且在 HTML 代码中使用正确的路径就行。

**By the Way**

`<img />` 标签的 `src` 和 `alt` 属性是 XHTML 网页要求的。`src` 属性标识图像文件，`alt` 属性指定关于图像的描述性文本。当用户不能看到图像时，将看到 `alt` 属性指定的文本。在“用文本描述图像”小节中将学到关于 `alt` 属性的更多知识。

**注意:** `<img />` 标签也支持 `title` 属性，该属性用于描述图像。与 `alt` 属性不同的是，`title` 属性是在图像可见时提供图像描述。`alt` 属性的作用更为重要，主要是在图像不能显示时——如盲人用户用屏幕阅读软件“浏览”网页时——发挥作用。`alt` 属性是必需的，但同时提供 `alt` 和 `title` 属性更好，如果要保证图像在两种情况都有描述的话。

**By the Way**

程序清单 11.1 是一个使用 `<img />` 标签的示例，在页面顶端的文本段落之前插入一幅图像。浏览器显示程序清单 11.1 中的 HTML 文件时，自动读取并且显示图像文件，如图 11.1 所示。

### 程序清单 11.1 使用 `<img />` 标签在网页上放置图像

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>A Spectacular Yosemite View</title>
  </head>

  <body>
    <h1>A Spectacular Yosemite View</h1>
    <p></p>
    <p><strong>Half Dome</strong> is a granite dome in Yosemite National Park,
    located in northeastern Mariposa County, California, at the eastern
    end of Yosemite Valley. The granite crest rises more than 4,737 ft
    (1,444 m) above the valley floor.</p>
    <p>This particular view is of Half Dome as seen from Washburn Point.</p>
  </body>
</html>
```

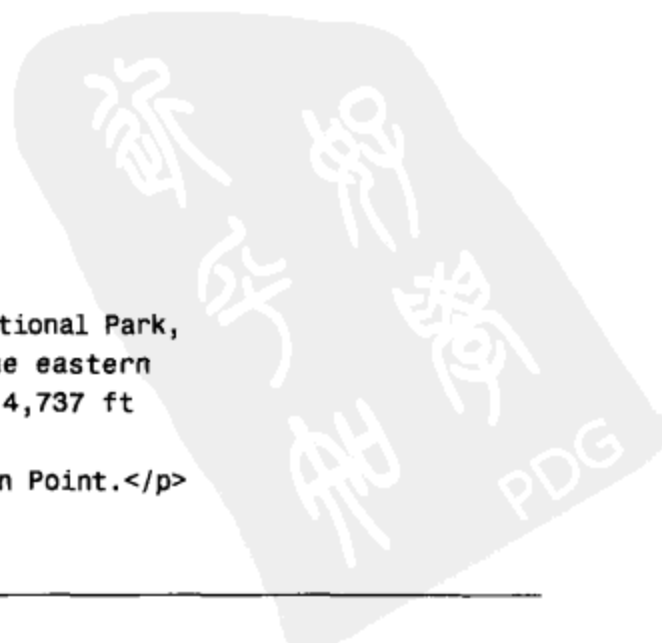
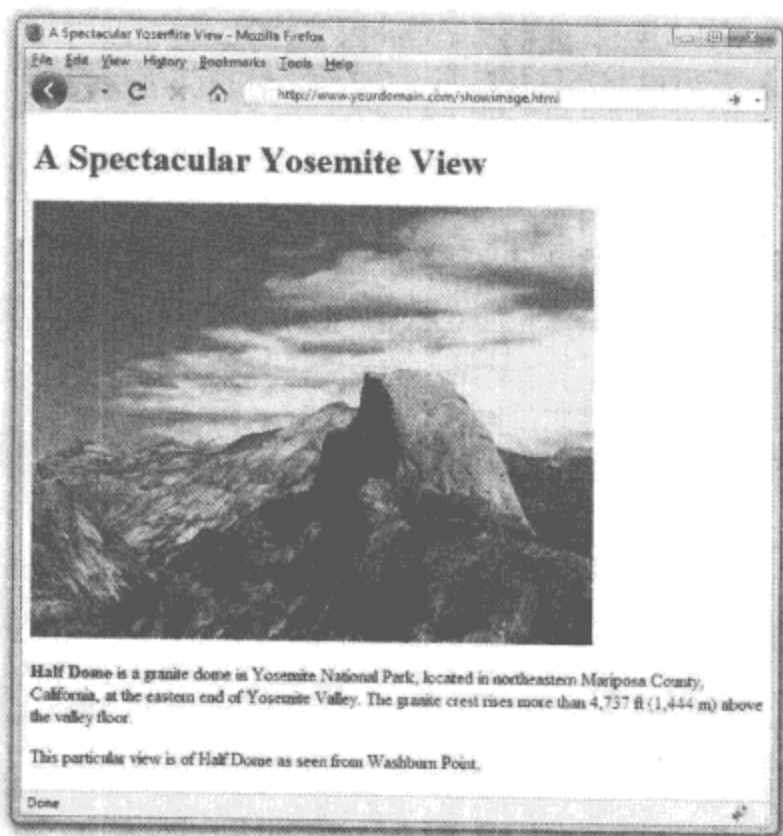




图 11.1

浏览器显示程序清单 11.1 中的 HTML 文件时，呈现 hd.jpg 图像



如果你猜到 `img` 表示 `image`（图像），那非常正确。而 `src` 表示 `source`（源），也就是图像文件所在的位置。在本书前面讨论过，网页图像总是存储在与网页文本独立的文件中，虽然它在浏览器中是作为网页的一部分显示的。

与 `<a href>` 标签（第 3 章介绍过）一样，可以在 `<img />` 标签的 `src` 属性中指定完整的 Internet 地址。如果图像与 HTML 文件在同一文件夹下，也可以只用图像的文件名。还可以使用相对地址，如 `images/birdy.jpg` 或 `../smiley.gif`。

### By the Way

**注意：**从理论上说，可以在网页中包含来自任何 Internet 网页的图像。在这种情况下，图像在页面显示时从其他页面的 Web 服务器上读取。你可以这么做，但是不应该这么做！这不仅是个坏习惯，因为你将其他人的带宽用于你的利益，这也可能使你的页面显示更慢。你也没有办法控制图像的修改或者删除。

如果你得到重新发布其他网页上图像的授权，将该图像的副本传送到你的计算机上并且使用本地文件引用，如 `` 来代替 ``。但是，当你在专用于图像库的服务（如 Flickr）上存放图像时，这条建议不适用，Flickr 这样的服务为每个图像提供一个 URL，这些 URL 地址中都包含 Flickr 的域。

## 11.2 用文本描述图像

程序清单 8.1 中的每个 `<img />` 标签都包含一段简短的文本信息，如 `alt="Half Dome"`。`alt` 表示“alternate text”（替换文本），它是图像本身没有装入时出现在该位置的文本。图像可能因为地址不正确，或者用户关闭了 Web 浏览器首选项中的自动下载图像，以及互联网连接非常慢时的数据没有传输而不能装入。图 11.2 所示为一个用于图像位置的 `alt` 文本的示例。

即使图像取代 `alt` 消息后，将鼠标指向图像时，`alt` 消息通常显示在一个小框（工具提示）

中。alt 消息也能帮助有视力障碍（或使用基于语音界面阅读网页）的人。

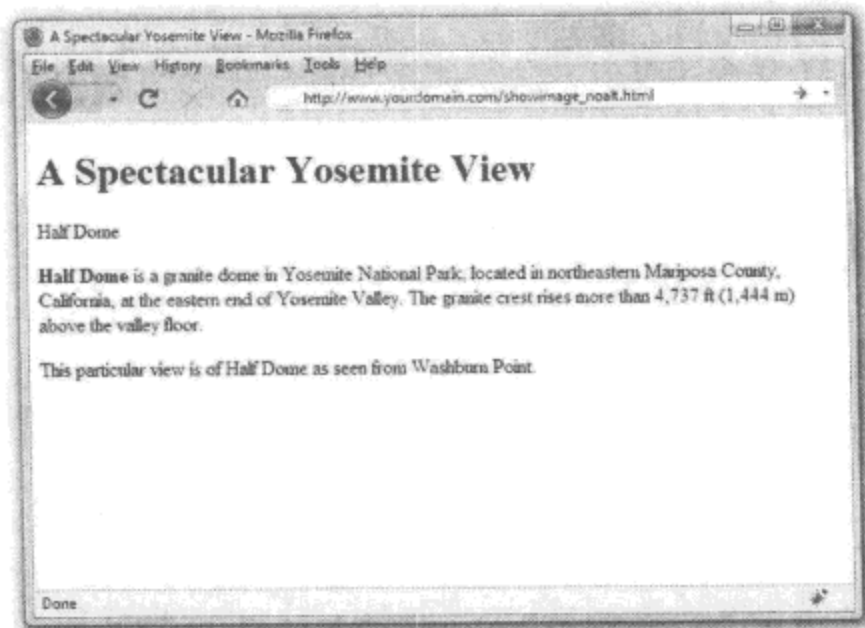


图 11.2

用户在图像没有出现时将看到 alt 信息

考虑到可能看到消息的用户情况是不同的，因此网页上的每个 `<img />` 标签必须包含一个合适的 alt 属性。通常最好使用对图像的简短描述，但网页制作者有时也在 alt 消息中放一些短小的广告消息或幽默笑话，不过不建议使用太多的幽默和不够的信息。对于小的或不重要的图像，完全可以省略 alt 属性，但这个属性是 `<img />` 标签要求的。这并不是说如果没有这个属性网页就不能正常显示，而是意味着违反了最新的 XHTML 标准。如果真的不需要这个属性，如对于一些小的或装饰性的图像，可以赋予它一条空文本消息 (`alt=""`)。

title 属性对 `<img />` 标签不是必需的。title 属性类似于 alt 属性，实际上，在两个属性都指定的情况下，在工具提示中 title 属性将覆盖 alt 属性。因此，通过文本来描述图像最好的解决方法是：同时使用 alt 和 title 属性来提供图像的相关注释和你认为作为工具提示或者通过屏幕阅读软件查看时有用的提示。

## 11.3 指定图像高度和宽度

因为文本在 Internet 上的传输比图像快，因此大多数网页浏览器将在图像之前显示网页的文本。这让用户在看到图像前有一些文本可看，使整个页面的显示显得更快些。

可以显式地指定每幅图像的宽度和高度，确保网页可以尽可能快地显示。这样的话，网页浏览器可以在加载网页，等待图像传输完毕的同时，为每幅图像留出准确的空间。

对于包含在网站中的图像，可以使用你的图像软件来确定准确的高度和宽度——以像素为单位。你也可以使用系统工具来找到这些图像的属性。例如，在 Windows 中，你可以右键单击一个图像，选择“属性”然后选择“摘要”查看图像的高度和宽度。知道了图像的高度和宽度，你可以在 `<img />` 标签中包含尺寸，如：

```

```

**提示：**为图像指定的 width 和 height 不一定要与图像实际的宽和高相等。两者不相等时，Web 浏览器将自动尝试将图像缩小或拉伸到指定的大小。然而，通常这样做并不好，因为浏览器在调整图像大小方面做得并不好。如果要使图像显示得小一点，最好在图像编辑器中调整好它的大小。

**Did you Know?**

## 11.4 对齐图像

与文本类似，也可以使用特殊的属性来对齐网页中的图像。不仅可以水平对齐图像，还可以垂直地将图像与包围图像的文本或其他图像对齐。

### 11.4.1 水平对齐图像

第5章讨论过，可用<div style="text-align:center">、<div style="text-align:right">和<div style="text-align:left">将元素居中对齐、右对齐或左对齐。这些样式设置可用于文本和图像，也可用于<p>标签。

和文本类似，图像一般与左边对齐，除非 style="text-align:center"或者 style="text-align:right"设置指出它们应该居中或者右对齐。换句话说，left是text-align样式属性的默认值。

也可以在<img />标签内直接使用float样式，使文本围绕图像。

在程序清单11.2中，<img style="float:left" />使图像左对齐，文本围绕在图像的右边。类似的，<img style="float:right" />使图像右对齐，文本围绕在图像的左边。图11.3所示为这些图像在网页上的对齐。没有将图像浮动在中央的概念，因为这样就无法确定文本应显示在图像的哪一边。

#### 程序清单 11.2 使用 text-align 样式在网页上对齐图像

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>More Spectacular Yosemite Views</title>
  </head>

  <body>
    <h1>More Spectacular Yosemite Views</h1>
    <p><strong>El
      Capitan</strong> is a 3,000-foot (910 m) vertical rock formation
      in Yosemite National Park, located on the north side of Yosemite
      Valley, near its western end. The granite monolith is one of the
      world's favorite challenges for rock climbers. The formation was
      named "El Capitan" by the Mariposa Battalion when it explored the
      valley in 1851.</p>
    <p><strong>Tunnel
      View</strong> is a viewpoint on State Route 41 located directly east
      of the Wawona Tunnel as one enters Yosemite Valley from the south.
      The view looks east into Yosemite Valley including the southwest face
      of El Capitan, Half Dome, and Bridalveil Falls. This is, to many, the
      first views of the popular attractions in Yosemite.</p>
  </body>
</html>
```

**By the  
Way**

**注意：**float 样式属性的功能实际上比这里介绍的还要强大，它不仅可用于图像，也可使用 float 样式属性创建一些有趣的页面布局，这将在稍后介绍。

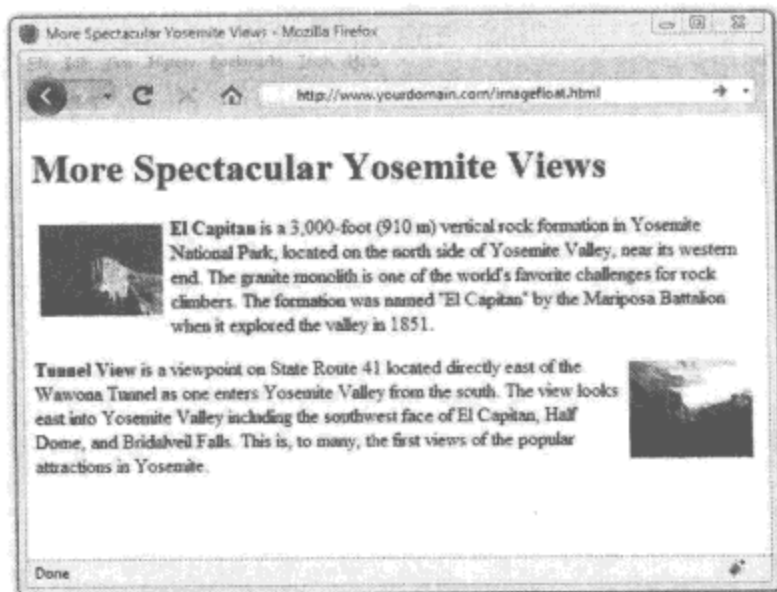


图 11.3

显示程序清单 11.2 中的图像对齐

**注意：**程序清单中的两个 `<img />` 标签都添加了填充（padding）样式属性。这种填充提供了图像和文本之间的“呼吸空间”——图像的四边都有 6 个像素。你将在第 13 章中学习到更多关于填充的内容。

**By the Way**

## 11.4.2 垂直对齐图像

有时可能希望在一行文本中插入一个小图像，或将一行文本放在图像旁边作为标题。在这些情况下，如果可以控制文本和图像在垂直方向上如何对齐将很方便。图像的底部与文本的底部对齐，还是文本和图像的中间对齐？可以在下列选项和其他一些选项中进行选择。

- 要使图像的顶部与同一行中最高的图像或文本的顶部对齐，使用 `<img style="vertical-align:text-top" />`。
- 要使图像的底部与文本的底部对齐，使用 `<img style="vertical-align:text-bottom" />`。
- 要使图像的中心与同一行中其他内容的中心对齐，使用 `<img style="vertical-align:middle" />`。
- 要使图像的底部与文本的基线对齐，使用 `<img style="vertical-align:baseline" />`。

**注意：**vertical-align 样式属性也支持 top 和 bottom 这两个值，这两个值分别用于将一行中图像的顶部和底部都对齐，而不管这一行中的文本。

**By the Way**

如图 11.4 所示，程序清单 11.3 演示了这 4 种选项。现在 4 个缩略图垂直排列在页面下半部分，每幅图像旁边都有一些描述性文本。每幅图像都使用不同的 vertical-align 样式属性来对齐图像与文本。

**提示：**如果在 `<img />` 标签中不包含 align 属性，图像的底部与文本的基线对齐。也就是说，实际上不必输入 `style="vertical-align:baseline"`，因为这是默认设置。然而，如果想为图像指定一个边距并更精确地对齐文本，可以显式地将 vertical-align 设置为 text-bottom。看图 11.4 中的最后一幅图像，由于图像设置了边距，且将 vertical-align 设置为 baseline，因此文本的位置低于图像的底部。

**Did you Know?**

## 程序清单 11.3 使用 vertical-align 样式对齐文本和图像

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

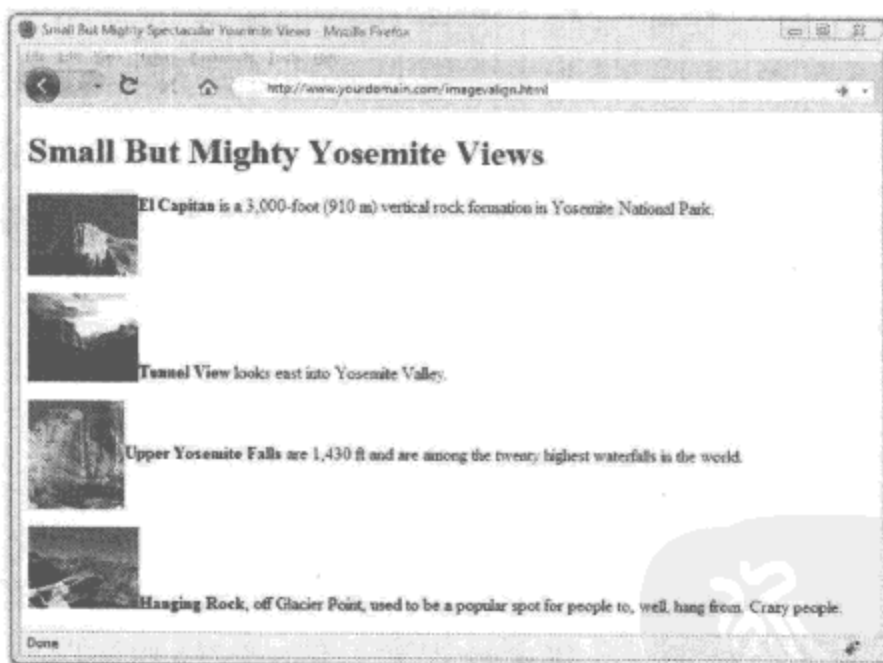
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Small But Mighty Spectacular Yosemite Views</title>
  </head>

  <body>
    <h1>Small But Mighty Yosemite Views</h1>
    <p><strong>El
Capitan</strong> is a 3,000-foot (910 m) vertical rock formation
in Yosemite National Park.</p>
    <p><strong>Tunnel
View</strong> looks east into Yosemite Valley.</p>
    <p><strong>Upper
Yosemite Falls</strong> are 1,430 ft and are among the twenty highest
waterfalls in the world. </p>
    <p><strong>Hanging
Rock</strong>, off Glacier Point, used to be a popular spot for people
to, well, hang from. Crazy people.</p>
  </body>
</html>

```

图 11.4

程序清单 11.3 种使  
用的垂直对齐选项


**TIP OF YOURSELF**
**自己动手尝试一下：试验图像对齐**

现在在网页中添加一些图像，并尝试使用 text-align、vertical-align 和 float。开始之前，先看看如何在网页中添加一张鱼的图像（fish.jpg）。

1. 将图像文件 fish.jpg 复制到 HTML 文件所在的目录下（或者将图像放在它的当前位置并且确认记住这一位置）。
2. 使用文本编辑器，在 HTML 文件中要显示图像的位置添加 ``。



3. 如果要使图像居中，在- 4. 如果还有时间的话，可以试着将多个大小不同的图像以不同的垂直对齐设置进行组合。

## 11.5 将图像变为链接

你可能注意到了，图 11.1 中页面上的图像很大，如果要显示多幅图像，这样的图像就不是很理想。这种情况下，最好创建较小的缩略图，链接到每幅图像的较大版本。这样就可以将缩略图安排在一个网页上，用户可以比较方便地看到所有内容，即使他们看到的只是要显示的一些样本。缩略图是使用图像链接改善网页的一种方式。

可以像创建文本链接一样，使用前面学过的<a href>标签将任何图像变为指向另一个网页或图像的可单击链接。程序清单 11.4 修改了程序清单 11.2 的代码，它使用了缩略图来提供指向图像较大版本的链接。为了确保用户知道单击缩略图，图像和一些帮助文本包含在一个<div>中，如图 11.5 所示。

### 程序清单 11.4 使用缩略图作为有效的图像链接

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>More Spectacular Yosemite Views</title>
    <style type="text/css">
      div.imageleft {
        float:left;
        clear: all;
        text-align:center;
        font-size:9px;
        font-style:italic;
      }
      div.imageright {
        float:right;
        clear: all;
        text-align:center;
        font-size:9px;
        font-style:italic;
      }
      img {
        padding: 6px;
        border: none;
      }
    </style>
  </head>
  <body>
    <h1>More Spectacular Yosemite Views</h1>
    <p><div class="imageleft">
      <a href="http://www.flickr.com/photos/nofancyname/614253439/"><img
```



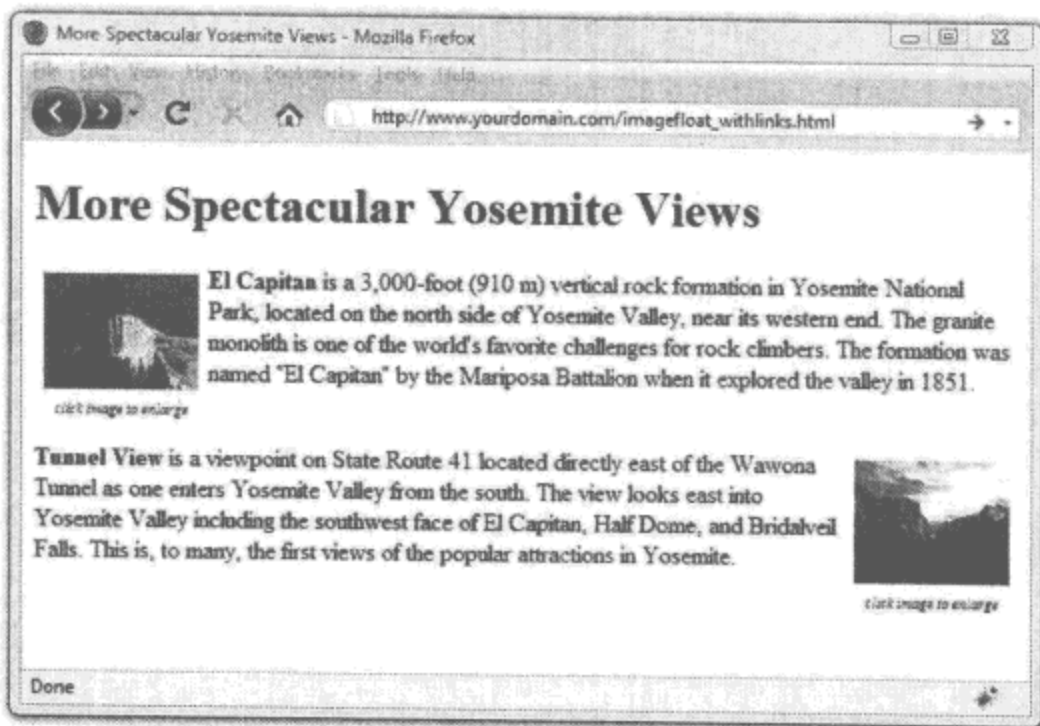
```

src="elcap_sm.jpg" alt="El Capitan" width="100" height="75"/></a>
<br/>click image to enlarge</div><strong>El Capitan</strong>
is a 3,000-foot (910 m) vertical rock formation in Yosemite National
Park, located on the north side of Yosemite Valley, near its western
end. The granite monolith is one of the world's favorite challenges
for rock climbers. The formation was named "El Capitan" by the
Mariposa Battalion when it explored the valley in 1851.</p>
<p><div class="imageright">
<a href="http://www.flickr.com/photos/nofancyname/614287355/"></a>
<br/>click image to enlarge</div><strong>Tunnel View</strong> is a
viewpoint on State Route 41 located directly east of the Wawona Tunnel
as one enters Yosemite Valley from the south. The view looks east into
Yosemite Valley including the southwest face of El Capitan, Half Dome,
and Bridalveil Falls. This is, to many, the first views of the popular
attractions in Yosemite.</p>
</body>
</html>

```

图 11.5

用缩略图作为链接  
改进了使用大的图  
像的页面布局



程序清单 11.4 中的代码使用的样式在第 14 章和第 15 章中更详细地解释，但是你可以领会以下基础知识。

- `<a>` 标签将这些特殊图像链接到更大的版本，在这个例子中这些图像保存在外部服务器（Flickr）。
- `<div>` 标签及其样式用于对齐这些图像和标题文本（还包括一些填充）。

除非另做说明，否则 Web 浏览器在图像链接边缘显示一个彩色矩形。与文本链接一样，通常矩形在还没有访问链接前是蓝色的，而访问后变为紫色——除非你在样式单中指定不同颜色的链接。因为通常都不希望显示这个难看的边框，因此应在 `<img />` 标签中包含样式 `style="border-style:none"`。在这个例子中，`border:none` 成为 `img` 元素的样式单输入项的一部分，因为我们两次使用相同的样式。

当你单击示例页面上的缩略图像，链接在浏览器中打开，如图 11.6 所示。



图 11.6

单击链接的缩略图  
打开链接的目标

## 11.6 使用背景图像

你在前一章中学过，可以使用背景图像作为容器元素的一种“墙纸”，这样文本或者其他图像出现在基础设计之上。

创建背景的基本样式属性如下。

- **background-color**: 指定元素的背景颜色。虽然不与图像相关，但是它是背景相关属性集的一部分。
- **background-image**: 指定作为元素背景的图像，语法如：`url("imagename.gif")`。
- **background-repeat**: 指定图像重复的方式，可以是水平或者垂直方向。默认（不指定任何值）情况下，背景将在垂直和水平方向都重复。其他的选项为：`repeat`（与默认值相同）、`repeat-x`（水平）、`repeat-y`（垂直）和 `no-repeat`（图像仅出现一次）。
- **background-position**: 指定图像开始相对于其容器的位置。选项包括：`top-left`（左上）、`top-center`（上中）、`top-right`（右上）、`center-left`（中左）、`center-center`（居中）、`center-right`（中右）、`bottom-left`（下左）、`bottom-center`（下中）、`bottom-right`（右下），以及具体的像素和百分比位置值。

指定一个背景图像时，你可以将所有规格放到一个属性中，如：

```
body {
  background: #ffffff url('imagename.gif') no-repeat top right;
}
```

在前一个样式单输入项中，网页的 `body` 元素将为白色并且在右上角包含一个名为 `imagename.gif` 的图像。`Background` 属性的另一个用处是为无序列表创建一个自定义项目符号。为了将图像作为项目符号，首先为 `<ul>` 标签定义如下样式：

```
ul {
  list-style-type: none;
  padding-left: 0;
  margin-left: 0;
}
```

接下来，修改 `<li>` 标签的声明为：

```
li {
    background: url(mybullet.gif) left center no-repeat
}
```

确保 `mybullet.gif`（或者其他图像名称）在 Web 服务器上并且可以访问。在这个例子中，所有无序列表项目将显示一个自定义图像而不是标准的实心圆点项目符号。

在后面的章节中将 CSS 用于整体页面布局时，我们将回到背景属性的具体使用上。

## 11.7 使用图像映射

有时候，你可能想要将图像作为导航，但是超出在网站上常见的基于按钮或者链接的简单导航的范畴。例如，你可能有一个具有医药信息的网站，希望显示人体的图像，链接到提供关于身体各部分的信息的页面。或者你有一个提供让用户可以点击的世界地图，用于访问各国的信息。你可以将图像根据用户点击的位置分为链接到不同文档的区域。这被称为图像映射（`imagemap`），任何图像都可以组成一个图像映射。

### 11.7.1 为什么图像映射不总是必要的？

关于图像映射，首先要说明的一点是，除非在很特殊的情况下，否则不需要使用图像映射。通常使用多幅图像，将它们排列在一起，每幅图像作为一个单独链接指向不同的页面，这会比较简单，也更高效。

如图 11.7 所示，这是显示 12 个不同的公司标志的网页。这个示例是商业界常见的网页类型，其中你为伙伴们做一个小小的免费广告。你可以将这些标志作为一个大的图像并且创建一个图像映射，提供指向 12 个公司的链接。用户可以单击图像中的每个标志来访问相应的公司。你也可以使用 12 单独的图像——每个用于一个公司，每个图像包含一个指向特定公司的链接。图像映射是具有多个部分的、排列方式古怪的，或者图像设计本身过于复杂而难以分为单独图像这些情况下的最佳选择。图 11.8 所示为一个最适合作为图像映射的图像。

图 11.7

具有 12 个不同标志的网页；这可能作为一个图像映射，也可以分为 12 个单独的部分



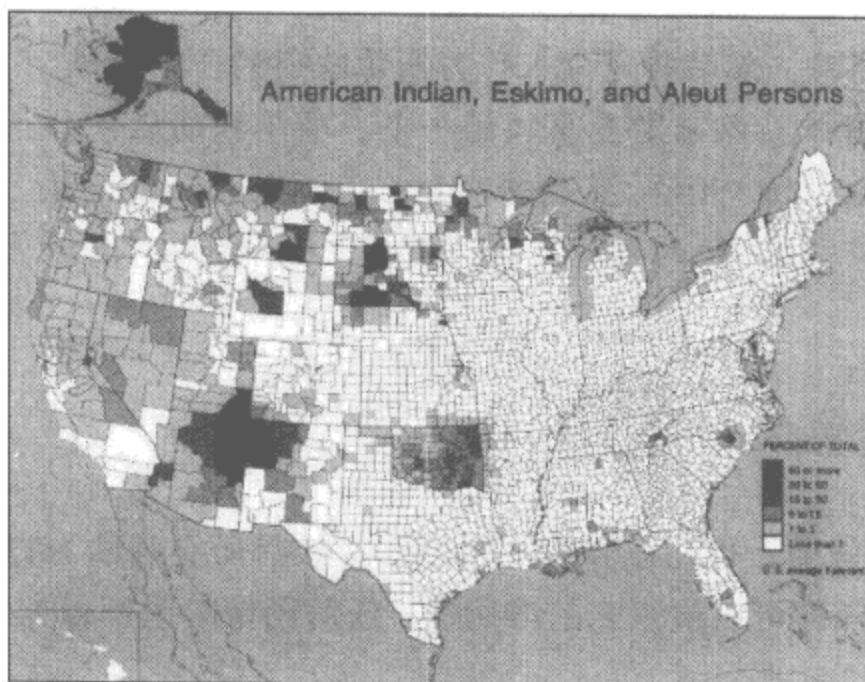


图 11.8  
难以分割的图像  
——作为一个图像  
映射更好些

### 11.7.2 映射图像中的区域

要创建图像映射，首先需要知道图像内要变为可单击链接的每个区域的数字像素坐标。这些可单击链接又称为区域（area）。你的图像软件可能提供找到这些坐标的简易方法，你也可能希望使用单独的图像映射工具如 [Mapedit](http://boutell.com/mapedit/) (<http://boutell.com/mapedit/>) 或者 <http://www.image.maps.com/> 上的在线图像映射制作程序。除了帮助你映射坐标以外，这些工具还提供建立映射所需要的 HTML 代码。

图像映射工具的使用通常只是简单地使用鼠标在希望成为一个链接的区域周围画一个矩形（或者自定义形状）。图 11.9 所示为这种矩形选择的结果，以及添加该链接的 URL 及标题或者替换文本的界面。创建图像映射需要几个部分的信息：坐标、目标 URL、链接标题、链接替换文本。

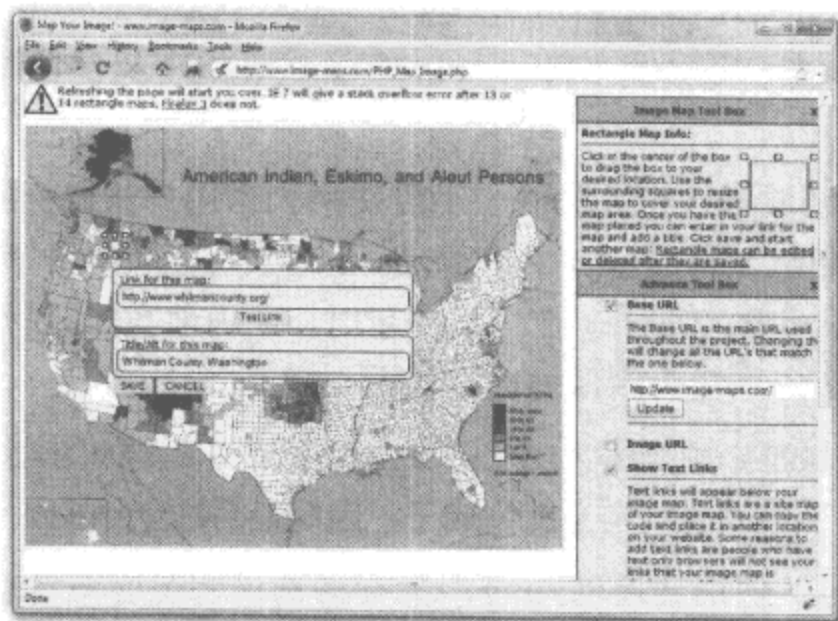


图 11.9  
使用图像映射工具  
创建一个图像的  
链接区域

#### 自己动手尝试一下：创建自己的图像映射

如果你有自己的图像并在继续学习本章时将它用作图像映射，那么就更容易记住如何创建图像映射。

- 对于初学者，最简单的是选择一个相当大的图像，大致分为几个矩形区域。

TRY IT YOURSELF ▼

- 如果你手头没有合适的图像,那么可以使用你最喜欢的图像软件创建一个。或者使用包含多个人的照片,然后将每个人作为图像映射的一个区域。
- 尝试一些不同的图像映射工具以确定最喜欢的一个。从单独的软件如 MapEdit (<http://boutell.com/mapedit/>) 开始,然后转向 <http://www.image.maps.com/> 上的在线图像映射制作程序。其他的程序可以使用搜索引擎查找“图像映射软件”主题。

### 11.7.3 创建用于图像映射的 HTML

如果使用一个图像映射生成器,你将已经有了创建图像映射所必须的 HTML。但是,理解各部分代码是个好主意,这样你可以检查它的准确性。任何图像映射都以如下的 HTML 代码开始:

```
<map name="mapname">
```

要记住, <map> 标签的 id 可用任意名称,但最好使用具有说明性的名称。接下来是图像的每个区域各使用一个 <area /> 标签。下面是用于图像映射示例中的一个 <area> 示例:

```
<area shape="rect" coords="100,136,116,152"
      href="http://www.whitmancounty.org/"
      alt="Whitman County, WA"
      title="Whitman County, WA" />
```

<area /> 标签有 5 个属性,可用于描述图像映射中的每个区域。

- shape 指明该区域为矩形 (shape="rect")、圆形 (shape="circle") 还是不规则多边形 (shape="poly")。
- coords 给出区域的像素坐标。对于矩形,给出左上角的 x、y 坐标以及右下角的 x、y 坐标;对于圆形,给出圆心的 x、y 坐标和半径,以像素为单位;对于多边形,按连接顺序列出所有顶点的 x、y 坐标。
- href 指定区域链接的页面。可以像在普通 <a href> 链接标签中那样使用任何地址或文件名。
- alt 让你能够提供与该区域形状相关联的一小段文字。当用户将鼠标指向该区域时,大多数浏览器 (除 Firefox) 将显示一个小窗口。这些文字为不理解图像映射的用户提供提示,虽然小却很重要。Firefox 在提供提示时,正确地在 alt 属性之外使用 title 属性,这也就是为什么应尽量为图像同时提供两个属性。

图像映射中的每个可单击区域都必须描述为一个区域,也就是说典型的图像映射由一系列区域组成。编写完 <area /> 标签后,图像映射的定义就完成了,因此最后加上 </map> 结束标签。

创建图像映射的最后一步是,使用普通的 <img /> 标签将真正的映射图像放在网页中。然而,有一个额外的 usemap 属性,如下所示:

```

```

指定 usemap 属性的值时,使用 <map> 标签中 name 属性指定的名称 (别忘了符号 #)。在



程序清单中，还使用了 `style` 属性，使图像映射的边框不显示出来，你自己的图像映射可以保留也可以不保留。

程序清单 11.5 显示了包含映射图像、图像映射和一些映射区域的样板网页。

### 程序清单 11.5 用 `<map>` 和 `<area/>` 标签定义图像映射区域

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Testing an Imagemap</title>
  </head>

  <body>
    <h1>Testing an Imagemap</h1>
    <p style="text-align:center">Click on a logo to go to the
    county's web site.<br/>
    </p>

    <map name="countymap" id="countymap">
      <area shape="rect" coords="100,136,116,152"
        href="http://www.whitmancounty.org/"
        alt="Whitman County, WA" title="Whitman County, WA" />
      <area shape="rect" coords="29,271,42,283"
        href="http://www.sccgov.org/" alt="Santa Clara County, CA"
        title="Santa Clara County, CA" />
      <area shape="rect" coords="535,216,548,228"
        href="http://visitingmifflincounty.com/"
        alt="Mifflin County, PA" title="Mifflin County, PA" />
    </map>
  </body>
</html>
```

**注意：**如果你注重细节的话，可能注意到这个网页是 XHTML 1.0 文档，而不像本书其他大部分例子那样是 XHTML 1.1 文档（看看代码的前几行）。这样做的原因是，有些浏览器（如 Internet Explorer）不支持 XHTML 1.1 中关于如何使用图像映射的小修改。这里说的是 `usemap` 属性，在 XHTML 1.1 中，映射名称前面不要求符号 `#`。实际上，这在 XHTML 1.1 中是不允许的，这也就是我将该网页设计成 XHTML 1.0 文档的原因。符号 `#` 在 XHTML 1.0 中是允许的，因此，为满足当前 Web 浏览器的需要，同时提供有效的网页，我在这个例子中选择了 XHTML 1.0。

**By the  
Way**

图 11.10 所示为图像映射的运行情况。注意，在浏览器窗口（这个例子中是 Firefox）的底部显示了鼠标悬停的区域的链接地址。此外，当你的鼠标悬停于某个区域，该区域的 `alt` 或者 `title` 文本——在这个例子中是 Whitman County 显示在图像映射之上。

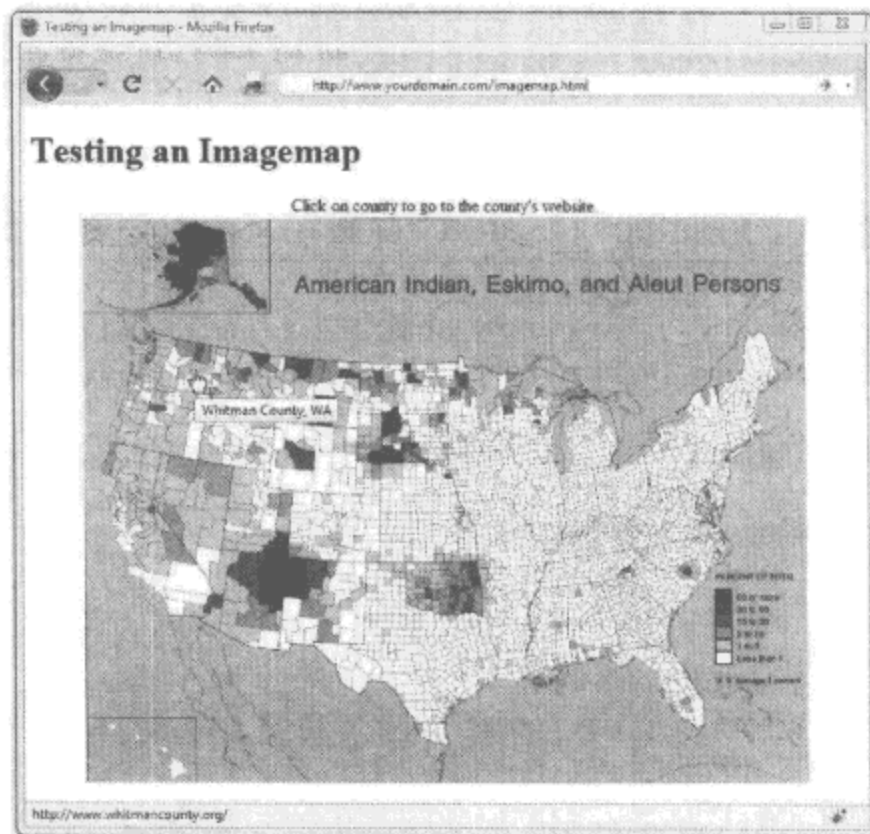
**注意：**有一种方法仅依赖 CSS 而不是 `<map>` 标签来生成映射图像。你将在第 16 章中学习到。

**By the  
Way**



图 11.10

程序清单中定义的  
图像映射在网页上  
的显示



## 11.8 总结

本章说明了使用标签在网页上放置图像的方法。你学习了包含在图像装入的时候以及用户的鼠标指针位于图像之上时显示的简短文本消息的方法。你还学习了控制图像的水平垂直对齐以及在图像的左右卷绕文本的方法。

你还学习了如何将图像用作链接——通过使用标签包围图像或者创建图像映射。你还学习了一些在容器元素中使用图像的知识。

表 11.1 总结了本章中介绍的标签，以及相关的样式属性。

表 11.1 第 11 章中介绍的 HTML 标签和属性

标签/属性	功能
<code>&lt;img /&gt;</code>	将图像放在网页中
<code>&lt;map&gt;...&lt;/map&gt;</code>	客户端图像映射，由
<code>&lt;area /&gt;</code>	在用户端图像映射中定义可单击的链接
<b>属性</b>	
<code>src="地址"</code>	图像的地址或文件名
<code>alt="取代图像的描述"</code>	显示在图像位置上的对图像的描述，主要提供给不能看到图像的用户看的
<code>title="标题"</code>	作为图像标题显示的文本信息，通常显示在图像上的一个弹出窗口（工具提示）中
<code>width="宽度"</code>	图像的宽度，单位是像素
<code>height="高度"</code>	图像的高度，单位是像素
<code>style="border-style:none"</code>	如果图像是链接，就去掉图像周围的边框
<code>style="vertical-align:alignment"</code>	图像在垂直方向上按 text-top、top、text-bottom、bottom、middle 或 baseline 对齐
<code>style="float:float"</code>	使图像浮动在一边，文本环绕在另一边。可选的值有 left、right 和 none（默认值）
<code>usemap="名称"</code>	为客户端图像映射指定的图像映射名，用于与<map>及<area />

续表

属性	
shape="值"	可单击区域的形状，这个属性有效的取值有 rect、poly 和 circle
coords="值"	图像中可单击区域的坐标，其含义和设置随区域类型而异
href="链接 URL"	单击区域时应加载的 URL

## 11.9 问与答

问：放在<img />标签的 alt=后面的信息可以有多长？

答：从理论上说，想多长都可以。实际上，应使信息足够短，使其可以在图像本身的空  
间内显示。对于大图像，10 个单词比较好，但我也见过有人放置一小段文本。对于小图像，  
一个单词就够了。

问：我按照你所说的那样使用了<img />标签，但浏览网页时只看到一个包含“x”或类  
似的形状的小框。这是为什么呢？

答：你看到的是图像受损标记，可能是因为以下两种情况之一：网页浏览器找不到图像  
文件，或浏览器不能理解图像文件的格式。要解决这个问题，首先检查图像确实在应在的位  
置。然后再图像编辑器中打开图像，将其存为 GIF、JPG 或者 PNG 格式。

问：如果图像映射上的区域重叠会发生什么？

答：可以重叠图像映射区域。只要记住，在确定使用哪个链接时，一个区域优先于另一  
个就行了。优先级根据图像映射中列出的顺序来分配。例如，映射中的第一个区域优先于  
第二个区域，也就是说点击两个区域之间的重叠部分将会链接到第一个区域。如果在图像  
映射中有一个区域没有链接到任何目标（被称为“死区”），可以使用重叠方法来避免它链  
接到任何目标。为此，只要将死区放在其他区域之前，这样死区与它们重叠，然后设置 href  
属性为“”。

## 11.10 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，  
然后再看后面的答案。

### 11.10.1 测验

1. 如何将图像文件 elephant.jpg 插入到网页的顶部？
2. 如何在网页浏览器不能显示图像 elephant.jpg 时显示单词 Elephant？
3. 你希望创建一个 200 像素×200 像素的图像 quarters.gif 的图像映射。单击图像左上的  
1/4 部分时，进入页面 topleft.html；单击右上四分之一时，进入页面 topright.html；单击图像  
左下的 1/4 部分时，进入页面 bottomleft.html；单击图像右下的 1/4 部分时，进入页面  
bottomright.html。实现这个图像映射要使用什么样的 HTML？

## 11.10.2 测验答案

1. 将这幅图像文件复制到 HTML 文本文件所在的文件夹，然后在 HTML 文本文件的 `<body>` 标签后输入

```
<p></p>
```

2. 使用以下 HTML:

```

```

3. 创建如下图像映射:

```
<map name="quartersmap" id="quartersmap">
  <area shape="rect" coords="0,0,99,99" href="topleft.html"
    alt="top left" />
  <area shape="rect" coords="100,0,199,99" href="topright.html"
    alt="top right" />
  <area shape="rect" coords="0,100,99,199" href="bottomleft.html"
    alt="bottom left" />
  <area shape="rect" coords="100,100,199,199"
href="bottomright.html"
    alt="bottom right" />
</map>

```

## 11.11 练习

- 练习本章中的图像放置方法将对确定图像在网站中所能扮演的角色有很大的帮助。使用一些示例图像，练习使用 `float` 样式放置图像和相关的文本。记住，`float` 属性的可能值为 `left`、`right` 或 `none`（默认）。
- 图像对齐也是好的 Web 视觉设计的重要方面。使用一些示例图像，练习使用 `vertical-align` 样式来放置图像和相关文本。记住，`image-align` 的可能值为 `text-top`、`top`、`text-bottom`、`bottom`、`middle` 或 `baseline`。



## 第 12 章

# 在网页中使用多媒体

本章中你将学到：

- 如何链接多媒体文件
- 如何嵌入多媒体文件
- 使用多媒体的更多技巧

多媒体这一术语包含了我们在网页上看到和听到的一切：音频、视频和动画，以及静态的图像和文本。在前一章中，你学习了图像和文本的使用，在本章中，你将学习如何在网站中包含其他类型的多媒体。你不用学习特殊的文本、视频或者动画的创建，而将要学习通过链接或者嵌入内容将这些文件包含到网站的方法。

在将多媒体包含到网站之前，要记住不是所有用户都具有能够播放你的媒体类型的设备，也不是所有用户都有快速传输大文件的宽带连接。始终警告你的访问者：他们所点击的链接将带给他们多媒体文件，并且提供给他们查看或者聆听该内容的选择——不要强迫他们使用这些文件。

### TRY IT YOURSELF ▼

**自己动手尝试一下：创建或者寻找一些用于网站的多媒体**

要了解如何将多媒体放在网页中，必须有一些多媒体内容。

创建任何一种多媒体内容都可能是复杂而充满挑战的任务。如果计划从头开始创建多媒体内容，需要远比本书多的知识，才能成为优秀的多媒体开发人员。然而，如果已经有一些内容，本章将教你如何将其放到网页中。

对于不是很有艺术才能的人，有几种方法可获得有用的多媒体材料。除很显然的方式（如雇用艺术家）外，可参考下列建议。

- Internet 上的许多材料都是免费的。当然，向媒体内容的作者或当前拥有人核实是个不错的主意，你肯定不希望因为版权问题而被起诉。另外，美国政府的各个机构生成的内容，从法律上说属于所有美国人（如 NASA 的在线电影胶片可以免费使用）。

TRY IT YOURSELF ▼

➤ 许多搜索引擎 (google.com、yahoo.com 和 lycos.com 等) 都能够搜索多媒体文件。只要谨慎对待版权问题, 通过这种方式很容易找到与特定主题相关的多媒体。简单地搜索“Flash 动画样板”、“QuickTime 电影样板”、“音频文件样板”就能得到多得无法处理的结果。

➤ 如果你有创造力, 确定你最喜欢的媒体—可能是视频制作, 或者音频制作, 还有人可能想试试动画。确定了出发点, 寻求各种使你能创建这些艺术作品的软件。许多公司提供多媒体软件, 如 Adobe (<http://www.adobe.com/>) 和 Apple (<http://www.apple.com/>)。

## 12.1 链接到多媒体文件

要将视频或音频文件包含到网站中, 最简单、最可靠的方法是使用<a href>链接它, 就像链接到另一个 HTML 文件一样。

例如, 下面一行代码可用于提供曲棍球游戏的 AVI 视频:

```
<a href="hockey.mov">View the hockey video clip.</a>
```

### By the Way

**注意:** 不管本章中所展示的媒体类型为何, 将多媒体组合到网页中的过程都相似。

当用户单击“View the hockey video clip.”时, QuickTime 视频文件 hockey.mov 将传输到其计算机上。他安装的助手应用程序或插件将在文件下载完毕时自动启动。如果没有找到与 AVI 兼容的助手程序或插件, 网页浏览器将为用户提供一个下载适当插件或将视频保存到硬盘中供以后浏览的机会。

程序清单 12.1 包含了使用简单图像链接来播放 Windows Media 格式视频的代码。除了图像链接以外, 还有一个文本链接为用户提供上下文。

### 程序清单 12.1 使用<a>将一个图像链接到 Windows Media 视频

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Fun in the Pond</title>
  </head>

  <body>
    <h1>Fun in the Pond</h1>
    <p><a href="pond.wmv"></a>
      Michael's backyard pond is not only a fun hobby but also an
      ongoing home improvement project that is both creative and
      relaxing. He has numerous fish in the pond, all Koi from various
      places as far as Japan, Israel, and Australia. Although they
      don't bark, purr, or fetch anything other than food, these fish
      are his pets, and good ones at that. You can <a href="pond.wmv">
      click here</a> or on the animated graphic on the left
      to see a movie clip of some fish in the pond.</p>
  </body>
</html>
```



**注意：**你可能不熟悉助手程序（helper applications，简称为 helper apps），它们是 Web 浏览器调用来显示自己不能处理的文件类型的外部程序。一般来说，助手应用程序与一个文件类型相关，Web 浏览器不能显示这些文件类型时就调用助手。

插件（plug-in）是直接安装到 Web 浏览器的特种助手应用程序，它们使你能够在浏览器窗口中查看多媒体内容。

**By the Way**

这段代码使用 GIF 动画 projector.gif 作为指向 pond.wmv 视频片段的链接。图 12.1 所示为具有投影机图像的 pond 示例页面，当图像被单击，Windows 媒体播放器将被调用并且开始播放电影。

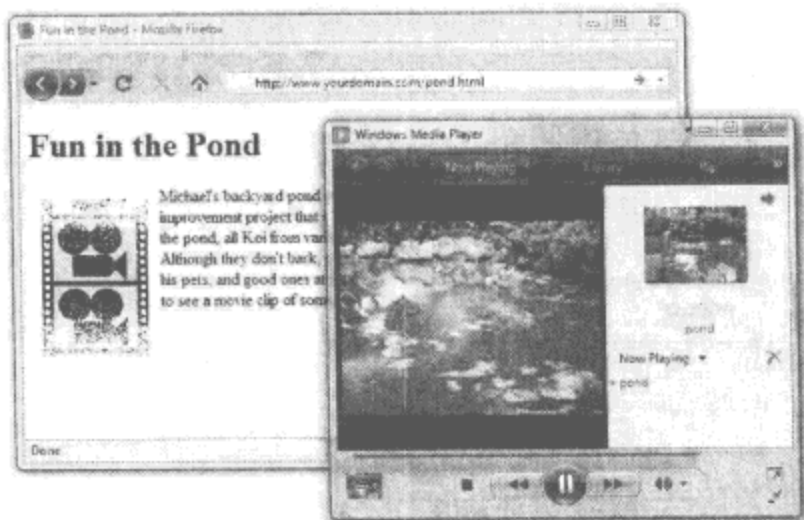


图 12.1

GIF 动画 projector.gif 作为指向 Windows Media 文件的图像链接，启动一个外部助手应用程序

要查看这段视频，你只需要单击投影机动画（或者该段落中的文本链接）。这一操作导致浏览器在插件（如果找到一个能够播放该片段的）的帮助下播放视频或者使用一个合适的助手应用程序。

如果你将 pond.wmv（Windows Media）改为 pond.mov（QuickTime），浏览器对链接的处理不同。如图 12.2 所示，QuickTime 插件不需要启动其他程序，而是让你在浏览器窗口中直接查看电影片段。

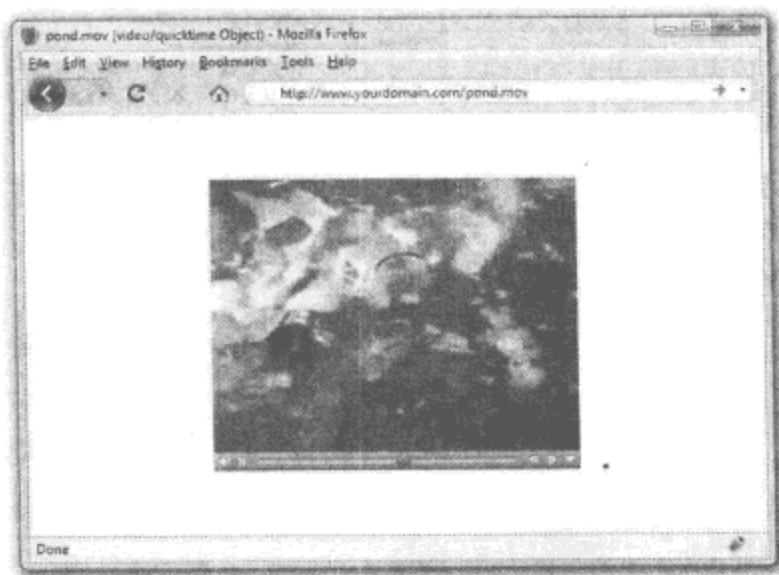


图 12.2

当你选中图像链接，QuickTime 电影 pond.mov 使用 QuickTime 浏览器插件播放

你可能猜到了，这种使用简单链接来播放多媒体文件的方法提供了最好的向后兼容，因为它将确定如何播放多媒体剪辑的所有职责都交给浏览器。这种方法的缺点是你不能控制多媒体剪辑如何播放，且无法直接在网页中播放多媒体剪辑。



**By the Way**

**注意：**如果你的浏览器不支持 QuickTime，可从 Apple 网站免费下载 QuickTime 播放器，其网址为 <http://www.apple.com/quicktime/>。即使安装了 QuickTime，有些浏览器对 QuickTime 视频的播放也将随是否安装了插件而异。例如，在我的 Windows 计算机上，Internet Explorer 和 Firefox 都直接在浏览器窗口中通过插件播放 QuickTime，而 Opera 将 QuickTime 作为助手程序启动。

## 12.2 嵌入多媒体文件

XHTML 包含一个标准的 <object> 标签，这是在网页中嵌入各种多媒体的首选方式。这个标签用于替代在一些 HTML 源代码中仍然能看到的 <embed/> 标签。

将多媒体文件嵌入到页面中产生一组软件控制，使文件能够直接播放——不需要辅助窗口，也没有必要离开所在的页面。下面是使用 <object> 标签嵌入前面看到的池塘视频的代码：

```
<object classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6"
width="320" height="305">
  <param name="type" value="video/x-ms-wmv" />
  <param name="URL" value="pond.wmv" />
  <param name="uiMode" value="full" />
  <param name="autoStart" value="false" />
</object>
```

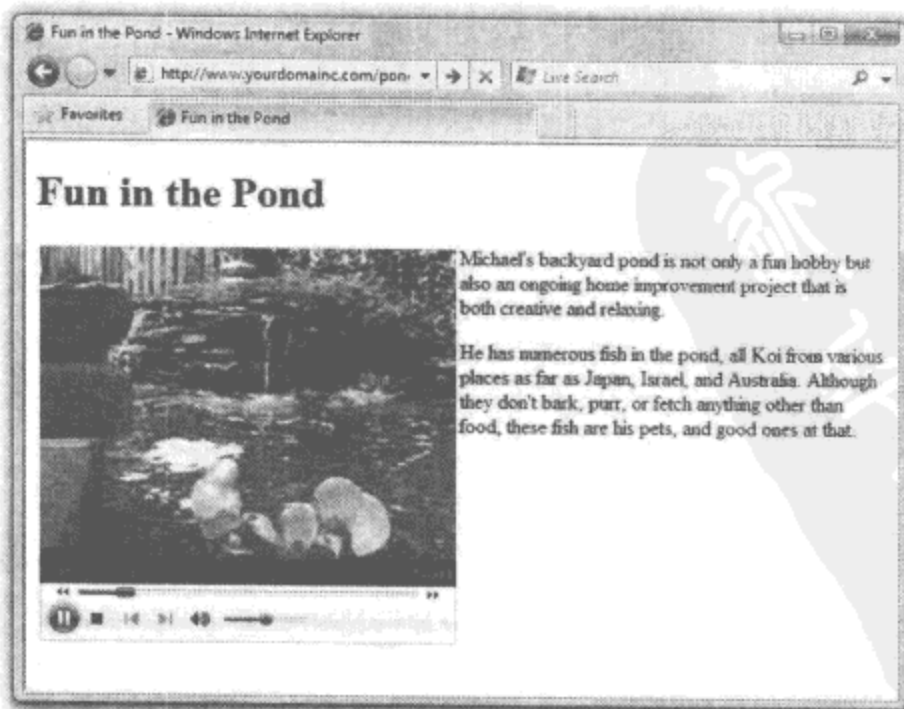
如果考虑到这段代码将一段视频嵌入到网页中，如图 12.3 所示，它并不很复杂。最复杂的部分是 <object> 标签的 classid 属性，它设置为一个很长的字母数字编号。这个编号是 Microsoft Windows 媒体播放器的全局 ID (global ID)，它告诉 <object> 标签将 Windows 媒体播放器嵌入网页中以播放视频剪辑。你可以将这段代码复制并粘贴到自己的网页中。

**By the Way**

**注意：**需要指出的是，Media Player 是一个复杂的媒体播放器，能够自动将多媒体流化，即下载开始的一两秒钟内容后就开始播放，其他内容在播放前面的内容的同时在后台下载，这样访问者在浏览或收听多媒体内容时不必长时间地等待下载完毕。

图 12.3

<object> 标签让你能够在网页中嵌入视频剪辑，并指定用于播放它的媒体播放器



<object>标签的 `width` 和 `height` 属性决定嵌入的 Windows Media Player 窗口的大小。如果没有设置这两个属性，有些浏览器将自动调整窗口的大小以适应内容，而有些将不显示内容——因此要安全地播放，应将这两个属性设置为多媒体内容播放时的大小。

<object>标签内有 4 个 <param> 标签，负责指定关于如何播放剪辑的更多细节。每个 <param> 标签都有两个属性：`name` 和 `value`，负责将数据 (`value`) 同特定设置 (`name`) 关联起来。在这个例子中，媒体剪辑的 URL 设置为 `pond.mov`。第三个参数 `uiMode` 决定 Windows Media Player 可用哪些按钮和用户界面选项：`full` 表示所有用户界面元素都可用，如控制按钮和音量滑块。最后，`autoStart` 参数设置为 `false`，这样，当页面在浏览器中打开时，视频剪辑不自动开始播放。

`type` 参数可能是最复杂的参数。它指定要播放的媒体类型，这里为 Windows Media Video (WMV) 文件。媒体类型必须指定为标准的 Internet MIME 类型之一。

MIME 类型是唯一标识 Internet 上不同类型的媒体对象的标识符。MIME 是 Multipurpose Internet Mail Extensions (多用途 Internet 邮件扩展) 的缩写，之所以取这样的名字，是因为 MIME 类型原来是用于表示电子邮件附件的。这些 MIME 类型应在 <object> 标签的 `type` 属性中使用，用于标识 `data` 属性引用的多媒体对象类型。

下面是几种流行的可在网页中使用的 MIME 声音和视频格式。

- WAV 音频: `audio/x-wav`。
- AU 音频: `audio/basic`。
- MP3 音频: `audio/mpeg`。
- MIDI 音频: `audio/midi`。
- WMA 音频: `audio/x-ms-wma`。
- RealAudio: `audio/x-pn-realaudio-plugin`。
- AVI: `video/x-msvideo`。
- WMV: `video/x-ms-wmv`。
- MPEG 视频: `video/mpeg`。
- QuickTime: `video/quicktime`。

程序清单 12.2 是池塘网页的相关代码，这里你可以看到 <object> 标签出现。

#### 程序清单 12.2 使用 <object> 标签直接嵌入 WMV 视频剪辑

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Fun in the Pond</title>
  </head>

  <body>
    <h1>Fun in the Pond</h1>
    <div style="float:left; padding:3px">
```

```

<object classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6"
width="320" height="305">
  <param name="type" value="video/x-ms-wmv" />
  <param name="URL" value="pond.wmv" />
  <param name="uiMode" value="full" />
  <param name="autoStart" value="false" />
  <embed width="320" height="305" type="video/x-ms-wmv"
src="pond.wmv" controls="All" loop="false" autostart="false"
pluginspage="http://www.microsoft.com/windows/windowsmedia/" />
</object>
</div>
<p>Michael's backyard pond is not only a fun hobby but also
an ongoing home improvement project that is both creative and
relaxing.</p>
<p>He has numerous fish in the pond, all Koi from various places
as far as Japan, Israel, and Australia. Although they don't bark,
purr, or fetch anything other than food, these fish are his pets,
and good ones at that.</p>
</body>
</html>

```

**By the Way**

**注意：**由于 XHTML 不支持 `<embed />` 标签，因此使用该标签的网页是无效的。不幸的是，这个问题实在没有其他解决方法——只能等待浏览器完全支持 `<object>` 标签或者转移到 HTML5 的 `<embed />` 元素。

你可能注意到，还有一些代码没有出现在前面的 `<object>` 标签示例中。不幸的是，正如本章前面提到的，并非所有网页浏览器对 `<object>` 标签的支持都完全一致。因此，必须在 `<object>` 标签内包含 `<embed />` 标签，以处理浏览器的不一致性。这并不是理想的解决方法，但只能这样做，因为浏览器开发商迟迟不能在标准上达成一致。如果仔细研究，将发现 `<object>` 标签包含的信息在 `<embed />` 标签也都包含。

这里介绍的 `<object>` 标签的用法比较简单，这是因为仅播放多媒体内容并不需要知道如何使用 `<object>` 标签的更复杂方面。换句话说，要在网页中共享多媒体剪辑，并不需要成为多媒体专家。

**By the Way**

**注意：**使用 `<object>` 和 `<embed>` 标签不仅能在网页中包含视频文件。任何多媒体文件的添加过程都相同。为了确定准确的 `classid` 和 `codebase` 属性，以及附加的参数（在 `<param>` 标记中），使用搜索引擎查找“对象嵌入 媒体类型”，其中媒体类型是 Real Audio、QuickTime、Flash 或者其他你想要的类型。

## 12.3 使用多媒体的其他技巧

在添加视频，音频或者动画到网站之前，首先问问自己。当你使用这些多媒体类型时，确定这么做的理由。不必要的声音和视频和不必要的图像一样，可能扰乱你的信息。其次，如果你的信息是“关注我制作的视频”或者“倾听我的音乐并下载一些歌曲”，那么多媒体绝对应该在网站上扮演一个角色。

下面是需要记住的一些技巧。

- 不要在页面中包含多媒体并且将它设置为页面装入时自动播放。始终让用户选择开始（和结束）音频或者视频。

- 只要可能，让用户选择多媒体播放器。不要将自己限制在仅能用一种类型的播放器或者在一种操作系统上播放的多媒体内容。
- 多媒体文件比典型的图像和文本文件大，这就意味着你必须在 Web 服务器上有存储它们的空间，请求它们也就需要传输带宽的分配。
- 如果你的网站完全是音频或者视频，提供很少的文本或者图像，要理解你的一些目标用户因为他们的系统或者带宽的限制不能看到或者听到你所介绍的内容，为这些用户提供其他得到你的信息的选项。
- 利用免费的在线视频宿主服务，比如 YouTube (<http://www.youtube.com/>)。YouTube 不仅提供视频剪辑的存储，还将提供给你将视频插入你的网页所需的代码。图 12.4 所示为 YouTube 的“世界上最可爱的宠物”页面。如果你从图中所示的“Embed”区域复制并且粘贴文本，你将得到如下代码：

```
<object width="425" height="344">
<param name="movie"
value="http://www.youtube.com/v/yPxiHd2B0po&rel=0&color1=0xb1b1b1
&color2=0xcfcfcf&feature=player_profilepage&fs=1"></param>
<param name="allowFullScreen" value="true"></param>
<param name="allowScriptAccess" value="always"></param>
<embed
src="http://www.youtube.com/v/yPxiHd2B0po&rel=0&color1=0xb1b1b1&c
olor2=0xcfcfcf&feature=player_profilepage&fs=1"
type="application/x-shockwave-flash" allowfullscreen="true"
allowScriptAccess="always" width="425" height="344"></embed>
</object>
```

然后你可以将这些代码插入网页。



图 12.4

YouTube 为视频文件提供存储以及用于你自己的网页的链接和 `<object>` 代码

## 12.4 总结

本章读者学习了如何在网页中嵌入视频与声音。知道了如何使用一个指向多媒体剪辑的简单链接，这种播放多媒体内容的方法得到了最广泛的支持，但最不灵活。然后学习了如何使用 `<object>` 标签直接在网页中嵌入媒体播放器。不仅如此，还了解到，为最大程度地与所有浏览器兼容，可结合使用标签 `<embed />` 和 `<object>`。标签 `<object>` 和 `<embed />` 可用于包含各种媒体类型，除前面使用过的 AVI、WMV 和 QuickTime 外，还有 WAV、MP3、RealAudio

和 MIDI 文件，而这只是其中的一小部分。

表 12.1 总结了本章介绍的标签。

表 12.1 第 12 章介绍的 HTML 标签与属性

标签/属性	功能
<object>...</object>	在文档中插入图像、视频、Java 程序、ActiveX 控件或其他对象
<param>...</param>	对象的运行阶段设置，如在网页中占据的宽度与高度
属性	
name="名称"	名称参数属性
value="值"	名称参数属性的值
<embed />	嵌入由插件程序读取或显示的多媒体文件，这个标签从技术上说已被摒弃，但由于浏览器并不都支持<object>标签，所以仍然很有用
width="宽度"	嵌入对象的宽度，单位为像素
height="高度"	嵌入对象的高度，单位为像素
type="mimetype"	多媒体内容的 MIME 类型
src="媒体 URL"	嵌入文件的 URL
control="控件"	媒体播放器的用户输入控件的配置，若使用 All，则所有控件都可用
loop="循环"	媒体剪辑只播放一次还是多次播放，设置为 true 或 false
autostart="自动开始"	是否在打开网页时就开始播放媒体剪辑，设置为 true 或 false
pluginspage="插件 URL"	播放媒体所需插件的 URL

## 12.5 问与答

问：我听说很多关于流式视频与音频的内容，它们指的是什么？

答：过去，通过大部分调制解调器下载视频与音频文件都需要花很多分钟甚至几小时，这样严重地限制了视频与音频在网页中的应用。所有人的目标都是朝流式视频与音频发展，它可以一边接收数据一边播放。也就是说，没有完全下载剪辑时，就可以开始看视频或听音频。

现在流式播放在大部分媒体播放器中得到了广泛支持，有独立版本也有插件版本。使用<object>标签嵌入媒体对象时，支持它的媒体播放器将自动地以流方式播放媒体剪辑——如果这个播放器支持流。

问：应如何在视听文件格式 QuickTime、Windows AVI/WAV、RealVideo/RealAudio 和 MPEG 之间做出选择？它们之间有很大的区别吗？

答：QuickTime 是 Macintosh 用户最常用的视频格式，但也有用于 Windows 的 QuickTime 播放器。同样，AVI/WMV 和 WAV/WMA 是 Windows 用户首选的视频和音频格式，但也有支持这些格式的 Macintosh 播放器。MPEG 是另外一个流行的视频与音频标准。MPEG-3 简称为 MP3，是极其流行的高保真度音频标准。另一种基于 MPEG 的音频格式是 Apple 的 AAC 格式，你可能知道它是 iTunes 使用的音乐格式。

如果大部分用户都使用 Windows，就选择视频格式 AVI/WMV 和音频格式 WAV/WMA。如果用户中很大一部分是 Macintosh 用户，就选择 QuickTime 或 RealVideo/RealAudio，或至少将其作为一种可选择的方式。对于 Mac 音频来说，MP3 也是可行的选择。如果跨平台兼容性非常重要，考虑使用 MP3 作为音频格式，RealVideo/RealAudio 作为视频格式——但要观



看 RealVideo/RealAudio 剪辑，必须从 <http://www.real.com/player/> 下载专门的软件。

## 12.6 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 12.6.1 测验

1. 要让尽可能多的用户能够观看你的网站中的视频，最简单的方法是什么？
2. 编写 HTML，在网页中嵌入名为 myvideo.avi 的视频，使所有主要网络浏览器的用户都能够播放它。视频需要在网页中占据 320 像素×305 像素的区域。
3. 如何在 <object> 标签中加入 <param> 标签来重复播放多媒体剪辑？

### 12.6.2 测验答案

1. 只需这样链接它：

```
<a href="myvideo.avi">my video</a>
```

2. 由于视频剪辑是 Microsoft 格式 (AVI)，所以应嵌入一个 Windows Media Player 对象。使用下列 HTML 代码：

```
<object classid="CLSID:6BF52A52-394A-11d3-B153-00C04F79FAA6"
width="320" height="305">
  <param name="type" value="video/x-ms-avi" />
  <param name="URL" value="myvideo.avi" />
  <param name="uiMode" value="full" />
  <param name="autoStart" value="false" />
  <embed width="320" height="305" type="video/x-ms-avi"
src="myvideo.avi" controls="All" loop="false" autostart="false"
pluginspage="http://www.microsoft.com/windows/windowsmedia/">
</embed>
</object>
```

3. <param name="loop" value="true" />

## 12.7 练习

- 创作自己的视频剪辑，然后将它嵌入到网页中，或者从 Web 查找一些免费的剪辑并且练习将它们放入文本中。
- 本章介绍的嵌入媒体的方法和标签也可用于 Macromedia Flash 文件。要了解如何使用 Flash 在网页中放置交互式的动画，请参见 Flash 主页，其网址为：<http://www.macromedia.com/software/flash/>。



# 第 13 章

## 使用框架

本章中你将学到：

- 建立框架集的方法
- 在框架和窗口之间链接的方法
- 使用内联框架的方法

你可能见过这样的网站，浏览器窗口似乎允许在几个不同的网页间切换。真相是浏览器让用户能够将浏览器窗口分为几个包含不同网页的区域，从而一次看到多个页面，这些区域称为框架（Frame）。当然，从用户的角度看，所有 Web 内容形成一个窗口，但实际上有几个不同的网页在工作。

### 13.1 什么是框架？

框架是浏览器窗口中的一个矩形区域，每个框架显示一个网页，与其他网页的框架相邻。乍一看，如图 13.1 所示，框架与普通网页没什么不同，但实际上它是两个不同的 HTML 页面，显示在同一个 Web 浏览器窗口中。每个网页都显示在自己的框架中，两个框架上下相邻，由一条水平线分开。

图 13.1

框架允许一次显示  
超过一个网页



如果使用框架，它们一般被用于创建利用静态的导航链接集的框架化网站；你可以从图 13.1 上面的框架中看到这些链接。单击这个例子中的链接时，上面的框架不变，而下面的框架将加载并显示一个新网页，如图 13.2 所示。



图 13.2

单击 Products 链接在下半部分打开一个新的页面但是上面的框架保持不变

你应该知道，框架很长时间以来是 Web 设计中的一个争论不休的问题。它的优点从来没有真正超过缺点，由于浏览器对 HTML 和 CSS 标准的不同支持，框架被看作达到某些目标的一种方法，而不考虑其缺点。作为 Web 开发人员，我不建议使用框架，原因如下。

- 框架与 Web 的基础概念冲突，它是可以通过一个 Web 地址（URL）访问的 Web 内容实例之间的超文本连接。
- 框架化的网站的打印部分非常困难，除非你单击了希望打印的具体框架，并且选择上下文菜单中的“打印此框架”（如果有这个项目），否则打印的将是框架集本身而没有内容。
- 如果框架缺乏合适的编码，或者有合适的编码但是被用于邪恶的目的，用户可能在框架化的网站中无法看到框架外部的内容。
- 框架历史上被用来代替 Web 开发和设计的方法、专业、易于理解的方法。没有理由在可以使用 CSS 布局这种更好的选择的情况下选择较差的方案。
- 因为这些（和其他）原因，框架已经从 HTML 5 标准中移除。未来，过时的 `<frame/>`、`<frameset>` 和 `<noframes>` 标签将不可用。

尽管有这些缺点，你仍然将在本章中学习如何创建非常简单的框架化网站。你仍然很可能遇到框架化网站，可能需要知道如何才能在不使用框架的情况下重新创建这种观感。在这种情况下，理解框架的构建很重要，这样你才能成功地将其解体。

此外，你将学习框架的一种类型——`<iframe>`，它起很重要的作用，并将在 HTML 5 中继续存在。

本书稍后，你将学习如何使用 XHTML 和 CSS 生成不使用框架的相同功能。现在，随着这个简单的框架化网站的构建，你可以学习将来应该撤销并且以不同方式重新创建的一个过程。

## 13.2 建立框架集

这个小节说明如何创建如图 13.1 和图 13.2 所示的简单框架化网站。每个框架的内容被

创建为普通的网页。这些页面是 `top.html` (用于导航)、`home.html`、`products.html`、`services.html`、`contact.html`。这些页面包含的标签都是在其他章节中见过的标签。一个被称为框架集 (Frameset) 文档的特殊页面被用来将页面放在一起。在这个例子中, 这个文档为 `index.html`。

### 13.2.1 创建一个框架集文档

框架集文档是一个 HTML 页面, 指示 Web 浏览器将窗口分为几个框架, 并指定每一框架应显示哪个网页。

框架集文档实际上并不包含内容, 它只是告诉浏览器要加载哪些网页以及如何在浏览器窗口中排列这些网页。程序清单 13.1 显示了如图 13.1 和图 13.2 所示的框架化网站的框架集文档。

程序清单 13.1 图 13.1 中所示的网站的框架集文档

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Sample Framed Site</title>
  </head>

  <frameset rows="50,*">
    <frame src="top.html" name="top" />
    <frame src="home.html" name="main" />
  <noframes>
    <body>
      <h1>Sample Framed Site</h1>
      Your browser does not support frames. Sorry!
    </body>
  </noframes>
</frameset>
</html>
```

这个程序清单包含一个 `<frameset>` 标签而没有 `<body>` 标签。通常, 可用于 `<body>` 标签内的标签都不能用于 `<frameset>` 标签内。在这个例子中, `<frameset>` 标签包含一个 `rows` 属性, 意思是框架应上下相邻, 就像表格中的行。如果要使框架左右相连, 使用 `cols` 属性代替 `rows` 属性。

#### By the Way

**注意:** 这个网页中使用的 DTD 不是本书一直使用的 XHTML 1.1 DTD。这是因为标准 XHTML 1.1 DTD 不支持框架。因此, 要让使用框架的网页有效, 必须使用 XHTML 1.0 Frameset DTD, 这是专门为使用框架的网页设计的 DTD。

必须指定 `rows` 或 `cols` 的大小, 这可使用精确的像素值或浏览器窗口的百分比。也可用星号 (“\*”) 来表明框架应尽可能占据窗口的可用空间。如果有多个框架有 \* 值, 剩余的空间将在这些框架之间平均分配。

在程序清单 13.1 中, `<frameset rows="50,*">` 表示将浏览器窗口垂直分为两框架。上面的框架高 50 像素, 下面的框架占据剩余的窗口空间。上面的框架包含文档 `top.html` (见程序清单 13.2), 下面的框架包含 `home.html` (见程序清单 13.3)。

### 程序清单 13.2 框架化网站样板的导航栏 top.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Sample Framed Site</title>
  </head>

  <body style="background-color:#0000FF;">
    <div style="text-align:center;color:#FFFFFF;font-weight:bold;
font-size:16pt">
      <a style="color:#FFFFFF;" href="home.html"
target="main">HOME</a> ::
      <a style="color:#FFFFFF;" href="products.html"
target="main">PRODUCTS</a> ::
      <a style="color:#FFFFFF;" href="services.html"
target="main">SERVICES</a> ::
      <a style="color:#FFFFFF;" href="contact.html"
target="main">CONTACT</a>
    </div>
  </body>
</html>
```

**提示：**在程序清单 13.1 的框架集后，我在<body>标签和</body>标签之间包含了完整的网页。注意，这个网页在图 13.1 或图 13.2 中根本没有显示出来。支持框架的所有网页浏览器都会忽略<noframes>标签和</noframes>标签之间的所有内容。

当今所有主要浏览器都支持框架，因此现在框架的兼容问题不像几年前那么重要了。即使这样，包含<noframes>标签是很容易的，应考虑到那些仍使用旧浏览器的用户。

**Did you Know?**

### 程序清单 13.3 框架化网站模板中的一个内容框架 home.html

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Sample Framed Site</title>
  </head>

  <body style="background-color:#FFFFFF">
    <h1 style="text-align:center">Sample Framed Site: Home</h1>
    <p style="text-align:center">This is an example of the "home"
page.</p>
  </body>
</html>
```

**注意：**程序清单 13.2 和程序清单 13.3 使用 XHTML 1.0 Transitional DTD。虽然 XHTML 1.1 DTD 更新也更严格，但为使文档有效，使用框架时必须遵守 XHTML 1.0，因此在框架内显示的网页也应使用 XHTML 1.0。

**By the Way**

这个例子中，上面的导航框架有 50 个像素的固定高度。但因为不能预测用户浏览网页时窗口的大小，因此通常使用百分比来指定行和列的大小比使用像素更方便。例如，要使左边的框架占据浏览器窗口宽度的 20%，右边的框架占据余下的 80%，可用下列的代码：

```
<frameset cols="20%,80%">
```

当你以像素为单位指定框架大小，最好在相同框架集中包含至少一个可变 (\*) 宽度的框架，这样文档将能够充满任何尺寸的窗口。

### 13.2.2 添加单独的框架

在<frameset>标签和</frameset>标签之间，应包含<frame />标签，指定每个框架要显示哪个 HTML 文档。如果<frame />标签的数量比<frameset>标签内定义的框架少，剩下的框架将为空。

在每个<frame>标签包含 src 属性，指明要在该框架中加载的网页地址。如果框架只包含一个图像，可使用图像文件的地址。

## 13.3 在框架和窗口之间链接

在<frame />标签中，使用 name 属性赋予框架唯一的名称后，框架真正的威力便开始显示出来。然后，可以使用<a>标签的 target 属性，使网页中的链接改变框架的内容。例如，程序清单 13.1 包含下列标签：

```
<frame src="home.html" name="main" />
```

### By the Way

**注意：**从技术上说，name 属性已经过时了，已被 id 属性取代。然而，现在的 Web 浏览器在识别目标框架时仍然依赖于 name 而不是 id。因此，现在需要使用 name 属性来标识框架。当然，两个属性都用也可以。

这段代码在框架中显示 home.html 页面，并将该框架命名为 main。

在上面的框架代码（程序清单 13.2）中，你看到如下链接：

```
<a style="color:#FFFFFF;" href="services.html"
  target="main">SERVICES</a>
```

当用户单击这个链接，services.html 将显示在名为 main（下面的框架）的框架中。如果省略 target="main" 属性，services.html 将显示在当前框架（上面的）中。

为节省空间，我没有提供 services.html 页面的程序清单，它只是一个普通网页，没有与框架相关的特性。从图 13.2 中可以看到该网页。

你可以在框架代码中使用一些 HTML 属性来去掉框架分隔符，减少边距大小来增加小的框架中的空间，以及强制框架没有滚动条。如图 13.3 所示，程序清单 13.4 显示了程序清单 13.1 代码的一个修改版本。该代码作了两处修改，添加了<frame>标签的两个属性：scrolling="no"和 frameborder="0"。

**程序清单 13.4 图 13.3 中所示的网站的框架集文档**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Sample Framed Site</title>
  </head>

  <frameset rows="50,*">
    <frame src="top.html" name="top" scrolling="no" frameborder="0" />
    <frame src="home.html" name="main" scrolling="no" frameborder="0" />
  </frameset>
</html>
```



图 13.3

这是程序清单 13.4 中添加了 `<frame>` 标签的属性之后的代码所显示的网页

## 13.4 使用内联框架

内联 (inline) 框架没有普通框架的可用性问题的，但是内联框架的用途不同。`<iframe>` 不是纯粹的布局技巧，而更像 `<object>` 标签——用于放置现有文档中的一大块内容。在 `<object>` 标签中的这一大块一般是多媒体。你可以使用 `<iframe>` 嵌入整个 HTML 文档、图像或者其他资源。程序清单 13.5 和程序清单 13.6 展示了生成如图 13.4 所示的内联框架的代码。

**程序清单 13.5 调用 `<iframe>` 的 XHTML 代码**

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Using an iframe</title>
  </head>
  <body style="background-color:#CCCCCC">
    <h1 style="text-align:center">Inline Frame Example</h1>
```



```

<div style="text-align:center">
  <iframe src="iframe_src.html"
    style="width:500px;height:100px;border:1px solid black;
    background-color:#FFFFFF">
    <p>Uh oh...your browser does not support iframes.</p>
  </iframe>
</div>
</body>
</html>

```

在程序清单 13.5 中，你还没有遇见过的唯一 XHTML 代码就是 `<iframe>` 本身。你可以看到它需要一个 `src` 属性值——这是来源——你可以使用样式定义宽度、高度、边框类型和背景颜色（以及其他）。程序清单 13.6 显示了 `<iframe>` 的源代码，这只是一个具有文本和一些样式的常规文件。

### 程序清单 13.6 程序清单 13.5 中调用的 `<iframe>` 源代码

```

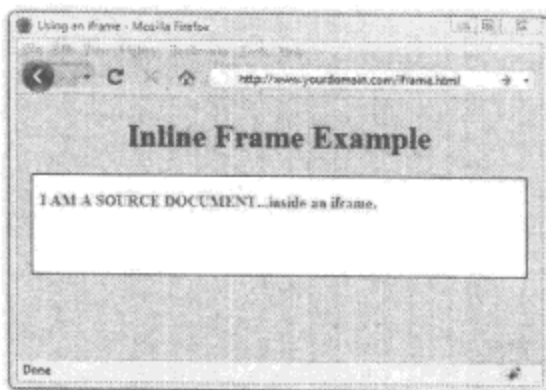
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>iframe source</title>
  </head>
  <body>
    <p style="color:#FF0000;font-weight:bold">I AM A
    SOURCE DOCUMENT...inside an iframe.</p>
  </body>
</html>

```

图 13.4

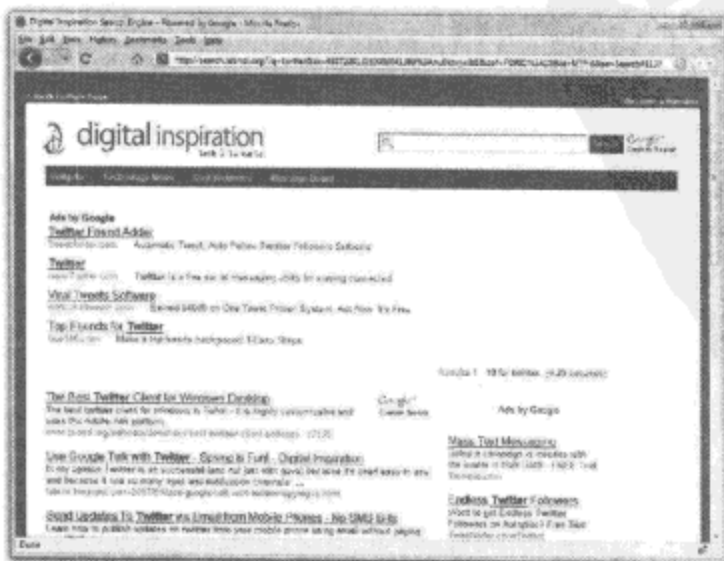
程序清单 13.5 调用程序清单 13.6 中的内联框架代码



内联框架常常用于从其他网站中引入内容。常见的用法包括从第三方广告服务为用户提供广告以及使用 Google 的网站搜索为用户显示搜索结果（利用 Google 的搜索技术）。图 13.5 所示为一个 `<iframe>` 用于将搜索引擎引入自定义网站模板的例子。

图 13.5

使用 `<iframe>` 显示 Google 自定义搜索结果



在图 13.5 中，白色区域中的所有内容是<iframe>的实际内容，源代码是 Google 网站上的一个脚本，它运行后显示 Digital Inspiration 网站上的模板中的内容。如果你仔细观察图 13.5——我不相信你能够在图中看到它——你可以看到实际的<iframe>周围有一个暗灰色的边框。

和本章中学到的<frame/>不同，<iframe>仍然是 HTML 5 的一部分。

## 13.5 总结

本章中你学习了如何通过将网页浏览器窗口划分为框架，从而同时显示多个页面。你学习了使用框架集文档来定义框架的大小与排列方式，以及每个框架要载入的网页或图像；学习了如何创建链接来改变选定框架的内容，同时保持其他框架不变；还了解到一些可选设置，它们可以控制框架的可调整边框和滚动条；最后，你学习了如何使用内联框架显示来自自己的网站或者其他网站的内容。

表 13.1 总结了本章介绍的标签与属性。

表 13.1 第 16 章介绍的标签与属性

标签/属性	功能
<frame />	在<frameset>中定义框架
<b>属性</b>	
src="url"	要在框架中显示的文档的 URL
id="名称"	在<a href>链接的 target 属性中指向这个框架时使用的名称。符合 XHTML 标准，但在当前网络浏览器对框架的支持中这个属性并不太受支持
name="名称"	在<a href>链接的 target 属性中指向这个框架时使用的名称。最终将被 id 取代，但现在仍然在当前网络浏览器中使用
scrolling="yes/no/auto"	决定一个框架是否有滚动条，可能的值有 yes、no 和 auto
noresize="noresize"	防止用户使用鼠标调整框架（和邻近框架）的大小
<frameset>..</frameset>	将主窗口划分成一组可以显示不同文档的框架
row="行数"	将窗口或框架集纵向地分为数行，可用数字（如 7）、窗口宽度的百分比（如 25%）或星号（*）来指定。星号表示这个框架占据剩余的所有空间或平均分配这些空间（如果指定了多个*框架）
cols="列数"	效果与 rows 相似，只是将窗口或框架集横向地分为数列
frameborder="yes/no"	指定框架是否显示边框，选项有 1（是）与 0（否）
<noframes>...</noframes>	针对不支持框架的浏览器，在<frameset>文档中提供一个替代文档主体（通常包含<body> ...</body>）
<iframe>...</iframe>	创建内联框架，<iframe>接受所有<frame />接受的属性

## 13.6 问与答

**问：**我是否可以在框架中显示来自 Internet 的其他人的网页，同时在另一个框架中显示我自己的页面？如果那些网站也使用框架，将发生什么？

**答：**可以在框架中加载 Internet（或 Intranet）中的任何文档。如果这个文档是框架集，其框架将调整大小以适应加载它的框架。

例如，可以在框架中放入最喜爱的链接列表，然后让这些链接引用的网页显示在单独的

框架中。这样可以很容易地提供到其他网站的链接，且不会使用户迷失而无法返回你的网站。

另外，将别人的网页作为框架将导致它们看起来是你网站的一部分，这将给你带来法律方面的麻烦，所以，如果将别人的网页放在你的框架中，要确保得到了网页的制作者的书面许可，就像将他们的网站中的图像或文字放在你的网页中一样。

问：需要在所有框架中放入<title>吗？如果这样做，哪个标题将显示在窗口顶部？

答：只有框架集文档的标题会显示。框架集文档不要求<head>和<title>这两个标签，但所有页面都有标题是个好主意，以防有人在框架外打开其中一个页面。

## 13.7 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 13.7.1 测验

1. 编写 HTML 代码，在一个占浏览器窗口左边 25% 的框架中列出名字 Mickey、Minnie 和 Donald，并使得用户单击每个名字时，占浏览器窗口右边 75% 的框架中将出现相应的网页。
2. 生成一个无边框的，具有 98% 页面宽度、250 个像素高的白色背景的<iframe>，应该使用什么样的代码？

### 13.7.2 测验答案

1. 需要 5 个不同的 HTML 文档。第一个文档是框架集：

```
<html>
  <head>
    <title>Our Friends</title>
  </head>

  <frameset cols="25%,75%">
    <frame src="index.html" />
    <frame src="mickey.html" name="mainframe" />
  </frameset>
</html>
```

然后，要将 index.html 文档用于左边的框架：

```
<html>
  <head>
    <title>Our Friends Index</title>
  </head>

  <body>
    <p>Pick a friend:</p>
    <p><a href="mickey.html" target="mainframe">Mickey</a><br />
      <a href="minnie.html" target="mainframe">Minnie</a><br />
      <a href="donald.html" target="mainframe">Donald</a></p>
  </body>
</html>
```



最后，需要 3 个 HTML 页面，其名称分别为 `mickey.html`、`minnie.html` 和 `donald.html`。它们包含各个朋友的信息。

## 2. 使用如下代码：

```
<iframe src="some_source.html"
style="width:98%;height:250px;border:none;
background-color:#FFFFFF">
<p>Put message here for people not able to see the inline frame.</p>
</iframe>
```

## 13.8 练习

- 考虑你使用框架化布局的可能原因，然后在纸上描画出这个布局。保存这个草稿，用于接下来的课程中使用 XHTML 和 CSS 设计布局的学习，这些技术提供给你与标准兼容并且用户友好的方法来达到相似的显示效果和功能。
- 考虑在网站中使用一两个 `<iframe>` 的方式——可能用于广告，或者用于利用免费的 Google 网站搜索。在你的设计中为这个元素留下空间。



## 第 14 章

# 使用边距、填充、对齐和浮动

本章中你将学到：

- 在元素周围添加边距的方法
- 在元素中添加填充的方法
- 对齐的方法
- float 属性的使用

现在你已经学习了一些创建 Web 内容的基础知识，本章将学习使用 CSS 改进这些内容的实质。在前面的章节中，你已经学习了如何将基本的 CSS 用于显示的目的（如字体大小和颜色）。在接下来的章节中，你将开始使用 CSS 控制整个网页而不只是单独的文本或者图形的各个方面。

在处理页面布局之前，单独理解 4 个 CSS 属性之后再将它们组合起来是很重要的。

- margin 和 padding 属性——用于添加元素周围的间距。
- align 和 float 属性——用于将元素相对其他内容放置。

本章中提供的示例不是所创建过的内容中最时髦的，但是它们的意图不在于此。这些示例清晰地说明了 XHTML 和 CSS 的协同。通过本章和其他章节，一旦你掌握了 CSS，你就可以创建基于 Web 的如图 14.1 所示的 CSS Zen Garden（CSS 禅意花园）那样的杰作。

CSS Zen Garden 网站看上去可能不像你平时访问的典型的电子商务或者社交网络网站。这些网站展示了使用 CSS 所能表现的艺术能力。别误会，这些网站经过了仔细的思考和计划，但是可能的设计是无限的。

**By the  
Way**

**注意：** CSS Zen Garden (<http://www.csszengarden.com/>) 中的网站展示了可以通过兼容标准的 CSS 完成的设计类型。这个花园中的所有用户提交的条目使用完全相同的 HTML 文件，但是艺术家们自由地修改 CSS 文件来创建自己的显示效果。图 14.1 中所示的例子是由 Stuff and Nonsense (<http://www.stuffandnonsense.co.uk/>) 的 Andy Clark 创作的。

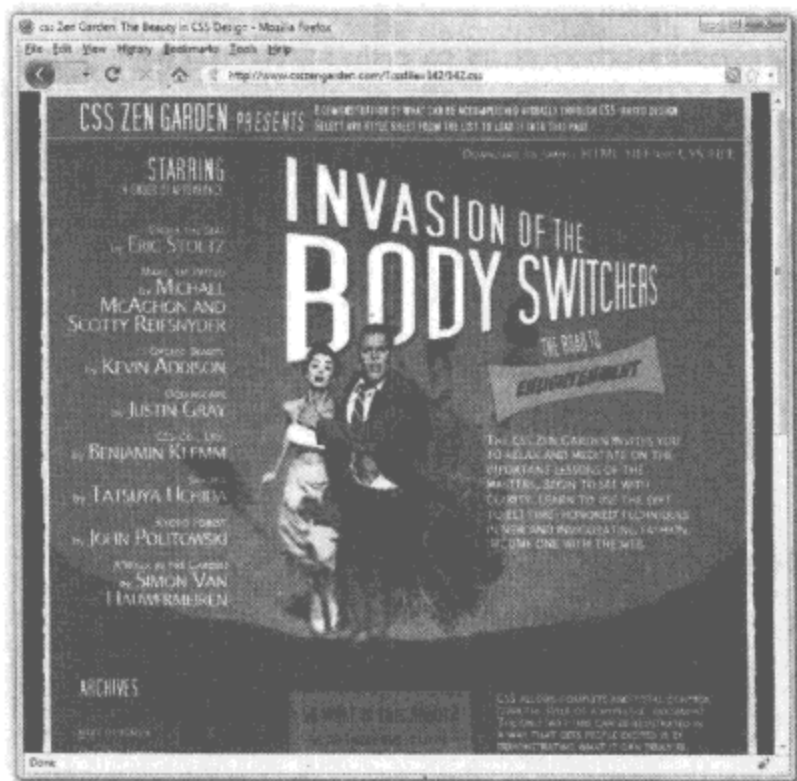


图 14.1

CSS Zen Garden 的许多 XHTML 和 CSS 工作示例之一

## 14.1 使用边距

样式单边距 (**margin**) 使你能够在网页上的元素的矩形区域外部添加空隙。需要记住的是, **margin** 属性处理的是元素外面的间距。

以下是设置边距的样式属性。

- **margin-top**: 设置上边距。
- **margin-right**: 设置右边距。
- **margin-bottom**: 设置下边距。
- **margin-left**: 设置左边距。
- **margin**: 在一个属性中设置上、下、左、右边距。

可以使用单独的边距属性来指定边距,也可以只使用 **margin** 属性来指定。边距可以设置为自动,这意味着浏览器自己设置边距为具体的长度(像素、点、**ems**)或者百分比。如果要将边距设置为百分比,别忘了百分比是基于整个网页而不是元素的大小的。因此,如果将 **margin-left** 属性设置为 25%,元素的左边距将是整个页面宽度的 25%。

程序清单 14.1 中的代码在页面上生成 4 个矩形,如图 14.2 所示,每个矩形宽度为 250 个像素,高度为 100 个像素,并且有 5 个像素的实线黑色边框,每个矩形(这个例子中为 `<div>`)有不同的背景颜色。我们希望每个 `<div>` 周围的边距为 15 个像素,可以使用如下代码:

```
margin-top:15px;
margin-right:15px;
margin-bottom:15px;
margin-left:15px;
```

你也可以使用简写的 **margin** 属性:

```
margin: 15px 15px 15px 15px;
```

当你使用边距属性(填充或者边框)并且希望所有 4 个值都相同,可以进一步简化:



```
margin: 15px;
```

### 程序清单 14.1 生成 4 个具有边框和边距的彩色<div>的简单代码

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Color Blocks</title>
    <style type="text/css">
      div {
        width:250px;
        height:100px;
        border:5px solid #000000;
        color:black;
        font-weight:bold;
        text-align:center;
      }

      div#d1 {
        background-color:red;
        margin:15px;
      }

      div#d2 {
        background-color:green;
        margin:15px;
      }

      div#d3 {
        background-color:blue;
      }

      div#d4 {
        background-color:yellow;
        margin:15px;
      }
    </style>
  </head>

  <body>
    <div id="d1">DIV #1</div>
    <div id="d2">DIV #2</div>
    <div id="d3">DIV #3</div>
    <div id="d4">DIV #4</div>
  </body>
</html>
```

#### **By the Way**

**注意：**你可能记得至少两种不同的简写顺序。第一种是，如果你将元素看作一个矩形，从上边开始，按照顺时针方向：上边、右边、下边、左边。你也可以使用首字母助记法“TRBL”，发音为“trouble”（麻烦），这个词能够表现你忘记边距属性的顺序时的状态。

还要指出一点，TRBL 顺序对填充和边框属性也有效。

实际上有 3 种使用简写设置边距、填充或者边框的方法，这根据设置属性时使用多少个数值而定。

- 一个值：所有边距的大小。
- 两个值：上、下边距和左、右边距（顺序是先上下后左右）。
- 4个值：上、下、左、右边距（顺序是上、下、左、右）。

你可能觉得坚持使用一个值或者所有4个值更容易些，但是这确实不是必须的。

接下来，处理程序清单 14.1 中的样式单中的 `margin` 属性，让我们来改变边距。在这个例子中，你不能真正地看到这些 `<div>` 元素周围的右边距，因为它们的右边没有其他内容并且没有被右对齐。记住这一点，我们将所有元素的 `margin-right` 设置为 `0px`。除此之外，接下来的目标是产生如下效果。

- 第一个色块周围没有边距。
- 第二个色块的左边距为 15 个像素，上边距为 5 个像素，没有下边距。
- 第三个色块左边距为 75 个像素，没有上下边距。
- 第四个色块左边距为 250 个像素，上边距为 25 个像素。

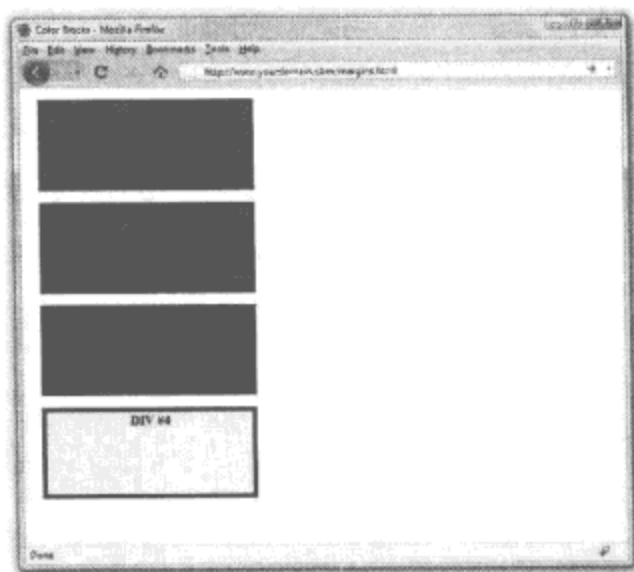


图 14.2

基本色块样板页面显示四个色块，每个都有相同的边距

这看上去很简单——第一个色块不设置边距。由于我们在第二个色块上边设置边距，所以即使我们没有指定第一个色块的边距，但是两个色块之间将会有有一个可见的边距。

4 个命名 `<div>` 的新样式单条目将为：

```
div#d1 {
  background-color:red;
  margin:0px;
}

div#d2 {
  background-color:green;
  margin:5px 0px 0px 15px;
}

div#d3 {
  background-color:blue;
  margin:0px 0px 0px 75px;
}

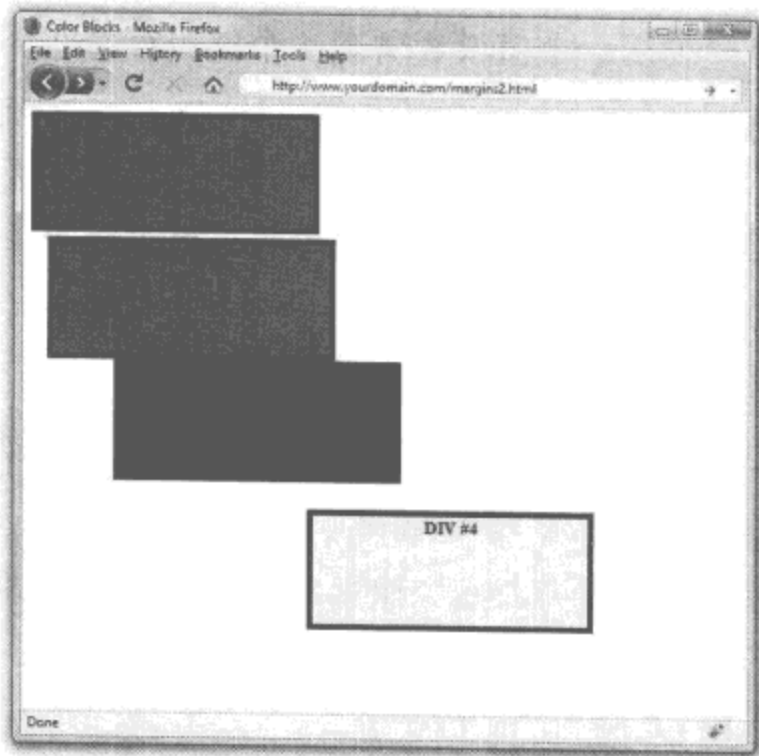
div#d4 {
  background-color:yellow;
  margin:25px 0px 0px 250px;
}
```



程序清单 14.2 的结果如图 14.3 所示，看上去很不规则，但是对于指出一些其他要点是相当有帮助的。例如，回忆一下目标之一是在第一个色块周围不生成边距，你可能预想色块的边框将会和浏览器窗口平齐。但是在图 14.3 中可以看到，页面内容和浏览器窗口边框有一个清晰的空隙。

图 14.3

对色块样板页面的修改显示了不同的边距



#### 程序清单 14.2 生成具有边框，边距和填充的 4 个彩色<div>的简单代码

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Color Blocks</title>
    <style type="text/css">
      body {
        margin:0px;
      }
      div {
        width:250px;
        height:100px;
        border:5px solid #000000;
        color:black;
        font-weight:bold;
        margin:25px;
      }

      div#d1 {
        background-color:red;
        text-align:center;
        padding:15px;
      }

      div#d2 {
        background-color:green;
        text-align:right;
        padding:25px 50px 6px 6px;
      }

      div#d3 {
        background-color:blue;
```



```
        text-align:left;
        padding:6px 6px 6px 50px;
    }

    div#d4 {
        background-color:yellow;
        text-align:center;
        padding:50px;
    }
</style>
</head>

<body>
  <div id="d1">DIV #1</div>
  <div id="d2">DIV #2</div>
  <div id="d3">DIV #3</div>
  <div id="d4">DIV #4</div>
</body>
</html>
```

如果我们要进行元素布置——这是下一章的学习内容——这个空隙将在你的布局中造成问题。为了确保位置和边距从浏览器的边缘开始计算，你需要处理<body>元素本身的边距。在这个例子中，你将需要在样式单中增加如下代码：

```
body {
  margin:0px;
}
```

需要记住的另一点是，如果你有两个有边框的元素互相堆叠，但是两者之间没有边距，它们的接触点将显示出有一个双倍的边框。然后你可能考虑将上面元素的border-bottom减半，而将下面的元素的border-top也减半。这样，边框看起来和其他各边相同。

而且，你可能想过，通过使用250个像素（<div>的宽度）的左边距，第4个色块将从第3个色块结束的地方开始。然而情况不是这样的，因为第3个色块有75个像素的左边距。为了使它们这样排列，第4个div的左边距值应该为325个像素。

将样式修改为下面的代码，将产生如图14.4所示的空隙。这使<body>元素没有边距，从而确保25个像素的左边距真的距离浏览器窗口边缘25个像素。这也使堆叠的第2个和第3个色块不会出现双倍的边框。此外，第4个色块从第3个色块结束的地方开始。

```
body {
  margin:0px;
}
div {
  width:250px;
  height:100px;
  color:black;
  font-weight:bold;
  text-align:center;
}
div#d1 {
  border:5px solid #000000;
  background-color:red;
```



```
margin:0px;
}
div#d2 {
border-width:6px 6px 3px 6px;
border-style:solid;
border-color:#000000;
background-color:green;
margin:10px 0px 0px 15px;
}
div#d3 {
border-width:3px 6px 6px 6px;
border-style:solid;
border-color:#000000;
background-color:blue;
margin:0px 0px 0px 15px;
}
div#d4 {
border:5px solid #000000;
background-color:yellow;
margin:0px 0px 0px 265px;
}
```

图 14.4

对色块的第三次  
修改拉近它们之  
间的关系

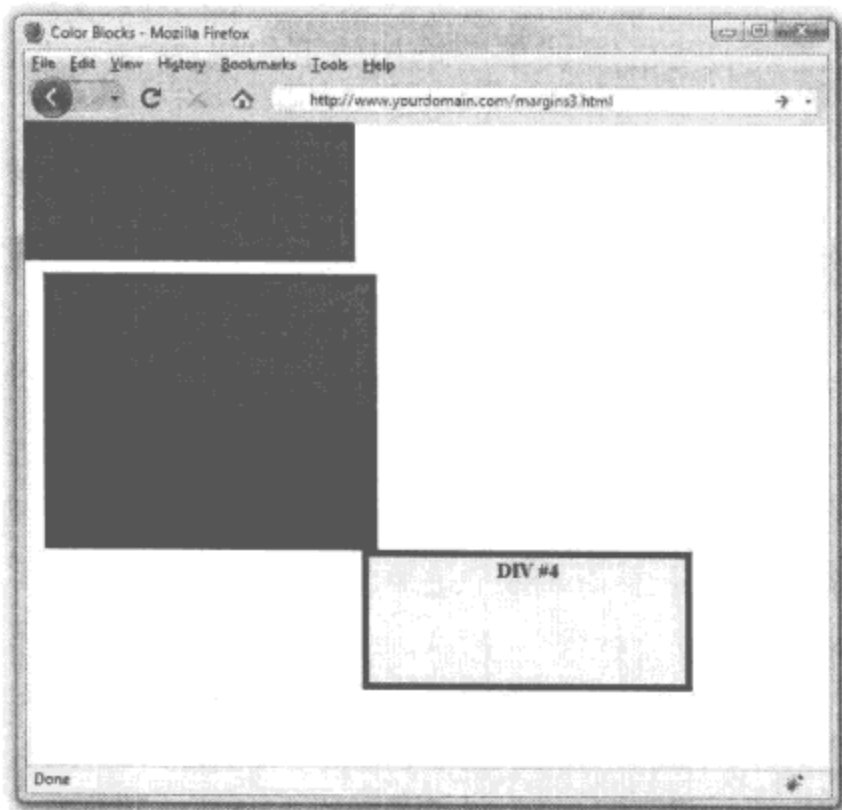


图 14.4 显示在第三个色块的右边和第四个色块的左边有一些重叠。如果色块宽度为 250 个像素，第 3 个色块的 `margin-left` 值为 15 个像素，那么第 4 个色块的左边距为 265 个像素，为什么会出现重叠？是的，确实有 265 个像素的边距，但是这个边距是不够的，因为我们还必须考虑 6 个像素的边框。如果我们将第 4 个色块的 `margin` 属性改变为如下的代码，第 3 个和第 4 个色块将按照计划排列，如图 14.5 所示：

```
margin: 0px 0px 0px 276px;
```

在这些示例中可以看到，边距规格对于元素的位置很有用，但是你必须小心设置这些规格。

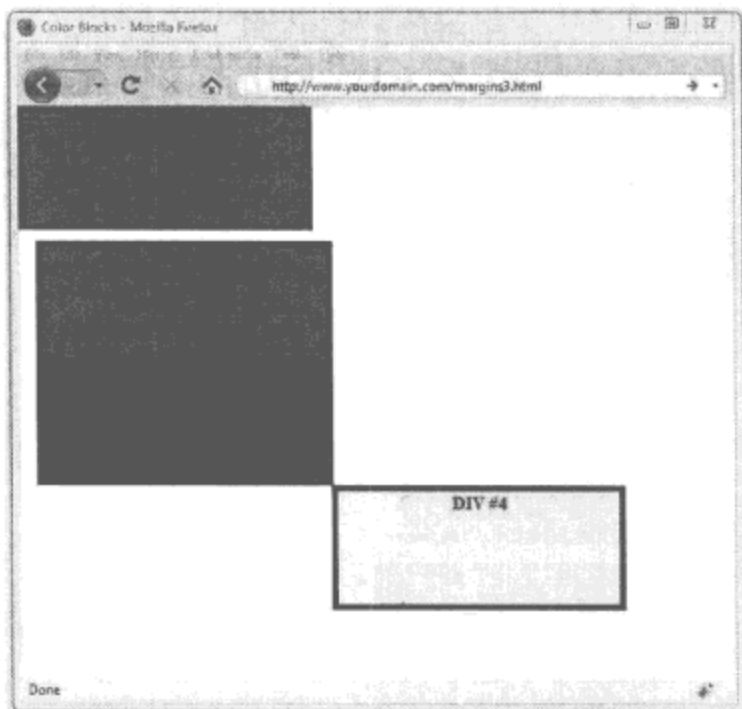


图 14.5

改变边距考虑 11 个像素的边框宽度

## 14.2 填充元素

填充（padding）和边距类似，为元素添加额外的空间，但是这些空间所在的位置有很大不同。回忆一下，边距被添加到元素外部，而填充在元素的矩形区域内部添加空间。举个例子，如果你创建一条样式规则，建立一个宽度为 50 像素，高度为 30 像素的元素，然后设置填充为 5 个像素，那么剩余的元素区域将为 40 像素×20 像素。而且，因为元素的填充出现在元素的内容区域之中，它将具有和元素内容相同的样式，包括背景颜色。

可使用填充属性之一来指定样式规则的填充，这些属性与边距属性很像。下面是可用于为样式规则的设置填充的填充属性。

- padding-top: 设置上填充。
- padding-right: 设置右填充。
- padding-bottom: 设置下填充。
- padding-left: 设置左填充。
- padding: 通过单个属性设置上、下、左、右填充。

与边距一样，可以使用单独的填充属性来设置样式规则的填充，也可以只用 padding 属性。填充也可用度量单位或百分比来表示。

下面的例子说明了如何设置样式规则的左、右填充，在元素内容的左、右两边空 10 像素：

```
padding-left:10px;  
padding-right:10px;
```

与边距一样，可用一个属性（padding 属性）来设置元素的所有填充。与 margin 属性一样，可使用 3 种方法来设置填充属性。下面的例子将样式规则的垂直（上、下）填充设置为 12 像素，将水平（左、右）填充设置为 8 像素：

```
padding:12px 8px;
```

下面是一个更清晰的实现相同效果的例子，它指定了所有填充值：

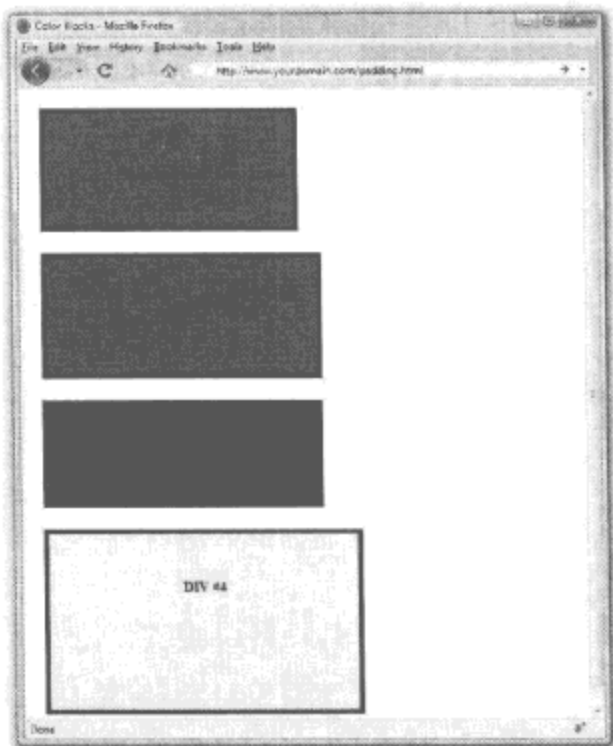
```
padding:12px 8px 12px 8px;
```



在前面的所有图中，你会注意到文本 DIV#1、DIV#2 等出现在色块的上面，文本和边框之间只有一点点空隙。这些空隙不是由填充值指定的，但是作为一种元素中的默认值出现。如果你希望对元素填充有精确的控制，程序清单 14.2 展示了一些例子。所有色块都有 250 个像素宽，100 个像素高，有 5 个像素的黑色实线边框，以及 25 个像素的边距，如图 14.6 所示。有趣的是每个<div>的填充值。

图 14.6

基本色块样板页面，显示 4 个有不同填充的色块

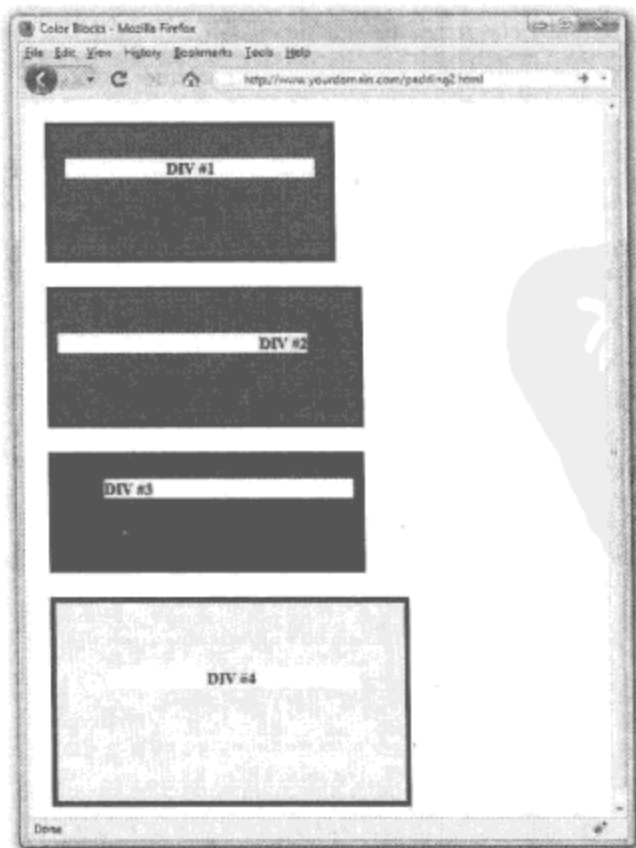


你应该立即意识到，这个示例中有些错误。色块应该为 250 个像素宽，100 个像素高。图 14.6 中的色块不统一，这是因为尽管我们控制<div>的大小，但是稍后应用的填充覆盖了原始尺寸声明。

如果你在一个<p>元素中放置文本并且给予元素一个白色背景，如图 14.7 所示，就能看到填充与文本的相对位置。当没有使用所有定义的填充的空间时，包围的元素必须作出调整。你将在第 15 章中学习到这种效果的细节。

图 14.7

显示填充与文本的相对位置



在使用 CSS 进行 Web 设计时做的最多的“调整”或者“移动”就是处理边距和填充。只要记住一点：边距在元素之外，填充在元素之内。

## 14.3 保持对齐

网页中的内容并不总是填满它所在的矩形区域，因此，控制内容的对齐通常很有帮助。即使矩形区域内的文本有多行，仍可使用对齐，因为你可能希望文本左、右或居中对齐。有两个样式属性可用于控制网页内容的对齐：`text-align` 和 `vertical-align`。

在第 11 章中，你见过这些样式属性的示例——对齐图像，但是并不影响在这里再次提起这些属性，因为对齐对于整个页面的设计也很重要。

复习一下，`text-align` 样式使元素在其定界区域内水平对齐，其取值可以是 `left`、`right`、`center` 或 `justify`。

`vertical-align` 属性与 `text-align` 属性类似，只是用于使元素在垂直对齐。`vertical-align` 属性指定元素如何与其父元素（在有些情况下，是与网页中元素的当前行）对齐。“当前行”指的是显示在元素内的元素所处的垂直位置，换句话说，就是内联元素。如果几个内联元素显示在同一行，可将其垂直对齐方式设置为相同，使它们垂直对齐。一个这样的例子是一行并排显示的图像——使用 `vertical-align` 属性可让它们垂直对齐。

下面是 `vertical-align` 属性的可能取值。

- `top`: 将元素的顶部与当前行对齐。
- `middle`: 将元素的中心与当前行对齐。
- `bottom`: 将元素的底部与当前行对齐。
- `text-top`: 将元素的顶部与父元素的顶部对齐。
- `baseline`: 将元素的基线与父元素的基线对齐。
- `text-bottom`: 将元素的底部与父元素的底部对齐。

对齐和边距，填充以及下一小节中学习的 `float` 属性一起使用，使你能够保持对设计的控制。

## 14.4 理解 Float 属性

理解 `float` 属性是理解基于 CSS 的布局和设计的基础，它是这些元素所构成的拼图的最后一块。简而言之，`float` 属性使元素能在设计中到处移动，从而让其他元素围绕它们。你会常常发现 `float` 用于图像（正如你在第 11 章中所看到的），但是你也能和许多设计师一样，在布局中浮动所有类型的元素。

元素的浮动是水平而不是垂直方向的，所以你所需要考虑的只有两种可能值：`right` 和 `left`。使用这两个值时，浮动元素将浮动到容器元素所允许的最右或者最左（根据 `float` 的值）。例如，如果你有 3 个 `<div>` 的浮动值为左，它们将在容器元素（正文）的左边排成一排。如果这些 `<div>` 在另一个 `<div>` 之中，即使那个 `<div>` 浮动到右边，它们也将排列在那个 `<div>` 的左边。

理解浮动最好通过几个示例，所以我们来看看程序清单 14.3。这个程序清单定义 3 个矩

形<div>并且将它们浮动到下一个相邻的元素（浮动到左边）。

### 程序清单 14.3 使用 float 来放置<div>

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Color Blocks</title>
    <style type="text/css">
      body {
        margin:0px;
      }
      div {
        width:250px;
        height:100px;
        border:5px solid #000000;
        color:black;
        font-weight:bold;
        margin:25px;
      }

      div#d1 {
        background-color:red;
        float:left;
      }

      div#d2 {
        background-color:green;
        float:left;
      }

      div#d3 {
        background-color:blue;
        float:left;
      }
    </style>
  </head>

  <body>
    <div id="d1">DIV #1</div>
    <div id="d2">DIV #2</div>
    <div id="d3">DIV #3</div>
  </body>
</html>
```

结果页面如图 14.8 所示，你已经可以看到问题——这 3 个色块应该相邻。它们确实相邻，但是浏览器窗口不足以显示这 3 个 250 个像素宽，互相之间有 25 个像素的边距的色块。因为它们是浮动的，第三个将显示在下一行。

你可以想象，这在设计布局可能是个问题，所以要注意你的边距、填充、对齐和浮动，同时还要在一个目标浏览器窗口大小中测试。诚然，图 14.8 中所示的浏览器窗口较小，使得浮动元素没有足够的空间而显示到下一行。换句话说，如果你在较大的浏览器窗口中打开这个 HTML 文件，可能不会看到这个问题——这就是应该在不同分辨率下检查网站，以了解是否需要修复的原因。这里的“修复”就是调整<div>边距和其他与大小相关的属性。

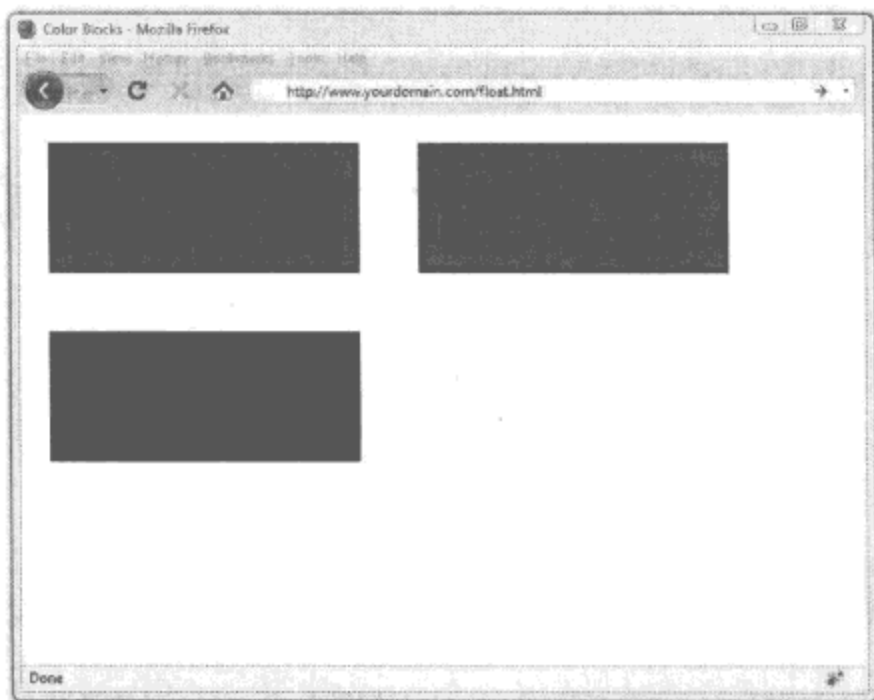


图 14.8

使用浮动来放置色块

图 14.9 所示为 float 属性的另一种有趣的用法。对程序清单 14.3 中的代码的修改只涉及将色块的宽度改为 100 个像素，减少边距为 10 个像素，并且将第 2 个色块的 float 对齐改为右（代替左）。

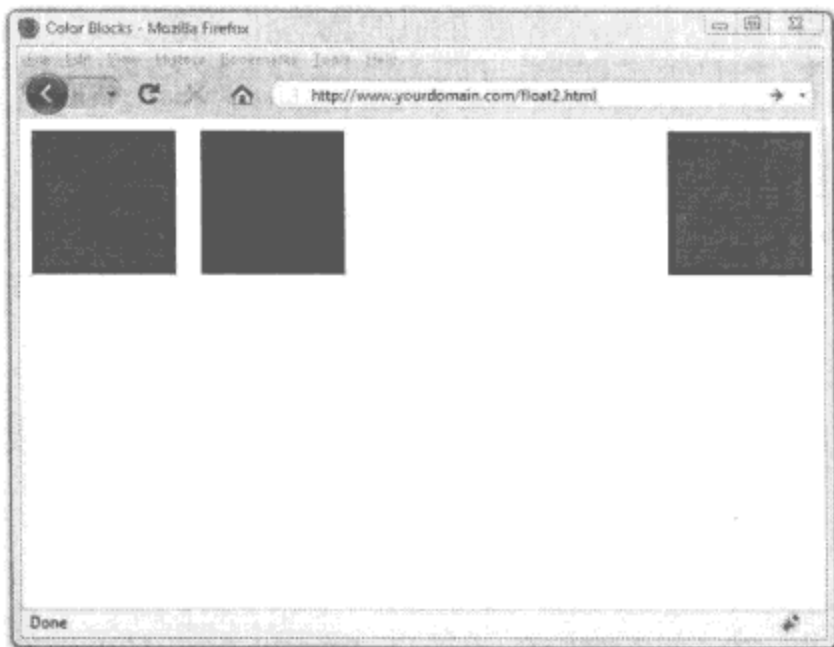


图 14.9

使用浮动来放置色块

但是，发生了一些非常有趣的现象。第二个色块现在变成了第三个色块，它与右边平齐。第二个色块的浮动值为右，所以它完全浮动到右边。第一和第三个色块尽可能浮动到左边，而不管 HTML 中的 <div> 代码：

```
<div id="d1">DIV #1</div>
<div id="d2">DIV #2</div>
<div id="d3">DIV #3</div>
```

浮动需要许多实践才能习惯，特别是页面中有其他元素而不仅仅是色块的时候。例如，如果添加一个基本段落会怎么样？所有浮动元素之后的元素将会环绕这个元素。为了避免这种现象，使用 clear 属性。

clear 属性有 5 个可能值：left、right、both、none 和 inherit。最常用的值为 left、right 和 both。指定 clear:left 将确保没有其他浮动元素向左浮动，clear:right 将确保没有其他浮动元素向右浮动，以此类推。浮动和清除是一个边做边学的过程，所以在本章稍后的作业中寻找更多种情况。

## 14.5 总结

本章介绍了基于 CSS 的设计中一些最基本的样式属性：`margin`、`padding` 和 `float`。你学习了 `margin` 属性对元素外围的空间的控制以及 `padding` 属性对元素内部空间的处理。

在复习了前一章中学习到的 `text-align` 和 `vertical-align` 属性之后，你学习到了 `float` 属性。`float` 属性考虑了元素的具体放置和元素周围的附加内容。

## 14.6 问与答

问：边距和填充示例中处理的都是方框和文本。可以将边距和填充应用到图像吗？

答：是的，可以将边距和填充应用到任何块级元素，比如 `<p>`、`<div>` 和 `<img/>`，以及列表（如 `<ul>` 和 `<ol>`）和列表项目（`<li>`）——这仅是几个例子。

## 14.7 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 14.7.1 测验

1. 为了将两个 `<div>` 元素并排放置，两者之间有 30 个像素的边距，样式单中使用什么条目？
2. 哪个 CSS 样式属性和值用于确保内容不会出现在浮动元素的左边？
3. 哪个 CSS 样式单条目用于在 `<div>` 中距离元素顶部 12 个像素的地方放置文本？

### 14.7.2 测验答案

1. 你可以使用多种方法。第一个 `<div>` 使用样式属性 `margin-right:15px`，第二个 `<div>` 使用 `margin-left:15px`。也可以在合适的元素上使用 30 个像素的 `margin-right` 或者 `margin-left` 属性。
2. 这个例子中使用 `clear:left`。
3. `Padding-top:12px`

## 14.8 练习

- 对边距、填充、对齐和浮动的完全理解需要实践。使用色块代码或者自己的 `<div>` 练习所有间距和浮动的风格和类型之后再进入下一章。下一章将 CSS “盒子模型” 作为整体来讨论，这个模型包含了本章中所讨论的所有单独项目。
- 练习在学习过的所有块级元素上应用边距和填充。熟悉在文本块中放置图像和在图像周围放置边距，使文本不会直接碰到图像的边缘。

## 第 15 章

# 理解 CSS 盒子模型和定位

---

本章中你将学到：

- CSS 盒子模型的概念
- 定位元素的方法
- 控制元素堆叠顺序的方法
- 管理文本流的方法

在前一章中，我几次提到了 CSS 盒子模型——本章从盒子模型的讨论开始，并且说明前一章中学习到的知识对理解这个模型的帮助。通过盒子模型的学习，你不会在创建设计然后发现元素没有对齐或者有些“偏离”之时感到焦躁不安。你将会知道在大部分情况下，只需要稍微调整一下边距、填充和边框等，就可以解决问题。

你还将学到更多关于 CSS 定位的知识，包括以三维的方式（不是垂直的方式）将元素堆叠在一起。最后，你将学习用 float 属性来控制元素周围的文本流。

### 15.1 CSS 盒子模型

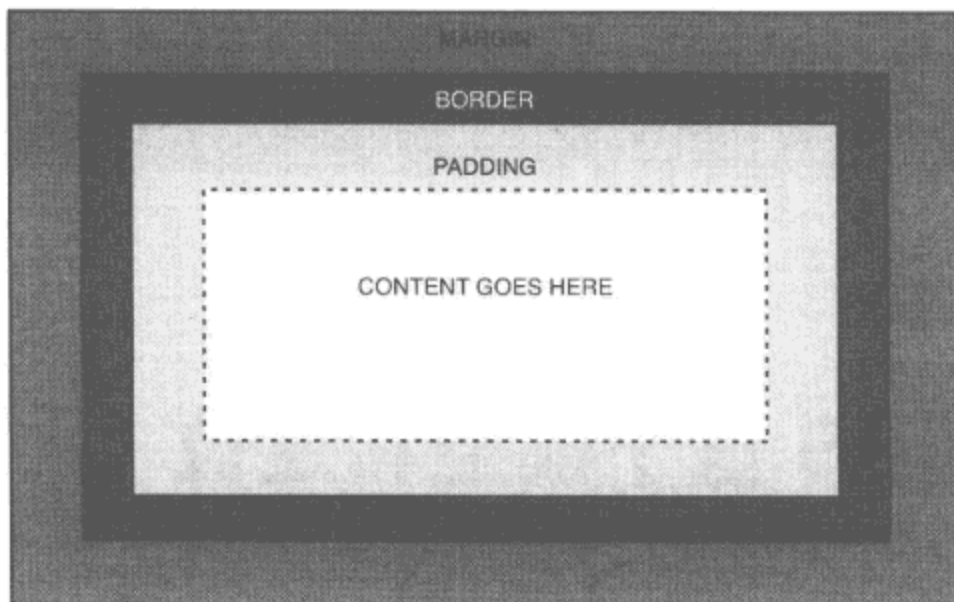
HTML 中的每个元素被考虑为一个“盒子”，不管这个元素是段落、<div>还是图像。盒子有一致的属性，不管我们能不能看到，也不管是否全部在样式单中指定，它们始终存在，作为设计人员，我们必须在创建布局时记得它们的存在。

图 15.1 是盒子模型的图解。盒子模型描述了每个 HTML 块级元素可能的边框、填充和边距，以及边框、填充和边距的应用方式。换句话说，所有元素的内容和边框之间都有一些填充。此外，边框可能可见也可能不可见，但是它的空间是存在的，正如元素边框和外部的其他内容之间有边距一样。



图 15.1

HTML 中的每个元素由 CSS 盒子模型代表



盒子模型还有另一种解释，从外往内如下所示。

- 边距是元素之外的区域。它绝没有颜色，始终是透明的。
- 边框从元素的周围扩展，在填充之外。边框可以有多种类型、宽度和颜色。
- 填充存在于内容外围并且继承内容区域的背景颜色。
- 内容被填充包围。

这里有个难点：要知道元素真正的高度和宽度，必须考虑盒子模型的所有要素。如果你记得前一章中的示例，尽管明确指定<div>宽度为 250 个像素，高度为 100 个像素，但是<div>应该扩展才能容纳在用的填充。

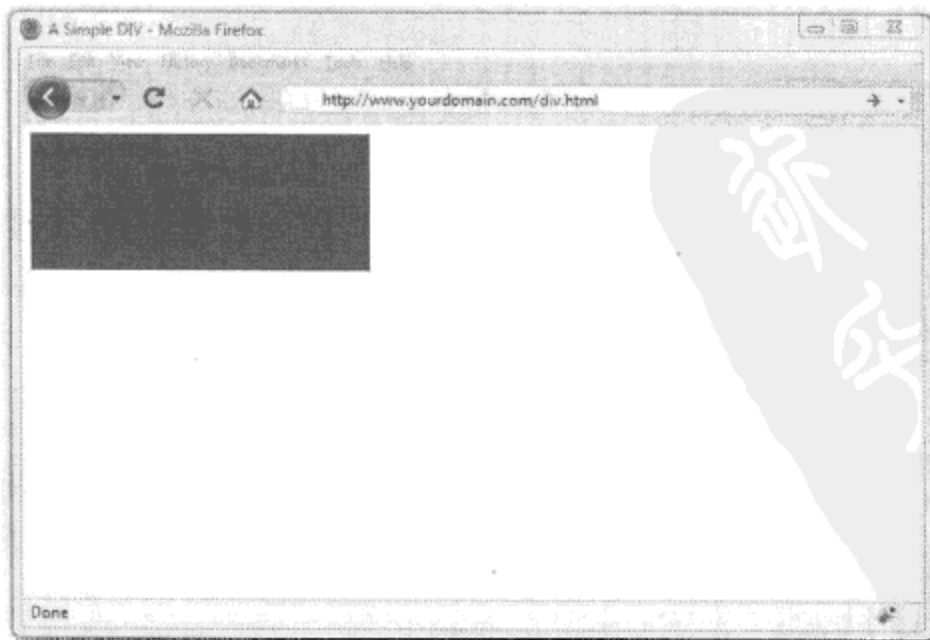
你已经知道使用属性来设置元素的宽度和高度。下面的示例说明如何定义 250 个像素宽、100 个像素高、红色背景、黑色的单像素边框的<div>：

```
div {  
  width: 250px;  
  height: 100px;  
  background-color: #ff0000;  
  border: 1px solid #000000;  
}
```

这个简单的<div>效果如图 15.2 所示。

图 15.2

这是个简单的  
<div>



如果我们用相同的属性定义第二个元素，但是添加特定大小的 margin 和 padding 属性，

就能看到元素尺寸的变化。这是因为盒子模型。

第二个<div>将定义如下，只添加了 10 个像素的边距和 10 个像素的填充：

```
div#d2 {  
  width: 250px;  
  height: 100px;  
  background-color: #ff0000;  
  border: 5px solid #000000;  
  margin: 10px;  
  padding: 10px;  
}
```

第二个<div>效果如图 15.3 所示，定义为和第一个等高等宽，但是包围元素的整个盒子的总体高度和宽度在设置了边距和填充之后变大了很多。

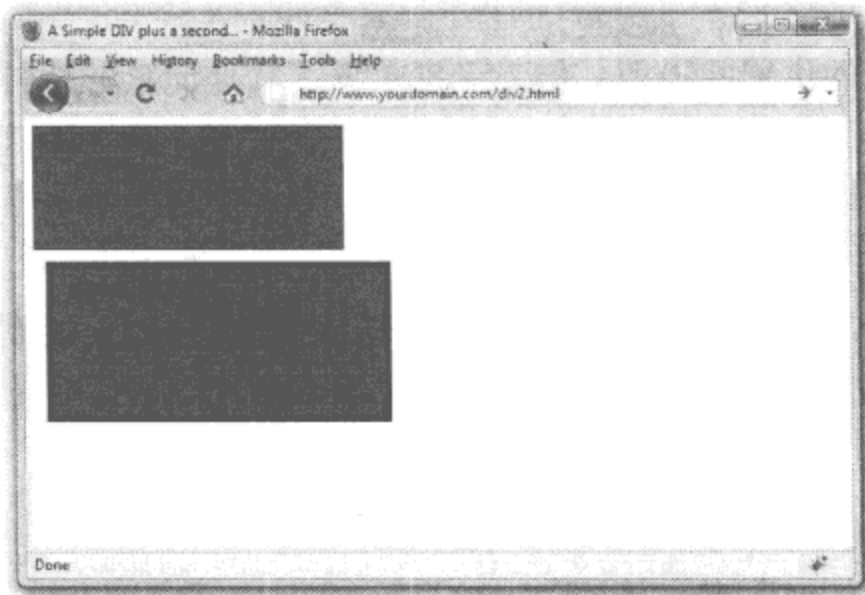


图 15.3

这是另一个简单的<div>但是盒子模型影响了第二个<div>的大小

总宽度如下：

宽度+左填充+右填充+左边框+右边框+左边距+右边距

总高度如下：

高度+上填充+下填充+上边框+下边框+上边距+下边距

因此，第二个<div>实际宽度为 300（250+10+10+5+5+10+10），实际高度为 150（100+10+10+5+5+10+10）。

现在你开始了解了盒子模型对设计的影响。我们假设你仅有 250 个像素的水平空间但是希望每边有 10 个像素的边距、10 个像素的填充和 5 个像素的边框。为了使希望显示的这些具有空间，必须指定<div>为仅有 200 个像素宽，这样  $200+10+10+5+5+10+10=250$ （可用的水平空间）。

现在你已经以 CSS 盒子模型的方式进行了训练，在本书余下的工作以及 Web 设计时记住它。此外，它将影响元素定位和内容的流向，这是接下来我们要处理的两个课题。

**注意：**本书中你被反复灌输 DOCTYPE 声明的使用——所有样板代码都包含 DOCTYPE。继续这种方法不仅使你的代码通过验证，而且是因为 Internet Explorer 在 CSS 盒子模型上的特殊问题：如果不定义 DOCTYPE，Internet Explorer 将以你意料之外的方式操纵元素的宽度和高度。这导致浏览器与你的布局不兼容，所以要记得包含 DOCTYPE。

By the  
Way

## 15.2 定位详解

相对定位是 HTML 使用的默认定位类型。可以将相对定位理解为棋盘上棋子的排列：棋子从左到右排列，达到边缘时移到下一行。样式属性 `display` 的值为 `block` 的元素自动放在下一行，而为 `inline` 的元素自动放在同一行，位于前一个元素后面。举个例子，`<p>`和`<div>`标签是块元素，而`<span>`标签是内联元素。

CSS 还支持另一种定位，称为绝对定位，因为它可以精确地设置 HTML 内容在网页中的位置。虽然绝对定位让你能够明确指定元素的位置，但位置仍是相对于网页中父元素的。换句话说，绝对位置让你能够指定元素的矩形局域相对于父元素区域的位置，这一点与相对定位截然不同。

你拥有指定元素在网页中的位置的自由，但当元素占据了其他元素的位置时，这就导致元素重叠的问题。你可以指定元素的绝对位置，从而导致元素重叠。在这种情况下，CSS 依靠 `z` 索引 (`z-index`) 来决定哪个元素在上方，哪个元素在下方。你将在本章稍后更详细地学习 `z` 索引。现在，先来看看如何控制样式规则使用相对定位还是绝对定位。

样式规则使用的定位类型（相对或绝对）是由 `position` 属性决定的，该属性有两个可能的值：`relative` 和 `absolute`。指定定位类型后，可用下列属性来提供具体的位置。

- `left`: 左边的位置偏移。
- `right`: 右边的位置偏移。
- `top`: 顶部的位置偏移。
- `bottom`: 底部的位置偏移。

你可能认为这些位置属性只适用于绝对位置，但实际上它们对两种定位都适用。在相对定位下，元素的位置用相对于元素原始位置的偏移来指定。因此，如果将 `left` 属性设置为 `25px`（像素），元素的左边将从原来的（相对）位置向右移 25 像素。而对于绝对定位，应用样式的元素位置是相对于父元素指定的。因此，如果在绝对定位下将 `left` 属性设置为 `25px`，元素的左边将距离父元素的左边 25 像素。而如果将 `right` 属性设置为同样的值，元素的右边将距离父元素的右边 25 像素。

让我们回到色块示例来说明定位的工作方式。在程序清单 15.1 中，4 个色块指定了相对定位。如图 15.4 所示，这些方块被垂直放置。

### 程序清单 15.1 用 4 个色块说明相对定位

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Positioning the Color Blocks</title>
    <style type="text/css">
      div {
        position:relative;
        width:250px;
```

```
    height:100px;
    border:5px solid #000;
    color:black;
    font-weight:bold;
    text-align:center;
}
div#d1 {
    background-color:red;
}

div#d2 {
    background-color:green;
}

div#d3 {
    background-color:blue;
}
div#d4 {
    background-color:yellow;
}
</style>

</head>
<body>
    <div id="d1">DIV #1</div>
    <div id="d2">DIV #2</div>
    <div id="d3">DIV #3</div>
    <div id="d4">DIV #4</div>
</body>
</html>
```



图 15.4

色块被一个接一个地垂直放置

`<div>`元素的样式单条目设置 `position` 样式属性为 `relative`。因为剩下的样式规则继承自 `<div>` 样式规则，也就继承了相对定位。实际上，其他样式规则的唯一差别是背景颜色不同。

注意，在图 15.4 中，`<div>` 元素一个接一个显示，这是使用相对定位所预期的。但是为了更有趣（这正是这里要做的），你可以将定位改为绝对的并且明确地指定颜色的位置。在程序清单 15.2 中，样式单条目被修改为使用绝对定位来排列色块。

## 程序清单 15.2 使用绝对定位的色块

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Positioning the Color Blocks</title>
    <style type="text/css">
      div {
        position:absolute;
        width:250px;
        height:100px;
        border:5px solid #000;
        color:black;
        font-weight:bold;
        text-align:center;
      }
      div#d1 {
        background-color:red;
        left:0px;
        top:0px;
      }
      div#d2 {
        background-color:green;
        left:75px;
        top:25px;
      }
      div#d3 {
        background-color:blue;
        left:150px;
        top:50px;
      }
      div#d4 {
        background-color:yellow;
        left:225px;
        top:75px;
      }
    </style>
  </head>
  <body>
    <div id="d1">DIV #1</div>
    <div id="d2">DIV #2</div>
    <div id="d3">DIV #3</div>
    <div id="d4">DIV #4</div>
  </body>
</html>
```

这个样式单将 `position` 属性设置为 `absolute`，这对于样式单使用绝对定位来说是必要的。此外，每个继承的 `<div>` 样式规则都设置了 `left` 和 `top` 属性。但是，每个规则的位置被设置为元素互相重叠，如图 15.5 所示。

现在我们谈论的是布局！图 15.5 说明了绝对定位使你能将元素精确地放在需要的地方。它还揭示了使元素互相重叠有多么简单。你可能很好奇，浏览器如何知道哪个元素被绘制在重叠部分的上面。下一小节介绍堆叠顺序的控制。

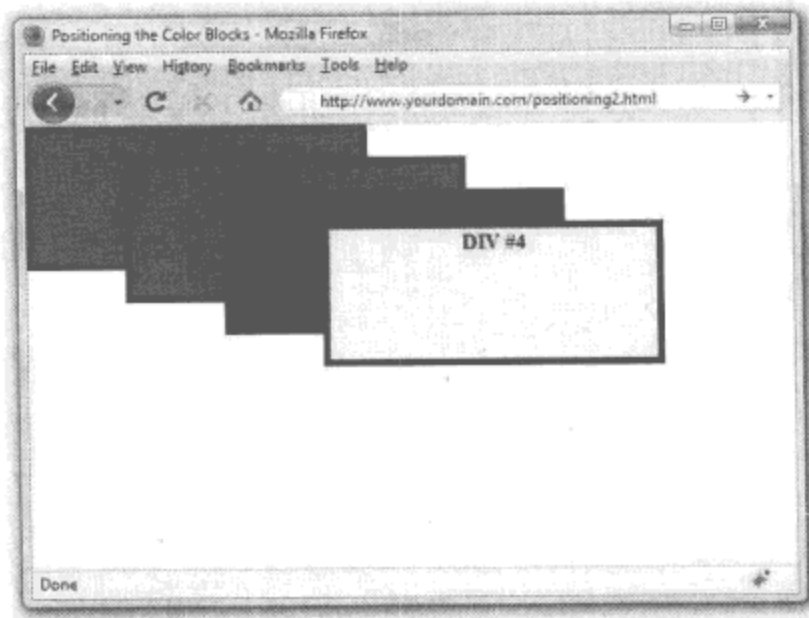


图 15.5

使用绝对定位显示  
的色块

## 15.3 控制元素的堆叠顺序

在有些情况下，肯定需要仔细地控制元素在网页中堆叠顺序。`z-index` 样式属性让你能够设置元素的堆叠顺序。`z-index` 这个名称看上去有些奇怪，它表示除计算机屏幕上水平和垂直的两维外，指向计算机屏幕的第三维。也可以将 `z-index` 看作是一本杂志在一摞杂志中的相对位置。靠近顶部的杂志的 `z-index` 比下面的杂志大，同样，堆叠元素时，`z-index` 值较大的元素在 `z-index` 值较小的元素上面。

`z-index` 属性用于设置一个数值，指示样式规则的相对 `z-index`。`z-index` 值仅在与样式表中的其他样式规则比较时才有意义，即设置单个样式规则的 `z-index` 属性没有意义。另一方面，如果为应用于重叠元素的多个样式规则设置 `z-index`，则 `z-index` 较大的元素将显示在 `z-index` 较小的元素上面。

**注意：**不管样式规则设置的 `z-index` 是多少，应用该样式规则的元素都将显示在父元素上面。

**By the  
Way**

### 程序清单 15.3 使用 `z-index` 改变色块样板中元素的显示

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Positioning the Color Blocks</title>
    <style type="text/css">
      div {
        position:absolute;
        width:250px;
        height:100px;
        border:5px solid #000;
        color:black;
        font-weight:bold;
        text-align:center;
      }
    </style>
  </head>
  <body>
    <div style="position:absolute; top:50px; left:50px; width:250px; height:100px; background-color:black; color:white; font-size:24px; text-align:center; z-index:1000;">
      DIV #4
    </div>
  </body>
</html>
```



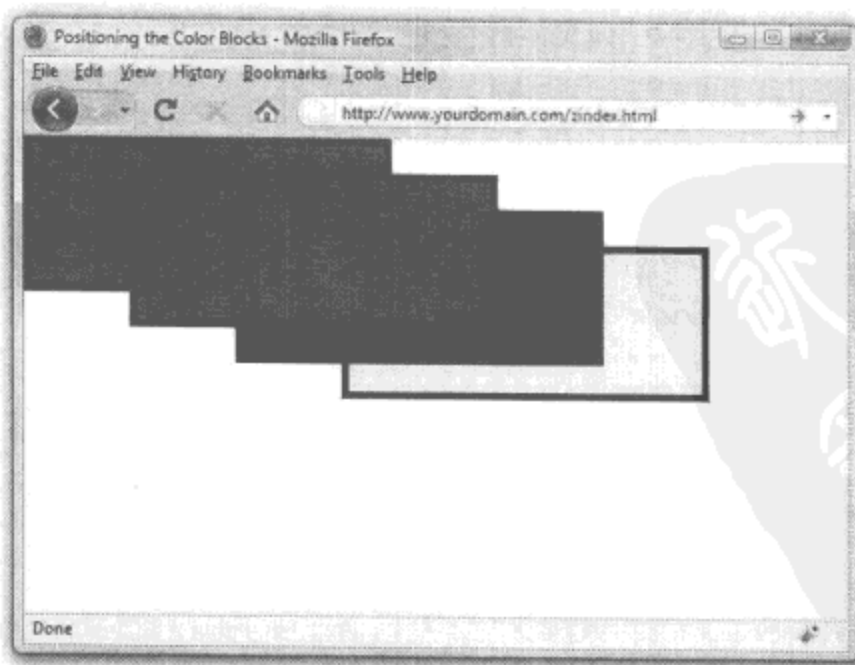


```
div#d1 {
  background-color:red;
  left:0px;
  top:0px;
  z-index:0;
}
div#d2 {
  background-color:green;
  left:75px;
  top:25px;
  z-index:3;
}
div#d3 {
  background-color:blue;
  left:150px;
  top:50px;
  z-index:2;
}
div#d4 {
  background-color:yellow;
  left:225px;
  top:75px;
  z-index:1;
}
</style>
</head>
<body>
  <div id="d1">DIV #1</div>
  <div id="d2">DIV #2</div>
  <div id="d3">DIV #3</div>
  <div id="d4">DIV #4</div>
</body>
</html>
```

你在程序清单 15.3 中看到的唯一变化是在每个编号的 div 样式类中添加了 z-index 属性。注意，第一个编号 div 中 z-index 设置为 0，这使其成为 z-index 值最低的元素，而第二个 div 的 z-index 最高。图 15.6 所示为这个样式单所显示的色块页面，它清晰地展示了 z-index 对显示内容的影响，并且使精细地控制元素的覆盖成为可能。

图 15.6

使用 z-index 改变  
色块显示



尽管显示色块的示例都是简单的<div>元素，但是 z-index 属性可以影响任何 HTML 内容，包括图像。

## 15.4 管理文本流

现在你已经看到了将元素相对其他元素放置或者绝对放置的一些示例，可以重新探讨元素周围的内容流了。“当前行”的概念指的是用于放置网页元素的不可见行。这一行与网页中的元素流有关，它在元素在网页中依次排列时起作用。元素流的一部分是文本流。当文本与其他元素（如图像）混合时，控制文本如何沿其他元素绕排很重要。

你已经在第 14 章中看到两个这种样式属性，下面是一些提供文本流控制的样式属性。

- **float**: 决定文本如何沿元素绕排。
- **clear**: 禁止文本沿元素绕排。
- **overflow**: 在元素太小而不能包含所有文本时控制文本的溢出方式。

**float** 属性用于控制文本如何沿元素绕排。它可设置为 **left** 或 **right**。这两个值决定元素相对于流动文本应如何放置。因此，如果将图像的 **float** 属性设置为 **left**，图像将放在文本的左边。

正如上一章中所学习到的，使用 **clear** 属性可防止文本出现在元素旁边，**clear** 属性可设置为 **none**、**left**、**right** 或 **both**。**clear** 属性的默认值是 **none**，表示排列文本时不考虑元素；**left** 导致文本不出现在元素左边，除非网页左边没有元素；同样，**right** 值表示文本出现在元素右边；**both** 值表示文本出现在元素两边。

**overflow** 属性处理溢出文本，溢出文本指的是在元素的矩形区域内放不下的文本，如果将元素的 **width** 和 **height** 设置得太小，就可能发生这种问题。**overflow** 属性可设置为 **visible**、**hidden** 或 **scroll**。**visible** 自动扩大元素使矩形区域可容纳溢出文本，这是 **overflow** 属性的默认值。**hidden** 保持元素的大小不变，使溢出文本不可见；**scroll** 可能是最有趣的值，它使元素添加滚动条，以便可以通过滚动看到所有文本。

## 15.5 总结

本章从关于 CSS 盒子模型以及考虑边距、填充和边框时元素高度和宽度的计算方法开始，这一讨论非常重要。接着处理元素的绝对定位，然后你学习了使用 **z-index** 的定位。最后学习了一些使你能够控制页面上的文本流的样式属性。

## 15.6 问与答

问：如何选择相对定位和绝对定位？

答：关于相对定位与绝对定位的选择，虽然没有固定的指导方针，但总体思路是，绝对定位只用于需要精确控制内容的位置时。因为绝对定位在定位内容时让你能够精确到像素，而相对定位在内容如何定位上的不可预测性更大。这不并不是说相对定位不能将网页元素定位好，只是说绝对定位更精确。当然，这也使得绝对定位在屏幕大小改变时更易受影响，而这是你无法控制的。

问：如果两个相互重叠的元素没有指定 **z-index**，如何知道哪个元素将显示在方面？

答：如果相互重叠的元素没有指定 `z-index`，在网页中较后出现的元素将显示在上面。这一点很容易记住，只要想象一下浏览器从HTML文档逐一读取元素并绘制它们，后面读取的元素将绘制在前面读取的元素上面。

## 15.7 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 15.7.1 测验

1. 绝对定位与相对定位的区别是什么？
2. 使用什么 CSS 样式属性来控制元素的堆叠顺序？
3. 编写 HTML 代码，使句子“Where would you like to”从在浏览器窗口的左上角开始显示，使“GO TODAY?”以大字体显示在窗口左上角下方 80 像素和右边 20 像素处。

### 15.7.2 测验答案

1. 在相对定位中，内容根据网页的走向显示，在 HTML 代码中，每个元素显示在它前面的元素后面，而绝对定位让你能够设置网页内容的精确位置。
2. 使用 `z-index` 控制元素的重叠顺序。
- 3.

```
<span style="position:absolute;left:0px;top:0px">  
Where would you like to</span>  
<h1 style="position:absolute;left:80px;top:20px">GO TODAY?</h1>
```

## 15.8 练习

- 创建不同边距，填充和边框的一系列元素，并且查看这些属性对高度和宽度的影响，以此来实践复杂的 CSS 盒子模型。
- 寻找喜欢的一组图像并且使用绝对定位甚至一些 `z-index` 值来将它们排列成画廊形式。尝试将这些图像排列成一个图案（如矩形、三角形或者圆形）。

## 第 16 章

# 使用 CSS 进一步控制列表

---

本章中你将学到：

- CSS 盒子模型对列表的影响
- 定制列表项目标志的方法
- 使用列表项目创建图像映射的方法

在第 5 章中，为你介绍了 3 种 HTML 列表，在第 14 章中学习了元素的边距、填充和对齐。本章中，你将学习将边距、填充和对齐应用到不同类型的 HTML 列表的方法，帮助你用纯粹的 HTML 和 CSS 生成一些强有力的设计元素。

具体地说，你将学习如何修改列表项目的外观——用第 5 章学过的 `list-style-type` 属性之外的方法——以及使用 CSS 样式列表来替换第 11 章中学习到的客户端图像映射。你将实践许多到目前为止学过的 CSS 样式，本章中得到的知识将直接地进入到第 17 章中的项目。

### 16.1 复习 HTML 列表

在第 5 章中学过，有 3 种基本的 HTML 列表。根据类型和上下文，每种列表内容的表现略有不同：

- 有序列表是在每个列表项目前有数字或者字母的缩进列表。有序列表以 `<ol>` 标签开始，`</ol>` 标签结束。列表项包含在 `<li></li>` 标签对中。这种列表通常用于显示编号的步骤或者内容的级别。
- 无序列表是在每个列表项目前有一个项目符号或者其他符号的缩进列表。无序列表以 `<ul>` 标签开始，`</ul>` 标签结束。这种列表通常用于提供后续信息的简短而具体的视觉提示。

- 定义列表通常用于显示术语及其含义，从而提供列表上下文中的信息层次——很类似有序列表但是没有编号。定义列表以<dl>开始，</dl>结束。<dt></dt>标签对包含每个项目，<dd></dd>标签对包含每个定义。

在内容要求下，可以嵌套有序和无序列表——就是将列表放在别的列表中。嵌套列表产生一个内容层次，所以保留在内容确实有你希望显示的层次结构（如内容大纲或者目录）时使用。你将在第17章中学到，可以在网站导航包含子导航元素时使用嵌套列表。

## 16.2 CSS 盒子模型对列表的影响

特有的列表相关样式包括 `list-style-image`（放置图像作为列表项标志）、`list-style-position`（表示列表项目标志的位置）和 `list-style-type`（列表项目标志的类型）。但是在这些样式控制列表结构和列表项目的同时，你可以使用 `margin`、`padding`、`color` 和 `background-color` 样式来完成更加特殊的列表显示。

### By the Way

**注意：**有些老的浏览器以不同方式处理边距和填充，特别是列表和列表项目周围。但是，在编写本书的时候，本章和其他章节中的HTML和CSS在主要Web浏览器（Apple Safari、Google Chrome、Microsoft Internet Explorer、Mozilla Firefox和Opera）的当前版本中的显示相同。当然，你仍然应该在将Web内容发布到网上之前在所有浏览器中审核，但是容纳浏览器的怪癖的“Hacking”样式单的需求正在逐渐消失。

在第14章中，你学习到每个元素在内容与其边框之间有一些填充，你还学习到了在元素边框和其他内容之间有边距。对于列表也是如此，当你为列表添加样式时，必须记住“列表”实际上由两个元素组成：上级列表元素类型（<ul>或者<ol>）和单独的列表项目。每个元素都有可由样式单影响的边距和填充。

本章中的示例为你展示不同CSS样式对HTML列表和列表项目显示的影响。记住这些基本的不同，你将能够完全控制列表，正如你在第17章将要学到的，你还将能够使用列表来完成网站导航中的高级视觉效果。

程序清单16.1创建一个基本的列表，包含3个项目。在这个程序清单中，无序列表本身（<ul>）有一个蓝色的背景、黑色边框，宽度为100个像素，如图16.1所示。列表项目（单独的<li>）有灰色的背景和黄色的边框，列表项目文本和标志（项目符号）为黑色。

### 程序清单 16.1 创建具有颜色和边框样式的基本列表

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>List Test</title>
    <style type="text/css">
      ul {
        background-color: #6666ff;
        border: 1px solid #000000;
```

```
        width:100px;
    }
    li {
        background-color: #cccccc;
        border: 1px solid #ffff00;
    }
</style>
</head>

<body>
  <h1>List Test</h1>
  <ul>
    <li>Item #1</li>
    <li>Item #2</li>
    <li>Item #3</li>
  </ul>
</body>
</html>
```

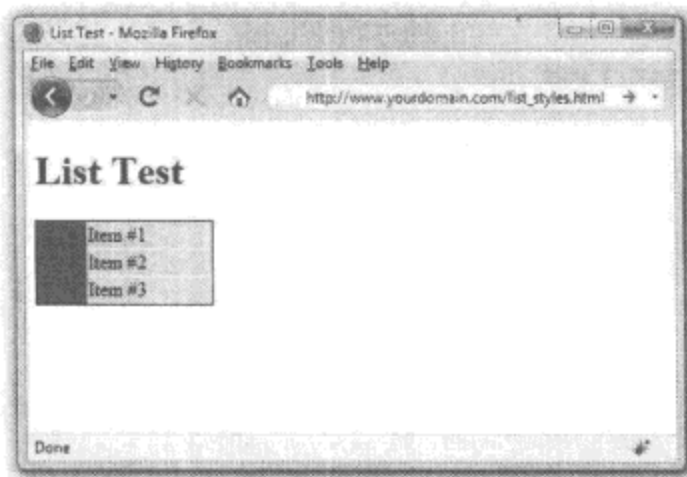


图 16.1

为列表和列表项目  
设置颜色和边框

如图 16.1 所示，`<ul>` 创建一个单独列表项目所在的盒子。在这个示例中，整个盒子都有蓝色的背景。但是要注意单独的列表项目——在这个示例中，它们使用一个灰色背景和黄色边框——没有扩展到 `<ul>` 创建的盒子的左边缘。

这是因为浏览器自动地在 `<ul>` 的左边添加一定的填充。浏览器没有将填充加到边距，因为边距出现在盒子的外部，它们仅仅添加盒子中左边的填充，填充值大约 40 个像素。

**注意：**你可以创建一个简单的测试文件（见程序清单 16.1），然后在样式单中的 `ul` 声明中添加 `padding-left: 40px;` 来测试不同浏览器默认的左填充值。如果你重新装入页面并且显示没有变化，那么就可以知道测试浏览器使用 40 个像素的默认左填充值。

**By the  
Way**

不管什么列表类型，默认的左边填充值保持不变。如果你在样式单中添加如下代码行，将发现填充值保持不变，如图 16.2 所示：

```
list-style-type:none;
```

当你创建一个包含列表的页面布局时，尝试用填充来将项目整齐地放在页面上。同样，列表没有默认的边距并不意味着不能设置它，在 `ul` 选择符的声明中添加 `margin` 值将提供额外的布局控制。

但是要记住，目前我们仅仅使用了列表定义本身，我们还没有将样式应用到单独的列表项目。在图 16.1 和图 16.2 中，列表项目的灰色背景和黄色边框说明没有默认的填充或者边距。



图 16.3 所示为将填充和边距值应用到列表项目而不是列表“盒子”的不同效果。

图 16.2

不管有没有列表项目标记，默认左填充值保持不变

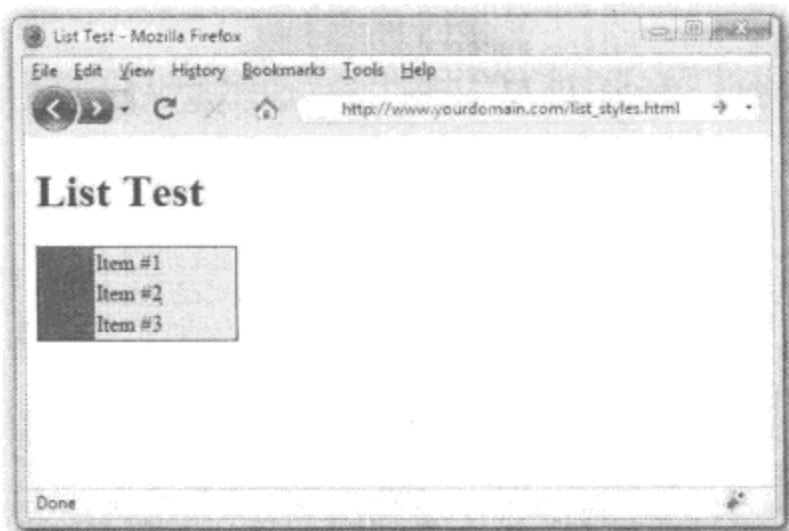
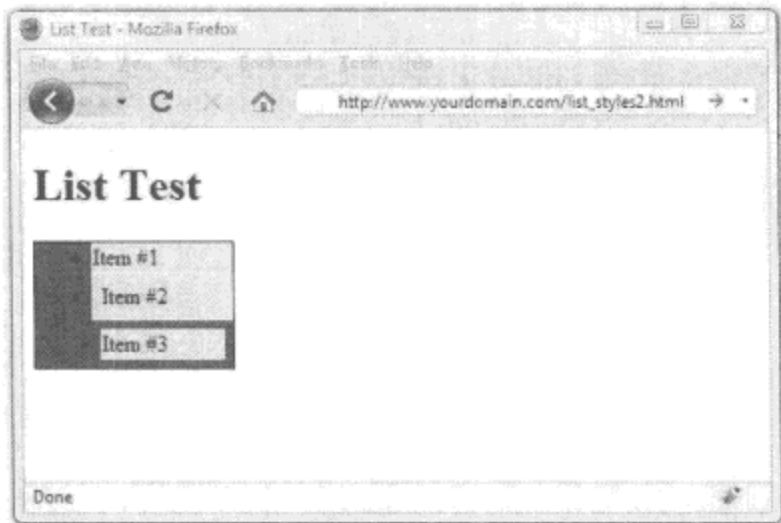


图 16.3

不同的值影响列表项目的填充和边距



第一个列表项目是没有应用填充或者边距的基本项目。但是，第二个列表项目使用 `style="padding: 6px; "`，你可以看到四边都有 6 个像素的填充（在内容和包围元素的黄色边框之间）。注意，项目符号的位置仍然和第一个项目一样。第三个列表项目使用 `style="margin: 6px; "` 在列表项目周围应用 6 个像素的边距，这个边距使 `<ul>` 的蓝色背景显露出来。

### 16.3 放置列表项目标志

对边距和填充的讨论提出了另一个问题：列表项目标志（如果使用）的控制以及文本的卷绕（或者不卷绕）。`list-style-position` 属性的默认值是“outside”——这个位置意味着项目符号、编号或者其他标志在文本的左边，`<li></li>` 标签对创建的盒子之外。当文本在列表项目中卷绕时，它在盒子中卷绕并且保持与元素的左边框平齐。

但是，当 `list-style-position` 属性的值为“inside”时，标志在 `<li></li>` 标签对创建的盒子之内。不仅列表项目标志被进一步缩进（它们实际上成为文本的一部分），而且文本卷绕到每个项目标志之下。

`list-style-position` 值为外部和内部的一个示例如图 16.4 所示。产生该图显示的代码与程序清单 16.1 相比的唯一变化（不包括“Item#2”和“Item#3”后添加的填充文本）是第二个列表项目包含 `style="list-style-position: outside"`，第三个列表项目包含 `style="list-style-position: inside"`。

第二个列表项目中的填充文本显示了列表宽度被定义为不足于一行中显示所有文本时，文本卷绕的方式。没有使用 `style="list-style-position: outside"` 也将得到相同的结果，因为

这是代码中没有明确指明时的默认值。

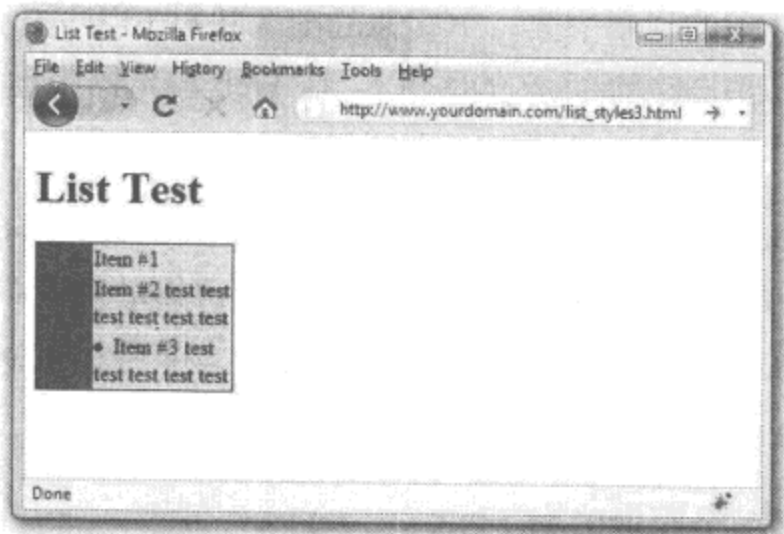


图 16.4

`list-style-position` 值为 `outside` 和 `inside` 之间的不同

但是，你可以清楚地看到使用“`inside`”位置时的不同。在第三个列表项目中，项目符号和文本都在黄色边框的灰色区域（列表项目本身）中。当 `list-style-position` 值为 `inside` 时，边框和填充对列表项目有不同的影响，如图 16.5 所示。

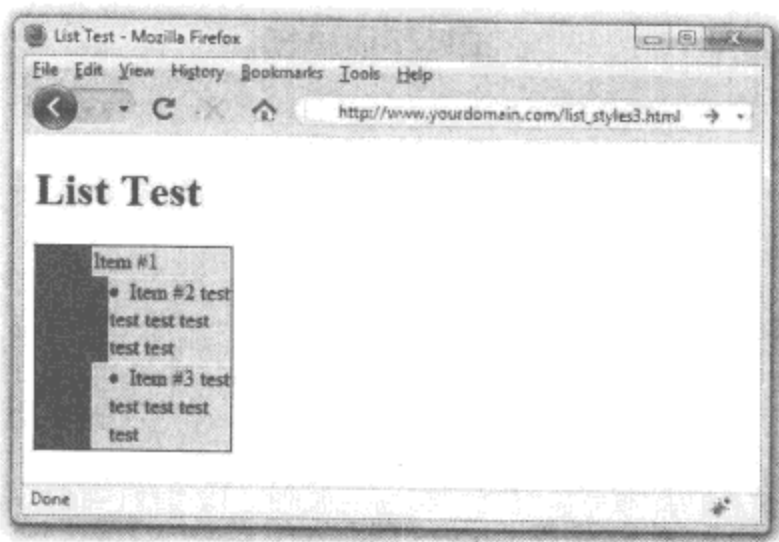


图 16.5

边距和填充改变 `list-style-position` 值为 `inside` 的列表项目

在图 16.5 中，第二个和第三个列表项目的 `list-style-position` 值为 `inside`。但是，第二个列表项目的 `margin-left` 值为 12 个像素，第三个列表项目的 `padding-left` 值为 12 个像素。两个内容块（列表标记加上文本）都显示文本围绕项目符号卷绕，定义列表项目的灰色区域中这些块的位置相同，受影响的区域是列表之中的列表项目。

不出意料，`margin-left` 值为 12 个像素的列表项目显示，12 个像素的红色从列表项周围的透明边距中显露出来。同样，`padding-left` 值为 12 个像素在内容开始之前显示 12 个像素的灰色背景（列表项目的）。填充在元素内部，边距在元素外部。

通过理解边距和填充对列表项目和所在列表的影响，可以创建网页中纯 CSS 的导航元素，不需要依赖外部的图像。在第 17 章中，你将学习创建垂直和水平导航菜单以及下拉菜单的方法。

## 16.4 用列表项目和 CSS 创建图像映射

在第 11 章中，你学习了使用 HTML 中的 `<map>` 标签创建客户端图像映射的方法。图像映射使你能定义图像的一个区域并且分配一个链接给这个区域（而不是必需将图像分块，为单块应用链接，并且将图像用 HTML 重新拼凑起来）。但是，你也可以纯粹用有效的 XHTML

和 CSS 创建一个图像映射。

如图 16.6 所示，程序清单 16.2 中的代码生成一个相似的图像映射（程序清单 16.2 中的代码不生成图中的红色边框。这个边框是用来突出定义的区域）。在 Web 浏览器中运行这段代码，看上去像一个放置了图像的网页。当你的鼠标悬停在一个“热区”上时，发生了一些动作。

### By the Way

**注意：**访问 <http://designreviver.com/tutorials/css-image-map-techniques-and-tutorials/>可以得到指向许多创建 XHTML 和 CSS 图像的教程的链接。这些教程的交互程度各有不同，一些教程可能介绍本书范围之外的客户端代码。但是解释非常全面。

### 程序清单 16.2 用 CSS 创建一个图像映射

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>CSS Image Map Example</title>
    <style type="text/css">
      #theImg {
        width:500px;
        height:375px;
        background:url(tea_shipment.jpg) no-repeat;
        position:relative;
        border: 1px solid #000000;
      }
      #theImg ul {
        margin:0px;
        padding:0px;
        list-style:none;
      }
      #theImg a {
        position:absolute;
        text-indent: -1000em;
      }
      #theImg a:hover {
        border: 1px solid #ffffff;
      }
      #ss a {
        top:0px;
        left:5px;
        width:80px;
        height:225px;
      }
      #gn a {
        top:226px;
        left:15px;
        width:70px;
        height:110px;
      }
      #ib a {
        top:225px;
        left:85px;
        width:60px;
        height:90px;
      }
    </style>
  </head>
  <body>
    <img alt="A tea shipment image with three hotspots labeled 'ss', 'gn', and 'ib'." data-bbox="131 300 616 901"/>
  </body>
</html>
```



```
#iTEA1 a {
  top:100px;
  left:320px;
  width:178px;
  height:125px;
}
#iTEA2 a {
  top:225px;
  left:375px;
  width:123px;
  height:115px;
}
</style>
</head>
<body>
  <div id="theImg">
    <ul>
      <li id="ss"><a href="[some URL]"
        title="Sugarshots">Sugarshots</a></li>
      <li id="gn"><a href="[some URL]"
        title="Golden Needle">Golden Needle</a></li>
      <li id="ib"><a href="[some URL]"
        title="Irish Breakfast">Irish Breakfast</a></li>
      <li id="iTEA1"><a href="[some URL]"
        title="IngenuiTEA">IngenuiTEA</a></li>
      <li id="iTEA2"><a href="[some URL]"
        title="IngenuiTEA">IngenuiTEA</a></li>
    </ul>
  </div>
</body>
</html>
```

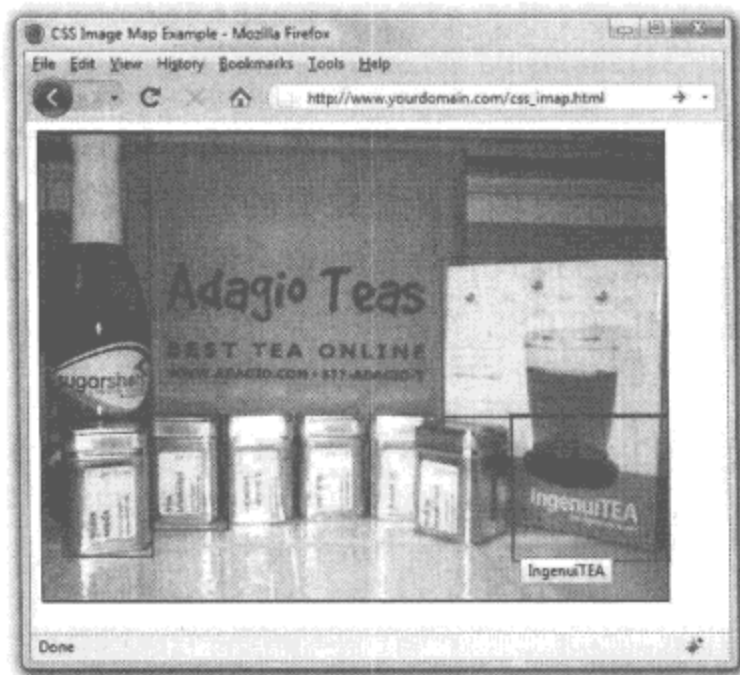


图 16.6

CSS 使你能在图像映射中定义热区

在程序清单 16.2 中可以看到，样式单有许多条目但是实际的 HTML 相当短。列表项目被用于创建 5 个不同的可点击区域，这些“区域”是给出明确的高度和宽度并且位于背景图像之上的列表区域。如果图像被从包含列表的<div>的背景中删除，列表项目仍然存在并且可以点击。

让我们仔细看看样式单，以便理解组成这个 XHTML 和 CSS 图像映射的各部分，从最基本的级别上说，这些部分只是链接的一个列表。

链接列表包含在名为“theimg”的<div>中。在样式单中，这个<div>被定义为黑色的，宽度为500个像素，高度为375个像素，具有1个像素的实线黑色边框的元素。这个元素的背景是名为tea\_shipment.jpg的图像，这个图像被放在一个位置并且不重复。接下来的HTML是无序列表(<ul>)的开始。在样式单中，这个无序列表的边距和填充被设置为0，list-style被设置为none——列表项目前面没有任何图标。

列表项目文本本身不会出现，因为<div>中的所有<a>标签使用了下面这种样式单技巧：

```
text-indent: -1000em;
```

将文本缩进-1000em，你可以确定这些文本不会出现。它确实存在，但是在浏览器窗口左边之外1000em的不可见区域中。换句话说，如果你将左手放在显示器的边上，text-indent:-1000em 将把文本放在你的小指的左边。但是这正是我们所要的，因为我们不需要这些文本链接。我们只需要一个区域被定义为链接，这样用户的光标将在位于网站的链接之上时变化。

当用户的鼠标悬停于包含链接的列表项之上时，列表项目显示一个白色的单像素实线边框，这归功于以下样式单条目：

```
#theImg a:hover {  
    border: 1px solid #ffffff;  
}
```

然后，列表项目被定义并放置，位置根据作为可点击区域的图像区域而定。例如，id为“ss”的列表项目，用于图中名为“Sugarshots”——左上角位于<div>的顶部左起5个像素的地方。这个列表项目宽度为80个像素，高度为225个像素。同样为“#gn”、“#ib”、“#iTEA1”和“#iTEA2”列表项目建立样式声明，这样链接区域与出现在图像相关位置的id关联。

## 16.5 总结

本章从列表和列表项目受到填充和边距样式影响的示例开始。你首先学习了与列表相关的默认填充和控制填充的方法。接下来你学习填充和边距值的修改和将列表项目标志放在项目内部和外部的的方法，由此你可以开始思考样式和列表对整个网站设计的影响。最后，你学习了利用列表和列表项目创建纯XHTML和CSS的图像映射的方法，从而减少了链接图像分块或者<map/>标签的使用。

本章的所有示例目的在于使你“跳出（列表的）固定思维”，如果你能够这么思考，下一章中就可以接受使用无序列表来产生网站中的垂直或者水平导航。

## 16.6 问与答

**问：**有很多网页讨论关于边距和填充的“盒子模型hack”，特别是在列表和列表项目周围。你确定我不需要使用hack？

**答：**在本章的开始，你知道本章（和其他章节）中的HTML和CSS在主要Web浏览器的当前版本中外观相同。在现代的浏览器开始根据CSS规范而不是自己的癖好来处理事情之前，多年来Web开发人员必须使用代码hack和其他技巧。此外，越来越多的时候，可以摆脱需要大部分hack的非常旧的浏览器的用户。所以，我不一定建议你仅为主要浏览器的当前

版本进行设计，也不建议你花费很多时间来实现旧版本浏览器的 hack——这些浏览器的用户低于互联网人口的 5%。你应该继续编写有效和坚持设计准则的代码，在最能反映你的目标用户的一组浏览器中测试网页，并且向世界发布你的网站。

问：CSS 图像映射看上去很费事。<map/>标签不好吗？

答：<map/>标签并不是完全不好，在 XHTML 和 HTML5 中也都有效。但是，确定在客户端图像映射中的坐标可能很困难，特别是在没有图像软件或者创建客户端图像映射专用的软件时。CSS 版本给你定义和现实可点击区域的更多选择，这里你看到的只是其中之一。

## 16.7 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 16.7.1 测验

1. list-style-position 值为 outside 和 inside 之间有何不同？默认值是什么？
2. list-style 值为 “none” 是否仍然产生结构化的有序或者无序列表？
3. 使用 HTML 代码创建一个 350 个像素宽、100 个像素高、绿色背景、有两个像素的黑色虚线边框并将列表项目标志放在容器之内的列表项目。

### 16.7.2 测验答案

1. list-style-position 值为 “inside” 将列表项目标志放在列表项目创建的块之中。“outside” 将列表项目标志放在块之外。当值为 “inside” 时，内容在列表项目标志之下卷绕。默认值为 “outside”。
2. 是的。唯一的不同是列表项目中的内容之前没有列表项目标志。
3. 使用如下代码：

```
<ul>
<li style="width:350px; height:100px; background-color:#00ff00;
border:2px dashed #000000; list-style-position:inside;">text goes
here</li>
</ul>
```

## 16.8 练习

- 寻找一个图像并且尝试使用本章中说明的技术映射各个区域。选择一个具有可以使用“热区”或者可点击区域的图像，指向网站上的其他网页或者其他网站。然后创建定义可点击区域和它们所应该指向的 URL 的 HTML 和 CSS。
- 准备使用下一章中的导航元素列表，考虑网站的结构并且草拟顶级导航和这些主要选择中的二级导航链接。考虑普遍使用的导航方法将是水平还是垂直导航。



## 第 17 章

# 使用 CSS 设计导航

---

本章中你将学到：

- 导航列表与常规列表的不同
- 使用 CSS 创建垂直导航的方法
- 使用 CSS 创建水平导航的方法

在前一章中，你学到了操纵列表外观和将列表用于表现其他内容（而不仅仅是带有项目符号和编号的项目集）的方法。在本章中，你将学习将列表用作垂直或者水平导航的一些方法，包括使用列表创建下拉式菜单。

本章中说明的方法只是使用列表创建各种导航方法的一小部分。但是，概念都类似，不同的结果来自于你自己的创意和这些基本概念的应用。为了帮助你激发创意，我在本章结束时将提供指向其他基于 CSS 的导航示例的链接。

### 17.1 导航列表与常规列表的不同

当我们讨论使用列表创建导航元素时，实际的意思是使用 CSS 将内容按照网站访问者预期的导航外观显示——简而言之，和简单的带有项目符号或者编号的列表不同。尽管导航元素本质上还是链接的列表，但是这些链接一般以使用户清晰地与内容交互的形式显示。

- 用户的鼠标指针必须改变，以指示该元素可点击。
- 元素周围的区域在鼠标悬停的时候改变外观。
- 内容区域在视觉上与常规文本不同。

创建导航的旧方法倾向于依赖图像——比如带有斜面的图像和使用背景和文本的对比颜色——加上根据鼠标操作切换图像的客户端 JavaScript 编程。但是使用纯粹的 CSS 创建来自列表项目的导航生成更可用、灵活和搜索引擎友好的显示，可以为使用所有风格和种类设备的用户访问。

不管导航元素的布局——水平或者垂直——本章讨论两种导航级别：初级和次级。初级导航将用户带到网站主要部分的介绍页面，次级导航反映某个部分中的页面。

### 自己动手尝试一下：创建网站范围内的导航

TRY IT YOURSELF ▼

在前一章最后的练习中，要求你考虑在自己的网站中使用的导航结构。如果你构思了信息架构——信息的分段以及在有必要时的子分段——这是个好的开始。你可以在本章中使用这些概要来创建以此结构为基础的垂直和水平导航。

## 17.2 用 CSS 创建垂直导航

根据你的网站架构——你已经创建的显示模板和网站中信息分类的风格——你可能发现自己使用垂直导航作为初级或者次级导航。

例如，假设你为公司创建了一个网站，初级部分为 About Us（关于我们）、Products（产品）、Support（支持）和 Press（新闻）。在初级的“关于我们”部分，你可能有多个其他页面，比如 Mission（任务）、History（历史）、Executive Team（执行团队），和 Contact Us（联系我们）——这些其他页面是“关于我们”部分中的次级导航。

程序清单 17.1 建立一个次级页面，页面的侧面有垂直导航，内容位于页面中央。侧面的链接和内容区域中的链接是基本的 HTML 列表元素。

这个程序清单和如图 17.1 所示的示例提供了一个出发点，向你展示 CSS 怎样使你能够将两个相似的 HTML 结构转化为两种不同的显示效果（从而成为两种不同的环境）。

### 程序清单 17.1 带有列表中的垂直导航的基本页面

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>About Us</title>
    <style type="text/css">
      body {
        font: 12pt Verdana, Arial, Georgia, sans-serif;
      }
      #nav {
        width:150px;
        float:left;
        margin-top:12px;
        margin-right:18px;
      }
      #content {
        width:550px;
        float:left;
      }
    </style>
  </head>

  <body>
    <div id="nav">
      <ul>
        <li><a href="#">Mission</a></li>
        <li><a href="#">History</a></li>
```



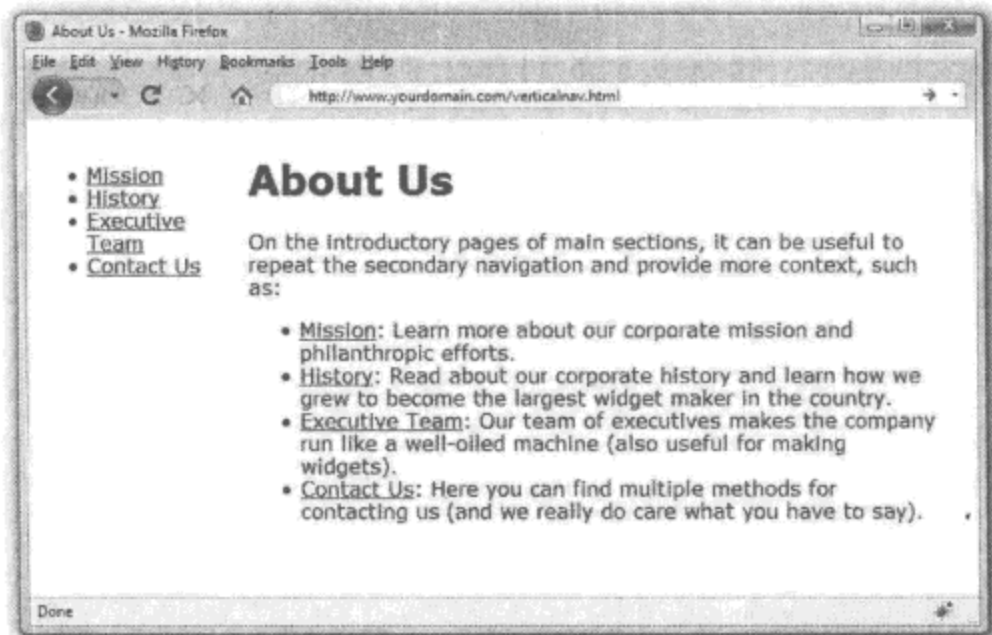
```

    <li><a href="#">Executive Team</a></li>
    <li><a href="#">Contact Us</a></li>
</ul>
</div>
<div id="content">
  <h1>About Us</h1>
  <p>On the introductory pages of main sections, it can be useful
  to repeat the secondary navigation and provide more context,
  such as:</p>
  <ul>
  <li><a href="#">Mission</a>: Learn more about our corporate
  mission and philanthropic efforts.</li>
  <li><a href="#">History</a>: Read about our corporate history
  and learn how we grew to become the largest widget maker
  in the country.</li>
  <li><a href="#">Executive Team</a>: Our team of executives makes
  the company run like a well-oiled machine (also useful for
  making widgets).</li>
  <li><a href="#">Contact Us</a>: Here you can find multiple
  methods for contacting us (and we really do care what you
  have to say).</li>
  </ul>
</div>
</body>
</html>

```

图 17.1

出发点：未添加样式的列表导航



这个页面的内容在两个相邻的<div>元素中：一个 id 值为 nav，另一个 id 值为 content。这个基本页面中分配的唯一样式是与<div>元素相关的宽度、边距和浮动，没有对列表元素应用样式。

为了区别内容区域和侧面导航中的列表内的链接，在样式单中添加如下样式：

```

#nav a {
  text-decoration: none;
}
#content a {
  text-decoration: none;
  font-weight: bold;
}

```

这些样式描述 id 为 nav 的<div>中的所有<a>链接没有下划线，id 为 content 的<div>中的所有<a>链接没有下划线并且为粗体。不同之处如图 17.2 所示。

为了真正地使网站导航列表有特殊的外观，必须更加深入挖掘样式单。

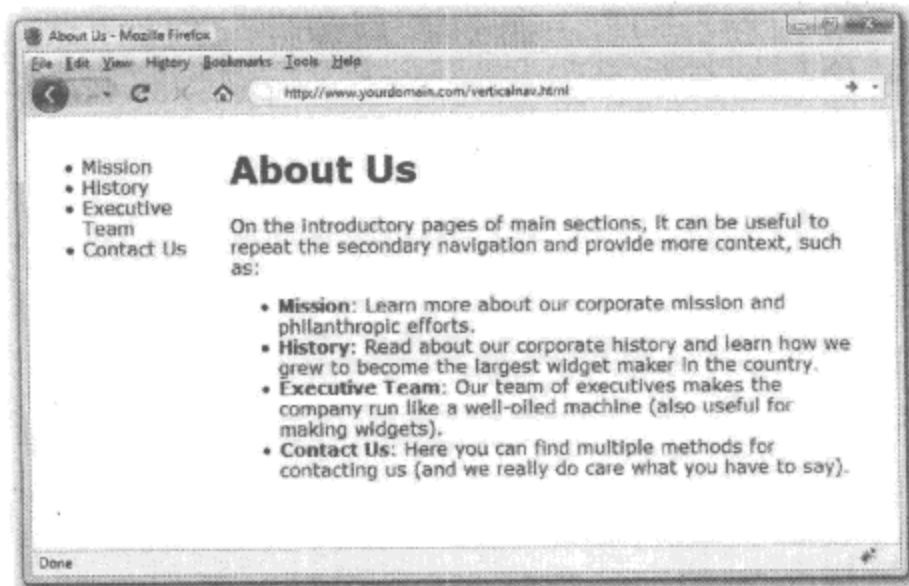


图 17.2

使用 CSS 区别列表元素

### 17.2.1 设置单级垂直导航样式

这个特别的导航元素集的目标是将它们表现为没有项目符号，具有根据链接状态（常规链接、访问过的链接、悬停连接和活动链接）改变的背景和文本颜色的链接块。这一过程的第一步已经完成：将导航从内容中分离。我们已经通过将导航放在 id 为 nav 的 <div> 中完成了这一步。

接下来，你必须修改定义 nav <div> 中链接的 <ul>。我们移除列表标志并且确定除了上边距之外，没有其他边距或者填充。上边距用于将导航的顶部与页面内容区域中的“About Us”标题文本对齐。

```
#nav ul {
  list-style: none;
  margin: 12px 0px 0px 0px;;
  padding: 0px;
}
```

因为导航列表项目本身显示为彩色区域，所以为每个列表项目设置一个下边框，以便在视觉上与内容分离：

```
#nav li {
  border-bottom: 1px solid #ffffff;
}
```

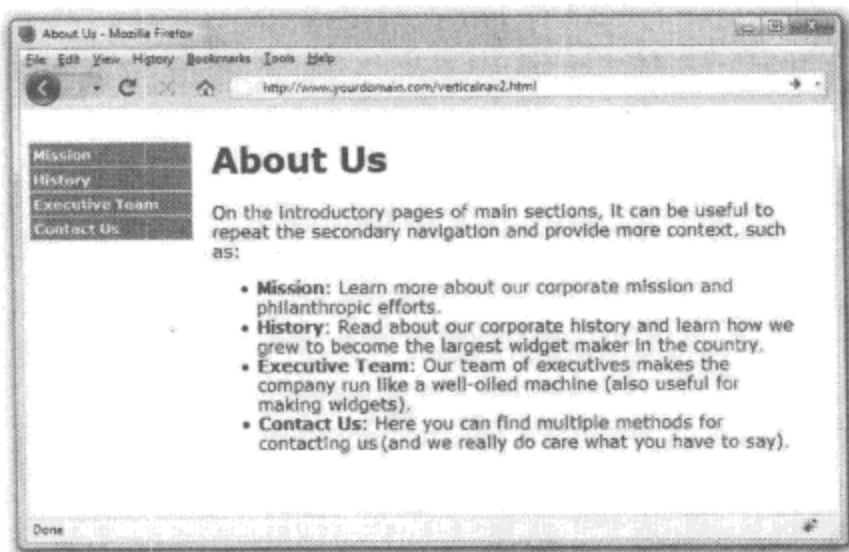
现在建立列表项目的余下部分。思路是列表项目仅仅作为链接时，它们是具有特殊的蓝色阴影的粗体白色文本（但是它们的字体比正文小）。为此，添加如下规则：

```
#nav li a:link, #nav li a:visited {
  font-size: 10pt;
  font-weight: bold;
  display: block;
  padding: 3px 0px 3px 3px;
  background-color: #628794;
  color: #ffffff;
}
```

前面使用的所有样式对你来说都很熟悉，只有 `display: block;` 可能是个例外。设置显示属性为块（block）确保整个 <li> 元素在用户鼠标悬停时起作用。图 17.3 所示为应用这些新的样式的垂直列表菜单。

图 17.3

垂直列表开始象导航菜单了



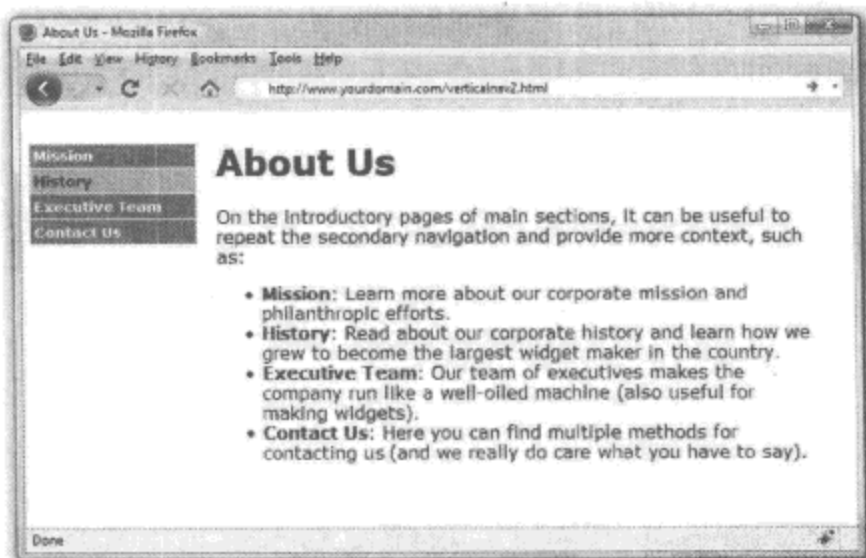
当用户的鼠标悬停于导航列表元素时，会发生一些视觉变化，使用户知道元素可点击。这与大部分软件菜单在用户鼠标悬停于菜单项目之上时改变颜色的方式相似。在这个例子中，我们将改变列表项目的背景颜色并将改变其文本颜色，它们将与前面显示的蓝色和白色不同。

```
#nav li a:hover, #nav li a:active {
    font-size: 10pt;
    font-weight: bold;
    display: block;
    padding: 3px 0px 3px 3px;
    background-color: #6cac46;
    color: #000000;
}
```

图 17.4 所示为目前所有样式的结果。通过使用样式单中的几个条目，简单的列表被转换为在视觉上有所不同的菜单。

图 17.4

列表项目在鼠标悬停时改变了颜色



## 17.2.2 设置多级垂直导航的样式

如果网站架构要求更多级别的导航，而你希望用户任何时候都能看到这些导航，该怎么做？这些导航由嵌套列表（你在前一章中学习过）和更多的样式单条目表现。在这个例子中，假定在 Executive Team（执行团队）链接下有 4 个导航项目，按照如下代码修改列表：

```
<ul>
  <li><a href="#">Mission</a></li>
  <li><a href="#">History</a></li>
  <li><a href="#">Executive Team</a>
    <ul>
```

```

<li><a href="#">&raquo; CEO</a>
<li><a href="#">&raquo; CFO</a>
<li><a href="#">&raquo; COO</a>
<li><a href="#">&raquo; Other Minions</a>
</ul>
</li>
<li><a href="#">Contact Us</a></li>
</ul>

```

这段代码在执行团队链接下产生一个嵌套列表，如图 17.5 所示。&raquo;实体产生一个新链接中文本之前显示的右向箭头。

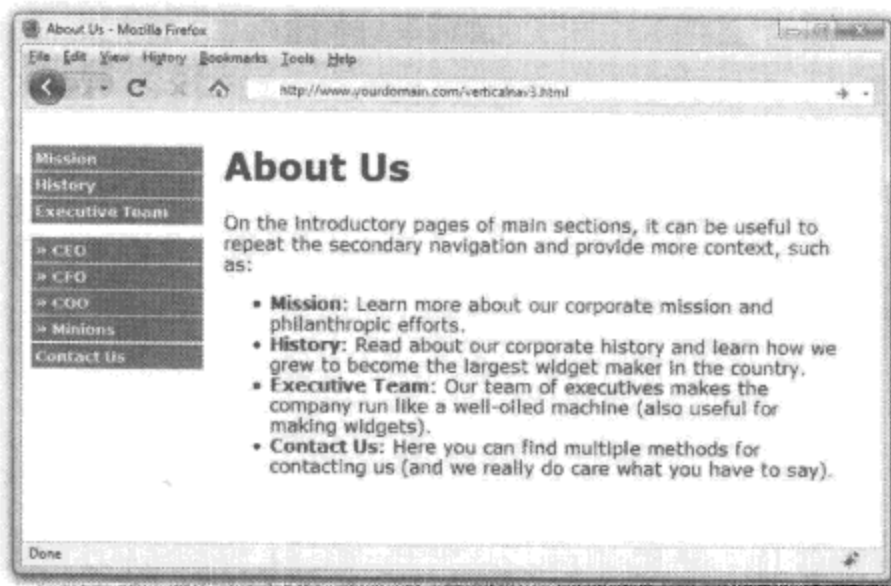


图 17.5

创建一个嵌套导航列表（还没有设置好样式）

新的项目作为列表中的块元素出现，但是信息的层次结构没有在视觉上表现出来。为了添加一些将这些项目标识为连接到执行团队链接的子导航元素的视觉元素，再次修改样式单添加一些缩进。

但是在这么做以前，还要修改其他的一些样式单条目。在前一小节，我们添加了选择符如 `#nav ul` 和 `#nav li`，这些选择符分别包含了“叫做 `nav` 的 `<div>` 中的所有 `<ul>`”和“叫做 `nav` 的 `<div>` 中的所有 `<li>`”。但是，现在我们有二个 `<ul>` 实例和另一组 `nav <div>` 中的 `<li>` 元素，我们希望这一组 `<li>` 与原来的显示有所不同。

为了确保两组列表项目有合适的样式，就要确保样式单选择符清晰地表现列表的层次。为此，将 `#nav ul` 和 `#nav ul li` 用于第一级列表，`#nav ul ul` 和 `#nav ul ul li` 用于第二级列表。程序清单 17.2 展示了新版本的样式单条目和产生如图 17.6 所示菜单的 HTML。

### 程序清单 17.2 列表中的多级垂直导航

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>About Us</title>
    <style type="text/css">
      body {
        font: 12pt Verdana, Arial, Georgia, sans-serif;
      }
      #nav {
        width:150px;
        float:left;

```



```
margin-top:12px;
margin-right:18px;
}
#content {
width:550px;
float:left;
}
#nav a {
text-decoration: none;
}
#content a {
text-decoration: none;
font-weight: bold;
}
#nav ul {
list-style: none;
margin: 12px 0px 0px 0px;
padding: 0px;
}
#nav ul li {
border-bottom: 1px solid #ffffff;
}
#nav ul li a:link, #nav ul li a:visited {
font-size: 10pt;
font-weight: bold;
display: block;
padding: 3px 0px 3px 3px;
background-color: #628794;
color: #ffffff;
}
#nav ul li a:hover, #nav ul li a:active {
font-size: 10pt;
font-weight: bold;
display: block;
padding: 3px 0px 3px 3px;
background-color: #c6a648;
color: #000000;
}
#nav ul ul {
margin: 0px;
padding: 0px;
}
#nav ul ul li {
border-bottom: none;
}
#nav ul ul li a:link, #nav ul ul li a:visited {
font-size: 8pt;
font-weight: bold;
display: block;
padding: 3px 0px 3px 18px;
background-color: #628794;
color: #ffffff;
}
#nav ul ul li a:hover, #nav ul ul li a:active {
font-size: 8pt;
font-weight: bold;
display: block;
padding: 3px 0px 3px 18px;
background-color: #c6a648;
color: #000000;
}
}
```



```
</style>
</head>

<body>
  <div id="nav">
    <ul>
      <li><a href="#">Mission</a></li>
      <li><a href="#">History</a></li>
      <li><a href="#">Executive Team</a>
        <ul>
          <li><a href="#">&raquo; CEO</a>
          <li><a href="#">&raquo; CFO</a>
          <li><a href="#">&raquo; COO</a>
          <li><a href="#">&raquo; Other Minions</a>
        </ul>
      </li>
      <li><a href="#">Contact Us</a></li>
    </ul>
  </div>
  <div id="content">
    <h1>About Us</h1>
    <p>On the introductory pages of main sections, it can be useful
    to repeat the secondary navigation and provide more context,
    such as:</p>
    <ul>
      <li><a href="#">Mission</a>: Learn more about our corporate
      mission and philanthropic efforts.</li>
      <li><a href="#">History</a>: Read about our corporate history
      and learn how we grew to become the largest widget maker
      in the country.</li>
      <li><a href="#">Executive Team</a>: Our team of executives makes
      the company run like a well-oiled machine (also useful for
      making widgets).</li>
      <li><a href="#">Contact Us</a>: Here you can find multiple
      methods for contacting us (and we really do care what you
      have to say.</li>
    </ul>
  </div>
</body>
</html>
```

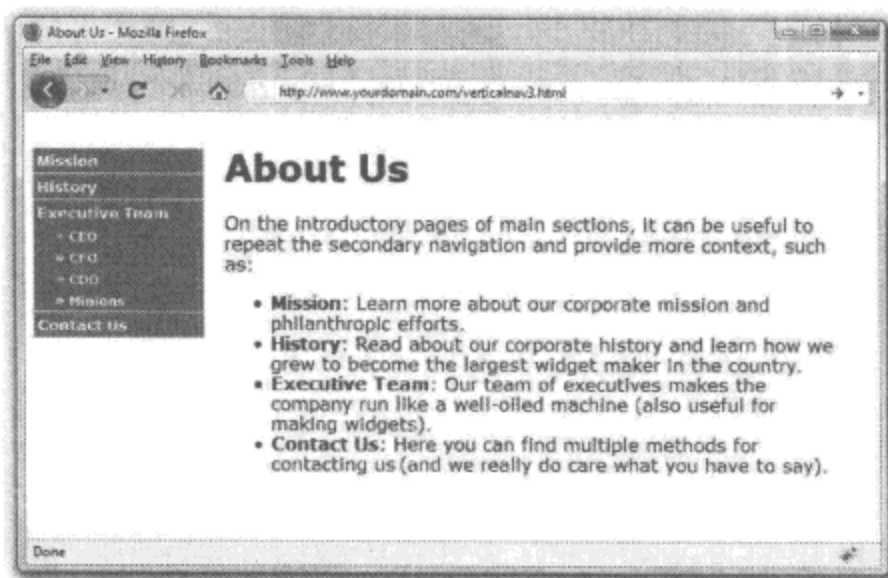


图 17.6

使用 CSS 创建两  
级垂直导航

设置垂直导航样式的不同方法仅受限于你的创意。你可以使用颜色、填充、背景图像，以及其他有效的 CSS 来产生相当灵活和容易修改的垂直导航。如果在搜索引擎中输入“CSS 垂直导航”，将会找到几千个示例——它们都根据在本章中学习到的简单原则。

## 17.3 用 CSS 创建水平导航

本章从垂直导航开始是因为你可能将列表项目垂直地写在纸上，垂直导航的外观与此类似可能更易于掌握。创建水平导航时，你仍然使用 HTML 列表元素，但是通过使用 `<ul>` 和 `<li>` 的 `display` 属性的 `inline` 值，代替 `block` 值所完成的垂直显示。确实就是这么简单。

程序清单 17.3 展示了表现水平导航的页面的一个出发点。该页面包含了两个主要的 `<div>` 元素：一个用于标题，一个用于内容。标题 `<div>` 包含一个标志 `<div>` 和一个导航 `<div>`，两者互相浮动。出现在导航 `<div>` 中的列表及其列表项目的 `display` 值为 `inline`。如图 17.7 所示，你可以在图中看到这些元素及其位置。

程序清单 17.3 来自列表的基本水平导航

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>ACME Widgets LLC</title>
    <style type="text/css">
      body {
        font: 12pt Verdana, Arial, Georgia, sans-serif;
      }
      #header {
        width: auto;
      }
      #logo {
        float:left;
      }
      #nav {
        float:left;
      }
      #nav ul {
        list-style: none;
        display: inline;
      }
      #nav li {
        display: inline;
      }
      #content {
        width: auto;
        float: left;
        clear: left;
      }
      #content a {
        text-decoration: none;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <div id="header">
      <div id="logo">
        
      </div>
```



```

<div id="nav">
  <ul>
    <li><a href="#">About Us</a></li>
    <li><a href="#">Products</a></li>
    <li><a href="#">Support</a></li>
    <li><a href="#">Press</a></li>
  </ul>
</div>
</div>
<div id="content">
  <p><strong>ACME Widgets LLC</strong> is the greatest widget-maker
  in all the land.</p>
  <p>Don't believe us? Read on...</p>
  <ul>
    <li><a href="#">About Us</a>: We are pretty great.</li>
    <li><a href="#">Products</a>: Our products are the best.</li>
    <li><a href="#">Support</a>: It is unlikely you will need support,
    but we provide it anyway.</li>
    <li><a href="#">Press</a>: Read what others are saying (about how
    great we are).</li>
  </ul>
</div>
</body>
</html>

```

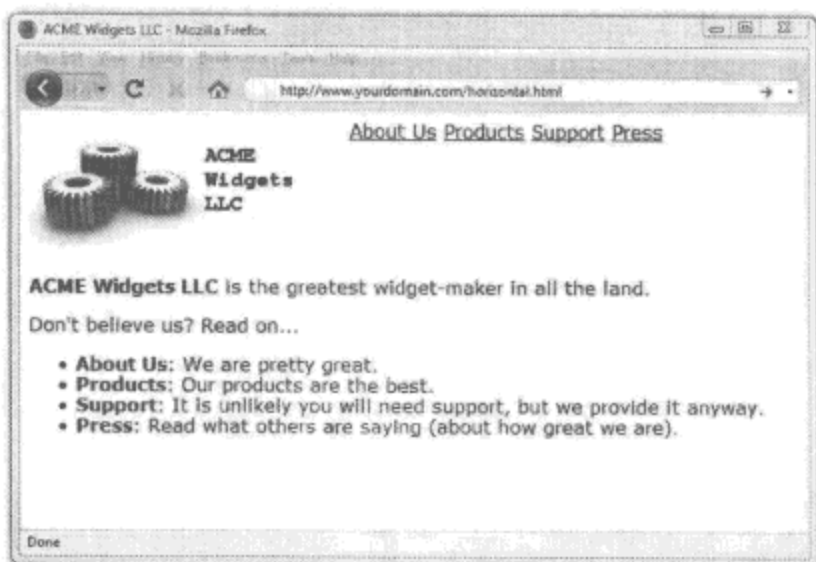


图 17.7

使用内联列表项目创建起作用（但不一定漂亮）的水平导航

对列表显示的修改完全通过 CSS，HTML 中的内容结果已经设置好。为了完成需要的显示，使用如下的 CSS。首先，id 为 nav 的 <div> 修改为特定的宽度，显示背景颜色和边框，并且使用 85 个像素的上边距（这样它靠近标志的底部显示）。

```

#nav {
  float:left;
  margin: 85px 0px 0px 0px;
  width: 400px;
  background-color: #628794;
  border: 1px solid black;
}

```

<ul> 的定义保持和程序清单 17.3 相同，只是修改了边距和填充：

```

#nav ul {
  margin: 0px;
  padding: 0px;
  list-style: none;
  display: inline;
}

```

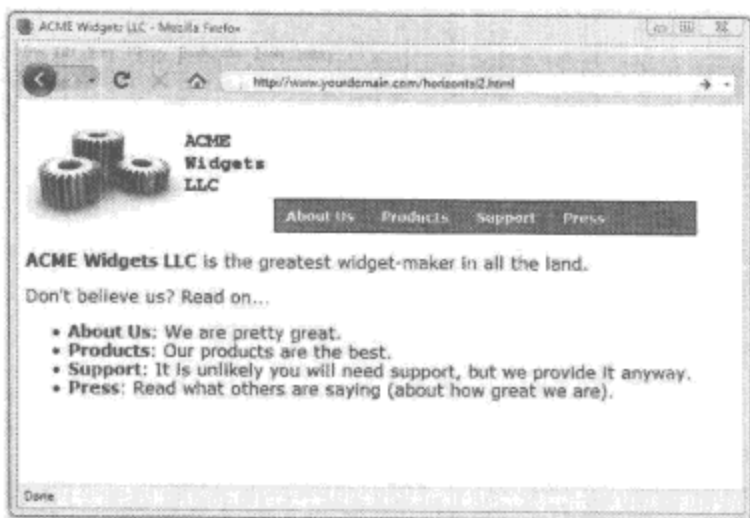
链接样式与垂直导航中类似，这些条目有不同的填充值，但是颜色和字体大小保持不变：

```
#nav li {
    display: inline;
    line-height: 1.8em;
}
#nav ul li a:link, #nav ul li a:visited {
    font-size: 10pt;
    font-weight: bold;
    text-decoration: none;
    padding: 7px 10px 7px 10px;
    background-color: #628794;
    color: #ffffff;
}
#nav ul li a:hover, #nav ul li a:active {
    font-size: 10pt;
    font-weight: bold;
    text-decoration: none;
    padding: 7px 10px 7px 10px;
    background-color: #c6a648;
    color: #000000;
}
```

合并这些样式，产生如图 17.8 所示的效果。

图 17.8

使用一些样式创建  
水平导航



当用户切换导航元素时，背景和文本颜色和用户鼠标悬停于垂直导航菜单之上时一样改变。而且，正如对垂直导航菜单的处理一样，你也可以使用嵌套列表来在水平菜单中产生下拉式功能。你可以动手试试！

## 17.4 总结

在本章中，你学习了使用 CSS 将简单的 HTML 无序列表转换为水平和垂直导航的方法。通过使用 CSS 代替图像、JavaScript 或者其他技术，你将在网站的显示和维护上有更大的灵活性。通过本章，你学到了在样式单中使用少数条目，就能够将普通的下划线文本链接转换为具有边框和背景颜色以及其他文本样式的区域。此外，你还学习了在菜单中表现嵌套列表的方法。

## 17.5 问与答

问：我可以在导航菜单中将图像作为自定义列表标志吗？

答：可以。你可以在列表项目的 HTML 文本中使用图像或者将图像作为 <li> 元素中的背

景图像。你可以像设置其他列表项目样式那样，为导航元素设置样式。HTML 无序列表和基于 CSS 的水平和垂直导航列表的唯一不同是你对它的命名，并将无序列表用于文本主体之外的目的。除此之外，你设置该列表的样式来向用户表明它确实是不同的东西——也可以使用小的图像来强调列表。

问：我在哪里可以找到使用列表的更多示例？

答：我最近一次在搜索引擎中输入 CSS 导航，得到了大约 4400 万个结果。以下是一些起点：

- A List Apart's CSS articles——<http://www.alistapart.com/topics/code/>
- Maxdesign's CSS Listamatic——<http://css.maxdesign.com.au/listamatic>
- Vitaly Friedman's CSS Showcase——<http://www.alvit.de/css-showcase/>

## 17.6 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 17.6.1 测验

1. 创建基于列表的导航时，可以使用多少级的嵌套列表？
2. 是非题：`display` 属性值 `inline` 生成水平列表。
3. 创建任何类型的导航列表时，`a` 选择符的 4 个伪类可否为相同值？

### 17.6.2 测验答案

1. 从技术上说，可以嵌套任意多的级别。但是从可用性角度看，嵌套列表的级数是有限的。这个限制通常是 3 级——如果超过这一限制，网站的组织可能很糟糕，用户在任何时候都看到了过多的选项。

2. 正确，如果 `display` 属性用于样式单中的 `ul` 和 `li` 选择符的话。

3. 当然可以，但是之后用户可能不会意识到漂亮的菜单确实是个菜单（因为鼠标操作时没有视觉效果）。

## 17.7 练习

- 使用多级垂直列表的技术，为本章末尾创建的垂直列表添加子导航。
- 观察用于网站的许多基于 CSS 的导航示例并且寻找一些看上去很复杂的操作。使用浏览器的“查看源代码”功能，查看这些网站使用的 CSS 并且尝试实现类似的功能。



## 第 18 章

# 使用鼠标操作修改文本显示

---

本章中你将学到：

- 用 CSS 创建工具提示的方法
- 用 CSS 显示附加的翻转文本的方法
- 访问事件的方法
- 使用 onclick 改变<div>外观的方法

在本书中，你已经看到了鼠标操作影响文本显示的示例。最简单的示例是样式单中定义了<a>链接的鼠标悬停伪类时，在用户鼠标悬停时以不同的颜色显示文本。在前一章中，你学习了悬停伪类对文本颜色和<li>元素的背景颜色的改变。

在本章中，你将学习如何完成两个组合 CSS 和鼠标操作的特殊操作——在使用鼠标操作项目时显示附加文本，以及使用鼠标点击来改变容器元素的颜色。这两种操作在一些场合中很有用，但是更重要的是，它们提供了在以后的 Web 开发中你可能需要的更高级工作的一个进入点。

### 18.1 用 CSS 创建工具提示

工具提示 (tooltip) 是图形用户界面的一个元素——这个界面可以是软件程序，也可以是网站——工具提示在鼠标悬停于项目之上时显示附加的信息。图 18.1 展示了一个运行中的工具提示：鼠标悬停于链接文本“HTML”之上，工具提示在一个小方框中显示“HTML”。在这个例子中，显示在工具提示中的文本来自<a>标签的 title 属性。

图 18.1 所示的工具提示由软件本身控制。也就是说，它的显示和外观不是你（作为 Web 开发人员）所能控制的。但是，你可以应用在前面的章节中已经学习到的概念，用少量的 CSS 来创建自己的工具提示。

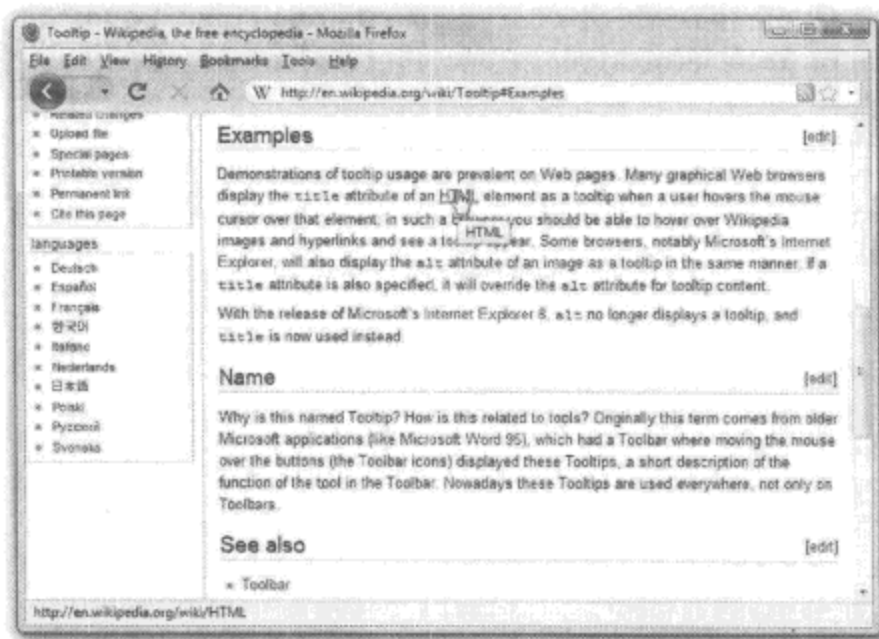


图 18.1

标准的工具提示

程序清单 18.1 包含了用于包含图像和链接的一个网页的样式单和 HTML, 这个网页还有一个显示自定义提示的链接。

### 程序清单 18.1 用 CSS 创建一个简单的工具提示

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Steptoe Butte</title>
    <style type="text/css">
      a {
        text-decoration: none;
        font-weight: bold;
      }
      a.tip {
        position: relative;
        z-index: 24;
      }
      a.tip: hover {
        z-index: 25;
      }
      a.tip span {
        display: none;
      }
      a.tip: hover span {
        font-weight: normal;
        display: block;
        position: absolute;
        top: 20px;
        left: 25px;
        width: 150px;
        padding: 3px;
        border: 1px solid #000;
        background-color: #ddd;
        color: #000;
      }
    </style>
  </head>
  <body>
    <h1>Steptoe Butte</h1>
```



```

<p>Steptoe Butte is a quartzite island jutting out of the
silty loess of the <a class="tip"
href="http://en.wikipedia.org/wiki/Palouse">Palouse <span>Learn more
about the Palouse!</span></a> hills in Whitman County, Washington. The
rock that forms the butte is over 400 million years old, in contrast
with the 15-7 million year old
<a href="http://en.wikipedia.org/wiki/Columbia_River">Columbia River</a>
basalts that underlie the rest of the Palouse (such "islands" of ancient
rock have come to be called buttes, a butte being defined as a small hill
with a flat top, whose width at top does not exceed its height).</p>
<p>A hotel built by Cashup Davis stood atop Steptoe Butte from 1888 to
1908, burning down several years after it closed. In 1946, Virgil McCroskey
donated 120 acres (0.49 km2) of land to form Steptoe Butte State Park,
which was later increased to over 150 acres (0.61 km2). Steptoe Butte is
currently recognized as a National Natural Landmark because of its unique
geological value. It is named in honor of
<a href="http://en.wikipedia.org/wiki/Colonel_Edward_Steptoe">Colonel Edward
Steptoe</a>.</p>
<p>Elevation: 3,612 feet (1,101 m), approximately 1,000 feet (300 m) above
the
surrounding countryside.</p>
<p><em>Text from
<a href="http://en.wikipedia.org/wiki/Steptoe_Butte">Wikipedia</a>, photo by
the author.</em></p>
</body>
</html>

```

注意，文本中的第一个链接有一个相关的类。这个类被称为 `tip`——该名称使其区别于其他不包含工具提示的链接。但是，因为这是一个 `class` 而不是 `id`，所以也可以用于页面中的所有其他链接。

注意，`tip` 类自身定义了 `position` 和 `z-index`。`position` 值 (`relative`) 确保了它保持自然地出现在文本中，这里的 `z-index` 与其他相关样式中使用的 `z-index` 相对，确保了链接文本留在其他有更高 `z-index` 值的文本下面。例如，使用 `tip` 类的链接的悬停状态具有更高的 `z-index` 值。

接下来的两个样式与在用的 `<span>` 标签相关。`<a>` 标签中的 `<span>` 标签的文本将不会显示，除非鼠标悬停于可见链接之上而调用工具提示。例如，链接文本如下：

```

<a class="tip" href="http://en.wikipedia.org/wiki/Palouse">Palouse
<span>Learn more about the Palouse!</span></a>

```

当你查看这一小段 HTML 时，链接文本好像应该是“Palouse Learn more about the Palouse!”。但是因为“Learn more about the Palouse!”包含在 `<span>` 标签中，而样式单中有应用到这个标签的具体样式，所以在用户鼠标悬停于实际的链接 (Palouse) 上之前这句话不会出现。图 18.2 所示为这一结果。

HTML 中，“Learn more about the Palouse!”被放置在一个 `<span>` 标签中，但是在样式单中设置了样式。也就是说，`a.tip:hover span` 选择符创建了一个 150 个像素宽，具有灰色背景和黑色边框的方框。这个方框出现在距离元素顶部 20 个像素，父元素左边 25 个像素的地方，这创建了工具提示。

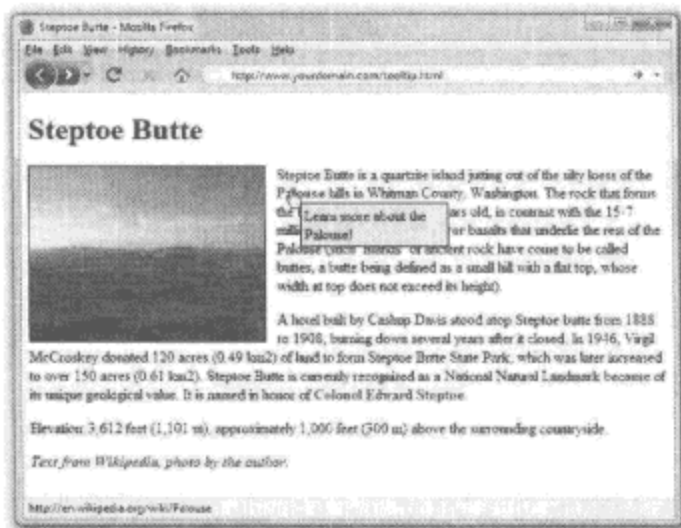


图 18.2

当用户悬停与页面第一个链接时，显示一个自定义工具提示

## 18.2 用 CSS 显示附加的翻转文本

工具提示有一个特殊的用途：显示附加到链接的文本“提示”。但是你可以使用相同的概念来根据鼠标操作显示布局中其他地方的隐含文本。例如，第 17 章中的 ACME Widgets LLC 主页——具有水平导航的版本。通过使用程序清单 17.1 中的概念，你可以在用户操作一个主要部分链接时在菜单之上显示附加文本。图 18.3 所示为一个这种操作，我占用了标志右边的空白部分。

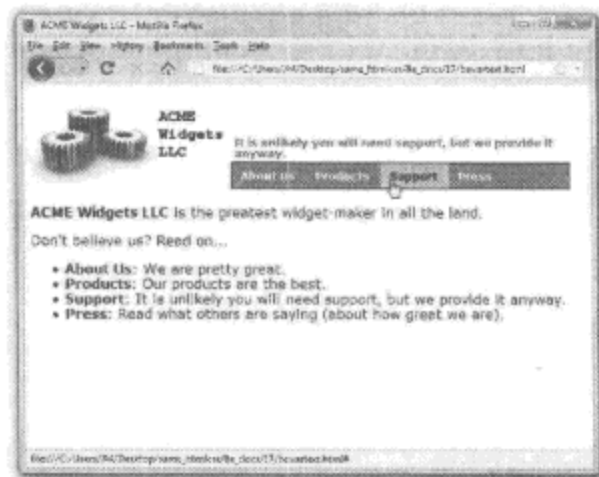


图 18.3

在另一个元素上的自定义文本，仅在鼠标操作时可见

程序清单 18.3 中对 CSS 的唯一修改是添加了下面 4 个样式。它们执行与程序清单 18.1 中的 tip 类相同的任务，但是显示与其略有不同。

```

a.more {
  position: relative;
  z-index: 24;
}
a.more: hover {
  z-index: 25;
}
a.more span {
  display: none;
}
a.more: hover span {
  font-weight: bold;
  display: block;
  position: absolute;
  top: -35px;
  width: 400px;
  padding: 3px;
  color: #ff0000;
  line-height: 1em;
}

```



-25 个像素的 `top` 属性值将 `<span>` 标签中的文本放置在父标签左上角之上 25 个像素的地方，这和前一个示例将文本放置在父标签左上角之下 20 个像素相反。对 HTML 的改变与前一小节中相似，文本被隐藏，直到 `<a>` 链接中的 `<span>` 标签发生操作：

```
<ul>
  <li><a class="more" href="#">About Us <span>We are pretty
  great.</span></a></li>
  <li><a class="more" href="#">Products <span>Our products are
  the best.</span></a></li>
  <li><a class="more" href="#">Support <span>It is unlikely you
  will need support, but we provide it anyway.</span></a></li>
  <li><a class="more" href="#">Press <span>Read what others are
  saying (about how great we are).</span></a></li>
</ul>
```

使用这些基本概念，你可以建立出现在任何地方的文本，这些文本在用户鼠标悬停于 `<a>` 链接之上时根据鼠标操作显示。在下一小节，你将学习如何使用事件属性加上少量 JavaScript 来进一步处理鼠标操作和 CSS。

### 18.3 访问事件

鼠标点击或者按键这些用户交互被称为事件 (event)。根据事件采取措施的脚本过程被称为事件处理 (event handling)。使用特殊的属性可以将事件处理脚本代码与网页元素关联。

下面是一些常用的事件属性，以及它们发生的时间描述，以下属性在 JavaScript 中迟早会用到。

- `onload`——浏览器装载元素。
- `onkeydown`——用户按下一个键。
- `onkeyup`——用户放开按键。
- `onclick`——用户在元素上单击鼠标左键。
- `ondblclick`——用户在元素上双击鼠标左键。
- `onmousedown`——用户在鼠标指针位于元素之上时按下任意鼠标按钮。
- `onmouseup`——用户在鼠标指针位于元素之上时放开任意鼠标按钮。
- `onmouseover`——用户将鼠标指针移入元素区域之中。
- `onmousemove`——用户在鼠标指针位于元素之上时移动鼠标。
- `onmouseout`——用户将鼠标指针移出元素区域。

你可以看到，事件属性用于响应常见的用户输入事件，如鼠标点击和按键。你可以将代码分配给事件属性，将 JavaScript 关联到事件，如：

```
<h1 onclick="this.style.color='red';">I turn red when clicked.</h1>
```

在上面的代码示例中，分配给 `<h1>` 标签的 `onclick` 事件属性一个 JavaScript 片段，也就是说这段代码对用户文本上单击鼠标左键作出响应。脚本代码设置文本颜色为红色。这样，通过相应鼠标单击改变文本颜色，使通常平淡的文本增加了交互性。这是客户端脚本与浏览器协作的基础。

在下一个小节，你将看到使用事件处理改变<div>外观的示例。具体地说，<div>内容根据鼠标点击显示或者消失。

## 18.4 使用 onclick 改变<div>外观

onclick 事件可以用于调用所有操作，你可能将鼠标单击看作单击按钮来提交表单的一种方式，也可以用在网页中提供交互性。在这个示例中，你将看到使用 onclick 事件可以显示或者隐藏包含在<div>中的内容。使用程序清单中相同的文本和图像，你将看到使用户能单击一段文本来显示前面隐藏的文本，从而为页面增加交互性。我们称之为“一段文本”是因为严格地说，该文本不是一个链接。也就是说，它的样子和表现都象链接，但不是用<a>标签标示的。

程序清单 18.2 提供这个示例的完整代码，效果如图 18.4 所示。

### 程序清单 18.2 使用 onclick 显示或者隐藏内容

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Steptoe Butte</title>
    <style type="text/css">
      a {
        text-decoration: none;
        font-weight: bold;
        color: #7a7abf;
      }
      #hide_e {
        display: none;
      }
      #elevation {
        display: none;
      }
      #hide_p {
        display: none;
      }
      #photos {
        display: none;
      }
      #show_e {
        display: block;
      }
      #show_p {
        display: block;
      }
      .fakelink {
        cursor:pointer;
        text-decoration: none;
        font-weight: bold;
        color: #E03A3E;
      }
    </style>
  </head>
  <body>
```





```

<h1>Steptoe Butte</h1>
<p>Steptoe Butte is a quartzite island jutting out of the
silty loess of the <a class="tip"
href="http://en.wikipedia.org/wiki/Palouse">Palouse <span>Learn more
about the Palouse!</span></a> hills in Whitman County, Washington. The
rock that forms the butte is over 400 million years old, in contrast
with the 15-7 million year old
<a href="http://en.wikipedia.org/wiki/Columbia_River">Columbia

```

```
River</a>
```

```

basalts that underlie the rest of the Palouse (such "islands" of ancient
rock have come to be called buttes, a butte being defined as a small hill
with a flat top, whose width at top does not exceed its height).</p>

```

```

<p>A hotel built by Cashup Davis stood atop Steptoe Butte from 1888 to
1908, burning down several years after it closed. In 1946, Virgil McCroskey
donated 120 acres (0.49 km2) of land to form Steptoe Butte State Park,
which was later increased to over 150 acres (0.61 km2). Steptoe Butte is
currently recognized as a National Natural Landmark because of its unique
geological value. It is named in honor of

```

```

<a href="http://en.wikipedia.org/wiki/Colonel_Edward_Steptoe">Colonel Edward
Steptoe</a>.</p>

```

```

<div class="fakelink"
  id="show_e"
  onclick="this.style.display='none';
  document.getElementById('hide_e').style.display='block';
  document.getElementById('elevation').style.display='inline';
">&raquo; Show Elevation</div>

```

```

<div class="fakelink"
  id="hide_e"
  onclick="this.style.display='none';
  document.getElementById('show_e').style.display='block';
  document.getElementById('elevation').style.display='none';
">&raquo; Hide Elevation</div>

```

```

<div id="elevation">3,612 feet (1,101 m), approximately 1,000 feet (300 m)
above the surrounding countryside.</div>

```

```

<div class="fakelink"
  id="show_p"
  onclick="this.style.display='none';
  document.getElementById('hide_p').style.display='block';
  document.getElementById('photos').style.display='inline';
">&raquo; Show Photos from the Top of Steptoe Butte</div>

```

```

<div class="fakelink"
  id="hide_p"
  onclick="this.style.display='none';
  document.getElementById('show_p').style.display='block';
  document.getElementById('photos').style.display='none';
">&raquo; Hide Photos from the Top of Steptoe Butte</div>

```

```

<div id="photos"></div>

```

```

<p><em>Text from

```

```

<a href="http://en.wikipedia.org/wiki/Steptoe_Butte">Wikipedia</a>, photos
by the author.</em></p>

```



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

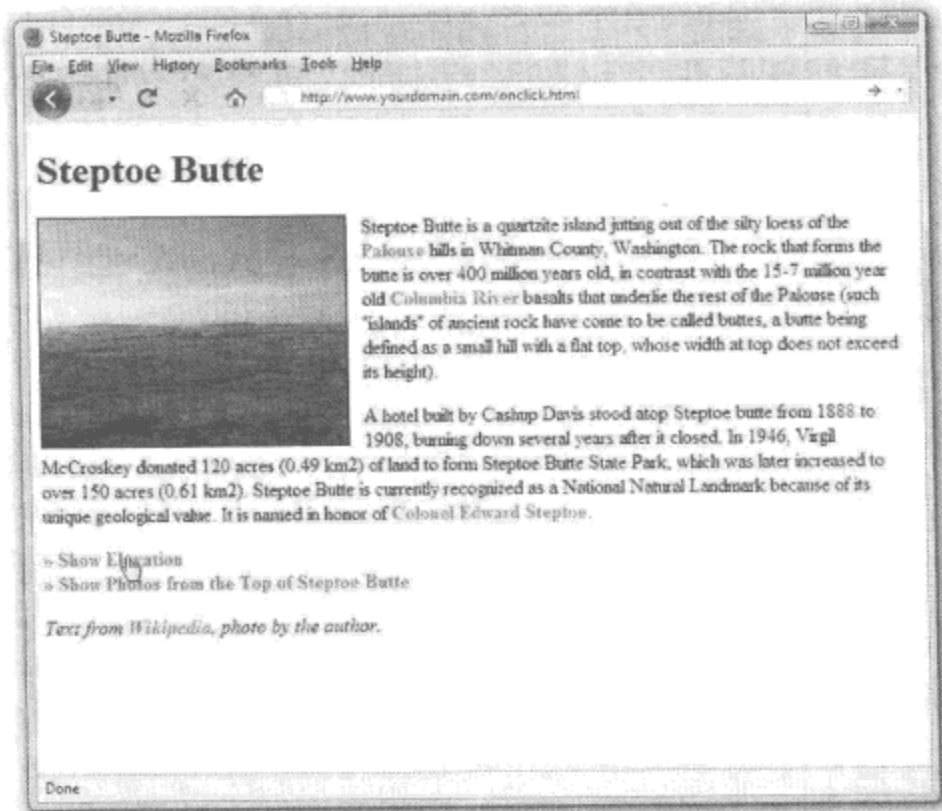


图 18.4

程序清单 18.2 的初始显示。注意，在鼠标悬停于红色文本上时，尽管它不是[<a>链接](#)，鼠标的指针还是变成手型指针

先看看样式单中的 6 个条目。第一个条目简单地给 `<a>` 标签中的链接设置样式，这些链接显示为无下划线、粗体的蓝色链接。你可以在图 18.4 中看到两段文本中（还有页面底部的文本行）的常规链接。

接下来的 4 个条目用于具体的 ID，这些 ID 都被设置为在页面装入时不可见 (`display:none`)。后面的 2 个 ID 被设置为在页面刚装入时显示为块元素。同样严格地说，这两个 ID 没有必要定义，因为这是默认的显示，样式单包含这些条目是为了表现不同之处。如果你计算程序清单 18.1 中的 `<div>` 元素数目，你将发现代码中一共有 6 个：4 个在页面装入时不可见，2 个可见。

这个示例的目标是在另一个 ID 被点击时改变两个 ID 的显示值。但是你首先必须确定用户意识到这段文本是可以点击的，这一般发生在用户看到鼠标指针改变以反映存在一个链接的时候。注意图 18.4 中鼠标指针变为手型，手指指向特定的链接。

这个功能通过为特定文本定义一个类来完成，该类称为 `fakelink`，正如你在以下这段代码中看到的：

```
<div class="fakelink"
      id="show_e"
      onclick="this.style.display='none';
              document.getElementById('hide_e').style.display='block';
              document.getElementById('elevation').style.display='inline';
              ">&raquo; Show Elevation</div>
```

`fakelink` 类确保文本被显示为无下划线、粗体、红色；`cursor: pointer` 使鼠标指针改变，这样用户认为这段文本是通常包含在 `<a>` 标签对之中的链接类型。但是真正有趣的是我们将 `onclick` 属性与 `<div>` 关联。在刚刚看到的示例片段中，`onclick` 属性值是改变 CSS 元素当前值的一系列命令。

让我们单独来看看这些命令：

```

this.style.display='none';
document.getElementById('hide_e').style.display='block';
document.getElementById('elevation').style.display='inline';

```

你所看到的是用于改变特定元素的不同 JavaScript 方法。你将在第 21 章中学到一些关于 JavaScript 的内容。总体来说，JavaScript 超出了本书的范围，但是我认为你可以按照这里介绍的方法进行。

在第一行中，`this` 关键字引用元素本身。换句话说，`this` 引用 ID 为 `show_e` 的 `<div>`。关键字 `style` 引用样式对象，样式对象包含所有分配给元素的 CSS 样式。在这个例子中，我们最感兴趣的是显示样式。因此，`this.style.display` 的意思是“`show_e` 的显示样式”，这里我们所做的是在文本被单击时设置显示样式为不显示 (`none`)。

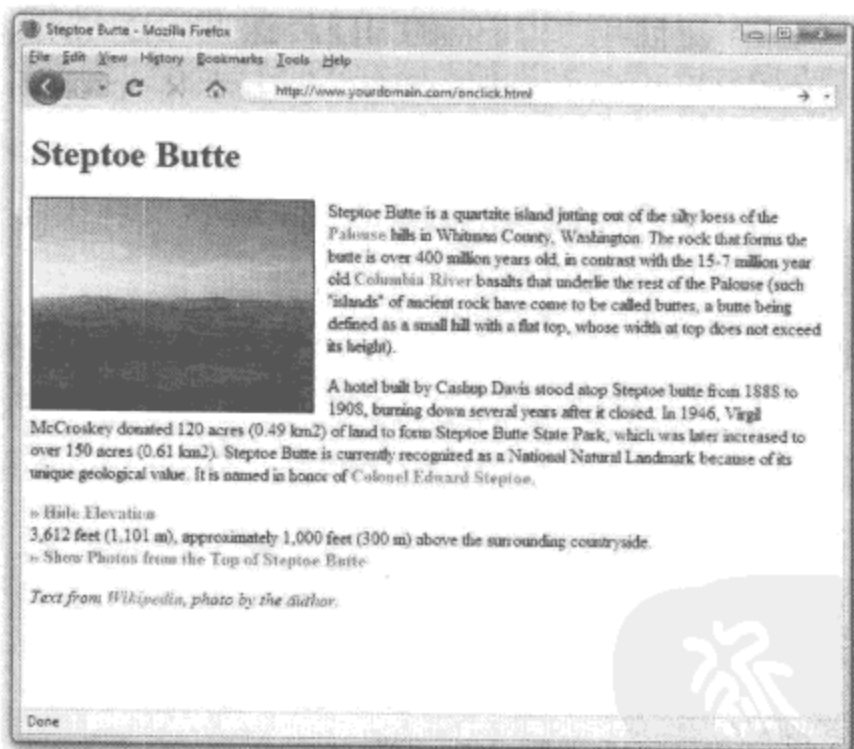
但是这不是我们所做的全部工作，因为在 `onclick` 属性中有 3 个操作。另两个操作从 `document.getElementById()` 开始并且在括号中包含一个特定的 ID 名称。我们用 `document.getElementById()` 代替 `this` 是因为第二个和第三个操作设置的不是父元素的 CSS 样式属性。你在片段中可以看到，在第二个和第三个操作中，我们为 ID 为 `hide_e` 和 `elevation` 的元素设置显示属性值。当用户单击当前可见的 `show_e <div>` 时，发生如下情况。

- `show_e <div>` 变为不可见。
- `hide_e <div>` 变为可见并且显示为块。
- `elevation <div>` 变为可见并且在行内显示。

这些操作的结果如图 18.5 所示。

图 18.5

单击“Show Elevation”之后，根据 `onclick` 属性中的命令，它和其他 `<div>` 元素的可见性发生变化



程序清单 18.2 中有另一组 `<div>` 元素，控制附加的照片的可见性。这些元素不受与海拔 (`elevation`) 相关的元素中的 `onclick` 操作的影响。也就是说，当你单击“Show Elevation”或者“Hide Elevation”时，这些与照片相关的 `<div>` 没有变化。你可以显示海拔而不显示照片（见图 18.5），显示照片而不显示海拔，也可以同时显示海拔和照片，如图 18.6 所示。

这个简短的示例已经向你初步展示了当你掌握了 CSS 与事件的协同之后，为布局和交互带来的可能性。例如，你可以编码网页，使用户能够修改样式单元素或者修改为一个完全不同的样式单，将文本块转移到其他位置，进行问卷调查或者提交表单，还能实现其他许多的

功能。

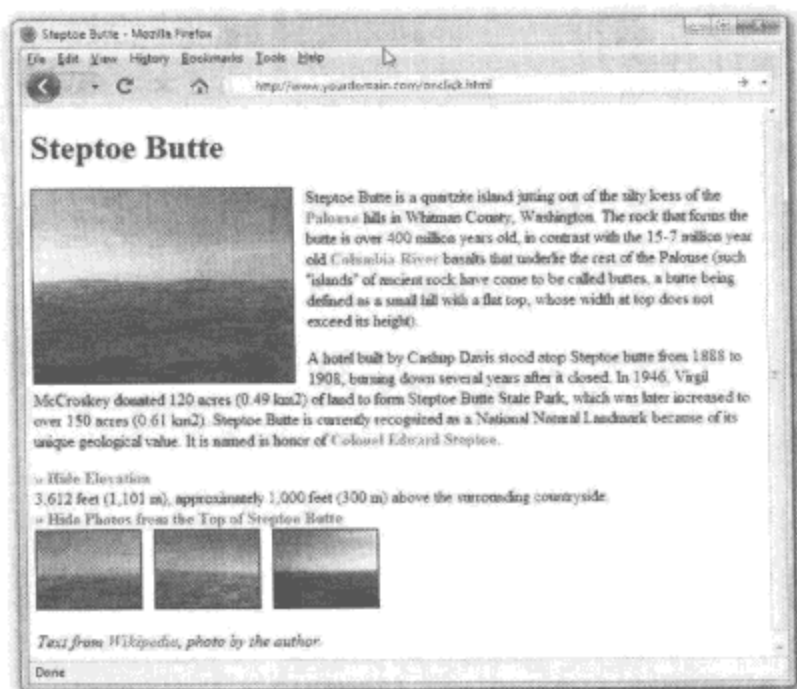


图 18.6

单击“Show Elevation”和“Show Photos from the Top of Steptoe butte”之后的页面

## 18.5 总结

在本章中，你学习了鼠标操作对文本显示的影响。在前两个小节中，这些操作根据与标签的鼠标悬停伪类相关的样式，而在之后的部分，介绍了与特定的用户交互事件相关联的操作。示例中的事件是 onclick 事件，但是你也看到了其他可能事件的列表，如 onload 和 onmouseover。

代码中介绍的唯一新内容是 cursor 属性的使用。将 cursor 属性设置为 pointer 使你能够指示用户特定的文本表现得和链接一样，尽管它们没有包含在熟悉的标签中。

## 18.6 问与答

问：本章介绍的一些事件很象标签的伪类，两者差别在哪里？

答：确实，onmousedown 事件很象活动状态，而 onmouseup 和 onmouseover 很象鼠标悬停状态。对于何时使用事件代替伪类没有特别的规则，但是很显然，如果没有其他理由使用标签，就不应该使用伪类。

问：能否在文本之外的元素（如图象）上捕捉鼠标或者键盘事件？

答：可以，这些事件类型可以应用到与点击或者翻转图像相关的操作。但是，其他多媒体对象，如嵌入的 YouTube 视频或者 Flash 文件不能以相同的方式交互，因为这些对象通过适用于其他鼠标或者键盘操作的软件播放。例如，如果你单击嵌入到网页中的 YouTube 视频，你与 YouTube 播放器而不是实际网页交互——这种操作不能以相同的方式捕捉。

## 18.7 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 18.7.1 测验

1. 使用负数的 `top` 属性值会发生什么？
2. 当用户将鼠标移出元素边界的时候，哪个事件用于改变页面？
3. 如果你在样式单中看到 `cursor:crosshair`，你认为它的作用是什么？

### 18.7.2 测验答案

1. 使用负数的 `top` 值将内容放在父元素的左上角之上（而不是下面）。
2. 使用 `onmouseout` 事件。
3. 将用户的鼠标指针改变为大的加号（或者准星）。但是这么做的原因不明显，因为这是鼠标的一种典型表现。换句话说，用户将疑惑于你为什么在界面中这么做，他们不会欣赏你的设计决策。

## 18.8 练习

- 创建自己的具有翻转工具提示文本的网页，使用颜色和其他与你的显示模板匹配的样式。
- 在程序清单 18.2 中的 `onclick` 属性中添加命令，使得在同一时间只有一个 `<div>` 元素（海拔或者照片）可见。



## 第 19 章

# 创建固定或者流动的布局

本章中你将学到：

- 固定布局的工作方式
- 流动布局的工作方式
- 创建固定/流动混合布局的方法

本书的主体部分已经向你传授了设置 Web 内容样式的所有知识，从字体大小和颜色到图像、块元素、列表等。但是还没有讨论页面布局的高级概要。一般来说，有两种布局——固定和流动——但是也可以组合这两种布局，一些元素是固定的而其他元素是流动的。

在本章中，你将首先学习这两种布局的特性，并且看到一些使用这两种布局的网站示例。在本章结束时，你将了解组合这两种布局的基本模板。最终，布局的类型取决于你——只要网站遵循 HTML 和 CSS 标准，就不容易犯错误。

### 自己动手尝试一下：寻找喜欢的布局示例

寻找流动布局示例的一个好去处是 WordPress Theme Gallery (<http://wordpress.org/extend/themes/>)。WordPress 是一个博客平台，正在越来越多地被用作为非博客网站的管理工具。这个主题画廊展示了固定布局和流动布局的数百个示例，这些示例不仅给你的代码，也给所需要创建的内容带来思路。尽管你没有将 WordPress 博客作为本书中练习的一部分，但是在这个模板画廊中你可以看到并且与设计的许多变种交互。

花点时间看看 WordPress 和 CSS 禅意花园 (<http://www.csszengarden.com>) 上的示例，这将帮助你感受所喜欢的布局类型，而不会被布局中的内容所左右。

TRY IT YOURSELF ▼

知  
道  
PDFG



## 19.1 理解固定布局

固定布局（或称固定宽度布局）是页面主体被设置为特定宽度的布局。宽度一般由一个包含所有内容的主“包装器”（wrapper）<div>来控制。这个<div>的宽度属性将在样式属性中设置，如果<div>有一个 ID 值如“main”或者“wrapper”（这个名称取决于你），可以在一个样式单条目中设置。

创建一个固定宽度布局时，最重要的决策是确定容许的最小屏幕分辨率。在许多年中，800 像素×600 像素是设计人员的“最小通用基准”，结果是典型的固定宽度大约为 760 个像素。但是，2007 年以后，使用 800 像素×600 像素屏幕分辨率的用户已经少于 8%（现在已经下降到大约 4%）。因此，许多 Web 设计人员考虑 1024 像素×768 像素为当前最小屏幕分辨率，使得固定宽度的设计在 800 像素到 1000 像素之间。

### Watch Out!

**警告：**记住，Web 浏览器窗口包含不可查看区域，包括滚动条。所以如果你针对 1024 像素宽的屏幕分辨率，实际上不能使用所有 1024 像素。

创建固定宽度布局的主要原因是这样做你可以精确地控制内容区域的外观。但是，如果用户用比设计时所用的更小或者更大得多的屏幕分辨率访问固定宽度网站，他们可能会碰到滚动条（如果分辨率较小）或者大量的空白（如果分辨率较大）。

当前的 ESPN.com 主页提供了这个问题的一个示例，它的内容区域固定为 964 像素。如图 19.1 所示，浏览器窗口宽度为 800 像素。在图像的右边，重要的内容被切断了（在该图的底部，浏览器显示了一个水平滚动条）。

图 19.1

在较小的屏幕尺寸下的固定宽度示例



但是，图 19.2 所示为在浏览器窗口超过 1300 个像素宽时的网站外观：在主要内容两边都有大量的空白（或称“实业”）。

创建固定宽度布局还有另一个需要考虑的因素：将内容靠左或者居中放置。将内容靠左放置仅在右边造成额外的空白；居中的内容区域在两边都会造成额外的空白。



图 19.2  
在较大的屏幕尺寸下的固定宽度示例

## 19.2 理解流动布局

流动布局又称不固定布局——页面的主题不使用以像素表示的特定宽度，它可能包含在一个使用百分比宽度的主“包装器”<div>中。流动布局背后的思路是如果用户具有很小或者很宽的屏幕，布局仍然完全可用并且保持整体的设计美学。

图 19.3、图 19.4 和图 19.5 所示为 3 个流动布局的示例。

在图 19.3 中，浏览器窗口大约为 770 个像素宽。这个示例显示了不会出现水平滚动条的合理最小屏幕宽度。另一方面，图 19.4 显示了很小的浏览器窗口（545 个像素宽）。

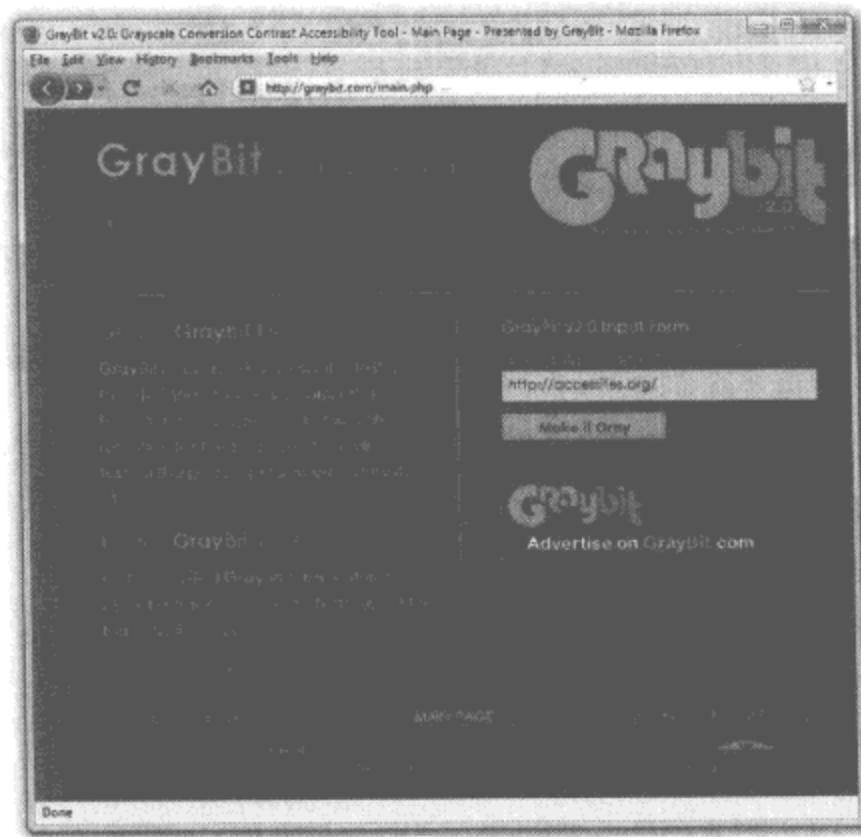


图 19.3  
在相对小的屏幕上查看流动布局

在图 19.4 中，你可以看到水平滚动条，在页面内容的标题区，标志图像开始覆盖于文本之上。但是页面的大部分仍然相当可用。页面左边的信息内容仍然容易辨认并且与右边的输

入表单分享可用空间。

图 19.4

在很小的屏幕上查看流动布局

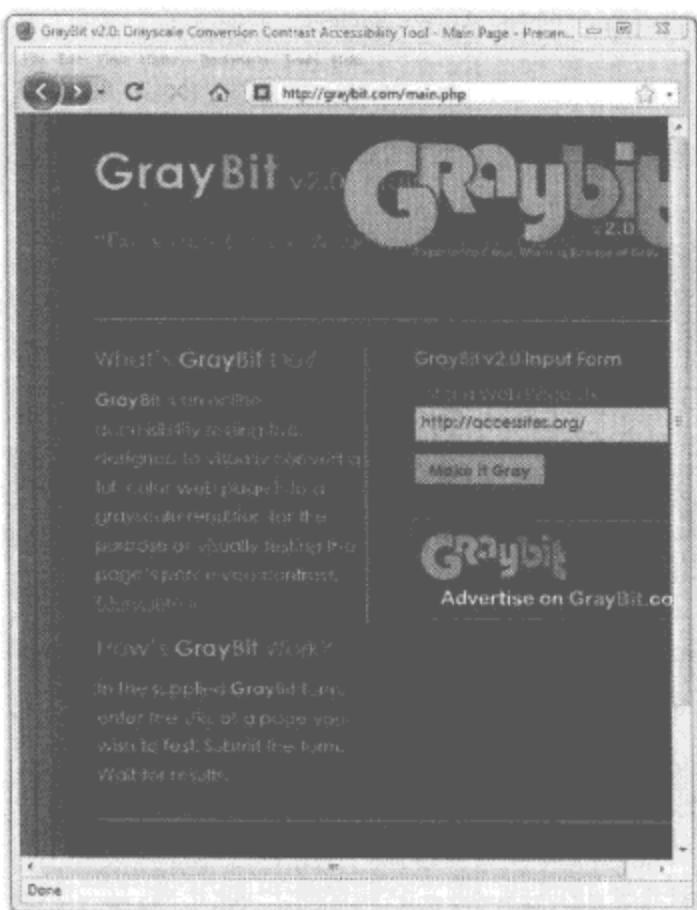
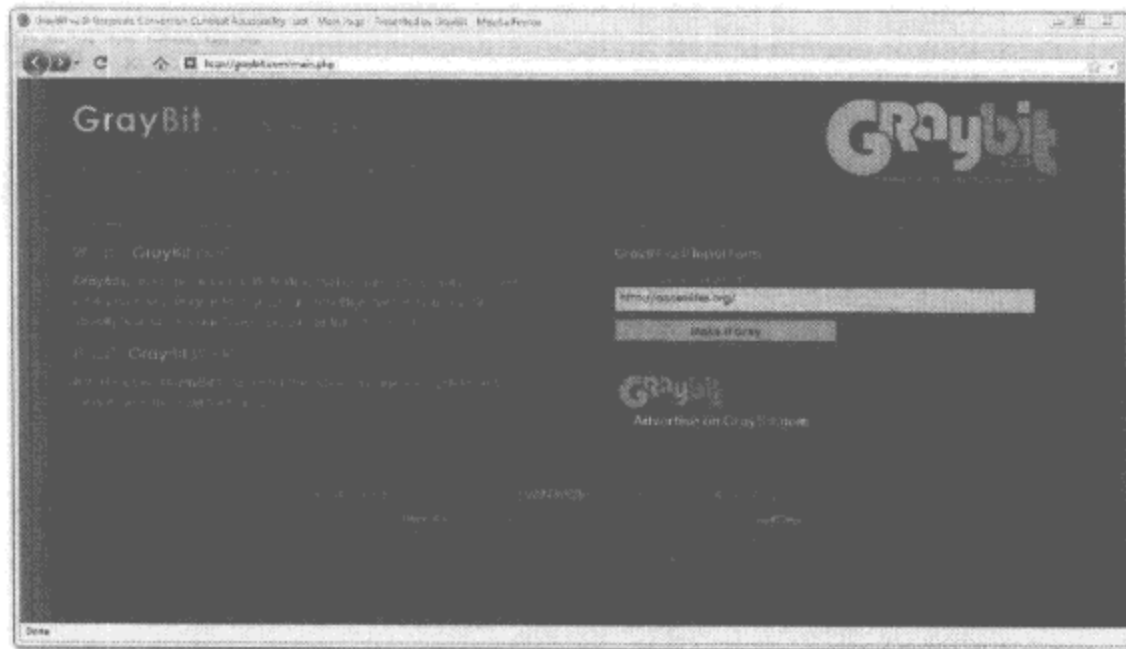


图 19.5 显示了在非常宽的屏幕中的相同页面。

图 19.5

在宽的屏幕上查看流动布局



在图 19.5 中，浏览器窗口大约 1330 像素宽。页面上的所有内容有很丰富的分布空间。这种流动布局因为所有设计元素指定了百分比宽度（代替固定宽度）而得以完成。在这一过程中，布局使用了所有的可用浏览器空间。

乍一看，流动布局方法似乎是最好的方法——毕竟，谁不想利用所有可用的屏幕空间？在利用空间和不允许内容“呼吸”之间仅仅一线之隔。太多的内容令人不知所措，太少的内容令人失望。

纯粹的流动布局可能相当吸引人，但是需要大量的测试来确保它可用于各种屏幕分辨率和各种浏览器。你可能没有时间和精力来完成这样的设计，在这种情况下，合理的妥协是固定/流动混合布局。

## 19.3 创建固定/流动混合布局

固定/流动混合布局包含两种布局元素。例如，你可以拥有一个流动布局，在正文区域或者锚元素（比如左边栏或者顶部导航栏）中包含固定宽度的内容区域。你甚至可以创建一个像第 13 章中介绍的框架一样工作的固定内容区域，固定的内容区域甚至在用户在内容中滚动时仍然保持固定。

### 19.3.1 从基本的布局结构开始

在这个示例中，你将学习创建一个流动的模板，但是这个布局中在正文区域的两侧有两个固定宽度的栏目（你可以将正文当作第三个栏目，只是这个栏目比其他两个宽得多）。该模板也有描述性的标题和页脚区域。程序清单 19.1 展示了这个布局的基本 HTML 结构。

程序清单 19.1 基本固定/流动混合布局结构

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Sample Layout</title>
    <link href="layout.css" rel="stylesheet" type="text/css" />
  </head>
  <body>
    <div id="header">HEADER</div>
    <div id="wrapper">
      <div id="content_area">CONTENT</div>
      <div id="left_side">LEFT SIDE</div>
      <div id="right_side">RIGHT SIDE</div>
    </div>
    <div id="footer">FOOTER</div>
  </body>
</html>
```

首先，注意这个布局的样式单用 `<link>` 标签链接而不是包含在模板中。因为模板用于多个页面，你希望能够以尽可能有组织的方式来控制模板元素的显示。这意味着你仅需要在一个地方——样式单中改变这些元素的定义。

接下来，你将注意到基本的 HTML 不过如此，极其简单。而且，说实话，这个基本的 HTML 结构可以用于固定布局，流动布局或者固定/流动混合布局，因为所有建立固定，流动或者混合布局的样式都存在于样式单中。

在程序清单 19.1 中，你实际拥有的 HTML 结构是希望包含在网站中的内容区域的标识。这个计划对任何开发都是关键，你甚至必须在考虑使用的布局类型之前就知道所希望包含的内容，更别提应用到布局的具体样式了。

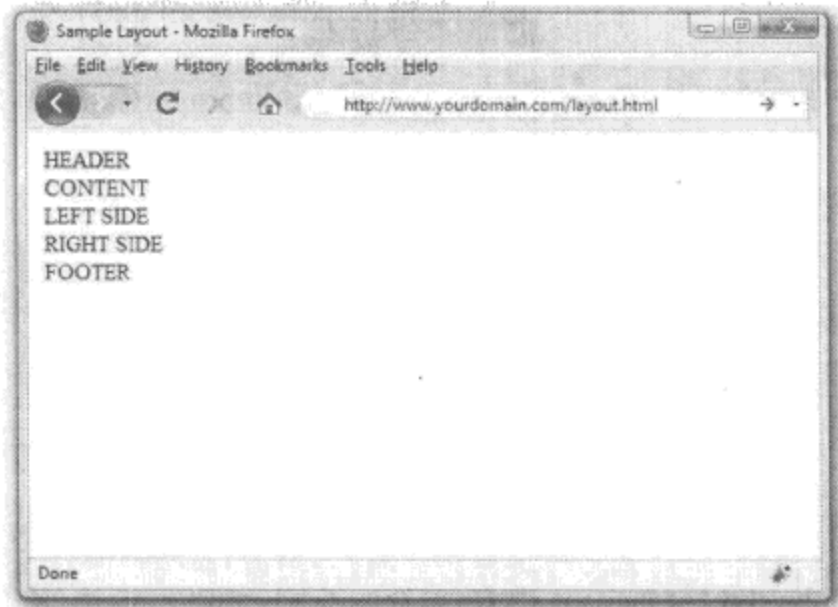
在这个阶段，`layout.css` 文件仅包含如下条目：

```
body {  
    margin:0;  
    padding:0;  
}
```

如果你看了程序清单 19.1 中的 HTML 并且对自己说“可是这些<div>元素只是互相堆叠而没有任何样式”，那你说对了。如图 19.6 所示，没有任何布局可言。

图 19.6

没有为<div>元素应用任何样式的基本 HTML 模板



### 19.3.2 在固定/流动混合布局中定义两个栏目

我们首先从最简单的事情开始。因为这个布局应该是流动的，我们知道放在标题和页脚区域的任何内容将扩展到浏览器窗口的宽度，而不管窗口宽窄。

添加如下代码到样式单，让标题和页脚区域宽度都为 100%，背景颜色相同：

```
#header {  
    float: left;  
    width: 100%;  
    background-color: #7152F4;  
}  
#footer {  
    float: left;  
    width: 100%;  
    background-color: #7152F4;  
}
```

现在工作越来越难了。我们必须在中间的栏目之外，在页面两侧定义两个固定栏目。在 HTML 中，注意有一个称作“包装器”的包含所有 3 个栏目的<div>。这个元素定义如下：

```
#wrapper {  
    float: left;  
    padding-left: 200px;  
    padding-right: 125px;  
}
```

这两个填充定义的用处实际上是为页面左右的两个固定宽度栏目保留空间。左边的栏目将为 200 个像素宽，右边的栏目将为 125 个像素宽，每个栏目都有不同的背景颜色。但是我们还必须相对于项目在 HTML 未设置样式时所在的位置（见图 19.6）进行定位。也就是在样式单中每个栏目所用的条目中添加 `position: relative`。此外，我们指出这些<div>元素应该向左浮动。

但是在 `left_side <div>` 中，我们还指出希望最右边的空白边缘与内容的边缘有 200 个像素的距离（这在定义为 200 个像素宽的栏目之外）。我们还希望左边的边距为完全的负数边距，这将使栏目就位（你很快会看到）。`right_side <div>` 没有包含 `right` 值，但是包含一个右边的负数边距：

```
#left_side {
  position: relative;
  float: left;
  width: 200px;
  background-color: #52f471;
  right: 200px;
  margin-left: -100%;
}
#right_side {
  position: relative;
  float: left;
  width: 125px;
  background-color: #f452d5;
  margin-right: -125px;
}
```

现在让我们定义内容区域，使其具有白色背景，占据 100% 的可用区域，并且浮动到所在位置的左边：

```
#content_area {
  position: relative;
  float: left;
  background-color: #ffffff;
  width: 100%;
}
```

这时，基本的布局将如图 19.7 所示，区域已经得到了清晰的描述。

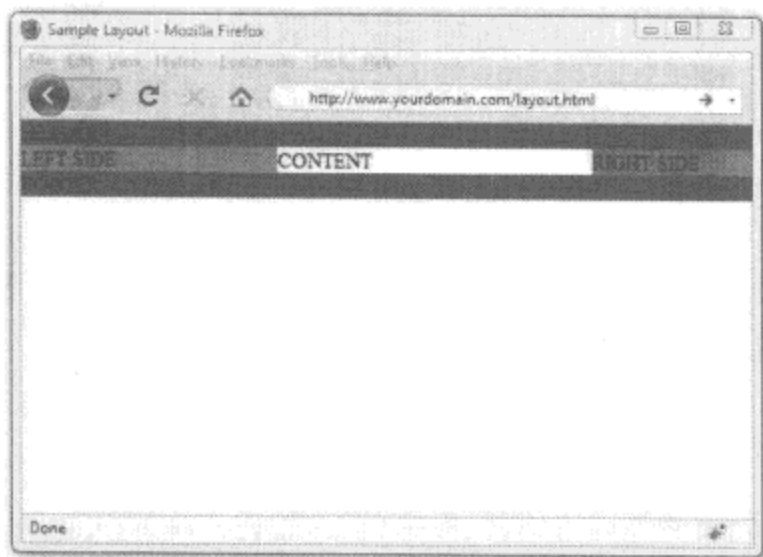


图 19.7

设置了一些样式之后的基本 HTML 模板

但是，如果窗口大小改变为某个宽度以下时，这个模板将出现问题。因为左栏目为 200 个像素宽，右栏目为 125 个像素宽，而你希望内容区域中至少有一些文本，可以想象，如果窗口仅为 350~400 个像素宽，这个页面将“破坏”。图 19.8 所示为窗口尺寸被改为 400 个像素以下（准确地说是 390 个像素）时的情况。

### 19.3.3 设置布局的最小宽度

虽然用户不太可能使用低于 400 个像素的浏览器窗口访问你的网站，但是这个示例的目



的在于说明本书的页面的局限。你可以广泛地类推和应用这一信息，即使在固定/流动混合网站中，除非你做些处理，否则存在一个极限点，在此布局将遭到破坏。

图 19.8

窗口大小在 400 个像素以下时的基本模板：糟糕！



需要做的处理是使用 `min-width` 属性。`min-width` 属性设置元素的最小宽度，这个宽度不包括填充、边框和边距。图 19.9 所示为 `<body>` 元素应用 `min-width` 属性的情况。

图 19.9

窗口大小在 400 个像素以下时的基本模板：好！

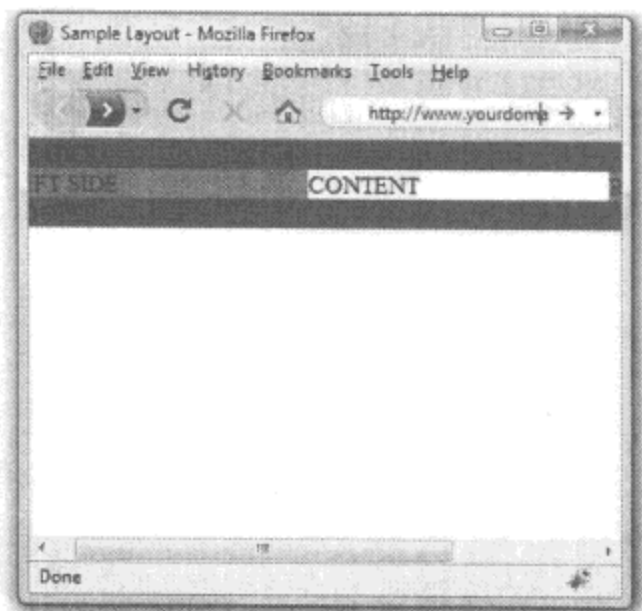


图 19.9 在向右滚动后显示出右栏目的一角，但是布局在尺寸低于最小宽度时并没有破坏。在这个例子中，最小的宽度为 525 个像素：

```
body {
  margin: 0;
  padding: 0;
  min-width: 525px;
}
```

在这个示例中出现了水平滚动条，因为浏览器窗口宽度小于 500 个像素。当窗口宽度稍大于 525 个像素时滚动条消失，而在浏览器窗口宽度大约为 875 个像素时它就完全不见了，如图 19.10 所示。

### 19.3.4 处理固定/流动混合布局中的栏目高度

这个示例一切都很好，只有一个问题：没有内容。为各个元素添加内容时，出现了更多的问题。如图 19.11 所示，栏目的高度变成所包含的内容所需的高度。

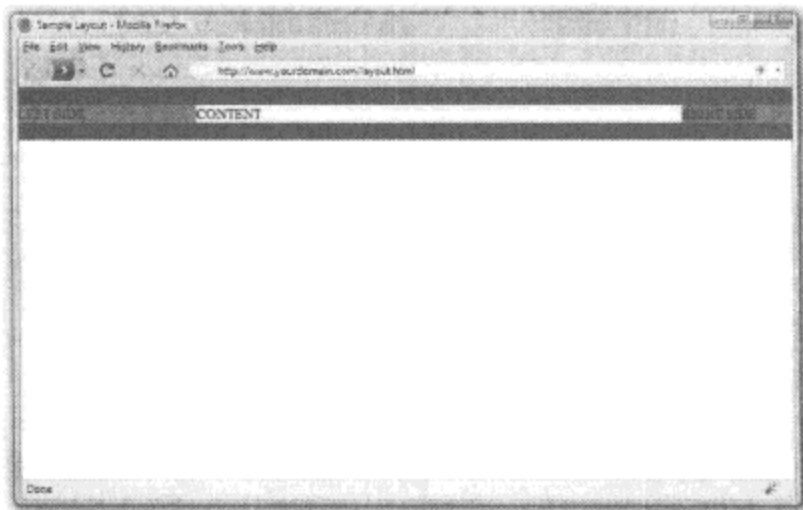


图 19.10  
在宽度大于800个像素的浏览器窗口中看到的基本HTML模板

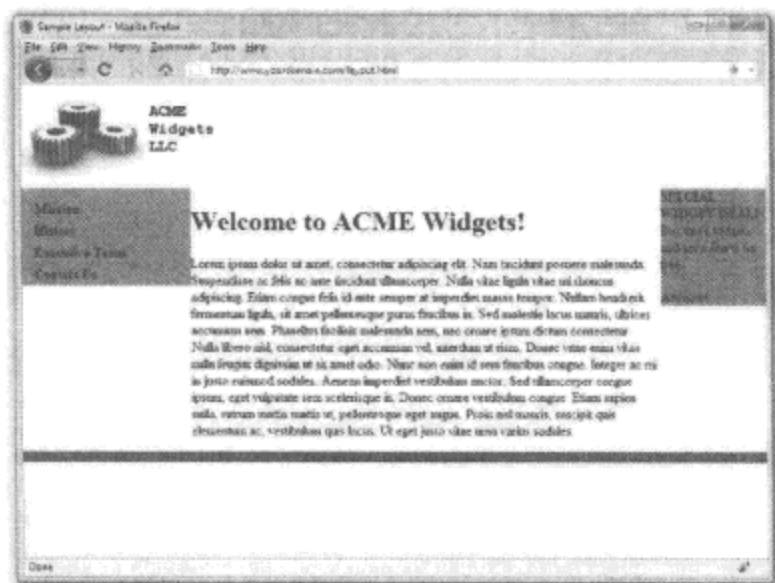


图 19.11  
栏目仅与内容等高

因为你无法指望用户的浏览器有一个确定的高度，或者内容长度始终相同，所以可能认为这是固定/流动布局的一个问题。事实并非如此，如果你稍微打破常规，就能够应用一些样式，将所有部分结合起来。

首先，在 `left_side`、`right_side` 和 `content_area` 的样式单条目中添加如下声明：

```
margin-bottom: -2000px;
padding-bottom: 2000px;
```

这些声明在所有 3 个元素的底部添加了荒谬的填充和大的边距值。你还必须为页脚 ID 添加 `position: relative`，使其在这个填充值下仍然可见。

这时，页面看上去如图 19.12 所示——仍然不是你想要的，但是更接近了。

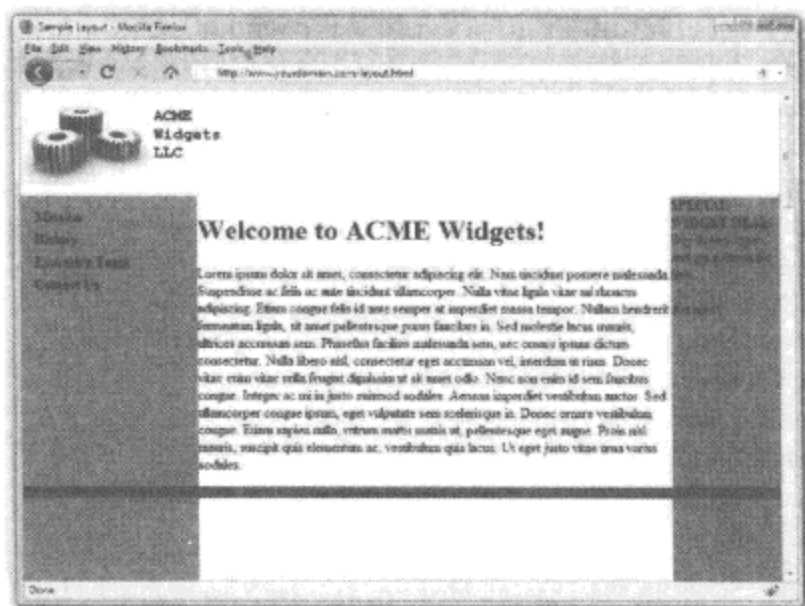


图 19.12  
彩色区域目前不管栏目中的内容为何都可见



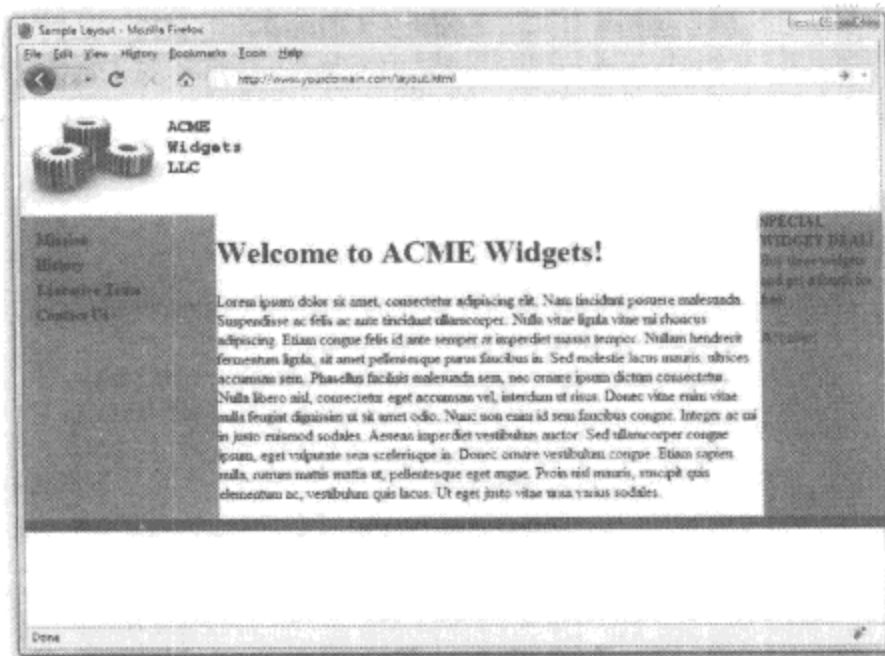
为了剪除多余的颜色，在样式单中添加如下用于包装器 ID 的样式：

```
overflow:hidden;
```

图 19.13 所示为最终的结果：一个具有必要的栏目空间的固定宽度/流动混合布局。

图 19.13

恭喜！这是一个固定宽度/流动混合布局



完整的 HTML 代码见程序清单 19.2，最终的样式单如程序清单 19.3 所示。

#### 程序清单 19.2 基本固定宽度/流动混合布局结构（包含内容）

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Sample Layout</title>
    <link href="layout.css" rel="stylesheet" type="text/css" />
  </head>

  <body>
    <div id="header"></div>
    <div id="wrapper">
      <div id="content_area">
        <h1>Welcome to ACME Widgets!</h1>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.
          Nam tincidunt posuere malesuada. Suspendisse ac felis ac ante
          tincidunt ullamcorper. Nulla vitae ligula vitae mi rhoncus
          adipiscing. Etiam congue felis id ante semper at imperdiet
          massa tempor. Nullam hendrerit fermentum ligula, sit amet
          pellentesque purus faucibus in. Sed molestie lacus mauris,
          ultrices accumsan sem. Phasellus facilisis malesuada sem, nec
          ornare ipsum dictum consectetur. Nulla libero nisl,
          consectetur eget accumsan vel, interdum ut risus. Donec
          vitae enim vitae nulla feugiat dignissim ut sit amet odio.
          Nunc non enim id sem faucibus congue. Integer ac mi in justo
          euismod sodales. Aenean imperdiet vestibulum auctor. Sed
          ullamcorper congue ipsum, eget vulputate sem scelerisque in.
          Donec ornare vestibulum congue. Etiam sapien nulla, rutrum
          mattis mattis ut, pellentesque eget augue. Proin nisl mauris,
          suscipit quis elementum ac, vestibulum quis lacus. Ut eget
          justo vitae urna varius sodales. </p>
        </div>
      </div>
    </div>
  </body>
</html>
```

```
</div>
<div id="left_side">
  <ul>
    <li><a href="#">Mission</a></li>
    <li><a href="#">History</a></li>
    <li><a href="#">Executive Team</a></li>
    <li><a href="#">Contact Us</a></li>
  </ul>
</div>
<div id="right_side"><strong>SPECIAL WIDGET DEAL!</strong><br/>
  Buy three widgets and get a fourth for free.<br/><br/>
  Act now!
</div>
</div>
<div id="footer"> Copyright information usually goes here.</div>
</body>
</html>
```

---

### 程序清单 19.3 用于固定宽度/流动混合布局的完整样式单

---

```
body {
  margin:0;
  padding:0;
  min-width: 525px;
}
#header {
  float: left;
  width:100%;
  background-color: #ffffff;
}
#footer {
  float: left;
  width:100%;
  background-color: #7152f4;
  font-size: 8pt;
  font-weight: bold;
  text-align: center;
  position: relative;
}
#wrapper {
  float: left;
  padding-left: 200px;
  padding-right: 125px;
  overflow: hidden;
}
#left_side {
  position: relative;
  float: left;
  width: 200px;
  background-color: #52f471;
  right: 200px;
  margin-left: -100%;
  padding-bottom: 2000px;
  margin-bottom: -2000px;
}
#right_side {
  position: relative;
  float: left;
  width: 125px;
  background-color: #f452d5;
```



```
margin-right: -125px;
padding-bottom: 2000px;
margin-bottom: -2000px;
}
#content_area {
position: relative;
float: left;
background-color: #ffffff;
width: 100%;
padding-bottom: 2000px;
margin-bottom: -2000px;
}
#left_side ul {
list-style: none;
margin: 12px 0px 0px 12px;
padding: 0px;
}
#left_side li a:link, #nav li a:visited {
font-size: 12pt;
font-weight: bold;
padding: 3px 0px 3px 3px;
color: #000000;
text-decoration: none;
display: block;
}
#left_side li a:hover, #nav li a:active {
font-size: 12pt;
font-weight: bold;
padding: 3px 0px 3px 3px;
color: #ffffff;
text-decoration: none;
display: block;
}
```

## 19.4 总结

在本章中，你看到了 3 种主要布局类型（固定、流动和固定/流动混合）的一些实用的示例。在本章的第三节，你看到了一个扩展的示例，使你经历了一步一步创建完全通过 HTML 和 CSS 验证的一个固定/流动混合布局的过程。记住，创建布局的最重要部分是将你认为在设计中应该考虑的内容分组。

## 19.5 问与答

**问：**我听说过弹性布局，这与流动布局有何不同？

**答：**弹性布局的内容区域尺寸在用户改变文本大小时变化。弹性布局使用 `ems`，这个单位与文本和字体大小关联。一个 `em` 是与当前字体以点（`point`）为单位的大小相同的排版单位。在弹性布局中使用 `ems` 时，如果用户使用 `Ctrl` 键和鼠标滚轮增大或者减小文字，包含文本的区域按照比例增大或者减小。弹性布局很难达到，更常用于像集而不是实用的网站，因为完善它们要花费很长的时间。

**问：**你已经花费了很多时间来谈论流动或者混合布局——它们比纯粹固定的布局更好吗？

答：“更好”是个最主观的词，本书关注的是与标准兼容的代码。大部分设计人员会告诉你流动布局的创建（和完善）要花费更多的时间，但是对可用性的改进值得花费这些时间。那么什么时候这些花费不值得呢？如果你的客户没有意见，他们的付费是统一付费而不是按小时付费时。在那种情况下，你这么做仅为了展示你的技能——但是这对你可能是值得的。

## 19.6 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 19.6.1 测验

1. 一般来说，哪种布局最好：固定、流动还是混合？
2. 你能够在页面上的任何地方放置固定布局吗？
3. `min-width` 的功能是什么？

### 19.6.2 测验答案

1. 这是很难回答的问题；没有“最佳”的布局。这取决于你的内容和目标用户群的要求。
2. 当然。尽管大部分固定布局是靠左或者居中的，但是你可以在 `xy` 轴上为包含所有其他布局 `<div>` 的 `<div>` 定位。
3. `min-width` 属性设置元素的最小宽度，不包括填充、边框或者边距。

## 19.7 练习

- 图 19.13 展示了“完成”的固定/流动混合布局，但是要注意有几个地方可以改进：右栏目的文本周围没有任何空间，正文和栏目之间没有间距，页脚栏有些稀疏，等等。花一点时间修复这些设计元素。
- 在第一个练习中添加合适的边距和填充之后，根据第 17 章中所学到的知识，为网页添加水平导航条和更有趣的垂直导航。





## 第 20 章

# 创建打印友好的网页

本章中你将学到：

- 使页面打印友好的因素
- 应用媒体专用的样式单的方法
- 为打印页面创建样式单的方法
- 在打印预览模式中查看网页的方法

如果使用过在线地图工具，如 MapQuest 或 Google Maps，肯定有过需要打印网页的经历。同样，只在网上提供的许多优惠券、在线销售商的产品订购发票以及基于 Web 的航班登记，还有从你的家用计算机上打印登记牌都增加了打印友好页面的需求。网页并不只是完全为在屏幕上浏览而设计的。你可能没有认识到这一点，但是有时候是可能要为需要打印网页脱机阅读的用户特别设计并提供打印友好（print-friendly）的网页版本的。实际上，CSS 使得创建可根据浏览方式不同而改变外观的网页非常简单。本章将介绍如何创建这样的网页。

### TRY IT YOURSELF ▼

**自己动手尝试一下：查看你的内容的打印友好性**

在本章的学习过程中，考虑自己的网页打印出来是否好看。然后考虑如何修改这些网页，使其打印出来的效果更好。下面是一些要考虑的因素。

1. 即使在前面的章节中已经做出了警告，网页是否仍然使用了与文本反差强烈的背景图像或背景颜色？这种网页可能因为背景而难以打印，因此可考虑在打印版本中不使用背景图像或者颜色并只使用黑色文本。准备打印的页面时，尽可能坚持使用白色背景上的黑色文本。
2. 网页是否包含许多链接？如果是，可考虑在打印时改变链接的外观，使链接不那么突出。别忘了，在纸上不能单击链接！
3. 最后，网页中的图像是否都是必要的？色彩丰富的图像在大多数打印机上都很费墨，因此可考虑在打印时删除一些图像。

## 20.1 如何成为打印友好的网页

在前面的“自己动手试一试”部分中简要地介绍过该主题，但还需要探讨如何使网页易于打印。首先，有些网页已经是打印友好的。如果网页使用白色背景和黑色文本，包含的图像很少，就无需考虑设计打印友好的版本。然而，如果网页的背景为黑色，包含动态链接和许多图像，这样的网页对大多数打印机来说是不友好的。

考虑使网页更易于打印时，要记住的主要一点是介质的限制。换句话说，什么使打印的网页与显示在计算机屏幕上的网页不同？一个明显的区别是大小——打印的网页大小是固定的，通常是 8.5 英寸×11 英寸，而计算机屏幕有多种大小。除大小外，打印的网页也有颜色限制，即使在彩色打印机上也如此。当用户只对网页的文本感兴趣时，很少用户会浪费油墨来打印彩色背景。

大多数用户对网页中的主要内容文本外的内容不感兴趣。例如，如图 20.1 所示，这是从加州的圣何塞到华盛顿州的旅游路线图。

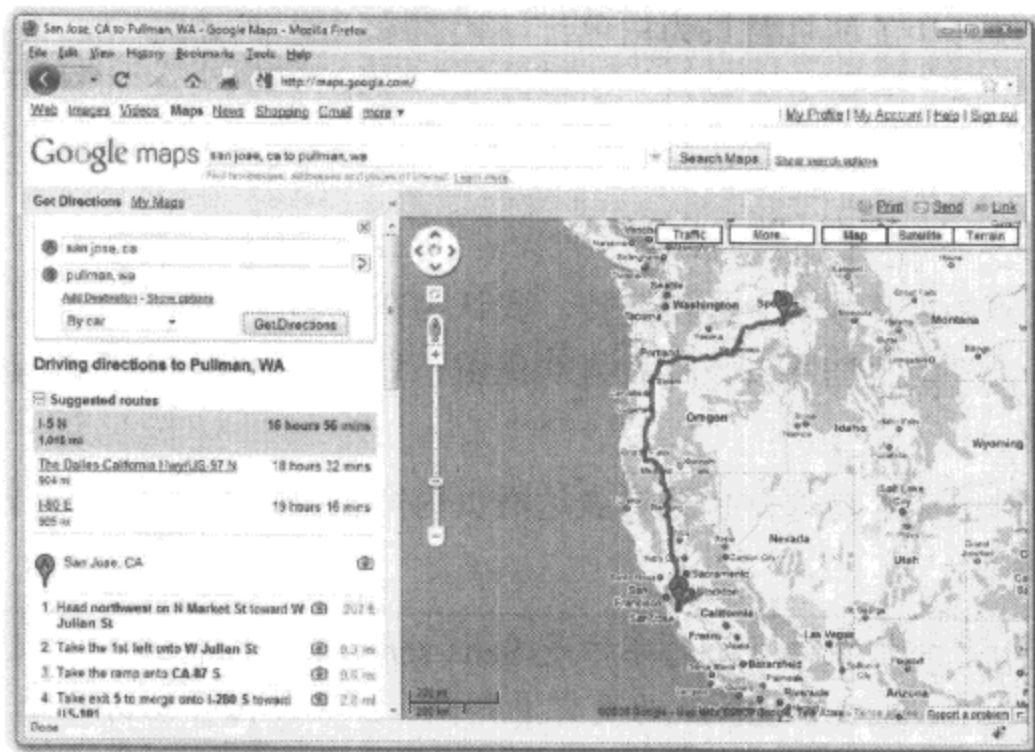


图 20.1

这个页面不是非常打印友好的，因为表单输入和大的图像以及自身的显示控制

图 20.1 中所示的页面包含了表单输入字段，一个可以控制（移动、缩放）的大图像，以及其他在 Web 内容中经常遇到的其他辅助项目。在地图之上显示一组你能进行的操作，其中之一是打印页面的链接。这时，你可能觉得奇怪，为什么不能单击浏览器中的打印按钮？你确实可以这么做，但是这样打印的页面和你在屏幕上看到的一样——包括了表单输入字段和图像元素，而你希望知道的只是驾车经过这条路线时所应该转弯的地方。

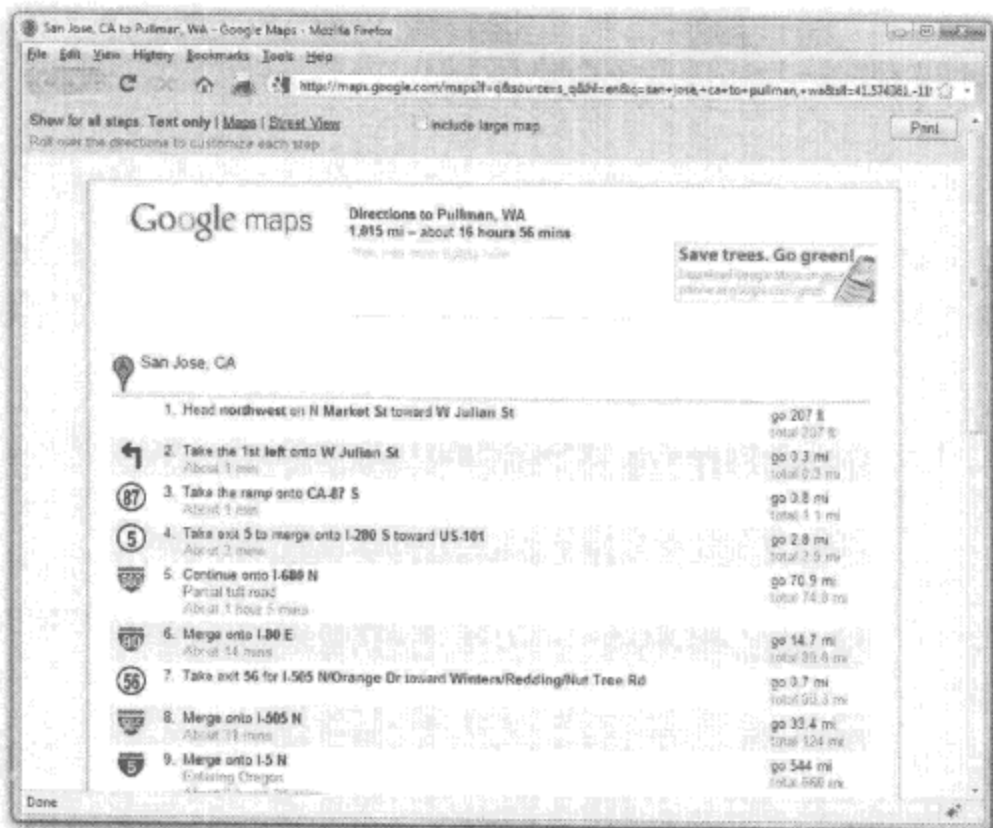
如果你单击页面正文中的打印链接，你的浏览器将显示一个页面，如图 20.2 所示，Google 将这个页面格式化为专用于打印。

从上图可以看出，打印友好的版本对原来的网页有很大改进，至少从打印机的角度看是这样的，所有表单输入和图像都被删除。

为让读者理解创建打印友好的网页时需要特别考虑的因素，下面列出了至少应该考虑的一些修改。

图 20.2

页面的打印友好版本将驾车指南独立出来，这样便于打印



- 删除页面背景，使打印的网页背景为白色。
- 将文本颜色改为黑色，文本可有颜色，但最好使用黑色。
- 保证字体足够大，使打印出来的网页易于阅读——可能需要针对不同的字体大小进行测试。
- 删除链接格式或转换为基本的下划线链接。我认为在打印的网页中保留特殊的链接外观真的没有意义，但有些网页设计者喜欢保留下划线，这样用户能够知道原来的网页中包含链接。
- 删除所有不必要的图像。这通常包括对表达网页内容不重要的图像，如导航按钮、大多数广告和动画。

### By the Way

**注意：**如果面向打印的页面的字体是 Sans-serif，一些 Web 设计人员建议将字体改为 serif，这种字体被认为更容易辨认。如果你在自己的页面上使用了 Sans-serif 字体，是否在打印网页时保持核心的外观取决于你——也就是说你不用将字体改为 Serif。

除这些建议外，在标题下添加一行关于网页作者、网页 URL 和网页版权等信息也可能有用。这些是用户离开网站、只留下网页的打印版本时可能失去的信息。

你可能不需要现在就将所有这些修改都应用于网页。这里只是说明如何使网页打印友好，这样在创建面向打印机的样式单时就可以做得更好。可创建只在打印网页时才应用的样式单。下一节将介绍如何创建这样的样式单。

## 20.2 应用与媒体相关的样式单

图 20.1 包含了一个小型打印机图标，让用户能够浏览网页的打印友好版本。这种图标在许多新闻网站上很流行，这是一个重要特性，如果没有它，用户打印网页时将很麻烦，而且

会在伴随文章的图像和广告上浪费纸墨。虽然打印机图标是一种直观而有效的方法，但是还有一种选择不需要这种特殊的链接。

我说的是使用面向打印的样式单，当用户选择打印网页时，该样式单就自动应用于网页。CSS 支持“与介质相关的样式单”的概念，这是指面向具体介质（如屏幕或打印机）的样式单。然而，CSS 不仅限于支持这两种介质，下面是 CSS2 支持使用独特样式单的介质列表。

- **all**: 所有设备。
- **aural**: 用于语音合成（在 CSS1 中称为 **speech**）。
- **braille**: 用于盲文设备。
- **embossed**: 用于页式盲文打印机。
- **handheld**: 用于屏幕大小和带宽受限的手持设备。
- **print**: 用于打印预览模式中在屏幕上查看的打印资料和文档。
- **projection**: 用于投影机。
- **screen**: 用于彩色计算机屏幕。
- **tty**: 用于使用固定宽度字符设备（如终端、电传打字机或者显示能力有限的手持设备）。
- **tv**: 用于电视设备，这种设备一般分辨率较低，颜色较少，并且只有有限的滚动能力。

这些介质当中最有趣的可能是声音，它使网页能够被朗读和聆听。显然，CSS 的架构师预见到的 Web 比我们现在想到的还要广泛，因为现在我们设计的网页主要是为了在计算机屏幕上显示。虽然你可能不需要考虑太多声音网页设计，但这可能成为将来发展方向。

关于应用于其他介质的样式单的好消息是，你不需要学习新东西。可能在设计声音网页时需要学习一些新技巧，但现在你可以使用学习过的样式属性来创建面向打印的样式单。真正的技巧在于知道如何将样式单应用于特定的介质。

前面提到，`<link />` 标签用于将样式单链接到网页。这个标签支持属性 **media**，可使用它来指定样式单将应用的介质。默认情况下，这个属性设置为 **all**，即如果没有专门指定，外部样式单将用于所有介质。其他可接受的值对应于前面列出的介质。

为网页指定面向打印的样式单，需要使用两个 `<link />` 标签，一个用于打印机，一个用于其他所有介质。下面是执行这项任务的代码：

```
<link rel="stylesheet" type="text/css" href="standard.css" media="all" />
<link rel="stylesheet" type="text/css" href="for_print.css" media="print" />
```

在这个例子中，网页链接了两个样式单。第一个样式单将 **media** 属性设置为 **all**，因此面向所有介质。如果没有其他设置，**standard.css** 样式单将应用于所有介质。然而，由于有了第二个样式单，**for\_print.css** 样式单将用于打印网页。

**注意：**也可以使用 `@import` 命令来链接面向介质的样式单。例如，下列代码的功能与上面的 `<link />` 代码相同：

```
@import url(player.css) all;
@import url(player_print.css) print;
```

**By the Way**

**注意:**可在一个<link>标签中指定多种介质类型,彼此用逗号分开,如下所示:

```
<link rel="stylesheet" type="text/css" href="for_pp.css" media="print, projector" />
```

这段代码使 for\_pp.css 样式单只应用于 print 和 projector 这两种介质类型。

**Watch Out!**

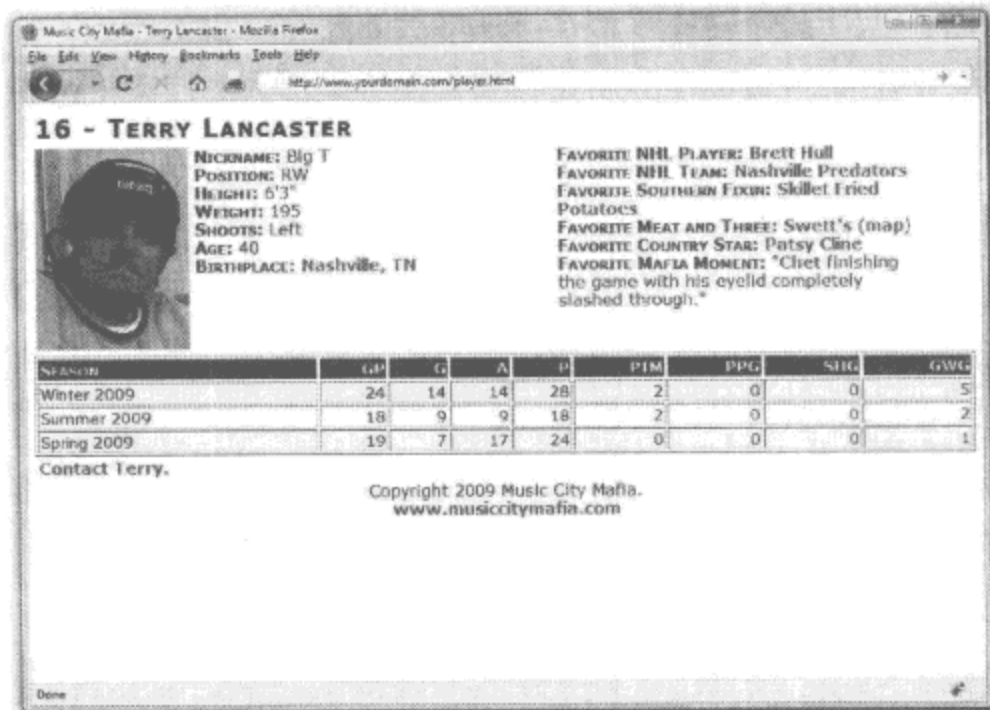
**警告:**你可能想在前一段代码中第一个链接的样式单中指定 media="screen"。虽然在普通网页浏览器中浏览网页时这样也可行,然而,如果用户使用手持浏览器或其他类型的介质,这可能引起问题。换句话说,样式单只能应用于 media 属性中指定的介质类型,不能应用于其他介质类型。

## 20.3 设计用于打印页面的样式单

本章前面讲过创建打印友好的网页所需的修改,现在就来创建打印友好的样式单。先来看一下使用普通(屏幕)样式单的网页,如图 20.3 所示。

图 20.3

在普通 Web 浏览器中查看使用 CSS 样式的网页

**By the Way**

**注意:**即使没有链接到外部样式单,你也可以为样式单指定媒体类型。<style> 标签也使用和<link/>标签相同的 media 属性。

该图显示了网页在普通网页浏览器中显示的结果。实际上,这个网页还不是打印友好的,可再进行一些改进。

下面的修改可使该网页打印出来的效果更好。

- 将所有文本的颜色改为黑色。
- 删除链接的格式(粗体及其颜色)。
- 将队员的两部分信息垂直排列,因为这些信息在打印的网页上可能水平排列不下。
- 删除整个联系人链接。

前两个对常规样式单的修改很简单,主要是修改或删除已有的样式。然而,第三个修改就需要思考一下。因为打印的网页大小是固定的,因此在打印的网页上所有元素都可以使用

绝对定位，这比较容易将内容区域放到希望的位置。最后，列表中最后一条很容易实现，只要将 `contact` 元素的 `display` 样式属性设置为 `none`。

**警告：**虽然绝对定位在这个曲棍球队员网页中可行，但在指定面向打印的网页的样式时它并不总是可行。具体地说，如果网页的内容打印出来不只一页，最好使用相对定位，让内容分为多页。

**Watch  
Out!**

程序清单 20.1 包含了样式单 `player_print.css` 的 CSS 代码，它包含上述修改，形成一个特别适于打印曲棍球队员网页的样式单。

#### 程序清单 20.1 面向打印的曲棍球队员样式单的 CSS 代码

```
body {
  font-family:Verdana, Arial;
  font-size:12pt;
  color:black;
}

div {
  padding:3px;
}

div.title {
  font-size:18pt;
  font-weight:bold;
  font-variant:small-caps;
  letter-spacing:2px;
  position:absolute;
  left:0in;
  top:0in;
}

div.image {
  position:absolute;
  left:0in;
  top:0.5in;
}

div.info {
  position:absolute;
  left:1.75in;
  top:0.5in;
}

div.favorites {
  position:absolute;
  left:1.75in;
  top:2in;
}

div.footer {
  position:absolute;
  text-align:left;
  left:0in;
  top:9in;
}

table.stats {
  width:100%;
```





```
text-align:right;
font-size:11pt;
position:absolute;
left:0in;
top:3.75in;
}

div.contact {
display:none;
}

.label {
font-weight:bold;
font-variant:small-caps;
}

tr.heading {
font-variant:small-caps;
background-color:black;
color:white;
}

tr.light {
background-color:white;
}

tr.dark {
background-color:#EEEEEE;
}

th.season, td.season {
text-align:left;
}

a, a:link, a:visited {
color:black;
font-weight:normal;
text-decoration:none;
}
```

这段代码中最绝妙的地方在于，所有绝对定位代码都以英寸（in）为度量单位。考虑到打印的网页大小以英寸为单位，而不是以像素为单位，因此这是合理的。仔细研究这段代码，可以发现所有文本都是黑色的，链接的特殊格式都已经删除，内容区域现在是绝对定位的，这样可以使内容准确地显示在希望的位置。

## 20.4 预览网页的打印结果

图 20.4 是打印友好的曲棍球队员网页版本在 Internet Explorer 的“打印预览”窗口中显示的样子。

如果图 20.4 已经显示了整个页面——11 英寸的高度，那么你将会注意到——页面的打印友好版本在最底部包含了脚注，如图 20.5 所示。

图 20.5 说明了在实际情况下如何使用打印友好的网页。该图在 Adobe Acrobat Reader 中显示同一曲棍球队员网页的 PDF 文档。

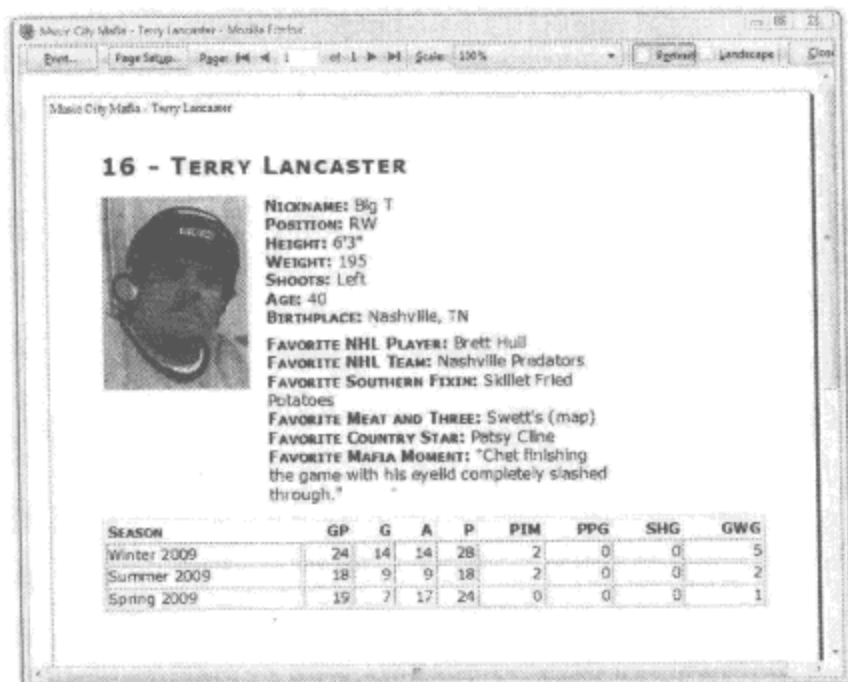


图 20.4

可用网页浏览器的“打印预览”来查看打印友好的网页版本

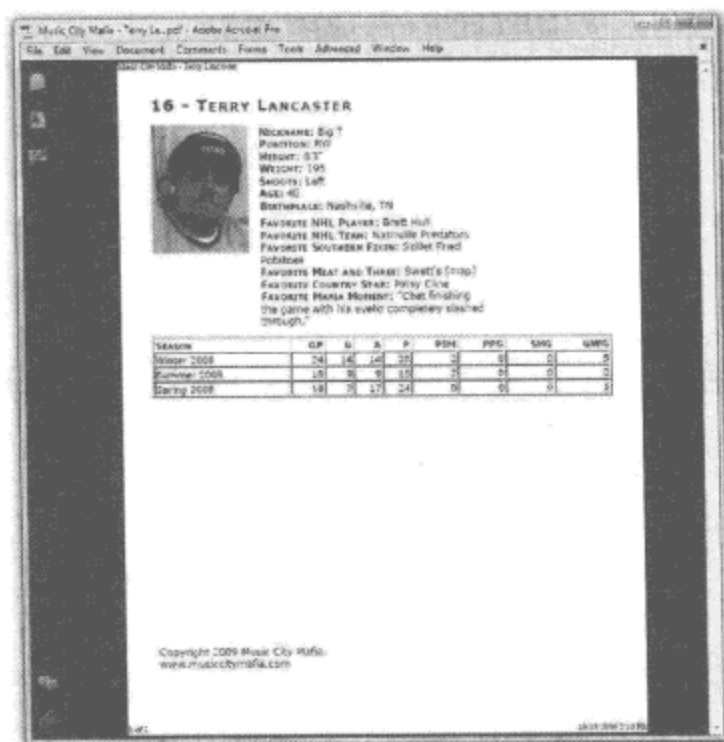


图 20.5

通过打印为 Adobe PDF，曲棍球队员页面被转换为 PDF 文档

Adobe 的虚拟打印机可以用于将曲棍球队员网页“打印”到 PDF 文档。你还可以找到比 Adobe Acrobat 软件更便宜的 PDF 转换程序如 DoPDF (<http://www.dopdf.com/>)。打印到 PDF 有效地创建打印友好网页的一个版本，这个版本的格式容易以电子形式共享打印。

**注意：** Acrobat 的更多内容可访问 <http://www.adobe.com/products/acrobat/>。

**By the Way**

## 20.5 总结

本章关注于 CSS 的一种实际应用，即解决一种常见问题：打印网页。你在本章首先学习了什么是打印友好的网页。在这里，你了解了内置于 CSS 的一种机制，即允许页面区分显示它的介质，然后选择对应的样式单。最后，你创建了一个面向打印的样式单，它指定网页打印时的样式。虽然大部分用户更喜欢在较大的计算机屏幕上浏览网页，而不是在纸上，但有时打印网页是必须的。通过提供易于打印的网页，可为网页访问者提供最

大的灵活性。

## 20.6 问与答

问：可以使用<link />标签的 media 属性创建一个专门用于在手持设备中浏览网页的样式单吗？

答：可以。将<link />标签的 media 属性设置为 handheld，可以将样式单设为专门针对手持设备。所有手机网站最终都可能通过这种方法来提供手机网页，而不是使用专门的标记语言，如无线标记语言（wireless markup language, WML）。

问：我还需要在网页上提供一个打印机图标，让用户能够打印网页吗？

答：不需要。在本章学习的链接样式单方法让你能够不通过任何特别的网页链接来支持打印友好的网页。然而，如果让用户能够在浏览器中看到打印友好的网页版本，可以链接到另一个版本的网页，它使用打印专用的样式单作为主要（浏览器）样式单。也可以在网页中提供一些说明（fine print），提示用户使用浏览器的打印预览功能来查看打印友好的网页版本。

## 20.7 作业

作业包含一些测试题及练习，帮助你巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 20.7.1 测验

1. 有一个指向打印友好页面的按钮是否就意味着该页面确实是打印友好的？
2. 如果外部样式单被链接到一个没有指定任何 media 属性的网页，结果是什么？
3. 如何链接一个名为 freestyle.css 的样式单到网页，使得仅在电视机上显示网页时才应用该样式单？

### 20.7.2 测验答案

1. 不——你仍然必须链接到一个将特殊的样式单应用到内容的页面，这样它才能显示为“打印友好”的。
2. Media 属性的默认值是 all，这意味着该样式单将针对所有介质。
3. 代码如下：

```
<link rel="stylesheet" type="text/css" href="freestyle.css" media="tv" />
```

## 20.8 练习

- 选择一个有许多颜色和图像的网页，为它创建一个打印友好的样式单。务必在网页

中添加一个<link />标签，它链接到面向打印的样式单。

- 使用一个<link />标签，将其 **media** 属性设置为 **handheld** 值，为网页创建面向手持设备的版本。涉及的概念与创建面向打印的页面相同，但需要应对的是很小的屏幕，而不是打印纸。可以发布页面，然后使用手机或手持计算机浏览器打开它，对该页面进行测试。



## 第 21 章

# 理解动态网站

---

本章中你将学到：

- 不同类型的动态内容的概念
- 在 HTML 中包含 JavaScript 的方法
- 用 JavaScript 显示随机文本的方法
- 使用 JavaScript 和用户事件改变图像的方法

动态这个术语的意思是活动的事物或者激发其他事物活动的事物。

谈到网站，动态的网站将交互性集合到功能和设计中，而且促使用户采取行动——进一步阅读、购买产品等。在本章中，你将学习一些建立动态网站的不同类型的交互，包括服务器端和客户端脚本（还有一些后者的实用示例）。

我在本书的其他地方已经提及了客户端脚本，在第 18 章中你已经使用事件属性和 JavaScript 来改变特定元素的样式——这被称为操纵文档对象模型（DOM）。你将在本章中再进行一些这种类型的操纵。具体地说，在学习不同的技术之后，你将使用 JavaScript 在页面装入时显示一些随机的格言并且根据用户的交互更换图像。

### 21.1 理解不同类型的脚本

在 Web 开发中，有两种脚本类型：服务器端和客户端。两种类型的脚本——实际上是计算机编程——都超出了本书的范围。但是，超出的不太多。Sams Teach Yourself 系列中的两本非常有用而流行的书籍是很自然的扩展：Sams Teach Yourself PHP, MySQL and Apache All-in-one（PHP, MySQL 和 Apache 入门经典，用于服务器端脚本）和 Sams Teach Yourself JavaScript in 24Hours（JavaScript 入门经典，用于客户端脚本）。

服务器端脚本指的是在 Web 服务器上运行，然后将结果发送给浏览器的脚本。如果你曾经在网站上提交过表单，包括使用搜索引擎，就曾经体验过服务器端脚本的结果。流行的服务器端脚本语言如下，更多的信息可以访问这里列出的网站。

- PHP (PHP: 超文本预处理器) ——<http://www.php.net/>
- JSP (Java 服务器页面) ——<http://java.sun.com/products/jsp/>
- ASP (活动服务器页面) ——<http://www.asp.net/>
- Perl ——<http://www.perl.org/>
- Python ——<http://www.python.org/>
- Ruby ——<http://www.ruby-lang.org/>

另一方面，客户端脚本指运行在你的浏览器中的脚本——这些脚本的运行不需要与 Web 服务器交互。最流行的客户端脚本语言是 JavaScript。多年来，研究显示 93% 的 Web 浏览器启用 JavaScript。

**注意：** 尽管名称为 JavaScript，但是不是面向对象编程语言 Java 的衍生物，也没有其他亲缘关系。Java 于 1995 年由 Sun Microsystems 公司发布，它与服务器端脚本语言 JSP 有很紧密的关系。JavaScript 也于 1995 年，由 Netscape Communications 公司发布，这个名称指出它与 Java 外表上的相似度，但是没有直接的关系。

**By the  
Way**

另一个客户端脚本语言是 Microsoft 的 VBScript (Visual Basic 脚本版本)。这种语言只可用于 Microsoft Internet Explorer 浏览器，因此，除非你非常确定所有网站用户都使用这个浏览器访问（比如在封闭的公司环境），否则不应该使用这种语言。为了得到最大的可能目标用户群，本章假定使用 JavaScript 进行客户端脚本编程，所有的代码示例都是 JavaScript。

## 21.2 在 HTML 中包含 JavaScript

JavaScript 代码可以存在于文件中的两个位置之一。

- 在以 .js 为扩展名的独立文件中。
- 直接存在于 HTML 文件中。

外部文件通常用于脚本库（你可以在许多页面中重用的），而直接出现在 HTML 文件中的代码是为了完成面向这些单独页面的特殊功能。不管 JavaScript 存在于哪里，浏览器通过 `<script></script>` 标签来了解它的存在。

当你把 JavaScript 保存在外部文件时，这样引用：

```
<script type="text/javascript" src="/path/to/script.js">
```

`<script></script>` 标签一般放置在 `<head></head>` 标签之间，因为它严格地说不属于页面正文中的内容。相反，`<script>` 标签建立一组 JavaScript 函数或者页面其余部分可使用的其他信息。但是，必要的时候，你也可以将 JavaScript 函数或者代码片断用 `<script>` 标签封装并且将它们放置在页面的任何地方。程序清单 21.1 显示了一个放在 HTML 文档正文部分的 JavaScript



片段示例。

### 程序清单 21.1 使用 JavaScript 打印一些文本

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>JavaScript Example</title>
  </head>

  <body>
    <h1>JavaScript Example</h1>
    <p>This text is HTML.</p>
    <script type="text/javascript">
      <!-- Hide the script from old browsers
      document.write('<p>This text comes from JavaScript.</p>');
      // Stop hiding the script -->
    </script>
  </body>
</html>
```

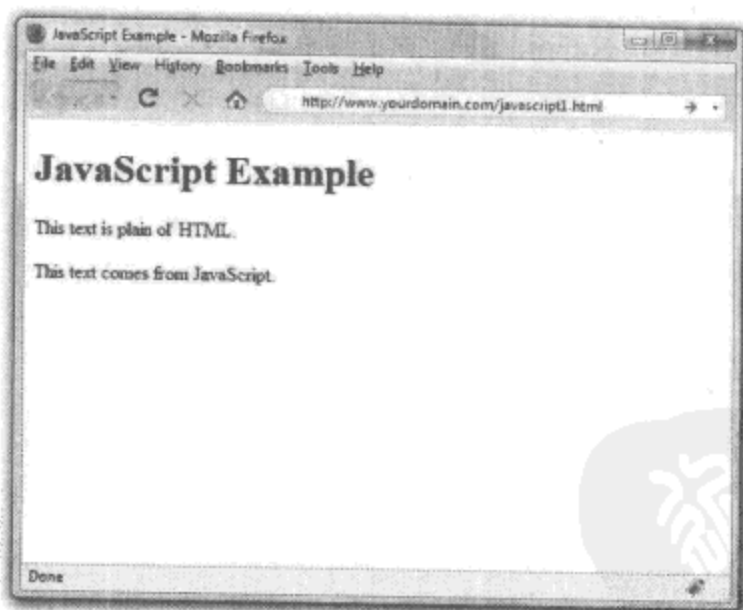
在<script></script>标签之间是一条输出如下 HTML 的 JavaScript 命令：

```
<p> This text comes from JavaScript.</p>
```

浏览器呈现这个 HTML 页面时，它看到<script></script>标签之间的 JavaScript，停下一毫秒来执行这条命令，然后返回呈现包含了来自 JavaScript 命令的 HTML 输出。图 21.1 所示为这个页面，和其他任何 HTML 页面一样。这是一个 HTML 页面，但是有一小部分 HTML 来自一条 JavaScript 命令。

图 21.1

JavaScript 片段的输出和其他输出看上去相似



注意：你可能注意到程序清单 21.1 中的两行：

```
<!-- Hide the script from old browsers
// Stop hiding the script -->
```

这是一个 HTML 注释。任何在<!--和-->之间的内容在源代码中可见但是不会被浏览器呈现。在这个例子中，万一访问者运行很旧的浏览器或者关闭 JavaScript，HTML 注释将包围 JavaScript 代码。



```

    // Stop hiding the script -->
</script>
</head>

<body>
  <h1>Quotable Quotes</h1>
  <p>Following is a random quotable quote. To see a new quote just
  reload this page.</p>
  <script type="text/javascript">
    <!-- Hide the script from old browsers
    getQuote();
    // Stop hiding the script -->
  </script>
</body>
</html>

```

尽管这段代码看上去很长，但是如果你仔细察看，将会发现许多代码由 4 个可用于在页面上显示的格言组成。忽略长度，代码本身并不特别复杂。

在第一组 `<script></script>` 标签之间的大量代码行创建一个函数 `getQuote()`。定义了这个函数，它可以在页面的其他地方调用。注意，如果函数存在于一个外部文件，就可以被所有页面调用。

如果你仔细地察看这些代码，将会看到一些这样的代码行：

```

// Create the arrays
和
// Initialize the arrays with quotes

```

这些代码行都是代码注释。开发人员使用这些注释在代码中留下注解，这样阅读者将了解特定场所代码的作用。在第一个关于创建数组的注释之后，你可以看到创建了两个数组——一个称为 `quotes`，另一个称为 `sources`——每个数组包含 4 个元素：

```

quotes=new Array(4);
sources= new Array(4);

```

在第二个注释（关于用格言初始化数组）之后，数组中添加了 4 个元素。我们将仔细地观察这些元素，第一句格言来自马克·吐温：

```

quotes[0] = "When I was a boy of 14, my father was so " +
"ignorant...but when I got to be 21, I was astonished at " +
"how much he had learned in 7 years.";
sources[0] = "Mark Twain";

```

已经知道数组被命名为 `quotes` 和 `sources`。但是在这个例子中被赋值的是 `quotes[0]` 和 `sources[0]`。因为格言和来源都是数组，数组中的项目都有其位置。使用数组时，其中的第一个项目不是 1 号，而是 0 号。换句话说，你从 0 开始而不是从 1 开始计数。因此，第一句格言的文本（值）被赋值给 `quotes[0]`（变量）。同样，第一个来源值被赋给 `sources[0]`。

字符串包含在引号中。但是，在 JavaScript 中，换行表示命令的结束，所以下面的代码将出现问题：

```

quotes[0] = "When I was a boy of 14, my father was so
ignorant...but when I got to be 21, I was astonished at
how much he had learned in 7 years.";

```

因此，你看到这个字符串由一系列包含在引号之中，用加号连接的字符串组成。

接下来的代码块看上去最像程序；这一个代码行得到一个随机数：

```
i=Math.floor(Math.random()*quotes.length);
```

但是你不能仅仅选取随机数，因为随机数的目的是确定打印的格言和来源——这里只有 4 条格言。所以，这一 JavaScript 代码行将会有以下特色。

- 使用 `Math.random()` 得到 0 到 1 之间的随机数。例如，结果可能是 0.5482749。
- 将随机数乘以格言数组的长度，目前是 4，数组的长度是元素的数量。如果随机数是 0.5482749（如前所述），乘以 4 等于 2.1930996。
- 使用 `Math.floor()` 将结果取整为最靠近的整数。换句话说，2.1930996 变为 2。
- 将 2 赋给变量 `i`。

函数的余下部分应该很熟悉，只有少数的例外。首先，在本章的前面学习过，`documents.write()` 用于写入浏览器呈现的 HTML。接下来，字符串被分离，清晰地表示将被做不同处理的内容，比如当反斜杠需要被打印时或者代替变量值时脱离引号。实际打印的格言和来源匹配 `quotes[i]` 和 `sources[i]`，`i` 是由以上的数学函数确定的数值。

但是简单地编写这个函数并不意味着将会创建输出。在 HTML 中，你接着会在 `<script></script>` 标签之间看见 `getQuote()`；——这是函数调用的方式。调用函数的地方就是函数输出所放置的地方。在这个例子中，输出显示在介绍该格言的段落之下。

图 21.2 所示为浏览器中的 Quotable Quote（值得引用的格言）页面。当页面重新装入时，显示这 4 条格言之一——完全是随机的！

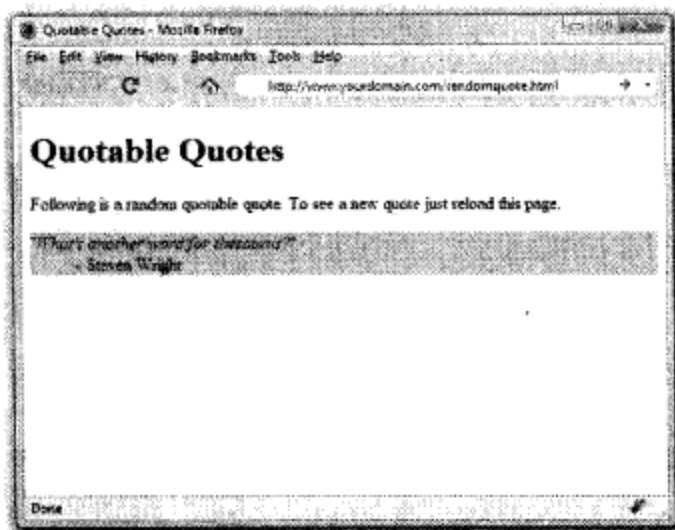


图 21.2

Quotable Quote 页面每次装入时显示随机的格言

记住，你可以简单地修改该页面以包含自己的格言或者其他希望随机显示的文本。你还可以在 `quotes` 和 `sources` 数组中添加更多的条目以增加可用的格言数量。

如果你将 Quotable Quote 页面用作出发点，就可以简单地修改脚本并且创建自己有趣的变种。如果你在这里出现了错误，就随它去吧。对付脚本代码错误的诀窍是耐心并且小心地分析输入的代码。你始终可以删除代码以简化脚本，直到能够正常工作，然后每次添加一些代码以确保这些代码正常工作。

## 21.4 理解文档对象模型

使用 JavaScript 的客户端交互一般采用以某种方式操纵文档对象模型（DOM）的形式。

DOM 是所有文档的不可见结构——不是 HTML 结构或者应用格式化级别的方式，而是一种整体的框架或者容器。如果这个描述有些模糊，那是因为它本来如此，DOM 不是个有形的对象。

整体的容器对象被称为文档。文档中的任何容器都有由文档引用的 ID。例如，如果你有一个 ID 为 `wrapper` 的 `<div>`，那么在 DOM 中该元素这样引用：

```
document.wrapper
```

在第 17 章中，你通过改变关联的 `style` 对象中的一些内容改变了特定元素的可见性。如果你希望访问 ID 为 `wrapper` 的 `<div>` 的背景颜色样式，可以如此引用它：

```
document.wrapper.style.background.color
```

为了按照交互用户事件将样式值改变为其他值，使用如下命令将颜色改变成白色：

```
document.wrapper.style.background.color="#ffffff"
```

DOM 是这样引用元素及相关对象的能力背后的框架。很显然，这是相当复杂的概念的一种简单概述，但是至少现在你可以开始掌握文档一句点一属性的引用方法。要学习更多关于 DOM 的指示，访问万维网联盟关于 DOM 的信息 (<http://www.w3.org/DOM/>)。

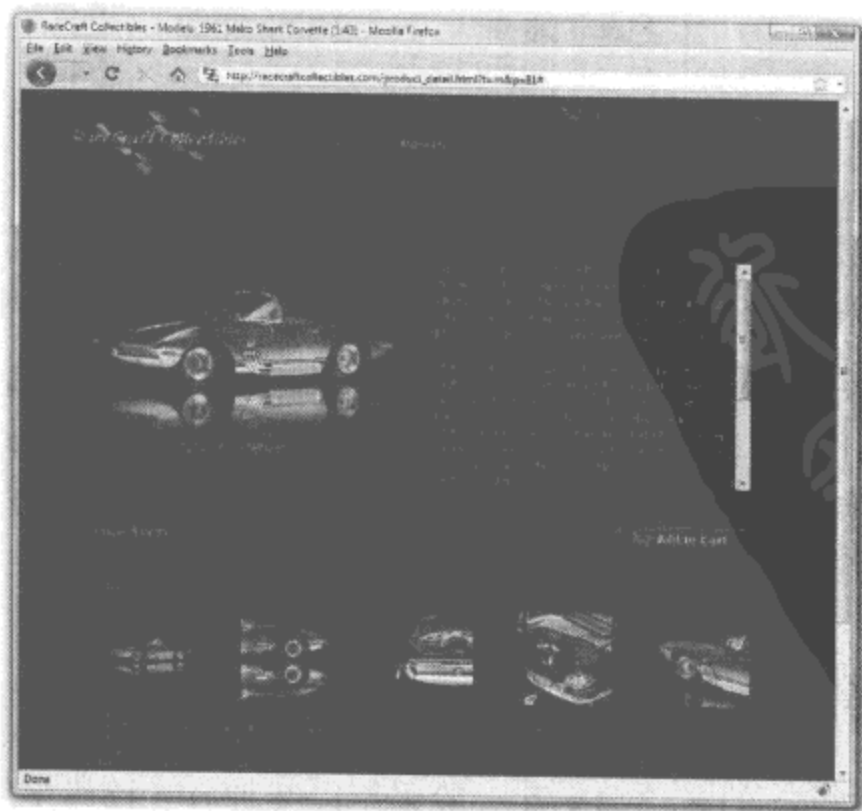
## 21.5 根据用户交互改变图像

第 18 章向你介绍了不同类型的用户交互事件，如 `onclick`、`onmouseover`、`onmouseout` 等。在那一章中，你根据用户的交互改变了文本，在这一节，你将看到一种常见的交互的示例，既实用又是动态的。

图 21.3 所示为一家回收公司的在线目录中的一个页面。目录中的每个页面都显示一个大的图像、货物的信息以及页面底部的一组较小的图像。在这种类型的目录中，每种货物的细节的特写照片对潜在购买者来说很重要，但是在一个页面上放置多个大的图像从显示和带宽方面看都是不灵便的。

图 21.3

用户第一次装入的  
目录项页面



页面上的大图像使用<img/>标签调用:

```

```

你可以看到, 这个图像被命名为 `product_img`。因此, 该图像在 DOM 中的存在形式为 `documents.product_img`。这很重要, 因为 JavaScript 的一些功能使我们能够动态改变 `documents.product_img.src` 的值, 这是图像的来源 (`src`)。

下面的代码片段创建图 21.3 底部显示的一组 5 个图像中的第 4 个小图像。onmouseover 事件指出, 用户对这个图像进行翻转时, `documents.product_img.src` (大图像位置) 的值被填充为匹配的一个大图像的路径。

```
<a href="#" onmouseover="javascript:document.product_img.src =  
'/path/to/large4.jpg'"></a>
```

图 21.4 所示为相同的页面——这不是用户重新装入的——当用户翻转页面底部的小图像时, 大图像的位置被不同的图像占据。鼠标指针悬停于右起的第二个图像之上。随着用户翻转内部图像的较小版本, 大的版本出现在页面的顶部区域。

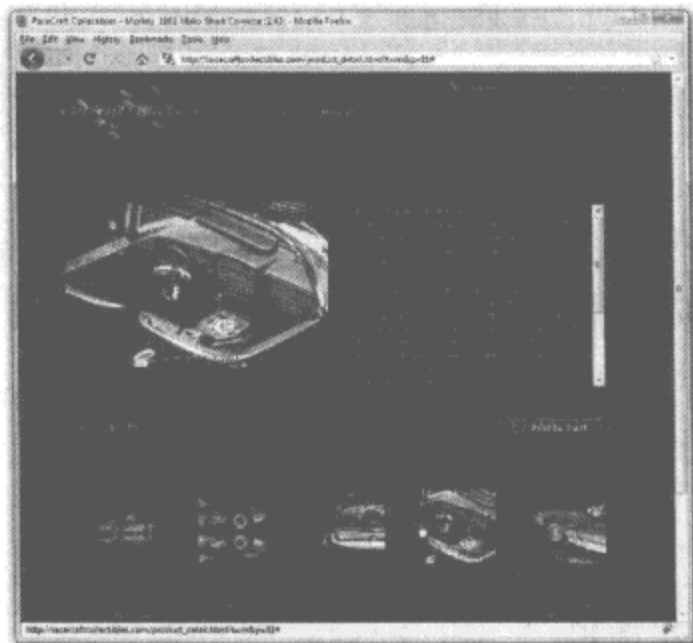


图 21.4

用户翻转小图像时,  
大的图像被替换

## 21.6 总结

本章中, 你学习了服务器端和客户端脚本之间的不同, 并且学习了在 HTML 中包含 JavaScript 为网站添加一些交互性的方法。你还学习了使用 JavaScript 的 `document.write()` 方法来在页面装入时显示随机的格言。最后, 你学习了文档对象模型的知识。

应用前面的章节中得到的知识, 你已经学习了使用客户端脚本将网页上的图像与鼠标移动对应的方法。这些任务都不需要太多编程技巧, 但是它们能鼓舞你学习更多的 JavaScript 和服务器端编程语言的知识, 为你的页面带来更复杂的交互特性。

## 21.7 问与答

问: 如果我希望使用本节课中的随机格言脚本, 但是我希望有一个大的格言库, 必须在



每个页面上放置所有格言吗？

答：是的，数组中的每个项目都必须存在。这里你可以看到适合用客户端和服务端脚本实现的功能之间的差别。如果你有一个真正的随机格言库，并且每次只显示一个，将这些项目保存在一个数据库并且使用少量服务器端脚本来连接这个数据库，读取文本，并且在网页上打印可能是最好的方案。你也可以始终继续在 JavaScript 中携带所有格言，但是至少应该把 JavaScript 函数放到不同的文件中，这样可以和文本分开维护。

问：我曾经见过一些在线目录将大的图像作为网站内容之上的一层来显示——可以看到图像下面的常规网站内容，但是焦点在大的图像上。这是如何做到的？

答：这一描述听起来像是 JavaScript 程序库“LightBox”的效果。LightBox 库使你能在一个在网站内容之上的图层中显示一个或者一组图像。这是一个非常流行的程序库，用于显示大图像细节或者重要到足以在内容之上展示的图像。该程序库在创造者的网站上 (<http://www.huddletogether/projects/lightbox>) 可以免费得到。根据软件包含的说明安装和使用它，你将能够使用在本书中得到的知识集成它。

## 21.8 作业

### 21.8.1 测验

1. 你建立了一个按钮图像，命名为 `button.gif`。你还建立了一个按钮的简单 GIF 动画，在绿色和白色之间闪烁。你将这个 GIF 命名为 `flashing.gif`。要让用户将鼠标指针放到按钮之上时使鼠标闪烁，并且在用户单击按钮时链接到 `gohere.html` 页面，应该使用什么样的 HTML 和 JavaScript 代码？

2. 如何修改问题 1 中的代码，使用户的鼠标移到按钮之上时按钮产生闪烁，即使鼠标移开，按钮还继续闪烁？

3. 下面的上下文中，加号是什么意思：

```
document.write('This is a text string' + 'that I have created.');
```

### 21.8.2 测验答案

1. 使用下面的代码：

```
<a href="gohere.html"
onmouseover="javascript:document.flasher.src='flashing.gif'"
onmouseout="javascript:document.flasher.src='button.gif'">
</a>
```

2. 使用下面的代码：

```
<a href="gohere.html"
onmouseover="javascript:document.flasher.src='flashing.gif'">
</a>
```

3. 加号用于连接两个字符串。



## 21.9 练习

- 你是否有些页面，如果鼠标经过导航图标或者其他图像时，这些图像有所改变能够使页面更生动或者更容易理解？如果有，尝试创建图像的一些突出显示版本，试着使用本章介绍的信息修改页面。
- 你可以显示随机的图像——比如图像横幅或者广告——和本章前面学到的使用 JavaScript 显示随即内容的方法相同。不打印文本，而只打印你希望显示的图像的 `<img/>` 标签。



## 第 22 章

# 使用基于 Web 的表单

---

本章中你将学到：

- HTML 表单工作原理
- 创建 HTML 表单前端的方法
- 命名表单数据各部分名称的方法
- 在表单中包含隐藏数据的方法
- 选择正确的表单输入控件的方法

到现在为止，本书都集中于输出信息。但也可以使用网页向阅读网页的人收集信息。

Web 表单让你能够接收反馈、订单和来自网页浏览者的其他信息。如果用过 Google、Yahoo!或 Bing 等搜索引擎，应熟悉 HTML 表单。这些单字段输入表单带有一个按钮，按下按钮时，给出所寻找的全部信息。产品订单也是表单极其常见的一种用途，如果你曾经在 Amazon.com 订购或者从 eBay 购买过东西，就用过这种表单。在本章中，你学习创建自己的表单的方法，但是学习的只是表单前端的创建。对表单后端的处理需要编程语言的知识，这超出了本书的范畴。

HTML 表单是网页的一部分，包含了用户输入信息的区域，这些信息发送给你，你所指定的其他 E-mail 地址，你所管理的数据库或者其他系统（如用于表单的第三方管理系统，比如 Salesforce.com）。

### 22.1 HTML 表单的工作原理

学习用于创建表单的 HTML 标签前，应理解访问者在表单中填写的信息如何发送给你。幕后的实际过程（服务器端或者后端）需要至少一种编程语言的知识，或者至少有根据说明使用别人的服务器段脚本处理表单输入的能力。在这一过程中，你应该与具有技术知识的人

一起工作，或者自己学习一些基础知识。简单的表单处理并不困难，而且很可能你的 Web 主机提供商有许多可以直接使用的后端脚本。

**注意：**PHP 是最流行的服务器端编程语言，任何称职的 Web 主机提供商都支持它。你可以在 <http://www.php.net/> 上学到更多关于 PHP 的知识，也可以通过本书的姊妹篇《Sams Teach Yourself PHP, MySQL and Apache All-in-one》立刻从头开始这个编程语言的学习（加上第 21 章讨论过的数据库交互）。尽管还有许多关于 PHP 的书籍，但是我偏好这本书，因为这是由我编写的。这本书是针对 PHP 或者其他编程语言的初学者的。

**By the Way**

每个表单都必须包含一个按钮，让用户单击以提交表单，这个按钮可以是自己创建的图像或者由 `<input>` 标签和 `submit` 的 `type` 值创建的标准 HTML 表单按钮。当用户点击表单提交按钮时，所有表单中输入的信息被发送到 `<form>` 标签的 `action` 属性中指定的 URL。该 URL 指向处理表单的特殊脚本，通过 E-mail 发送表单内容或者执行交互过程中的另一个步骤（比如从搜索引擎请求结果或者将货物放在在线购物筐中）。

当你开始考虑更复杂的表单内容处理，而不只是将结果通过 E-mail 发送给你时，需要更多的知识。例如，如果你希望创建一个在线商店，接受信用卡并且处理业务，有一些精心建立的方法来完成这些工作，这些方法都针对确保客户数据的安全。这不是你能轻易进行的工作，你需要比本书所提供的更多的知识。

在你将表单放到网上之前，应该看一看 Web 主机提供商的用户指南，了解一下他们所提供的表单处理脚本。你可能会搜索 CGI（通用网关接口）这个关键字。CGI 是程序（如表单处理脚本）与 Web 服务器交互的标准方式。

**注意：**从技术上说，可以不通过服务器端脚本来发送表单数据，但是这种方法（在 `<form>` 的 `action` 属性中使用 `mailto` 链接）会产生不一致的结果。各种 Web 浏览器，以及个人安全设置，将会导致该操作的响应与你的意图以及用户的预期不同。当用户提交一个表单，他们预计会调用一个脚本，执行他们所看不见的一个过程，然后返回表示这个过程结束的一条信息。而调用 `mailto` 的表单不是这种情况。

**By the Way**

## 22.2 创建表单

每个表单都必须以 `<form>` 标签开始，该标签可放在 HTML 文档主体的任何位置。`form` 标签通常有两个属性，即 `method` 和 `action`：

```
<form method="post" action="mailto:me@mysite.com">
```

最常用的 `method` 属性为 `post`，这表示将表单输入结果作为文档发送。在有些特殊情况下，可能需要使用 `method="get"`，这将表单结果作为 URL 报头的一部分提交。例如，通过 Web 表单提交查询给搜索引擎时有时使用 `get`。由于读者不是表单专家，因此只要使用 `post` 即可，除非 Web 主机提供商的文档告诉你使用别的。

`action` 属性指定要将表单数据发送到的地址，有两种选择。

- 输入 Web 服务器中表单处理程序或脚本的地址，这样表单数据将发送给该程序。
- 输入 `mailto:` 和你的电子邮件地址，这样表单数据将直接发送给你。然而，这种方法完全依赖于用户计算机是否正确配置了电子邮件客户端程序。从公共计算机访问网站而没有电子邮件客户端程序的人将无法提交表单数据给你。

程序清单 22.1 所创建的代码和如图 22.1 所示的表单包含当前可在 HTML 表单中使用的各种用户输入。阅读下述对每种输入元素的解释时，可参阅这个程序清单和图示。

### 程序清单 22.1 具有各种用户输入组件的表单

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Guest Book</title>

    <style type="text/css">
      .formlabel {
        font-weight:bold;
        width: 250px;
        margin-bottom: 12px;
        float: left;
        text-align: left;
        clear: left;
      }
      .formfield {
        font-weight:normal;
        margin-bottom: 12px;
        float: left;
        text-align: left;
      }

      input, textarea, select {
        border: 1px solid black;
      }
    </style>
  </head>
  <body>
    <h1>My Guest Book</h1>
    <p>Please sign my guest book. Thanks!</p>
    <form method="post" action="URL_to_script">

      <div class="formlabel">What is your name?</div>
      <div class="formfield"><input type="text" name="name"
        size="50" /></div>

      <div class="formlabel">What is your e-mail address?</div>
      <div class="formfield"><input type="text" name="email"
        size="50" /></div>

      <div class="formlabel">Please check all that apply:</div>
      <div class="formfield">
        <input type="checkbox" name="website_response[]" value="I
          really like your Web site." />I really like your Web site.<br />
        <input type="checkbox" name="website_response[]" value="One
          of the best sites I've seen." />One of the best sites I've
          seen.<br />
      </div>
    </form>
  </body>
</html>
```



```
<input type="checkbox" name="website_response[]" value="I sure
wish my site looked as good as yours." />I sure wish my site
looked as good as yours.<br />
<input type="checkbox" name="website_response[]" value="I have
no taste and I'm pretty dense, so your site didn't do much for
me." />I have no taste and I'm pretty dense, so your site
didn't do much for me.<br />
</div>

<div class="formlabel">Choose the one thing you love best about my
web site:</div>
<div class="formfield">
  <input type="radio" name="lovebest" value="me" />That gorgeous
  picture of you.<br />
  <input type="radio" name="lovebest" value="cats" />All the
  beautiful pictures of your cats.<br />
  <input type="radio" name="lovebest" value="childhood" />The
  inspiring recap of your suburban childhood.<br />
  <input type="radio" name="lovebest" value="treasures" />The
  detailed list of all your Elvis memorabilia.<br />
</div>
<div class="formlabel">If my web site were a book, how many copies
would it sell?</div>
<div class="formfield">
  <select size="3" name="sales">
    <option value="Millions, for sure." selected="selected">Millions,
    for sure.</option>
    <option value="100,000+ (would be Oprah's favorite)">100,000+
    (would be Oprah's favorite)</option>
    <option value="Thousands (an under-appreciated classic)">Thousands
    (an under-appreciated classic)</option>
    <option value="Very few: not banal enough for today's public">Very
    few: not banal enough for today's public.</option>
    <option value="Sell? None. Everyone will download it for free.">Sell?
    None. Everyone will download it for free.</option>
  </select>
</div>

<div class="formlabel">How can I improve my web site?</div>
<div class="formfield">
  <select name="suggestion">
    <option value="Couldn't be better." selected="selected">Couldn't
    be better.</option>
    <option value="More about the cats.">More about the cats.</option>
    <option value="More about the family.">More about the family.</option>
    <option value="More about Elvis.">More about Elvis.</option>
  </select>
</div>

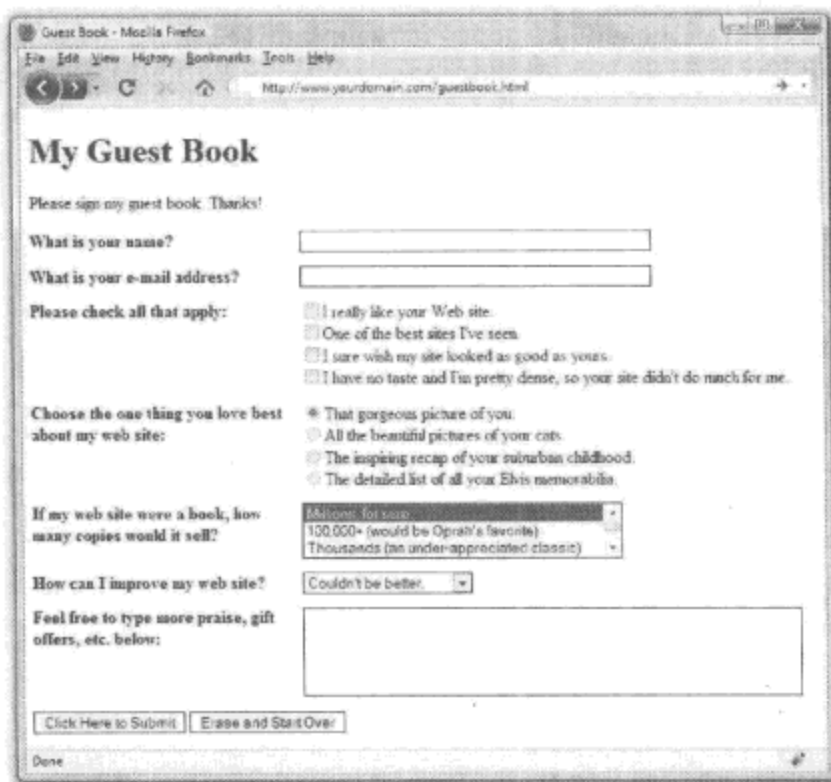
<div class="formlabel">Feel free to type more praise, gift offers, etc.
below:</div>
<div class="formfield">
  <textarea name="comments" rows="4" cols="55"></textarea>
</div>

<div style="float:left;">
  <input type="submit" value="Click Here to Submit" />
  <input type="reset" value="Erase and Start Over" />
</div>
</form>
</body>
</html>
```



图 22.1

程序清单 22.1 中的代码使用几乎所有 HTML 表单输入元素



程序清单 22.1 中的代码使用包含许多 `<input/>` 标签的 `<form>` 标签。每个 `<input/>` 标签对应特定的用户输入组件，比如复选框或者单选按钮。输入、选择和文本区域元素在样式单中包含边框，所以很容易看到表单中元素的轮廓。记住，你可以在这些元素上应用各种 CSS。

下面几个小节仔细研究 `<input/>` 和其他表单相关的标签。

## 22.3 接受文本输入

要求用户在表单中输入信息，可使用 `<input />` 标签。该标签必须位于 `<form>` 和 `</form>` 标签之间，但与网页中任何文本、图像和其他 HTML 标签之间的相对关系没什么限制。例如，要询问用户的姓名，可以输入下列代码：

```
What's your name? <input type="text" size="50" maxlength="100"
name="name" />
```

`type` 属性指明要显示的是哪种表单元素，这里是单行文本框（接下来的每小节将单独介绍一种元素）。

`size` 属性指明文本框的宽度大约为多少个字符。如果使用的不是等宽字体，则输入框的宽度将随用户输入的字符而异。如果输入太长，不能全部显示在文本框中，大多数网页浏览器将自动向左滚动文本。

`maxlength` 决定文本框允许用户输入的最大字符数。如果用户的输入超过指定的长度，多余的字符将不会显示。指定的长度可以比文本框大，也可以比它小，还可以相等。`size` 和 `maxlength` 只能用于 `type="text"`，因为其他输入类型（复选框、单选按钮等）的大小是固定的。

### Did you Know?

**提示：** 如果希望用户输入文本而不显示在屏幕上，可用 `<input type="password" />` 来代替 `<input type="text" />`。这样用户输入的文本将显示为星号 (\*\*\*)。属性 `size`、`maxlength` 和 `name` 在 `type="password"` 中的用法与在 `type="text"` 中相同。记住，隐藏密码的方法只能提供视觉上的保护——密码传输时没有加密或其他保护措施。

## 22.4 标识每项表单数据

不管是哪种输入元素，都必须为其收集的数据赋予名称。可以为每个输入项使用任何名称，只要它们在表单中互不相同（本章稍后讨论的单按钮和复选框除外）。后端脚本处理表单时，每项数据都是根据名称来标识的。这个名称变成一个变量，填写一个值。这个值是用户在表单中输入的或者与用户选择的元素相关的值。

例如，如果用户在前面定义的文本框中输入 Jane Doe，一个变量被发送给表单处理脚本，这个变量是 name，值为 Jane Doe。表单处理脚本使用这种变量名称和值。

接下来的小节中有更多的名称/值对的示例。

**注意：**为了在本书的范围内进行解释表单处理脚本在这里被简化了。处理脚本可用的变量的真正表现（或者名称）取决于脚本的编程语言。但是从概念上，在后端上输入元素的名称变成变量名称，输入元素的值变成变量值是没有错的。

**By the Way**

## 22.5 在表单中包含隐藏数据

要将数据项发送给处理表单的服务器脚本而不希望用户看到这些数据，可使用具有 type="hidden" 属性的 <input> 标签。这个属性对显示没有任何影响，它只是在表单结果提交时添加指定的名称和值。

如果你使用 Web 主机提供商提供的表单处理脚本，可用这个属性告诉脚本将表单结果发送到哪个电子邮件地址。例如，下面的代码指示将结果发送到 me@mysite.com:

```
<input type="hidden" name="mail_to" value="me@mysite.com" />
```

脚本往往使用至少一两个隐藏输入元素来传送接收表单提交结果时可能有用的附加数据，一些例子包含 E-mail 地址和主题。如果你使用 Web 主机提供商的脚本，参考随脚本提供的文档，以了解可能需要的隐藏字段。

## 22.6 研究表单输入控件

有多种输入控件可用于从用户那里收集信息。你已经了解了一种文本输入选项。接下来的几个小节将介绍设计表单时可使用的其他表单输入控件。

### 22.6.1 复选框

最简单的输入类型是复选框，它显示为一个小方框，用户可以单击选中或取消一个或者多个项目。例如，程序清单 22.1 中在“Please check all that apply”标签之后显示的复选框列表，标签提示用户确实可以选中所有需要应用的项目。

程序清单 22.1 中的复选框 HTML 说明所有复选框的 name 属性都是: website\_response[]。

```



```

方括号 ([]) 的使用为处理脚本指明了，有一系列值而不仅是一个值被放到这个变量中（如果用户只选择一个复选框，也可能仅有一个值）。如果用户选择第一个复选框，“I really like your Web site.” 这个字符串将被放到 `website_response[]` 的方括号中。如果用户选择第三个复选框，“I sure with my site looked as good as yours.” 这个字符串也将被放到 `website_response[]` 的方括号中。处理脚本将把这个变量作为一个数组而不是一个输入来处理。

### Did you Know?

**提示：**如果发现输入元素的标签与元素靠得太近，只要在 `<input />` 标签和标签文本之间添加一个空格，如下所示：

```
<input type="checkbox" name="mini" /> Mini Piano Stool
```

但是，你可能也会看到一组有不同变量名的复选框。例如，下面的代码是另一种复选框组的写法：

```



```

在这些复选框中，第一个复选框的变量名为“`liked_site`”，值为“`yes`”。

如果你希望浏览器呈现表单时某个复选框默认选中，可以包含 `checked` 属性。例如，下面的代码创建两个复选框，第一个被默认选中：

```



```

### Watch Out!

**警告：**XHTML 要求所有属性都要有等号和值。这就是我为什么使用 `checked="checked"` 表示复选框被选中，而不是只使用 `checked` 的原因。该规则适用于所有的 HTML 布尔（`true/false`、`on/off` 和 `yes/no` 等）属性。

这个示例中标签为“`I really like your Web site.`”的复选框被选中。用户可以单击复选框来表示他们有其他的选择。“`One of the best sites I've seen.`”复选框开始未选中，所以用户必须点击它才能打开。

没有选中的复选框不会包含在表单输出中。

## 22.6.2 单选按钮

单选按钮每次只能选择一个，实现起来和复选框一样容易。单选按钮的最简单用法是用于是非题或者仅能选择一个候选人的投票。

创建一个单选按钮，只要使用 `type="radio"`，每个单选按钮放在一个 `<input />` 标签中。同一组单选按钮的名称都相同，但不要使用复选框中的 `[]`：

```
<input type="radio" name="vote" value="yes" checked="checked" /> Yes<br />
<input type="radio" name="vote" value="no" /> No <br />
```

`value` 可以是你所选择的任何名称。如果你包含了 `checked` 属性，该按钮被默认选中。在相同名称的单选按钮中只能选中一个。

在设计表单并且在复选框和单选按钮中做出选择时，试问自己：这个问题是否只有一个答案？如果是这样，使用单选按钮。

**注意：**如果你查看其他 Web 设计人员的表单，可能会看到复选框使用相同的名称但是有不同的值，如：

```
<input type="checkbox" name="pet" value="dog"> dog<br />
<input type="checkbox" name="pet" value="cat"> cat<br />
<input type="checkbox" name="pet" value="iguana"> iguana<br />
```

如果用户选中超过一个复选框，很可能只有最后一个值为脚本所处理。这就是应该对复选框组，复选框组名称和提交之前发送给脚本的值加以考虑的原因。

**By the  
Way**

## 22.6.3 选择列表

滚动列表和下拉选择列表都由 `<select>` 标签创建。可将该标签与 `<option>` 标签一起使用，如下面的例子所示：

```
<select name="extras" size="3" multiple="multiple">
  <option value="Electric windows" selected="selected">Electric
  windows</option>
  <option value="Sunroof">Sunroof</option>
  <option value="AM/FM Radio">AM/FM Radio</option>
  <option value="CD Player">CD Player</option>
  <option value="GPS">GPS</option>
</select>
```

标签 `<select>` 和 `</select>` 之间除 `<option>` 标签和 `</option>` 标签外，不应包含其他标签。

与文本输入元素不同的是，这里的 `size` 属性决定选择列表可同时显示多少项。如果在上述代码中使用 `size="2"`，则只能同时看到两项，列表旁边将显示滚动条，用户可以向下滚动以查看第三项。

`multiple` 属性让用户能够同时选择多项，`selected` 属性指定默认被选中的项。表单提交时，返回的是被选中的选项的 `value` 属性指定的文本。

**提示：**如果没有指定 `size` 属性或指定 `size="1"`，列表将为下拉选择列表。下拉选择列表不允许同时选中多项。从逻辑上说，它相当于一组单选按钮。例

**Did you  
Know?**

如，选择 yes/no 答案可采用另一种方法：

```
<select name="vote">
  <option
value="yes">Yes</option>
  <option
value="no">No</option>
</select>
```

## 22.6.4 文本区域

前面介绍的 `<input type="text">` 只允许用户输入一行文本。如果希望用户能够输入多行文本，可使用标签 `<textarea>` 和 `</textarea>`，包含在这两个标签之间的文本默认显示在文本区域中，下面是一个例子：

```
<textarea name="comments" rows="4" cols="20">Please send more
information.
</textarea>
```

你可能猜到了，`rows` 和 `cols` 属性控制输入框文本的行数和列数。`cols` 属性不像 `rows` 属性那样精确，它是一行文本包含的大概字符数。然而，文本区域也有滚动条，因此用户可以输入比显示区域更多的文本。

## 22.7 提交表单数据

每个表单都必须包含一个用于将表单数据提交给服务器的按钮。可用 `value` 属性在这个按钮上添加任何标签：

```
<input type="submit" value="Place My Order Now!" />
```

按钮为灰色，大小由 `value` 属性包含的文本长度决定。用户单击该按钮时，表单中的所有数据都将发送到表单的 `action` 属性指定的电子邮件地址或程序脚本。

也可在表单中包含一个重置按钮，让用户能够在改变主意或犯错时清除表单中的输入。为此，可使用下列代码：

```
<input type="reset" value="Clear This Form and Start Over" />
```

如果觉得标准的 `Submit` 和 `Reset` 按钮太乏味，记住，可以使用 CSS 设置样式。如果这还不够好，有一种简单的方法可在这些按钮中使用自己的图像。要为提交按钮使用自己选择的图像，输入如下代码：

```
<input type="image" src="button.gif" alt="Order Now!" />
```

图像 `button.gif` 将显示在网页中，用户单击该图像时，表单将提交给服务器。也可以包含可用于 `<img />` 标签的属性，如 `alt` 和 `style`（第 8 章介绍了 `<img />` 标签）。

## 22.8 总结

本章演示了创建 HTML 表单的方法，这使你的网页访问者能够输入特殊的信息。我们没

有介绍对这些信息的处理，因为表单处理需要外部的脚本。

你学习了所有主要表单元素的知识，以及表单处理脚本对这些元素的名称和值的解读。当你准备好尝试后端表单处理脚本时，已经相当精通前端的细节了。

表 22.1 总结了本章介绍 HTML 标签和属性。

表 22.1 第 22 章介绍的 HTML 标签与属性

标签/属性	功能
<form>...</form>	指定输入表单
<b>属性</b>	
action="脚本 URL"	处理表单输入的脚本的地址
method="post/get"	表单输入如何传送到服务器，通常设置为 post，而不是 get
<input />	表单的输入元素
type="控制类型 "	输入控件的类型，可能的值包含 checkbox、hidden、radio、reset、submit、text 和 image
name="名称 "	传送给脚本时，该项使用的唯一名称
value="值 "	文本或隐含项的默认值。对于复选框或单选按钮，是与表单一起提交的值；对于重置或提交按钮，是按钮本身的标签
src="图像 URL"	图像的源文件
checked="checked"	对于复选框或单选按钮，指定该项是被选中
size="宽度 "	文本区域的宽度，单位为字符
maxlength="最大长度 "	文本区域中可以输入的最大字符数
<textarea>...</textarea>	表单输入元素多行文本，可包含默认文本
name="名称 "	要传送给脚本的名称
rows="行数 "	文本区域显示的行数
cols="字符数 "	文本区域显示的列（字符）数
<select>...</select>	创建一个包含可能选项的下拉列表或滚动列表
name="名称 "	要传送给脚本的名称
size="元素数 "	显示的项数。如果指定了 size，则为滚动列表；如果没有指定 size，则为下拉列表
multiple="multiple"	允许在列表中选择多项
<option>...</option>	指定<select>元素中的选项
selected="selected"	包含该属性的选项默认被选中
value="值 "	提交表单时，如果该选项被选中，将提交这个值

## 22.9 问与答

**问：**我听说通过 Internet 传送信用卡号很危险。窃贼会在数据传送给我的过程中截取表单数据吗？

**答：**表单数据（和任何网页或电子邮件信息）在 Internet 的传送过程中可能被截取。如果在表单中请求信用卡号或其他敏感信息，你应该在安全的服务器上使用 SSL（安全套接字层）来实施安全浏览。这些 SSL 安全证书通常需要从安全服务提供商，如 Thawte



(<http://www.thawte.com/>) 或 VeriSign (<http://www.verisign.com/>) 那里获得, 但是你应该了解你的 Web 托管提供商安装和销售的 SSL 证书。

要截取 Internet 中传送的信息, 比在餐厅或零售店中从别人背后偷看难得多。虽然如此, 还是应该总是使用安全网页来处理敏感的金融信息(如信用卡号), 尤其是在别人非常信任你的处理时。

问: 我可以将表单放入 CD/DVD-ROM 吗? 它们必须在 Internet 上吗?

答: 可以放置网页的地方就可以放置表单。如果表单在硬盘或 CD-ROM 里而不是 Web 服务器上, 则不管填写者是否连接到 Internet 都可以填写它们。当然, 单击提交按钮时, 他们必须连接到 Internet (或本地 Intranet), 否则无法将信息传送给处理脚本。

## 22.10 作业

作业包含一些测试题及练习, 帮助巩固对本章介绍的内容的理解。试着回答所有问题, 然后再看后面的答案。

### 22.10.1 测验

1. 编写 HTML 代码来创建一个访客本表单, 它询问姓名、性别、年龄和电子邮件地址。假定你在 `/cgi/generic` 有一个表单处理脚本, 需要包含如下隐藏输入元素来告诉脚本将这个表单结果传送到哪里:

```
<input type="hidden" name="mailto" value="you@yoursite.com" />
```

2. 如果创建了一幅名为 `sig-in.gif` 的图像, 如何将其用作问题 1 中访客本的提交按钮?

### 22.10.2 测验答案

1.

```
<html>
  <head>
    <title>My Guestbook</title>
  </head>

  <body>
    <h1>My Guestbook: Please Sign In</h1>
    <form method="post" action="/cgi/generic">
      <p>
        <input type="hidden" name="mailto" value="you@yoursite.com" />
        Your name: <input type="text" name="name" size="20" /><br />
        Your sex: <input type="radio" name="sex" value="male" /> male
        <input type="radio" name="sex" value="female" /> female<br />
        Your age: <input type="text" name="age" size="4" /><br />
        Your e-mail address: <input type="text" name="email"
size="30" />
        <br />
        <input type="submit" value="sign in" />
        <input type="reset" value="erase" />
      </p>
    </form>
  </body>
</html>
```



```
    </p>  
  </form>  
</body>  
</html>
```

2. 将 `<input type="submit" value="Sign In" />` 替换为 `<input type="image" src="sign-in.gif" alt="Sign In" />`

## 22.11 练习

- 创建一个使用各种输入元素和选择列表的表单，以确保理解它们的工作原理。
- 调查你的 Web 托管提供商的表单处理选项并且使用它提供的一个脚本处理前一个练习中创建的表单。



## 第 23 章

# 组织和管理网站

---

本章中你将学到：

- 如何确定何时一个页面足以处理所有内容
- 简单网站的组织方法
- 较大网站的组织方法
- 编写易于维护的 HTML 代码的方法

本书的大部分篇幅引导你学习了 Web 内容的设计和创建，从文本到图像和多媒体。在这期间，我已经说明了一些对内容的生命周期的思考方法，而在本章中你将学习如何将工作作为整体来考虑。

本章介绍如何组织和展示多个网页，让用户能够在这些网页中导航、浏览，而不致于造成困惑。你还将学习如何使网站令人难忘，使访问者愿意再次访问。Web 开发人员使用“黏着”来形容人们不愿离开的网页。这一章将帮助你使网页更有黏着力！

因为网站可以是（而且通常也应该是）经常更新的，因此，创建易于维护的网页很重要。本章将教你如何在网页中增加注释和其他文档，以便你或其他同事能够理解和修改网页。

### TRY IT YOURSELF ▼

#### 自己动手尝试一下：评估你的组织

现在，你应该已经有了足够的 HTML 知识，能够创建绝大部分网站。你可能有许多网页，甚至已经将它们发布到网上。

阅读本章课程时，考虑一下你的网页是如何组织的以及如何改进。你在 HTML 中使用过注释，或者用于未来网站维护的关于组织的文档吗？如果没有，现在是开始这么做的好时机了。在这个过程中，如果决定重新设计并修改几乎所有网页，不要为此感到惊讶，这种努力很可能是值得的！

## 23.1 一个网页就足够时

创建和组织具有吸引力的高效网站并不总是一项复杂的工作。如果你要创建的是用于一个只需要少数特殊信息的实体（如区域会议）的网站，可以在一个网页中展示许多有用的信息，而不需要许多花哨的图像。实际上，单页网站有几个优点。

- 网站上的所有信息都可以尽快地下载。
- 尽管网页打印出来需要几张纸，但只需一个打印命令就可以将整个网站打印到纸上。
- 访问者很容易将网站保存在其硬盘上以便将来引用，特别是当网站使用很少的图像时。
- 同一网页上不同部分的链接的响应速度通常比不同网页之间的链接快。

图 23.1 所示为网页的第一部分，这作为一个长的网页比使用多页网站更好。与大多数介绍性网页一样，这个网页开始简单地介绍了这个页面的主要内容和面向的读者。详细的目录让访问者能够直接跳到最感兴趣的内容。它包含 8 页纸的文本，解释购买房屋的流程——访问者可以考虑打印出来，稍后阅读，也可以做笔记。

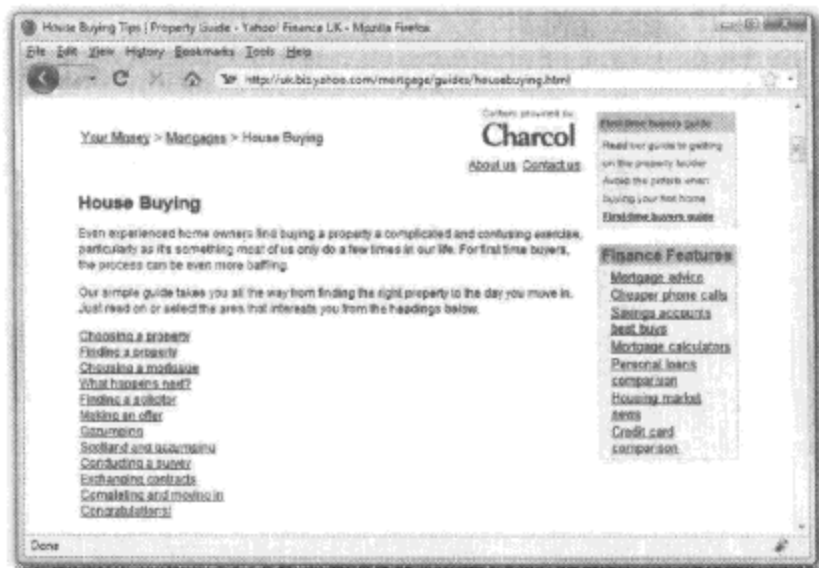


图 23.1

一个好的目录可以使冗长的页面容易导航

如图 23.2 所示，页面中每小节后面都有一个链接指向目录，因此，在这个网页中导航感觉就像在多个网页之间导航一样。因为该网页旨在提供方便的参考，访问者肯定更愿意将单个网页作为书签或保存下来，而不愿对 8 个或 10 个网页进行操作。

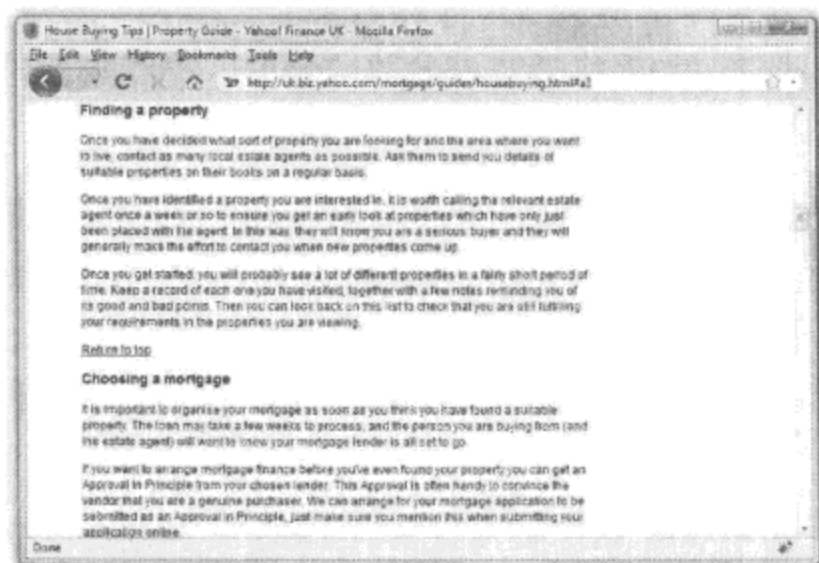


图 23.2

在长页面的每个部分之后，始终提供回到目录的链接

在本书见过各种花哨的图像和布局技巧后，你可能忘记了这样一点：通常良好的老式大

纲是组织网站中的冗长页面的最清晰也最有效的方法。

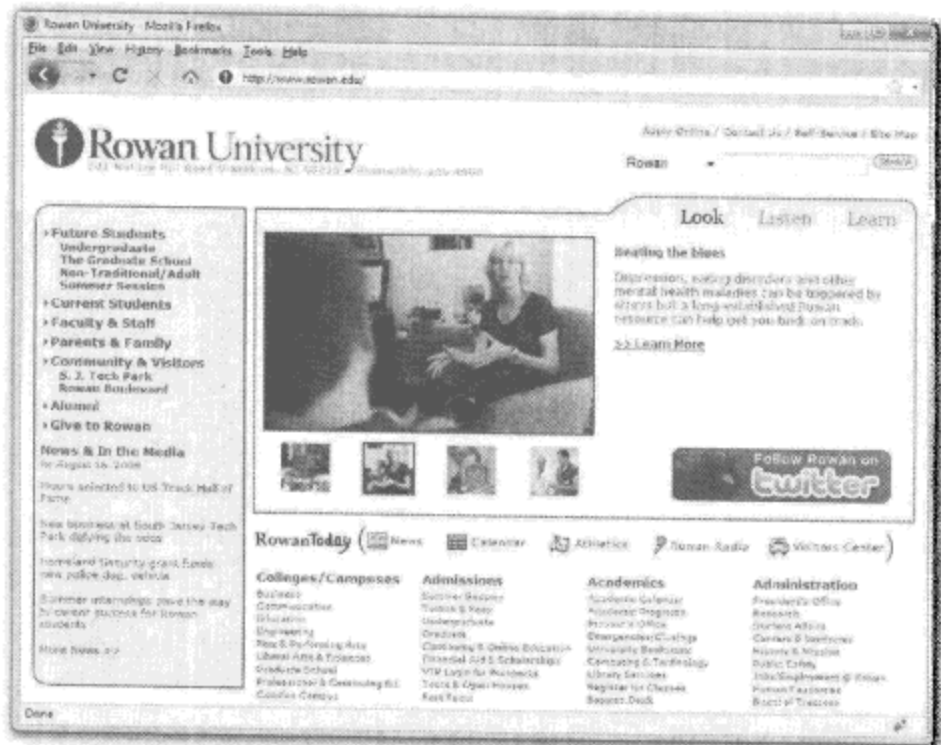
## 23.2 组织简单的网站

虽然有些网站只有一个网页，但大部分公司和个人将网站分为能够快速阅读的多个短小网页，使用图形图标在网页间导航。这样，用户可以在几次点击之间就看到几乎所有信息。另外，还减少了需要在网页中滚动的次数，这对于使用手机设备来浏览网站或显示器分辨率比较低（小于 800 像素×600 像素）的用户很方便。

“主页”的目标是使组织在互联网上“可见”，但是更重要的是，作为网站中包含的信息的门户。网站的主页应该为用户提供足够的信息，使其清晰地了解组织结构，还有传统的电话联络信息以及可以提交问题和反馈的一个 E-mail 地址。它还应该提供进入网站其他页面的高度结构化信息的路径。图 23.3 中的主页提供了这些良好特性的示例：基本信息、联络信息以及为多个目标用户群提供的信息路径。

图 23.3

这个大学的主页使用基本的设计、最少但有用的图像以及清晰的结构来吸引用户搜索更多的信息



### Did you Know?

**提示：**不管网站有多大，都应仔细组织资源。例如，将网页的图像放在独立的文件夹 images 中。同样，如果有一些文件可供下载，可将其放在文件夹 download 下。这样更易于根据网页资源的类型（HTML 页面、GIF 图像等）对其进行跟踪。此外，如果你将网站分为“公司”、“产品”、“新闻报道”等部分，将页面放入类似名称的目录中进行组织。

网站开发新手最常见的错误之一是，网站上每个页面都与前面的页面看起来不一样。另一个同样严重的错误是使用其他成千上万网页制作者都在使用的公共剪贴画。别忘了，在 Internet 上，单击一下就可以将你带到全世界。要使网站突出并给人印象深刻，唯一的方法是使所有网页都遵循独特的明确无误的视觉主题。换句话说，要尽力使自己的网站与其他网站相比具有独特性，但整个网站内部是统一的。

作为统一性使网站更加内聚的例子，考虑你可能访问过的大型流行网站，如 ESPN.com。如果你访问 ESPN.com 的 MLB（如图 23.4）部分和 NFL（见图 23.5）部分，你会注意到非常相似的结构。



图 23.4  
ESPN.com 的 MLB 部分

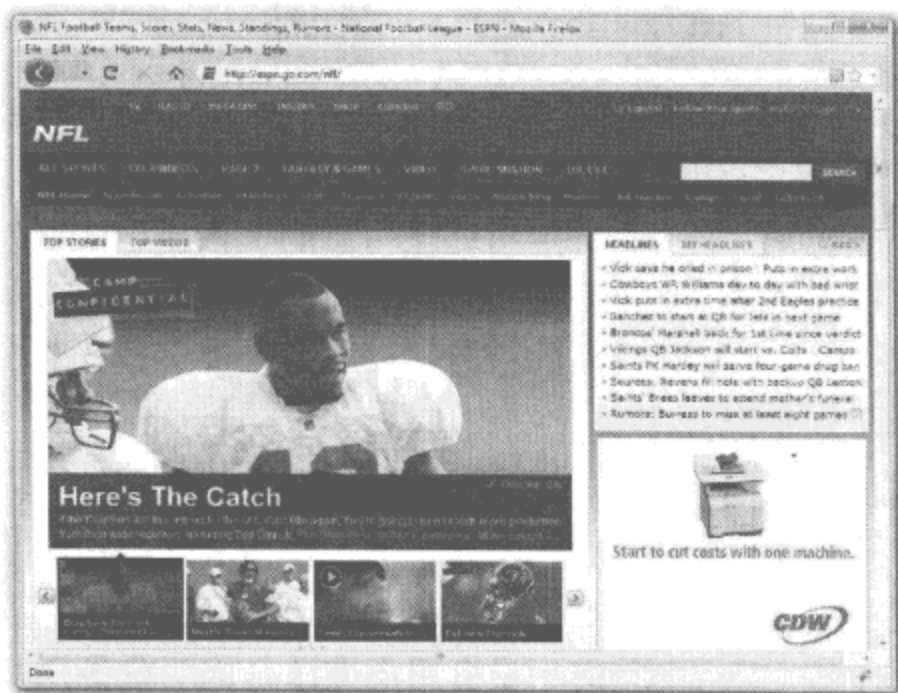


图 23.5  
ESPN.com 的 NFL 部分

在两个例子中，你都能看到页面顶部的导航元素（包含一些子导航元素），页面中间的大区域中的特性图像，右边的一个方框中包含当前头条新闻的链接，下面的第二个方框中用于显示一个广告。MLB 部分和 NFL 部分之间唯一的不同是配色方案：MLB 是蓝色方案为主导，NFL 则以绿色为主。但是，在两个部分中，你都知道如果希望阅读流行的新闻，要看页面的右边。如果你希望导航到网站的其他部分或者主页，你可以查看页面左上角的导航元素。

这些一致的元素确保用户能够充满信心地在内容中浏览。从维护的角度看，一致的结构化模板使你能够重用代码。这种代码重用一般发生在动态编程中（在本书的范围之外），但是一般来说它意味着 HTML 仅存在于一个位置并且动态地应用到内容，而不是一再复制和粘贴相同的 HTML。因此，你仅需要做一次修改，而不是修改上千个文件。

### 23.3 组织较大的网站

对于复杂的网站，将精心设计的布局 and 图像统一地应用于所有页面，有助于组织网站和改进网站的风格。下面看一个需要展示大量信息给多类用户的网站，看看如何使艺术与组织一同发挥作用。



如图 23.6 所示为 Amazon.com 的主页，特别是选中的侧导航栏。Amazon 的业务是销售产品，普通而简单。因此，对于 Amazon 来说，将产品类别作为主导航元素是有意义的，如图所示。

图 23.6

Amazon.com 将产品类别作为主要导航元素

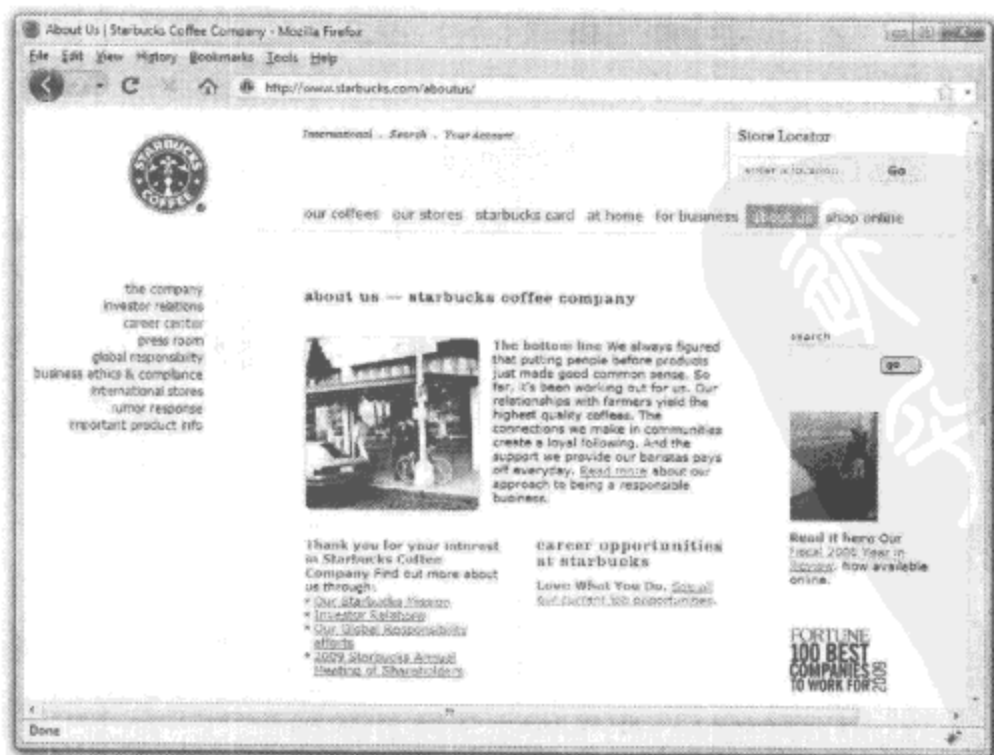


尽管 Amazon 的业务是销售产品，但仍然必须提供关于他们是谁、如何与他们联络以及其他重要的辅助信息来改进公司与客户的关系。指向这些信息的链接出现在网站页脚（页面最后的部分）——在屏幕截图可视范围之外。在创建网站模板时，你必须决定最重要的内容区域以及内容的组织，还要记住为用户提供基本的信息——尤其是能够改进你的形象并让用户感觉到他们的意见对你的价值的信息。

下一个示例是 Starbucks.com（星巴克）网站中的一个二级页面。Starbucks.com 网站中的所有页面遵循相同的导航和子导航显示方法：用于主导航的一个横条，左边的垂直栏目中放置二级元素。如图 23.7 所示，用户当前浏览的部分（“About Us”）高亮显示。这种视觉提示帮助用户确定自己在网站中的位置。使用视觉提示是有用的方法，因为你的用户可能通过搜索引擎或者其他网站的链接进入一个页面。一旦用户进入，你希望他们感觉像在家里一样——或者至少感觉他们知道自己与你的网站有关的地方。

图 23.7

Starbucks.com 二级页面显示一个被选中的主导航元素，二级导航在页面的左侧



你可以从不同的主导航元素——“our coffees”（我们的咖啡）、“our stores”（我们的商店）、“starbucks card”（星巴克卡）、“at home”（在家里）、“for business”（商务）、“about us”（关于我们）和“Shop online”（网上商店）——看到星巴克网站必须为因不同的原因进入网站的不同类型的人的需求服务。在你组织自己的网站内容时，确定对你来说最重要的信息，以及对用户最重要的信息，并且创建达到最佳的折衷的导航框架。

图 23.8 所示为导航样式的另一个示例，这一次对标准的顶部导航/左侧导航方案有一点变化。在这个示例中，左侧导航（在这个例子中的二级导航）还出现在主导航下的下拉菜单（具体做法参见第 17 章）。将鼠标悬停于任何其他导航元素之上显示类似的菜单。这种方案使用户能够在指尖上拥有整个网站地图，因为他们能够在其他页面上用一次点击到达网站上的任何地方。



图 23.8

BAWSI.org 网站显示连接到主导航元素的子导航

你还将会注意到侧导航窗口中的“Overview”连接的样式有所不同——具有比其他链接更深的紫色文本，指出访问者所在的页面。这种视觉细节与你在星巴克网站上看到的相似，是为用户提供其在当前导航框架中所在位置的一种方法。

有许多不同类型的导航样式和方法指出用户所在的位置和下一步可能前往的位置。记住如下的事实：研究不断表明，人们同时面对遇到超过 7 种选择时，将感到困惑不安，人们面对 5 种以下选择时感觉最舒服。因此，应尽可能地避免连续展示超过 5 个链接（不管是列表还是图标），一定要避免通过使用 7 个以上的链接。需要在导航列表中展示超过 7 个链接时，可将它们分为多个列表，每个列表包含 5~7 个链接，正如你在 Amazon.com 示例中所看到的如图 23.6 所示。

如果网页中不指向主页的链接不超过两个（或最多三个），也有助于访问者更好地导航，在访问者浏览完从属网页后，也应让读者能够返回到主分类网页（或主页）中。换句话说，应尽量设计比较扁平的链接结构，使大多数网页到主页的链接深度只有一两级。你肯定不希望用户严重依赖于浏览器的“返回”按钮来导航你的网站。

## 23.4 编写易于维护的 HTML 代码

如果你编写过程序，就知道编写可维护的代码有多重要——也就是说，可维护性指的是

你或其他人以后在阅读你的代码时，不会完全看不懂。困难在于要使代码尽可能容易理解。相信我，你肯定会有这样的时刻，阅读自己编写的网页时完全不知道当时想的是什么或为什么这样编写代码。幸运的是，有一种方法可以解决这种问题！

### 23.4.1 使用注释说明代码

开发 HTML 页面时，请牢记你或其他人总有一天需要对它进行修改。简单的本网页通常很容易阅读和修改，但包含图像、表格和其他布局技巧的复杂网页就较难理解。

在 Internet Explorer 中单击“查看”菜单，然后选择“源文件”，或者在 Firefox 中单击“查看”，然后选择“网页源文件”，看看任何一个网页的源文件，你就会明白我说的。你很可能看到一堆混乱的、难以解读的 HTML 代码。这主要是因为标示是使用内容管理软件系统动态生成的。也可能是因为维护者没有注意结构、易读性、代码注释，以及使代码容易理解的其他方法。为方便维护网页，建议调整网页代码使其更有序。

#### By the Way

**注意：**要在 JavaScript 代码中包含注释，在每行注释前加上//（JavaScript 注释不需要结束标签）。在样式单中，注释以/\*开始，以\*/结束。HTML 标签<!--和-->标签不能用于脚本或样式单！然而，可以也应该在<script>或<style>标签后包含<!--，并在与之匹配的</script>或</style>标签前包含标签-->。注释标签可以防止老式浏览器将脚本或样式当成普通文本显示在网页上。

你在其他章节中已经看到过，可以将为自己或合作者添加的注释放在标签<!--和-->之间。在浏览器中浏览时，这些注释不会显示在网页上，但是使用文本编辑器、字处理器、Web 开发工具或网页浏览器的“查看源文件”（或者查看网页源代码）命令查看 HTML 代码时，可以看到这些注释。下面是一个如何使用注释的例子：

```
<!-- This image needs to be updated daily. -->

```

在这段代码中，<img />标签之前的注释对如何使用图像进行了解释。有人读到这段代码时，就知道这是一个必须每天更新的图像。注释中的文本将完全被浏览器忽略。

#### Did you Know?

**提示：**注释的一种用途是隐藏网页中正在建设的部分。可以通过注释使正在建设的文本和图像隐藏起来，而不是显示它们然后解释它们正在建设中。这样可以逐步构建网页，只显示完成的部分。

#### TRY IT YOURSELF ▼

#### 自己动手尝试一下：为代码添加注释

现在花些时间检查已创建的所有网页、脚本和样式单，添加一些有助于以后查看网页源文件的注释。下面是一些建议。

1. 在比较复杂的格式化或布局技巧使用的标签前添加注释，对其进行解释。
2. 在<img />标签前使用注释，简要地描述那些根据 alt 信息无法知道其功能的重要图像。
3. 考虑使用一条（或多条）注释总结<table>标签中的单元格是如何组织的。

4. 如果使用了十六进制的颜色代码（如`<div style="color:#8040B0">`），插入一条注释说明它实际表示什么颜色（蓝紫色）。
5. 将注释缩进，使注释和 HTML 更易于区别和阅读。HTML 本身也要缩进，使其更容易阅读。

## 23.4.2 缩进使代码更清晰

坦白地说，在本书中，我一直在灌输一种 HTML 代码开发的风格，但没有明确指出。现在到了讲解这种风格的时候了。你肯定注意到了，本书的所有 HTML 代码都使用一种缩进模式。具体地说，每个子标签都比父标签缩进两个空格。另外，标签内部超出一行的内容在标签内也缩进。

要了解缩进的重要性，最好的方法是看一些没有缩进的 HTML 代码。大家都知道这样的歌词：“you don't know what you've got [ ' ]til it's gone（直到失去才意识到曾经拥有）。”下面是没有缩进的简单表格代码：

```
<table>
<tr><td>Cell One</td><td>Cell Two</td></tr>
<tr><td>Cell Three</td><td>Cell Four</td></tr>
</table>
```

不仅没有缩进，且表格内的行和列也没有分开。现在，将这段代码与下列描述同一各表格的代码进行比较：

```
<table>
  <tr>
    <td>Cell One</td>
    <td>Cell Two</td>
  </tr>
  <tr>
    <td>Cell Three</td>
    <td>Cell Four</td>
  </tr>
</table>
```

对 HTML 代码的易读性和可维护性来说，一致的缩进风格甚至比注释还重要。你不一定要使用我的缩进策略，使用 3 个或 4 个空格，也是可以的。如果要使代码更紧凑，不缩进标签内的内容也没有问题。重要的是要形成自己的编码风格并坚持使用。

**提示：**如果你与其他人一起工作或计划与其他人一起开发网页，应考虑建立一个小组，制定一致的编码风格。

**Did you Know?**

## 23.5 总结

本章提供了一些示例和解释，帮助你将网页组织成一个信息丰富、引人入胜、易于导航的网站。网络用户已经非常精明，他们希望网站设计良好，如果他们觉得你的网页因设计粗劣而难以导航，就会马上抛弃你的网页。

本章还讨论了通过加入注释和缩进使 HTML 代码易于维护的重要性。注释不仅可以在以后重看代码时提示你，而且可以指导继承代码的人。缩进可能像是美学问题，但确实有助于

一看网页就能快速地分析和理解网页的结构。

## 23.6 问与答

**问：**我见过的一些网页，它们要求访问者进入主页前调整浏览器窗口宽度或其他设置。这是为什么？

**答：**尖锐的回答是，这些网站创建者不关心他们的用户。决不要强迫用户用不同的方式处理浏览器，尤其是不要总是自动改变浏览器的大小。这些都是最大的可用性禁止事项。当网站告诉你修改设置时，这是因为网站创建者认为如果可以稍微控制访问者的窗口或字体大小，他们可以提供更好的展示。当然，很少有人会改变其设置（他们也不应该改变），所以这些网站看起来比较奇怪或者难以阅读。最好使用在这本书学到的技巧使网站在任何窗口大小和各种浏览器设置中都是容易阅读的，且是吸引人的。网站组织得越好，对访问者来说就越有用。

**问：**浏览网页时，较多的注释和空格会使网页加载变慢吗？

**答：**网页中少量的额外文本相对于其他更大的网页资源（如图像）来说，是可以忽略不计的。另外，较慢的拨号调制解调器通常会很好地压缩传输的文本，所以加入空格调整 HTML 格式通常根本不会影响传输时间。加入数百个注释才会在网页载入时导致一秒的额外延迟。请切记，现在很多人都拥有宽带连接（电缆、DSL 等），文本传输是极其快速的。是图像使网页传输速度变慢，所以要尽量压缩图像（参阅第 10 章），而放心地使用文本注释。

## 23.7 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 23.7.1 测验

1. 哪 3 种方法可以帮助人们体会到你所有的网页组成一个连贯一致的网站？
2. 在人们首次访问的网站看到的主页中，应包含哪两种信息？
3. 如果想对网页的未来编辑说“Don't change this image of me. It's my only chance at immortality”（不要更换这幅图像，这是我不朽的唯一机会），但不希望浏览网页的人看到这种信息，该如何做？

### 23.7.2 测验答案

1.
  - (a) 使用一致的背景、颜色、字体和样式。
  - (b) 在所链接到的网页顶部都使用相同的链接文本或图像。



(c) 在网站每个网页中都使用相同的小标题、按钮或其他元素。

2.

(a) 足够的信息，让他们能够立刻知道网站的名称以及网站内容是哪方面的。

(b) 任何要传达给目标读者的重要信息，都要直接而简洁地表述出来。无论是使命声明还是营销口号，都要保证它们清晰地展示出来。

3. 在<img />标签前加入如下注释：

```
<!-- Don't change this image of me.  
      It's my only chance at immortality. -->
```

## 23.8 练习

- 拿一支铅笔（老式的那种）画出网站草图，草图由小矩形和它们之间的箭头组成。画出每个页面的草图，用弯线指出文本的位置，用涂鸦表示图像的位置。每个箭头从涂鸦图标开始，该图标对应于导航到箭头指向的网页的按钮。即使拥有最优秀的网站管理工具，但手工画出网站草图也可以让你非常直观地理解网站中哪些网页易于访问，以及相邻网页的布局如何协同工作，这些都应在投入时间开始编写实际HTML，将这些网页连接到一起之前完成。信不信由你，我开始设计网站时仍像这样画出网站草图。有时没有东西能胜过铅笔和纸！
- 打开标示你当前网站的HTML文件，检查注释和代码缩进。有哪些代码需要解释给将来可能的读者吗？如果有，添加解释性的注释。你是否难以说出代码的层次——是否难以看到标题和分段？如果这样，缩进HTML，使结构与层次匹配，从而使你能够很快地跳转到需要编辑的分段。





## 第 24 章

# 帮助人们找到你的网站

---

本章中你将学到：

- 宣传网站的方法
- 在主要搜索网站上列出网站的方法
- 网站搜索引擎优化的方法

网站只有被用户访问时才有用，如果没有人能够找到你的网站，你辛辛苦苦地创建就是白费。本章将介绍的 HTML 标签和技巧不会使网站在视觉方面不同，但这些标签和技巧非常重要，有助于目标用户更容易看到你的网站。对于大多数网站制作者来说，本章可能是最简单的，但却是最重要的。你将学习如何添加网站元素和构建网站架构，使人们搜索与你的话题或者公司相关的关键词时，增加搜索引擎返回你的网站链接的可能性，这被称为搜索引擎优化（SEO）。与你从希望向你销售服务的公司那里听到的不同，没有什么魔法可以使你出现在每个搜索列表的前面，但有许多免费的最佳方法可确保你的网站很容易被发现。

### 24.1 宣传网站

你肯定希望你的网站能够吸引别人的注意，否则你不会花功夫来创建它们。如果只是将网站放到局域网或公司内部网上，或者只是通过磁盘或电子邮件来发布网站，那么，帮助人们找到你的网站可能就不是问题。然而，如果将网站加入到 Internet 上浩如烟海的网站中，则让目标用户能够找到你的网站将是巨大的挑战。

要解决这个问题，需要对大多数人如何决定要浏览哪些网站有基本了解。下面是人们获知网站的 3 个基本途径。

- 有人告诉他们并提供网站的地址，他们可以直接在 Web 浏览器中输入该地址。
- 从其他人的链接进入你的网站。
- 从搜索引擎（如 Google、Yahoo! 或 Bing）的数据库中的索引找到你的网站。

如果花些时间和工夫，可使通过这 3 种途径找到你的网站的人更多。要使听说你的网站的人增多，需要通过口耳相传以及其他任何交流渠道。如果有联系人数据库或邮件列表，可通过它们将网站向这些人公布。将网站的地址添加到你的名片或公司的宣传册中。还有，如果有钱，可以利用电视和广播将你的 Internet 地址广而告之。总之，要进行市场营销，即使在 Internet 上，传统的口头营销仍是最好的方法。

增加指向你的网站的链接数量也很简单，但这并不意味着工作量很小。如果存在有关你的主题的专门目录，不管是在线的还是印刷的，要确保你的网站位列其中。参加社交网络，包括为你的服务或者业务实施 Facebook 的“爱好者”页面（如果适用）。创建一个 Twitter 账户来广播新闻并且与客户连接——同样，在适用于你的网站的情况下。进入你的客户可能存在的空间，比如评论你感兴趣的特定主题的博客，并且参加这些社区。这不是说你应该找到关于你的主题或者服务的一个论坛，并且向那里的用户发送指向你的网站的链接。要像你的领域中的专家那样，提供忠告和建议，加上你自己的网站 URL。在这方面本书中不能给你太多的帮助，需要你走出去做。

**注意：** Mashable (<http://www.mashable.com/>) 是一个非常流行、高流量并且深受尊敬的网站（这归功于它的准确和所增加的价值），提供在社交网络中交互的技巧，尤其适用于商业用户。

**By the Way**

我能帮助你的主要是确保内容为搜索引擎所收集并且正确地索引。这是个合理的假设：如果你的内容没有在 Google 的数据库中，那么你就有麻烦了。

搜索引擎基本上都是索引尽可能多的互联网内容（包括视频和富媒体）的巨大数据库。它们使用自动化过程来搜索网站，使用被称为机器人或者蜘蛛的程序来搜索内容页面并且建立数据库。在内容索引之后，搜索应用程序使用成熟的页面排序技术来确定在用户输入搜索词时，哪些内容首先显示。

搜索引擎处理用户查询时，它查找包含用户的搜索关键词和短语的内容。但是这不是一个像“如果这个页面包含这个短语，作为一个结果返回”那样的简单匹配，因为内容根据关键词和短语的出现频率和上下文排序，来自其他网站的链接数也为排序增加了可信度。本章将教你几种方法，以确保你的内容根据你所提供的内容和上下文正确地出现在搜索引擎中。

## 24.2 在主要的搜索网站列出你的网站

如果希望人们找到你的网站，就必须向每个主要的搜索网站提交请求，让它们来索引你的网站。尽管搜索引擎自动索引 Web 内容，但是这是确保你的网站存在于他们的网站的最佳方式。每个搜索网站都会提供一个表格，让你填写网站的地址及简单的介绍，有时候还要求网站相关的分类或关键词列表。这些表格很容易填写，一个小时就可以填完，省下的时间可用于将你的网站在一两个专门目录中列出（如何找到这些专门目录呢？当然是通过主要的搜索网站）。

### 列出你的网站之前

等一下！在你仓促地提交列表请求之前，先阅读本章的余下部分。否则，你将有很严重

的问题，而且可能已经错过解决问题的最佳时机。

为了理解我的意思，可以想象一下这种场景：你发布了一个销售自动灭蟑螂器的网站。我是一个遇到蟑螂问题的互联网用户，并且对灭蟑螂喷雾过敏。我打开自己的膝上型计算机，扫清键盘上的蟑螂，登录到所喜欢的搜索网站，并且输入搜索词“蟑螂”。搜索引擎立即显示了 10254 个包含蟑螂这个词的网站中的头十个。你已经提交了列表请求，所以你知道自己的页面在列表中的某个地方。

我有没有提到过我是个富人？我有没有提到过两只蟑螂正在我的脚上？你甚至提供在我的区域中的当日交付服务。你希望你的网站出现在列表中的第 3 位还是第 8542 位？好的，现在你理解了这个问题了。仅仅在搜索引擎中列出是不够的——你需要提升你的排名。

虽然在主要搜索引擎中列出很简单也很快，但也令人困惑：对于要注册网站必须单击的地方，每个搜索网站使用的术语各不相同。下面的列表可帮助减少困惑，它列出了一些免费地将你的网站包含在内的搜索引擎的地址以及注册时要单击的链接词。

- Google——访问 <http://www.google.com/addurl/>，输入网站的地址及简单介绍，然后输入页面上显示的花体验证文本（被称为 CAPTCHA，全自动区分计算机和人类的图灵测试）。最后单击 Add URL 按钮，将网站添加到 Google。
- Yahoo! Search——访问 <http://siteexplorer.search.yahoo.com/submit>，输入网站的地址，然后单击 Submit URL 按钮。
- Bing——访问 <http://www.bing.com/doc/submit.aspx>，输入验证文本，输入网站的地址，然后单击 Submit URL 按钮。
- AllTheWeb——AllTheWeb 的搜索结果是由 Yahoo! Search 提供的，因此只要将网站提交给 Yahoo! Search，如上所述。
- Alta Vista——Alta Vista 的搜索结果也是由 Yahoo! Search 提供的，因此只要将网站提交给 Yahoo! Search，如上所述。

### Did you know?

**提示：**有些网站提供可自动提交给所有主要搜索引擎和一些小型搜索引擎的表单，如 <http://www.scrubtheWeb.com/>、<http://www.submitexpress.com/> 和 <http://www.hypersubmit.com/>。在这些网站当中，有许多也提供将网站列在其他许多目录和索引下的收费服务。这些服务对你来说可能有价值也可能没有，这取决于你的目标用户，但我强烈推荐你直接到前面列出的主要搜索网站上，使用它们的表单来提交请求。这样可以确保准确地回答问题（每个网站的问题都有点不一样），并确保你确切地知道网站将如何列在每个网站中。

## 24.3 为搜索引擎提供提示

事实上，你绝对无法确保在主要搜索引擎中使用特定的单词或词组进行搜索时，自己的网站出现在搜索结果的前十名中，也就是说，不能从搜索网站购买广告空间。毕竟，如果可以的话，每个人都希望自己的网站列在第一名。你能够做的只是避免自己的网站显示在列表的最后，使自己与其他人有同等的机会出现在最前面。这被称为搜索引擎优化（SEO），也就是优化网站的内容和结果使搜索引擎更喜欢你的网站。

每个搜索引擎都使用不同的方法来确定哪些网站是最为相关的，因此应列在搜索结果列表的前面。但你不必太关心它们的区别，因为它们都使用一些基本的标准组合。下面的列表包含了几乎每个搜索引擎在评价哪些网站与关键词最匹配时都会考虑的因素。

- 关键词是否出现在页面的<title>标签中？
- 关键词是否出现在页面的前几行中？
- 关键词是否出现在页面的<meta />标签中？
- 关键词是否出现在页面的<h1>标题中？
- 关键词是否出现在页面的图像文件名和 alt 文本中？
- 网站中有多少页面链接到该页面？
- 其他网站中有多少页面链接到该页面？又有多少其他页面链接到这些页面？
- 在上一次搜索列表结果中该页面被选中了多少次？

很显然，为将排名提前，最重要的工作是考虑目标用户最可能输入的关键词组合。建议不要关心常用的单个单词（如“食物”）的检索——在这种情况下，生成的列表通常很长，试图使网站出现在前面就像买彩票一样。相反，应将重点放在与你的主题相关相关性最强的、不常用词和两三个词的组合上（如用“南方家常菜”来代替“食物”）。确保这样的单词和词组多次出现在页面中，并将最重要的词放在<title>标签以及第一个标题或介绍性段落中。

**注意：**有些激进的网站制作者在页面中将同一个单词重复几十个甚至上百次，有时候以很小的字体或难以看见的颜色显示，这样做只是为了使搜索引擎在该关键词的搜索结果中将该页面显示在前面。这种做法称为搜索引擎欺骗（spamming）。

不要做这种事情，所有主要的搜索引擎都反对这种做法，它们设置了“欺骗探测器”，检查以可疑的模式重复相同的词或词组的页面，一旦发现立刻将其从数据库中删除。页面中多次出现重要的搜索关键词是可以的（也有好处），但一定要在正常的句子或词组中使用这些关键词，这样才不会被欺骗探测策略删除。

**By the  
Way**

在上面列出的所有搜索引擎评估标准中，误解最多的可能是<meta />标签。有些人以为使用它们，页面就立即能够出现在每个搜索列表的前面。另一些人却认为<meta />标签没什么用。这两种极端想法都是错误的。

<meta />标签是一个通用标签，可放在任何文档的<head>部分中，用于指定关于页面的不属于<body>文本的信息。大多数主要搜索引擎都查找<meta />标签是否提供了对页面的简单介绍和标识网页的一些关键词。例如，你的自动灭蟑螂器订单表单可能包括下列两个标签：

```
<meta name="description"
content="Order form for the SuperSquish cockroach flattener." />
<meta name="keywords"
content="cockroach, roaches, kill, squish, supersquish" />
```

**警告：**<meta />应放在<head>、<title>和</title>之后，但在结束标签</head>之前。根据 XHTML 标准，在每个文档的<head>部分中，<title>必须是第一个标签。

**Watch  
Out!**

在这个例子中，第一个标签确保搜索引擎将网站显示在搜索结果列表中时，对网站有准确的描述。第二个<meta />标签使得搜索查询中包含指定的关键词时，你的网站排名将靠前一点。

要使网站被搜索引擎索引，应在页面中包含<meta />标签，并指定属性 name="description" 和 name="keywords"。这样做对网站在搜索列表中的排名可能不会有很显著的效果，也不是所有的搜索引擎都会查找<meta />标签，但可能有所帮助。

### Did you Know?

**提示：**搜索引擎专家建议，在<meta />标签中，网页描述的理想长度是 100 ~ 200 个字符。对于关键词，推荐长度是 200 ~ 400 个字符。专家还建议不要在关键词之间浪费空格，这在蟑螂的例子中很明显。最后，不要疯狂地重复同一个关键词——如果做得过头，一些搜索引擎会惩罚你。

程序清单 24.1 所示的页面是一个说明如何改进搜索引擎结果的具体例子。

这个页面应该很容易找到，因为它涉及一个具体的话题，并多次包含一些不常见的技术术语，让对这个主题感兴趣的人搜索。然而，还可以采取一些措施，以提高网站在搜索结果列表中的排名。

### Did you Know?

**提示：**如果不希望网站包含在搜索引擎的数据库中（这种可能性不大），可将如下<meta />标签放在页面的<head>部分中：

```
<meta name="robots" content="noindex, noindex" />
```

这样会使有些搜索引擎忽略你的页面。要使用更健壮的保护措施避过搜索机器人的眼睛，可以请 Web 服务器管理员将你的网站地址包含在服务器的 robots.txt 文件中（他知道怎么做）。这样所有主要搜索引擎都会忽略你的网站，这适用于不想被公开搜索的公司内部网站。

## 程序清单 24.1 在互联网网站搜索中可见性较差的一个页面

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Fractal Central</title>
  </head>

  <body>
    <div style="text-align:center">
      
    </div>
    <div style="width:133px; float:left; padding:6px; text-align:center;
border-width:4px; border-style:ridge">
      Discover the latest software, books and more at our online store.<br />
      <a href="orderform.html"></a>
    </div>
    <div style="float:left; padding:6px">
      <h2>A Comprehensive Guide to the<br />
      Art and Science of Chaos and Complexity</h2>
      <p>What's that? You say you're hearing about "fractals" and "chaos" all
```



```
over the place, but still aren't too sure what they are? How about a
quick summary of some key concepts:</p>
<ol>
  <li><p>Even the simplest systems become deeply complex and richly
beautiful when a process is "iterated" over and over, using the
results of each step as the starting point of the next. This is how
Nature creates a magnificently detailed 300-foot redwood tree from a
seed the size of your fingernail.</p></li>
  <li><p>Most "iterated systems" are easily simulated on computers,
but only a few are predictable and controllable. Why? Because a tiny
influence, like a "butterfly flapping its wings," can be strangely
amplified to have major consequences such as completely changing
tomorrow's weather in a distant part of the world.</p></li>
  <li><p>Fractals can be magnified forever without loss of detail, so
mathematics that relies on straight lines is useless with them.
However, they give us a new concept called "fractal dimension" which
can measure the texture and complexity of anything from coastlines to
storm clouds.</p></li>
  <li><p>While fractals win prizes at graphics shows, their chaotic
patterns pop up in every branch of science. Physicists find beautiful
artwork coming out of their plotters. "Strange attractors" with
fractal turbulence appear in celestial mechanics. Biologists diagnose
"dynamical diseases" when fractal rhythms fall out of sync. Even pure
mathematicians go on tour with dazzling videos of their
research.</p></li>
</ol>
<p>Think all these folks may be on to something?</p>
</div>
<div style="text-align:center">
  <a href="http://netletter.com/nonsense/"></a>
</div>
</body>
</html>
```

现在比较程序清单 24.1 与修改过的程序清单 24.2，这两个页面看上去几乎一样，然而，对于搜索机器人和搜索引擎来说，这两个页面非常不同。下面的列表总结了其中的修改以及这样修改的原因。

- 在页面的<title>标签和第一个标题中添加了一些重要的搜索关键词。原来的页面甚至没有在这两个关键的地方包含单词 `fractal`。
- 添加了<meta />标签，为搜索引擎提供描述和关键词。
- 在第一个<img />标签中添加了极具描述性的 `alt` 属性。不是所有搜索引擎都会读取并索引 `alt` 文本，但的确有一些搜索引擎这样做。
- 删除了技术术语（如“`fractal`”和“`iterated`”）的引号，因为有些搜索引擎认为“`fractal`”和 `fractal` 是两个不同的单词。使用字符实体`&quot;`来表示引号，搜索引擎不会考虑这个字符实体。这是个好主意，因为 XHTML 鼓励 Web 开发人员使用`&quot;`实体而不是直接使用引号。
- 在订单表单框的文本中两次添加了关键词 `fractal`。

无法量化人们在搜索 `fratal` 和 `chaos` 时，能够找到程序清单 23.2 和程序清单 23.1 所示页面的概率差，但这些修改肯定能够提高搜索引擎找到该页面的可能性。通常来说，针对搜索引擎所做的改进也可能使页面的主题对于人来说更容易理解。这使得针对搜索引擎的网站优



化是双赢的!

## 程序清单 24.2 对程序清单 24.1 的改进

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>Fractal Central: A Guide to Fractals, Chaos, and Complexity</title>
    <meta name="description" content="A comprehensive guide to fractal
      geometry, chaos science and complexity theory." />
    <meta name="keywords" content="fractal,fractals,chaos science,chaos
      theory,fractal geometry,complexity,complexity theory" />
  </head>

  <body>
    <div style="text-align:center">
      
    </div>
    <div style="width:133px; float:left; padding:6px; text-align:center;
      border-width:4px; border-style:ridge">
      Discover the latest fractal software, books and more at the
      <span style="font-weight:bold">Fractal Central</span> online store.<br />
      <a href="orderform.html"></a>
    </div>
    <div style="float:left; padding:6px">
      <h2>A Comprehensive Guide to Fractal Geometry,<br />
      Chaos Science, and Complexity Theory</h2>
      <p>What's that? You say you're hearing about &quot;fractals&quot; and
      &quot;chaos&quot; all over the place, but still aren't too sure what
      they are? How about a quick summary of some key concepts:</p>
      <ol>
        <li><p>Even the simplest systems become deeply complex and richly
          beautiful when a process is &quot;iterated&quot; over and over, using
          the results of each step as the starting point of the next. This is
          how Nature creates a magnificently detailed 300-foot redwood tree from
          a seed the size of your fingernail.</p></li>

        <li><p>Most &quot;iterated systems&quot; are easily simulated on
          computers, but only a few are predictable and controllable. Why?
          Because a tiny influence, like a &quot;butterfly flapping its
          wings,&quot; can be strangely amplified to have major consequences
          such as completely changing tomorrow's weather in a distant part of
          the world.</p></li>

        <li><p>Fractals can be magnified forever without loss of detail, so
          mathematics that relies on straight lines is useless with them.
          However, they give us a new concept called &quot;fractal
          dimension&quot; which can measure the texture and complexity of
          anything from coastlines to storm clouds.</p></li>

        <li><p>While fractals win prizes at graphics shows, their chaotic
          patterns pop up in every branch of science. Physicists find beautiful
          artwork coming out of their plotters. &quot;Strange attractors&quot;
          with fractal turbulence appear in celestial mechanics. Biologists
          diagnose &quot;dynamical diseases&quot; when fractal rhythms fall out
          of sync. Even pure mathematicians go on tour with dazzling videos of
          their research.</p></li>
      </ol>
      <p>Think all these folks may be on to something?</p>
    </div>
  </body>
</html>
```

```
</div>
<div style="text-align:center">
  <a href="http://netletter.com/nonsense/"></a>
</div>
</body>
</html>
```

这些修改最终将使这个网站的内容更可能被正确地索引。除了易于索引之外，还要记住内容的质量——以及其他链接到你的网站——也是很重要的。

## 24.4 更多搜索引擎优化技巧

我能够给你的关于搜索引擎优化的最重要提示是，如果 SEO 公司承诺确定的结果，不要雇用他们来实施你的 SEO 工作。如果公司承诺你的网站将是 Google 搜索结果中的第一位，躲开他们，保护好你的支票本——没有人能保证这种结果，因为搜索算法有许多变量，在一个星期中第一位可能多次变化。这并不是说所有 SEO 公司都是骗子，一些进行 SEO 工作的合法的网站内容和架构顾问被和发送主动的 E-mail 的垃圾邮件制造者混为一谈，这种主动的 E-mail 类似这样：

“亲爱的 google.com,

我访问了你的网站并且注意到，你没有在大部分搜索引擎和目录中被列出……”

这个示例邮件被 Google 作为网站管理者指南中的一个示例，并加上了这样的注解：“对关于搜索引擎的主动邮件，像你对待燃烧脂肪的夜间节食药或者帮助被罢免的独裁者转移资金的请求一样，做个怀疑论者”。是的，有人确实给 Google 发了一封垃圾邮件，关于如何在 Google 中增进它们的搜索排名。更多好的建议，可以访问 <http://www.google.com/Webmasters/>。

下面是你可以采取的其他一些措施，这些措施是免费的，可以在搜索引擎中优化你的内容。

- 使用准确的页面标题。你的标题应该简短，但是具有描述性并且是独特的。不要尝试用关键词填充你的标题。
- 创建人类友好的 URL，比如使用用户容易记忆的词语。[http://www.mycompany.com/products/super\\_widget.html](http://www.mycompany.com/products/super_widget.html) 比起 <http://www.mycompany.com?c=p&id=4&id=49f8sd7345fea> 要容易记忆得多，也更容易让搜索引擎以相关的方式索引。
- 创建反映你的目录结构的 URL。这首先假设你具有应有的目录结构。
- 在可能的情况下，使用文本而不是图形元素作为导航。
- 如果有好几级的内容，使用面包屑轨迹，这样用户能够找到回家的路。面包屑轨迹还提供给搜索引擎更多索引词。例如，如果你关注一个关于食物的网站的南方菜类别中的饼干的烹饪方法，这个特定的页面的面包屑轨迹可能是这样：  
主页〉南方菜〉烹饪方法〉饼干。
- 在页面的内容中，正确使用标题（<h1>、<h2>、<h3>）。

除了为用户提供丰富和有用的内容，你应该按照这些提示改进网站在页面排名中的地位。

## 24.5 总结

本章介绍了一些极其重要的主题：如何提供提示给搜索引擎（如 Google、Bing、Yahoo!），让人们能够更容易地在 Internet 找到你的网站。你还看到了一个完全合理的网页 HTML 重整的示例，这使页面对搜索引擎更加友好。最后，你学习了另外一些优化网站索引的技巧。

表 24.1 列出了本章介绍的标签和属性。

表 24.1 第 24 章介绍的标签与属性

标签/属性	功能
<meta />	指定文档的元信息（关于文档本身的信息）。最常见的用途是加入页面描述、指定关键词。在文档的<head>中使用
属性	
name="名称 "	用于指定 content 属性中包含关于文档的哪类信息。例如 name="key-words"表示 content 中包含页面的关键词
content="值 "	http-equiv 或 name 中指定的信息类型的实际信息或值。例如，如果 http-equiv 属性设置为 refresh，content 属性应该设置为要等待的秒数，接着是分号以及要载入的网站地址

## 24.6 问与答

**问：**我的网站有许多页面，我需要在搜索网站为每个页面填写单独的表格吗？

**答：**不需要。如果只提交主页（它可以链接到所有其他页面），搜索蜘蛛将搜索这个页面的所有链接（还有被链接页面的所有链接，这样不断重复下去），直到它将网站中所有页面都编入索引为止。

**问：**我提交了在搜索引擎中列出网站的请求，但即使我输入公司的特有名称，网站依然从未出现。我该怎么办呢？

**答：**大多数大型搜索引擎提供一个表格，填写后你可以立即查询特定地址是否包含在其数据库中。如果发现还没有，可以再提交申请。有时要在提交申请几天甚至几星期后，网络蜘蛛才将你的网站编入索引。

**问：**在<meta />标签中加入关键词时，需要包含每种可能的拼写或大小写吗？

**答：**不必担心大小写，几乎所有的搜索都是以小写输入的。然而，要包含所有明显的变体或常见拼写错误，将其作为关键词。虽然从理论上说很简单，但在利用<meta />标签时，还有很多本章没有介绍的高级策略。关于<meta />标签的各种属性以及用法请访问 [http://en.wikipedia.org/wiki/Meta\\_element](http://en.wikipedia.org/wiki/Meta_element)。

**问：**可以使用<meta />标签使页面每隔几秒或几分钟自动地刷新自己吗？

**答：**可以，但通常不需要这样做，除非在 Web 服务器上设置了某种程序或脚本为页面提供新信息。如果出现了那种情况，你可以使用 AJAX 以不同的方式进行这种刷新（AJAX 的基本信息参见第 21 章）。由于可用性的原因，W3C 和用户不赞成使用<meta />来刷新内容。

## 24.7 作业

作业包含一些测试题及练习，帮助巩固对本章介绍的内容的理解。试着回答所有问题，然后再看后面的答案。

### 24.7.1 测验

1. 如果发布了一个关于小狗收养 (puppy adoption) 的网站，如何确保在主要的 Internet 搜索网站输入 puppy、dog 与/或 adoption 时可以找到该网站？
2. 假设你决定在你的 HTML 代码中将关键词粘贴几百次，在白色背景上使用白色的字体，使用户不能看到它们。搜索引擎蜘蛛会如何处理？
3. 将所有内容放到一个目录中，还是将它组织为多个目录更好？

### 24.7.2 测验答案

1. 首先，确保 puppy、dog、adoption 经常出现在主页中（你可能已经这样做了），并使 Puppy Dog Adoption 出现在页面标题中。然后，在页面的 <head> 部分加入下列 <meta /> 标签：

```
<meta name="description"
content="dog adoption information and services" />
<meta name="keywords" content="puppy, dog, adoption" />
```

2. 搜索引擎将忽略这些重复，并且可能将你列入垃圾制造者的黑名单。
3. 绝对是将内容组织到目录中更好。这将提供更简便的维护，而且使你有机会利用有意义的目录结构创建人类可读的 URL，还将创建导航面包屑轨迹。

## 24.8 练习

- 你已经到达本章的末尾。如果你已经有了一个网站，重新审视内容和结构以进行最大可能的优化，然后将地址提交给所有主要搜索引擎。



## 附录 A

# 完整的 XHTML 1.1 和 CSS 2 快速参考

XHTML1.1 将 HTML 作为一种 XML 应用进行重新表述，使这种语言的扩展更加容易定义和实现。本附录提供你最可能看到和使用的 XHTML1.1 元素和属性，以及 CSS2 的样式属性的快速参考。完整的规范可访问 <http://www.w3.org/>。

为方便读者查找信息，本附录根据功能将 HTML 元素放在不同的小节中，并按如下顺序排列各个小节。

- 结构
- 文本短语和段落
- 文本格式化元素
- 列表
- 链接
- 表格
- 嵌入内容
- 样式
- 表单
- 脚本

在每节中，元素按字母顺序排列，并提供如下信息。

- 用法：元素的总体描述。
- 开始/结束标签：指出标签是必需的、可选还是非法的。
- 属性：列出元素的属性，并简要说明其作用。用于鼠标控制或者调用客户端脚本的属性在这里没有指出，这些面向操作的属性请参见 W3C 网站上的完整规范。
- 是否可为空：指出元素是否可以为空。
- 备注：使用元素时需要特别考虑的因素。

CSS 样式属性以类似的方式排列，只是其中包含的是可接受的值，而不是属性。

### **By the Way**

**注意：**XHTML 1.1 包含几个基本属性，可用于很多元素中。在每个元素中，这些属性被称为属性组 core、i18n 和 events。列出所有 XHTML 元素后，将详细说明这些属性组，指出每个属性组包含哪些属性。

## A.1 XHTML 结构

XHTML 依靠几个元素来定义文档结构（与结构化文档内的文本相对），并提供给浏览器或搜索引擎使用的信息。

### 注释<!--...-->

用法	用于插入注释或脚本，浏览器不显示它们
开始/结束标签	必需/必需
属性	无
是否可为空	是
备注	注释不限于一行，而是可以为任意长。结束标签不必与开始标签位于同一行

### <!doctype>

用法	版本信息出现在 HTML 文档的第一行，它是 SGML 声明而不是元素
----	-------------------------------------

### <body>...</body>

用法	包含文档的内容
开始/结束标签	可选/可选
属性	core、i18n 和 events onload="eventcode": 文档加载时触发的内部事件 onunload="eventcode ": 文档卸载时触发的内部事件
是否可为空	否
备注	只能有一个<body>，必须紧跟在<head>的后面。如果你使用框架（请不要这么做），<body>元素可由<frameset>元素代替

### <div>...</div>

用法	用于给文本块添加结构
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否
备注	不能在 p 元素内使用

**注意：**你可能遇到这样的 HTML 网页，使用带 align 属性的<div>元素。在 XHTML 和 HTML 中，该属性已被删除，并使用 CSS 样式属性 text-align 来对齐文本。该样式属性将在本附录后面介绍。



**<h1>...</h1>到<h6>...</h6>**

用法	6 种标题 (h1 是最重要的), 用于在 body 中以层次方式组织信息
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否
备注	浏览器根据标题的重要程度使用不同的字号, <h1>最大, 而 <h6>最小

**<head>...</head>**

用法	这是文档头, 包含向用户和搜索引擎提供信息的其他元素
开始/结束标签	可选/可选
属性	i18n profile="url": 指定元数据位置的 URL
是否可为空	否
备注	每个文档只能有一个<head>, 必须在开始标签<html>后面, 且在<body>的前面

***Did you know?***

**提示:** profile 属性在 HTML5 中不允许。

**<hr />**

用法	用于分隔网页各部分的水平线
开始/结束标签	必需/非法
属性	core、i18n 和 events
是否可为空	是

**<html>...</html>**

用法	html 元素包含整个文档
开始/结束标签	可选/可选
属性	i18n
是否可为空	否
备注	声明<!doctype...>中包含版本信息, 因此在这里不是必需的

**<meta />**

用法	提供有关文档的信息
开始/结束标签	必需/非法
属性	i18n http-equiv="servercmd": HTTP 响应报头的名称

name="name": 元信息的名称  
 content="value": 元信息的内容  
 scheme="scheme": 指定用于解释元数据的模式

是否可为空 是

**提示:** scheme 属性在 HTML5 中不允许。

**Did you  
Know?**

### <span>...</span>

用法 通过定义文本跨度来组织文档  
 开始/结束标签 必需/必需  
 属性 core、i18n 和 events  
 是否可为空 否

### <title>...</title>

用法 给网页指定的名称。<title>标签位于在<head>标签中，其内容显示在浏览器窗口的标题栏中  
 开始/结束标签 必需/必需  
 属性 i18n  
 是否可为空 否  
 备注 每个文档只能有一个标题

## A.2 XHTML 文本短语和段落

可根据特定的用途结构化文本短语（块），如创建段落。不应将其同修改文本格式混为一谈。

### <blockquote>...</blockquote>

用法 用于显示长引用  
 开始/结束标签 必需/必需  
 属性 core、i18n 和 events  
 cite="url": 要引用的文本的地址  
 是否可为空 否

### <br />

用法 强行换行  
 开始/结束标签 必需/非法  
 属性 core、i18n 和 events  
 是否可为空 是

**<cite>...</cite>**

用法	引用参考
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

**<code>...</code>**

用法	标识要显示的代码段
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

**<h1>...</h1>到<h6>...</h6>**

用法	文本标题
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

**<p>...</p>**

用法	定义段落
开始/结束标签	必需/可选
属性	core、i18n 和 events
是否可为空	否

**<pre>...</pre>**

用法	显示预先格式化的文本
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

**<strong>...</strong>**

用法	更强的强调
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

**<sub>...</sub>**

用法	创建下标
开始/结束标签	必需/必需



属性	core、i18n 和 events
是否可为空	否
<b>&lt;sup&gt;...&lt;/sup&gt;</b>	
用法	创建上标
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

### A.3 XHTML 文本格式化元素

文本特征如字号、粗细和样式可使用这些元素来修改，但更好的方法是使用 CSS 样式属性。本附录后面包含有关这些属性的完整参考，这些属性让你能够对文本进行难以置信的控制。

**<b>...</b>**

用法	粗体文本
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

**<big>...</big>**

用法	大文本
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

**提示：**这个元素在 HTML5 中已被删除，因为它纯粹用于表现，因而用 CSS 处理更好。

**Did you Know?**

**<i>...</i>**

用法	斜体文本
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

**<small>...</small>**

用法	小文本
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

**<tt>...</tt>**

用法	电传打字机（等宽）文本
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

**Did you know?**

**提示：**这个元素在 HTML5 中已被删除，因为它纯粹用于表现，因而用 CSS 处理更好。

## A.4 XHTML 列表

通过创建列表，可以将文本组织成结构化程度更高的大纲。列表可嵌套。

**<dd>...</dd>**

用法	在<dl>（定义列表）元素中使用的定义描述
开始/结束标签	必需/可选
属性	core、i18n 和 events
是否可为空	否
备注	可包含块级的内容，如<p>元素

**<dl>...</dl>**

用法	创建定义列表
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否
备注	必须包含至少一个<dt>或<dd>元素

**<dt>...</dt>**

用法	<dl>（定义列表）中使用的定义项
开始/结束标签	必需/可选
属性	core、i18n 和 events
是否可为空	否
备注	必须包含文本（文本用文本标记元素进行修改）

**<li>...</li>**

用法	定义列表中的列表项
开始/结束标签	必需/可选
属性	core、i18n 和 events
是否可为空	否



**<ol>...</ol>**

用法	创建有序列表
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否
备注	必须包含至少一个列表项

**<ul>...</ul>**

用法	创建无序列表
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否
备注	必须包含至少一个列表项

## A.5 XHTML 链接

超级链接是 XHTML 的基础。这些元素让你能够链接到其他文档、当前文档的其他地方或外部文件。

**<a>...</a>**

用法	用于定义链接和锚
开始/结束标签	必需/必需
属性	core、i18n 和 events charset="encoding": 源的字符编码 name="name": 定义锚 href="linkurl": 链接源的 URL rel="linktype": 前向链接类型 rev="linktype": 逆向连接类型 shape="value": 让你能够使用定义的形状 (default、rect、circle、poly) 定义客户端图像映射 coords="values": 使用像素或百分比设置形状的大小
是否可为空	否

**提示:** charset、name、rev、shape 和 coords 属性在 HTML 5 中不允许使用。

**Did you know?**

**<base />**

用法	文档中的其他所有 URL 都相对于这个位置进行解析
----	---------------------------



开始/结束标签	必需/非法
属性	href="linkurl": 链接源的 URL
是否可为空	是
备注	位于文档的<head>中
<hr/>	
<b>&lt;link/&gt;</b>	
<hr/>	
用法	定义链接和源之间的关系
开始/结束标签	必需/非法
属性	core、i18n 和 events charset="encoding": 源的字符编码 href="linkurl": 源的 URL rel="linktype": 前向链接类型 rev="linktype": 逆向连接类型 type="contenttype": Internet 内容类型 media="media": 定义目标介质(screen、print、projection、braille、speech、all) target="placement": 指定在哪里显示源(用户定义的名称、blank、parent、self、top)
是否可为空	是
备注	位于文档的<head>中

**Did you know?**

**提示:** charset、rev 和 target 属性在 HTML 5 中不允许使用。

## A.6 XHTML 表格

表格是一种以表格方式显示数据的方法。在 XHTML 之前, 表格被广泛地用于页面布局, 但随着样式单的出现, W3C 和本书的作者不再鼓励这样使用表格。

**<caption>...</caption>**

用法	显示表格标题
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否
备注	可选的

**<col />**

用法	为共享属性值将列编成组
----	-------------

开始/结束标签	必需/非法
属性	core、i18n 和 events <span>span="numcols": 组包含的列数</span> <span>width="width": 列宽, 可以是百分比、像素数或最小值</span> <span>align="alignment": 水平对齐单元格的内容 (left、center、right、justify、char)</span> <span>char="charalignment": 设置列与之对齐的字符</span> <span>charoff="charoffset": 一行中到第一个对齐字符的偏移量</span> <span>valign="verticalalignment": 垂直对齐单元格的内容 (top、middle、bottom、baseline)</span>
是否可为空	是

**提示:** width、align、char、charoff 和 valign 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

**Did you Know?**

#### <colgroup>...</colgroup>

用法	定义一个列组
开始/结束标签	必需/非法
属性	core、i18n 和 events <span>span="numcols": 组中包含的列数</span> <span>width="width": 列宽</span> <span>align="alignment": 水平对齐单元格的内容 (left、center、right、justify、char)</span> <span>char="charalignment": 设置列与之对齐的字符</span> <span>charoff="charoffset": 一行中到第一个对齐字符的偏移量</span> <span>valign="verticalalignment": 垂直对齐单元格的内容 (top、middle、bottom、baseline)</span>
是否可为空	否

**提示:** width、align、char、charoff 和 valign 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

**Did you Know?**

#### <table>...</table>

用法	创建表格
开始/结束标签	必需/必需
属性	core、i18n 和 events <span>width="width": 表宽</span>

`cols="numcols"`: 列数

`border="borderwidth"`: 表格边框的宽度 (单位为像素)

`frame="frame"`: 设置表格边框的可视性 (`void`、`above`、`below`、`hsides`、`lhs`、`rhs`、`vsides`、`box`、`border`)

`rules="rules"`: 设置表格内的可视规则 (`none`、`group`、`rows`、`cols`、`all`)

`cellspacing="cellspacing"`: 单元格之间的间距

`cellpadding="cellpadding"`: 单元格内的间距

`summary="description"`: 提供表格的文本描述, 以方便使用

是否可为空

否

### Did you Know?

**提示:** `width`、`align`、`border`、`frame`、`rules`、`cellspacing`、`cellpadding` 和 `summary` 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

### <tbody>...</tbody>

用法

定义表体

开始/结束标签

可选/可选

属性

`core`、`i18n` 和 `events`

`align="alignment"`: 水平对齐单元格的内容 (`left`、`center`、`right`、`justify`、`char`)

`char="charalignment"`: 设置列与之对齐的字符

`charoff="charoffset"`: 一行中到第一个对齐字符的偏移量

`valign="verticalalignment"`: 垂直对齐单元格的内容 (`top`、`middle`、`bottom`、`baseline`)

是否可为空

否

### Did you Know?

**提示:** `align`、`char`、`charoff` 和 `valign` 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

### <td>...</td>

用法

定义单元格的内容

开始/结束标签

必需/可选

属性

`core`、`i18n` 和 `events`

`abbr="name"`: 缩写的名称

`axis="axisnames"`: 列出单元格对应的行头和列头

`rowspan="numrows"`: 单元格跨越的行数

`colspan="numcols"`: 单元格跨越的列数

`align="alignment"`: 水平对齐单元格的内容 (left、center、right、justify、char)

`char="charalignment"`: 设置列与之对齐的字符

`charoff="charoffset"`: 一行中到第一个对齐字符的偏移量

`valign="verticalalignment"`: 垂直对齐单元格的内容 (top、middle、bottom、baseline)

`headers="headers"`: 单元格的表头信息

`scope="scope"`: 指出单元格是否为其他单元格提供表头信息

是否可为空

否

**提示:** `axis`、`align`、`char`、`charoff`、`valign` 和 `scope` 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

***Did you Know?***

`<tfoot>...</tfoot>`

用法

定义表格脚注

开始/结束标签

必需/可选

属性

`core`、`!l8n` 和 `events`

`align="alignment"` 水平对齐单元格的内容 (left、center、right、justify、char)

`char="charalignment"`: 设置列与之对齐的字符

`charoff="charoffset"`: 一行中到第一个对齐字符的偏移量

`valign="verticalalignment"`: 垂直对齐单元格的内容 (top、middle、bottom、baseline)

是否可为空

否

**提示:** `align`、`char`、`charoff` 和 `valign` 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

***Did you Know?***

`<th>...</th>`

用法

定义表头单元格的内容

开始/结束标签

必需/可选

属性

`core`、`!l8n` 和 `events`

`abbr="name"`: 缩写的名称

`axis="axisnames"`: 列出单元格对应的行头和列头

`rowspan="numrows"`: 单元格跨越的行数

`colspan="numcols"`: 单元格跨越的列数

`align="alignment"`: 水平对齐单元格的内容 (left、center、right、



justify、char)

char="charalignment": 设置列与之对齐的字符

charoff="charoffset": 一行中到第一个对齐字符的偏移量

valign="verticalalignment": 垂直对齐单元格的内容(top、middle、bottom、baseline)

headers="headers": 单元格的表头信息

scope="scope": 指出单元格是否为其他单元格提供表头信息

是否可为空

否

**Did you Know?**

**提示:** axis、align、char、charoff、valign 和 scope 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

**<thead>...</thead>**

用法

定义表头

开始/结束标签

必需/可选

属性

core、i18n 和 events

align="alignment" 水平对齐单元格的内容(left、center、right、justify、char)

char="charalignment": 设置列与之对齐的字符

charoff="charoffset": 一行中到第一个对齐字符的偏移量

valign="verticalalignment": 垂直对齐单元格的内容(top、middle、bottom、baseline)

是否可为空

否

**Did you Know?**

**提示:** align、char、charoff 和 valign 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

**<tr>...</tr>**

用法

定义表格中一行单元格

开始/结束标签

必需/可选

属性

core、i18n 和 events

align="alignment" 水平对齐单元格的内容(left、center、right、justify、char)

char="charalignment": 设置列与之对齐的字符

charoff="charoffset": 一行中到第一个对齐字符的偏移量

valign="verticalalignment": 垂直对齐单元格的内容(top、middle、bottom、baseline)

是否可为空	否
-------	---

**提示:** align、char、charoff 和 valign 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

**Did you know?**

## A.7 XHTML 嵌入内容

也被称为包含 (inclusion), 嵌入内容适用于图像、图像映射、Java 小程序、Flash 动画以及其他多媒体或编程内容, 它们放在网页中提供额外功能。

### <area />

用法	用于定义链接和锚
开始/结束标签	必需/非法
属性	core、i18n 和 events shape="value": 让你能够使用定义的形状 (default、rect、circle、poly) 定义客户端图像映射 coords="values": 使用像素或百分比设置形状的大小 href="linkurl": 链接源文档的 URL nohref="nohref": 表示区域没有动作 alt="alttext": 替代文本
是否可为空	是

### <img />

用法	在文档中包含一幅图像
开始/结束标签	必需/非法
属性	core、i18n 和 events src="sourceurl": 图像的 URL alt="alttext": 替代文本 width="height": 图像的宽度 height="width": 图像的高度 border="border": 边框宽度 hspace="horizontal space": 图像同其他内容的水平间距 vspace="vertical space": 图像同其他内容的垂直间距 usemap="mapurl": 指向客户端图像映射的 URL ismap="ismap": 指定服务器端图像映射
是否可为空	是



**Did you Know?**

**提示:** `hspace` 和 `vspace` 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

**<map>...</map>**

用法	与<area>元素一起使用时, 创建客户端图像映射
开始/结束标签	必需/必需
属性	core、i18n 和 events <code>name="name"</code> : 要创建的图像映射的名称
是否可为空	否

**<object>...</object>**

用法	包含一个对象
开始/结束标签	必需/必需
属性	core、i18n 和 events <code>declare="declare"</code> : 声明但不创建对象的标记 <code>classid="objecturl"</code> : 对象的 URL <code>codebase="codebaseurl"</code> : 用于解析其他属性指定的 URL 的 URL <code>data="dataurl"</code> : 对象数据的 URL <code>type="datatype"</code> : 数据的 Internet 内容类型 <code>codetype="codetype"</code> : 代码的 Internet 内容类型 <code>standby="waitmsg"</code> : 加载时显示消息 <code>height="height"</code> : 图像的高度 <code>width="width"</code> : 图像的宽度 <code>border="border"</code> : 在对象周围显示边框 <code>hspace="horizontal space"</code> : 对象边界同其他其他页面内容之间的间距 <code>vspace="vertical space"</code> : 对象的顶部和底部同其他页面内容之间的间距 <code>usemap="mapurl"</code> : 图像映射的 URL <code>shape="shapes"</code> : 让你能够定义搜索超级链接的区域, 如果对象是图像 <code>name="nameurl"</code> : 作为表单的一部分提交的 URL
是否可为空	否

**Did you know?**

**提示:** `hspace` 和 `vspace` 属性在 HTML5 中已被删除, 因为它们纯粹用于表现, 因而用 CSS 处理更好。

**<param />**

用法	初始化对象
开始/结束标签	必需/非法
属性	<code>neme="name"</code> : 定义参数名 <code>value="value"</code> : 对象参数的值 <code>valuetype="valuetype"</code> : 定义值类型 ( <code>data</code> 、 <code>ref</code> 、 <code>object</code> ) <code>type="contenttype"</code> : Internet 介质类型
是否可为空	是

**A.8 XHTML 样式**

通过使用 `<style>` 元素, 可以将样式单 (内联和外部的) 加入到 HTML 文档中。

**<style>...</style>**

用法	创建内部样式单
开始/结束标签	必需/必需
属性	<code>il8n</code> <code>type="contenttype"</code> : Internet 内容类型 <code>media="media"</code> : 定义目标介质 ( <code>screen</code> 、 <code>print</code> 、 <code>projection</code> 、 <code>braille</code> 、 <code>speech</code> 、 <code>all</code> ) <code>title="title"</code> : 样式的名称
是否可为空	否
备注	位于 <code>&lt;head&gt;</code> 元素中

**A.9 XHTML 表单**

表单创建让用户能够选择选项、输入信息并将数据发送给 Web 服务器进行处理的界面。

**<button>...</button>**

用法	创建按钮
开始/结束标签	必需/必需
属性	<code>core</code> 、 <code>il8n</code> 和 <code>events</code> <code>name="name"</code> : 按钮名 <code>value="value"</code> : 按钮的值

是否可为空	<p><code>type="type"</code>: 按钮类型 (<code>button</code>、<code>submit</code>、<code>reset</code>)</p> <p><code>disabled="disabled"</code>: 将按钮状态设置为禁用</p> <p>否</p>
<b>&lt;fieldset&gt;...&lt;/fieldset&gt;</b>	
用法	将相关控件编组
开始/结束标签	必需/必需
属性	<code>core</code> 、 <code>idl8n</code> 和 <code>events</code>
是否可为空	否
<b>&lt;form&gt;...&lt;/form&gt;</b>	
用法	创建一个表单用于放置用户输入控件
开始/结束标签	必需/必需
属性	<p><code>core</code>、<code>idl8n</code> 和 <code>events</code></p> <p><code>action="actionurl"</code>: 服务器动作的 URL</p> <p><code>method="post/get"</code>: HTML 方法 (<code>get</code>、<code>post</code>), 不提倡使用 <code>get</code></p> <p><code>enctype="mediatype"</code>: 指定 MIME (Internet 媒体类型)</p> <p><code>accept="contentype"</code>: 服务器能够接受的内容类型列表</p> <p><code>accept-charset="encodings"</code>: 字符编码列表</p>
是否可为空	否
<b>&lt;input/&gt;</b>	
用法	定义在表单中使用的控件
开始/结束标签	必需/非法
属性	<p><code>core</code>、<code>idl8n</code> 和 <code>events</code></p> <p><code>type="controltype"</code>: 输入控件的类型 (<code>text</code>、<code>password</code>、<code>checkbox</code>、<code>radio</code>、<code>submit</code>、<code>reset</code>、<code>file</code>、<code>hidden</code>、<code>image</code>、<code>button</code>)</p> <p><code>name="name"</code>: 控件的名称 (除 <code>submit</code> 和 <code>reset</code> 外都需要)</p> <p><code>value="value"</code>: 控件的初始值 (单选按钮和复选框都需要)</p> <p><code>checked="checked"</code>: 将单选按钮的状态设置为被选中</p> <p><code>disabled="disabled"</code>: 禁用控件</p> <p><code>readonly="readonly"</code>: 适用于文本密码类型</p> <p><code>size="size"</code>: 除文本和密码控件外其他控件的宽度 (像素数), 文本和密码控件的宽度用字符数指定</p> <p><code>maxlength="maxlength"</code>: 可输入的最大字符数</p> <p><code>src="imageurl"</code>: 图像控件类型的 URL</p>

	<p><code>alt="alttext"</code>: 替代文本描述</p> <p><code>usemap="mapurl"</code>: 到客户端图像映射的 URL</p> <p><code>accept="filetypes"</code>: 允许上传的文件类型</p>
是否可为空	是
<b>&lt;label&gt;...&lt;/label&gt;</b>	
用法	给控件加上标签
开始/结束标签	必需/必需
属性	<p>core、i18n 和 events</p> <p><code>for="control"</code>: 将控件与标签关联起来</p> <p><code>onfocus="eventcode"</code>: 元素获得焦点时发生的事件</p> <p><code>onblur="eventcode"</code>: 元素失去焦点时发生的事件</p>
是否可为空	否
<b>&lt;label&gt;...&lt;/label&gt;</b>	
用法	给控件加上标签
开始/结束标签	必需/必需
属性	<p>core、i18n 和 events</p> <p><code>for="control"</code>: 将控件与标签关联起来</p>
是否可为空	否
<b>&lt;option&gt;...&lt;/option&gt;</b>	
用法	指定<select>元素中的选项
开始/结束标签	必需/可选
属性	<p>core、i18n 和 events</p> <p><code>selected="selected"</code>: 指定选项是否被选中</p> <p><code>disabled="disabled"</code>: 禁用控件</p> <p><code>label="label"</code>: 为选项组定义标签</p> <p><code>value="value"</code>: 控件提交时的值</p>
是否可为空	否
<b>&lt;select&gt;...&lt;/select&gt;</b>	
用法	创建一些选项供用户选择
开始/结束标签	必需/必需
属性	<p>core、i18n 和 events</p> <p><code>name="name"</code>: 元素的名称</p>

	size="size": 以行为单位的宽度
	multiple="multiple": 允许多项选择
	disabled="disabled": 禁用控件
是否可为空	否
<b>&lt;textarea&gt;...&lt;/textarea&gt;</b>	
用法	创建一个区域供用户输入多行文本
开始/结束标签	必需/必需
属性	core、i18n 和 events
	name="name": 元素的名称
	rows="numrows": 以行为单位的宽度
	cols="numcols": 以列为单位的高度
	disabled="disabled": 禁用控件
	readonly="readonly": 设置显示的文本为只读状态
是否可为空	否
备注	将要显示的文本放在开始和结束标签之间

## A.10 XHTML 脚本

使用脚本可以处理数据和执行其他动态事件。使用<script>元素将脚本包含的网页中，该元素还指出了使用的脚本语言（JavaScript、VBScript 等）。

### <noscript>...</noscript>

用法	提供浏览器不能执行脚本时使用的替代内容
开始/结束标签	必需/必需
属性	core、i18n 和 events
是否可为空	否

### <script>...</script>

用法	<script>元素包含由浏览器执行的客户端脚本
开始/结束标签	必需/必需
属性	type="scripttype": 脚本语言的 Internet 内容类型
	src="scripturl": 外部脚本的 URL
	defer="defer": 指出脚本不修改文档内容
是否可为空	否
备注	可在<meta/>元素中设置默认的脚本语言

## A.11 XHTML 常用属性

在前面几节中，下面 6 个属性被统称为 core。

- `id="id"`: 全局标识符。
- `class="styleclasses"`: 由空格分隔的类列表。
- `style="styles"`: 样式信息。
- `title="title"`: 为特定元素提供其他信息；而<title>元素指定整个网页的标题。
- `accesskey="shortcut"`: 设置用于访问元素的快捷键。
- `tabindex="taborder"`: 设置元素的 Tab 顺序。

在前几节，下面两个用于国际化 (i18n) 的属性统称为 i18n。

- `lang="lang"`: 语言标识符
- `dir="textdir"`: 文本方向 (ltr, rtl)

下列内部事件统称为 events。关于这些事件在具体元素中的应用的更多信息，参见 W3C 规范。

- `OnClick="eventcode"`: 指针设备（如鼠标）被单击。
- `OnDbClick=" eventcode"`: 指针设备（如鼠标）被双击。
- `OnMouseDown=" eventcode"`: 按住鼠标按钮。
- `OnMouseUp=" eventcode"`: 松开被按住的鼠标按钮。
- `OnMouseOver=" eventcode"`: 鼠标移到对象上。
- `OnMouseMove=" eventcode"`: 鼠标移动。
- `OnMouseOut=" eventcode"`: 鼠标离开对象。
- `OnKeyPress=" eventcode"`: 按下键然后松开。
- `OnKeyDown=" eventcode"`: 键被按住。
- `OnKeyUp=" eventcode"`: 按住的键被松开。

## A.12 CSS 尺寸样式属性

很多 CSS 样式规则依赖于一个或另一个表单的尺寸属性。如果不使用这些尺寸属性，将难以指定表单中元素的尺寸。

<b>height</b>	
用法	设置元素的高度
值	auto、length、%
<b>line-height</b>	
用法	设置元素行之间的距离
值	normal、length、%



<b>max-height</b>	
用法	设置元素的最大高度
值	none、length、%
<b>max-width</b>	
用法	设置元素的最大宽度
值	none、length、%
<b>min-height</b>	
用法	设置元素的最小高度
值	length、%
<b>min-width</b>	
用法	设置元素的最小宽度
值	length、%
<b>width</b>	
用法	设置元素的宽度
值	auto、length、%

## A.13 CSS 文本和字体样式属性

CCS 的核心是文本和字体样式属性，它们让你能够以难以置信的方式控制网页文本的外观。

<b>color</b>	
用法	设置文本的颜色
值	color
<b>direction</b>	
用法	设置文本的方向，从左到右还是从右到左
值	ltr、rtl
<b>font</b>	
用法	让你能够在—个地方设置所有字体属性
值	font-style、font-variant、font-weight、font-size/line-height、font-family
<b>font-family</b>	
用法	用于元素的顺序字体名和/或通用字体名列表
值	family-name、generic-family
<b>font-size</b>	
用法	设置字体大小

值	xx-small、x-small、small、medium、large、x-large、xx-large、smaller、larger、 <i>length</i> 、%
<b>font-style</b>	
用法	设置字体样式
值	family-name、generic-family
<b>font-variant</b>	
用法	以小型大写字体或常规字体显示文本
值	normal、small-caps
<b>font-weight</b>	
用法	设置字体的粗细
值	normal、bold、bolder、lighter、100、200、300、400、500、600、700、800、900
<b>letter-spacing</b>	
用法	增大或缩小文本的字符间距
值	normal、 <i>length</i>
<b>text-align</b>	
用法	对齐元素中的文本
值	left、right、center、justify
<b>text-decoration</b>	
用法	将修饰应用于文本
值	none、underline、overline、line-through、blink
<b>text-indent</b>	
用法	缩进元素中文本的首行
值	<i>length</i> 、%
<b>text-transform</b>	
用法	控制文本的大小写
值	none、capitalize、uppercase、lowercase
<b>white-space</b>	
用法	指定如何处理元素中的空格
值	normal、pre、nowrap
<b>word-spacing</b>	
用法	增大或缩小字间距
值	normal、 <i>length</i>

## A.14 CSS 背景样式属性

有几个 CSS 样式属性可用于修改页面的背景以及页面上各个元素的背景。

### background

用法	让你能够在一个地方设置所有的背景属性
值	background-color、background-image、background-repeat、background-attachment、background-position

### background-attachment

用法	指定背景图像是固定的还是随其他页面内容滚动
值	scroll、fixed

### background-color

用法	设置元素的背景色
值	color-rgb、color-hex、color-name、transparent

### background-image

用法	将图像用作背景
值	url、none

### background-position

用法	设置背景图像的起始位置
值	top left、top center、top right、center left、center center、center right、bottom left、bottom center、bottom right、x-% y-%、x-pos y-pos

### background-repeat

用法	设置是否以及如何重复背景图像
值	repeat、repeat-x、repeat-y、no-repeat

## A.15 CSS 边框样式属性

每个块元素都有样式可设置的边框。虽然可以让边框不可见，但有多种样式可应用于元素的边框。

### border

用法	让你能够在一个地方设置所有 4 条边框的所有属性
值	border-width、border-style、border-color

### border-bottom

用法	让你能够在一个地方设置下边框的所有属性
值	border-bottom-width、border-style、border-color

**border-bottom-color**

用法 设置下边框的颜色

值 *border-color*

**border-bottom-style**

用法 设置下边框的样式

值 *border-style*

**border-bottom-width**

用法 设置下边框的宽度

值 *thin*、*medium*、*thick*、*length*

**border-color**

用法 设置四条边框的颜色

值 *color*

备注 可以使用 1~4 种颜色指定

**border-left**

用法 让你能够在—个地方设置左边框的所有属性

值 *border-left-width*、*border-style*、*border-color*

**border-left-color**

用法 设置左边框的颜色

值 *border-color*

**border-left-style**

用法 设置左边框的样式

值 *border-style*

**border-left-width**

用法 设置左边框的宽度

值 *thin*、*medium*、*thick*、*length*

**border-right**

用法 让你能够在—个地方设置右边框的所有属性

值 *border-right-width*、*border-style*、*border-color*

**border-right-color**

用法 设置右边框的颜色

值 *border-color*

**border-right-style**

用法 设置右边框的样式

值	<i>border-style</i>
<b>border-right-width</b>	
用法	设置右边框的宽度
值	<i>thin</i> 、 <i>medium</i> 、 <i>thick</i> 、 <i>length</i>
<b>border-style</b>	
用法	设置 4 个边框的样式
值	<i>none</i> 、 <i>hidden</i> 、 <i>dotted</i> 、 <i>dashed</i> 、 <i>solid</i> 、 <i>double</i> 、 <i>groove</i> 、 <i>ridge</i> 、 <i>inset</i> 、 <i>outset</i>
备注	可以使用 1~4 种样式来指定
<b>border-top</b>	
用法	让你能够在 一个地方设置上边框的所有属性
值	<i>border-top-width</i> 、 <i>border-style</i> 、 <i>border-color</i>
<b>border-top-color</b>	
用法	设置左边框的颜色
值	<i>border-color</i>
<b>border-top-style</b>	
用法	设置左边框的样式
值	<i>border-style</i>
<b>border-top-width</b>	
用法	设置左边框的宽度
值	<i>thin</i> 、 <i>medium</i> 、 <i>thick</i> 、 <i>length</i>
<b>border-width</b>	
用法	让你能够在 一个地方设置 4 条边框的宽度
值	<i>thin</i> 、 <i>medium</i> 、 <i>thick</i> 、 <i>length</i>
备注	可以使用 1~4 个宽度值来指定

## A.16 CSS 边距样式属性

边距让你能够在元素的外边缘周围（元素边框的外面）增加空间。

### margin

用法 让你能够在 一个地方设置所有边距属性

值 *margin-top*、*margin-right*、*margin-bottom*、*margin-left*

### margin-bottom

用法 设置元素的下边距

值 *auto*、*length*、*%*

**margin-left**

用法	设置元素的左边距
值	auto、 <i>length</i> 、%

**margin-right**

用法	设置元素的右边距
值	auto、 <i>length</i> 、%

**margin-top**

用法	设置元素的上边距
值	auto、 <i>length</i> 、%

## A.17 CSS 填充样式属性

填充让你能够在元素周围（边框内部）增加空间。

**padding**

用法	让你能够在—个地方设置所有填充属性
值	<i>padding-top</i> 、 <i>padding-right</i> 、 <i>padding-bottom</i> 、 <i>padding-left</i>

**padding-bottom**

用法	设置元素的下填充
值	<i>length</i> 、%

**padding-left**

用法	设置元素的左填充
值	<i>length</i> 、%

**padding-right**

用法	设置元素的右填充
值	<i>length</i> 、%

**padding-top**

用法	设置元素的上填充
值	<i>length</i> 、%

## A.18 CSS 布局和显示样式属性

在指定如何在页面上布局和排列元素方面，CSS 布局和显示样式属性扮演了至关重要的角色。

**bottom**

用法	设置元素下边缘同其父元素下边缘之间的偏移量
值	auto、 <i>length</i> 、%



<b>clear</b>	
用法	指定元素周围的哪些地方不能有其他元素
值	left、right、both、none
<b>clip</b>	
用法	设置元素的形状
值	auto、 <i>shape</i>
备注	显示时元素被裁剪为指定的形状
<b>cursor</b>	
用法	指定要显示的鼠标形状
值	<i>url</i> 、auto、crosshair、default、pointer、move、e-resize、ne-resize、nw-resize、n-resize、se-resize、sw-resize、s-resize、w-resize、text、wait、help
<b>display</b>	
用法	设置是否以及如何显示元素
值	none、inline、block、list-item、run-in、compact、marker、table、inline-table、table-row-group、table-header-group、table-footer-group、table-row、table-column-group、table-column、table-cell、table-caption
<b>float</b>	
用法	设置图像或文本相对于另一个元素的位置
值	left、right、none
<b>left</b>	
用法	设置元素左边缘同其父元素左边缘之间的偏移量
值	auto、 <i>length</i> 、%
<b>overflow</b>	
用法	指定如果元素的内容容纳不下如何处理
值	auto、visible、hidden、scroll
<b>position</b>	
用法	指定元素的布局为静态的、相对的、绝对的还是固定的
值	static、relative、absolute、fixed
<b>right</b>	
用法	设置元素右边缘同其父元素右边缘之间的偏移量
值	auto、 <i>length</i> 、%

<b>top</b>	
用法	设置元素上边缘同其父元素上边缘之间的偏移量
值	auto、 <i>length</i> 、%
<b>vertical-align</b>	
用法	设置元素的垂直对齐方式
值	baseline、sub、super、top、text-top、middle、bottom、text-bottom、 <i>length</i> 、%
<b>visibility</b>	
用法	指定应显示（可见）还是隐藏（不可见）元素
值	visible、hidden、collapse
<b>z-index</b>	
用法	设置元素的 z 顺序（堆叠顺序）
值	auto、 <i>number</i>

## A.19 CSS 列表和项目符号样式属性

你可能还没有意识到，通过 CSS 指定列表样式有多灵活。有几个 CSS 样式属性可用于列表和列表中的列表项目符号。

<b>list-style</b>	
用法	让你能够在一个地方设置所有的列表属性
值	<i>list-style-type</i> 、 <i>list-style-position</i> 、 <i>list-style-image</i>
<b>list-style-image</b>	
用法	设置用作列表项目符号的图像
值	<i>none</i> 、 <i>url</i>
<b>list-style-position</b>	
用法	指定项目符号在列表中的位置
值	inside、outside
<b>list-style-type</b>	
用法	设置列表项目符号的类型
值	none、disc、circle、square、decimal、decimal-leading-zero、lower-roman、upper-roman、lower-alpha、upper-alpha、lower-greek、lower-latin、upper-latin、hebrew、armenian、georgian、cjk-ideographic、hira-gana、katakana、hiragana-iroha、katakana-iroha

## A.20 CSS 表格样式属性

有几个高级表格属性让你能够对表格的显示方式进行微调。

### **border-collapse**

用法	设置表格边框的模式
值	collapse、separate

### **border-spacing**

用法	设置相邻单元格边框之间的间距
值	<i>length length</i>

### **caption-side**

用法	设置表题相对于表格的位置
值	top、bottom、left、right

### **empty-cells**

用法	指定没有可见内容的单元格是否有边框
值	show、hide

### **table-layout**

用法	指定表格布局
值	auto、fixed
备注	如果将其设置为 fixed，表格大小将是固定的，这样可以提高浏览器显示表格的速度



# HTML与CSS

SAMS

入门经典 (第8版)

Teach Yourself

## 学习如何：

- 创建网页并将其放到网上；
- 格式化文本，使其得到最大限度的清晰度和可读性；
- 创建到其他网页和网站的链接；
- 在网页中添加图像、颜色和视觉效果；
- 使用透明图像和背景图像；
- 使用CSS设计站点布局和排版；
- 使用Web表单得到用户的输入；
- 宣传你的网站并且使其对搜索引擎友好；
- 用不同的浏览器测试网站的兼容性；
- 使网站易于维护和更新。

## 24章阶梯教学

通过阅读本书，读者将学会如何使用HTML和CSS来设计、创建和维护世界级的网站。本书采用直观、循序渐进的方法，引导读者掌握从基本知识到最新功能的所有内容。本书每章内容都建立在已学的知识之上，即使读者没有任何HTML与CSS知识，也可以通过本书，走向成功的坚实之路。

循序渐进的示例引导读者完成最常见的任务。

问与答、测验和练习帮助读者检验知识的掌握情况。

“注意”、“提示”和“警告”指出捷径和解决方案。

美术编辑 王建国

分类建议：计算机/网页制作/HTML  
人民邮电出版社网址：[www.ptpress.com.cn](http://www.ptpress.com.cn)

ISBN 978-7-115-25660-7



9 787115 256607 >

ISBN 978-7-115-25660-7

定价：45.00 元