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
- “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




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9版）

[美] Julie Meloni 著
陈宗斌 译

人民邮电出版社
POSTS & TELECOM PRESS

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# HTML与CSS入门经典（第9版）

[美] Julie Meloni 著
陈宗斌 译



人 民 邮 电 出 版 社
北 京

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**图书在版编目（CIP）数据**

HTML与CSS入门经典 : 第9版 / (美) 梅洛尼(Meloni,J.) 著 ; 陈宗斌译. -- 北京 : 人民邮电出版社, 2014.9
ISBN 978-7-115-36529-3

Ⅰ. ①H… Ⅱ. ①梅… ②陈… Ⅲ. ①超文本标记语言－程序设计－教材②网页制作工具－教材 Ⅳ. ①TP312②TP393.092

中国版本图书馆CIP数据核字(2014)第171132号

## 版 权 声 明

Julie Meloni: Sams Teach Yourself HTML and CSS in 24 Hours

ISBN: 0672336146

Copyright © 2014 by Pearson Education, Inc.

Authorized translation from the English languages edition published by Pearson Education, Inc.

All rights reserved.

本书中文简体字版由美国 **Pearson** 公司授权人民邮电出版社出版。未经出版者书面许可，对本书任何部分不得以任何方式复制或抄袭。

版权所有，侵权必究。

◆ 著　　[美] Julie Meloni
译　　陈宗斌
责任编辑　傅道坤
责任印制　彭志环　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路 11 号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
大厂聚鑫印刷有限责任公司印刷

◆ 开本：787×1092　1/16
印张：21.25
字数：528 千字　　2014 年 9 月第 1 版
印数：1 – 3 500 册　　2014 年 9 月河北第 1 次印刷
著作权合同登记号　图字：01-2012-7072 号

定价：45.00 元

**读者服务热线：(010)81055410　印装质量热线：(010)81055316**
**反盗版热线：(010)81055315**

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 内容提要

本书是经典畅销图书《HTML 与 CSS 入门经典》的最新版本，与过去的版本相同，本书采用直观、循序渐进的方法，为读者讲解使用 HTML5 与 CSS3 设计、创建并维护世界级 Web 站点的过程，以方便读者掌握。

本书分为 24 章，其涵盖的内容有理解 Web 的工作方式，构造 HTML 文档，理解层叠样式表、理解 HTML5 页面架构，处理文本块和列表，处理字体，处理颜色和边框，使用外部链接和内部链接，使用表格和栏，创建用于 Web 的图像，在 Web 站点中使用图像和多媒体，处理边距、填充、对齐和浮动，理解 CSS 方框模型和定位，创建固定或流动布局，使用 CSS 进一步处理列表并设计导航，使用鼠标操作修改文本显示，实现 CSS3 变形、渐变和动画，创建打印友好的 Web 界面，理解动态 Web 站点和 HTML5 应用，处理基于 Web 的表单，组织和管理 Web 站点，以及帮助人们找到你的 Web 页面等内容。

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。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 关于作者

**Julie Meloni** 是一位居住在华盛顿特区的软件开发经理和技术顾问。她编写过多本与 Web 编程语言和数据库相关的图书，并发表了若干篇文章，其中包括畅销书 *Sams Teach Yourself PHP, MySQL and Apache All in One*。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 前　言

2012 年，有超过 28 亿人访问过 Internet，仅仅在美国就有 2.55 亿人，另外还有 5.67 亿中国用户、6900 万德国用户、5400 万英国用户、7600 万俄罗斯用户和 9900 万巴西用户，由此可以看出“世界”（World）这个词语在术语“万维网”（World Wide Web，WWW）中的意义。许多 Internet 用户也在为 Web 创建内容，你可能就是其中一员。尽管难以得到 Web 页面总数的准确统计，可我们还是知道 Google 在 2008 年中期就突破了 1 万亿个索引页面标记。

在接下来的 24 小时，将有数亿个新页面出现在 Internet 的可访问区域中。其中有许多页面将放置在私有内联网上，它们在那里将被通过局域网连接的商务人士访问。其中每个页面——像已经在线的数十亿个页面一样——都将使用超文本标记语言（Hypertext Markup Language，HTML）编写。

当学完本书后，你的 Web 页面也会出现在 Internet 上。本书还可以给你提供一个基础，用于开发当今世界中最有价值的技能之一：精通 HTML。

你真的不用借助任何专业软件，而且无需与费用高昂的设计顾问进行交互，就能花费比预期更少的时间，自行学会创建最高质量的 Web 页面吗？这本篇幅短小、容易阅读的图书真的能让你自学最新的技术，用于开发和发布 Web 内容吗？

是的。事实上，通过本书前两章的学习，即使是没有任何 HTML 经验的读者，也可以编写出能够上线运行的 Web 页面。

怎样才能如此快地学习 Web 的语言呢？本书基于 HTML 和 CSS，把课程内容组织到简单的步骤中，然后准确显示了如何处理每个步骤。其中许多 HTML 代码示例都配有由代码运行后生成的图片。你将看到它是如何做到的，它们的工作机制是什么，然后你就可以立即对自己的页面做相同的事情。几分钟后，可以继续执行下一个步骤。

不久以后，你会对自己发布在 Internet 上的页面感到惊叹。

## 超越 HTML

本书介绍的内容并不仅限于 HTML，因为 HTML 不是今天创建 Web 内容所需知道的唯

异步社区会员 Kensuke(18319066421) 专享 尊重版权

一知识。本书的目标是给你提供一个坚实的基础，使你仅通过 24 章内容的学习，就能掌握创建符合标准的现代 Web 站点所需的技能。本书将介绍以下关键的技能和技术。

- HTML5 是用于创建 Web 页面的最新建议，本书中的每个示例都使用 HTML5 验证过。
- 经测试，本书中的所有示例都与每种主流的 Web 浏览器兼容，包括 Apple Safari、Google Chrome、Microsoft Internet Explorer、Mozilla Firefox 和 Opera。你将从头开始学习与过去的标准保持兼容，同时为将来也做好准备。
- 本书广泛介绍了层叠样式表（Cascading Style Sheet，CSS）的最新建议，它允许仔细控制布局、字体、颜色以及 Web 页面每个方面的格式化效果，包括文本和图像。在创建引人注目的 Web 页面时，CSS 远远超过了仅凭传统的 HTML 本身可以做的事情。
- 上述的技术知识不足以解决所有问题，因此本书还包括了在建立 Web 站点来实现目标时所需了解的建议。本书足够深入地介绍了所有的关键细节，包括设计有效的页面布局、利用 FTP 软件把页面发布到 Internet 上、组织和管理多个页面，以及使页面在所有主流的 Internet 搜索站点中出现在查询列表中的较高位置，它们使你能够轻松跨越通常会使初学者感到挫败的障碍。

本书之前的 8 个版本重点讲解了许多基本知识，这个更新版本也不例外。本书中的所有示例都进行了更新，并且对内容中的重要部分进行了修订，以匹配新的示例和新的技术，完全兼容 HTML5 和 CSS3。

## 可视化示例

本书中的每个示例都分为两个部分进行演示。

- 首先显示创建 HTML 页面的文本，并且会着重显示 HTML 和 CSS 代码。
- 然后给出用户利用世界上最流行的 Web 浏览器查看时所显示的结果 Web 页面。

你通常根本不必阅读任何相关的文本（尽管强烈建议你阅读其余的文本），就能够修改示例，使之适合于你自己的页面。

本书中的所有示例都符合标准，并且可以与 Apple Safari、Google Chrome、Microsoft Internet Explorer、Mozilla Firefox 和 Opera 协同工作。尽管所有的截图都取自 Chrome，仍然可以保证所有的代码在所有其他的浏览器中都可以良好运行。

## 问题、测验和练习

每章末尾都带有与该章内容相关的简短问题以及相应的答案，用于解决读者可能会问到的一些问题。一份简明而完整的测验可以让你测试自己的水平，确保你理解了该章中介绍的所有内容。最后，还提供了一两个可选的练习，使你能够在继续学习后面的内容之前有机会实践新的技能。



异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 目　录


第 1 章　理解 Web 的工作方式 ……1

1.1　HTML 和 WWW 简史 ……2
1.2　创建 Web 内容 ……2
1.3　理解 Web 内容递送 ……3
1.4　选择 Web 托管提供商 ……4
1.5　利用多种 Web 浏览器执行测试……6
1.6　创建一个示例文件 ……7
1.7　使用 FTP 传输文件……7
1.7.1　选择 FTP 客户端……8
1.7.2　使用 FTP 客户端……9
1.8　了解在 Web 服务器上放置文件的位置……11
1.8.1　基本的文件管理……11
1.8.2　使用索引页面 ……12
1.9　在不涉及 Web 服务器的情况下分发内容 ……13
1.9.1　在本地发布内容……13
1.9.2　在博客上发布内容……14
1.10　关于测试 Web 内容的提示……14
1.11　小结 ……15
1.12　问与答……15
1.13　测验……16
1.13.1　问题 ……16
1.13.2　答案 ……17
1.13.3　练习 ……17

第 2 章　构造 HTML 文档 ……18

2.1　准备工作……18
2.2　从一个简单的 Web 页面开始 ……19
2.3　每个 Web 页面都必须具有的 HTML 标签……22
2.4　利用段落和换行符组织页面 ……23
2.5　利用标题组织内容 ……25
2.6　理解语义元素 ……27
2.7　验证 Web 内容……28
2.8　小结 ……30
2.9　问与答……31
2.10　测验……31
2.10.1　问题……31
2.10.2　答案……31
2.10.3　练习 ……32

第 3 章　理解层叠样式表……33

3.1　CSS 的工作方式 ……34
3.2　基本的样式表 ……35
3.3　CSS 样式的基础知识 ……39
3.3.1　基本的布局属性 ……39
3.3.2　基本的格式化属性 ……40
3.4　使用样式类……42
3.5　使用样式 ID……44


异步社区会员 Kensuke(18319066421) 专享 尊重版权

3.6 内部样式表和内联样式……45
3.7 小结……47
3.8 问与答……47
3.9 测验……48
3.9.1 问题……48
3.9.2 答案……48
3.9.3 练习……48

第 4 章 密切观察 HTML5 页面结构……50

4.1 概念化页面……50
4.2 以多种方式使用<header>元素……54
4.3 理解<section>元素……55
4.4 使用<article>元素……56
4.5 实现<nav>元素……56
4.6 何时使用<aside>元素……57
4.7 有效地使用<footer>元素……58
4.8 小结……58
4.9 问与答……59
4.10 测验……59
4.10.1 问题……59
4.10.2 答案……59
4.10.3 练习……60

第 5 章 处理文本块和列表……61

5.1 在页面上对齐文本……62
5.1.1 使用属性……62
5.1.2 对齐块级元素中的文本……62
5.2 3 类 HTML 列表……64
5.3 在列表内放置列表……66
5.4 小结……70
5.5 问与答……71
5.6 测验……71
5.6.1 问题……71
5.6.2 答案……71
5.6.3 练习……72

第 6 章 处理字体……73

6.1 处理特殊字符……74
6.2 粗体、斜体和特殊的文本格式化……76
6.3 调整字体……79
6.4 使用 Web 字体……82
6.5 小结……84
6.6 问与答……85
6.7 测验……85
6.7.1 问题……85
6.7.2 答案……86
6.7.3 练习……86

第 7 章 处理颜色和边框……87

7.1 选择颜色的最佳实践……87
7.2 理解 Web 颜色……88
7.3 使用十六进制的颜色值……90
7.4 使用 CSS 设置背景、文本和边框颜色……91
7.5 创建圆角……93
7.6 小结……95
7.7 问与答……95
7.8 测验……96
7.8.1 问题……96
7.8.2 答案……96
7.8.3 练习……96

第 8 章 使用外部链接和内部链接……97

8.1 使用 Web 地址……97
8.2 使用锚点在页面内链接……99
8.2.1 利用锚点标识页面中的位置……100
8.2.2 链接到锚点位置……100
8.3 在自己的 Web 内容之间创建链接……102
8.4 链接到外部 Web 内容……105
8.5 链接到电子邮件地址……106
8.6 在新的浏览器窗口中打开链接……107
8.7 使用 CSS 编排超链接的样式……107
8.8 小结……110
8.9 问与答……111
8.10 测验……112
8.10.1 问题……112



异步社区会员 Kensuke(18319066421) 专享 尊重版权

8.10.2 答案 ……112
8.10.3 练习 ……112

第 9 章 使用表格和栏 ……114

9.1 创建一个简单的表格……114
9.2 控制表格大小 ……118
9.3 表格内的对齐和跨越……120
9.4 利用表格进行页面布局……123
9.5 使用 CSS 栏 ……123
9.6 小结……127
9.7 问与答……127
9.8 测验……127
9.8.1 问题 ……128
9.8.2 答案 ……128
9.8.3 练习 ……128

第 10 章 创建用于 Web 的图像 ……129

10.1 选择图形软件 ……129
10.2 关于图形需要知道的最少知识 ……130
10.3 准备照片图像 ……131
10.3.1 裁剪图像 ……132
10.3.2 调整图像大小……133
10.3.3 调整图像颜色……134
10.3.4 控制 JPEG 压缩……135
10.4 创建横幅和按钮 ……135
10.5 减少或删除图像中的颜色……137
10.6 创建拼贴式背景图像……137
10.7 创建动画式 Web 图形 ……139
10.8 小结……139
10.9 问与答……140
10.10 测验……140
10.10.1 问题 ……140
10.10.2 答案 ……141
10.10.3 练习 ……141

第 11 章 在 Web 站点中使用图像 ……142

11.1 在 Web 页面上放置图像……142
11.2 利用文本描述图像……145
11.3 指定图像的高度和宽度……145
11.4 对齐图像 ……146
11.4.1 水平图像对齐……146
11.4.2 垂直图像对齐 ……148
11.5 把图像转变成链接 ……149
11.6 使用背景图像 ……152
11.7 使用图像映射 ……153
11.7.1 为什么图像映射并非总是必要的 ……153
11.7.2 映射图像内的区域……154
11.7.3 为图像映射创建 HTML 代码……155
11.8 小结……157
11.9 问与答……158
11.10 测验……158
11.10.1 问题 ……159
11.10.2 答案 ……159
11.10.3 练习 ……159

第 12 章 在 Web 站点中使用多媒体 …160

12.1 链接到多媒体文件 ……161
12.2 嵌入多媒体文件 ……163
12.3 为音频和视频播放使用纯 HTML5 代码……165
12.4 关于使用多媒体的额外提示 ……167
12.5 小结……168
12.6 问与答……169
12.7 测验……170
12.7.1 问题 ……170
12.7.2 答案 ……170
12.7.3 练习 ……170

第 13 章 处理边距、填充、对齐和浮动……171

13.1 使用边距……171
13.2 填充元素……177
13.3 使所有的一切保持对齐 ……180
13.4 理解 float 属性……181
13.5 小结……183
13.6 问与答……183
13.7 测验……184
13.7.1 问题 ……184
13.7.2 答案 ……184
13.7.3 练习 ……184



异步社区会员 Kensuke(18319066421) 专享 尊重版权

第 14 章　理解 CSS 方框模型和定位 …185

14.1　CSS 方框模型 ……185
14.2　关于定位的独家新闻……188
14.3　控制元素的堆叠方式……191
14.4　管理文本流 ……193
14.5　小结……194
14.6　问与答……194
14.7　测验……194
14.7.1　问题 ……194
14.7.2　答案 ……195
14.7.3　练习 ……195

第 15 章　创建固定布局或流动布局 ……196

15.1　理解固定布局 ……197
15.2　理解流动布局 ……198
15.3　创建固定/流动混合布局……200
15.3.1　从基本的布局结构开始 ……200
15.3.2　在固定/流动混合布局中定义两栏 ……201
15.3.3　设置布局的最小宽度 ……203
15.3.4　在固定/流动混合布局中处理栏高度 ……204
15.4　考虑响应性 Web 设计 ……208
15.5　小结……209
15.6　问与答……209
15.7　测验……209
15.7.1　问题 ……209
15.7.2　答案 ……210
15.7.3　练习 ……210

第 16 章　使用 CSS 进一步处理列表 …211

16.1　HTML 列表复习 ……211
16.2　CSS 方框模型如何影响列表……212
16.3　放置列表项指示符 ……215
16.4　利用列表项和 CSS 创建图像映射……216
16.5　小结……219
16.6　问与答……219
16.7　测验……219
16.7.1　问题……220
16.7.2　答案……220
16.7.3　练习 ……220

第 17 章　使用 CSS 设计导航系统……221

17.1　导航列表与常规的列表有何区域……221
17.2　利用 CSS 创建垂直导航系统 ……222
17.2.1　编排单级垂直导航系统的样式……224
17.2.2　编排多级垂直导航系统的样式……226
17.3　利用 CSS 创建水平导航系统 ……229
17.4　小结……232
17.5　问与答……232
17.6　测验……233
17.6.1　问题……233
17.6.2　答案……233
17.6.3　练习 ……233

第 18 章　使用鼠标动作修改文本显示……234

18.1　利用 CSS 创建工具提示 ……234
18.2　利用 CSS 显示额外的翻转文本……237
18.3　访问事件……238
18.4　使用 onclick 更改 的外观 ……239
18.5　小结……244
18.6　问与答……244
18.7　测验……245
18.7.1　问题……245
18.7.2　答案……245
18.7.3　练习 ……245

第 19 章　实现 CSS3 变形、渐变和动画 ……246

19.1　考虑浏览器的差别 ……246
19.2　使用 2D 变形 ……247



异步社区会员 Kensuke(18319066421) 专享 尊重版权

19.2.1 平移元素 ……247
19.2.2 旋转元素 ……249
19.2.3 缩放元素 ……250
19.2.4 扭曲元素 ……252
19.3 使用 3D 变形 ……253
19.3.1 应用透视图 ……253
19.3.2 在 3D 空间中移动和旋转元素 ……255
19.3.3 创建和操纵立方体 ……255
19.4 实现 CSS3 渐变 ……259
19.5 初识 CSS3 动画 ……260
19.6 小结 ……261
19.7 问与答 ……261
19.8 测验 ……262
19.8.1 问题 ……262
19.8.2 答案 ……262
19.8.3 练习 ……262

第 20 章 创建打印友好的 Web 页面 ……263

20.1 是什么使页面成为打印友好的 ……264
20.2 应用特定于媒体的样式表 ……266
20.3 设计用于打印页面的样式表 ……267
20.4 在 Print Preview（打印预览）中查看 Web 页面 ……270
20.5 小结 ……272
20.6 问与答 ……272
20.7 测验 ……272
20.7.1 问题 ……272
20.7.2 答案 ……273
20.7.3 练习 ……273

第 21 章 理解动态 Web 站点和 HTML5 应用程序 ……274

21.1 理解不同类型的脚本 ……274
21.2 在 HTML 中包括 JavaScript 代码 ……275
21.3 显示随机内容 ……277
21.4 理解文档对象模型 ……280
21.5 基于用户交互更改图像 ……281
21.6 提前考虑开发 HTML5 应用程序 ……282
21.7 小结 ……283
21.8 问与答 ……283
21.9 测验 ……284
21.9.1 问题 ……284
21.9.2 答案 ……284
21.9.3 练习 ……285

第 22 章 处理基于 Web 的表单 ……286

22.1 HTML 表单是如何工作的 ……286
22.2 创建表单 ……287
22.3 接受文本输入 ……291
22.4 命名每一份表单数据 ……292
22.4.1 标记每一份表单数据 ……292
22.4.2 组合表单元素 ……293
22.4.3 在表单中包括隐藏的数据 ……294
22.5 探讨表单输入控件 ……294
22.5.1 复选框 ……294
22.5.2 单选按钮 ……296
22.5.3 选择列表 ……296
22.5.4 文本框、文本区和其他输入类型 ……297
22.6 使用 HTML5 表单验证 ……298
22.7 提交表单数据 ……300
22.8 小结 ……301
22.9 问与答 ……302
22.10 测验 ……302
22.10.1 问题 ……302
22.10.2 答案 ……303
22.10.3 练习 ……303

第 23 章 组织和管理 Web 站点 ……304

23.1 当一个页面就足够时 ……305
23.2 组织简单的站点 ……306
23.3 组织更大的站点 ……308
23.4 编写易于维护的代码 ……311
23.4.1 利用注释记录代码 ……311
23.4.2 缩进代码以使之更清晰 ……312



异步社区会员 Kensuke(18319066421) 专享 尊重版权

23.5 考虑版本控制 ……313
23.6 使用 HTML 和 CSS 框架 ……314
23.7 小结……315
23.8 问与答……315
23.9 测验……316
23.9.1 问题 ……316
23.9.2 答案 ……316
23.9.3 练习 ……316

**第 24 章 帮助人们找到你的 Web 页面** ……318

24.1 宣传你的 Web 站点 ……318
24.2 利用主要的搜索站点列出你的页面 ……320
24.3 为搜索引擎提供线索 ……321
24.4 关于搜索引擎优化的额外提示 ……326
24.5 小结……327
24.6 问与答……327
24.7 测验……328
24.7.1 问题……328
24.7.2 答案……328
24.7.3 练习……328



异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 1 章

# 理解 Web 的工作方式

在本章中你将学到：

- 万维网（World Wide Web，WWW）简史；
- 术语 Web 页面（web page）的含义是什么，为什么这个术语并不总会反映所涉及的全部内容；
- 内容怎样从你的个人计算机到达别人的 Web 浏览器上；
- 怎样选择 Web 托管提供商；
- 不同的 Web 浏览器和设备类型可能怎样影响内容；
- 怎样使用 FTP 把文件传输到 Web 服务器上；
- 应该把文件存放在 Web 服务器上的什么位置；
- 在不涉及 Web 服务器的情况下怎样分发 Web 内容；
- 怎样使用其他发布方法，比如博客；
- 关于测试 Web 内容的外观和功能的提示。

在学习超文本标记语言（Hypertext Markup Language，HTML）和层叠样式表（Cascading Style Sheet，CSS）的复杂知识之前，对技术有深入的理解很重要，它们有助于把这些纯文本文件转换成你在计算机或手持设备上浏览 WWW 时看到的丰富的多媒体显示。

例如，如果不使用 Web 浏览器查看，那么包含标记和客户端代码的 HTML 和 CSS 文件将是无用的，并且除非建立 Web 服务器，否则除你之外的其他人将无法查看你的内容。Web 服务器使你的内容可供其他人使用，他们反过来又使用他们的 Web 浏览器导航到一个地址并等待服务器给他们发送信息。你将密切地参与这个发布过程，因为你必须创建文件，然后把它们放到服务器上，使它们在第一时间可用，并且必须确保你的内容像预期的那样

异步社区会员 Kensuke(18319066421) 专享 尊重版权

出现在最终用户面前。

## 1.1 HTML 和 WWW 简史

在月球上还没有任何人类的足迹之前，一些有远见的人决定看看他们能否连接几个主要的计算机网络。我将和你分享他们的名称和故事（可以说非常丰富），但是最终的结果是“所有网络之母”，我们称之为 Internet。

到 1990 年，通过 Internet 访问信息相当具有技术性。事实上，它是如此困难，以至于那些拥有博士学位的物理学家在尝试交换数据时也通常会遭受挫折。有这样一位物理学家，即现在非常著名的 Tim Berners-Lee 爵士，他设计了一种方式，可以通过超文本链接轻松地交叉引用 Internet 上的文本。

这不是一种新思想，但是他开发的简单 HTML 却成功地兴盛起来，虽然更多雄心勃勃的超文本项目步履维艰。超文本（hypertext）最初意指以电子形式存储的文本，并且在页面之间具有交叉引用的链接。它现在是一个更宽泛的术语，指可以链接到其他对象的任何几何对象（文本、图像、文件等）。超文本标记语言（Hypertext Markup Language）是一种用于描述如何组织和链接文本、图形以及包含其他信息的文件的语言。

到 1993 年，全世界只有 100 台左右的计算机能提供 HTML 页面。这些互联的页面被称为万维网（World Wide Web，WWW），并且编写了几个 Web 浏览器程序，允许人们查看 Web 页面。由于 Web 日益普及，几个程序员不久编写了可以查看文本以及图形图像的 Web 浏览器。从此开始，Web 浏览器软件的持续开发以及 HTML（和 XHTML）语言的标准化带领我们进入了今天我们生活的世界，在这个世界里，有超过 5 亿的 Web 站点提供数十亿的文本和多媒体文件。

上面几段文字实际上描述了 Web 发展简史中的一个不同寻常的阶段。今天的大学生从来也不了解 WWW 不存在的那段时间，并且永远在线的信息和无处不在的计算的思想将对我们以后生活的各个方面产生深远的影响。不要把 Web 内容创建和管理视作少数技术人员（如果你愿意的话，也可以称他们为怪才）所拥有的一种技能，到学完本书时，你将把它们看作是任何人都可以掌握的技能，而不仅限于那些天生的怪才。

**By the Way**

**注意**：有关 WWW 发展历史的更多信息，参见关于这个主题的维基百科的文章：http://en.wikipedia.org/wiki/History_of_the_Web。

## 1.2 创建 Web 内容

你可能注意到本书中使用的术语是“Web 内容”（web content），而不是“Web 页面”（web page），我们是有意这样做的。尽管我们谈论的是“访问 Web 页面”，但是我们实际的意思是“查看我们计算机上的一个地址上的所有文本和图像”。我们阅读的文本和查看的图像都是通过 Web 浏览器呈现的，在单独的文件中可以找到关于它们的某些指导。

这些文件可以包含被 HTML 代码标记（mark up）或包围的文本，告诉浏览器如何显示文本——作为标题、作为段落或者以红色字体显示等。一些 HTML 标记告诉浏览器显示图像

异步社区会员 Kensuke(18319066421) 专享 尊重版权

或视频文件，而不是纯文本，由此得出以下结论：发送给 Web 浏览器的是不同类型的内容，因此仅仅称作 Web 页面开始不能完全涵盖它们。我们在这里代之以使用 Web 内容来涵盖完整范围的文本、图像、音频、视频以及在线发现的其他媒体。

在后面几章中，你将学习链接到或创建在 Web 站点中发现的各种类型的多媒体 Web 内容。此时，你只需记住你正在控制用户在访问你的 Web 站点时所查看的内容。从一个文件开始，其中包含要显示的文本或者告诉服务器给用户的 Web 浏览器发送一幅图形，你将不得不计划、设计和实现最终构成 Web 形象（web presence）的所有部分。正如你将在整本书中所学到的，它不是一个困难的过程，只要在学习过程中理解每一个小步骤即可。

就其最基本的形式而言，Web 内容开始于一个包含 HTML 标记的简单文本文件。在本书中，你将学习和创建符合标准的 HTML5 标记。编写符合标准的代码的许多好处之一是，在将来，将无需担心必须要回顾代码，以从根本上改变它。这样，你的代码将（很可能）总是适用的，只要 Web 浏览器遵守标准即可（在一段较长的时间内有望如此）。

## 1.3 理解 Web 内容递送

在许多不同的位置会发生多个过程，最终将产生你可以查看的 Web 内容。这些过程将发生得非常快（以毫秒级的速度），并且是在幕后发生的。换句话说，尽管我们可能认为我们所做的全部事情是打开 Web 浏览器，输入一个 Web 地址，并立即查看所请求的内容，后台的技术就会代表我们努力工作。图 1.1 显示了浏览器与服务器之间的基本交互。

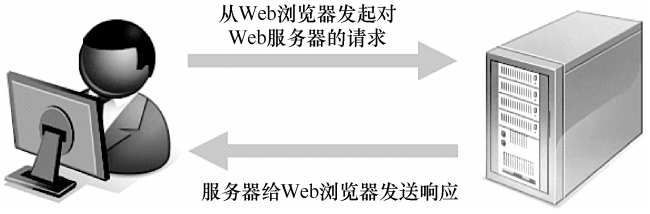



图 1.1
浏览器请求和服务器响应

不过，在看到所请求的站点的全部内容之前，这个进程将会涉及多个步骤，并且可能在浏览器与服务器之间会往返多次。

假设你想执行 Google 搜索，必然会在地址栏中输入“http://www.google.com”，或者从书签列表中选择 Google 书签。浏览器几乎立即就会显示如图 1.2 所示的内容。

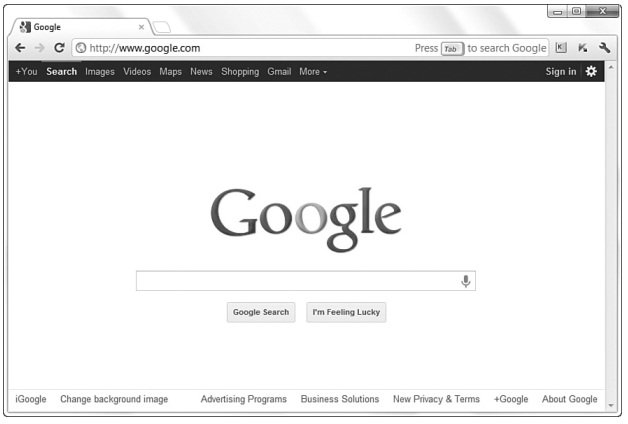



图 1.2
访问 www.google.com

异步社区会员 Kensuke(18319066421) 专享 尊重版权

图 1.2 显示了一个包含文本以及一幅图像（Google 标志）的 Web 站点。用于从 Web 服务器获取文本和图像、并将其显示在屏幕上的过程的简单版本如下。

1. Web 浏览器发送一个对位于 www.google.com 地址上的 index.html 文件的请求。index.html 文件不必是在地址栏中输入的地址的一部分，在学习本章后面的内容时，将会了解关于 index.html 文件的更多知识。
2. 在收到对特定文件的请求后，Web 服务器进程将在其目录内容中寻找特定的文件，打开它，并把该文件的内容发送回 Web 浏览器。
3. Web 浏览器接收 index.html 文件的内容，它是利用 HTML 代码标记的文本，并基于这些 HTML 代码呈现内容。在呈现内容时，浏览器将遇到用于 Google 标志的 HTML 代码，在图 1.2 中可以看到该标志。HTML 代码看起来如下所示。

```
<img alt="Google" src="/images/srpr/logo4w.png" width="275"height="95" />
```

   这个 HTML 标签提供了一些属性，告诉浏览器显示这个标志所必需的信息：文件源地址（src）、宽度（width）、高度（height）、边框类型（border）和替换文本（alt）。在后面的课程中将会学习到关于属性的更多知识。
4. 浏览器查看<img/>标签中的 src 属性，查找源位置。在这种情况下，可以在与浏览器获取 HTML 文件相同的 Web 地址（www.google.com）上的 images 目录中找到 logo3w.png 图像。
5. 浏览器请求 Web 地址 http://www.google.com/images/srpr/logo4w.png 上的文件。
6. Web 服务器解释请求，找到文件，并把该文件的内容发送给请求它的 Web 浏览器。
7. Web 浏览器在显示器上显示图像。

在 Web 内容递送过程的描述中可以看到，Web 浏览器所做的不仅仅是充当用于查看内容的图片框架。浏览器将会依据文件中的 HTML 命令组合 Web 内容成分，并排列那些部分。

也可以在本地或者在你自己的硬盘驱动器上查看 Web 内容，从而无需 Web 服务器。获取和显示内容的过程与上述步骤中列出的步骤相同，这是由于浏览器将寻找并解释 HTML 文件的代码和内容，但是往返行程更短，浏览器将在你自己的计算机的硬盘驱动器上（而不是在远程机器上）寻找文件。如果文件中嵌入了任何基于服务器的程序设计语言，将需要 Web 服务器解释它们，但是这超出了本书的范围。事实上，无需拥有自己的 Web 服务器，就可以顺利学完本书中的所有课程，但是这样的话，除你之外其他任何人都无法查看你的杰作。

## 1.4 选择 Web 托管提供商

尽管刚才告诉你无需具有 Web 服务器也可以顺利地学完本书中的所有课程，我们还是建议你使用 Web 服务器继续学习下面的内容。不要担心，获得托管提供商通常是一个快速、轻松并且相对廉价的过程。事实上，只需支付比你正在阅读的这本书稍微多一点的费用，就能获得你自己的域名和为期一年的 Web 托管。

如果在所选的搜索引擎中输入“web hosting provider”（Web 托管提供商），将获得数百万条纪录和无穷无尽的倡议性搜索结果（也称为广告）。如果不这样，许多 Web 托管提

异步社区会员 Kensuke(18319066421) 专享 尊重版权

供商将从世界上消失。即使查看的托管提供商列表是经过筛选的，它也可能令人不知所措，尤其是当你所寻找的只是一个位置，用于托管你自己或者公司的简单 Web 站点时。

你将希望在寻找提供商时缩小搜索的范围，并且选择最适合自己需要的提供商。对于 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”**：如果具有一种在线形象（online presence），你将希望确保人们实际上能够以一致的方式到达那里。
- **客户服务**：寻找用于联系客户服务的多种方法（电话、电子邮件、聊天），以及针对常见问题的在线文档。
- **服务器空间**：托管包将会包括足够的服务器空间以存放你计划在 Web 站点（如果有的话）中包括的所有多媒体文件（图像、音频、视频）吗？
- **带宽**：托管包将会包括足够的带宽，从而可以满足所有人访问你的站点并下载文件，而无需额外付费吗？
- **域名购买和管理**：托管包将会包括域名吗，或者你必须单独从托管账户购买和维护域名吗？
- **价格**：不要为托管支付过多的费用。你将会看到所提供的广泛价格，并且应该立即就想知道：“它们有什么区别？”通常，它们之间的区别几乎与服务的质量无关，而只与公司的系统开销以及公司认为它可以怎样侥幸地逃避收费有关。一个良好的经验法则是：如果每年为基本的托管包和域名支付 75 美元以上的费用，那么你支付的费用就可能太高了。

下面列出了 3 家可靠的 Web 托管提供商，它们的基本托管包以相对较低的费用提供丰富的服务器空间和带宽（以及域名和额外的好处）。如果你对它们中的任何一家都不感兴趣，至少可以使用它们基本的托管包描述，作为你货比三家的指导原则。

- **A Small Orange（www.asmallorange.com）**：它的 Tiny 和 Small 托管包是新的 Web 内容发布者的完美起点。
- **DailyRazor（www.dailyrazor.com）**：甚至它的 Rookie 个人托管包也是全功能的和可靠的。
- **LunarPages（www.lunarpages.com）**：它的 Basic 托管包非常适合于许多个人和小企业的 Web 站点。

**注意**：作者曾使用过以上三家提供商（后来还使用了其他一些提供商），建议使用其中任何一家提供商都没有问题；她主要使用 DailyRazor 作为 Web 托管提供商，尤其是对于高级开发环境则更是如此。

**By the Way**

良好的托管提供商的一个特点是，它会提供一个“控制面板”，让你管理自己账户的各个方面。图 1.3 显示了作者自己在 Daily Razor 上的托管账户的控制面板。许多 Web 托管提供商都提供了这种特殊的控制面板软件，或者某种在设计上类似的控制面板——清楚标记的图标引导你可以执行的任务，用以配置和管理你的账户。

你可能永远也不需要使用控制面板，但是使之可用将能够简化以下操作：安装数据库及

异步社区会员 Kensuke(18319066421) 专享 尊重版权

其他软件、查看 Web 统计信息和添加电子邮件地址（以及许多其他的特性）。如果你可以遵循指导，就可以管理你自己的 Web 服务器——不需要特殊的培训。

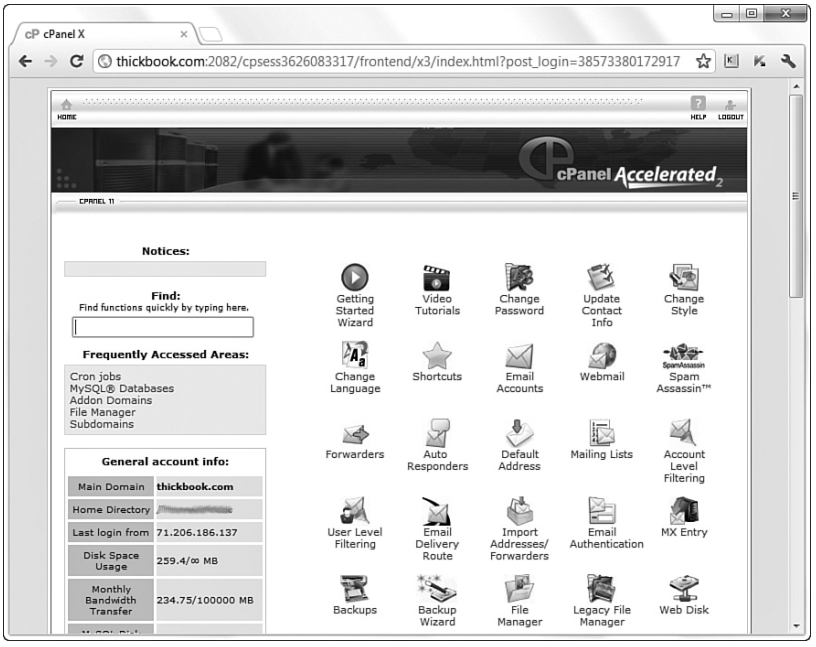



图 1.3
一个控制面板示例

## 1.5 利用多种 Web 浏览器执行测试

刚才讨论了 Web 内容递送的过程以及如何获得 Web 服务器，现在回过头来并讨论利用多种 Web 浏览器测试 Web 站点似乎有点奇怪。不过，在继续学习关于利用 HTML 和 CSS 创建 Web 站点的所有知识之前，要记住下面这句非常重要的话：你的 Web 站点的每位访问者将可能使用与你自己不同的硬件和软件配置。他们的设备类型（台式机、笔记本电脑、上网本、智能手机）、屏幕分辨率、浏览器类型、浏览器窗口大小、连接的速度都可能与你不同。记住：当访问者查看你的站点时，你不能控制他们使用的任何方面。因此，在你设置 Web 托管环境并准备好工作时，要考虑下载多种不同的 Web 浏览器，以便可以使用本地测试工具套件。下面让我解释为什么这一点很重要。

尽管所有的 Web 浏览器都会以相同的常规方式处理信息，但是它们当中的一些特定的区别将导致事情在不同的浏览器中看起来并不总是相同的。甚至相同 Web 浏览器的相同版本的用户也可能通过选择不同的显示选项和/或更改他们的查看窗口的大小，来改变页面的显示方式。所有主流的 Web 浏览器都允许用户利用他们自己的选择覆盖 Web 页面作者指定的背景和字体。当页面第一次出现在人们的显示屏幕上时，屏幕分辨率、窗口大小和可选的工具栏也可能改变他们所看到的页面大小。你只能确保自己编写的是符合标准的 HTML 和 CSS。

在任何情况下，都不要花时间创建只会在你自己的计算机上看起来完美无缺的最终设计——除非你在朋友的计算机上、大街上的咖啡店里的计算机上或者你的 iPhone 上查看它时能够坦然地面对失望。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**注意：**在第 15 章中，你将学习一点响应性 Web 设计的概念，其中将依赖于用户的行为和查看环境（屏幕大小、设备等）自动转换和改变站点的设计。

By the Way

应该总是尽可能多地在下面这些 Web 浏览器上测试你的 Web 站点。

- 用于 Mac 和 Windows 的 Apple Safari（http://www.apple.com/safari/）。
- 用于 Windows 的 Google Chrome（http://www.google.com/chrome）。
- 用于 Mac、Windows 和 Linux 的 Mozilla Firefox（http://www.mozilla.com/firefox/）。
- 用于 Windows 的 Microsoft Internet Explorer（http://www.microsoft.com/ie）。
- 用于 Mac、Windows 和 Linux/UNIX 的 Opera（http://www.opera.com/）。

既然你已经建立了开发环境，或者对将来想要建立的开发环境至少有了某种想法，现在就让我们继续创建一个测试文件。

## 1.6 创建一个示例文件

在开始前，让我们看看程序清单 1.1。这个程序清单代表一份简单的 Web 内容，其中包含几行 HTML 代码，用于在两行上以大号、加粗字母打印“"Hello World! Welcome to My Web Server.”，并在浏览器窗口内居中显示它们。在继续学习本书后面的内容时，你将对这个文件内使用的 HTML 和 CSS 有更多的了解。

**程序清单 1.1 我们的示例 HTML 文件**

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

为了利用这些内容，可以打开所选的文本编辑器，比如“记事本”（在 Windows 上）或 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，因为这些程序创建的文件类型不同于我们用于创建 Web 内容的纯文本文件。

输入在程序清单 1.1 中看到的内容，然后使用 sample.html 作为文件名保存文件。.html 扩展名告诉 Web 服务器你的文件的确完完全全是 HTML。当把文件内容发送给请求它的 Web 浏览器时，浏览器也会知道它是 HTML，并将相应地呈现它。

既然你已经有了一个要使用的示例 HTML 文件——并且希望把它放到某个地方，比如 Web 托管账户，现在就让我们发布你的 Web 内容。

## 1.7 使用 FTP 传输文件

就像你迄今为止所学到的，必须把 Web 内容放到 Web 服务器上，才能使之为其他人访

异步社区会员 Kensuke(18319066421) 专享 尊重版权

问。这个过程通常是使用文件传输协议（File Transfer Protocol，FTP）发生的。要使用 FTP，需要一个 FTP 客户端，这个程序用于把你的计算机上的文件传输到 Web 服务器上。

FTP 客户端需要 3 份信息来连接到 Web 服务器。在你建立了账户之后，你的托管提供商将把这些信息发送给你。

- 你将连接到的主机名或地址。
- 你的账户的用户名。
- 你的账户的密码。

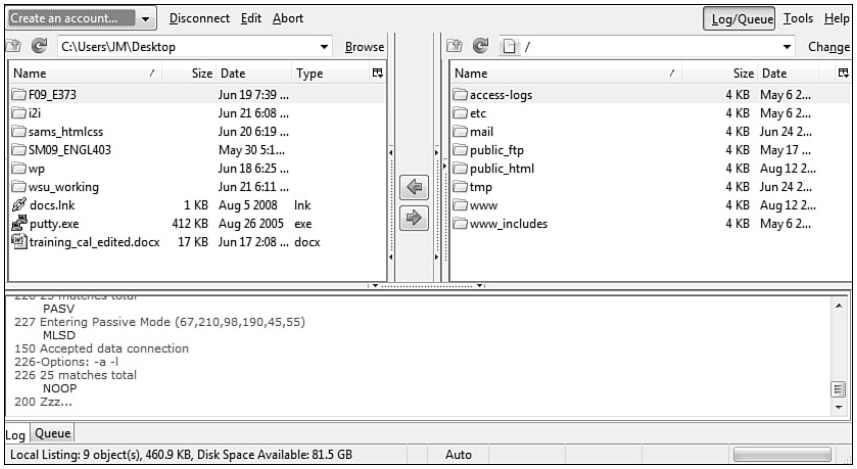
当你具有了这些信息后，就准备好使用 FTP 客户端，把内容传输到 Web 服务器。

## 1.7.1 选择 FTP 客户端

无论使用的 FTP 客户端是什么，它们一般都会使用相同的界面。图 1.4 显示了一个 FireFTP 示例，它是 Firefox Web 浏览器使用的 FTP 客户端。本地机器（你的计算机）的目录清单出现在屏幕左边，远程机器（Web 服务器）的目录清单则出现在右边。通常会看到右箭头和左箭头按钮，右箭头把所选的文件从计算机发送到 Web 服务器上；左箭头则把文件从 Web 服务器发送到计算机上。许多 FTP 客户端还允许简单地选择文件，然后把那些文件拖放到目标机器上。

图 1.4

FireFTP 界面




许多 FTP 客户端可以免费使用，但是也可以通过 File Manager（文件管理）工具传输文件，它很可能是 Web 服务器的控制面板的一部分。不过，这种文件传输方法通常会在过程中引入更多的步骤，并且几乎不像在你自己的计算机上安装 FTP 客户的过程那样流线化（或简单）。

下面列出了一些流行的免费 FTP 客户端。

- 用于 Mac 和 Windows 的 Classic FTP（http://www.nchsoftware.com/classic/）。
- 用于 Mac 的 Cyberduck（http://cyberduck.ch/）。
- 用于 Mac 的 Fetch（http://fetchsoftworks.com/）。
- 用于所有平台的 FileZilla（http://filezilla-project.org/）。
- 用于所有平台的 FireFTP（http://fireftp.mozdev.org/）Firefox 扩展。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

当选择一个 FTP 客户端并在计算机上安装它时，就准备好从 Web 服务器上传和下载文件。在下一节中，将使用程序清单 1.1 中的示例文件解释这个过程的工作方式。

## 1.7.2 使用 FTP 客户端

下面的步骤说明了如何使用 Classic FTP 连接到 Web 服务器并传输文件。不过，所有的 FTP 客户端都使用类似的（如果不是完全相同的话）界面。如果理解了下面的步骤，就应该能够使用任何 FTP 客户端。

记住，首先需要主机名、账户的用户名和账户的密码。

1. 启动 Classic FTP 程序，并单击 Connect 按钮，将提示你填写与你想连接到的站点相关的信息，如图 1.5 所示。

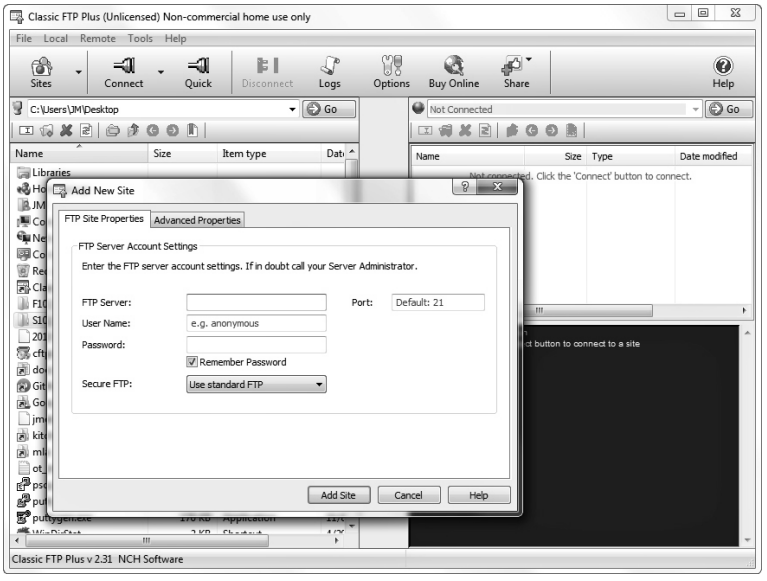


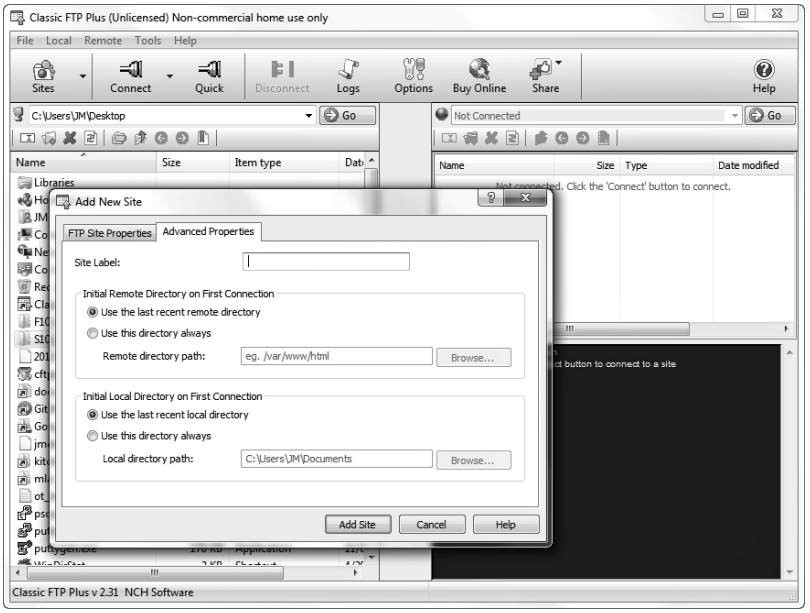

图 1.5
在 Classic FTP 中连接到新的站点

2. 填写图 1.5 中显示的每个项目，如下所示。
   - FTP 服务器是需要发送 Web 页面的 Web 服务器的 FTP 地址，你的托管提供商将给你提供这个地址。它可能是 yourdomain.com，但是要检查你在签订服务合同时接收到的信息。
   - 使用托管提供商提供的信息完成 User Name 框和 Password 框。
3. 你可能要切换到 Advanced 选项卡，并修改以下可选的项目，如图 1.6 所示。
   - Site Label 是你将用于称呼自己站点的名称。其他任何人都不会看到这个名称，因此可以输入你想要的任何名称。
   - 可以更改 Initial Remote Directory on First Connection 和 Initial Local Directory on First Connection 的值，但是你可能想等待，直到你习惯了使用 FTP 客户端并且建立了工作流程为止。
4. 完成设置后，单击 Add Site 保存设置。然后可以单击 Connect，建立一条与 Web 服务器的连接。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

图 1.6

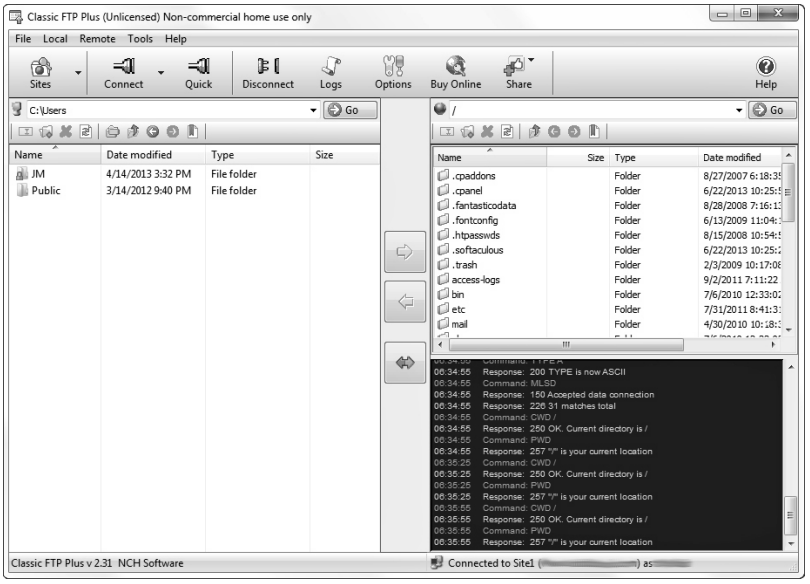
Classic FTP 中的 Advanced 连接选项




你将看到一个对话框，指示 Classic FTP 尝试连接到 Web 服务器。一旦成功连接，将会看到一个如图 1.7 所示的界面，在左边显示本地目录的内容，并在右边显示 Web 服务器的内容。

图 1.7

通过 Classic FTP 成功连接到远程 Web 服务器




5. 现在几乎就准备好将文件传输到 Web 服务器，剩余的全部工作是把目录改为所谓的 Web 服务器的文档根目录（document root）。Web 服务器的文档根目录被指定为 Web 内容的顶级目录，它是目录结构的起点，在本章后面将了解关于它的更多信息。通常，这个目录被命名为 public_html、www（因为 www 被创建为 public_html 的别名）或 htdocs。你自己不必创建这个目录，托管提供商将为你创建它。

   双击文档根目录名称以打开它。FTP 客户界面的右边将变成显示该目录的内容（此时它可能是空的，除非 Web 托管提供商代表你在该目录中放置了占位符文件）。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

6．目标是把你以前创建的 sample.html 文件从你的计算机传输到 Web 服务器上。在 FTP 客户端界面的左边列出的目录中查找文件（如果需要，可以进行导航），并单击它一次，以高亮显示文件名。

7．单击 FTP 客户端界面中间的右箭头按钮，把文件发送到 Web 服务器。当文件传输完成时，客户界面的右边将会刷新，以显示发送到目的地的文件。

8．单击 Disconnect 按钮关闭连接，然后退出 Classic FTP 程序。

无论何时你想通过 FTP 把文件发送到 Web 服务器上，从概念上讲都要采取这些类似的步骤。也可以使用 FTP 客户端在远程 Web 服务器上创建子目录。要使用 Classic FTP 创建子目录，可以单击 Remote 菜单，然后单击 New Folder。不同的 FTP 客户端具有不同的界面选项，来实现相同的目标。

## 1.8 了解在 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将如何组织该内容——不仅便于用户查找，而且便于你维护服务器上的内容。把文件放在目录中有助于管理那些文件。

在 Web 服务器上命名和组织目录以及开发针对文件维护的规则完全取决于你自己。不过，在这个漫长的过程中，维护一个组织良好的服务器可以使其内容管理更高效。

### 1.8.1 基本的文件管理

在你浏览 Web 时，可能注意到当你在 Web 站点中导航时 URL 会改变。例如，如果查看一家公司的 Web 站点，并且单击通往公司的产品或服务的图形导航元素，URL 可能会从

http://www.companyname.com/

变为

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

或

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

在上一节中，我使用了术语文档根目录（document root），但是没有真正解释它的有关含义。Web 服务器的文档根目录实质上是完整 URL 中的末尾斜杠。例如，如果域是 yourdomain.com，并且 URL 是 http://www.yourdomain.com/，那么文档根目录就是通过末尾斜杠（/）表示的目录。文档根目录是你在 Web 服务器上创建的目录结构的起点；Web 服务器将从这个位置开始寻找 Web 浏览器请求的文件。

如果像前面所指示的那样把 sample.html 文件放在文档根目录中，将能够通过 Web 浏览器利用以下 URL 访问它：

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

如果把这个 URL 输入到 Web 浏览器中，将会看到呈现的 sample.html 文件，如图 1.8 所示。

异步社区会员 Kensuke(18319066421) 专享 尊重版权




图 1.8
通过 Web 浏览器访问的 sample.html 文件

不过，如果在文档根目录内创建一个新目录，并把 sample.html 文件放在该目录中，则将利用下面这个 URL 访问该文件：

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

如果把 sample.html 文件放在一连接到你的服务器时就最初看到的目录中，也就是说，你没有改变目录并把文件放在文档根目录中，那么将不能从你的 Web 服务器利用任何 URL 访问 sample.html 文件。该文件仍然位于你称为 Web 服务器的机器上，但是由于文件不在服务器软件知道从此开始寻找文件的文档根目录中，永远都没有人能够通过 Web 浏览器访问到它。

底线就是：在开始传输文件之前，总是要导航到 Web 服务器的文档根目录。

对于图形和其他多媒体文件尤其如此。Web 服务器上的公共目录被称为 images，正如你可能想到的，其中将存放所有的图像资源以便检索。其他流行的目录包括用于存放样式表文件的.css 文件（如果使用多个样式表文件的话），以及用于存放外部 JavaScript 文件的.js 文件。此外，如果你知道在 Web 站点上将具有一个区域，访问者可以从中下载许多不同类型的文件，就可能简单地把该目录命名为 downloads。

无论它是一个包含你的艺术选辑的 ZIP 文件，还是一个带有销售数字的 Excel 电子数据表，在 Internet 上发布不仅仅是 Web 页面的文件通常是有用的。为了使一个不是 HTML 文件的文件在 Web 上可用，只需把该文件像一个 HTML 文件一样上传到 Web 站点，并遵循本章前面给出的关于上传文件的指导即可。在把文件上传到 Web 服务器之后，就可以创建一个指向它的链接（在第 8 章将学习这方面的知识）。换句话说，Web 服务器不仅仅能提供 HTML 文件。

下面给出了一段示例 HTML 代码，在本书后面将学习关于它的更多知识。下面的代码将用于一个名为 artfolio.zip 的文件，它位于 Web 站点的 downloads 目录中，并且会显示链接文本“Download my art portfolio!”：

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

## 1.8.2 使用索引页面

在你想到索引时，可能会想起本书后面的那一节，它告诉你在哪里寻找各个关键词和主题。Web 服务器目录中的索引文件可以服务于此目的——如果你像那样设计它的话。事实上，

异步社区会员 Kensuke(18319066421) 专享 尊重版权

该名称最初就起源于此。

index.html 文件（或者采用它通常的叫法即索引文件（index file））是你给一个页面提供的名称，当人们导航到你的 Web 站点中的特定目录时，你希望他们把该页面视为默认的文件。

索引文件的另一个作用是，访问你的站点上的某个目录的用户将具有一个索引页面，但是没有指定该页面仍将登录到用于站点的那个区域（或者用于站点本身）的主页上。

例如，可以输入以下 URL 之一，并登录到 Apple 的 iPhone 信息页面上：

http://www.apple.com/iphone/

http://www.apple.com/iphone/index.html

iphone 目录中没有 index.html 页面，结果将依赖于 Web 服务器的配置。如果服务器被配置成禁止目录浏览，用户在尝试访问不带有指定的页面名称的 URL 时将会看到一条“Directory Listing Denied”（“目录列表被拒绝”）消息。不过，如果服务器被配置成允许目录浏览，用户将会看到那个目录中的文件列表。

你的托管提供商将已经确定了这些服务器配置选项。如果你的托管提供商允许你通过控制面板修改服务器设置，你就可以更改这些设置，使得服务器基于你自己的需要来响应请求。

索引文件不仅在子目录中使用，也在 Web 站点的顶级目录（或文档根目录）中使用。Web 站点的第一个页面——首页（home page）或主页（main page），或者当用户第一次访问你的域时你希望他们看到的 Web 内容的你所喜欢的任何称呼——都应该被命名为 index.html，并且放在 Web 服务器的文档根目录中。这确保当用户在他们的 Web 浏览器中输入 http://www.yourdomain.com/时，服务器将利用你打算让他们看到的内容来响应（而不是给他们提供“Directory Listing Denied”（目录列表被拒绝）消息或者其他某种意外的后果）。

## 1.9 在不涉及 Web 服务器的情况下分发内容

在线发布 HTML 和多媒体文件显然是学习 HTML 和创建 Web 内容的主要原因。不过，在另外一些情况下，其他的发布形式简直是不可行的。例如，你可能希望在商品展示会上分发 CD-ROM、DVD-ROM 或 USB 驱动器，其中带有设计为 Web 内容的市场营销材料——即可以通过 Web 浏览器查看的超链接的文本，但是不涉及 Web 服务器。你还可能希望在可移动媒体上包括基于 HTML 的指导手册，以便在培训研讨会上给学生演示它们。这些只是说明在不涉及 Internet 的发布场景中如何使用 HTML 页面的两个示例。

这个过程也称为创建本地（local）站点。即使没有涉及 Web 服务器，仍然可以把这些大量的超文本内容称为站点（site）。之所以使用本地这个术语，是因为文件是在本地而不是在远程（通过 Web 服务器）访问的。

### 1.9.1 在本地发布内容

让我们假定你需要创建一个希望在 USB 驱动器上分发的本地站点。现在，甚至最便宜的 USB 驱动器也能够保存如此多的数据（并且基本的超文本文件相当小），以至于可以在一个

异步社区会员 Kensuke(18319066421) 专享 尊重版权

小驱动器上分发整个站点和全功能的 Web 浏览器（fully functioning web browser）。

**By the Way**

**注意：**在创建和分发本地站点时，分发 Web 浏览器不是必需的，尽管这样做可以起到锦上添花的作用。可以合理地假定用户具有他们自己的 Web 浏览器，并且会打开某个目录中的 index.html 文件，开始浏览超链接的内容。不过，如果你希望在 USB 驱动器上分发 Web 浏览器，可以访问 http://www.portableapps.com/，并寻找 Portable Firefox 或 Portable Chrome。

可以简单地把 USB 驱动器的目录结构视作就像是 Web 服务器的目录结构。USB 驱动器的目录结构的顶级目录可以是文档根目录。或者，如果连同内容一起分发 Web 浏览器，就可能具有两个目录，例如，一个名为 browser，另一个名为 content。在这种情况下，content 目录将是文档根目录。在文档根目录内，可以具有额外的子文件夹，可以在其中放置内容和其他多媒体资源。

就像远程站点一样，使本地站点维持组织有序也很重要，以便可以避免 HTML 文件中的断开的链接。在第 8 章中将学习关于把文件链接到一起的具体细节的更多知识。

### 1.9.2　在博客上发布内容

你可能具有一个被第三方（如 WordPress、Tumblr 或 Blogger）托管的博客，因此在没有专用的 Web 服务器或者甚至不知道任何 HTML 的情况下已经发布了内容。除了源代码编辑器（source editor）之外，这些服务还提供可视化编辑器（visual editor），这意味着可以输入你的文字，并添加展示格式化效果，如粗体、斜体或字体颜色，而不必知道用于这些动作的 HTML 代码。当你单击这些编辑器中的 Publish（发布）按钮时，内容仍然会变成实际的 HTML 代码。

不过，利用你在全书中获得的知识，可以增强你写博客的能力，因为你将能够为博客帖子内容和博客模板使用源代码编辑器，从而可以对该内容的外观和感觉提供更多的控制。这些动作的发生不同于你学习用于创建 HTML 文件并通过 FTP 把它上传到你自己的专用 Web 服务器的过程，但是如果我没有指出写博客事实上是一种 Web 发布形式，那我就是不负责任的。

## 1.10　关于测试 Web 内容的提示

无论何时把文件传输到 Web 服务器上或者把它们存放在可移动媒体上以便于本地浏览，都应该立即彻底地测试每个页面。下面的检查表有助于确保 Web 内容像你期望的那样工作。注意：此时，有一些术语你可能还不熟悉，但是在你逐步学习本书并创建更大的项目时，可以回过头来看看这个检查表。

- 在传输文件之前，要在本地机器上测试它们，以确保链接都会工作，并且内容反映了你想要的视觉设计。在把页面传输到 Web 服务器或可移动设备上之后，要把它们都再次测试一遍。
- 利用尽可能多的浏览器执行这些测试，Chrome、Firefox、Internet Explorer、Opera

异步社区会员 Kensuke(18319066421) 专享 尊重版权

和 Safari 就构成了一个良好的列表，并且要同时在 Mac 和 Windows 两种平台上执行测试。如果可能，要在低分辨率（800×600 像素）和高分辨率（1920×1080 像素）下检查页面。

- 在开始测试之前，要关闭 Web 浏览器中的自动图像加载功能，以便可以查看每个页面在没有图形时的外观。检查 alt 标签的消息，然后再打开图像加载功能以加载图像，并且再次仔细地检查页面。
- 使用浏览器的字体大小设置，以不同的字体大小查看每个页面，如果用户利用他们自己的字体规范覆盖了你的字体规范，确保你的布局不会散架。
- 等待每个页面完全加载完成，然后一直向下滚动，以确保所有的图像都出现在它们应该出现的位置。
- 测定每个页面在加载时要花多长的时间。它要花几秒钟以上的时间才能加载完吗？如果是这样，那个页面上的信息是否有足够的价值，从而可以使用户在页面加载完之前不会去往别的地方？诚然，宽带连接很普遍，但是这并不意味着应该加载带有 1MB 图像的页面。

如果你的页面通过了所有这些测试，你就可以高枕无忧了，你的站点已经准备好被公众查看。

## 1.11 小结

本章介绍了使用 HTML 标记文本文件来制作 Web 内容的概念。你还获悉 Web 内容不仅仅是“页面”，Web 内容还包括图像、音频和视频文件。所有这些内容都存在于 Web 服务器上，它是一台通常远离你自己的计算机的远程机器。在你的计算机或其他设备上，你使用 Web 浏览器请求、获取并且最终在屏幕上显示 Web 内容。

你学习了在确定 Web 托管提供商是否适应你的需求时要考虑的标准。在选择了 Web 托管提供商之后，就可以使用 FTP 客户端开始把文件传输到 Web 服务器，你也学习了如何执行该操作。你还学习了一点关于 Web 服务器目录结构和文件管理的知识，以及在给定的 Web 服务器目录中的 index.html 文件的非常重要的用途。此外，你还学习了可以在可移动媒体上分发 Web 内容，并且学习了如何动手构造文件和目录，以实现在不使用远程 Web 服务器的情况下查看内容的目标。

最后，你学习了在把你的工作成果放到 Web 服务器上之后在多种浏览器中对其进行测试的重要性。编写有效的、符合标准的 HTML 和 CSS 代码有助于确保你的站点对于所有的访问者看上去都是相当相似的，但是如果没有接收到来自开发团队之外的潜在用户的输入，那么仍然不应该进行设计——当你是设计团队中的一员时，获取其他人的输入甚至更重要！

## 1.12 问与答

**问：我查看了 Internet 上的一些 Web 页面的 HTML 源代码，它们看上去极难学习。我**

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**必须像一名计算机程序员那样思考，才能够学习这种材料吗？**

**答：**尽管复杂的 HTML 页面看上去可能的确令人畏缩不前，但是学习 HTML 比学习实际的程序设计语言（比如 C++或 Java）要容易得多。HTML 是一种标记语言，而不是程序设计语言。使用它标记文本，以便浏览器可以以某种方式呈现文本。与开发计算机程序相比，这是一组完全不同的思考过程。你实际上不需要任何作为计算机程序员的经验或技能，就能成为一名成功的 Web 内容作者。

许多商业 Web 站点背后的 HTML 代码看上去比较复杂的原因之一是它们很可能是通过可视化 Web 设计工具创建的，该工具是一个“所见即所得”（what you see is what you get，WYSIWYG）编辑器，在某些情况下会使用其软件开发人员告诉它的任何标记（与手工编码相对，其中可以完全控制得到的标记）。在本书中，将从头开始介绍基本的编码，它通常会导致干净的、易于阅读的源代码。可视化 Web 设计工具倾向于使代码难以阅读，以及产生错综复杂并且不符合标准的代码。

**问：运行你建议的所有测试将要花费比创建我的页面更长的时间！我不能利用较少的测试蒙混过去吗？**

**答：**如果你的页面没有打算用于赚钱或者提供重要的服务，那么当它们在某些用户看来很滑稽或者偶尔会产生错误时，它可能不是一个大问题。在这种情况下，只需利用两种不同的浏览器测试每个页面，并且每天访问它一次。不过，如果需要展现专业的图像，除了进行严格的测试之外，将别无选择。

**问：说真的，谁在乎我是怎样组织我的 Web 内容的呢？**

**答：**无论相信与否，你的 Web 内容的组织结构与搜索引擎和站点的潜在访问者密切相关。但是，总之，具有组织有序的 Web 服务器目录结构有助于跟踪你很可能会频繁更新的内容。例如，如果你具有专用的目录用于存放图像或多媒体，就可以确切知道在哪里寻找你想更新的文件，而无需搜寻包含其他内容的目录。

## 1.13 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 1.13.1 问题

1. 你将需要把多少个文件存储在 Web 服务器上，用以产生单个 Web 页面，并且它上面具有一些文本和两幅图像？
2. 在 Web 托管提供商中，要关注的一些特点是什么？
3. 通过 FTP 连接到 Web 服务器需要哪 3 份信息？
4. index.html 文件的用途是什么？
5. Web 站点必须包括一种目录结构吗？

异步社区会员 Kensuke(18319066421) 专享 尊重版权

### 1.13.2 答案

1. 将需要 3 个文件：一个用于 Web 页面本身，它包括文本和 HTML 标记，还有两个文件分别用于每一幅图像。
2. 要关注可靠性、客户服务、Web 空间和带宽、域名服务、站点管理的额外事项和价格。
3. 需要主机名、你的账户的用户名和你的账户的密码。
4. index.html 文件通常是用于 Web 服务器内的某个目录的默认文件。它允许用户访问 http://www.yourdomain.com/somedirectory/，而不必使用末尾的文件名，并且最终仍然会到达合适的位置。
5. 不是。使用一种目录结构以进行文件组织完全取决于你自己，尽管强烈建议这样做，因为它可以简化内容维护。

### 1.13.3 练习

- 按次序获得你的 Web 托管——你将在自己的计算机上通过在本地查看文件来学习本书中的各章内容吗，或者你将使用 Web 托管提供商吗？注意：大多数 Web 托管提供商在你购买托管计划的当天就能使你正常运行自己的站点。
- 如果你使用的是外部托管提供商，那么就使用 FTP 客户端，在 Web 站点的文档根目录内创建一个子目录。把 sample.html 文件的内容粘贴进另一个名为 index.html 的文件中，把<title>与</title>标签之间的文本更改成新的内容，以及把<h1>与</h1>标签之间的文本更改成新的内容。保存文件，并把它上传到新的子目录。使用 Web 浏览器导航到 Web 服务器上的新目录，并且会看到显示的是 index.html 文件中的内容。然后，使用 FTP 客户端，从远程子目录中删除 index.html 文件。利用 Web 浏览器返回到那个 URL，重新加载页面，并且查看在没有 index.html 文件的情况下服务器如何做出响应。
- 使用在上一个练习中创建的相同文件集，把这些文件放在一个可移动的媒体设备上，例如，CD-ROM 或 USB 设备。使用浏览器导航你的示例 Web 站点的这个本地版本，并且考虑你将不得不利用这个可移动媒体分发的指导，以便其他人可以使用它。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 2 章

# 构造 HTML 文档

在本章中你将学到：

- 怎样用 HTML 创建一个简单的 Web 页面；
- 怎样包括每个 Web 页面必须具有的所有 HTML 标签；
- 怎样利用段落和换行符组织页面；
- 怎样利用标题组织内容；
- 怎样使用 HTML5 的语义元素；
- 怎样验证 Web 内容。

在第 1 章中，你基本了解了创建 Web 内容并且在线（或者在本地，如果还没有 Web 托管提供商的话）查看它的背后的过程。在本章中，我们将言归正传，解释必须出现在 HTML 文件中的各种元素，使之在 Web 浏览器中适当地显示。

一般来讲，本章将快速总结 HTML 的基础知识，并给出一些实用的提示，帮助你最大限度地利用你的时间，成为 Web 页面作者和发布者。不过，本章并不全是介绍理论知识，你将看到真实的 Web 页面及其背后的 HTML 标记。

## 2.1 准备工作

下面是你在准备学习本书余下的内容之前，需要做些什么。

1. 获得一台计算机。我使用一台 Windows 笔记本电脑测试示例 Web 内容，并且抓取了本书中的图片，但是你可以使用任何 Windows、Macintosh 或者 Linux/UNIX 机器创建和查看你的 Web 内容。
2. 获得通往 Internet 的连接。无论你是使用拨号、无线还是宽带连接，都不影响你创

异步社区会员 Kensuke(18319066421) 专享 尊重版权

建和查看 Web 内容，但是连接的速度越快，总的体验将越好。提供 Internet 连接的 Internet 服务提供商（ISP）、学校或企业将给你提供详细的信息，可以帮助你正确地设置它。此外，许多公共场所（比如咖啡店、书店和图书馆）还提供了免费的无线 Internet 服务，如果你的笔记本电脑具有 Wi-Fi 网络支持，那么就可以使用它们。

3. 获得 Web 浏览器软件。这是计算机获取和显示 Web 内容所需的软件。如你在第 1 章中所学到的，最流行的浏览器软件（以字母顺序）是：Apple Safari、Google Chrome、Microsoft Internet Explorer、Mozilla Firefox 和 Opera。安装其中多个浏览器是一个好主意，这样就可以试验并且确保内容跨所有这些浏览器看起来是一致的，你不能指望其他人都使用一样的浏览器。

**警告：**尽管所有的 Web 浏览器都以相同的常规方式处理信息，但是它们之间的一些特定的区别意味着事物在不同的浏览器中看起来并不总是相同的。一定要在多个浏览器中检查 Web 页面，以确保它们看起来基本上是一致的。

*Watch Out!*

4. 使用 Web 浏览器在 Internet 上转转，寻找与你要创建的那些内容或外观相似的 Web 站点。注意一些页面是怎样使你受挫的，是什么吸引你并使你继续阅读其他页面，以及是什么使你回过头来反复查看某些页面的。如果特定的主题使你感兴趣，可以考虑使用诸如 Google（www.google.com）或 Bing（www.bing.com）之类的流行的搜索引擎搜索它。

**注意：**如第 1 章中所讨论的，如果计划把 Web 内容放在 Internet 上（与在 CD-ROM 或本地内联网上发布它相对），将需要把它传输到一天 24 小时都连接到 Internet 的计算机上。为你提供 Internet 访问的相同公司或学校也可能提供 Web 空间；如果不是这样，将可能需要给托管提供商付费来获得服务。

*By the Way*

## 2.2 从一个简单的 Web 页面开始

在第 1 章中，你学习到 Web 页面只是一个被 HTML 代码标记（或包围）的文本文件，这些代码告诉浏览器如何显示文本。要创建这些文本文件，可以使用诸如“记事本”（在 Windows 上）或 TextEdit（在 Mac 上）之类的文本编辑器，不要使用“写字板”、Microsoft Word 或其他全功能的文字处理软件，因为它们创建的文件类型与我们用于创建 Web 内容的纯文本文件不同。

在开始工作前，应该从你想要放在 Web 页面上的一些文本开始。

1. 查找（或者编写）几个关于你自己、你的家庭、你的公司、你的垒球队或者你感兴趣的其他某个主题的文本段落。
2. 把这段文本另存为标准的 ASCII 纯文本。“记事本”（在 Windows 上）和大多数简单的文本编辑器总是把文件另存为纯文本，但是如果使用另一个程序，则可能需要选择这种文件类型作为一个选项（在选择“文件”>“另存为”命令之后）。

在学习本章的过程中，将给文本文件添加 HTML 标记（称为标签[tag]），从而把它转变

异步社区会员 Kensuke(18319066421) 专享 尊重版权

成 Web 内容。

在保存包含 HTML 标签的文件时，总是要给它们提供以.html 结尾的名称。这很重要：如果在保存文件时忘记了在文件名末尾输入.html，大多数文本编辑器都会给它提供某个其他的扩展名（比如.txt）。如果发生这种情况，当尝试利用 Web 浏览器查看文件时，也许不能找到它；如果找到文件，也肯定不能正确地显示。换句话说，Web 浏览器期望 Web 页面文件具有.html 文件扩展名并且具有纯文本格式。

**Watch Out!**

**警告：**我们反复申明这一点，因为它对于结果和学习过程本身非常重要：不要利用 Microsoft Word 或者任何其他 HTML 兼容的字处理器创建第一个 HTML 文件；其中大多数程序都试图以怪异的方式为你重写 HTML 代码，这可能会把你完全搞糊涂。在使用 Microsoft Word 和“另存为”HTML 时也是如此——你很可能获得充斥大量 HTML 代码的冗长且不兼容的文件，它将不会经过验证，并且会在编辑时让你头痛。

此外，我还建议不要使用所见即所得（what you see is what you get，WYSIWYG）的图形编辑器，比如 Adobe Dreamweaver。你很可能发现，在开始学习 HTML 时，从一个简单的文本编辑器入手将更容易、更有教育意义。

在访问 Web 站点时，还可能遇到文件扩展名为.htm 的页面，它是另一个可接受的文件扩展名。你还可能发现在 Web 上使用的其他文件扩展名，比如.jsp（Java Server Pages）、.asp（Microsoft Active Server Pages）或.php（PHP：Hypertext Preprocessor），但是这些文件类型使用服务器端技术，它们超出了 HTML 和整本书中的课程的范围。不过，除了程序设计语言之外，这些文件中也包含 HTML。尽管这些文件中的程序设计代码是在服务器端编译的，并且你在客户端看到的一切都是 HTML 结果，但是如果查看源文件，很可能会看到程序设计代码和标记代码错综复杂地交织在一起。

程序清单 2.1 显示了一段可以输入并保存的文本示例，用于创建一个简单的 HTML 页面。如果利用 Chrome 打开这个文件，将会看到如图 2.1 所示的页面。你创建的每个 Web 页面都必须包括一个<!DOCTYPE>声明，以及<html></html>、<head></head>、<title></title>和<body></body>标签对。

**程序清单 2.1 <html>、<head>、<title>和<body>标签**

```
<!DOCTYPE html>
<html 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.
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

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

在程序清单 2.1 中，与在每个 HTML 页面中一样，以“<”开头且以“>”结尾的单词实际上都是编码的命令。这些编码的命令被称为 HTML 标签（HTML tag），因为它们给文本段“加了标签”，并且告诉 Web 浏览器它是什么类型的文本。这允许 Web 浏览器适当地显示文本。

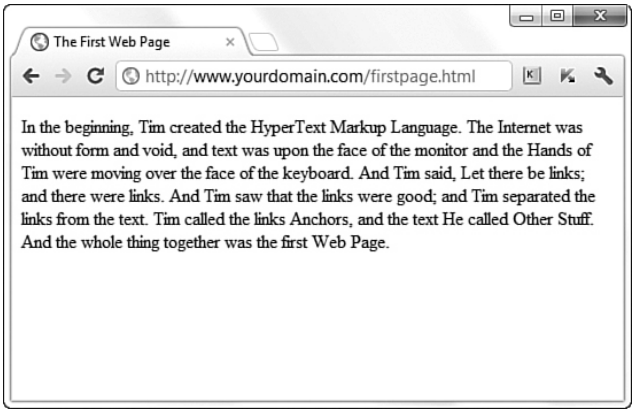



图 2.1

当把程序清单 2.1 中的文本另存为 HTML 文件并利用 Web 浏览器查看时，将只会显示实际的页面标题和正文文本

文档中的第一行是文档类型声明。你声明它是 html（确切地讲是 HTML5），因为 html 是用于在<!DOCTYPE>标签中将文档声明为 HTML5 的值。

**Try It Yourself：创建并查看基本的 Web 页面**

在学习程序清单 2.1 中使用的 HTML 标签的含义之前，你可能想准确地了解我将如何创建并查看文档本身。可遵循下面这些步骤。

1. 在 Windows“记事本”中（或者使用 Macintosh TextEdit 或另一个文本编辑器）输入程序清单 2.1 中的所有文本，包括 HTML 标签。
2. 选择“文件”>“另存为”命令，确保选择纯文本（或 ASCII 文本）作为文件类型。
3. 把文件命名为 firstpage.html。
4. 选择硬盘驱动器上想要用于保存 Web 页面的文件夹，并且记住你所选的文件夹！然后单击“保存”或“确定”按钮保存文件。
5. 现在启动你最喜爱的 Web 浏览器（仍然保持运行“记事本”，以便可以轻松地在查看和编辑页面之间切换）。

   在 Internet Explorer 中，选择“文件”>“打开”命令，并单击“浏览”按钮。如果使用的是 Firefox，则可选择 File>Open File 命令。然后导航到合适的文件夹，并选择 firstpage.html 文件。一些浏览器和操作系统也允许把 firstpage.html 文件拖放到浏览器窗口上以查看它。

   你应该会看到如图 2.1 所示的页面。

如果获得了一个 Web 托管账户，此时就可以使用 FTP，把 firstpage.html 文件传输到 Web 服务器上。事实上，从本章往后，所有的指导都假定你具有托管提供商，并且熟悉通过 FTP 来回发送文件。如果不是这样，请在继续学习下面的内容之前复习一下第 1 章的内容。此外，如果有意选择在本地（没有 Web 主机）处理文件，就要准备调整指导，以适应

异步社区会员 Kensuke(18319066421) 专享 尊重版权

你的特定需求（比如忽略“传输文件”和“输入URL”命令）。

**By the Way**

**注意：**不需要连接到Internet，也可以查看存储在你自己的计算机上的Web页面。默认情况下，每次启动Web浏览器时，它都将尝试连接到Internet，这在大多数时间都是有意义的。不过，如果在硬盘驱动器上本地（脱机）开发页面，这可能会造成麻烦，并且会遇到页面未找到的错误。如果一天24小时都通过LAN、线缆调制解调器、Wi-Fi或DSL连接到Internet，这就是个尚无定论的问题，因为浏览器永远也不会抱怨脱机。否则，就要依赖于浏览器的类型采取合适的动作。可以检查浏览器的Tools（工具）菜单下面的选项。

## 2.3 每个Web页面都必须具有的HTML标签

现在应该给你揭示HTML标签的密语了。当你理解了这种语言之后，将具有超越其他人的创造力。不要告诉其他人，但它确实十分容易。

第一行代码是文档类型声明；在HTML5中，它非常简单，如下：

```
<!DOCTYPE html>
```

这个声明把文档标识为HTML5，这确保Web浏览器可以预期会发生什么事情，并且准备好以HTML5呈现内容。

许多HTML标签具有两个部分：开始标签（opening tag）和封闭标签（closing tag），其中前者指示文本段的开始位置，后者指示文本段的结束位置。封闭标签开始于一个“/”（正斜杠），它位于“<”符号之后。

另一类标签是空标签（empty tag），它是不同的，这是由于它不包括匹配的开始标签和封闭标签对。作为替代，空标签包含单个标签，它开始于“<”，结束于“/>”符号。尽管结尾斜杠在HTML5中不再明确需要，但它确实有助于同XHTML保持兼容。如果在Web站点中具有一堆旧的XHTML代码，在进行升级时，它将不会损坏。

下面快速总结了这3个标签，只是为了确保你理解每个标签所起的作用。

- 开始标签是指示HTML命令开始的HTML标签，受命令影响的文本出现在开始标签之后。开始标签总是开始于“<”，结束于“>”，比如在<html>中。
- 封闭标签是指示HTML命令结束的HTML标签，受命令影响的文本出现在封闭标签之前。封闭标签总是开始于“</”，结束于“>”，比如在</html>中。
- 空标签是在页面中发出HTML命令并且不会封闭任何文本的HTML标签。空标签总是开始于“<”，并且结束于“/>”，比如在<br />和<img />中。

**By the Way**

**注意：**你肯定会注意到，在程序清单2.1中，有一些额外的代码与<html>标签相关联。这种代码包含语言属性（lang），它用于指定与标签相关的额外信息。在这里，它指定HTML内的文本的语言是英语。如果以不同的语言编写代码，可以用相关的语言标识符替换en（用于英语）。

例如，程序清单2.1中的<body>标签告诉Web浏览器页面中的实际正文文本的开始位

异步社区会员 Kensuke(18319066421) 专享 尊重版权

置，</body>则指示其结束位置。<body>与</body>标签之间的所有内容都出现在 Web 浏览器窗口的主要显示区域，如程序清单 2.1 中所示。

浏览器窗口的最上方（见图 2.1）显示页面标题文本，它是位于<title>与</title>之间的任何文本。页面标题文本还会在浏览器的 Bookmarks（书签）或 Favorites（收藏夹）菜单上标识页面，这依赖于你使用的是哪种浏览器。为页面提供页面标题很重要，以便页面的访问者可以正确地给它们建立书签以便将来引用，搜索引擎也使用页面标题来提供指向搜索结果的链接。

在你创建的每个 HTML 页面中都将使用<body>和<title>标签对，因为每个 Web 页面都需要页面标题和正文文本。你还将使用<html>和<head>标签对，它们是程序清单 2.1 中显示的另外两个标签。把<html>放在文档的开始处可以简单地指示文档是一个 Web 页面。末尾的</html>则指示 Web 页面结束了。

在页面内，有一个头部区域和一个主体区域。每个区域分别通过<head>和<body>标签标识。其思想是：页面头部中的信息以某种方式描述页面，但实际上不会被 Web 浏览器显示出来。不过，放在主体中的信息会被 Web 浏览器显示。<head>标签总是出现在页面的 HTML 代码开始处附近，仅仅位于<html>开始标签之后。

<title>标签对用于标识出现在页面头部内的页面标题，这意味着它将放在<head>开始标签之后和</head>封闭标签之前。在后面几章中，你将学习可以放在<head>与</head>之间的其他一些高级的头部信息，比如用于格式化页面的样式表规则。

**Did you Know?**

**提示：**你可能发现创建和保存“裸”页面（也称为骨架[skeleton]页面或模板[template]）很方便，它只具有 DOCTYPE 以及<html>、<head>、<title>和<body>开始和封闭标签，类似于程序清单 2.1 中的文档。以后，无论何时想要创建一个新的 Web 页面，都可以打开该文档作为起点，这样就可以使自己避免每次都要输入所有那些必需标签的麻烦。

程序清单 2.1 中的<p>标签封闭了一个文本段落。只要有可能，就应该把文本块封闭在合适的容器元素中。在本书后面，你将学习关于容器元素的更多知识。

## 2.4 利用段落和换行符组织页面

Web 浏览器在显示 HTML 页面时，它将会忽略换行符或者单词之间的空格数。例如，图 2.2 上方显示的诗歌版本在所有单词之间只显示一个空格，即使在程序清单 2.2 中不是这样输入它的。这是由于 HTML 代码中的额外空白会被自动缩减为一个空格。此外，当文本到达浏览器窗口的边缘时，它将自动换到下一行，而不管在原始 HTML 文件中换行符出现在什么位置。

**程序清单 2.2 包含分段符和换行符的 HTML**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>The Advertising Agency Song</title>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

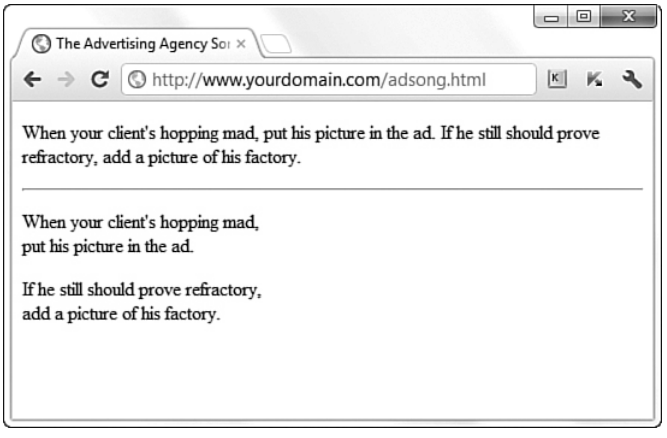
```
  </head>

  <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 2.2

当把程序清单 2.2 中的 HTML 显示为 Web 页面时，换行符和分段符只出现在<br />和 标签中




**By the Way**

**注意：**如果封闭斜杠不是空元素所必需的，你可能会问为什么在整本书中都会使用它。一个原因是：多年来，封闭标签从不是必需的变成必需的，后来又变成不是必需的（再一次），并且本书的作者仍然坚持使用完全有效但是不再必需的封闭斜杠。另一个原因是：由于中间那个时间段相对较长，许多代码编辑器、代码生成器和模板都使用封闭斜杠，因此你将看到使用的时间要多于不使用它的时间。你选择哪种书写方式是无关紧要的，因为两种方式都是有效的，只需确保遵循其中一种编码风格，并在使用时保持一致即可。

如果你想控制换行符和分段符实际出现的位置，则必须使用 HTML 标签。当把文本封闭在 容器标签内时，就假定换行符位于封闭标签之后。在后面各章中，你将学习使用 CSS 控制换行符的高度。<br />标签在段落内强制进行换行。与你迄今为止见过的其他标签不同，<br />不需要封闭标签</br>，这是前面讨论过的那些空标签之一。

程序清单 2.2 和图 2.2 中的诗歌显示了<br />和 标签，它们用于分隔广告代理歌曲的行和诗节。你还可能注意到程序清单中的<hr />标签，它可以导致水平标线出现在页面

异步社区会员 Kensuke(18319066421) 专享 尊重版权

上（见图 2.2）。利用<hr />标签插入水平标线还会导致换行，即使没有与之一起包括一个<br />标签。像<br />一样，<hr />水平标线标签是一个空标签，因此从来不会获得一个</hr>封闭标签。

**Try It Yourself：在 HTML 中格式化文本**

自己动手试试把一段文本格式化为正确的 HTML。

1. 把<html><head><title>My Title</title></head><body>添加到文本的开头（使用你自己的页面标题代替“My Title”）。还要在页面顶部包括样板代码，负责满足标准 HTML 的要求。
2. 在文本的末尾添加</body></html>。
3. 在每个段落开头添加一个 标签，并在每个段落末尾添加一个 标签。
4. 在任何想要单倍行距的位置使用<br />标签。
5. 使用<hr />绘制水平标线，用于分隔主要的文本区域，或者在你想看到一根线条横跨页面的任意位置绘制它。
6. 将文件另存为 mypage.html（使用你自己的文件名代替 mypage）。
7. 在 Web 浏览器中打开文件，查看 Web 内容（如果具有 Web 托管账户，则可通过 FTP 把文件发送给它）。
8. 如果有什么看起来不合适，可以回到文本编辑器中进行校正，并再次保存文件（如果具有 Web 托管账户，则可把文件发送给它）。然后需要单击浏览器中的 Reload（重新加载）/Refresh（刷新）按钮，查看所做的任何改变。

**警告：**如果使用字处理器创建 Web 页面，那么一定要以纯文本或 ASCII 格式保存 HTML 文件。

**Watch Out!**

## 2.5 利用标题组织内容

在浏览 Internet 上的 Web 页面时，你将注意到其中许多页面都在顶部具有一个标题，它们比其余的文本更大、更粗。程序清单 2.3 是用于一个简单的 Web 页面的示例代码和文本，其中包含一个标题示例，它与正常的段落文本形成了鲜明对比。<h1>与</h1>标签之间的任何文本都将显示为大标题。此外，<h2>和<h3>用于创建逐渐变小的标题，依此类推，直到<h6>。

**程序清单 2.3 标题标签**

```
<!DOCTYPE html>

<html 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>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    <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>
```

**By the Way**

**注意：**迄今为止，你可能理解了以下事实：HTML 代码通常会由它的作者进行缩进，以呈现 HTML 文档的不同部分之间的关系，并且便于阅读。这种缩进完全是自愿的——可以轻松地把所有标签集中在一起，并且不使用空格或换行符，当在浏览器中查看时，它们看起来仍然很好。缩进便于你快速查看充满代码的页面，并且理解它们是如何组成一个整体的。缩进代码是另一个良好的 Web 设计习惯，并且最终会使页面更容易维护，对于你自己以及可能接手你留下的工作的任何人都是如此。

在图 2.3 中可以看到，创建标题的 HTML 相当简单。在这个示例中，使用<h1>标签突出显示短语“My Widgets”。为了创建最大的（一级）标题，只需把<h1>标签放在你想用作标题的文本的开头，并把</h1>标题放在它的末尾即可。对于稍小一点的（二级）标题（heading），比如其重要性比页面标题（title）低一些的信息，可以在文本周围使用<h2>和</h2>标签。对于其重要性甚至不及二级标题的内容，可以在文本周围使用<h3>和</h3>标签。

图 2.3

在这个示例产品页面上使用 3 种标题级别显示内容的层次结构




不过，要记住：标题应该遵循内容的层次结构。只使用一个一级标题，在一级标题后面可以具有一个（或多个）二级标题，直接在二级标题后面使用三级标题，等等。不要落入给内容指定标题只是为了使内容以某种方式显示的陷阱。作为替代，要确保合适地对内容进

异步社区会员 Kensuke(18319066421) 专享 尊重版权

行分类（作为主标题、次级标题等），同时使用显示样式使文本在 Web 浏览器中以特定的方式呈现。

理论上讲，还可以使用<h4>、<h5>和<h6>标签创建越来越不重要的标题，但是它们不是经常使用。无论如何，Web 浏览器很少在这些标题与<h3>标题之间显示出显著的区别。尽管可以利用你自己的 CSS 控制它，并且在显示内容时，通常不会需要 6 种标题级别，以显示内容的层次结构。

记住页面标题（title）与标题（heading）之间的区别很重要。在日常英语中，这两个单词通常可以互换使用，但是在谈论 HTML 时，<title>用于给整个页面提供一个标识名称，它不会显示在页面自身上，而只会出现在浏览器窗口的标题栏上。另一方面，标题标签可以使页面上的某些文本在显示时进行视觉强调。每个页面只能有一个<title>，它必须出现在<head>与</head>标签内；另一方面，可以根据需要具有许多<h1>、<h2>和<h3>标题，并且按你设想的那样以任意顺序排列它们。不过，如前所述，应该使用标题标签紧密地控制内容的层次结构，而不要把标题用作一种实现特殊外观的方式，因为这是 CSS 所做的事情。

**警告：**不要忘记，放在 Web 页面头部中的任何内容都不打算在页面上显示出来，而页面主体中的所有内容都打算被查看。

Watch Out!

### 窥探其他设计师的页面

鉴于在许多流行的 Web 页面上存在时尚的视觉效果，有时也有一些吸引人的音频效果，你可能意识到本课程中讨论的简单页面只是 HTML 的冰山一角。既然你已经知道了基础知识，就可能会惊奇自己能够通过查看 Internet 上其他人的页面来学习多少剩余的知识。可以在任何 Web 浏览器中右击并选择 View Source（查看源文件）命令，查看任何页面的 HTML 代码。

如果你自己还不能解释某些 HTML 标签可以做什么或者如何准确地使用它们，也不要担心。随着你继续学习本书后面的内容，所有这些问题都将得到解决。不过，现在预览源代码将显示你知道标签的应用，并且可以让你体验自己很快就能够对 Web 页面做什么。

## 2.6 理解语义元素

HTML5 包括一些标签，它们可让你增强在标记过的文本中所提供信息的语义（含义）。不要简单地把 HTML 用作一种表示语言，就像在 HTML 的早期时代所做的那样，当时把<b>用于粗体以及把<i>用于斜体是一种标准规范，现代的 HTML 的目标之一是把表示与含义分隔开。虽然使用 CSS 提供表示的指导原则，但是 HTML 的创作者可以在他们的标记内为各个元素提供有意义的名称，不仅通过使用 ID 和类名（你将在本书后面学到），而且可以通过使用语义元素来实现这一点。

HTML5 中的一些可用的语义元素如下。

- <header></header>：这似乎是违反直觉的，但是可以在单个页面内使用多个<header>标签。<header>标签应该用作介绍性信息的容器，因此在页面中只可能出

异步社区会员 Kensuke(18319066421) 专享 尊重版权

现一次（很可能位于顶部），但是如果把页面内容分成多个区域，那么也可能多次使用它。任何容器元素都可以具有一个<header>元素，只需确保使用它包括关于其中包含的元素的介绍性信息即可。

- <footer></footer>：<footer>标签用于包含关于其包含元素（页面或区域）的额外信息，比如版权和作者信息，或者指向相关资源的链接。
- <nav></nav>：如果站点具有导航元素，比如指向站点内或者甚至页面自身内的其他区域的链接，这些链接将位于<nav>标签内。通常可以在<header>标签的第一个实例中发现<nav>标签，这只是因为人们倾向于把导航系统放在顶部，并将其视作介绍性信息，但是并非必须如此。你可以把<nav>元素放在任意位置（只要它包括导航系统即可），并且可以根据需要在页面上具有许多<nav>元素（通常不超过两个，但是你可能另有所图）。
- <section></section>：<section>标签包含主题上相关的任何内容，它还可以包含一个用于介绍性信息的<header>标签，并且可能包含一个用于其他相关信息的<footer>标签。可以把<section>视作携带有比标准的<p>（段落）或 （分区）标签更多的含义，后面两个标签通常根本不会传达任何含义，使用<section>可以传达它所包含的内容元素之间的关系。
- <article></article>：<article>标签就像是<section>标签，这是由于它可以包含<header>、<footer>以及其他的容器元素，比如段落和分区。但是利用<article>标签携带的额外含义是它就像报纸或其他出版物中的文章一样。在发表的博客、新闻文章、评论及其他适合这种描述的项目周围可以使用这个标签。<article>与<section>之间的一个关键区别是<article>是一个独立的作品体系，而<section>则是一个主题信息组。
- <aside></aside>：使用<aside>标签指示次级信息，如果<aside>标签位于<section>或<article>内，关系将是那些容器，否则次级关系将是整个页面或站点本身。把<aside>视作侧栏可能是有意义的，针对页面上的所有内容或者针对文章或信息的其他主题容器。

随着你练习使用它们，这些语义元素将变得更清晰。一般而言，使用语义元素是一个好主意，因为它们不仅为你自己以及阅读和处理你的标记的其他设计师和程序员提供了额外的含义，而且为机器也提供了额外的含义。Web 浏览器和屏幕阅读器将会对语义元素做出响应，它们将使用这些元素来确定文档的结构。屏幕阅读器将给用户报告更深层的含义，从而提高材料的可访问性。

在整本书中，除了 HTML 提供的标准类型的块元素之外，你还将把这些及其他语义元素用于页面结构。

## 2.7 验证 Web 内容

在第 1 章中，我讨论了一些测试页面的方式，一种非常重要的测试页面的方式是验证（validate）它们。可以这样考虑它：它就像是设计和绘制一组漂亮的房屋平面图，但是对于建筑师来说则是另一回事，他要将其标记为适合建造的安全结构。验证 Web 页面是一个类

异步社区会员 Kensuke(18319066421) 专享 尊重版权

似的过程。不过，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，验证是利用特殊的应用程序测试页面的过程，它将搜索错误并确保页面遵循严格的 XHTML 标准。验证很简单，事实上，负责开发 Web 标准的标准机构万维网联盟（World Wide Web Consortium，W3C）提供了一个在线验证工具可以让你使用。要验证页面，可以遵循下面这个 URL：http://validator.w3.org/。图 2.4 显示了 W3C 标记验证服务（Markup Validation Service）。

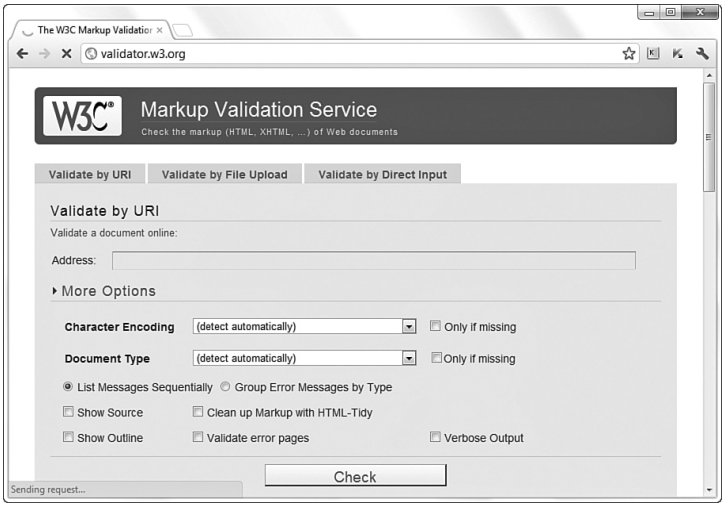



图 2.4

W3C Markup Validation Service 允许验证 HTML（XHTML）文档，以确保它编码正确

如果你已经在线发布了页面，就可以使用 Validate by URI 选项卡。Use the Validate by File Upload 选项卡用于验证存储在本地计算机文件系统上的文件；Validate by Direct Input 选项卡则允许从文本编辑器中粘贴文件的内容。如果一切顺利，页面应该会获得通过验证报告，如图 2.5 所示。

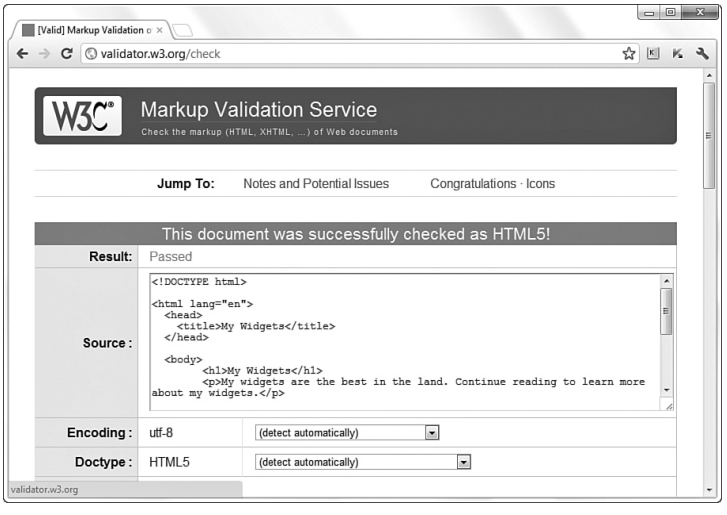



图 2.5

如果页面通过了 W3C Markup Validation Service 的验证，你就知道它已经准备好

如果 W3C Markup Validation Service 在 Web 页面中遇到错误，它将提供具体的详细信息（包括有问题的代码的行号）。这种方式非常适合于追查问题，并清除页面中有错误的代码。验证不仅会通知你页面何时构造正确，而且可以在你发布页面供全世界查看之前帮助你查找并修正问题。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**Did you Know?**

**提示：**一些 Web 开发工具包括内置的验证特性，可以使用它们代替 W3C Markup Validation Service。一些示例包括浏览器扩展，比如 Firebug（http://getfirebug.com/）和 HTML Validator（http://users.skynet.be/mgueury/mozilla/），但是许多其他的程序也提供了类似的功能——可以检查用户文档。

## 2.8 小结

本章介绍了 Web 页面是什么，以及它们如何工作的基础知识。读者学习了编码的 HTML 命令包括在文本文件中，并且看到自己输入 HTML 文本要好于使用图形编辑器为你创建 HTML 命令，尤其是在学习 HTML 语言时。

本章介绍了最基本、最重要的 HTML 标签。通过把这些编码的命令添加到任何纯文本文档中，可以快速把它转换成真正的 Web 页面。读者学习了创建 Web 页面的第一步是把几个必需的 HTML 标签放在开头和末尾，包括为页面添加页面标题。然后可以标记段落和行的结束位置，并且添加水平标线和标题（如果想要它们的话）。读者还体验了 HTML5 中的一些语义标签，它们通过描述页面包含的内容的类型（而不仅仅是内容本身），用于提供额外的含义。表 2.1 总结了本章中介绍的所有标签。

表 2.1 本章介绍的 HTML 标签

| 标签 | 功能 |
|---|---|
| <html>...</html> | 封闭整个 HTML 文档 |
| <head>...</head> | 封闭 HTML 文档的头部。在<html>标签对内使用 |
| <title>...</title> | 指示文档的页面标题。在<head>标签对内使用 |
| <body>...</body> | 封闭 HTML 文档的主体。在<html>标签对内使用 |
| ... | 段落。在段落之间跳过一行 |
| <br /> | 换行符 |
| <hr /> | 水平标线 |
| <h1>...</h1> | 1 级标题 |
| <h2>...</h2> | 2 级标题 |
| <h3>...</h3> | 3 级标题 |
| <h4>...</h4> | 4 级标题（很少使用） |
| <h5>...</h5> | 5 级标题（很少使用） |
| <h6>...</h6> | 6 级标题（很少使用） |
| <header>...</header> | 页面或区域的头部 |
| <footer>...</footer> | 页面或区域的脚注 |
| <nav>...</nav> | 页面或整个站点的导航链接 |
| <section>...</section> | 内容的主题组 |
| <article>...</article> | 一段独立的内容，比如新闻故事 |
| <aside>...</aside> | 与包含它的内容相关的辅助内容 |

异步社区会员 Kensuke(18319066421) 专享 尊重版权

最后，本章学习了验证 HTML 的重要性以及如何使用在线验证工具执行该操作。

## 2.9 问与答

**问：我创建了 Web 页面，但是当我在 Web 浏览器中打开文件时，我看到的全是文本，包括 HTML 标签。有时，我曾经在页面顶部看到过怪异的天书一样的字符。我做错了什么？**

**答：**你没有将文件另存为纯文本。尝试再次保存文件，要小心将其另存为 Text Only（纯文本）或 ASCII Text（ASCII 文本）。如果不能完全搞清楚如何让字处理器执行该操作，也不要紧张。只需代之以在“记事本”或 TextEdit 中输入 HTML 文件，一切都应该会工作得很好（此外，总是要确保 Web 页面的文件名以.html 或.htm 结尾）。

**问：我在 Internet 上看到过一些 Web 页面在开始处没有<!DOCTYPE>或<html>标签。你说过，页面总是必须以这些标签开头，这是怎么回事？**

**答：**如果你忘记了包括<!DOCTYPE>或<html>标签，许多 Web 浏览器将会原谅你，并且无论如何都会正确地显示页面。不过，还是应该写上它们，因为一些软件确实需要它，以将页面识别为有效的 HTML。此外，你还希望页面是真正的 HTML 页面，以使它们遵守最新的 Web 标准。

## 2.10 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 2.10.1 问题

1. 每个 HTML 页面都需要哪 5 个标签？
2. 用于产生以下主体内容的 HTML 标签和文本是什么？
   - 具有单词“We are Proud to Present...”的标题。
   - 具有一个单词“Orbit”的次级标题。
   - 具有单词“The Geometric Juggler”的不太重要的标题。
3. 你将使用什么代码创建一个完整的 HTML Web 页面，它具有页面标题“Foo Bar”，顶部有一个标题“Happy Hour at the Foo Bar”，然后是单词“Come on down!”，它采用常规字体？尝试使用你刚刚学过的一些语义元素。

### 2.10.2 答案

1. 每个 HTML 页面都需要<html>、<head>、<title>和<body>标签（以及它们的封闭标签</html>、</head>、</title>和</body>），在第一行还需要<!DOCTYPE html>标签。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

2．主体内的代码看起来如下所示：

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

3．你的代码可能看起来如下所示：

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Foo Bar</title>
  </head>

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

## 2.10.3 练习

- 即使你阅读本书的主要目标是为你的企业创建 Web 内容，也可能希望仅仅出于实践的目的而创建个人 Web 页面。输入几个段落，向全世界介绍你自己，并使用你在本章中学过的 HTML 标签，把它们转变成 Web 页面。
- 在整本书中，你将依照代码示例创建你自己的页面。现在花一点时间建立一个基本的文档模板，其中包含文档类型声明以及用于创建核心 HTML 文档结构的标签。这样，无论何时需要，都可以准备好复制并粘贴该信息。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 3 章

# 理解层叠样式表

在本章中你将学到：

- 怎样创建基本的样式表；
- 怎样使用样式类；
- 怎样使用样式 ID；
- 怎样构造内部样式表和内联样式。

上一章学习了 HTML 的基础知识，包括如何为所有的 Web 内容建立一个骨架式的 HTML 模板。本章将学习如何使用层叠样式表（Cascading Style Sheet，CSS）微调 Web 内容的显示。

样式表背后的概念很简单：创建一个样式表文档，指定字体、颜色、间距和其他特征，用于建立 Web 站点的独特外观；然后将每个应该具有那种外观的页面链接到样式表，而不是在每个单独的文档中重复指定所有那些样式。因此，当你决定更改公司官方字体或配色方案时，只需更改样式表中的一两个条目，即可同时修改所有的 Web 页面，而不必在所有的静态 Web 文件中更改它们。因此，样式表（stylesheet）是一个格式化指令组，可以同时控制多个 HTML 页面的外观。

样式表允许设置大量的格式化特征，包括准确的字体控制、字间距和行间距以及边距和页面边框等，不一而足。样式表还允许以熟悉的单位指定尺寸及其他度量标准，比如英寸、毫米、磅和派卡（pica）。此外，你还可以使用样式表在 Web 页面上的任意位置精确定位图形和文本，可以通过特定的坐标或者相对于页面上的其他项目来进行。

简而言之，样式表带给 Web 一种高级的显示方式，并且它们是利用样式做到这一点的。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 3.1 CSS 的工作方式

样式表背后的技术称为 CSS 或层叠样式表。CSS 是一种定义样式构造的语言，比如字体、颜色和定位，它们描述了如何在 Web 页面上格式化和显示信息。CSS 样式可以直接存储在 HTML Web 页面中，或者存储在单独的样式表文件中。无论采用哪种方式，样式表都将包含一些样式规则，对给定类型的元素应用样式。当在外部使用时，将把样式表规则存放在外部样式表文档中，其文件扩展名为.css。

**By the Way**

**注意：**如果具有 3 个或更多的 Web 页面共享类似的格式化和字体，当你阅读了本章内容后，就可能想为它们创建一个样式表。即使你选择不创建完整的样式表，也会发现直接在 Web 页面内对各个 HTML 元素应用样式是有用的。

样式规则（style rule）是一种格式化指令，可以应用于 Web 页面上的元素，比如文本段落或链接。样式规则由一个或多个样式属性以及它们关联的值组成。内部样式表（internal stylesheet）直接存放在 Web 页面内，而外部样式表（external stylesheet）则存在于单独的文档中，并通过一个特殊的标签简单地链接到 Web 页面，稍后将介绍关于这个标签的更多知识。

层叠样式表（CSS）这个名称中的“层叠”部分是指对 HTML 文档中的元素应用样式表规则的方式。更确切地讲，CSS 样式表中的样式构成了一种层次结构，其中更具体的样式将覆盖更一般的样式。由 CSS 负责依据这种层次结构来确定样式规则的优先级，它将建立一种层叠效果。如果这听起来有点令人糊涂，只需把 CSS 中的层叠机制视作类似于基因遗传，其中一般的特点将从父母传递给孩子，但是更具体的特点则完全特定于孩子。基本样式规则是在整个样式表中应用的，但是可以被更具体的样式规则覆盖。

下面用一个示例进行说明。查看下面的代码，看看你是否可以断定文本的颜色将会发生什么事情：

```
<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>
```

在前一个示例中，通过 color 样式属性将绿色应用于 标签。因此， 标签中的文本被涂成绿色。由于两个<p>标签都是 标签的孩子，绿色文本样式将层叠到它们下方。不过，第一个<p>标签会覆盖颜色样式，并把它变成蓝色。最终的结果是：第一行（未被段落标签包围）是绿色，第一个正式的段落是蓝色，第二个正式的段落则会保留层叠的绿色。

**By the Way**

**注意：**你可能注意到我在本章中相当多地使用了元素（element）这个术语（在本书余下部分也是这样做的）。元素仅仅只是 Web 页面中的一份信息（内容），比如图像、段落或链接等。标签用于标记元素，可以把元素视作标签，并且在标签内带有描述性信息（如属性、文本、图像等）。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

如果你自己通过这段描述来创建它，并且最终未遭受挫折，那么要祝贺你，这就成功了一半。理解 CSS 不像理解火箭科学，实践得越多，它就会变得越清晰。真正的技巧是开发具有美感的设计，然后可以通过 CSS 将其应用于你的在线状态（online presence）。

像许多 Web 技术一样，CSS 已经演进多年了。CSS 的原始版本（称为 CSS1[Cascading Style Sheets Level 1，层叠样式表级别 1]）创建于 1996 年。后来的 CSS2 标准创建于 1998 年，它在今天仍然在使用，所有现代的 Web 浏览器都支持 CSS2。最新的 CSS 版本是 CSS3，它建立在由其以前的版本打下的强大基础之上，但是添加了一些高级功能，用于增强在线体验。在整本书中，你将学习核心 CSS，包括 CSS3 的新元素，它们适用于本书介绍的基本设计和功能。因此，当我在整本书中谈论 CSS 时，都指的是 CSS3。

在 http://www.w3.org/Style/CSS/上可以找到 CSS 的完整参考指南。本章余下的内容将解释良好地应用 CSS 的基础知识。

## 3.2 基本的样式表

尽管样式表具有强大的威力，但是它们非常容易创建。考虑图 3.1 和图 3.2 中显示的 Web 页面。这些页面共享了几个视觉属性，可以把它们放入一个公共样式表中。

- 它们为标题使用大号、加粗的 Verdana 字体，并为正文文本使用正常大小和粗细的 Verdana 字体。
- 它们使用了一幅名为 logo.gif 的图像，它浮动在内容内，并且出现在页面右边。
- 除子标题外，所有其他的文本都是黑色的，而子标题是紫色的。
- 它们在左边和上边都具有边距。
- 它们在文本行之间具有垂直空间。
- 它们包括一个脚注，它居中显示，并且采用较小的印刷字体。

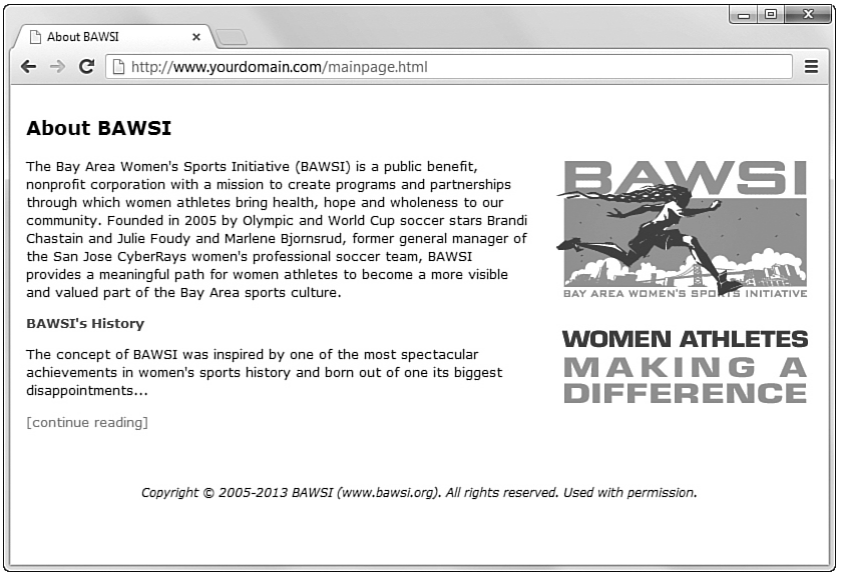



图 3.1 这个页面使用样式表来微调文本和图像的外观和间距

异步社区会员 Kensuke(18319066421) 专享 尊重版权

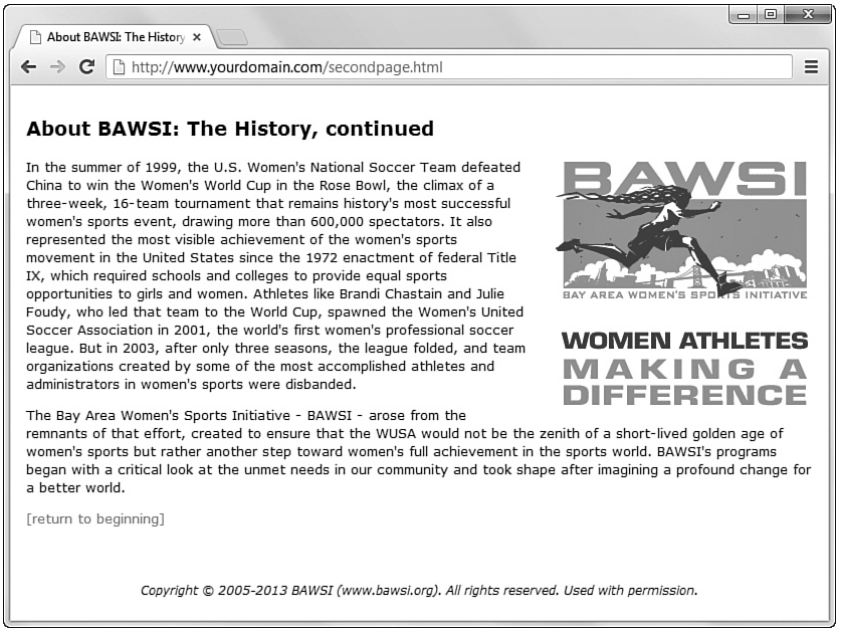



图 3.2

这个页面使用与图 3.1 中相同的样式表，从而维持了一致的外观和感觉

程序清单 3.1 显示了样式表中使用的 CSS，用于指定这些属性。

### 程序清单 3.1 单个外部样式表

```
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;
}
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

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

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

这最初看上去可能有许多代码，但是如果仔细观察，就会看到每一行代码上并没有许多信息。把各个样式规则放在它们自己的行上是相当标准的做法，有助于使样式表更容易阅读，但这是一种个人偏好，也可以把所有的规则都放在一行上，只要保持使用分号隔开每个规则即可（稍后将介绍更多相关知识）。谈到代码的可读性，关于这种样式表代码，也许你将注意到的第一件事是：它看起来一点也不像正常的 HTML 代码。CSS 完全使用它自己的语法来指定样式表。

当然，程序清单中也包括一些熟悉的 HTML 标签（尽管并非所有的标签都需要样式表中的一个条目）。如你可能猜到的，样式表中的 body、h1、p、img、a 和 footer 指示将应用样式表的 HTML 文档中的对应标签。每个标签名后面的大括号描述了那个标签内的所有内容应该如何显示。

在这里，样式表指示所有的 body 文本都应该以 10 磅的大小和 Verdana 字体（如果可能的话）显示，并且颜色为黑色，行间距为 14 磅。如果用户没有安装 Verdana 字体，样式表中的字体列表代表浏览器应该搜索要使用的字体的顺序：依次是 Geneva、Arial 和 Helvetica。如果用户所有这些字体都没有安装，浏览器将使用任何可用的默认 sans serif（无衬线）字体。此外，页面的左、右和上边距各为 5 磅。

<h1>标签内的任何文本都应该以 14 磅的大小和加粗的 Verdana 字体显示。而且，任何只使用<p>标签的段落都将继承通过主体元素指示的样式。不过，如果<p>标签使用名为 subheader 的特殊类，文本将以粗体显示，并且颜色为#593d87（紫色）。

程序清单中每个度量尺寸后面的 pt 指磅（point）（每英寸有 72 磅）。如果你喜欢，也可以用英寸（in）、厘米（cm）、像素（px）或“字母 m 的宽度”（em）指定任何样式表度量尺寸。

你可能注意到程序清单中的每个样式规则都以分号（;）结尾。分号用于把样式表相互分隔开。因此，利用分号结束每个样式规则就是一种习惯做法，以便可以轻松地在它后面添加另一个样式规则。审阅程序清单 3.1 中的样式表的余下内容，查看应用于额外标签的表示格式化。不要担心，在整本书中都将学习关于所有这些条目的更多知识。

**注意：**利用样式表，可以根据自己的意愿指定字体大小，尽管一些显示设备和打印机将不会正确地处理大小超过 200 磅的字体。

**By the Way**

要把这个样式表链接到 HTML 文档，可以在每个文档的<head>区域中包括一个<link />标签。程序清单 3.2 显示了用于图 3.1 中所示页面的 HTML 代码，它包括以下<link />标签：

异步社区会员 Kensuke(18319066421) 专享 尊重版权

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

这假定样式表存储在与 HTML 文档相同文件夹中一个名为 styles.css 的文件下。只要 Web 浏览器支持样式表（并且所有的现代浏览器都支持），样式表中指定的属性将应用于页面中的内容，而无需任何特殊的 HTML 格式化代码。这满足了 HTML 的目标之一，就是把 Web 页面中的内容与用于显示该内容所需的特定格式化分隔开。

**程序清单 3.2 用于图 3.1 中所示页面的 HTML 代码**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>About BAWSI</title>
    <link rel="stylesheet" type="text/css" href="styles.css" />
  </head>
  <body>
     <section>

     <header>
     <h1>About BAWSI</h1>
     </header>

     <p><img src="logo.gif" alt="BAWSI logo"/>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>
     </section>


     <footer>
     Copyright &copy; 2005-2013 BAWSI (www.bawsi.org).
     All rights reserved. Used with permission.
    </footer>
  </body>
</html>
```

***Did you Know?***

**提示：**在大多数 Web 浏览器中，可以通过打开.css 文件并选择“记事本”或另一个文本编辑器作为辅助应用程序查看文件，来查看样式表中的样式规则（要确定.css 文件的名称，可以查看链接到它的任何 Web 页面的 HTML 源代码）。要编辑自己的样式表，只需使用一个文本编辑器即可。

程序清单 3.2 中的代码很有趣，因为它不包含任何类型的格式化。换句话说，HTML 代码中没有任何命令指示文本和图像将如何显示——没有颜色、没有字体，什么也没有。然而

异步社区会员 Kensuke(18319066421) 专享 尊重版权

仍然会仔细地格式化页面并呈现给屏幕，这要归功于指向外部样式表 styles.css 的链接。这种方法的真正好处是：可以轻松地创建具有多个页面的站点，并且维持一致的外观和感觉。并且还具有把页面的视觉样式隔离到单个文档（样式表）中的好处，这样执行一次更改就能够影响所有的页面。

**注意：**并非所有的浏览器对 CSS 的支持都是毫无瑕疵的。要对比主流浏览器对 CSS 的支持有何不同，可以查看下面这个 Web 站点：http://www.quirksmode.org/css/contents.html。

By the Way

Try It Yourself：创建你自己的样式表

从头开始，创建一个名为 mystyles.css 的新文本文档，并为下面这些基本的 HTML 标签创建一些样式规则：<body>、<p>、<h1>和<h2>。在创建了样式表之后，创建一个包含这些基本标签的新 HTML 文件。试验不同的样式规则，看看在样式表文件中利用一处简单的更改来修改段落中的整个文本块有多容易。

## 3.3 CSS 样式的基础知识

你现在已经基本了解了 CSS 样式表，以及它们如何基于描述 Web 页面中的信息的外观的样式规则。本章中的下面几节快速概述了一些最重要的样式属性，并且允许你开始在自己的样式表中使用 CSS。

CSS 包括几种样式属性，它们用于控制字体、颜色、对齐方式和边距等，不一而足。CSS 中的样式属性可以一般可以分成两大类。

- 布局属性，由影响 Web 页面上的元素定位的属性组成，比如边距、填充和对齐方式。
- 格式化属性，由影响 Web 站点内的元素的视觉显示的属性组成，比如字体类型、大小和颜色。

### 3.3.1 基本的布局属性

CSS 布局属性确定了如何在 Web 页面上放置内容。最重要的布局属性之一是 display 属性，它描述了如何相对于其他元素显示一个元素。display 属性具有 4 个基本的值。

- block：在新行上显示元素，比如在一个新段落中。
- list-item：在新行上显示元素，并在其旁边带有一个列表项标记（项目符号）。
- inline：利用当前段落内联显示元素。
- none：不显示元素，它是隐藏的。

**注意：**display 属性依赖于称为相对定位（relative positioning）的概念，它意味着元素将相对于页面上的其他元素进行定位。CSS 还支持绝对定位（absolute positioning），它允许独立于其他元素把一个元素放在页面上的精确位置。在第 14 章中将学习关于这些定位类型的更多知识。

By the Way

异步社区会员 Kensuke(18319066421) 专享 尊重版权

如果形象地表示出每个元素在显示时在 Web 页面上所占据的矩形区域——display 属性可以控制这个矩形区域的显示方式，将更容易理解 display 属性。例如，block 值将导致元素独自放在一个新行上，而 inline 值则把元素放在它前面的内容旁边。display 属性是可以在大多数样式规则中应用的少数几个样式属性之一。下面给出了一个如何设置 display 属性的示例：

```
display:block;
```

可以利用 width 和 height 属性控制用于元素的矩形区域的大小。和许多与大小相关的 CSS 属性一样，可以用多种不同的度量单位指定 width 和 height 属性值。

- in：英寸。
- cm：厘米。
- mm：毫米。
- %：百分比。
- px：像素。
- pt：磅。

无论在样式表内怎样选择，都可以混用和匹配单位，但是在一组类似的样式属性当中保持一致是一个好主意。例如，你可能想坚持为字体属性使用磅，并为尺寸使用像素。下面给出了一个使用像素单位设置元素宽度的示例：

```
width: 200px;
```

### 3.3.2 基本的格式化属性

CSS 格式化属性用于控制 Web 页面上的内容的外观，与控制内容的物理定位相对。最流行的格式化属性之一是 border 属性，它利用一个方框或者部分方框在元素周围建立一条可见的边界。注意：边框总是存在，这是由于总是为其预留了此空间，但是边框不会以你可以看见的方式出现，除非给它提供使之可见的属性（比如颜色）。下面的 border 属性提供了一种描述元素边框的方式。

- border-width：边框边缘的宽度。
- border-color：边框边缘的颜色。
- border-style：边框边缘的样式。
- border-left：边框的左边。
- border-right：边框的右边。
- border-top：边框的上边。
- border-bottom：边框的下边。
- border：边框的各条边。

border-width 属性建立边框边缘的宽度，通常用像素表示，如下面的代码所示：

```
opborder-width:5px;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

border-color 和 border-style 属性用于设置边框的颜色和样式。下面给出了如何设置这两个属性的示例：

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

可以把 border-style 属性设置为以下基本的值之一（在本书后面将会学习一些更高级的边框技巧）。

- solid：单线边框。
- double：双线边框。
- dashed：短划虚线边框。
- dotted：点线边框。
- groove：具有沟槽外观的边框。
- ridge：具有脊状外观的边框。
- inset：具有内嵌外观的边框。
- outset：具有外凸外观的边框。
- none：无边框。
- hidden：实际上等同于 none。

border-style 属性的默认值是 none，这就是元素为什么没有边框的原因，除非把边框属性设置为一种不同的样式。尽管 solid 是最常见的边框样式，也可以看到其他样式在使用。

border-left、border-right、border-top 和 border-bottom 属性允许单独为元素的每一边设置边框。如果想让边框在全部 4 条边上都显示相同的样式，可以只使用单个 border 属性，它期望以下通过空格隔开的样式：border-width、border-style 和 border-color。下面是一个使用 border 属性的示例，它把边框设置包含两条（双）红线，它们的宽度总共为 10 像素：

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

元素边框的颜色是利用 border-color 属性设置的，而元素的内部区域的颜色则是使用 color 和 background-color 属性设置的。color 属性设置元素中的文本（前景）的颜色，background-color 属性则设置文本后面的背景的颜色。下面给出了一个示例，显示了把两个颜色属性设置为预先定义的颜色：

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

也可以利用十六进制值（将在第 7 章中更详细地介绍）或者 RGB（Red Green Blue，红、绿、蓝）十进制值指定颜色，给这些属性分配自定义的颜色：

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

不用太费事，还可以控制 Web 页面内容的对齐方式和缩进。这是利用 text-align 和 text-indent 属性完成的，如下面的代码所示：

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

在适当地对齐和缩进元素之后，就可能有兴趣设置它的字体。下面的基本字体属性可

异步社区会员 Kensuke(18319066421) 专享 尊重版权

以设置与字体关联的多个参数（在第 6 章中将学习一些更高级的字体用法）。

- font-family：字体系列。
- font-size：字体的大小。
- font-style：字体的样式（normal 或 italic）。
- font-weight：字体的粗细（normal、lighter、bold、bolder 等）。

font-family 属性指定字体系列名称的优先级列表。使用优先级列表代替单个值，以防某种字体在给定的系统上不可用时可以提供替代选择。font-size 属性使用某个度量单位（通常是磅）指定字体的大小。最后，font-style 属性设置字体的样式，font-weight 属性则设置字体的粗细。下面给出了一个设置这些字体属性的示例：

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

既然你已经知道了相当多的样式属性以及它们的工作方式，现在可以回顾一下程序清单 3.1，并且查看它是否具有更多的意义。下面简要重述一下该样式表中使用的样式属性，你可以把它用作理解样式表如何工作的指南。

- font：可让你同时设置许多字体属性。可以指定通过逗号分隔的字体名称的列表，如果第一个字体名称不可用，就试试下一个，依此类推。还可以包括单词 bold 和/或 italic 或字体大小。此外，还可以利用 font-family、font-size、font-weight 和 font-style 单独设置其中每个字体属性。
- line-height：在出版界也称为行距（leading）。它用于设置每个文本行的高度，通常以磅为单位。
- color：使用标准的颜色名称或十六进制颜色代码设置文本颜色（参见第 7 章，了解更多详细信息）。
- text-decoration：用于关闭链接的下划线，可以简单地将其设置为 none。也支持 underline、italic 和 line-through 这些值。第 8 章将更详细地介绍对链接应用样式。
- text-align：将文本对齐到左边（left）、右边（right）或中间（center），以及利用 justify 值两端对齐文本。
- padding：给元素的左边、右边、上边和下边添加填充，可以用度量单位或页面宽度的百分比表示这个填充。如果想独立给元素的左边和右边添加填充，可以使用 padding-left 和 padding-right，也可以根据需要使用 padding-top 或 padding-bottom 给元素的上边或下边添加填充。在第 13 章以及第 14 章中将学习关于这些样式属性的更多知识。

## 3.4 使用样式类

本书是一本自学用书，因此不必进入单独某一章来学习如何给页面提供极好的样式，尽管确实需要学习某个样式类是什么。无论何时希望页面上的一些文本看起来与其他文本有

异步社区会员 Kensuke(18319066421) 专享 尊重版权

所不同，都可以创建一个规则，其作用相当于自建的 HTML 标签。你定义的每一类特殊格式化的文本被称为样式类（style class）。样式类是一组自定义的格式化规范，可以应用于 Web 页面中的任何元素。

在向你展示样式类之前，我需要快速回顾一下所学的内容并阐释一些 CSS 术语。首先，CSS 样式属性（style property）是一种特定的样式，可以为其指定一个值，比如 color 或 font-size。可以使用选择器把样式属性和它各自的值与 Web 页面上的元素关联起来，选择器（selector）用于标识应用样式的页面上的标签。下面给出了选择器、属性和值的示例，它们都包括在一个基本的样式规则中：

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

在这行代码中，h1 是选择器，font 是样式属性，36pt Courier 是值。选择器很重要，因为它意味着字体设置将应用于 Web 页面中的所有 h1 元素。但是，你可能想区分一些 h1 元素，那么该怎么办呢？答案在于样式类。

假设你想在文档中使用两种不同类型的<h1>标题，就可以把以下 CSS 代码放在一个样式表中，为每种标题创建一个样式类：

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

注意：这些选择器在 h1 后面包括一个句点（.），其后接着一个描述性的类名称。要在两个样式类之间做出选择，可以使用 class 属性，如下：

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

或者可以使用以下代码：

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

当在 HTML 代码中引用样式类时，可以简单地在元素的 class 属性中指定类名称。在前一个示例中，“Marvin's Munchies Inc.”将以 36 磅的 Comic Sans 字体显示，假定你在 Web 页面顶部包括了一个指向样式表的<link />，并且用户安装了 Comic Sans 字体。“MMI Investor Information”则将以 36 磅的 Arial 字体显示。在程序清单 3.2 中可以看到另一个使用类的示例：寻找 subheader<p>类。

如果想创建可以应用于任何元素而不仅仅是标题或者其他一些特殊标签的样式类，则该怎么做？在 CSS 中，简单地使用一个句点（.），其后接着你构建的任何样式类名称以及你所选择的任何样式规范。这个类可以同时指定任何数量的字体、间距和边距设置。无论何时想要在页面中应用自定义的标签，只需使用一个 HTML 标签以及 class 属性，其后接着你创建的类名称。

例如，程序清单 3.1 中的样式表包括以下样式类规范：

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

该样式类在程序清单 3.2 中与下述标签一起使用。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
<P class="subheader">
```

**Did you Know?**

**提示：**当样式规则包括多个属性时，你可能会注意到编码风格中的变化。对于具有单独一种样式的样式规则，通常会看到属性放在与规则相同的行上，如下所示：

```
p.subheader{ font-weight: bold; }
```

不过，当样式规则包含多个样式属性时，如果每行列出一个属性，那么阅读和理解代码将要容易得多，如下所示：

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

在程序清单 3.2 中，该标签与 封闭标签之间的所有内容都将以加粗的紫色文本显示。

使样式类如此有价值的是：它们把样式代码与 Web 页面分隔开，实际上允许你使自己的 HTML 代码重点关注页面中的实际内容，而不是关注它将如何出现在屏幕上。这样，你就可以通过微调样式表，重点关注如何将内容呈现到屏幕上。你可能感到惊奇的是，样式表中相对较少的代码如何跨整个 Web 站点产生重大的影响，这使页面的维护和操纵要容易得多。

## 3.5 使用样式 ID

在创建自定义的样式类时，可以根据自己的意愿使用那些类许多次——它们并不是独一无二的。不过，在一些情况下，你将希望出于布局或格式化目的（或者二者兼顾）精确控制独特的元素。在这样的情况下，可以考虑使用 ID 代替类。

样式 ID（style ID）是一组自定义的格式化规范，只能应用于 Web 页面上的一个元素。可以跨一组页面使用 ID，但是在每个页面内每个时间只能使用一次。

例如，假设在所有页面的主体内具有一个页面标题。每个页面只有一个页面标题，但是所有页面本身都包括该页面标题的一个实例。下面给出了一个选择器的示例，它具有一个指定的 ID，以及一个属性和一个值：

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

注意：这个选择器在“p”后面包括一个散列标记或磅符号（#），其后接着一个描述性 ID 名称。在 HTML 代码中引用样式 ID 时，只需在元素的 id 属性中指定 ID 名称即可，如下所示：

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

开始和封闭 标签之间的所有内容都将以 24 磅的 Verdana 文本显示，但是在任何给定的页面上只会出现一次。你通常会看到出于布局的目的使用样式 ID 定义页面的特定部分，比如标头（header）区域、脚注（footer）区域、主体区域等。页面中的这类区域只会在每个页面上出现一次，因此使用 ID 是比使用类更合适的选择。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 3.6 内部样式表和内联样式

在一些情况下，你可能希望指定将只在一个 Web 页面中使用的样式。可以把样式表封闭在<style>和</style>标签之间，并直接在 HTML 文档中包括它。以这种方式使用的样式表必须出现在 HTML 文档的<head>中。无需<link />标签，并且不能从任何其他页面引用该样式表（除非把它也复制到那个文档的开始处）。这类样式表被称为内部样式表，在本章前面你已经学过它。

程序清单 3.3 显示了一个说明如何指定内部样式表的示例。

**程序清单 3.3 具有内部样式表的 Web 页面**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Some Page</title>

    <style type="text/css">
      footer {
        font-size: 9pt;
        line-height: 12pt;
        text-align: center;
      }
    </style>
  </head>
  <body>
  ...
  <footer>
  Copyright 2013 Acme Products, Inc.
  </footer>
  </body>
</html>
```

在程序清单的代码中，footer 样式类是在内部样式表中指定的，它出现在页面头部。样式类现在可以在这个页面的主体内使用。事实上，在页面的主体中使用它来设置版权通知的样式。

如果你想创建一个将在单个页面内多次使用的样式规则，那么内部样式表就很方便。不过，在一些情况下，可能需要对一个特殊的元素应用独特的样式，这就需要内联样式规则，它允许只为页面的一小部分（比如单个元素）指定样式。例如，可以通过 style 属性在<p>、 或标签内创建并应用样式规则。这类样式被称为内联样式（inline style），因为它是在 HTML 代码的中间指定的。

**By the Way**

**注意：**和是伪标签，除了指定应用所添加的任何 style 属性的内容范围之外，它们自身不做其他任何事情。 与之间的唯一区别是： 是一个块元素，因此将强制进行换行，而则不然。因此，如果想修改出现在句子或段落中间的任意文本部分的样式，并且不进行任何换行，那么就应该使用。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

下面显示了一个示例 style 属性：

```
<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>
```

这段代码利用标签显示如何在内联样式规则中应用 color 样式属性。事实上，这个示例中的<p>标签和标签都把 color 属性用作一种内联样式。color:red 样式属性将覆盖与标签之间的文本的 color:green 样式属性，理解这一点很重要。然后，在第二个段落中，将不会应用任何一种 color 样式，因为它是一个全新的段落，将遵循整个页面的默认颜色。

**Watch Out!**

**警告：**如果超越页面级调式的范围或者以一种受控的设置试验新事物，那么使用内联样式不被认为是一种最佳实践。最佳实践是使页面链接到在中心维护的样式表，使得所做的更改会立即反映到使用它的所有页面中。

## 验证样式表

就像验证 HTML 或 XHTML 标记很重要一样，验证样式表也很重要。在 http://jigsaw.w3.org/css-validator/上可以找到用于 CSS 的特定验证工具。与第 2 章中讨论的验证工具一样，可以把该工具指向一个 Web 地址，上传文件，或者把内容粘贴进所提供的表单字段中。最终的目标是如图 3.3 所示的结果：有效！

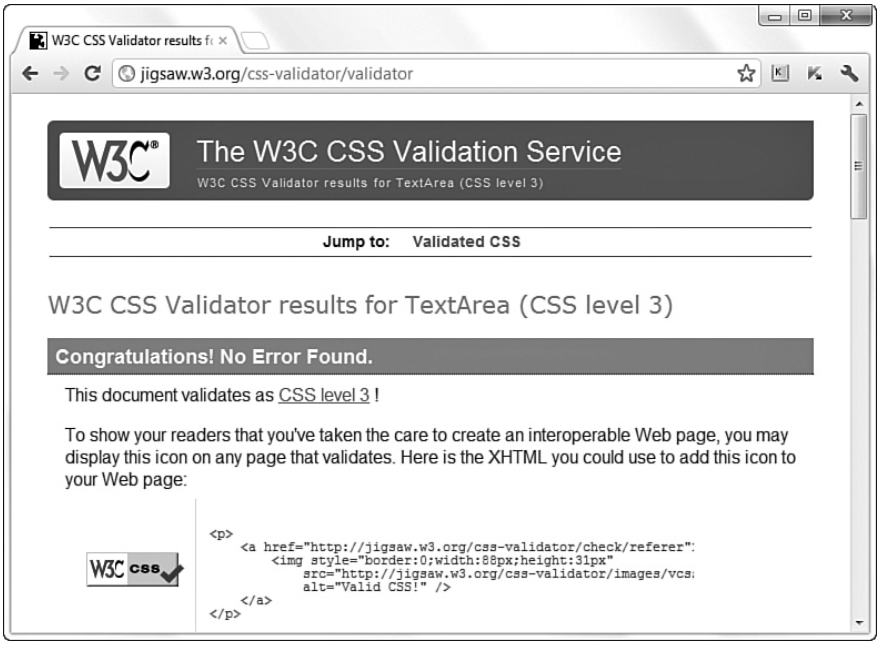



图 3.3

W3C CSS Validator 显示程序清单 3.1 的样式表内容中没有错误

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 3.7 小结

本章学习了样式表可以同时控制许多 HTML 页面的外观，它还可以让你极其精确地控制 HTML 元素的排版式样、间距和定位。我们还学习了通过给几乎所有的 HTML 标签添加一个 style 属性，就可以控制 HTML 页面的任何部分的样式，而不必引用单独的样式表文档。

我们学习了在 Web 站点中包括样式表的 3 种主要方法：在文档中的<head>中链接扩展名为.css 的单独的样式表文件，将样式规则的集合放置在文档头部的<style>标签内，以及通过 style 属性直接把规则放在 HTML 标签中（尽管后一种方法不是长期使用的最佳实践）。

表 3.1 总结了本章中讨论的标签。参阅 http://www.w3c.org 上的 CSS 样式表标准，了解关于在<style>标签或 style 属性后面可以包括什么选项的详细信息。

**表 3.1　　第 3 章中介绍的 HTML 标签和属性**

| 标签/属性 | 作用 |
|---|---|
| <style>...</style> | 允许在文档内包括内部样式表，在<head>与</head>之间使用 |
| **属性** | |
| type="contenttype" | Internet 内容类型（对于 CSS 样式表，总是使用"text/css"） |
| <link /> | 链接到外部样式表（或其他文档类型）。在文档的<head>区域中使用 |
| **属性** | |
| href="url" | 样式表的地址 |
| type="contenttype" | Internet 内容类型（对于 CSS 样式表，总是使用"text/css"） |
| rel="stylesheet" | 链接类型（对于样式表，总是使用"stylesheet"） |
| ... | 不做任何事情，但是会提供一个位置，用于放置 style 或其他属性（类似于 ... ，但是不会导致换行） |
| **属性** | |
| style="style" | 包括内联样式规范（可以用在、 、<body>及其他大多数 HTML 标签中） |

## 3.8 问与答

**问：假如我把一个样式表链接到我的页面，它指示所有的文本都应该是蓝色的，但是在页面中的某个位置有一个标签，那么这段文本将显示为蓝色还是红色？**

**答：**红色。本地内联样式总是具有比外部样式表更高的优先级。位于页面顶部的<style>与</style>标签之间的任何样式规范也具有比外部样式表更高的优先级（但是其优先级不高于同一个页面中后面某个位置的内联样式）。这是我在本章前面所提到的样式表的层叠作用。可以把层叠样式表的作用视作开始于外部样式表，它会被内部样式表覆盖，后者又会被内联样式覆盖。

**问：可以把多个样式表链接到单个页面吗？**

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**答**：当然可以。例如，你可能具有一个用于格式化（如文本、字体、颜色等）的样式表以及另一个用于布局（如边距、填充、对齐等）的样式表，对这两个样式表都只包括一个<link />。从技术上讲，CSS 标准要求 Web 浏览器给用户提供一个选项，以便当通过多个<link />标签提供多个样式表时可以在它们之间做出选择。不过，在实际中，所有主流的 Web 浏览器都只简单地包括每个样式表，除非它具有 rel="alternate"属性。用于链接多个样式表的首选技术涉及使用特殊的@import 命令。下面显示了一个利用@import 导入多个样式表的示例：

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

与<link />标签类似，@import 命令必须放在 Web 页面的头部。在第 20 章中将学习关于这个方便的小命令的更多知识，其中你将学习如何创建专用于打印 Web 页面的样式表。

## 3.9 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 3.9.1 问题

1. 你将使用什么代码创建一个样式表，指定以 30 磅的蓝色 Arial 字体显示标题，并以 10 磅的蓝色 Times Roman 字体（或者默认的浏览器字体）显示所有其他的文本？
2. 如果把为“问题 1”创建的样式表另存为 corporate.css，那么将怎样把它应用于名为 intro.html 的 Web 页面？
3. 可以用多少种不同的方式确保将样式规则应用于内容？

### 3.9.2 答案

1. 样式表将包括以下内容：

```
h1 { font: 30pt blue Arial; }
body { font: 10pt blue "Times New Roman"; }
```

2. 把以下标签放在 intro.html 文档的<head>与</head>标签之间：

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

3. 有 3 种方式：外部、内部和内联。

### 3.9.3 练习

- 使用在本章前面创建的样式表，向样式表中添加一些样式类。为了查看你的劳动果实，可以把这些类应用于你也在本章中创建的 HTML 页面。结合使用类与<h1>和

异步社区会员 Kensuke(18319066421) 专享 尊重版权

<p>标签，获得对事物的感觉。

- 为你的 Web 站点开发一个标准的样式表，并把它链接到你的所有页面中（为需要偏离它的页面使用内部样式和/或内联样式）。如果你是在为一家公司工作，机会就很好，它已经为打印材料开发了字体和样式规范。获得这些规范的一份副本，并使公司的 Web 页面也遵循它们。
- 确保在 http://www.w3.org/Style/CSS/上探索官方样式表规范，并且试验一些本章中未做介绍的更深奥的样式属性。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 4 章

# 密切观察 HTML5 页面结构

在本章中你将学到：

- 如何使用语义标签指示标头和脚注内容；
- 如何使用语义标签指示导航和辅助内容；
- 如何使用语义标签更好地构造主体内容？

在第 2 章中讲到，HTML5 包括一些标签，它们可让你增强在标记过的文本中所提供的信息的语义（含义）。在本章中，将密切观察 6 个元素，它们构成了文档的稳固语义结构的基础，这 6 个元素是：<header>、<section>、<article>、<nav>、<aside>和<footer>。

在本书余下的全部内容中，你将看到在代码示例中适当地使用了这些标签，因此在继续学习后面的内容之前，本章将确保你很好地领会了它们的含义。

## 4.1 概念化页面

理解 HTML5 语义元素的最佳方式之一是查看它们的使用，但是当这些元素的主要目的是提供**含义**（meaning）而不是设计时，这可能会有点困难。这并不是说不能给这些元素添加设计——当然可以，并且在后面的章节中将会这样做。但是，语义元素的“动作”是保持内容，并通过这样做来提供含义，如图 4.1 所示，它显示了用于基本 Web 页面的公共语义元素。

起初，你可能认为：“当然，这是完全有意义的，并且标头出现在顶部，而脚注则出现在底部”，并且对于理解语义元素初看上去自我感觉相当好——也应该如此！然后，再看一眼应该会提出一些问题：如果想希望导航系统出现在标头下方的水平位置，则该怎么做？边注（从字面上讲）必须位于页面一侧吗？如果不想要任何边注，则该如何？在主体区域内再次使用<header>和<footer>，则会发生什么事情？等等，不一而足！你可能会问的另一个问题是：<article>元素适合放在什么位置？在这个示例中没有显示它，但它是本章的一部分。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

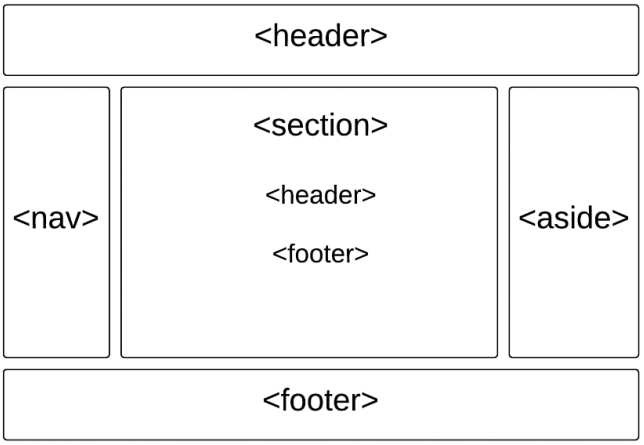



图 4.1

显示 Web 页面中的基本语义元素

现在，概念化页面（确切地讲是你想创建的页面）该闪亮登场了。如果你理解想要标记的内容，并且理解你可以使用任何或所有的语义元素或者不使用它们，仍然可以创建有效的 HTML 文档，那么就可以开始以一种对页面和你自己（并且有望对你的读者）最有意义的方式组织页面的内容。

**注意：**尽管不需要使用语义元素来创建有效的 HTML 文档，仍然建议使用一个最小的集合，以便 Web 浏览器和屏幕阅读器可以确定文档的结构。屏幕阅读器能够给用户报告更深刻的含义，从而提高你的内容的可访问性（如果在 HTML 文档中给这条“注意”加标记，它将使用<aside>元素）。

**By the Way**

在继续学习第二个示例之前，让我们看看图 4.1 中使用的元素，然后查看对各个元素本身的更深入的探索。在图 4.1 中，将在页面顶部看到一个<header>，并在底部看到一个<footer>——如前所述，非常直观。在页面左边使用的<nav>元素与用于导航的公共显示区域相匹配，而页面右边的<aside>元素则与用于辅助注释、引文、帮助文本以及用于获取关于内容的“更多信息”链接的公共显示区域匹配。在图 4.2 中，你将看到其中一些元素改变了位置，不要担心，图 4.1 并不是语义标记的某种不可变的示例。

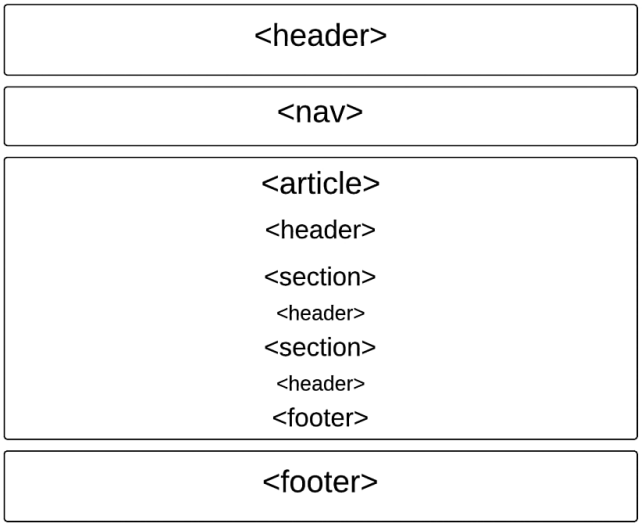



图 4.2

使用嵌套的语义元素给内容添加更多的含义

异步社区会员 Kensuke(18319066421) 专享 尊重版权

你在图 4.1 中看到的可能令人感到奇怪的内容是<section>元素内的<header>和<footer>。如你稍后将学到的，<header>元素的作用是引入第二个示例，然后更深入地探讨各个元素本身。在图 4.1 中，你在页面顶部看到了一个<header>，并在页面底部看到了一个<footer>，如前所述，非常直观。在页面左边使用的<nav>元素与其后的内容匹配，<header>元素本身在文档大纲中不会传达任何级别。因此，可以根据需要使用许多<header>元素相应地标记内容，页面开头的<header>元素可能包含关于作为一个整体的页面的介绍性信息，<section>元素内的<header>元素可能非常容易，并且相应地包含关于其中内容的介绍性信息。对于这个示例中的<footer>元素的多种外观也是如此。

让我们转向图 4.2，它移动了<nav>元素，还引入了<article>元素的使用。

在图 4.2 中，页面开头的<header>和<nav>元素以及页面底部的<footer>元素应该会给你完美的感觉。而且，尽管我们还没有谈论<article>元素，但是如果把它视作一个具有一些区域（甚至<section>!）的容器元素，并且其中每个区域都有它自己的标题，那么图形中间的语义元素块也应该是有意义的。可以看到，没有单独一种方式用于概念化页面内容，对页面上的每种单独的内容都应该进行概念化。

如果标记了图 4.2 中所示的结构中的一些内容，它可能看上去就像程序清单 4.1。

**程序清单 4.1 基本内容的语义标记**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Semantic Example</title>
  </head>
  <body>
    <header>
       <h1>SITE OR PAGE LOGO GOES HERE</h1>
    </header>
    <nav>
       SITE OR PAGE NAV GOES HERE.
    </nav>
    <article>
       <header>
          <h2>Article Heading</h2>
       </header>
       <section>
          <header>
             <h3>Section 1 Heading</h3>
          </header>
          <p>Section 2 content here.</p>
       </section>
       <section>
          <header>
             <h3>Section 2 Heading</h3>
          </header>
          <p>Section 2 content here.</p>
       </section>
       <footer>
          <p>Article footer goes here.</p>
       </footer>
    </article>
    <footer>
       SITE OR PAGE FOOTER HERE
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

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

如果在 Web 浏览器中打开这个 HTML 文档，将会看到如图 4.3 所示的内容——一个完全未编排样式的文档，但它具有语义含义（即使没有人可以“看到”它）。

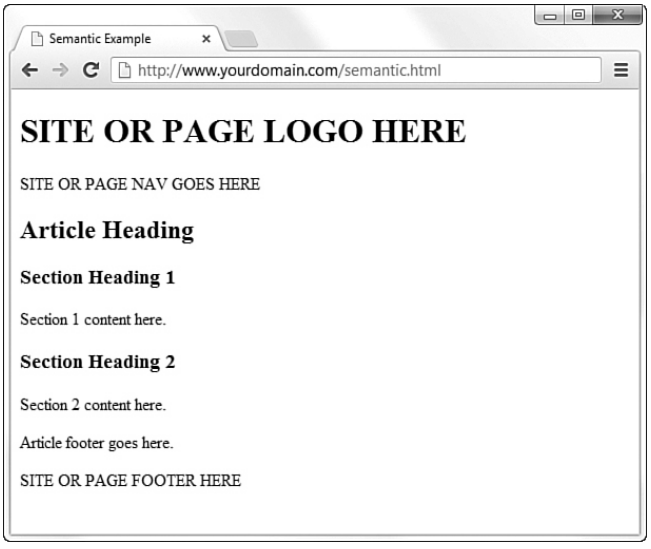



图 4.3

程序清单 4.1 的输出结果

仅仅由于没有可见的样式，并不意味着含义丢失了。如本节前面所指出的，机器可以解释通过语义元素提供的文档结构。在图 4.4 中，可以看到这个基本文档的大纲，它显示了这个文件在经过 http://gsnedders.html5.org/outliner/上的 HTML5 Outline 工具检查后的输出结果。

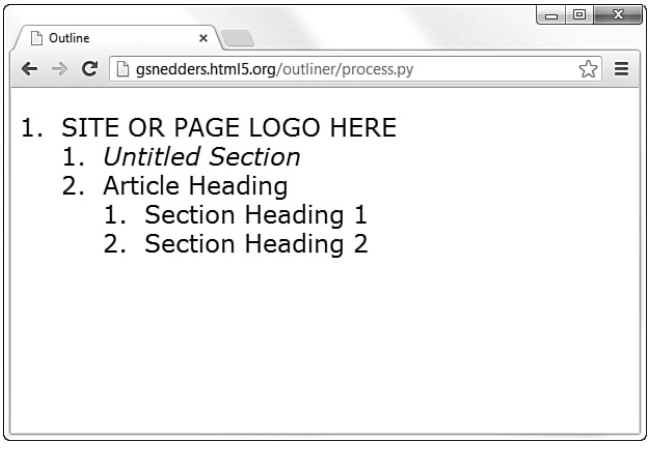



图 4.4

这个文档的大纲遵循语义标记

既然你已经见过了概念化文档中所表示信息的一些示例，就为开始标记那些文档做了更好的准备。下面几节将分别探讨各个语义元素。

**提示：**使用 HTML5 Outline 工具是一种检查你创建了标头、脚注和区域的良好方式。如果检查文档并在任意位置看到“无标题区域”，并且那些无标题的区域与<nav>或<aside>元素不匹配（它们对于包含标头具有更宽松的指导原则），那么你就要做一些额外的工作。

**Did you Know?**

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 4.2 以多种方式使用<header>元素

在最基本的层面上，<header>元素包含介绍性信息。该信息可能采用实际的<h1>（或其他级别）元素的形式，或者它可能只是<p>或 元素内包含的标志图像或文本。内容的含义本质上应该是介绍性的，以保证它包括在<header></header>标签对内。

在本章中迄今为止的示例中可以看到，<header>元素的公共位置位于页面的开始处。当以这种方式使用它时，包含标志或<h1>级页面标题就是有意义的，如下所示：

```
<header>
    <img src="acmewidgets.jpg" alt="ACME Widgets LLC"/>
</header>
```

或者甚至如下所示：

```
<header>
    <img src="acmewidgets.jpg" alt="ACME Widgets LLC"/>
    <h1>The finest widgets are made here!</h1>
</header>
```

两个代码段都有效地使用了<header>，因为它们内部包含的信息是对页面总体上的介绍。

在本章中还看到，不限于只使用一个<header>。可以疯狂地使用<header>元素，只要它们充当介绍性信息的容器即可。程序清单 4.1 显示了为一个<article>内的多个<section>元素使用<header>元素，并且这是该元素的完全有效的使用：

```
<section>
   <header>
     <h3>Section 1 Heading</h3>
   </header>
   <p>Section 2 content here.</p>
</section>
<section>
   <header>
     <h3>Section 2 Heading</h3>
   </header>
   <p>Section 2 content here.</p>
</section>
```

<header>元素可以包含流式内容（flow content）类别中的其他任何元素，并且它也是其中一个成员。这意味着如果想要的话，<header>可以包含<section>元素，并且这是完全有效的标记。不过，在概念化内容时，首先要考虑嵌套类型是否有意义。

**By the Way**

**注意**：一般来讲，流式内容（flow content）元素是包含文本、图像或其他多媒体嵌入式内容的元素，HTML 元素属于多个类别。

如果想要了解关于把元素分类进内容模型中的更多信息，可以参见www.w3.org/TR/2011/WD-html5-20110525/content-models.html。

<header>内所允许内容的唯一例外是：<header>元素不能包含其他<header>元素，也不能包含<footer>元素。类似地，<header>元素不能被包含在<address>或<footer>元素内。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 4.3 理解<section>元素

<section>元素具有一个简单的定义：它是“文档的一个普通区域”，是“内容的一个主题组”，通常具有一个标题。对我而言，这听起来十分简单，对你可能也是如此。因此，你可能很奇怪地发现，如果在所选的搜索引擎中输入“HTML5 中的区域与文章之间的区别”，将会查找到成千上万个条目谈论它们的区别，因为定义一直会绊倒人们。我们首先讨论<section>元素，然后介绍<article>元素，并且希望避免似乎会使 Web 开发人员新手感到痛苦的任何误解。

在程序清单 4.1 中，你看到了一个在<article>内使用<section>的直观示例（在这里重复）；在这个示例中，可以轻松地设想<section>包含一个“内容的主题组”，它们都具有自己的标题这一事实支持了这一点。

```
<article>
    <header>
      <h2>Article Heading</h2>
    </header>
    <section>
      <header>
        <h3>Section 1 Heading</h3>
      </header>
      <p>Section 2 content here.</p>
    </section>
    <section>
      <header>
        <h3>Section 2 Heading</h3>
      </header>
      <p>Section 2 content here.</p>
    </section>
    <footer>
      <p>Article footer goes here.</p>
    </footer>
</article>
```

但是，下面这个示例完全有效地使用了<section>，并且看不到<article>元素：

```
<section>
    <header>
      <h1>Super Heading</h1>
    </header>
    <p>Super content!</p>
</section>
```

那么，开发人员该怎么做呢？比如说你具有一些普通的内容，你知道自己想把它们划分到一些区域中，它们都具有各自的标题。在这种情况下，可以使用<section>。如果只需要形象地划分无需额外标题的内容块（比如利用分段符），那么使用<section>就不合适了，可代之以使用<p>或 。

由于<section>元素可以包含任何其他的流式内容元素，并且可以被包含在其他任何流式内容元素内（<address>元素除外），很容易明白的是，对于<section>元素的使用，如果具有通用的指导原则但是没有其他的限制，那么有时会误解它。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 4.4 使用<article>元素

就我个人而言，我相信关于使用<section>与<article>的许多误解不得不涉及<article>元素的名称。当我考虑一篇文章时，确切地讲，我考虑的是报纸或杂志中的文章。我不会自然地考虑“任何独立的作品体系”，这就是通常定义<article>元素的方式。HTML5 建议的规范把它定义为“文档、页面、应用程序或站点中的一个完整或自含式的作品，在理论上讲，它是可独立分发或重用的”，比如“论坛上的帖子、杂志或报纸中的文章、博客作品、用户提交的评论、交互式构件或小器具，或者其他任何独立的内容项目”。

换句话说，<article>元素可用于包含 Web 站点的整个页面（无论它是否是出版物中的一篇文章）、出版物中的实际文章、任意位置的博客帖子、论坛中的主题讨论的一部分、关于博客帖子的评论，以及作为显示你所在城市的当前天气的容器。因此，在搜索“HTML5 中的区域与文章之间的区别”时得到成千上万条结果就毫不令人感到奇怪了。

一个良好的经验法则是，在尝试搞清楚何时使用<article>以及何时使用<section>时，只需回答下面的问题：这个内容自身有意义吗？如果是，那么无论内容对你来说看起来像什么（例如，一个静态 Web 页面，而不是 *New York Times* 中的一篇文章），首先都要使用<article>元素。如果你发现自己对它进行了拆分，就要在<section>中执行该操作。并且如果你发现自己认为你的“文章”事实上是一个更大的整体的一部分，那么可以把<article>标签改为<section>标签，并且找到文档的开始位置，然后从此处通过在更高级别更合适地放置<article>标签来包围文档。

## 4.5 实现<nav>元素

<nav>元素似乎是如此简单（<nav>意指导航），并且它最终也是这样，但是它也可能被错误地使用。在本节中，你将学习一些基本的使用方法，还要了解一些错误的使用方法以便避免它们。如果你的站点在站点级或者在较长的内容页面中具有任何导航元素，就会有效地使用<nav>元素。

对于那种站点级导航，通常会在主<header>元素内找到<nav>元素。你不需要这样做，但是如果你希望导航内容是介绍性的（并且在模板中无所不在），就可以轻松地为主<nav>元素提出一个出现在主<header>元素内的充分理由。更重要的是，这是有效的 HTML（就像<header>之外的<nav>一样），<nav>元素可以出现在任何流式内容中，以及包含任何流式内容。

下面的代码段显示了一个 Web 站点的主要的导航链接，它放置在一个<header>元素中：

```
<header>
   <img src="acmewidgets.jpg" alt="ACME Widgets LLC"/>
   <h1>The finest widgets are made here!</h1>
   <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>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    </nav>
</header>
```

在文档中不限制只使用一个<nav>元素，对于站点开发人员来说，在创建模板时同时包括主导航系统和辅助导航系统是一种良好的做法。例如，你可能在页面顶部看到水平的主导航系统（通常包含在一个<header>元素内），然后在页面的左侧栏中看到垂直导航系统，它代表主要区域内的辅助页面。在这种情况下，只需简单地使用另一个<nav>元素，它不用包含在<header>元素内，但是以不同的方式放置和编排样式，以便除了语义之外还从视觉上把两类导航系统区分开。

记住，<nav>元素用于主要的导航内容。主导航系统和辅助导航系统都包括在内，还会包括页面内的内容的表格。对于<nav>元素的良好、有用的语义用法，不要简单地将其应用于每个链接，以允许用户到处导航。注意：我说的是“良好、有用的”语义用法，而不一定是“有效”使用，可以把<nav>应用于任何链接列表，依据 HTML 规范它将是有效的，因为链接是流式内容。但它将不会特别有用，它不会添加含义，利用<nav>元素包围指向社交媒体共享工具的链接列表。

## 4.6 何时使用<aside>元素

从这一整本书中我提供的为数众多的提示和注意中可以看出，我非常喜欢最适合于在<aside>元素内标记的内容类型。<aside>元素意指包含与它周围的内容只有稍微有些关联的任何内容——额外的解释、指向相关资源的链接、醒目引文、帮助文本等。你可能把<aside>元素视作侧栏，但是要小心的是，不要只把它视作可视的侧栏，或者位于页面侧边的栏，其中可以插入想要的任何内容，而无论它是否与手头的内容或站点相关。

在图 4.5 中，可以看到<aside>中的内容如何用于创建醒目引文（pull quote），或者专门放在一边以吸引注意的内容摘要。在这里，<aside>用于突出显示文本中的一个重要区域，但它也可以用于定义术语或者指向相关文档的链接。

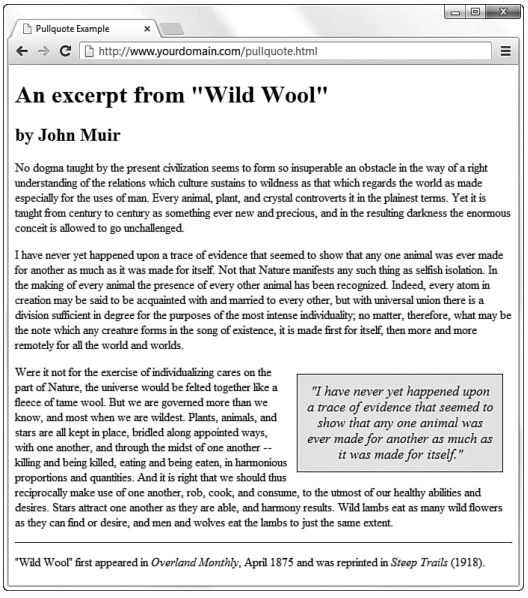



图 4.5

使用<aside>创建有意义的醒目引文

异步社区会员 Kensuke(18319066421) 专享 尊重版权

在确定是否使用<aside>元素时，可以考虑想要添加的内容。它与将包含<aside>的内容直接相关吗，比如文章中使用的术语的定义或者用于文章的相关链接的列表？如果你的回答是轻松的“是”，那就太好了！把<aside>用于核心内容。如果考虑在本身填满内容的包含元素之外包括一个<aside>元素，只需确保<aside>的内容与整体总体上合理相关并且没有把<aside>元素用于视觉效果即可。

## 4.7 有效地使用<footer>元素

与<header>元素相对应的元素是<footer>元素，它包含关于其包含元素的额外信息。<footer>元素的最常见的应用是在页面底部包含版权信息，如下所示：

```
<footer>
   <p>&copy; 2013 Acme Widgets, LLC. All Rights Reserved.</p>
</footer>
```

与<header>元素类似，<footer>元素可以包含流式内容类别中的其他任何元素（其他<footer>或<header>元素除外），并且它也是其中一员。此外，<footer>元素不能包含在<address>元素内，但是<footer>元素可以包含<address>元素。事实上，<footer>元素是存放<address>元素的常见位置。

在<footer>元素内放置有用的<address>内容是<footer>元素的最有效的应用之一（更不要说<address>元素了），因为它提供了关于它所引用的页面或页面区域的特定上下文信息。下面的代码段显示了<address>在<footer>内的使用：

```
<footer>
   <p>&copy; 2013 Acme Widgets, LLC. All Rights Reserved.</p>
   <p>Copyright Issues?Contact:</p>
      <address>
      Our Lawyer<br/>
      123 Main Street<br/>
      Somewhere, CA 95128<br/>
      <a href="mailto:lawyer@richperson.com">lawyer@richperson.com</a>
      </address>
</footer>
```

与<header>元素一样，不限于只使用一个<footer>元素。可以根据需要使用许多<footer>元素，只要它们是关于包含元素的额外信息的容器即可。程序清单 4.1 显示了将<footer>元素用于页面和一个<article>元素，它们二者都是有效的。

## 4.8 小结

本章解释了 6 种语义元素，可以帮助读者更好地指示标头和脚注内容、导航和辅助内容，以及文档中的普通正文内容。本章还讲解了在开始标记总体的文档结构之前如何概念化它，以便在进行标记时，可以使用合适的元素帮助给读者传达额外的含义。

表 4.1 总结了本章介绍的 HTML 标签。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

表 4.1 本章介绍的 HTML 标签

| 标签 | 作用 |
| --- | --- |
| <header>...</header> | 包含介绍性信息 |
| <section>...</section> | 包含主题相似的内容，比如图书中的某一章或者页面中的某个区域 |
| <article>...</article> | 包含属于独立作品体系的内容，比如新闻文章 |
| <nav>...</nav> | 包含导航元素 |
| <aside>...</aside> | 包含用于其包含元素的辅助信息 |
| <footer>...</footer> | 包含用于其包含元素的补充材料（通常是版权通知或作者信息） |
| <address>...</address> | 包含与其最近的<article>或<body>元素相关的地址信息，通常包含在<footer>元素内 |

## 4.9 问与答

**问：我根本不需要使用语义标记吗？你不是在本章中一直都在说有没有它页面都是有效的吗？**

**答：**是的，所有这些元素都不是有效的 HTML 文档所必需的。你不必使用其中任何标记，但是我强烈建议你不要只考虑把这些标记用于视觉显示，也要考虑用于语义含义。视觉显示对于屏幕阅读器毫无意义，但是语义元素可以通过这些机器传达大量的信息。

**问：我对于何时使用<section>以及何时使用<aside>仍然懵懂无知，你能更清楚地加以说明吗？**

**答：**我不会责备你。HTML5 Doctor Web 站点上有可用的资源，它是我见过的最好的资源之一，可以帮助你消除混淆。它是用于 HTML5 分区的流程图，可以在 http://html5doctor.com/downloads/h5d-sectioning-flowchart.png 上找到它。这幅流程图将询问你一些关于内容的恰当问题，并且帮助你确定要使用的正确的容器元素。

## 4.10 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 4.10.1 问题

1．本章中讨论的哪个语义元素适合于包含在文章中使用的词语的定义？

2．在<header>元素内必须使用<h1>、<h2>、<h3>、<h4>、<h5>或<h6>元素吗？

3．在单个页面内可以具有多少个不同的<nav>元素？

### 4.10.2 答案

1．<aside>元素适合于此任务。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

2．否。除了另一个<header>元素或<footer>元素之外，一个<header>元素可以包含任何其他的流式内容。不过，标题元素（<h1>~<h6>）在<header>元素中不是必需的。

3．可以根据需要具有许多<nav>元素。诀窍是只“需要”少数几个<nav>元素（也许只需要用于主导航系统和辅助导航系统的<nav>元素）；否则，含义将会丢失。

### 4.10.3 练习

- 在第 2 章中，有一个练习是让你创建个人 Web 页面。返回到该文档，并添加你在本章中学到的所有相关的语义元素。
- 构建单个页面，创建另外几份相关的内容。记住，你的一些页面可能包含没有分区的<article>元素，但是其他页面也可能包含<section>元素以及<header>和<footer>元素。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 5 章

# 处理文本块和列表

在本章中你将学到：

- 怎样在页面上对齐文本；
- 怎样使用 3 种类型的 HTML 列表；
- 怎样在列表内放置其他列表。

在 Web 的早期时代，只以一种字体和一种大小显示文本。不过，如你在本书前几章中所学过的，HTML 与 CSS 的组合现在使得有可能控制文本的外观，以及如何在 Web 页面上对齐和显示它。

本章将介绍文本对齐的基础知识，并且指导你通过接下来的步骤：使用列表。由于列表是如此常见，HTML 提供了一些标签，用于自动缩进文本，以及在每个列出的项前面添加编号、项目符号或其他符号。你将学习如何格式化不同类型的列表，它们是在 Web 站点中显示内容的许多方式的一部分。

Try It Yourself：准备示例文本

如果你具有一些可以缩进、居中显示或者执行其他操作的示例文本，在这一整章中，就可以最大限度地学习如何编排文本样式。

- 任何类型的大纲、演示文稿中的项目编号要点、带编号的步骤、术语表或者数据库中的文本信息的列表都是可以处理的良好材料。
- 任何文本都可以，但是要尽量找到（或输入）一些你想放到 Web 页面上的文本。公司宣传册或者你的简历中的文本可能是一个好的选择。
- 如果你将使用的文本来自字处理或数据库程序，一定要把它以纯文本或 ASCII 格式保存到一个新文件中。然后在学习本书的过程中可以添加合适的 HTML 标签和样式属性来格式化它。
- 在使用本章中介绍的代码格式化正文文本之前，要添加在前面各章中使用的 HTML 基本标签集（<!DOCTYPE>以及<html>、<head>、<title>和<body>标签对）。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 5.1 在页面上对齐文本

在阅读 Web 上的信息时，很容易想当然地认为大多数段落都会自动左对齐。不过，肯定会存在一些情况，其中你可能选择把内容右对齐，甚至对齐到页面的中间。HTML 提供了用于对齐单个 HTML 块级元素的选项，比如包含在 或 内的文本。不过，在我们深入研究对齐块元素的细节之前，让我们简要说明一些属性的工作方式。

### 5.1.1 使用属性

属性（attribute）提供了与 HTML 标签相关的额外信息，它们是在 HTML 标签内使用的特殊代码字，用于精确控制标签所做的事情。甚至在最简单的 Web 内容中，它们也非常重要，因此熟练地使用它们就是很重要的。

属性涉及使用将应用于特定标签的样式、类或 ID。如果在样式表中定义一个特定的类或 ID，如你在第 3 章中所学到的，那么就可能在标签自身内使用 class="someclass"或 id="someid"调用那个类或 ID。当浏览器呈现内容以进行显示时，它将查看样式表，以精确确定将如何依据关联的样式定义显示内容。类似地，可以使用 style 属性包括用于特定元素的样式信息，而不必把元素连接到实际的样式表，这是你在第 3 章中学过的内联样式格式。

例如，当利用 标签开始一个段落时，可以通过设置 style 属性，指定那个特定段落中的文本是应该对齐到左边距、右边距，还是页面的中间。如果希望把那个特定段落与现有的类或 ID 相关联，可以设置 class 或 id 属性。

在下面的示例中，可以左对齐每个段落：

```
<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 style 属性内，这对于调试或者短期的格式化测试是有用的，但是对于持续维护 Web 内容的用处不是那么大。在第二个段落中，如果用于 leftAlignStyle 类的样式表条目包括 text-align 语句，那么段落将左对齐。记住，使用类意味着其他标签可以重用那个类。类似地，如果用于 firstLeftAlign id 的样式表条目包括 text-align 语句，第三个段落也将左对齐；记住，使用 id 意味着这些样式只能应用于一个确定的标签。

### 5.1.2 对齐块级元素中的文本

若要把块级元素（比如 ）中的文本对齐到右边距，同时又不在样式表中创建单独的类或 ID，可以简单地在段落开头的 标签内放置 style="text-align:right"（或者在类或 id 中定义它）。类似地，要把元素中的文本居中显示，可以使用 。要把段落中的对齐到左边，可以使用 。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

style 属性的 text-align 部分被称为样式规则（style rule），这意味着它将设置 HTML 元素的特定的样式方面。可以使用许多样式规则，细心控制 Web 内容的格式化效果。

text-align 样式规则并不仅仅是为<p>标签预留的。事实上，可以把 text-align 样式规则用于任何块级元素（包括语义元素，比如<section>和<header>），以及<h1>、<h2>、其他标题级别的标签和 标签等。 标签特别方便，因为它可以包容其他的块级元素，从而允许同时控制大部分 Web 内容的对齐方式。 标签中的“div”意指分隔（division）。

**提示：**HTML 中的每个属性和样式规则都具有一个默认值，如果你自己没有设置属性，那么就假定使用默认值。就<p>标签的 text-align 样式规则而言，默认值是 left，因此使用“裸”的<p>标签与使用<p style="text-align:left">的效果相同。学习常用样式规则的默认值是成为良好的 Web 页面开发人员的一个重要部分。

Did you Know?

程序清单 5.1 利用不同的块级元素演示了 style 属性和 text-align 样式规则，结果如图 5.1 所示。

**程序清单 5.1　结合使用 style 属性与 text-align 样式规则**

```
<!DOCTYPE html>

<html lang="en">
  <head>
     <title>Bohemia</title>
  </head>

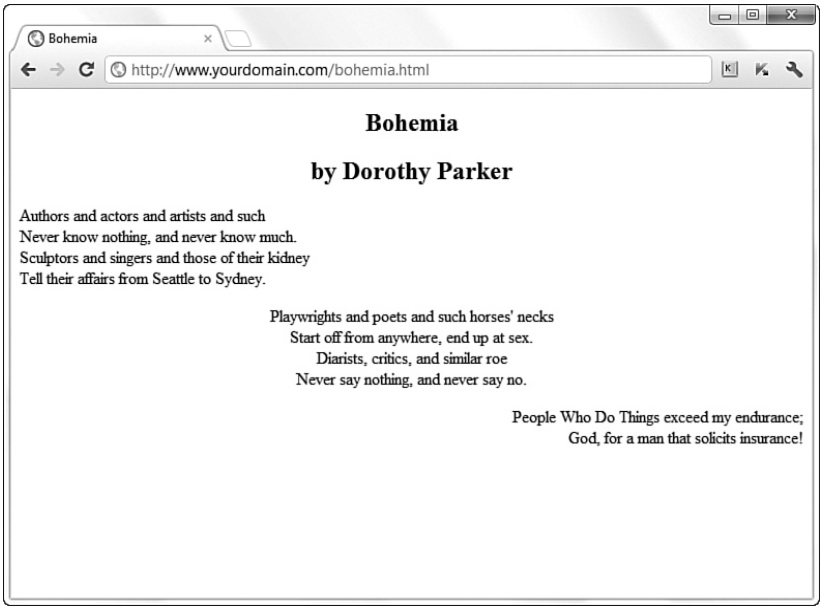
  <body>
    <section style="text-align:center">

    <header>
      <h1>Bohemia</h1>
      <h2>by Dorothy Parker</h2>
    </header>
    </section>
    <section>
      <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>
      </section>
  </body>
</html>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

图 5.1

使用程序清单 5.1 中的文本对齐方式的结果



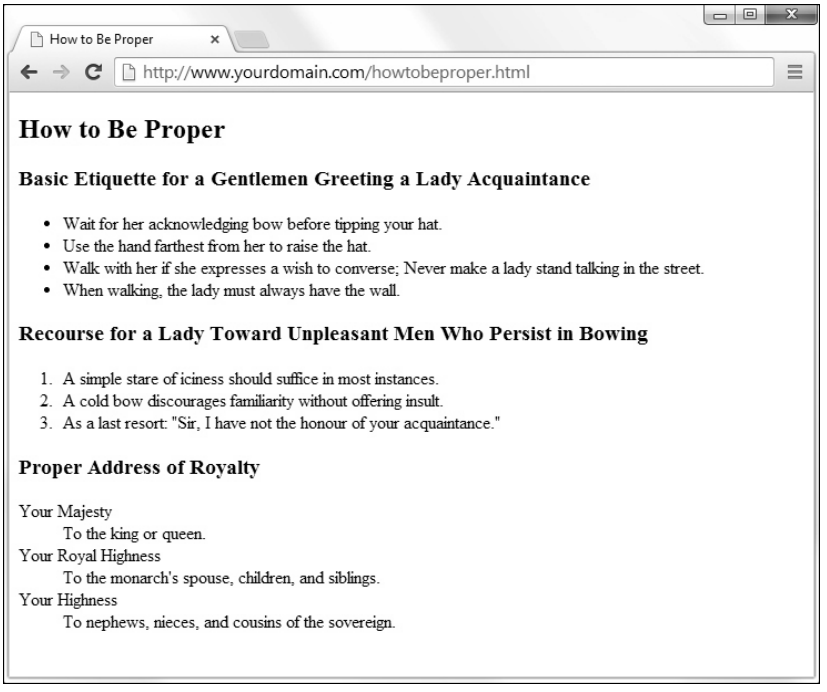

使用<section style="text-align:center">确保内容区域（包括两个标题）居中显示。不过，应用于<section>内的各个段落的内联样式将覆盖设置，并且确保第一个段落的文本左对齐，第二个段落居中显示，第三个段落则右对齐。

## 5.2 3 类 HTML 列表

为了清晰起见，在 Web 页面上把信息展示为项目列表通常是有用的。HTML 提供了 3 类基本的列表，如图 5.2 所示，程序清单 5.3 则揭示了用于构造它们的 HTML 代码。

图 5.2

3 类基本的 HTML 列表




异步社区会员 Kensuke(18319066421) 专享 尊重版权

- 有序列表：一种在每个列表项前具有数字或字母的缩进列表。有序列表开始于<ol>，结束于</ol>封闭标签。列表项封闭在<li></li>标签对内，并且在每个<li>开始标签处会自动出现换行符。整个列表是缩进的。
- 无序列表：一种在每个列表项前具有项目符号或其他符号的缩进列表。无序列表开始于<ul>标签，并且结束于</ul>。与有序列表一样，它的列表项封闭在<li></li>标签对内。换行符和符号出现在每个<li>开始标签处，并且整个列表是缩进的。
- 定义列表：术语及其意义的列表。这类列表在每个列表项前没有特殊的数字、字母或符号，它开始于<dl>，并结束于</dl>。<dt></dt>标签对封闭每个列表项，<dd></dd>标签对则封闭每个定义。换行符和缩进效果是自动出现的。

**程序清单 5.2 无序列表、有序列表和定义列表**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>How to Be Proper</title>
  </head>

  <body>
    <article>

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

    <section>
       <header>
       <h1>Recourse for a Lady Toward Unpleasant Men Who Persist in
       Bowing</h1>
       </header>
       <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>
    </section>

    <section>
       <header>
       <h1>Proper Address of Royalty</h1>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      </header>

      <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>
   </section>
   </article>
  </body>
</html>
```

注意程序清单 5.2 中的语义元素（<article>、<section>和<header>）的使用，它们有助于更好地理解内容大纲，包括文本块之间如何彼此相关联。其中每个元素都可以具有应用于它们自己的样式，它们将进一步在视觉上把元素分隔开。

**By the Way**

**注意：**记住，不同的 Web 浏览器可能以相当不同的方式显示 Web 内容。HTML 标准不会精确指定 Web 浏览器应该如何格式化列表，因此使用较旧的 Web 浏览器的用户所看到的缩进效果可能与你看到的不完全相同。可以使用 CSS 对列表项进行精确控制，在本章后面将学习有关这方面的知识。

## 5.3 在列表内放置列表

尽管定义列表被官方设想用于定义术语，但是许多 Web 页面在想要看到某种缩进效果的任意位置都使用它们。在实际中，如果想缩进任何文本，只需把<dl><dd>放在它的开头，并把</dd></dl>放在末尾，然后忽略<dt></dt>标签对即可。

由于在使用 CSS 时可以精确控制项目的显示，因此无需使用嵌套的（nested）列表来实现缩进的视觉外观。把嵌套的列表预留用于需要它的内容。换句话说，使用嵌套的列表显示信息的层次结构，比如在程序清单 5.3 中。

有序列表和无序列表可以彼此嵌套，直至你想要的许多层级。在程序清单 5.3 中，通过几个无序列表构造了一个复杂的缩进大纲。注意，在图 5.3 中，Web 浏览器自动为前 3 级缩进都使用不同类型的项目符号，使得列表非常容易阅读。这在现代浏览器中很常见。

**Did you Know?**

**提示：**嵌套指一个标签完全出现在另一个标签内。嵌套的标签也称为包含它们的（父）标签的子标签。对嵌套的标签进行缩进是一种常见（但并非必需）的编码实践，以便可以轻松地查看它们与父标签的关系。

**程序清单 5.3 使用列表构建大纲**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Vertebrates</title>
  </head>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
<body>
  <section>

  <header>
  <h1>Vertebrates</h1>
  </header>

  <ul>
    <li><strong>Fish</strong>
      <ul>
        <li>Barramundi</li>
        <li>Kissing Gourami</li>
        <li>Mummichog</li>
      </ul>
    </li>
    <li><strong>Amphibians</strong>
      <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><strong>Reptiles</strong>
      <ul>
        <li>Nile Crocodile</li>
        <li>King Cobra</li>
        <li>Common Snapping Turtle</li>
      </ul>
    </li>
  </ul>
 </body>
</html>
```

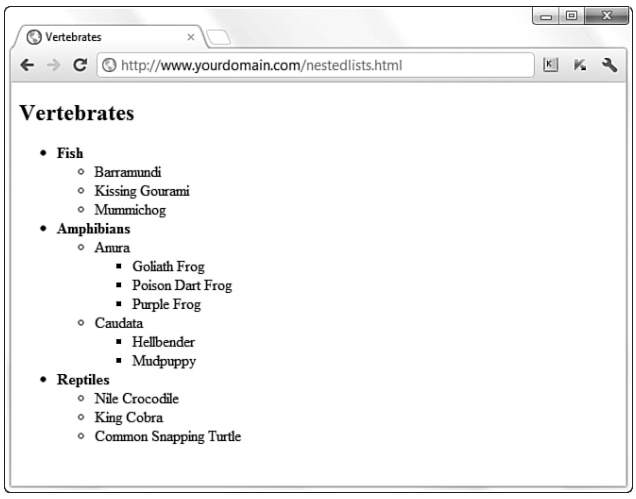



图 5.3

多级无序列表整齐地缩进并添加了项目符号，以改进可读性

异步社区会员 Kensuke(18319066421) 专享 尊重版权

如图 5.3 所示，Web 浏览器通常为第一级项目符号使用一个实心圆盘，为第二级项目符号使用一个空心圆，并为所有更深的层级使用实心正方形。不过，可以为任何层级明确选择使用哪种类型的项目符号，其方法是通过内联方式或者在特定的样式表中使用<ul style="list-style-type:disc">、<ul style="list-style-type:circle">或<ul style="list-style-type:square">代替<ul>。

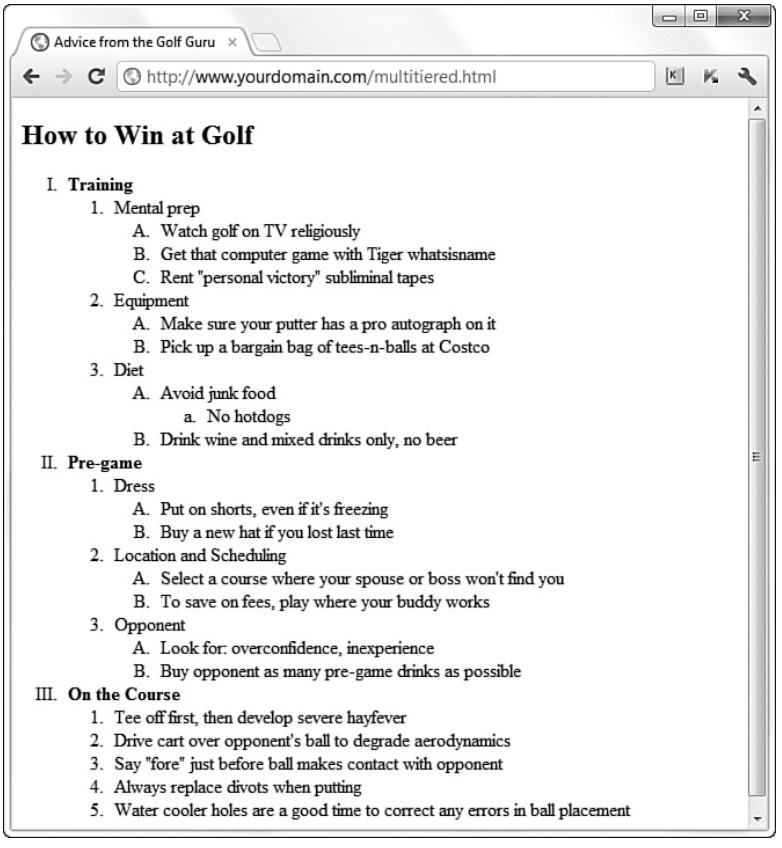
甚至可以通过在<li>标签中使用 list-style-type 样式规则，更改无序列表内的任何一个要点的项目符号。例如，下面的代码将在单词“extra”和“super”前面显示一个空心圆，并在单词“special”前面显示一个实心正方形：

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

list-style-type 样式规则还可以处理有序列表，但它不是选择一种项目符号，而是选择要放置在每个列表项前面的数字或字母类型。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）、大字字母（list-style-type:upper-alpha）、小写字母（list-style-type:lower-alpha）和普通的数字。所得到的结果如图 5.4 所示，它进行了漂亮的格式化。

图 5.4

良好格式化的大纲可以使几乎所有的计划看起来更合理




尽管程序清单 5.4 只把 list-style-type 样式规则用于<ol>标签，但是也可以把它用于列表内特定的<li>标签（虽然难以想象你希望这样做的情况）。还可以利用 list-style-type:decimal

异步社区会员 Kensuke(18319066421) 专享 尊重版权

明确指定普通的编号，并且可以利用 list-style-type:lower-roman 指定小写的罗马数字。

程序清单 5.4 在多级列表中结合使用 list-style-type 样式规则与 style 属性

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Advice from the Golf Guru</title>
  </head>

  <body>
    <article>

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

    <ol style="list-style-type:upper-roman">
      <li><strong>Training</strong>
        <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><strong>Pre-game</strong>
        <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
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
          <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><strong>On the Course</strong>
      <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>
 </article>
 </body>
</html>
```

## 5.4 小结

本章讲解了一些属性可用于指定许多 HTML 标签的选项和特殊行为，以及结合使用 style 属性与 CSS 样式规则来对齐文本。本章还讲解了如何创建和结合 3 种基本类型的 HTML 列表：有序列表、无序列表和定义列表。可以把列表放置在其他列表内，来创建大纲和其他复杂的文本排列效果。

表 5.1 总结了本章中讨论的标签和属性。不要觉得必须记住所有这些标签，有一些非常好的针对 HTML 和 CSS 的参考资料可用，比如 Mozilla Developer Network（开发人员网络）上的指南。参见 https://developer.mozilla.org/en-US/docs/Web/HTML/Element 和 https://developer.mozilla.org/en-US/docs/Web/CSS，它们分别提供了关于 HTML 和 CSS 的指南。

**表 5.1 本章介绍的 HTML 标签和属性**

| 标签/属性 | 作用 |
|---|---|
| … | 要格式化的文本区域 |
| <dl>…</dl> | 一个定义列表 |
| <dt>…</dt> | 定义术语，作为定义列表的一部分 |
| <dd>…</dd> | 定义术语对应的定义，作为定义列表的一部分 |
| <ol>…</ol> | 一个有序（编号）列表 |
| <ul>…</ul> | 一个无序（项目符号）列表 |
| <li>…</li> | 用于<ol>或<ul>的列表项 |
| **属性** | |
| style="text-align:alignment" | 把文本对齐到中间（center）、左边（left）或右边（right）（可以用于<section>、<article>、<p>、<h1>、<h2>、<h3>等） |
| style="list-style-type:numtype" | 用于标记列表的数字类型。一些常见的值是 decimal、lower-roman、upper-roman、lower-alpha、upper-alpha 和 none |
| style="list-style-type:bullettype" | 用于标记列表项的项目符号图标。一些可能的值是 disc、circle、square 和 none |

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 5.5 问与答

**问：我见过一些 Web 页面使用三维小球或其他特殊图形作为项目符号，它们是怎样做到的呢？**

**答：**这种技巧稍微有点超越了本章中介绍的内容，你将在第 11 章中学习如何自己执行该操作。

**问：我怎样“完全两端对齐”文本，使得文本与左、右边距齐平？**

**答：**可以在样式声明中使用 text-align:justify。

## 5.6 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 5.6.1 问题

1. 怎样在整个页面上居中显示所有的文本？
2. 怎样缩进单个单词，并在它前面放置一个正方形项目符号？
3. 你将使用什么来创建定义列表，以显示单词“glunch”的含义是“一种蔑视、生气或不悦的神情”，而单词“glumpy”的含义是“阴沉、郁郁寡欢或闷闷不乐”？

### 5.6.2 答案

1. 如果考虑紧接在页面顶部的<body>标签后面放置一个利用"text-align:center"编排样式的 ，并把 放在页面末尾的</body>标签之前。不过，text-align 样式也直接受到<body>标签支持，这意味着可以抛弃 标签或其他容器元素，并以内联方式或者在样式表中直接对<body>标签应用 text-align:center。转眼之间，整个页面都将居中显示！
2. 可以使用下面这段代码：

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

（把 style="list-style-type:square"放在<li>标签中，或者在样式表中定义它，它们可以提供相同的结果，因为这个列表中只有一个列表项。）

3. 可以使用下面这段代码：

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

异步社区会员 Kensuke(18319066421) 专享 尊重版权

### 5.6.3 练习

- 使用文本对齐样式属性，把文本块放在 Web 页面中的多个不同的位置。尝试嵌套段落和分区（<p>和 ），以便理解样式将如何通过内容层次结构执行层叠或者不层叠。
- 尝试制作一个有序列表，概括你想要放在 Web 页面上的信息。这可以让你实践格式化 HTML 列表，还可以让你提前考虑将在本书后面的章节中介绍的问题。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 6 章

# 处理字体

在本章中你将学到：

- 怎样使用特殊字符；
- 怎样使用粗体字、斜体和特殊文本格式化效果；
- 怎样使用文本效果；
- 怎样调整字体；
- 怎样使用 Web 字体。

在 Web 的早期时代，只以一种字体和一种大小显示文本，但是那些日子已经一去不复返了。HTML 与 CSS 的组合使得有可能控制文本外观的所有方面。在本章中，你将学习如何更改字体的视觉显示（它的字体系列、大小和粗细），以及如何把粗体字、斜体字、上标、下标和加删除线的文本纳入到页面中。你还将学习如何更改字形和字体大小。

Try It Yourself：准备示例文本

如果你具有一些可以显示不同字体和颜色并且可以缩进、居中显示或者执行其他操作的示例文本，那么在这一整章中，就可以最大限度地学习如何编排文本样式。它实际上与你将使用的文本类型无关，因为你具有如此多种不同的样式可能性要尝试，以至于它们无论如何都永远不会同时出现在相同的 Web 页面上（除非你想使访问者陷入疯狂之中）。利用这个机会，感受一下文本级改变可能如何影响内容的外观。

- 如果你将使用的文本来自字处理或数据库程序，一定要把它以纯文本或 ASCII 格式保存到一个新文件中。然后在学习本书的过程中可以添加合适的 HTML 标签和样式属性来格式化它。
- 任何文本都可以，但是要尽量找到（或输入）一些你想放到 Web 页面上的文本。公司宣传册或者你的简历中的文本可能是一个好的选择。
- 任何类型的大纲、演示文稿中的项目编号要点、带编号的步骤、术语表或者数据库

异步社区会员 Kensuke(18319066421) 专享 尊重版权

中的文本信息的列表都是可以处理的良好材料。

- 在使用本章中介绍的代码格式化正文文本之前，要添加在前面各章中使用的 HTML 基本标签集（至少要使用<html>、<head>、<title>和<body>标签）。

## 6.1 处理特殊字符

在我们急匆匆地介绍字体改变之前，先让我们简要讨论一下字体内的特殊字符。大多数字体都包括用于欧洲语言的特殊字符，比如 Café 中的重音“é”。你还将发现少量的数学符号和特殊的标点符号，比如圆形项目符号（•）。

**By the Way**

**注意：**在查看其他设计师的 Web 内容时，你可能注意到其中标记文本的方法不同于本书中讲述的那些方法。一些格式化文本的老方法包括：使用标签对来指示何时应该对单词进行加粗，使用标签对指示何时应该以斜体格式显示单词，以及使用<font></font>标签对指定字体系列、大小和其他属性。不过，这种方法将被 HTML 逐步淘汰，并且 CSS 要强大得多。

在 HTML 文档中，可以使用如表 6.1 所示的合适代码在任何位置插入这些特殊字符。在 http://www.webstandards.org/learn/reference/named_entities.html 上甚至可以找到一个用于多种字符集的更广泛的代码列表。

表 6.1 常用的英语特殊字符

| 字符 | 数字代码 | 代码名称 | 描述 |
|---|---|---|---|
| “” | " | " | 引号 |
| & | & | & | “与”符号 |
| < | < | < | 小于号 |
| > | > | > | 大于号 |
| ¢ | ¢ | ¢ | 分币符号 |
| £ | £ | £ | 英镑 |
| ¦ | ¦ | ¦或&brkbar; | 间断竖线 |
| § | § | § | 分节号 |
| © | © | © | 版权 |
| ® | ® | ® | 注册商标 |
| o | ° | ° | 度的符号 |
| ± | ± | ± | 加或减 |
| 2 | ² | ² | 上标 2 |
| 3 | ³ | ³ | 上标 3 |
| · | · | · | 中间点 |
| 1 | ¹ | ¹ | 上标 1 |

异步社区会员 Kensuke(18319066421) 专享 尊重版权

续表

| 字符 | 数字代码 | 代码名称 | 描述 |
|---|---|---|---|
| ¼ | ¼ | ¼ | 1/4 |
| ½ | ½ | ½ | 1/2 |
| ¾ | ¾ | ¾ | 3/4 |
| Æ | Æ | Æ | 大写 AE 连体字母 |
| æ | æ | æ | 小写 ae 连体字母 |
| É | É | É | 重读大写字母 E |
| é | é | é | 重读小写字母 e |
| × | × | × | 乘号 |
| ÷ | ÷ | ÷ | 除号 |

例如，可以使用以下任何一种方法创建单词“café”：

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

**提示**：你正在寻找版权（©）和注册商标（®）符号吗？它们的代码分别是©和®。要创建未注册的商标（ ）符号，可以使用™。

**Did you Know?**

尽管可以通过数字指定字符实体，每个符号还具有一个助记名称，它们通常更容易记住。

HTML 使用称为字符实体（character entity）的特殊代码来表示诸如©和®之类的特殊字符。字符实体的指定总是开始于“&”，并且结束于“;”。表 6.1 列出了最常用的字符实体，尽管 HTML 支持的字符实体要多得多。

表 6.1 包括用于尖括号、引号以及“与”符号的代码。如果希望这些符号出现在页面上，就必须使用那些代码；否则，Web 浏览器将把它们解释为 HTML 命令。

程序清单 6.1 和图 6.1 显示了表 6.1 中的几个符号的实际应用。

**程序清单 6.1 特殊字符代码**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Punctuation Lines</title>

    <style type="text/css">
    section {
     margin-bottom: 20px;
    }
    </style>

  </head>

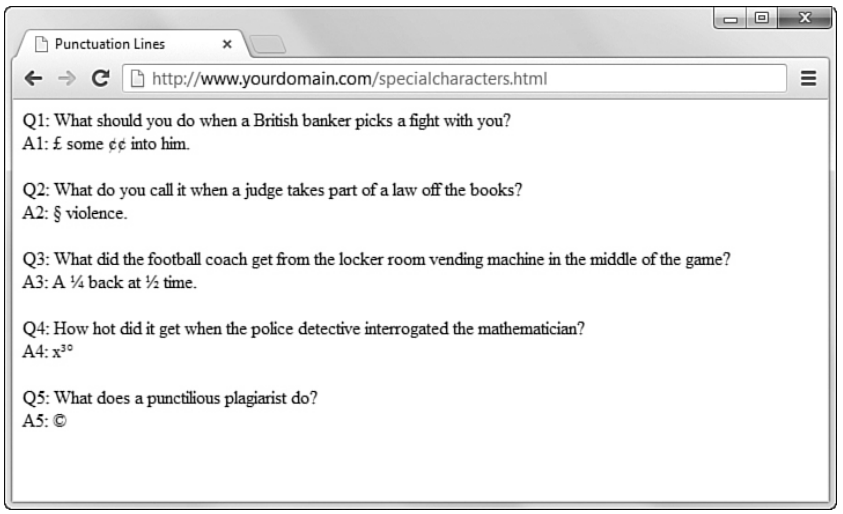
  <body>
    <section>
      Q: What should you do when a British banker picks a fight
      with you?<br />
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      A: &pound; some &cent;&cent; into him.
    </section>
    <section>
      Q: What do you call it when a judge takes part of a law
      off the books?<br />
      A: &sect; violence.
    </section>
    <section>
      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.
    </section>
    <section>
      Q: How hot did it get when the police detective interrogated
      the mathematician?<br />
      A: x&sup3;&deg;
    </section>
    <section>
      Q: What does a punctilious plagiarist do?<br />
      A: &copy;
    </section>
  </body>
</html>
```

图 6.1

这是程序清单 6.1 中的 HTML 页面在大多数 Web 浏览器中所显示的效果




## 6.2 粗体、斜体和特殊的文本格式化

回到打字机的时代，我们满足于纯文本显示，以及使用临时的下划线来突出重点。今天，粗体（boldface）和斜体（italic）文本在所有纸面交流中都变成了社交礼节所要求的。自然，也可以向 Web 内容中添加粗体和斜体文本。多个标签和样式规则使文本格式化成为可能。

给文本添加粗体和斜体格式化效果的老派方法——这里将简要讨论，因为如果你选择查看源代码，那么将不可避免地会在许多 Web 站点的源代码中看到它，包括和标签对。对于粗体文本，将用和标签包围文本。类似地，要使任何文本以斜体显示，将在和标签内封闭它。尽管这种方法在浏览器中仍然会工作得很好，但它

异步社区会员 Kensuke(18319066421) 专享 尊重版权

不像用于文本格式化的 CSS 样式规则那样灵活或强大，应该避免使用。

**By the Way**

**注意：**尽管只要有可能，你都希望使用样式来影响表示，但是在处理粗体和斜体文本时，样式规则的一种替代方法涉及和标签对。在大多数浏览器中，标签所做的事情与<b>标签相同，而标签就像<i>标签一样，用于把文本格式化为斜体。当然，也可以按你喜欢的方式设置这些标签的样式，但是这些是默认设置。和标签被认为是<b>和<i>之上的改进，因为它们只暗示文本应该接受特殊的强调。它们没有明确规定应该如何实现这种效果。换句话说，浏览器不必把解释为意指粗体或者把解释为意指斜体。这使和更适合于 HTML5，因为它们给文本添加了含义，并且会影响文本的显示方式。

本书第 III 部分“利用 CSS 进行高级 Web 页面设计”更深入地介绍了 CSS 样式规则，但是这里做一点铺垫是合适的，以便你可以理解一些基本的文本格式化选项。font-weight 样式规则使你能够使用一个样式规则设置字体的粗细或鲜明度，用于 font-weight 的标准设置包括 normal、bold、bolder 和 lighter（其中 normal 是默认设置）。斜体文本是通过 font-style 规则控制的，可以把它设置为 normal、italic 或 oblique。如果希望应用多个样式规则，也可以一起指定样式规则，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是：必须用分号（;）把它们隔开。

不过，并不仅限于在段落中使用字体样式。下面的代码显示了如何把项目符号列表中的文本设置为斜体字：

```
<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>
```

也可以在标题内使用 font-weight 样式，但是一种更粗重的字体通常不会影响标题，因为它们默认已经是粗体。

尽管使用 CSS 允许对文本应用丰富的格式化效果，但是当不必像 CSS 所允许的那样具体时，另外几个有效的 HTML5 标签也适合于给文本添加特殊的格式化效果。下面给出了其中一些标签，所有其他的文本格式化实际上应该都是利用 CSS 完成的。每个标签的实际应用如程序清单 6.2 和图 6.2 所示。

- <sup></sup>：上标文本。
- <sub></sub>：下标文本。
- ：强调的（斜体）文本。
- ：醒目的（粗体）文本。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

- <pre></pre>：预先格式化的（通常是等宽的）文本，保留空格和换行符。

**Watch Out!**

**警告：**在过去，<u>标签用于创建加下划线的文本，但是你现在将不想使用它，这是由于下面两个原因。第一，用户期望加下划线的文本是一个链接，因此如果加下划线的文本不是链接，那么他们可能会糊涂。第二，<u>标签是过时的，这意味着它将 HTML 语言逐步淘汰（就像<strike>标签一样）。在 Web 浏览器中仍然支持这两个标签，并且很可能持续相当长的一段时间，但是使用 CSS 是创建加下划线和删除线的文本的首选方法。

### 程序清单 6.2 特殊格式化的标签

```
<!DOCTYPE html>

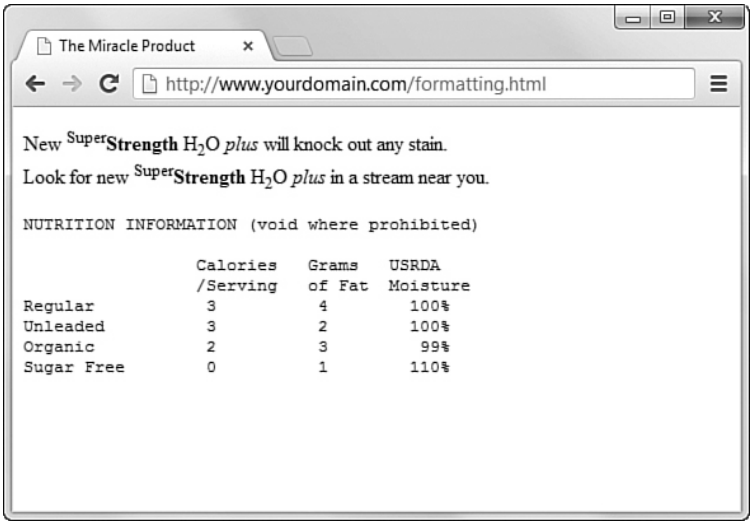
<html lang="en">
  <head>
    <title>The Miracle Product</title>
  </head>

  <body>
    <p>
    New <sup>Super</sup><strong>Strength</strong> H<sub>2</sub>O
    <em>plus</em> will knock out any stain.<br/>Look for new
    <sup>Super</sup><strong>Strength</strong> H<sub>2</sub>O <em>plus</em>
    in a stream near you.
    </p>
    <pre>
NUTRITION INFORMATION (void where prohibited)

                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 6.2

这是程序清单 6.2 中的字符格式化的效果




异步社区会员 Kensuke(18319066421) 专享 尊重版权

<pre>标签导致文本以等宽字体显示，并且会做另外某件独特而有用的事情。如你在第 2 章中所学到的，在 HTML 文件中通常会忽略多个空格和换行符，但是<pre>可以保留额外的间距和换行符。例如，如果没有<pre>，图 6.2 末尾的文本看起来将如下所示：

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

即使在每一行末尾添加<br />标签，列也不会正确地对齐。不过，当在开头放置<pre>并在末尾放置</pre>时，列将正确地对齐，因为这样会保留精确的空格，而无需使用<br />标签。<pre>标签提供了一种快捷、容易的方式，用于以最少的工作量保留你可能想转换成 Web 页面的任何等宽文本文件的对齐方式。

CSS 提供了一些用于对齐文本（实际上可以对文本做任何事情）的更健壮的方法，在本书整个第 III 部分将学习关于它们的更多知识。

## 6.3 调整字体

有时，你可能想对文本的大小和外观进行更多一点控制，而不仅仅是将其设置为粗体或斜体。在我深入讨论使用 CSS 调整字体的合适方式之前，让我们简要探讨在 CSS 出现之前是怎样操作的：当你查看其他 Web 站点的源代码时，仍有可能发现这种方法的一些示例。记住，仅仅由于这些老方法仍在使用并不意味着你应该如法炮制。

在样式表投入使用之前，现在被逐步淘汰的<font>标签用于控制 Web 页面文本中的字体。

**注意：**我不能充分强调<font>标签将不会被使用！这里仅仅出于演示和历史目的而使用它。

**By the Way**

例如，曾经使用下面的 HTML 代码更改页面上的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 size 和 color 属性使得可以不用太费力地改变文本的字体。尽管这种方法工作得很好，但是由于 CSS 样式规则，它已经被一种更高级的字体格式化方法所取代。下面给出了几种用于控制字体的主要的样式规则。

- font-family：设置字体系列（字形）。
- font-size：设置字体的大小。
- color：设置字体的颜色。

font-family 样式规则允许设置用于显示文本的字形。你可以（并且通常应该）为这个样式指定多个值（用逗号隔开），使得如果用户系统上的第一种字体不可用，那么浏览器将可以尝试一种替代字体。

**注意：**在第 7 章中将学习关于控制页面上的文本颜色的更多知识。该章还将说明如何创建你自己的自定义颜色，以及控制文本链接的颜色。

**By the Way**

提供替代字体系列很重要，因为每个用户都可能会安装不同的字体集，至少会超越常

异步社区会员 Kensuke(18319066421) 专享 尊重版权

用的基本核心字体集（如 Arial、Times New Roman 等）。通过提供替代字体列表，当无法找到理想的字体时，就有更好的机会使页面优雅地转而依靠一种已知的字体。下面给出了一个 font-family 样式的示例，它用于设置一个文本段落的字形：

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

这个示例具有几个有趣的部分。第一，将 arial 指定为主字体。大小写不影响字体系列，因此 arial 与 Arial 或 ARIAL 没有什么不同。关于这段代码的另一个有趣的方面是：字体名称“times roman”是用单引号括住的，因为它中间有一个空格。不过，由于'times roman'出现在 sans-serif 的通用规范之后，将不太可能使用它。由于 sans-serif 位于第二个位置，因此它告诉浏览器：“如果这台机器上没有安装 Arial，那么就使用默认的 sans-serif 字体。”

font-size 和 color 样式规则也常用于控制字体的大小和颜色。font-size 样式可以设置为一种预先定义的大小（如 small、medium 或 large）；color 则可以设置为一种预先定义的颜色（如 white、black、blue、red 或 green），或者可以把它设置为一种特定的十六进制颜色（如 #ffb499）。下面是前一个段落示例的更好版本，并且指定了字体大小和颜色：

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

程序清单 6.3 中的示例 Web 内容以及如图 6.3 所示的结果，使用了一些字体样式规则来创建一份基本的在线简历的开始部分。

**By the Way**

**注意**：在第 7 章将学习十六进制颜色。目前，只需理解 color 样式规则允许指定精确的颜色，而不仅仅是使用 green、blue、orange 等。

**程序清单 6.3 使用字体样式规则创建基本的简历**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>R&eacute;sum&eacute; for Jane Doe</title>

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

      header {
        text-align: center;
      }

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

      p.contactinfo {
        font-size:14px;
      }
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

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

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

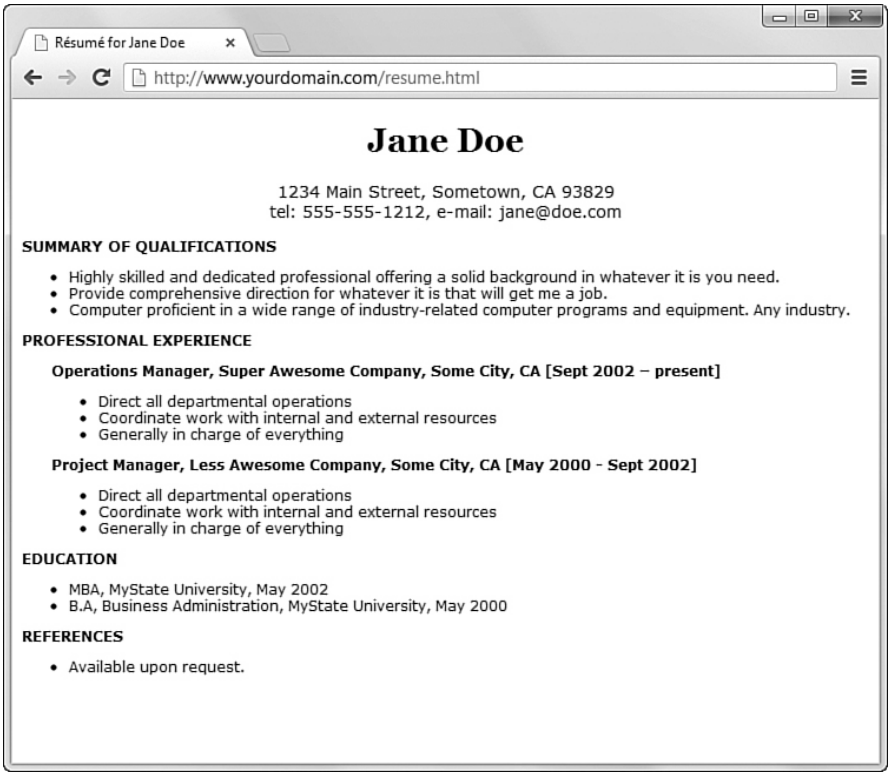
     </section>
     <section>
     <p class="categorylabel">Professional Experience</p>
     <div class="indented">
             <p><strong>Operations Manager,
             Super Awesome Company, Some City, CA [Sept 2002 –
             present]</strong></p>
             <ul>
             <li>Direct all departmental operations</li>
             <li>Coordinate work with internal and external
             resources</li>
             <li>Generally in charge of everything</li>
             </ul>
             <p><strong>Project Manager,
           Less Awesome Company, Some City, CA [May 2000 - Sept
           2002]</strong></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>
     </section>
     <section>
     <p class="categorylabel">Education</p>
     <ul>
     <li>MBA, MyState University, May 2002</li>
     <li>B.A, Business Administration, MyState University,
     May 2000</li>
     </ul>
     </section>
     <section>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

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

图 6.3

这是程序清单 6.3 中使用的代码所显示的效果




使用 CSS，如你在第 3 章中所学到的，它把样式集组织成类，可以看到文本格式化是如何应用于这段内容的不同区域的。如果密切查看 div.indented 类的定义，将会看到 margin-left 样式的使用。这个样式对元素的左边应用一定的空间量（在这个示例中是 25 像素），这个空间将导致图 6.3 中所示的缩进效果。

## 6.4 使用 Web 字体

在上一节中，看到我们十分确信所用的字体系列驻留在每个人的计算机上。也就是说，可以保证大多数计算机都会呈现 Arial 或 Times New Roman，或者具有用于 serif 和 sans-serif 的所必需的默认字体，如果这是样式表所需要的话。但是，利用 CSS3 中包括的 @font-face 特性，甚至可以对在线放置的内容发挥更大的设计能力。

简而言之，@font-face 特性允许在 HTML5 标记中定义要使用的字体，使得无论用户是否在他们的计算机上安装了那些字体（并且很有可能用户没有在他们自己的计算机上安装你所选的奇特字体），都会显示出它们。字体的定义可以是本地的（对于你的 Web 服务器而言，如果你在意在那里包括字体文件的话），或者是远程的（可以链接到许多字体存

异步社区会员 Kensuke(18319066421) 专享 尊重版权

储的位置）。

在样式表中，要定义在整个页面中使用的新字体，可以简单地使用下面的结构：

```
@font-face {
     font-family: 'some_name_goes_here';
     src: url('some_location_of_the_font_file');
}
```

在定义它之后，可以在样式表中的其他任何位置引用 font-family，如下所示：

```
h1 {
  font-family:some_name_goes_here;
  font-size:28px;
  text-align:center;
}
```

但是，你可能会问，从哪里获得字体呢？可以从许多位置获得字体——有一些字体是免费的，其他字体则不然。一个广泛流行的位置是 Google Web Fonts（http://www.google.com/fonts），这不仅是因为这里的字体是免费的，而且因为 Google 被广泛认可为提供了一个稳定的平台，如果你的 Web 排版样式依赖于驻留在别人的 Web 服务器上的字体，这就很重要。其他一些用于获得字体的可靠的付费站点是 TypeKit（www.typekit.com/）和 Fontspring（www.fontspring.com）。付费站点不一定很糟糕，艺术家们也必须挣钱。我在 TypeKit 上进行了个人订购，并且对它们的服务非常满意，但是我也把 Google Web Fonts 用于许多项目。

让我们试着修改程序清单 6.3 中的代码，包括进一种 Google Web Font 字体，用于 h1 元素。如果转到 http://www.google.com/fonts，并选择一种你喜欢的字体，Google 提供了要在 HTML 和 CSS 中包括的代码。我选择了一种名为 Cherry Swash 的字体，Google 建议我在自己的 HTML 模板中的<head>区域中包括以下代码：

```
<link href='http://fonts.googleapis.com/css?family=Cherry+Swash:400,700'
     rel='stylesheet' type='text/css' />
```

**By the Way**

**注意：**如果查看前面的链接位置，可以看到 Google 的@font-face 定义已经为我们完成了。确切地讲，其定义如下：

```
@font-face {
  font-family: 'Cherry Swash';
  font-style: normal;
  font-weight: 400;
  src: local('CherrySwash'), local('CherrySwashRegular'), url(http://themes.
  googleusercontent.com/static/fonts/cherryswash/v1/HqOk7C7J1TZ5i3LejF0vnhCUOGz
  7vYGh680lGh-uXM.woff) format('woff');
}
@font-face {
  font-family: 'Cherry Swash';
  font-style: normal;
  font-weight: 700;
  src: local('Cherry SwashBold'), local('CherrySwashBold'), url(http://themes.
  googleusercontent.com/static/fonts/cherryswash/v1/CfyMyQqfucZPQNB0nvYyHl4twX
  kwp3_u9ZoePkT564.woff)
format('woff');
}
```

既然我的代码知道在哪里寻找字体，我们就只需引用它即可：

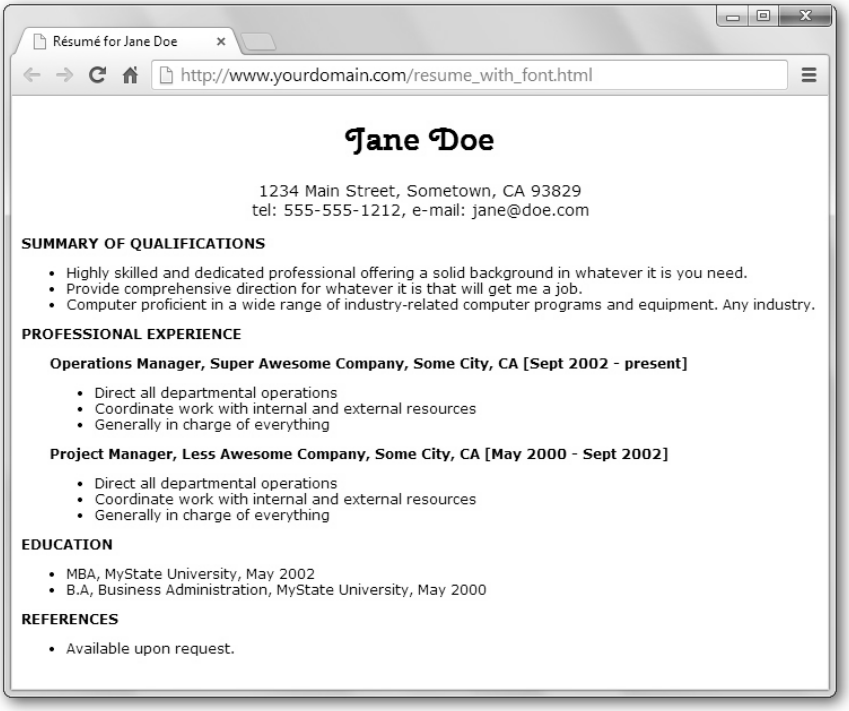
异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
h1 {
  font-family:'Cherry Swash';
  font-size:28px;
  text-align:center;
}
```

图 6.4 显示了使用 Web 字体的新简历。

图 6.4

这份简历在标题中使用 Cherry Swash 作为字体




## 6.5 小结

本章讲解了如何使文本显示为粗体或斜体，以及如何编码上标、下标、特殊符号和重音字母。本章还讲解了如何在预先格式化的等宽文本段落中使文本正确对齐，以及如何控制 Web 页面上的任何文本区域的大小、颜色和字形。本章还介绍了一些用于指定许多 HTML 标签的选项和特殊行为的属性，并且讲解了结合使用 style 属性和 CSS 样式规则来影响文本的外观。最后，本章介绍如何使用 Web 字体，来丰富页面的表示。

表 6.2 总结了本章中讨论的标签和属性。

表 6.2 本章介绍的 HTML 标签和属性

| 标签/属性 | 作用 |
|---|---|
| ... | 强调（通常是斜体） |
| ... | 更重的强调（通常是粗体） |
| <pre>...</pre> | 预先格式化的文本（将保留精确的换行符和间距，通常以等宽字体呈现） |
| <sub>...</sub> | 下标 |
| <sup>...</sup> | 上标 |

异步社区会员 Kensuke(18319066421) 专享 尊重版权

续表

| 属性 | |
|---|---|
| style="font-family:typeface" | 字体的字形（系列），它是字体的名称，比如 Arial（也可用于<p>、<h1>、<h2>、<h3>等） |
| style="font-size:size" | 字体的大小，可以设置为 small、medium、large、x-small、x-large 等，也可以设置为具体的磅大小（比如 12pt） |
| style="color:color" | 更改文本的颜色 |

## 6.6 问与答

**问：我怎样查找自己的计算机上所安装的字体的准确名称？**

**答：**在 Windows 计算机上，打开“控制面板”，并单击 Fonts 文件夹，这将列出系统上的字体（Vista 用户可能不得不切换到“控制面板”中的“经典视图”）。在 Mac 上，在 Applications 文件夹中打开 Font Book。当在 font-family 样式规则中指定字体时，可以使用字体名称的准确拼写。不过，字体名称不区分大小写。

**问：我怎样把日文、阿拉伯语、中文和其他非欧洲字符放在我的页面上？**

**答：**首先，需要阅读你的页面上的这些字符的用户必须安装合适的语言字体。他们还必须为其 Web 浏览器选择该语言字符集及其关联的字体。在 Windows 中可以使用“字符映射表”程序（或者其他操作系统中的类似程序），获取任何语言字体中的每个字符的数字代码。要找到“字符映射表”，可以单击“开始”>“所有程序”>“附件”>“系统工具”命令（在 Mac 上，可以打开 Applications 文件夹中的 Font Book）。如果你想要的字符具有代码 214，就使用Ö把它放在 Web 页面上。如果找不到“字符映射表”程序，可以使用操作系统的内置“帮助”功能查找特定的位置。

在亚洲语言中包括短消息（比如“We Speak Tamil-Call Us!”）的最佳方式是将其作为图形图像来包括它。这样，每个用户都将会看到它，即使他们使用英语作为 Web 浏览的主要语言也是如此。但是，甚至对于在图形中使用一种语言字体，也可能不得不为操作系统下载特定的语言包。同样，可以检查系统的“帮助”功能，获取特定的指导。

## 6.7 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 6.7.1 问题

1. 怎样使用样式代替<b>或标签创建一个段落，并使其中前 3 个单词显示为粗体？
2. 怎样表示水的化学公式？
3. 怎样在 Web 页面上显示“© 2013, Webwonks Inc.”？

异步社区会员 Kensuke(18319066421) 专享 尊重版权

4．怎样居中显示整个页面上的所有文本？

## 6.7.2 答案

1．可以使用下面这段代码：

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

2．使用以下代码：

```
H<sub>2</sub>O
```

3．可以使用以下代码之一：

```
&copy; 2013, Webwonks Inc.
```

或

```
&#169; 2013, Webwonks Inc.
```

4．如果考虑把 或者另一个容器元素紧接着放在页面顶部的<body>标签后面，并在页面底部的</body>标签之前使用 ，那么你就是正确的。不过，在<body>标签中也直接支持 text-align 样式，这意味着可以放弃 标签，并且 style="text-align:center"样式直接放在<body>标签中。这样，整个页面将迅速居中显示。

## 6.7.3 练习

- 对多个块级元素（如<p>、 、<ul>和<li>列表项）以及一些语义元素应用字体级样式属性。尝试嵌套元素，以便感受样式如何通过内容层次结构进行层叠或者不层叠。
- 使用文本对齐样式属性把文本块放在 Web 页面上的多个不同的位置。尝试嵌套段落和分区（<p>和 ），以便感受样式如何通过内容层次结构进行层叠或者不层叠。
- 疯狂体验 Web 字体。通过 Google Web Fonts 找到一些 Web 字体，然后在样式表并且最终在页面自身的布局中包括它们。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 7 章

# 处理颜色和边框

在本章中你将学到：

- 怎样为在 Web 上工作的 Web 站点选择颜色；
- 怎样使用 CSS 设置背景、文本和边框颜色；
- 怎样在使用边框的元素上创建圆角。

每一项任务都比较短并且令人轻松愉快，可以帮助你把 Web 开发体验从本书中迄今为止的白色背景/黑色文本示例上转移到更有趣的（或者至少是彩色的）示例上。但是，这并不是说浅色背景上的深色文本就很糟糕，事实上，它是你将在线发现的最常见的颜色组合。

关注配色方案并且制作引人注目的 Web 站点很重要，但是你不必成为一名职业的艺术家，也能在 Web 站点中实现有感染力的配色方案，或者在单调、乏味的方形世界中创造出令人侧目的繁花盛景。本章将帮助你理解颜色理论的最基本的知识，以及如何使用 CSS 修改颜色。

## 7.1 选择颜色的最佳实践

我无法确切地告诉你在 Web 站点中要使用哪些颜色，但是我可以帮助你理解在你自己选择那些颜色时要考虑的某些事项。你使用的颜色可能极大地影响你的访问者。例如，如果你正在运营一个电子商务站点，就可能希望使用能诱惑用户查看商品目录并且最终会购物的颜色。如果你是在创建具有大量文本的站点，就可能希望确保配色方案会导致易于阅读的文本。总之，你希望确保审慎并且怀有敬意地使用颜色。

你可能想知道，在谈论颜色时怎样怀有敬意地做到协调相容，但是要记住，WWW（World Wide Web，万维网）是一个国际性社区，人们的理解会有所差异。例如，粉红色在日本十分流行，但是在东欧国家则非常不受欢迎。类似地，绿色在美国是货币的颜色，但是

异步社区会员 Kensuke(18319066421) 专享 尊重版权

绝大多数其他的国家都具有彩色的纸币，因此“货币的颜色”根本不是单独一种颜色，这样的话，其喻意对于国际访问者将没有价值。

除了使用文化敏感的颜色之外，其他最佳实践包括以下几点。

- 使用自然的调色板。这并不意味着应该使用土色调，而是要使用在城镇里闲逛时自然看到的颜色，避免可能导致眼睛受损的超亮颜色。
- 使用较小的调色板。不需要使用 15 种不同的颜色来实现你的目标。事实上，如果页面中包括具有 15 种不同颜色的文本和图像，就可能要重新评估你尝试发送的消息。关注 3、4 种主色，最多再添加少数几种补色。
- 考虑你的访问者统计数据。你不太可能控制访问者统计数据，因此必须找到一个可以包容每一个人的中间立场。年轻人喜欢的颜色不一定与老年人欣赏的颜色相同，就像男人与女人之间以及来自不同地理区域和文化的人之间会存在颜色偏见一样。

你现在可能认为颜色选项是受限的，事实并非如此，你只需在做决策之前加以考虑。在所选的搜索引擎中搜索“颜色理论”，应该会给你的考虑提供更多的食粮，就像使用色轮一样。

色轮（color wheel）是一张图表，以圆形方式显示了颜色的组织。它的显示方法是尝试帮助你形象地表示主色、辅色与补色之间的关系。配色方案是通过操作色轮开发的，理解配色方案有助于确定要在整个 Web 站点中一致地使用的调色板。例如，知道一些关于颜色关系的知识将有希望使你能够避免在淡蓝色背景上使用橙色文本，或者在棕褐色背景上使用翠蓝色文本。

Web 设计中的一些常见的配色方案如下。

- 类似色：使用色轮上彼此相邻的颜色，比如黄色和绿色。其中一种颜色是主色，其相似的邻近颜色可以丰富显示效果。
- 补色：使用色轮上彼此相对的颜色，比如暖色（红色）和冷色（绿色）。
- 三色：使用在色轮上间隔相等的 3 种颜色。三色方案可以提供平衡感，同时仍然允许使用丰富的颜色。

有些图书和课程是专门介绍颜色理论的，因此在本书中继续这种讨论实在有些偏离主题。不过，如果你打算从事 Web 设计和开发，理解颜色理论的基础知识将受益匪浅。花一些时间学习有关它的知识，在线搜索可以提供大量的相关信息。

此外，还要花一些时间实际动手操作色轮。http://colorschemedesigner.com/上的 Color Scheme Designer 允许从基色开始着手，创建单色、补色、三色、四色、类似色和重类似色配色方案。

## 7.2 理解 Web 颜色

为 Web 页面指定一种不同于白色的背景色比你可能认识到的更容易。例如，要指定蓝色作为页面的背景色，可以把 style="background-color:blue"放在<body>标签内或者放在用于主体元素的样式表规则中。当然，还可以使用许多不同于蓝色的颜色。事实上，W3C 标准

异步社区会员 Kensuke(18319066421) 专享 尊重版权

列出了 17 种颜色：浅绿色、黑色、蓝色、紫红色、灰色、绿色、石灰色、茶色、深蓝色、橄榄色、橙色、紫色、红色、银色、鸟蓝色、白色和黄色。

显然，Web 页面上显示的远不止这 17 种颜色，事实上，可以使用 140 个颜色名称，并且保证所有的浏览器都将以类似的方式显示这些颜色。下面给出了 140 个描述性颜色名称的部分列表：azure、bisque、cornflowerblue、darksalmon、firebrick、honeydew、lemonchiffon、papayawhip、peachpuff、saddlebrown、thistle、tomato、wheat 和 whitesmoke。

但是名称具有主观性。例如，如果查看 140 个跨浏览器的颜色名称的颜色表，将会看到你无法区分 fuchsia 和 magenta。用于这两个名词（fuchsia 和 magenta）的关联的十六进制颜色值也完全相同：#ff00ff。在下一节中将学习关于十六进制颜色值的更多知识，但是目前，要知道的是，如果希望符合标准，并且使用比 W3C 标准规定的 16 个颜色名称更多的颜色，那么只要有可能，就需要使用十六进制颜色代码。

**注意：**要查看 140 个描述性颜色名称的完整列表，以及它们的十六进制代码和浏览器所显示的颜色示例，可以访问 http://www.w3.org/TR/SVG/types.html#ColorKeywords。

**By the Way**

十六进制颜色代码可以产生 1600 万种可能的颜色，并且大多数现代计算机都可以显示所有这些颜色。不过，要知道的是，并非所有的计算机显示器都以相同的色调显示颜色。在你的显示器上可能显示为漂亮的淡蓝色的背景色在另一个用户的显示器上则可能更多地显示紫色色调。中性的土色调的颜色（比如中灰色、棕黄色和乳白色）在许多计算机显示器上甚至可以产生更不可预知的结果。这些颜色甚至可能改变单个显示器上的颜色，这依赖于房间里的光线条件或者一天中的时间。

**提示：**值得指出的是，颜色名称不区分大小写。因此，Black、black 和 BLACK 都是黑色，尽管大多数 Web 设计师坚持使用小写或者混合大小写（如果他们根本不使用颜色名称——大多数设计师都使用十六进制表示法，就会产生一种区别更明显的颜色使用方法）。

**Did you Know?**

除了把页面的背景改为不同于白色的颜色之外，还可以改变文本链接的颜色，包括多个链接属性（比如当用户把鼠标指针悬停在链接上时的颜色以及当用户单击链接时的颜色，就像在前面几章中所学到的那样）。还可以设置容器元素（比如段落、div、醒目引文和表格单元格）的背景色，并且可以使用颜色指定那些元素周围的边框。在本章后面将会看到颜色和容器元素的一些示例。

有大量非常糟糕的 Web 站点，其中一些是由热心人创建的，无论如何，这里没有一丝嘲讽的意思。不过，如图 7.1 所示的 World’s Worst Website 是有意创建的，用于显示 Web 站点设计的一些更臭名昭著的过错，尤其是它的颜色使用。一幅截屏图还不能完全说明问题，你自己可以访问并体验该站点，地址是：http://www.angelfire.com/super/badwebs/main.htm。

如果在搜索引擎中搜索“糟糕的 Web 站点示例”，将会找到许多站点，它们汇集了糟糕设计的示例，并且解释了为什么这样的站点应该归于“耻辱堂”（Hall of Shame），而不是“名人堂”（Hall of Fame）。许多站点由于其视觉显示效果而被认为很糟糕，并且这种显示效果开始于颜色选择。因此，理解颜色及其规范和使用的细微差别是创建良好的 Web 站点的

异步社区会员 Kensuke(18319066421) 专享 尊重版权

至关重要的一步。

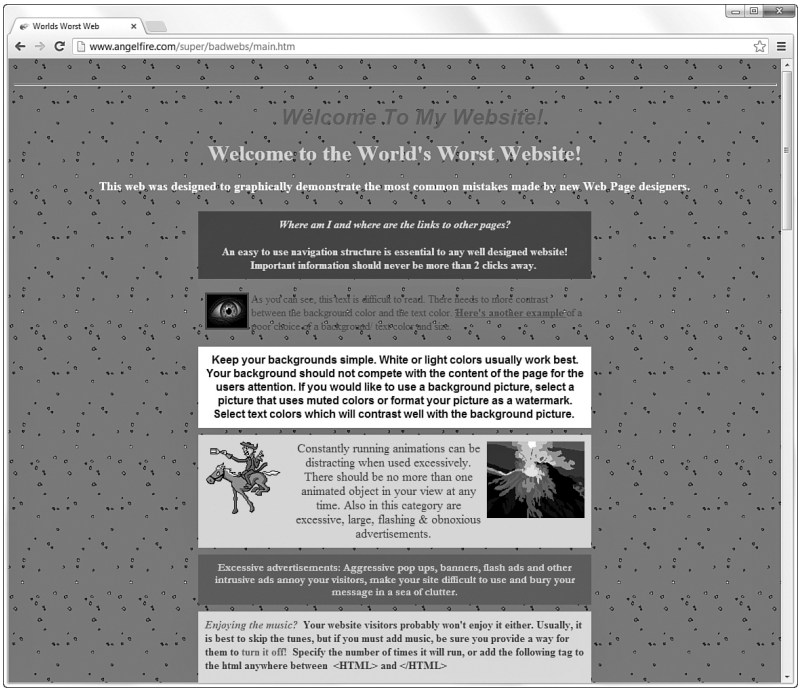



图 7.1

World's Worst Website 的部分截屏图

## 7.3　使用十六进制的颜色值

为了保持符合标准，以及保持对 Web 站点中的颜色的精确控制，可以通过它们的十六进制值来引用颜色。十六进制的颜色值指示应该把多少红色、绿色和蓝色的光线混入每种颜色中。它的工作方式有点像培乐多彩泥（Play-Doh），恰到好处地混入想要的红色、蓝色和绿色量，以获得合适的颜色。

十六进制的颜色格式是#rrggbb，其中 rr、gg 和 bb 是 2 位数字的十六进制值，分别用于颜色中的红色（rr）、绿色（gg）和蓝色（bb）成分。如果你不熟悉十六进制数字，也不要感到焦虑。只要记住：ff 是最大值，00 是最小值。为每种成分使用以下代码。

- ff 意指完全亮度。
- cc 意指 80%的亮度。
- 99 意指 60%的亮度。
- 66 意指 40%的亮度。
- 33 意指 20%的亮度。
- 00 意指没有这种颜色成分。

例如，鲜红色是#ff0000，碧绿色是#003300，蓝紫色是#660099，中灰色是#999999。为了创建具有红色背景和碧绿色文本的页面，可以在内联样式内使用以下 HTML 代码：

```
<body style="background-color:#ff0000; color:#003300">
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

尽管这里只给出了两位数字的十六进制值的 6 个示例，但是它们实际上有 225 种组合：0~9 和 a~f 配对。例如，f0 是一个可能的十六进制值（十进制值为 240），62 也是一个可能的十六进制值（十进制值为 98），等等。

如前所述，#rrggbb 十六进制颜色代码格式中的 rr、gg 和 bb 分别代表颜色中的红色、绿色和蓝色成分。其中每个成分都具有一个十进制值，其范围在 0（无颜色）~255（完全颜色）之间。

因此，白色（或#ffffff）可以转换成红色值 255、绿色值 255 和蓝色值 255。类似地，黑色（#000000）可以转换成红色值 0、绿色值 0 和蓝色值 0。真正的红色是#ff0000（全都是红色，没有绿色，也没有蓝色），真正的绿色是#00ff00（没有红色，全都是绿色，也没有蓝色），真正的蓝色是#0000ff（没有红色，也没有绿色，全都是蓝色）。所有其他的十六进制表示法也都可以转换成这 3 种颜色中每种颜色的 255 个可能值的某个变体。跨浏览器兼容的颜色名称 CornflowerBlue 与十六进制表示法#6495ed 相关联，其中红色值为 100，绿色值为 149，蓝色值为 237（几乎所有可用的蓝色值）。

在挑选颜色时，可以通过图形程序进行，或者在线查找自己喜欢的颜色。你可能会看到十六进制或十进制的颜色表示。如果在搜索引擎中输入“十六进制的颜色转换器”，将会找到众多选项，可以帮助你把颜色值转换成可以在样式表中使用的代码。

## 7.4 使用 CSS 设置背景、文本和边框颜色

在使用 CSS 时，可以在 3 种情况下使用颜色值：在指定背景色、文本颜色和元素的边框颜色时。在第 8 章中，将学习为不同的链接状态使用颜色，因此在本章中，我们将重点介绍基本的元素显示。

图 7.2 显示了一个使用颜色的示例，它可以非常轻松地进入 Web 设计的“耻辱堂”。我不能想象曾经在一个严肃的 Web 站点中使用颜色和样式的这些组合，但它在此可以充当一个示例，说明可能如何将颜色样式应用于不同的元素。本书中印刷的图像很可能不会很好地显示所用的令人恐怖的颜色，因此一定要打开示例文件或者输入程序清单 7.1 中的代码，并在浏览器中加载它。

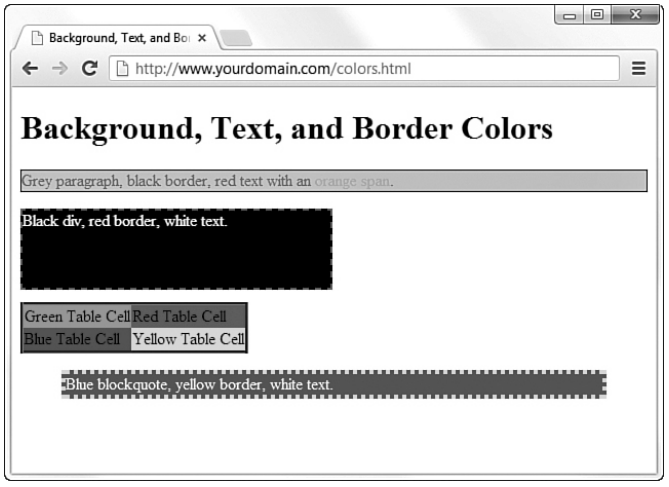



图 7.2

可以使用 CSS 设置背景、文本和边框的颜色

异步社区会员 Kensuke(18319066421) 专享 尊重版权

程序清单 7.1 显示了用于产生图 7.2 的 HTML 和 CSS 样式。

**程序清单 7.1 使用样式产生背景、文本和边框的颜色**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Background, Text, and Border Colors</title>

    <style type="text/css">
    #uglyparagraph {
       background-color:#cccccc;
       color:#ff0000;
       border:1px solid #000000;
    }
    .orange {
       color:#ffa500
    }
    #uglydiv {
       width:300px;
       height:75px;
       margin-bottom: 12px;
       background-color:#000000;
       border:2px dashed #ff0000;
       color: #ffffff;
    }
    table {
       border-width: 1px solid #000;
       border-spacing: 2px;
       border-style: outset;
       border-collapse: collapse;
    }
    .greencell {
       background-color: #00ff00;
    }
    .redcell {
       background-color: #ff0000;
    }
    .bluecell {
       background-color: #0000ff;
    }
    .yellowcell {
       background-color: #ffff00;
    }
    #uglybq {
       background-color:#0000ff;
       border:4px dotted #ffff00;
       color:#ffffff;
    }
    </style>
  </head>

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

       <p id="uglyparagraph">Grey paragraph, black border, red text
       with an <span class="orange">orange span</span>.</p>

       <div id="uglydiv ">Black div, red border, white text. </div>

       <table>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
        <tr>
        <td class="greencell">Green Table Cell</td>
        <td class="redcell">Red Table Cell</td>
        </tr>
        <tr>
        <td class="bluecell">Blue Table Cell</td>
        <td class="yellowcell">Yellow Table Cell</td>
        </tr>
        </table>

        <blockquote id="uglybq">
        Blue blockquote, yellow border, white text.
       </blockquote>
  </body>
</html>
```

查看程序清单 7.1 中的样式，除了一些边框样式之外，所有其他的样式应该基本上都能够弄明白了。在 CSS 中，如果边框还没有宽度和类型，将不能指定一种颜色。在程序清单 7.1 中的第一个示例中，对于 uglyparagraph，边框宽度是 1px，边框类型是 solid。在用于 uglydiv 的示例中，边框宽度是 2px，边框类型是 dashed。在 uglybq 示例中，边框宽度是 4px，边框类型是 dotted。

在为 Web 站点挑选颜色时，要记住“差之毫厘，谬以千里”。如果确实喜欢明亮、壮观的颜色，可以把它用作强调色，而不要在整个主要设计元素中都使用它。出于易读性考虑，记住深色文本搭配浅色背景要比浅色文本搭配深色背景阅读起来容易得多。

最后，要考虑有一部分不容忽视的观众可能是色盲。出于可访问性的目的，你可能考虑使用 http://colorfilter.wickline.org/上的 Colorblind Web Page Filter 工具，看看你的站点在色盲观众看来是什么样子的。

## 7.5 创建圆角

我们将在本章中介绍的最后一个与颜色相关的主题是在元素上创建圆角。你可能想问把方角变成圆角怎么是与颜色相关的主题呢，但是如果把它分解为各个成分，那么制作圆角实际上就是修改元素的边框，并且确保在正确的区域显示正确的颜色。

在最基本的层面上，CSS 的 border-radius 属性用于控制圆角。可以应用边框的任何元素也都可以应用 border-radius 属性，流行的浏览器的所有当前版本（包括大多数移动浏览器）都支持这个属性。

简而言之，创建圆角依赖于椭圆的定义，然后它将代替以前的方角。现在，如果你像我一样并且对所有与几何学相关的内容有畏惧情绪，那么这处理起来要复杂得多。但是，如果我告诉你 CSS 定义了与具有圆角的 元素一起使用的类，那么事情将开始看到一点曙光：

```
.rounded {
   background-color:#00ff00;
   border-radius: 20px;
   height: 150px;
   width: 150px;
}
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

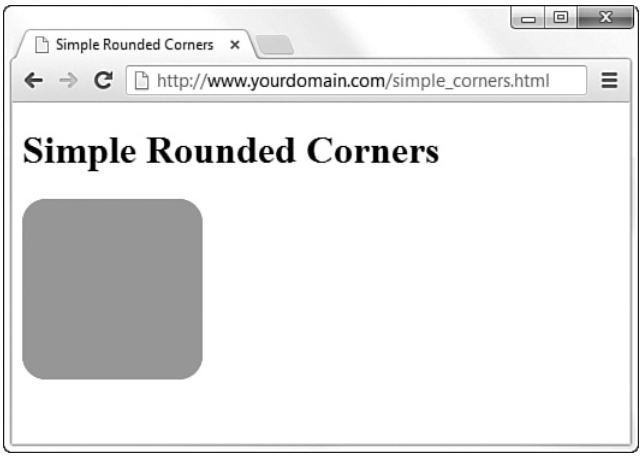



图 7.3

使用 border-radius 显示一个具有圆角的  元素

你已经知道 background-color、height 和 width 属性的作用是什么，因此这里唯一的新项目是 border-radius 属性。在这里，它的值是 20px。border-radius 属性是一个简写的属性，意味着它指代多个其他的属性。border 属性是 border-top、border-right、border-bottom 和 border-left 这些属性的简写形式并且组合了它们，与之非常相似，border-radius 则组合了 border-top-left-radius、border-top-right-radius、border-bottom-right-radius 和 border-bottom-left-radius 这些属性。

这就是说，像下面这一行代码：

```
border: 5px;
```

等价于：

```
border-top: 5px;
border-right: 5px;
border-bottom: 5px;
border-left: 5px;
```

因此，下面这一行代码：

```
border-radius: 20px
```

则等价于：

```
border-top-left-radius: 20px;
border-top-right-radius: 20px;
border-bottom-right-radius: 20px;
border-bottom-left-radius: 20px;
```

现在讨论有些复杂的部分：“20px”实际上指什么？在这里，它指示构成每个圆角的椭圆在长度和高度上的水平和垂直半径都是 20 像素，如图 7.4 所示。

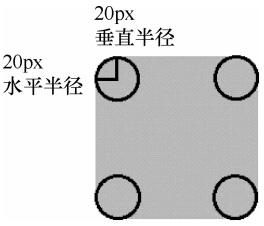



图 7.4

确定用于创建圆角的椭圆的半径

异步社区会员 Kensuke(18319066421) 专享 尊重版权

这里有一种更方便一点的简写形式。由于每个椭圆都有两个半径（水平半径和垂直半径），可以为每个椭圆同时定义它们。不过，如果希望它们是相同的，就可以使用一个值。如果希望两个半径不同，可以使用如下代码：

```
border-top-left-radius: 20px 10px;
```

前面的示例为元素的左上角设置 20px 的水平半径和 10px 的垂直半径。

如你可能猜到的，不仅圆角的水平半径和垂直半径可以具有不同的值，而且每个圆角也可以具有不同的值。换句话说，可以任意定义圆角。图 7.5 显示了另外几个圆角的示例，以及用于创建它们的 border-radius 值。

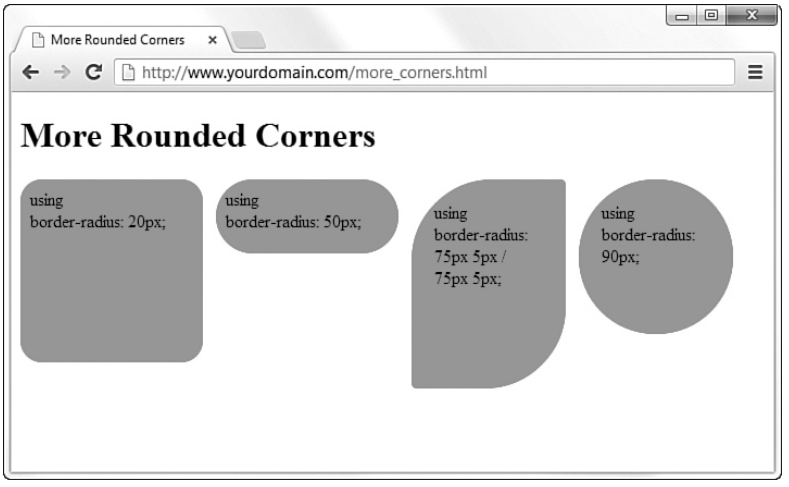



图 7.5
使用 border-radius 在 CSS 中创建形状

## 7.6 小结

本章讲解了几条关于使用颜色的最佳实践，还讲解了如何使用色轮查找可以衬托文本的颜色。此外，本章还介绍了颜色的十六进制表示法——利用与颜色中的红色、绿色和蓝色的数量相关的表示法来表示所有的颜色，并且看到了十六进制表示法怎样允许你对元素应用细致入微的颜色。更重要的是，读者在本章学习了 3 个与颜色相关的样式属性，可以使用 CSS 对容器背景、边框和文本应用颜色。最后，本章介绍了只使用 CSS 和一点数学知识在元素上创建圆角的方法。

## 7.7 问与答

**问：白色背景上的黑色文本确实最适合于在线阅读文本吗？我发现颜色组合在我看来确实很难。**

**答：**白色背景上的黑色文本是用于阅读的最常见的颜色组合之一，因为它是在前景色和背景色之间实现高对比度的最容易的方式——高对比度意味着更好的清晰度。但是纯白色的、明亮的显示器的亮光确实会烦扰一些人，这就是为什么大多数设计师设法坚持指南的精神（如果不是法律条文的话）的原因。也就是说，看到稍浅一点的白色背景和较深的灰色文本代替纯白色和纯黑色变得更常见。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 7.8　测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 7.8.1　问题

1. 怎样给 Web 页面提供一个黑色背景，并使所有的文本都以碧绿色显示？基于你在本章中学到的知识，你希望使用这种颜色组合吗？
2. 你将使用什么 CSS 属性和值来确保某个段落具有白色背景、橙色文本和 3 像素宽的虚线、绿色边框？
3. 如果希望全部 4 个圆角的水平半径和垂直半径都相同（如 45 像素），那么你将为 border-radius 使用什么值？

### 7.8.2　答案

1. 把下面的代码放在 Web 页面的开始处，或者使用针对 body 元素的样式规则：

```
<body style="background-color:#000000; color:#00FF00">
```

2. 可以使用下面的属性和值：

```
background-color: #ffffff;
color: #ffa500;
border: 3px dashed #00ff00;
```

3. 可以使用下面的代码：

```
border-radius: 45px;
```

### 7.8.3　练习

- 选择一种你喜欢的基色——也许是一种可爱的蓝色或者土色，并且使用 http://colorschemedesigner.com/上的 Color Scheme Designer（配色方案设计器）找到一组可以在 Web 站点中使用的颜色。建议使用四色或者较重的类似配色方案。
- 当具有一组颜色（或者几个用于颜色集的选项）时，就可以利用<h1>元素、文本段落以及一些列表项创建一个基本的 HTML 页面。使用你在本章中学到的与颜色相关的样式，更改页面的背景色和多个块级元素的文本，查看这些颜色集可能如何协同工作。看看它们将如何交互，并且确定哪些颜色最适用于容器，以及哪些颜色最适用于纯文本、标头文本和链接文本。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 8 章

# 使用外部链接和内部链接

在本章中你将学到：

- 怎样使用锚点链接；
- 怎样在你自己的站点上的页面之间创建链接；
- 怎样链接到外部内容；
- 怎样链接到电子邮件地址；
- 怎样把窗口作为链接的目标；
- 怎样利用 CSS 编排链接的样式。

迄今为止，你学习了如何使用 HTML 标签创建一些基本的 Web 页面。不过，此时，这些内容本身就像一座座“孤岛”，与任何其他的内容都没有联系（尽管我在示例中偷偷地设置了几个页面链接）。为了把你的工作变成真正的 Web 内容，需要把它连接到 Web 的其余部分，或者至少要连接到你自己的个人站点或公司站点内的其他页面。

本章将说明如何创建指向自己文档内的内容的超文本链接，以及如何链接到其他外部文档。此外，你还将学习如何编排超文本链接的样式，使得它们以你想要的颜色和装饰显示，而不一定显示默认的蓝色加下划线的效果。

## 8.1 使用 Web 地址

为单个 Web 站点存储 Web 内容的最简单的方式是：把所有文件一起存放在同一个文件夹中。当像这样把文件存储在一起时，可以简单地在<a>标签的 href 属性内提供文件的名称，来链接到它们。

属性（attribute）是一份与标签关联的额外信息，用于提供关于标签的进一步的详细信息。例如，<a>标签的 href 属性用于确定你将链接到的页面的地址。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

当具有不止几个页面时，或者当站点中的内容开始具有一种组织结构时，就应该把文件存放在目录（或文件夹[folder]中，如果你愿意这样称呼它的话），并且其名称将会反映其中包含的内容。例如，所有的图像都可以存放在 images 目录中，公司信息则应该存放在 about 目录中，等等。无论怎样在自己的 Web 服务器内组织文档，都可以使用相对地址，它只包括用于查找一个页面的足够信息。

**By the Way**

**注意：**在我们开始前，你可能想复习一下把文件存放在服务器上的什么位置以及在一组目录内如何管理文件的基础知识。在 Web 内容中创建链接时，知道这些信息很重要。可以回头参考一下第 1 章，尤其是 1.8 节的内容。

相对地址（relative address）描述了从一个 Web 页面到另一个 Web 页面的路径，而不是完整的（或绝对的[absolute]）Internet 地址。

回忆第 1 章可知，Web 服务器的文档根目录是被指定为 Web 内容的顶级目录的目录。在 Web 地址中，利用正斜杠（/）表示这个文档根目录。所有后续的目录级别都利用相同类型的正斜框隔开。例如：

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

假设你在文档根目录中创建一个名为 zoo.html 的页面，并且希望包括两个链接，分别指向 elephants 子目录中名为 african.html 和 asian.html 的页面。这些链接看起来如下所示：

```
<a href="/elephants/african.html">Learn about African elephants.</a>
<a href="/elephants/asian.html">Learn about Asian elephants.</a>
```

**Watch Out!**

**警告：**在 HTML 中，总是使用正斜杠（/）来隔开目录。不要使用反斜杠（\，通常在 Windows 中使用它）来隔开目录。记住，Web 中所有的一切都是向前移动的，因此要使用正斜杠。

这些具体的地址实际上称为相对根地址（relative-root address），这是由于它们是缺少整个域名的相对地址，但是它们明确相对于通过正斜杠指定的文档根目录。

使用常规的相对地址，可以省略初始正斜杠。这类地址允许链接变成相对于它们所处的任何目录，它可以是文档根目录，或者是从文档根目录往下一级或更多级的另一个目录：

```
<a href="elephants/african.html">Learn about African elephants.</a>
<a href="elephants/asian.html">Learn about Asian elephants.</a>
```

可以用以下任何一种方式将 elephants 子目录中的 african.html 和 asian.html 文档链接回 zoo.html 主页面：

```
<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。在绝对链接内，链接应该到达的位置是绝对确定的，因为提供了完整的 URL，包括域名。

第二个链接是相对根链接。它相对于目前浏览的域，因此不需要协议类型（如 http://）

异步社区会员 Kensuke(18319066421) 专享 尊重版权

或域名（如 www.yourdomain.com）。这里提供了初始正斜杠，以显示地址开始于文档根目录。

在第三个链接中，双点（..）是一个特殊命令，指示包含当前文件夹的文件夹，即父文件夹（parent folder）。无论何时看到双点，你自己就要想到在目录结构中“向上一级”。

如果在整个 Web 页面内一致地使用相对寻址，就可以把页面的目录移到另一个文件夹、磁盘驱动器或 Web 服务器，而不会改变链接。

如果必要，相对地址可以跨越相当复杂的目录结构。第 23 章中提供了关于组织和链接大量 Web 页面的更详细的建议。

**提示：**围绕相对寻址（elephants/african.html）与绝对寻址（http://www.takeme2thezoo.com/elephants/african.html）的一般规则是：当链接到存储在一起的文件（比如全都是同一个 Web 站点的一部分的文件）时，应该使用相对寻址。当链接到别处（比如另一台计算机、另一个磁盘驱动器，或者更常见的是 Internet 上的另一个 Web 站点）的文件时，就要使用绝对寻址。

**Did you Know?**

Try It Yourself：创建一个简单的站点架构

如果顺利的话，到现在为止，你在学习本书的过程中应该已经创建了自己的一两个页面。遵循下面这些步骤，添加另外几个页面，并把它们链接在一起。

1. 使用主页作为主要入口，以及连接到它的所有其他页面的中枢。如果创建了一个关于你自己或者你的企业的页面，就可以使用该页面作为你的主页。你现在出于此目的还可能喜欢创建一个新页面。
2. 在主页上，放置一个指向你创建的其他 HTML 文件（或者你计划不久将创建的 HTML 文件的占位符）的链接列表。确保在每个链接中文件名的准确拼写（包括任何大小写）都是正确的。
3. 在除主页之外的其他所有页面上，在底部（或顶部）包括一个带回到主页的链接，这可以使站点导航变得简单、容易。
4. 你可能还想在主页或者单独的链接页面上包括一个指向相关或有趣站点的链接列表。人们通常在他们自己的主页上包括他们朋友的个人页面的列表。不过，企业应该小心行事，不要把潜在的顾客过快地带往其他的站点。不能保证他们将记得为你自己的页面之间的链接使用相对寻址，并为指向其他站点的链接使用绝对寻址。

## 8.2 使用锚点在页面内链接

<a>标签（负责 Web 上的超链接的标签）得名于单词 anchor，它意味着链接充当 Web 页面中的某个区域的标示。在本书中迄今为止的示例中，你学习发如何使用<a>标签链接到别的位置，但这只是它的一半用途。让我们开始使用锚点链接，来链接到相同页面内的内容。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

### 8.2.1 利用锚点标识页面中的位置

<a>标签可以用于把页面上的某个区域标记为锚点，使你能够创建一个指向精确区域的链接。本章稍后一点将展示的程序清单 8.1 演示了一个指向页面内的锚点的链接。要查看这样的链接是怎样创建的，让我们提前快速偷窥一下程序清单中的第一个<a>标签：

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

<a>标签通常使用 href 属性来指定超链接的目标。<a href>就是你单击的位置，<a id>则是你单击时所转到的位置。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。作为替代，<a>标签给页面上出现标签的特定位置提供了一个名称。必须包括</a>标签，并且必须给 id 属性分配一个唯一的名称，但是<a>和</a>之间不是必须要有文本。

**Did you Know?**

**提示**：实际上可以在 HTML5 中的许多容器元素上使用 id 属性，并且还可以使用</a>标签指向那些元素作为锚点链接。

### 8.2.2 链接到锚点位置

程序清单 8.1 显示了一个具有多个锚点的站点，这些锚点遍布在单个页面中。查看程序清单 8.1 中的最后一个<a>标签，可以看到一个示例：

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

#符号意味着单词“top”引用的是当前文档内的命名锚点，而不是单独的页面。当用户单击“Return to Index”时，Web 浏览器将显示以<a id="top">开头的页面部分。

**程序清单 8.1 结合使用<a>标签与 id 属性设置锚点**

```
<!DOCTYPE html>

<html lang="en">
 <head>
  <title>Alphabetical Shakespeare</title>
 </head>

 <body>
   <article>

   <header>
     <h1><a id="top"></a>First Lines of Shakespearean Sonnets</h1>
   </header>

   <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>

   <p style="text-align:center"><strong>Alphabetical Index</strong></p>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
<div 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>
</div>
<hr />

<section>

<header>
  <h1><a id="A"></a>A</h1>
</header>
<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">Return to Index.</a></p>
</section>
<hr />
<!-- continue with the alphabet -->
<section>

<header>
  <h1><a id="Z"></a>Z</h1>
</header>

<p>(No sonnets start with Z.)</p>
<p><a href="#top"><em>Return to Index.</em></a></p>
</section>

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

**注意：**在程序清单 8.1 的结尾附近，可以看到如下一行代码:

**By the Way**

```
<!-- continue with thealphabet -->
```

这段文本（HTML 注释）将出现在源代码中，但是不会被浏览器显示出来。在第 23 章中将学习关于给代码加注释的更多知识。

程序清单 8.1 中的每个<a href>链接都用于创建一个通往相应<a id>锚点的加下划线的链接，或者如果我填写了所有的文本，它将会创建相应的链接。在这个示例中，只有“A”和“Z”会工作，因为只有“A”和“Z”链接具有要链接到的相应文本，但是你自己可以自由地填写余下的内容。例如，如图 8.1 所示，单击“Alphabetical Index”下面的字母“Z”，将

异步社区会员 Kensuke(18319066421) 专享 尊重版权

把你带到如图 8.2 所示的页面部分。

**Watch Out!**

**警告：**通过<a>标签中的 id 属性指定的锚点名称必须开始于一个字母数字字符。因此，如果你希望对锚点的 ID 简单地进行编号，就要确保它们开始于文本（比如 photo1、photo2 等），而不仅仅是 1、2 等。

图 8.1

程序清单 8.1 中的<a id>标签根本不会出现在 Web 页面上，<a href>则会显示为加下划线的链接

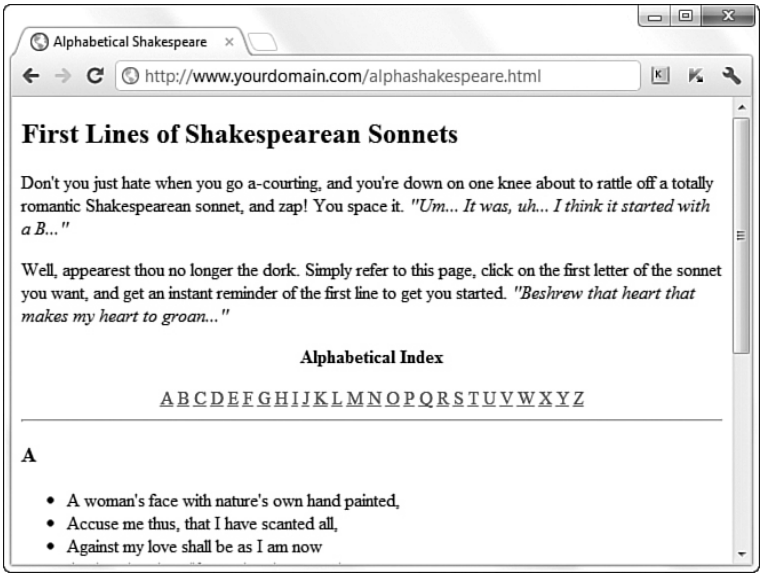
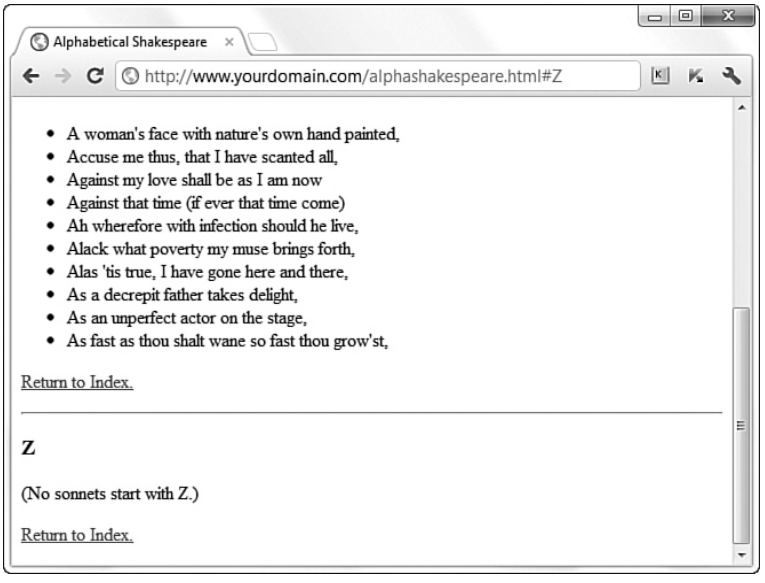



图 8.2

单击图 8.1 中所示页面上的字母“Z”，将把你带到相同页面上的相应区域




在掌握了链接到单个页面内的文本区域的概念之后，现在就可以学习连接其他的 Web 内容。

## 8.3 在自己的 Web 内容之间创建链接

如在本章前面所学过的，在链接到域内（或者同一台计算机上，如果正在本地查看站

异步社区会员 Kensuke(18319066421) 专享 尊重版权

点的话）的内容时，无需在 href 属性中指定的每个地址之前包括“http://”。在创建从一个文件指向相同域内或者相同计算机上的另一个文件的链接时，无需指定完整的 Internet 地址。事实上，如果两个文件存储在同一个文件夹中，就可以只简单地使用 HTML 文件的名称：

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

例如，程序清单 8.2 和图 8.3，显示了一个测验页面，它具有一个链接，指向程序清单 8.3 和图 8.4 中所示的答案页面。答案页面包含一个返回测验页面的链接。由于程序清单 8.2 中的页面链接到相同目录中的另一个页面，就可以使用文件名代替完整的地址。

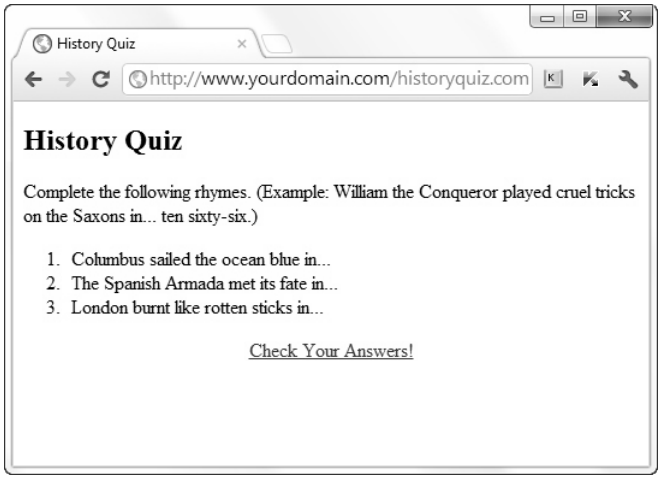



图 8.3

这是在程序清单 8.2 中列出并且被程序清单 8.3 中的链接引用的 historyquiz.html 文件

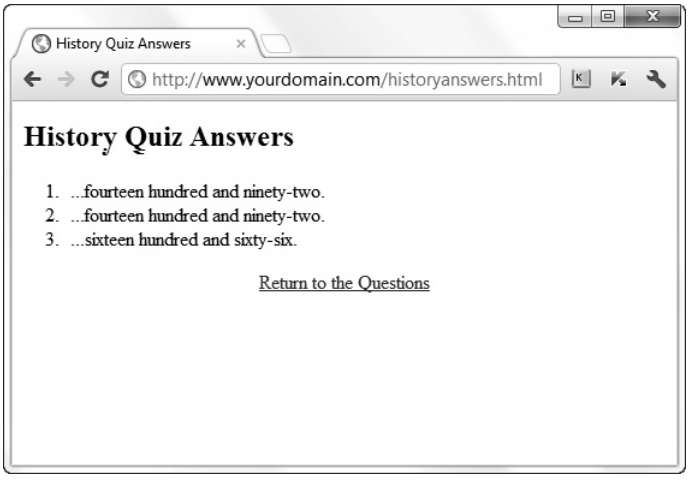



图 8.4

图 8.3 中的“Check Your Answers!”链接将把你带到这个答案页面；“Return to the Questions”则将把你带回图 8.3 中所示的页面

**程序清单 8.2　historyquiz.html[1]文件**

```
<!DOCTYPE html>

<html lang="en">
 <head>
  <title>History Quiz</title>
 </head>
```

1 原文是 historyanswers.html，根据上下文和程序清单中的代码可以看出原文有误。——译者注

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
 <body>
  <section>

 <header>
  <h1>History Quiz</h1>
  </header>

  <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;">
  <a href="historyanswers.html">Check Your Answers!</a>
  </p>

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

程序清单 8.3 historyquiz.html 文件链接到的 historyanswers.html 文件

```
<!DOCTYPE html>

<html lang="en">
 <head>
  <title>History Quiz Answers</title>
 </head>

 <body>
  <section>

  <header>
  <h1>History Quiz Answers</h1>
  </header>

  <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;">
  <a href="historyquiz.html">Return to the Questions</a>
  </p>
  </section>
 </body>
</html>
```

**By the Way**

**注意：**在程序清单 8.2 和程序清单 8.3 中，你将看到在大块内容周围使用了<section></section>标签对。你可能想知道这是否完全必要，毕竟，它是页面上的唯一内容。答案是：不是，它不是完全必要的。HTML 将很好地经过验证，并且即使<section></section>标签不存在，那么任何查看这段代码的人也不会被它的组织结构弄糊涂。这里使用它们只是为了确保你习惯于在所有这些代码示例中都看到它们，以及允许我编写这个“注意”，提醒你在将来某个时间可能怎样使用<section></section>标签。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

例如，如果你想要把问题部分和答案部分放在一个页面上，并且应用样式和一点基于 JavaScript 的交互性，就可以隐藏一个部分（问题），直到阅读者单击一个链接，然后显示另一个部分（答案）。这个动作超出了本书的范围，但它可以在后面作为一个示例，说明如何使用一点最简单的标记为你安排更大的事情。

**By the Way**

使用文件名代替完整的 Internet 地址可以节省许多输入工作。更重要的是，无论页面组存储在什么位置，页面之间的链接都会正确地工作。当文件仍然位于计算机的硬盘驱动器上时，就可以测试链接。然后可以把它们移到 Web 服务器、CD-ROM、DVD 或内存卡上，并且所有的链接仍将正确地工作。这种标识 Web 页面的简化方法没有什么神秘之处，如你已经学过的，它只涉及 Web 页面寻址。

## 8.4 链接到外部 Web 内容

链接到自己站点内的页面与链接到外部内容之间的唯一区别是，当链接到站点外部的内容时，将需要包括该内容的完整地址。完整的地址包括：“http://”、域名和文件的完整路径名（如 HTML 文件、图像文件、多媒体文件等）。

例如，要包括一个从你自己的 Web 页面之一内指向 Google 的链接，将在<a>链接中使用下面这种类型的绝对寻址：

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

**警告：**你可能已经知道，在把任何地址输入到大多数 Web 浏览器中时，可以省略前面的“http://”。不过，在把 Internet 地址输入到 Web 页面上的<a href>链接中时，将不能省略这个部分。

**Watch Out!**

可以应用在前几节中所学的知识，创建指向其他页面上的命名锚点的链接。链接的锚点不受限于相同的页面。可以通过包括地址或文件名，其后接着“#”和锚点名称，来链接到另一个页面上的命名锚点。例如，下面的链接将把你带到 www.takeme2thezoo.com（虚拟）域上的 elephants 目录内的 african.html 页面内的名为 photos 的锚点上。

```
<a href="http://www.takeme2thezoo.com/elephants/african.html#photos">
Check out the African Elephant Photos!</a>
```

如果从已经位于 www.takeme2thezoo.com 域上的另一个页面链接（事实上，因为你是站点维护人员），那么链接可能简单地如下所示：

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

如你已经学过的，在这种情况下，“http://”和域名将不是必要的。

**警告：**确保只在<a href>链接标签中包括“#”符号。不要把该符号放在<a id>标签中；否则，指向该名称的链接将不会工作。

**Watch Out!**

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 8.5 链接到电子邮件地址

除了页面之间以及单个页面的各个部分之间的链接以外，<a>标签还允许链接到电子邮件地址。这是使 Web 页面访问者能够给你反馈信息的最简单的方式。当然，可以保给访问者提供你的电子邮件地址，并且信任他们会把它输入到他们使用的任何电子邮件程序中，但是这将会增加出错的可能性。通过提供一个指向你的电子邮件地址的可单击的链接，可以让他们几乎毫不费力地给你发送消息，并且消除了出现输入错误的可能性。

指向电子邮件地址的 HTML 链接看起来如下所示：

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

“Send me an email message”这些单词将像任何其他的<a>链接一样显示。

如果希望人们看到你的实际的电子邮件地址（以便他们可以注意到它，或者使用不同的电子邮件程序发送消息），可以把它包括在 href 属性中，并且作为<a>和</a>标签之间的消息的一部分，如下所示：

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

在大多数 Web 浏览器中，当用户单击链接时，那个人将获得一个窗口，他或她可以在其中输入消息，并立即发送给你，将自动使用那个人用于发送和接收电子邮件的电子邮件程序。在链接中可以提供一些额外的信息，使得消息的主题和正文也具有默认值，其方法是在 mailto 链接中添加 subject 和 body 变量，利用问号（?）把变量与电子邮件地址分隔开，然后利用“与”符号（&）把每个变量/值对分隔开。此时，不必理解变量/值的术语。下面给出了一个示例，为上述电子邮件示例指定主题和正文：

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

**Did you Know?**

**提示**：如果只想指定电子邮件主题，而不想指定正文，则只需省略“与”符号（&）和 body 变量、等于号以及值文本字符串即可，如下所示：

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

当用户单击这个链接时，将创建一封电子邮件，其中利用“author@somedomain.com”作为收件人，利用“Book Question”作为邮件的主题，并且利用“When is the next edition coming out?”作为邮件的正文。

在你完成设置并开始把电子邮件地址贴满所有的 Web 页面之前，我不得不提出一点警告，并且让你参与一个方便的技巧。你无疑熟悉垃圾邮件发送者，他们构建电子邮件地址的数据库，然后利用垃圾邮件广告兜售这些地址。垃圾邮件发送者获得电子邮件地址的一种方式是：使用可以自动搜索 Web 页面以获取 mailto 链接的程序。

幸运的是，一个小技巧即可挫败许多（但是并非全部）垃圾邮件发送者。这个技巧涉及使用字符实体编码电子邮件地址，它可以迷惑试图从 Web 页面中获取电子邮件地址的 scraper 程序。例如，考虑电子邮件地址 jcmeloni@gmail.com。如果利用对应的字符实体替

异步社区会员 Kensuke(18319066421) 专享 尊重版权

换地址中的字母，将使一些电子邮件获取程序不知所措。小写的 ASCII 字符实体开始于“a”，用于字母“a”，然后按字母表的顺序递增。例如，字母“j”是“&#106”，c 是“&#99”，依此类推。利用 ASCII 属性替换所有的字符将产生如下结果：

```
<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>
```

由于浏览器把字符编码解释为字符，因此从浏览器的角度看最终的结果是相同的。不过，自动的电子邮件获取程序将搜索页面的原始 HTML 代码，在这种情况下，将显示看起来相当混乱的电子邮件地址。如果不希望查明你自己的地址的字符编码，只需在搜索引擎中输入“email address encoder”（电子邮件地址编码器），就会发现一些在线服务，它们可以为你产生编码的字符串。

**提示**：如果把电子邮件联系人链接放在你的所有 Web 页面的脚注中，那么其他人就很容易联系你。你给他们提供了一种方式，告诉你关于页面的任何问题，而它们可能是你在测试时所遗漏的。不要忘记使用电子邮件地址字符实体技巧，使得你的地址可以逃离垃圾邮件发送者的雷达的搜捕。

*Did you Know?*

## 8.6 在新的浏览器窗口中打开链接

既然你已经理解了如何为链接（包括内部链接（在站点内）和外部链接（指向其他站点））创建地址，下面将要介绍另外一种链接方法：强制用户在新窗口中打开链接。

你无疑听说过弹出式窗口（pop-up window），它们是浏览器窗口，通常用于给产品或服务做广告，打算自动打开和显示，而无需用户准许（许多现代的浏览器都会禁止这种行为）。不过，打开另一个窗口或者把另一个位置作为目标的概念在一些情况下具有正当的目的。例如，你可能想在一个较小的辅助浏览器窗口中展示信息，但是仍然允许用户在主窗口中查看信息。当单击一个指向动画演示、电影剪辑或其他多媒体元素的链接时，通常会发生这种情况。当链接到站点外面的内容时，你也可能希望把新的浏览器窗口作为目标。

target 这个单词很重要，因为它是用于<a>标签的属性的名称。target 属性指向有效的浏览环境，或者“要打开的新窗口”。

将在新窗口中打开的一个有效的 HTML 链接构造如下：

```
<a href="/some/file.html" target="_blank">Open a Window!</a>
```

记住，代表用户打开一个新的浏览器窗口（尤其是当它是一个全尺寸的新窗口时）违反了有用性和可用性的一些原则，因此要斟酌是否以及何时要这样做。

## 8.7 使用 CSS 编排超链接的样式

基于文本的超链接在 Web 页面上默认将显示为加下划线的蓝色文本。你还可能注意到，你以前访问过的链接显示为加下划线的紫色文本，这种颜色也是默认的。如果你在 Web 上花费了很多时间，那么还会注意到并非所有的链接都是蓝色或紫色的，对于这种情

异步社区会员 Kensuke(18319066421) 专享 尊重版权

况，我认为，我们都应该心怀感激。使用一点 CSS 并且了解用于<a>链接的各个伪类，就可以使链接看起来就像自己想要的。

**By the Way**

**注意：**可以通过在<a>开始标签与</a>封闭标签之间放置一个<img />标签，使用图形作为链接（代替使用文本作为链接）。

伪类（pseudoclass）是一个描述用于元素的样式的类，它们适用于某些环境，比如用户与该元素交互的多种不同的状态。

例如，用于<a>标签的公共伪类是 link、visited、hover 和 active。可以利用助记符“Love–Hate”——LV（love）HA（hate）记住它们。

- a:link 描述以前未被访问的超链接的样式。
- a:visited 描述以前被访问过并且存在于浏览器的记忆中的超链接的样式。
- a:hover 描述当用户的鼠标悬停于其上时（并且在单击它之前）的超链接的样式。
- a:active 描述正在被单击但是尚未释放鼠标的超链接的样式。

例如，假如你希望创建一个具有以下样式的链接。

- 字体是加粗的 Verdana 字体（并且没有下划线，意味着它没有文本装饰）。
- 基色是淡蓝色。
- 当用户把鼠标悬停于其上或者当他们单击它时变成红色。
- 当用户访问过它之后变成灰色。

样式表条目看起来可能如下所示：

```
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;
}
```

**By the Way**

**注意：**这个示例中的颜色是通过它们的十六进制值指示的。

由于无论处于什么状态，示范性的链接是加粗的 Verdana 字体（并且没有下划线），因此可以把这 3 个属性/值对放在用于 a 选择器的规则中。不过，由于每个伪类都必须具有与之关联的特定颜色，我们可以为每个伪类使用一个规则，如代码示例中所示。伪类将继承父规则的样式，除非用于伪类的规则明确覆盖该规则。换句话说，上一个示例中的所有伪类都将是加粗的 Verdana 字体（并且没有下划线）。不过，如果我们使用了以下用于 hover 的伪类，那么当用户把鼠标悬停在文本上时，将以 Comic Sans 字体显示它（事实上，如果用户

异步社区会员 Kensuke(18319066421) 专享 尊重版权

安装了 Comic Sans 字体的话）：

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

此外，由于 active 和 hover 伪类使用相同的字体颜色，因此可以把用于它们的规则结合起来：

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

程序清单 8.4 把这 3 个代码段放在一起，使用样式化的伪类创建页面，图 8.5 显示了这段代码的结果。




图 8.5

链接可以使用特定的样式控制视觉显示

**程序清单 8.4　使用样式显示链接伪类**

```
<!DOCTYPE html>

<html 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>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    <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>
```

如果在 Web 浏览器中查看示例，的确应该以淡蓝色、加粗并且未加下划线的 Verdana 字体显示链接。如果把鼠标悬停在链接上，或者单击链接并且不释放鼠标，它应该变成红色。如果单击并释放链接，页面将简单地重新加载，因为该链接指向具有相同名称的文件。不过，在那个时刻，链接将处于浏览器的记忆中，因此将被显示为一个访问过的链接，并且它将显示为灰色，而不是蓝色。

可以使用 CSS 对链接应用广泛的与文本相关的改变。可以改变字体、大小、粗细度、装饰等。有时，你可能希望样式表中有几组不同的链接样式。在这种情况下，可以创建类，不限于只为<a>标签使用一组样式。下面的示例是一组用于 footerlink 类的样式规则，这个类用于我可能想要放在 Web 站点的脚注区域中的链接。

```
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;
}
```

从下面的示例中可以看到，类名（footerlink）出现在选择器名称（a）之后，它们之间用点隔开，并且出现在伪类名（hover）之前，它们之间用冒号隔开：

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

## 8.8 小结

<a>标签使文本变成“超”文本。利用它，可以在页面之间创建链接，以及创建指向任何页面上的特定锚点的链接。本章重点介绍了使用相对或绝对寻址标识页面，创建指向其他页面的简单链接，并且编排它们的样式。

你认识到在创建指向其他人的页面的链接时，在<a href>标签中包括每个页面的完整 Internet 地址很重要。对于你自己的页面之间的链接，只需包括文件名和足够的目录信息，即可从一个页面到达另一个页面。

本章还讲解了如何在页面内创建命名锚陆海空，以及如何创建指向特定锚点的链接。此外，还介绍了如何链接到电子邮件地址，以便用户可以轻松地给你发送消息，以及如何保护电子邮件地址，以防它们被垃圾邮件发送者获取。最后，介绍了一些使用 CSS 控制链接的显示效果的方法。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

表 8.1 总结了本章中讨论的<a>标签。

表 8.1 本章介绍的 HTML 标签和属性

| 标签/属性 | 作用 |
| --- | --- |
| <a>...</a> | 利用 href 属性，创建指向另一个文档或锚点的链接。利用 id 属性，创建一个可以链接到的锚点 |
| 属性 | |
| href="address" | 要链接到的文档或锚点的地址 |
| id="name" | 文档中的这个锚点的名称 |

## 8.9 问与答

**问：如果链接到 Internet 上的某个页面，然后拥有该页面的人删除或移动它，则会发生什么事情？**

**答：**这依赖于外部页面的维护人员是怎样建立他的 Web 服务器的。通常，在单击被移动过或者删除的链接时，将会看到“page not found”（页面未找到）消息或者类似的效果。此时，仍然可以单击 Back（后退）按钮返回到你的页面。作为一名站点维护人员，可以周期性地运行链接检查程序，确保内部和外部链接都是有效的。它是一个示例是 Link Checker 服务，位于 http://validator.w3.org/checklink 上。

**问：我的 Web 站点上的其中一个内部链接在我的计算机上工作得很好，但是当我把页面放到 Internet 上时，该链接却不再会工作。这是怎么啦？**

**答：**最有可能是由于以下原因。

- 大小写问题。在 Windows 计算机上，利用<a href="myfile.html">链接到名为 MyFile.html 的文件将会工作。在大多数 Web 服务器上，该链接必须是<a href="MyFile.html">（否则必须把文件改名为 MyFile.html）。甚至使事情更糟糕的是，一些文本编辑器和文件传输程序实际上会在不告诉你的情况下改变大小写。最佳的解决方案是坚持为 Web 页面都使用小写的文件名。
- 文件名中的空格。大多数 Web 服务器不允许文件名中带有空格。例如，永远也不应该把 Web 页面命名为“my page.html”。可代之以将其命名为“mypage.html”，或者甚至是“my_page.html”或“my-page.html”（使用下划线或短划线代替空格）。
- 本地绝对地址。如果使用本地绝对地址链接到一个文件，比如 C:\mywebsite\news.html，当把该文件放到 Internet 上时，链接将不会工作。永远也不应该使用本地绝对地址。当发生这种情况时，它通常是由创建用于测试页面某个部分的临时链接引发的意外事件。因此，在把页面发布到 Web 上之前，要仔细地删除任何测试链接。

**问：我能够把 href 和 id 放在同一个<a>标签中吗？我将希望出于任何原因这样做吗？**

**答：**你能够这样做，并且如果你具有彼此相邻的命名锚点和链接，它可以节省一些输

异步社区会员 Kensuke(18319066421) 专享 尊重版权

入工作。不过，单独使用<a href>和<a id>一般会更好，这样就可以避免混淆，因为它们在HTML 文档中起着非常不同的作用。

**问：如果我意外地拼写错了锚点的名称或者忘记了在它前面添加“#”，则会发生什么事情？**

**答：**如果链接到页面内不存在的锚点名称或者拼写错了锚点名称，链接将转到那个页面的顶部。

## 8.10 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 8.10.1 问题

1. 你上小学时的好朋友在 Internet 上找到了你，并且说他希望交换主页链接。你怎样把一个指向他的站点（http://www.supercheapsuits.com/~billybob/）的链接放在你的页面之一上？
2. 你将使用什么 HTML 代码，使得某个单击页面顶部的“About the Authors”这些单词的人有可能向下跳转到页面上别的位置的工作人员列表？
3. 如果你的电子邮件地址是 bon@soir.com，你将怎样把文本“goodnight greeting”变成一个链接，使得人们可以单击它，以便写一封电子邮件，并发送给你。

### 8.10.2 答案

1. 把下面的代码放在你的页面上：

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

2. 在页面顶部输入以下代码：

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

3. 在工作人员区域的开始处输入以下代码：

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

4. 在 Web 页面上输入以下代码：

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

### 8.10.3 练习

- 创建一个 HTML 文件，其中包括你最喜爱的 Web 站点的格式化的列表。你可能已

异步社区会员 Kensuke(18319066421) 专享 尊重版权

经在 Web 浏览器中收藏了这些站点，这样就可以访问它们，在浏览器的地址栏中发现准确的 URL。

- 如果为一个 Web 站点创建了任何页面，可以仔细检查它们，并且考虑文本中是否有任何空间，你打算使用它让人们可以轻松地联系你。在那里包括一个指向你的电子邮件地址的链接。你可能永远也不会提供太多的机会让人们联系你，并且告诉你他们需要什么或者他们是怎样看待你的产品的，尤其是当你在运营一家企业时。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 9 章

# 使用表格和栏

在本章中你将学到：

- 怎样创建简单的表格；
- 怎样控制表格的大小；
- 怎样对齐表格内的内容并且跨越行和列；
- 怎样使用 CSS 栏。

在本章中，你将学习如何构建 HTML 表格，可以利用它们来控制 Web 内容中的表格式数据的间距、布局和外观。尽管可以使用 CSS 实现类似的结果，但是肯定有些时间使用表格在行和列中展示信息是最佳的方式。在本章中还将看到设计师在过去为何必须为页面布局使用表格，以及在将来如何避免这样做。最后，你将学习出于布局目的使用 CSS 创建栏的基本知识。

在我们开始前，记住表格只是将内容有序地安排进垂直的列和水平的行中。

## 9.1 创建一个简单的表格

表格由信息行组成，其中包含单独的单元格。要创建表格，必须从一个<table>标签开始。当然，要利用</table>标签结束表格。CSS 包含许许多多的属性，使你能够修改表格本身，比如你在前几章学过的多个边框属性。

***Did you Know?***

**提示：**在学习本章的过程中，考虑怎样把文本安排进表格或栏中可以使 Web 内容受益。下面列出了一些要记住的具体思想。

- 表格的最明显的应用是组织表格式信息，比如多列的名称和编号列表，非常像电子数据表。
- 无论何时需要多栏的文本或图像，比如在传统的报纸印刷品中，都可以考虑使用 CSS 栏。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

在具有了<table>标签之后，接下来需要<tr>标签。<tr>标签用于创建表格行，在该标签与</tr>封闭标签之间包含一个或多个信息单元格。要创建这些单独的单元格，可以使用<td>标签（<td>代表表格数据[table data]）。把表格信息放在<td>与</td>标签之间。单元格（cell）是一个矩形区域，可以包含任何文本、图像和 HTML 标签。表格中的每一行都包含至少一个单元格，一行中的多个单元格构成表格中的列。

在构建表格时还会涉及另一个基本的标签，即<th>标签。<th>标签的工作方式与<td>标签非常相似，只不过<th>指示单元格是表格标题的一部分。大多数 Web 浏览器都自动把<th>单元格中的文本呈现为居中和粗体样式，就像利用 Chrome 浏览器在如图 9.1 所示的那样。不过，如果浏览器没有利用内置的样式自动呈现这个元素，就可以使用 CSS 编排元素的样式，而不必使用类来区分不同类型的表格数据元素。

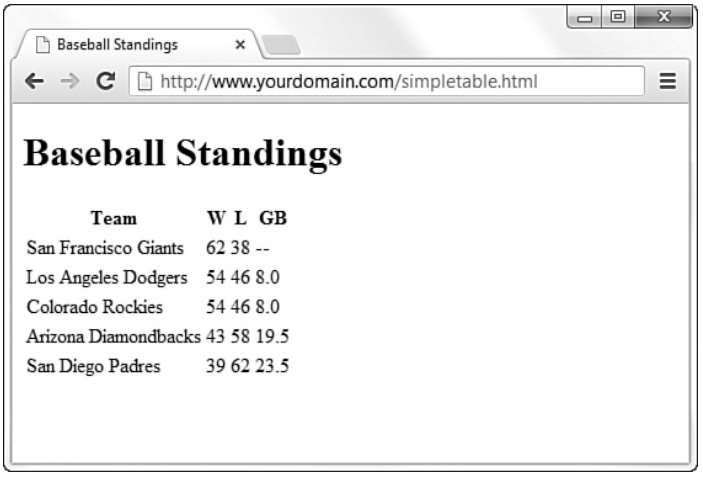



图 9.1

程序清单 9.1 中的代码创建一个 6 行、4 列的表格

可以根据需要创建许多单元格，但是表格中的每一行都应该具有与其他行相同的列数。程序清单 9.1 中的 HTML 代码只使用了我迄今为止提及的 4 个表格标签创建一个简单的表格。图 9.1 显示了在 Web 浏览器中查看时的结果页面。

**程序清单 9.1　利用<table>、<tr>、<td>和<th>标签创建表格**

```
<!DOCTYPE html>

<html 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>
        <td>San Francisco Giants</td>
        <td>54</td>
        <td>46</td>
        <td>8.0</td>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      </tr>
      <tr>
        <td>Los Angeles Dodgers</td>
        <td>62</td>
        <td>38</td>
        <td>—</td>
      </tr>
      <tr>
      <tr>
        <td>Colorado Rockies</td>
        <td>54</td>
        <td>46</td>
        <td>8.0</td>
      </tr>
      <tr>
        <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：**你可能发现：如果使用空格缩进<tr>和<td>标签，就像我在程序清单 9.1 中所做的那样，那么 HTML 表格将更容易阅读（并且不太容易出现费时的错误）。记住，浏览器在呈现 HTML 时将会忽略空格，因此代码的布局不会影响人们将看到的表格的布局。

示例中的表格包含棒球排行榜，它们非常适合于安排进行和列中，但是它们有一点朴素。例如，这个示例甚至没有任何边框，你稍后将学习使它们变得活泼一点。表格的标题显示排行榜中的 Team（球队）、Wins（W，胜场）、Losses（L，负场）和 Games Behind（GB，落后战绩最领先球队的胜场数）。

添加以下样式表条目，它们负责在表格及其单元格周围添加一条基本的边框：

```
table, tr, th, td {
   border: 1px solid black;
   border-collapse: collapse;
   padding: 3px;
}
```

你可能想知道为什么必须为用于创建表格的全部 4 个元素都指定这些样式，而不仅仅是为总的表格元素本身指定它们。基本上，这是因为表格是由它的元素组成的，并且每个元素都可以应用这些样式。下面的图形演示了对不同的元素应用或者不应该样式时表格的外观，用于强调这一点。

图 9.2 显示了刚才列出的样式的输出结果。border-collapse 属性（值为 collapse）使<table>、<tr>以及<th>或<td>元素的所有边框都折叠成一条共享的边框。填充则给单元格的

异步社区会员 Kensuke(18319066421) 专享 尊重版权

内容添加了一点呼吸的空间。

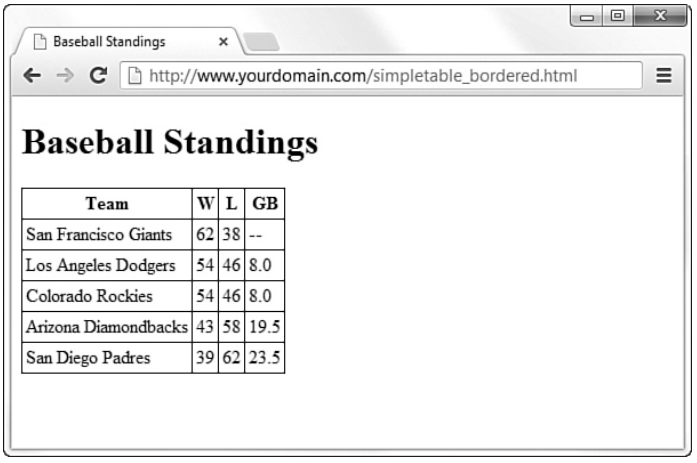



图 9.2

给表格添加一些 CSS 样式，包括使用 border-collapse 属性

如图 9.3 所示，可以看到表格看上去就像没有指定 border-collapse 属性一样（这样，默认值将起作用，它是 separate，用于单独的边框）。

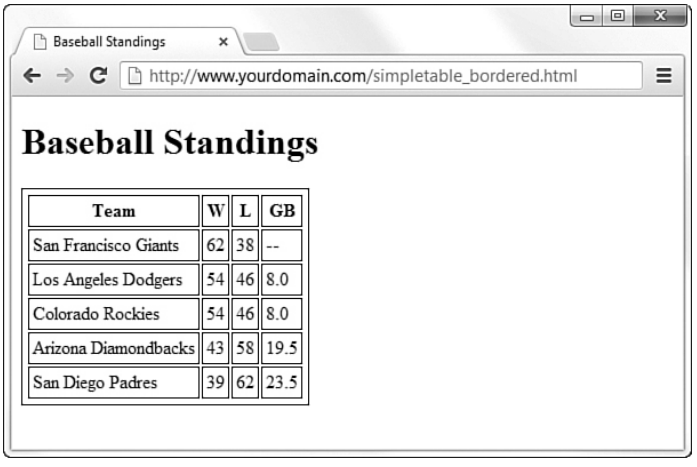



图 9.3

删除 border-collapse 属性将显示所有元素的边框

在图 9.4 中可以看到表格看上去就像没有为<th>和<td>元素指定任何以前的样式一样，注意缺少指示列的边框。

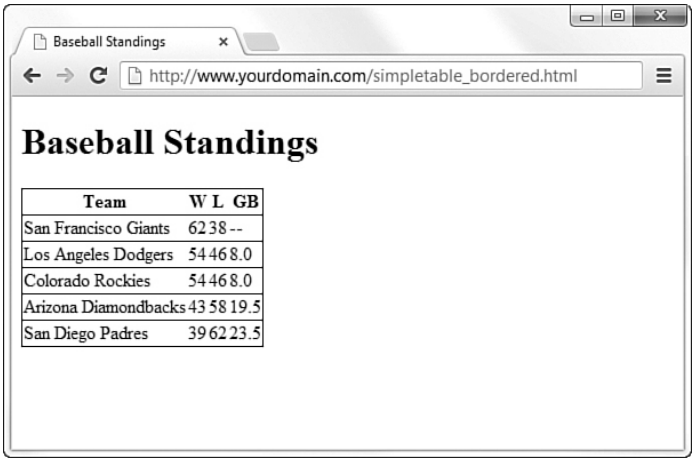



图 9.4

删除用于<th>和<td>元素的样式

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**Did you Know?**

**提示：**在创建简单的表格时，可以利用另外 3 个有用但是并非必需的与表格相关的标签。

- <thead></thead>：包装这个元素中的标头行，给组添加更多的含义，还允许这些标头行跨所有页面打印出来（如果表格足够长的话）。然后可以编排<thead>元素（而不是单个<th>单元格）的样式。
- <tbody></tbody>：包装这个元素中构成这个表格的“主体”的行（除标头和脚注行以外的所有行），给组添加更多的含义，然后还可以把<tbody>元素作为一个整体来编排样式，而不是编排各个<td>单元格的样式。
- <tfoot></tfoot>：与<thead>元素非常相似，使用它包装这个元素中的脚注行（如果具有任何脚注行的话），给组添加更多的含义，并将其作为一个整体编排样式。脚注行的一个示例可能是列中展示的数据之和，比如财务总计。

## 9.2 控制表格大小

当没有指定表格宽度时，表格的大小及其各个单元格将会自动扩展，以适应放入其中的数据。不过，可以通过为<table>元素定义 CSS width 属性，来控制整个表格的宽度。也可以通过给<td>元素的 CSS width 属性，定义每个单元格的宽度。可以把 width 属性指定为像素、“字母 m 的宽度”（em）或百分比。

为了使表格的第一个单元格占据表格总宽度的 20%，并使第二个单元格占据表格宽度的 80%，可以使用以下属性定义：

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

注意：表格的大小被设置为 100%，这确保它可以填满浏览器窗口的整个宽度。当使用百分比代替固定的像素尺寸时，表格将自动调整大小，以适应任何尺寸的浏览器窗口，同时维持你所寻求的具有美感的平衡。在这里，表格内的两个单元格将自动分别调整为表格总宽度的 20%和 80%。

在程序清单 9.2 中，扩展了程序清单 9.1 中的简单表格（以及与边框相关的样式），以显示对表格单元格宽度的非常精确的控制（并且添加了与边框相关的样式）。

**程序清单 9.2 指定表格单元格的宽度**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Baseball Standings</title>

    <style type="text/css">
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    table, tr, th, td {
       border: 1px solid black;
       border-collapse: collapse;
       padding: 3px;
    }
    </style>
  </head>

 <body>
  <h1>Baseball Standings</h1>
    <table>
      <tr>
        <th style="width:200px;">Team</th>
        <th style="width:25px;">W</th>
        <th style="width:25px;">L</th>
        <th style="width:25px;">GB</th>
      </tr>
      <tr>
        <td>San Francisco Giants</td>
        <td>62</td>
        <td>38</td>
        <td>--</td>
      </tr>
      <tr>
        <td>Los Angeles Dodgers</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>
      <tr>
        <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>
```

在图 9.5 中可以看到一致的列宽度。

为第一行中的每个<th>元素添加特定的宽度样式来定义列的宽度。第一列被定义为 200px 宽，而第二列、第三列和第四列的宽度都是 25px。在图 9.5 中，可以在第一列中看到文本后面的空白，指示指定的宽度确实大于列本身所具有的宽度，从而允许表格呈现出来，而无需明确的宽度指示符。

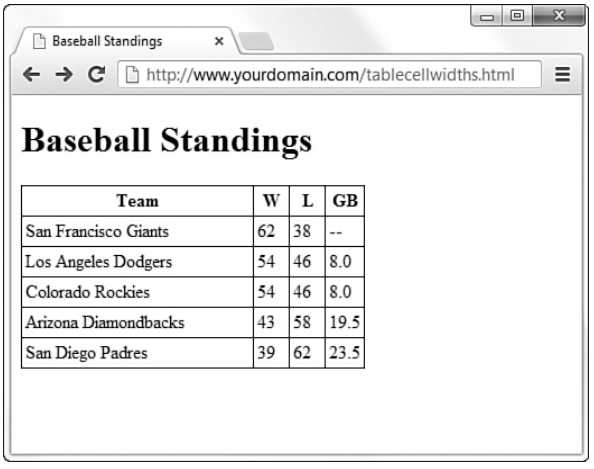
另请注意：在后面的行中的<td>元素中没有重复这些宽度。从技术上讲，只需要在第一

异步社区会员 Kensuke(18319066421) 专享 尊重版权

行中定义宽度，余下的行将亦步亦趋，因为它们都是相同表格的一部分。不过，如果使用另一种格式化样式（比如用于更改字体大小或颜色的样式），将不得不为每个应该具有那些显示属性的元素重复该样式。

图 9.5

程序清单 9.2 中的代码创建一个 6 行、4 列的表格，并且为每一列使用特定的宽度




## 9.3 表格内的对齐和跨越

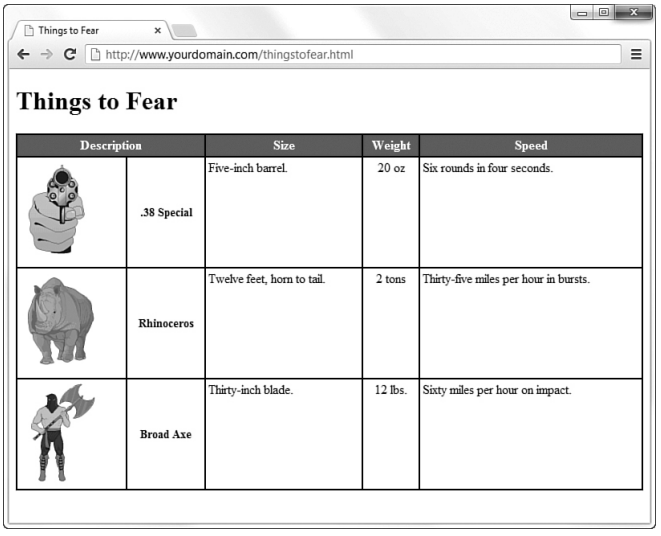
默认情况下，放在表格单元格内的任何内容都会左对齐并且垂直居中。本章中迄今为止所有的图形都显示了这种默认的对齐方式。不过，可以利用 text-align 和 vertical-align 样式属性水平和垂直地对齐表格单元格的内容。

可以对任何<tr>、<td>或<th>标签应用这些对齐属性。分配给<tr>标签的对齐属性将应用于该行中的所有单元格。依赖于表格的大小，可以通过在<tr>级别（而不是在每个<td>或<th>标签中）应用这些属性，使自己节省一些时间和工作。

程序清单 9.3 中的 HTML 代码使用了一种文本对齐样式的组合，对某一行应用默认的对齐方式，但是在几个单独的单元格中覆盖了它。图 9.6 显示了程序清单 9.3 中的代码的结果。

图 9.6

程序清单 9.3 中的代码显示了 colspan 和一些对齐样式的用法




异步社区会员 Kensuke(18319066421) 专享 尊重版权

下面列出了一些更常用的 vertical-align 样式属性值：top、middle、bottom、text-top、text-bottom 和 baseline（用于文本）。在垂直对齐表格数据时，这些属性值给你提供了非常大的灵活性。

**程序清单 9.3　表格中的对齐、单元格间距、边框和背景色**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Things to Fear</title>
    <style type="text/css">
    table {
       border: 2px solid black;
       border-collapse: collapse;
       padding: 3px;
       width: 100%
    }

    tr, th, td {
       border: 2px solid black;
       border-collapse: collapse;
       padding: 3px;
    }

    thead {
       background-color: #ff0000;
       color: #ffffff;
    }

    .aligntop {
       vertical-align:top;
    }

    .description {
       font-size: 14px;
       font-weight:bold;
       vertical-align:middle;
       text-align:center;
    }
    .weight {
       text-align:center;
    }
    </style>

  </head>
  <body>
    <h1>Things to Fear</h1>
    <table>
      <thead>
      <tr>
        <th colspan="2">Description</th>
        <th>Size</th>
        <th>Weight</th>
        <th>Speed</th>
        </tr>
      </thead>
      <tr class="aligntop">
        <td><img src="handgun.gif" alt=".38 Special"/></td>
        <td class="description">.38 Special</td>
        <td>Five-inch barrel.</td>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      <td class="weight">20 oz.</td>
      <td>Six rounds in four seconds.</td>
    </tr>
      <tr class="aligntop">
      <td><img src="rhino.gif" alt="Rhinoceros" /></td>
      <td class="description">Rhinoceros</td>
      <td>Twelve feet, horn to tail.</td>
      <td class="weight">2 tons</td>
      <td>Thirty-five miles per hour in bursts.</td>
    </tr>
    <tr class="aligntop">
      <td><img src="axeman.gif" alt="Broad Axe" /></td>
      <td class="description">Broad Axe</td>
      <td>Thirty-inch blade.</td>
      <td class="weight">12 lbs.</td>
      <td>Sixty miles per hour on impact.</td>
    </tr>
  </table>
 </body>
</html>
```

在图 9.6 顶部，一个单元格（Description）跨越了两列。这是利用针对那个单元格的<th>标签中的 colspan 属性完成的。如你可能猜到的，还可以使用 rowspan 属性创建一个跨越多行的单元格。

**Did you Know?**

**提示：**在创建其中的单元格将跨越多列或多行的表格时，保持行和列的结构组织有序可能是最困难的部分。最微小的错误通常也可能会把整个事情弄得一团糟。在开始编写 HTML 代码实现复杂的表格之前，可以先在纸上画画它们，这样可以节省时间，并使自己免遭挫折。

跨越（spanning）是强制一个单元格横跨表格的多行或多列的过程。colspan 属性导致单元格横跨多列；rowspan 属性则对行产生相同的效果。

此外，在样式表中还定义了文本样式，并且应用于 Description 列中的单元格，创建既垂直对齐到单元格中间又水平对齐到单元格中心的粗体文本。

程序清单 9.3 中的几个技巧还没有解释。你可以给整个表格以及表格中的每个单独的行或单元格提供它自己的背景，并使之与你可能在 Web 页面自身上使用的任何背景区分开。通过在<table>、<tr>、<td>、<th>、<thead>、<tbody>或<tfooter>标签中放置 background-color 或 background-image 样式来执行该操作（在样式表中为这些元素赋值），与你将在<body>标签中所做的完全一样。在程序清单 9.3 中，只有最上面一行具有背景色；样式表将<thead>元素定义为在那一行中的单元格中具有红色的背景和白色的文本。

与 background-color 样式属性类似的是 background-image 属性（这个示例中未显示），它用于为表格背景设置一幅图像。如果你想把图像 leaves.gif 设置为表格的背景，将在用于<table>元素的样式表条目中使用 background-image:url("leaves.gif")。注意，图像文件放在圆括号内，并在前面放置单词“url”，它指示你正在描述图像文件位于何处。

**By the Way**

**注意：**你通常会在表格中看到交替的行颜色。例如，其中一行可能具有灰色的背景，下一行可能具有白色的背景。交替的行颜色有助于用户更清楚地阅读表格的内容，尤其是当表格比较大时。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

调整表格不仅限于使用样式属性。如程序清单 9.3 所示，可以通过对单元格应用某种填充，在其边框内控制单元格内容周围的空白。如果想在单元格本身的边框之间添加一些空白，可以使用 CSS border-spacing 属性，这允许定义水平和垂直间距，如下所示：

```
border-spacing: 2px 4px;
```

在示例中，水平边框之间的间距被定义 2 像素的空白，垂直边框之间则定义为 4 像素的空白。如果只使用一个值，则该值将应用于全部 4 条边框。

## 9.4 利用表格进行页面布局

在本章开头，我指示设计师把表格用于页面布局，以及显示表格式信息。如果你窥视另一位设计师的源代码，仍将发现许多基于表格的布局示例。这种设计方法源于在 20 世纪 90 年代中期到 21 世纪早期在浏览器中对 CSS 的支持不一致。由于所有的浏览器都支持表格，并且一般以相同的方式提供这种支持，因此 Web 设计师对基于表格的内容布局方法不离不弃，以跨所有的浏览器实现相同的可视化页面显示。不过，既然现在跨所有主流的浏览器都对 CSS 提供了相对类似的支持，设计师就可以遵循长期的基于标准的建议规范，而不是将表格用于页面布局。

万维网联盟（World Wide Web Consortium，W3C）——监管 Web 未来的标准组织——长期提倡将样式表（而不是使用表格）作为设计页面布局的正确方式。样式表最终将比表格强大得多，这就是为什么本书用大量篇幅介绍如何将样式表用于页面布局的原因。

避免将表格用于布局的主要原因如下所示。

- 表格将表示与内容混合在一起：CSS 和符合标准的 Web 设计的一个目标是把表示层与内容层分隔开。
- 表格会导致徒劳无益的、困难的重新设计：要改变基于表格的布局，必须改变站点的每个页面上的基于表格的布局（除非它是一个复杂的、动态驱动的站点的一部分，在这种情况下，必须撤消所有的动态部分并重新创建它们）。
- 解决可访问性问题：屏幕阅读软件将期望内容的表格，并且通常尝试将布局表格作为内容表格来阅读。
- 在移动设备上呈现：表格布局通常不够灵活，不能下缩到较小的屏幕上（参见第 15 章）。

这些只是基于表格的 Web 设计中的少数几个问题。要深入了解其中一些问题，可以参见 http://www.hotdesign.com/seybold/everything.html 上的受欢迎的演示文稿“Why Tables for Layout is Stupid”（为什么把表格用于布局是愚蠢的）。

## 9.5 使用 CSS 栏

如果你具有大量只包含文本的信息，就可能希望像实物报纸那样展示它：分栏展示。一百多年的研究显示行的长度与阅读速度之间具有相关性，并且表明行的“甜蜜点”或最佳长度能够让你获得快速且令人愉悦的阅读体验。这个“甜蜜点”仍然存在（行的长度大约为

异步社区会员 Kensuke(18319066421) 专享 尊重版权

4 英寸），这就是实物报纸仍然分栏展示信息的原因。

如果有许多信息要展示给读者，或者如果只是想简单地模仿报纸布局的美感，可以使用 CSS 栏。确实，也可以使用表格，因为表格是由行和列组成的，但是上一节解释了避免使用基于表格的布局的一些原因。此外，栏不仅仅用于文本，可以把想要的任何内容放在定义的栏中，比如把广告或相关的文本放在侧栏中。

在图 9.7 中可以看到 CSS 栏的基本应用，用于定义传统的报纸类型的布局。

图 9.7

程序清单 9.4 中的代码显示了一个 3 栏式布局




### 程序清单 9.4 使用 CSS 栏创建一个 3 栏式布局

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Breaking News!</title>
    <style type="text/css">
    article {
       -webkit-column-count: 3;
       -webkit-column-gap: 21px;

       -moz-column-count: 3;
       -moz-column-gap: 21px;
       column-count: 3;
       column-gap: 21px;
    }

    h1 {
       text-align: center;
       -webkit-column-span: all;
       -moz-column-span: all;
       column-span: all;
    }

    p {
       margin-top: 0px;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      margin-bottom: 12px;
    }

    footer {
       -webkit-column-span: all;
       -moz-column-span: all;
       column-span: all;
    }
    </style>
  </head>

  <body>
    <article>

      <header>
        <h1>Breaking News!</h1>
      </header>

        <p>Breaking news, also known as a special report or news bulletin,
        is a current issue that broadcasters feel warrants the interruption
        of scheduled programming and/or current news in order to report its
        details. Its use is also assigned to the most significant story of
        the moment or a story that is being covered live. It could be a
        story that is simply of wide interest to viewers and has little
        impact otherwise. Many times, breaking news is used after the news
        network has already reported on this story. When a story has not
        been reported on previously, the graphic and phrase "Just In" is
        sometimes used instead.</p>

        <p>In early coverage of a breaking story, details are commonly
        sketchy, usually due to the limited information available at the
        time. For example, during the Sago Mine disaster, initial reports
        were that all twelve miners were found alive, but news
        organizations later found only one actually survived.</p>

        <p>Another criticism has been the diluting of the importance of
        breaking news by the need of 24-hour news channels to fill time,
        applying the title to soft news stories of questionable importance
        and urgency, for example car chases. Others question whether the
        use of the term is excessive, citing occasions when the term is
        used even though scheduled programming is not interrupted.</p>
      <footer>
        <em>Text courtesy of Wikipedia:
        http://en.wikipedia.org/wiki/Breaking_news</em>
      </footer>

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

程序清单 9.4 中的代码来自于一篇伪造的新闻文章，因此我使用了<article>元素来存放所有的内容。在<article>元素内，是一个<header>元素，其中包含“Breaking News!”标题（在<h1>级别），其中接着 3 个文本段落和一个<footer>元素。所有的样式都是在程序清单开头的样式表中处理的，并为刚才提到的 4 个元素（<article>、<h1>、<p>和<footer>）提供了样式。

在样式表中，我在<article>元素内应用了栏的主要定义。我使用 column-count 定义了 3 栏，并且使用 column-gap 把栏之间的空白定义为 21 像素宽。接下来，我为<h1>元素添加了

异步社区会员 Kensuke(18319066421) 专享 尊重版权

一个定义，首先使文本在页面的中间对齐，其次确保文本跨越所有栏。出于同样的原因，我对用于<footer>元素的条目应用了相同的 column-span 属性。

在用于<h1>元素的条目之后，我给<p>元素添加了一些具体的边距，即上边距为 0 像素，下边距为 12 像素。我可能留出了足够的空白，并且只允许<p>元素显示为默认样式，但是这将在每个段落顶部创建一个边距。你可能会问：“这有什么大不了的？”因为无论怎样，它看起来就像我在段落之间手动添加了一些空白，并且也确实如此。不过，我在每个段落后面都添加了空白，并且去掉了每个段落前面的空白，使得第一栏不是开始于一块空白，从而不会错误地对齐 3 个栏的顶部。

**By the Way**

**注意：**在样式表中，要注意的是，对于样式表中每个与栏相关的属性，我都添加了另外两外条目，它们具有-webkit 和-moz 前缀。由于浏览器对与栏相关的 CSS 的支持仍然有一点不一致，可以使用具有这些前缀的属性来确保属性将在所有的浏览器中工作。

甚至还可以在栏之间添加垂直线条，如图 9.8 所示。下面显示了我添加的用于实现这种外观效果的样式表条目，注意它们看上去非常类似于你定义边框的方式（如你在第 7 章中所学到的）。

```
-webkit-column-rule-width: 1px;
-moz-column-rule-width: 1px;
column-rule-width: 1px;
-webkit-column-rule-style: solid;
-moz-column-rule-style: solid;
column-rule-style: solid;
-webkit-column-rule-color: #000;
-moz-column-rule-color: #000;
column-rule-color: #000;
```

图 9.8

程序清单 9.4 中的代码，并且在栏之间添加了垂直线条




异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 9.6 小结

本章篇幅较短，主要介绍了把文本和图像安排进称为表格的有组织的行和列中，介绍了用于创建表格的基本标签，以及几个用于控制表格的对齐、间距和外观的 CSS 属性。同时，本章还介绍了永远都不要把表格用于布局目的，但是可以使用 CSS 栏实现多栏布局。

表 9.1 总结了本章中介绍的标签。

**表 9.1　　本章介绍的 HTML 标签**

| 标签 | 作用 |
|---|---|
| <table>...</table> | 创建一个可以包含任意数量的行和列的表格 |
| <thead>...</thead> | 定义表格的标题行 |
| <tbody>...</tbody> | 定义表格的正文行 |
| <tfoot>...</tfoot> | 定义表格的脚注行 |
| <tr>…</tr> | 定义包含一个或多个单元格（<td>标签）的表格行 |
| <th>...</th> | 定义表格的标题单元格（接受与<td>相同的所有样式） |
| <td>...</td> | 定义表格的数据单元格 |

## 9.7 问与答

**问：我制作了一张大表格，并且当我加载页面时，页面上长时间都没有任何事情发生。为什么要等待？**

**答：**复杂的表格可能要花相当长的时间才能出现在屏幕上。Web 浏览器在可以显示表格的任何部分之前，必须查明表格中的任何内容的大小。可以通过总是为表格内的每一幅图形图像包括 width 和 height 属性，把显示速度加快一点。为<table>和<td>元素定义具体的宽度也是有用的。

**问：我能够使用 CSS 栏创建出超越“报纸”外观的布局吗？**

**答：**是的，如果你有这种期望，就可以使用 CSS 栏为整个 Web 站点创建一种基于栏的布局。例如，一栏用于导航，下一栏比较宽，用于正文文本，接下来是第三栏，用于广告或其他辅助内容，这是一种常见的模板设计。

## 9.8 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

### 9.8.1 问题

1. 怎样才能创建一个简单的两行、两列的表格，并利用 1 像素的黑色边框框住这个表格？
2. 扩展问题 1，怎样才能在每个单元格内的内容周围添加 9 像素的空白？
3. 怎样使元素的内容跨越多列？

### 9.8.2 答案

1. 使用以下 HTML 代码：

```
<table style="border: 1px solid #000000; border-collapse:
collapse;">
  <tr>
    <td>Top left...</td>
    <td>Top right...</td>
  </tr>
  <tr>
    <td>Bottom left...</td>
    <td>Bottom right...</td>
  </tr>
</table>
```

2. 在样式声明中添加 padding: 9px。
3. 对元素应用 column-span 属性，并将其值设置为 all。

### 9.8.3 练习

- 你是否具有任何页面，访问者可能感兴趣于将其中的信息作为列表或表格来查看？使用表格来展示一些表格式信息。确保每一列都具有它自己的标题（或者也许是它自己的图形）。试验你在本章中学过的不同类型的对齐方式和间距。
- 你经常会在表格中看到交替的行颜色，其中一行具有灰色背景，下一行则具有白色背景。表格行中的交替颜色的目标是在快速查看填满数据的表格时，使得更容易分辨单独的行。创建一个具有交替的行颜色和文本颜色（如果必要的话）的表格。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 10 章

# 创建用于 Web 的图像

在本章中你将学到：

- 怎样选择要使用的软件包；
- 怎样准备在线使用的照片；
- 怎样创建横幅和按钮；
- 怎样减少图像中的颜色数量；
- 怎样创建透明的图像；
- 怎样为拼贴式背景准备图像；
- 怎样创建动画式 Web 图形。

相信我，你的职业不必是一位艺术家（尽管这确实会有帮助），也能在 Web 页面上放置具有高度感染力的图形。你不必在软件包上花费数百或数千美元，只是为了处理数码照片或者你可能想使用的其他源图形。本章的主题将帮助你开始创建可以在 Web 站点中使用的图像。

尽管本章中的示例图形使用的是针对 Windows、Mac 和 Linux 用户的流行且免费的图形程序（GNU Image Manipulation Program（GNU 图像处理程序）或 GIMP），你也可以将在本章中学到的知识应用于任何主要的 Windows 或 Macintosh 图形应用程序，尽管这些软件的菜单和选项看上去有所不同。

在你学会了创建图形本身之后，将准备好使用 HTML、CSS 或者它们二者在 Web 站点中包括它们。

## 10.1 选择图形软件

你可以使用几乎所有的图形程序为 Web 站点创建和编辑图像，从通常由计算机的操作

异步社区会员 Kensuke(18319066421) 专享 尊重版权

系统免费提供的简单的绘图或绘画程序，到昂贵的专业程序，比如 Adobe Photoshop。类似地，如果计算机连接有数码相机或扫描仪，它们可能带有一些能够创建适合于在线使用的图像的图形软件。还有几款免费的图像编辑器可供下载，或者甚至作为 Web 应用程序在线使用，并且它们正好可用于处理照片元素。

如果你已经具有自认为可能适合于创建 Web 图形的软件，就可以尝试使用它来做下面几节中描述的一切事情。如果你的软件不能处理这里介绍的一些任务，它可能就不是一种适合于 Web 图形的良好工具。在这种情况下，就要从 www.gimp.org 下载并安装 GIMP。这是一个全功能的图形程序，完全免费，并且肯定可以执行本章中展示的各类动作。

**By the Way**

**注意：**毋庸置疑，在谈到图像编辑程序时，Adobe Photoshop 是出类拔萃的。不过，它比较昂贵，并且如果你没有处理计算机图形的经验，那么它还相当复杂。有关 Adobe 产品的更多信息，可以访问 Adobe Web 站点（http://www.adobe.com/）。如果你有意购买该公司的某种产品，可以先从这个站点下载免费评估版。

### 使用在别处找到的图像

要想节省为 Web 页面创建图形和媒体文件的时间，最佳方式之一当然是避免完全创建它们。可以非常轻松地获取任何 Web 页面中的图形，只需右击它（或者按住 Option 键并单击 Apple 鼠标），并选择 Save Image As（图像另存为）或 Save Picture As（图片另存为）（依赖于浏览器）。从页面中提取背景图像同样非常容易，右击它并选择 Save Background As（背景另存为）即可。

不过，如果没有得到所有者的明确许可（通过询问或者寻找 Creative Commons 许可证），那么永远也不应该使用这样的图像。在没有得到明确许可的情况下使用这些图像是一种侵犯版权的行为（也是令人反感的）。

你还可能希望考虑无需为专利权付费的剪贴画，它不需要获得版权许可。在线剪贴画的良好来源是 Microsoft 的 Office 网上剪辑和媒体 Web 站点，其地址为 http://office.microsoft.com/clipart/。Clipart.com 是另一个受欢迎的剪贴画目的地，只需支付很少的费用，就能访问数千幅储备的图像。

如果 GIMP 不适合你，可以考虑下载 Adobe Photoshop 或 Corel DRAW 的评估版。仅仅出于图片处理的目的，可以有许多选项，它们都具有有益的特性。Google 的 Picasa（可以在 http://picasa.google.com/上免费获得）就是这样一个选项，Pixlr（pixlr.com）是另一个良好的选项。这两个程序都适合于编辑图像，而不是从头开始创建它们，并且 Picasa 还适合于组织数码照片集。这些类型的程序不一定可以帮助你为站点设计横幅或按钮图像，但是它们有助于处理一些辅助图像，并且它们的功能足够强大，值得你查看。

## 10.2 关于图形需要知道的最少知识

当涉及在 Internet 上发布图形和多媒体时，有两股力量意见不一致。用户的眼睛和耳朵希望你的所有内容都尽可能详细和准确，并且他们还希望信息会立即显示。一个难以解决的

异步社区会员 Kensuke(18319066421) 专享 尊重版权

问题是，彩色图形意味着较大的文件尺寸，这会增加传输时间，甚至在快速连接上也是如此。怎样才能最大化表示的质量同时又最小化文件尺寸？为了做出这些选择，需要理解颜色和分辨率如何协作，产生主观的质量感觉。

图像的分辨率是构成图像的单个点或像素的数量（通常为 72 点/英寸或 72dpi）。与较小的低分辨率图像相比，较大的高分辨率图像一般要花较长的时间进行传输和显示。图像尺寸通常被指定为图像的宽度乘以图像的高度，用像素表示。例如，300×200 的图像是指图像的宽度为 300 像素，高度为 200 像素。

分辨率并不是决定图像文件的存储尺寸（和传输时间）的最重要的因素，当你发现这一点时可能会感到奇怪。这是由于 Web 页面上使用的图像总是以压缩的形式存储和传输的。图像压缩是对图像进行的数学处理，以挤压出重复的模式。图像压缩的数学知识很复杂，但是基本的思想是：当在磁盘上存储图像时，可以挤压出重复的模式或者较大的同色区域。这使图像文件要小得多，并且允许在 Internet 上更快地传输。以后，在显示图像时，Web 浏览器将恢复图像的原始外观。

**By the Way**

**注意：**目前在使用多种类型的图像分辨率，包括像素、空间、光谱、时间和辐射。仅仅为了学习每种类型的分辨率，你可能就会花几个小时的时间。并且如果你是在学习图形设计课程，就可能会这样做。不过，目前，你只需记住较大的图像要花更长的时间下载，还会使用许多显示空间。显示尺寸和存储或传输尺寸是在设计 Web 站点时所要考虑的因素。

在下面几节中，你将学习如何创建具有较强的视觉冲击力但是文件尺寸又比较小的图形。用于完成这项任务的技术依赖于每一幅图像的内容和用途。Web 图形的使用像 Web 页面一样多，但是有 4 类图形是迄今为止最常见的。

- 人、产品和地理位置的照片。
- 图形横幅和标志。
- 用于指示动作和提供链接的按钮或图标。
- 用于容器元素的背景纹理。

## 10.3 准备照片图像

要把照片放在 Web 页面上，需要把基于打印的照片转换成数字图像，或者使用数码相机以数字方式创建照片，这包括智能手机中无处不在的相机。对于一些型号较老的硬件，可能需要使用设备自带的自定义软件把图像传输到硬盘驱动器上，但是在大多数情况下，应该能够连接设备，然后把文件拖放到硬盘驱动器上。如果使用扫描仪创建打印照片的数字版本，就可以直接从所选的图形程序中控制几乎所有的扫描仪，关于这一点，你可以参见软件文档，了解详细信息。

**Did you Know?**

**提示：**如果你没有扫描仪或数码相机，那么几乎所有的胶片开发商都提供了一种服务，收取适度的费用以把照片从 35mm 的胶片上传输到 CD-ROM 或 DVD-ROM 上。然后你就可以把文件复制到硬盘驱动器上，并且使用图形程序打开和修改图像文件。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

在把数字图像文件传输到计算机上之后，就可以使用图形程序对它们进行裁剪、调整大小、校色和压缩，以使它们准备好在 Web 站点中使用。

## 10.3.1 裁剪图像

由于你希望 Web 页面上的图形尽可能简洁，因此通常需要裁剪数码照片。在裁剪照片时，可以选择你想显示的区域，并把其余的裁剪掉。

**Try It Yourself：在 GIMP 中裁剪图像**

GIMP 工具箱允许快速访问裁剪工具及其可能的属性。找到一个图像文件，可以是你利用相机拍摄并存储在硬盘驱动器上的数字图像，或者是你在网上找到的图像。在 GIMP 中打开图像之后，可以执行以下步骤，在 GIMP 中裁剪它。

1. 如图 10.1 所示，在 GIMP 工具箱中单击 Crop（裁剪）工具。依赖于你所选的工具，可能具有额外的属性可以选择。例如，图 10.1 显示了用于裁剪工具的属性（如宽高比、位置、大小等）。

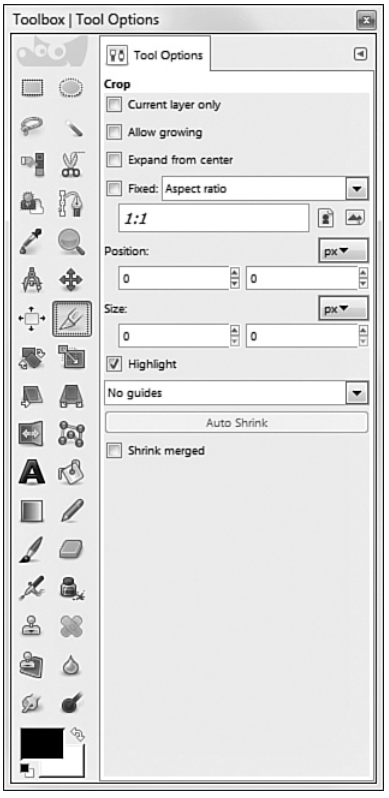



图 10.1
从工具箱中选择 Crop 工具

2. 在你想裁剪的图像中，单击你想保留的图像部分的左上角，并按住鼠标左键，向下拖动到右下角，绘制一个方框包围所选的内容，如图 10.2 所示。
3. 单击所选内容的其中一个角，应用裁剪。

   你的图形程序很可能具有一种不同于这里所显示的方法，但是概念是相同的：选取要保留的区域，然后裁剪掉其余的部分。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

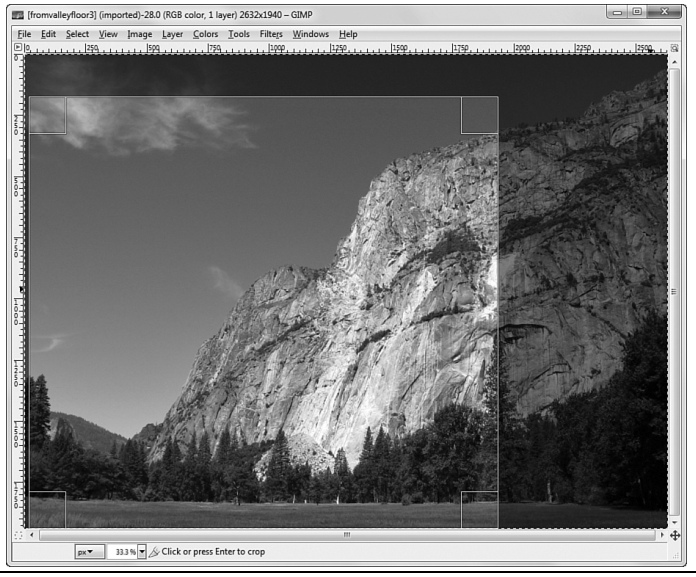



图 10.2

选取你想显示的图像区域

即使在裁剪了图像之后，它仍然可能比用于 Web 页面所需的尺寸要大一些。依赖于特定 Web 页面的设计，你可能想限制大图像不超过 800×600 像素（如果只在页面上单独显示它的话，比如产品目录），或者甚至是 640×480 像素或更小。当在文本旁边显示时，图像的宽度倾向于在 250~350 像素之间，以便为文本也留出足够的空间。在一些情况下，你可能还想提供图像的缩略图版本，以链接到更大的版本。这样，你将可能坚持缩略图的较大尺寸接近于 100 像素。

**提示：**你的图形软件很可能在图像窗口自身内的某个位置显示出随时可见的图像大小。在 GIMP 中，可以在窗口的标题栏中看到当前的图像大小。其他程序可能在右下角或左下角显示它。你还可能会在窗口中看到放大比率，并且能够通过缩放来改变它。

**Did you Know?**

## 10.3.2 调整图像大小

更改图像大小所需的准确工具依赖于使用的程序。在 GIMP 中，可以进入 Image 菜单，并且单击 Scale Image，打开 Scale Image 对话框，如图 10.3 所示。

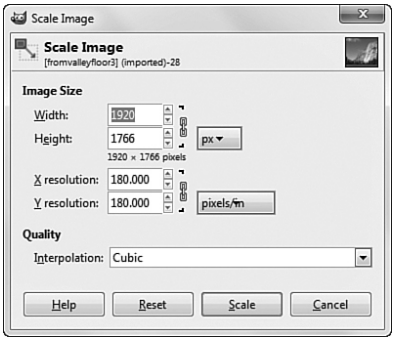



图 10.3

使用 Scale Image 对话框更改图像的大小

异步社区会员 Kensuke(18319066421) 专享 尊重版权

你几乎总是希望使用现有的宽高比调整大小，这意味着当你输入想要的图像宽度时，将会自动计算高度（反之亦然），以防止图像被挤压变形。在 GIMP 中，宽高比默认是锁定的，如图 10.3 所示的 Width 和 Height 选项旁边显示的链环所指示的那样。单击链环一次，可解除锁定它，使你能够指定自己选择的像素宽度和高度，以及是否会被挤压。

**By the Way**

**注意：**与 GIMP 中的许多特性一样，Scale Image 对话框出现在包含要调整大小的图像的窗口前面。这种布置使你能够在对话框中执行更改，应用它们，并且立即查看结果。

在大多数（如果不是全部的话）图形程序中，还可以基于百分比调整图像大小，而不用提供具体的像素尺寸。例如，如果我的图像开始时的大小是 1815×1721 像素，并且我不想执行数学运算，以确定将其显示为那个宽度的一半所需要的值，我可以简单地选择 Percent（在这个实例中，可以从图 10.3 中显示的像素旁边的下拉列表中选择它），并且把默认设置（100）改为 50。然后，图像将变成 908（宽度）×861（高度）像素，其中不需要进行任何数学运算。

## 10.3.3 调整图像颜色

如果你是在编辑照片图像，而不是创建你自己的图形，就可能需要使用一些校色工具，以获得最合适的照片。就像在许多图像编辑程序中一样，GIMP 提供了几个选项，用于调整图像的亮度、对比度和颜色平衡，还提供了一个滤镜，用于减少令人讨厌的红眼。为了使用 GIMP 删除红眼，可以进入 Filters，单击 Enhance，然后单击 Red Eye Removal。

其中大多数选项都非常直观。如果你希望图像变得更亮一些，可以调整亮度。如果希望图像显得更红一些，可以调整颜色平衡。在 GIMP 中，Colors 菜单提供了众多工具可供访问。与上一节中描述过的 Scale Image 对话框一样，每个工具都会在工作区的前台显示一个对话框。在调整颜色时，图像将会反映所做的更改。这种预览功能是大多数图像编辑软件中都会包括的一个特性。

如图 10.4 所示，Adjust Hue/Lightness/Saturation 工具是 Colors 菜单上提供的许多工具之一。可以使用对话框中的多个不同的滑块调整你正在处理的值，来实现许多与颜色相关的改变。Preview 特性使你能够在做一件事时查看自己正在做什么。Reset Color 按钮将把图像恢复到它的原始状态，而不会应用任何更改。

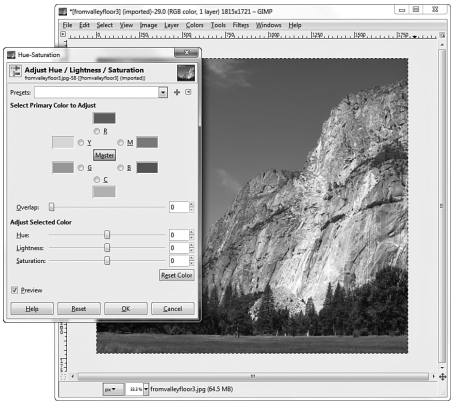



图 10.4

Adjust Hue/Lightness/Saturation 工具是 GIMP 中提供的许多基于滑块的颜色修改工具之一

异步社区会员 Kensuke(18319066421) 专享 尊重版权

由于众多工具可供使用，并且每个工具都具有预览功能，简单地试验一下是查明每个工具可以做什么的最佳方式。

### 10.3.4 控制 JPEG 压缩

对于 Web 上的照片图像，当以 JPEG 文件格式（而不是 GIF 格式）保存时，它们将工作得最好。JPEG 使你能够保留文件中的颜色数量，同时仍然使文件的总体大小保持在一个可管理的水平。在调整完照片的大小和外观时，可以选择 File>Export 命令，并选择 JPEG 作为文件类型。你的图形程序很可能提供另一个对话框，用于控制多个 JPEG 选项，比如压缩。

如图 10.5 所示，在 GIMP 中导出一个 JPEG 文件时将会看到 Export Image as JPEG 对话框。在该对话框中可以看到，在保存 JPEG 文件时，可以通过在 1（质量较低、文件较小）~100（质量较高、文件较大）之间调整 Quality 滑块，来控制压缩比。

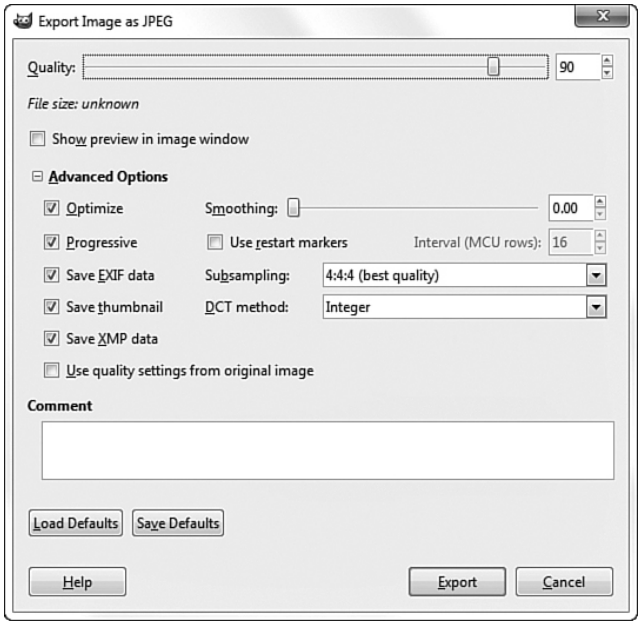



图 10.5

GIMP 以 JPEG 格式保存文件后，能够减小文件大小，同时仍然保留图像质量

你可能希望试验一下，查看多种不同的 JPEG 压缩级别如何影响图像的质量，但是对于大多数照片图像来说，85%的质量（或者 15%的压缩）一般是文件大小（因此也就是下载速度）与质量之间的良好折衷。

## 10.4 创建横幅和按钮

对于从头创建的图形（比如横幅和按钮），将需要你做出与众不同的考虑，它们与针对照片所做的考虑不同。

在制作横幅或按钮时，需要做出的第一个决定是它应该有多大。大多数接受 Web 的人现在都具有一台计算机，其屏幕分辨率至少是 1024×768 像素。例如，我的屏幕目前设置为 1440×900 像素，1366×768 像素是另一种流行的分辨率。一个看起来不错的尝试是：计划图形，使得它们总是能够适应较小的屏幕（1024×768 像素），并且还有用于滚动条和边距的多

异步社区会员 Kensuke(18319066421) 专享 尊重版权

余空间。不过，现如今，如果你是为台式机设计页面，那么把最小宽度增加几个像素将不会遇到很大的麻烦。

**Did you Know?**

**提示：**多年来，为 800×600 像素的屏幕分辨率设计页面是一种标准的规范。现在仍然要谨记这个较低的数字，因为许多人不会在全屏幕模式下打开应用程序。不过，以 1024×768 像素的屏幕分辨率作为设计的底线也是一个合理的设计。

假定你把 1024×768 像素的最低分辨率作为目标，那么完全尺寸的横幅和标题图形的宽度应该不超过 900 像素，这一般是在考虑了滚动条、工具栏及浏览器窗口的其他部分之后的可供查看的页面宽度。在页面内，正常的照片及其他图像的每个尺寸应该是 100~300 像素，较小的按钮和图标的高度和宽度应该是 20~100 像素。多年来，为横幅图形及其他广告大小的图像使用了一些被普遍接受的尺寸。如果查看可供会员组织和合作伙伴程序使用的横幅广告，就会看到这些尺寸的使用，比如 Amazon Affiliate 程序（参见 https://affiliate-program.amazon.com/）。公共尺寸包括：120×600 像素、234×60 像素和 300×250 像素，等等。

要在 GIMP 中创建一幅新图像，可以进入 File 菜单，并选择 New 命令，出现一个 Create a New Image 对话框，如图 10.6 所示。如果不确信图像需要有多大，只需接受 640×480 像素的默认大小即可。或者可以在 Template 下拉列表中选择其他预先确定的尺寸之一，比如“Web Banner Common 468×60”或“Web Banner Huge 728×90”。这两个设置都比较保守，然而完全可接受，它们很好地解释了“常见”和“巨大”的横幅。在这个对话框中，还可以为新图像输入你自己的宽度和高度。

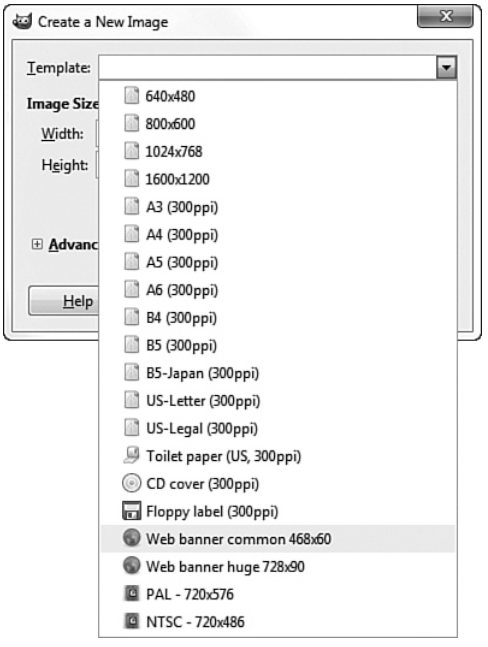



图 10.6

在开始处理图像之前，必须先确定它的大小

对于图像的背景色，通常应该选择白色，以匹配大多数 Web 浏览器用于 Web 页面的背景（尽管如你在上一章中所学到的，可以改变这个颜色）。当你知道自己将创建的页面背景不是白色时，就可以为图像选择一种不同的背景色。或者你可能想创建一幅根本没有背景的

异步社区会员 Kensuke(18319066421) 专享 尊重版权

图像，在这种情况下，可以选择 Transparency 作为背景色。当最终的、准备好用于 Web 的图像包括一种透明的背景时，将允许图像后面的 Web 页面（及其背景色）显示出来。在 GIMP 中，可以在 Create a New Image 对话框中打开 Advanced 选项，为新图像选择背景色。

在输入了图像的宽度和高度（以像素为单位）并且单击 OK 按钮之后，将会看到一张黑色的画布。如果你像我们大多数人一样患有艺术恐惧症，那么它看上去将令人生畏！不过，可以使用如此多的图像创建教程（更不用说整本书了），引导你通过这个过程，并让你挥洒自己的创意。本节全都用于介绍当创建在站点中使用的图形时需要记住的事情，它不一定会准确地教你怎么做，因为熟悉你所选的工具是驾驭它们的第一步。

## 10.5 减少或删除图像中的颜色

减少图像尺寸（及其下载时间）的最有效的方式之一是减少图像中使用的颜色数量。这可能会显著降低一些照片图像的视觉质量，但是它非常适合于大多数横幅、按钮及其他图标。

你将很高兴地知道有一种特殊的文件格式用于颜色数量有限的图像，它就是 GIF（Graphics Interchange Format，图形交换格式）。当把图像保存或导出为 GIF 时，可能会提示你拼合图标，或者通过把它转换成索引图像（indexed image）来减少颜色数量，因为这些是 GIF 的要求，检查你的软件中关于图层和索引颜色的帮助文件，彻底了解你可能需要做什么。

记住，GIF 图像格式设计用于包含纯色区域的图像，比如 Web 页面标题及其他演示图形；GIF 格式并不适合于照片（可代之以使用 JPG 或 PNG 文件）。

PNG（读作“ping”）是所有主流的 Web 浏览器中都支持的一种有用的文件格式。GIF 图像格式允许指定单独一种透明色，这意味着 Web 页面的背景将透过图像的那些区域显示出来，而 PNG 格式则更进一步，它允许指定不同的透明度。

你可能见过一些 Web 站点在它们的容器元素中使用背景色或图像，而且使图像存在于前景中，允许背景透过前景图形的某些部分显示出来。在这些情况下，前景中的图像具有一些透明的部分，使得图像本身（它们总是位于矩形画布上）不会显示出画布上设计还没有发生的区域。你通常希望使用这些类型的部分透明的图像，以使图形在你所具有的任何背景色或背景图像上看起来都不错。

要使图像的一部分成为透明的，必须以 GIF 或 PNG 文件格式保存图像。如本章前面所述，大多数支持 GIF 格式的图形程序都允许指定一种颜色是透明的，而 PNG 图像则允许指定一个透明度范围。一般来讲，由于这个透明度范围，PNG 格式要优于 GIF 格式。所有最新的 Web 浏览器都已经支持 PNG 图像。

创建透明图像的过程依赖于你创建的图像类型（GIF 或 PNG）以及你用于创建它的图形软件。要获得相关的指导，可以查看你的图形程序的帮助文件，或者在搜索引擎中输入“利用××软件创建透明的图像”。

## 10.6 创建拼贴式背景图像

可以使用任何 GIF、JPG 或 PNG 图像作为容器元素内的背景贴图，但是在你开始创建

异步社区会员 Kensuke(18319066421) 专享 尊重版权

拼贴式背景（尤其是高度图案化的拼贴式背景）之前，要问问自己拼贴式背景给 Web 站点的总体外观和感觉添加了什么效果。更重要的是，要问问自己当把站点的文本放在那种图案上时，它们是否容易阅读。

考虑你每天都频繁光顾的 Web 站点，并且考虑以下事实：很少有站点在它们的整个页面上使用拼贴式的、高度图案化的背景。如果限制自己只浏览公司、产品、体育队的 Web 站点或者其中的信息（主要是文本）享有特权的其他站点，那么具有拼贴式、高度图案化的背景的站点数量甚至会进一步减少。Web 能够承受每一个人都具有个性的设计，但是如果你是为自己的企业创建站点，就可能希望避免使用高度图案化的背景与颜色对比强烈的文本。

如果你为整个站点使用拼贴式背景图像，要记住当你不能辨别出它们是拼贴的图像时，它们将看起来效果最佳。换句话说，当背景贴图的顶部边缘与底部边缘无缝地匹配并且左边缘与右边缘也匹配时，你就知道自己具有良好的图像。

背景贴图的使用如图 10.7 和图 10.8 所示，它们二者都具有无缝的背景，但是所产生的效果有所不同。

图 10.7

这是一个无缝背景图像的示例，其中可以辨别出背景是拼贴而成的，因为可以看到 6 个完全相同的形状

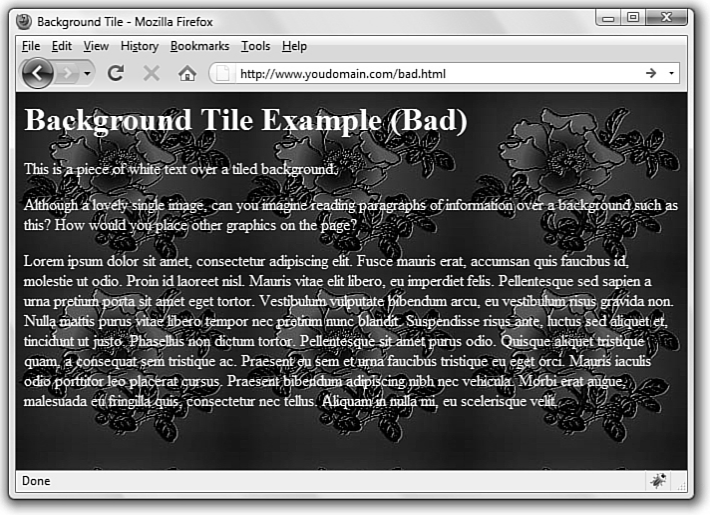
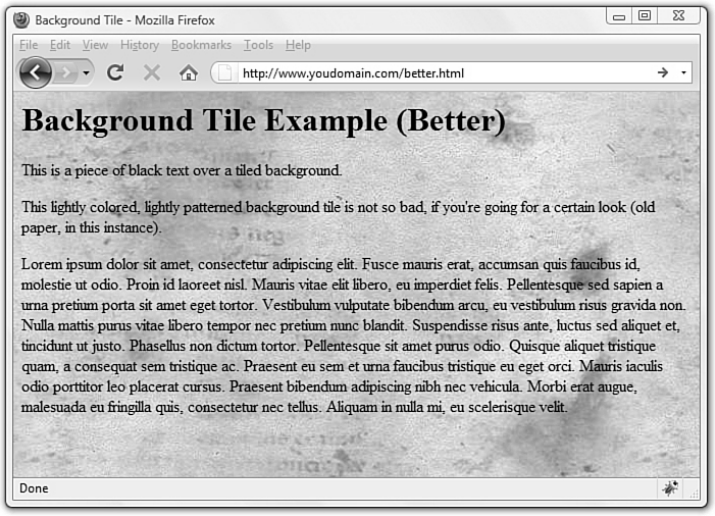



图 10.8

这也是一个无缝背景图像的示例，但是不能辨别它是拼贴而成的




异步社区会员 Kensuke(18319066421) 专享 尊重版权

在第 11 章中，你将学习如何在容器元素内放置背景图像。尽管我在本节中迄今为止给出了一些警告，但是背景图像仍然可能是你的设计武器库中的一件强大的武器，只要不是高度图案化的即可。在 Subtle Patterns Web 站点（http://subtlepatterns.com/）上可以找到背景图像的一些优秀（并且可免费使用）的示例，通常称之为纹理（texture）。我还没有展示一个良好、细微的拼贴式背景的示例，因为它是一个良好设计的元素，导致它很可能是如此细微，以至于在印刷出来的图书中看不到它。

## 10.7 创建动画式 Web 图形

GIF 图像格式使你能够创建动画式图像，它们添加了一些运动效果，从而给任何 Web 页面增添了一些情趣。与通常用于实现类似效果的视频或多媒体文件相比，动画式 GIF 图像的传输速度要快得多。利用 GIMP 软件包，可以通过在一幅图像内创建多个图层，然后在保存文件时修改 Animated GIF 选项，来创建动画式 GIF。此外，如果你具有一系列想要创建动画的图像，可以使用 Gickr（www.gickr.com）上的免费的、基于 Web 的 GIF 动画服务。

创建 GIF 动画的第一步是：创建一系列将一个接一个地显示的图像，或者创建一系列图层，这依赖于特定的软件程序。其中每一幅图像都称为帧（frame）。就像拼合电影或动画片一样，你在电影或电视屏幕上看到的图像和动画式图形中的图像都是由许多单独的帧组成的，它们在外观上具有细微的差别。当你记住了图形帧之后，把它们连接在一起的过程就相对比较简单——计划阶段是最困难的。花一些时间以故事板方式描绘帧，尤其是当你计划具有多个帧时。当你知道将如何把帧组合起来时，就可以使用本节前面提到的 Gickr 服务，或者阅读你的图形软件的文档，以学习把它们都串到一起的特定过程。

**Try It Yourself：准备在 Web 站点中使用的图像**

你现在应该已经准备好了两三幅图像，以便在你学习本章余下的内容时，可以尝试把它们放到你自己的页面上。如果你具有一些已经以 GIF、PNG 或 JPEG 格式保存起来的图像文件（文件名以.gif、.png 或.jpg 结尾），就可以使用它们。使用你在阅读上一节时创建的任何图形也是不错的。

像 Google 这样的搜索引擎通过把你引领到与你自己的主题相关的站点，可以变成图像的金矿。搜索引擎还可以帮助你发现许许多多的站点，它们专用于提供对有用的媒体集合进行免费和廉价的访问。同时不要忘记 Microsoft 的 Office 网上剪辑和媒体 Web 站点（http://office.microsoft. com/clipart/）上的大型剪贴画库。其他有价值的资源包括 Google Images（images.google.com）和 Flickr（www.flickr.com），使用 Creative Commons 许可证寻找图像，转载时注明出处即可免费使用。

## 10.8 小结

本章讲解了准备在 Web 页面上使用的图形的基本知识。这是一个复杂的主题，本章介绍的相关内容足可以激起你的兴趣。本章中的示例使用流行（且免费）的 GIMP 软件包，但是可以自由地使用最适合的图形软件。本章还介绍了如何裁剪、调整大小和调整图像颜色等

异步社区会员 Kensuke(18319066421) 专享 尊重版权

动作，以及不同的文件格式。在站点中包括图形时，必须谨记许多考虑事项，包括图形大小和分辨率，以及如何使用透明度、动画式 GIF 和拼贴式背景。

## 10.9 问与答

**问：我不能聘请一位图形艺术家设计我的页面吗？这样，我就不用自己学习所有这些材料了。**

**答：**这个问题回答起来比较困难——它来自于某个为一家 Web 开发和设计代理公司工作了很长时间的人。对于个人和小企业，代理公司并非总是最佳的解决方案。聘请图形设计师要花时间和金钱。此外，许多图形艺术家也不会制作出适合于 Web 的作品。确切地讲，他们是基于印刷的艺术家，而印刷世界与网上世界具有相当大的区别。此外，聘请图形设计师创建 Web 站点可能不会发挥出那位特定的图形设计师的长处。换句话说，设计师可能擅长设计 Web 站点的图形元素，但是他或她可能不是优秀的内容架构师，或者所做的工作可能不会超越使用 HTML 和 CSS。如果你的站点只是一个个人站点，你可能不想把钱花在完全外包的专业设计上。但是，如果你的站点打算宣传企业、产品、学校或其他方面，其中图像是取得成功的基本组成部分，就值得你在某个时间花费时间（和金钱）咨询一位 Web 设计专业人员。

**问：我创建了用于印刷在纸张上的图形，Web 页面图形有什么区别吗？**

**答：**是的。事实上，用于印刷图形的许多规则都在 Web 上保留了下来。Web 页面图形必须是低分辨率的，而印刷图形应该具有尽可能高的分辨率。在计算机屏幕上，白色会冲洗掉黑色；而在纸张上，黑色将混入白色中。此外，当只下载了一半图形时，有人可能会停止 Web 页面加载，而在查看印刷的图像时，这不是一个要考虑的事项。如果你完成了许多印刷图形设计，就要尽量避免陷入旧习惯中。

## 10.10 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 10.10.1 问题

1. 你有一张骏马的扫描图片，需要把它放到 Web 页面上。你应该使它有多大，应该以哪种文件格式保存它？
2. 当选择拼贴式背景图像时，要注意的一些问题是什么？
3. 如果具有一幅透明背景上的蓝色花朵的正方形图像，并且容器元素的背景色是灰色的，那么花朵图像是将作为正方形还是某种其他的形状出现在页面上？

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 10.10.2 答案

1. 依赖于图像对于页面有多重要，应该使之像 100×40 像素那样小，或者像 300×120 像素那样大。JPEG 格式（具有大约 85%的压缩）将是最佳的。当然，还可以提供一个缩略图链接，指向可以单独查看的大图像。
2. 你需要确保当贴图重复时，它的“缝隙”是不可见的，图案不会喧宾夺主，并且贴图中的颜色不会与页面上使用的其他任何图像或颜色形成强烈的对比。
3. 它将显示为花朵的形状，因为图像具有透明的背景。包含元素的灰色背景将显露出来。

## 10.10.3 练习

- 在开始为一个重要的企业站点设计图形之前，可以在一张纸上描绘你可能想要包括的图形元素的大致尺寸和形状。参阅本章开头的内容，了解关于典型的图形尺寸和屏幕分辨率的信息，以确保获得最合适的图形元素。
- 使用你所选的图形软件程序，创建几个简单的动画式图形，用于放置在 Web 站点上。即使只把几个帧组合在一起，也有机会实践把帧组合起来，以实现用于显示的最佳帧速率。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 11 章

# 在 Web 站点中使用图像

在本章中你将学到：

- 怎样把图像放在 Web 页面上；
- 怎样利用文本描述图像；
- 怎样指定图像的高度和宽度；
- 怎样对齐图像；
- 怎样使用背景图像；
- 怎样使用图像映射。

上一章学习了为 Web 站点创建合适图像的一些基本知识，本章将讲解如何借助 HTML 和 CSS 在 Web 站点中使用这些文件。你不仅将学习使用 HTML<img/>标签包括图像以便在 Web 浏览器中显示的基本知识，还将学习如何提供这些图像的描述（以及使用它们的原因）。然后，将学习图像放置，包括如何使用图像作为不同元素的背景。最后，你将学习如何使用图像映射，它允许使用单独一幅图像作为指向多个位置的链接。

## 11.1 在 Web 页面上放置图像

要开始在 Web 站点上放置图像，首先要把图像文件移到与 HTML 文件相同的文件夹中，或者移到名为 images 的目录中，通常使用它以便于组织站点内容。

在这第一个示例中，让我们假定把一个名为 myimage.gif 的图像放在与你想用于显示它的 HTML 文件相同的目录中。要显示该图像，可以在想显示图像的文本中的那个位置插入以下 HTML 标签，并且使用图像文件的名称替换 myimage.gif：

```
<imgsrc="myimage.gif" alt="My Image" />
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

如果图像文件位于文档根目录下的 images 目录中，则将使用以下代码，可以看到它现在包含 images 目录中的 myimage.gif 文件的完整路径：

```
<img src="/images/myimage.gif" alt="My Image" />
```

<img />标签的 src 和 alt 属性是有效的 HTML Web 页面所需要的。src 属性标识图像文件，alt 属性则使你能够指定关于图像的描述性文本。如果用户由于图像不可访问或者由于用户正在使用纯文本浏览器或屏幕阅读器，alt 属性就打算用于替换图像。在后面的“利用文本描述图像”一节中将阅读到关于 alt 属性的更多知识。

**注意：**把图像放在什么位置与 Web 服务器、Web 浏览器或最终用户无关，只要你知道它们在什么位置并且在 HTML 代码中使用正确的路径即可。就我个人而言，我更喜欢把所有的图像都放在单独的 images 目录中，或者放在通用的 assets 目录的一个子目录（如 assets/images）中，使得所有的图像或其他资源（如多媒体和 JavaScript 文件）都组织有序。

**By the Way**

作为一个说明如何使用<img />标签的示例，程序清单 11.1 在页面顶部的文本段落之前插入了一幅图像。无论何时 Web 浏览器显示程序清单 8.2 中的 HTML 文件，它都会自动检索并显示图像文件，如图 11.1 所示。




图 11.1

当 Web 浏览器显示程序清单 11.1 中显示的 HTML 代码时，它将呈现 hd.jpg 图像

**注意：**<img />标签是也支持 title 属性的 HTML 标签之一，也可以使用这个属性来描述图像，与 alt 属性非常相似。不过，title 属性有些问题，这是由于在不同的用户代理之间将以不一致的方式显示它，因此是不可信赖的。你可能会看到 title 属性的使用，并且在使用它时 HTML 是有效的，但是请不要使用它代替 alt 属性。如果这样做，将会限制站点在许多类型的设备上的有用性。

**By the Way**

异步社区会员 Kensuke(18319066421) 专享 尊重版权

程序清单 11.1 使用<img />标签把图像放置在 Web 页面上

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <title>A Spectacular Yosemite View</title>
 </head>

 <body>
  <section>
    <header>
      <h1>A Spectacular Yosemite View</h1>
    </header>
    <img src="hd.jpg" alt="Half Dome" />
    <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>
  </section>
 </body>
</html>
```

**By the Way**

**注意：**可以在你自己的页面内包括任何 Web 站点中的图像。在这些情况下，无论何时显示你的页面，都将从另一个页面的 Web 服务器中检索图像。你可以但是不应该这样做！这不仅是一种糟糕的方式，因为你将使用另一个人的带宽来使自己获益，而且会使你的页面显示得更慢。此外，当图像可能被更改或删除时，你将无法进行控制。

如果授权你再次发布另一个 Web 页面中的图像，那么总是要把该图像的一个副本传输到你的计算机上，并且使用本地文件引用，比如使用<imgsrc="myimage.jpg"/>代替<imgsrc="http://www.otherserver.com/theirimage.jpg" />。

不过，当把你的图像（比如照片）托管给明确打算作为图像库的服务（比如 Flickr）时，这个建议将不适用。诸如 Flickr 之类的服务将为每一幅图像提供一个 URL，并且每个 URL 都会在地址中包括 Flickr 的域。如果你想链接到利用移动应用程序（比如 Instagram 或 Streamzoo，这里只指出了其中的两个）拍摄的图像，情况也是如此——这些服务也会提供图像的完整 URL，然后可以在你自己的 Web 站点中链接到它。

如果你猜测 img 指“image”（图像），那就猜对了。同样，src 指“source”（源），或者指向图像文件的位置的引用。如本书开头所讨论的，图像总是存储在一个与 Web 页面的文本（HTML 文件）分隔开的文件中，即使在浏览器中查看时它看上去像是相同页面的一部分。

与用于超链接的<a>标签一样，可以在<img />标签的 src 属性中指定任何完整的 Internet 地址，作为图像文件的位置。也可以使用相对地址，比如/images/birdy.jpg 或../smiley.gif。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 11.2 利用文本描述图像

程序清单 11.1 中的<img />标签中包括一条短文本消息，在这里是 alt="Half Dome"。alt 代表替代文本（alternate text），如果图像没有加载，就会显示该消息，用以代替图像。如果图像的地址不正确，如果 Internet 连接非常慢并且数据还没有传输，或者如果用户使用的是纯文本浏览器或屏幕阅读器，那么图像可能不会加载。图 11.2 显示了一个使用 alt 文本代替图像的示例。每种 Web 浏览器都以不同的方式呈现 alt 文本，但是当它是 HTML 文档的一部分时，仍然会提供信息。

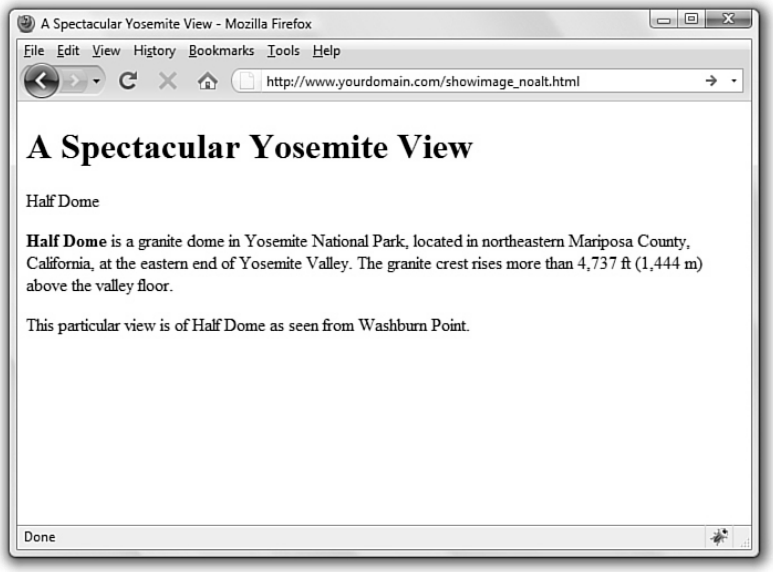



图 11.2 当没有显示图像时，用户将会看到 alt 消息

甚至当图形已经完全加载并且在 Web 浏览器中可见时，无论何时鼠标指针经过图像，alt 消息都可能会出现在一个小方框中（称为工具提示[ToolTip]）。alt 消息也可以帮助任何具有视觉障碍（或者使用基于语音的界面阅读 Web 页面）的用户。

必须在 Web 页面上的每个<img />标签中都包括合适的 alt 属性，要牢记在各种情况下人们都可能会看到该消息。图像的非常简短的描述通常是最佳的，但是 Web 页面作者有时会在他们的 alt 消息中放入简短的广告消息或者微妙的幽默语。不建议放入太多的幽默语或者不充分的信息，因为它根本没有什么用。对于较小的或者不重要的图像，可能会诱使人们完全省略 alt 消息，但是 alt 属性是<img />标签必需的属性。这并不意味着页面将不会正确地显示，但是意味着你将违反 HTML 标准。如果你绝对不需要 alt 属性，我建议给它分配一个空文本消息（alt=""），对于较小的或者装饰性的图像，有时就是这样。

## 11.3 指定图像的高度和宽度

由于文本在 Internet 上的移动速度要远远快于图形，大多数 Web 浏览器最终都会在显示图像之前，先显示页面上的文本。这使用户在等待查看图片前可以先阅读一些内容，从而使整个页面看上去似乎加载得更快。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**Did you Know?**

**提示：**为图像指定的高度和宽度不必与图像的实际高度和宽度匹配。Web 浏览器将尝试挤压或者拉伸图像，以显示你指定的任何大小。不过，这一般是一个糟糕的主意，因为浏览器不是特别擅长于调整图像的大小。如果你知道自己希望图像显示得更小一些，那么在图像编辑器中调整它的大小无疑是更好的做法。

可以通过明确指出每一幅图像的高度和宽度，确保页面上的所有内容尽可能快地显示出来，并且出现在正确的位置。这样，Web 浏览器可以在它布局页面以及等待图像传输完时，立即并且准确地为每一幅图像留出空间。

对于你想包括在站点中的每一幅图像，都可以使用图形程序确定它的准确的高度和宽度（以像素为单位）。你还可以使用系统工具，查找这些图像的属性。例如，在 Windows 中，可以通过右击图像，选择 Properties（属性），然后选择 Details（详细信息）命令，查看图像的高度和宽度。知道了图像的高度和宽度之后，就可以在<img />标签中包括它的尺寸，如下所示：

```
<imgsrc="myimage.gif" alt="Fancy Picture" width="200" height="100" />
```

**By the Way**

**注意：**在编写本书时，Web 开发人员和设计师正在讨论所谓的响应性图像的创建和实现，这类图像是指依赖于浏览器和媒体类型以不同的大小和分辨率显示的图像。目前引入了一个名为<picture>的新的 HTML5 元素，但它尚未成为标准的一部分，也没有被许多 Web 浏览器实现。如果你计划在将来对 Web 上的图像进行任何处理，就要记录这类信息，并且时不时地在 http://picture.responsiveimages.org/上检查这个新元素的状态。

## 11.4 对齐图像

就像可以在页面上对齐文本一样，也可以使用特殊的属性在页面上对齐图像。可以相对于周围的文本和其他图像，在水平和垂直方向上对齐图像。

### 11.4.1 水平图像对齐

第 5 章讲到，可以使用 CSS text-align 属性使元素内的内容居中、对齐到右边或者对齐到左边。这些样式设置将同时影响文本和图像，并且可以在任何块元素（比如<p>）内使用它们。

像文本一样，图像通常对齐到左边，除非另一种对齐设置指示它们应该居中或者对齐到右边。换句话说，left 是 CSS text-align 属性的默认值。

也可以直接在<img />标签内使用 CSS float 属性，使文本环绕在图像周围。

在程序清单 11.2 中，<img style="float:left" />把第一幅图像对齐到左边，并使文本环绕在它的右边。类似地，<img style="float:right" />把第二幅图像对齐到右边，并使文本环绕在它的左边。图 11.3 显示了这些图像在 Web 页面上是怎样对齐的。没有把一幅图像浮动到中

异步社区会员 Kensuke(18319066421) 专享 尊重版权

间的概念，因为无法确定怎样使文本环绕在它的两边。

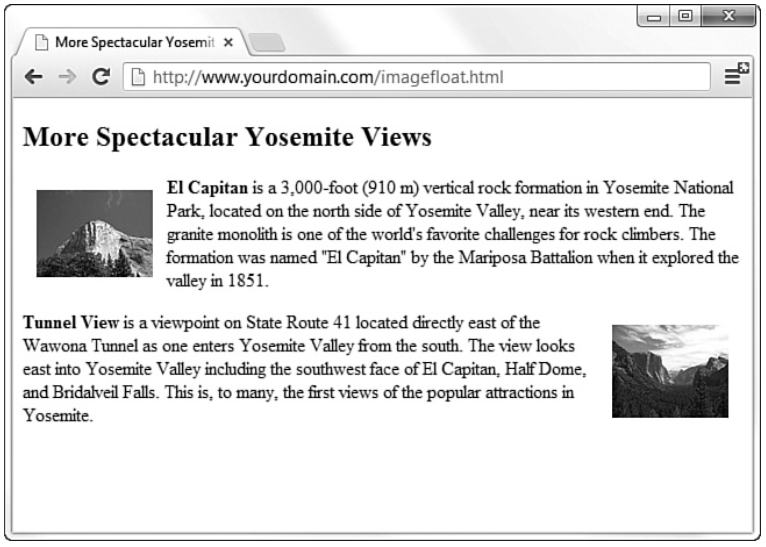



图 11.3 显示程序清单 11.2 中的图像对齐方式

**程序清单 11.2　使用 float 样式属性在 Web 页面上对齐图像**

```
<!DOCTYPE html>

<html lang="en">
 <head>
  <title>More Spectacular Yosemite Views</title>
 </head>

 <body>
  <section>
    <header>
    <h1>More Spectacular Yosemite Views</h1>
    </header>
    <p><img src="elcap_sm.jpg" alt="El Capitan" width="100"
    height="75" style="float:left; padding: 12px;"/><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><img src="tunnelview_sm.jpg" alt="Tunnel View" width="100"
    height="80" style="float:right; padding: 12px;"/><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>
  </section>
 </body>
</html>
```

**注意：**注意在程序清单 11.2 中，为使用的两个<img/>标签在样式属性中添加了填充。这种填充在图像与文本之间提供了一些呼吸的空间——在图像的 4 边都是 12 像素。在第 13 章中将学习关于填充的更多知识。

**By the Way**

异步社区会员 Kensuke(18319066421) 专享 尊重版权

### 11.4.2 垂直图像对齐

有时，你希望在文本行的中间插入一幅较小的图像，或者希望把单独一行文本放在图像旁边作为文字说明。在任何一种情况下，对文本和图像在垂直方向上的对齐进行一些控制将很方便。图像的底部应该与字母的底部对齐吗，或者应该把文本和图像全部进行排列以使得它们的中间对齐吗？你可以选择这些以及另外几个选项。

- 要把一幅图像的顶部与同一行上最高的图像或字母的顶部对齐，可以使用<img style="vertical-align:text-top" />。
- 要把图像的底部与文本的底部对齐，可以使用<img style="vertical-align:text-bottom" />。
- 要把图像的中间与行上的所有内容的总体垂直中心对齐，可以使用<img style="vertical-align:middle" />。
- 要把图像的底部与文本的基线对齐，可以使用<img style="vertical-align:baseline" />。

**By the Way**

**注意：**CSS vertical-align 属性还支持 top 和 bottom 值，无论行上具有任何文本，它们都可以使图像与元素所在行的总体顶部或底部对齐。

程序清单 11.3 中使用了所有这些选项，如图 11.4 所示。4 幅缩略图现在沿着页面垂直列出，并且每一幅图像旁边还带有描述性文本。这里使用了 CSS vertical-align 属性的多种设置，用于对齐每一幅图像及其相关的文本。这肯定不是最漂亮的页面，但是应该可以让你清楚各种对齐方式。

**程序清单 11.3 使用 vertical-align 样式对齐文本与图像**

```
<!DOCTYPE html>

<html lang="en">
 <head>
  <title>Small But Mighty Spectacular Yosemite Views</title>
 </head>

 <body>
  <section>
    <header>
      <h1>Small But Mighty Yosemite Views</h1>
    </header>
    <p><img src="elcap_sm.jpg" alt="El Capitan" width="100"
    height="75" style="vertical-align:text-top;"/><strong>El
    Capitan</strong> is a 3,000-foot (910 m) vertical rock formation
    in Yosemite National Park.</p>
    <p><img src="tunnelview_sm.jpg" alt="Tunnel View" width="100"
    height="80" style="vertical-align:text-bottom;"/><strong>Tunnel
    View</strong> looks east into Yosemite Valley.</p>
    <p><img src="upperyosefalls_sm.jpg" alt="Upper Yosemite Falls"
    width="87" height="100" style="vertical-align:middle;"/><strong>Upper
    Yosemite Falls</strong> are 1,430 ft and are among the twenty highest
    waterfalls in the world. </p>
    <p><img src="hangingrock_sm.jpg" alt="Hanging Rock" width="100"
    height="75" style="vertical-align:baseline;"/><strong>Hanging
    Rock</strong>, off Glacier Point, used to be a popular spot for people
    to, well, hang from. Crazy people.</p>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

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

**提示：**如果没有在<img />标签中或者用于该标签的类中指定任何 CSS vertical-align 属性，那么图像的底部将与它旁边的任何文本的基线对齐。这意味着实际上永远不必使用 vertical-align:baseline，因为默认就假定使用这种对齐方式。不过，如果为图像指定了边距，并且打算更准确地对齐文本，那么就可能希望明确地把 vertical-align 属性设置为 text-bottom。

**Did you Know?**

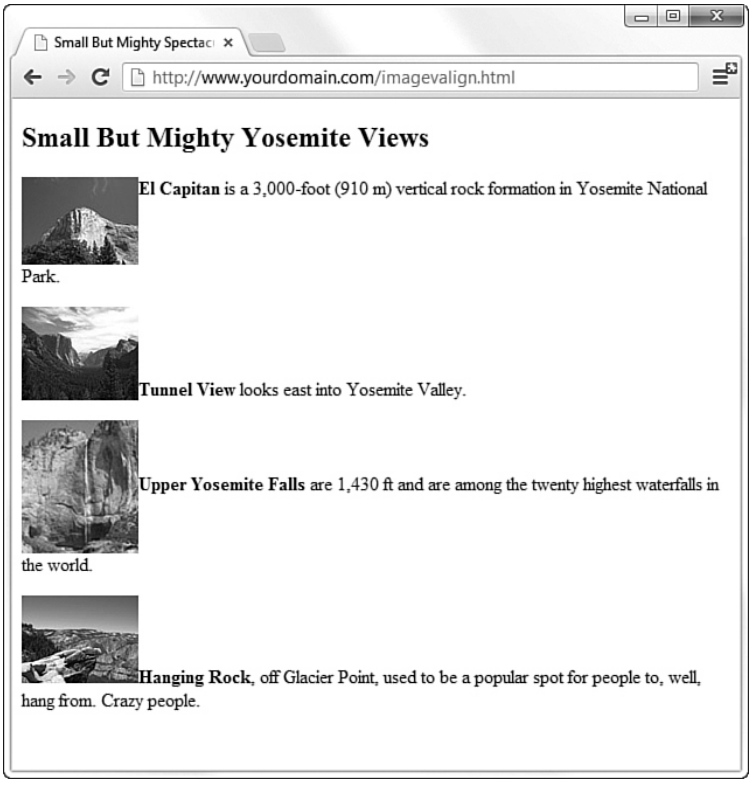



图 11.4 显示程序清单 11.3 中使用的垂直图像对齐选项

## 11.5 把图像转变成链接

在图 11.1 中，你可能注意到页面上的图像相当大。在这个特定的示例中，这样做很好，但是当你尝试展示多幅图像时，它就不那么理想了。创建较小的图像缩略图链接到每一幅图像的较大版本将更有意义。然后，可以在页面上排列缩略图，使得访问者可以轻松地查看所有的书面内容，即使他们看到的只是实际（较大）图像的较小版本。缩略图是可以使用图像链接给页面添加情趣的许多方式之一。

要把任何图像转变成指向另一个页面或图像的可单击的链接，可以使用在第 8 章中学过的<a>标签来创建文本链接。程序清单 11.4 包含用于在文本内显示图像缩略图的代码，并且那些缩略图链接到图像的较大版本。为了确保用户知道单击缩略图，将把图像和一些帮助文本封闭在一个 中，如图 11.5 所示。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

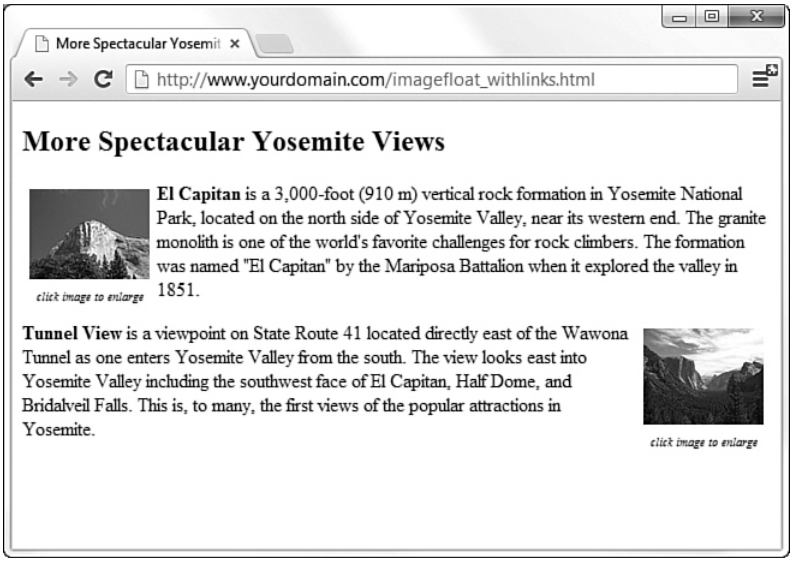



图 11.5

使用缩略图作为链接，可以改进使用大图像的页面的布局

## 程序清单 11.4 使用缩略图作为有效的图像链接

```
<!DOCTYPE html>

<html lang="en">
 <head>
  <title>More Spectacular Yosemite Views</title>
   <style type="text/css">
   div.imageleft {
    float:left;
    clear: all;
    text-align:center;
    font-size:10px;
    font-style:italic;
   }
   div.imageright {
    float:right;
    clear: all;
    text-align:center;
    font-size:10px;
    font-style:italic;
   }
   img {
    padding: 6px;
    border: none;
   }
   </style>
 </head>
 <body>
  <section>
    <header>
    <h1>More Spectacular Yosemite Views</h1>
    </header>
    <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>
    <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
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    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>
    <div class="imageright">
    <a href="http://www.flickr.com/photos/nofancyname/614287355/"><img
    src="tunnelview_sm.jpg" alt="Tunnel View" width="100" height="80"/></a>
    <br/>click image to enlarge</div>
    <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>
    </section>
  </body>
</html>
```

程序清单 11.4 中的代码使用了一些额外的样式，在后面的章节中将更详细地解释它们，但是你应该能够搞清楚一些基本的知识。

- <a>标签把这些特定的图像链接到较大的版本，在这里，它们存储在外部服务器上（在 Flickr 上）。
- 标签以及它们的样式用于对齐那些图形与文字说明集（并且还包括一些填充）。

除非另有指示，否则 Web 浏览器将会在每个图像链接的边缘周围显示一个彩色的矩形。与文本链接一样，对于近来未被访问的链接，该矩形通常显示为蓝色；对于近来访问过的链接，则显示为紫色，除非在样式表中指定不同颜色的链接。由于你很少（如果曾经有过的话）希望在链接的图像周围显示这种难看的线条，通常应该在链接内的任何<img />标签中包括 style="border:none"。在这种情况下，border:none 样式就成为了用于<img/>元素的样式表条目的一部分，因为我们把相同的样式使用了两次。

当你单击所显示的示例页面上的缩略图之一时，将在浏览器中打开链接，如图 11.6 所示。

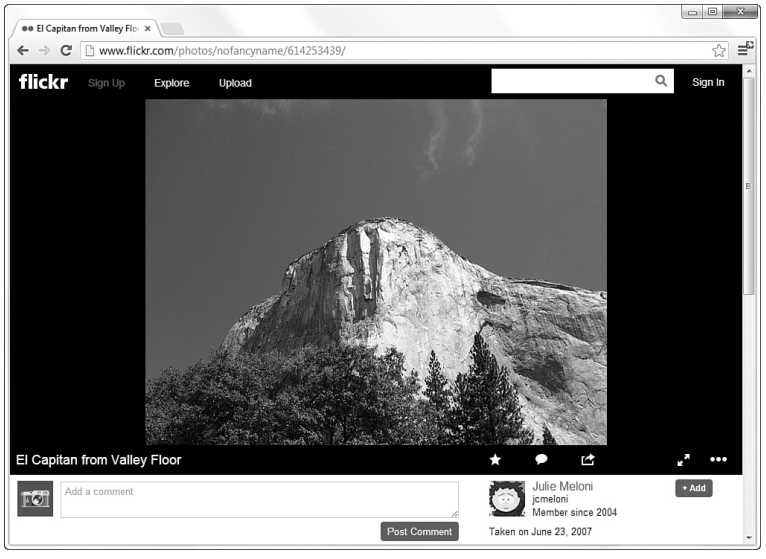



图 11.6 单击链接的缩略图，打开链接的目标

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 11.6 使用背景图像

在第 10 章中学过，可以使用背景图像充当容器元素中的一种墙纸，使得文本或其他图像出现在这种底层设计的顶部。

协作用于创建背景的基本 CSS 属性如下。

- background-color：指定元素的背景色。尽管它不是图像相关的，但它是与背景相关的属性集的一部分。如果图像是透明的或者没有加载，作为替代，用户将会看到背景色。
- background-image：使用以下语法指定将图像用作元素的背景：url('imagename.gif')。
- background-repeat：指定图像应该怎样在水平方向上和垂直方向上重复。默认情况下（未做任何指定），背景图像将同时在水平方向上和垂直方向上重复。其他选项包括 repeat（与默认值相同）、repeat-x（水平）、repeat-y（垂直）和 no-repeat（图形只出现一次）。
- background-position：指定图像相对于它的容器最初应该置于何处。选项包括 top-left、top-center、top-right、center-left、center-center、center-right、bottom-left、bottom-center、bottom-right，以及特定的像素和百分比位置。

在指定背景图像时，可以把所有这些规范放在一个属性中，比如：

```
body {
   background: #ffffff url('imagename.gif') no-repeat top right;
}
```

在前面的样式表条目中，Web 页面的主体元素将是白色的，并且将在右上方包括一个名为 imagename.gif 的图形。background 属性的另一种用途是为无序列表创建自定义的项目符号。要把图像用作项目符号，首先需要为<ul>标签定义样式，如下所示：

```
ul {
  list-style-type: none;
  padding-left: 0;
  margin-left: 0;
}
```

接下来，更改<li>标签的声明，如下所示：

```
li {
  background: url(mybullet.gif) left center no-repeat
}
```

确保 mybullet.gif（或者给图形提供的任何名称）位于 Web 服务器上并且是可访问的。在这里，所有的无序列表项都将显示自定义的图像，而不是标准的实心圆形项目符号。

在后面的学习中，当把 CSS 用于总体的页面布局时，我们将回过头来讨论背景属性的特定使用。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 11.7 使用图像映射

有时，你希望使用图像作为导航系统，但这超越了你通常在 Web 站点中看到的基于简单的按钮或者基于链接的导航系统。例如，也许你具有一个提供医疗信息的 Web 站点，并且你希望显示一幅人的身体的图像，它链接到一些页面，它们提供了关于各个身体部位的信息。或者你可以具有一个 Web 站点，它提供了一幅世界地图，用户可以单击以访问有关国家的信息。可以把一幅图像划分成一些区域，链接到不同的文档，这依赖于用户在该图像内单击的位置。这称为图像映射（imagemap），可以把任何图像制作成图像映射。

### 11.7.1 为什么图像映射并非总是必要的

关于图像映射要知道的第一点是：除了在非常特殊的情况下之外，可能不需要使用它们。使用多幅普通的图像，相互紧挨着摆放在一起，并为每一幅图像提供单独的链接，这几乎总是更容易、更高效的做法。

例如，图 11.7 是一个 Web 页面，其中显示了 12 个不同的公司标志。这个示例是商业世界里的一种常见的 Web 页面类型，其中你可以给合作伙伴提供一点免费的广告。你可以把这些标志展示为一幅大图像，并创建一个图像映射，提供分别指向 12 家公司的链接。用户可以单击图像中的每个标志，访问每家公司的站点。但是，每次希望向图像映射中添加新标志时，都将不得不修改整个图像，并重新映射热点（hot spot），从而会浪费一些时间。作为替代，在这种情况下，当事实证明图像映射并不合适时，只需像这个示例中一样简单地显示图像，通过使用 12 幅单独的图像（每一幅图像用于一家公司），并使每一幅图像包括一个指向那家特定公司的链接即可。

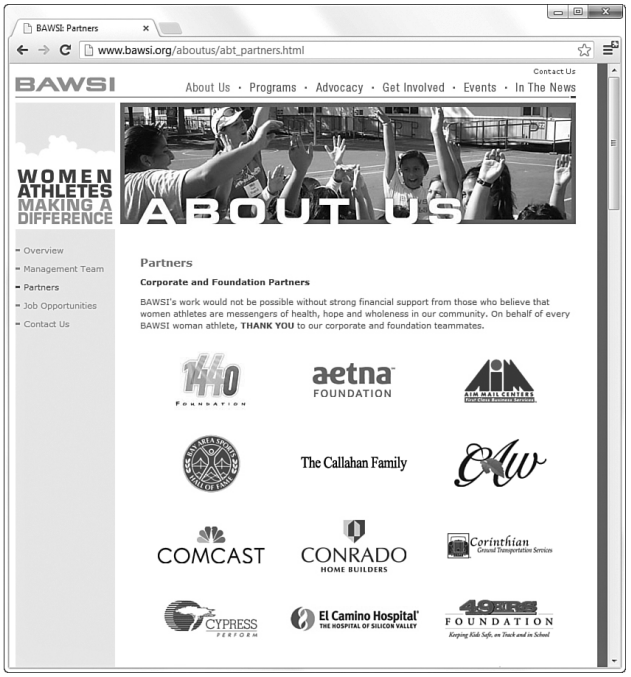



图 11.7

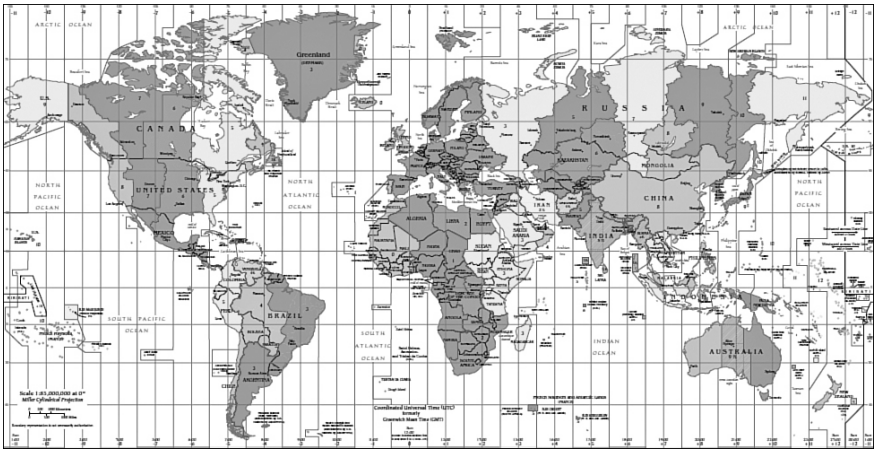
具有 12 个不同标志的 Web 页面，它可以展示为单个图像映射，或者分成 12 个单独的部分

异步社区会员 Kensuke(18319066421) 专享 尊重版权

如果一幅图像具有许多部分，它们怪异地排列在一起，或者图像的设计本身太复杂，以至于不能分成单独的图像，那么图像映射就是最佳的选择。图 11.8 显示了一幅最适合作为图像映射的图像，它是由美国中央情报局提供的世界标准时区的公共领域图像。

图 11.8

如果把这幅图像切分成完全相等的部分，结果不是很好——更好的做法是把它转变成图像映射



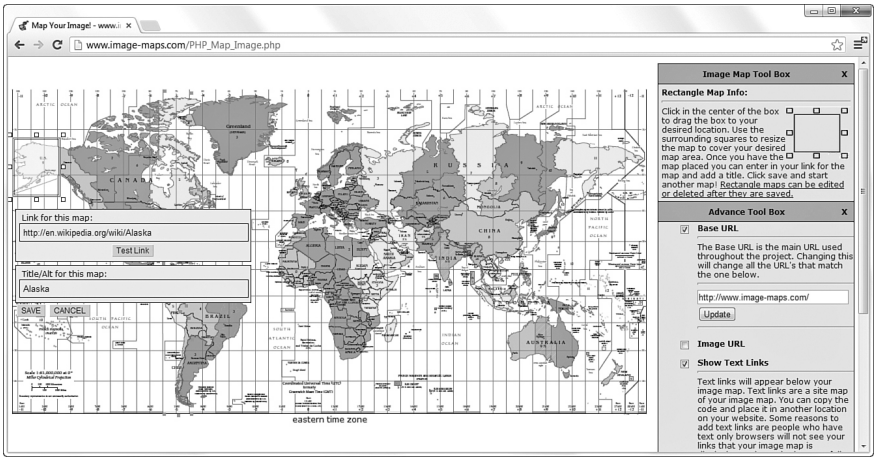

## 11.7.2 映射图像内的区域

要创建任何类型的图像映射，需要搞清楚图像内想转换成可单击链接的每个区域的数字像素坐标，这些可单击的链接也称为区域（area）。你的图形程序可能提供一种容易的方式来查找这些坐标。或者你可能想使用一个独立的图像映射工具（如 Mapedit[http://www.boutell.com/mapedit/]），或者一个在线图像映射制作器（如 http://www.image-maps.com/ 上的图像映射制作器）。除了帮助你映射坐标之外，这些工具还提供了使映射工作所需的 HTML 代码。

使用图像映射工具通常与使用鼠标在你想变成链接的区域周围绘制一个矩形（或者一种自定义的形状）一样简单。图 11.9 显示了这些矩形选区之一的结果，以及用于为这个链接添加 URL 以及标题或替换文本的界面。有几份信息是为图像映射创建 HTML 代码所必需的，包括链接的坐标、目标 URL 和替换文本。

图 11.9

使用图像映射工具创建单个图形的链接区域




异步社区会员 Kensuke(18319066421) 专享 尊重版权

Try It Yourself：创建你自己的图像映射

如果你有一幅自己的图像，并且在继续学习本章内容时把它转换成图像映射，那么你将更有可能记住制作图像映射。

- 对于初学者，最容易的做法是选择一幅相当大的图像，然后形象地把它分成粗略的矩形区域。
- 如果你手头没有合适的图像，可以使用你最喜爱的图形程序创建一幅图像。也许使用单独一张显示了几个人物的照片，并且使用每个人物作为图像映射的一个区域。
- 试用几个不同的图像映射工具，确定你最喜欢哪一个。从诸如 MapEdit（http://www.boutell.com/mapedit/）之类的独立软件开始，并转向 http://www.image-maps.com/上的在线图像映射制作器。还有其他的图像映射制作工具，可以使用所选的搜索引擎查找关于图像映射软件主题的各类信息。

### 11.7.3 为图像映射创建 HTML 代码

如果使用图像映射生成器，它将提供创建图像映射所需的 HTML 代码。不过，理解代码的各个部分是一个好主意，以便可以检查它正确与否。下面的 HTML 代码是开始创建任何图像映射所必需的：

```
<map name="mapname">
```

记住：可以使用你想要用于<map>标签的 name 的任何名称，尽管使之尽可能具有描述性将是有帮助的。接下来，需要为图像的每个区域提供一个<area />标签。下面给出了单个<area />标签的示例，它是通过如图 11.9 所示的动作创建的：

```
<area shape="rect" coords="1,73,74,163"
   href="http://en.wikipedia.org/wiki/Alaska"
   alt="Alaska" title="Alaska" />
```

这个<area />标签具有 5 个属性，可以把它们用于在图像映射中描述的每个区域。

- shape 指示区域是一个矩形（shape="rect"）、圆形（shape="circle"），还是一个不规则的多边形（shape="poly"）。
- coords 给出了区域的精确像素坐标。对于矩形，将给出左上角的 *x*、*y* 坐标，其后接着右下角的 *x*、*y* 坐标。对于圆形，将给出中心点的 *x*、*y* 坐标，其后接着半径（以像素为单位）。对于多边形，则按连线-点的顺序列出各个角的 *x*、*y* 坐标。

下面显示了一个映射的多边形的示例——它们可以获得一点疯狂的外观效果：

```
<area shape="poly"
coords="233,0,233,20,225,22,225,101,216,121,212,154,212,167,212,
181,222,195,220,209,226,214,226,234,232,252,224,253,223,261,231,
264,232,495,254,497,274,495,275,482,258,463,275,381,270,348,257,
338,266,329,272,313,271,301,258,292,264,284,262,262,272,263,272,
178,290,172,289,162,274,156,274,149,285,151,281,134,272,137,274,3"
href="http://en.wikipedia.org/wiki/Eastern_Time_Zone"
alt="Eastern Time Zone" title="Eastern Time Zone"/>
```

- href 指定区域链接到的位置。可以使用你将在普通的<a>链接标签中使用的任何地址或文件名。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

- alt 允许提供一段与形状关联的文本。你在前面学过，提供这段文本对于利用纯文本浏览器或屏幕阅读器浏览 Web 页面的用户很重要。使用 title 属性可以确保当用户访问指定的区域时，包含信息的工具提示也是可见的。

图像映射中的每个独特的可单击区域都必须被描述为单个区域，这意味着典型的图像映射包含一个区域的列表。在编码了<area />标签之后，就完成了图像映射的定义，因此利用</map>封闭标签把内容包装起来。

创建图像映射的最后一步是把它绑定到实际的映射图像上。使用普通的<img />标签把映射图像放在页面上。不过，额外的 usemap 属性的编码如下：

```
<img src="timezonemap.png" usemap="#timezonemap"
  style="border:none; width:977px;height:498px"
  alt="World Timezone Map" />
```

在指定 usemap 属性的值时，可以使用你放在<map>标签的 id 中的名称（并且不要忘记#符号）。还要包括 style 属性，指定图像的高度和宽度，以及关闭图像映射周围的边框，至于你是否选择在图像映射中保留边框，则取决于你自己。

程序清单 11.5 显示了一个示例 Web 页面的完整代码，该页面包含映射图形、它的图像映射以及几个映射的区域。

**程序清单 11.5 利用<map>和<area />标签定义图像映射的区域**

```
<!DOCTYPE html>

<html lang="en">
 <head>
  <title>Testing an Imagemap</title>
  </head>

 <body>
  <section>
    <header>
      <h1>Testing an Imagemap</h1>
    </header>
    <div style="text-align:center"> Click on an area to learn more about
    that location or time zone.<br/>
    <img src="timezonemap.png" usemap="#timezonemap"
    style="border:none; width:977px;height:498px "
    alt="World Timezone Map" /></div>

    <map name="timezonemap" id="timezonemap">
    <area shape="poly" coords=" 233,0,233,20,225,22,225,101,216,121,212,
     154,212,167,212,181,222,195,220,209,226,214,226,234,232,252,224,253,
     223,261,231,264,232,495,254, 497,274,495,275,482,258,463,275,381,270,
     348,257,338,266,329,272,313,271,301,258,292,264,284,262, 262,272,263,
     272,178,290, 172,289,162,274,156,274,149,285,151,281,134,272,137,
     274,3 "
    href="http://en.wikipedia.org/wiki/eastern_time_zone"
    alt="Eastern Time Zone" title="Eastern Time Zone" />
    <area shape="rect" coords="1,73,74,163 "
    href="http://en.wikipedia.org/wiki/Alaska"
    alt="Alaska" title="Alaska" />
    </map>
  </section>
 </body>
</html>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

图 11.10 显示了图像映射的应用。当把鼠标指针悬停在一个区域上时，将在图像映射上显示该区域的 alt 或 title 文本，在这个示例中是“Eastern Time Zone”。

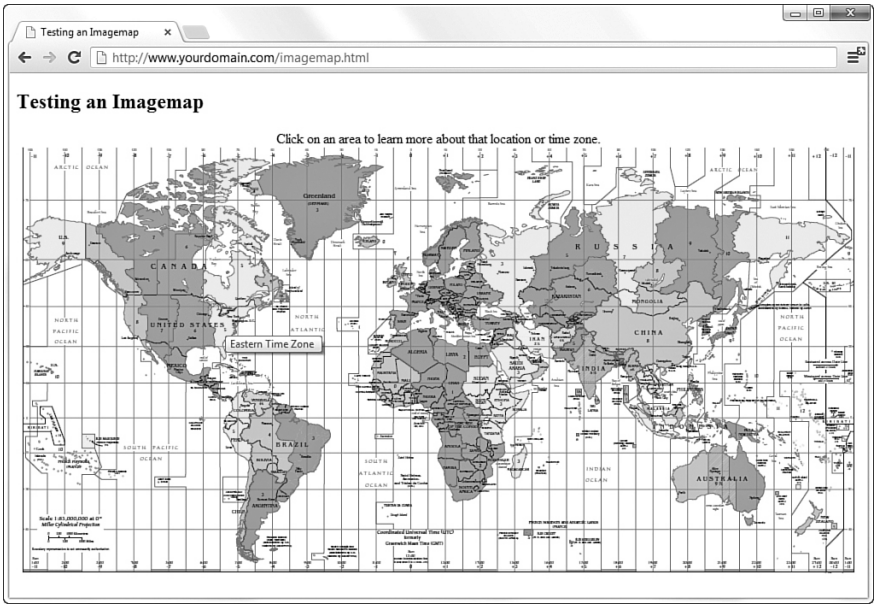



图 11.10

程序清单 11.5 中定义的图像映射在 Web 页面上的显示效果

**注意：**制作映射图像的一种方法只依赖于 CSS，而不是 HTML <map>标签。在第 16 章中将学习关于这方面的更多知识。

**By the Way**

## 11.8 小结

本章介绍了在 Web 页面中如何使用图像。在创建、编辑或者找到一些可用的图像之后，你将知道如何使用<img />标签把它们放在页面中。本章还介绍了如何包括一条短文本消息，用于在图像加载时代替它，以及用户无论何时把鼠标指针移到图像上，都会显示该消息。此外，本章介绍了如何控制每一幅图像的水平和垂直对齐，以及如何在图像的左边或右边环绕文本。

本章还介绍了如何把图像用作链接——通过使用<a>标签包围图像或者通过创建图像映射。本章最后讲解了一些在容器元素的背景中使用图像的相关知识。

表 11.1 总结了本章中介绍的标签和属性。

**表 11.1　　本章介绍的标签和属性**

| 标签 | 作用 |
|---|---|
| <img /> | 在页面内放置一个图像文件 |
| <map>...</map> | 通过<imgusemap="..." />引用的客户端图像映射。包括一个或多个<area />标签 |
| <area /> | 在客户端图像映射内定义可单击的链接 |
| **属性/样式** | |
| style="background-color:color" | 设置元素（比如<body>、<p>、 、<blockquote>及其他容器）的背景色 |

异步社区会员 Kensuke(18319066421) 专享 尊重版权

续表

| | |
|---|---|
| style="color:color" | 设置元素内的文本的颜色 |
| style="border:size type color" | 设置元素周围的 4 条边框的颜色。如果没有同时指定边框的宽度和类型，则不能使用边框颜色 |
| src="address" | 给出图像的地址或文件名 |
| alt="altdescription" | 给出图像的替代描述，显示它用于代替图像，主要针对不能查看图像本身的用户 |
| title="title" | 显示为图像标题的文本消息，通常位于图像上面的一个小弹出式方框中（工具提示） |
| width="width" | 指定图像的宽度（以像素为单位） |
| height="height" | 指定图像的高度（以像素为单位） |
| style="border:attributes" | 如果图像充当链接，就清除图像周围的边框 |
| style="vertical-align:alignment" | 把图像垂直对齐到 text-top、top、text-bottom、bottom、middle 或 baseline |
| style="float:float" | 把图像浮动到一边，使得文本可以环绕在它周围。可能的值有 left、right 和 none（默认） |
| usemap="name" | 用于客户端图像映射的图像映射规范的名称。用于<map>和<area /> |
| shape="value" | 在<area />标签内，指定可单击区域的形状。用于这个属性的有效的选项是 rect、poly 和 circle |
| coords="values" | 在<area />标签内，指定图像内的可单击区域的坐标。它的含义和设置因区域的类型而异 |
| href="linkurl" | 在<area />标签内，指定当区域被单击时应该加载的 URL |

## 11.9 问与答

**问：我就像你建议的那样使用<img />标签，但是当我查看页面时，我看到的只是一个小方框，里面具有一些形状。哪里出错了？**

**答：**你看到的破坏的图像图标可能意味着以下两件事之一：要么是 Web 浏览器找不到图像文件，要么是没有以浏览器可识别的格式保存图像。为了解决这些问题，首先要进行检查，以确保图像位于希望它出现的位置。如果是这样，那么就在图形编辑器中打开图像，并再次将其另存为 GIF、JPG 或 PNG 格式。

**问：如果我重叠了图像映射上的一些区域，则会发生什么事情？**

**答：**允许在图像映射上重叠区域。只需记住：在确定要遵循哪个链接时，一个区域将具有比另一个区域更高的优先级。优先级是依据在图像映射中先列出的是哪些区域来分配的。例如，图像映射中的第一个区域就具有比第二个区域更高的优先级，这意味着在区域的重叠部分单击将链接到第一个区域。如果图像映射内有一个区域未链接到任何内容（称为死区［dead area］），就可以使用这种重叠技巧有意阻止这个区域链接到任何内容。为此，只需把死区放在其他区域之前，使得死区重叠它们，然后把它的 href 属性设置为""。

## 11.10 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 11.10.1 问题

1. 怎样才能在 Web 页面的开始处插入一个 elephant.jpg 图像文件？
2. 当 Web 浏览器不能显示实际的 elephant.jpg 图像或者如果在使用屏幕阅读器时，怎样把单词 Elephant 提供给用户？
3. 如果具有一个名为 myimage.png 的图像，并且希望对齐它，使得文本行对齐图像的中间，那么你将使用什么样式属性？

## 11.10.2 答案

1. 把图像文件复制到与 HTML 文本文件相同的目录文件夹中。紧接在 HTML 文本文件中的<body>标签后面，输入 <imgsrc="elephant.jpg" alt="" /> 。
2. 使用以下 HTML 代码：

```
<imgsrc="elephant.jpg" alt="elephant" />
```

3. 你将使用 vertical-align:middle，确保文本对齐图像的中间。

## 11.10.3 练习

- 练习本章中介绍的任何图像放置方法对于确定图像可以并且将会在你设计的 Web 站点中所起的作用大有帮助。使用少数几幅示例图像，练习使用 float 样式，彼此相关地放置图像和文本。记住，用于 float 的可能的值包括 left、right 和 none（默认）。
- 与上一个练习非常相似，使用多种不同类型的垂直对齐方式，在你选择的 Web 页面上布置图像和文本。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 12 章

# 在 Web 站点中使用多媒体

在本章中你将学到：

- 怎样链接到多媒体文件；
- 怎样嵌入多媒体文件；
- 怎样使用 HTML5 音频和视频元素；
- 关于使用多媒体的额外提示。

术语多媒体（multimedia）包含我们在 Web 页面上看到和听到的一切内容：音频、视频和动画，以及静态图像和文本。在以前的章节中，你学习了如何使用图像和文本。在本章中，你将学习在 Web 站点中包括其他类型的多媒体。你将不会学习如何创建任何特定的音频或视频，而是将学习如何在站点中包括这样的文件，这是通过链接或嵌入内容实现的。

甚至在考虑在站点中包括多媒体之前，要记住并非每一个用户都具有可以播放你的媒体类型的设备，也不是所有的用户都具有允许快速传输这些大型文件的宽带 Internet 连接。总是要警告你的访问者，他们单击的链接将把他们带到多媒体文件上，并允许他们选择是否查看或收听内容，不要把这些文件强加给他们。

Try It Yourself：创建或查找一些多媒体以在你的 Web 站点中使用它们

在你学习如何把多媒体放到你的 Web 页面上之前，需要具有一些多媒体内容。

创建任何类型的多媒体都可能是一项具有挑战性的、复杂的任务。如果你计划从头开始创建自己的内容，将需要掌握比本书多得多的知识，变成下一位多媒体开发人员高手。不过，当你具有一些内容时，就可以使用本章中的提示来展示如何把你新创建的内容放入 Web 页面中。

如果你具有艺术方面的挑战，可以用多种替代方式获得有用的多媒体资源。除了显而易见的方式（比如聘请一位艺术家）之外，下面还给出了几条建议。

- Internet 上的大量材料都是免费的。当然，与内容的作者或当前所有者复核仍然是一

异步社区会员 Kensuke(18319066421) 专享 尊重版权

个好主意，你总不希望被起诉侵权。除此之外，美国政府的多个办公室产生的内容依法属于全体美国人（例如，在网上找到的任何 NASA 镜头都可供你免费使用）。

- 许多搜索引擎都具有用于查找多媒体文件的特定搜索能力。只要你小心应对版权问题，这可能是一种找到与特定主题相关的多媒体的轻松方式。简单地搜索 sample Flash animations、sample QuickTime movie 或 sample audio files 将产生比你可以处理的更多结果。
- 如果你具有创造性倾向，可以确定你最喜欢的媒体。例如，视频制作、音频制作或动画。当你具有了一个起点时，可以深入研究多种类型的软件，它们将使你能够创建这样的艺术杰作。许多公司（包括 Adobe[www.adobe.com]和 Apple[www.apple.com]）都提供了多媒体软件。

## 12.1 链接到多媒体文件

把视频或音频文件纳入到 Web 站点中的最简单、最可靠的选项是利用<a>标签简单地链接它，就像链接到另一个 HTML 文件一样。

例如，可以使用以下代码行，提供一只可爱的小猫的 MOV 视频：

```
<a href="cute_kitten.mov">View an adorable kitten!</a>
```

当用户单击单词“View an adorable kitten!”时，将会从你的 Web 服务器把 QuickTime 视频文件 cute_kitten.mov 传输到他或她的计算机上。一旦文件下载完成，就会自动启动用户安装的任何辅助应用程序或插件。如果没有找到兼容的辅助应用程序或插件，Web 浏览器将给用户提供一个机会以下载合适的插件，或者把视频保存到硬盘驱动器上以便以后查看。

**By the Way**

**注意：**如果你不熟悉辅助应用程序（helper application，或者简写为 helper app），下面就做一下解释，它们是 Web 浏览器调用的外部程序，用于显示它自己不能处理的任何类型的文件。一般来讲，无论何时 Web 浏览器不能独自显示某种类型的文件，就会调用与该文件类型关联的辅助应用程序。插件（plug-in）是直接安装在 Web 浏览器中的一种特殊的辅助应用程序。它们使你能够直接在浏览器窗口中查看多媒体内容。

程序清单 12.1 包含用于一个 Web 页面的代码，它使用一个简单的图像链接以 Windows Media 文件格式播放视频。除了图像链接之外，还在文本内放置了一个链接，用于给用户提供环境。

**程序清单 12.1　把图像链接到一个 Windows Media 视频**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Fun in the Pond</title>
  </head>

  <body>
    <h1>This Kitten Loves to Play!</h1>
    <div style="border-style:none; float:left; padding:12px"><a
    href="kitten.wmv"><img src="projector.gif" alt="Kitten Video" /></a>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    </div>
    <p>All kittens love to play, but this one is particularly fond of her
    yellow stuffed banana toy!</p>
    <p>Click <a href="kitten.wmv">here</a> or on the projector graphic to
    see a movie clip of this kitten in action.</p>
  </body>
</html>
```

这段代码简单地使用 GIF 图像 projector.gif 作为一个指向 kitten.wmv 视频剪辑的链接。图 12.1 显示了一个形象的示例页面，以及正在查看的放映机图像。当单击该图像时，就会调用 Windows Media Player（Windows 媒体播放器），并且开始播放影片。

图 12.1

将 GIF 图像 projector.gif 用作一个指向 Windows Media 文件的图像链接，用以启动外部辅助应用程序

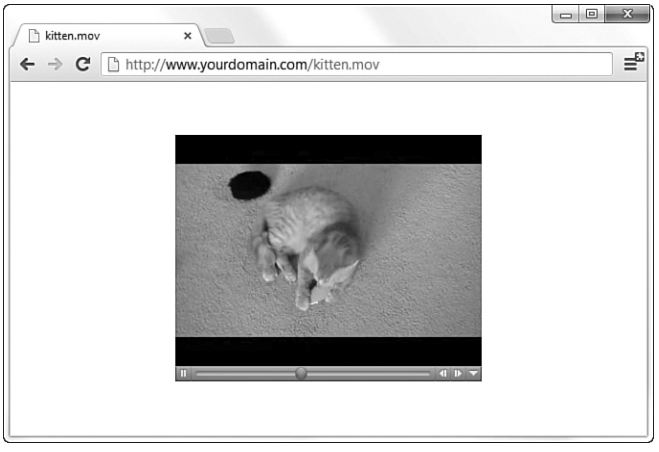



要查看视频，只需单击动画式放映机（或者段落中的文本链接）。这个动作将导致浏览器在插件的帮助下播放视频（如果找到一个可以播放剪辑的插件的话），或者把它提交给合适的辅助应用程序处理。

如果把链接从 kitten.wmv（Windows Media）改为 kitten.mov（QuickTime），浏览器将以不同的方式处理链接。这将不会启动另一个程序，QuickTime 插件使你能够直接在浏览器窗口中查看影片剪辑，如图 12.2 所示。

图 12.2

当沿着图像链接前进时，将使用 QuickTime 浏览器插件播放 QuickTime 影片 kitten.mov




如你可能猜到的，这种使用一个简单的链接播放多媒体文件的方法提供了最好的向后

异步社区会员 Kensuke(18319066421) 专享 尊重版权

兼容性，因为浏览器将承担搞清楚如何播放多媒体剪辑的所有责任。其缺点是不能对剪辑的播放方式进行过多的控制，并且肯定不能直接在页面的环境内播放剪辑。

**注意：**如果浏览器不支持 QuickTime，可以从 Apple 免费下载 QuickTime 播放器，下载地址是：http://www.apple.com/quicktime/。即使你没有安装 QuickTime，一些浏览器也会基于是否安装了插件以不同的方式播放 QuickTime 影片。例如，在我的 Windows 计算机上，Internet Explorer 和 Firefox 都通过插件直接在浏览器窗口中播放 QuickTime 影片，而 Opera 则会启动 QuickTime 作为辅助应用程序。

By the Way

## 12.2 嵌入多媒体文件

HTML 长时间以来都包含一个标准的<object>元素，用于在 Web 页面中嵌入多媒体文件。这个元素在桌面和移动浏览器中受到广泛支持，用于代表将会在浏览器中查看或播放但是要使用外部插件的资源类型。

在 Web 页面中嵌入多媒体文件将会产生一组软件控件，允许直接播放文件，无需辅助窗口，也不需要导航离开所在的页面。下面给出了用于嵌入你在前面见过的小猫视频的代码，它使用了<object>标签：

```
<object
  width="320"
  height="240"
  type="video/x-ms-wmv"
  data="pond.wmv"
</object>
```

**注意：**Windows Media Player 是一款足够先进的媒体播放器，它可以自动流线化多媒体文件，这意味着仅仅在加载了前几秒的内容之后，它就可以开始播放它们，注意到这一点很重要。其思想是：当你正在观看或收听前面的部分时，将在后台加载余下的内容。因此，访问者在观看或收听多媒体剪辑时，不必长时间等待下载完成。

By the Way

在图 12.3 中可以查看这段代码的结果。这段代码并不比向 Web 页面中添加图像复杂多少，唯一真正的区别是你在这里指定了文件类型（在这里是 video/x-ms-wmv）。

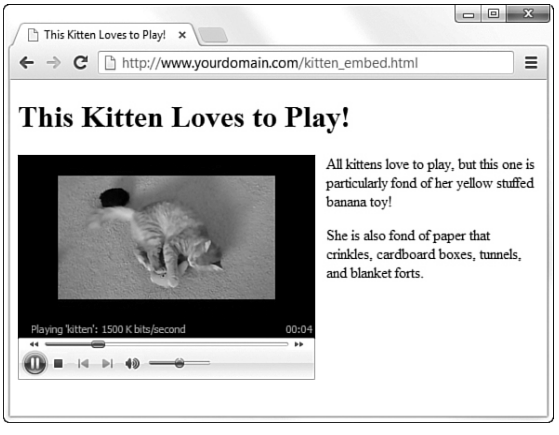



图 12.3

<object> 标签使你能够在 Web 页面中嵌入视频剪辑

异步社区会员 Kensuke(18319066421) 专享 尊重版权

<object>元素的 width 和 height 属性确定了嵌入式播放器（在这个示例中是 Windows Media Player）的大小。如果没有设置这些属性，一些浏览器将自动调整嵌入式播放器的大小，以适应内容，而其他的浏览器则不会显示任何内容。通过把它们设置为一种适合要播放的多媒体内容的大小，可以安全地播放它。

还只有另外两个属性是显示合适的视频文件所必需的，它们是 data 属性和 type 属性，其中前者的值是想要播放的文件名（或者文件的 URL），后者则用于标识要播放的媒体的类型。在这里，文件是 Windows Media Video（WMV）文件，通过 MIME 类型指定。MIME 类型（MIME type）是一个标识符，用于独特地标识 Internet 上的媒体对象的不同类型。MIME 代表多用途网际邮件扩展（Multipurpose Internet Mail Extension），并且这个名称来自于以下事实：MIME 类型最初用于标识电子邮件附件。这些 MIME 类型应该在<object>元素的 type 属性中使用，用于标识 data 属性所引用的多媒体对象的类型。

下面列出了你可能希望在 Web 页面中使用的几种流行的声音和视频格式的 MIME 类型。

- WAV Audio：audio/vnd.wave。
- MP3 Audio：audio/mpeg。
- MP4 Audio：audio/mp4。
- WMV：video/x-ms-wmv。
- MPEG Video：video/mpeg。
- MPEG4 Video：video/mp4。
- QuickTime：video/quicktime。

程序清单 12.2 显示了用于那个 Web 页面的相关代码，其中可以看到<object>元素出现在上下文中。

**程序清单 12.2 使用<object>元素直接嵌入 WMV 视频剪辑**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>This Kitten Loves to Play!</title>
  </head>

  <body>
    <h1>This Kitten Loves to Play!</h1>
    <div style="float:left; padding:12px">
    <object
       width="320"
       height="240"
       data="kitten.wmv"
       type="video/x-ms-wmv">
    </object>
    </div>
    <p>All kittens love to play, but this one is particularly fond
    of her yellow stuffed banana toy!</p>
    <p>She is also fond of paper that crinkles, cardboard boxes, tunnels,
    and blanket forts.</p>
  </body>
</html>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

管理本节讨论的是使用<object>元素“嵌入”多媒体对象，也可以使用另一个元素来实现相同的目标：适当地命名的<embed />元素。至于我为什么没有先讨论<embed />元素，这有一个良好的理由：具有负面的、令人混淆的历史。在 HTML5 之前的日子里，可以用两种方式把多媒体对象嵌入到 Web 页面中：使用<object>（官方规范的一部分）或者使用<embed />（不是官方规范的一部分，但是被老式的 Netscape 浏览器实现）。在这段时间，如果你希望确保嵌入式多媒体对象在所有浏览器中都是可见的，可以使用一点技巧，并且把<embed />元素包装在<object>元素内，如下所示：

```
<object
       width="320"
       height="240"
       data="kitten.wmv"
       type="video/x-ms-wmv">
       <embed
              width="320"
              height="240"
              type="video/x-ms-wmv"
              src="kitten.wmv" />
</object>
```

知道如何处理<object>元素的浏览器将会采用前一种方式，并且会忽略<embed/>元素。但是将会忽略<object>元素并且知道如何处理<embed/>元素的浏览器则会采用后一种方式。这使 Web 开发人员非常苦恼，就像现在解释起来会令人混淆一样。

现在，在 HTML5 的时代，<object>和<embed />元素都是有效的，可以使用其中任何一个元素，而不必同时使用两个元素。这两个元素的一般用法之间的唯一区别是：<object>元素使用 data 属性引用多媒体资源，而<embed />元素则使用 src 属性（非常像<img />标签）。

## 12.3 为音频和视频播放使用纯 HTML5 代码

在上一节中，你比自己可能想到的更多地了解了<object>和<embed />元素的令人混淆的历史。这段令人混淆的历史的最终结果是：尽管你当然可以继续使用<object>或<embed />元素在 Web 页面中嵌入音频或视频，并且依靠外部插件播放那些文件，但是 HTML5 包含一些与生俱来的元素用于播放音频和视频。在这种情况下，“与生俱来的”（native）是指浏览器无需外部插件即可播放多媒体文件的能力，并且所有平台（包括移动平台）上的所有主流的浏览器都支持这些元素。这将是往后播放音频和视频的首选方法。

程序清单 12.3 显示了如何使用<audio>元素嵌入将被浏览器与生俱来地播放的音频文件。<audio>元素相当简单，只需要一个属性（src），或者你想播放的资源的位置。不过，如你在程序清单 12.3 中所看到的，你将可能希望使用另外几个方便的属性。

**程序清单 12.3 使用<audio>元素嵌入并播放音频文件**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Let's Hear Some Music</title>
  </head>

  <body>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    <h1>Let's Hear Some Music</h1>
    <p>Better yet, let's use the HTML5 &lt;audio&gt; element to do so!</p>
    <audio
         src="manhattan_beach.mp3"
         preload="auto"
         controls
         autoplay
         loop>

         <!-- Message to display in case the audio element isn't
         supported. -->
         <p>Your browser does not support the audio element.</p>
    </audio>
  </body>
</html>
```

在这里，src 属性具有一个值 manhattan_beach.mp3，因为这是我想播放的音频文件的名称，除了 src 属性之外，我还将在这个<audio>元素中使用另外 4 个属性。

- preload：具有 3 个可能的值：none、auto 和 metadata。如果不希望缓冲文件，就使用 none；如果要缓冲文件，可使用 auto；如果希望只缓冲文件的元数据，则使用 metadata。
- controls：如果存在，则显示用于音频播放器的控件。
- autoplay：如果存在，则一加载完文件就播放它。
- loop：如果存在，则继续重复播放文件，直到手动停止它为止。

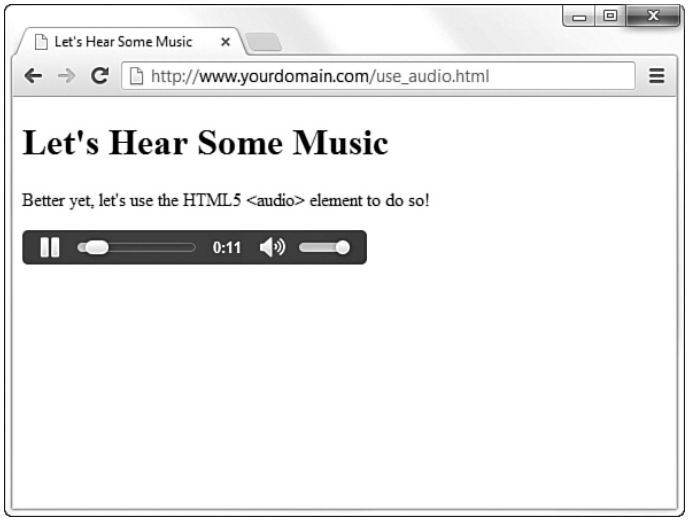
**Did you Know?**

**提示**：要注意在<audio>元素内给用户包括一条消息。尽管所有主流的浏览器的当前版本都支持<audio>元素，但是如果用户的浏览器不支持，那么他将代之以看到 标签内的消息。

图 12.4 显示了 Chrome Web 浏览器呈现的程序清单 12.3 中的页面。

图 12.4

使用<audio>元素播放声音文件




在实际中，你可能不希望在 Web 站点中自动播放和循环播放声音文件，这样做通常被认为是一种特别负面的用户体验。不过，如果你确实在自动播放一个声音文件（尽量不要这样做），就一定要包括播放器控件，以便用户可以立即关闭声音。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

使用 HTML5<video>元素与使用<audio>元素相当类似。程序清单 12.4 显示了它的一个示例，其中除了一些文本变化以外，唯一真正的改变是使用 height 和 width 属性定义被视频播放器和视频剪辑占据的空间。

**程序清单 12.4　使用<video>元素嵌入和播放视频文件**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Let's Watch a Video</title>
  </head>

  <body>
    <h1>Let's Watch a Video</h1>
    <p>Better yet, let's use the HTML5 &lt;video&gt; element to do so!</p>
    <video
          src="kitten.mov"
          preload="auto"
          width="320"
          height="240"
          controls
          autoplay
          loop>

          <!-- Message to display in case the video element isn't
          supported. -->
          <p>Your browser does not support the video element.</p>
    </video>
  </body>
</html>
```

preload、controls、autoplay 和 loop 属性为<video>元素所做的事情与它们为<audio>元素所做的事情完全相同。图 12.5 显示了 Chrome Web 浏览器呈现的程序清单 12.4 中的页面。

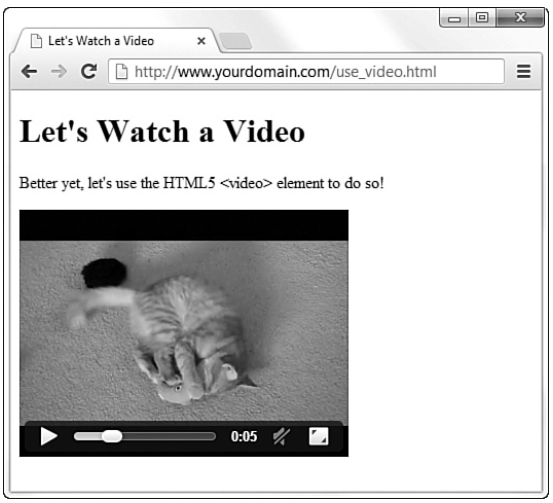



图 12.5

使用<video>元素播放视频文件

## 12.4　关于使用多媒体的额外提示

在把视频、音频或动画添加到 Web 站点中之前，要问问自己是否确实应该这么做。在

异步社区会员 Kensuke(18319066421) 专享 尊重版权

使用这些类型的多媒体时，一定要找出这样做的理由。不必要的声音和视频就像不必要的图像一样，可能会影响访问者关注站点总的主题思想。不过，如果你的主题思想是“查看我制作的视频”或者“收听我的音乐并下载一些歌曲”，那么多媒体绝对会在 Web 站点发挥重要作用。

要牢记以下几点额外的提示。

- 不要在页面中包括多媒体，并且不要把它设置成在页面加载时自动播放。总是要给用户提供选项，来开始（和停止）你的声音或视频。
- 当提供文件用于直接下载时，要允许用户选择文件类型。不要限制自己提供只能被一种类型的播放器或者一种操作系统播放的多媒体内容。
- 多媒体文件比典型的图形和文本文件更大，这意味着在 Web 服务器上需要有存储它们的空间，以及用于传输它们的带宽配额，以便把它们传输给任何通过你的 Web 站点请求它们的人。
- 如果你的站点全都是音频或视频，并且只附带提供了非常少的文本或图形，就要理解某个范围的受众将由于他们的系统或带宽的限制，而不会看到或听到你想展示的内容。要给这些用户提供额外的选项，以便获取你的信息。
- 利用免费的在线视频托管服务，比如 YouTube（www.youtube.com）。YouTube 不仅为你的视频剪辑提供了存储空间，而且还给你提供了必要的代码，用于在你自己的 Web 页面中嵌入视频。

## 12.5 小结

本章讲解了如何在 Web 页面中嵌入视频和声音，还讲解了如何使用指向多媒体文件的简单链接，这种用于播放媒体内容的方法受到了最广泛的支持，但它的灵活性最差。然后，介绍了如何使用<object>或<embed />元素直接在 Web 页面中嵌入媒体播放器，它可以用于包括广泛的媒体类型，包括 WAV、MP3、AVI、WMV 和 QuickTime 等。最后，本章介绍了 HTML5 中的<audio>和<video>元素，它们使浏览器能够与生俱来地呈现音频和视频文件，而无需外部的插件支持。

表 12.1 总结了本章中讨论的标签。

**表 12.1　　本章介绍的 HTML 标签和属性**

| 标签/属性 | 作用 |
|---|---|
| <object>...</object> | 在文档中嵌入图像、视频、Java 小程序、ActiveX 控件或其他对象 |
| **属性** | |
| width="width" | 指定嵌入式对象的宽度（以像素为单位） |
| height="height" | 指定嵌入式对象的高度（以像素为单位） |
| data="mediaurl" | 给出你想嵌入的资源的 URL |
| type="mimetype" | 指定多媒体内容的 MIME 类型 |
| <embed /> | 嵌入要被插件应用程序阅读或显示的多媒体文件 |

异步社区会员 Kensuke(18319066421) 专享 尊重版权

续表

| 属性 | |
|---|---|
| width="width" | 指定嵌入式对象的宽度（以像素为单位） |
| height="height" | 指定嵌入式对象的高度（以像素为单位） |
| src="mediaurl" | 给出要嵌入的文件的 URL |
| type="mimetype" | 指定多媒体内容的 MIME 类型 |
| <audio>...</ audio > | 在浏览器中与生俱来地播放音频文件 |
| **属性** | |
| src="mediaurl" | 给出要嵌入的文件的 URL |
| preload="preloadtype" | 指示是否预加载媒体文件。选项有：none、auto 和 metadata |
| controls | 指示浏览器显示音频播放器控件 |
| autoplay | 指示浏览器在完成加载时播放文件 |
| loop | 指示浏览器播放文件，直到明确停止它 |
| <video>...</ video > | 在浏览器中固有地播放视频文件 |
| **属性** | |
| src="mediaurl" | 给出要嵌入的文件的 URL |
| preload="preloadtype" | 指示是否预加载媒体文件。选项有：none、auto 和 metadata |
| width="width" | 指定嵌入式对象的宽度（以像素为单位） |
| height="height" | 指定嵌入式对象的高度（以像素为单位） |
| controls | 指示浏览器显示视频播放器控件 |
| autoplay | 指示浏览器在完成加载时播放文件 |
| loop | 指示浏览器播放文件，直到明确停止它 |

## 12.6 问与答

**问：我听过关于流线化视频和音频的许多事情，它们意指什么？**

**答：**在过去，通过大多数调制解调器获取视频和音频文件要花几分钟有时甚至是几小时的时间，这严重地限制了在 Web 页面上包括视频和音频。每一个人努力的目标是流线化视频或音频，这使得可以在接收数据的同时播放它们。换句话说，不必完全下载剪辑，就可以开始观看或收听它。

现在通过大多数媒体播放器广泛支持流线化播放，包括在独立版本和插件中。当使用<object>或<embed />元素嵌入一个媒体对象时，如果底层的媒体播放器支持流线化，那么它将自动流线化媒体剪辑。当使用<video>元素时，可以更精细地控制多媒体资源的缓冲和播放。

**问：在 QuickTime、Windows AVI/WAV、MPEG 及其他文件格式当中，怎样选择一种视听（audiovisual）文件格式？它们之间有任何重要的区别吗？**

**答：**QuickTime 是 Macintosh 用户当中最流行的视频格式，尽管 QuickTime 播放器也可供 Windows 使用。类似地，AVI/WMV 和 WAV/WMA 是用于 Windows 用户的视频和音频格

异步社区会员 Kensuke(18319066421) 专享 尊重版权

式，但是可以使用于 Macintosh 的播放器支持这些格式。MPEG 是另一个流行的音频和视频标准。MPEG-3（MP3）作为一种可供选择的高保真的音频标准，已经非常流行。另一种基于 MPEG 的音频格式是 Apple 的 AAC 格式，你可能更熟悉它，因为这是原始的 iTunes 音乐格式。

如果你的大多数受众都使用 Windows，则可以为视频使用 AVI/WMV 格式，并为音频使用 WAV/WMA 或 MP3 格式。如果你的受众包括大量的 Macintosh 用户，则可使用 QuickTime 或者至少提供某种替代选择。MP3 也是用于 Mac 音频的一个选项。

## 12.7 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 12.7.1 问题

1. 使可能最广泛的受众能够访问你的 Web 站点上的视频的最简单的方法是什么？
2. 你将用于在 Web 页面中嵌入一个名为 myvideo.mov 的视频文件的基本 HTML5 代码是什么？该视频在页面上需要一个大小为 320×305 像素的区域，并且你希望显示视频控件，但是不希望自动播放或循环播放文件。
3. 怎样编码<audio>元素，使得媒体剪辑将会重复播放？假定音频文件的名称为 myaudio.mp3。

### 12.7.2 答案

1. 只需链接到它：

```
<a href="myvideo.mov">my video</a>
```

2. 代码如下：

```
<video width="320" height="305" src="myvideo.mov" controls></video>
```

3. 使用下面的代码：

```
<audio src="myaudio.mp3" autoplay></audio>
```

### 12.7.3 练习

- 在 Web 上找到一些免费的音频和视频剪辑，并且练习使用 HTML5<audio>和<video>元素把它们放在你的文本内。
- 本章中介绍的利用<object>和<embed/>元素嵌入媒体的技术也适用于 Flash 文件。要查明怎么使用 Flash 向 Web 页面中添加交互式动画，可以检查 http://www.adobe.com/products/flash.html 上的 Flash 主页。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 13 章

# 处理边距、填充、对齐和浮动

在本章中你将学到：

- 怎样在元素周围添加边距；
- 怎样在元素内添加填充；
- 怎样使所有的一切都保持对齐；
- 怎样使用 float 属性。

既然你已经学习了一些创建 Web 内容的基础知识，在本章中，你将学习一些使用 CSS 的基本知识，用于增强该内容。在后面的章节中，你将深入研究使用 CSS 控制整个 Web 页面的各个方面，而不仅仅是单独的文本或图形。

不过，在处理页面布局之前，逐一理解 4 个特定的 CSS 属性很重要，然后就可以组合使用它们。

- margin 和 padding 属性：用于增加元素周围的空间。
- align 和 float 属性：用于相对于其他元素来放置元素。

本章中提供的示例并不是曾经创建的 Web 内容的最时尚的示例，但是它们也没有打算这样。作为替代，这些示例只清楚地说明了 HTML 与 CSS 如何协同工作。当你通过学习本章和本书的其他章节，并通过不断地实践所学的知识掌握了 CSS 时，将能够使用你自己的设计技能来增强使用相对基础的底层技术创建的内容。

## 13.1 使用边距

样式表边距（margin）让你能够在用于 Web 页面上的元素的矩形区域外面（outside）的四周添加空白。margin 属性处理的是元素外面的空间，记住这一点很重要。

下面列出了用于设置边距的样式属性。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

- 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：将上、右、下、左边距作为单个属性进行设置。

可以使用任何单独的边距属性或者使用单个 margin 属性指定边距。可以把边距指定为 auto，意味着浏览器本身将以特定的长度（像素、磅、“字母 m 的宽度”（em））或百分比设置边距。如果决定把边距设置为百分比，要记住百分比是基于包含元素的大小计算的。因此，如果把主体内的某个元素的 margin-left 属性设置为 25%，那么该元素的左边距最终将为整个页面宽度的 25%。不过，如果把那个元素内的某个元素的 margin-left 属性设置为 25%，它将是原来计算出的 25%的宽度的 25%。

**By the Way**

**注意**：你可以用至少两种不同的方式记住简写顺序。首先，如果把一个元素视作矩形，从顶部开始并按顺时针方向处理各条边：上边、右边、下边、左边。或者可以使用首字母助记法，并记住 TRBL（读作“trouble”或“tribble”，如果你是 *Star Trek* 游戏迷的话），它也代表一种可能的状态，可以防止你忘记边距属性的顺序。
另请注意：TRBL 顺序对于填充属性和边框属性也是有效的。

程序清单 13.1 中的代码在页面上产生 4 个矩形，每个矩形的宽度都是 250 像素，高度是 100 像素，并且具有 5 像素的纯黑色边框（见图 9.2）。每个矩形（在这里是 ）都具有一种不同的背景色。我们希望每个 周围的边距在各条边上都是 15 像素，因此可以使用以下代码：

```
margin-top:15px;
margin-right:15px;
margin-bottom:15px;
margin-left:15px;
```

也可以使用 margin 属性将其简写如下：

```
margin:15px 15px 15px 15px;
```

当你使用 margin 属性（或者 padding 或 border 属性）并且希望全部 4 个值都相同时，甚至可以进一步简化它，并使用以下代码：

```
margin:15px;
```

当使用简写法设置边距、填充或边框时，3 种方法都是适用的，它们基于在设置属性时使用了多少个值而有所不同。

- 一个值：所有边距的大小。
- 两个值：上/下边距和左/右边距（以这种顺序）的大小。
- 3 个值：上边距、左边距和右边距（给它们提供相同的值）、下边距（以这种顺序）的大小。
- 4 个值：上、右、下、左边距（以这种顺序）的大小。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

你可能发现坚持使用一个值或者使用全部 4 个值将更容易，但是当然不会要求必须如此。

**程序清单 13.1　产生 4 个具有边框和边距的彩色 的简单代码**

```
<!DOCTYPE html>

<html 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;
          margin:15px;
       }

       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>
```

在图 13.1 中可以查看程序清单 13.1 的输出结果。

接下来，只处理程序清单 13.1 中的样式表条目中的 margin 属性，让我们改变边距。在这个示例中，你确实看不到其中任何 元素的右边距，因为它们的右边没有任何内容，并且它们也没有对齐到右边。在记住这一点的情况下，让我们在所有这些元素中都把 margin-right 设置为 0px。除此之外，下一组目标是产生以下效果。

- 第 1 个色块周围没有边距。
- 在第 2 个色块周围，具有 15 像素的左边距和 5 像素的上边距，并且没有下边距。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

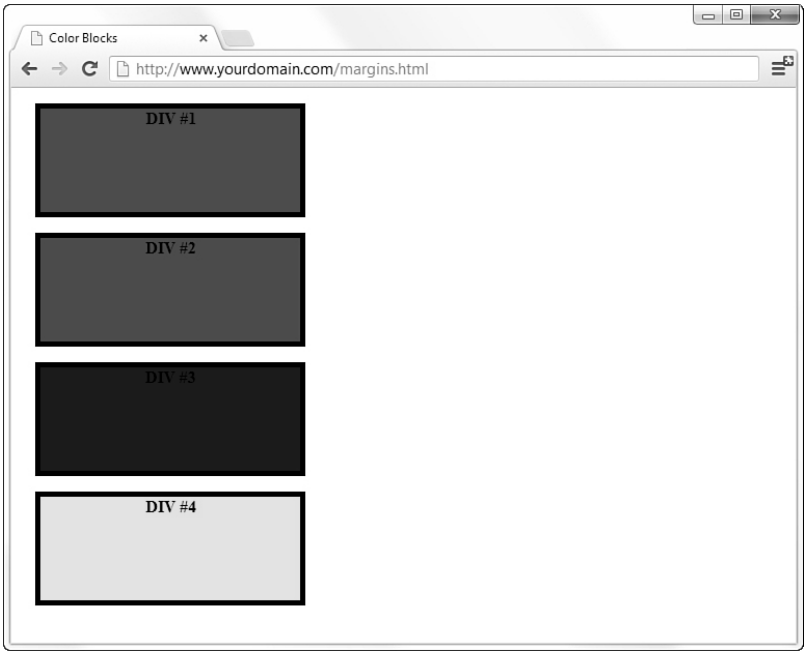



图 13.1

基本色块示例页面显示了 4 个色块，它们每个都具有相等的边距

- 在第 3 个色块周围，具有 75 像素的左边距，并且没有上边距或下边距。
- 在第 4 个色块周围，具有 250 像素的左边距和 25 像素的上边距。

这似乎比较直观，在第 1 个方块周围没有设置边距。但是，我们希望第 2 个方块顶部具有边距，因此在第 1 个方块与第 2 个方块之间确实具有可见的边距，即使我们没有为第一个方块指定边距。

用于 4 个命名的 的新样式表条目现在如下所示：

```
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;
}
```

代码改变的结果（见图 13.2）似乎是随意的，但是实际上相当有用，可用于指出另外几个要点。例如，当你通过回忆而知道目标之一是在第 1 个色块周围不产生任何边距时，你可能期望色块的边框与浏览器窗口齐平。但是，如图 13.2 所示，在页面的内容与浏览器窗口的框架之间具有明显的空白。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

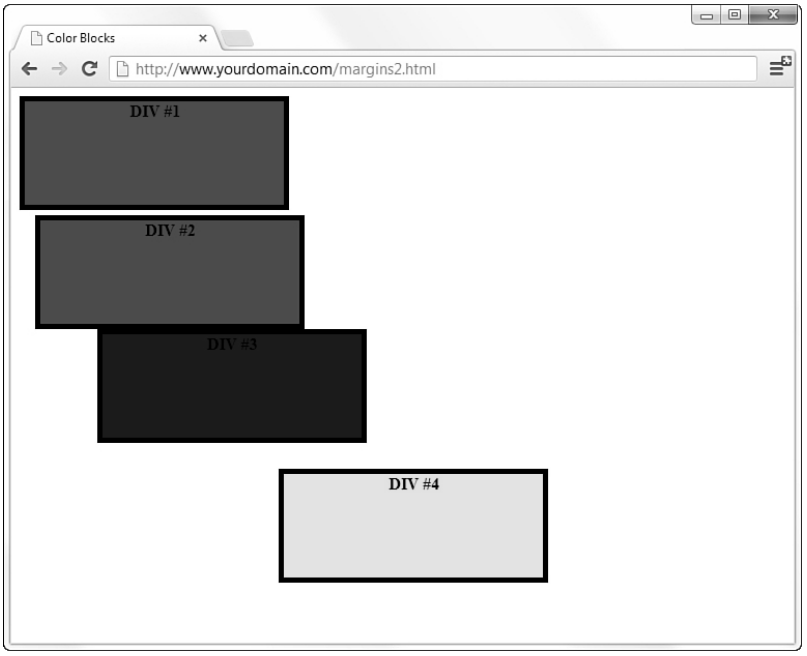



图 13.2
修改色块示例页面，显示一些不同的边距

如果我们处理元素放置（在下一章中将学习这方面的知识），这将会在布局中引发一个问题。为了确保元素放置与边距是从与浏览器齐平的位置开始计算的，需要处理<body>元素本身的边距。在这里，给样式表中添加以下代码：

```
body {
  margin:0px;
}
```

要记住的另一个“要点”是：如果有两个带边框的元素彼此堆叠在一起，但是它们之间没有边距，那么它们接触的位置似乎具有双边框。然后，你可能考虑使顶部元素的 border-bottom 的宽度减半，再使底部元素的 border-top 的宽度也减半。如果这样做，在彼此堆叠时，边框看上去将与其他边具有相同的宽度。

此外，你还可能认为使用 250 像素的左边距（ 的宽度）将在第 3 个色块结束的位置开始第 4 个色块。不过，事实并非如此，因为第 3 个色块具有 75 像素的 margin-left。为了使它们甚至更接近于对齐，用于第 4 个 div 的 margin-left 值必须是 325 像素。

像以下代码中所示的那样更改样式，产生如图 13.3 所示的间距：

```
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;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
}
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 6px6px;
  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;
}
```

这些更改将把<body>元素的边距设置为 0，从而确保 25 像素的 margin-left 值确实距离浏览器框架的边缘 25 像素。它还显示第 2 个和第 3 个色块彼此堆叠在一起，但是由于修改了 border 元素，使得双边框不会出现。此外，第 4 个色块开始于第 3 个色块结束的位置。

如图 13.3 所示，在第 3 个色块的右边缘与第 4 个色块的左边缘之间出现了一些重叠。为什么会这样，如果色块的宽度是 250 像素，第 3 个色块的 margin-left 值为 15 像素，那么第 4 个色块的左边被指望具有 265 像素的边距吗？好吧，它确实具有 265 像素的边距，但是这个边距大小并不够，因为你还必须把 6 像素的边框计算在内。更改第 4 个色块的 margin 属性，以反映出下面的代码将使第 3 个色块与第 4 个色块按计划的那样对齐，如图 13.4 所示：

```
margin:0px 0px 0px 276px;
```

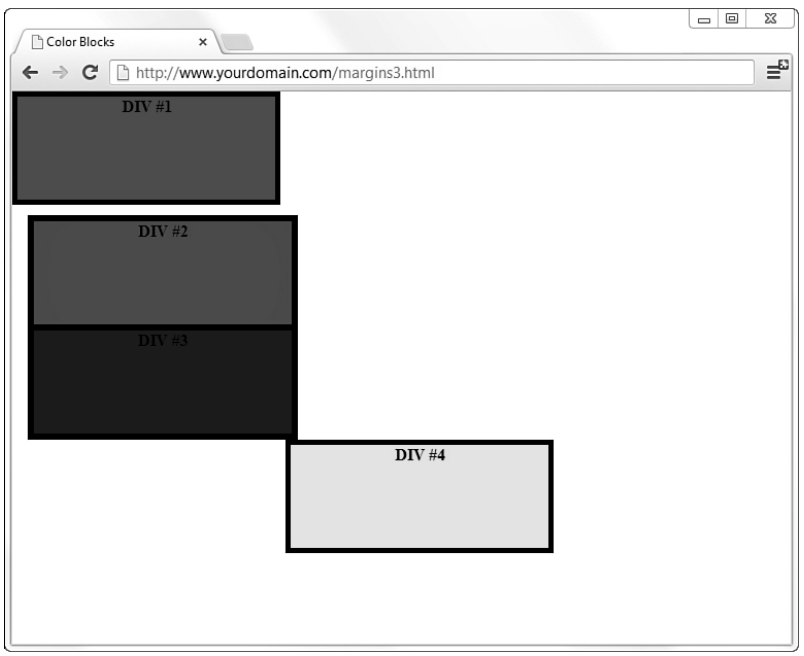



图 13.3

对色块进行的第三次修改将把项目拖到彼此更接近的位置

异步社区会员 Kensuke(18319066421) 专享 尊重版权

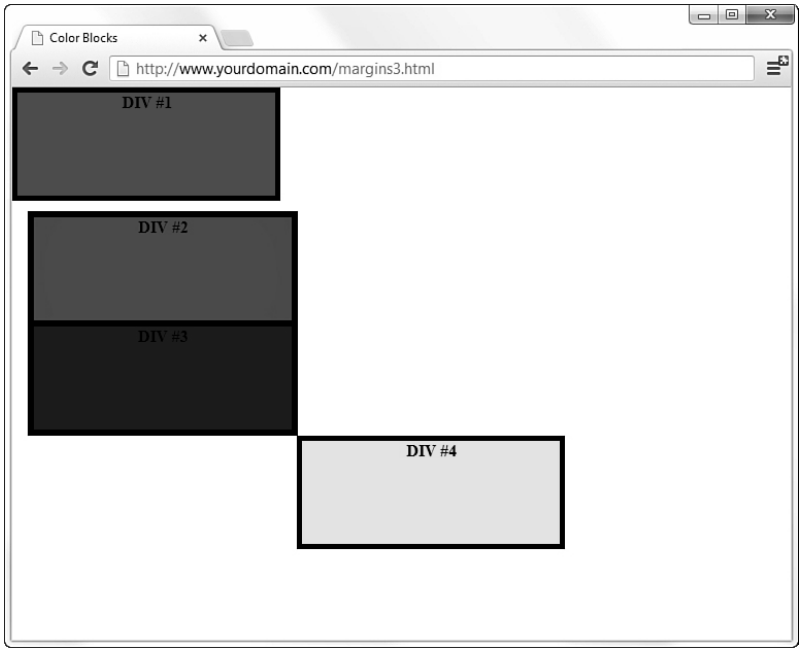



图 13.4

更改边距，允许 11 像素的边框宽度

如这些示例中所示，边距规范对于元素的放置极其有用，但是在设置这些规范时必须小心谨慎。

## 13.2 填充元素

填充类似于边距，这是由于它给元素添加额外的空间。它们之间的最大区别在于这个空间所在的位置。回忆可知：边距是添加在元素的外面的。另一方面，填充则是在元素的矩形区域内添加空间。例如，如果为元素创建一个样式规则，建立 50 像素的宽度和 30 像素的高度，然后把规则的填充设置为 5 像素，那么余下的内容区域将是 40 像素×20 像素。此外，由于元素的填充出现在元素的内容区域内，它将具有与元素的内容相同的样式，包括背景色。

使用 padding 属性之一指定样式规则的填充，其工作方式与 margin 属性非常相似。在设置样式规则的填充时，可以使用以下填充属性。

- 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：将上、右、下、左填充作为单个属性进行设置。

与边距一样，可以使用单独的填充属性或者单个 padding 属性设置样式规则的填充。也可以使用度量单位或百分比表示填充。

下面显示了一个示例，说明如何设置样式规则的左、右填充，使得元素的内容两边都具有 10 像素的填充：

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
padding-left:10px;
padding-right:10px;
```

与边距一样，可以利用单个属性（padding 属性）设置元素的所有填充。要设置 padding 属性，可以使用 margin 属性使用的 3 种同样的方法。

下面显示了一个示例，说明如何把样式规则的垂直填充（上/下）设置为 12 像素，以及把水平填充（左/右）设置为 8 像素：

```
padding:12px 8px;
```

下面给出了一段更明确的代码，它通过指定所有的填充值来执行相同的任务：

```
padding:12px 8px 12px 8px;
```

在所有前面的图形中，注意文本“DIV #1”、“DIV #2”等出现在色块的顶部，并且边框与文本之间只有一点空间。任何填充值都没有指定那个空间，但是它在元素内显示为一种默认设置。如果你希望对元素填充进行特定的控制，程序清单 13.2 显示了一些示例。所有的色块都是 250 像素宽和 100 像素高，具有 5 像素的纯黑色边框以及 25 像素的边距，如图 13.5 所示。在每个单独的 的填充值内会发生有趣的事情。

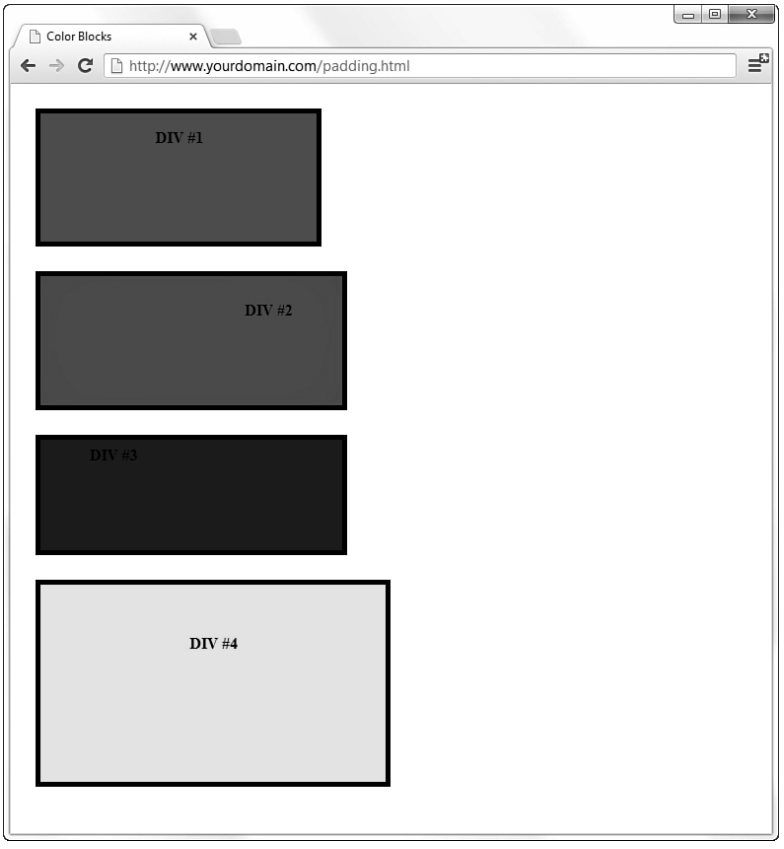



图 13.5 基本色块示例页面显示了 4 个具有可变填充的色块

**程序清单 13.2 用于产生 4 个具有边框、边距和填充的彩色 的简单代码**

```
<!DOCTYPE html>

<html lang="en">
  <head>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    <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>
```

你应该立即意识到这个示例中的某个方面出了差错。色块都被指望是 250 像素宽和 100 像素高。图 13.5 中的色块大小不一，因为尽管我们努力控制 的大小，但是后来应用的填充覆盖了初始大小声明。

如果在一个<p>元素中放置文本，并给该元素提供白色的背景，如图 13.6 所示，可以看到填充相对于文本的位置。当没有空间使用所有定义的所有填充时，周围的元素将不得不做出调整。在第 14 章中将详尽学习这种效果。

在 Web 设计中，利用 CSS 执行调整或微调次数最多的就是边距和填充。只需记住，边距是在元素外面，填充则是在其内。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

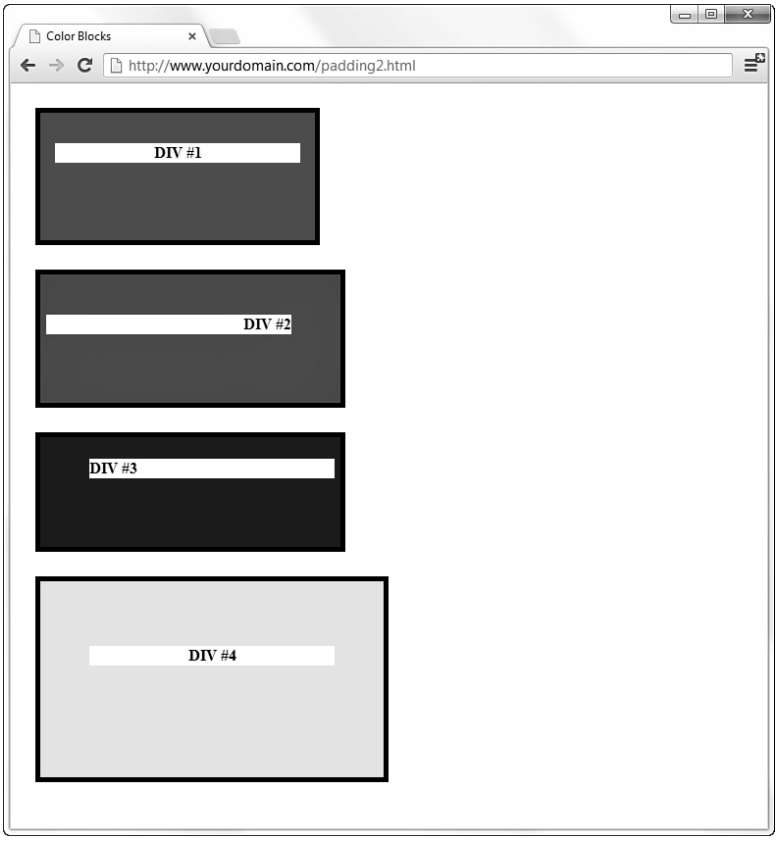



图 13.6
显示填充相对于文本的位置

## 13.3 使所有的一切保持对齐

要知道的是，Web 页面上的内容并不总会填充显示它的矩形区域的整个宽度，控制内容的对齐方式通常是有帮助的。即使矩形区域内的文本扩展到多行上，在对齐时也可能会进入图片，因为你可能希望文本左对齐、右对齐或居中。有两个样式属性使你能够控制元素的对齐，它们是 text-align 和 vertical-align。

在第 11 章中，你见过了这些样式属性实际应用的示例（在对齐图像时），但这无碍于在这里再次提及这些属性，因为对齐在总体页面设计中也起着重要作用。

复习一下，使用 text-align 在其边界区内水平地对齐元素，并且可以把它设置为 left、right、center 或 justify。

vertical-align 属性类似于 text-align，只不过它用于垂直地对齐元素。vertical-align 属性指定一个元素如何同其父元素对齐，或者在某些情况下与页面上的一些元素的当前行对齐。“当前行”指出现在相同父元素内的元素（即内联元素）的垂直放置方式。如果多个内联元素出现在同一行上，可以设置它们的垂直对齐方式，就像垂直地对齐它们一样。一个良好的示例是一个接一个地出现在一排图像，此时 vertical-align 属性可以使你能够垂直地对齐它们。

下面列出了 vertical-align 属性使用的公共值。

- top：将元素的顶部与当前行对齐。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

- middle：将元素的中间与其父元素的中间对齐。
- bottom：将元素的底部与当前行对齐。
- text-top：将元素的顶部与其父元素的顶部对齐。
- baseline：将元素的基线与其父元素的基线对齐。
- text-bottom：将元素的底部与其父元素的底部对齐。

对齐可以与边距、填充和 float 属性（将在下一节中学习它）协同工作，使你能够维持对设计的控制。

## 13.4 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，它是组合使用所有这些元素的智力游戏的最后几块拼图之一。简而言之，float 属性允许元素在设计中四处移动，使得其他元素可以环绕在它们周围。你通常会发现 float 属性与图像结合使用（如你在第 11 章中所看到的），但是你可以（并且许多设计师也这样做）在布局中浮动各种元素。

元素是水平（而非垂直）浮动的，因此你自己只需关心两个可能的值：right 和 left。在使用时，浮动的元素将尽可能远地向右边或向左边浮动（依赖于 float 的值），只要包含元素允许即可。例如，如果你具有 3 个 float 值为 left 的 ，它们都将对齐到包含主体元素的左边。如果在一个 内具有另一个 ，它们都将对齐到那个元素的左边，即使那个元素本身是浮动到右边的。

通过查看几个示例，可以最好地理解浮动，因此让我们转向程序清单 13.3。这个程序清单简单地定义了 3 个矩形 ，并把它们浮动到彼此的旁边（浮动到左边）。

**程序清单 13.3　使用 float 放置 **

```
<!DOCTYPE html>

<html 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;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

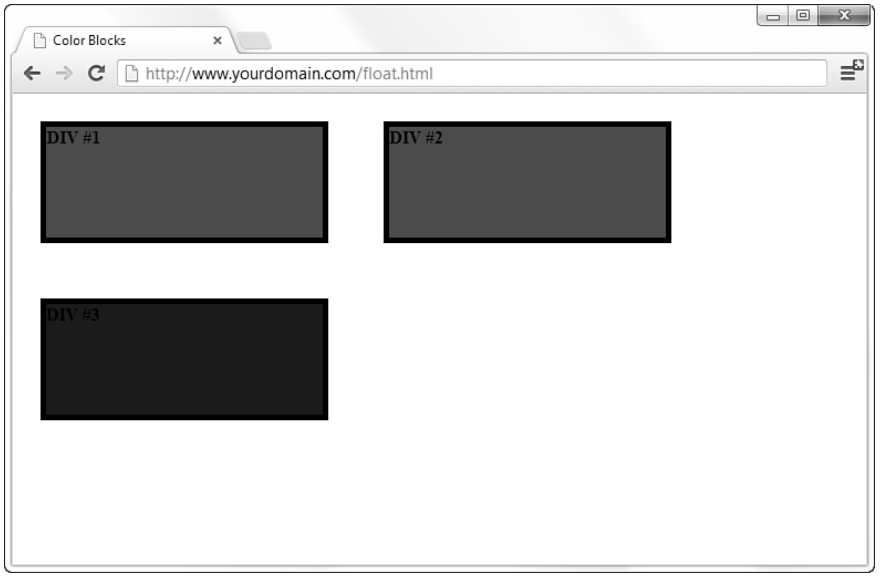
```
            float:left;
        }

        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>
```

图 13.7 显示了结果页面。你可能已经看到了一个问题：这 3 个色块被指望浮动到彼此的旁边。好吧，它们实际上是浮动到了彼此的旁边，但是浏览器窗口的宽度不够，无法显示这 3 个 250 像素宽的方块以及它们之间的 25 像素的边距。由于它们都是浮动的，第 3 个色块将简单地浮动到下一行。

图 13.7
使用 float 放置色块




可以想象到，在特殊设计的可视化布局中，这可能是一个问题，因此要注意边距、填充、对齐和填充，同时还要在目标浏览器窗口大小中进行测试。诚然，图 13.7 中的浏览器窗口是一个小窗口，当没有足够的空间在一行放下浮动的元素时，将把它们移到下一行。换句话说，如果利用较大的浏览器窗口打开相同的 HTML 文件，可能不会看到这个问题，这就是为什么还应该利用不同的分辨率检查站点的原因，以便查看是否需要进行修正。这里采用的修正措施是：调整 的边距及其他与大小相关的属性。

图 13.8 显示了在使用 float 属性时的另一种有趣的可能性。对程序清单 13.3 中的代码所做的唯一修改涉及使色块只有 100 像素宽，把边距减小到 10px，以及把第二个色块的 float 改为对齐到右边（而不是左边）。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

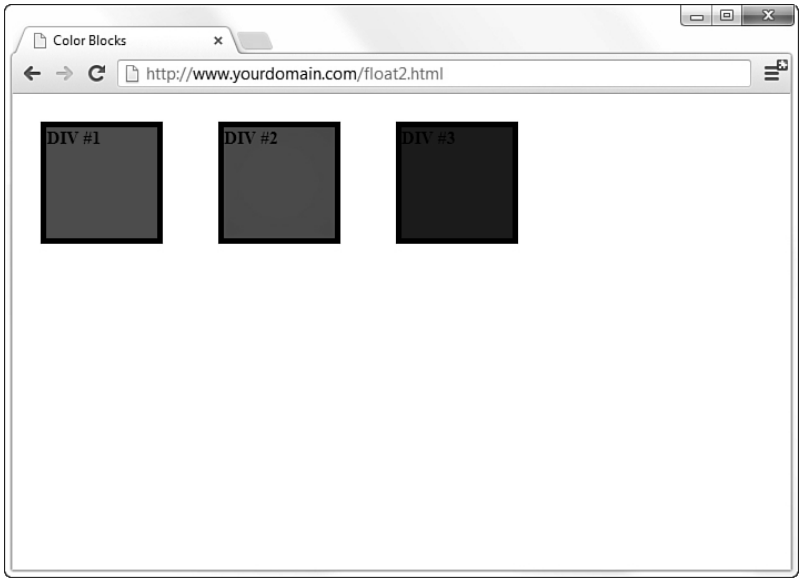



图 13.8
使用 float 放置色块

不过，有趣的事情发生了。第 2 个色块现在看上去显示为第 3 个色块，因为它是右对齐的。第 2 个色块的 float 值为 right，因此它浮动到了右边。第 1 个和第 3 个色块则尽可能浮动到左边，而不管 代码是怎样出现在 HTML 中的，如下所示：

```
<div id="d1">DIV #1</div>
<div id="d2">DIV #2</div>
<div id="d3">DIV #3</div>
```

浮动要进行许多次练习才能习惯，尤其是当页面中具有额外的元素而不仅仅是几个色块时。例如，当向色块组合中添加一个基本的段落时会发生什么事情？放置在浮动元素后面的所有元素都将在该元素周围浮动。为了避免这种情况，可以使用 clear 属性。

clear 属性具有 5 个可能的值：left、right、both、none 和 inherit，其中最常见的值是 left、right 和 both。指定 clear:left 可以确保左边不允许出现其他的浮动元素；clear:right 则可以确保右边不允许出现其他的浮动元素，等等。浮动和清除是一个边干边学的过程，因此可以在本章末尾的“测验”中查找更多的情况。

## 13.5 小结

本章介绍了基于 CSS 的设计中的一些最基本的样式属性：margin、padding 和 float。通过本章的学习，读者可以知道 margin 属性如何控制元素外面的空间，以及 padding 属性如何处理元素内部的空间。

在复习了上一章介绍过的 text-align 和 vertical-align 属性之后，本章介绍了 float 属性，该属性允许特定地放置元素以及那些元素周围的额外内容。

## 13.6 问与答

**问：边距和填充的示例都必须涉及方框和文本。我能够把边距和填充也应用于图像吗？**

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**答**：是的，可以把边距和填充应用于任何块级元素，如<p>、 、<img/>、列表（比如<ul>和<ol>）以及列表项（<li>）等。

## 13.7 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 13.7.1 问题

1. 要彼此相邻地放置两个 元素，但是它们之间具有 30 像素的边距，可以在样式表中使用什么条目？
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. 哪个样式表条目用于把文本放置在一个 内，并使之出现在距离元素顶部 12 像素的位置？

### 13.7.2 答案

1. 可以使用多个条目。第一个 使用样式属性 margin-right:15px；第二个 使用样式属性 margin-left:15px。或者可以根据需要使用 margin-right 或 margin-left 把全部 30 像素都分配给其中一个 。此外，至少第一个 需要给它分配 float:left。
2. 在这种情况下，可使用 clear:left。
3. `padding-top:12px`

### 13.7.3 练习

- 需要多加练习，才能完全理解边距、填充、对齐和浮动。在学习下一章的内容之前，使用你自己的色块代码或 ，练习各种方式以及各类间距和浮动。下一章将从整体上讨论 CSS 方框模型，它包含本章中讨论过的各个项目。
- 在你埋头工作时，练习对你迄今为止学过的每个块级元素应用边距和填充。熟悉在文本块内放置图像，并设置图像周围的边距，使得文本不会接触到图形的边缘。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 14 章

# 理解 CSS 方框模型和定位

在本章中你将学到：

- 怎样概念化 CSS 方框模型；
- 怎样定位元素；
- 怎样控制元素的堆叠方式；
- 怎样管理文本流。

上一章中，几次提到了 CSS 方框模型。本章开始讨论方框模型，并且解释你在上一章中学到的信息将如何帮助你理解该模型。如果你知道方框模型是如何工作的，在创建设计时就不会感到狂躁，然后就会意识到元素没有对齐或者它们似乎有点“偏”。你将知道，在大多数情况下，某些方面（边距、填充、边框）需要进行一点调整。

你还将学习 CSS 定位，包括以一种三维的方式（而不是垂直方式）把元素彼此堆叠起来。最后，你将学习使用 float 属性控制元素周围的文本流。

## 14.1 CSS 方框模型

HTML 中的每个元素都被视作是一个“方框”，无论它是一个段落、一个 ，还是一幅图像等。方框具有一致的属性，无论我们是否看到了它们，也无论样式表是否指定了它们。它们总是存在的，并且作为设计师，我们在创建布局时必须牢记它们是存在的。

图 14.1 显示了方框模型的图形。方框模型描述了每个 HTML 块级元素对于边框、填充和边距所具有的潜力，确切地讲是指将如何应用边框、填充和边距。换句话说，所有的元素在元素的内容与边框之间都具有一些填充。此外，边框可能是或者可能不是可见的，但是用于它的空间是存在的，就像元素的边框与元素外面的其他任何内容之间具有边距一样。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

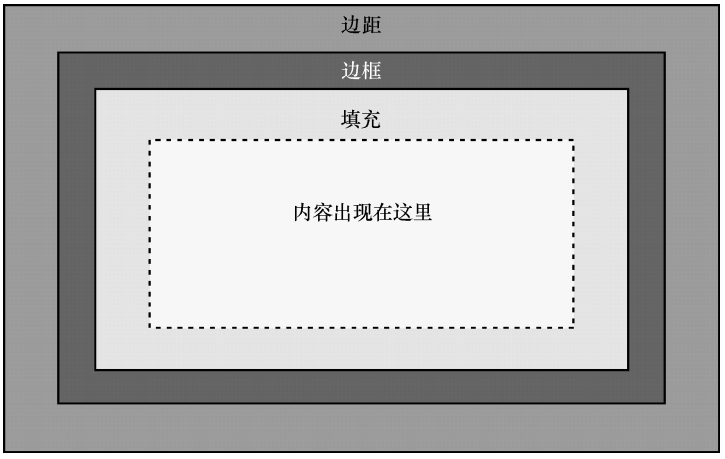



图 14.1

HTML 中的每个元素都是通过 CSS 方框模型表示的

下面还按照从外向里的顺序，给出了方框模型的另一种解释。

- 边距（margin）是元素外面的区域。它永远都没有颜色，总是透明的。
- 边框（border）扩展到元素周围，位于任何填充的外部边缘上。边框可以具有多种类型、宽度和颜色。
- 填充（padding）存在于内容周围，并且会继承内容区的背景色。
- 内容（content）被填充包围。

此时就涉及到一个比较棘手的部分：要知道元素的真正高度和宽度，就必须把方框模型的所有元素都考虑在内。回想一下上一章中的示例，尽管明确指示一个 应该是 250 像素宽和 100 像素高，那个 还是必须变得更大，以容纳使用的填充。

你已经知道如何使用宽度和高度属性设置元素的宽度和高度。下面的示例显示了如何定义一个宽 250 像素、高 100 像素的 ，并且它具有红色的背景和 1 像素的黑色边框：

```
div {
  width: 250px;
  height: 100px;
  background-color: #ff0000;
  border: 1px solid #000000;
}
```

图 14.2 显示了这个简单的 。

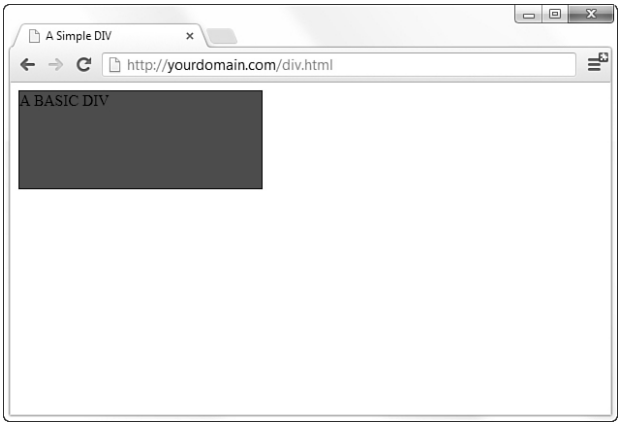



图 14.2

这是一个简单的

异步社区会员 Kensuke(18319066421) 专享 尊重版权

如果我们利用这些相同的属性定义第二个元素，但是还添加某种大小的 margin 和 padding 属性，那么将开始看到元素的大小如何变化。这是由于方框模型。

第二个 将定义如下，只给元素添加 10 像素的边距和 10 像素的填充：

```
div#d2 {
  width: 250px;
  height: 100px;
  background-color: #ff0000;
  border: 5px solid #000000;
  margin: 10px;
  padding: 10px;
}
```

图 14.3 中所示的第二个 被定义为具有与第一个 相同的高度和宽度，但是在添加了边距和填充之后，包围元素本身的整个方框的总高度和总宽度要大得多。

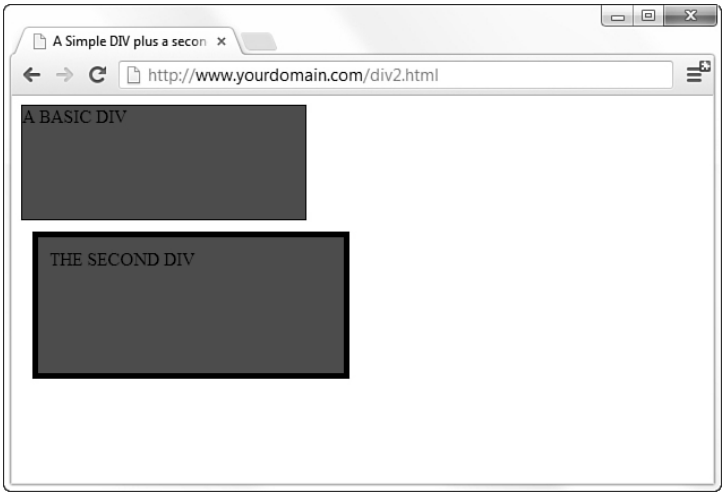



图 14.3

这被认为是另一个简单的 ，但是方框模型会影响第二个 的大小

元素的总宽度（width）是以下属性值之和：

```
width + padding-left + padding-right + border-left + border-right +
margin-left + margin-right
```

元素的总高度（height）是以下属性值之和：

```
height + padding-top + padding-bottom + border-top + border-bottom +
margin-top + margin-bottom
```

因此，第二个 的实际宽度是 300（250 + 10 + 10 + 5 + 5 + 10 + 10），实际高度是 150（100 + 10 + 10 + 5 + 5 + 10 + 10）。

**注意：**在整本书中，都在训练你使用 DOCTYPE 声明——所有的示例代码都包括一个 DOCTYPE。继续保持这种惯例，这不仅可以使你的代码经过验证，而且由于 Internet Explorer 的某些版本和 CSS 方框模型会引发一个非常具体的问题：如果没有定义 DOCTYPE，Internet Explorer 的某些版本可能会以你不想要的方式操纵元素的高度和宽度。这将导致你的布局出现浏览器不兼容性问题，因此要记得包括 DOCTYPE。

**By the Way**

目前，可以开始查看方框模型如何影响你的设计。假定你只有 250 像素的水平空间，

异步社区会员 Kensuke(18319066421) 专享 尊重版权

但是你在每一边都想要 10 像素的边距、10 像素的填充和 5 像素的边框。为了利用你所具有的显示空间来容纳想要的内容，必须把 的 width 指定为仅仅 200 像素，使得 200 + 10 + 10 + 5 + 5 + 10 + 10 等于 250 像素的可用水平空间。

模型的数学运算也很重要。在动态驱动的站点或者用户交互驱动客户端显示（比如通过 JavaScript 事件）的站点中，服务器端或客户端代码可以动态地绘制和重绘容器元素。换句话说，你的代码将产生数字，但是必须提供界限。

既然你已经理解了方框模型的工作方式，就要在本书中所做的余下工作和 Web 设计中始终牢记它。它除了能做其他事情外，还可以影响元素定位和内容流，而这些是我们接下来要探讨的两个主题。

## 14.2 关于定位的独家新闻

相对定位是 HTML 使用的默认定位类型，可以把相对定位视作类似于在西洋跳棋棋盘上布置跳棋子：跳棋子从左到右排列，当到达棋盘边缘时，就移到下一行。利用 display 样式属性的 block 值编排样式的元素会自动放置在新行上，而 inline 元素则将被放置在与它们之前的那个元素相同的行上。例如，<p>和 标签被视作是块元素，而标签则被视作是内联元素。

CSS 支持的另一种定位被称为绝对定位（absolute positioning），因为它允许设置 HTML 内容在页面上的精确位置。尽管绝对定位允许自由地精确指出元素出现的位置，但是其位置仍然是相对于出现在页面上的任何父元素的。换句话说，绝对定位允许指定元素的矩形区域相对于其元素的区域的精确位置，它与相对定位差别很大。

由于可以自由地把元素放置在页面上想要的任何位置，当一个元素占据另一个元素使用的空间时，就可能会遇到重叠问题。没有什么能阻止你指定元素的绝对位置，以使得它们重叠。在这种情况下，CSS 依靠每个元素的 Z-索引来确定哪个元素位于顶部，以及哪个元素位于底部。在本书后面将学习关于元素的 Z-索引的更多知识。目前，让我们精确地探讨如何控制样式规则是使用相对定位，还是使用绝对定位。

特定的样式规则使用的定位类型（相对定位或绝对定位）是由 position 属性确定的，它能够具有以下两个值之一：relative 或 absolute。在指定了定位的类型之后，可以使用以下属性提供特定的位置。

- left：左边的位置偏移量。
- right：右边的位置偏移量。
- top：顶部的位置偏移量。
- bottom：底部的位置偏移量。

你可能认为这些位置属性只对于绝对定位才是有意义的，但是它们实际上适用于两种类型的定位。在相对定位下，一个元素的位置被指定为相对于该元素的原始位置的偏移量。因此，如果把元素的 left 属性设置为 25px，元素的左边将将从它的原始（相对）位置移动 25 像素。另一方面，绝对定位是相对于应用样式的元素的父元素指定的。因此，如果在绝对定位下把一个元素的 left 属性设置为 25px，元素的左边出现在父元素的左边缘的

异步社区会员 Kensuke(18319066421) 专享 尊重版权

右边 25 像素处。另一方面，使用具有相同值的 right 属性将把元素定位于父元素的右边缘的右边 25 像素处。

让我们回到色块示例上来，说明定位是如何工作的。在程序清单 14.1 中，4 个色块都指定了相对定位，如图 14.4 所示，色块是垂直定位的。

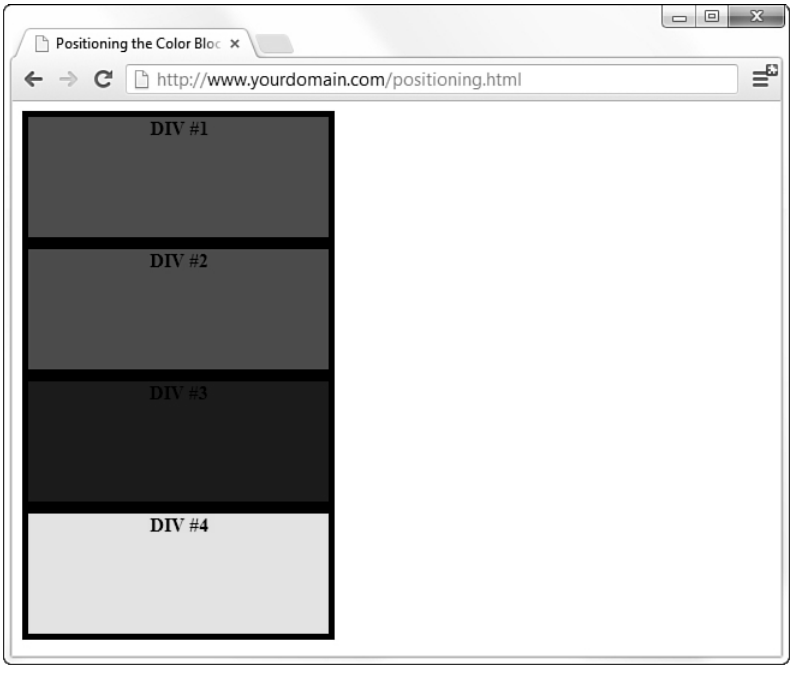



图 14.4

色块是垂直定位的，彼此堆叠在一起

### 程序清单 14.1　利用 4 个色块显示相对定位

```
<!DOCTYPE html>

<html 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;
    }
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    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>
```

用于 元素本身的样式表条目把用于 元素的 position 样式属性设置为 relative。由于其余的样式规则都继承自 样式规则，因此它们将继承它的相对定位。事实上，其他样式规则之间的唯一区别在于它们具有不同的背景色。

注意：在图 14.4 中， 元素是一个接一个地显示的，这是你利用相对定位所期望达到的效果。但是，为了使事情变得更有趣，我们在这里会执行一些特定的操作，你可以把定位类型改为绝对定位，并且明确指定颜色的放置。在程序清单 14.2 中，将把样式表条目更改为使用绝对定位，以安排色块。

**程序清单 14.2 使用色块的绝对定位**

```
<!DOCTYPE html>

<htmllang="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;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    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>
```

这个样式表把 position 属性设置为 absolute，它是样式表使用绝对定位所必需的。此外，还为每个继承的 样式规则设置了 left 和 top 属性。不过，还设置了其中每个规则的位置，使得元素彼此重叠地显示，如图 14.5 所示。

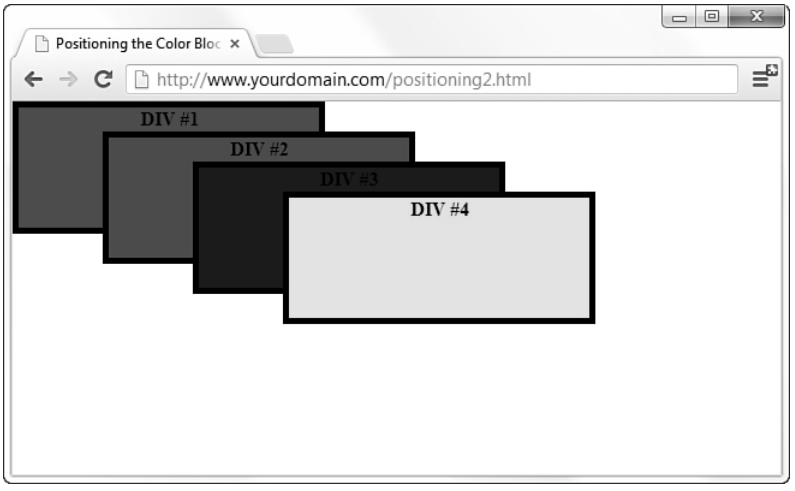



图 14.5
使用绝对定位显示色块

现在，我们讨论布局。图 14.5 显示了绝对定位怎样使你能够精确地把元素放在需要的位置。它还展现了把元素排列成重叠的效果有多容易。你可能感到好奇的是，在元素重叠时，Web 浏览器怎样知道在顶部绘制哪些元素。下一节将介绍如何控制堆叠顺序。

## 14.3 控制元素的堆叠方式

在某些情况下，你可能希望仔细控制元素在 Web 页面上彼此重叠的方式。z-index 样式属性使你能够设置元素的顺序，从而控制它们如何彼此堆叠。Z 索引（z-index）这个名称听起来可能有点奇怪，但它指的是指入计算机屏幕里面的第三维（z 轴）的概念，另外两维是横穿屏幕的 *x* 轴和在屏幕上垂直向下的 *y* 轴。考虑 Z 索引的另一种方式是考虑单独一本杂志在一堆杂志里的相对位置，其中更靠上的杂志比下面的杂志具有更高的 Z-索引。类似地，具有更高 Z 索引的重叠元素将出现在具有较低 Z 索引的元素上面。

z-index 属性用于设置一个数值，指示样式规则的相对 Z 索引。分配给 z-index 的数字只相对于样式表中的其他样式规则才是有意义的，这意味着为单个规则设置 z-index 属性并没有多大的意义。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 属性，具有更高 z-index 值的元素将出现在具有较低 z-index 值的元素上面。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**By the Way**

**注意：**无论为一个样式规则设置什么 z-index 值，利用该规则显示的元素总会出现在其父元素的上面。

程序清单 14.3 包含色块样式表和 HTML 代码的另一个版本，它使用 z-index 设置改变元素的自然重叠顺序。

**程序清单 14.3　使用 z-index 改变色块示例中的元素的显示顺序**

```
<!DOCTYPE html>

<html 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;
     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>
```

这段代码与你在程序清单 14.2 中看到的代码相比唯一的改变是：在每个编号的 div 样式类中添加了 z-index 属性。注意，第一个编号的 div 的 z-index 设置为 0，根据 Z-索引，这

异步社区会员 Kensuke(18319066421) 专享 尊重版权

应该使之成为最低的元素，而第二个 div 则具有最高的 Z-索引。图 14.6 显示了利用这个样式表显示的色块页面，它清楚地显示了 Z-索引如何影响显示的内容，并使得有可能仔细控制元素的重叠。

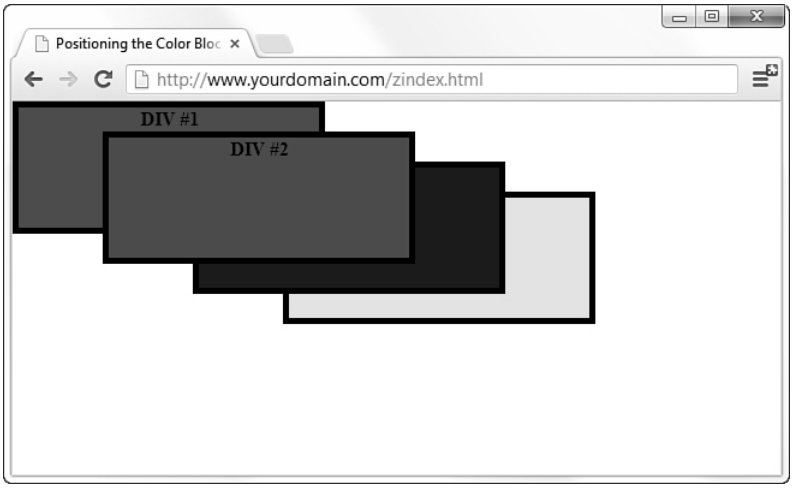



图 14.6
使用 z-index 属性改变色块的显示

尽管示例显示色块是简单的 元素，z-index 样式属性仍然可以影响任何 HTML 内容，包括图像。

## 14.4 管理文本流

既然你已经见过了一些相对于其他元素来放置元素或者绝对地放置它们的示例，现在就应该再次讨论元素周围的内容流。概念上的当前行（current line）是一个用于在页面上放置元素的不可见的行。这一行涉及页面上的元素流，当在页面的水平和垂直方向上一个接一个地排列元素时，它就可以派上用场。元素流的一部分是页面上的文本流。当把文本与其他元素（比如图像）混合在一起时，控制文本如何在那些其他的元素周围流动就很重要。

在第 13 章中，你已经见过了其中两个样式属性。下面列出了用于控制文本流的一些样式属性。

- float：确定文本如何在元素周围流动。
- clear：阻止文本在元素周围流动。
- overflow：当元素太小以至于不能包含所有的文本时，控制文本的溢出。

float 属性控制文本如何在元素周围流动。它可以设置为 left 或 right，这些值确定相对于流动的文本把一个元素定位于何处。因此，把图像的 float 属性设置为 left，将把图像定位到流动文本的左边。

上一章讲到，可以使用 clear 属性阻止文本流动到某个元素旁边，该属性可以设置为 none、left、right 或 both。它的默认值是 none，指示在流动文本时将不会对元素做任何特殊考虑。left 值将导致文本停止在元素周围流动，直到页面的左边没有元素为止。同样，right 值意味着文本不会在元素的右边流动。both 值指示文本不会在元素的任何一边流动。

overflow 属性处理溢出文本，它指的是文本在其矩形区域内放不下。如果把元素的

异步社区会员 Kensuke(18319066421) 专享 尊重版权

width 和 height 设置得太小，就可能会发生这种情况。visible 设置可以自动扩大元素，使得溢出文本能够放入其中，这是该属性的默认设置。hidden 值将使元素保持相同的大小，并且允许溢出文本从视图中隐藏起来。也许最有趣的值是 scroll，它可以给元素添加滚动条，使得可以来回移动并查看文本。

## 14.5 小结

本章开篇讨论了 CSS 方框模型，以及在把边距、填充和边框考虑在内时如何计算元素的宽度和高度，然后讨论了元素的绝对定位，并且介绍了使用 z-index 属性进行定位。本章介绍了几个出色的样式属性，使你能够控制页面上的文本流。

本章篇幅较短，但是充满了关于控制站点设计的基本信息。它值得反复阅读，并逐步完成其中的示例，以便为你的工作打下良好的基础。

## 14.6 问与答

**问：我怎样确定何时使用相对定位以及何时使用绝对定位？**

**答：**尽管对于使用相对定位或绝对定位没有固定的准则，但是一般的思想是：仅当希望对如何定位内容施加更精细的控制时，才需要绝对定位。这涉及以下事实：绝对定位使你能够把内容定位到精确的像素上，而相对定位就其定位内容的方式而言可预测性要低得多。这并不是说相对定位不能做很好的在页面上定位元素的工作，它只是意味着绝对定位更精确。当然，这也使得绝对定位潜在地更容易受到屏幕大小改变的影响，而这是你确实无法控制的。

**问：如果我没有指定两个彼此重叠的元素的 Z-索引，我怎样才能知道哪个元素将出现在上面？**

**答：**如果没有为重叠的元素设置 z-index 属性，那么在 Web 页面中后来显示的元素将出现在上面。记住这一点的容易方式是：考虑 Web 浏览器是在从 HTML 文档中读取每个元素时在页面上绘制它的，在文档后面读取的元素将绘制在前面读取的那些元素上面。

## 14.7 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 14.7.1 问题

1．相对定位与绝对定位之间的区别是什么？

2．哪个 CSS 样式属性控制元素彼此重叠的方式？

3．可以使用什么 HTML 代码精确地从浏览器窗口的左上角开始显示单词“Where would you like to”，并且精确地距离这个角左边 20 像素并向下 80 像素以大号字体显示单词

异步社区会员 Kensuke(18319066421) 专享 尊重版权

“GO TODAY?”？

### 14.7.2 答案

1. 在相对定位中，内容是依据页面流显示的，其中每个元素都物理地出现在 HTML 代码中位于它前面的元素之后。另一方面，绝对定位允许设置内容在页面上的精确位置。
2. z-index 样式属性控制元素彼此重叠的方式。
3. 可以使用下面这段代码：

```
<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>
```

### 14.7.3 练习

- 通过创建一系列具有不同边距、填充和边框的元素，练习处理 CSS 方框模型的复杂状况，并且查看这些属性如何影响它们的高度和宽度。
- 找到一组你喜欢的图像，并且使用绝对定位，甚至可能还要使用一些 z-index 值，在一种图库中排列它们。在放置图像时，要尝试使它们构成某种设计（如正方形、三角形或圆形）。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 15 章

# 创建固定布局或流动布局

在本章中你将学到：

- 固定布局的工作方式；
- 流动布局的工作方式；
- 怎样创建固定/流动混合布局；
- 怎样考虑并开始实现一种响应性设计。

迄今为止，你已经学习了关于编排 Web 内容的样式的许多知识，从字体大小和颜色到图像、块元素、列表等。但是，我们还没有讨论的是页面布局的高级概述。一般来讲，有两种类型的布局——固定布局和流动布局。但是也有可能组合使用这两种布局，其中一些元素是固定的，另外一些元素则是流动的。

在本章中，你首先将学习这两类布局的特征，并且查看几个使用它们的 Web 站点的示例。在本章末尾，你将看到一个组合了两类布局的元素的基本模板。最终，你所选择的布局类型取决于你自己——只要你的站点遵循 HTML 和 CSS 标准，那么它就不容易出错。

Try It Yourself：查找你喜欢的布局示例

提供流动布局示例的一个良好的位置是 WordPress Theme Gallery（http:// wordpress.org/themes/）。WordPress 开始时是一个博客平台，但是近年来日益被人们作为一种非博客内容或站点管理工具使用。WordPress Theme Gallery 显示了数百个固定宽度和流动布局的示例，它们可以给你创建的内容提供某种思想（如果不是全部代码的话）。作为本书中的练习的一部分，即使你不是在处理 WordPress 博客或站点，也可以查看模板库，并与许多设计变体交互。

花一些时间查看 WordPress 示例，也许还要查看 http://www.csszengarden. com/上的“CSS 禅意花园”（CSS Zen Garden）。这有助于你感受自己喜爱的布局类型，而不会受到布局中的内容影响。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 15.1 理解固定布局

固定布局（或固定宽度的布局）就是指其中的页面主体被设置为特定宽度的布局，这个宽度通常受包含所有内容的主“包装器”元素控制。如果给包装器元素（比如一个 ）提供了一个 ID 值（比如 main 或 wrapper（尽管这个名称由你自己确定）），那么就可以在样式表条目中设置包装器元素的 width 属性。

在创建固定宽度的布局时，最重要的决策是确定你希望包容的最低屏幕分辨率。多年来，800×600 像素成为了 Web 设计师的“最小公分母”，导致典型的固定宽度大约是 760 像素。不过，对于非移动浏览器，使用 800×600 像素的屏幕分辨率的人数现在不到 4%。鉴于此，许多 Web 设计师考虑将 1024×768 像素作为当前的最低屏幕分辨率，因此如果他们创建固定宽度的布局，固定宽度通常在 800~1000 像素之间。

创建固定宽度布局的主要原因是：使得可以精确控制内容区域的外观。不过，如果用户在访问你的固定宽度的站点时使用的屏幕分辨率比你设计站点时所牢记的分辨率更小或者大得多，那么他们将会遇到滚动条（如果他们的分辨率更小的话）或者大量的空白（如果他们的分辨率更大的话）。今天，在大多数流行的 Web 站点中很难找到固定宽度的布局，因为站点设计师知道他们需要迎合可能最大范围的受众（因此不会对浏览器大小做出假定）。不过，固定宽度的布局仍然被广泛采用，尤其是被使用具有严格模板的内容管理系统的站点管理员采用。

下面的图形显示了这样一个站点，它用于圣何塞州立大学（大学的 Web 站点通常使用严格的模板和内容管理系统，因此这是一个很容易找到的示例），具有一个宽度固定为 960 像素的包装器元素。如图 15.1 所示，浏览器窗口在 960 像素的宽度下有少许内容会被遮挡起来。在图像的右边，挡住了一些重要的内容（在图形底部，在浏览器中出现了一个水平滚动条）。

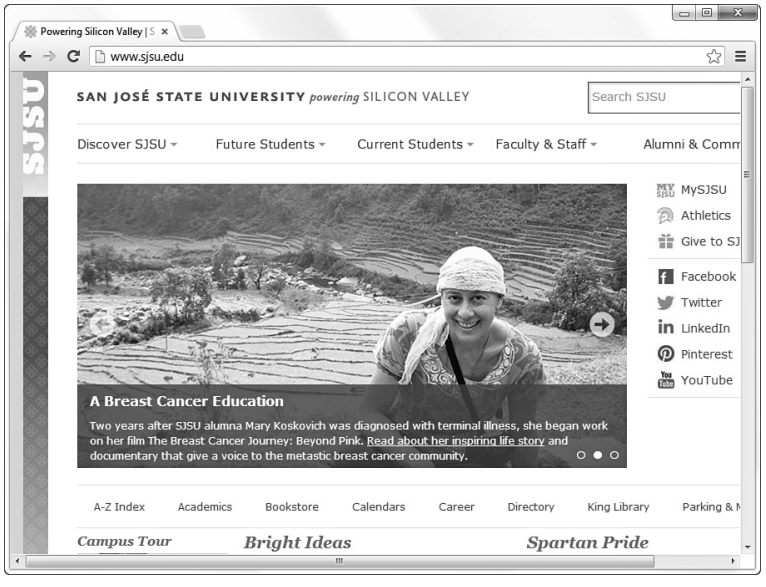



图 15.1

在较小屏幕中的固定宽度的示例

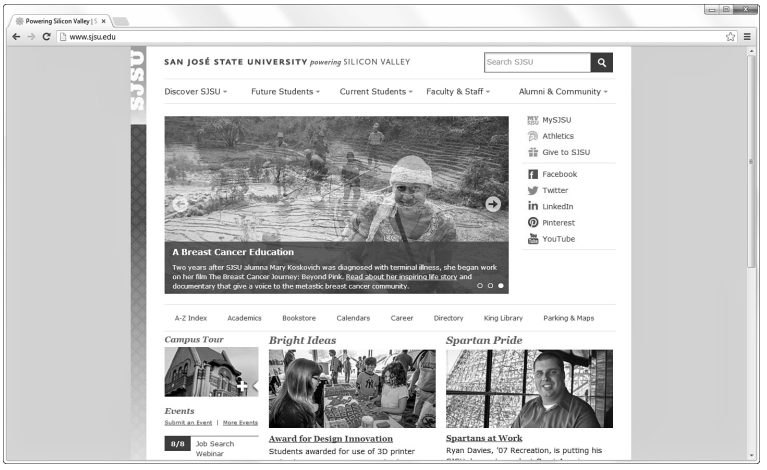
不过，图 15.2 显示了当浏览器窗口的宽度大于 1400 像素时站点的外观：你会在主体内

异步社区会员 Kensuke(18319066421) 专享 尊重版权

容两边看到许多空白（或“实际空间”），一些人认为它们不美观，令人生厌。

图 15.2

在较大屏幕中的固定宽度的示例




除了首先决定创建固定宽度的布局之外，还要确定是使固定宽度的内容左对齐还是居中放置它。使内容左对齐将只在右边产生额外的空间；使内容区域居中则会在两边都产生额外的空间。不过，居中至少可以提供平衡，而左对齐设计最终可能看上去像一个较小的矩形挤入浏览器的角落，这依赖于用户显示器的大小和分辨率。

## 15.2 理解流动布局

流动布局——也称为流式布局（fluid layout）——是指页面的主体不使用指定宽度（以像素为单位）的布局，尽管它可能封闭在使用百分比宽度的主“包装器”元素中。流动布局背后的思想是：它可以非常有用，并且即使用户具有非常小或非常宽的屏幕，它仍然可以保持总体的设计美感。

图 15.3、图 15.4 和图 15.5 显示了 3 个实际应用的流动布局的示例。

图 15.3

在相对较小的屏幕中查看时的流动布局




异步社区会员 Kensuke(18319066421) 专享 尊重版权

在图 15.3 中，浏览器窗口大约是 770 像素宽。这个示例在水平滚动条出现之前显示了适当的最小屏幕。事实上，直到浏览器宽度达到 735 像素，才会出现滚动条。另一方面，图 15.4 显示了一个非常小的浏览器窗口（宽度小于 600 像素）。

在图 15.4 中，可以看到一个水平滚动条。在页面内容的标头区域，标志图形正开始接替文本，并出现在它们的顶部。但是页面的大部分内容仍然相当有用。页面左边的信息性内容仍然清晰易读，并且与右边的输入表单共享可用的空间。

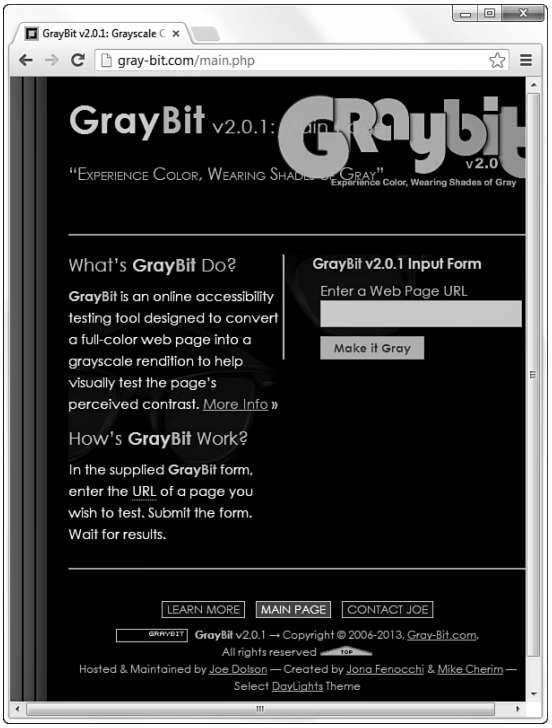



图 15.4

在非常小的屏幕中查看时的流动布局

图 15.5 显示了在非常宽的屏幕中查看这个相同页面的效果。图中浏览器窗口的宽度大约是 1330 像素。有大量的空间可以让页面上的所有内容伸展开。之所以能够实现这种流动布局，是因为所有的设计元素都具有指定的百分比宽度（而不是固定宽度）。这样，布局就可以利用所有可用的浏览器实际空间。

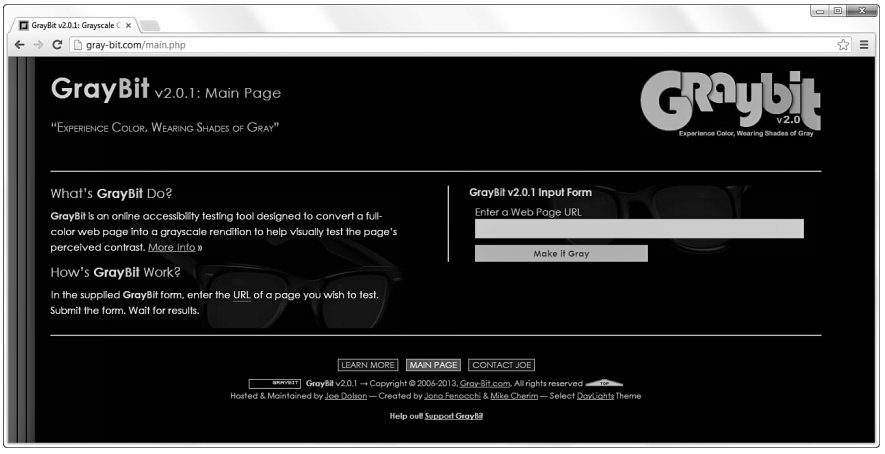



图 15.5

在宽屏幕中查看时的流动布局

异步社区会员 Kensuke(18319066421) 专享 尊重版权

初看上去，流动布局方法似乎像是最佳的方法，毕竟谁不希望利用所有可用的屏幕实际空间呢？但是，充分利用空间与不允许内容呼吸之间有一条明显的界线。太多的内容会令人窒息，而开阔的空间中如果内容不足则会令人感觉平淡。

纯粹的流动布局可能给人留下相当深刻的印象，但它需要进行大量的测试，以确保它在广泛的浏览器中和变化的屏幕分辨率下都是有用的。你可能没有时间和精力创建这样的设计，在这种情况下，合理的折衷是实现固定/流动混合布局，或者一种完全响应性的设计，在本章后面将会学到它。

## 15.3 创建固定/流动混合布局

创建固定/流动混合布局是指包含两类布局元素的布局。例如，你可以具有一种流动布局，它在主体区域内包括固定宽度的内容区域，或者作为锚点元素（比如左边的栏或者顶部的导航条）。你甚至可以创建一个固定的内容区域，使其像框架一样工作，其中内容区域将保持固定，甚至在用户滚动经过内容时也是如此。

### 15.3.1 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，但它在主体区域的两侧具有两个固定宽度的栏（主体区域是第三栏，如果你这样认为的话，那么它只是比其他两栏要宽得多）。模板还具有清楚划定的标头和脚注区域。程序清单 15.1 显示了这种布局的基本 HTML 结构。

**程序清单 15.1 基本的固定/流动混合布局结构**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Sample Layout</title>
    <link href="layout.css" rel="stylesheet" type="text/css" />
  </head>

  <body>
     <header>HEADER</header>
     <div id="wrapper">
       <div id="content_area">CONTENT</div>
       <div id="left_side">LEFT SIDE</div>
       <div id="right_side">RIGHT SIDE</div>
       </div>
       <footer>FOOTER</footer>
  </body>
</html>
```

首先，注意：利用<link>标签链接到用于这种布局的样式表，而不是将其包括在模板中。由于模板将用于多个页面，你将希望能够以尽可能组织有序的方式控制模板的元素显示。这意味着只能在一个位置（样式表中）更改那些元素的定义。

接下来，注意最基本的 HTML 代码。说实话，这个基本的 HTML 结构可以用于固定布

异步社区会员 Kensuke(18319066421) 专享 尊重版权

局、流动布局或者你在这里看到的固定/流动混合布局，因为使之成为固定、流动或混合布局的所有实际的样式设置都发生在样式表中。

**注意：**我在这个示例中使用了具有命名标识符的元素代替语义元素，比如<section>或<nav>，因为我在以尽可能简单的方式演示要点，而没有受内容本身的束缚。不过，如果你知道左边的 将用于保留导航，就应该使用<nav>标签，而不要使用其 id 为 left_side 的 元素。

**By the Way**

对于程序清单 15.1 中的 HTML 结构，你实际上具有想包括在站点中的内容区域的标识。这种规划对于任何开发都是至关重要的，在考虑将要使用的布局类型之前，必须知道想要包括什么，更不用说将应用于该布局的特定样式了。

在这个阶段，layout.css 文件只包括下面这个条目：

```
body {
   margin:0;
   padding:0;
}
```

如果查看程序清单 15.1 中的 HTML 代码，并自言自语地说道：“但是那些 元素将不带任何样式地彼此堆叠起来”，你是正确的。如图 15.6 所示，这里没有提及任何布局。

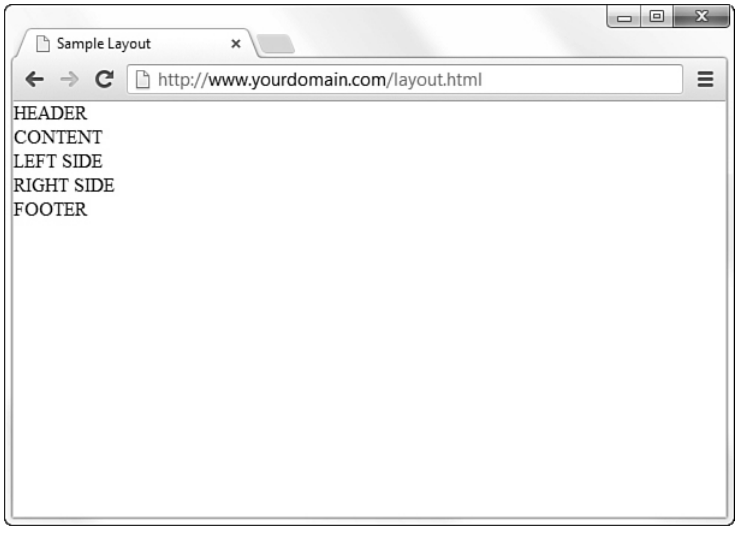



图 15.6

一个基本的 HTML 模板，其中没有对容器元素应用任何样式

## 15.3.2 在固定/流动混合布局中定义两栏

我们可以从容易的事情开始。由于这个布局被指望是流动的，我们知道放在标头和脚注区域中的任何内容都将扩展到浏览器窗口的宽度，而无论该窗口可能有多窄或者有多宽。

向样式表中添加以下代码，给标头和脚注区域都提供 100%的宽度，以及相同的背景色和文本颜色：

```
header, footer {
  float: left;
  width: 100%;
  background-color: #7152f4;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
  color: #ffffff;
}
```

现在，事情变得有点棘手了。我们必须在页面的两边定义两个固定的栏，还要在页面中间定义一栏。在这里使用的 HTML 代码中，注意一个名为 wrapper 的 元素，它包围了这两栏，在样式表中将这个元素定义如下：

```
#wrapper {
  float: left;
  padding-left: 200px;
  padding-right: 125px;
}
```

两个填充定义实质上为页面左、右两边的两个固定宽度的栏预留空间。左栏的宽度将是 200 像素，右栏的宽度则是 125 像素，并且每一栏都将具有不同的背景色。但是，如果 HTML 仍然没有编排样式（见图 15.6），我们还必须相对于将放置项目的位置来定位它们。这意味着向用于其中每一栏的样式表条目中添加 position:relative。此外，我们还指示 元素应该浮动到左边。

但是，就 left_side 而言，我们还指示我们希望最右边的边距边缘距离边缘在 200 像素内（这包括了将被定义为 200 像素宽的栏）。我们还希望左边的边距是一个完全为负的边距，这将把它拖入合适的位置（你稍后将会看到）。right_side 没有包括一个用于 right 的值，但它确实在右边包括一个负的边距：

```
#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;
}
```

此时，让我们还定义内容区域，使得它具有白色的背景，占据 100%的可用区域，并且相对于其位置浮动到左边：

```
#content_area {
  position: relative;
  float: left;
  background-color: #ffffff;
  width: 100%;
}
```

此时，基本的布局看起来应该如图 15.7 所示，并且清楚地描绘了一些区域。

不过，如果把窗口调整到某个宽度以下，这个模板就会出现一个问题。由于左栏的宽度是 200 像素，右栏的宽度是 125 像素，并且你希望内容区域中至少会有一些文本，可以想象：如果窗口的宽度只有 350~400 像素，那么这个页面将会破坏。在下一节中将

异步社区会员 Kensuke(18319066421) 专享 尊重版权

处理这个问题。

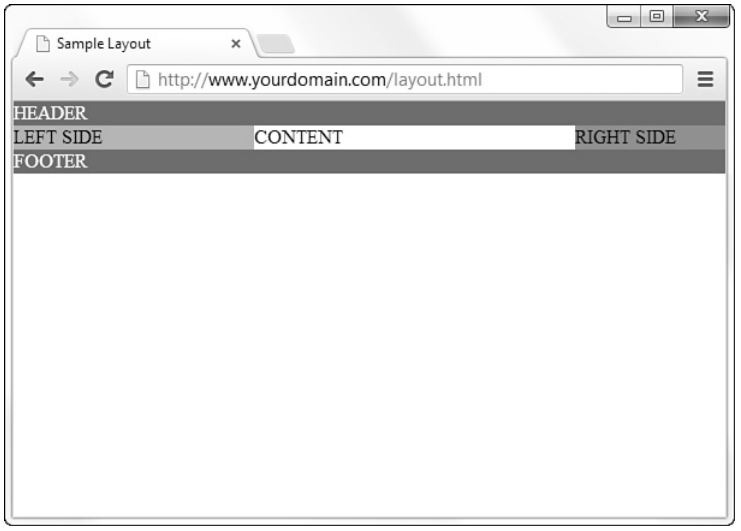



图 15.7

添加一些样式后的基本 HTML 模板

## 15.3.3 设置布局的最小宽度

尽管用户不太可能利用显示宽度在 400 像素以下的桌面浏览器访问你的站点，但是这个示例在本章的界限内展示了其特定的目的。你可以进行推断并广泛地应用该信息：甚至在固定/流动混合站点中，在某个时间，你的布局也会崩溃，除非对它进行某些事情。

要做的“某些事情”之一是使用 CSS min-width 属性。该属性用于设置元素的最小宽度，不包括填充、边框或边距。图 15.8 显示了当把 min-width 属性应用于<body>元素时所发生的事情。

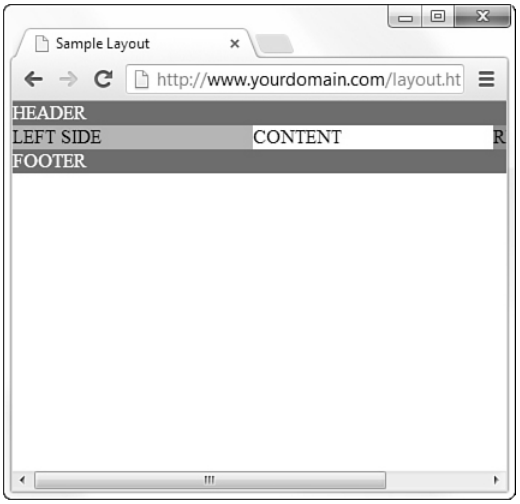



图 15.8

调整到 400 像素以下的基本 HTML 模板，并且应用了最小宽度

图 15.8 显示了在滚动到右边后的右栏的一小部分，但是当布局调整到最小宽度以下时它并没有支离破碎。在这里，最小宽度是 525 像素：

```
body {
  margin: 0;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
  padding: 0;
  min-width: 525px;
}
```

在这个示例中会显示水平滚动条，因为浏览器窗口本身的宽度小于 500 像素。当窗口的宽度稍大于 525 像素时，滚动条将会消失。

## 15.3.4 在固定/流动混合布局中处理栏高度

除了一个问题之外，这个示例一切都好，这个问题就是：它没有内容。当把内容添加到不同的元素中时，更多的问题出现了。如图 15.9 所示，栏将变得足够高，以容纳它们所包含的内容。

图 15.9
栏变得与它们所包含的内容一样高




由于不能指望用户的浏览器具有特定的高度或者内容总是具有相同的长度，你可能认为这将导致固定/流动混合布局出现一个问题。事实并非如此。如果稍微深入考虑一下，就可以应用另外几种样式，把各个部分组合在一起。

**By the Way**

**注意：**由于我们超越了基本的布局示例，并且我还冒失地从标头和脚注中删除了背景色和文本颜色属性，这就是为什么这个示例不再在非常暗的背景上显示白色文本的原因。此外，我还在<footer>元素中居中显示了文本，它现在具有浅灰色背景。

首先，在用于 left_side、right_side 和 content_area id 的样式表条目中添加以下声明：

```
margin-bottom: -2000px;
padding-bottom: 2000px;
```

这些声明添加了荒唐的填充，并给所有这 3 个元素的底部分配了太大的边距。你还必

异步社区会员 Kensuke(18319066421) 专享 尊重版权

须给样式表中的脚注元素定义添加 position:relative，使得尽管有这种荒唐的填充，脚注仍然是可见的。

此时，页面看上去如图 15.10 所示，尽管仍然不是我们所想要的，但是更接近了。

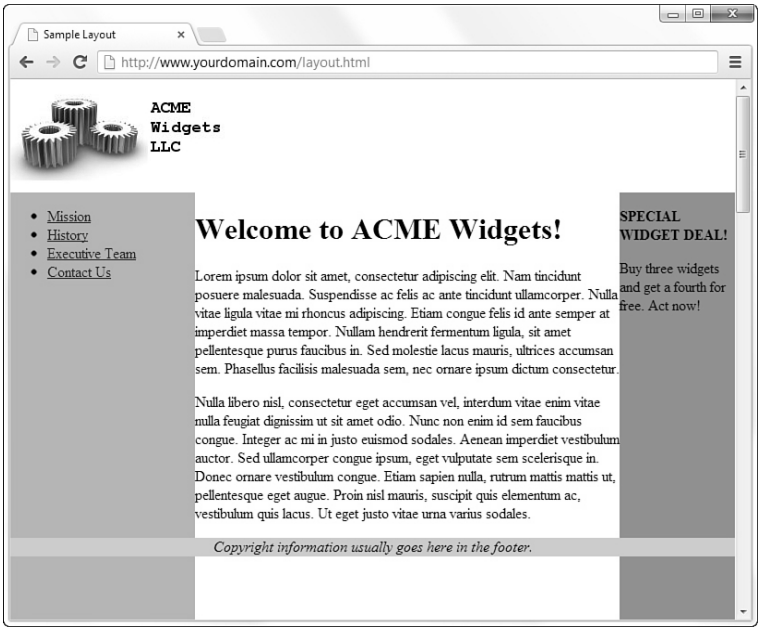



图 15.10

不管栏中有多少内容，彩色区域现在是可见的

为了裁剪掉所有多余的颜色，可以为包装器 ID 向样式表中添加以下代码：

```
overflow: hidden;
```

图 15.11 显示了最终的结果：一种固定宽度/流动混合布局，并且具有必要的栏间距。我还随意地编排了导航链接的样式，并且调整了欢迎消息周围的边距。在程序清单 15.3 中可以查看完整的样式表。

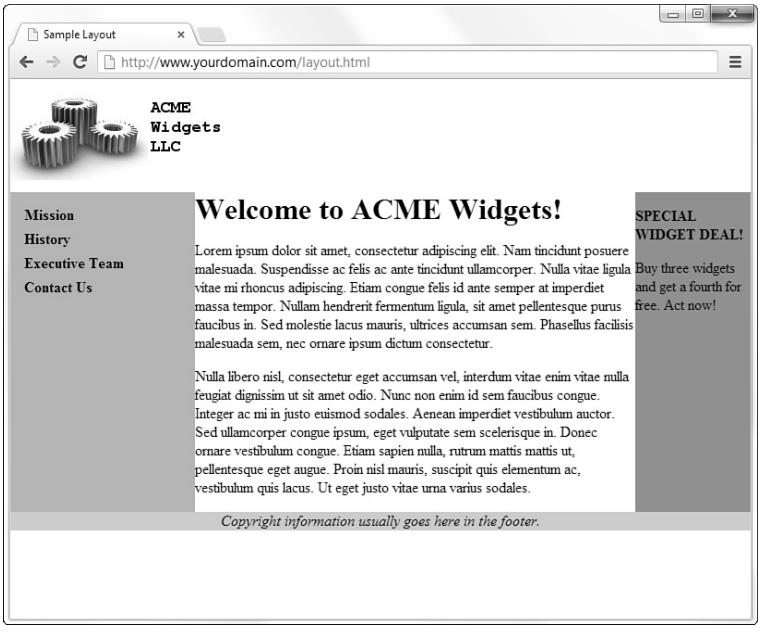



图 15.11

祝贺！它是固定宽度/流动混合布局（尽管你希望对那些颜色进行一些处理）

异步社区会员 Kensuke(18319066421) 专享 尊重版权

完整的 HTML 代码出现在程序清单 15.2 中，程序清单 15.3 则显示了最终的样式表。

**程序清单 15.2 基本的固定/流动混合布局结构（带有内容）**

```
<!DOCTYPE html>

<html lang="en">
  <head>
      <title>Sample Layout</title>
      <link href="layout.css" rel="stylesheet" type="text/css" />
  </head>

  <body>
    <header><img src="acmewidgets.jpg" alt="ACME Widgets
       LLC"/></header>
    <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.</p>
        <p>Nulla libero nisl, consectetur eget accumsan vel, interdum
        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 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">
         <p><strong>SPECIAL WIDGET DEAL!</strong></p>
         <p>Buy three widgets and get a fourth for free. Act now!</p>
       </div>
    </div>
    <footer>
    Copyright information usually goes here in the footer.
    </footer>
  </body>
</html>
```

**程序清单 15.3 用于固定/流动混合布局的完整样式表**

```
body {
  margin:0;
  padding:0;
  min-width: 525px;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
}

header {
  float: left;
  width: 100%;
}

footer {
  position:relative;
  float: left;
  width: 100%;
  background-color: #cccccc;
  text-align:center;
  font-style: italic;
}

#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%;
  margin-bottom: -2000px;
  padding-bottom: 2000px;
}

#right_side {
  position: relative;
  float: left;
  width: 125px;
  background-color: #f452d5;
  margin-right: -125px;
  margin-bottom: -2000px;
  padding-bottom: 2000px;
}

#content_area {
  position: relative;
  float: left;
  background-color: #ffffff;
  width: 100%;
  margin-bottom: -2000px;
  padding-bottom: 2000px;
}

h1 {
  margin: 0;
}

#left_side ul {
  list-style: none;
  margin: 12px 0px 0px 12px;
  padding: 0px;
}
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
#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;
}
```

## 15.4 考虑响应性 Web 设计

2010 年，Web 设计师 Ethan Marcotte 创造了术语响应性 Web 设计（responsive web design），它指的是一种 Web 设计方法，构建在你刚才学过一点的流式设计的基础之上。响应性 Web 设计的目标是：无论用于查看 Web 内容的设备类型和大小是什么，都很容易查看、阅读和导航它们。换句话说，计划创建响应性 Web 站点的设计师正在这样做，以确保在大型桌面显示器、小智能手机或中型平板电脑上查看站点的受众可以获得相似的享受和有用性。

响应性设计基于流式（流动）网格布局，与你在本章前面学过的非常相似，但是做了几处修改和增补。首先，那些网格布局总是应该采用相对单位（而不是绝对单位）。换句话说，设计师应该使用百分比（而不是像素）来定义容器元素。

第二，所有的图像都应该是灵活的（这是我们在以前的章节中没有讨论的）。对于这一点，我的意思是不要为每一幅图像使用特定的高度和宽度，而要使用相对百分比，使得图像总会显示在包含它们的（相对大小的）元素内。

最后，花一些时间为每种媒体类型开发特定的样式表，并且使用媒体查询基于媒体类型利用这些不同的规则。在第 20 章中，将学习关于特定于媒体的样式表的更多知识，但是一般来讲，要知道可以像下面这样指定样式表的链接：

```
<link rel="stylesheet" type="text/css"
   media="screen and (max-device-width: 480px)"
   href="wee.css" />
```

在这个示例中，media 属性包含一种类型和一个查询：类型是 screen，查询部分是(max-device-width: 480px)。这意味着如果尝试呈现内容的设备具有一个屏幕（与（比如说）打印机相对，在第 20 章将会看到它），并且水平分辨率（设备宽度）小于 480 像素宽——与智能手机一样，那么就加载名为 wee.css 的样式表，并且使用在其中找到的规则呈现内容。

当然，本书中的几个短段落并不能证明响应性 Web 设计是完全合理的。在自己打下 HTML5 和 CSS3 的坚实基础后，强烈建议阅读 Marcotte 的图书 *Responsive Web Design*

异步社区会员 Kensuke(18319066421) 专享 尊重版权

（http://www.abookapart.com/products/responsive-web-design）。此外，在第 23 章中讨论的几个 HTML 和 CSS 框架也利用了响应性设计的原理，它们构成了一个用于构建响应性站点的非常好的起点，并利用流式网格、图像调整大小和媒体查询对其进行修修补补，使之能够实现这个目标。

## 15.5 小结

本章讲解了 3 种主要的布局类型（固定布局、流动布局和固定/流动混合布局）的一些实用的示例。在 15.3 节中，我们看到了一个扩展的示例，它引领我们通过了创建一种固定/流动混合布局的过程，其中 HTML 和 CSS 都正确地经过了验证。记住，创建布局的最重要的部分是搞清楚你认为可能需要在设计中考虑到的内容区域。

最后，本章介绍了响应性 Web 设计的概念，这个主题本身足够编写一本书。这里提供了相关的简要信息，比如使用流式网格布局、响应性图像和媒体查询，这样读者就具有一些基本的概念，可以自己开始进行测试。

## 15.6 问与答

**问：我听说过弹性布局（elastic layout）的概念，它与流动布局有什么区别？**

**答：**弹性布局是指当用户调整文本的大小时其内容区域也会调整大小的布局。弹性布局使用 em（“字母 m 的宽度”），它固有地与文本和字体大小成正比。em 是一个印刷度量单位，等于当前字体的磅大小。当在弹性布局中使用 em 时，如果用户使用 Ctrl 键和鼠标滚轮强制增加或减小文本大小，那么包含文本的区别也会成比例地增加或减小。弹性布局通常实现起来相当困难。

**问：你花了许多时间讨论流动布局和混合布局，它们比纯粹的固定布局更好吗？**

**答：**“更好”是一个主观用语。在本书中，关注的是符合标准的代码。大多数设计师将告诉你滚动布局要花更长的时间创建（和完善），但它可以增强有用性，因此值得这样做，尤其是当它可以导致响应性设计时。那么何时不值得这样做呢？如果你的客户没有意见，并且按统一费率（而不是按小时）给你付费，就不需要采用这种布局。在这种情况下，你只是为了展示自己的技能（不过，对你来说可能值得这样做）。

## 15.7 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 15.7.1 问题

1．一般来讲，哪种布局是要使用的最佳布局：固定布局、流动布局或混合布局？

2．你能够在页面上的任意位置定位固定布局吗？

异步社区会员 Kensuke(18319066421) 专享 尊重版权

3．min-width 的作用是什么？

## 15.7.2 答案

1．这是一个难以回答的问题；没有所谓的“最佳”布局。它依赖于你的内容和受众的需求（注意：作者最终倾向于流动布局和响应性布局是“最佳”的，不过，如果你有时间和资源这样做的话）。

2．当然可以。尽管大多数固定布局都是左对齐或居中的，也可以在 *x*、*y* 轴上分配一个固定位置，可以在那里放置一个 ，包括所有其他的布局 。

3．min-width 属性用于设置元素的最小宽度，不包括填充、边框或边距。

## 15.7.3 练习

- 图 15.11 显示了完成的固定/流动混合布局，但是要注意几个要改进的区域：右栏中的文本周围没有任何空白，并且主体文本与任何一个栏之间没有任何边距，此外脚注条有一点稀疏，等等。花一些时间修正这些设计元素。
- 使用上一个练习中修正的代码，尝试使之具有响应性，并且只使用你在本章中学到的简短信息。要使模板可以在智能手机或其他小设备上查看，仅仅把容器元素转换成相对大小应该要走一段很长的路，但是媒体查询和替代样式表肯定不会对其产生不良影响。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 16 章

# 使用 CSS 进一步处理列表

在本章中你将学到：

- CSS 方框模型怎样影响列表；
- 怎样自定义列表项指示符；
- 怎样使用列表项和 CSS 创建图像映射。

第 5 章介绍了 3 种类型的 HTML 列表。第 13 章介绍了元素的边距、填充和对齐。在本章中，你将学习如何对不同类型的 HTML 列表应用边距、填充和对齐样式，纯粹以 HTML 和 CSS 帮助你产生一些强大的设计元素。

确切地讲，你将学习如何修改列表元素的外观——超越了使用你在第 5 章中学过的 list-style-type 属性。你将把迄今为止学过的许多 CSS 样式应用到实践中，并且你在本章中获得的知识将使你能够把列表用于更多的地方，而不仅仅用于展示一组带有项目符号或编号的列表项。

## 16.1 HTML 列表复习

如你在第 5 章所学过的，HTML 列表具有 3 种基本类型。其中每种列表都基于其类型和环境，以一种稍微不同的方式展示内容。

- 有序列表（ordered list）是一种在每个列表项前显示数字或字母的缩进列表。有序列表被<ol>和</ol>标签包围，并且列表项封闭在<li></li>标签对中。这类列表通常用于显示带编号的步骤或内容的层级。
- 无序列表（unorderedlist）是一种在每个列表项前显示项目符号或其他符号的缩进列表。无序列表被<ul>和</ul>标签包围，并且列表项封闭在<li></li>标签对中。这类列表通常用于提供视觉提示，简述下面将要介绍的具体信息。
- 定义列表（definition list）通常用于显示列表项及其含义，从而在列表本身的上下

异步社区会员 Kensuke(18319066421) 专享 尊重版权

文内提供信息层次结构——与有序列表非常像，但是没有编号。定义列表被<dl>和</dl>标签包围，并且<dt>和</dt>标签封闭列表项，而<dd>和</dd>标签则封闭定义。

当内容需要时，可以嵌套有序和无序列表，或者把列表放在其他列表内。嵌套列表将产生一种内容层次结构，因此要把它们留用于内容实际地具有一种你想显示的层次结构时（比如内容大纲或目录）。或者，如你以后将学到的，当站点导航系统包含子导航元素时，就可以使用嵌套列表。

## 16.2 CSS 方框模型如何影响列表

与列表相关的特定样式包括：list-style-image（用于放置一幅图像作为列表项标记）、list-style-position（指示把列表项标记放在什么位置）和 list-style-type（列表项标记本身的类型）。但是，通过这些样式控制列表和列表项的结构，可以使用 margin、padding、color 和 background-color 样式利用列表实现甚至更具体的显示效果。

**By the Way**

**注意：**一些老式的浏览器以不同的方式处理边距和填充，尤其是在列表和列表项周围。不过，在编写本书时，本章及本书其他章中的 HTML 和 CSS 在主流 Web 浏览器（Apple Safari、Google Chrome、Microsoft Internet Explorer、Mozilla Firefox 和 Opera）的当前版本中都是完全相同地显示的。当然，在把你的 Web 内容发布到网上之前，仍然应该在所有的浏览器中复查它们，但是“修改”样式表以包容浏览器独特的呈现特点的要求正日益减弱。

在第 13 章中，你学习了每个元素在元素的内容与边框之间都具有某种填充，你还学习了在元素的边框与任何其他的内容之间都具有边距。对于列表也是归路经，在编排列表的样式时，必须记住“列表”实际上由两个元素组成：父列表元素类型（<ul>或<ol>）和各个列表项本身。其中每个元素都具有可以被样式表影响的边距和填充。

本章中的示例显示了不同的 CSS 样式如何影响 HTML 列表和列表项的视觉显示。在本章中练习处理列表时，要牢记这些基本的区别，并且你将能够使用列表在站点导航系统内实现高级视觉效果。

程序清单 16.1 创建一个包含 3 个列表项的基本列表。在这个程序清单中，给无序列表本身（<ul>）提供了蓝色的背景、黑色的边框和 100 像素的特定宽度，如图 16.1 所示。列表项（各个<li>）则具有灰色的背景和黄色的边框。列表项文本和指示符（项目符号）是黑色的。

**程序清单 16.1 利用颜色和边框样式创建基本的列表**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>List Test</title>
    <style type="text/css">
      ul {
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
        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 利用颜色和边框编排列表和列表项的样式

如图 16.1 所示，<ul>创建一个方框，并把各个列表项置于其中。在这个示例中，整个方框都具有蓝色的背景。但是还要注意各个列表项（在这个示例中，它们使用灰色的背景和黄色的边框）没有扩展到由<ul>创建的方框的左边缘。

**注意：**可以通过创建一个简单的测试文件，比如图 16.1 中的文件，然后在样式表中向 ul 选择器的声明中添加 padding-left: 40px;，测试不同的浏览器所显示的默认 padding-left 值。如果重新加载页面并且显示没有改变，你就知道测试浏览器使用 40 像素作为 padding-left 的默认值。

**By the Way**

这是由于浏览器给<ul>的左边自动添加了某种填充。浏览器不会给边距添加填充，因为它将出现在方框外面四周。它们把填充添加在方框里面，并且只添加到左边。这个填充值大约是 40 像素。

无论列表的类型是什么，左边的默认填充值仍然保持相同。如果向样式表中添加以下一行代码，创建一个没有列表项指示符的列表，将会看到填充仍然保持相同，如图 16.2 所示：

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
list-style-type: none;
```

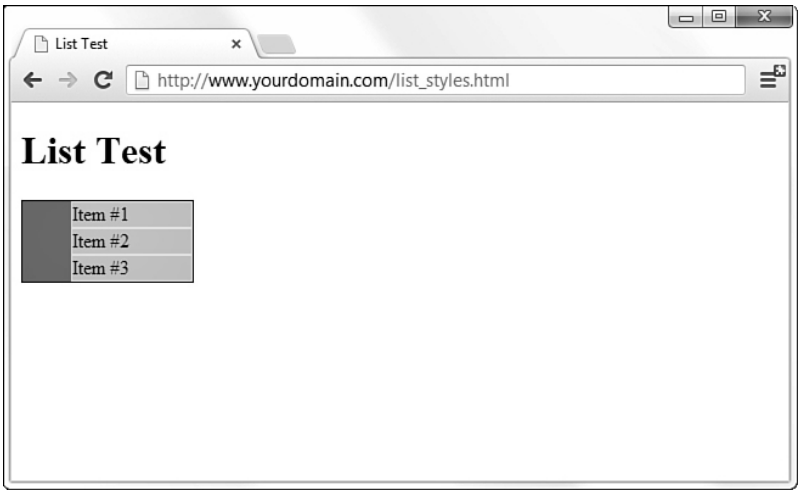



图 16.2

无论是否具有列表项指示符，左边的默认填充都将保持相同

在创建包含任何类型的列表的页面布局时，都要设置填充以把列表项“井井有条”地放在页面上。类似地，仅仅由于没有默认的边距与列表相关联并不意味着不能给显示的内容指定某种边距，向 ul 选择器的声明中添加 margin 值可以提供额外的布局控制。

但是要记住，迄今为止我们处理的只是列表定义本身，我们还没有处理对各个列表项应用的样式。在图 16.1 和图 16.2 中，列表项的灰色背景和黄色边框没有显示默认的填充或边距。通过对列表项（而不是对整个列表“方框”本身）应用填充或边距值可以创建不同的效果，如图 16.3 所示。

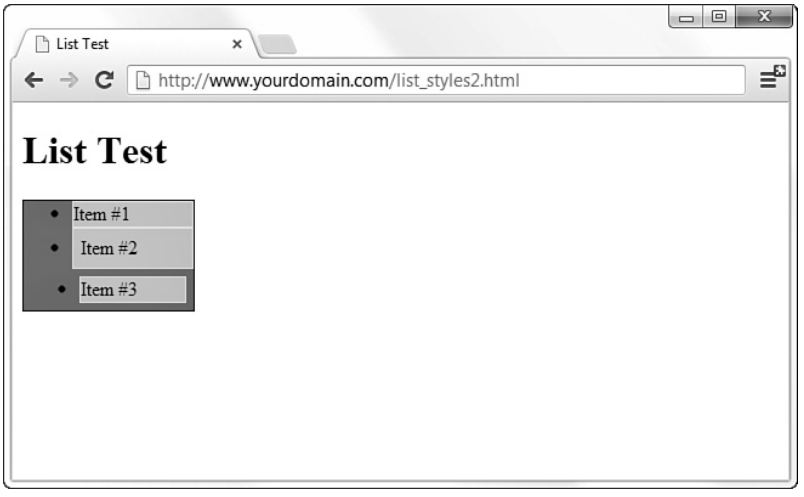



图 16.3

不同的值将会影响列表项的填充和边距

第一个列表项是基本列表项，没有对它应用填充或边距。不过，第二个列表项使用了一个名为 padded 的类，在样式表中将其定义为 padding: 6px，在每一边都可以看到 6 像素的填充（在内容与包围元素的黄色边框之间）。注意：项目符号的放置仍然保持与第一个列表项的放置方式相同。第三个列表项使用了一个名为 margined 的类，在样式表中将其定义为 margin: 6px，以在列表项周围应用 6 像素的边距，这个边距允许<ul>的蓝色背景显示出来。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 16.3 放置列表项指示符

上述关于边距和填充的所有讨论引发了另一个问题：列表项指示符（在使用时）的控制以及文本应该如何环绕在它们周围（或者不环绕）。list-style-position 属性的默认值是 outside，这种放置方式意味着项目符号、编号及其他指示符将保持在由<li></li>标签对创建的方框外面的文本左边。当文本在列表项内换行时，它将在该方框内换行，并且保持左对齐到元素的左边框。

但是，当 list-style-position 的值是 inside 时，指示符位于由<li></li>标签对创建的方框内。不仅列表项指示符随后将进一步缩进（它们实质上变成了文本的一部分），而且文本将在每个列表项指示符底下换行。

图 16.4 显示了外部和内部列表样式位置的示例。程序清单 16.1 与用于产生图 16.4 中的示例的代码之间的唯一变化（不包括添加给“Item #2”和“Item #3”的填充文本）是第二个列表项使用了一个名为 outside 的类，它在样式表中被定义为 list-style-position: outside，并且第三个列表项使用了一个名为 inside 的类，它在样式表中被定义为 list-style-position: inside。

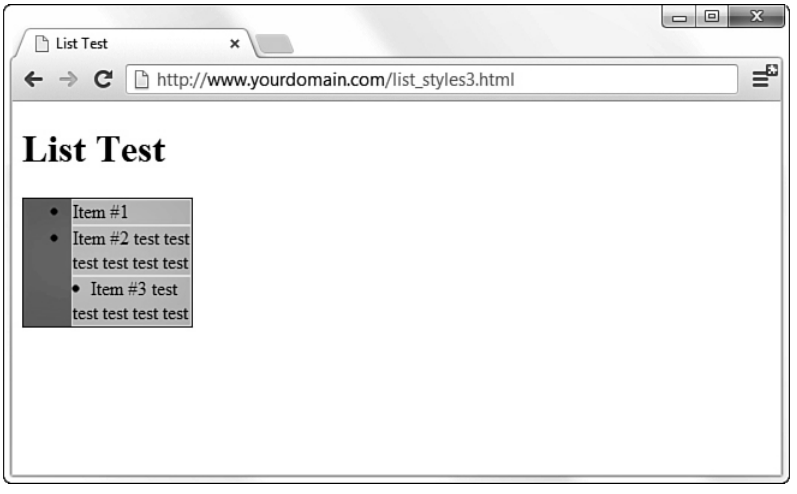



图 16.4 list-style-position 属性的 outside 值与 inside 值之间的区别

用于第二个列表项的额外的填充文本显示了：当把列表的宽度定义为一个太窄以至于不能在一行上显示全部文本的值时文本是如何换行的。无需使用 list-style-position: outside 就能够实现相同的结果，因为这是 list-style-position 的默认值，因此代码中无需任何显式的声明。

不过，可以清楚地看到使用 inside 位置时的区别。在第三个列表项中，项目符号和文本都位于被蓝色边框包围的灰色区域（列表项本身）内。当 list-style-position 的值为 inside 时，边距和填充将以不同的方式影响列表项，如图 16.5 所示。

在图 16.5 中，第二个和第三个列表项都具有值为 inside 的 list-style-position 属性。不过，第二个列表项具有 12 像素的 margin-left 值，第三个列表项则具有 12 像素的 padding-left 值。尽管两个内容块（列表指示符以及文本）显示文本环绕在项目符号周围，并且这些块在定义列表项的灰色区域内的放置方式是相同的，但是受影响的区域是列表自身内的列表项。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

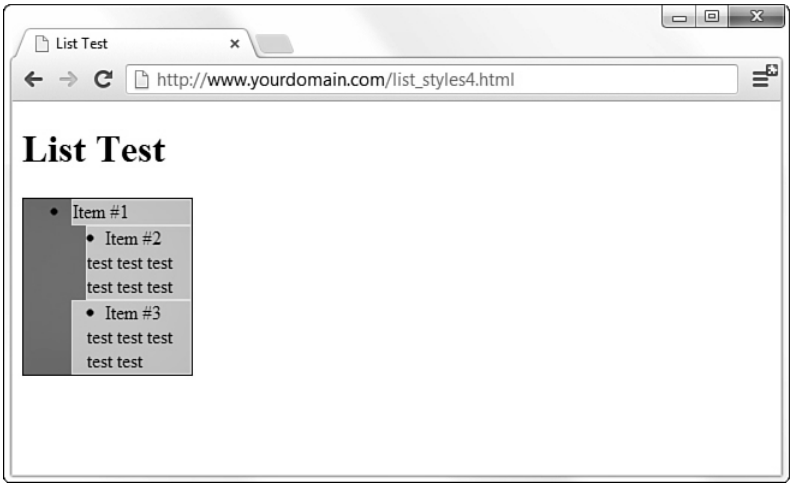



图 16.5

边距和填充使用值为 inside 的 list-style-position 改变显示效果

如你所期望的，具有 12 像素的 margin-left 值的列表项将透过列表项周围的透明边距显示出 12 像素的红色。类似地，具有 12 像素的 padding-left 值的列表项将在内容开始之前显示 12 像素的（列表项的）灰色背景。填充位于元素内；边距则位于元素外。

通过理解边距和填充影响列表项以及包含它们的列表的方式，可以在 Web 站点中创建导航元素，它们是纯粹的 CSS，并且不依赖于外部图像。在本章后面，你将学习如何创建垂直和水平导航菜单，以及下拉菜单。

## 16.4 利用列表项和 CSS 创建图像映射

在第 11 章中，学习了如何使用 HTML 中的<map>标签创建客户端图像映射。图像映射使你能够定义一个图像区域，并分配一个指向该区域的链接（而不必把图像切分成块，并对各个块应用链接，然后再在 HTML 中把图像缝合在一起）。不过，也可以纯粹利用有效的 HTML 和 CSS 创建图像映射。

程序清单 16.2 中的代码产生了如图 16.6 所示的图像映射。当在 Web 浏览器中呈现该代码时，它看起来简直就像是一个 Web 页面中放置了一幅图像。当鼠标悬停在“热”区（hot area）上时，如你在图 16.6 中可以看到的：较粗的白色边框和图像的 alt 文本显示了鼠标悬停的区域，并且在浏览器窗口的左下方，可以看到分配给该热点（hot spot）的 URL。

**程序清单 16.2 使用 CSS 创建图像映射**

```
<!DOCTYPE html>

<html 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;
    }
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    #theImg ul {
        margin:0px;
        padding:0px;
        list-style:none;
    }
    #theImg a {
        position:absolute;
        text-indent: -1000em;
    }
    #theImg a:hover {
        border: 4px 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;
    }
    #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>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权




图 16.6

CSS 使你能够在图像映射中定义热点

如程序清单 16.2 所示，样式表具有相当多的条目，但是实际的 HTML 代码相当短。列表项用于创建 5 个不同的可单击区域，这些“区域”都是列表项，给它们分配了特定的高度和宽度，然后放置在位于背景中的图像上。如果从包围列表的 的背景中删除图像，那么列表项仍将存在，并且仍然是可单击的。

让我们逐步解释样式表，以使你能够理解构成这个 HTML 和 CSS 图像映射的各个部分，在其最基本的层面上，该图像映射只是链接的列表。

链接的列表封闭在一个名为 theImg 的 中。在样式表中，这个 被定义一个块元素，其宽度为 500 像素，高度为 375 像素，并且具有 1 像素的纯黑色边框。这个元素的背景是一幅名为 tea_shipment.jpg 的图像，它放置在一个位置，并且不会重复。你看到的下一段 HTML 代码是无序列表（<ul>）的开头。在样式表中，把这个无序列表四周的边距和填充值都设置为 0 像素，并把 list-style 设置为 none——列表项前将不会放置任何图标。

由于在用于 内的所有<a>标签的样式表条目中采用了以下技巧，列表项文本自身永远不会显示给用户：

```
text-indent: -1000em;
```

通过把文本缩进负 1000 em，可以向你保证文本永远也不会出现。它确实存在，但它存在于一个不可查看的区域，距离浏览器窗口左边 1000 em。换句话说，如果你举起左手，并把它放在计算机显示器的一边，text-indent:-1000em 把文本放在你的小手指左边的某个位置。但是这就是我们所想要的，因为我们不需要看到文本链接。我们只需将一个区域定义为链接，使得用户的光标能够改变，就像它移到 Web 站点中的任何链接上一样。

当用户的光标悬停在包含一个链接的列表项上时，由于样式表中的下面这个条目，该列表项将显示 4 像素的纯白色边框：

```
#theImg a:hover {
   border: 4px solid #ffffff;
}
```

然后定义列表项本身，并且基于被指望成为可单击区域的图像区域把它们放在特定的

异步社区会员 Kensuke(18319066421) 专享 尊重版权

位置。例如，ID 为 ss 的列表项（用于 Sugarshots——图中显示的列表项名称）的左上角距离 顶部和左边缘分别是 0 像素和 5 像素。这个列表项的宽度是 80 像素，高度是 225 像死缠烂打。对#gn、#ib、#iTEA1 和#iTEA2 这些列表项创建了类似的样式声明，使得与这些 ID 关联的链接区域出现在相对于图像的某些位置。

## 16.5 小结

本章开始介绍了一些示例，说明列表和列表元素如何受到填充和边距样式影响。然后，本章介绍了与列表关联的默认填充，以及如何控制这种填充。接下来，又讲解了如何修改填充和边距值，以及如何把列表项指示符放在列表项的里面或外面，你开始考虑样式和列表如何影响总体的站点设计。最后，本章讲解了如何利用列表和列表元素创建纯粹的 HTML 和 CSS 图像映射，从而减小切分链接的图像或者使用<map>标签的需要。

在学习了“考虑（列表）方框外面”之后，读者还将学到如何使用无序列表在 Web 站点内创建水平或垂直导航系统。通过使用 CSS 代替图形，在站点的显示和维护方面就具有了更大的灵活性。通过本章，读者学习了利用样式表中的几个条目，可以把加下划线的普通文本链接转换成具有边框和背景色以及其他文本样式的区域。此外，读者还学习了如何在菜单内提供嵌套列表。

## 16.6 问与答

**问：许多极佳的 Web 页面都讨论了关于边距和填充的“方框模型重构（hack）”，尤其是对于列表和列表元素。你确信我没有使用这样的重构吗？**

**答：**在本章开头，你获悉本章（及其他章）中的 HTML 和 CSS 在主流 Web 浏览器的当前版本中看上去都是相同的。这是在现代浏览器开始依据 CSS 规范（而不是它们自己的特质）处理事情之前 Web 开发人员多年来不得不进行代码重构以及采用其他技巧的产物。此外，Internet 用户正日益摆脱那些首先需要其中大多数重构的非常老的 Web 浏览器。因此，尽管我不一定建议你只为主流 Web 浏览器的当前版本设计 Web 页面，但是我也不建议你花大量的时间为较老的浏览器版本实现重构，顺便说一下，这些较老的浏览器版本在 Internet 全体用户中只有不到 5%的人在使用。你应该继续编写可以经过验证并且遵守设计原则的稳定代码，在最能反映你的受众的一系列浏览器中测试页面，并把你的站点向全世界发布。

**问：创建 CSS 图像映射好像要做许多工作。<map>标签是如此糟糕吗？**

**答：**<map>标签根本不糟糕，并且确实是有效的 HTML5。不过，确定客户端图像映射中使用的坐标可能很困难，尤其是在没有图形软件或者打算用于创建客户端图像映射的软件的情况下。CSS 版本可以给你提供更多的选项，用于定义和显示可单击区域，你在这里只看见了其中一个这样的区域。

## 16.7 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要

异步社区会员 Kensuke(18319066421) 专享 尊重版权

尝试尽量回答所有的问题。

### 16.7.1 问题

1. list-style-position 属性的 inside 值与 outside 值之间的区别是什么？它的默认值是什么？
2. 值为 none 的 list-style 属性仍然会产生一个结构化的列表吗，它是有序列表还是无序列表？
3. 在使用无序列表创建图像热点时，怎样绘制热点区域？

### 16.7.2 答案

1. 值为 inside 的 list-style-position 属性将把列表项指示符放在由列表项创建的块内。如果值为 outside，则把列表项指示符放在块外。当值为 inside 时，内容将环绕在列表项指示符底下。默认值是 outside。
2. 是的。唯一的区别是在列表项内的内容之前没有列表项指示符。
3. 绘制热点区域的方法是：给列表项提供特定的 height 和 width，以及通过 right 值和 left 值指示的起点。

### 16.7.3 练习

- 就像我们在本章开头所做的那样，通过创建列表和列表项，然后调整相应的属性值，确信你确实理解了填充和边距的不同作用，以及使用 inside 和 outside 放置列表项的方式。
- 找到一幅图像，并使用本章中展示的技术亲自动手创建映射区域。选择一幅可以划分一些区域的图像，其中可以使用热点或可单击区域通向你的站点上的其他 Web 页面或者别人的站点。然后创建 HTML 和 CSS 定义可单击区域以及它们应该通向的 URL。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 17 章

# 使用 CSS 设计导航系统

在本章中你将学到：

- 导航列表与常规的列表有何区别；
- 怎样利用 CSS 创建垂直导航系统；
- 怎样利用 CSS 创建水平导航系统。

上一章讲解了如何将边距、填充和对齐样式应用于不同类型的 HTML 列表，帮助你纯粹以 HTML 和 CSS 创建一些强大的设计元素。本章将讲解许多方式中的其中几种，把列表用作垂直或水平导航系统，包括如何使用列表创建下拉菜单。

本章中解释的方法代表可以使用列表创建的许许多多、各种各样的导航方法的一个非常小的子集。不过，概念都是相似的，不同的结果出自于你自己的创意以及对这些基本概念的应用。为了使你能够才思泉涌，我提供了一些指针，它们指向本章末尾的其他一些基于 CSS 的导航系统。

## 17.1 导航列表与常规的列表有何区域

在我们讨论使用列表创建导航元素时，我们真实的用意是使用 CSS 以 Web 站点访问者对导航外观的所期望的方式显示内容。简言之，不同于简单的项目符号或编号列表。尽管一组导航元素实质上就是链接的列表，但是那些链接通常以一种用户显然应该与内容交互的方式显示。

- 用户的鼠标光标会改变，以指示元素是可单击的。
- 当鼠标悬停在元素上时，元素周围的区域会改变外观。
- 内容区域在视觉上显得与普通的文本有所不同。

创建导航的老方法倾向于依靠图像（比如具有斜面边缘以及为背景和文本使用对比色的

异步社区会员 Kensuke(18319066421) 专享 尊重版权

图形）以及利用 JavaScript 进行客户端编程，基于鼠标动作来处理图像交换。但是，使用纯粹的 CSS 通过列表元素创建导航系统可以产生更有用、更灵活并且对搜索引擎更友好的显示效果，用户可以使用各种方式和各种设备访问它们。

无论导航元素的布局是怎样的（水平或垂直），本章都讨论了两种导航级别：主导航（primary navigation）和次级导航（secondary navigation）。主导航把用户带到站点的主要区域的介绍性页面；次级导航则在某个区域内反映这些页面。

## 17.2 利用 CSS 创建垂直导航系统

依赖于站点架构——你创建的显示模板以及在站点中对信息进行分类的方式，你可能发现自己为主导航或次级导航使用垂直导航系统。

例如，假设为你公司创建了一个 Web 站点，它具有“About Us”、“Products”、“Support”和“Press”这些主要区域。在“About Us”主要区域内，你可能还具有另外几个页面，比如“Mission”、“History”、“Executive Team”和“Contact Us”，这些其他的页面就是“About Us”主要区域内的次级导航。

程序清单 17.1 建立了一个基本的次级页面，在页面的侧面具有垂直导航系统，内容则位于页面的中间。页面侧面的链接和页面的内容区域中的链接是基本的 HTML 列表元素。

这个程序清单和图 17.1 所示的示例提供了一个起点，用于说明 CSS 怎样使你能够把两种类似的 HTML 结构转换成两种不同的视觉显示效果（以及两种不同的环境）。

**程序清单 17.1 在列表中具有垂直导航系统的基本页面**

```
<!DOCTYPE html>

<html 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;
      }
      section {
         width:550px;
         float:left;
      }
    </style>
  </head>

  <body>
    <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>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      </ul>
      </nav>
      <section>
      <header>
      <h1>About Us</h1>
      </header>
      <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>
    </section>
   </body>
</html>
```

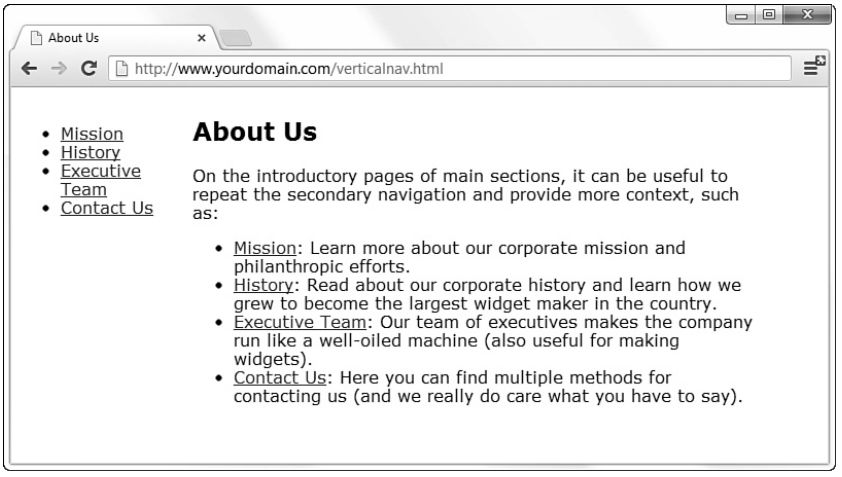



图 17.1

起点：未编排样式的列表导航系统

这个页面的内容建立在两个区域中：一个<nav>元素和一个<section>元素，其中前者包含导航系统，后者则包含页面的主要文本。在这个页面中指定的唯一样式是与每个元素关联的宽度、边距和浮动值，并且没有为列表元素应用样式。

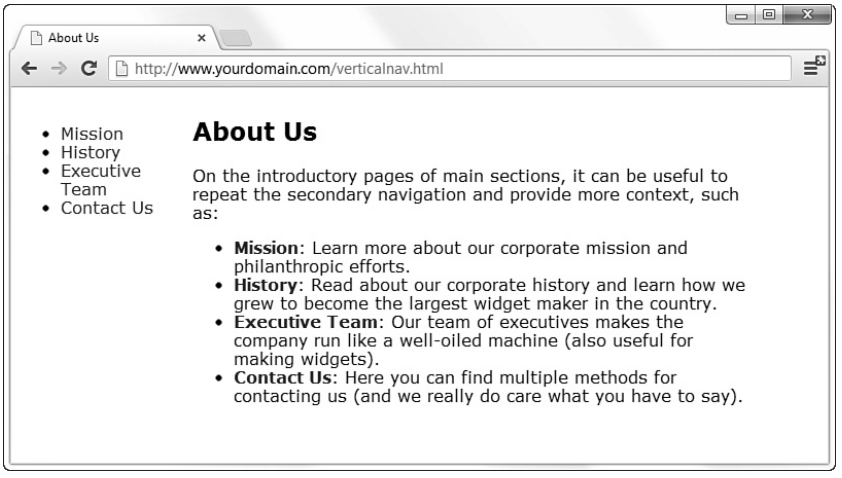
为了区分出现在内容区域中的列表中的链接与出现在侧面导航中的列表中的链接，可以向样式表中添加以下样式：

```
nav a {
   text-decoration: none;
}
section a {
   text-decoration: none;
   font-weight: bold;
}
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

这些样式简单地指示<nav>中的所有<a>链接都没有下划线，并且<section>中的所有<a>链接也没有下划线，但是它们是加粗的。图 17.2 显示了它们之间的区别。

图 17.2
使用 CSS 区分列表元素




但是，为了真正使侧面导航列表看起来比较特殊，将不得不更深入地研究样式表。

## 17.2.1 编排单级垂直导航系统的样式

这个特定的导航元素集的目标只是把它们展示为没有项目符号的链接块，并且背景色和文本颜色将依赖于它们的链接状态（普通链接、访问过的链接、悬停在链接上或者激活的链接）而改变。这个过程的第一步已经完成了：即把导航与内容分隔开，我们通过把导航放在一个<nav>元素中来实现这一点。

接下来，需要修改在<nav>元素内定义链接的<ul>。让我们去掉列表指示符，并且确保除了上边距外没有额外的边距和填充。上边距用于把导航系统的顶部与页面内容区域中的“About Us”标头文本的顶部对齐：

```
Nav ul {
   list-style: none;
   margin: 12px 0px 0px 0px;;
   padding: 0px;
}
```

由于导航列表面本身将显示为彩色的区域，可以给每个列表项提供一个下边框，使得可以在视觉上把内容分隔开：

```
nav li {
   border-bottom: 1px solid #ffffff;
}
```

现在继续构建余下的列表项。这里的思想是：当列表项只是简单地充当链接时，它们将具有特殊的蓝色阴影，以及白色的加粗文本（尽管它们的字体大小比正文文本自身要小一些）。为了实现这种效果，可以添加以下代码：

```
nav li a:link, nav li a:visited {
   font-size: 10pt;
   font-weight: bold;
   display: block;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    padding: 3px 0px 3px 3px;
    background-color: #628794;
    color: #ffffff;
}
```

也许除了样式表条目中使用的 display: block;之外，你应该熟悉以前使用过的所有样式。把 display 属性设置为 block，确保当用户把鼠标悬停在<li>元素上时，整个元素都在起作用。图 17.3 显示了应用过这些新样式的垂直列表菜单。

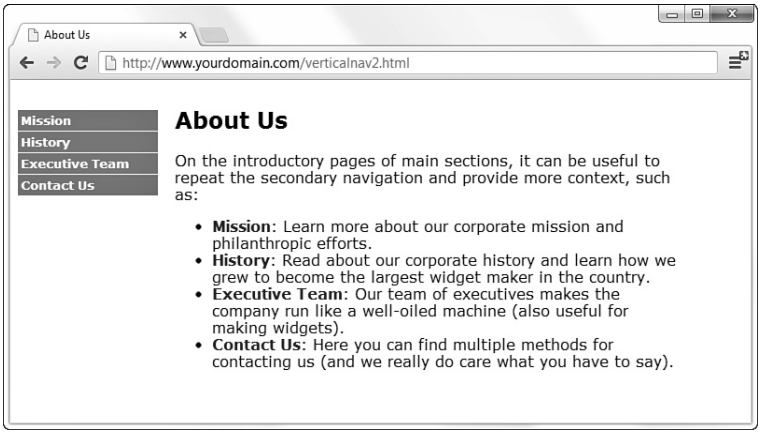



图 17.3

垂直列表开始看起来像导航菜单

当用户的鼠标悬停在导航列表元素上时，需要使某种视觉改变发生，使得用户知道元素是可单击的。这类似于当用户把光标悬停在菜单项上时，大多数软件菜单是如何改变颜色的。在这种情况下，我们将改变列表项的背景色，并改变列表项的文本颜色，它们将不同于前面所示的蓝色和白色。

```
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 显示了迄今为止的所有样式方面的工作的结果。样式表中的几个条目把简单的列表转换成视觉上有差别的菜单。

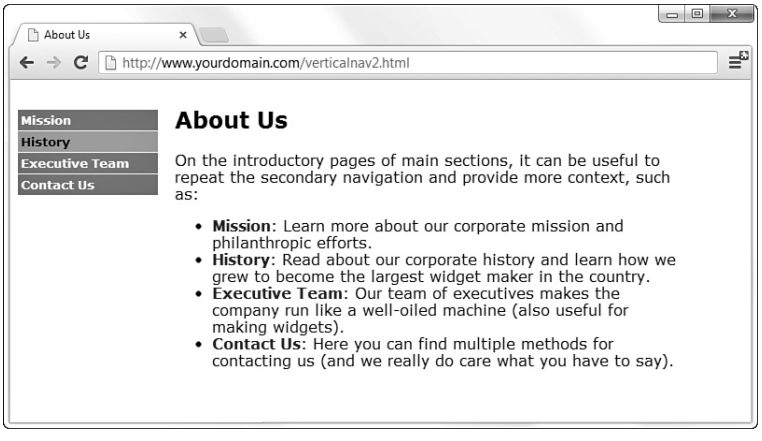



图 17.4

当鼠标悬停在列表项上时，它们现在将改变颜色

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 17.2.2 编排多级垂直导航系统的样式

如果站点架构需要另一级导航系统并且你希望用户一直都能看到它，则该如何？可以通过嵌套列表（在前面几章中学习过它）和更多的样式表条目表示它。在这里，假定在“Executive Team”链接下面有 4 个导航元素。在 HTML 中修改列表，如下所示：

```
<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“Executive Team”链接下面产生一个嵌套列表，如图 17.5 所示。HTML 实体»产生右指的箭头，它们出现在新链接中的文本之前。

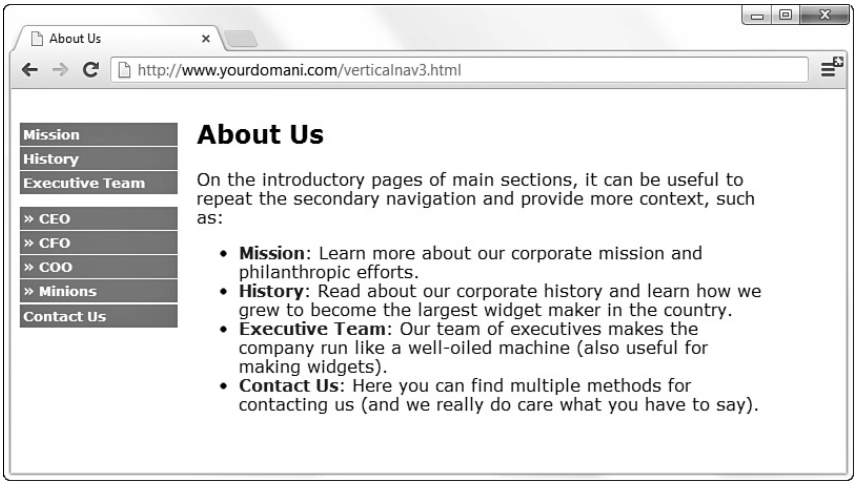



图 17.5 创建嵌套的导航列表（但是它还没有很好地编排样式）

新项目作为块元素出现在列表内，但是没有形象地表示出信息的层次结构。为了添加某种视觉元素，将这些项目标识为附加到“Executive Team”链接上的子导航元素，可以再次修改样式表，添加某种缩进效果。

但是在这样做之前，你还可能希望修改一些其他的样式表条目。在上一节中，我们添加了诸如 navul 和 nav li 之类的选择器，它们分别指示“<nav>元素中的所有<ul>元素”和“<nav>元素中的所有<li>元素”。不过，我们现在连同<nav>元素还具有两个<ul>的实例和另一组<li>元素，我们希望它们都以不同于原始元素集的方式显示。

为了确保合适地编排两个列表项集的样式，要确保样式表选择器清楚地指示列表的层次结构。为此，可以为列表的第一级使用诸如 navul 和 navul li 之类的条目，并为列表的第二级使用 navulul 和 navulul li。程序清单 17.2 显示了用于产生如图 17.6 所示菜单的样式表

异步社区会员 Kensuke(18319066421) 专享 尊重版权

条目和 HTML 的新版本。

**程序清单 17.2 列表中的多级垂直导航系统**

```
<!DOCTYPE html>

<html 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;
      }
      section {
         width:550px;
         float:left;
      }
      nav a {
         text-decoration: none;
      }
      section 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 {
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
            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>
    <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>
        <li><a href="#">&raquo; CFO</a></li>
        <li><a href="#">&raquo; COO</a></li>
        <li><a href="#">&raquo; Other Minions</a></li>
        </ul>
      </li>
      <li><a href="#">Contact Us</a></li>
    </ul>
    </nav>
    <section>
      <header>
      <h1>About Us</h1>
      </header>
      <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>
    </section>
  </body>
</html>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

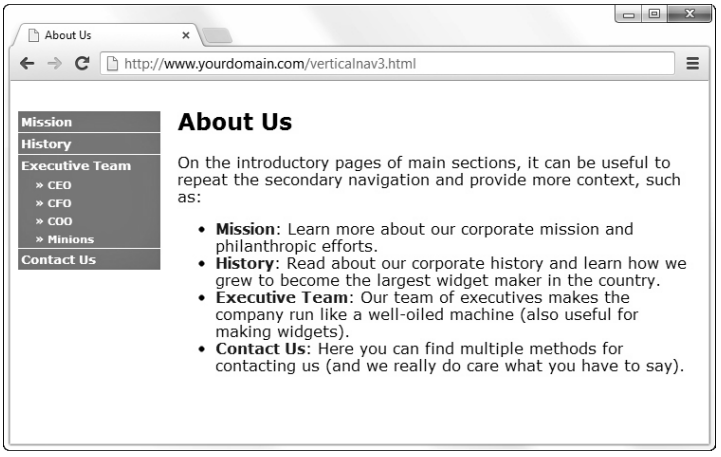



图 17.6
使用 CSS 创建两级垂直导航系统

只有你自己的创意才能限制编排垂直导航系统样式的不同方式。你可以使用颜色、边距、填充、背景图像以及任何其他的有效 CSS 创建相当灵活并且容易修改的垂直导航系统。如果在搜索引擎中输入“CSS 垂直导航”，将会找到数百万个示例，并且它们都基于你在本章中学到的简单原理。

## 17.3 利用 CSS 创建水平导航系统

关于导航的章节开始时都是介绍垂直导航系统，因为当导航系统看起来仍然像你可能在一张纸上垂直地书写的项目列表（比如杂货列表）时，把列表转换成导航系统的概念更容易理解。在创建水平导航系统时，仍会使用 HTML 列表元素，但是将不会为<ul>和<li>使用 display 属性的 inline 值来实现垂直显示效果，而是代之以使用 display 属性的 block 值。它确实就这么简单。

程序清单 17.3 显示了用于创建具有水平导航系统的页面的起点。该页面包含一个<header>元素和一个<section>元素，前者用于标志和导航，后者用于内容。在<header>元素内，包含标志的 将浮动到包含导航链接的<nav>元素旁边。出现在<nav>元素中的列表对于列表和列表项，其 display 属性值都是 inline。在图 17.7 中可以看到这些元素以及它们的放置方式。

**程序清单 17.3 通过列表创建的基本水平导航系统**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>ACME Widgets LLC</title>
    <style type="text/css">
      body {
         font: 12pt Verdana, Arial, Georgia, sans-serif;
      }
      header {
         width: auto;
      }
      #logo {
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
        float:left;
      }
      nav {
        float:left;
      }
      nav ul {
        list-style: none;
        display: inline;
      }
      nav li {
        display: inline;
      }
      section {
        width: auto;
        float: left;
        clear: left;
      }
      section a {
        text-decoration: none;
        font-weight: bold;
      }
    </style>
  </head>
  <body>
    <header>
      <div id="logo">
        <img src="acmewidgets.jpg" alt="ACME Widgets LLC" />
      </div>
      <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>
      </nav>
    </header>
    <section>
      <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>
    </section>
  </body>
</html>
```

修改这个列表的显示效果纯粹是通过 CSS 进行的，HTML 自身内的内容结构已经设置好了。为了实现想要的显示效果，可以使用以下 CSS 代码。首先，把<nav>元素修改成特定

异步社区会员 Kensuke(18319066421) 专享 尊重版权

的宽度，显示背景色和边框，以及使用 85 像素的上边距（使得它出现在标志底下）。

```
nav {
   float:left;
   margin: 85px 0px 0px 0px;
   width: 400px;
   background-color: #628794;
   border: 1px solid black;
}
```




图 17.7

使用内联列表元素创建实用的——但是不一定很漂亮的——水平导航系统

除了边距和填充中的变化之外，<ul>的定义将保持与程序清单 17.3 中的相同：

```
nav ul {
   margin: 0px;
   padding: 0px;
   list-style: none;
   display: inline;
}
```

<li>的定义也将保持与程序清单 17.3 中的相同，只不过给它的 line-height 属性提供的值是 1.8em：

```
nav li {
   display: inline;
   line-height: 1.8em;
}
```

链接样式类似于垂直导航系统中使用的那些样式，这些条目具有不同的填充值，但是颜色和字体大小将保持相同：

```
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: px 10px 7px 10px;
   background-color: #c6a648;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

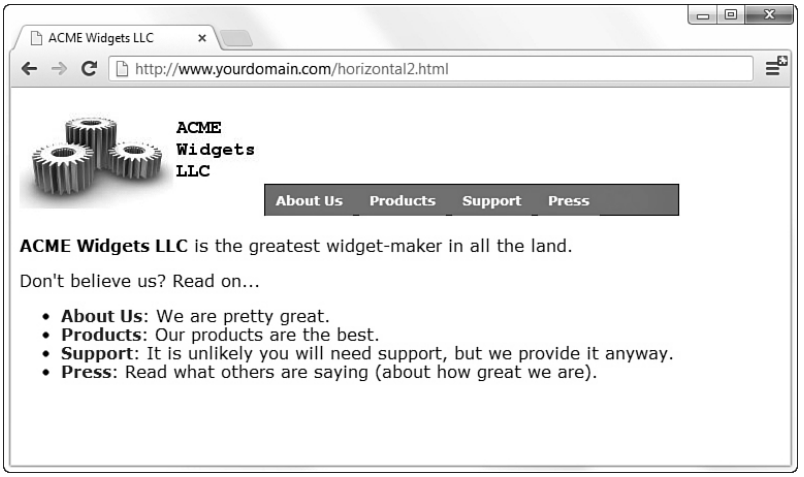
```
    color: #000000;
}
```

把这些样式放在一起，就可以产生如图 17.8 所示的显示效果。

当用户把鼠标指针悬停在导航元素上时，背景色和文本颜色将会以某种方式改变，就像用户把鼠标指针悬停在垂直导航菜单上一样。此外，就像对垂直导航菜单所做的那样，可以使用嵌套列表在水平菜单中产生下拉功能。自己试一试吧！

图 17.8

创建具有某种样式的水平导航系统




## 17.4 小结

本章主要介绍了如何使用无序列表在 Web 站点内创建水平或垂直导航系统。通过使用 CSS 代替图形，可以在站点的显示和外观方面获得更大的灵活性。通过本章，读者知道利用样式表中的几个条目，就可以把加下划线的普通文本链接转变成带有边框和背景色及其他文本样式的区域。此外，本章还介绍了如何在菜单内展示嵌套列表。

## 17.5 问与答

**问：我可以在导航菜单中使用图形作为自定义的列表指示符吗？**

**答：**是的。可以在列表项的 HTML 文本内使用图形，或者将其作为<li>元素内的背景图像。可以像编排任何其他的列表元素的样式那样编排导航元素的样式。HTML 无序列表与基于 CSS 的水平或垂直导航列表之间的唯一区别是它们的叫法不同，并且可以在文本主体外面出于特定的目的使用无序列表。除此之外，还可以编排列表的样式，向用户显示它的确有所不同，并且可以利用较小的图形凸显列表来实现此目的。

**问：我在哪里可以找到关于利用列表可以做什么的更多示例？**

**答：**据我上一次检查，在搜索引擎中输入“CSS 导航”返回了将近 4400 条结果。下面列出了几个起始位置。

- A List Apart 的 CSS 文章，位于：http://alistapart.com/topic/css。
- Maxdesign 的 CSS Listamatic，位于：http://css.maxdesign.com.au/listamatic/。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

- Vitaly Friedman 的 CSS 展示，位于：www.alvit.de/css-showcase/。

## 17.6 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 17.6.1 问题

1. 要创建水平列表，可以使用 inline 或 block_display 属性吗？
2. 在创建基于列表的导航系统时，可以使用多少层级的嵌套列表？
3. 在创建任何类型的导航列表时，用于 a 选择器的 4 个伪类可以具有相同的值吗？

### 17.6.2 答案

1. 水平列表使用 display 属性的 inline 值；垂直列表使用 block 值。
2. 从技术上讲，可以按需要的深度嵌套列表。但是，从有用性的观点看，对于你想要用于嵌套列表的层级数量是有限制的。这个界限通常是 3 级列表，如果超过限制，就会存在以下风险：创建的站点组织结构不佳，或者给用户提供的选项多于他们需要随时看到的选项。
3. 当然可以，但是会冒着用户没有意识到漂亮的菜单就是真正的菜单的风险（由于没有针对鼠标动作产生相应的视觉显示）。

### 17.6.3 练习

- 使用所显示的用于多级垂直列表的技术，向本章末尾创建的垂直列表中添加子导航项目。
- 查看 Web 站点中使用的基于 CSS 的导航的众多示例，找到一些看上去富有技巧的动作。使用 Web 浏览器的 View Source（查看源文件）功能，查看这些站点使用的 CSS，并且尝试为你自己实现类似的效果。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 18 章

# 使用鼠标动作修改文本显示

在本章中你将学到：

- 当用户移动鼠标时怎样显示工具提示；
- 怎样使用鼠标事件更改关于页面的信息。

在本书中，我们见过了一些说明鼠标动作可能如何影响文本显示的示例。最简单的示例是在样式表中定义<a>链接的 hover 伪类，用于当用户的鼠标悬停在文本上时以不同的颜色显示它。上一章学习了 hover 伪类如何改变<li>元素的文本颜色和背景色。

本章将学习如何实现两个结合了 CSS 和鼠标动作的特定动作：当用户将鼠标移动到某个项目上时显示额外的文本，以及使用鼠标单击来改变容器元素的颜色。当然，这两个动作在某些情况下是有用的，但是更重要的是，它们为你以后在 Web 开发中可能希望处理的更高级的工作提供了一个入口点。

## 18.1 利用 CSS 创建工具提示

工具提示（ToolTip）是图形用户界面（可能是一个软件程序或者一个 Web 站点）内的一个元素，当光标悬停在特定的项目上时它将提供额外的信息。图 18.1 显示了一个工具提示的实际应用：鼠标移动到链接文本“markup language”上，工具提示则在一个小方框中显示“Markup language”。在这里，工具提示中显示的文本来自于<a>标签的 title 属性。

图 18.1 中的工具提示的显示是由软件本身控制的。也就是说，它的显示和外观并不是你（即 Web 开发人员）所能控制的。不过，你可以通过应用在前几章中已经实践过的概念利用一点 CSS 创建你自己的工具提示。

程序清单 18.1 包含一些样式表条目和 HTML 代码，它们用于一个包含图像和链接的页面，但是有一个链接还会显示自定义的工具提示。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

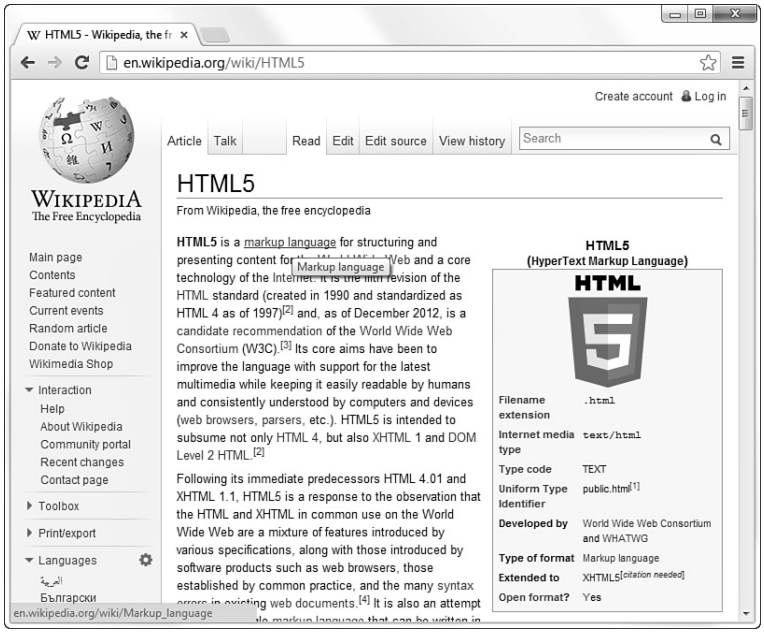



图 18.1

标准的工具提示的实际应用

## 程序清单 18.1　利用 CSS 创建简单的工具提示

```
<!DOCTYPE html>

<html 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;
       font-style: italic;
       display: block;
       position: absolute;
       top: 20px;
       left: 25px;
       width: 150px;
       padding: 3px;
       border:1px solid #000;
       background-color:#ddd;
       color:#000;
    }
    img {
       float:left;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      margin-right: 12px;
      margin-bottom: 6px;
      border: 1px solid #000;
    }
    </style>
  </head>
  <body>
    <header>
      <h1>Steptoe Butte</h1>
    </header>
    <p><img src="steptoebutte.jpg" alt="View from Steptoe Butte" />
    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>
    <footer>
      <em>Text from
      <a href="http://en.wikipedia.org/wiki/Steptoe_Butte">Wikipedia</a>,
      photo by the author.</em>
    </footer>
  </body>
</html>
```

注意：文本中的第一个链接具有一个与之关联的类，它是名称是 tip，这个名称使之与其他不包含工具提示的链接区分开。不过，由于这是一个类并且没有 id，它也可以用于页面中的所有其他的链接。

注意：tip 类本身具有定义好的 position 和 z-index。position 值（relative）确保它在展示时就像自然地出现在文本中一样。与其他相关样式使用的 z-index 属性相比，tip 类的 z-index 属性利用较高的 z-index 值确保链接的文本保持出现在其他文本之下。例如，使用 tip 类的链接的悬停状态就具有较高的 z-index 值。

后两种样式与使用的标签相关。除非用户通过把鼠标悬停在可见的链接上来调用工具提示，否则将不会显示<a>标签内的标签内的文本。例如，链接文本看起来如下所示：

```
<a class="tip" href="http://en.wikipedia.org/wiki/Palouse">Palouse
<span>Learn more about the Palouse!</span></a>
```

当查看这段 HTML 代码时，看起来好像“Palouse Learn more about the Palouse!”这些单词应该全都是一个链接。不过，由于“Learn more about the Palouse!”封闭在标签

异步社区会员 Kensuke(18319066421) 专享 尊重版权

中，并且在样式表中对该应用于特定的样式，直到用户的鼠标悬停在实际的链接（单词“Palouse”）上时，这些单词才会显示出来。图 18.2 显示了这种结果。




图 18.2

当鼠标悬停在页面中的第一个链接上时，将会显示自定义的工具提示

文本“Learn more about the Palouse!”放置在 HTML 中的标签中，但它是在样式表内编排样式的。也就是说，a.tip:hover span 选择器创建了一个方框，它的宽度是 150 像素，并且具有灰色的背景和黑色的边框。这个方框距离元素的顶部 20 像素，并且距离父元素的左边 25 像素。这将创建工具提示，并且可以根据需要自定义它。

## 18.2 利用 CSS 显示额外的翻转文本

工具提示服务于特定的目的：显示与链接相关的文本“提示”。但是，可以基于鼠标动作使用相同的概念在布局中的另外某个位置显示隐藏的文本。例如，第 17 章中的 ACME Widgets LLC 主页——具有水平导航系统的版面。通过使用与程序清单 17.1 中的那些相同的概念，可以在用户把鼠标移到主要选择链接之一上时，在菜单上方显示额外的文本。图 18.3 显示了其中的操作之一，我重新捕获了导航上面和标志右边的一些空白。

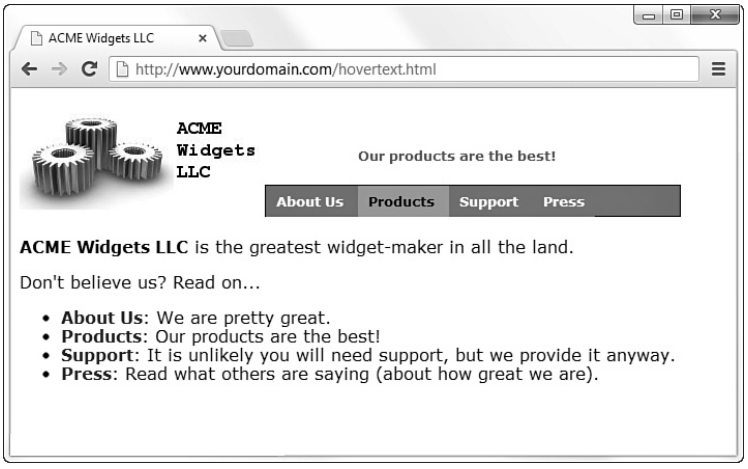



图 18.3

另一个元素上面的自定义文本，只在翻转时才可见

异步社区会员 Kensuke(18319066421) 专享 尊重版权

在第 17 章中，对程序清单 17.3 中的 CSS 所做的唯一改变以及其后的修改是添加了下面 4 种样式。这 4 种样式所执行的任务与程序清单 18.1 中使用的 tip 类样式相同，但是显示的内容稍微不同于程序清单 18.1 所产生的结果。

```
nav a.more {
   position:relative;
   z-index:24;
}
nav a.more:hover {
   z-index:25;
}
nav a.more span {
   display: none;
}
nav a.more:hover span {
   font-weight: bold;
   display: block;
   position: absolute;
   top: -35px;
   width: 300px;
   color:#ff0000;
   line-height: 1em;
}
```

top 属性的值−25px 用于把封闭在标签中的文本放置在父标签的左上角上方 25 像素，这与前一个示例相对，该示例把它放在父标签的左上角下方 20 像素。对 HTML 所做的改变类似于上一节中的所做的改变，这是由于文本是隐藏的，直到<a>链接内的标签中出现翻转为止。

```
<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>
```

使用这些基本的概念，可以基于用户的鼠标悬停在<a>链接上时所执行的动作使文本出现在想要的任意位置。在下一节中，你将学习如何使用事件属性以及一点 JavaScript，对鼠标动作和 CSS 进行甚至更多的处理。

## 18.3 访问事件

诸如鼠标单击或按键之类的用户交互就称为事件（event），脚本基于事件采取动作的过程被称为事件处理（event handling）。可以使用特殊的属性把事件处理脚本代码与 Web 页面上的元素相关联。JavaScript 或 JavaScript 库（比如 jQuery（http://jquery.org/））的使用超出了本书的范围，但是它可用于处理这个常见的问题，并使你开始考虑在学习过程中下一步要做什么。

下面列出了 JavaScript 中的一些迟早会派上用场的常用事件属性，并且描述了当它们发

异步社区会员 Kensuke(18319066421) 专享 尊重版权

生时对 Web 页面元素所执行的操作。

- onload：浏览器加载元素。
- onkeydown：用户按键。
- onkeyup：用户释放键。
- onclick：用户利用鼠标左键单击元素。
- ondblclick：用户利用鼠标左键双击元素。
- onmousedown：当鼠标指针位于元素上时，用户按下任何一个鼠标键。
- onmouseup：当鼠标指针位于元素上时，用户释放任何一个鼠标键。
- onmouseover：用户把鼠标指针移入元素的边界内。
- onmousemove：当鼠标指针位于元素上时，用户移动鼠标指针。
- onmouseout：用户把鼠标指针移出元素的边界之外。

可以看到，事件属性用于响应常见的用户输入事件，比如鼠标单击或按键。可以通过把代码分配给事件属性，将 JavaScript 与某个事件关联起来，如下所示：

```
<h1 onclick="this.style.color = 'red';">I turn red when clicked.</h1>
```

在上面的代码段中，把一点 JavaScript 代码分配给<h1>标签的 onclick 事件属性，这意味着代码将运行以响应用户在文本上单击鼠标左键。脚本代码将通过把文本的颜色设置为红色来做出响应。这样，就通过更改文本的颜色来响应鼠标单击，给正常的普通文本添加了交互性。这是客户端脚本怎样与 Web 浏览器协同工作的基础。

在下一节中，你将看到一个示例，说明如何使用事件处理来改变 的外观。确切地讲， 的内容将基于鼠标单击而显示或消失。

## 18.4 使用 onclick 更改 的外观

可以使用 onclick 事件调用各种动作。你可能把鼠标单击视作一种通过单击按钮提交表单的方式，但是你也可以捕获这个事件，并使用它在页面内提供交互性。在这个示例中，你将看到如何使用 onclick 事件显示或隐藏一个 中包含的信息。使用看上去与程序清单 18.1 中相当类似的文本和图像，你将看到如何给页面添加交互性，它允许用户在单击一段文本时显示以前隐藏的信息。我之所以把该文本称为“一段文本”，是因为严格来讲该文本不是链接。也就是说，它看起来像是链接，但是并没有将其标记在<a>标签内。

程序清单 18.2 提供了用于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 显示或隐藏内容**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Steptoe Butte</title>
    <style type="text/css">
    a {
       text-decoration: none;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      font-weight: bold;
    }
    img {
      margin-right: 12px;
      margin-bottom: 6px;
      border: 1px solid #000;
    }
    .mainimg {
      float: left;
    }
    #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;
    }
    section {
      margin-bottom: 6px;
    }
    </style>
  </head>
    <body>
      <header>
        <h1>Steptoe Butte</h1>
      </header>

      <section>
        <p><img src="steptoebutte.jpg" alt="View from Steptoe Butte"
        class="mainimg" />Steptoe Butte is a quartzite island jutting out of the
        silty loess of the <a
        href="http://en.wikipedia.org/wiki/Palouse">Palouse </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
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      Edward Steptoe</a>.</p>
    </section>

    <section>
      <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>
    </section>

    <section>
      <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"><img src="steptoe_sm1.jpg" alt="View from Steptoe
      Butte" /><img src="steptoe_sm2.jpg" alt="View from Steptoe
      Butte" /><img src="steptoe_sm3.jpg" alt="View from Steptoe
      Butte" /></div>
    </section>

    <footer>
      <em>Text from
      <a href="http://en.wikipedia.org/wiki/Steptoe_Butte">Wikipedia</a>,
      photos by the author.</em>
    </footer>
  </body>
</html>
```

首先，查看样式表中的 11 个条目。第一个条目简单地编排由<a></a>标签对包围的链接的样式，这些链接显示为未加下划线的、加粗的蓝色链接。在图 18.4 中的两个文本段落（以及页面底部的行）中可以看到这些普通的链接。下两个条目确保页面中使用的图像具有合适的边距，用于<img />元素的条目将设置一些边距和一条边框，并且.mainimg 类允许对页面上的主要图像应用一种样式，但是不会对页面底部的一组 3 幅图像应用样式。

下 4 个条目用于特定的 ID，并且这些 ID 都被设置成在页面最初加载时是不可见的（display: none）。与之相反，其后两个 ID 被设置成在页面最初加载时显示为块元素。同样，严格来讲，这两个 ID 不必被定义为块元素，因为它是默认的显示规范。不过，这个样式表

异步社区会员 Kensuke(18319066421) 专享 尊重版权

包括了这些条目，用以说明两组元素之间的区别。如果统计程序清单 18.2 中的 元素的数量，将会在代码中找到 6 个 ：其中在页面加载时有 4 个是不可见的，另外两个则是可见的。

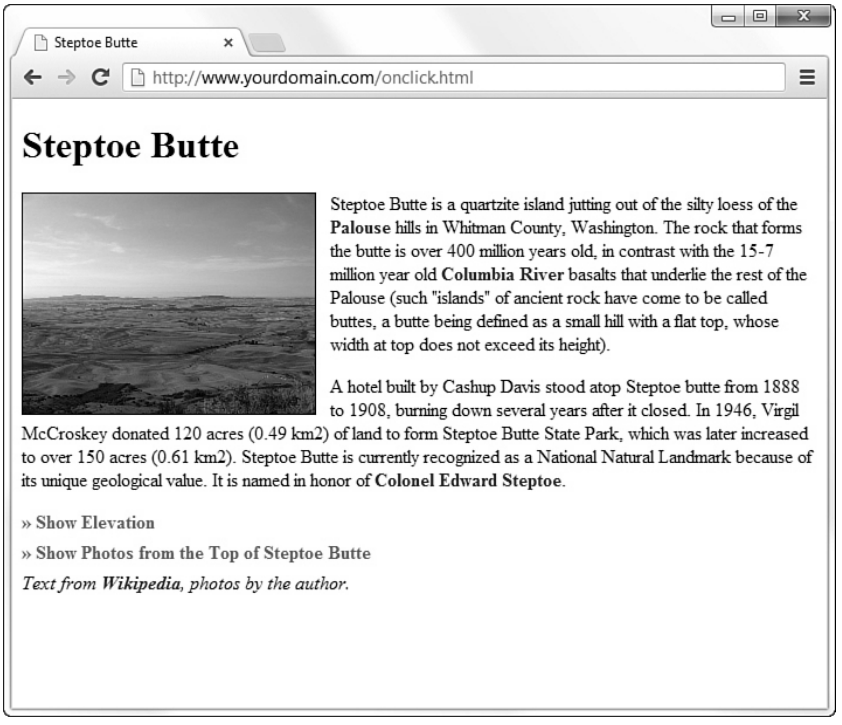



图 18.4

程序清单 18.2 的初始显示。注意：当鼠标指针悬停在红色文本上时，它将变成人手形状，尽管事实上该文本不是一个<a>链接

这个示例的目标是：在单击一个 ID 时，更改另外两个 ID 的显示值。但是，首先必须确保用户意识到一段文本是可单击的，并且这通常发生在用户看到他们的鼠标指针改变了形状以反映链接存在时。尽管在图 18.4 中看不到它，但是如果在你的机器上加载示例代码并在浏览器中查看它，鼠标指针将变成人手形状，并且有一根手指指向特定的链接。

这个功能是通过为这段特定的文本定义一个类来实现的，这个类称为 fakelink，如下面这个代码段中所示：

```
<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 类确保将文本呈现为未加下划线的、加粗的红色文本，cursor: pointer 导致鼠标指针改变形状，使得用户认为文本是通常封闭在一个<a>元素中的链接类型。但是，当我们把 onclick 属性与 相关联时，真正有趣的事情发生了。在刚才所示的示例代码段中，onclick 属性的值是一系列命令，它们可以改变 CSS 元素的当前值。

让我们单独查看它们：

```
this.style.display='none';
document.getElementById('hide_e').style.display='block';
document.getElementById('elevation').style.display='inline';
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

你在这里看到的是一种不同的 JavaScript 方法，它打算用于更改特定的元素。一般来讲，JavaScript 超出了本书的范围，但是我认为你可以理解这里所发生的事情（并且你将在第 21 章中学习更多一点关于 JavaScript 的知识）。

在代码段的第一行中，this 关键字指代元素本身。换句话说，this 指代 ID 为 show_e 的 。关键字 style 指代 style 对象，该对象包含分配给元素的所有 CSS 样式。在这里，我们最感兴趣的是 display 样式。因此，this.style.display 意指“show_e ID 的显示样式”，并且在单击文本自身时，我们将把 display 样式的值设置为 none。

但是，在 onclick 属性内还发生了 3 个动作。另外两个动作开始于 document.getElementByID()，并且在圆括号内包括一个特定的 ID 名称。我们使用 document.getElementByID()代替 this，是因为第二个和第三个动作将为不是父元素的元素设置 CSS 样式属性。在代码段中可以看到，在第二个和第三个动作中，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名为 show_e 的 时，将会发生以下事情。

- show_e 变成不可见的。
- hide_e 变成可见的，并且显示为一个块。
- elevation 变成可见的，并且显示为内联元素。

图 18.5 显示了这些动作的结果。

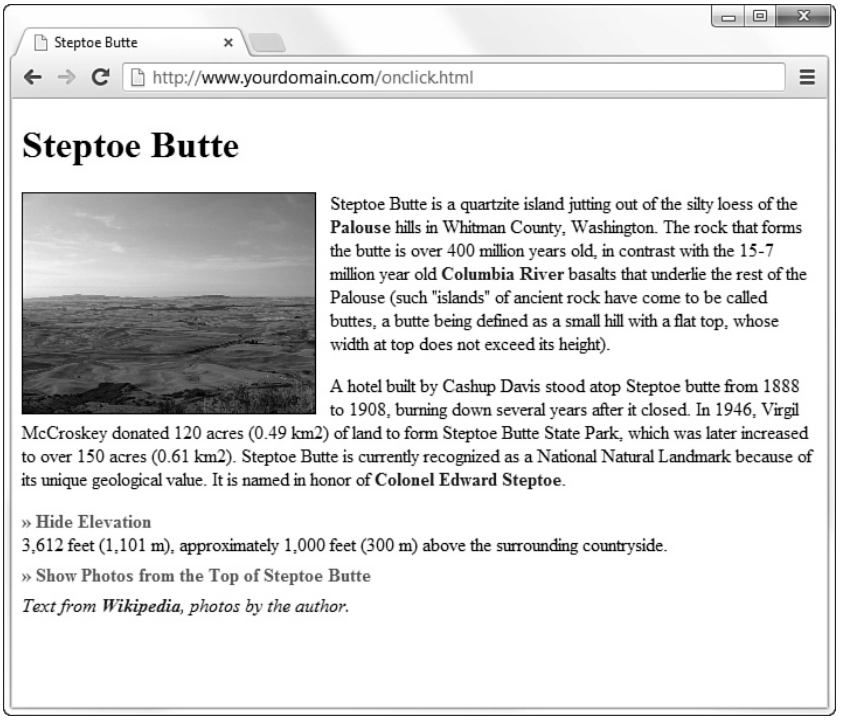



图 18.5
在单击“Show Elevation”之后，将基于 onclick 属性中的命令改变它及其他 元素的可见性

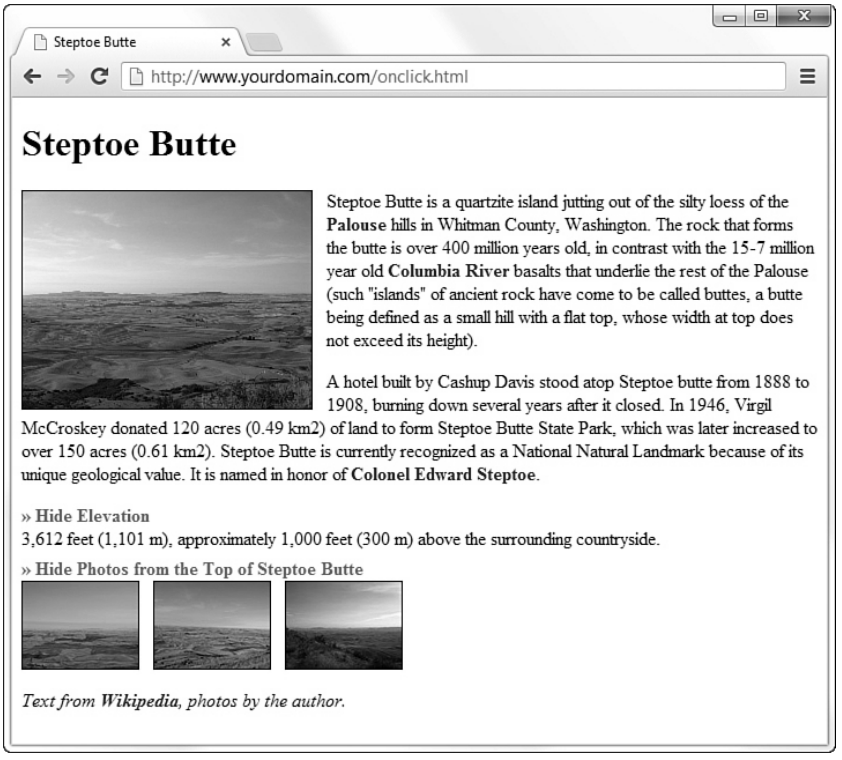
在程序清单 18.2 中的代码中还存在另一组 元素，它们用于控制额外的照片的可见性。这些元素不受与正视图相关的元素中的 onclick 动作影响。也就是说，当单击“Show Elevation”或“Hide Elevation”时，与照片相关的 元素不会改变。可以显示正视图但不显示照片（如图 18.5 所示），也可以显示照片但不显示正视图，或者同时显示正视图和照

异步社区会员 Kensuke(18319066421) 专享 尊重版权

片，如图 18.6 所示。

图 18.6

在单击“Show Elevation”和“Show Photos from the Top of Steptoe Butte”之后的页面




这个简短的示例显示了非常基本的布局和交互方式，等到你掌握了 CSS 以及事件之后，就可以很好地控制它们。例如，你可以编码页面，使得用户可以改变样式表的元素或者改成一个完全不同的样式表、把文本块移到布局中的其他位置、参加测验或提交表单，等等。

## 18.5 小结

本章讲解了鼠标动作可能如何影响文本的显示。在前两节中，这些动作基于同<a>标签的 hover 伪类关联的样式。本章后面介绍了与特定的用户交互事件关联的动作。示例中的特定事件是 onclick 事件，但是你也看到了其他可能事件的列表，比如 onload 和 onmouseover 等。

代码自身中介绍的唯一的新代码是使用 cursor 属性。给 cursor 属性赋予 pointer 值，使你能够指示用户特定的文本在充当链接，即使它没有像用户可能习惯看到的那样封闭在<a></a>标签中。

## 18.6 问与答

**问：本章中介绍的一些事件听上去比较像用于<a>标签的伪类。它们之间有什么区别吗？**

**答：**确实如此，onmousedown 事件像活动状态，onmouseup 和 onmouseover 非常像悬停状态。没有特定的规则介绍何时使用事件代替伪类，但是显而易见的是，如果没有其他的原因使用<a>标签，就不应该只使用<a>标签以使用伪类。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**问：你能够捕获除文本以外的元素（比如图像）上的鼠标或键盘事件吗？**

**答：**是的，可以把这些类型的事件应用于同单击或翻转图像以及文本相关的动作。不过，用户不会以相同的方式与其他多媒体对象交互，比如嵌入式 YouTube 视频或 Flash 文件，这些对象是通过额外的软件播放的，它们具有其他适用的鼠标或键盘动作。例如，如果单击嵌入在 Web 页面中的 YouTube 视频，就会与 YouTube 播放器交互，而不再与实际的 Web 页面交互，因为不能以相同的方式捕获该动作。

## 18.7 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 18.7.1 问题

1. 当为 top 属性使用一个负值时，会发生什么事情？
2. 当用户把鼠标移出特定元素的边界之外时，哪个事件用于改变与页面有关的某些方面？
3. 如果在样式表中看到 cursor: crosshair 样式，你认为它将做什么？

### 18.7.2 答案

1. 为 top 属性使用一个负值将把内容放在父元素的左上角的上方（而不是其下方）。
2. 使用 onmouseout 事件。
3. 你很可能认为它将把用户的鼠标指针变成一个大加号（或十字形）。不过，不清楚你为什么要这样做，因为这是光标的非典型展示。换句话说，用户可能想知道你为什么要对他们的界面那样做，他们将不会欣赏你的这种设计决策。

### 18.7.3 练习

- 采用你自己的带有悬停的工具提示文本的页面，并且使用颜色和其他样式处理你自己的显示模板。
- 向程序清单 18.2 中的 onclick 属性添加一些命令，使得每次只有其中一个 元素（正视图或照片）是可见的。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 19 章

# 实现 CSS3 变形、渐变和动画

在本章中你将学到：

- 怎样移动、旋转、缩放和扭曲 2D 元素；
- 怎样应用透视图以及移动和旋转 3D 元素；
- 怎样为在 HTML 和 CSS 中创建的对象制作动画。

HTML5 与 CSS3 的融合产生了一种极其强大的设计手段，本书只是浅尝辄止，毕竟，HTML5 和 CSS3 构成了许多移动应用程序的基础，并且还开始涉足视频游戏开发（所有这些都超出了本书的范围）。但是，我坚定地相信应该给初学者展示这些高级主题的基础，并且本章是一个基本的示例。使用 CSS3 的高级交互性（在编写本书时）仍然处于试验阶段，但是设计师已经尝试它有几年的时间了，因此你也能够这样做。

## 19.1 考虑浏览器的差别

在我们开始涉足 CSS3 变形（transformation）、渐变（transition）和动画（animation）的试验性领域之前，理解“试验”一词在这种情况下的含义以及如何解释它很重要。CSS3 的这些特定的元素已经使用多年了，但是仍被认为处于试验阶段，因为针对变形、渐变和动画的 CSS3 规范还没有正式定稿。不同 Web 浏览器的开发商通过特定于供应商的前缀开始着手规范的实现。

供应商前缀，比如“-webkit”（用于基于 WebKit 的浏览器，比如 Chrome 和 Safari）、“-moz”（用于 Mozilla Firefox）、“-ms”（用于 Microsoft Internet Explorer）和“-o”（用于 Opera），被浏览器开发商用于添加对试验特性的支持，但它是以一种临时（并且有时是特定的浏览器）的方式进行的。在规范稳定并定稿之后，并且在创建了关于特定特性的实现的规则时，浏览器供应商将以一种标准化的方式实现那些规则，而不再需要那些供应商前缀。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**注意：**你第一次遇到特定于供应商的前缀是在第 9 章学习 CSS3 栏时。

By the Way

在本章下面的示例中，将结合使用供应商前缀与标准的属性，使得示例在现在和将来都是有用的。书写带有供应商前缀的属性有一点麻烦，如下所示：

```
-webkit-transform: translate(100px) rotate(20deg);
-webkit-transform-origin: 60% 100%;
-moz-transform: translate(100px) rotate(20deg);
-moz-transform-origin: 60% 100%;
-ms-transform: translate(100px) rotate(20deg);
-ms-transform-origin: 60% 100%;
-o-transform:translate(100px) rotate(20deg);
-o-transform-origin:60% 100%;
```

用以代替标准的属性：

```
transform: translate(100px) rotate(20deg);
transform-origin: 60% 100%;
```

至少你可以确信覆盖的范围尽可能广。规范尽早会定稿，并且可以丢弃供应商前缀，可以在 http://caniuse.com 上检查方便的 Web 站点，看看所提供的支持随着时间的推移是如何改变的。

## 19.2 使用 2D 变形

当我们查看 Web 页面上的内容时，我们查看的是存在于相同平面上的二维（2D）对象，并且它们都具有长度和宽度。想展示你的小猫的照片吗？该图像具有高度和宽度，它是一幅 2D 图像。对于 HTML 容器元素也是如此， 、<p>和<section>元素都具有长度和宽度，可以指定和操纵它们。

使用 CSS3 执行 2D 变形允许更改这些元素的位置、长度和宽度，这是通过使用 CSS3 transform 属性平移、旋转和缩放元素实现的。在下面几节中将查看其中的每一种用例。

**注意：**对于本章中的所有图形，开始时元素都是静态放置的，然后才会做某件事情。在使用时，图形将显示元素的初始方位，但是应该在浏览器中打开示例 HTML 文件，查看代码的完全效果和元素的移动。

By the Way

### 19.2.1 平移元素

在 2D 空间中平移元素听起来比实际的有趣得多，平移（translate）只是意味着“移动”。使用 CSS3 transform 属性的 translate()函数允许沿着 *x* 轴（水平方向）或 *y* 轴（垂直方向）放置元素。这种方法似乎最容易理解，因此让我们从它开始入手并以此为基础。

图 19.1 显示了一个 2D 方框，它是程序清单 19.1 中的代码的结果。

程序清单 19.1 中的代码使用一些样式表条目来编排一个 ID 为 theBox 的 的样式，它的宽度和高度都是 250 像素，具有黑色的边框和灰色的背景，并且包含一些对齐的文本。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

但是第二个样式表条目用于 ID 为 theBox 的 的 hover 状态，它包含一些函数用于平移（移动）方框，这是使用 CSS3 transform 属性实现的。确切地讲，这些函数（其中每个函数针对一个供应商前缀，外加“标准”函数）告诉浏览器当把鼠标指针悬停在元素上时，就沿着 *x* 轴把元素平移（移动）75 像素，并沿着 *y* 轴平移 100 像素。

图 19.1

显示一个基本的 2D 元素




### 程序清单 19.1 对 2D 元素进行变形

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Translating a 2D Element</title>
    <style type="text/css">
    #theBox {
       height: 250px;
       width: 250px;
       background-color: #ddd;
       border: 1px solid #000;
       text-align: center;
       display:table-cell;
       vertical-align:middle;
       font-size: 150%;
       font-weight: bold;
     }
    #theBox:hover {
       -moz-transform: translate(75px,100px);
       -ms-transform: translate(75px,100px);
       -o-transform: translate(75px,100px);
       -webkit-transform: translate(75px,100px);
       transform: translate(75px,100px);
     }
    </style>
  </head>

  <body>
    <div id="theBox">
        I am a box!<br/>Watch me move!
    </div>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

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

你可能注意到移动是快速的并且不是非常平滑。如果你幻象从一个位置到另一个位置的平滑、轻松流畅的渐变，不要感到绝望，我们将在本章后面介绍如何使用 CSS3 渐变实现这种效果。目前，让我们只利用基本的 2D 和 3D 变形打下稳固的基础。

### 19.2.2 旋转元素

在 2D 空间中旋转元素意味着从原点开始转动它，并沿着顺时针方向从 0° 到 360° 继续旋转（也可以使用负的度数指示进行逆时针移动）。为了实现旋转，可以使用 CSS3 transform 属性的 rotate()函数，并且指定度数作为参数。例如，下面的代码将把一个元素沿着顺时针方向旋转 45°：

```
transform: rotate(45deg);
```

图 19.2 显示了一个 2D 方框，它开始时已经沿着逆时针方向旋转了 45°。程序清单 19.2 中的代码把它设置成右侧朝上。

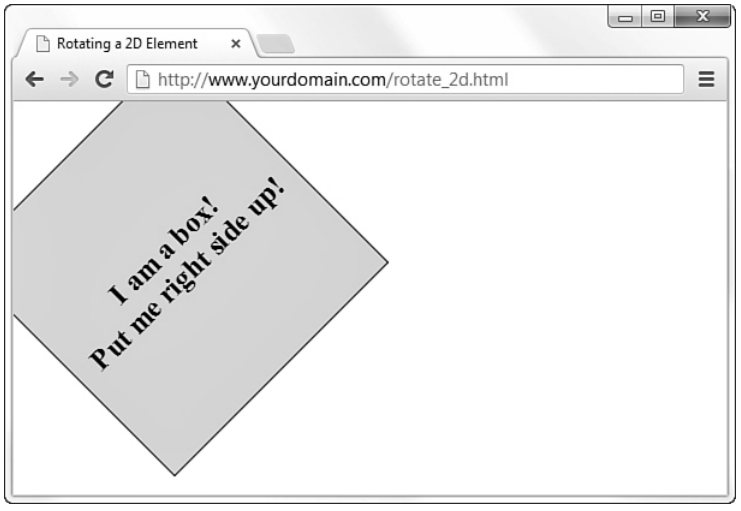



图 19.2 显示一个已经旋转过的基本 2D 元素

**程序清单 19.2 旋转一个 2D 元素**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Rotating a 2D Element</title>
    <style type="text/css">
    #theBox {
       height: 250px;
       width: 250px;
       background-color: #ddd;
       border: 1px solid #000;
       text-align: center;
       display:table-cell;
       vertical-align:middle;
       font-size: 150%;
       font-weight: bold;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      -moz-transform: rotate(-45deg);
      -ms-transform: rotate(-45deg);
      -o-transform: rotate(-45deg);
      -webkit-transform: rotate(-45deg);
      transform: rotate(-45deg);
    }
    #theBox:hover {
      -moz-transform: rotate(0deg);
      -ms-transform: rotate(0deg);
      -o-transform: rotate(0deg);
      -webkit-transform: rotate(0deg);
      transform: rotate(0deg);
    }
    </style>
  </head>
  <body>
    <div id="theBox">
       I am a box!<br/>Put me right side up!
    </div>
  </body>
</html>
```

**By the Way**

**注意：**所有的旋转（以及扭曲及其他变形）都开始于一个原点，可以使用 transform-origin 属性设置它。如果没有为 transform-origin 指定一个值，就使用默认值；默认值就好像在样式表中指定以下条目：

```
transform-origin: 50% 50%;
```

这个默认值意味着变形的原点应该位于沿着 *x* 轴和 *y* 轴各 50%的位置，或者恰好位于元素的中间。

与程序清单 19.1 一样，程序清单 19.2 中的代码使用一些样式表条目来编排 ID 为 theBox 的 的样式，它的宽度和高度都是 250 像素，具有黑色的边框和灰色的背景，并且包含一些对齐的文本。另请注意：我们使用了 transform 属性的 rotate()函数把该 的初始显示设置为−45°（逆时针 45°）。

第二个样式表条目用于 ID 为 theBox 的 的 hover 状态，并且它包含一些函数，用于把方框旋转回垂直的位置（0°）。如果把鼠标指针悬停在方框上，方框将顺时针旋转 45°，即从其−45°（或 315°，或者你更喜欢这样考虑它的话）的位置旋转到 0°的位置。

### 19.2.3 缩放元素

在 2D 空间中缩放一个元素意味着它看上去沿着 *x* 轴和 *y* 轴改变了大小。为了实现缩放，可以使用 CSS3 transform 属性的 scale()函数，并指定一个小于或大于 1 的值，这依赖于你想要如何缩放它——如果使用一个 0.01~0.99 之间的值，则会把元素缩放得比原始大小更小一些；如果使用一个大于 1 的值，则会把元素缩放得比原始大小更大一些。例如，下面的代码将把元素缩放成原始大小的一半：

```
transform: scale(.5);
```

可以使用以下语法，沿着 *x* 轴或 *y* 轴缩放元素：

```
transform: scale(value_for_x_axis, value_for_y_axis);
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

如果没有指定单独的值，那么就假定你打算同时沿着 *x* 轴和 *y* 轴以类似的方式缩放元素，这是典型的情况。因此，上一个示例中的 scale(.5)将沿着 *x* 轴和 *y* 轴同等地缩放元素。

程序清单 19.3 中的代码把我们迄今为止在示例中使用的基本方框缩放成它的一半大小。试一试吧！

**程序清单 19.3　缩放一个 2D 元素**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Scaling a 2D Element</title>
    <style type="text/css">
    #theBox {
       height: 250px;
       width: 250px;
       background-color: #ddd;
       border: 1px solid #000;
       text-align: center;
       display:table-cell;
       vertical-align:middle;
       font-size: 150%;
       font-weight: bold;
     }
    #theBox:hover {
       -moz-transform: scale(.5);
       -ms-transform: scale(.5);
       -o-transform: scale(.5);
       -webkit-transform: scale(.5);
       transform: scale(.5);
    }
    </style>
  </head>

  <body>
    <div id="theBox">
        I am a box!<br/>Watch me shrink!
    </div>
  </body>
</html>
```

与本章中以前的程序清单一样，程序清单 19.3 中的代码使用一些样式表条目来编排 ID 为 theBox 的 的样式，它的宽度和高度都是 250 像素，具有黑色的边框和灰色的背景，并且包含一些对齐的文本。第二个样式表条目用于 ID 为 theBox 的 的 hover 状态。它包含一些函数，用于把方框缩放成它的一半大小。如果希望把方框缩放到它的原始大小的 3 倍，将使用下面一行代码：

```
transform: scale(3);
```

如果希望只把方框的高度缩放到它的原始大小的 3 倍，将使用以下代码，并且会看到如图 19.3 所示的内容：

```
transform: scale(1, 3);
```

注意：不仅仅会缩放边框，容器元素内的一切内容也会被缩放，就像这个示例中的文本一样。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

图 19.3

显示一个基本的 2D 元素，并把它的高度缩放到它的原始大小的 3 倍




## 19.2.4 扭曲元素

在 2D 空间中扭曲一个元素是指沿着 *x* 轴或 *y* 轴把它扭曲一定的度数。要扭曲一个元素，可以使用 CSS3 transform 属性的 skewX()或 skewY()函数（依赖于你想要如何扭曲它），并指定一个范围在−90° ~90° 之间的角度值（以度为单位）。例如，下面的代码将沿着 *x* 轴把一个元素扭曲 45º：

```
transform: skewX(45deg);
```

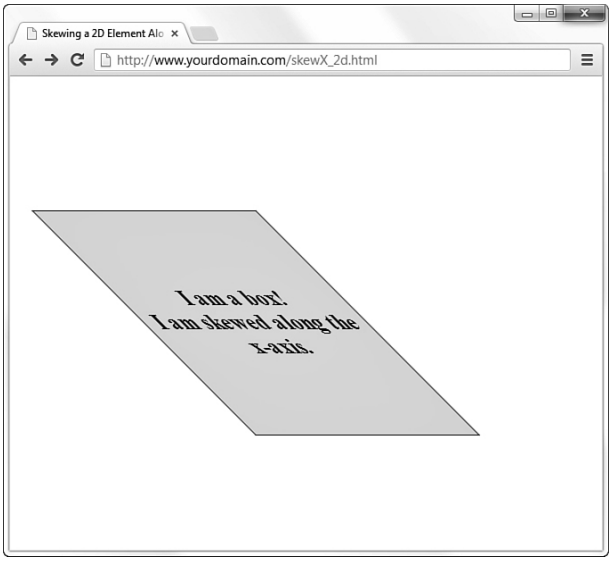
**By the Way**

**注意**：尽管可以指定一个−90° ~90° 之间的角度值，但是−90、90 或 0 这些值将导致一个察觉不到的元素。角度就在那里，并且会适当地扭曲，但是浏览器将不能呈现它。如果使用 1 或 89 的角度值，将开始明白为什么会这样。

图 19.4 和图 19.5 显示了我们在本章中一直使用的基本灰色方框，并沿着 *x* 轴然后沿着 *y* 轴扭曲到 45°（例如，根据需要使用 skewX(45deg)和 skewY(45deg)）。

图 19.4

显示一个沿着 *x* 轴扭曲的基本 2D 元素




异步社区会员 Kensuke(18319066421) 专享 尊重版权

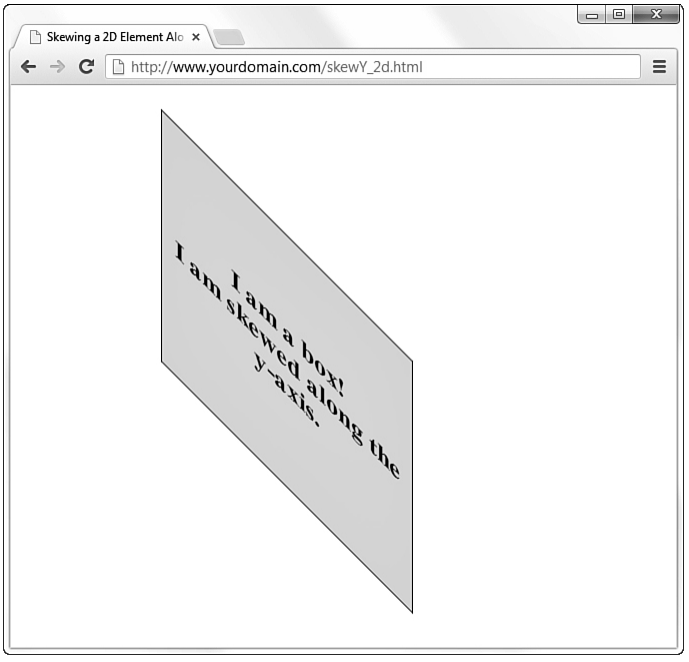



图 19.5

显示一个沿着 *y* 轴扭曲的基本 2D 元素

## 19.3 使用 3D 变形

如前所述，当我们查看 Web 页面上的内容时，实际上是在 2D 空间中查看它们。那么我们怎样才能从 2D 空间跃入 3D 空间（它实质上仍然是一个 2D 界面）呢？使用 CSS，我们可以创建一个空间，它通过给构成二维空间的高度和宽度添加深度，看上去好像是在三维空间中显示对象。在下面几节中，你将看到如何执行你在本章前面在 2D 空间中所执行的一些相同类型的变形，只是这里是在 3D 空间中进行的。

### 19.3.1 应用透视图

一般来讲，透视图（perspective）是以一种特定的方式绘制 2D 对象的过程，用以提供当从特定点（没影点[vanishing point]）查看时它们彼此之间的高度、宽度和深度的正确表示。

在处理 2D 对象时，使用的是 *x* 轴和 *y* 轴；现在，*z* 轴派上了用场。CSS perspective 属性的语法非常简单，如下所示：

```
perspective: length;
```

**注意：**默认情况下，元素的没影点位于元素的中心，但是可以使用 CSS perspective-origin 属性修改它，其方式与 transform-origin 具有默认值但是也可以手动设置它相同。

**By the Way**

长度的单位可以是像素或 em，但是必须大于 0。

也可以使用函数表示法，把 perspective()函数添加到 transform 属性的函数列表中，如下所示：

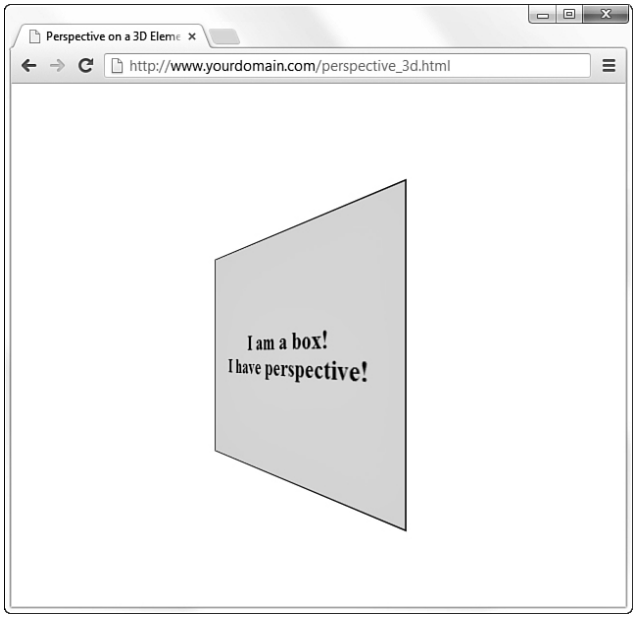
异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
transform: perspective(length) property(value);
```

对元素应用透视图本身不会在显示中产生任何差别，它必须发生在结合使用 3D 变形之一时，稍后将看到这一点。但是为了在继续学习之前更好地理解透视图，可以回过头看一眼图 19.2，它是一个旋转了−45° 的 2D 元素。把该图像与图 19.6 做一下比较。

图 19.6

具有 300 像素的透视值并且在 *y* 轴上旋转了−45° 的元素




你应该能够看到添加给对象的深度。在这里，perspective 值是 300 像素，它提供用于一种可见的效果。不过，在图 19.7 中，perspective 值是 3000px，这导致一种几乎察觉不到的 3D 效果，因为值越大，观察者与对象之间的距离也越大，从而导致产生透视图的更难以观测的外观。

图 19.7

具有 3000 像素的透视值并且在 *y* 轴上旋转了−45° 的元素




异步社区会员 Kensuke(18319066421) 专享 尊重版权

最后，把该对象与图 19.8 中的对象做比较，其中后者具有 100 像素的透视值——效果非常显著！

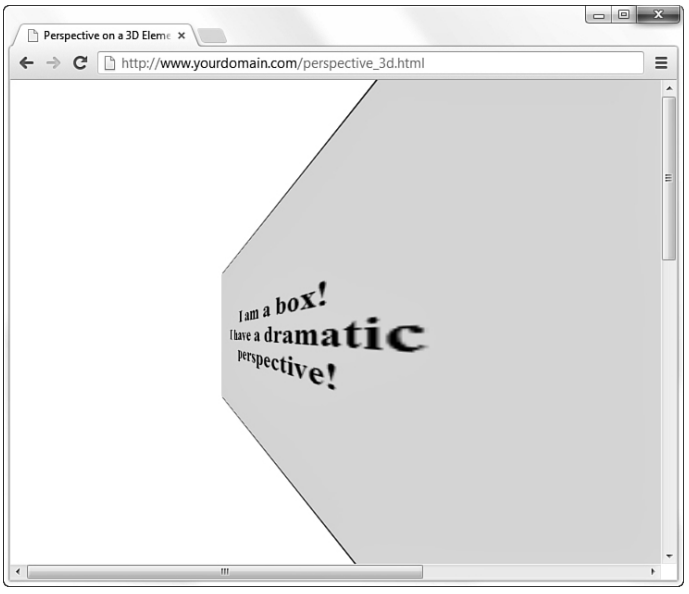



图 19.8

具有 100 像素的透视值并且在 *y* 轴上旋转了−45° 的元素

## 19.3.2 在 3D 空间中移动和旋转元素

在 3D 空间中移动（或平移）和旋转对象与在 2D 空间中执行相同的操作相当类似。在 2D 空间中，如你以前所看到的，可以使用 transform 属性的 translate()和 rotate()函数执行这些动作。在 3D 空间中，函数如下。

- rotateX(angle)、rotateY(angle)和 rotateY(angle)_rotateX(angle)：围绕合适的轴旋转对象，与 2D rotate(angle)变形函数形成对比，后者只是沿着 *x* 轴或 *y* 轴移动对象。当比较图 19.2 和图 19.6 时，就能看到这种区别。
- translateZ(value)：把对象定位到距离观察者更近或更远。

## 19.3.3 创建和操纵立方体

获取 3D 对象上的句柄的最佳方式是使之可以轻松地表示成一个 3D 对象——也许是 6 个面的立方体。利用 HTML 和 CSS 创建一个 3D 立方体似乎有些令人畏缩不前，但是我相信你将能够理解它，并把各个部分组合在一起（事实上，对于 6 个面的立方体，要把 6 个部分组合在一起）。

我开始创建了一个 3D 立方体，参见程序清单 19.4。这个程序清单的结果如图 19.9 所示。在程序清单后面，我解释了创建该立方体的每个步骤。

**程序清单 19.4 创建一个 3D 立方体**

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Creating a 3D Cube</title>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
<style type="text/css">
body {
   margin: 75px 0px 0px 75px ;
}

#wrapper {
   -moz-perspective: 500px;
   -webkit-perspective: 500px;
   perspective: 500px;
}

#the_cube {
   -moz-transform-style: preserve-3d
   -webkit-transform-style: preserve-3d
   transform-style: preserve-3d
}

#the_cube div {
   width: 150px;
   height: 150px;
   position: absolute;

   border: 1px solid black;
   background: rgba(200,200,200,0.5);

   font-size: 125%;
   font-weight: bold;
   line-height: 125px;
   text-align: center;
}

.front {
   -moz-transform: translateZ(75px);
   -ms-transform: translateZ(75px);
   -o-transform: translateZ(75px);
   -webkit-transform: translateZ(75px);
   transform: translateZ(75px);
}

.back {
   -moz-transform: rotateY(180deg) translateZ(75px);
   -ms-transform: rotateY(180deg) translateZ(75px);
   -o-transform: rotateY(180deg) translateZ(75px);
   -webkit-transform: rotateY(180deg) translateZ(75px);
   transform: rotateY(180deg) translateZ(75px);
}

.right {
   -moz-transform: rotateY(90deg) translateZ(75px);
   -ms-transform: rotateY(90deg) translateZ(75px);
   -o-transform: rotateY(90deg) translateZ(75px);
   -webkit-transform: rotateY(90deg) translateZ(75px);
   transform: rotateY(90deg) translateZ(75px);
}

.left {
   -moz-transform: rotateY(-90deg) translateZ(75px);
   -ms-transform: rotateY(-90deg) translateZ(75px);
   -o-transform: rotateY(-90deg) translateZ(75px);
   -webkit-transform: rotateY(-90deg) translateZ(75px);
   transform: rotateY(-90deg) translateZ(75px);
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    }

    .top {
       -moz-transform: rotateX(90deg) translateZ(75px);
       -ms-transform: rotateX(90deg) translateZ(75px);
       -o-transform: rotateX(90deg) translateZ(75px);
       -webkit-transform: rotateX(90deg) translateZ(75px);
       transform: rotateX(90deg) translateZ(75px);
    }

    .bottom {
       -moz-transform: rotateX(-90deg) translateZ(75px);
       -ms-transform: rotateX(-90deg) translateZ(75px);
       -o-transform: rotateX(-90deg) translateZ(75px);
       -webkit-transform: rotateX(-90deg) translateZ(75px);
       transform: rotateX(-90deg) translateZ(75px);
    }
    </style>
  </head>

  <body>

    <div id="wrapper">
       <div id="the_cube">
          <div class="front">FRONT</div>
          <div class="back">BACK</div>
          <div class="left">LEFT</div>
          <div class="right">RIGHT</div>
          <div class="top">TOP</div>
          <div class="bottom">BOTTOM</div>
       </div>
    </div>
  </body>
</html>
```

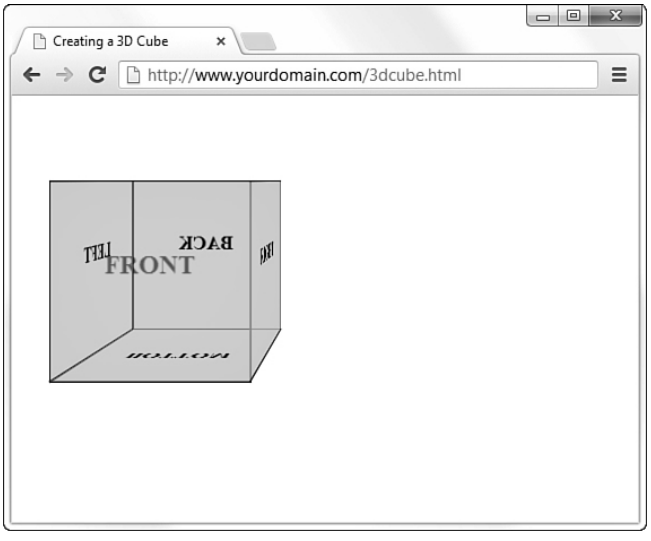



图 19.9

利用 HTML 和 CSS 创建的一个基本的 3D 立方体

这一大段代码全都是样式表条目，由于需要为多个 CSS 属性使用特定于供应商的条目，因此它们本身显得比较长。目前，可以跳过所有这些条目，并且更细致地查看 集。

- ：用它自身包装所有其他的 ，以及充当 3D 空间，并为其设置一个透视值（在这里是 500px）。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

- ：通过使用 transform-style 属性并将其值设置为 preserve-3d，为我们打算保持为三维形状的物体建立舞台。
- （及其类名对应于立方体的各个面的所有其他的 ）充当立方体的一个面。

**By the Way**

**注意：**即使使用供应商前缀，处理变形也是一件复杂的事情，并且可能随着所发布的浏览器不同版本而改变。如果你的结果与我的稍有差异，也不要感到惊惶失措，尤其是当你使用非 Chrome 浏览器时。可以把它视作一个对代码进行较小的改动以设法使之工作的好机会，处理代码的目的只是为了查看所发生的事情，专业人员也是这么做的，并且不要让任何人告诉你其他情况！

现在回过头来更细致地查看样式表，从用于所有 的条目开始，这些 都是 ID 为#the_cube 的 的子元素——换句话说，它们是立方体的每一个面：

```
#the_cube div {
   width: 150px;
   height: 150px;
   position: absolute;

   border: 1px solid black;
   background: rgba(200,200,200,0.5);

   font-size: 125%;
   font-weight: bold;
   line-height: 125px;
   text-align: center;
}
```

这个条目确保立方体的每一个面都具有相同的高度、宽度、边框、背景色和字体属性。在图 19.9 中可以看到，立方体的各个面的确看上去是相同的（除了在每个面上打印的特定文本以外）。

现在看看比较困难的部分：逐一处理每个面的变形。为了避免混乱，在下面引用样式表条目时，我将只使用基本的属性（如 transform），而不是所有特定于供应商的版本，在程序清单 19.3 中可以查看它们。让我们从正面开始：

```
.front {
   transform: translateZ(75px);
}
```

正面是立方体上我们希望首先看到的那个面，因此我们不会旋转它或者应用任何变形，只是在 *z* 轴上把它前移它的一半长度或 75 像素。这似乎足够简单。如果你想查看当更改了 translateZ()的值时所发生的事情，那就这么做吧，你将注意到立方体的各个面之间有间隙。

接下来我们开始旋转立方体的其余的面，使得它们的边缘与所有其他的面对齐。看看用于左面的条目：

```
.left {
   transform: rotateY(-90deg) translateZ(75px);
}
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

在这里，150×150 像素的正方形面维持 500 像素的透视值，它是为 3D 视域设置的，并在其 *y* 轴上旋转了−90°。把这个面在 *z* 轴上前移 75 像素，使其边缘对齐，如图 19.10 所示。

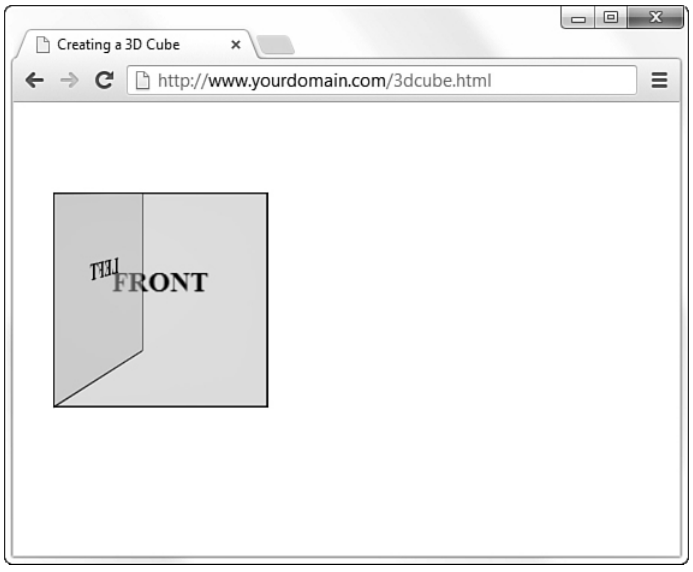



图 19.10
对齐 3D 立方体的正面和左面

其余的面是以类似的方式定义的：首先将它们沿着 *x* 轴或 *y* 轴旋转到位，然后沿着 *z* 轴移动它们：

```
.back {
   transform: rotateY(180deg) translateZ(75px);
}
.right {
   transform: rotateY(90deg) translateZ(75px);
}
.top {
   transform: rotateX(90deg) translateZ(75px);
}
.bottom {
   transform: rotateX(-90deg) translateZ(75px);
}
```

结果就是你在图 19.9 中看到的 3D 立方体。为了更好地感受沿着 *x* 轴或 *y* 轴旋转立方体的面是如何工作的，我强烈建议你改变 rotateX()和 rotateY()的值，查看各个面有没有对齐。在本章后面，我们将在这个 3D 立方体示例的基础上学习利用 CSS3 动画使之持续自旋。

## 19.4 实现 CSS3 渐变

在本章前面，你看到了当用户把鼠标指针悬停在 2D 元素上时移动、旋转和缩放它们的一些示例。当时，你可能注意到移动的速度很快并且不是那么平滑，并且我承诺在本章后面的部分修正这个问题。现在就将介绍修正的方法，这是通过使用 CSS 渐变实现的。

当在 CSS 属性之间移动（比如从对象的初始状态移到一种缩小的状态）时，就会发生渐变。如果不控制渐变（本章中的最初几个示例就是这样），用户将会看到第一种状态，然

异步社区会员 Kensuke(18319066421) 专享 尊重版权

后就会立即看到另一种状态，从而使“渐变”几乎感觉不到。

为了控制渐变，可以使用 CSS transition 属性（不要忘记用于 WebKit、Mozilla 和 Opera 的合适的供应商前缀，Internet Explorer 不需要供应商前缀），它使用以下语法：

```
transition: property duration timing-function delay;
```

尽管 property 指将进行渐变的属性的名称——例如，使用 transform 缩放对象——duration、timing-function 和 delay 这些值仍然是你所不熟悉的。duration 值是两种状态之间的持续时间（以秒为单位），timing-function 是帮助你定义渐变加速的几个函数之一（例如，是在渐变的开始还是结束时速度比较快），delay 是渐变开始前的时间（以秒为单位）。

**By the Way**

**注意：**还可以单独指定属性，而不是把它们都放在 transition 属性中，可以在样式表中单独的行上使用 transition-property、transition-duration、transition-timing-function 和 transition-delay 这些属性。

参考程序清单 19.3，它把一个 2D 元素缩放成其原始大小的一半。为了控制元素与其收缩版本之间的渐变，将在用于#theBox 的样式表条目中添加下面这几行代码：

```
-moz-transition: -moz-transform 5s ease-in 1s;
-o-transition: -o-transform 5s ease-in 1s;
-webkit-transition: -webkit-transform 5s ease-in 1s;
transition: transform 5s ease-in 1s;
```

这些条目告诉浏览器通过等待 1 秒钟然后用 5 秒钟的时间缓入成一个新值，来渐变到 transform 属性的新值（在这里，它是 scale(.5)）。如果在浏览器中加载修改过的代码，将会看到：的确，当把鼠标指针悬停在 2D 方框上时，它将暂停片刻，然后缓慢地开始收缩，5 秒钟后结束。不可能在一幅静态的图像中显示这个过程，因此你必须自己试试。

有关 CSS3 渐变的更多信息，尤其是用于 transition-timing-function 的选项，参见 https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Using_CSS_transitions。

## 19.5 初识 CSS3 动画

可以把 CSS3 动画视作一组加强的渐变控制。在屏幕上发生的过程的确是属性之间的渐变，但是它更多地考虑的是 CSS animation 属性及其子属性，而不是 CSS transition 属性。

在第 10 章中，你可能回想起关于创建动画式 Web 图形的简短讨论。确切地讲，创建动画式 Web 图形的第一步是创建一系列图像，然后一个接一个地显示它们，就像电影和动画片把单独的帧放在一起创建移动的图片一样。在 CSS3 动画中，通过定义关键帧（keyframe）来构建动画，这些关键帧本身定义了浏览器在动画序列期间应该如何显示特定的元素。

在样式表中使用以下语法定义关键帧（不要忘记用于全部 4 个供应商的合适的供应商前缀）：

```
@keyframes name_of_keyframe {
   from {
      property: value;
  }
  to {
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    property: value;
  }
}
```

为了使用一个实用的示例，让我们返回到在程序清单 19.3 中创建的 3D 立方体上来。假设你想持续自旋该立方体，使之好像是在空中翻滚，这将需要围绕 *x* 轴和 *y* 轴从 0° 到 360° 旋转它：

```
@keyframes spin {
    from { transform: rotateY(0deg) rotateX(0deg); }
    to { transform: rotateY(360deg) rotateX(360deg);}
}
```

因此创建一个名为 spin 的关键帧，我们将需要在 animation 属性中引用它。在这个示例中，animation 属性将应用于#the_cube 的定义。看看完整的示例：

```
#the_cube {
   -moz-transform-style: preserve-3d;
   -webkit-transform-style: preserve-3d;
   transform-style: preserve-3d;

   -webkit-animation: spin 5s infinite linear;
   -moz-animation: spin 5s infinite linear;
   -ms-animation: spin 5s infinite linear;
   -o-animation: spin 5s infinite linear;
   animation: spin 5s infinite linear;
}
```

在这个示例中，用于 animation 属性的语法如下：

```
animation: keyframe_name time iteration_count timing_function
```

在这个示例中，我把动画定义成无限地持续自旋，但是将在 5 秒钟内完成其线性旋转。如果我把 linear 改为 ease，你将注意到在每次迭代之前都会稍稍有一点停顿。与关于渐变一节中的前一个示例一样，不可能在静态的图像中显示这个过程——你必须自己试试它。

要学习关于 animation 属性的众多子属性的更多知识，我建议访问 https://developer.mozilla.org/en-US/docs/Web/CSS/animation。

## 19.6 小结

本章介绍了移动、旋转、缩放和扭曲 2D 和 3D 元素（以及其中的内容）的基础知识。你获悉了通过 2D 元素创建 3D 元素开始于透视图的应用，并且看到了如何控制该透视图以实现想要的外观。最后，你执行了控制 CSS 属性之间的渐变的前几个步骤，并且使用了两组不同的渐变和动画控制。

## 19.7 问与答

**问：如果所有这些 CSS 都是试验性的，那么使用它真的安全吗？**

**答：**HTML5 和 CSS3 的大部分规范仍然被标记为试验性的（事实上，HTML5 仍然只是一个建议，还没有官方的规范），然而设计师使用这样的元素有几年的时间了，因为浏览器

异步社区会员 Kensuke(18319066421) 专享 尊重版权

包括针对试验性元素的呈现支持。如果在实现试验性特性时使用特定于供应商的前缀，并为甚至较老的浏览器提供退而求其次的选项，那么将无需等待规范最终定稿，并且如果你确定试验性功能可以给用户提供更好的体验，就可以凭空设想利用了它们的设计和交互性。与以往一样，继续在你的站点用户当中最流行的浏览器中检查和测试站点，以确保他们具有你所想要的查看体验。

## 19.8 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 19.8.1 问题

1．怎样把一个 2D 元素沿逆时针方向旋转 90°？确保在答案中使用供应商前缀。

2．scale(3)和 scale(1,3)之间的区别是什么？

3．z 轴代表什么？

### 19.8.2 答案

1．可以使用以下代码：

```
-moz-transform: rotate(-90deg);
-ms-transform: rotate(-90deg);
-o-transform: rotate(-90deg);
-webkit-transform: rotate(-90deg);
transform: rotate(-90deg);
```

2．第一个示例在宽度和高度上同等地把元素缩放到其原始大小的 3 倍；第二个示例则保持宽度与原来的相同，但是把高度缩放到原始大小的 3 倍。

3．z 轴代表深度。

### 19.8.3 练习

- 花一些时间处理 3D 立方体示例，并且修改 6 个面的大小、颜色和放置方式，以更好地感受这个立方体的各个部分是如何组合在一起的。
- 研究可供 transition 和 animation 属性使用的众多选项（分别参见 https://developer.mozilla.org/en-US/docs/Web/CSS/transition 和 https://developer.mozilla.org/en-US/docs/Web/CSS/animation），并且修改 2D 缩放和旋转以及 3D 持续自旋立方体示例，利用这些属性的不同方面。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 20 章

# 创建打印友好的 Web 页面

在本章中你将学到：

- 什么构成了打印友好的页面；
- 怎样应用特定于媒体的样式表；
- 怎样为打印页面创建样式表；
- 怎样在 Print Preview（打印预览）模式中查看 Web 页面。

如果你曾经在前往一个没有智能手机信号的地区旅行之前使用了诸如 MapQuest 或 Google Maps 之类的网上地图应用程序，那么无疑将体验打印 Web 页面的需要。类似地，只在网上提供的优惠券发放、来自网上经销商的商品购买收据、基于 Web 的航班办理登机手续以及从家用计算机打印登机牌的能力增加了对打印友好的页面的需求。的确，并非所有的 Web 页面都完全是为了在屏幕上查看而设计的。

你可能没有意识到这一点，但是为那些希望打印一份副本以便脱机阅读的用户明确设计并提供页面的打印友好的版本是可能的，Google Maps 在显示了内容的屏幕上的版本之后就会提供一个这样的打印版本。我是所有的事情都无纸化的超级粉丝，但是一些人恰恰更喜欢拥有一些文档（和方位）的打印稿备份。使用 CSS，很容易创建可以基于查看方式而改变外观的 Web 页面。在本章中，你将学习如何创建这样的页面。

Try It Yourself：审查内容是否是打印友好的

在学习本章的过程中，考虑你自己的看起来可能适合打印的 Web 页面。然后考虑你将希望更改它们，以使它们在打印的页面上看起来甚至更好。下面给出了一些要考虑的思想。

- 甚至违背本书前面章节中的警告，你的页面使用了图案式的背景图像或者与文本形成鲜明对比的不同寻常的背景色吗？由于背景，这类页面可能难以打印，因此要考虑提供页面的打印版本，它没有使用自定义的背景图像或颜色并且只简单地使用黑

异步社区会员 Kensuke(18319066421) 专享 尊重版权

色的文本。当准备页面以进行打印时，如果可能，就要坚持在白色的背景上使用黑色的文本。

- 你的页面包括许多链接吗？如果是，就要考虑为打印更改链接的外观，使得它们不怎么显眼。例如，删除任何下划线。记住，你不能单击一张纸！
- 最后，页面上的每一幅图像都绝对必不可少吗？在大多数打印机上，彩色的图像在打印时都要耗费昂贵的油墨，因此要考虑从打印友好的页面中去掉一些（如果不是全部）图像。

## 20.1 是什么使页面成为打印友好的

有一些 Web 页面已经是打印友好的，指出这一点很重要。如果页面使用白色的背景以及对比鲜明的深色文本和少量的图像，那么你自己甚至可能不需要关注特殊的打印友好的版本。另一方面，事实证明，具有深色背景、动态链接和多幅图像的页面可能是普通的打印机难以处理的（或者只会令阅读者恼火）。

在考虑怎样使页面更加打印友好时要记住的主要事情是由媒介强加的限制。换句话说，是什么把打印的页面与计算机屏幕独特地区分开？显而易见的区别是大小——打印的页面具有固定的大小，在美国通常是 8½×11 英寸，而屏幕的大小则可能变化很大。打印的页面还具有颜色限制（甚至在彩色打印机上也是如此）。当用户确实只希望打印页面上的文本时，只有极少的人想要浪费打印全彩背景所需的油墨。

大多数用户还只对打印充当页面焦点的文本感兴趣，而对其他内容不感兴趣。例如，图 20.1 显示了一条在地图上标出的从密苏里州的独立城到俄勒冈州的俄勒冈城的旅行路线（近似于历史上的俄勒冈小道）。

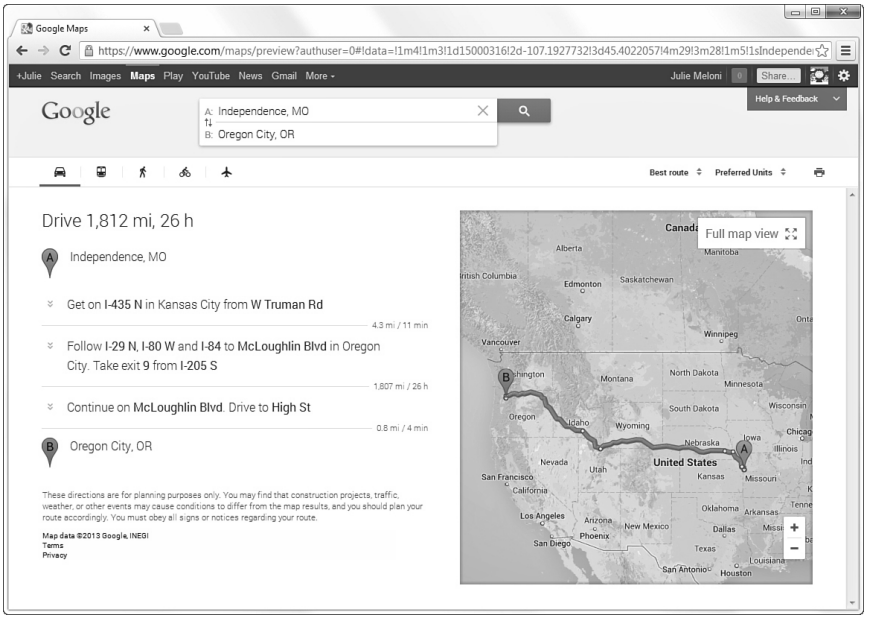



图 20.1

由于具有表单输入元素并且大图像具有它自己的显示控件，因此这个页面并不是非常打印友好

图 20.1 中的页面包含表单输入框、一幅可以控制自身（移动、缩放）的大图像，以及你在 Web 内容中所期望的其他辅助项。Google 施展了一点魔法（你也将学习如何这样做），

异步社区会员 Kensuke(18319066421) 专享 尊重版权

以确保在打印这个页面时，其中一些辅助项会消失不见，并且在网上查看时当前隐藏的更多信息会实际地显示出来。如果没有利用这一点魔法——包括打印友好的样式表——那么在这段 1800 英里的旅程中你所考虑到的只有两次转向，而不是在打印页面时所看到的大量的方位。如果单击页面主体中的 Print 链接，或者使用 Web 浏览器的打印功能，那么 Web 浏览器将会显示一个 Google 明确地格式化成用于打印的页面，如图 20.2 所示。

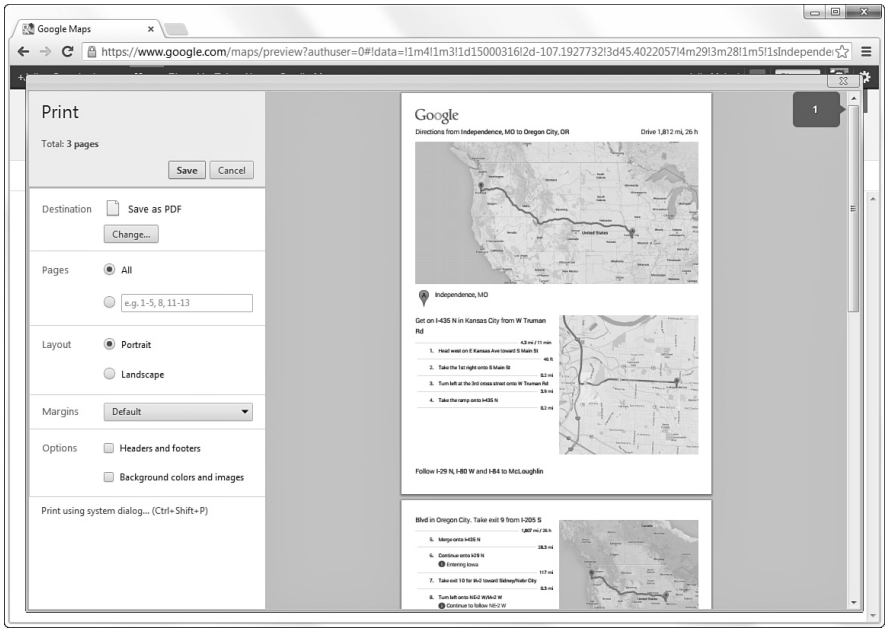



图 20.2

页面的打印友好版本的预览包括所有的驾驶指导（即使它们没有显示在你的屏幕上），并且删除了其他的页面元素

如图所示，这个页面的打印友好的版本代表了在原始页面基础上的改进，至少从打印机的角度讲是这样。所有的表单输入元素都被删除了，Web 站点本身的导航元素也删除了，并且使所有的方位都是可见的。有人可能坚持更打印友好的版本甚至还会删除所有的图像。

**By the Way**

**注意：**如果特定于打印机的页面的字体是无衬线字体（sans serif），一些 Web 设计师建立把字体改为衬线字体（serif），一般认为这种字体在打印稿中更容易阅读。如果在页面上使用一种无衬线字体，就由你自己决定是否希望在打印页面时维持它的核心外观——这意味着不要把字体改为衬线字体。

本着让你更好地理解在转向创建打印友好的页面时（确切地讲）需要考虑什么的精神，下面列出了你至少希望考虑的一些改变。

- 如果页面具有背景的话，就删除它，给打印的页面提供一个白色的背景。一些浏览器将自动负责处理这个任务，但是没有理由不能明确地指出可以包容那些不会自动删除背景的较老的浏览器。
- 把文本颜色改为黑色，在文本中具有某种颜色是错的，但是黑色是首选颜色。
- 确保字体足够大，使得页面在打印时可以轻松地阅读。你可能必须测试一些不同的大小。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

- 删除链接格式化效果，或者简单地恢复为基本的加下划线的链接。一些设计师喜欢保留下划线，只是为了使访问者知道原始页面中存在链接。
- 删除任何和所有不是必不可少的图像。这通常包括对于传达页面中的内容不是至关重要的任何图像，比如导航按钮、广告和动画式图像（它们当然不会在打印的页面上展示为动画）。

除了这些建议之外，你还可能发现添加一个署名行是有用的，它带有关于页面作者的信息，以及一个用于页面的 URL 和版权信息。当用户离开你的 Web 站点并且手中只有页面的打印版本时，可能会丢失这些信息。

你可能现在还不需要对页面执行这些改变。这里的思想是：播下构成打印友好的页面的种子，以便你可以更好地创建特定于打印机的样式表。这是正确的，创建一个仅当打印时才应用于页面的样式表——在下一节中将学习这方面的知识。

## 20.2 应用特定于媒体的样式表

图 20.1 显示了一家公司（尽管在这种实践中 Google 并不孤单）使用链接到打印友好的页面的小打印机图标，使用户能够预览然后打印更适合于打印而不是在线阅读的页面。这种特性对于用户是有帮助的，因为他们也许不想受到打印页面困扰，并且不希望在与他们想阅读的内容一起提供的图形和广告上浪费纸张和油墨。尽管打印机图标和链接方法很直观并且工作得非常好，这里还是提供了一个选项，针对的是不需要创建单独的打印友好的内容。

第二个选项涉及使用特定于打印的样式表，当用户选择打印页面时它会自动应用于页面。第 15 章中讲到，CSS 支持特定于媒体的样式表（media-specific style sheets）的概念，它们是以特定的媒介（比如屏幕或打印机）为目标的样式表或样式表条目。不过，CSS 并没有在这两种形式的媒体上止步不前。检查下面的特定媒体类型的列表，CSS 使你能够利用独特的样式表支持它们。

- all：用于所有设备。
- aural：用于语音合成器。
- braille：用于盲人触觉反馈设备。
- embossed：用于分页式盲人打印机。
- handheld：用于屏幕大小和带宽均有限的手持式设备。
- print：用于在 Print Preview（“打印预览”）模式下在屏幕上查看的打印材料和文档。
- projection：用于放映的演示文稿。
- screen：用于彩色计算机屏幕。
- tty：用于使用固定字宽的字符网格的设备（比如终端、电传打字机或者显示屏大小有限的手持式设备）。
- tv：用于电视类型的设备，它们通常具有较低的分辨率并且是彩色的，同时具有有限的滚动能力。

这些媒体中，也许最有趣的媒体是 aural 类型，用于可以大声阅读或倾听的 Web 页面。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

显然，CSS 的架构师构想的 Web 要比大多数人在主要为计算机屏幕设计页面时所考虑的要宽泛得多。尽管你可能现在还不需要过多地关心语音型 Web 页面设计，但是如果你选择一种 Web 设计方面的职业，则它对于你可能需要顾及到什么充当了一个优秀的“参谋”。

在应用于其他媒体时，关于样式表的好消息是：它们不需要你学习任何新知识。不错，或许对于语音型 Web 页面来说，你将需要学习几种新技巧，但是就目前而言，可以使用你已经学过的相同样式属性来创建特定于打印的样式表。技巧在于知道如何为特定的媒介应用样式表。

回忆可知，<link />标签用于把外部样式表链接到 Web 页面。这个标签支持一个名为 media 的属性，它用于指定应用样式表的媒介的名称。默认情况下，该属性被设置为 all，这意味着如果没有指定别的媒体，那么外部样式表将用于所有的媒体。其他可接受的属性值对应于前面的列表中提供的媒体列表。

为 Web 页面建立特定于打印的样式表涉及使用两个<link />标签，其中一个用于打印机，另一个用于其余的所有媒介。下面的代码用于处理这项任务：

```
<link rel="stylesheet" type="text/css" href="standard.css" media="all" />
<link rel="stylesheet" type="text/css" href="for_print.css" media="print"/>
```

**By the Way**

**注意：**也可以使用@import 命令链接特定于媒体的样式表。例如，下面的代码就像前面的<link />代码一样工作：

```
@import url(standard.css) all;
@import url(for_print.css)
print;
```

在这个示例中，将两个样式表链接到 Web 页面。第一个样式表通过把 media 属性设置为 all，把所有的媒体都作为目标。如果没有做其他任何事情，standard.css 样式表将应用于所有的媒体。不过，第二个样式表的存在导致 for_print.css 样式表将用于打印页面。

可以在单个<link />标签中指定多种媒体类型，并用逗号隔开这些类型即可，如下所示：

```
<link rel="stylesheet" type="text/css" href="for_pp.css"
      media="print, projector" />
```

这段代码导致将 for_pp.css 样式表只应用于 print 和 projector 媒体类型，而别无其他。

**Watch Out!**

**警告：**在前面的代码中，你可能忍不住要在第一个链接的样式表中指定 media="screen"。尽管这适合于在正常的 Web 浏览器中查看页面，但是如果用户使用手持式浏览器或者任何其他类型的媒体查看页面，那么它将引发问题。换句话说，样式表只将应用于在 media 属性中提及的特定媒体类型，而不会应用于其他任何类型。

## 20.3 设计用于打印页面的样式表

使用打印友好的 Web 页面所需的建议修改列表，现在应该尝试创建一个打印友好的样式表。让我们首先查看一个使用正常的（screen）样式表显示的页面，如图 20.3 所示。

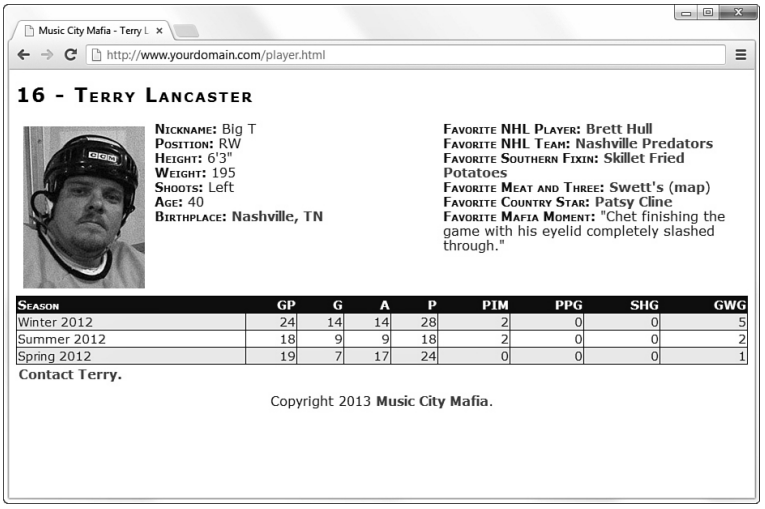
这幅图呈现了页面在正常的 Web 浏览器中的外观。实际上，这个页面还远没有做好打

异步社区会员 Kensuke(18319066421) 专享 尊重版权

印的准备，但是它仍然可以受益于某些改进。

图 20.3

在正常的 Web 浏览器中查看的一个利用 CSS 编排样式的页面




下面的更改可能有助于使这个 Web 页面更加打印友好。

- 把所有文本的颜色都改为黑色。
- 删除链接格式化效果（加粗和颜色）。
- 把两位运动员的信息区域垂直码放整齐，因为在打印的页面上不太可能水平地放下它们。
- 完全删除联系人链接。

对正常样式表的前两处修改比较直观，它们主要涉及更改或撤消现有的样式。不过，第三处修改需要稍做一点思考。因为你知道打印的页面是固定大小的，因此应该为打印页面上的所有元素使用绝对定位。这使得把内容区域精确地放置到希望它们出现的位置要容易得多。最后，列表中的最后一项非常容易实现：只需简单地把 contact 元素的 display 样式属性设置为 none 即可。

**By the Way**

**注意：**即使没有链接到外部版本，也可以为样式表指定一种媒体类型。<style>标签也使用与<link />标签相同的 media 属性。

**Watch Out!**

**警告：**尽管绝对定位适合于冰球运动员示例页面，但它对于编排特定于打印的页面的样式并非总是一个好主意。更确切地讲，如果页面中包含的内容值得使用多个打印页面，那么最好使用相对定位，并且让内容在多个页面上流动。

程序清单 20.1 显示了用于 player_print.css 样式表的 CSS 代码，它把这些修改纳入到一个样式表中，该样式表非常适合于打印冰球运动员页面。

**程序清单 20.1 用于特定于打印的冰球运动员样式表的 CSS 代码**

```
body {
  font-family:Verdana, Arial;
  font-size:12pt;
  color: #000000;
}
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
h1 {
  font-size:18pt;
  font-weight:bold;
  font-variant:small-caps;
  letter-spacing:2px;
  position:absolute;
  left:0in;
  top:0in;
}

img {
  position:absolute;
  left:0in;
  top:0.5in;
}

footer {
  position:absolute;
  text-align:left;
  left:0in;
  top:9in;
}

a, a:link, a:visited {
  color: #000000;
  font-weight:normal;
  text-decoration:none;
}

div {
  padding:3px;
}

div.player_info {
  position:absolute;
  left:1.75in;
  top:0.5in;
}

div.player_favorites {
  position:absolute;
  left:1.75in;
  top:2in;
}

div.contact {
  display:none;
}

.label {
  font-weight:bold;
  font-variant:small-caps;
}

table, tr, td {
  border: 1px solid black;
  border-collapse: collapse;
}

table.stats {
  width:100%;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
  text-align:right;
  font-size:11pt;
  border: 1px solid black;
  border-collapse: collapse;
  position:absolute;
  left:0in;
  top:4in;
}

thead {
  font-variant:small-caps;
  background-color: #000000;
  color:white;
}

tr.light {
  background-color: #ffffff;
}

tr.dark {
  background-color: #eeeeee;
}

th.season, td.season {
  text-align:left;
}
```

这段代码中可能最大的区别是：它使用英寸（in）作为通过 CSS 发生的所有绝对定位的度量单位。当你考虑我们依据英寸（而不是像素）来规划页面时，这就是有意义的。如果仔细研究代码，你将注意到文本都是黑色的，从链接中删除所有特殊的样式格式化效果，并且内容区域现在是绝对定位的（使得它们精确出现在希望它们出现在一张纸上的位置）。

**By the Way**

**注意：**在程序清单 20.1 中，在一些情况下使用了 0in 的左边距。你可能发现浏览器需要一些边距，否则它将会切除一些内容。如果是这样，只需把它调整为 0.1in 或 0.25in 或者某个其他的值——你仍然应该能够领会放置内容的意义。

## 20.4 在 Print Preview（打印预览）中查看 Web 页面

图 20.4 显示了在浏览器的 Print Preview（打印预览）窗口中查看的冰球运动员页面的打印友好的版本。

仅仅为了进行良好的度量，如图 20.5 所示，显示了完整的打印页面（作为 PDF），打印预览是准确的！

**By the Way**

**注意：**Adobe 的虚拟打印机可以用于把冰球运动员 Web 页面打印到 PDF 文档。你还可能找到诸如 DoPDF（www.dopdf.com）之类的 PDF 转换器，它们将以比 Adobe Acrobat 软件更低的费用为你所用。打印到 PDF 实际上将以一种可以轻松地以电子方式共享的格式创建一个打印友好的 Web 页面版本，以便进行打印。

要了解关于 Acrobat 的更多信息，可以访问 http://www.adobe.com/products/acrobat.html。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

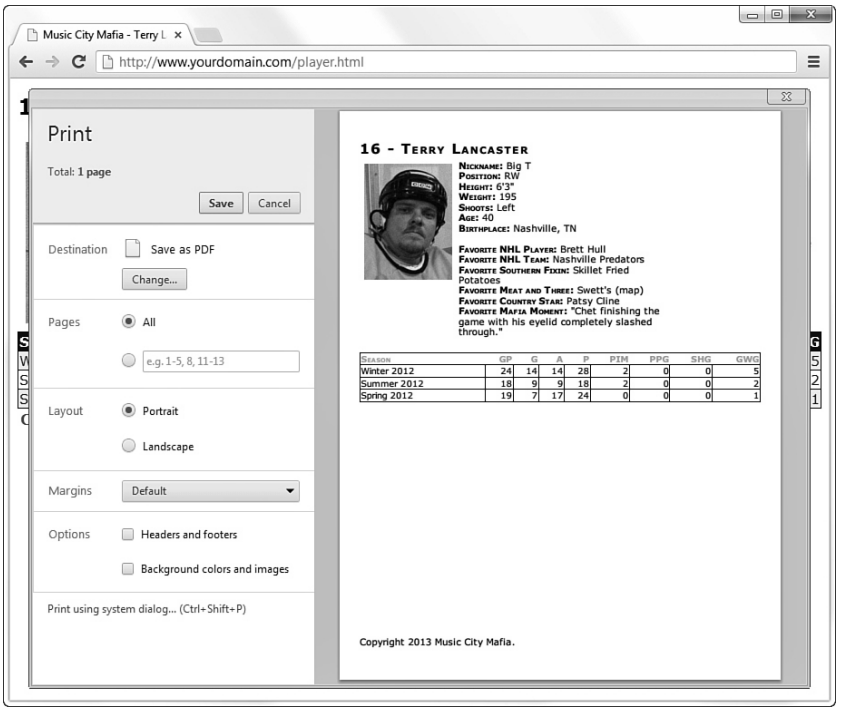



图 20.4

在打印 Web 页面前，可以使用 Print Preview（打印预览）查看它的打印友好的版本

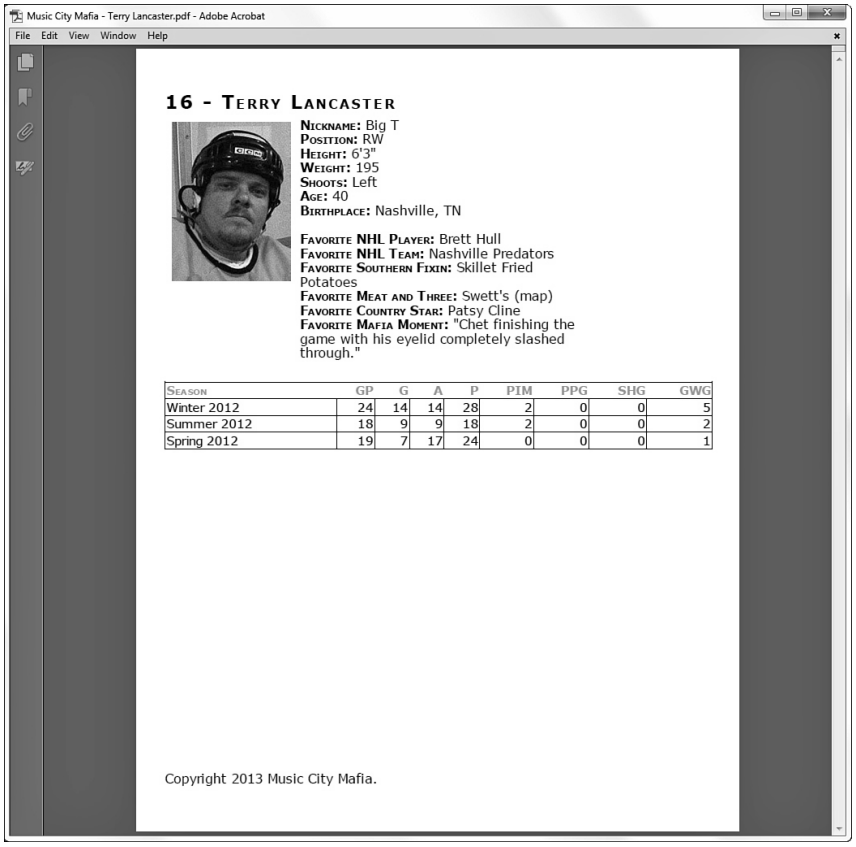



图 20.5

通过把冰球运动员页面打印（另存）为 Adobe PDF，把它转换成一个 PDF 文档

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 20.5 小结

本章重点介绍了 CSS 解决公共需求的实际应用，即打印 Web 页面。本章首先介绍了构成打印友好的 Web 页面的准确成分，然后介绍构建到 CSS 中的机制，它允许区分使用不同媒体呈现的页面，然后讲解了如何相应地选择样式表。

最后，你创建了一个特定于打印的样式表，用于编排仅为了进行打印的页面的样式。尽管大多数用户更喜欢在较大的计算机屏幕上查看页面而不是在纸上阅读它，但是有时打印 Web 页面是必需的。通过提供打印友好的页面，确保给 Web 页面访问者提供最大限度的灵活性。

## 20.6 问与答

**问：我可以使用<link />标签的 media 属性创建明确用于在手持式设备上查看 Web 页面的样式表吗？**

**答：**是的。通过把<link />标签的 media 属性设置为 handheld，利用样式表明确地把手持式设备作为目标。不过，一些设备不会适当地或者一致地解释它，这就是为什么许多设计师结合使用 screen 与额外的查询来确定宽度乃至布局的原因。

**问：我仍然需要在页面上提供打印机图标以使得它们能够被打印吗？**

**答：**不需要，那只是一个方便的视觉提示，而不是必需的。你在本章中学习的链接的样式表技术允许支持打印友好的 Web 页面，而无需页面上具有任何特殊的链接。不过，如果你希望使用户能够在浏览器中查看页面的打印友好的版本，可以链接到页面的另一个版本，它使用特定于打印的样式表作为它的主（浏览器）样式表。或者可以在页面上提供某个“良好打印”图标，指示用户使用浏览器的 Print Preview（打印预览）特性查看页面的打印友好的版本。

## 20.7 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 20.7.1 问题

1. 具有一个指向打印友好的页面的按钮是否意味着页面实际上是打印友好的？
2. 如果没有指定任何 media 属性，那么在链接到页面的外部样式表上会发生什么事情？
3. 怎样把名为 freestyle.css 的样式表链接到页面，使得仅当在电视上查看页面时才会应用它？

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 20.7.2 答案

1. 不是。你仍然必须具有一个指向页面的链接，并对内容应用特定的样式表，使得它看起来是打印友好的。
2. media 属性假定它的默认值是 all，这导致样式表把所有的媒体类型都作为目标。
3. 在 HTML 的<head>区域中使用以下代码：

```
<link rel="stylesheet" type="text/css" href="freestyle.css"
media="tv" />
```

## 20.7.3 练习

- 为具有相当多的颜色和图像的页面创建一个打印友好的样式表，确保向页面中添加一个额外的<link />标签，在特定于打印的样式表中链接它。
- 如果你感觉确实有些雄心勃勃，可以尝试使用<link />标签的 media 属性的 handheld 值，创建你的 Web 页面之一的特定于手持的版本。它的概念与创建打印友好的页面相同，只不过在这种情况下，处理的是极其受限的屏幕大小，而不是打印的页面。可以发布页面，然后在手机或手持式设备上打开它来进行测试。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 21 章

# 理解动态 Web 站点和 HTML5 应用程序

在本章中你将学到：

- 怎样概念化不同类型的动态内容；
- 怎样在 HTML 中包括 JavaScript 代码；
- 怎样利用 JavaScript 显示随机化的文本；
- 怎样使用 JavaScript 和用户事件更改图像；
- 怎样开始提前考虑把各个部分组合起来创建 HTML5 应用程序。

术语动态（dynamic）意味着某件事物是活动的，或者某件事物激发另一个人变成活动的。在讨论 Web 站点时，动态 Web 站点是指不仅在其功能和设计中纳入了交互性，而且还激发用户采取某个动作（阅读更多内容、购买产品等）的站点。在本章中，你将学习可以使站点成为动态站点的不同类型的交互性，包括关于服务器端和客户端脚本的信息（以及后面的一些实用的示例）。

我在本书中的别的位置已经提过客户端脚本，并且你在第 18 章中已经使用过一点，当时你使用事件属性和 JavaScript 更改特定元素的样式——称为操作 DOM（Document Object Model，文档数据模型）。在本章中，你将执行更多的这类操作。确切地讲，在学习了不同的技术之后，你将使用 JavaScript 在页面加载时显示随机引文，并且基于用户交互来交换图像。最后，在至少学习了一些关键字以及把 HTML、CSS 和 JavaScript 这些部分组合起来的基本概念之后，将向你介绍在创建 HTML5 应用程序时可能存在的一些情况。

## 21.1 理解不同类型的脚本

在 Web 开发中，存在两种不同类型的脚本：服务器端脚本和客户端脚本。这两类脚本（它们事实上都是一种计算机编程形式）超出了本书的范围，不过，它们并没有超出太多。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

*Sams Teach Yourself* 系列中的两本非常有用且流行的图书是本书的自然延伸，这两本图书是：*Sams Teach Yourself PHP, MySQL and Apache All-in-One*（针对服务器端脚本）和 *Sams Teach Yourself JavaScript in 24 Hours*（针对客户端脚本）

服务器端脚本（server-sidescripting）指运行在 Web 服务器上的脚本，它们用于把结果发送给 Web 浏览器。如果你曾经在 Web 站点上提交过表单（包括使用搜索引擎），就体验过服务器端脚本的结果。一些流行（并且相对容易学习）的服务器端脚本语言包括以下列表（要了解更多信息，可以访问列出的 Web 站点）。

- PHP（PHP: Hypertext Preprocessor）：http://www.php.net/。
- Ruby：http://www.ruby-lang.org/。
- Python：http://www.python.org/。
- Perl：http://www.perl.org/。

另一方面，客户端脚本（client-side scripting）指运行在 Web 浏览器内的脚本，即无需与 Web 服务器交互即可使脚本运行。迄今为止，最流行的客户端脚本语言是 JavaScript。多年来，研究表明，在所有 Web 浏览器中，有超过 98%的浏览器都支持 JavaScript。

**Did you Know?**

**提示：**尽管 JavaScript 与面向对象程序设计语言 Java 的名称相近，但是 JavaScript 并不是源于 Java，也不是它的什么近亲。Java 由 Sun Microsystems 于 1995 年发布，与服务器端脚本语言 JSP 密切相关。JavaScript 由 Netscape Communications 也于 1995 年创建，这样命名是为了指示它只在表面上与 Java 类似，但是二者之间并无直接联系。

另一种客户端脚本语言是 Microsoft 的 VBScript（Visual Basic Scripting Edition，Visual Basic 脚本版）。这种语言只能被 Microsoft Internet Explorer Web 浏览器使用，因此除非你非常确信用户将利用这种浏览器访问你的站点（比如在封闭的公司环境里），否则不应该使用它。鉴于我们希望覆盖尽可能广泛的受众，本章假定为客户端脚本使用 JavaScript，并且本章中的编码示例全都是 JavaScript。

## 21.2 在 HTML 中包括 JavaScript 代码

JavaScript 代码可以存在于文件内的以下两个位置之一。

- 位于它自己的具有.js 扩展名的文件内。
- 直接位于 HTML 文件内。

外部文件通常用于脚本库（可以在许多页面中重用的代码），而直接出现在 HTML 文件内的代码倾向于实现特定于那些单独页面的功能。无论 JavaScript 代码位于何处，浏览器都可以通过<script></script>标签对的使用而知晓它的存在。

当在外部文件中存储 JavaScript 代码时，可以用下面这种方式引用它：

```
<script type="text/javascript" src="/path/to/script.js"></script>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**By the Way**

**注意：**另外一种相当常见的情况是：可以在页面底部并且正好在<body>元素的结尾之前找到加载的脚本。这确保无需等待加载脚本即可呈现页面。不过，如果你还使用脚本把内容写到页面上，就不能使用这种加载脚本的方法——它将只在末尾写出内容，而这很可能不是你所想要的。

这些<script></script>标签通常位于<head></head>标签之间，因为严格来讲，它们并不是属于页面的<body>中的内容。作为替代，<script>标签使一组 JavaScript 函数或其他信息可供页面的余下部分使用。不过，也可以利用<script>标签封装 JavaScript 函数或代码段，并且根据需要把它们在页面中的任意位置。程序清单 21.1 显示了一个放在 HTML 文档的<body>中的 JavaScript 代码段的示例。

**程序清单 21.1 使用 JavaScript 打印一些文本**

```
<!DOCTYPE html>

<html 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>
```

**Did you Know?**

**提示：**在程序清单 21.1 中，你可能注意到下面这两行代码：

```
<!-- Hide the script from old browsers
// Stop hiding the script -->
```

这是 HTML 注释。位于以<!--开始并以-->结尾之间的任何内容在源代码中都是可见的，但是将不会被浏览器呈现。在这里，JavaScript 代码被 HTML 注释包围，以防访问者正在运行非常老的 Web 浏览器或者禁用了 JavaScript。现在，几乎所有的浏览器都会适当地使用或忽略 JavaScript，但是为根本不会处理 JavaScript 的非常老的浏览器或屏幕阅读器注释掉它们不会造成什么损害。

位于<script></script>标签之间的是单个 JavaScript 命令，它用于输出以下 HTML：

```
<p>This text comes from JavaScript.</p>
```

当浏览器呈现这个 HTML 页面时，它将会看到位于<script></script>标签之间的 JavaScript 代码，并暂停一毫秒以执行命令，然后返回以呈现输出结果，它现在包括来自 JavaScript 命令的 HTML 输出。图 21.1 显示这个页面像任何其他的 HTML 页面一样出现。它是一个 HTML 页面，但是只有一小部分 HTML 来自于 JavaScript 命令。

异步社区会员 Kensuke(18319066421) 专享 尊重版权




图 21.1

JavaScript 代码段的输出结果看起来像任何其他的输出结果

## 21.3 显示随机内容

每次页面加载时，都可以使用 JavaScript 显示不同的内容。或许你具有一个文本或图像集合，你发现它们非常有趣，值得包括在页面中。

我非常痴迷于好的引文。如果你像我或者其他许多创建个人 Web 站点的人一样，你可能发现在 Web 页面中纳入一条不断变化的引文是有趣的。要创建一个页面，它具有一条每次页面加载时都会改变的引文，必须先收集所有的引文，以及它们各自的来源。然后把这些引文放在一个 JavaScript 数组（array）中，它是程序设计语言中的一种特殊的存储单元，可以方便地处理项目列表。

在把引文加载到数组中之后，用于随机提取引文的 JavaScript 代码就相当简单了（我们稍后将解释它）。你已经见过了将输出结果打印到 HTML 页面上的代码段。

程序清单 21.2 包含完整的 HTML 和 JavaScript 代码，它们可以让 Web 页面在每次加载时显示一条随机的引文。

**程序清单 21.2　带有随机引文的 Web 页面**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Quotable Quotes</title>

    <script type="text/javascript">
      <!-- Hide the script from old browsers
      function getQuote() {
        // Create the arrays
        quotes = new Array(4);
        sources = new Array(4);

        // Initialize the arrays with quotes
        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";
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
        quotes[1] = "Everybody is ignorant. Only on different " +
        "subjects.";
        sources[1] = "Will Rogers";

        quotes[2] = "They say such nice things about people at " +
        "their funerals that it makes me sad that I'm going to " +
        "miss mine by just a few days.";
        sources[2] = "Garrison Keillor";

        quotes[3] = "What's another word for thesaurus?";
        sources[3] = "Steven Wright";

        // Get a random index into the arrays
        i = Math.floor(Math.random() * quotes.length);

        // Write out the quote as HTML
        document.write("<p style='background-color: #ffb6c1' >\"");
        document.write(quotes[i] + "\"");
        document.write("<em>- " + sources[i] + "</em>");
        document.write("</p>");
      }
      // 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>
```

尽管这段代码看起来比较长，但是其中许多内容都由可用于显示在页面上的 4 条引文组成。

第一组<script></script>标签之间的大量代码行用于创建一个名为 getQuote()的函数。在定义了一个函数之后，就可以在相同页面的其他位置调用它，稍后将在代码中看到。注意：如果函数存在于外部文件中，那么就可以从所有页面中调用它。

如果密切观察代码，将会看到如下一些行：

```
// Create the arrays
```

和

```
// Initialize the arrays with quotes
```

这些都是代码注释。开发人员使用这些类型的注释在代码中留下说明，使得任何阅读它的人都清楚代码在那个特定的位置正在做什么。在第一条关于创建数组的注释后面，可以看到创建了两个数组：一个名为 quotes，另一个名为 sources，它们都包含 4 个元素：

```
quotes = new Array(4);
sources = new Array(4);
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

在第二条注释（关于利用引文初始化数组）后面，向数组中添加了 4 个数据项。让我们密切观察其中一个数据项，即第一条引文，它出自 Mark Twain：

```
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";
```

你已经知道数组被命名为 quotes 和 sources，但是被赋值的变量（在这个实例中）则称为 quotes[0]和 sources[0]。因为 quotes 和 sources 是数组，数组中的数据项则都具有它们自己的位置。在使用数组时，数组中的第一个数据项不是在#1 槽中，它位于#0 槽中。换句话说，是从 0 而不是从 1 开始计数，这是程序设计中的典型情况，可以把它归档为一个用于将来的有趣和有用的说明（或者一个不错的普通答案）。因此，将把第一条引文的文本（值）赋予 quotes[0]（变量）。类似地，把第一个来源的文本赋予 source[0]。

文本字符串封闭在引号中。不过，在 JavaScript 中，换行符指示命令的结尾，因此下面的书写方式将在代码中引发问题：

```
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，你将看到字符串被构建为一系列封闭在引号中的字符串，并且利用加号（+）把它们连接起来（这个加号被称为连接运算符（concatenation operator））。

下一个代码块肯定看起来最像程序设计；这一行代码生成一个随机数，并把那个值赋予一个名为 i 的变量：

```
i = Math.floor(Math.random() * quotes.length);
```

但是，你不能挑选任何随机数。随机数的目的是确定应该打印哪条引文和来源，并且只有 4 条引文。因此，这一行 JavaScript 代码将做以下事情。

- 使用 Math.random()获取一个 0~1 的随机数。例如，0.5482749 可能就是 Math.random()的结果。
- 用随机数乘以 quotes 数组的长度，后者目前是 4。数组的长度是数组中的元素个数。如果随机数是 0.5482749（如前所示），把它乘以 4 将得到 2.1930996。
- 使用 Math.floor()把结果向下取整为最接近的整数。换句话说，把 2.1930996 变为 2。
- 给变量 i 赋值 2（例如）。

函数的余下部分看起来应该很熟悉，只有少数几个例外。首先，如你在本章前面学过的，document.write()用于编写浏览器将要呈现的 HTML。接下来，把字符串分隔开，以清楚指示何时需要以不同的方式处理什么事情，比如当应该以字面意义打印引号时利用反斜杠（\）对它们进行转义，或者在代入变量的值时。所打印的实际引文和来源将与 quotes[i]和 sources[i]匹配，其中 i 是由前述的数学函数确定的数字。

但是，简单地编写函数的举动并不意味着将会产生任何输出结果。在 HTML 中再往下看，可以看到两个<script></script>标签之间的 getQuote();——这是调用函数的方式。无论何时执行函数调用，都会把函数的输出结果放在那个位置。在这个示例中，将把输出结果显示

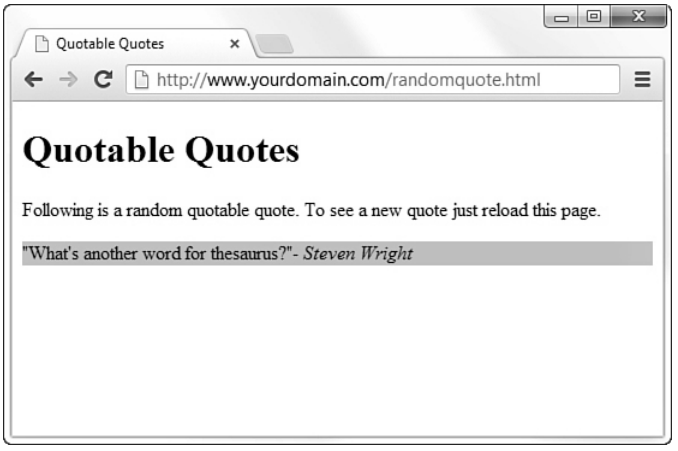
异步社区会员 Kensuke(18319066421) 专享 尊重版权

在介绍引文的段落下面。

图21.2显示了在加载到Web浏览器中时所显示的Quotable Quotes页面。当页面重新加载时，将有1/4的几率显示一条不同的引文——毕竟，它是随机的。

图21.2

每次加载Quotable Quotes页面时，它都会显示一条随机的引文




记住：可以轻松地修改这个页面，以包括进你自己的引文或者你想随机显示的其他文本。也可以通过在代码中在quotes和sources数组中添加更多的条目，增加可供显示的引文数量。当然，还可以修改HTML输出结果，并以你喜欢的任何方式编排它的样式。

如果使用Quotable Quotes页面作为起点，就可以轻松地改变脚本，并按自己的想法创建你自己的有趣的变体。如果在过程中犯了错误，也没有关系。在脚本代码中找出过去的错误的技巧是要有耐心，并且仔细分析所输入的代码。你总是可以删除代码来简化脚本，直到使之可以工作为止，然后一次添加一段新代码，以确保每个代码段都会工作。

## 21.4 理解文档对象模型

本节也可以被称为“开始使用文档对象模型以及在将来学习更多相关知识至少需要知道什么”，但是即便如此，这也将构成一个可怕的节标题。如你所学过的，使用JavaScript的客户端交互性通常采用以某种方式操作DOM（Document Object Model，文档对象模型）的形式。DOM是所有文档的不可见的结构，它不是HTML结构或者应用语义格式化的方式，而是一种总体框架或容器。如果这种描述似乎有些模糊，那是由于它本来就是这样，它不是一个切实的对象。

总的容器对象被称为document。你在文档内创建并提供一个ID的任何容器都可以通过该ID来引用。例如，如果具有一个ID为wrapper的 ，那么在DOM中，就可以用以下方式引用该元素：

```
document.wrapper
```

回忆可知，在第18章中，通过更改与特定元素关联的style对象中的某些内容，可以改变该元素的可见性。类似地，如果希望访问ID为wrapper的 的background-color样式（然后对它进行某种处理），将像下面这样引用它：

```
document.wrapper.style.background-color
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

要改变该样式的值，也许是基于交互式的用户事件来执行该操作，可以使用以下代码把颜色改为白色：

```
document.wrapper.style.background-color="#ffffff"
```

DOM 是在背后支持你以这种方式引用元素及其关联对象的框架。显然，这简短地概述了一些相当复杂的事情，但是至少你现在可以理解这种 document.xx 表示法是关于什么的。要了解关于 DOM 的更多信息，可以访问 http://www.w3.org/DOM/，获取万维网联盟（World Wide Web Consortium）提供的关于 DOM 的信息。

## 21.5 基于用户交互更改图像

第 18 章介绍了不同类型的用户交互事件，比如 onclick、onmouseover 和 onmouseout。在该章中，你基于用户交互触发了文本中的改变，在本节中，你将看到一个可见的交互类型的示例，它既是实用的，也是动态的。

图 21.3 显示了一个页面，它包含一幅大图像，旁边有一些文本，页面往下较远的位置还有 3 幅小图像。如果密切观察小图像的列表，可能会注意到第一幅小图像事实上是所显示的大图像的较小版本。这是小图库的常见显示类型，比如你可能在在线目录中看到的小图库，其中的项目具有一条描述以及少量的产品替代视图。尽管产品详情的特写图像对于潜在的购买者很重要，但是在页面上使用多幅大图像从显示和带宽的角度看将使页面显得臃肿，因此这种图库视图是显示替代图像的流行方式。我个人没有要销售的产品，但是我确实具有大树的图片，因此可以把它们用作示例。

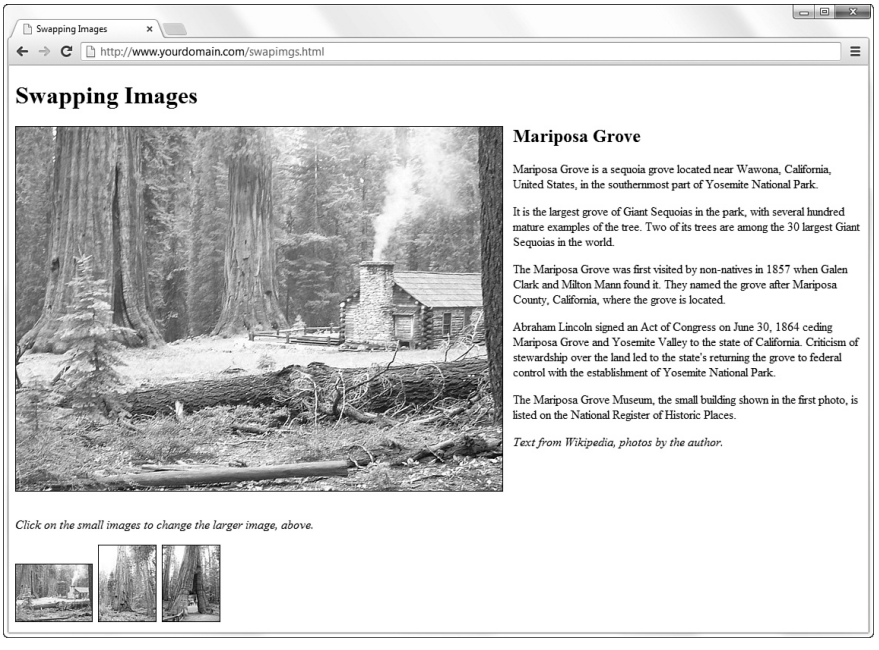



图 21.3 具有主图像和准备好被单击并查看的替代图像的信息页面

使用下面这个<img/>标签指定页面上的大图像：

```
<img
   id="large_photo"
   style="border: 1px solid black; margin-right: 13px;"
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
  src="mariposa_large_1.jpg"
  alt="large photo" />
```

在游戏的这个阶段，style、src 和 alt 属性应该对于你都是有意义的。此外，可以看到，给这幅图像提供的 ID 是 large_photo。因此，这幅图像在 DOM 中是作为 document.images['large_photo']存在的——通过它们的 ID 引用图像。这很重要，因为一点 JavaScript 功能就使我们能够动态地改变 document.images['large_image'].src 的值，它是图像的来源（src）。

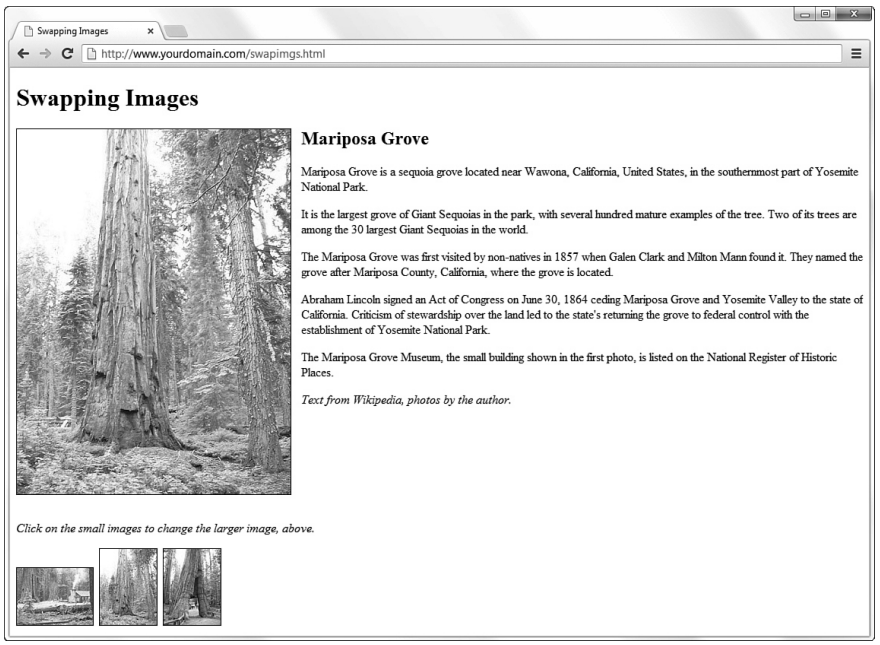
下面的代码段在图 21.3 底部所示的 3 幅图像的组中创建第三幅小图像。onclick 事件指示当用户单击这幅图像时，就利用匹配的大图像的路径填充 document.images['large_image'].src 的值（大图像的空位）。

```
<a href="#"
  onclick="javascript:document.images['large_photo'].src = 'mariposa_
  large_1.jpg'">
<img
  style="border: 1px solid black; margin-right: 3px;"
  src="mariposa_small_1.jpg"
  alt="photo #1" /></a>
```

图 21.4 显示了相同的页面，但是没有被用户重新加载。当用户单击页面底部的第三幅小图像时，大图像的空位将被一幅不同的图像填充。

图 21.4

当用户单击较小的图像时，将会替换大图像




## 21.6 提前考虑开发 HTML5 应用程序

我没有撒谎，利用 HTML 和 CSS 构建的基本 Web 站点与使用 HTML5 的一些高级特性的高级应用程序之间具有非常大的区别。但是，你对 Web 的语言 HTML 的理解很重要，这样就能够考虑可以在多大程度上扩展它（事实证明，可以对它进行非常大的扩展）。除了基本的标记之外，HTML5 还为复杂的应用程序扩展到包括了 API（application programming interface，应用程序编程接口），如你在以前章节中所学到的，从音频和视频元素的自然集成

异步社区会员 Kensuke(18319066421) 专享 尊重版权

开始，一直到允许访问和运行成熟的应用程序的内置脱机存储机制（和存储在客户端的数据），即使没有网络连接亦可如此。

尽管 HTML5 的功能极其丰富，高度交互式的 HTML5 Web 站点和应用程序（包括移动应用程序）的创建也不是孤立地发生的。当配合使用 HTML5 与客户端语言（比如 JavaScript）时，交互性就应运而生了，客户端语言然后将返回到服务器中，并通过称为 Web 套接字（Web socket）的持久连接与服务器端语言（比如 PHP、Ruby、Python 等）通信。通过打开这条连接，并与一些（例如）操作数据库或执行某种计算的服务器端代码通信，浏览器可以转发将被 JavaScript 额外处理并且最终以 HTML5 呈现的大量信息。在涉及应用程序创建时，无论它是视频游戏、字处理程序，还是电子邮件或 Twitter 客户等流行的 HTML5 应用程序，HTML5 的高级特性与 JavaScript（确切地讲，是功能丰富的 JavaScript 库，比如 jQuery）的结合确实使得机会无限。

HTML5 应用程序创建中涉及的技术深度超出了这本基本图书的范围，但是你应该具有符合标准的 HTML5 和 CSS3 的基础，如果你开始考虑基本 Web 站点以外的事情，它们将非常有用。要学习关于 HTML5 应用程序创建的更多知识，可以阅读《HTML 移动应用开发入门经典》一书，其中介绍了一些核心特性。通常学习本书的全部内容，你将牢固地掌握 HTML5 的基础知识。我自信：借助额外的指导，你就可以更进一步，开始学习并且在 HTML5 应用程序中构建基本的交互。

## 21.7 小结

本章介绍了服务器端脚本与客户端脚本之间的区别，并且介绍了如何在 HTML 文件中包括 JavaScript，以给 Web 站点添加一点交互性。本章还介绍了如何使用 JavaScript document.write()方法在页面加载时显示随机引文。最后，本章讲解了一点关于文档对象模型的知识。

通过应用之前章节中学到的知识，读者将掌握如何使用客户端脚本使 Web 页面上的图像响应鼠标移动事件。所有这些任务都不需要太多的编程技能，但是它们可以激励你学习关于 JavaScript 或者服务器端程序设计语言的更多知识，以便可以给页面提供更复杂的交互特性，包括进入创建 HTML5 应用程序的世界。

## 21.8 问与答

**问：如果我想使用本章中的随机引文脚本，但是我希望具有一个包含许多条引文的库，我将不得不把所有的引文放在每个页面中吗？**

**答：**是的。数组中的每一项都必须出现在页面中。在这里，你可以开始看到有些事情可以在客户端处理，而另外一些事情在服务器端处理会更好一些。如果你具有一个真正的随机引文库，并且在任何给定的时间只展示一条引文，那么最好的做法可能是把那些项目存储在一个数据库表中，并且使用一点服务器端脚本连接到该数据库，检索文本，并把它打印到页面上。此外，总是可以继续利用 JavaScript 携带所有的引文，但是至少应该把那个 JavaScript 函数放在一个不同的文件中，以便可以独立于文本单独进行维护。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**问：我见过了一些在线目录，在 Web 站点内容的顶部一层中显示一幅大图像，并且在它底下可以看到普通的 Web 站点内容，但是焦点是在大图像上。怎样实现这种效果？**

**答：**图像的描述听起来像是由称为 Lightbox 的 JavaScript 库创建的效果。Lightbox 库使你能够在站点内容上面的一层中显示显示一幅图像，或者一个图库。这是一个非常流行的库，用于显示大图像的详细信息，或者只显示一组被认为很重要从而足以展示在内容“上面”的图像。这个库的创建者在 http://lokeshdhakar.com/projects/lightbox/上免费提供了它。要安装并使用它，可以遵循与软件一起包括的指导；你将能够使用在本书中迄今为止所获得的知识把它集成进你的站点中。

## 21.9 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 21.9.1 问题

1. 你制作了一幅按钮的图片，并把它命名为 button.gif。你还制作了该按钮的简单的 GIF 动画，使之闪现绿色和白色。你把该 GIF 命名为 flashing.gif。你可以使用什么 HTML 和 JavaScript 代码，使得每当用户把鼠标指针移到按钮上时它都会闪光，并且当用户单击按钮时它还会链接到一个名为 gohere.html 的页面？
2. 你可以怎样修改为问题 1 编写的代码，使得当用户把鼠标指针移到按钮上时它将会闪光，并且会继续不停地闪光，即使用户后来把鼠标指针从它上面移开也是如此？
3. 在下面的代码中，加号意味着什么？

```
document.write('This is a text string ' + 'that I have created.');
```

### 21.9.2 答案

1. 你的代码看起来可能如下所示：

```
<a href="gohere.html"
onmouseover="javascript:document.images['flasher'].src=
'flashing.gif'"
onmouseout="javascript:document.images['flasher'].src='button.gif'">
<img src="button.gif" id="flasher" style="border-style:none" /></a>
```

2. 你的代码看起来可能如下所示：

```
<a href="gohere.html"
onmouseover="javascript:document.images['flasher'].src=
'flashing.gif'">
<img src="button.gif" id="flasher" style="border-style:none" /></a>
```

3. 加号（+）把两个字符串连接在一起。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

### 21.9.3 练习

- 你是否具有任何页面，如果当鼠标指针经过导航图标或其他图像时它们会改变，那么页面看起来将更华丽、更容易理解？如果是这样，可以尝试创建图像的一些高亮版本，并且尝试使用本章中介绍的信息修改你自己的页面。
- 你可以利用在本章前面学过的使用 JavaScript 显示随机内容的相同方式来显示随机图像，比如图形横幅或广告。不用打印文本，只需打印用于你想显示的图像的<img/>标签。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 22 章

# 处理基于 Web 的表单

在本章中你将学到：

- HTML 表单怎样工作；
- 怎样创建 HTML 表单的前端；
- 怎样命名各份表单数据；
- 怎样在表单中包括隐藏的数据；
- 怎样为相应的情况选择正确的表单输入控件；
- 怎样提交表单数据；
- 怎样验证表单数据；
- 通过电子邮件发送表单结果。

至此，本书中的大部分内容重点关注的是把信息传达给其他人。但是，你也可以使用 Web 页面，从阅读它们的人那里收集信息。

Web 表单使你能够从访问 Web 页面的用户那里接收反馈、订单或其他信息。如果你曾经使用过搜索引擎，比如 Google、Yahoo!或 Bing，就应该熟悉 HTML 表单，即那些单个输入框的表单，并且具有一个按钮，当按下它时，将提供你正在寻找的所有信息。产品订购表单也是表单的一种极其流行的应用，如果从 Amazon.com 订购任何商品或者从 eBay 卖家那里购物，你就已经使用了表单。在本章中，你将学习如何创建自己的表单，但是只将学习如何创建那些表单的前端。处理表单的后端需要具备程序设计语言的知识，而这超出了本书的范围。

## 22.1 HTML 表单是如何工作的

HTML 表单是 Web 页面的一部分，它们包括一些区域，用户可以在其中输入信息，然

异步社区会员 Kensuke(18319066421) 专享 尊重版权

后把它发回给你，或者发送到你指定的另一个电子邮件地址、你管理的数据库，或者完全发送到另一个系统，比如用于表单的第三方管理系统，如 Salesforce.com。

在学习用于制作自己的表单的 HTML 标签之前，至少应该从概念上理解如何将来自那些表单的信息发回给你。实际的幕后（服务器端[server-side]或后端[back-end]）过程需要了解至少一门程序设计语言，或者在使用别人的服务器端脚本处理表单输入时至少能够遵循特定的指导。在过程中的那个时间，你要么应该与某个具有技术知识的人协同工作，要么自己学习一些基础知识。简单的表单处理并不困难，并且你的 Web 托管提供商很可能具有多个后端脚本，只需进行最少量的自定义即可使用它们。

**By the Way**

**注意：**PHP 是最流行的服务器端程序设计语言；任何称职的 Web 托管提供商都支持它。在 http://www.php.net/上可以学习关于 PHP 的更多知识，或者可以在 *Sams Teach Yourself PHP, Apache, and MySQL All-in-One* 一书中从头开始深入学习这门程序设计语言（以及数据库交互性）。尽管有多本其他的关于 PHP 的图书及相关技术可用，我仍然偏爱这一本书，因为是我编写它的。它适合于 PHP 或其他任何程序设计语言的绝对初学者。

表单包括一个按钮，可以让用户提交表单，这个按钮可以是你自己创建的一幅图像，或者是一个标准的 HTML 表单按钮，它是在创建表单的<input>标签时创建的，并给 type 赋值为 submit。当某个人单击表单提交按钮时，将把在表单中输入的所有信息都发送到在<form>标签的 action 属性中指定的 URL。该 URL 应该指向一个将处理表单的特定脚本，然后通过电子邮件发送表单内容，或者执行交互式过程中的另一个步骤（比如从搜索引擎请求结果，或者把物品放入在线购物车中）。

当你开始考虑对表单内容做更多事情而不仅仅是通过电子邮件把结果发送给你自己时，将需要额外的技术知识。例如，如果你希望创建一家网上商店，接受信用卡并处理交易，那么有一些行之有效的做法用于处理这个任务，它们都适合于确保客户数据的安全性。这项操作执行起来不是那么轻松自如，你将需要比本书提供的更多知识。

**By the Way**

**注意：**有一种方法无需服务器端脚本即可发送表单数据，你将在本章后面学习该方法——它使用<form>的 action 属性中的 mailto 链接。但是，在你尝试该方法时，要知道它可能会产生不一致的结果；各个 Web 浏览器以及个人安全设置可能会导致动作以不同于你想要的方式做出响应。

在把表单发布到网上之前，应该仔细阅读 Web 托管提供商的用户指南，查看它在表单处理脚本方面提供了什么。你很可能发现只需最少量的配置即可轻松使用 Perl 或 PHP 脚本。

## 22.2 创建表单

每个表单都必须开始于一个<form>标签，它可以位于 HTML 文档主体内的任何位置。<form>标签通常具有 3 个属性：name、method 和 action。

```
<form name="my_form" method="post" action="myprocessingscript.php">
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

最常见的 method 是 post，它把表单输入结果作为文档进行发送。在一些情况下，需要使用 method="get"，它代之以把结果作为 URL 查询字符串的一部分进行提交。例如，当从 Web 表单把查询提交给搜索引擎时，有时就会使用 get。由于你还不是一名表单方面的专家，可以只使用 post，除非你的 Web 托管提供商的文档告诉你采用另外的方法。

**By the Way**

**注意**：HTML5 已经经历了许多次改进和添加功能，用于创建表单和表单元素，但是在编写本书时，它们并非都能够以你期望的方式使用。好消息是即使在表单中使用一种较老的浏览器在技术上不支持的输入类型，它仍将显示一个有用的普通输入框。总之，本章只将讨论你可以自信地使用的输入类型。要了解关于对与表单相关的新元素和属性的浏览器支持的最新信息，可以在 http://caniuse.com/#feat=forms 上检查“Can I Use…”，它汇集了关于 HTML5 和 CSS3 的浏览器支持的各类信息。

action 属性指定用于发送表单数据的地址。这里有两个选项。

- 可以输入 Web 服务器上的表单处理程序或脚本的位置，然后将把表单数据发送给那个程序。这是迄今为止最常见的方案。
- 可以输入“mailto:”，其后接着你的电子邮件地址，无论何时某个人填写了表单，这将把表单数据直接发送给你。不过，这种方法完全依赖于用户的计算机正确地配置了电子邮件客户。通过未安装电子邮件客户的公共计算机访问你的站点的用户将被冷落：

```
<form name="my_form" method="post" action="mailto:me@mysite.com">
```

在程序清单 22.1 中创建如图 22.1 所示的表单，包括你目前可以在现代浏览器中的 HTML 表单中使用的几乎每一种用户输入组件。在你阅读下面关于每一类输入元素的解释时，可以参考该图和程序清单。

**程序清单 22.1 具有多种用户输入组件的表单**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Guest Book</title>

    <style type="text/css">

      fieldset {
         width: 75%;
         border: 2px solid #ff0000;
      }

      legend {
         font-weight: bold;
         font-size: 125%;
      }

      label.question {
         width: 225px;
         float: left;
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
      text-align: left;
      font-weight: bold;
    }

    span.question {
      font-weight: bold;
    }

    input, textarea, select {
      border: 1px solid #000;
      padding: 3px;
    }

    #buttons {
      margin-top: 12px;
    }

  </style>
</head>
<body>
  <h1>My Guest Book</h1>
  <form name="gbForm" method="post" action="URL_to_script">

  <fieldset>
    <legend>Personal Information</legend>

    <p><label class="question" for="the_name">What is your
    name?</label>
    <input type="text" id="the_name" name="the_name"
           placeholder="Enter your full name."
           size="50" required autofocus /></p>

    <p><label class="question" for="the_email">What is your e-mail
           address?</label>
    <input type="email" id="the_email" name="the_email"
           placeholder="Please use a real one!"
           size="50" required /></p>
  </fieldset>

  <fieldset>
    <legend>Survey Questions</legend>

    <p><span class="question">Please check all that apply:</span><br/>
    <input type="checkbox" id="like_it" name="some_statements[]"
           value="I really like your Web site." />
    <label for="like_it">I really like your Web site.</label><br/>
    <input type="checkbox" id="the_best" name="some_statements[]"
           value="It's one of the best sites I've ever seen" />
    <label for="the_best">It’s one of the best sites I’ve ever
           seen.</label><br/>
    <input type="checkbox" id="jealous" name="some_statements[]"
           value="I sure wish my site looked as good as yours." />
    <label for="jealous">I sure wish my site looked as good as
           yours.</label><br/>
    <input type="checkbox" id="no_taste" name="some_statements[]"
           value="I have no taste and I'm pretty dense, so your site
           didn't do much for me." />
    <label for="no_taste">I have no taste and I’m pretty dense, so your
           site didn’t do much for me.</label></p>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
        <p><label for="choose_scale"><span class="question">Please rate my
              site on a scale of 1 (poor) to 10 (awesome):</span></label>
        <input type="number" id="choose_scale" name="choose_scale"
               min="0" max="10" step="1" value="5"/></p>

        <p><span class="question">Please choose the one thing you love best
               about my web site:</span><br/>
        <input type="radio" id="the_picture" name="best_thing"
               value="me" />
        <label for="the_picture">That amazing picture of you</label><br/>
        <input type="radio" id="the_cats" name="best_thing"
               value="cats" />
        <label for="the_cats">All the cat photos, of course</label><br/>
        <input type="radio" id="the_story" name="best_thing"
               value="childhood story" />
        <label for="the_story">The inspiring recap of your suburban
               childhood</label><br/>
        <input type="radio" id="the_treasures" name="best_thing"
               value="Elvis treasures" />
        <label for="the_treasures">The detailed list of all your Elvis
               memorabilia</label></p>

        <p><label for="how_improve"><span class="question">How can I
               improve my web site?</span></label><br/>
         <select id="how_improve" name="how_improve" size="4" multiple>
               <option value="You can't. It couldn't be better.">You can't.
               It 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></p>

      </fieldset>

      <fieldset>
          <legend>Free for All!</legend>
          <p><label for="message"><span class="question">Feel free to send
          more praise, gift offers, etc.:</span></label>
          <textarea id="message" name="message" rows="7" cols="55">
          </textarea></p>
      </fieldset>
      <div id="buttons">
        <input type="submit" value="Click Here to Submit" /> or
        <input type="reset" value="Erase and Start Over" />
      </div>

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

程序清单 22.1 中的代码使用了一个<form>标签，它包含相当多的<input />标签。每个<input />标签对应于一种特定的用户输入组件，比如复选框或单选按钮。输入、选择和文本区元素在样式表中包含边框，因此很容易在表单中看到这些元素的外框。记住：可以对这些元素应用各类 CSS。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

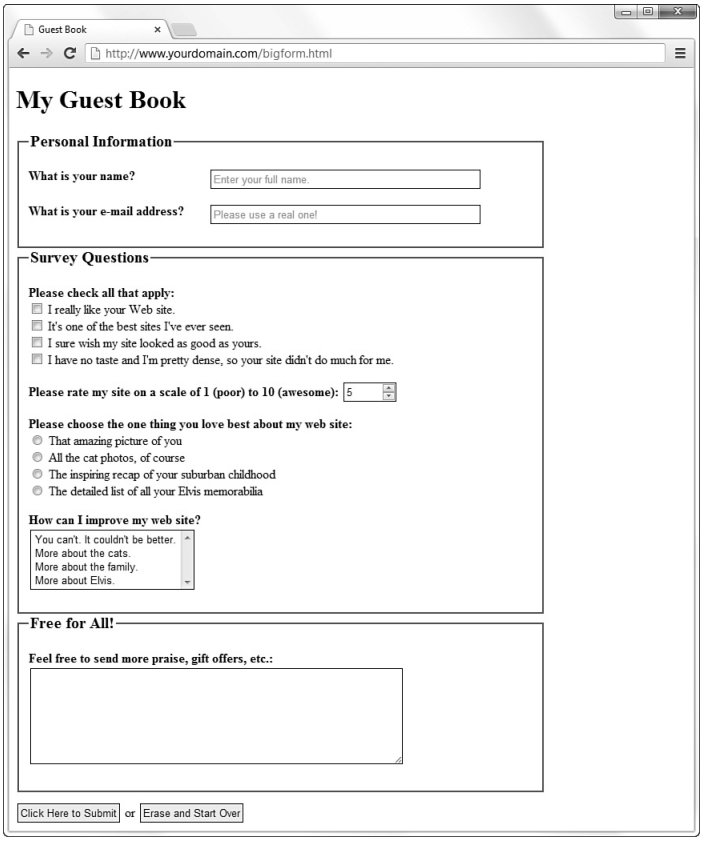



图 22.1

程序清单 22.1 中的代码使用了许多常见的 HTML 表单元素

下面几节将详细介绍<input />标签以及其他与表单相关的标签。

## 22.3 接受文本输入

为了要求用户在表单内提供特定的信息，可以使用<input />标签。尽管该标签并不明确需要出现在<form>与</form>标签之间，但是这样做是一个良好的实践，并且可以使代码更容易理解。可以把<input />标签放在页面上与文本、图像以及其他 HTML 标签相关的任意位置。例如，为了要求提供某个人的名字，可以输入以下文本，其后紧接着一个<input />字段：

```
<p><label class="question" for="the_name">What is your name?</label>
<input type="text" id="the_name" name="the_name"
       placeholder="Enter your full name."
       size="50" required autofocus /></p>
```

type 属性指示要显示的是什么类型的表单元素，在这里是一个简单的单行文本输入框（在本章中将单独讨论每种元素类型）。在这个示例中，注意 placeholder、required 和 autofocus 属性的使用。你将在本章后面学习 required 属性，一旦浏览器呈现表单，autofocus 属性就会自动把用户的光标聚焦在这个文本框中。一个表单只能有一个 autofocus 字段。placeholder 属性使你能够定义一些出现在文本框中的文本，但是当你开始输入时，它们将会消失。使用这个属性，可以在用户完成你的表单时给他们提供更多一点的指导。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**Did you Know?**

**提示：**如果你希望用户在输入文本时它们不会显示在屏幕上，可以使用<input type="password" />代替<input type="text" />。这样，将会显示星号（***），代替用户输入的文本。无论是 type="password"还是 type="text"，size、maxlength 和 name 属性的工作方式都完全相同。记住：这种隐藏密码的技术只提供了视觉保护，而没有加密或其他保护措施与将要传送的密码相关联。

size 属性指示文本输入框的宽度应该能够容纳大约多少个字符。如果使用一种比例间隔字体，那么输入的宽度将因用户输入的内容而异。如果输入太长而在框中放不下，大多数 Web 浏览器将自动把文本滚动到左边。

maxlength 属性确定允许用户在文本框中输入的字符数。如果用户尝试输入超过指定长度的内容，那么额外的字符将不会出现。你可以指定一个更长、更短或者与文本框的物理大小相同的长度。size 和 maxlength 属性只用于那些打算用于文本值的输入字段，比如 type="text"、type="email"、type="URL"和 type="tel"，而不适用于复选框和单选按钮，因为它们具有固定的大小。

## 22.4 命名每一份表单数据

无论输入元素的类型是什么，都必须给它收集的数据提供一个名称。可以为每个输入项目使用你所喜欢的任何名称，只要表单中的每个名称都不相同即可（单选按钮和复选框除外，在本章后面将讨论相关内容）。当表单被后端脚本处理时，将按名称确定每个数据项。这个名称变成了一个变量，它填充有一个值。这个值要么是用户在表单中输入的，要么是与用户选择的元素关联的值。

例如，如果用户在前面定义的文本框中输入“Jane Doe”，就会把一个变量发送给表单处理脚本，变量是 user_name，变量的值则是“Jane Doe”。表单处理脚本将处理这些类型的变量名称和值。

**By the Way**

**注意：**为了在本书的范围内进行解释说明，有意把这里的表单处理脚本设计得过于简单。可供处理脚本使用的变量的准确外观（或名称）依赖于那个脚本的程序设计语言。但是，从概念上讲，输入元素的名称变成了变量的名称，并且输入元素的值在后端变成了那个变量的值，这种说法是有效的。

要在 JavaScript 中使用这个文本框（或其他字段），记住文本对象使用 name 属性，像下面这样引用前面的代码段中的字段的值：

```
document.formname.user_name.value
```

### 22.4.1 标记每一份表单数据

标记表单数据与使用 name 或 id 属性标识表单元素以便往后使用并不是一回事。它代之以使用<label></label>标签对包围充当表单元素的一种文字说明的文本。表单元素<label>为元素提供了额外的环境，它对于屏幕阅读器软件特别重要。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

在程序清单 22.1 中可以看到两个不同的示例。首先，你可以看到包围用户询问的第一个问题（What is your name?）的<label>。使用 for 属性把这个标签绑定到具有相同 id（在这里是 the_name）的<input />元素：

```
<p><label class="question" for="the_name">What is your name?</label>
<input type="text" id="the_name" name="the_name"
       placeholder="Enter your full name."
       size="50" required autofocus /></p>
```

屏幕阅读器将读给用户：“What is your name?”然后还会读出“文本框”，提醒用户利用合适的信息完成文本框。在程序清单 22.1 中的另一个示例中，你看到使用<label>包围复选框列表中的不同选项（在该程序清单后面，还会看到使用它包围单选按钮列表）：

```
<p><span class="question">Please check all that apply:</span><br/>
<input type="checkbox" id="like_it" name="some_statements[]"
       value="I really like your Web site." />
<label for="like_it">I really like your Web site.</label><br/>
<input type="checkbox" id="the_best" name="some_statements[]"
       value="It's one of the best sites I've ever seen" />
<label for="the_best">It's one of the best sites I've ever
       seen.</label><br/>
<input type="checkbox" id="jealous" name="some_statements[]"
       value="I sure wish my site looked as good as yours." />
<label for="jealous">I sure wish my site looked as good as
       yours.</label><br/>
<input type="checkbox" id="no_taste" name="some_statements[]"
       value="I have no taste and I'm pretty dense, so your site
       didn't do much for me." />
<label for="no_taste">I have no taste and I'm pretty dense, so your
       site didn't do much for me.</label></p>
```

在这种情况下，屏幕阅读器将读出被<label>标签包围的文本，其后接着“复选框”，以提醒用户选择给定的选项之一。标签应该用于所有的表单元素，并且可以利用与其他容器元素相同的方式使用 CSS 编排样式。样式不会影响屏幕阅读器，但它确实有助于改进布局的美感和易读性。

## 22.4.2 组合表单元素

在程序清单 22.1 中，可以看到使用了<fieldset>和<legend>元素 3 次，用于创建 3 组不同的表单字段。<fieldset>元素正是用于执行该任务，它把表单元素组包围起来，给用户提供额外的环境，无论他们是直接在 Web 浏览器中还是在屏幕阅读器软件的帮助下访问它。<fieldset>元素就定义了这种组合，<legend>元素包含将显示或大声读出以描述这种组合的文本，比如程序清单 22.1 中的如下代码：

```
<fieldset>
    <legend>Personal Information</legend>
    <p><label class="question" for="the_name">What is your name?</label>
    <input type="text" id="the_name" name="the_name"
        placeholder="Enter your full name."
        size="50" required autofocus /></p>
...
</fieldset>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

在这种情况下，当屏幕阅读器读取与某个表单元素关联的<label>时，如你在上一节中所学过的，它还会追加<legend>文本，比如“Personal Information. What is yourname? Text box”。<fieldset>和<legend>元素可以使用 CSS 编排样式，因此在 Web 浏览器中可以轻松地把组合元素的视觉提示变成可见的（如你以前在图 22.1 中所看到的）。

### 22.4.3 在表单中包括隐藏的数据

你是否曾经希望把某些数据项发送给处理表单的服务器脚本，但是不希望用户看到那些数据项？此时，可以使用具有 type="hidden"属性的<input />标签。这个属性不会对显示产生任何影响，它只会在提交表单时把你指定的任何名称和值添加到表单结果中。

如果你使用的是 Web 托管提供商提供的表单处理脚本，就可能会指示你使用这个属性，告诉脚本通过电子邮件把表单结果发送到哪里。例如，包括进以下代码，在提交表单后，将通过电子邮件把结果发送到 me@mysite.com：

```
<input type="hidden" name="mailto" value="me@mysite.com" />
```

在你接收表单提交的结果时，有时会看到脚本使用隐藏的输入元素携带可能有用的额外数据，隐藏的表单字段的一些示例包括电子邮件地址和电子邮件的主题。如果你使用的是 Web 托管提供商提供的脚本，可以参考与该脚本一起提供的文档，了解关于潜在需要的隐藏字段的额外详情。

## 22.5 探讨表单输入控件

有多种输入控件可用于从用户那里获取信息。你已经见过了一个文本输入选项，下面几节将介绍其余大多数可用于设计表单的表单输入选项。

### 22.5.1 复选框

除了文本框之外，最简单的输入类型之一就是复选框（check box），它显示了一个小正方形。用户可以单击复选框，以选择或取消选择组中的一个或多个项目。例如，程序清单 22.1 中列出的复选框出现在文本“Please check all that apply”后面，暗示用户的确可以选中所有适用的选项。

用于程序清单 22.1 中的复选框的 HTML 代码显示 name 属性的值对于所有的复选框都是相同的：

```
<p><span class="question">Please check all that apply:</span><br/>
<input type="checkbox" id="like_it" name="some_statements[]"
       value="I really like your Web site." />
<label for="like_it">I really like your Web site.</label><br/>
<input type="checkbox" id="the_best" name="some_statements[]"
       value="It's one of the best sites I've ever seen" />
<label for="the_best">It's one of the best sites I've ever
       seen.</label><br/>
<input type="checkbox" id="jealous" name="some_statements[]"
       value="I sure wish my site looked as good as yours." />
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
<label for="jealous">I sure wish my site looked as good as
       yours.</label><br/>
<input type="checkbox" id="no_taste" name="some_statements[]"
       value="I have no taste and I'm pretty dense, so your site
       didn't do much for me." />
<label for="no_taste">I have no taste and I'm pretty dense, so your
       site didn't do much for me.</label></p>
```

name 属性中的方括号（[ ]）的使用指示后端脚本将把一系列值放入这一个变量中，而不是只使用一个值（如果用户只选择一个复选框，那么它可能就只是一个值）。如果用户选择第一个复选框，则将把文本字符串“I really like your Web site.”放入 website_response[]桶中。如果用户选择第三个复选框，则将把文本字符串“I sure wish my site looked as good as yours.”也放入 website_response[]桶中。处理脚本然后将把变量作为一个数据的数组（而不仅仅是单个条目）进行处理。

**提示：**如果发现输入元素的标签出现的位置离元素太近，就要在<input />标签的末尾和标签文本的开头之间添加空间，如下所示：

```
<input type="checkbox" name="mini" /><label>Mini Piano Stool</label>
```

**Did you Know?**

不过，你可能看到复选框组为组中的变量使用单独的名称。例如，下面显示了编写复选框组的另一种方式：

```
<p><span class="question">Please check all that apply:</span><br/>
<input type="checkbox" id="like_it" name="liked_site" value="yes"
       value="I really like your Web site." />
<label for="like_it">I really like your Web site.</label><br/>
<input type="checkbox" id="the_best" name="best_site" value="yes"
       value="It's one of the best sites I've ever seen" />
<label for="the_best">It's one of the best sites I've ever
       seen.</label><br/>
<input type="checkbox" id="jealous" name="my_site_sucks" value="yes"
       value="I sure wish my site looked as good as yours." />
<label for="jealous">I sure wish my site looked as good as
       yours.</label><br/>
<input type="checkbox" id="no_taste" name="am_dense" value="yes"
       value="I have no taste and I'm pretty dense, so your site
       didn't do much for me." />
<label for="no_taste">I have no taste and I'm pretty dense, so your
       site didn't do much for me.</label></p>
```

在这第二个复选框列表中，当被后端处理脚本处理时，第一个复选框的变量名称是"liked_site"，值（如果选中的话）是"yes"。

如果希望在 Web 浏览器呈现表单时复选框默认是选中的，就要包括 checked 属性。例如，下面的代码将创建两个复选框，并且第一个复选框默认是选中的：

```
<input type="checkbox" id="like_it" name="liked_site" value="yes"
       value="I really like your Web site." checked />
<label for="like_it">I really like your Web site.</label><br/>
<input type="checkbox" id="the_best" name="best_site" value="yes"
       value="It's one of the best sites I've ever seen" />
```

在这个示例中，标记为“I really like your site.”的复选框默认是选中的。用户必须单击该复选框才能取消选中它，从而指示他们对你的站点具有另外一种观点。标记为“One of the best I've seen.”的复选框开始时未选中，因此用户必须单击才能启用它。未选择的复选

异步社区会员 Kensuke(18319066421) 专享 尊重版权

框不会出现在表单输出中。

### 22.5.2 单选按钮

单选按钮（radio button）指一次只能选择其中一个选项，其实现方式与复选框几乎同样简单。单选按钮的最简单的应用是用于“是/否”问题，或者当只能选择一个候选人时用于投票表决。

要创建单选按钮，可以使用 type="radio"，并给每个选项提供它自己的<input />标签。为组中的所有单选按钮使用相同的 name，但是不要使用在创建复选框时使用的“[ ]”，因为不必容纳多个答案：

```
<input type="radio" id="vote_yes" name="vote" value="yes" checked />
<label for="vote_yes">Yes</label><br />
<input type="radio" id="vote_no" name="vote" value="no" />
<label for="vote_no">No</label>
```

value 可以是你所选的任何名称或代码。如果包括 checked 属性，默认就会选择该按钮。不能选中多个具有相同 name 的单选按钮。

**By the Way**

**注意：**单选按钮得名于它们与老式的按键式收音机上的按钮比较相似。那些按钮使用机械排列，使得当按下其中一个按钮时，其他的按钮都将会弹起。

在设计表单并在复选框与单选按钮之间做出选择时，要问问自己所问的或暗示的问题是否只能用一种方式回答。如果是，就使用单选按钮。

### 22.5.3 选择列表

滚动列表（scrolling list）和下拉选择列表（pull-down pick list）都是利用<select>标签创建的。可以结合使用这个标签与<option>标签，如下面的示例所示（取自程序清单 22.1）：

```
<p><label for="how_improve"><span class="question">How can I
     improve my web site?</span></label><br/>
<select id="how_improve" name="how_improve" size="4" multiple>
     <option value="You can't. It couldn't be better.">You can't.
          It 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></p>
```

与你在上一节中简单学过的 text 输入类型不同，这里的 size 属性用于确定在选择列表上同时显示多少个列表项。如果在上面的代码中使用 size="2"，则将只有前两个选项是可见的，并且在列表旁边会出现滚动条，使得用户可以向下滚动，以便查看第三个和第四个选项。

包括 multiple 属性将使用户能够一次选择多个选项；selected 属性用于使选项默认最初是选择的。在提交表单时，在用于每个选项的 value 属性中指定的文本将伴随所选的选项。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**提示：**如果省略 size 属性或者指定 size="1"，那么列表将创建一个简单的下拉选择列表。选择列表不允许进行多项选择，它们逻辑上等价于单选按钮组。下面的示例显示了为一个问题选择 yes 或 no 的另一种方式：

Did you Know?

```
<select name="vote">
  <option value="yes">Yes
  </option>
  <option value="no">No
  </option>
</select>
```

除了<option>和</option>之外，应该没有其他的 HTML 标签出现在<select>和</select>标签之间，只有<optgroup>标签例外（程序清单 22.1 中没有显示它）。如下面的代码段中所示，使用<optgroup>标签允许利用列表中显示的一个标签创建选项组（这就是 optgroup 得名的原因），但是不能选作表单字段的“答案”。例如，下面这个代码段：

```
<select name="grades">
     <optgroup label="Good Grades">
          <option value="A">A</option>
          <option value="B">B</option>
     </optgroup>
     <optgroup label="Average Grades">
          <option value="C">C</option>
     </optgroup>
     <optgroup label="Bad Grades">
          <option value="D">D</option>
          <option value="F">F</option>
     </optgroup>
</select>
```

产生一个如下所示的下拉列表：

```
Good Grades
   A
   B
Average Grades
   C
Bad Grades
   D
   F
```

在这种情况下，只有 A、B、C、D 和 F 是可选择的，但是<optgroup>标签是可见的。

## 22.5.4 文本框、文本区和其他输入类型

本章前面提到的<input type="text">属性只允许用户输入单行文本。当你希望允许在单个输入项中出现多行文本时，可以使用<textarea>和</textarea>标签创建一个文本区，而不仅仅是一个文本框。包括在这两个标签之间的任何文本都将显示为那个框中的默认条目。下面显示了一个来自程序清单 22.1 中的示例：

```
<textarea id="message" name="message" rows="7" cols="55">Your
      message here.</textarea>
```

如你可能猜到的，rows 和 cols 属性控制放入输入框中的文本的行数和列数。cols 属性

异步社区会员 Kensuke(18319066421) 专享 尊重版权

的准确度略低于 rows 属性，大约是放入一行文本中的字符数。不过，文本区确实具有滚动条，因此用户输入的文本可以多于显示区中放入的文本。

不过，让我们暂且回到基本的<input />元素上来，因为 HTML5 为输入提供了多得多的 type 选项，而不仅仅是“文本”，比如内置的日期选择器。其缺点是：并非所有的浏览器都完全支持其中的许多选项（比如内置的日期选择器）。下面列出了几种不同的输入类型（其中一些是新增的，另外一些则不然），它们都受到完全支持，但是在本章中将不会详细讨论它们。

- type="email"：显示为常规的文本框，但是当使用表单验证时，内置的验证器将会检查它是否是形式有效的电子邮件地址。一些移动设备默认会显示相关的按键（例如，@符号），而不需要额外的用户交互。
- type="file"：打开一个对话框，使你能够在计算机上搜索一个要上传的文件。
- type="number"：这种类型将不会利用<option>标签为每个数字创建一个<select>列表，而是使你能够指定一个 min 和 max 值，以及数字之间的 step（步长），以在浏览器端自动生成一个列表。在程序清单 22.1 中可以看到它的使用。
- type="range"：与刚才介绍的 number 类型非常相似，这种类型使你能够指定一个 min 和 max 值，以及数字之间的步长，但是在这种情况下，它将显示为水平滑块。
- type="search"：显示为常规的文本框，但是具有额外的控件，有时用于允许用户使用“x”或类似的字符清除搜索框。
- type="url"：显示为常规的文本框，但是在使用表单验证时，内置的验证器将检查它是否是形式有效的 URL。一些移动设备默认会显示相关的按键（例如，.com 虚拟按键），而不需要额外的用户交互。

可以使用 https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input 上的图表，跟踪这些及其他<input>类型的最新状态。

## 22.6 使用 HTML5 表单验证

HTML5 中的许多特性使 Web 开发人员成为非常快乐的人。其中最简单但是最显著地改变了生活的特性之一可能是包括了表单验证。在 HTML5 表单验证问世之前，我们不得不创建基于 JavaScript 的复杂的表单验证，它会使涉及的每一个人都感到头痛。

但是这一切都一去不复返了！HTML5 默认会验证表单，除非在<form>元素中使用 novalidate 属性。当然，如果没有在任何表单字段本身中使用 required 属性，那将没有什么需要验证。如你在上一节中所学到的，不仅要针对内容（任何内容）验证字段，还要依据它们的类型对其进行验证。例如，在程序清单 22.1 中，我们具有一个必需的字段，用于电子邮件地址：

```
<p><label class="question" for="the_email">What is your e-mail
     address?</label>
<input type="email" id="the_email" name="the_email"
     placeholder="Please use a real one!"
     size="50" required /></p>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

如图 22.2 和图 22.3 所示，可以看到表单会自动验证内容是否存在，而且当你尝试在字段中输入一个无用的字符串而不是电子邮件地址时还会提醒你输入错误。

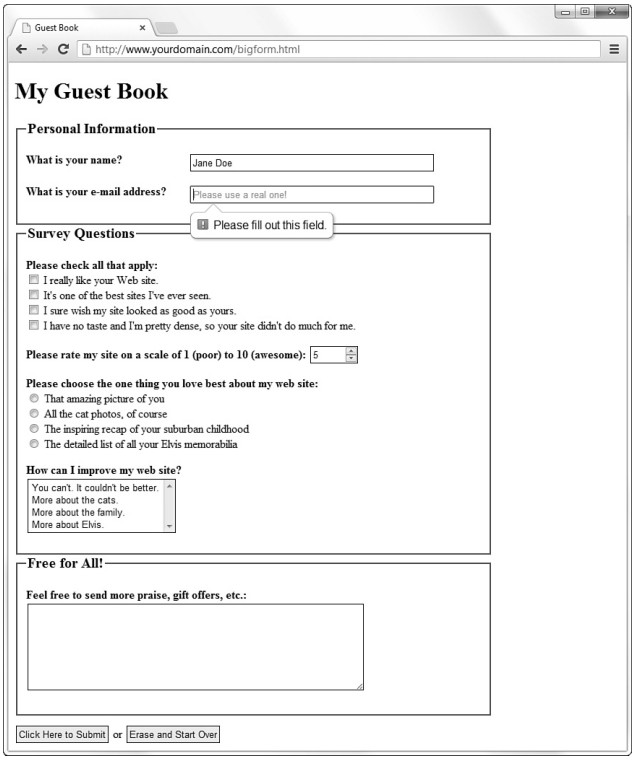



图 22.2

如果在必需的字段中没有输入内容，那么在尝试提交表单时会导致验证错误

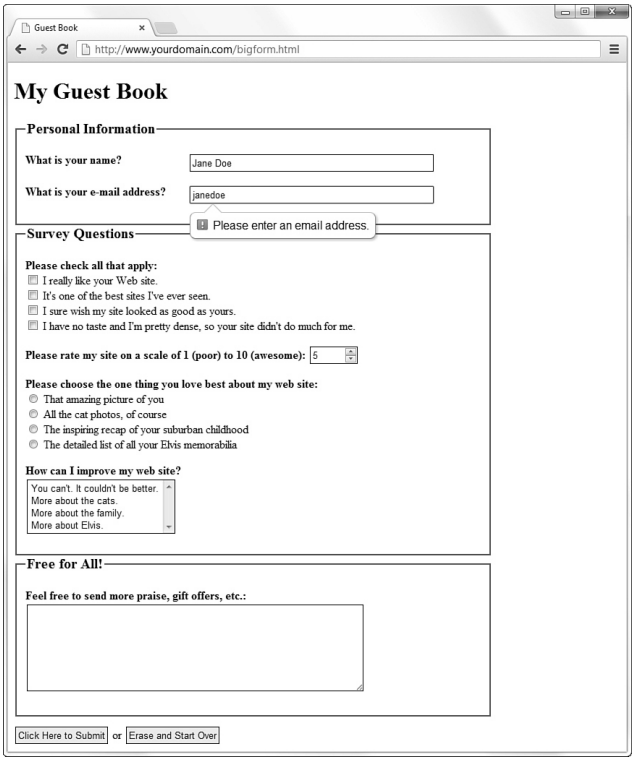



图 22.3

如果在期望一个电子邮件地址的字段中输入格式错误的内容，那么在尝试提交表单时会导致验证错误

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**By the Way**

**注意：**电子邮件地址的验证开始和结束于它只是看上去像一个电子邮件地址。在缺少费时的后端处理脚本的情况下，这种模式匹配实际上是可以对电子邮件地址所进行的唯一一种“验证”类型。

可以使用<input />字段的 pattern 属性指定你自己的模式匹配要求。pattern 属性使用正则表达式，这是一个非常大的主题，足以编写一本它自己的图书。但是考虑一下示例。如果你希望确保<input />元素只包含数字和字母（没有特殊字符），可以使用以下代码：

```
<input type="text" id="the_text" name="the_text"
     placeholder="Please enter only letters and numbers!"
     size="50" pattern="[a-z,A-Z,0-9]" required />
```

这里的模式指示如果字段包含 a~z 之间的任何字母、A~Z 之间的任何字母（大写）以及 0~9 之间的任何数字，它就是有效的。如果希望在不购买一整本书的情况下学习关于正则表达式的更多知识，可以看看 http://regexone.com/上的在线教程。

## 22.7 提交表单数据

表单通常包括一个按钮，用于把表单数据提交给服务器上的脚本或者调用一个 JavaScript 动作。可以利用 value 属性把你喜欢的任何标签放在 Submit（提交）按钮上：

```
<input type="submit" value="Place My Order Now!" />
```

除非使用 CSS 更改样式，否则一个灰色按钮将调整大小，以适应你放在 value 属性中的标签。当用户单击它时，将把表单上的所有数据项都发送给在表单的 action 属性中指定的电子邮件地址或脚本。

还可以包括一个 Reset（重置）按钮，用于清除表单上所有输入的内容，使得如果用户改变主意或者犯了错误，那么他们可以重新开始。使用的代码如下：

```
<input type="reset" value="Clear This Form and Start Over" />
```

如果标准的 Submit（提交）和 Reset（重置）按钮在你看来有点平淡，记住可以使用 CSS 编排它们的样式。如果这还不够好，你将很高兴地知道有一种轻松地方式用你自己的图形代替这些按钮。要为 Submit（提交）按钮使用你所选的图像，可以使用以下代码：

```
<input type="image" src="button.gif" alt="Order Now!" />
```

button.gif 图像显示在页面上，当用户单击该图像时，也会提交表单。可以包括通常用于<img />标签的任何属性，比如 alt 和 style。

表单元素还包括一种普通按钮类型。当在<input />标签中使用 type="button"时，将会得到一个自己不会执行任何动作的按钮，但是可以使用 JavaScript 事件处理程序（比如 onclick）给它指定一个动作。不过，由于为此需要 JavaScript 交互，因此最好坚持使用 submit、rest 或 image 类型，或者使用实际的<button>元素（在 IE 9 或更高版本中），它具有 type 属性 submit 和 rest，并且与具有相同 type 的<input>元素相比，可以稍微灵活一些地编排它的样式。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 22.8 小结

本章演示了如何创建 HTML 表单，当它们挂钩到一个后端处理脚本时，将允许访问者给你提供信息（这超出了本书的范围）。

本章介绍所有主要的表单元素，包括一点关于表单处理脚本如何解释那些元素的名称和值属性的知识。当你准备好试验一个后端处理脚本时，应该熟谙前端的细节。

我们没有深入讨论该信息，因为表单处理需要外部脚本处理该表单。不过，还有大量的事情可以做，像你希望的那样设置表单的外观和动作，包括表单验证，因此在下一步涉足表单交互性之前，还可以进行许多实践。

表 22.1 总结了本章中介绍的 HTML 标签和属性。

**表 22.1　　本章介绍的 HTML 标签和属性**

| 标签/属性 | 作用 |
|---|---|
| <form>...</form> | 指示一个输入表单 |
| **属性** | |
| action="scripturl" | 提供用于处理该表单输入的脚本的地址 |
| method="post/get" | 指示如何将表单输入发送给服务器。通常设置为 post，而不是 get |
| <label>...</label> | 为与之关联的表单元素提供信息 |
| <fieldset>...</fieldset> | 组合一组相关的表单元素 |
| <legend>...</legend> | 给一组相关的表单元素提供一个标签 |
| <input /> | 一个用于表单的输入元素 |
| **属性** | |
| type="controltype" | 给该输入构件提供一种类型。一些可能的值有：checkbox、hidden、radio、reset、submit、text 和 image 等 |
| name="name" | 在把该项目传递给脚本时给它提供一个独特的名称 |
| value="value" | 文本或隐藏项目的默认值。对于复选框或单选按钮，它是与表单一起提交的值。对于重置或提交按钮，它是按钮本身的标签 |
| src="imageurl" | 显示图像的源文件 |
| checked | 用于复选框或单选按钮，指示该项目是选中的 |
| autofocus | 在加载表单时把焦点放在元素上 |
| required | 指示应该依据类型（在合适时）验证字段的内容 |
| pattern="pattern" | 指示应该相对于这个正则表达式验证该字段的内容 |
| size="width" | 指定文本输入区的宽度（字符数） |
| maxlength="maxlength" | 指定在文本区中可以输入的最大字符数 |
| <textarea>...</textarea> | 指示多行文本输入表单元素。可以包括默认的文本 |
| **属性** | |
| name="name" | 指定传递给脚本的名称 |
| rows="numrows" | 指定该文本区显示的行数 |

异步社区会员 Kensuke(18319066421) 专享 尊重版权

续表

| 属性 | |
|---|---|
| cols="numchars" | 指定该文本区显示的列数（字符数） |
| autofocus | 在加载表单时把焦点放在元素上 |
| required | 指示应该依据类型（在合适时）验证字段的内容 |
| pattern="pattern" | 指示应该相对于这个正则表达式验证该字段的内容 |
| <select>...</select> | 创建可能项目的菜单或滚动列表 |
| **属性** | |
| name="name" | 显示传递给脚本的名称 |
| size="numelements" | 要显示的元素个数。如果指定了 size，选项将变成一个滚动列表。如果没有提供 size，选项将是一个下拉选择列表 |
| multiple | 允许在列表中进行多项选择 |
| required | 指示应该为选项验证字段 |
| <optgroup>...</optgroup> | 指示一组<option>元素 |
| **属性** | |
| label="label" | 为组提供一个标签 |
| <option>...</option> | 指示<select>元素内可能的项目 |
| **属性** | |
| selected | 通过包括这个属性，在列表中默认将选择<option> |
| value="value" | 在提交表单时如果选择了该<option>，指定要提交的值 |

## 22.9 问与答

**问：有任何方式创建大量的文本框，而无需为它们都使用不同的名称吗？**

**答：**是的。如果为表单中的多个元素使用相同的名称，它们的对象将构成一个数组。例如，如果利用名称 member 定义 20 个文本框，则可以把它们称为 member[0]~member[19]。这也适用于其他类型的表单元素。

**问：如果 HTML5 包含表单验证，我还必须再次关心验证吗？**

**答：**是的，你需要。尽管 HTML5 表单验证非常棒，你仍然应该在后端验证发送给你的表单信息。后端处理超出了本书的范围，但是作为一条规则，你永远也不应该信任任何用户输入——在执行使用它们的动作之前总是要对其进行检查（尤其是当与数据库交互时）。

## 22.10 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 22.10.1 问题

1．你将使用什么 HTML 代码创建一个留言板表单，要求某个人提供他或她的名字和性

异步社区会员 Kensuke(18319066421) 专享 尊重版权

别？假定你在/scripts/formscript 中建立了一个表单处理脚本，并且需要包括以下隐藏的输入元素，以告诉脚本把表单结果发送到哪里：

```
<input type="hidden" name="mailto" value="you@yoursite.com" />
```

2．如果你创建了一个名为 submit.gif 的图像，将怎样把它用作在“问题 1”中创建的表单的 Submit（“提交”）按钮？

3．<form>标签的下面哪个属性用于确定将把数据发送到哪里？

```
a. action
b. method
c. name
```

## 22.10.2 答案

1．可以使用如下所示的 HTML 代码（当然，还要包括合适的 DOCTYPE 及其他结构标记）：

```
<form name="form1" method="post" action="/scripts/formscript">
<input type="hidden" name="mailto" value="you@yoursite.com" />
<p><label for="name">Your Name:</label>
<input type="text" id="name" name="name" size="50" /></p>
<p>Your Gender:
<input type="radio" id="male" name="gender"
     value="male" /><label for="male">male</label>
<input type="radio" id="female" name="gender"
     value="female" /><label for="female">female</label>
<input type="radio" id="go_away" name="gender"
     value="mind your business" />
    <label for="go_away">mind your business</label></p>
<p><input type="submit" value="Submit Form" /></p>
</form>
```

2．利用下面这段代码：

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

替换以下代码：

```
<input type="submit" value="Submit Form" />
```

3．a。action 属性确定把数据发送到哪里。

## 22.10.3 练习

- 创建一个使用了所有不同类型的输入元素和选择列表的表单，确保你理解了它们都是如何工作的。
- 学习一点有关正则表达式的知识，并使用 pattern 属性实现一些自定义的验证。
- 调查你的 Web 托管提供商的表单处理选项，并且使用 Web 托管提供商提供给你使用的脚本来处理在前面的练习中创建的表单。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 23 章

# 组织和管理 Web 站点

在本章中你将学到：

- 怎样确定一个页面是否足以处理你的所有内容；
- 怎样组织简单的站点；
- 怎样组织更大的站点；
- 怎样编写易于维护的代码；
- 怎样开始进行版本控制。

本书的主要篇幅都在引领你设计和创建自己的 Web 内容，从文本到图形和多媒体。在这个过程中，我指出了你可以考虑内容的生命周期的一些方式。但是在本章中，你将学习如何作为一个整体看待自己的工作。

本章将说明如何考虑组织和展示多个 Web 页面，使得访问者将能够自如在它们当中导航，而不会被弄糊涂。你还将学习一些方式，使 Web 站点令人难忘，从而使访问者能够一遍又一遍地反复访问它。Web 开发人员使用术语黏性（sticky）来描述人们不想离开的页面。希望本章可以帮助你使自己的 Web 站点具有十足的粘性。

由于 Web 站点可以（并且通常应该）频繁更新，因此有必要创建可以轻松地维护的页面。本章将说明如何向页面中添加注释和其他文档，以便你（或者你的其他任何工作人员）可以理解和修改你的页面。它还将介绍版本控制，以便你可以单独或者作为团队的一份子进行创新，而不会覆盖你可能希望保存的工作。

Try It Yourself：评估你的组织方式

到本书中的这个位置，你应该具备了足够多的 HTML 和 CSS 知识，来制作 Web 站点的大部分内容。你可能已经创建了许多页面，并且也许甚至把它们发布到了网上。

在学习本章的过程中，要考虑现在将如何组织你的页面，以及可以怎样改进该组织方

异步社区会员 Kensuke(18319066421) 专享 尊重版权

式。你在 HTML 中使用过注释或者创建过文档以便为将来的 Web 站点维护人员说明内容的组织方式吗？如果没有，现在就应该开始这样做。在这个过程中，如果你决定重新进行设计，并且这涉及更改几乎所有的页面，也不要感到吃惊，其结果很可能值得付出这样的努力。

## 23.1 当一个页面就足够时

构建和组织一个有吸引力并且有效的 Web 站点并非总是需要成为一项复杂的任务。如果为单个实体（比如一个本地事件）创建 Web 展示，它只需要少量非常具体的信息，那么就可以在单个页面上有效地展示该信息，而无需许多华丽的图形。事实上，单个页面的 Web 展示具有多个积极的特点。

- 站点上的所有信息将比内容更广泛的站点下载得更快。
- 可以利用单个打印命令把整个站点打印到纸上，即使它的页面有几张纸那么长。
- 访问者可以轻松地把站点保存在他们的硬盘驱动器上以便将来参考，尤其是当它使用最少量的图形时。
- 相同页面的不同部分之间的链接通常能够比指向其他页面的链接更快地做出响应。

图 23.1 显示了一个 Web 页面的第一部分，它作为单个长长的页面比作为一个多页面站点能够更好地为其目标受众服务。与大多数介绍性页面一样，该页面首先简洁地解释了页面是关于什么的，以及谁将希望阅读它。详细的目录允许访问者直接跳转到包含他们发现最感兴趣的材料所在的部分。如果打印这个“页面”，它将包含大约 6 页纸的关于驱动 Web 站点流量的文本，访问者可能考虑打印它们并在以后阅读，也许同时还会做笔记。

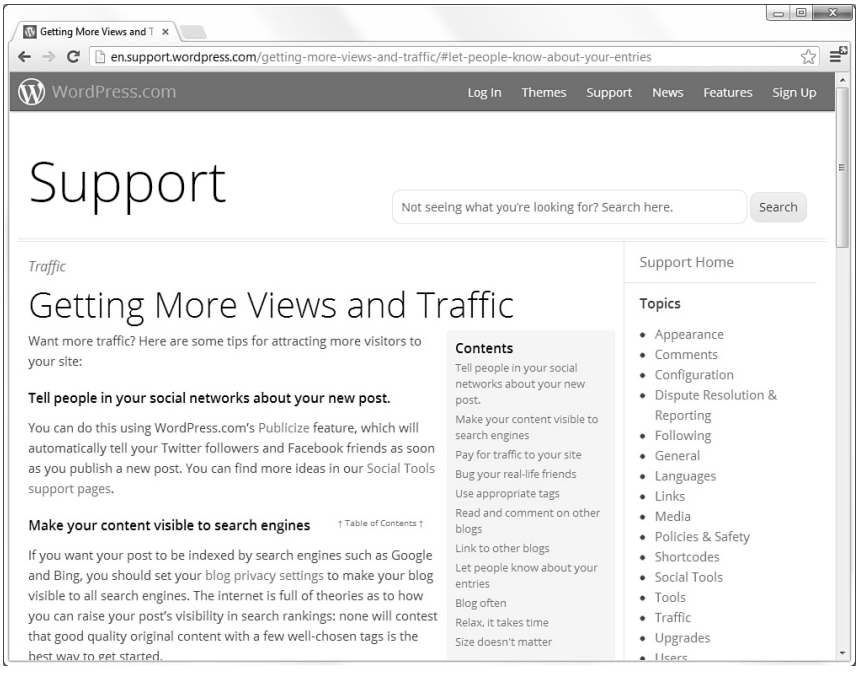



图 23.1

良好的目录可以使长页面更容易导航

当页面包含目录以把信息区域分隔开时，使每个较短的部分包含一个指回目录的链接

异步社区会员 Kensuke(18319066421) 专享 尊重版权

也是一种常见的情况，这样在页面上四处导航时就会感觉到非常像在多页面站点上导航。由于长页面的这些类型的内容打算作为一个方便的参考，访问者肯定更喜欢创建书签或者保存单个页面的便利性，而不必保存 8 个或 10 个单独的页面。维基百科（Wikipedia）内包含有最常见的单页面信息 Web 站点的示例，参见 www.wikipedia.org。如果考虑每个条目都充满了丰富的内容作为它自己的“站点”，维基百科（Wikipedia）内的单页面站点——以及它们自己的目录——将代表数百万个打印的页面。

在网上体验过许多漂亮、有效的图形布局之后，你可能忍不住要忘记一种良好的老式大纲，在站点内组织充满基于文本的内容的长页面时，它通常是一种最清晰、最高效的方式。对于会被大量访问的单页面界面（也称为单页面应用程序（application））尤其如此，它尝试把桌面应用程序的所有交互性都引入 Web 浏览体验中。这些应用程序通常是使用 HTML 和 JavaScript 框架构建的，并且包括重要的视觉设计元素。事实上，这些站点通常用于发布设计代表作选辑，而不是我在这里使用的基于文本的内容类型。

## 23.2 组织简单的站点

除了上述的单页面应用程序和代表作选辑站点的特例之外，单页面 Web 站点倾向于只服务于“即将到来”或占位符目的。如果在 Web 上花费一点时间，很快就会知晓大多数公司和个人通过给他们的读者提供了更好的服务，他们采用的方法是：把他们的站点划分成较短的、可以快速阅读的页面，周围环绕着图形导航元素，允许他们收集在几次单击内可能想要的几乎所有的信息。而且，使用多个页面代替一系列非常长的页面最小化了滚动页面的需要，而对于那些使用移动设备查看完整的站点或者具有低分辨率的显示器（小于 800×600 像素）的访问者来说，滚动页面可能特别麻烦。

***Did you Know?***

**提示：**无论站点有多大，仔细组织资源都是一个好主意。例如，把用于 Web 页面的图像放在一个名为 images 的单独的文件夹中就是朝着组织站点所迈出的一步。类似地，如果具有可供下载的文件，就把它们放在一个名为 downloads 的文件夹中。这使得基于特定的类型（HTML 页面、PNG 图像等）跟踪 Web 页面资源要容易得多。此外，如果把站点组织成区域，比如 Company、Products、Press 等，出于相同的组织原因，可以把各个文件放入类似地命名的目录（如 company、products、press 等）中。

Web 站点的基本目标是使个人或组织在 Internet 上可见，而且更重要的是充当站点自身内的信息的门户。站点的主页应该给用户提供足够多的信息，以提供站点组织结构的清晰图景，以及传统的联系信息或电子邮件地址，以便提交问题或反馈。它还应该提供通往站点中的其他页面内的高度结构化信息的路径。如图 23.2 所示的主页提供了所有这些良好特性的示例：基本的信息、联系信息，以及通往针对多个受众的信息的路径。

开发人员在开始创建 Web 站点时最常犯的错误之一，是创建与站点上的其他页面看起来根本不同的页面。一个同样严重的错误是使用其他数千位 Web 创建者也在使用的公共可用的相同剪贴画。记住，在 Internet 上，点击一次就可以带你周游世界。使页面难忘并且被认可为一个有粘附力的站点的唯一方式是使所有页面遵循独特的、清楚明白的视觉主题。换句话说，与其他站点相比要努力追求独特性，而在站点自身内则要保持一致性。

异步社区会员 Kensuke(18319066421) 专享 尊重版权




图 23.2

这个大学的主页使用了一种基本设计、最小量但是有用的图形以及清晰的结构，可以吸引用户探索更多的信息

下面举例说明一致性如何有助于使站点更有粘附力，考虑你可能访问过的大型的流行站点，比如 ESPN.com。如果访问 ESPN.com 上的 MLB 区域，如图 23.3 所示，然后访问 NFL 区域（见图 23.4），将注意到非常相似的结构。

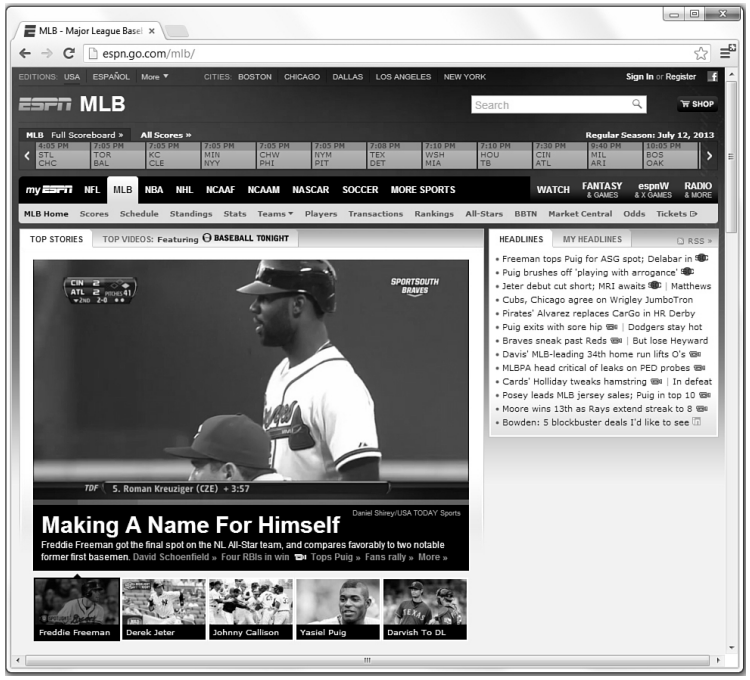



图 23.3

ESPN.com 上的 MLB 区域

异步社区会员 Kensuke(18319066421) 专享 尊重版权

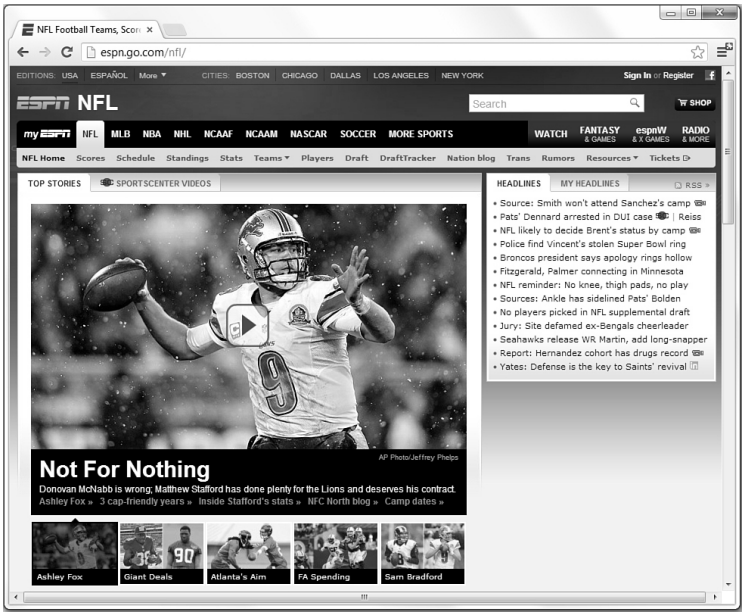



图 23.4

ESPN.com 上的 NFL 区域

在这两个示例中，你看到导航元素位于页面顶部（包括一些子导航元素），在页面中间有一大块区域用于特色项目图形，右边的矩形包含指向头条新闻的链接，主图像下面有一组次级矩形，把读者引导至额外的新闻故事。MLB 区域与 NFL 区域之间的唯一区别在于配色方案：MLB 区域是以蓝色为主导色的配色方案的一部分，而 NFL 区域的主导色则是绿色。不过，在这两个区域中，你知道如果你想阅读流行的新闻故事，都要查看页面的右边。如果你想导航到站点中的另一个区域或者站点的主页，就要寄望于页面左上方的导航元素。

一致的设计和组织有序的元素有助于确保用户将能够自信地在你的全部内容中导航。从维护的角度讲，一致的结构模板将使你能够重用底层的代码段。这种代码重用通常是通过动态服务器端程序设计发生的，这超出了本书的范围，但是一般来讲，它意味着将不需要一遍又一遍地反复复制和粘贴相同的 HTML、CSS 和 JavaScript 代码，而只需要客户端代码存在于仅仅一个位置，就可以动态地应用于内容。因此，例如，如果想把背景从蓝色改为绿色，将不需要更改数千个文件，而只需执行一次更改即可。

## 23.3 组织更大的站点

对于复杂的站点，当在所有的页面中一致地使用精致的布局和图形时，它们将有助于组织和改进站点的外观。要查看怎样密切合作地创建具有美感和组织有序的作品，让我们看看几个站点的导航示例（以及底层的组织结构），这些站点给多个不同的受众展示了大量的信息。

图 23.5 显示了 Amazon.com 主页，特意选择了侧面的导航元素。Amazon 从事产品销售，朴素且简单。因此，让 Amazon 将产品目录显示为主导航元素就是有意义的。

尽管 Amazon 从事产品销售，它还不得不提供关于它是谁、如何联系它等信息，以及其他辅助但很重要的信息，以增进企业与顾客之间的关系。指向这类信息的链接出现在 Amazon.com Web 站点的脚注中（或底部），位于这幅截屏图的可视区域之外。在创建站点

异步社区会员 Kensuke(18319066421) 专享 尊重版权

模板时，必须确定最重要的内容区域以及如何组织该内容，还要记住给用户提供基本的信息，尤其是当该信息可以增强图像并且使用户感觉好像你很重视他们必须说点什么时。

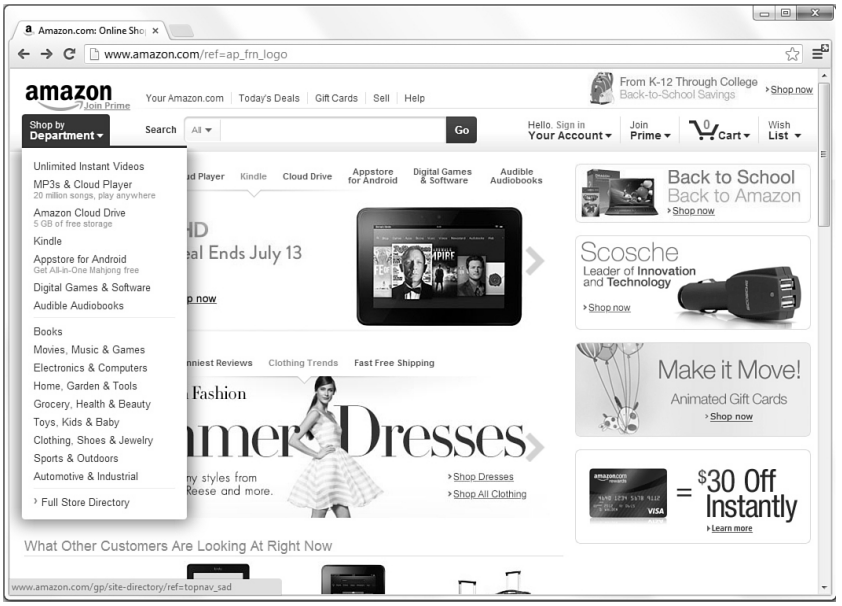



图 23.5

Amazon.com 将产品目录显示为主导航元素

下一个示例是 Peet's Coffee & Tea Web 站点（www.peets.com）内的一个次级页面。该 Web 站点中的所有页面都遵循展示导航和子导航系统的常见类型之一：水平条用于主导航系统，用于该区域的次级元素则放在左边的垂直栏中。如图 23.6 所示，用户当前浏览的区域（Community）在主导航系统中是高亮显示的（通过细微的字体颜色变化），次级导航系统中的特定页面也是如此（通过改变元素的背景色）。这些类型的视觉指示可以帮助用户在站点内确定自身的方位。使用视觉指示是一种有用的策略，因为你的用户可能通过搜索引擎或者另一个 Web 站点中的链接到达某个页面。在用户到达后，你将希望他们有一种宾至如归的感觉，至少感觉好像他们知道自己在你的站点中所处的位置。

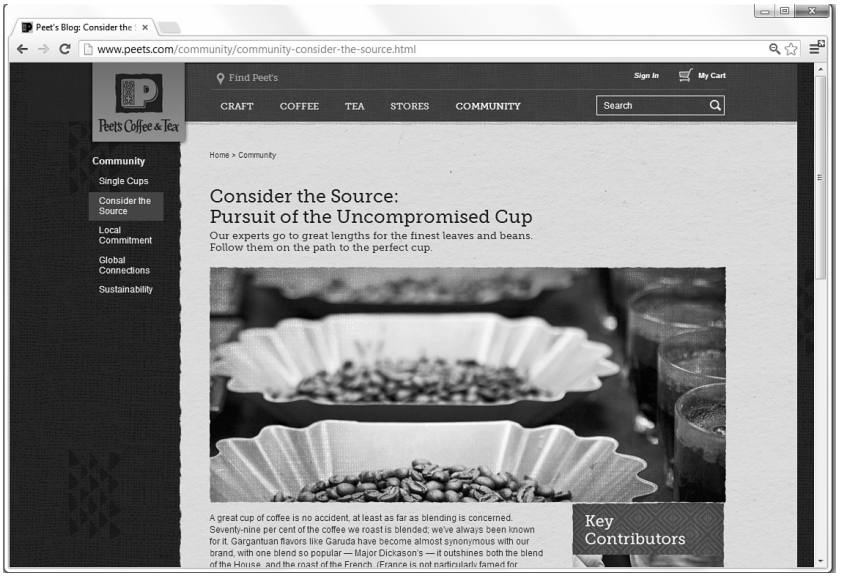



图 23.6

Peet's Coffee & Tea Web 站点的次级页面显示一个主导航元素以及页面左边的次级导航元素处于选择状态

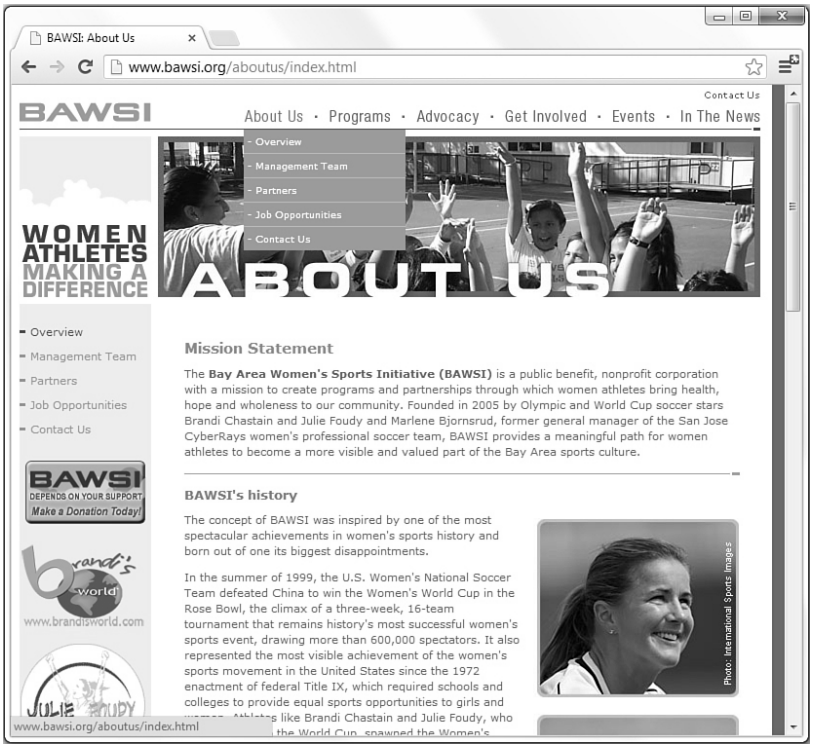
异步社区会员 Kensuke(18319066421) 专享 尊重版权

通过不同的主导航元素（Craft、Coffee、Tea、Stores 和 Community）可以看到：Peet's Coffee & Tea Web 站点必须为出于许多不同的原因而来到 Web 站点的许多不同类型的人的需求提供服务。在组织你自己的站点内容时，在确定对你最重要的信息，以及对你的用户最重要的信息，并创建一种导航模式，它能够在两者之间找到一种折衷办法。

图 23.7 是导航风格的另一个示例，这一次稍稍改变了标准的顶部导航/左侧导航模式。在这个示例中，左侧导航系统（在这里是次级导航）也出现在主导航系统下面的一个下拉菜单中（参阅第 17 章，了解关于如何实现它的信息）。把鼠标指针悬停在其他任何主导航元素上都将显示类似的菜单。这种模式使用户对完整的站点地图了如指掌，因为他们在其他任何页面上单击一次就能到达站点中的任意位置。

图 23.7

BAWSI.org Web 站点显示了依附于每个主导航元素的子导航元素




另外请注意，侧面的导航窗口中的 Overview 链接的样式与窗口中的其他链接相比有点不同，它是利用更重的紫色文本编排样式的，指示访问者他们位于哪个页面上。这种视觉细节与你在 Peet's Coffee & Tea 站点上看到的相似，利用一种低调的方式让用户感觉他们在当前导航模式内的什么位置。

有许多种不同类型的导航风格，指示用户位于何处以及他们接下来可能想去哪里的方式也是多种多样，你可以从中做出选择。记住以下事实：研究反复表明，当一次提供 7 个以上的选择时，人们将会变得糊涂并且会恼怒，他们感到最舒适的是 5 个或更少的选择。因此，只要有可能，就应该避免一个接一个地展示 5 个以上的链接（无论是在列表中还是作为图形图标）——当然一定要避免同时展示 7 个以上的链接。Amazon.com 在这里获得了通行证，因为它是一家 Internet 超市，用户期望在他们到达那里时有许多“部门”可以购物。但是，当需要在导航列表中展示 7 个以上的链接时，可以把它们分成多个列表，给每 5 个或 7 个项目提供单独的标题。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

如果你避免在任何页面上放置两个以上（或者至多 3 个）离开主页的链接，它也可以帮助读者头脑清醒地导航你的站点。你还总是应该在读者阅读了一个附属页面之后给他们发回一个主目录页面（或主页）。换句话说，要尽量设计一种稍微有些扁平的链接结构，其中大多数页面的链接深度都不超过 1 个或两个。就算要这样，你也不希望访问者不得不过于依赖他们的浏览器的 Back（“后退”）按钮来导航你的站点。

## 23.4 编写易于维护的代码

如果你在阅读本书前做过了任何编码工作，就已经知道编写可以维护的代码是多么重要。也就是说，你或者其他人应该能够在以后查看你的代码，并且不会被它彻底地弄糊涂。这里的挑战是使你的代码能够被尽可能快地理解。当你回过头来查看以前编写的页面，并且对当时在考虑什么或者为什么要这样编写代码没有毫无头绪时，就会对此深有感触。幸运的是，有一种方式可以抵御这个明显失忆的问题。

### 23.4.1 利用注释记录代码

无论何时开发 HTML 页面或 CSS 代码段，都要记住你或者其他人几乎肯定需要在某一天修改它。简单的文本 Web 页面通常容易阅读和修改，但是带有图形、表格和其他布局技巧的复杂页面可能相当难以解释。

**注意：**要在样式表中包括注释，可以利用“/*”开始注释，并利用“*/”结束它们（注释的代码应该位于这些字符之间）。

HTML 的“<!--”和“-->”注释语法在样式表中不会正确地工作。

**By the Way**

为了查看我正在谈论什么，可以在 Web 浏览器中访问几乎任何页面，并查看它的源代码。如果使用 Internet Explorer，可以右击任何页面，并选择 View Source（“查看源”）命令。如果使用 Chrome 或 Firefox，可以右击任何页面，并选择 View Page Source 命令。你可能会看到一堆乱七八糟的代码，很难解释为纯粹的 HTML。这可能是由于内容管理软件系统动态地生成了标记，或者可能是由于它的人类维护者不重视结构、阅读的简易性、代码注释以及使代码可以被人类轻松阅读的其他方法。出于维护你自己的页面的原因，我鼓励你对 HTML 标记和样式表条目强加更多一点的秩序。对于你可能在将来编写的任何 JavaScript 代码也是如此：正确地缩进它可以使你（以及你将来的开发合作伙伴）受益匪浅。

**提示：**注释的一种方便应用是隐藏 Web 页面中目前正在构造的部分。这将不会使文本和图形可见并解释它们正在构造，而可以从视图中完全隐藏它们，只需仔细地在你不希望显示的 HTML 内容周围放置开始和结束注释指示符即可。这种方法非常适合于循序渐进地处理页面的各个部分，并且仅当你完成时才向世界显示最终的结果。

**Did you Know?**

可以使用 HTML 开始和结束注释语法（即“<!--”和“-->”）来封闭注释，使之只能被你自己或者你的合作者处理。在利用浏览器查看时，这些注释将不会出现在 Web 页面上，但是可以被在文本编辑器中或者通过 Web 浏览器的 View Source（“查看源”）或 View Page

异步社区会员 Kensuke(18319066421) 专享 尊重版权

Source 功能检查 HTML 代码的任何人阅读。下面的示例提供了一点复习知识，只是为了说明如何编码注释：

```
<!-- This image needs to be updated daily. -->
<img src="headline.jpg" alt="Today's Headline" />
```

如这段代码所揭示的，位于<img />标签之前的注释提供了如何使用图像的提示。任何阅读这段代码的人都立即就会知道这是一幅必须每天更新的图像。Web 浏览器将完全忽略注释中的文本。

**Try It Yourself：注释代码**

现在值得你花时间把迄今为止创建的所有 Web 页面和样式表检查一遍，并添加你或其他人在将来修改它们时可能发现有用的任何注释。下面给出了操作方法。

1. 插入一条注释，解释任何奇特的格式化或布局技术，把该注释就插入在使之发生的标签前面。
2. 在<img />标签之前，使用一条注释简要描述任何重要的不能从 alt 消息明显看出其作用的图形。
3. 考虑使用一条注释（或者多条注释）总结<table>的单元格被期望将如何对齐。
4. 如果使用十六进制颜色代码（比如 ），就要插入一条注释，指示它实际上是哪种颜色（蓝紫色）。
5. 缩进注释，帮助它们引人注目，并使注释以及 HTML 或 CSS 代码更容易阅读。不要忘记在 HTML、CSS 和 JavaScript 代码自身中也使用缩进，使它们也更容易阅读，在下一节中将讨论这个主题。

## 23.4.2 缩进代码以使之更清晰

我必须坦白，在全书中，我认真地向你灌输了一种 HTML 代码开发风格，但是没有真正解释为什么要这样做。现在应该揭示这个秘密了。你无疑注意到了一种一致的模式，即本书中的所有 HTML 代码都采用了缩进。更确切地讲，每个子标签都向右进行了缩进，使其距离它的父标签两个空格。而且，标签内的内容如果跨越多行，则在该标签内进行缩进。

了解缩进价值的最佳方式是查看一些没有缩进的 HTML 代码。你知道如何演唱这首歌：“You don’t know what you’ve got ‘til it’s gone”（在你失去前永远不知道自己得到过什么）。无论如何，下面显示了一个非常简单的表格，它被编码成没有任何缩进效果：

```
<table><tr><td>Cell One</td><td>Cell Two</td></tr>
<tr><td>Cell Three</td><td>Cell Four</td></tr></table>
```

这里不仅没有缩进，而且没有描绘表格内的行和列。现在把这段代码与下面的代码做一下比较，它描述了相同的表格：

```
<table>
  <tr>
    <td>Cell One</td>
    <td>Cell Two</td>
  </tr>
  <tr>
    <td>Cell Three</td>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    <td>Cell Four</td>
  </tr>
</table>
```

这段大量缩进的代码使得我们非常清楚是如何通过<tr>和<td>标签划分行和列的。

在涉及使 HTML 代码更容易理解和维护时，一致的缩进甚至可能比注释更重要。你不必认同这种特定的缩进策略。如果你愿意使用 3 个或 4 个空格代替两个空格，那也不错。如果你希望使内容更紧凑一点并且在标签内不缩进内容，它也会工作。从本节中获得的要点是：开发一种你自己的编码样式然后不折不扣地坚持它很重要。

**提示：**如果你是与其他人合作（或者计划这样做）开发 Web 页面，可以考虑聚在一起成立一个小组，制定一种一致的编码风格。这样，每个人都将在相同的页面上，对于缩进也是如此。

Did you Know?

## 23.5 考虑版本控制

如果你曾经使用过 Google Docs，就会遇到一种版本控制形式。在 Google Docs 中，Google 会在你输入时自动保存工作的修订版。这不同于简单地自动保存你的工作（尽管它也会这样做），因为你可以恢复到在这个过程中所执行的任何修订。在使用流行的博客创作软件（比如 Blogger 或 WordPress）时，或者甚至是在编辑百科时，你可能遇到了这个概念——它们二者也使用户能够修订他们的工作，而不必覆盖从而永远删除他们以前的工作。

你可能感到迷惑的是："好吧，这与 HTML 或 CSS 何干？你正在谈论的是文档。"答案很简单，就像你可能希望恢复到文章或信件的以前版本一样，你也可能希望恢复到 HTML 或 CSS 的以前版本。这可能是由于你自始至终都遵循一个良好的思想，但是你的标记被证明是难以维持的，并且你不想完全重新开始——你只想回退到修改路径中的某个位置。

版本控制不仅仅涉及修改历史。当你开始使用版本控制系统维护代码时，将会听到如下术语。

- 提交（commit）/签入（check in）和签出（check out）：当把一个对象放入代码库中时，就在提交那个文件；在签出一个文件时，就在从代码库中获取它（其中存储了所有当前的和历史的版本）并对其进行处理，直到准备好再次提交或签入文件为止。
- 分枝（branch）：置于版本控制之下的文件在任何时刻都可以分枝或分支（fork），从而创建两条或更多的开发路径。假设你想试验一些新的显示布局或者表单交互性，但是不希望以任何方式修改现有的站点。你可能开始于一组主文件，但是后来把这组文件进行了分支以用于新的站点，并继续独立地开发它们。如果继续开发原始的文件集，则处理的将是主干（trunk）。
- 更改（change）/差异（diff）：这只是版本控制下的修改的术语（可以称之为更改或差异）。你可能也听过把差异用作一个动词，比如"我比较了文件的差异"，用于指比较一个对象的两个版本的动作（有一个底层的 UNIX 命令称为 diff）。
- 分支（fork）：当找到一种开源 GitHub 代码库时，你可能想要把它用作自己工作的

异步社区会员 Kensuke(18319066421) 专享 尊重版权

基础（或者想要为它做贡献），就可以分支该代码库，然后创建它的一个副本，以便可以按自己的进度继续处理它。从分支的代码库中，可以把提交推送（push）到自己的版本，从原始代码库中提取（fetch）更改，如果你想要把自己的更改贡献给所分支的原始代码库，还可以给原始代码库的所有者发出拉取请求（pull request）。

你将听说的术语远远不止这里列出的这几个，但是如果你概念化了代码库、（本地）工作副本，以及签入和签出文件的过程，将能够很好地为你的数字对象实现版本控制。

### 23.5.1 使用版本控制系统

有多个不同的版本控制系统可供使用，其中一些是免费和开源的，另外一些则是专有的。两个最流行的系统是Subversion（subversion.apache.org）和Git（git-scm.com）。如果你具有允许安装Subversion的Web托管服务，就可以创建自己的代码库，并且使用Subversion客户连接到它。

一个日益流行的工具是Git，它是一种版本控制的分散式方法，并且还为那些想要从代码库开始起步但是不一定希望、需要或理解随之产生的所有额外的安装和维护开销的用户提供了众多工具和托管选项。用于Git代码库的这样一个托管选项是GitHub（github.com），它允许用户创建账户，然后根据需要免费存储和维护尽可能多的代码库（只要它们是开源的即可），同时还为那些希望维护私有代码库的用户提供了付费解决方案。

对于任何想要试水版本控制的人，我建议使用Git和GitHub，因为它们相对容易使用，并且是免费的跨平台工具。GitHub Help站点是开始的极佳位置，参见http://help.github.com/。已经免费的GitHub账户的一个额外的优点是能够使用Gist（gist.github.com）与其他人共享代码段（或整个页面），那些代码段本身就是Git代码库，因而进行了版本化，并且自身是可分支的。GitHub代码库（包括Gist）也是对你的工作开始进行版本控制的优秀方式。

## 23.6 使用HTML和CSS框架

如果使用诸如Wordpress（www.wordpress.org）或Drupal（www.drupal.org）之类的内容管理系统（CMS）给你的Web站点提供动力，那么你最终将使用一个设计用于这些系统之一的展示模板，但是对于那些不想使用CMS而是想要一个涉足高级HTML和CSS展示的起点的人来说该怎么办？在过去几年，Web开发世界见证了HTML和CSS（或“前端”）框架的兴趣，它们有助于解决这个问题。其中许多框架都是开源的，并且可供下载或者从GitHub代码库进行分支。这些框架通常还包括高级JavaScript功能，但是不会让这吓到你，有大量的示例和文档可以帮助你通过工作的初始阶段。

我建议使用3个流行的前端框架。

- Bootstrap：该框架是由Twitter（推特）上的工程师在内部开发的，它是开源软件，任何人都可以免费使用它来试水现代设计元素。在http://getbootstrap.com/上可以了解到更多信息，包括简单的“Get Started”部分，其中解释了它包括什么以及如何

异步社区会员 Kensuke(18319066421) 专享 尊重版权

使用它。

- Foundation：另一个开源框架，Foundation 强调响应性设计，使得具有各种不同设备（从台式机到手机）的人都可以享受并使用你的 Web 站点。在 http://foundation.zurb.com/上可以了解到更多信息，它包括一个内容广泛的“Getting Started”部分，其中详细描述了你可以使用的显示模板的成分。
- HTML5 Boilerplate：最简洁的框架之一，它可能对于初学者是最有用的，因为它提供了你所需的基础，而不会由于种种可能性而使你不知所措。在 http://html5boilerplate.com/上可以了解到更多信息，并且可以查看在 GitHub 代码库内维护的文档。

尽管前端框架对于加快 Web 开发的一些基础性工作的速度可能非常有用，但是会冒着落入“俗套”（cookie cutter）的陷阱，其中你的站点看起来像所有其他的站点（至少像使用相同框架的站点）。不过，利用一点创意，就可以避开那个陷阱。

## 23.7 小结

本章提供了一些示例和解释，帮助你把 Web 页面组织到一个条理分明的站点中，它将是信息丰富、有吸引力并且易于导航的。Web 用户已经变得相当机智，期望看到良好设计的 Web 站点，如果他们体验的是一种难以导航的拙劣设计，那么他们很快就会放弃你的站点。

本章还讨论了通过添加注释和缩进使代码易于维护的重要性。注释很重要，不仅可以在你以后回过头来查看以前编写的代码时给你以提醒，而且可以在别人继承你的代码时给他们提供指导。缩进似乎是一个美感问题，但它可以帮助你在很短的时间内快速分析和理解 Web 页面的结构。

由于你可能很快就需要代码管理工具，为你自己或者在你的小组中为你自己及其他开发人员提供服务，本章介绍了几个版本控制的概念。版本控制使你能够进行创新，而不会丢失稳定的、生产质量的工作，还可以为其他开发人员提供更多的机会在你的代码库内工作。

最后，本章介绍了一点关于 HTML 和 CSS 框架的知识，这些框架有许多个，本章只介绍了其中几个。这些框架通过提供已经包含现代的、经过验证的标记的模板，可以帮助你加快 Web 开发项目的速度。

## 23.8 问与答

**问：在某个人查看我的页面时，添加许多注释和空格不会使它们加载得更慢吗？**

**答：**与其他更大的 Web 页面资源（比如图像和多媒体）相比，页面中的额外一点文本的大小可以忽略不计。除此之外，较慢的拨号式调制解调器连接通常会在传输文本时对它们进行相当好的压缩，因此添加空格来格式化 HTML 通常不会更改传输时间。必须输入数百个注释文字，才可能在加载页面时导致额外 1 秒钟的延迟。另请记住：利用许多人使用的宽带连接（如电缆、DSL 等），文本传输得非常快。图形是减慢页面加载速度的成分，因此尽

异步社区会员 Kensuke(18319066421) 专享 尊重版权

管需要竭尽所能地优化图像，但是可以自由地使用文本注释。在 https://developers.google.com/speed/docs/best-practices/payload?hl=fr#MinifyHTML 上还可以学习关于“缩小”HTML 和 CSS（以及 JavaScript 代码）的概念的更多知识。

**问：使用版本控制对于我的小型个人 Web 站点似乎有些矫枉过正。我必须使用它吗？**

**答：**当然不是。任何类型（个人或其他类型）的 Web 站点都不需要置于版本控制或其他备份系统之下。不过，大多数人都经历过某种数据丢失或 Web 站点崩溃的情况，因此如果没有使用版本控制，我强烈建议至少执行某种自动化备份，把文件备份到一个外部系统。至于“外部系统”，我的意思是指任何外部设备，无论是连接到计算机的物理驱动器，还是基于云的备份服务，比如 Dropbox（www.dropbox.com）。

## 23.9 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 23.9.1 问题

1. 用于确保所有的页面构成单个黏性 Web 站点的 3 种方式是什么？
2. 在主页中总是应该包括哪两类信息？
3. 你想要对将来的 Web 页面编辑器说：“不要改变我的这幅图像，它是我永垂不朽的唯一机会。”但是你不希望查看页面的用户看到该消息，怎样做到这一点？

### 23.9.2 答案

1. 使用一致的背景、颜色、字体和样式。在链接导向的页面顶部重复使用相同的链接文字或图形。在站点的每个页面上重复使用相同的小标题、按钮或其他元素。
2. 使用足够多的辨识信息，以使用户能够立即看到站点的名称并且理解它是关于什么的。此外，无论你想传达给目标受众的最重要的消息是什么，都要直接并且简洁地陈述它。无论它是你的使命陈述还是注册商标式的市场营销口号，都要确保它位于朴素的视图中。
3. 把以下注释直接放在<img />标签之前：

```
<!-- Don't change this image of me.
    It's my only chance at immortality. -->
```

### 23.9.3 练习

- 拿一支铅笔（老式的那种），把你的 Web 站点绘制成一串小矩形，并在它们之间绘制一些箭头。为每个页面的外观草绘一份粗略的概览，其方法是：在显示文本的位置放上一些胡乱写的字，并在显示图像的位置放上涂鸦的绘画。每个箭头都应该开

异步社区会员 Kensuke(18319066421) 专享 尊重版权

始于一个涂鸦的图标，它对应于箭头导向的页面的导航按钮。即使具有最新的、极其优秀的 Web 站点管理工具（它们通常不仅仅用于创建站点本身），手工绘制站点也可以让你更直观地领会站点上的哪些页面将很容易到达，以及邻近页面的布局将如何协同工作，所有这些都发生在你投入时间编写实际的 HTML 来连接页面之前。无论相信与否，当我第一次设计 Web 站点时，我仍将像这样绘制它们的草图。有时，你无法打败铅笔和纸张！

- 打印构成当前 Web 站点的 HTML 文件，并把它们都检查一遍，了解注释和代码缩进情况。是否存在一些区域，需要把其中的代码向将来可能查看它的任何人做一下解释？如果是，就添加解释性注释。辨别代码的层次结构对你来说困难吗？难以查看标题和区域吗？如果是这样，就要缩进 HTML 代码，使其结构与代码的层次结构匹配，从而使你能够快速跳转到需要编辑的区域。
- 在 GitHub 上创建一个账户，并为你的个人 Web 站点或其他基于代码的项目创建一个代码库。从此开始，通过把你所做的更改提交到 GitHub 代码库，使你的代码库与你在个人计算机上所做的工作保持同步。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# 第 24 章

# 帮助人们找到你的 Web 页面

在本章中你将学到：

- 怎样宣传你的 Web 站点；
- 怎样利用主要的搜索站点列出你的页面；
- 怎样为搜索引擎优化你的站点？

你的 Web 页面最终仅当可以被人们访问时才是有用的，如果没有人能够找到你的页面，你为创建一种有用的架构、提供有趣的内容以及正确地编码它们所付出的努力终将是徒劳无功的。你将在本章中发现的额外 HTML 标签不会在你的 Web 页面中产生任何可见的差别，但是它们极其重要，因为它们将帮助你的受众更轻松地找到你的 Web 页面。

对于大多数 Web 站点创建者而言，本章可能是本书中最容易的一章，但也是最重要的一章。你将学习如何向页面中添加元素以及如何构造站点架构，以便在某人搜索与你的主题或公司相关的词语时，增加搜索引擎将返回指向你的站点的链接的可能性，这称为搜索引擎优化（Search Engine Optimization，SEO）。

与你可能从尝试给你销售搜索引擎优化服务的公司那里所听到的相反，没有神奇的秘诀可以保证你将处于每一份搜索结果列表的顶部。不过，你可以自己遵循一组自由的最佳实践，以确保站点能够被尽可能轻松地找到。

## 24.1 宣传你的 Web 站点

你大概希望 Web 站点吸引某人的注意，否则你当初就不会费心地创建它。如果只把页面放在局域网或者公司内联网上，或者如果只把站点分布在可移动的存储介质上，那么帮助用户找到你的页面可能就不是一个太大的问题。但是，如果是把你的 Web 站点的内容添加到通过搜索引擎建立索引的数十亿个其他的内容页面中，那么把目标受众带到你的站点的确就是一项重大的挑战。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

为了处理这个问题，将需要基本理解大多数人如何决定要查看哪些页面。人们会通过 3 种基本方式知道你的 Web 站点。

- 有人把你的站点告诉了他们，并给他们提供了地址，他们直接在 Web 浏览器中输入该地址。
- 他们从别人的站点中、从聚合器和推荐服务（如 Digg 或 Reddit）或者从社交网络站点（如 Facebook、Twitter 或 Google+）上的链接或提及你的站点的内容中沿着一条链接到达你的站点。
- 他们在给 Google、Bing 或 Yahoo!等搜索引擎提供动力的数据库中发现了你的建立索引的站点。

可以利用少许时间和努力来增加 Web 站点的流量。要增加在口头上听说过你的人数，可以使用你的嘴巴，以及可供你使用的任何其他的交流渠道。如果你具有现有的联系人数据库或邮件列表，就可以向那些人宣传你的 Web 站点。把站点地址添加到你的名片或公司文献中。如果你有钱，可以购买电视和收音机广告，广播你的 Internet 地址。简而言之，要做一些市场营销的事情。良好的老式口头营销仍然是最佳的方式，甚至在 Internet 上也是如此，我们具有更多的工具可供我们在线使用。

增加从其他站点进入你的站点的链接数也是非常直截了当的，尽管这并不意味着它不需要做许多工作。如果有关于你的主题的专门的目录，无论是在线的还是印刷版本，都要确保在其中把你列出来。参与社交网络，包括实现针对你的服务或业务的 Facebook 和 Google+页面（如果合适的话）。创建一个 Twitter（推特）账户，用于广播新闻消息并与客户联系，同样，如果这适合于你的在线形象（online presence）的话。进入你的客户可能所在的空间，比如评论你感兴趣的特定主题的博客，并参与那些社区。这并不是说你应该找到一个关于你的主题或服务的论坛，并把指向你的站点的链接兜售给它的用户。担当你所给定领域的专家，并连同你自己的站点 URL 一起提供忠告和建议。在本章中我无法做太多的事在这方面给你提供帮助，你只有自己积极地参与实践。

我可以帮助你的主要事情是：确保搜索引擎可以正确地收集你的内容并为它们建立索引。一个合理的假设是：如果你的内容不在 Google 的数据库中，那你就有麻烦了。

**注意**：Mashable（www.mashable.com）是一个非常流行的、高流量并且很受人尊敬的站点（由于它的准确性和额外的价值），它提供了关于在社交网络空间里进行交互的提示，尤其适合于商业用户。

**By the Way**

搜索引擎基本上都是巨大的数据库，它们都尽可能地为 Internet 上的大量内容建立索引，包括视频和其他丰富的媒体。它们使用自动化处理来搜索站点，并且使用被称为机器人（robot）或蜘蛛（spider）的程序搜索有关内容的页面，以及构建数据库。在为内容建立了索引之后，搜索应用程序本身将使用高度先进的技术给页面划分等级，以确定当用户输入一个搜索项时首先将显示哪些内容，接着将显示哪些内容，然后又会显示哪些内容，等等。

当搜索引擎处理用户查询时，它将寻找包含用户正在寻找的关键词和短语的内容。但它并不是简单的匹配，比如“如果这个页面包含这个短语，就返回它作为结果”，因为内容是依据关键词和短语出现的频率和环境以及从其他站点带给它信誉的链接数划分等级的。本章将介绍几种方式，确保你的内容基于你所提供的内容和环境适当地出现在搜索引擎中。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 24.2 利用主要的搜索站点列出你的页面

如果希望用户找到你的页面，你绝对必须给每个主要的搜索站点提交一个请求，为你的页面建立索引。即使搜索引擎会自动为 Web 内容建立索引，这也是确保你的站点会存在于它们的站点上的最佳方式。其中每个这样的站点都具有一个表单，需要填写 URL 地址、站点的简短描述，并且在一些情况下还要填写应该与你的清单关联的关键词分类或列表。这些表单很容易填写，可以轻松地在一个小时内完成它们，并且利用剩下的时间在你可能找到的一两个专用目录中列出自己（怎样找到专用目录？当然是通过主要的搜索引擎!）。

### 在列出你的页面之前

但是，请等等。在你急匆匆地提交列出页面请求之前，请先阅读本章的余下内容。否则，你将遇到一个非常严重的问题，并且将已经失去了解决它的最好机会。

要明白我的意思，可以想象下面这个场景：你发布了一个页面，用于销售自动的蟑螂碾压机。我是一位被蟑螂问题困扰的 Internet 用户，并且我对杀虫剂过敏。我打开了我的笔记本电脑，掸掉键盘上的蟑螂，登录到我最喜爱的搜索站点，并且输入“蟑螂”作为搜索项。搜索引擎迅速给我展示了 10400000 个包含词语“蟑螂”的 Web 页面中的前 10 个页面的列表。你已经提交了你的列出页面请求，因此你知道你的页面将出现在该列表中的某个位置。

我提过我很富有吗？我提过有两只蟑螂在我的脚上交配吗？你甚至在我所在的地区提供了当日交货的服务。你希望你的页面是出现在列表上的第 3 条，还是第 8 542 条？好吧，现在你理解了问题。仅仅列出在搜索引擎中是不够的，你需要利用内容和工作来提升等级。

***Did you Know?***

**提示：**一些站点提供了一个表单以及几个小型的搜索引擎，其中前者可以自动把它自身提交给所有主要的搜索引擎。这些站点（比如 www.scrubtheweb.com、www.submitexpress.com 和 www.hypersubmit.com）都是尝试给你销售一种优质服务（也就是把你还列出在许多其他的目录和索引中）的流行的站点示例。依赖于你的目标受众，这些服务可能有价值，也可能没有价值，但是我强烈建议你应该直接转到本章前面列出的主要的搜索站点，并使用它们自己的表单来提交要列出的请求。这样，就可以确保准确地回答问题（它们在每个站点上稍有不同），并且你将准确地知道在每个搜索引擎中将如何列出你的站点。

即使利用主要的搜索引擎列出你的站点是容易且快速的，它也可能有点令人混淆：每个搜索引擎都使用不同的术语，来确定你应该单击什么位置来注册你的页面。下面的列表可能让你避免一些挫折，它包括一些可以免费包括你的站点的流行的搜索引擎的地址，以及你应该单击以进行注册的指示链接的准确措辞。

- Google：访问 https://www.google.com/webmasters/tools/submit-url，输入你的站点地址，然后输入页面上显示的潦草书写的验证文本，称为 CAPTCHA 或验证码

异步社区会员 Kensuke(18319066421) 专享 尊重版权

（completely automated public turing test to tell computers and humans apart，完全自动化的公共图灵测试，用于把计算机和人类区分开）。然后单击 Submit（“提交”）按钮，把你的站点添加到 Google 中。

- Bing（以及 Yahoo!额外提供的列表）：访问 http://www.bing.com/toolbox/submit-site-url 以登录进去，然后输入验证文本，输入你的站点地址，并单击 Submit URL（“提交 URL”）按钮。当提交到 Bing 时，将获得 Yahoo!额外提供的列表，因为如果转到 Yahoo 自己的 Submit Your Site（“提交你的站点”）页面（地址是 http://search.yahoo.com/info/submit.html），它将重定向到 Bing 的提交表单。

## 24.3 为搜索引擎提供线索

事实上，绝对不能做任何事情，来保证你的站点在任何主要的搜索引擎中都将出现在针对特定词语或短语的前 10 个搜索结果中（也就是说，除了从搜索站点购买广告空间外将别无他法）。毕竟，如果具有这样的保证，那么为什么希望出现在列表上第一位的其他任何人不能采用同样的方法呢？你能做的就是避免出现在列表末尾，并像任何其他人一样给自己提供一个排在第一位的好机会；这称为搜索引擎优化（search engine optimization，SEO），或者优化你的页面的内容和结构，使得搜索引擎将比其他页面更青睐你的页面。

每个搜索引擎都使用一种不同的方法来确定哪些页面很可能最有相关性，因此应该排在搜索结果列表的顶部。不过，你不必过分纠结于这些区别，因为它们都使用相同的基本标准的某种组合。下面的列表包括搜索引擎在尝试评估哪些页面最匹配一个或多个关键词时将会考虑的几乎所有的事情。

- 你的页面标题在页面的<title>标签中自然地包括了一些有用的关键词吗？
- 页面的正文内容自然地包括了用户将会搜索的关键词吗？
- 关键词是否出现在页面中的<meta />标签中的简洁描述里？
- 你的站点是否利用了诸如<section>、<article>、<header>和<nav>之类的语义元素，这使得甚至更容易将某种含义应用于你的内容？
- 在页面中的编号级标题中出现了重要的文本吗？
- 在图像文件的名称以及用于页面中的图像的 alt 文本中使用了良好的描述吗？
- Web 站点内有多少个其他的页面链接到这个页面？
- 其他的 Web 站点内有多少个其他的页面链接到这个页面？又有多少个其他的页面链接到那些页面？
- 用户有多少次从以前的搜索结果列表中选择这个页面？

显然，提升位置可以做的最重要的事情是：考虑你的目标受众最有可能输入的关键词。不要让自己关注普通的单个词语的搜索，比如“食物”，它们生成的列表通常是如此之长，以至于使之出现在顶部就像买彩票一样。要代之以重点关注不同寻常的词语以及两个或 3 个词语的组合，它们最有可能指示与你的主题的相关性（例如，用“南方的家常菜”代替简单的“食物”）。确保那些名词和短语在你的页面上出现多次，并且一定要把最重要的词语

异步社区会员 Kensuke(18319066421) 专享 尊重版权

放在<title>标签以及第一个标题或介绍性段落中。

**By the Way**

**注意：**一些急于求成的 Web 页面作者把数十个甚至上百个相同的词语重复放在他们的页面上，有时使用的是小号字体或者难以查看的颜色，只是为了在用户每次搜索那个词语时使搜索引擎把该页面定位于列表的顶部。这种做法被称为搜索引擎作弊（search engine spamming）。
不要尝试做这种事，所有主要的搜索引擎都知道这种做法，并且会立即从它们建立了作弊检测器的数据库中删除任何这样的页面，这是通过以一种可疑的模式重复相同的词语或一组词语来实现的。在页面内容中，使重要的搜索词语多次自然地出现仍然是比较好的（并且相当有利）。不过，要确保在正常的句子或短语中使用那些词语，这样作弊政策将不会招惹你。

在刚才列出的所有搜索引擎评估标准中，<meta />标签的使用可能是最不被理解的。一些人大肆吹捧<meta />标签，好像使用它们可以立即使你移到每个搜索列表的顶部。其他人则认为<meta />标签是无效的和无用的，而弃之不用。这些极端情况都是不正确的。

<meta />标签是一个可以放在任何文档的<head>部分中的通用标签，用于指定一些不属于<body>文本的关于页面的信息。大多数主要的搜索引擎都查看<meta />标签来获取页面的简要描述和一些关键词，以确定你的页面是关于什么的。例如，你的自动蟑螂碾压机的订购表单可能包括以下两个标签：

```
<meta name="description"
      content="Order the SuperSquish cockroach flattener." />
<meta name="keywords"
      content="cockroach,roaches,kill,squish,supersquish" />
```

**Watch Out!**

**警告：**总是要把<meta />标签放在<head>、<title>和</title>标签之后，但是要放在</head>封闭标签之前。

这个示例中的第一个标签确保搜索引擎具有关于页面的准确描述，用于展示在它的搜索结果列表上（如果它们是这样做的话，就会多次这样做）。无论何时在搜索查询中包括有你指定的任何关键字，如果搜索引擎在它们的算法中使用了这类标签，那么第二个<meta />标签可能稍微会提升你的页面在列表上的排位。

**Did you Know?**

**提示：**在不太可能的情况下，你根本不希望页面包括在搜索引擎数据库中，此时就可以把以下<meta />标签放在该页面的<head>部分中：

```
<meta name="robots"content="noindex,noindex" />
```

这导致一些搜索机器人会忽略页面。为了更健壮地阻挡窥视机器人的眼睛，可以要求你的 Web 服务器管理员在服务器的 robots.txt 文件中包括你的页面地址（他或她将知道这意味着什么以及如何执行该操作，如果不是这样，可以参考 www.robotstxt.org 上的方便的信息）。所有主要的搜索蜘蛛然后都必定会忽略你的页面。这可能适合于你不情愿通过公共搜索而使之可用的内部公司页面。

在你希望搜索引擎建立索引的任何页面中，总是要包括<meta />标签以及name="description"和 name="keywords"属性。这样做可能不会对你的页面在搜索列表中的排

异步社区会员 Kensuke(18319066421) 专享 尊重版权

位具有显著的影响，并且并非所有的搜索引擎都会寻找<meta />标签，但是它有益无害。不过，一定要谨慎行事并克制自己，审慎且自然地使用描述和关键词标签，不要在它们当中塞满你能想到的每一个词语。

下面举一个具体的示例，说明如何改进搜索引擎结果，考虑程序清单 24.1 中的页面。

**程序清单 24.1　在 Internet 站点搜索期间几乎可见性的页面**

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <title>Fractal Central</title>
  </head>

  <body>
    <header style="text-align:center">
      <img src="fractalaccent.gif" alt="a fractal" />
    </header>
    <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"><img src="orderform.gif"
         alt="Order Form" style="border-style:none" /></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>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.</li>
        <li>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.</li>
        <li>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.</li>
        <li>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.</li>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    </ol>
    <p>Think all these folks may be on to something?</p>
  </div>
  <div style="text-align:center">
    <a href="http://netletter.com/nonsense/">
       <img src="findout.gif" alt="Find Out More"
            style="border-style:none" /></a>
  </div>
 </body>
</html>
```

这个页面应该很容易找到，因为它涉及特定的主题，并且有一些不同寻常的技术性术语多次出现，而对这个主题感兴趣的用户很可能会搜索它们。不过，你可以做几件事，增加这个页面出现在搜索引擎结果列表上的较高位置的几率。

现在把程序清单 24.1 中的页面与程序清单 24.2 中对这个页面所做的更改做一下比较。两个页面看起来几乎完全相同，但是对于搜索机器人和搜索引擎来说，这两个页面看上去具有相当大的差别。下面的列表总结了页面中所做的更改，以及这些更改如何影响页面的索引。

- 向<title>标签和页面上的第一个标题中添加了重要的搜索项。原始的页面甚至在这两个关键位置都没有包括单词“fractal”。
- 添加了<meta />标签，利用描述和关键词给搜索引擎提供帮助。
- 提高了大字标题的价值。
- 使用语义标签通过其结构对内容应用额外的含义。
- 向第一个<img />标签中添加了非常具有描述性的 alt 属性。并非所有的搜索引擎都会读取 alt 文本并对其建立索引，但是有一些会这样做。
- 删除了括住技术性术语的引号（如"fractal"和"iterated"），因为一些搜索引擎会把"fractal"视作不同于 fractal 的单词。利用字符实体"替换引号，搜索机器人将会简单地忽略它。
- 向订购表单框中的文本中把关键字“fractal”添加了两次。

**程序清单 24.2 对程序清单 24.1 中的页面所做的改进**

```
<!DOCTYPE html>

<html 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>
    <header style="text-align:center">
      <img src="fractalaccent.gif" alt="Fractal Central: A
           Guide to Fractals, Chaos, and Complexity " />
    </header>
```

异步社区会员 Kensuke(18319066421) 专享 尊重版权

```
    <section>
    <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 <strong>Fractal
      Central</strong> online store <br />
      <a href="orderform.html"><img src="orderform.gif"
         alt="Order Form" style="border-style:none" /></a>
     </div>
      <header>
      <h1>A Comprehensive Guide to the<br />
      Art and Science of Chaos and Complexity</h1>
      </header>
      <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>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.</li>
        <li>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.</li>
        <li>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.</li>
        <li>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.</li>
      </ol>
      <p>Think all these folks may be on to something?</p>
    </section>
    <footer style="text-align:center">
      <a href="http://netletter.com/nonsense/">
          <img src="findout.gif" alt="Find Out More"
               style="border-style:none" /></a>
    </footer>
  </body>
</html>
```

用户在搜索关于分形和混沌的信息时，他们能够找到程序清单 24.2 中所示页面的次数与找到程序清单 24.1 中所示的页面相比会增加多少是不可能量化的，但是可以肯定的是这些更改可以改进页面对搜索引擎的可见性。通常，为搜索蜘蛛所做的改进可能使页面的主题

异步社区会员 Kensuke(18319066421) 专享 尊重版权

也更容易被人类认可和理解。这使得为搜索引擎优化页面获得了双赢的结果。

这些更改使这个站点的内容更有可能被适当地建立索引。除了具有良好的、建立了索引的内容之外，还要记住内容的质量，以及链接到你的站点的其他站点的数量，这也很重要。

## 24.4 关于搜索引擎优化的额外提示

我可以给你的关于搜索引擎优化的最重要的提示是：如果一家 SEO 公司给你承诺了特定的结果，就不要给它付费来执行你的 SEO 任务。如果一家公司承诺你的站点将是 Google 搜索中的第一个结果，那么一定要溜之大吉，并随身携带好支票簿——没有人能够承诺这一点，因为搜索算法具有如此多的变量，以至于在给定的一个星期的时间内顶部的结果可能会改变多次。这并不是说所有的 SEO 公司都是骗子。人们经常把一些执行 SEO 任务的合法的站点内容和架构顾问与发送不请自来的电子邮件的垃圾邮件发送者混为一谈，比如：

“亲爱的 google.com，我访问了您的 Web 站点，并且注意到您没有列出在大多数搜索引擎和目录中……”

在 Google 自己的针对网站管理员的指导原则中，把这封示例电子邮件用作一个示例，并且指出：“对那些关于搜索引擎的不请自来的电子邮件，就像那些宣称一夜见效的减肥药一样，您都需要持有相同的怀疑态度。”是的，确实有人给 Google 发送了一封垃圾电子邮件，关于如何提高它们在 Google 中的搜索排名。要了解来自 Google 的更好的建议，可以访问 http://www.google.com/webmasters/。

下面给出了你可以采取的一些额外的措施，可让你免费为搜索引擎优化自己的内容。

- 使用准确的页面标题。你的页面标题应该简短，但是具有描述性并且是独特的。不要尝试在页面标题中塞满关键词。
- 创建人性化的 URL，比如其中包含有用户可以轻松记住的单词的那些 URL。记住一个像 http://www.mycompany.com/products/super_widget.html 这样的 URL 比记住像 http://www.mycompany.com?c=p&id=4&id=49f8sd7345fea 这样的 URL 要容易许多，并且搜索引擎更容易以一种相关的方式对其建立索引。
- 创建反应了你的目录结构的 URL。这假定你首先具有一种目录结构，而这是你应该做到的。
- 如果可能，就要为导航使用文本，而不要使用图形元素。
- 如果内容具有多级深度，可以使用浏览路径记录（breadcrumb trail），使得用户可以找到他们回家的路。浏览路径记录还给搜索引擎提供了更多的词语来建立索引。例如，如果你正在查看一个与食物相关的 Web 站点的 Southern Cooking（南方烹饪）类别中的烤饼的食谱，用于这个特定页面的浏览路径记录可能看起来如下：
  Home > Southern Cooking > Recipes > Biscuits
- 在你的页面内容内，适当地使用语义元素和带有编号的各级标题（<h1>、<h2>、<h3>）。

除了为用户提供丰富且有用的内容之外，还应该遵循这些提示，提高站点在页面排名中的声望。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 24.5 小结

本章通过探讨如何给搜索引擎（如 Google 和 Bing）提供线索，介绍了一些极其重要的领域，使得用户可以更轻松地找到你的页面。你还看到了对一个非常合理的 Web 页面背后的 HTML 代码进行改进以使之对搜索引擎更友好的示例。最后，本章介绍了在总体上对站点的索引进行优化的几条额外提示。

表 24.1 列出了本章中介绍的标签和属性。

表 24.1 本章介绍的 HTML 标签和属性

| 标签/属性 | 作用 |
| --- | --- |
| <meta /> | 指示关于该文档的元信息（关于文档本身的信息）。最常用于添加页面描述和指定关键词。在文档<head>中使用 |
| 属性 | |
| name="name" | 可用于指定在 content 属性中包含关于文档的哪些类型的信息。例如，name="keywords"意味着用于页面的关键词位于 content 中 |
| content="value" | 在 name 中指定的信息的实际消息或值。例如，在 content 属性中可以查看关键词的列表 |

## 24.6 问与答

**问：在我的站点中具有许多页面。我需要在每个搜索站点上为每个页面填写单独的表单吗？**

**答：**不需要。如果你只提交主页（它大概会链接到所有其他的页面），搜索蜘蛛将爬过页面上的所有链接（以及链接的页面上的所有链接等），直到它们对站点上的所有页面都建立了索引为止。

**问：我提交了一个请求，利用搜索引擎列出我的站点，但是当我搜索我的页面时，它并没有出现—甚至当我输入我的公司的独特名称时也是如此。我能够做什么？**

**答：**大多数大型的搜索引擎都提供了一个表单，可以填写它以即时检查在它们的数据库中是否包括有一个特定的地址。如果发现未包括该地址，可以提交另一个请求表单。有时，在提交了请求之后，蜘蛛可能要花数天或数周的时间才能设法给你的页面建立索引。

**问：当我把关键词放在<meta />标签中时，我需要包括关于拼写和大小写的每种可能的变体吗？**

**答：**不用担心大小写，几乎所有的搜索都是全都以小写字母输入的。要包括任何明显的变体或常见的拼写错误作为单独的关键词。尽管这在概念上比较简单，在涉及操作<meta />标签时，还有比我在本章中能够介绍的更高级的策略可用。可以访问 http://en.wikipedia.org/wiki/Meta_element，了解关于这个标签的多个属性、如何使用它以及何时可能忽略它的良好信息。

**问：我听说可以使用<meta />标签，使页面每隔几秒钟或几分钟就自动重新加载它自**

异步社区会员 Kensuke(18319066421) 专享 尊重版权

**身。真是这样吗？**

**答：**是的，但是这样做没有意义，除非在 Web 服务器上设置了某种程序或脚本，提供关于页面的新信息。不过，出于有用性考虑，W3C 和用户一般都反对使用<meta />标签来刷新内容。

## 24.7 测验

本测验包含一些问题和练习，可帮助读者巩固本章所学的知识。在查看答案之前，要尝试尽量回答所有的问题。

### 24.7.1 问题

1. 如果你发布一个关于领养小狗的页面，怎样才能确保该页面可以被在所有主要的 Internet 搜索站点中输入“小狗”、“狗”和“领养”这几个词语的用户找到？
2. 假设你决定在 HTML 代码中把关键词粘贴几百次，并且在白色的背景上使用白色的字体，使得读者不会看到它们。搜索引擎蜘蛛会怎样处理这种情况？
3. 是把所有的内容存放在一个目录中更好，还是把它们组织进多个目录中更好？

### 24.7.2 答案

1. 确保“小狗”、“狗”和“领养”这几个词语都频繁出现在主页上（因为它们可能已经这样做了），并在“领养小狗”所在的行上命名页面标题。当你处于此位置时，在页面的<head>部分中放入以下<meta />标签：

```
<meta name="description"
content="dog adoption information and services" />
<meta name="keywords" content="puppy, dog, adoption" />
```

   在线发布你的页面，然后访问每个主要的搜索引擎（在本章前面列出了它们）的站点提交页面，填写站点提交表单。
2. 搜索引擎蜘蛛将会忽略重复出现的词语，并且可能把你列入黑名单而移出它们的索引，以及把你标记为垃圾信息发送者。
3. 当然是把内容组织进多个目录中更好。这使得内容更容易维护，还给你提供了一个机会，利用目录结构创建人类可读的、有意义的 URL，以及创建导航式浏览路径记录。

### 24.7.3 练习

- 你已经学完了本书。如果你具有一个站点，准备让全世界的人查看，就要检查内容和结构，以便进行可能最好的优化，然后把地址提交给所有主要的搜索引擎。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

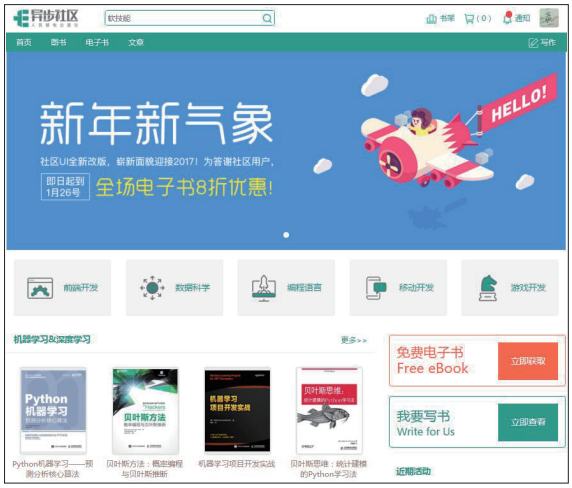
异步社区
人民邮电出版社
www.epubit.com.cn

# 欢迎来到异步社区！

## 异步社区的来历

异步社区（www.epubit.com.cn）是人民邮电出版社旗下 IT 专业图书旗舰社区，于 2015 年 8 月上线运营。

异步社区依托于人民邮电出版社 20 余年的 IT 专业优质出版资源和编辑策划团队，打造传统出版与电子出版和自出版结合、纸质书与电子书结合、传统印刷与 POD 按需印刷结合的出版平台，提供最新技术资讯，为作者和读者打造交流互动的平台。




## 社区里都有什么？

### 购买图书

我们出版的图书涵盖主流 IT 技术，在编程语言、Web 技术、数据科学等领域有众多经典畅销图书。社区现已上线图书 1000 余种，电子书 400 多种，部分新书实现纸书、电子书同步出版。我们还会定期发布新书书讯。

### 下载资源

社区内提供随书附赠的资源，如书中的案例或程序源代码。

另外，社区还提供了大量的免费电子书，只要注册成为社区用户就可以免费下载。

### 与作译者互动

很多图书的作译者已经入驻社区，您可以关注他们，咨询技术问题；可以阅读不断更新的技术文章，听作译者和编辑畅聊好书背后有趣的故事；还可以参与社区的作者访谈栏目，向您关注的作者提出采访题目。

## 灵活优惠的购书

您可以方便地下单购买纸质图书或电子图书，纸质图书直接从人民邮电出版社书库发货，电子书提供多种阅读格式。

对于重磅新书，社区提供预售和新书首发服务，用户可以第一时间买到心仪的新书。

用户账户中的积分可以用于购书优惠。100 积分 =1 元，购买图书时，在 0 使用积分 里填入可使用的积分数值，即可扣减相应金额。

异步社区会员 Kensuke(18319066421) 专享 尊重版权

## 特 别 优 惠

购买本书的读者专享**异步社区购书优惠券**。

使用方法：注册成为社区用户，在下单购书时输入 57AWG 使用优惠码，然后点击“使用优惠码”，即可享受电子书8折优惠（本优惠券只可使用一次）。

### 纸电图书组合购买

社区独家提供纸质图书和电子书组合购买方式，价格优惠，一次购买，多种阅读选择。




## 社区里还可以做什么？

### 提交勘误

您可以在图书页面下方提交勘误，每条勘误被确认后可以获得 100 积分。热心勘误的读者还有机会参与书稿的审校和翻译工作。

### 写作

社区提供基于 Markdown 的写作环境，喜欢写作的您可以在此一试身手，在社区里分享您的技术心得和读书体会，更可以体验自出版的乐趣，轻松实现出版的梦想。

如果成为社区认证作译者，还可以享受异步社区提供的作者专享特色服务。

### 会议活动早知道

您可以掌握 IT 圈的技术会议资讯，更有机会免费获赠大会门票。

## 加入异步

扫描任意二维码都能找到我们：

| 异步社区 | 微信服务号 | 微信订阅号 | 官方微博 | QQ 群：436746675 |
|---|---|---|---|---|

社区网址：www.epubit.com.cn

官方微信：异步社区

官方微博：@ 人邮异步社区，@ 人民邮电出版社 - 信息技术分社

投稿 & 咨询：contact@epubit.com.cn

异步社区会员 Kensuke(18319066421) 专享 尊重版权

# HTML与CSS

# 入门经典（第9版）

SAMS

Teach Yourself

## 学习如何：

- 创建自己的Web页面，并立即上线运行；
- 对文本进行格式化，使其具备最佳的清晰度和可读性；
- 创建指向其他页面和站点的链接；
- 在Web页面中添加图形、颜色以及其他效果；
- 使用透明图像和背景图像；
- 使用CSS设计站点布局和版式；
- 使用Web表单收集用户输入的信息；
- 推广你的站点，使其容易被搜索引擎找到；
- 使用不同的浏览器测试网站的兼容性；
- 使你的网站易于维护和升级；
- 使用HTML5语义标记更好地组织页面架构；
- 使用CSS创建栏；
- 在站点中添加基于HTML5的视频和音频；
- 使用CSS3变形、转换和动画创建极佳的视觉特效；
- 使用HTML5强大的表单验证功能。

## 24章阶梯教学

通过阅读本书，读者将学会如何使用HTML与CSS来设计、创建和维护专业水准的网站。本书采用直观、务实的写作方法，每章内容都以前面所学的章节为基础，环环相扣，即使读者没有任何经验，也可以通过本书完全掌握HTML与CSS的基本知识。

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。



读者可通过http://pan.baidu.com/s/1sjBEJMl下载本书所有的源代码。




ISBN 978-7-115-36529-3




分类建议：计算机／程序设计／Web开发
人民邮电出版社网址：www.ptpress.com.cn

美术编辑 王建国

异步社区会员 Kensuke(18319066421) 专享 尊重版权