

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

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




涵盖
HTML5与
CSS3

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

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/1sjBEJMI>下载本书所有的源代码。



人民邮电出版社 - 信息技术分社
<http://weibo.com/ptpitbooks>

ISBN 978-7-115-36529-3



9 787115 365293 >

ISBN 978-7-115-36529-3

定价：45.00 元

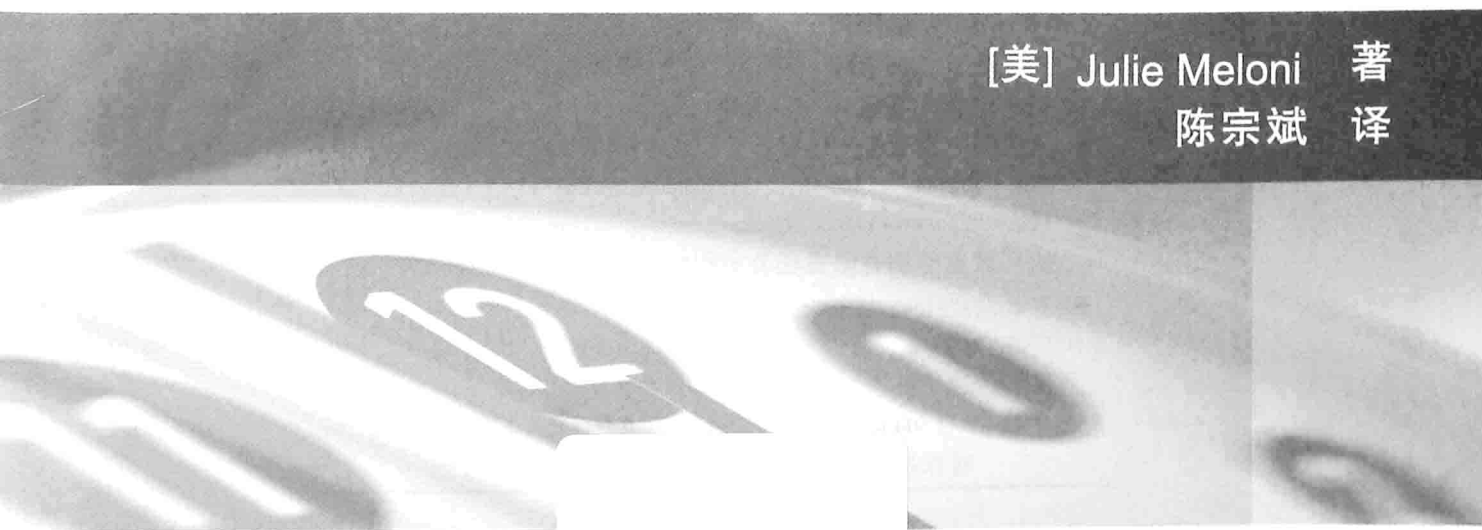
分类建议：计算机 / 程序设计 / Web开发
人民邮电出版社网址：www.ptpress.com.cn

美术编辑 王建国

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

图书在版编目 (C I P) 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. ①TP312
②TP393.092

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

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。

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*。

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，用于开发当今世界上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

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。

目 录

第 1 章 理解 Web 的工作方式	1	第 2 章 构造 HTML 文档	18
1.1 HTML 和 WWW 简史	2	2.1 准备工作	18
1.2 创建 Web 内容	2	2.2 从一个简单的 Web 页面 开始	19
1.3 理解 Web 内容递送	3	2.3 每个 Web 页面都必须 具有的 HTML 标签	22
1.4 选择 Web 托管提供商	4	2.4 利用段落和换行符组织 页面	23
1.5 利用多种 Web 浏览器执行 测试	6	2.5 利用标题组织内容	25
1.6 创建一个示例文件	7	2.6 理解语义元素	27
1.7 使用 FTP 传输文件	7	2.7 验证 Web 内容	28
1.7.1 选择 FTP 客户端	8	2.8 小结	30
1.7.2 使用 FTP 客户端	9	2.9 问与答	31
1.8 了解在 Web 服务器上放置 文件的位置	11	2.10 测验	31
1.8.1 基本的文件管理	11	2.10.1 问题	31
1.8.2 使用索引页面	12	2.10.2 答案	31
1.9 在不涉及 Web 服务器的 情况下分发内容	13	2.10.3 练习	32
1.9.1 在本地发布内容	13	第 3 章 理解层叠样式表	33
1.9.2 在博客上发布内容	14	3.1 CSS 的工作方式	34
1.10 关于测试 Web 内容的提示	14	3.2 基本的样式表	35
1.11 小结	15	3.3 CSS 样式的基础知识	39
1.12 问与答	15	3.3.1 基本的布局属性	39
1.13 测验	16	3.3.2 基本的格式化属性	40
1.13.1 问题	16	3.4 使用样式类	42
1.13.2 答案	17	3.5 使用样式 ID	44
1.13.3 练习	17		

3.6	内部样式表和内联样式	45	6.3	调整字体	79
3.7	小结	47	6.4	使用 Web 字体	82
3.8	问与答	47	6.5	小结	84
3.9	测验	48	6.6	问与答	85
3.9.1	问题	48	6.7	测验	85
3.9.2	答案	48	6.7.1	问题	85
3.9.3	练习	48	6.7.2	答案	86
第 4 章	密切观察 HTML5 页面结构	50	6.7.3	练习	86
4.1	概念化页面	50	第 7 章	处理颜色和边框	87
4.2	以多种方式使用<header>元素	54	7.1	选择颜色的最佳实践	87
4.3	理解<section>元素	55	7.2	理解 Web 颜色	88
4.4	使用<article>元素	56	7.3	使用十六进制的颜色值	90
4.5	实现<nav>元素	56	7.4	使用 CSS 设置背景、文本和边框颜色	91
4.6	何时使用<aside>元素	57	7.5	创建圆角	93
4.7	有效地使用<footer>元素	58	7.6	小结	95
4.8	小结	58	7.7	问与答	95
4.9	问与答	59	7.8	测验	96
4.10	测验	59	7.8.1	问题	96
4.10.1	问题	59	7.8.2	答案	96
4.10.2	答案	59	7.8.3	练习	96
4.10.3	练习	60	第 8 章	使用外部链接和内部链接	97
第 5 章	处理文本块和列表	61	8.1	使用 Web 地址	97
5.1	在页面上对齐文本	62	8.2	使用锚点在页面内链接	99
5.1.1	使用属性	62	8.2.1	利用锚点标识页面中的位置	100
5.1.2	对齐块级元素中的文本	62	8.2.2	链接到锚点位置	100
5.2	3 类 HTML 列表	64	8.3	在自己的 Web 内容之间创建链接	102
5.3	在列表内放置列表	66	8.4	链接到外部 Web 内容	105
5.4	小结	70	8.5	链接到电子邮件地址	106
5.5	问与答	71	8.6	在新的浏览器窗口中打开链接	107
5.6	测验	71	8.7	使用 CSS 编排超链接的样式	107
5.6.1	问题	71	8.8	小结	110
5.6.2	答案	71	8.9	问与答	111
5.6.3	练习	72	8.10	测验	112
第 6 章	处理字体	73	8.10.1	问题	112
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			

8.10.2 答案	112	11.4.2 垂直图像对齐	148
8.10.3 练习	112	11.5 把图像转变成链接	149
第 9 章 使用表格和栏	114	11.6 使用背景图像	152
9.1 创建一个简单的表格	114	11.7 使用图像映射	153
9.2 控制表格大小	118	11.7.1 为什么图像映射并非总是必要的	153
9.3 表格内的对齐和跨越	120	11.7.2 映射图像内的区域	154
9.4 利用表格进行页面布局	123	11.7.3 为图像映射创建 HTML 代码	155
9.5 使用 CSS 栏	123	11.8 小结	157
9.6 小结	127	11.9 问与答	158
9.7 问与答	127	11.10 测验	158
9.8 测验	127	11.10.1 问题	159
9.8.1 问题	128	11.10.2 答案	159
9.8.2 答案	128	11.10.3 练习	159
9.8.3 练习	128		
第 10 章 创建用于 Web 的图像	129	第 12 章 在 Web 站点中使用多媒体	160
10.1 选择图形软件	129	12.1 链接到多媒体文件	161
10.2 关于图形需要知道的最少知识	130	12.2 嵌入多媒体文件	163
10.3 准备照片图像	131	12.3 为音频和视频播放使用纯 HTML5 代码	165
10.3.1 裁剪图像	132	12.4 关于使用多媒体的额外提示	167
10.3.2 调整图像大小	133	12.5 小结	168
10.3.3 调整图像颜色	134	12.6 问与答	169
10.3.4 控制 JPEG 压缩	135	12.7 测验	170
10.4 创建横幅和按钮	135	12.7.1 问题	170
10.5 减少或删除图像中的颜色	137	12.7.2 答案	170
10.6 创建拼贴式背景图像	137	12.7.3 练习	170
10.7 创建动画式 Web 图形	139		
10.8 小结	139	第 13 章 处理边距、填充、对齐和浮动	171
10.9 问与答	140	13.1 使用边距	171
10.10 测验	140	13.2 填充元素	177
10.10.1 问题	140	13.3 使所有的一切保持对齐	180
10.10.2 答案	141	13.4 理解 float 属性	181
10.10.3 练习	141	13.5 小结	183
第 11 章 在 Web 站点中使用图像	142	13.6 问与答	183
11.1 在 Web 页面上放置图像	142	13.7 测验	184
11.2 利用文本描述图像	145	13.7.1 问题	184
11.3 指定图像的高度和宽度	145	13.7.2 答案	184
11.4 对齐图像	146	13.7.3 练习	184
11.4.1 水平图像对齐	146		

第 14 章 理解 CSS 方框模型和定位	185	16.7 测验	219
14.1 CSS 方框模型	185	16.7.1 问题	220
14.2 关于定位的独家新闻	188	16.7.2 答案	220
14.3 控制元素的堆叠方式	191	16.7.3 练习	220
14.4 管理文本流	193	第 17 章 使用 CSS 设计导航系统	221
14.5 小结	194	17.1 导航列表与常规的列表	
14.6 问与答	194	有何区域	221
14.7 测验	194	17.2 利用 CSS 创建垂直导航	
14.7.1 问题	194	系统	222
14.7.2 答案	195	17.2.1 编排单级垂直导航	
14.7.3 练习	195	系统的样式	224
第 15 章 创建固定布局或流动布局	196	17.2.2 编排多级垂直导航	
15.1 理解固定布局	197	系统的样式	226
15.2 理解流动布局	198	17.3 利用 CSS 创建水平导航	
15.3 创建固定/流动混合布局	200	系统	229
15.3.1 从基本的布局结构		17.4 小结	232
开始	200	17.5 问与答	232
15.3.2 在固定/流动混合布局		17.6 测验	233
中定义两栏	201	17.6.1 问题	233
15.3.3 设置布局的最小		17.6.2 答案	233
宽度	203	17.6.3 练习	233
15.3.4 在固定/流动混合		第 18 章 使用鼠标动作修改文本	
布局中处理栏高度	204	显示	234
15.4 考虑响应性 Web 设计	208	18.1 利用 CSS 创建工具提示	234
15.5 小结	209	18.2 利用 CSS 显示额外的	
15.6 问与答	209	翻转文本	237
15.7 测验	209	18.3 访问事件	238
15.7.1 问题	209	18.4 使用 onclick 更改<div>	
15.7.2 答案	210	的外观	239
15.7.3 练习	210	18.5 小结	244
第 16 章 使用 CSS 进一步处理列表	211	18.6 问与答	244
16.1 HTML 列表复习	211	18.7 测验	245
16.2 CSS 方框模型如何影响		18.7.1 问题	245
列表	212	18.7.2 答案	245
16.3 放置列表项指示符	215	18.7.3 练习	245
16.4 利用列表项和 CSS 创建		第 19 章 实现 CSS3 变形、渐变	
图像映射	216	和动画	246
16.5 小结	219	19.1 考虑浏览器的差别	246
16.6 问与答	219	19.2 使用 2D 变形	247

19.2.1 平移元素	247	21.6 提前考虑开发 HTML5 应用程序.....	282
19.2.2 旋转元素	249	21.7 小结.....	283
19.2.3 缩放元素	250	21.8 问与答.....	283
19.2.4 扭曲元素	252	21.9 测验.....	284
19.3 使用 3D 变形	253	21.9.1 问题.....	284
19.3.1 应用透视图	253	21.9.2 答案.....	284
19.3.2 在 3D 空间中移动和 旋转元素.....	255	21.9.3 练习.....	285
19.3.3 创建和操纵立方体.....	255	第 22 章 处理基于 Web 的表单.....	286
19.4 实现 CSS3 渐变	259	22.1 HTML 表单是如何工作的	286
19.5 初识 CSS3 动画	260	22.2 创建表单.....	287
19.6 小结.....	261	22.3 接受文本输入	291
19.7 问与答.....	261	22.4 命名每一份表单数据	292
19.8 测验.....	262	22.4.1 标记每一份表单 数据	292
19.8.1 问题.....	262	22.4.2 组合表单元素	293
19.8.2 答案.....	262	22.4.3 在表单中包括隐藏的 数据	294
19.8.3 练习.....	262	22.5 探讨表单输入控件	294
第 20 章 创建打印友好的 Web 页面.....	263	22.5.1 复选框.....	294
20.1 是什么使页面成为打印 友好的.....	264	22.5.2 单选按钮.....	296
20.2 应用特定于媒体的样式表.....	266	22.5.3 选择列表	296
20.3 设计用于打印页面的 样式表.....	267	22.5.4 文本框、文本区和 其他输入类型.....	297
20.4 在 Print Preview (打印 预览) 中查看 Web 页面.....	270	22.6 使用 HTML5 表单验证.....	298
20.5 小结.....	272	22.7 提交表单数据.....	300
20.6 问与答.....	272	22.8 小结.....	301
20.7 测验.....	272	22.9 问与答.....	302
20.7.1 问题.....	272	22.10 测验.....	302
20.7.2 答案.....	273	22.10.1 问题.....	302
20.7.3 练习.....	273	22.10.2 答案.....	303
第 21 章 理解动态 Web 站点和 HTML5 应用程序.....	274	22.10.3 练习.....	303
21.1 理解不同类型的脚本.....	274	第 23 章 组织和管理 Web 站点.....	304
21.2 在 HTML 中包括 JavaScript 代码.....	275	23.1 当一个页面就足够时	305
21.3 显示随机内容	277	23.2 组织简单的站点	306
21.4 理解文档对象模型	280	23.3 组织更大的站点	308
21.5 基于用户交互更改图像.....	281	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

23.5	考虑版本控制	313	24.2	利用主要的搜索站点列出 你的页面	320
23.6	使用 HTML 和 CSS 框架	314	24.3	为搜索引擎提供线索	321
23.7	小结	315	24.4	关于搜索引擎优化的 额外提示	326
23.8	问与答	315	24.5	小结	327
23.9	测验	316	24.6	问与答	327
23.9.1	问题	316	24.7	测验	328
23.9.2	答案	316	24.7.1	问题	328
23.9.3	练习	316	24.7.2	答案	328
第 24 章	帮助人们找到你的 Web 页面	318	24.7.3	练习	328
24.1	宣传你的 Web 站点	318			

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

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

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。

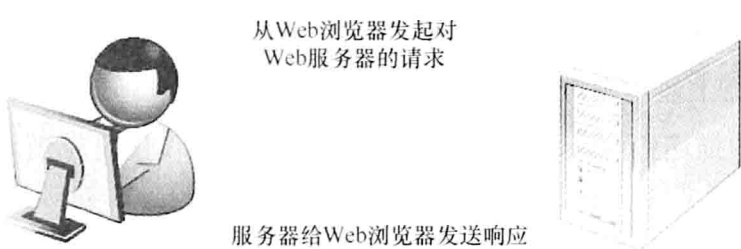


图 1.1
浏览器请求和服务
器响应

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

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。

```

```

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 `` 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

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**

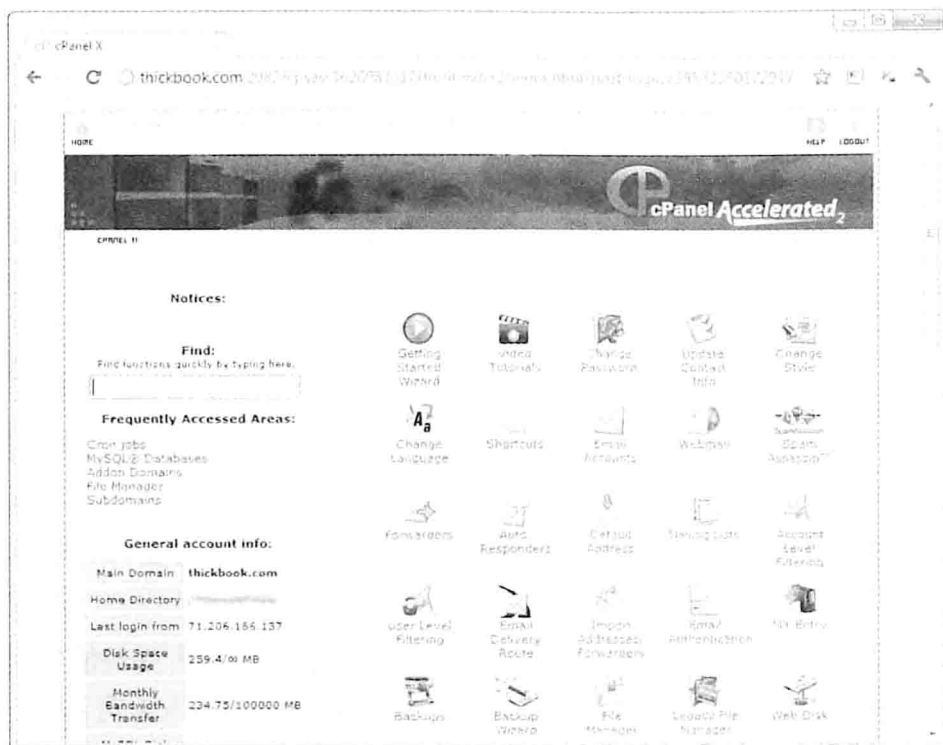
良好的托管提供商的一个特点是, 它会提供一个“控制面板”, 让你管理自己账户的各个方面。图 1.3 显示了作者自己在 Daily Razor 上的托管账户的控制面板。许多 Web 托管提供商都提供了这种特殊的控制面板软件, 或者某种在设计上类似的控制面板——清楚标记的图标引导你可以执行的任务, 用以配置和管理你的账户。

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

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。

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

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。

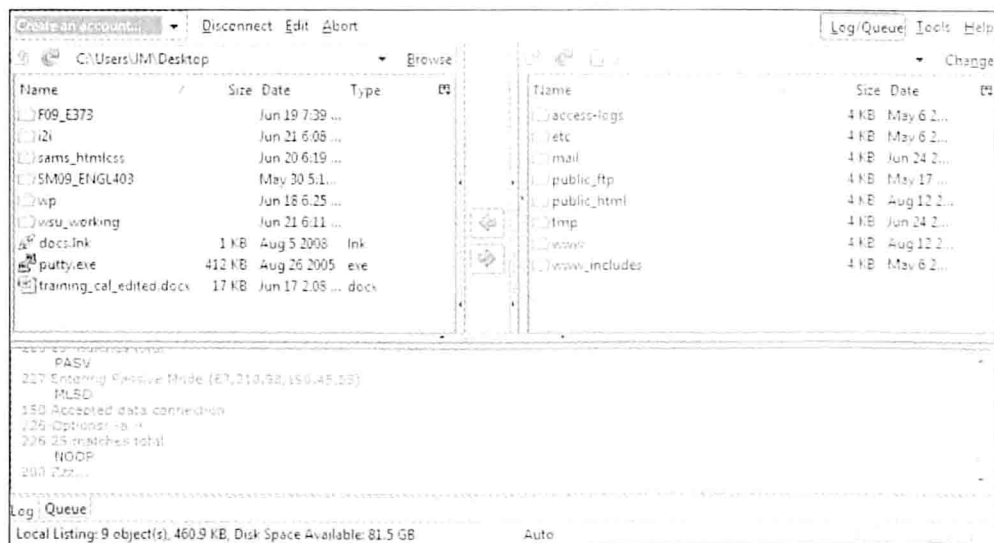
当你具有了这些信息后，就准备好使用 FTP 客户端，把内容传输到 Web 服务器。

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。

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。

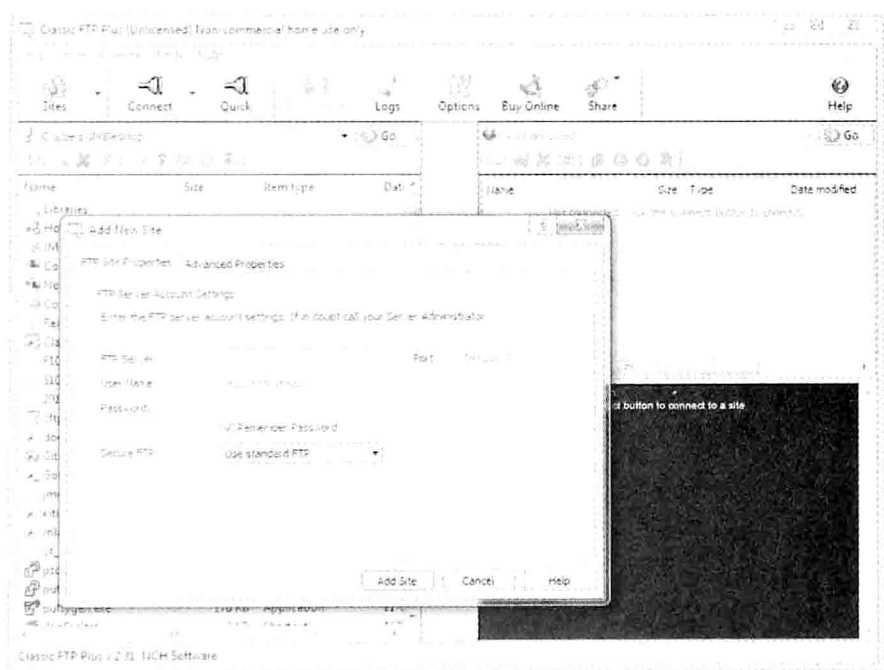


图 1.5

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。

6. 目标是你以前创建的 `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。

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，

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

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

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

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？

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机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。

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

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

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.
```

```

</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。

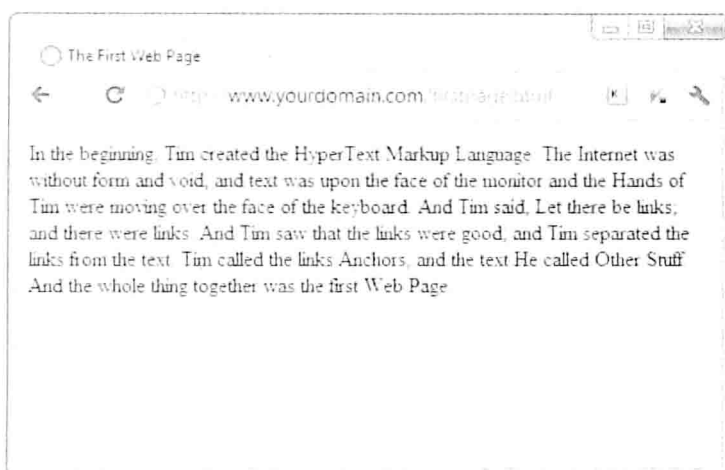


图 2.1

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

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