

PROGRAMMER TO PROGRAMMER™



Beginning Web Programming with HTML,
XHTML, and CSS, 2nd Edition

Web编程入门经典

——HTML、XHTML和CSS

(第2版)



(美) Jon Duckett 著
杜静 敖富江 译



清华大学出版社

Web 编程入门经典

——HTML、XHTML 和 CSS

(第 2 版)

(美) Jon Duckett 著
杜静 敖富江 译

TP3P3.0P2

D008

清华大学出版社

北 京

Jon Duckett

Beginning Web Programming with HTML, XHTML, and CSS, 2nd Edition

EISBN: 978-0-470-25931-3

Copyright © 2008 by Wiley Publishing, Inc.

All Rights Reserved. This translation published under license.

本书中文简体字版由 Wiley Publishing, Inc. 授权清华大学出版社出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2009-2971

本书封面贴有 Wiley 公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

Web 编程入门经典——HTML、XHTML 和 CSS(第 2 版)/(美) 达科特(Duckett, J.)著; 杜静, 敖富江 译.
—北京: 清华大学出版社, 2010.1

书名原文: Beginning Web Programming with HTML, XHTML, and CSS, 2nd Edition

ISBN 978-7-302-21597-4

I. ①W… II. ①达… ②杜… ③敖… III. ①超文本标记语言, HTML、XHTML—主页制作—程序设计 ②主页制作—软件工具, CSS IV. ①TP312②TP393.092

中国版本图书馆 CIP 数据核字(2009)第 226811 号

责任编辑: 王 军 于 平

装帧设计: 孔祥丰

责任校对: 成凤进

责任印制: 孟凡玉

出版发行: 清华大学出版社

地 址: 北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编: 100084

社 总 机: 010-62770175

邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者: 北京密云胶印厂

装 订 者: 三河市新茂装订有限公司

经 销: 全国新华书店

开 本: 185×260 印 张: 40.75 字 数: 992 千字

版 次: 2010 年 1 月第 1 版 印 次: 2010 年 1 月第 1 次印刷

印 数: 1~4000

定 价: 79.80 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770177 转 3103 产品编号: 030084-01

前言

目前关于设计和构建 Web 页面的书籍很多,因此首先感谢您选择阅读本书。本书与其他书籍的不同点是什么呢? Web 概念已经出现了十多年,在这期间引入了大量用于创建 Web 页面的技术,其中一些技术目前还在使用,而其他技术已经消失。许多介绍如何编写 Web 页面的书籍是相同书籍早期版本的修订,因此仍然采用与以前版本相同的介绍方法。而本书的目的是介绍如何为当今以及将来的 Web 创建页面。因此,读者阅读完本书之后,仍可将其作为一本有用的参考书放在身边,在需要时随时翻阅。

曾经有一段时间,编写 Web 页面的程序员仅需要掌握一种编程语言,即 HTML 语言。但是随着 Web 技术的发展,为了创建有效并吸引人的 Web 页面,程序员需要学习更多的技术,需要掌握多种不同语言,主要包括:

- **HTML 和 XHTML:** HTML 和 XHTML 用于解释 Web 页面的结构。它们用于指明哪些文本作为题头,段落的起始位置和结束位置在何处,哪些图像应当出现在文档中,以及指定不同页面之间的链接。不应当将 HTML 和 XHTML 看作是两种独立的语言。相反,可以将 XHTML 看作是与 HTML 的最新版本非常相似的语言。
- **CSS:** CSS 用于控制文档的外观。例如,可以使用 CSS 指定字型应该是较大的、粗体的 Arial 字型,或者指定页面的背景应该是亮绿色。另外,还可以使用 CSS 控制不同项在页面上的位置。例如,利用 CSS 将文本放置在同一个页面的两列中。
- **JavaScript:** 利用 JavaScript 可以在创建的 Web 页面上添加交互性,并且可以操作显示 Web 页面的浏览器。

尽管事实上需要掌握多种语言(而不只是 HTML),但是可以将 HTML 看作熟悉 Web 的好机会,因为 HTML 中用于创建 Web 页面的许多技术拥有成熟的、有利的方法;或者将其看作“最佳实践”,因为可以利用它创建完整的 Web 站点。

本书简介

本书将介绍如何利用 HTML 和 XHTML 控制 Web 页面的结构,如何利用 CSS 赋予 Web 页面样式,如何利用 JavaScript 添加交互性。但是,只是学习最新的技术并不能确保编写出优秀的 Web 页面。由于用于编写 Web 页面的技术不断改进,因此浏览器(用于访问 Web 的程序和设备)也在改进。浏览器反映——并且有时甚至是通知——用于创建 Web 页面的语言的发展方向。问题在于并不是每个人都在其计算机上安装了最新软件,因此人们不仅希望编写出的 Web 页面能够充分利用浏览器的最新功能,而且希望确保 Web 页面能够在当今仍然流行的较老浏览器上正确显示。

因为 Web 页面的构建方式不断改变,并且存在多种不同版本的 Web 浏览器,所以本书中列举的某些功能被标记为“逐渐淘汰”,这意味着虽然该功能仍然能够工作于现代的浏

览器中，但是不再建议使用它，因为软件可能不会始终支持它。

在编写 Web 页面时需要了解的另一个问题是，能够访问 Web 的设备正在不断增加，例如移动电话、PDA(个人数字助理)和电视机顶盒。这些设备通常利用相同的语言，本书中将介绍这些语言——通过学习结合使用 XHTML 和 CSS，相比于仅采用旧的 HTML 编写的 Web 站点，您将能够创建出生命力更持久的 Web 站点。

最近几年来，Web 技术改变的另外一个领域是越来越强调可用性和可访问性。可用性是指使 Web 站点易于被用户浏览，并使用户获得他们到达您的站点所希望查找的内容；而可访问性是指使尽可能多的用户能够访问 Web 站点，特别是一些具有行为障碍的人(他们可能具有受损的视力或者无法使用鼠标)。在世界上的许多国家，如果站点不能满足严格的访问指导原则，政府不会同意构建该 Web 站点。在构建 Web 站点之前仔细思考，将意味着视力受损的人们能够查看具有较大文本的站点，或者通过使用屏幕阅读器阅读站点。有些书籍专门讨论可用性和可访问性主题，它们的目标是需要学习如何使代码具有更好的可访问性和可用性的 Web 开发人员，但本书的目标是使开发人员从一开始就牢记这些原则。

阅读完本书之后，读者编写的 Web 页面将不仅能够使用最新的技术，而且仍然能够通过较老的浏览器查看。外观优美的页面仍然能够被视力受损或身体受损的人们访问。这些页面不仅能够满足当今访问者的需求，也能够工作于正在出现的技术。并且，读者掌握的技巧将会在相当长的一段时间内有效。

本书读者对象

本书适用于希望学习如何创建 Web 页面的读者；也适用于已经从事过 Web 页面的编写(或许使用过某种 Web 页面制作工具)，但是希望真正理解 Web 编程语言以便创建更优秀页面的读者。

有经验的 Web 开发人员也能够从本书受益，因为本书介绍了某些最新的技术，例如 XHTML，并且鼓励读者接受某些 Web 标准，这些标准不仅满足能够访问 Web 的新设备的需求，而且能够使 Web 站点获得更多的访问者。

本书的读者不需要具有任何编程经验，这是一本编程的入门书籍。无论读者是编程爱好者，还是希望从事 Web 编程职业，本书都将教授 Web 编程的基础知识。当然，术语“程序员”通常与“低层次”联系在一起，但是阅读完本书之后，即使希望被称作为 Web 设计人员，也需要知道如何编写代码以便创建出完善的 Web 站点。

本书主要内容

阅读完本书之后，读者将能够创建出具有专业外观并且代码优美的 Web 页面。

读者不仅将学习如何编写由标记语言(例如 XHTML)组成的代码，而且将看到如何应用这些代码创建具有复杂布局的页面、定位文本和图像在页面中的出现位置，以及获得所需的颜色和字体。在这个过程中，您将看到如何使页面能够易于使用，并且获得尽可能多的访问者。本书还将介绍一些实用技术，例如如何使 Web 站点能够在 Internet 上被访问，如

何使搜索引擎识别 Web 站点。

本书介绍的主要技术包括 HTML、XHTML 和 CSS，并且介绍关于 JavaScript 的基础知识，这些知识足以用于在一些示例页面中添加交互性，并使读者能够处理一些基本脚本。在这个过程中，本书还向读者介绍将来可能希望学习的一些其他技术。

本书鼓励读者编写的代码遵循 Web 标准；HTML、XHTML 和 CSS 均由 World Wide Web 联盟创建和维护，World Wide Web 联盟也简称为 W3C(网址为 www.w3.org/)，这是一个专门致力于创建 Web 规范的组织。本书也介绍某些非标准的功能，当读者遇到这样的标记并需要知道它的作用时，了解这些功能是非常有帮助的(遇到这种情况时，将专门指明这些功能不是标准的组成部分)。

使用本书的要求

使用本书时，读者需要的全部条件包括：一台安装了 Web 浏览器(最好是 Firefox 2 或更高版本、Safari 2 或更高版本、Internet Explorer 6 或更高版本)的计算机以及一个简单的文本编辑器(例如 Windows 的 Notepad 或 Mac 的 TextEdit)。

如果读者具有一个 Web 页面编辑器程序，例如 Macromedia Dreamweaver 或 Microsoft FrontPage，也可以使用该程序，但本书不会讲解如何使用这些程序。这些程序各有不同，并且已经有介绍它们的专门书籍。即使具有这样的程序，但在真正理解这些程序生成的代码之后，则能够编写出更好的站点，因为经常需要手动编写和编辑这些代码。

本书组织结构

本书第 1 章将介绍如下内容：创建 Web 站点的主要任务是利用其构成元素和属性标记出现在站点上的文本。元素和属性描述了文档的结构(哪些是题头，哪些是文本的段落，哪些是链接等)。

本书前 6 章描述了组成 HTML 和 XHTML 的不同元素和属性，以及如何使用它们编写 Web 页面。以任务相关的方式将这几章组织成不同的领域，例如将一个文档结构化为题头和段落，创建页面之间的链接，添加颜色和图像，显示表等。在介绍每一个任务和主题时，本书都将首先给出示例，向读者演示可以实现的功能；然后更详细地介绍使用的元素和属性。

需要提醒的是，读者在第一次阅读本书时不需要阅读元素每个方面的详细解释——只需要了解该元素可以实现的功能即可。出于完整性(并将相关信息保持在相同位置)的原因，本书中介绍了一些可能会很少用到的功能。当发现需要使用某些比较难以理解的功能时，可以返回来重新仔细阅读它们。因此，如果希望更快地阅读本书，只需要记住标识的要点，然后继续下去。

每一章的结束部分给出了一些练习，以方便读者温习所学的概念。如果需要返回并重新阅读该章的内容以完成练习，不要为此感到担心。本书的目的是为读者在未来的几年中提供有用的参考，因此不要感觉需要用心学习所有内容。在阅读本书的过程中，读者将看到每个元素能够被哪些浏览器支持，并将学习到大量有用的提示和技巧以及创建专业 Web

页面的技术。

学习如何利用 HTML 和 XHTML 创建和结构化文档之后,读者接下来将学习如何使用层叠样式表(Cascading Style Sheet, CSS)创建更具吸引力的 Web 页面,并且学习如何改变所使用字体的字型和大小、文本的颜色、各项的背景和周围的边框,以及如何将对象对齐在页面的中间、左边或右边等知识。

学习完这两章(第 8 章和第 9 章)之后,利用本书中的示例将能够编写非常复杂的 Web 页面。这些章节的内容可以作为有用的参考,在将来可以重新查阅它们,其中的示例可以作为构建自己的 Web 站点的工具箱。

第 9 章和第 10 章着重介绍 Web 页面设计问题。读者将看到一些流行的 Web 页面布局示例以及构造它们的方式,并且学习如何创建优秀的导航栏,以使用户在您的站点找到所需的页面;您将发现使表单有效的各个方面;并且将学习如何使 Web 站点能够被尽可能多的人访问。这两章实际上基于本书前半部分中介绍的理论,有助于创建具备专业外观的页面,以便吸引用户并使站点更易于使用。

第 11 章和第 12 章介绍 JavaScript,这是一种称为脚本语言的编程语言,可将其用于 Web 页面中。虽然完整的 JavaScript 语言非常庞大,无法在这两章中全面介绍它,但是读者应当能够了解它的工作方式,并了解如何将脚本集成到页面中。

第 13 章(也就是最后一章)将介绍如何为将站点发布到 Internet 上做准备,并介绍 Web 主机托管、FTP 和代码验证方面的知识。最后,该章给出阅读完本书后可以继续执行什么操作的概念;您可能希望在 Web 站点上添加大量其他内容,或者希望通过学习进一步提高 Web 编程技能,本章将给出其他可以实现的功能以及需要学习的相关知识。

本书包含了几个有用的附录,包括 XHTML 元素和 CSS 特性的参考。其中一个附录解释了 XHTML 和 CSS 如何指定颜色。其他附录分别介绍可利用的字符编码、语言代码以及一些能够用于 HTML、XHTML、CSS 和 JavaScript 的转义字符。最后一个附录列出了一些较老的标记,实际上不应当再使用这些标记,这里列举它们只是为了防止读者在创建站点时遇到一些较老的 Web 技术。

源代码

在读者学习本书中的示例时,可以手工输入所有的代码,也可以使用本书附带的源代码文件。本书使用的所有源代码都可以从本书合作站点 <http://www.wrox.com/> 或 www.tupwk.com.cn/download 上下载。登录到站点 <http://www.wrox.com/>, 使用 Search 工具或使用书名列表可以找到本书。接着单击本书细目页面上的 Download Code 链接,可以获得所有的源代码。

注释:

由于许多图书的标题都很类似,所以按 ISBN 搜索是最简单的,本书英文版的 ISBN 是 978-0-470-25931-3。

在下载了代码后,只需用自己喜欢的解压缩软件对它进行解压缩即可。另外,也可以

进入 <http://www.wrox.com/dynamic/books/download.aspx> 上的 Wrox 代码下载主页, 查看本书和其他 Wrox 图书的所有代码。

勘误表

尽管我们已经尽了各种努力来保证文章或代码中不出现错误, 但是错误总是难免的, 如果您在本书中找到了错误, 例如拼写错误或代码错误, 请告诉我们, 我们将非常感激。通过勘误表, 可以让其他读者避免受挫, 当然, 这还有助于提供更高质量的信息。

请给 wkservice@vip.163.com 发电子邮件, 我们会检查您的反馈信息, 如果是正确的, 我们将在本书的后续版本中采用。

要在网站上找到本书英文版的勘误表, 可以登录 <http://www.wrox.com>, 通过 Search 工具或书名列表查找本书, 然后在本书的细目页面上, 单击 Book Errata 链接。在这个页面上可以查看到 Wrox 编辑已提交和粘贴的所有勘误项。完整的图书列表还包括每本书的勘误表, 网址是 www.wrox.com/misc-pages/booklist.shtml。

P2P.WROX.COM

要与作者和同行讨论, 请加入 p2p.wrox.com 上的 P2P 论坛。这个论坛是一个基于 Web 的系统, 便于您张贴与 Wrox 图书相关的消息和相关技术, 与其他读者和技术用户交流心得。该论坛提供了订阅功能, 当论坛上有新的消息时, 它可以给您传送感兴趣的论题。Wrox 作者、编辑和其他业界专家和读者都会到这个论坛上来探讨问题。

在 <http://p2p.wrox.com> 上, 有许多不同的论坛, 它们不仅有助于阅读本书, 还有助于开发自己的应用程序。要加入论坛, 可以遵循下面的步骤:

- (1) 进入 p2p.wrox.com, 单击 Register 链接。
- (2) 阅读使用协议, 并单击 Agree 按钮。
- (3) 填写加入该论坛所需要的信息和自己希望提供的其他信息, 单击 Submit 按钮。
- (4) 您会收到一封电子邮件, 其中的信息描述了如何验证账户, 完成加入过程。

注释:

不加入 P2P 也可以阅读论坛上的消息, 但要张贴自己的消息, 则必须加入该论坛。

加入论坛后, 可以张贴新消息, 响应其他用户张贴的消息。可以随时在 Web 上阅读消息。如果要想让该网站给自己发送特定论坛中的消息, 可以单击论坛列表中该论坛名旁边的 Subscribe to this Forum 图标。

关于使用 Wrox P2P 的更多信息, 可阅读 P2P FAQ, 了解论坛软件的工作情况以及 P2P 和 Wrox 图书的许多常见问题。要阅读 FAQ, 可以在任意 P2P 页面上单击 FAQ 链接。

目 录

第 1 章 创建结构化文档.....1	
1.1 结构化文档组成的 Web.....1	
1.2 XHTML 简介.....2	
1.3 核心元素和属性.....8	
1.3.1 <html>元素.....8	
1.3.2 <head>元素.....9	
1.3.3 <title>元素.....9	
1.3.4 <body>元素.....10	
1.4 属性组.....10	
1.4.1 核心属性.....11	
1.4.2 国际化属性.....12	
1.4.3 UI 事件.....14	
1.5 基本文本格式.....14	
1.5.1 空格和流.....15	
1.5.2 使用 hn 元素创建题头.....16	
1.5.3 使用<p>元素创建段落.....18	
1.5.4 使用 元素创建换行.....19	
1.5.5 使用<pre>元素创建预先 格式化的文本.....20	
1.6 表现元素.....23	
1.6.1 元素.....24	
1.6.2 <i>元素.....24	
1.6.3 <u>元素(逐渐淘汰).....24	
1.6.4 <s>元素和<strike>元素 (逐渐淘汰).....24	
1.6.5 <tt>元素.....25	
1.6.6 <sup>元素.....25	
1.6.7 <sub>元素.....25	
1.6.8 <big>元素.....25	
1.6.9 <small>元素.....26	
1.6.10 <hr />元素.....26	
1.7 短语元素.....26	
1.7.1 元素添加强调.....27	
1.7.2 元素添加着重强调.....27	
1.7.3 用于缩写词的<abbr>元素.....28	
1.7.4 用于首字母缩写词 的<acronym>元素.....28	
1.7.5 用于特殊术语的<dfn>元素.....29	
1.7.6 用于引用文本的 <blockquote>元素.....29	
1.7.7 用于短引用的<q>元素.....30	
1.7.8 用于引证的<cite>元素.....31	
1.7.9 用于代码的<code>元素.....31	
1.7.10 用于通过键盘输入的 文本的<kbd>元素.....31	
1.7.11 用于编程变量的 <var>元素.....32	
1.7.12 用于程序输出的 <samp>元素.....32	
1.7.13 用于地址的<address>元素.....32	
1.8 列表.....33	
1.8.1 利用 元素创建无序列表.....33	
1.8.2 有序列表.....34	
1.8.3 定义列表.....36	
1.8.4 列表的嵌套.....37	
1.9 编辑文本.....40	
1.9.1 使用<ins>元素指示 新添加的文本.....41	
1.9.2 使用元素指示 删除的文本.....42	
1.10 利用字符实体表示 特殊字符.....42	
1.11 注释.....43	
1.12 元素(逐渐淘汰).....43	
1.13 理解块级元素和内联元素.....44	

1.14 利用<div>元素和元素分组元素	45	3.6 练习	95
1.15 本章小结	45	第 4 章 表	97
1.16 练习	46	4.1 表简介	97
第 2 章 链接和导航	47	4.2 基本表元素和属性	100
2.1 基本链接	47	4.2.1 创建表的<table>元素	100
2.1.1 链接到其他文档	48	4.2.2 包含表行的<tr>元素	104
2.1.2 链接到 e-mail 地址	50	4.2.3 表示表单元格的<td>元素和<th>元素	106
2.2 理解目录和目录结构	50	4.3 高级表	112
2.2.1 链接的目标位置	52	4.3.1 将表划分为表头、表主体和表尾	112
2.2.2 URL 的组成	52	4.3.2 在表中添加<caption>	114
2.2.3 绝对 URL 和相对 URL	54	4.3.3 使用 colspan 属性跨越多列	114
2.2.4 <base>元素	57	4.3.4 使用 rowspan 属性跨越多行	115
2.3 利用<a>元素创建链接	58	4.3.5 使用<colgroup>元素分组列	116
2.3.1 利用 href 属性创建源锚点	58	4.3.6 利用<col>元素让列共享样式	118
2.3.2 利用 name 和 id 属性创建目的地锚点(链接到页面的特定部分)	59	4.4 表的可访问性问题	118
2.3.3 <a>元素的其他属性	60	4.4.1 表的线性化	118
2.4 高级 e-mail 链接	66	4.4.2 用于布局的表线性化	119
2.5 本章小结	67	4.4.3 用于数据的表线性化	121
2.6 练习	67	4.5 本章小结	121
第 3 章 图像和对象	69	4.6 练习	122
3.1 在站点中添加图像	69	第 5 章 表单	123
3.1.1 图像格式的类型	70	5.1 表单简介	123
3.1.2 位图图像	70	5.2 利用<form>元素创建表单	125
3.1.3 矢量图像	77	5.2.1 action 属性	125
3.1.4 使用元素添加图像	77	5.2.2 method 属性	125
3.2 利用<object>元素添加其他对象	83	5.2.3 id 属性	126
3.2.1 <object>元素的属性	84	5.2.4 name 属性(逐渐淘汰)	126
3.2.2 <param>元素	87	5.2.5 onsubmit 属性	126
3.2.3 在页面中添加 Flash 电影	87	5.2.6 onreset 属性	127
3.3 使用图像作为链接	89	5.2.7 enctype 属性	127
3.4 图像映射	90	5.2.8 accept-charset 属性	127
3.4.1 服务器端图像映射	90	5.2.9 accept 属性	128
3.4.2 客户端图像映射	91	5.2.10 target 属性	128
3.5 本章小结	94		

5.2.11 空白和<form>元素.....	128	6.4.6 scrolling 属性.....	174
5.3 表单控件	128	6.4.7 longdesc 属性.....	174
5.3.1 文本输入.....	129	6.5 <noframes>元素	175
5.3.2 按钮	133	6.6 创建框架之间的链接.....	175
5.3.3 复选框	135	6.7 框架集的嵌套.....	177
5.3.4 单选按钮.....	137	6.8 利用<iframe>元素创建 浮动框架或内联框架.....	181
5.3.5 选项框	139	6.9 本章小结.....	186
5.3.6 文件选项框.....	144	6.10 练习.....	186
5.3.7 隐藏控件.....	145		
5.3.8 对象控件.....	146		
5.4 利用<label>元素为控件 创建标签	150	第7章 层叠样式表	188
5.5 利用<fieldset>元素和 <legend>元素结构化表单	151	7.1 CSS 简介	189
5.6 焦点	153	7.1.1 一个基本的示例	190
5.6.1 焦点移动顺序.....	154	7.1.2 继承.....	193
5.6.2 访问键	155	7.2 添加 CSS 规则的位置.....	194
5.7 禁用的或只读的控件.....	156	7.2.1 <link>元素	194
5.8 向服务器发送表单数据.....	157	7.2.2 <style>元素.....	196
5.8.1 HTTP get.....	158	7.2.3 外部 CSS 样式表的优点	197
5.8.2 HTTP post.....	159	7.3 CSS 特性	198
5.9 本章小结	162	7.4 控制字体.....	199
5.10 练习	162	7.4.1 font-family 特性.....	200
第6章 框架	164	7.4.2 font-size 特性.....	201
6.1 框架集简介	164	7.4.3 font-weight 特性	203
6.2 使用框架的时机	166	7.4.4 font-style 特性	203
6.3 <frameset>元素	167	7.4.5 font-variant 特性	204
6.3.1 cols 属性	168	7.4.6 font-stretch 特性	204
6.3.2 rows 属性.....	170	7.4.7 font-size-adjust 特性	205
6.3.3 针对 <frameset>元素的 浏览器专用扩展	170	7.5 文本格式化.....	205
6.4 <frame>元素.....	172	7.5.1 color 特性	206
6.4.1 src 属性.....	172	7.5.2 text-align 特性	206
6.4.2 name 属性.....	173	7.5.3 vertical-align 特性	206
6.4.3 frameborder 属性.....	173	7.5.4 text-decoration 特性.....	208
6.4.4 marginwidth 属性和 marginheight 属性	173	7.5.5 text-indent 特性	208
6.4.5 noresize 属性	174	7.5.6 text-shadow 特性	209
		7.5.7 text-transform 特性	209
		7.5.8 letter-spacing 特性	210
		7.5.9 word-spacing 特性	210
		7.5.10 white-space 特性	211
		7.5.11 direction 特性.....	212

7.5.12	unicode-bidi 特性	212	8.2.6	background 特性(获得良好支持的简写形式)	252
7.6	文本伪类	213	8.3	列表	253
7.6.1	first-letter 伪类	213	8.3.1	list-style-type 特性	253
7.6.2	first-line 伪类	213	8.3.2	list-style-position 特性	254
7.7	选择器	216	8.3.3	list-style-image 特性	255
7.7.1	通用选择器	216	8.3.4	list-style 特性(简写形式)	256
7.7.2	类型选择器	217	8.3.5	marker-offset 特性	256
7.7.3	类选择器	217	8.4	表	256
7.7.4	id 选择器	217	8.4.1	表的特性	258
7.7.5	子选择器	218	8.4.2	border-collapse 特性	259
7.7.6	后继选择器	218	8.4.3	border-spacing 特性	260
7.7.7	相邻兄弟选择器	218	8.4.4	caption-side 特性	261
7.7.8	利用子选择器和相邻兄弟选择器降低标记中类的相关性	219	8.4.5	empty-cells 特性	261
7.7.9	属性选择器	220	8.4.6	table-layout 特性	263
7.8	长度	222	8.5	外边框	263
7.8.1	绝对单位	222	8.5.1	outline-width 特性	264
7.8.2	相对单位	222	8.5.2	outline-style 特性	264
7.8.3	百分比	223	8.5.3	outline-color 特性	264
7.9	框模型简介	223	8.5.4	outline 特性(简写形式)	264
7.9.1	演示框模型的示例	225	8.6	:focus 伪类和:active 伪类	265
7.9.2	Border 特性	227	8.7	生成的内容	265
7.9.3	padding 特性	230	8.7.1	:before 和:after 伪元素	266
7.9.4	margin 特性	231	8.7.2	content 特性	266
7.9.5	面积	232	8.8	其他特性	269
7.10	本章小结	241	8.8.1	cursor 特性	269
7.11	练习	241	8.8.2	display 特性	270
8.1	background-color 特性	246	8.8.3	visibility 特性	270
8.2	background-image 特性	247	8.9	额外的规则	271
8.2.1	background-repeat 特性	249	8.9.1	@import 规则: 模块化样式表	271
8.2.2	background-position 特性(用于固定背景的位置)	251	8.9.2	@charset 规则	272
8.2.3	background-attachment 特性(用于水印)	251	8.9.3	!important 规则	272
8.2.4	background-color 特性	246	8.10	CSS 的定位功能	273
8.2.5	background-image 特性	247	8.10.1	普通流	273
8.2.6	background-repeat 特性	249	8.10.2	position 特性	274
8.2.7	background-position 特性	251	8.10.3	框偏移特性	274
8.2.8	background-attachment 特性	251	8.10.4	相对定位	275
8.2.9	background-color 特性	246	8.10.5	绝对定位	276

8.10.6	固定定位	277	10.1.3	调整文本行高度以使 文本更具可读性	328
8.10.7	z-index 特性	278	10.1.4	宽列的文本更难以阅读	328
8.10.8	使用 float 特性浮动	279	10.1.5	背景图像会使文本 难以阅读	329
8.10.9	clear 特性	281	10.1.6	仔细选择字体	329
8.11	本章小结	286	10.1.7	固定大小的字体受屏幕 分辨率影响	331
8.12	练习	287	10.2	导航	331
第 9 章	页面布局	289	10.2.1	菜单	331
9.1	理解站点	289	10.2.2	链接	336
9.1.1	理解站点的目标	290	10.2.3	站点搜索功能	337
9.1.2	期望的站点访问者	291	10.3	在表的多行中添加阴影	339
9.1.3	新内容	291	10.4	表单	341
9.1.4	定义站点的内容	292	10.4.1	设计表单之前的工作	341
9.1.5	分组和分类	293	10.4.2	设计表单	343
9.1.6	创建站点地图	293	10.5	本章小结	360
9.1.7	标识每个页面的关键元素	295	10.6	练习	361
9.2	页面大小(和屏幕分辨率)	295	第 11 章	学习 JavaScript	362
9.3	设计页面	301	11.1	编程的定义	363
9.3.1	规划元素的位置	302	11.2	在页面中添加脚本的方式	364
9.3.2	引入样式	304	11.2.1	JavaScript 中的注释	366
9.3.3	导航	307	11.2.2	<noscript>元素	366
9.3.4	主页面	309	11.3	文档对象模型	368
9.3.5	内容页面	309	11.3.1	文档对象模型简介	368
9.4	构造页面	310	11.3.2	对象、方法和特性	370
9.4.1	单列布局	311	11.3.3	forms 集合	372
9.4.2	双列布局	314	11.3.4	表单元素	373
9.4.3	3 列布局	316	11.3.5	images 集合	376
9.4.4	牺牲列	318	11.3.6	不同类型的对象	379
9.4.5	利用 CSS 的高级布局	319	11.4	开始利用 JavaScript 编程	379
9.4.6	利用嵌套表创建布局	320	11.5	变量	380
9.5	本章小结	322	11.5.1	为变量赋值	380
9.6	练习	323	11.5.2	变量的生命周期	381
第 10 章	设计问题	324	11.6	运算符	381
10.1	文本	324	11.6.1	算术运算符	381
10.1.1	空白有助于制作更 吸引人的页面	325	11.6.2	赋值运算符	382
10.1.2	仔细对齐文本以使其 更具可读性	327	11.6.3	比较运算符	382

11.6.4	逻辑或布尔运算符	383	12.1.4	将脚本放置在 scripts 文件夹中	409
11.6.5	字符串运算符	384	12.2	表单验证	409
11.7	函数	384	12.2.1	什么时候验证	409
11.7.1	定义函数的方式	384	12.2.2	如何验证	410
11.7.2	调用函数的方式	384	12.3	增强表单的可用性	423
11.7.3	return 语句	385	12.3.1	关注第一个表单项	423
11.8	条件语句	385	12.3.2	字段之间的自动 焦点移动	424
11.8.1	if 语句	386	12.3.3	禁用文本输入框	425
11.8.2	if...else 语句	386	12.3.4	转换大小写	426
11.8.3	switch 语句	387	12.3.5	剪裁字段开头和 结尾的空格	427
11.8.4	条件(或三元)运算符	389	12.3.6	选择文本区域中的 所有内容	428
11.9	循环	389	12.3.7	选中或取消选中 所有复选框	429
11.9.1	while 循环	389	12.4	图像翻转	435
11.9.2	do...while 循环	390	12.5	随机脚本生成器	437
11.9.3	for 循环	390	12.6	弹出式窗口	438
11.9.4	无限循环和 break 语句	391	12.7	JavaScript 库	440
11.10	事件	391	12.7.1	利用 Scriptaculous 库 创建动画效果	440
11.11	内置对象	393	12.7.2	利用 Scriptaculous 库 拖放可排序列表	442
11.11.1	字符串对象	393	12.7.3	利用 MochiKit 库创建 可排序表	443
11.11.2	日期对象	396	12.7.4	利用 YUI 库创建日历	445
11.11.3	数学对象	399	12.7.5	利用 YUI 库创建自动 完成的文本输入框	446
11.11.4	数组对象	401	12.8	何时不使用 JavaScript	448
11.11.5	Window 对象	402	12.8.1	下拉导航菜单	448
11.12	编写 JavaScript 代码	404	12.8.2	隐藏 e-mail 地址	448
11.12.1	关于数据类型的 注意事项	404	12.8.3	快速跳转选择框	448
11.12.2	关键字	405	12.8.4	用户需要从站点中 获得的任何信息	449
11.13	本章小结	406	12.9	本章小结	449
11.14	练习	406	12.10	练习	450
第 12 章	应用 JavaScript	407			
12.1	关于编写脚本的一些 实用提示	407			
12.1.1	其他人是否已经 编写过这个脚本	407			
12.1.2	可重用的函数	408			
12.1.3	使用外部 JavaScript 文件	409			

第 13 章 在 Web 上发布站点	452	13.5.4 Flash	483
13.1 Meta 标签	453	13.5.5 学习图形程序包	484
13.1.1 name 属性和 content 属性	453	13.6 本章小结	485
13.1.2 http-equiv 属性和 content 属性	455	附录 A 练习题答案	487
13.1.3 scheme 属性	458	附录 B XHTML 元素参考	511
13.2 测试站点	459	附录 C CSS 特性	549
13.2.1 目录结构和相对 URL 的重要性	459	附录 D 颜色名和颜色值	575
13.2.2 验证 HTML、XHTML 和 CSS	460	附录 E 字符编码	582
13.2.3 检查链接	462	附录 F 特殊字符	585
13.2.4 检查不同的屏幕分辨率 和颜色深度	464	附录 G 语言代码	595
13.2.5 可访问性检验工具	464	附录 H MIME 媒体类型	598
13.2.6 开发服务器或 主运行服务器	464	附录 I 逐渐淘汰的和浏览器 专用的标记	607
13.2.7 在浏览器的不同版本 中执行检查	464		
13.2.8 引导测试	465		
13.2.9 校对	466		
13.3 发布站点	466		
13.3.1 获得域名	467		
13.3.2 主机托管	468		
13.3.3 搜索引擎策略	471		
13.3.4 其他 Web 市场 营销策略	474		
13.3.5 统计分析	475		
13.3.6 版本控制	477		
13.4 下一步执行的操作	478		
13.4.1 博客	478		
13.4.2 讨论板或论坛	479		
13.4.3 添加搜索实用程序	479		
13.5 其他技术简介	480		
13.5.1 服务器端 Web 编程: ASP.NET 和 PHP	480		
13.5.2 选择服务器端语言	480		
13.5.3 内容管理	481		

第 1 章

创建结构化文档

本章介绍编写 Web 页面所需要学习的首要技术：HTML 和 XHTML。事实上，您将学习的是 XHTML——后面会解释 HTML 和 XHTML 之间的区别。(可以将 XHTML 简单看作为 HTML 的最新版本)。

本章的主要目的是演示 XHTML 如何描述文档的结构。

在本章中将介绍以下内容：

- 学习标签、元素和属性之间的区别
- 了解 Web 页面如何使用标记描述结构化页面的方式
- 介绍用于标记题头和段落等文本的元素
- 学习许多其他元素，这些元素能够在文档中添加表示信息和语义
- 了解如何在文档中添加项目列表和编号列表
- 介绍一些用于区分 XHTML 中元素类型的核心概念

阅读完本章后，读者能够了解如何使用 XHTML 构造页面，并能够编写自己的第一个 Web 页面。

1.1 结构化文档组成的 Web

在日常生活中，人们会接触各种类型的打印文档——报纸、列车时刻表、保险单。Web 由大量链接在一起的文档所组成，这些文档类似于日常生活中的文档。因此，首先回顾日常生活中看到的一些文档的结构，并将它们与 Web 页面作比较。

每天早上我习惯阅读报纸。报纸由一些故事和文章(很可能还有一些广告)组成。每个故事具有一个标题、一些段落，或许还有一个副标题以及其他更多的段落，还可能包含一幅或两幅图片。

我一般不买日报，因为常常在线观看新闻，但 Web 站点中的文章的结构与报纸中文章的结构非常相似。每篇文章也是由题头、文本段落和图片组成。两者非常相似，仅有的区别是，Web 站点中的每个故事具有它自己的页面，访问该页面仅需要单击它的标题行或者简短概述，这些标题行和简短概述可能位于站点的主页面或者站点的某个分部(例如政治、体育或者娱乐部分)的主页面。

举一个例子：假设乘火车去看望一个朋友，因此需要查看列车时刻表以了解到他那里

的列车车次。列车时刻表的主要部分是一张表，表中列举了列车在各个站的到达和离开时间。如同许多文档具有标题和段落一样，许多其他的文档也使用表；从报纸财经增刊中的股票信息页面到电视节目播出列表，人们每天都会遇到一些包含信息的表——并且这些表通常也被创建在 Web 中。

另外一种常见的文档是表单。例如，桌面上有一张保险公司的表单。在这张表单中有一些需要填写的字段，包括姓名、地址、保险金额，也有一些方框，需要在这些方框中进行选择以指出房子中房间的数量和大门门锁的类型。事实上，Web 页面中存在大量表单，从询问查找内容的简单搜索框到在线订购书籍或 CD 之前要求填写的注册表单。

根据上面的介绍可以发现，日常生活中打印文档的结构和 Web 中页面的结构非常类似。因此，在开始编写 Web 页面时，您会很容易地理解代码将告诉 Web 浏览器需要显示信息的结构——哪些文本被放置在标题、段落或者表中等——以便浏览器能够正确将它们显示给用户。

为了告诉 Web 浏览器文档的结构——如何建立标题、段落、表等——您需要学习 HTML 和 XHTML。

1.2 XHTML 简介

XHTML(可扩展超文本标记语言)与它的前身 HTML 是 Web 上最广泛使用的语言。从名称中可以看出，XHTML 是一种标记语言，它看上去可能很复杂，但如果意识到每天都会遇到各种标记，这种观点就会改变。

当在文字处理软件中创建文档时，可以对文本添加样式以解释文档的结构。例如，可以使用题头样式(通常具有较大的字体)区分标题和文本主体，使用 Enter(或 Return)键开始一个新的段落，在文档中插入表以放置数据，或者为一系列相关的要点创建项目列表等。这些样式会影响文档的表现形式，但是这种标记的关键目的是提供一种结构，以使文档更容易理解。

当标记 Web 文档时，实际上执行的是一种非常相似的过程，不同的只是需要在 Web 文本上添加标签。对于 XHTML，关键是要记住添加标签以指明文档的结构，即文档的哪一部分是题头，哪些部分是段落，哪些内容属于表等。浏览器(例如 Internet Explorer、Firefox 和 Safari 等)将使用这种标记以用户熟悉的方式显示文本，它的行为类似于文字处理软件(标题的字体大于正文，段落间存在一些空格，项目列表的每一项前面有一个圆点)。然而，这些内容的表现方式由浏览器决定；XHTML 规范没有说明应当使用哪些字体或者该字体的尺寸。

注意：

早期版本的 HTML 允许控制文档的表现形式——例如文档应当使用哪些字体和颜色——但是 XHTML 标记不应该用于样式化文档，这些工作由 CSS 完成，第 7 章中将介绍 CSS。

首先查看一个非常简单的 Web 页面。前言中曾提到，编写 Web 页面不需要任何特殊的程序——可以只使用一个文本编辑器，例如 Windows 中的 Notepad 或 Mac 中的 TextEdit。