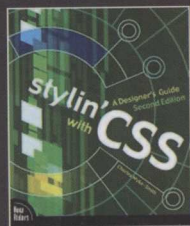


TURING

New
Riders

Second Edition
Stylin' with CSS: A Designer's Guide
写给大家看的CSS书
(第2版)

[美] Charles Wyke-Smith 著
李松峰 张程 等译



- 学习 CSS 就这么简单
- 在不知不觉中成为 Web 设计高手
- Amazon 全五星盛誉畅销著作



人民邮电出版社
POSTS & TELECOM PRESS

TURING

Second Edition
Stylin' with CSS: A Designer's Guide
写给大家看的CSS书
(第2版)

[美] Charles Wyke-Smith 著
李松峰 张程 等译

人民邮电出版社
北 京

图书在版编目 (CIP) 数据

写给大家看的 CSS 书：第 2 版 / (美) 威克 - 史密斯 (Wyke-Smith, C.) 著；李松峰等译. —北京：人民邮电出版社，2009.2

书名原文：Stylin' with CSS: A Designer's Guide, Second Edition
ISBN 978-7-115-19364-3

I. 写… II. ①威…②李… III. 主页制作 - 软件工具, CSS IV. TP393.092

中国版本图书馆 CIP 数据核字 (2008) 第 196312 号

内 容 提 要

本书介绍了 CSS 样式设计，主要涵盖利用 XHTML 为内容定义结构、CSS 的工作原理、字体和文本样式、定位元素、基本的页面布局等内容，详细讲述了所有程序员都需要掌握的两种技能：避免重写以前曾经写过的代码和掌握以最经济的方法编写代码。同时，书中的所有代码都可以从本书的网站 (www.stylinwithcss.com) 下载。而且，作者会在发现错误之后改正它们并更新下载文件。

本书适合 Web 开发人员阅读和参考。

写给大家看的 CSS 书 (第 2 版)

◆ 著 [美] Charles Wyke-Smith

译 李松峰 张程等

责任编辑 傅志红

执行编辑 杨爽

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

北京铭成印刷有限公司印刷

◆ 开本：800×1000 1/16

印张：18.5 彩插：6

字数：437千字 2009年2月第1版

印数：1-3 000册 2009年2月北京第1次印刷

著作权合同登记号 图字：01-2008-3337号

ISBN 978-7-115-19364-3/TP

定价：49.00元

读者服务热线：(010)88593802 印装质量热线：(010)67129223

反盗版热线：(010)67171154

版权声明

Authorized translation from the English language edition, entitled *Stylin' with CSS: A Designer's Guide, Second Edition* by Charles Wyke-Smith, published by Pearson Education, Inc., publishing as New Riders. Copyright © 2008 by Charles Wyke-Smith.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Simplified Chinese-language edition copyright © 2009 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Pearson Education Inc.授权人民邮电出版社独家出版。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。

版权所有，侵权必究。

译者序

CSS 不难学——但前提是必须有好的教材。

本书第 1 版之所以畅销，其中一个主要原因就是作者讲解清晰，内容通俗易懂。在第 1 版获得成功的基础上，作者根据自身实践，将 CSS 的最新发展和应用结合起来，基本上重写了第 1 版，最后将全新的版本呈现给读者。在这一版中，作者沿续了受到广泛好评的平易朴实的写作风格，将 CSS 中的关键概念和基础术语阐释得更加透彻、准确。相信有一定 CSS 经验的读者看罢这一版，一定会有相见恨晚之感！

第 2 版虽然可以说重写了第 1 版，但内容架构上仍然保持了第 1 版的按技术主题分类，以实例辅助讲解的组织方式。看全书目录好像比较扼要，但深入各个章节之后，就会发现其涵盖面之广几乎可以与一本权威指南之类的参考书媲美。而且，由于作者表达精练，经常一语中的，因此整本书的篇幅虽小，但信息量、实用性几乎不亚于大部头的 CSS 著作。译者个人认为，这种组织方式有利于读者提纲挈领，从全局角度理解并记住 CSS 的关键技术和主要应用。

第 2 版还有一个特色，就是随书提供的 Stylib 样式库。在这个库中，作者提供了多栏布局、导航链接、表单等 Web 设计常用组件的样式模板。读者只要根据书中的介绍，为页面中添加一些标记并定义相应的类，就可以自动拥有该库提供的样式化能力，从而轻松实现常用的界面元素设计。应该说，设计 CSS 样式库是一种创新，也是 CSS 应用进入高级阶段的标志。正如层出不穷的 JavaScript/Ajax 库一样，也许 Charles Wyke-Smith 提供 CSS 样式库的理念也折射出了 CSS 应用的一种发展趋势。

总之，用简短的语言来概括本书特色，就是讲解透彻、组织巧妙、富有新意，是学习 CSS 的一本好教材。

本书由李松峰负责翻译，参加翻译工作的还有张程、宋会敏、曹建辉、崔淑云、闫建旺、熊俊佳、熊俊芹、刘英、左艳波、贾爱华等。

前 言

很难相信，从我写本书第 1 版到现在已经过了 3 年。在此期间，我又参与了大量网站的开发并调整了使用 CSS 的方式。我原计划对本书第 2 版进行一些小的调整，涵盖 IE 7 并从整体上使内容与时代同步。但是，最终我却对前 3 章进行了大量改进，并且完全重写了书中剩余的内容。因此，也使原计划几周的工作量用了将近一年才完成。

我所做的修改反映了任何经常使用 CSS 的开发人员在技术上的进步，而且这些修改也更深入地涉及所有程序员都需要掌握的两种技能：避免重写以前曾经写过的代码和掌握以最经济的方法编写代码的方法。在新版本中，介绍了实现这两个有价值的目标的各种方式。

重用和DRY

第一个目标是代码重用，它是在好几章中都会涉及的一个主题。书中展示了创建功能性构建块的技术，无论是页面布局的框架结构，还是应用样式的美观的链接列表，将它们保存为可以迅速添加到标记中的 CSS 类，设计者能够对其进行个性化地修改。本书已经将这些构建块添加到一个名为 Stylib 的代码库中，其中包含各种界面元素并以 CSS 代码及其相关 XHTML 标记的形式组织。这个库及本书的示例代码也放在了网站 www.stylinwithcss.com 中。虽然 Stylib 库还不算成熟，但已经包含了很多有用的组件，这些组件曾为我节省了很多时间——但愿对你也同样适用。今后，我还要不断完善这个库并将它们放到网站上供读者下载。

另一个目标是代码节约，它是贯穿本书的另一个主题。举例来说，为文档层次中最高层的元素应用一个样式，该样式就能够影响到其他众多元素。我经常看到别人的 CSS 样式表中包含为每个标题和段落声明的相同的 font-family 规则，而通过把 font-family 属性应用给顶层的 body 标签则意味着只需在一个地方编写和维护该属性。代码节约的底层理念是一句编程名言，叫做 DRY，意思是 Don't Repeat Yourself（不要重复自己）。在本书前面几章中，我们会展示很多类似的例子。

掌握关键技术

我认为只要真正理解了少数重要的 CSS 技术，就能够使一名新手变成能够独当一面的 CSS 熟手。这些技术包括正确地使用定位和 display 属性，以及理解浮动和清除的工作原理。在本书第 4 章中，将专门介绍这些 CSS 中的重点技术，并通过简单的例子示范如何在实践中运用它们。任何对 CSS 感兴趣并希望把自己的技能提升到一个新水平的读者，都会发现第 4 章非常具

有启发性——我当然盼望如此。

使用正确的工具

在真正开始学习本书之前，我想解释一下如何做好准备工作，以便读者在最短的时间内完成跨大量浏览器的页面布局。

首先，需要一个优秀的代码编辑器来编写代码，需要一个 FTP 客户端向 Web 服务器上传文件。Adobe Dreamweaver（大约 400 美元）整合了这两种工具，也是我日常使用的编程工具。Dreamweaver 的代码视图非常有用，而且代码颜色及自动完成功能有助于区分代码中的不同元素，识别语法错误以及猜测可能的输入（只需输入前几个字母，其余的都能自动完成）。设计视图则会模仿浏览器的方式呈现页面，这当然也很有用，因为通过单击视图中的任何元素都会立即转到代码视图中的相关部分。当然，也有一些价格便宜、甚至免费的编辑器和 FTP 客户端，它们也能够提供类似的功能，同时为你省下一笔不菲的开销。不过，根本的问题不在于使用什么工具来编写和上传代码，而在于网页在用户浏览器中的最终效果如何。

只考虑4种浏览器

看到这个标题，你可能会因为只需要在 4 种浏览器中测试代码而感到奇怪。这 4 种浏览器如下。

- Firefox
- Safari
- IE 7
- IE 6

原因是这 4 种浏览器的市场占有率超过了 95%，而其他任何浏览器的市场占有率都没有超过 2%（数据来自维基百科上关于各种浏览器使用率统计的平均值，地址为 http://en.wikipedia.org/wiki/Usage_share_of_web_browsers）。

在这 4 种浏览器中，无论 IE 6 存在多少呈现 bug 和对 CSS 属性的支持有多匮乏，尽管人们已经慢慢（非常慢）开始放弃使用它了，但它仍然是主流浏览器。另外 3 种浏览器都是我们常说的 SCB，即 Standards Compliant Browser（符合标准的浏览器）。也就是说，它们基本上都能够严格遵循 W3C（World Wide Web Consortium，万维网联盟）的推荐规范来实现对 CSS 的支持。事实上，这 3 种浏览器都能够相当精确地呈现全部 CSS2 及部分 CSS3 属性（可以将 CSS2 和 CSS3 看成是 CSS 的版本，后面我们会介绍更多与此相关的内容）。换句话说，当它们呈现了有效的 CSS 样式的 XHTML 页面时，结果的差异很小。

在上面的列表中也可以包含 Netscape，即使它的市场份额最多也只有 1 个百分点。但是，因为 Netscape 和 Firefox 都是在 Mozilla 呈现引擎基础上构建的浏览器，所以在 Firefox 中能够正确

呈现的页面，在 Netscape 中也应该得到几乎相同的结果。因此，如果基于以上 4 种浏览器测试和调整页面并取得满意的结果，就可以确保页面在几乎任何用户的浏览器中都能够得到体现设计意图的呈现。这里，我们甚至不会提到在 IE 5.5 中测试；因为它当前只有不到 0.5% 的使用率，而任何使用一个具有 8 年历史的老浏览器的用户，恐怕都会有比呈现我们的网页更大的技术问题。

我的建议是绝对不要使用 IE 6 作为开发浏览器，而要使用前面 3 个 SCB 中的一种，在 SCB 中已经能得到满意的页面后再在 IE 6 中测试并调整。实际上，更进一步讲，我会强烈建议你开发期间使用 Firefox 来预览页面。之所以作此推荐，是因为我发现 Firefox 是所有 SCB 中最符合标准的一种浏览器。此外，在 Firefox 中安装 Web Developer Toolbar（Web 开发者工具条），就可以方便地验证 XHTML 和 CSS 而无需上传到服务器，启用和禁用 CSS，查看页面中 XHTML 元素的轮廓以便实际地确定元素是否按照计划进行布局，以及使用其他一些有用而且节省时间的工具。假如不安装 Web Developer Toolbar，那么大量开发时间就会白白浪费掉。而且，还可以为 Firefox 安装 Firebug 调试工具，有了这个附加软件，差不多就具有了进行 Web 开发的专业级环境。Firefox 和这两种附加软件都可以通过 www.getfirefox.com 免费下载。

请下载代码，别再自己敲了

尽管本书在出版之前进行了认真的编辑和审校，但错误仍然在所难免。我会及时修改书中代码存在的问题，并更新到本书网站（www.stylinwithcss.com）上。如果读者要试验本书的示例代码，请直接到本书站点下载。这样，不仅可以节省手工录入的时间，而且可以得到修正后的代码。

别忘了给我写信

如果你发现了错误，请到本书网站上发电子邮件告诉我，我会在确认之后把勘误公布到网站中。同时，我也希望看到读者的评论和建议，因此在你购买本书后要记着花几分钟给我写封信。我会尽最大努力回复每一封读者的来信，也会尽量回答读者遇到的任何问题。不过，假如你真的想把 CSS 代码发送给我看，请把它嵌入到 XHTML 文档的头部，并将页面中的图像指向你的服务器的绝对 URL；然后，将 XHTML 文档通过电子邮件发送给我。这样，只要打开页面，我就能看到相应的效果。

最后，感谢你购买这本书。希望它能够给你带来非常大的帮助。

本书中使用的图标说明



说明



提示



注意

① 可以在图灵网站 www.turingbook.com 本书网页免费注册下载。——编者注

致谢

一个人不可能单独完成这么一本书。因此，我要感谢鼓励我编写本书第2版的出版人——Peachpit的Nancy Ruenzel，感谢为我提供宝贵建议且富有幽默感的总编辑Nancy Davis，感谢New Riders的策划编辑Michael Nolan再一次让一切都顺利进行。对于生产团队，我要感谢我的项目编辑Marta Justak，感谢她在反复推敲本书时的付出和对我进行的指导，感谢生产编辑Hilal Sala始终如一的鼓励，感谢排版编辑David Van Ness认真制作本书的版式，感谢Doug Adrianson的认真校对。另外，还要特别感谢技术审稿人Virginia DeBolt，感谢她对书中代码及相关内容娴熟的审核。

接下来，要感谢本书第1版的所有读者，为我提供了你们的想法和反馈意见。但愿我理解并拿出了你们希望看到的东西，希望你们继续向我提出意见并将你们网站的URL通过电子邮件发送给我。

对于我工作的Benefitfocus.com，我想感谢我的上司、企业产品战略副总裁Nancy Sansom，感谢她给我编写本书所需的灵活性和对我工作的一贯鼓励和指导，也感谢她给了我一家令人赞叹的公司工作的机会。此外，我还要感谢Benefitfocus的CEO Shawn Jenkins和COO Jerry Lepore，感谢他们的支持和领导以及创建并运营着一家我见过的最好的公司。

下面，我要感谢User Experience Group团队能干而又富有创意的成员们，能够领导他们并向他们学习真是一种幸运，他们是Brad Bonham、Darin Cavanaugh、Daniel Nadeau、Michael Rosier、Mike Attisano、Leah Giorlando、Tony Malatanos和Matt Megrue。另外，我也非常高兴作为Product Strategy团队的一部分共同经受挑战，这个团队中有机敏睿智的Raymond Minnis、Nathan Reel、John“Coach”Wilson、Kinsey Rawe和Mike Fullan。另外还有一些使在Benefitfocus工作充满乐趣和激情的同事：Nina Sossamon-Pogue、Andrew Alberico、Elena Tudor、Will Deupree、Manon Husman、Kelli Hicks、Tracey Snyder、Annmarie Fini、Michelle Pagel、Nate DaPore、Randy Chitwood、Glenn Miller、Heather Norton、Rebecca Laney、Paul Sparrow、Robert Moss、Don Taylor和Jonathan Chandler——感谢你们给我的灵感、友谊和支持。

我还要特别感谢我的妻子Beth。她的图形设计及写作技巧，为本书的每一页都增添了光彩。她检查我的用词、简化我的表达，并且为书中过长的段落修改标点符号，这些对我来说非常有必要。而且，她还帮我制作图表，设计第7章重点介绍的本书网站（否则，我也不会把自己的照片搞那么大。干得好，亲爱的）。因此，在将本书送交编辑之前，基本保证了各章的完整性和连续性。本书之所以能有现在的样子，也得力于她对细节的持续关注和提出的改进建议。谢谢你，我的好妻子。

最后，感谢我的两个可爱的女儿Jemma和Lucy。我想说，你们俩在老爸写书时都表现出了极大的耐心和理解。现在，我们可以在星期六的早上一块出去吃冰淇淋了！

——Charles Wyke-Smith

2007年11月14日，南卡罗来纳，查尔斯顿

目 录

第 1 章 XHTML : 为内容定义结构	1
1.1 Web 标准	2
1.1.1 时至今日, 仍然会提到 IDWIMIE	2
1.1.2 内容、结构和表现	3
1.2 时代在改变	5
1.2.1 过去的方式	5
1.2.2 未来刚刚开始	7
1.3 XHTML 及编写规则	8
1.3.1 XHTML——规则	9
1.3.2 XHTML 模板	14
1.3.3 标记内容	16
1.3.4 文档流——块级元素和行内元素	16
1.3.5 文档层次: 认识 XHTML 家族	22
第 2 章 CSS 的工作原理	25
2.1 为文档应用样式的 3 种方式	26
2.1.1 内联样式	26
2.1.2 嵌入样式	27
2.1.3 链接样式	28
2.2 CSS 规则剖析	30
2.3 编写 CSS 规则	31
2.4 在文档层次中对准标签	32
2.4.1 使用上下文选择符	32
2.4.2 使用子选择符	36
2.4.3 添加类和 ID	37
2.4.4 ID 简介	41
2.4.5 ID 和类之间的区别	42
2.4.6 特殊的选择符	43
2.4.7 选择符小结	46

2.5 伪类	47
2.5.1 锚链接的伪类	47
2.5.2 其他有用的伪类	49
2.6 伪元素	50
2.7 继承	52
2.8 层叠机制	53
2.8.1 样式的来源	53
2.8.2 层叠规则	54
2.9 规则声明	58
2.9.1 数字值	58
2.9.2 颜色值	61

第 3 章 字体和文本样式

63

3.1 在 CSS 中指定字体	64
3.2 探索字体系列	67
3.2.1 使用嵌入样式 (仅现在)	69
3.2.2 为整个页面设置字体系列	70
3.3 设置字体大小	71
3.4 字体属性	77
3.4.1 font-style 属性	77
3.4.2 font-weight 属性	78
3.4.3 font-variant 属性	79
3.4.4 字体属性的简写方式	80
3.5 文本属性	80
3.5.1 text-indent 属性	82
3.5.2 letter-spacing 属性	84
3.5.3 word-spacing 属性	85
3.5.4 text-decoration 属性	85
3.5.5 text-align 属性	86
3.5.6 line-height 属性	87
3.5.7 text-transform 属性	89
3.5.8 vertical-align 属性	90
3.6 使用字体和文本属性	92

第 4 章 定位元素

97

4.1 理解盒模型	98
4.1.1 盒子的边框	99
4.1.2 盒子的内边距	102

4.1.3	盒子的外边距	102
4.1.4	折叠外边距	104
4.2	盒子到底有多大	105
4.3	浮动和清除	110
4.3.1	float 属性	110
4.3.2	clear 属性	112
4.4	position 属性	116
4.4.1	静态 (static) 定位	116
4.4.2	相对 (relative) 定位	117
4.4.3	绝对 (absolute) 定位	118
4.4.4	固定 (fixed) 定位	119
4.4.5	定位环境	120
4.5	display 属性	123
4.6	使用 position/display 属性的例子	124

第 5 章 基本的页面布局

129

5.1	有代表性的多栏布局	130
5.2	本书 CSS 库——Stylib 简介	133
5.3	宽度问题	133
5.4	浮动布局与绝对定位布局	134
5.4.1	简单的两栏式固定宽度布局	135
5.4.2	理解内部 div	140
5.4.3	防止不必要的溢出	140
5.4.4	按照需要为内部 div 添加样式	141
5.4.5	为文本添加样式	141
5.5	简单的两栏流动式布局	141
5.5.1	使用一点限制	143
5.5.2	浮动还是不要浮动	144
5.6	三栏式固定宽度布局	145
5.7	三栏流动式布局	149
5.8	设计长度相同的分栏	153
5.8.1	人造分栏	154
5.8.2	以编程方式扩展分栏 (并添加圆角)	158
5.9	绝对定位的布局	162

第 6 章 设计界面组件

169

6.1	为表格添加样式	170
6.2	为表单添加样式	183
6.2.1	表单的工作原理	183

6.2.2	表单的标记	184
6.2.3	表单的样式	192
6.3	为列表和菜单添加样式	199
6.3.1	列表	199
6.3.2	创建基于 CSS 的菜单	211

第 7 章 构建网页 225

7.1	本书网站简介	226
7.2	设置文件夹结构	228
7.3	创建站点结构	230
7.3.1	从库中复制必要的 CSS 文件	233
7.3.2	@import 规则	233
7.3.3	与文本和颜色有关的样式表	237
7.3.4	页面中的标记	242
7.3.5	背景图像	245
7.3.6	下拉菜单	248
7.3.7	透明的侧边栏面板	251
7.3.8	添加注册表单	256
7.3.9	文本样式	259
7.4	结束语	265

附录 A XHTML 标签参考 267

附录 B CSS 属性参考 271

第 1 章

XHTML：为内容定义结构

本书主要讲述如何以最有效、最简捷的方式构建符合标准的网站。并且，使这些网站容易被用户访问、也容易更新。Web 标准就是由 W3C（World Wide Web Consortium，万维网联盟）发布的一组推荐规范。如果所有浏览器制造商和所有 Web 程序员都遵循这些规范，那么从理论上讲，所有网页会在每个浏览器中都具有相同的外观和行为。但是，说起来容易，实行起来却很难。

2004 年底，当我撰写本书第 1 版时，Web 标准运动正处于风起云涌的态势。今天，大多数新建立的网站都以符合 Web 标准的方式来设计和编程。因此，Web 也已经发展到了一个更好的阶段。

由 Web 标准的倡导者推动的这一运动，使得 Web 不仅发展到了更好的阶段，而且也变得更加容易预知。这些倡导者同浏览器制造商共同努力，确保了新浏览器能够按照 W3C 的推荐规范来解释 3 种主要的客户端编程语言（XHTML、CSS 和 JavaScript），而不是从强调竞争优势的一己私利出发各自为战地使用专有标签和特性。

1.1 Web标准



我经常提到 CSS2 或 CSS3。这两个术语只是引用 CSS 标准的两个特定的版本。同其他技术一样, CSS 标准也会不断得到改进。目前, CSS2 在大多数浏览器中几乎都得到了完整的实现; CSS3 虽然已经出台一段时间了, 但仍然只得到 Firefox 和 Opera 的部分支持, 基本上没有得到 IE 7 的支持。本书后面会更多地提到两个版本的 CSS 标准。



如果只对 IE 6 或者对 IE 6 和 IE 7 都是这种情况, 我会一一指明。

在遵循最佳 Web 标准实践的基础上, 像你我一样的 Web 开发者能够为所有用户实现外观和性能几乎一致的网站。例如, 我们可能认为微软的 IE 是大多数符合 Web 标准的浏览器中最好的。但是, 不管其现有的优势怎样, 事实却不是如此。

根据 W3C 推荐规范, 其他一些浏览器在解释 CSS 方面做得很好。像 PC 平台上最新版的 Firefox 和 Opera, 以及 Mac 平台上的 Safari 和 Firefox 等, 都能够以几乎一致的方式呈现按照 CSS2 规范设计的 XHTML 样式。然而, 微软的 IE 6 却包含很多的未实现特性, 并且还存在着其他一些不标准的实现。

相对于 IE 6 来说, 微软在 2006 年 10 月发布的 IE 7 则有了极大的改进。为此, 我一直盼望着人们能够迅速地从 IE 6 切换到 IE 7。可是, 根据 thecounter.com 的统计, 截至 2007 年 7 月, 仍然有 50% 的上网用户使用 IE 6^①。

1.1.1 时至今日, 仍然会提到 IDWIMIE

不管怎样, 由于 IE 6 还迟迟不肯退出历史舞台, 我仍然会在提到某个 CSS 特性时告诉你 IDWIMIE (音 id-wimmy), 即 It Doesn't Work In Microsoft Internet Explorer (它在微软的 IE 中无效)。

针对 IE (及其他较早浏览器) 的缺点, 存在一种叫做 hack 的解决方案。所谓 hack, 就是指以非标准的方式来使用 CSS, 以达到欺骗特定的浏览器, 使其“看到”或忽略某些样式的目的。创建 hack 既乏味又耗时, 但只要 IE 6 普遍存在, hack 的创建就要持续下去。

对于网站设计者和访问者而言, Web 标准为在所有平台的所有浏览器中实现一致的网站外观和行为提供了一种预期。虽然这一预期还没有实现, 但是, 每种浏览器都支持不同的特性集, 从而导致结果不一致, 并使得跨浏览器 / 跨平台的 Web 开发效率低下、难以驾驭的时代 (似乎) 已经一去不复返了。今天, Web 标准已经深入人心。

^① 截至 2008 年 10 月 17 日, IE 7 的市场占有率 (42%) 已超过 IE 6 (36%), 参见 <http://www.thecounter.com/stats/2008/October/browser.php>。——译者注



严格来讲，XHTML 和 CSS 并不是编程语言，而是分别用来标记内容和为内容提供样式的机制，因此这里所说的“语言”只具有一般化的含义。



img 标签也包含两个属性，分别是图像的来源和替代文本——要了解更多有关属性的内容，请参考提示条“什么是属性”。

1.1.2 内容、结构和表现

因此，遵循 W3C 推荐的 Web 标准，本书将向你展示如何使用 XHTML 定义内容的结构，如何使用 CSS 定义内容的表现。

(1) 内容是一个综合的概念，它包括文本、图像、视频、声音、动画和文件（例如 PDF 文档）等，所有你想交付给读者的东西。

(2) XHTML (eXtensible HyperText Markup Language, 可扩展超文本标记语言) 允许你定义内容中的每个元素是什么。是标题或者段落呢？还是一个项目列表、一个超链接或者是一幅图像？一切都由你为内容添加相应的 XHTML 标记来决定。标记由用来识别内容中每个元素的标签（标签名被一对尖括号 < > 包围）组成。要创建一个 XHTML 元素（以下简称元素），既可以使用一个开始标签和一个结束标签来围绕一块内容，像这样：

```
<p>This tag defines the text content as a paragraph</p>
```

也可以对非文本内容（如本例中的图像）使用一个单独的标签：

```

```

本章主要讨论 XHTML 及如何使用它，但此刻最重要的是要知道：XHTML 定义一个文档的结构。

(3) CSS (层叠样式表) 允许你定义每一个被标记了的内容元素怎样在页面上表现出来。某个段落的字体用 Helvetica 还是 Times？是粗体还是斜体？是缩进还是与页边对齐？CSS 能够控制每个内容元素的格式及定位。要格式化一个段落中文字的大小，我可能会这样写：

```
p {font-size: 12px;}
```

这会使文字的高度为 12 像素。本书的全部内容几乎都是教你学习 CSS，但此刻最重要的事情是要知道：CSS 定义一个文档的表现。

什么是属性

属性能够添加到标签中,用以进一步定义标签。每个属性由两部分组成:属性名和属性值,格式为 name="value"。例如,下面的图像标签:

```

```

包含两个属性:图像的来源,这个属性的值 "images/fido.gif" 定义了图像在服务器上的相对位置;图像的替代描述文本,这个属性的值 "a picture of my dog" 会在图像加载失败时出现在屏幕上或者被屏幕阅读器大声地读出来。类似这些标签的属性都属于文档的结构部分。

在 Web 标准出现之前,人们经常把用于表现的属性也放到标签中,例如文本大小和颜色等属性。现在,通过把所有表现信息都转移到样式表中,可以极大地降低标记的复杂性,而属性也只用来定义文档的结构。

开发 Web 标准的一个核心目标就是提供一种将文档结构与表现分开的方法,这对于开发既具有可移植性(可以在多种设备中显示)又具有持久性(未来仍然可用)的内容是至关重要的。

基于标准编码的10大好处

你可能会感到奇怪——“为什么要费力地改变多年来标记网页的方式呢?”,以下是采用基于标准编码实践的10大理由。

(1) 可以交付到多用户代理。同一标记块中的内容可以容易地交付到各种用户代理(对能够阅读 XHTML 文档的设备的统称),例如浏览器、带浏览器的智能手机、蜂窝电话,以及为视力残障人士设计的阅读文本的屏幕阅读器等。只需为每种类型的设备创建一个不同的样式表,或者毫无修饰地显示 XHTML 即可。

(2) 改善性能。由于内容中只包含最少的结构化标记,网页会更轻(文件更小),因而下载更快。可以用一个样式表替换掉通常要在网站每个页面中加入的表现标记。稍后会看到,一个样式表可以为整个网站定义表现,而用户的浏览器只需下载这个样式表一次。

(3) 适合所有浏览器。只需一点努力,就可以使网页很好地兼容早期的浏览器,于是所有用户都会尽可能地获得基于其可用技术的最佳体验。

(4) 分离内容和表现。可以修改甚至彻底改变网站的内容或表现(即设计),但不会因为一方改动而影响另一方。