

CSS Mastery

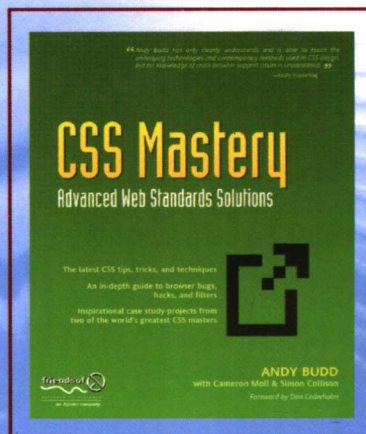
Advanced Web Standards Solutions

精通 CSS

高级 Web 标准解决方案

[英] Andy Budd 著
Cameron Moll
Simon Collison
陈剑瓯 译

- Amazon 第一 CSS 畅销书
- 最有用的 CSS 技术汇总
- 解密业界大师绝技



TURING

图灵程序设计丛书

精通CSS

高级Web标准解决方案

CSS Mastery: Advanced Web Standards Solutions

Andy Budd
[英] Cameron Moll 著
Simon Collison

陈剑瓯 译



人民邮电出版社
POSTS & TELECOM PRESS

版 权 声 明

Original English language edition, entitled *CSS Mastery: Advanced Web Standards Solutions* by Andy Budd, Cameron Moll, and Simon Collison, published by Apress L.P., 2560 Ninth Street, Suite 219, Berkeley, CA 94710 USA.

Copyright © 2006 by Andy Budd, Cameron Moll, and Simon Collison. Simplified Chinese-language edition copyright © 2006 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Apress L.P.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。

献给我漂亮的女友 Melanie,
感谢你过去 9 个月来给予的爱和支持。
—Andy Budd

献给妈妈、爸爸、Emma 和 Agenzia,
感谢你们的支持。
—Simon Collison

献给 Suzanne 和孩子们,
我不能没有你们。
—Cameron Moll

序

在网页设计的精彩世界里，实现同一个目标有无数的方法，而且新的方法还在不断地出现。对于特定的问题没有惟一正确的解决方法，丰富的选择使我们这些网页设计人员受益良多，同时也困扰着我们。这些选择虽然能使网页设计变得有趣，但同时也带来了很大的麻烦。本书将帮助你减少麻烦，理清头绪。

Andy Budd 多年来一直在编写、设计和宣讲基于标准的网页设计，我们现在有幸在本书中看到他以简洁清晰的方式讲授最重要的 CSS 技术。本书提供了一套网页设计人员不可或缺的解决方案、技巧和经验。

有些图书中仅仅提出一种实现某一目标的正确方法，我很不喜欢这样的做法，Andy 所做的正好相反，他为各种任务（如对链接应用样式、创建标签页式导航和创建多列布局等）提供了多种方法。掌握了常见设计元素的这些时髦漂亮的设计方法，你就可以更好地做出选择。

不只如此，Andy 还邀请两位出色的设计人员将这些技术组合在一起，通过两个实例研究向我们展示了这些基本技术如何组合在一起。我长期以来一直是 Cameron 和 Simon 的崇拜者，看到这两位分别讨论流体布局、无懈可击的布局和灵活的样式解决方案的实例研究，真的是受益匪浅。

好了，现在开始深入研究各种设计方法，祝你早日成为精通 CSS 的高手。

Dan Cederholm

于马萨诸塞州 Salem 市

(Dan 是 *Web Standards Solutions* 一书的作者)

前言

尽管 CSS 资源的数量越来越多，但是在 CSS 邮件列表上仍然反复看到同样的问题：如何让设计居中？最好的圆角框技术是什么？如何创建三列布局？如果你熟悉 CSS 设计社区，那么往往会想起提供了某篇文章或某种技术的网站。但是，如果你是 CSS 的初学者，或者没有时间阅读所有博客，那么这些信息可能并不好找。

在 CSS 的某些比较晦涩的方面（比如定位模型和特殊性），即使是有经验的 CSS 开发人员也会遇到问题。这是因为大多数 CSS 开发人员是靠自学的，他们从文章和别人的代码中学习经验，而没有全面理解规范。而且许多人认为，CSS 规范是复杂的并且常常自相矛盾，是为浏览器厂商而不是为网页开发人员编写的。

此外，还有浏览器问题。浏览器的 bug 和不一致性是现代 CSS 开发人员面对的最大问题之一。不幸的是，许多 bug 都没有良好的文档记录，它们的修复方法基本上只是在开发人员之间口口相传。你必须知道以某种方式做某件事，否则在某种浏览器中就会出问题。但是，你记不住是哪种浏览器以及为什么会出问题。

所以，就需要有一本书将最有用的 CSS 技术汇总在一起，并且集中介绍现实的浏览器问题，从而弥补人们 CSS 知识中的欠缺。本书会帮助你加快学习 CSS 的过程，使你的代码很快达到 CSS 专家的水平。

读者对象

本书适合具有(X)HTML 和 CSS 基本知识的任何人¹。无论你是刚刚接触 CSS 设计，还是已经开发纯 CSS 站点好几年了，本书中都有适合你的内容。如果你已经使用 CSS 一段时间了，但还不是专家，那么你能够从本书获得最大的收益。本书充满了实用的现实建议和示例，可以帮助你精通现代 CSS 设计。

本书结构

本书的前两章讨论基本的 CSS 概念和最佳实践，帮助你轻松地入门。你将学习如何建立代

1. 如果你不具备这些基础知识，可以阅读人民邮电出版社即将出版的《HTML XHTML 与 CSS 基础教程》（第 6 版）。——编辑注

码结构和加注释、CSS 定位模型的细节以及浮动和清理的工作原理。你可能已经掌握了其中的许多内容，但是可能会发现自己有遗漏或理解不充分的地方。因此，前两章是不错的 CSS 入门材料，也可以帮助你回顾已经知道的知识。

介绍了基本知识之后，后面五章讨论核心 CSS 技术，比如图像、链接、列表操纵、表单设计、数据表格设计以及纯 CSS 布局。每一章都由浅入深，直到建立比较复杂的示例。在这些章节中，你将学习如何创建圆角框、带透明阴影的图像、标签页式导航条和 flickr 风格的翻转。如果你想研究本书中的示例，可以从 www.turingbook.com 或 www.friendsofed.com 下载所有示例代码。

浏览器 bug 是许多 CSS 开发人员最头疼的问题，所以本书中的所有示例都关注创建跨浏览器的技术。不只如此，本书还用两章讨论招数 (hack)、过滤器、bug 和 bug 修复。在这两章中，你将学习一些最常用的过滤器，什么时候应该使用它们以及什么时候应该避免使用它们。还要学习 bug 捕捉技术，以及如何在 bug 造成问题之前发现并修复它们。甚至还会学习是什么造成了 Microsoft IE 中许多看似无规律的 CSS bug。

最后两章是真正的“大餐”。Simon Collison 和 Cameron Moll 是两位最杰出的 CSS 设计人员，他们将本书讨论的许多技术组合成两个实例研究。所以你不但将学习这些技术的工作原理，而且要学习如何将它们用在实际项目中。

本书可以从头到尾地阅读，也可以放在计算机旁边作为提示、技巧和技术的参考资料。这由你决定。

本书约定

本书使用几个需要注意的约定。本书采用以下术语：

- (X)HTML 指 HTML 和 XHTML 这两种语言。
- 除非特别声明，CSS 是指 CSS 2.1 规范。
- IE 5.x/Win 指 Windows 的 IE 5.0 和 IE 5.5。
- Windows 的 IE 6 和更低版本指 Windows 的 IE 5.0~6.0。

为了方便，假设本书中的所有(X)HTML 示例都嵌套在一个有效文档的<body>中，同时，CSS 包含在文档的<head>中。有时候为了简短，(X)HTML 和 CSS 放在同一个代码示例中。但是在真实的文档中，这些代码需要放在各自的位置上才能正常工作。

```
p {color: red;}
```

```
<p>I'm red</p>
```

最后，对于包含重复数据的(X)HTML 示例，并不给出每一行，而是使用省略号 (...) 表示代码延续：

```
<ul>
<li>Red</li>
<li>Yellow</li>
<li>Pink</li>
<li>Green</li>
...
</ul>
```

了解了这些约定之后，我们就可以上路了吧。

致谢

Andy Budd

感谢所有直接或间接地帮助我们撰写本书的人。

感谢 Chris 在我撰写本书的过程中一直引导着我，帮助我将想法变成现实。感谢所有不知疲倦地帮助本书按时出版的 Apress 工作人员。你们的奉献精神和专业精神令人敬佩。

感谢我在 Clearleft (www.clearleft.com)、Jeremy Keith (www.adactio.com) 和 Richard Rutter (www.clagnut.com) 的朋友和同事，他们在我撰写本书的过程中提供了鼓励和反馈意见。

感谢 Molly E. Holzschlag 允许我在本书中借用你的知识和经验。你的支持和指导是无价的，真不知道你是如何管理时间的。

感谢 Jamie Freeman 和 Jo Acres 为提高我的技能提供了完美的环境。我随时都有茶点可以享用。还要感谢布莱顿网页开发社区，尤其是 BNM 和 SkillSwap 邮件列表上的所有人。

感谢我的同事与我分享他们的网页开发经验和知识。如果没有下面各位以前的工作，本书是不可能完成的：Cameron Adams、John Allsopp、Nathan Barley、Holly Bergevin、Douglas Bowman、The BritPack、Dan Cederholm、Tantek Çelik、Joe Clark、Andy Clarke、Simon Collison、Mike Davidson、Garrett Dimon、Derck Featherstone、Nick Fink、Patrick Griffiths、Jon Hicks、Shaun Inman、Roger Johansson、Ian Lloyd、Ethan Marcotte、Drew McLellan、Eric Meyer、Cameron Moll、Dunstan Orchard、Veerle Pieters、D. Keith Robinson、Jason Andrew Andrew Santa Maria、Dave Shea、Ryan Sims、Virtual Stan、Jeffrey Veen、Russ Weakley、Simon Willison 和 Jeffrey Zeldman。

感谢我的博客的所有读者和过去一年中我在各种会议、讨论会和培训活动中遇到的所有人。你们的意见和思想丰富了本书的内容。

特别感谢 Mel 校对了每一章，而且在过去 9 个月里一直容忍着我的反复修改。

最后，感谢你阅读本书，希望本书能够帮助你将 CSS 技能提升到新的层次。

Cameron Moll

我想感谢所有帮助我编写实例研究的人。尤其感谢 Ryan Parman，Tuscany 布局的右上角使

用了他的 `TIMEDATE` 脚本生成日期。可以从 www.skyzyx.com/scripts/ 下载他的脚本。

感谢 **Veer** 提供了这个布局中使用的精美图像。如果没有他们的帮助, **Tuscany Luxury Resorts** 就会单调多了。不知道为什么, **Veer** 总是能够提供独特的视觉元素——照片、字型等, 这真是不同寻常。可以在 www.veer.com/ 访问他们的收藏。

Simon Collison

我必须感谢 **Jon Burgerman** (www.jonburgerman.com)、**Richard May** (www.richard-may.com) 和我在 **Black Convoy** (www.blackconvoy.com) 的所有朋友, 他们允许我使用他们的图像和名称, 并且为我的实例研究贡献了他们的才能。还要特别感谢瑞典的 **Roger Johansson** (www.456bereastreet.com), 他允许我使用他的圆角框并且在去年夏天款待了我。**More Than Doodles** 设计能够如此迅速高效地完成要感谢 **ExpressionEngine** (www.expressionengine.com) 出版平台中的模板系统, 这是我离不开的一个工具。最后, 感谢 **Agenzia** (www.agenzia.co.uk) 的朋友们, 你们容忍我近期狂热的写作行为。感谢所有人。

目 录

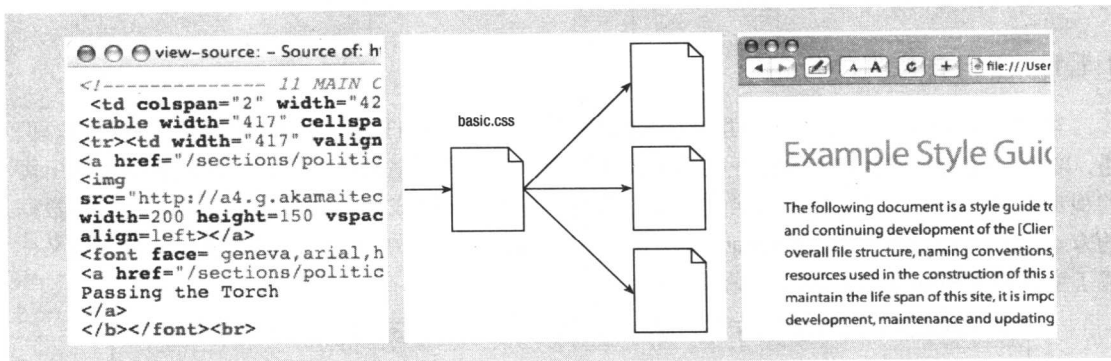
第 1 章 基础知识	1	3.2.2 山顶角	44
1.1 设计代码的结构	2	3.3 阴影	46
1.1.1 使用有意义的标记	2	3.3.1 简单的 CSS 阴影	46
1.1.2 文档类型、DOCTYPE 切换和浏览器模式	7	3.3.2 来自 Clagnut 的阴影方法	49
1.2 为样式找到目标	9	3.3.3 模糊阴影	49
1.2.1 常用的选择器	10	3.3.4 洋葱皮阴影	52
1.2.2 通用选择器	11	3.4 图像替换	54
1.2.3 高级选择器	11	3.4.1 FIR	55
1.2.4 层叠和特殊性	14	3.4.2 Phark	55
1.2.5 继承	16	3.4.3 Gilder/Levin 方法	56
1.3 规划、组织和维护样式表	17	3.4.4 IFR 与 sIFR	56
1.3.1 对文档应用样式	17	3.5 小结	58
1.3.2 对代码进行注释	18	第 4 章 对链接应用样式	59
1.3.3 样式指南	20	4.1 简单的链接样式	59
1.3.4 组织样式表以便简化维护	21	4.2 让下划线更有趣	60
1.4 小结	22	4.3 突出显示不同类型的链接	62
第 2 章 可视化格式模型	23	4.4 创建按钮和翻转	65
2.1 框模型概述	23	4.4.1 简单的翻转	66
2.1.1 IE/Win 和框模型	25	4.4.2 具有图像的翻转	66
2.1.2 空白边叠加	25	4.4.3 Pixy 样式的翻转	67
2.2 定位概述	27	4.5 已访问链接样式	68
2.2.1 视觉格式化模型	27	4.6 纯 CSS 工具提示	70
2.2.2 相对定位	28	4.7 小结	71
2.2.3 绝对定位	29	第 5 章 对列表应用样式和创建导航条	73
2.2.4 浮动	31	5.1 基本列表样式	74
2.3 小结	36	5.2 创建垂直导航条	75
第 3 章 背景图像和图像替换	37	5.3 在导航条中突出显示当前页面	77
3.1 背景图像基础	37	5.4 创建水平导航条	78
3.2 圆角框	39	5.5 简化的“滑动门”标签页式导航	80
3.2.1 固定宽度的圆角框	40	5.6 CSS 图像映射	82
		5.7 远距离翻转	90

5.8 对于定义列表的简短说明	93	8.1.2 明智地使用招数	135
5.9 小结	94	8.2 过滤单独的样式表	136
第 6 章 对表单和数据表格应用样式	95	8.2.1 IE 的有条件注释	136
6.1 对数据表格应用样式	95	8.2.2 带通过滤器	137
6.1.1 表格特有的元素	97	8.3 过滤单独的规则和声明	139
6.1.2 数据表格标记	98	8.3.1 子选择器招数	139
6.1.3 对表格应用样式	99	8.3.2 属性选择器招数	140
6.1.4 添加视觉样式	100	8.3.3 星号 HTML 招数	141
6.1.5 添加其他样式	101	8.3.4 IE/Mac 注释反斜线招数	142
6.2 简单的表单布局	101	8.3.5 转义属性招数	142
6.2.1 有用的表单元素	102	8.3.6 Tantek 的框模型招数	143
6.2.2 基本布局	103	8.3.7 修改后的简化框模型招数	143
6.2.3 其他元素	104	8.3.8 !important 和下划线招数	144
6.2.4 修饰	106	8.3.9 Owen 招数	144
6.3 复杂的表单布局	107	8.4 小结	145
6.3.1 可访问的数据输入元素	108	第 9 章 bug 和 bug 修复	147
6.3.2 多列复选框	109	9.1 bug 捕捉	148
6.3.3 表单反馈	111	9.2 bug 捕捉的基本知识	150
6.4 小结	113	9.2.1 隔离问题	151
第 7 章 布局	115	9.2.2 创建基本测试案例	152
7.1 让设计居中	115	9.2.3 修复问题, 而不是修复症状	153
7.1.1 使用自动空白边让设计居中	116	9.2.4 请求帮助	153
7.1.2 使用定位和负值空白边让 设计居中	117	9.3 拥有布局	153
7.2 基于浮动的布局	118	9.3.1 什么是布局	154
7.2.1 两列的浮动布局	118	9.3.2 布局有什么效果?	154
7.2.2 三列的浮动布局	121	9.4 常见 bug 及其修复方法	156
7.3 固定宽度、流体和弹性布局	122	9.4.1 双空白边浮动 bug	156
7.3.1 流体布局	122	9.4.2 3 像素文本偏移 bug	156
7.3.2 弹性布局	124	9.4.3 IE 6 重复字符 bug	159
7.3.3 弹性-流体混合布局	126	9.4.4 IE 6 躲躲猫 bug	160
7.3.4 流体和弹性图像	127	9.4.5 相对容器中的绝对定位	160
7.4 faux 列	129	9.4.6 停止对 IE 的批评	161
7.5 小结	132	9.5 小结	162
第 8 章 招数和过滤器	133	实例研究 1 More Than Doodles	163
8.1 招数和过滤器简介	134	CS1.1 关于这个实例研究	164
8.1.1 关于招数和过滤器的一个警告	134	CS1.2 用后代选择器控制内容区域	165
		CS1.2.1 XHTML	165
		CS1.2.2 三列布局	166

CS1.2.3 两列布局	167
CS1.2.4 单列布局	168
CS1.2.5 隐藏不需要的列	168
CS1.3 对列进行浮动	169
CS1.3.1 计算	169
CS1.3.2 将列浮动到正确的位置	170
CS1.4 根据主体类突出显示当前页面	172
CS1.5 列的 drop-in 框	173
CS1.6 直角和圆角——自己决定	174
CS1.6.1 一般的直角	174
CS1.6.2 为特殊效果做准备	175
CS1.7 透明的定制角和边框	175
CS1.7.1 图像	176
CS1.7.2 CSS	177
CS1.8 使用类组合来确定操作目标	178
CS1.9 使用图像类和例外	180
CS1.9.1 默认图像	180
CS1.9.2 拥有的图像	180
CS1.9.3 比较大的图像	181
CS1.10 处理链接	183
CS1.10.1 了解边栏链接	183
CS1.10.2 带符号的已访问链接	183
CS1.10.3 突出显示外部链接	185
CS1.11 浮动的阴影(图库)	186
CS1.11.1 形成阴影	186
CS1.11.2 对图像进行浮动	187
CS1.12 小结	188
实例研究 2 Tuscany Luxury Resorts	189
CS2.1 关于这个实例研究	189
CS2.2 流体布局	190
CS2.2.1 主体和容器	192
CS2.2.2 报头	192
CS2.2.3 内容和边栏	193
CS2.2.4 页脚	195
CS2.2.5 解决流体布局的问题	196
CS2.3 使用绝对定位对准元素	197
CS2.3.1 位置属性(top、bottom、left、right)	197
CS2.3.2 堆放次序(z-index)	199
CS2.4 背景图像技术	200
CS2.4.1 将上半部分分成三份	201
CS2.4.2 使背景“无懈可击”	202
CS2.5 图像替换	204
CS2.5.1 徽标图像替换	204
CS2.5.2 首字母大写图像替换	206
CS2.6 流体图像	207
CS2.7 为多个元素使用一个列表项	209
CS2.8 小结	212

第 1 章

基础知识



人类天生就是一种好奇的动物。我们都很喜欢摆弄事物。这不，最近我得到了一台新的 iMac G5，还没看说明书呢，自己就先把它摆弄了一番。我们喜欢自己摆弄东西，对它的作用和使用方式形成自己的看法。我们会自己胡乱设想一阵子，直到有什么东西出了错或者出乎我们的预料，才会去查阅手册。

学习 CSS（层叠样式表）最好的方式之一是直接开始使用它。但是，如果不小心的话，那么就可能误解重要的概念，或者在日后造成问题。在本章中，将讲解一些基本但常常被误解的概念，并讨论如何让(X)HTML 和 CSS 保持清晰且结构良好。

当使用 XHTML 这个术语时，指的是 Extensible Hypertext Markup Language；当使用 (X)HTML 这个术语时，指的是 XHTML 和 HTML。

在本章中，你将学习：

- 文档结构良好且有意义的重要性。
- 编码的最佳实践。
- 常见的编码错误。
- 文档类型、DOCTYPE 切换和浏览器模式。
- 调整样式的方式。
- 层叠、特殊性和继承。

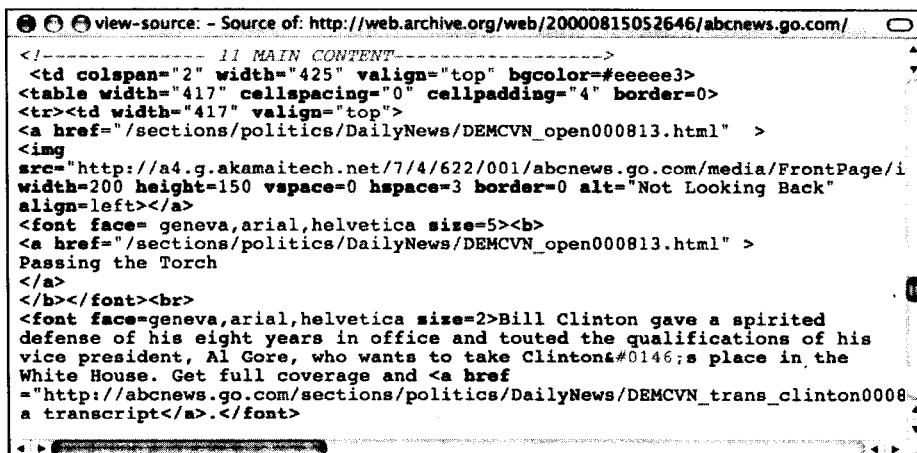
1.1 设计代码的结构

大多数人不关心建筑物的地基。但是,如果没有坚固的地基,建筑物的主体也就不会存在了。虽然本书讨论的是高级的 CSS 技术,但是如果没有结构良好且有效的(X)HTML 文档,那么我们要做的许多事情都是不可能实现的(或者说实现起来是非常困难的)。

在本节中,你将明白为什么结构良好且有意义的(X)HTML 文档在 CSS 开发中非常重要,还将学习如何在文档中增加更多的意义,从而让自己的开发工作更轻松。

1.1.1 使用有意义的标记

早期的 Web 仅仅是一系列相互链接的研究文档,并使用 HTML 添加基本的格式和结构。但是,随着万维网的流行,HTML 开始用来表现页面。人们使用字体和粗体标签的组合来创建所需的视觉效果,而不只是用标题元素突出显示页面的标题。表格成了一种布局工具而不是显示数据的方式,人们使用块引用(blockquote)来添加空白而不是表示引用。Web 很快就失去了意义,成了字体和表格标签的大杂烩(见图 1-1)。



```
view-source: - Source of: http://web.archive.org/web/20000815052646/abcnews.go.com/
<!------- 11 MAIN CONTENT----->
<td colspan="2" width="425" valign="top" bgcolor=#eeeeee3>
<table width="417" cellspacing="0" cellpadding="4" border=0>
<tr><td width="417" valign="top">
<a href="/sections/politics/DailyNews/DEMCVN_open000813.html" >
</a>
<font face= geneva,arial,helvetica size=5><b>
<a href="/sections/politics/DailyNews/DEMCVN_open000813.html" >
Passing the Torch
</a>
</b></font><br>
<font face=geneva,arial,helvetica size=2>Bill Clinton gave a spirited
defense of his eight years in office and touted the qualifications of his
vice president, Al Gore, who wants to take Clinton's place in the
White House. Get full coverage and <a href
="http://abcnews.go.com/sections/politics/DailyNews/DEMCVN_trans_clinton0008
a transcript</a>.</font>
```

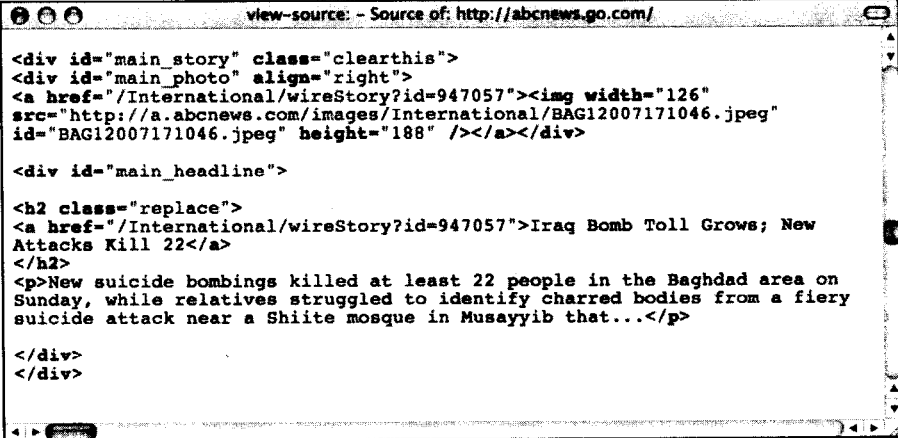
图 1-1 来自 abcnews.com (2000 年 8 月 14 日) 的新闻头条页面的标记,它使用表格进行布局,对标题使用大的粗体字。代码缺乏结构,很难理解

HTML 原本是一种简单且容易理解的标记语言。但是,随着网页变得越来越复杂,代码变得几乎不可能理解了。因此,需要使用复杂的 WYSIWYG (所见即所得) 工具来处理大量的无意义标签。不幸的是,这些工具并没有使事情简化,反而添加了它们自己的复杂标记。新千年开始以来,即使普通的网页也变得非常复杂,以致于几乎不可能进行手工编辑而不破坏代码。看起来不能这样下去了。

就在这种情况下, CSS 出现了。有了 CSS, 就可以控制页面的外观并且将文档的表现部分与

内容分隔开。表现标签（比如字体标签）可以去掉，而且可以使用 CSS 而不是表格来控制布局。标记重新变得简单了，人们又开始对底层代码感兴趣了。

意义重新回到了文档中。浏览器的默认样式可以被覆盖，所以可以将某些内容标为标题，而不需要为它指定大的、加粗的、难看的字体。可以创建列表，而这些列表不一定显示为一系列带黑点的列表项，可以使用没有相关联样式的块引用。开发人员开始按照(X)HTML 元素的原义使用它们，无需去管它们的外观（见图 1-2）。



```
<div id="main_story" class="clearthis">
<div id="main_photo" align="right">
<a href="/International/wireStory?id=947057"></a></div>

<div id="main_headline">

<h2 class="replace">
<a href="/International/wireStory?id=947057">Iraq Bomb Toll Grows; New
Attacks Kill 22</a>
</h2>
<p>New suicide bombings killed at least 22 people in the Baghdad area on
Sunday, while relatives struggled to identify charred bodies from a fiery
suicide attack near a Shiite mosque in Musayyib that...</p>

</div>
</div>
```

图 1-2 今年早些时候从 abcnews.com 获得的新闻头条页面的标记，它具有良好的结构，容易理解。虽然它仍然包含一些表现标记，但是与图 1-1 中的代码相比有了显著的改进

有意义的标记为开发人员提供了几个重要的好处。与表现性的页面相比，有意义的页面更容易处理。例如，假设需要修改页面中的一个引用。如果这个引用加上了正确的标记，那么很容易搜索代码，找到第一个块引用元素。但是，如果这个引用只是另一个段落元素标签，那么就很难寻找了。

除了人容易理解之外，程序和其他设备也可以理解有意义的标记（也称为语义标记）。例如，搜索引擎可以识别出标题行（因为它被包围在 h1 标签中）并给它分配更高的重要度。屏幕阅读器的用户可以依靠标题作为辅助的页面导航措施。

对于本书来说更重要的是，有意义的标记提供了将元素调整为你所需样式的简单方法。它在文档中添加了结构并且创建一个底层框架。可以直接设置元素的样式，而不需要添加其他标识符，因此避免了不必要的代码膨胀。

(X)HTML 包含丰富的有意义元素，比如：

- h1、h2 等。
- ul、ol 和 dl。

- strong 和 em。
- blockquote 和 cite。
- abbr、acronym 和 code。
- fieldset、legend 和 label。
- caption、thead、tbody 和 tfoot。

因此, 当存在意义合适的元素时, 就应该使用。

1. ID和类名

有意义的元素提供了很好的基础, 但是可用元素的列表并不全面。(X)HTML 是作为简单的文档标记语言创建的, 而不是界面语言。因此, 没有用于内容区域或导航栏等的专用元素。虽然可以使用 XML 创建自己的元素, 但是由于太复杂, 这在目前还不太现实。

次优的解决方案是使用现有的元素, 并且通过添加 ID 或类名给它们赋予额外的意义。这会在文档中添加额外的结构, 并给样式提供有用的“钩子”。因此, 可以建立一个简单的链接列表, 并且给它分配 ID mainNav, 从而创建出定制的导航元素:

```
<ul id="mainNav">
  <li><a href="#">Home</a></li>
  <li><a href="#">About Us</a></li>
  <li><a href="#">Contact</a></li>
</ul>
```

使用 ID 名标识页面上的单独元素 (比如站点导航), ID 必须是惟一的。ID 可以用来标识持久的结构性元素, 例如主导航或内容区域。ID 还可以用来标识一次性元素, 例如某个链接或表单元素。

在整个站点上, ID 名应该应用于概念上相似的元素以避免混淆。从技术上说, 如果联系人表单和联系人详细信息在不同的页面上, 那么可以给它们分配同样的 ID 名 contact。但是, 如果以后需要根据每个元素的上下文环境设置它们的样式, 那么就会遇到问题。所以, 使用不同的 ID 名 (比如 contactForm 和 contactDetails) 就会简单得多。

一个 ID 名只能应用于页面上的一个元素, 而同一个类名可以应用于页面上任意数量的元素。类非常适合标识内容的类型或相似的条目。例如, 假设有一个新闻页面, 其中包含每条新闻的日期。不必给每个日期分配不同的 ID, 而是可以给所有日期分配类名 date。

在分配 ID 和类名时, 一定要尽可能保持名称有意义并与表现方式无关。例如, 可以给导航元素分配 ID rightHandNav, 因为你希望它出现在右边。但是, 如果以后将它的位置改到左边, 那么 CSS 和(X)HTML 就会不同步。所以, 将这个元素命名为 subNav 或 secondaryNav 更合适。这种名称解释了这个元素是什么, 而没有涉及如何表现它。对于类名, 也是这样的。即使你希望所有错误消息以红色显示, 也不要使用类名 red, 而应该选择更有意义的名称, 比如 error 或 feedback (见图 1-3)。

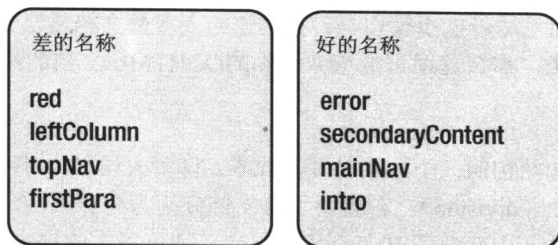


图 1-3 好的 ID 名和差的 ID 名

在写类名和 ID 名时，需要注意区分大小写。CSS 大体上是不区分大小写的语言。但是，在标记中实体（比如类名和 ID 名）是否区分大小写取决于标记语言是否区分大小写。如果使用 XHTML，那么类名和 ID 名是区分大小写的；如果使用常规的 HTML，那么是不区分大小写的。处理这个问题最好的方式是保持一致的命名约定。所以，如果在(X)HTML 类名中使用驼峰式大小写（camel case）¹，那么在 CSS 中也采用这种形式。

由于类具有灵活性，它们是非常强大的。同时，它们也可能被过度使用或滥用。CSS 新手常常在几乎所有东西上添加类，从而试图更精细地控制它们的样式。早期的 WYSIWYG 编辑器也倾向于在应用样式的每个地方都添加类。许多开发人员在使用编辑器生成的代码学习 CSS 时继承了这个坏习惯。这种现象称为“多类症”（classitis），在某种程度上，这和使用基于表格的布局一样糟糕，因为它在文档中添加了无意义的代码。

```
<h3 class="newsHead">Zeldman.com turns 10</h3>
<p class="newsText">
Another milestone for Jeffrey as zeldman.com turns 10 today
</p>
<p class="newsText"><a href="news.php" class="newsLink">More</a></p>
```

在前面的示例中，每个元素都通过使用一个与新闻相关的类名标识为新闻的一部分。这使新闻标题和正文可以采用与页面其他部分不同的样式。但是，不需要用这么多类来区分各个元素。可以将新闻条目放在一个部分中，并且加上类名 news，从而标识整个新闻条目。然后，可以使用层叠识别新闻标题或文本。

```
<div class="news">
<h3>Zeldman.com turns 10</h3>
<p>Another milestone for Jeffrey as zeldman.com turns 10 today</p>
<p><a href="news.php">More</a></p>
</div>
```

以这种方式删除不必要的类有助于简化代码，使页面更简洁。稍后，将讨论 CSS 选择器和为样式寻找目标。无论如何，这种对类名的过度依赖是不必要的。我常常只在不适合使用 ID 的

1. 所谓 camel case 是一种大小写约定，即组成标识符的多个单词除首词首字母小写外，其余单词首字母均大写，如图 1-3 中的 leftColumn 和 mainNav 等。