

与畅销书作者朱印宏一起参悟CSS布局之道

CSS

商业网站布局之道

- 最新的CSS技巧、诀窍和研究；
- 针对浏览器存在的Bug，深入剖析Hack和Filter之技；
- 学习经典的盒模型、布局模型、网页定位技术与CSS编码规范；
- 通过五个有代表性的综合实例体会实战开发的智慧，将超链接、导航菜单、滑动门、圆角、阴影、换肤、Tab等技术融入其中。



源代码光盘1CD

赠超值最新版

清华大学出版社

CSS中文完全参考手册

朱印宏著

CSS 商业网站布局之道

朱印宏 著

清华大学出版社
北 京

内 容 简 介

本书是一本 CSS 技术专著, 主要从布局这个角度全面、系统和深入地讲解 CSS 在标准网站布局之中的应用。很多读者经过初步的学习之后就能够使用 CSS 设计出一些漂亮的网页样式, 于是便乐在其中, 踌躇满志, 这是好事, 但千万不要自我陶醉, 因为你还未领略 CSS 的博大精深。用 CSS 容易, 难的是全部都用 CSS。CSS 的精髓是布局, 而不是样式, 布局是需要缜密的逻辑思维和系统设计的, 而样式只需要简单地编写代码或复制即可。本书正是基于这样的现实, 系统地探索了 CSS 在布局中的各种技术应用, 主要包括盒模型技术、布局模型技术、定位技术、浏览器兼容技术, 以及编码规范等, 同时结合 5 个综合实例切实帮助读者找到实战开发的体验和智慧。除此之外, 还就 CSS 开发中的一些技术难点和重点展开了详细的介绍, 例如表格、表单、图像、超链接、导航菜单、滑动门、圆角、阴影、换肤、Tab 技术等等。

本书注重基础、讲究实用、选材精当、深入浅出, 适合初中级读者学习使用, 也适合具有(X)HTML 和 CSS 基本知识的任何网页设计和开发人员阅读和参考。

本书封面贴有清华大学出版社防伪标签, 无标签者不得销售。

版权所有, 侵权必究。侵权举报电话: 010-62782989 13501256678 13801310933

图书在版编目(CIP)数据

CSS 商业网站布局之道 / 朱印宏著. —北京: 清华大学出版社, 2007. 10

ISBN 978-7-302-15810-3

I. C… II. 朱… 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 (2007) 第 113519 号

责任编辑: 陈 冰

责任校对: 张 剑

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

c-service@tup.tsinghua.edu.cn

社 总 机: 010-62770175 邮购热线: 010-62786544

投稿咨询: 010-62772015 客户服务: 010-62776969

印 刷 者: 北京市世界知识印刷厂

装 订 者: 三河市溧源装订厂

经 销: 全国新华书店

开 本: 203 × 260 印 张: 33.75 字 数: 878 千字

版 次: 2007 年 10 月第 1 版 印 次: 2007 年 10 月第 1 次印刷

印 数: 1 ~ 5000

定 价: 75.00 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: 025399-01

作者简介



朱印宏

资深网页设计师。从事网站开发与设计已有6年之久。对CSS、JavaScript、XHTML、XML、AJAX等网页前台技术有着深入的研究，并对这些技术的应用有着独到的见解；精通ASP和ASP.NET，能够融合前台和后台技术，实现网站混合开发。作者发表过Web设计文章十几篇，出版过6本Web技术专著。其中，《完美网页设计》系列、《ASP.NET全程开发实例》和《ASP.NET基础与实例教程》等图书深受读者欢迎和喜爱，图书多次重印，所写图书严谨、扎实，语言通俗易懂。个人所建立的www.css8.cn网站专注于Web标准技术与信息咨询。

前 言

CSS 是 Cascading Style Sheet 的英文缩写，中文翻译为层叠样式表单。通过这个意思，我们可以明白层叠性是 CSS 的一个基本特性。但是很多开发人员错误地把 CSS 与样式等同起来。

实际上，CSS 技术的核心是布局，而不是样式。CSS 的强大生命力在于它的布局能力，而不是细枝末节的修饰功能。当然，CSS 在样式设计方面的功能是非常完美的。如果读者受制于错误概念的影响，不能超脱现有的篱笆，会给自己的学习与实践背上很重的包袱。俗话说得好“站得高，望得远”，只有从布局的高度来学习 CSS 技术，读者才能抓住 CSS 的本质，才能在 CSS 实践中应用自如。

尽管目前有关 CSS 的学习资源的数量越来越多，但多是一些支离破碎的技巧和经验，学习这样的资料很容易产生身处山中却不得山貌的困惑。另一方面，CSS 的一些高级技术是比较晦涩的，例如，盒模型、布局模型、定位模型、CSS 特殊性和兼容性，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 开发人员是靠自学成才的，他们从网上文章和别人的代码中学习经验，而没有全面系统的理解 CSS 规范。

此外，浏览器兼容性问题和 CSS 开发工具的缺乏都会影响 CSS 的开发效率。浏览器对 CSS 的解析差异是 CSS 开发人员所面临的巨大挑战，很多设计师不得不花费大量时间来修补 CSS 代码的兼容性，由于没有系统的参考资料，这些经验基本上都是摸石头过河，严重影响开发速度。

因此，本书将从 CSS 最基础知识起步，全面系统的探索 CSS 核心功能，将最有用的 CSS 技术汇总在一起，全面弥补读者在 CSS 知识中的欠缺，帮助你加快学习 CSS 的进程，使你的代码很快能够达到 CSS 专家的水平。

读者对象

全书从最基本的概念开始讲起，步步深入，因此，即使你是完全的新手，对 (X)HTML 和 CSS 还没有任何概念，你也完全可以看懂本书。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。本书充满了实用的经验技术和示例，可以帮助你精通 CSS 技术。

本书专注于 CSS 技术，系统、全面和深入是本书的特色，因此，本书适用于所有网站设计师以及网站开发人员。全书力求用通俗的语言、简洁的结构、系统的内容与读者一起探索 CSS 技术，并以 CSS 布局为重点介绍 CSS 全部的应用技巧。

全书结构

全书共分 13 章，包括两大部分：第一部分侧重于 CSS 技术本身，全面系统的讲解 CSS 技术的方方面面；第二部分侧重于综合实例解析，帮助读者积累 CSS 开发实战经验。本书内容采用知识+实例的方式，一步步引导读者如何开始新的、符合 Web 标准的 CSS 布局设计。由于 CSS 布局

设计从本质上与传统的表格式布局有着截然不同的开发方式，因此本书的重点将放在如何转变思路 and 如何实际操作等问题上。

第 1 章，侧重于现实的困惑，介绍 Web 标准布局的历史发展，帮助读者找准方向，为后面学习奠定思想基础。

第 2 章，是 CSS 基础知识入门，系统的讲解了 CSS 语法，是初学者必须认真阅读的内容，后面所有内容都将以此为基础展开介绍。

第 3 章，介绍了 CSS 开发中两个最重要的编辑工具，熟练掌握其中一种 CSS 开发工具，可以帮助读者节省很多代码书写时间。

第 4 章，介绍 CSS 盒模型技术，盒模型是 CSS 布局的基础，所有布局都是建立在该技术之上，对于初学者来说应该算是一个技术难点。其中，盒模型本身比较容易理解，困难的是盒模型之间的关系不好把握，需要一定的经验积累。

第 5 章，介绍 CSS 版式设计，这一章主要从最基本的样式介绍起，涉及到一些开发技巧，但学习难度不是很大。

第 6 章，介绍 CSS 布局模型技术，布局是 CSS 的难点，也是一个重点，里面包含很多实用技术，如浮动、定位、布局、结构等。读者不容易学习的原因在于，一是布局本身比较抽象，有些效果不容易直接观察；二是浏览器的兼容性会给读者的学习和开发带来障碍；三是不同类型的元素混合布局时，会出现很多意想不到的问题。

第 7 章，介绍 CSS 兼容性技术，本章对于初学者来说，在学习上会存在一定的困难，不过当你初步掌握 CSS 布局之后，你会发现它非常有用，且具有相当的参考价值。

第 8 章，完全从开发实践的角度介绍如何统筹、规划 CSS 编码，以及如何优化 CSS 代码。学习中，需要读者结合实际开发过程，这样能够更容易理解和领会其中的意思。

从第 9 章开始，分 5 章介绍了 5 个综合实例的开发思路和过程，力求帮助读者体验不同类型网站开发的一般步骤和方法。在实例中，重在介绍 CSS 的构思和开发方法，并对其中一些技术难点展开讲解。

通过阅读本书，读者能够获得全新的网站设计思考方式、全新的 XHTML+CSS 开发方式，帮助你创建更具创造性、高效性，以及易于维护与访问的网站。

阅读提示

在写作中，全书采用循序渐进的方式，从最基本的概念讲解，并不断深入，对于没有基础的读者来说，建议不要跳跃式阅读，只有在领会前面内容的基础上，你才能够更好地学习和理解后面的知识。如果你有一定的 CSS 基础，可以把本书作为设计开发的参考手册，它将帮助你排解实践中遇到的各种问题。

本书为了使读者更好地理解与掌握 CSS 布局方法，在大部分章节之中，除了提供知识、实例外，还从网络上收集了在此方面具有独到之处的网站，就这些现实存在的网站在这些设计问题上的实现方法进行解析，以帮助读者快速理解并行之有效的应用到自己的工作之中。

为了帮助读者学习参考，本书还提供了最新的 CSS 中文完全参考手册，帮助读者随时查找在学习或开发中所用到的 CSS 语法和属性的详细解释。

本书介绍了大量的 XHTML 和 CSS 代码，用以帮助读者学习 CSS 布局，但由于篇幅所限，

不能够把书中每个示例的所有代码都放置在书中，因此示例代码基本上只显示了三部分：

- XHTML 结构代码，即网页中的 body 部分代码。
- CSS 布局代码，即网页中的 style 标签内代码。

网页其他部分代码被省略，显示如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>标题</title>
<style type="text/css">
—在这里输入书中显示的 CSS 样式或布局代码
</style>
</head>
<body>
—在这里输入书中显示的 XHTML 结构或框架代码
</body>
</html>
```

因此，读者在上机练习时，需要补全 XHTML 主体框架代码。如果不明白，可以直接参考光盘实例。在实际学习时，需要使用完整的 XHTML 代码结构，才能够让浏览器顺利解析，读者可以自己编写或借助于网页编辑器完成。

而另一方面，由于实例中的代码有不少内容过于复杂繁多，为了帮助读者更清晰的阅读，实例中的部分代码使用...省略号省略了不重要或重复性的代码。

目前 CSS 在兼容性上仍然存在着差异。其中比较明显的是 IE 5.5 及以下版本与 IE 6 和 Firefox 的差异，而 IE 6 与 Firefox 之间也有部分解析差异。因此，在本书示例中，会一直考虑代码的浏览器兼容问题。

目前浏览器市场主要是 IE 6 与 Firefox，但国内的用户 90% 以上都使用 IE 6 浏览器，因此，本书中的所有实例都以 IE 6 为试验平台，保证大部分用户可简单在 IE 6 平台上实现，同时也会兼顾 Firefox 以及其他类型的浏览器，对于 IE 5.5 及以下版本的浏览器，由于用户人数有限，且其本身也并不是以 Web 标准为核心的浏览器，因此不是重点试验平台，只有涉及到兼容技术时，本书才会介绍并显示不同 IE 版本的浏览效果。

本书在 CSS 布局的解读上，完全从 CSS 布局的思考方式入手，并不针对浏览器之间的兼容性技巧进行编写，因此，读者只要理解了 CSS 布局的思考方式和编写原理，就能够真正的了解 CSS 的布局技术，而这些浏览器之间的差异，大家在学习和应用的过程中也能够慢慢体会到。在本书的专门章节——CSS 浏览器兼容解决方案中，我们将深入探讨如何解决这些兼容性问题，以及对浏览器的解析行为做出分析，帮助读者更透彻的理解 CSS 布局。

致谢

本书写作历时半年之久，为了给读者贡献一本高质量的 CSS 技术专著，很多人为此而默默无

闻地提供了各种直接或间接的帮助。

特别要感谢我的妻子和孩子，为了支持我的写作，为我节省了大量的宝贵时间，同时还要忍受每个深夜里灯光和噪音的干扰，每当我处于困境和面临压力时，是她们给予我无微不至的生活照顾，并鼓励我勇敢地坚持下来。

感谢我的同事姜德元、杨兴国、张敏、袁江、陈万林等，是他们承担了更多的工作，才使得我能有更多的写作时间。

最后，我要感谢本书的策划编辑陈冰先生，是他提出了很多有建设性的意见，使得本书的内容更充实、更实用。同时，还要感谢为本书的出版付出努力的文稿编辑、校对编辑，及其他相关工作人员。

本书作者

本书由朱印宏编写。同时，参与资料整理及编写的还有常才英、袁祚寿、袁衍明、张敏、袁江、田明学、唐荣华、毛荣辉、卢敬孝、刘玉凤、李坤伟、旷晓军、陈万林、陈锐、周秀成、颜昌学、王幼平、冉原洲、李经键、胡厚成等，在此对大家的辛勤工作表示衷心的感谢！

由于时间有限，书中难免会有疏漏和不足之处，恳请广大读者提出宝贵意见。有关本书的任何问题请发电子邮件到 zhuyinhong@263.net，本书支持信息将会及时发布在 <http://www.css8.cn> 站点上。

朱印宏

2007年7月于北京

目 录

第 1 章 Web 标准布局概述	1	2.3.5 伪类和伪对象	42
1.1 从闪存帝国网站改版说起	1	2.3.6 伪类选择符	45
1.2 表格布局的神话与没落	2	2.3.7 伪对象选择符	46
1.2.1 表格为谁而生	3	2.3.8 分组选择符	47
1.2.2 表格为什么要没落	3	2.3.9 包含选择符	48
1.3 从 Table 到 CSS——坎坷的历程	8	2.3.10 元素指定选择符	49
1.3.1 现实的无奈	8	2.3.11 子对象选择符	50
1.3.2 勇敢的挑战	9	2.3.12 属性选择符	51
1.4 认识 Web 标准	10	2.3.13 相邻选择符	55
1.4.1 结构标准语言	10	2.4 CSS 属性	55
1.4.2 表现标准语言	11	2.5 CSS 属性值	58
1.4.3 行为标准语言	11	2.5.1 颜色值	58
1.5 CSS 布局的基本思路	12	2.5.2 绝对单位	59
1.6 我能学好 CSS 布局吗	14	2.5.3 相对单位	59
1.7 制作你的第一个 XHTML+CSS 页面	16	2.5.4 百分比	60
1.7.1 传统表格布局	16	2.5.5 URL	60
1.7.2 表格布局存在的问题	19	2.6 在网页中应用 CSS	61
1.7.3 XHTML+CSS 标准布局	19	2.6.1 内联样式	61
第 2 章 CSS 布局基础	24	2.6.2 内部样式	62
2.1 标准网页结构	24	2.6.3 外部样式	62
2.1.1 XHTML 文档结构	24	2.7 CSS 的继承、层叠和特殊性	64
2.1.2 XHTML 基本语法	26	2.7.1 继承	64
2.1.3 XHTML 元素分类	27	2.7.2 层叠	65
2.1.4 XHTML 常用元素简介	28	2.7.3 特殊性	66
2.1.5 XHTML 结构语义化研究	32	2.8 div 和 span 深入研究	70
2.1.6 XHTML 结构化应用案例分析	33	2.8.1 通用元素 div 和 span	70
2.2 CSS 基本语法	35	2.8.2 使用 div 元素	74
2.3 CSS 选择符	36	2.8.3 关于 div 元素嵌套的思考	76
2.3.1 类型选择符	37	2.8.4 使用 span 元素	79
2.3.2 id 选择符	38	2.9 id 和 class 深入研究	81
2.3.3 class 选择符	40	2.9.1 使用 id	81
2.3.4 通用选择符	42	2.9.2 关于 id 和 name	83
		2.9.3 使用 class	85

2.9.4 关于 id 和 class	88	4.5.2 解决非 IE 父元素不能自适应子元素高度问题	152
第 3 章 CSS 布局工具	89	4.5.3 解决 IE 子元素底边界不被解析问题	154
3.1 TopStyle	89	4.5.4 解决 IE 子元素溢出问题	155
3.1.1 认识 TopStyle 界面	89	第 5 章 CSS 格式化排版	157
3.1.2 创建样式	90	5.1 排版是个细活	157
3.1.3 编辑样式	92	5.2 文字排版	159
3.1.4 应用样式	95	5.2.1 关于字体	159
3.1.5 环境设置及使用技巧	96	5.2.2 关于字体大小	162
3.1.6 测试样式	97	5.2.3 增强字体大小的灵活性和可读性	166
3.2 Dreamweaver	98	5.2.4 关于字体颜色	167
3.2.1 CSS 样式面板	98	5.2.5 关于字体样式	168
3.2.2 属性面板	103	5.3 段落排版	170
3.2.3 CSS 可视化助理	104	5.3.1 缩进	170
3.2.4 CSS 样式呈现	108	5.3.2 对齐	172
3.2.5 CSS 视图辅助工具	112	5.3.3 行距	175
3.2.6 浏览器检查	113	5.3.4 分栏	176
3.2.7 验证标记	114	5.3.5 列表版式基础	178
第 4 章 CSS 盒模型	116	5.3.6 列表版式实战	180
4.1 盒模型基础	116	5.3.7 列表项行内显示	184
4.1.1 为什么要提出盒模型	116	5.3.8 列表符号的个性显示	185
4.1.2 盒模型结构	117	5.3.9 字距	188
4.1.3 关于元素宽和高的计算	118	5.3.10 文本行控制	188
4.1.4 关于元素并列间距的计算	121	5.3.11 空白	192
4.1.5 关于元素间垂直间距的计算	124	5.3.12 文本流控制	193
4.1.6 关于空盒模型	127	5.3.13 文本转换	197
4.2 边框	127	5.3.14 文本修饰	197
4.2.1 边框宽度	129	5.4 图文混排	199
4.2.2 边框颜色	129	5.4.1 构建图文混排结构	199
4.2.3 边框样式	130	5.4.2 图文规则混排	202
4.2.4 关于内联元素的边框	132	5.4.3 背景图像应用	204
4.3 边界	134	5.4.4 图文不规则混排	206
4.3.1 边界值	134	5.4.5 不规则图文混排的再研究	209
4.3.2 边界重叠	140	第 6 章 CSS 布局模型	212
4.3.3 内联元素的边界	147	6.1 关于 CSS 布局	212
4.4 补白	148		
4.5 盒模型兼容性应用	151		
4.5.1 解决 IE 元素浮动时边界误差问题	151		

6.1.1 从网友的求帖说起	212	7.3.2 @import 规则过滤器	280
6.1.2 关于盒模型补充	213	7.4 过滤规则和声明	283
6.1.3 盒模型的类型	214	7.4.1 !important 关键字过滤器	283
6.1.4 包含块	216	7.4.2 下划线属性过滤器	283
6.1.5 布局模型	220	7.4.3 转义属性过滤器	284
6.2 流动布局模型	221	7.4.4 *html 选择符过滤器	285
6.2.1 流动布局模型的特征	222	7.4.5 属性选择符过滤器	285
6.2.2 相对定位流动	223	7.4.6 子对象选择符过滤器	286
6.3 浮动布局模型	225	7.4.7 相邻选择符过滤器	286
6.3.1 浮动布局模型的特征	225	7.4.8 转义选择符过滤器	286
6.3.2 浮动清除	230	7.4.9 注释反斜杠过滤器	287
6.3.3 浮动嵌套	233	7.5 在 IE/Windows 中隐藏规则和声明	287
6.3.4 浮动与流动嵌套	234	7.5.1 隐藏单个声明	287
6.3.5 浮动与流动混合布局应用	236	7.5.2 隐藏多个声明	289
6.3.5.1 调整左右栏之间 的空隙	236	7.5.3 隐藏规则	289
6.3.5.2 调整上下栏之间 的空隙	238	7.6 网页 Bug 解决方案	290
6.3.5.3 混合布局沙盘推演	242	7.6.1 W3C CSS 验证服务	290
6.4 层布局模型	246	7.6.2 Web Developer 工具插件	291
6.4.1 定位类型	246	7.6.3 浏览器显示模式	293
6.4.2 绝对定位包含块	246	7.6.4 代码隔离与验证	296
6.4.3 绝对定位元素的边距	251	7.7 常见 Bug 及其修复	297
6.4.4 相对定位	253	7.7.1 盒模型 Bug	297
6.4.5 混合定位	253	7.7.2 双倍浮向 Bug	298
6.5 层叠等级	255	7.7.3 多出 3 像素 Bug	299
6.6 高度自适应	259	7.7.4 高度不适应 Bug	301
6.7 CSS 基本布局类型	261	7.7.5 多余字符 Bug	303
6.7.1 单行单列	262	7.7.6 定位 Bug	304
6.7.2 单行两列	262	7.7.7 捉迷藏 Bug	305
6.7.3 单行三列	265	7.7.8 部分区域隐藏 Bug	306
6.7.4 多行多列	268	7.7.9 百分比 Bug	308
6.7.5 伪列布局	272	7.7.10 项目符号隐藏 Bug	309
第 7 章 CSS 浏览器兼容解决方案	275	7.8 HTML 默认样式与浏览器解析差异	311
7.1 关于网页浏览器	275	7.8.1 HTML 4 默认样式	311
7.2 CSS Bug、Hack 和 Filter 简介	276	7.8.2 浏览器默认样式	313
7.3 过滤样式表文件	278	7.9 IE 7 浏览器兼容解决方案	315
7.3.1 IE 条件注释	279	7.9.1 盒模型溢出兼容解决方案	315
		7.9.2 IE 7 不再支持的过滤器小结	317
		7.9.3 推荐使用的过滤器	318

第 8 章 CSS 文档统筹与编码规范	320
8.1 CSS 文档统筹.....	320
8.1.1 根据页面类型分离文件.....	321
8.1.2 根据功能模块分离文件.....	321
8.1.3 根据标签类型分离文件.....	321
8.1.4 根据设备类型分离文件.....	322
8.1.5 根据代码规模综合分离文件.....	322
8.2 规则的组织.....	322
8.3 属性的组织.....	325
8.4 CSS 命名艺术.....	327
8.4.1 经典命名三法.....	327
8.4.2 CSS 命名规则.....	328
8.4.3 CSS 命名方法.....	329
8.5 CSS 代码缩写规则.....	331
8.5.1 盒模型代码简写.....	331
8.5.2 列表和背景缩写.....	334
8.5.3 颜色值缩写.....	335
8.5.4 字体缩写.....	335
8.6 CSS 代码书写格式.....	336
8.6.1 CSS 代码常用书写格式.....	336
8.6.2 CSS 代码清道夫.....	339
8.7 CSS 代码注释艺术.....	340
8.7.1 怎样写好注释.....	340
8.7.2 预防 CSS 中文注释 Bug.....	342
8.7.3 CSS 注释清除.....	343
8.8 CSS 代码优化技巧.....	344
8.8.1 利用继承性优化代码.....	344
8.8.2 利用默认值优化代码.....	345
8.8.3 利用公共类优化代码.....	345
8.8.4 利用选择符分组优化代码.....	346
8.8.5 利用层叠覆盖优化代码.....	347
第 9 章 个人 Blog 网站布局	348
9.1 关于 Veerle's blog 的研究.....	348
9.2 网站配色.....	351
9.3 全局属性设置.....	351
9.4 控制模块显示宽度.....	353
9.5 关于兼容性的解决方案.....	356
9.6 滑动门技术研究.....	358
9.6.1 什么是滑动门.....	359
9.6.2 滑动导航菜单.....	362
9.7 欣赏与研究背景图像的应用.....	365
9.7.1 主背景图像、艺术图像和 图标的使用.....	365
9.7.2 日历背景图像的使用.....	366
9.7.3 背景图像应用的思考.....	367
9.7.3.1 隐藏法.....	367
9.7.3.2 躲避法.....	368
9.7.3.3 覆盖法.....	368
9.8 圆角研究.....	369
9.8.1 不带背景图像的圆角.....	369
9.8.2 关于圆角的高级探讨.....	376
9.8.3 纯 CSS 圆角代码优化.....	381
9.8.4 用背景图像设计圆角.....	383
9.8.4.1 完全固定的圆角布局.....	383
9.8.4.2 高度固定的圆角布局.....	384
9.8.4.3 宽度固定的圆角布局.....	385
9.8.4.4 完全自适应圆角布局.....	387
9.8.4.5 优化完全自适应圆角 布局代码.....	390
9.8.5 关于圆角解决方案的更多 参考信息.....	391
9.9 阴影研究.....	393
9.9.1 定位法.....	393
9.9.2 浮动法.....	394
9.9.3 洋葱皮.....	395
第 10 章 公司宣传网站的布局	396
10.1 关于 Layout 的研究.....	396
10.2 全局属性设置.....	399
10.3 页面框架控制.....	400
10.4 关于列表布局的深入研究.....	403
10.4.1 图片切换技术.....	403
10.4.2 选项卡切换技术.....	406
10.5 导航菜单研究.....	408
10.5.1 水平菜单.....	409
10.5.2 垂直菜单.....	411

10.5.3 多级菜单	413	12.2 关于读客网的研究	462
10.5.4 CSS Tab Designer	420	12.3 全局属性设置	465
10.5.5 获取更多的导航菜单	421	12.4 页面框架控制	470
第 11 章 技术门户网站布局	423	第 13 章 网上商城网站布局	475
11.1 关于 CSS Beauty 的研究	423	13.1 关于阿里巴巴	475
11.2 全局属性设置	426	13.2 关于阿里巴巴中国网站的研究	476
11.3 页面框架控制	429	13.3 全局属性设置	481
11.4 关于 CSS Beauty 导航条的研究	432	13.4 页面框架控制	483
11.5 关于 CSS Beauty 数据表格的研究	434	13.5 网页换肤技术研究	486
11.5.1 Upcoming Events 数据表格	434	13.5.1 构思与准备	487
11.5.2 Current Job Openings 数据表格	436	13.5.2 设计皮肤	488
11.6 数据表格深入研究	438	13.5.3 用 JavaScript 控制皮肤	490
11.6.1 数据列分组	440	13.5.4 记忆皮肤	492
11.6.2 数据行分组	442	13.5.5 恢复皮肤记忆	494
11.6.3 表格中重要的排版属性	443	13.5.6 分析阿里巴巴中国网站 首页皮肤	495
11.7 数据表格的 CSS 布局研究	445	13.6 Tab 切换技术	502
11.7.1 边框控制	445	13.7 表单研究	504
11.7.2 改善数据表格的视觉效果	449	13.7.1 表单及其框架	504
第 12 章 Web 2.0 网站布局	457	13.7.2 表单版式	510
12.1 关于 Web 2.0 的技术探析	457	13.7.3 表单布局	518

第 1 章 Web 标准布局概述

“评价一种 Web 开发技术优劣的标准只有一个，那就是看这种技术能否在最恰当的时间和最恰当的地点，以最恰当的方式，为最需要信息的人提供最恰当的信息。”（摘自 2004 年 4 月《程序员》的《Web 开发技术史话》）

我很欣赏王咏刚先生的上面这段话。他说这段话的本意是就当时许多搞技术的人似乎已经忘记了 Web 架构的设计初衷，在自己开发的网站或 Web 应用中大肆堆砌各种所谓的先进技术，但最终用户能够在这些网站或应用中获得的有价值信息却寥寥无几。

在讨论 Web 标准布局之前，再来反思王先生的这段话，感觉 CSS 的发展正应了先生的观点，纵览今日中国的 Web 技术发展，虽然还存在不同的声音，但普及和使用 Web 标准及 CSS 技术已是一个不争的事实。

1.1 从闪客帝国网站改版说起

国内信息技术的发展似乎总慢于西方半个节拍，其他不说，但至少在 CSS 技术上是这样的。CSS 第一个正式规范早在 1996 年就已经发布了，也就是说 1996 年 W3C 组织（World Wide Web Consortium，万维网联盟）正式推出了 CSS 1.0 版本技术标准，到了 1998 年 CSS 2.0 版本又面世了。而我们的 Web 设计师们在新世纪钟声之后，才慢慢嗅觉到 CSS 技术的先进性，并开始在网站中小心翼翼地进行尝试。

W3C 组织是一个专门负责网络标准制定的非赢利组织，诸如 HTML、XHTML、CSS、XML 等标准就是由 W3C 制定的。根据 W3C 官方网站的介绍，W3C 会员包括生产技术产品及服务的厂商、内容供应商、团体用户、研究实验室、标准制定机构和政府部门，W3C 会员一起协同工作，致力在万维网发展方向上达成共识。关于 W3C 组织的更多信息，请读者访问 W3C 官方英文网站（<http://www.w3c.org>），或者通过万维网联盟香港办事处（<http://www.w3c.ust.hk/>，成立于 1998 年 11 月，地址在香港科技大学），以及万维网联盟中国办事处（<http://www.chinaw3c.org/index.htm>，成立于 2006 年 4 月，地址在北京航空航天大学）获取有关 W3C 的中文介绍。

CSS 的中国实践与普及当推《CSS 权威指南》一书的引进，相信很多早期的 Web 设计师都看过这本书。时间大致应该锁定在 2001 年 5 月，此后，前卫的设计师们开始在网页中局部尝试 CSS 技术。

2004 年 10 月闪客帝国（<http://www.flashempire.com/>）开始用 Web 标准对自己的网站进行重构，当时在业界产生了不小的轰动，因为这是第一个吃螃蟹的人。对于一个商业网站来说，使用未经普遍实践的技术是存在很大风险的，所幸他们很顺利的走过来。用当时网易学院采访边城浪子的话说：“网站重构是今后很多网站要面临的问题，标准化永远不会是个错误，我们知道我们早晚会走这一步。长痛不如短痛，所以，我们就下定决心了。”

闪客帝国网站于 2005 年完成了全部 CSS 布局，作为国内第一个大型应用案例，虽然在设计思想和技巧使用上不是最佳案例，但其在 CSS 布局应用上带来了很多宝贵经验。如图 1.1 所示是现在

的闪客帝国网站首页第一屏截图效果。



图 1.1

最近，浏览中央电视台官方网站（http://www.cctv.com/），发现它的网页全部被 Web 标准重构了，这实在是一个惊喜，印象中的中央官方网站都是比较严肃的，对于新技术的采用向来都很谨慎。可见 CSS 技术在国内已经得到绝大部分网页设计师的认可和喜爱。



图 1.2

左图是央视网站首页第一屏截图效果。浏览央视网站页面，虽然还存在一些技术问题，但足见 CSS 技术在国内网页设计中的影响力，相信央视网站的示范性和感召力将会更进一步刺激 CSS 技术在国内的普及与推广。

1.2 表格布局的神话与没落

表格在网页布局中应用已经有很多年了，由于多年的技术发展和经验积累，Web 设计工具功能不断增强，使表格布局在网页应用中达到登峰造极的地步。一个最典型例子就是网页设计师可以直接使用图像编辑器画图、切图，最后再由图像编辑器全自动生成表格布局的页面，像这样提供完整

而又完善的一条龙服务的网页设计工具实在很多，如设计师比较熟悉的 Adobe Photoshop、ImageReady 和 Fireworks 等。这些工具的使用都很简单、学习周期短、制作速度快，而且设计的页面效果都很漂亮。如果你是一位富有艺术感的读者，相信设计出来的页面必定很吸引眼球。

传统表格布局的快速与便捷加速了网页设计师对于页面创意的激情，而忽视了代码的理性分析，甚至刚学几天的初学者都敢说出“我可以设计出最完美的网页”，表格布局就这样被神话了。

1.2.1 表格为谁而生

提出这个疑问决非偶然，一直以来我就在不断探寻和思考表格的渊源。假设我们从网页设计这个相对独立的领域跨入另一个陌生的公司财会世界，你会大悟表格的真正使命。“表格在手，一清二楚”，这是财会人员的口头禅，记得以前看过一本书介绍，说是一企业高级白领，她最大的才干就是擅长设计表格，针对不同的工作、不同的岗位，设计出适合不同工作和岗位的表格，让不同工种和岗位的员工能将各工作更有序更高效地完成，并将管理成本降得更低。何为高级，原来是能够设计出各种表格的白领。

也许你会自以为擅长表格设计的人当属网页设计师，其实这种想法是错误的，随手翻阅高级管理人员案头的形形色色的表格，你才会明白什么是真正的表格，表格的真正价值所在。当你进入公司管理阶层工作时，你会发现若能为公司根据需要不断设计出行之有效的各类表格，那真的是一项非常重要的工作。

表格能够使繁复的数据分门别类，让人一目了然地清楚数据内在的整体情况，它比许多文字说明都要简明、有用得多。例如，如何将客户的不同模具的未付款帐单理清，要知道每一次签订制作的模具是分期付款的，同一客户的不同模具，因签订合同的时间不同，加上其他诸多因素等，其应付款和未付款的期数是各不相同的，只有用表格的形式，才能很清楚地表达不同模具应付款和未付款及所有模具总未付款的情况。

走出传统的纸制表格再来看看与网页毗邻的电子表格技术。提起电子表格，你一定会想到微软的 Excel，但你不会想到 VisiCalc。1979 年面世的 VisiCalc 是历史上第一个电子表格软件，是它直接奠定了电子表格软件的基础，VisiCalc 的发明人 Dan Bricklin 绝对是一位天才（个人主页：<http://www.bricklin.com/>），仅用 16KB 的代码就改变了整个计算机产业，直接推进了个人电脑的发展。

从界面上看，VisiCalc 完全左右了现代电子表格软件的发展。最早的 VisiCalc 是在 Apple II 系统上运行，因此给苹果公司带来了巨额利润。后来 Mitch Kapor 创办了 Louts 公司，推出了 Lotus 1-2-3，电子表格得到了更完善的发展。

如今 Lotus 1-2-3 电子表格软件也消失在用户的视野，提起电子表格大家都只知道 Excel，或者国内金山公司的电子表格。但不管电子表格软件如何发展，它们最终都是为了数据管理服务的，谁也不会想到用表格来进行设计，或用表格来排版布局。

1.2.2 表格为什么要没落

实际上，表格的功能不是用来进行布局。最初的网页没有今天这么复杂，仅显示文本或几个简单的图像，网页文档从上而下自然流动分布，不需要考虑版式设计问题。

后来随着网页内容的丰富，图像、声音、动画等多媒体不断充斥网页，网页内容不断膨胀，同

时用户对于网页视觉提出了更高要求，于是如何把传统印刷中的版式技术转移到网页设计中来就成为了一个比较紧迫的需求。

由于表格不仅可以控制单元格的宽度和高度，而且还可以嵌套，多列表格还可以把文本分栏显示，于是就有人试着在表格中放置其他网页内容，如图像、动画等，以打破比较固定的网页版式。

而网页表格对无边框表格的支持为表格布局奠定了基础，用表格实现页面布局慢慢就成为了一种设计习惯。现在网上的大多数页面都是使用表格来完成布局的，即使在 CSS 不断普及的情况下，这种状况也很难在短时间内得以改观。

表格布局是不标准的，用 W3C 制定的规范来说，表格的目的是用来显示数据的，而不是来完成布局，错把表格当布局缘于当时 Web 技术的缺乏和对标准需求的乏力。

为什么时至今日才会出现这么多人大声疾呼表格布局的非标准问题呢，为什么有这么多位设计师迫切的使用 XHTML+CSS 来重构自己的网页呢？

其实在表格布局之初，就有专家提出异议，而且那时 CSS 也已经被制定并公布于众，一套较完善的网页标准技术也已经被推向市场，但应者甚寡，微软的 IE 浏览器更是带头唱反调，标准就这样被非标准淫威了好几年，萎缩在一角默默地隐忍。

现在 Web 技术从 1.0 跨入 2.0 时代，表格布局已经不能再适应新技术发展的趋势了，同时用户对于互联网的需求上升到一个新的层次。如今的 Web 世界，用户体验是上帝，在民意需求的强势之下，标准占据了上风，连微软这样的巨无霸也不得不诚服于 W3C 所提倡的标准之下。

什么是 Web 1.0，什么是 Web 2.0 呢？

Web 2.0 在两年前也许还是一个新名词，曾几何时它像网上核弹一样爆发了，并以不可阻挡之势燃烧了整个互联网，其热度不压于当年的超女，又曾几何时它悄悄地走进了我们的生活，从陌生走向了熟悉，从概念走向了应用，今天，Web 2.0 构成了我们网络生活不可缺少的一部分，大家都在讨论、研究和实践 Web 2.0。

目前对什么是 Web 2.0 并没有很严格的定义，一般来说 Web 2.0（也有人称之为互联网 2.0）是相对 Web 1.0 的新一类互联网应用的统称。Web 1.0 的主要特点在于用户通过浏览器获取信息，Web 2.0 则更注重用户的交互作用，用户既是网站内容的消费者（浏览者），也是网站内容的制造者。

Web 1.0 到 Web 2.0 的转变，具体的说，从模式上是单纯的“读”向“写”、“共同建设”发展。所以互联网下一步，是要让所有的人都忙起来，用全民力量共同织出贴近生活的网。到目前为止，对于 Web 2.0 概念的说明，通常采用 Web 2.0 典型应用案例介绍，加上对部分 Web 2.0 相关技术的解释，例如，博客（BLOG）、播客、威客、掘客、RSS、百科全书（Wiki，也叫维客）、网摘、社会网络（SNS）、P2P、即时信息（IM）等，关于这些信息技术的解释与更多信息，读者可以参考第 12 章介绍。

Web 2.0 对于网页结构和布局提出更高要求，这不仅仅是为了节省几行代码、减少带宽的问题，实际上今天的带宽对于互联网的发展已经不是主要障碍了。要实现 Web 的交互与共建，就必须允许用户可以操纵网页，而这些都是必须建立在完全标准的网页结构之上，以及更加灵活的布局方式。

显然，Web 2.0 的这种要求与表格布局是格格不入的，如果你透过精美的表格页面，会吃惊的发现，用切图生成的网页代码如同天书，后期的维护和更新竟成了一场噩梦。例如，图 1.3 所示是千龙网千龙伊氏女性频道（<http://yessee.qianlong.com/>）的一个栏目，这是一个典型的用切图生成的页面，如果查看它的源代码，你会发现里面存在大量的冗余代码，网页结构与表现混杂在一起，如图 1.4 所示，要想在源代码中查找信息是非常困难的，更别说让用户参与网页共建之中。