

对标准化网页设计见解独特
一针见血揭示CSS技术的核心价值及应用
www.css8.cn网上讲座与答疑

精通CSS 网页布局



朱印宏 林小志 编著



附1DVD

- 共计893分钟的超强视频讲解。
- CSS、Html、JavaScript、ASP教程（CHM格式）
- 本书实例源代码



中国电力出版社
www.cepp.com.cn



精通 CSS 网页布局

朱印宏 林小志 编著



中国电力出版社
www.cepp.com.cn

内 容 提 要

本书深入、系统地讲解了使用 CSS 进行网页布局设计的相关知识和技巧,内容包括 CSS 基础、文字和版式设计、图像控制、超链接和导航菜单设计、表格和表单设计、浮动网页布局设计以及网页优化等。在书的最后,还对 CSS 框架设计进行案例分析,并通过一个购物网站设计实例来融会贯通所有知识点,以达到灵活应用的目的。

本书还附赠一张多媒体教学光盘,内含 12 课有关 CSS 的多媒体教学课件,总长度超过 13 个小时,与本书知识点基本同步,以帮助您快速了解 CSS 的核心技术,达到事半功倍的学习效果。

本书内容全面,示例丰富,讲解细致,非常适合网页设计与网站开发人员学习。

图书在版编目(CIP)数据

精通 CSS 网页布局 / 朱印宏, 林小志编著. —北京: 中国电力出版社, 2009
ISBN 978-7-5083-7962-3

I. 精… II. ①朱…②林… 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 172302 号

责任编辑: 马首鳌

责任校对: 王开云

责任印制: 郭华清

书 名: 精通 CSS 网页布局

编 著: 朱印宏 林小志

出版发行: 中国电力出版社

地址: 北京市三里河路 6 号 邮政编码: 100044

电话: (010) 68362602 传真: (010) 68316497

印 刷: 航远印刷有限公司

开本尺寸: 185mm × 260mm 印 张: 27 字 数: 680 千字

书 号: ISBN 978-7-5083-7962-3

版 次: 2009 年 2 月北京第 1 版

印 次: 2009 年 2 月第 1 次印刷

印 数: 0001—4000 册

定 价: 45.00 元(含 1DVD)

敬 告 读 者

本书封面贴有防伪标签,加热后中心图案消失

本书如有印装质量问题,我社发行部负责退换

版 权 专 有 翻 印 必 究

F

前言

Foreword

Hi, 大家好, 我是朱印宏, 自从去年 9 月份推出《CSS 商业网站布局之道》以来, 该书在广大读者之间产生了不小的轰动, 获得了 Web 前端设计圈内很多专业人士的推崇, 并连续几个月登上华储、卓越、当当网站排行榜。《CSS 商业网站布局之道》是我对 CSS 技术的思考和实践的一部分, 其实在那本书推出之后, 我仍然感觉有很多东西没有写入, 或者没有写透, 于是继续思考 CSS 的核心技术问题以及它的发展前景。可以这样说, 《CSS 商业网站布局之道》一书是我对于 CSS 深层次的理论思考, 而本书则是我在前一本书的基础上继续探索 CSS 在实践中的应用, 少了几分理论色彩, 却多了几分实践的摸索。

记得我曾经在《CSS 商业网站布局之道》一书中说过: “用 CSS 容易, 难的是全部都用 CSS”。虽然 CSS 语言比较好学习, 但是如果完全使用 CSS 技术来控制页面的表现仍然会很麻烦。于是很多人不理解, 经常在群里或“坛子”里与我辩论。例如, 蓝色经典论坛的版主 linxz 就曾与我在群里为此争论了一番:

样吧 21:55:47

用 CSS 容易, 难的是全部都用 CSS。

linxz™ 21:56:07

为什么要全部用 CSS 呢?

linxz™ 22:56:56

CSS 是用在该用的地方。你这样说太广泛了。

样吧 22:57:27

CSS 是用在该用的地方, 自然。

样吧 22:57:44

难道不需要 CSS, 非要硬塞一个样式吗?

linxz™ 22:59:39

难道用 JS, 你一定要用 CSS?

样吧 22:59:57

我没有这样说呀。

样吧 22:60:26

你非要硬塞, 不理解?

linxz™ 22:61:18

☺

大家有自己的想法这是可以理解的, 但是我这么说也有其事实根据: 第一, 枯燥的 CSS 代码远没有传统表格布局用得随心所欲; 第二, 为了使设计的页面能够兼容不同浏览器, 看似简单的 CSS 代码往往会把你弄得精疲力竭; 第三, 没有完美的可视化辅助工具, 你似乎感觉自己总处于石器时代的那种刀耕火种之中。

当然我说的是相对的, 随着 CSS 技术的不断普及和推广, 各种成熟的可视化工具大量涌现,

以及各种浏览器都支持 W3C 标准，也许到那时候，我再说这话可能就不合时宜了。

我曾经还在《CSS 商业网站布局之道》主题网站 (<http://www.css8.cn/csslayout/index.htm>) 写过这么一句话：**CSS 的精髓是布局，而不是样式，布局是需要缜密的结构分析和设计。**这句话也是我对于 CSS 思考而来的，有很多专业人士比较欣赏，当然也不乏有反对意见。记得 CSS 森林群主曾经围绕我这句话写了一篇《中国式 WEB 标准》的思考文章 (<http://www.cssforest.org/blog/index.php?id=93>)，虽然他的文章主题是谈论标准设计问题，但是这句对于 CSS 思考的话还是引起大家的共鸣。

我一直认为样式和布局应该是两个不同的概念，样式是指具体的应用效果，而布局则是网页结构的表现问题，它不仅仅是局部效果问题，还应涉及网页的框架结构。如果说学习 CSS 仅希望它能够美化一下字体样式，或者设计漂亮的段落版式，那么我认为你还没有把握住 CSS 的精髓，CSS 的精髓应该是网页的整体呈现方面，换句话说就是如何使用 CSS 把一个 HTML 结构给完美的支撑起来。说起来可能很容易，如何撑起网页的结构呢？这就涉及 CSS 核心技术问题，以及如何兼容不同浏览器问题，这远比传统表格布局中信手涂鸦要复杂许多。正是基于这样的思考，我才萌生了继续写作本书的念头。

从去年开始，国外就开始了 CSS 框架技术的试验和实践，也出现了几个比较有名的 CSS 框架技术，国内设计师还没有完全嗅到 CSS 框架的香味，唯有淘宝团队在今年开始进行 CSS 框架的摸索和实践。当然在本书上市之后，也许会有更多大型商业网站去实践 CSS 框架技术，考虑到这种技术是一种发展趋势，所以我也用了一章篇幅对此展开讲解，很多东西都是我对于 CSS 框架的思考和探索，相信未来几年内它一定会成为 CSS 技术的一个热门话题。

给本书读者的话

非常感谢您购买本书。在乱花渐欲迷人眼的 CSS 分类书架中，你能够挑选本书作为学习 CSS 技术的首选书籍，是我的荣幸。当然在准备阅读本书之前，我还是要叮嘱几句，希望你能够买得物有所值。

第一，请记住不怕慢就怕站。我不希望读者一目十行的来阅读本书，在群里经常会听到读者说 3 天翻阅一本图书的故事。当然这里可能会有两种情况：一是书的含金量确实不高，所以只能一目十行；二是你的 CSS 技术水平功力不浅，一目十行是为了搜索自己需要的知识。如果不是这两种情况，真心的建议你还是慢慢阅读本书，我整整花费了五个多月的时间来写本书，而你仅仅使用一个礼拜的时间来阅读本书，那么里面肯定会有很多金子被埋没了。

第二，请记住不怕笨就怕懒。要说到笨，其实我是最笨的学生，记得最初学习 CSS 时，居然抱住《CSS 参考手册》硬耗了 3 个月。学习最忌讳自大，但是最担心的就是发懒。抽懒筋，就是形象的描绘了人性的弱点，人发懒时，越想懒还想懒，以至于懒得抽筋。所以初学者最好多动手，对于书中得每行代码都能够动手输入试验一下，并能够有所思考。群里有位咆哮男郎的网友购买《CSS 商业网站布局之道》一书，居然用了 2~3 月的时间把书中每句话都咀嚼了一遍，每行代码都亲自动手输入试验，并帮助挑出了一些纰漏的地方汇集给我，让我非常感动。

第三，请记住不怕错就怕迷信。不要迷信任何真理，要敢于去思考，应能够举一反三，触类旁通。学习本书，不等于本书就是圣经，你要敢于去探索、穷思每个问题，不要被已有的观点束缚你的思想，只有这样你才能够走得更远，超过本书作者的水平。

最后，我把 CSS 聊吧中给初学者的一组话送给你，希望它能够成为你学习 CSS 的助推器。

- 不要被对象、属性、方法等词汇所迷惑，最根本的是先了解最基础知识。
- 不要放过任何一个看上去很简单的小问题，它们往往并不那么简单，或者可以引申出很多知识点，不会举一反三你就永远学不会。
- 知道一点东西，并不能说明你会全部，需要经验积累。
- 看再多的书是学不全的，要多实践。
- 把时髦的技术挂在嘴边，还不如把过时的技术记在心里。
- 学习的最好方法之一就是多练习。
- 在任何时刻都不要认为自己手中的书已经足够了。
- 看得懂的书，请仔细看；看不懂的书，请硬着头皮看。
- 别指望看第一遍书就能记住和掌握什么，请看第二遍、第三遍……
- 请把书上的例子亲手到电脑上实践，即使配套光盘中有源文件。
- 把在书中看到的有意义的例子扩充，并将其切实的运用到自己的工作中。
- 不要漏掉书中任何一个练习，请全部做完并记录下思路。
- 别心急，写代码确实不容易，水平是在不断的实践中完善和发展的。
- 每学到一个难点的时候，尝试着对别人讲解这个知识点并让他理解。你能讲清楚才说明你真的理解了。
- 记录下在和别人交流时发现的自己忽视或不理解的知识点。
- 保存好你做过的所有的源文件，那是你最好的积累之一。
- 对于网络，还是希望大家能多利用一下，你要学会自己找答案，如 Google、百度都是很好的搜索引擎，你只要输入关键字就能找到很多相关资料，别总是等待别人给你希望。
- 浮躁的人容易问：我到底该学什么？别问，学就对了。
- 浮躁的人容易问：CSS 有钱途吗？建议你去抢银行。
- 浮躁的人容易说：我要中文版！我英文不行！——不行？学呀！
- 浮躁的人分两种：只观望而不学的人，只学而不坚持的人。
- 浮躁的人永远不是一个高手，请不要做浮躁的人！

关于光盘的内容

本书赠送 DVD 光盘一张，其中包含：

- 893 分钟的超强视频讲解。视频与本书内容同步，辅助读者随堂、直观地学习本书核心技术。
- 本书用到的素材和网页源文件。
- CSS 完全中文参考手册、HTML 参考 CHM 文件。

本书作者

本书由朱印宏主持编写。同时，参与本书资料整理及编写的还有林小志、袁衍明、常才英、袁祚寿、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、钱佩林、苏敬波、冉东林、杨龙贵、张炜、王慧明、涂怀清、卢国才、苏恢定、司成向、胡体清、陈宗亮、徐清银、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。

技术支持

CSS 聊吧论坛: <http://www.css8.cn/css2/> (如果地址无效, 请从样吧首页链接进入)

CSS 技术交流群: 10780586 (群 1)、54430674 (群 2)、23660353 (群 3)、34678498 (群 4)、23038424 (群 5)、41091270 (群 6)

Dreamweaver 设计交流群: 6879926 (群 1)

出版社网站 (售后服务支持): <http://www.infopower.com.cn>

作者网站 (技术服务支持): <http://www.css8.cn>

作者电邮: zhuyinhong@css8.cn

作者 QQ: 364077976

朱印宏

2008 年 8 月于北京



目 录

Contents

前 言

第 1 章 CSS 布局基础 1

- 1.1 认识网页中的 CSS 1
 - 1.1.1 编写符合 CSS 布局的 HTML 结构 1
 - 1.1.2 DOCTYPE (文档类型) 与 CSS 的关系 3
- 1.2 CSS 基本语法和用法 4
 - 1.2.1 CSS 基本语法 4
 - 1.2.2 应用 CSS 样式 5
 - 1.2.3 建立外部样式表 6
 - 1.2.4 在 HTML 中导入外部样式表 7
- 1.3 CSS 的属性和属性值 8
 - 1.3.1 如何记忆 CSS 的属性 8
 - 1.3.2 CSS 的取值单位 9
- 1.4 CSS 常用选择符 10
 - 1.4.1 标签选择符 10
 - 1.4.2 类选择符 11
 - 1.4.3 ID 选择符 13
- 1.5 CSS 高级选择符 15
 - 1.5.1 认识 HTML 文档的树状结构 15
 - 1.5.2 子选择符 17
 - 1.5.3 相邻选择符 18
 - 1.5.4 属性选择符 19
- 1.6 灵活使用 CSS 选择符 23
 - 1.6.1 CSS 选择符分组 23
 - 1.6.2 通配选择符 24
 - 1.6.3 包含选择符 24
 - 1.6.4 伪类和伪元素选择符 26
 - 1.6.5 CSS 选择符的嵌套 27
- 1.7 CSS 的继承性 28
 - 1.7.1 认识 CSS 的继承性 28
 - 1.7.2 CSS 继承性的局限性 29
- 1.8 CSS 样式层叠和选择符优先级 31
 - 1.8.1 CSS 样式表的优先级 31
 - 1.8.2 CSS 样式的优先级 31
 - 1.8.3 CSS 选择符的优先级 31

第 2 章 使用 Dreamweaver 实现 CSS

布局 33

- 2.1 认识 Dreamweaver 的 CSS 布局 workflow 33
 - 2.1.1 增强 CSS 功能的【属性】面板 33
 - 2.1.2 【页面属性】对话框 35
 - 2.1.3 【CSS 样式】面板 36
 - 2.1.4 属性表 37
- 2.2 操作【CSS 样式】面板 38
 - 2.2.1 新建样式 38
 - 2.2.2 定义样式的属性 39
 - 2.2.3 应用样式 43
- 2.3 以 CSS 技术为核心的设计视图 45
 - 2.3.1 强大的 CSS 设备类型 45
 - 2.3.2 可视化助理 47
- 2.4 使用 DW 实现 CSS 布局实战 49
 - 2.4.1 使用 Dreamweaver CS3 构建页面结构 49
 - 2.4.2 使用【页面属性】统一基本样式 51
 - 2.4.3 使用【CSS 样式】面板定制页面基本布局 54

第 3 章 听话的文字和版式 56

- 3.1 看懂 Dreamweaver CS3 提供的模板 56
 - 3.1.1 善于利用示例页 56
 - 3.1.2 挖掘标准网页布局模板的价值 60
- 3.2 你的文字听话吗 62
 - 3.2.1 国际雅虎和中国雅虎 62
 - 3.2.2 选择好单位让你的文字更听话 64
 - 3.2.3 设计网页字体大小配置方案 65
 - 3.2.4 设计与字体大小相关联的网页布局 68
- 3.3 留心, 不要让你的文字“露怯” 70
 - 3.3.1 不要以为多此一举 71
 - 3.3.2 让网页字体显示更灵活 71
 - 3.3.3 别忘了通用字体 72
- 3.4 小家伙, 你向哪儿看齐 73
 - 3.4.1 文本对齐 74
 - 3.4.2 布局居中 74

3.4.3 布局元素向左、右对齐	75	5.2.3 模拟按钮样式	139
3.4.4 文本和元素垂直对齐	76	5.2.4 使用图像设计超链接样式	141
3.4.5 绝对定位元素居中显示	78	5.2.5 设计具有扩展性的超链接样式	143
3.4.6 混合结构中对齐处理	80	5.3 构建导航条列表框架及其基本样式	144
3.5 咱们比比个吧	81	5.3.1 为什么使用项目列表来构建导航条	145
3.5.1 什么样的行高更合适阅读	82	5.3.2 搭架结构合理的导航列表	145
3.5.2 行高、边距和伪行高	83	5.3.3 使用定义列表搭架导航列表	147
3.6 让网页看起来和颜悦色	85	5.3.4 控制项目列表的样式	147
3.6.1 在 CSS 中使用颜色	85	5.3.5 有趣的项目符号	148
3.6.2 CSS 布局与配色	86	5.3.6 自定义图片项目符号	149
第 4 章 扮靓网页的天使——图像	89	5.3.7 通过背景图像来定义项目符号	150
4.1 选择：前景图像和背景图像	89	5.3.8 使用行内显示法设计项目列表水平显示	151
4.1.1 前景图像	89	5.3.9 使用浮动法设计项目列表水平显示	153
4.1.2 背景图像	90	5.3.10 设计水平浮动列表的超链接样式	153
4.1.3 如何选用前景图像和背景图像	91	5.3.11 解决浮动溢出问题	154
4.1.4 走出迷失的图像 URL	91	5.3.12 使用定位法设计项目列表水平显示	155
4.2 图像边框和阴影	93	5.4 设计导航菜单	155
4.2.1 定义边距	93	5.4.1 淡雅的垂直导航菜单	156
4.2.2 巧设边距	94	5.4.2 巧妙的水平导航菜单	158
4.2.3 为图像增加阴影	96	5.4.3 背景图像在导航菜单中的应用	161
4.2.4 为 img 元素定义默认阴影样式	98	5.4.4 实用的多级菜单	164
4.2.5 设计晶莹剔透的水印	99	5.4.5 时尚的滑动门菜单	167
4.3 图文混合排版	101	第 6 章 更专业的表格和表单	170
4.3.1 单行图文定位	101	6.1 用好表格	170
4.3.2 单行图文对齐	103	6.1.1 正确使用表格	170
4.3.3 图文环绕	106	6.1.2 优化数据表格的结构	174
4.3.4 调整图文环绕的间距	108	6.1.3 设计表格样式	176
4.3.5 不规则图文环绕	108	6.2 表格样式设计实战	178
4.4 隐形的翅膀——背景图像	111	6.2.1 设计表格行样式	179
4.4.1 控制背景图像的重复显示	111	6.2.2 设计表格列样式	180
4.4.2 背景图像水平平铺应用示例	112	6.2.3 设计鼠标经过时表格行的样式	180
4.4.3 背景图像垂直平铺应用示例	113	6.3 用好表单	182
4.4.4 精确定位背景图像	114	6.3.1 认识表单	182
4.4.5 固定背景图像在浏览器中的位置	117	6.3.2 设计表单样式	184
4.4.6 如何增加多个背景图像	118	6.4 表单样式设计实战	186
4.4.7 设计更酷的圆角	120	6.4.1 设计高亮显示当前表单样式	187
4.5 背景图像布局应用——伪列布局	124	6.4.2 设计图标样式的表单效果	188
第 5 章 活泼的超链接和导航菜单	132	6.4.3 设计易用性表单效果	190
5.1 轻松掌握超链接样式	132	第 7 章 浮动的网页布局	194
5.1.1 怪异的伪类	132	7.1 CSS 的“水立方”模型	194
5.1.2 超链接的四种状态	133	7.1.1 盒模型	194
5.1.3 为超链接绑定不同的样式	134	7.1.2 盒模型解疑	195
5.1.4 为页面定义不同的超链接样式	135	7.1.3 盒模型的外边距	196
5.2 花枝招展的超链接样式	137		
5.2.1 为超链接定义普通样式	137		
5.2.2 避免影响版面晃动的超链接样式	138		

7.1.4 行内元素的外边距	197	8.2.2 绝对定位的坐标系	253
7.1.5 块状元素的外边距	198	8.2.3 绝对定位的相对论	255
7.1.6 浮动元素的外边距	198	8.2.4 相对定位参照物和坐标系	256
7.1.7 绝对定位元素的外边距	199	8.3 定位布局中元素层叠处理	257
7.1.8 盒模型的内边距	200	8.3.1 定位元素的层叠顺序	258
7.1.9 盒模型的边框	202	8.3.2 定位元素与文档流的层叠关系	259
7.1.10 盒模型的宽和高	203	8.3.3 定位元素的层叠包含关系	260
7.2 构建符合标准的网页结构	205	8.4 CSS 定位布局实战	262
7.2.1 是内容决定结构, 还是表现决定结构	205	8.4.1 米老鼠卡通画册	262
7.2.2 选择好结构标签	207	8.4.2 禅意花园展室	267
7.2.3 研究禅意花园的网页结构	207	8.4.3 蓝色的多瑙河	272
7.3 CSS 浮动布局基础	211	第9章 兼容性网页布局	278
7.3.1 认识网页布局类型	211	9.1 智能的 IE 条件语句	278
7.3.2 准备有趣的模块浮动游戏	211	9.1.1 认识 IE 条件语句	278
7.3.3 ①②③顺序的水平布局	212	9.1.2 IE 条件语句基本用法	279
7.3.4 ③②①顺序的水平布局	212	9.1.3 IE 条件语句在实践中应用	282
7.3.5 ②①③顺序的水平布局	213	9.2 探析 IE 浏览器渲染网页布局的特性	286
7.3.6 ③①②顺序的水平布局	214	9.2.1 认识 IE 浏览器的渲染特性——Layout	286
7.3.7 ①-②③结构布局	216	9.2.2 操控 IE 元素的 Layout 特性	287
7.4 CSS 浮动布局高级技术	217	9.2.3 网页元素的 Layout 特性	287
7.4.1 探究浮动元素的空间大小	217	9.2.4 行内元素的 Layout 特性	290
7.4.2 探究浮动元素的移动位置	218	9.2.5 与 Layout 特性相关的 IE 解析 Bug 和解决方法	290
7.4.3 探究浮动元素的环绕关系	220	9.2.6 能够更好兼容不同浏览器	293
7.4.4 浮动与清除	222	9.2.7 Layout 元素与包含浮动元素的关系	293
7.4.5 浮动元素包含其他对象的问题及其解决方法	224	9.2.8 Layout 元素与相邻浮动元素的关系	295
7.4.6 浮动元素被其他对象包含的问题及其解决方法	225	9.2.9 Layout 元素与列表元素的关系	296
7.4.7 浮动布局中元素垂直间距问题以及解决方法	227	9.2.10 Layout 元素与定位元素的关系	299
7.4.8 浮动布局中元素水平间距问题以及解决方法	228	9.3 浏览器兼容技术解析	300
7.4.9 浮动布局中浮动元素并列错位问题以及解决方法	230	9.3.1 你的浏览器符合标准吗	300
7.5 CSS 浮动布局实战	232	9.3.2 兼容 IE 6 版本浏览器	302
7.5.1 深红色咖啡馆	232	9.3.3 兼容 IE 5 系列版本浏览器	303
7.5.2 阳光灿烂喜洋洋	237	9.3.4 兼容 IE 7 版本浏览器	305
第8章 精确的网页布局	243	9.3.5 兼容 FF 等标准浏览器	305
8.1 CSS 精确布局的定位原理	243	9.4 深入剖析 CSS 层叠布局	306
8.1.1 认识 position	243	9.4.1 认识 CSS 层叠包含框	306
8.1.2 静态定位	244	9.4.2 CSS 层叠包含框嵌套关系	308
8.1.3 绝对定位	245	9.4.3 IE 在解析 CSS 层叠关系时的 Bug	309
8.1.4 相对定位	245	9.4.4 探析 CSS 层叠负值以及 IE 存在 Bug	310
8.1.5 固定定位	248	9.5 IE 浏览器常见 Bug 及其解决方法	312
8.2 定位布局中参照物和坐标系	250	9.5.1 超链接设计中存在的 Bug	312
8.2.1 绝对定位的参照物	250	9.5.2 IE 元素的内容与背景分离显示的 Bug	313
		9.5.3 IE 6 躲躲猫 Bug	314
		9.5.4 IE 6 多余字符 Bug	315

9.6 兼容性网页布局实战	316	第 11 章 CSS 框架设计及其案例解析	376
9.6.1 三列等高布局新法及其非 IE 存在的 Bug 和解决方法	316	11.1 CSS 框架设计	376
9.6.2 三列浮动布局中 IE 的垂直空隙 Bug 及其兼容方法	320	11.1.1 什么是 CSS 框架	376
第 10 章 网页结构化布局与实施	323	11.1.2 分析 CSS 框架的优缺点	377
10.1 设计网页基本结构	323	11.1.3 如何设计 CSS 框架	378
10.1.1 网页基本结构的设计思路	323	11.2 CSS 框架的抽象性及其设计	378
10.1.2 设计符合 SEO 的网页结构	326	11.2.1 认识 CSS 的类样式	379
10.2 单列版式布局	327	11.2.2 CSS 类样式的实施策略	379
10.2.1 单列版式的布局结构和设计方法	328	11.2.3 通用 CSS 类样式设计	380
10.2.2 单列固定宽度版式设计	330	11.2.4 CSS 框架中元素默认样式设计	383
10.2.3 单列固定宽度版式实施	332	11.2.5 CSS 框架中其他类型选择符样式的处理	386
10.2.4 江南水乡意象画布局设计	334	11.3 CSS 框架的可维护性及其设计	387
10.2.5 江南水乡意象画布局实战	337	11.3.1 CSS 框架的注释技巧及其可读性	387
10.3 单列液态框架版式布局	340	11.3.2 CSS 框架中类名和 ID 名的可读性	388
10.3.1 使用 CSS 设计框架结构	340	11.3.3 CSS 框架代码的排版格式应简明直观	389
10.3.2 使用 CSS 设计单列液态布局	343	11.3.4 框架内样式代码的结构顺序应符合一定的阅读习惯	391
10.4 两列版式布局	346	11.4 CSS 框架的可扩展性和兼容性及其设计	391
10.4.1 两列版式的布局结构和设计方法	347	11.4.1 CSS 的继承性及其利用	391
10.4.2 两栏浮动版式中兼容固定和自适应宽度的设计方法和探索	349	11.4.2 CSS 框架的继承与包含关系及其应用	393
10.4.3 两栏浮动版式中自适应宽度的设计方法和探索	351	11.4.3 通过样式表的分类、组织实现 CSS 框架的可扩展性	395
10.4.4 两栏浮动版式中兼容性宽度的设计方法和探索	353	11.4.4 CSS 框架的可兼容性	397
10.4.5 两栏固定版式的设计方法和探索	354	11.5 CSS 框架案例解析	397
10.4.6 黑色理想网页布局实施实战	355	11.5.1 解析 Elements CSS 框架	398
10.5 三列版式布局	361	11.5.2 解析 Blueprint CSS 框架	401
10.5.1 三列版式的布局结构和设计方法	361	第 12 章 购物网站结构与布局实施	409
10.5.2 解析 Dreamweaver CS3 提供的绝对定位布局及存在问题	362	12.1 设计购物网站的基本结构	409
10.5.3 三列液态版式的设计新方法和探索	364	12.1.1 网站内需分析	409
10.5.4 三列固定版式设计方法和探索	366	12.1.2 设计网站的基本结构	411
10.5.5 一列液态两列固定版式设计方法和探索	367	12.1.3 完善语义化结构	413
10.5.6 二列液态一列固定版式设计方法和探索	369	12.2 使用 CSS 完成基本结构的布局	415
10.5.7 山鹰之美网页布局实战	370	12.2.1 实施网页基本布局	415
		12.2.2 设计滑动门菜单	417
		12.2.3 设计圆角区域	419

CSS 布局基础

CSS 是 Cascading Style Sheets 的简写，中文翻译的意思为层叠样式表。在网页设计中，CSS 与 HTML、JavaScript 并列为网页前端设计的三种基本语言。其中，CSS 负责设计网页的表现效果，HTML 负责构建网页的基本结构，JavaScript 负责开发网页的交互效果。

CSS 布局是 CSS 语言的高级使命，它与我们常说的网页样式略有不同，布局涉及网页整体结构的呈现，而样式则过多关注局部效果的显示。由于 CSS 布局涉及 CSS 的核心技术，Web 设计师需要吃透 CSS 布局原理，能灵活统筹规划，并能兼容主流浏览器，因此其开发的难度也是可想而知的。本章是全书的基础，也是你准备学习 CSS 语言的基础，所以在系统讲解 CSS 基础的同时，也会略有侧重，避免在简单知识上重复叙述。

1.1 认识网页中的 CSS

如今当你随意查看一个页面中的源代码时，会看到类似下面的代码段：

```
<style type="text/css">
```

```
.....
```

```
</style>
```

和

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

这些正是本书要讲解的 CSS 语言在网页中的“家”，更准确地讲就是 CSS 源代码在网页中分隔符和引用指针。

CSS 语言不需要编译，也不需要特殊的处理，你只要把它们放在<style>和</style>标签之间，或者单独存储在一个文本文件之中，然后保存到扩展名为.css 的文件，最后利用<link>标签链接或导入到网页中即可，使用就是这么简单。

1.1.1 编写符合 CSS 布局的 HTML 结构

学习本书之前你需要有初步的 HTML 语言基础，并能够认识常用的 HTML 标签。其实这对于任何一个初学网页制作的读者来说，不应该存在什么障碍了。但是如果你习惯于传统网页制作方式，用表格来切图，然后生成网页，那么本节内容还需要你认真地阅读。

网页都是由标签组成的，这正如一张地图上的各种提示性标签，它标明了每个点所指示的方位、地理名称等。把这些标签串连在一起，才能够在你的大脑中形成一幅清晰的地图。而网页标签不仅要指明网页内容在网页结构中的位置关系和显示效果，更重要的是还要标明网页内

容的意思，即所谓的 HTML 标签的语义性。

例如，在下面这个简单的网页文档中，每个标签都包含一个字符串，但是它们所表示的意思是不相同的。

```
<!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>网页标签</title>
</head>
<body>
<h2>标题</h2>
<p>段落</p>
<ul>
<li>列表项目</li>
<li>列表项目</li>
</ul>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td>数据表</td>
<td>数据表</td>
</tr>
</table>
</body>
</html>
```

针对上面的 HTML 文档结构，从语义角度简单说明如下：

(1) <html>标签表示网页文档的意思，也可以说是 HTML 文档源代码的分隔符。

(2) <head>标签表示网页头部信息。

(3) <body>标签表示网页主体信息。

在<body>标签中包含了不同语义的标签。

1) <h2>标签表示二级标题信息。

2) <p>标签表示段落文本信息。

3) 和标签表示项目列表信息。

4) <table>、<tr>和<td>标签表示数据表信息等。

但是以表格为核心的传统网页布局中，设计师似乎对于各种标签所代表的语义特征并不感兴趣。设计师所追求的目标就是如何把网页设计得更精美漂亮。结果<table>、<tr>和<td>标签就被供奉为网页布局的三剑客，标题和段落文本全部被放置在<p>标签中，然后通过属性设置来控制标题和段落文本的显示效果。

如今表格布局不再使用了，CSS 布局被广泛使用，这是很自然的事情，但是对于习惯于传统布局思维，并且熟悉操作传统布局工具的设计师来说，所肩负的担子很重。

学习使用 CSS 进行布局，能够帮助你设计出更简洁的 HTML 结构。因为你不再需要在 HTML 文档中使用各种修饰性的标签和属性。整个页面只剩下构建页面结构的框架标签以及表达一定语义的服务性标签。整个文档的源代码变得简单了许多。

如果你还在继续使用如下标签，建议今后尽量不要再使用了：

(1) 不使用标签来设置文本样式。

(2) 不使用和<i>标签来设置文字粗体和斜体。如果确实需要使用可以选用和

`<em>` 标签来表示粗体和斜体。

(3) 不使用 `<table>` 标签进行布局, `<table>` 仅负责显示表格式的数据。

(4) 不要在 `<body>` 标签内通过各种属性来定义网页基本属性, 如 `background` (背景图像)、`bgcolor` (背景色)、`text` (文本)、`link` (链接)、`leftmargin` (左页边距)、`topmargin` (顶部页边距) 等。

(5) 尽量不使用 `<br>` 换行标签, 传统布局中习惯使用 `<br>` 和空格符号来设计多段文本, 换行排版等, 现在可以借助各种语义化标签和 CSS 来控制, 基本上不用再使用 `<br>` 了。

对于标准网页设计的初学者来说, 我们希望你一步步扎实学习, 但是下面几个简单的、符合标准的标签及其用法一定要先记住, 然后在此基础上不断去积累。

(1) `<div>`: 网页结构化标签, 它负责划分网页结构, 并负责包含不同的模块。

(2) `<span>`: 行内文本和对象包含标签, 它负责修饰行内对象样式。

(3) `<p>`: 负责管理段落文本内容的组织和管理。

(4) `<h>`: 标题标签, 根据重要性分为 6 个级别, `h1` 表示一级标题, 一般页面中只最好只包含一个一级标题。

(5) `<ul>`、`<ol>` 和 `<li>`: 项目列表标签, 主要负责网页内同类信息的列表。

1.1.2 DOCTYPE (文档类型) 与 CSS 的关系

如果在 Dreamweaver CS3 中新建网页文档, 你将会在新文档的首行看到 DOCTYPE 声明。DOCTYPE 是文档类型的简写, 它定义当前文档的基本类型。

其实 DOCTYPE 只是一组机器可读的规范, 虽然中间包含了文件的 URL, 但浏览器不会去读取这些文件, 仅用于识别, 然后决定以什么样的规范去执行页面中的代码。其具体格式如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

DTD 表示文档类型的定义, 它们定义 XML、XHTML 和 HTML 的特定的某一个版本中可以有, 不可以有什么, 在载入网页的时候, 浏览器会用既定的声明规范去检查页面的内容, 是不是有效, 然后采取相应的措施与编码解释文档中的代码。

那它们是如何工作的? 我们不去深究, 这对于初学者来说未免有点困难。但是你必须知道在标准网页布局中这是非常必要的。缺少它, 页面就可能出现一些异常的情况。因此, 当你开发符合标准的页面时, 要必须为 HTML 文档指定一种 DOCTYPE, 否则 CSS 渲染可能会出错。

XHTML 1.0 提供了三种 DTD 声明可供选择, 简单说明如下。

(1) 过渡型 (Transitional): 要求非常宽松的 DTD, 它允许用户继续使用 HTML4.01 的标签, 但是要符合 XHTML 的写法。完整代码如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

(2) 严格型 (Strict): 要求严格的 DTD, 用户不能使用任何表现层的标签和属性, 如 `<br>`。完整代码如下:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

(3) 框架型 (Frameset): 专门针对框架页面设计使用的 DTD, 如果你的页面中包含有框架, 就需要采用这种 DTD。完整代码如下:


```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

一般情况下，建议读者选择过渡型文档类型，这种 DTD 比较宽松，也比较容易通过 W3C 的代码校验，比较适合目前国内开发环境和大多数开发人员的水平。当然你的目标是向着严格型文档类型去努力。

此外，你还会发现在符合标准的 HTML 文档中，<html> 标签还会包含如下代码：

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="UTF-8">
```

xmlns 属性是 XHTML namespace 的缩写，中文翻译为名字空间。XHTML 是 HTML 向 XML 过渡的标识语言，它需要符合 XML 文档规则，因此也需要定义名字空间。因为 XHTML1.0 不能自定义标识，所以它的名字空间都相同，就是“http://www.w3.org/1999/xhtml”。对于 HTML 文档来说我们只要照抄代码就可以了。

1.2 CSS 基本语法和用法

CSS 是一种高级的、弱类型的语言，与 HTML 一样都是一种标识语言，在任何文本编辑器中都可以打开或编辑。因此，不管你有没有基础，初次接触 CSS 时会感到很简单，而一旦完全使用 CSS 来布局页面时，你又会发现它很复杂。因此，读者必须在开始学习就要打下扎实的基础，下面我们就来介绍 CSS 的基本语法和简单用法。

1.2.1 CSS 基本语法

CSS 的语法单元是样式，每个样式包含两部分内容：选择符和声明（或称为规则），如图 1.1 所示。

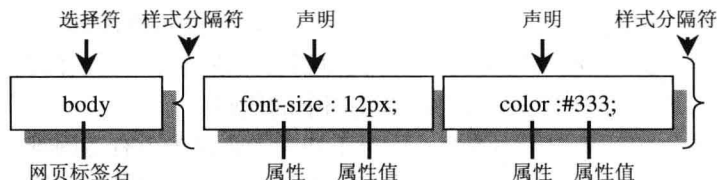


图 1.1 CSS 样式基本格式

(1) 选择符 (Selector)：选择符告诉浏览器该样式将作用于页面中哪些对象，这些对象可以是某个标签、所有网页对象、指定 Class 或 ID 值等。浏览器在解析这个样式时，根据选择符来渲染对象的显示效果。

(2) 声明 (Declaration)：声明可以增加一个或者无数个，这些声明命令浏览器如何去渲染选择符指定的对象。声明必须包括两部分：属性和属性值，并用分号来标识一个声明的结束，在一个样式中最后一个声明可以省略分号。所有声明被放置在一对大括号内，然后整体紧邻选择符的后面。

(3) 属性 (Property)：属性是 CSS 提供的设置好的样式选项。属性名由一个单词或多个单词组成，多个单词之间通过连字符相连，这样能够更直观地表示属性所要设置样式的效果。

(4) 属性值 (Value)：属性值用来显示属性效果的参数。它包括数值和单位，或者关键字。例如，定义网页字体大小为 12 像素，字体颜色为深灰色，我们则可以设置如下样式：

```
body{font-size: 12px; color: #CCCCCC;}
```

多个样式可以并列在一起，不需要考虑如何进行分隔，例如，我们再定义段落文本的背景色为紫色，则可以在上面样式基础上定义如下样式：

```
body{font-size: 12px; color: #CCCCCC;}p{background-color: #FF00FF;}
```

由于 CSS 语言忽略空格（除了选择符内部的空格外），因此我们可以利用空格来美化 CSS 源代码，则上面代码可以进行如下美化：

```
body {
    font-size: 12px;
    color: #CCCCCC;
}
p { background-color: #FF00FF; }
```

这样在阅读 CSS 源代码时就一目了然了。既方便阅读，又容易维护。

任何语言都需要注释，HTML 使用“<!-- 注释语句 -->”来进行注释，而 CSS 使用“/* 注释语句 */”来进行注释。例如，对于上面可以进行如下注释：

```
body { /* 页面基本属性 */
    font-size: 12px;
    color: #CCCCCC;
}
/* 段落文本基础属性 */
p { background-color: #FF00FF; }
```

1.2.2 应用 CSS 样式

CSS 的样式必须放置在特定类型的文件、标签或属性中，否则是无效的，浏览器会视其为普通的字符串，而不对其进行解析。CSS 代码一般可以放置在三个地方，详细说明如下：

(1) 直接放在标签的 style 属性中，例如，

```
<span style="color:red;">红色字体</span>
<div style="border:solid 1px blue; width:200px; height:200px;"></div>
```

这样当浏览器解析这些标签时，检测到该标签包含有 style 属性，于是就调用 CSS 引擎来解析这些样式码，并把效果呈现出来。

这种通过 style 属性直接把样式码放在标签内的做法被称之为行内样式，因为它与传统网页布局中在标签增加属性的设计方法没有什么两样，这种方法实际上还没有真正把 HTML 结构和 CSS 表现分开进行设计，因此不建议使用。除非为页面中个别元素设置某个特定样式效果而需要单独进行定义。

(2) 把样式代码放在<style>标签内，例如：

```
<style type="text/css">
body { /* 页面基本属性 */
    font-size: 12px;
    color: #CCCCCC;
}
/* 段落文本基础属性 */
p { background-color: #FF00FF; }
</style>
```

在设置<style>时应该指定 type 属性，告诉浏览器该标签包含的代码是 CSS 源代码。这样当浏览器遇到<style>之后，会自动调用 CSS 引擎进行解析。

这种 CSS 应用方式也被称为网页内部样式，如果仅为一个页面定义 CSS 样式时，使用这种方法比较高效，且管理方便。但是在一个网站中或在多个页面之间引用时，使用这种方法会产生代码冗余，不建议使用，而且多页管理样式也是不经济的。

内部样式一般放在网页的头部区域，目的是让 CSS 源代码早于页面源代码下载并被解析，这样避免当网页信息下载之后，由于没有 CSS 样式渲染而使页面信息无法正常显示。

(3) 把样式放置在单独的文件中，然后使用<link>标签或者@import 关键字导入。

这样当浏览器遇到这些代码时，会自动根据它们提供的 URL 把外部样式表文件导入到页面中并进行解析。这种应用样式的方式也被称为外部样式。一般网站都采用外部样式来设计网站的表现层问题，以便统筹设计 CSS 样式，并能够快速开发和高效管理。

1.2.3 建立外部样式表

外部样式表也就是一个文本文件，其扩展名为.css。当把不同的样式复制到一个文本文件中后，另存在扩展名为.css 文件中，则它就是一个外部样式表。如图 1.2 所示就是禅意花园的外部样式表。

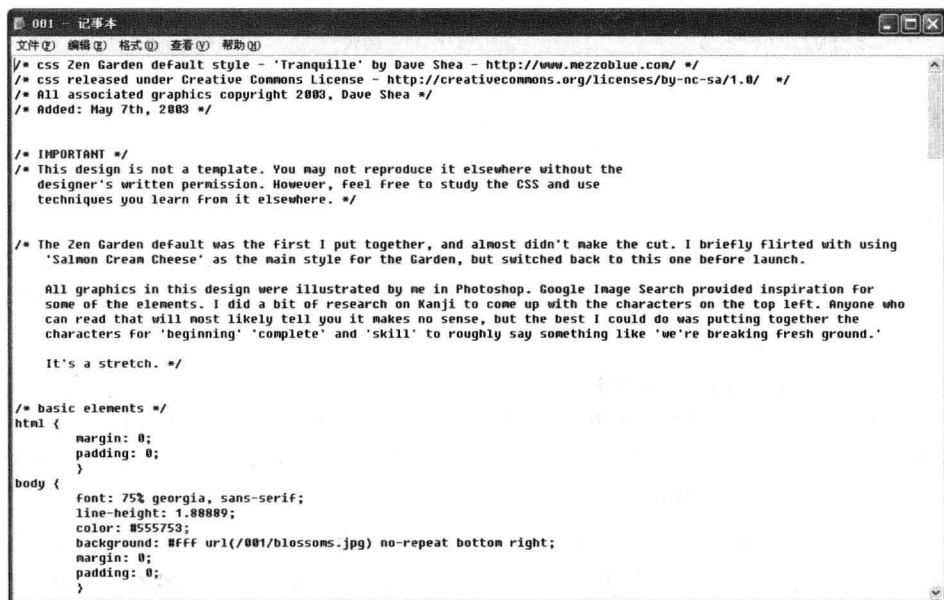


图 1.2 CSS 禅意花园外部样式表文件

外部样式表文件与内部样式表没有什么两样，都是由无数个样式组成的。你也可以在外部样式表文件顶部定义 CSS 源代码的字符编码。例如，下面代码定义样式表文件的字符编码为中文简体。

```
@charset "gb2312";
```

当你在 Dreamweaver CS3 中新建或打开一个 CSS 文件时，可以借助可视化操作快速进行设置或修改。方法是：选择【修改】|【页面属性】菜单命令，打开【页面属性】对话框（如图 1.3 所示），从中可以修改 CSS 文件的字符编码类型。

如果不设置 CSS 文件的字符编码，就可以保留默认设置，浏览器则会根据 HTML 文件的字符编码来解析 CSS 代码。