



本书内附光盘

网络精英系列

彭建翔 著

CSS2

网页结构 建筑师

- ◆ 想要成为优秀的网页设计师，就一定要了解CSS2
- ◆ CSS2能让您的网页如同鬼斧神工，成为WWW中的传奇作品
- ◆ 本书精细剖析CSS2的结构、参数及对浏览器的支持性，更透露了许多CSS2不为人知的重要关键技术，是一本网页设计者不可不看的经典好书，内容涉及：
 - 网页建筑师——CSS简介
 - 详细分析各种选择符
 - 如何指定一个文件在不同媒体之间显示不同的样式
 - 样式表重要概念——方块模型
 - 详细解说多种视觉模式细节
 - 产生内容及自动的数字和项目符号
 - 插页、分页、左右页等页媒体规则
 - 文字及字体变化与Microsoft WEFT工具
 - 网页中的表格功能及资料关系
 - 详细介绍CSS2参数，并提供许多扩展参数（来自CSS3）
 - RFC 1630重点说明及URI语法介绍
 - CSS重要资源网站介绍

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

CSS2

网页结构建筑师

彭建翔 著

中国铁道出版社
2002年·北京

(京)新登字 063 号

北京市版权局著作权合同登记号: 01-2001-4505 号

版 权 声 明

本书中文繁体字版由台湾金禾资讯股份有限公司出版(2001)。本书中文简体字版经台湾金禾资讯股份有限公司授权由中国铁道出版社出版(2001)。任何单位或个人未经出版者书面允许,不得以任何手段复制或抄袭本书内容。

图书在版编目(CIP)数据

CSS2 网页结构建筑师/彭建翔著 —北京:中国铁道出版社, 2001.12

ISBN 7-113-04468-9

I. C… II. 彭… III. 主页制作—软件工具, CSS2 IV. TP393.092

中国版本图书馆 CIP 数据核字(2001)第 090482 号

书 名: CSS2 网页结构建筑师

作 者: 彭建翔

出版发行: 中国铁道出版社(100054, 北京市宣武区右安门西街8号)

策划编辑: 严晓舟

特邀编辑: 邓庆容

封面设计: 孙天昭

印 刷: 北京市燕山印刷厂

开 本: 787×1092 1/16 印张: 17.75 字数: 420 千

版 本: 2002 年 1 月第 1 版 2002 年 1 月第 1 次印刷

印 数: 1~5000 册

书 号: ISBN 7-113-04468-9/TP·655

定 价: 32.00 元

版权所有 盗版必究

凡购买铁道版的图书,如有缺页、倒页、脱页者,请与本社计算机图书批销部调换。

出版说明

CSS 叫层叠样式表，是 Cascading Style Sheet 的缩写。它对网页中的布局、字体、颜色、背景和其他图文效果能够实现更加精确的控制。除此之外，它还有许多优点，如：只通过修改一个文件就可改变页数不定的网页的外观和格式；在所有浏览器和平台之间的兼容性；更少的编码、更少的页数和更快的下载速度等等，它为大部分的网页创新奠定了基石。CSS2 是在 CSS1 的基础上，扩展了更多的标记功能，实现了更多的效果。

本书详细讲解了 CSS2 的语法及主要参数，其结构清晰、语句通俗，同时对每个知识点均有范例帮助理解，而且这些范例文件都在本书的配盘中，相信您学习起来更轻松更容易。

若您对本书有任何疑问，请 E-mail 至 citymail@ms31.url.com.tw。

本书由台湾金禾资讯股份有限公司提供版权，经中国铁道出版社计算机图书项目中心审选，张翰文、李自运、马超、萧志军、陈贤淑及孟丽花等同志完成了本书的整稿及编排工作。

2002 年 1 月

目 录

第 1 部分

第 1 章	介绍 CSS	3
	什么是 CSS 呢?	4
	三种使用样式的方法.....	6
	一些名词介绍.....	7
	CSS 参数的定义.....	9
	CSS2 的基本数据类型.....	11
	inherit 参数值.....	12
	指定值、计算值、实际值.....	13
第 2 章	选择符	15
	选择符.....	16
	选择符的语法.....	16
	通用选择符.....	17
	类型选择符.....	18
	子代选择符.....	19
	子选择符.....	20
	邻近选择符.....	21
	属性选择符.....	22
	类选择符.....	24
	ID 选择符.....	24
	:first-child 伪类.....	25
	:link 及:visited 伪类.....	26
	:hover、:active、:foucs 伪类.....	27
	:left 及:right 伪类.....	28
	:first-line 伪元素.....	29
	:first-letter 伪元素.....	30
	:before 及:after 伪元素.....	31
	层叠选择符.....	31
第 3 章	媒体类型	33
	介绍媒体类型.....	34
	!important 规则.....	36
	各种媒体名称.....	37
	媒体组.....	38

第 4 章	方块	39
	方块模型	40
	显示模式	40
	查看区域	40
	嵌套区域	41
	最初的嵌套区域	41
	各种不同的方块	42
	区域等级标记及区域方块	42
	无名的区域方块	44
	行内等级标记及行内方块	45
	无名的行内方块	46
	密集方块	47
	run-in 方块	48
	位置结构	49
	区域格式内容	49
	行内格式内容	50
	相对位置	52
	悬浮方块	53
	绝对位置	54
	固定位置	55
第 5 章	更多显示模式的细节	57
	定义内容区域	58
	计算宽度及左右边的栏外空格	58
	计算高度及顶部、底部的栏外空格	59
	行高的计算	60
	leading 和 half-heading	60
第 6 章	生成内容、数字和项目符号	63
	引号	64
	自动生成数字	65
	计数器格式	66
	标记及清单	67
	项目符号	67
第 7 章	页媒体	69
	介绍页式媒体	70
	页方块和@page 规则	70
	左、右、第一页	70
	在页方块外的内容	71
	分页	71

第 8 章 文字及字体	73
@font-face 规则	74
Microsoft WEFT 工具	74
描述符	78
第 9 章 表格	79
CSS 的表格模型	81
无名的表格对象	82
字段选择	83
在视觉格式化模型中的表格	84
表格标题的位置及对齐方式	86
表格内容的排版	87
表格的层次	88
表格宽度的算法	89
固定的表格排版算法	89
自动的表格排版算法	91
表格高度的算法	94
列的水平对齐	94
动态行和列的影响	94
分开的边框模型	95
重叠的边框模型	96
边框冲突的解决	97

第 2 部分

CSS2 所定义的	102
background-color 参数	102
background-image 参数	103
background-repeat 参数	104
background-attachment 参数	106
background-position 参数	108
background 参数	109
border-top-width 参数	110
border-right-width 参数	111
border-bottom-width 参数	113
border-left-width 参数	114
border-width 参数	115
border-top-color 参数	116
border-right-color 参数	118
border-bottom-color 参数	119
border-left-color 参数	120

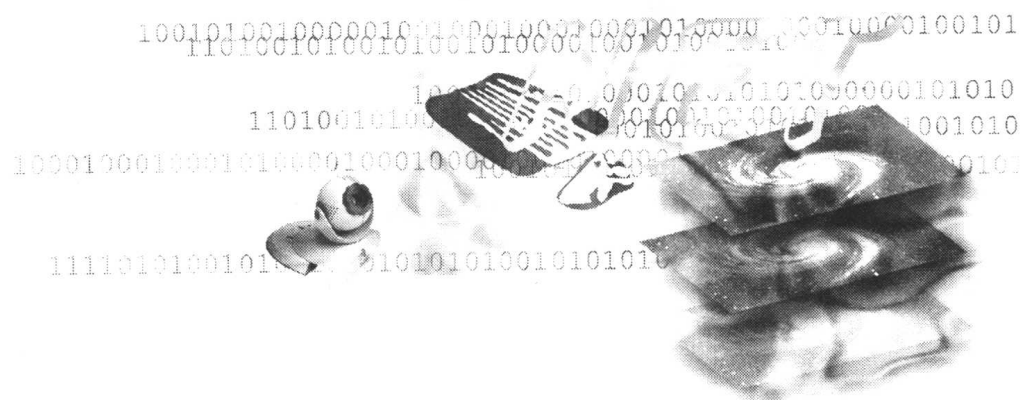
border-color 参数	121
border-top-style 参数	123
border-right-style 参数	124
border-bottom-style 参数	125
border-left-style 参数	127
border-style 参数	128
border-top 参数	130
border-right 参数	131
border-bottom 参数	132
border-left 参数	134
border 参数	135
border-collapse 参数	136
border-spacing 参数	138
bottom 参数	139
caption-side 参数	140
clear 参数	141
clip 参数	143
color 参数	144
content 参数	145
counter-increment 参数	147
counter-reset 参数	148
cursor 参数	150
direction 参数	151
display 参数	152
empty-cells 参数	155
float 参数	157
font-family 参数	158
font-size 参数	159
font-style 参数	161
font-variant 参数	162
font-weight 参数	163
font 参数	164
height 参数	166
left 参数	167
letter-spacing 参数	168
line-height 参数	169
list-style-type 参数	170
list-style-position 参数	173
list-style-image 参数	174

list-style 参数	175
margin-top 参数	176
margin-right 参数	177
margin-bottom 参数	178
margin-left 参数	180
margin 参数	181
max-height 参数	182
max-width 参数	184
min-height 参数	185
min-width 参数	186
orphans 参数	188
overflow 参数	188
padding-top 参数	189
padding-right 参数	190
padding-bottom 参数	192
padding-left 参数	193
padding 参数	194
page 参数	195
page-break-after 参数	196
page-break-before 参数	197
page-break-inside 参数	198
position 参数	199
quotes 参数	200
right 参数	201
size 参数	203
text-align 参数	203
text-decoration 参数	205
text-indent 参数	206
table-layout 参数	207
text-transform 参数	208
top 参数	210
vertical-align 参数	211
visibility 参数	213
widows 参数	214
width 参数	214
word-spacing 参数	215
z-index 参数	216
IE 的扩展样式	218
ime-mode 参数	218

layout-grid-char 参数	219
layout-grid-line 参数	220
layout-grid-mode 参数	221
layout-grid-type 参数	223
layout-grid 参数	224
overflow-x 参数	225
overflow-y 参数	226
scrollbar-3dlight-color 参数	227
scrollbar-arrow-color 参数	228
scrollbar-base-color 参数	230
scrollbar-darkshadow-color 参数	231
scrollbar-face-color 参数	232
scrollbar-highlight-color 参数	233
scrollbar-shadow-color 参数	234
scrollbar-track-color 参数	235
writing-mode 参数	236
zoom 参数	237
Netscape 的扩展样式	238
-moz-border-radius-bottomleft 参数	238
-moz-border-radius-bottomright 参数	239
-moz-border-radius-topleft 参数	240
-moz-border-radius-topright 参数	241
-moz-border-radius 参数	242
-moz-box-sizing 参数	243
-moz-opacity 参数	245
-moz-outline-color 参数	246
-moz-outline-style 参数	247
-moz-outline-width 参数	248
-moz-outline 参数	249
-moz-outline-radius-bottomleft 参数	250
-moz-outline-radius-bottomright 参数	251
-moz-outline-radius-topleft 参数	253
-moz-outline-radius-topright 参数	254
-moz-outline-radius 参数	255
-moz-user-focus 参数	256
-moz-user-input 参数	257
-moz-user-select 参数	258

第 3 部分

URI	262
资源.....	262
附表一 颜色表.....	264
附表二 选择符.....	268



第 1 部分

介绍CSS

第一章

什么是 CSS 呢?

CSS (Cascading Style Sheet) 叫做层叠样式表。在说明什么是层叠样式表之前先介绍一下什么是样式表。简单来说, 样式表是一种文件排版的技术, 它可以让元素按特定的样式显示。众所周知, B 标记会把文字变成粗体, EM 标记会把元素变成斜体, 它们都有默认的样式, 如果想改变文字颜色, 就得插入 FONT 元素, 到时候想要改变网页文字的颜色还不会太麻烦 (对于单张网页来说, 文字颜色的变换不会很多), 如果是一个网站时, 那光是在改变文字颜色上就会花费不少时间, 像 Microsoft 那种网站, 花费的时间更是不得了。有了样式表, 就可以把样式统一起来 (当然一个元素指定一个样式也是可以的, 不过这就同没有使用样式表是一样的), 当然样式表还不只有这些功能。还是来看看一个简单的例子——例 1-1。

例 1-1:

```
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=gb2312">
<TITLE>简单的 HTML 文件</TITLE>
</HEAD>
<BODY>
<DIV><FONT color="#FF0000" size="6">DIV 标记。</FONT></DIV>
<DIV><FONT color="#FF0000" size="6">红色。</FONT></DIV>
<DIV><FONT color="#FF0000" size="6">字体大小 24pt。</FONT></DIV>
</BODY>
</HTML>
```

上面的例子是一个很简单的 HTML 文件, 例中有三个 DIV 标记, 而且这三个元素的文字大小及颜色都是一样的。如果想要改变 DIV 中文字的大小或颜色就要逐个修改, 这样不但麻烦而且没有效率 (尽管现在有很多做网页的工具, 可以节省一些时间), 假如某天读者突然想改变自己网站 DIV 中文字的颜色, 那就要花费不少时间了。

现在看看例 1-2。

例 1-2:

```
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=gb2312">
<TITLE>简单的 HTML+CSS 文件</TITLE>
<STYLE>
DIV {font-size:24pt;color:#FF0000}
</STYLE>
</HEAD>
<BODY>
<DIV>DIV 标记。</DIV>
```

```
<DIV>红色。</DIV>
<DIV>字体大小 24pt。</DIV>
</BODY>
</HTML>
```

与例 1-1 不同之处就是，例 1-2 多了个 STYLE 标记和 DIV {font-size:24pt;color:#FF0000} 字符串，其中 STYLE 这个元素所包含的字符串就是 CSS，它的语法很简单，包括两部分：选择符(在例 1-2 中是 DIV)及样式声明(在例 1-2 中是 font-size:24pt 及 color:#FF0000)。而样式声明包括两个部分：参数名(在例 1-2 中是 font-size 及 color)和参数值(在例 1-2 中是 24pt 及 #FF0000)。最后再用大括号括起来就可以了，样式声明之间用分号隔开，在两个声明之间也可以插入空格。

那么使用样式表的好处到底是什么呢？是不是文件大小(Byte 数)更小了？答案是肯定的，在使用样式表之后，网页在网络上的传输效率就更高了。而且只要更改 STYLE 标记内部的字符串，就可以改变三个 DIV 中文字的大小及颜色了，而且还减少了编写网页的时间。

现在知道什么是样式表了吗？不过现在才要说明什么是层叠样式表。先是以例 1-3 来说明：

例 1-3：

```
<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=gb2312">
<TITLE>层叠样式</TITLE>
<STYLE>
DIV {font-size:24pt}
DIV {color:#FF0000}
</STYLE>
</HEAD>
<BODY>
<DIV>DIV 标记。</DIV>
<DIV>红色。</DIV>
<DIV>字体大小 24pt。</DIV>
</BODY>
</HTML>
```

这个例子与例 1-2 的不同之处在于：在例 1-2 中，STYLE 标记的内容是 DIV {font-size:24pt;color:#FF0000}，而在例 1-3 中它们是分开书写，不过二者的显示结果是一样的。层叠样式表可以把样式表串连起来，读者可以这么想，首先第一个样式(DIV {font-size:24pt})会把 DIV 中的文字大小改变成 24pt，而第二个样式(DIV {color:#FF0000})会把 DIV 中的文字颜色变成红色，由于两个样式没有关联，不会有冲突，所以无论谁先声明都没有关系，因为最后都是调用在同一个或同一群元素中。但是像下面这样编写页面就会出现错误：

例 1-4 :

```

<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=gb2312">
<TITLE>层叠样式</TITLE>
<STYLE>
DIV {font-size:24pt}
DIV {color:#FF0000}
DIV {color:blue}
</STYLE>
</HEAD>
<BODY>
<DIV>DIV 标记。</DIV>
<DIV>红色。</DIV>
<DIV>字体大小 24pt。</DIV>
</BODY>
</HTML>

```

因为浏览器先调用第二个样式(DIV{color:#FF0000}),再调用第三个样式(DIV {color:blue}),因此 DIV 中的文字会是蓝色的而不是红色的,只要样式声明不重复,把样式分开来写就不会影响样式的结果,这就是层叠样式表。

三种使用样式的方法

有三种方法可以用来使用样式,前面已经先介绍一种了,就是用 STYLE 标记来声明,这种方法比较适合于在一个网页中同样的元素使用同样的样式。而另一种方法是为个别元素指定样式,当然使用样式的元素必须带有 style 属性(每个元素都会提供 style 属性),请看例 1-5:

例 1-5 :

```

<HTML>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=gb2312">
<TITLE>style 属性</TITLE>
</HEAD>
<BODY>
<DIV style="font-size:24pt;color:#FF0000">DIV 标记。</DIV>
<DIV style="font-size:24pt;color:#FF0000">红色。</DIV>
<DIV style="font-size:24pt;color:#FF0000">字体大小 24pt。</DIV>
</BODY>
</HTML>

```

和 STYLE 标记不同的是,style 属性用两个双引号引住,其他语法都一样,不过最好不要使用这种方法,除非要为特定元素指定特定样式(不过还是可以在 STYLE 标记中为特定的