

丛书畅销
30万册

励志改变人生
编程改变命运

零基础学

HTML+CSS

第3版

11.5小时多媒体教学视频

张熠 等编著

本书特色

- ◎ 由浅入深，循序渐进，从零开始学HTML+CSS，一点都不难
- ◎ 编程基础、编程进阶、编程应用、项目实战、上机练习、面试指南
- ◎ 211个实例、3个案例、115个练习题、17个面试题

超值、大容量DVD

- ◎ 本书教学视频
- ◎ 本书源代码
- ◎ 本书教学PPT
- ◎ 本书习题答案

本书技术支持

- ◎ 论坛：<http://www.rzchina.net>



机械工业出版社
China Machine Press



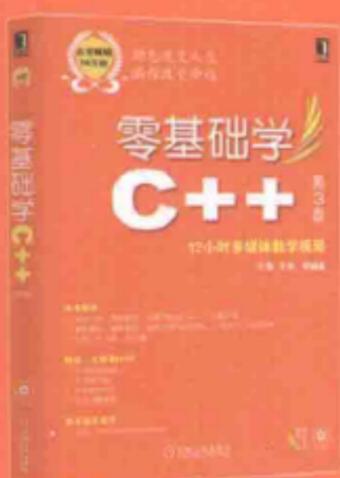
DVD-ROM



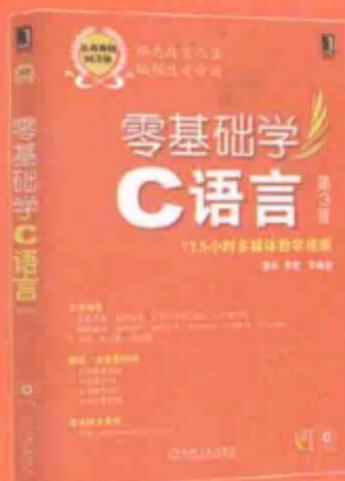
励志改变人生 编程改变命运

丛书特色

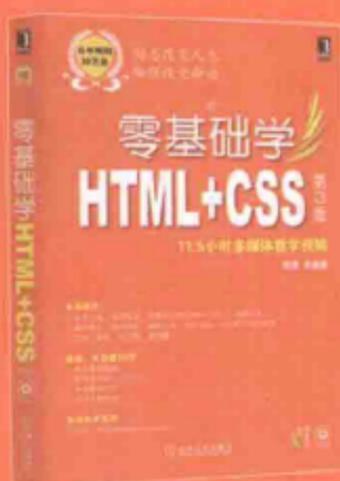
- ◎ 零门槛学习，没有任何基础便可轻松掌握
- ◎ 给出编程经验、编程陷阱、编程技巧和编程谬误
- ◎ 源代码提供了详细的注释，阅读起来无障碍
- ◎ 提供配套的多媒体教学视频，学习效果更好
- ◎ 每个知识点都对应相应的实例讲解，容易理解
- ◎ 提供上机实践练习题和常见面试题及解答
- ◎ 提供典型案例帮助读者提高开发水平
- ◎ 提供了专门的售后服务论坛：<http://www.rzchina.net>



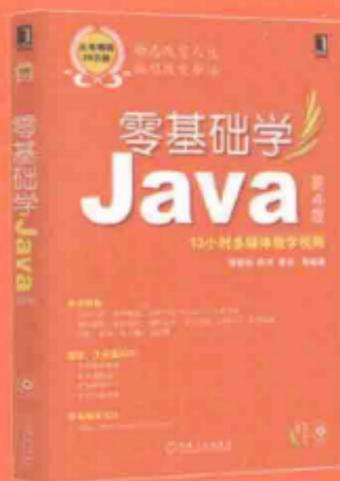
9 787111 468592 >
定价：79.00元（附光盘）



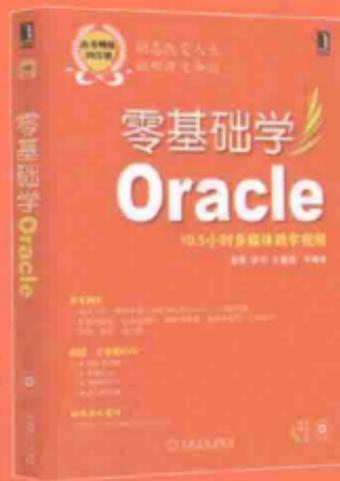
9 787111 461081 >
定价：69.00元（附光盘）



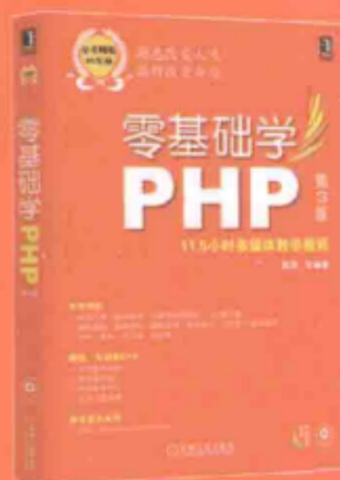
9 787111 463672 >
定价：69.00元（附光盘）



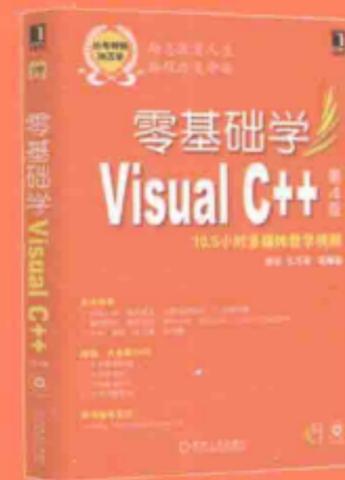
9 787111 466055 >
定价：79.00元（附光盘）



9 787111 468721 >
定价：69.00元（附光盘）



9 787111 467953 >
定价：69.00元（附光盘）



9 787111 468608 >
定价：79.00元（附光盘）



9 787111 449447 >
定价：79.00元（附光盘）



9 787111 468615 >
定价：79.00元（附光盘）



9 787111 461883 >
定价：79.00元（附光盘）



DVD-ROM

投稿热线：(010) 88379604
客服热线：(010) 88378991 88361066
购书热线：(010) 68326294 88379649 68995259

华章网站：www.hzbook.com
网上购书：www.china-pub.com
数字阅读：www.hzmedia.com.cn



上架指导：计算机/HTML

ISBN 978-7-111-46367-2



9 787111 463672 >

定价：69.00元（附光盘）

零基础学



HTML+CSS

第3版

11.5小时多媒体教学视频

张熠 等编著



机械工业出版社
China Machine Press



DVD-ROM

前 言

现在市面上许多可视化的网页编辑工具都达到了“所见即所得”的效果，这些图形化的开发工具可以直接编辑网页，而不用书写麻烦的标记。这使得用户在没有 HTML 语言基础的情况下，照样可以编写网页。这是网页图形编辑工具最大的优点，但也是它们最大的缺点。由于受到图形编辑工具自身的约束，将产生大量的垃圾代码。因为图形工具默认满足一般的网页设计需求，这样当生成网页元素时，就会生成该元素的所有属性代码，从而增加许多不必要的代码。所以一名明智的网页开发人员应该在掌握网页图形编辑工具的基础上进一步学会 HTML 语言，从而知道哪些是垃圾代码，如何消除无用的代码，进而快速制作出高质量的网页。

本书特色

本书系统全面地介绍 HTML 语言的知识，先从简单 HTML 语法入手，让读者对网页的信息展现有一个初步的了解，接着引入 DIV+CSS 的相关知识，让读者明白网页元素是如何布局的，最后通过 3 个案例的讲解，来巩固和加深前面学过的知识。

本书的特点主要体现在以下几个方面。

- ❑ 本书的编排采用循序渐进的方式，每章主题鲜明、要点突出，适合初级读者逐步掌握 HTML 的语法规则和设计思想。
- ❑ 本书的每一个知识点都配有一个范例，读者可容易地将代码复制到自己的机器上进行实验，自行实践和演练，直观体会所讲要点，感受使用 HTML 语言编写一张网页。本书的所有例子、源代码都附在随书光盘中，方便读者使用。
- ❑ 本书的每一章最后都有一个涉及该章知识的综合案例，该案例是对该章知识的总结、复习和提高，从而让读者对相关知识点理解得更深刻，这也是本书的一大亮点。
- ❑ 本书代码简洁、思路清晰、内容全面，兼顾 HTML 语言的所有知识点。
- ❑ 本书结合编者多年的 HTML 语言编写和网页设计经验，特别标注出易出错的技术点或初学者易误解的细节，同时对查错和调试页面给出指导思想，这样读者在学习时可少走弯路，加快学习进度。

本书内容

本书共分为四篇，共 24 章，先从 HTML 语言的基础知识讲起，使读者初步了解 HTML 语言的语法和网页生成。如果将编写 HTML 语言比作盖房子，第一篇的基础知识就相当于砖瓦水泥；

第二篇讲述 DIV+CSS 布局，就如同房子的框架设计；第三篇是 CSS 的相关知识点，本书对相关知识点做了详细介绍和范例演示；第四篇是案例实践和对差错的指导意见，以及常见面试题。

第一篇（第 1~9 章） HTML 基础知识

该篇讲述了 HTML 语言的基础知识，包括 HTML 语言介绍、如何通过 HTML 语言的标记在网页中创建和设置表格、列表、图片等元素，如何设置网页的背景颜色、文本的相关属性、网页的一些特别显示效果（如网页上的相关滚动信息）。

第二篇（第 10~11 章） DIV+CSS 布局

DIV+CSS 的使用已经是现在主流网站布局的主要实现方法，该篇通过 DIV 设计方法与以往的网页设计方法的对比，来突出用 DIV 设计网页的优势。初步介绍网页通过 DIV 来布局网页，以及如何结合 CSS 进行美观设计。

第三篇（第 12~20 章） CSS 部分

该篇全面详细地介绍 CSS 知识点，如选择符的种类、边框、字体、边距等样式的语法。

第四篇（第 21~24 章） 实践部分

该篇旨在让读者掌握 HTML 语言开发案例和实践项目，虽然只提供了 3 种常见网页布局的开发，却能帮助读者掌握 HTML 语言的语法和 DIV+CSS 网站的设计思想，最后编者根据自己的经验提供一些网页设计和检查的指导意见，并给出一些常见的 HTML+CSS 面试题，帮助读者顺利踏入职场。

本书读者对象

本书作为 HTML 语言的基础教程，适合以下读者阅读：

- 无 HTML 语言基础的初学者
- 了解 HTML 语言，但所学不全面的人员
- 高等院校学习网页设计的学生
- 使用 DIV+CSS 进行项目开发的人员
- 供其他相关技术人员

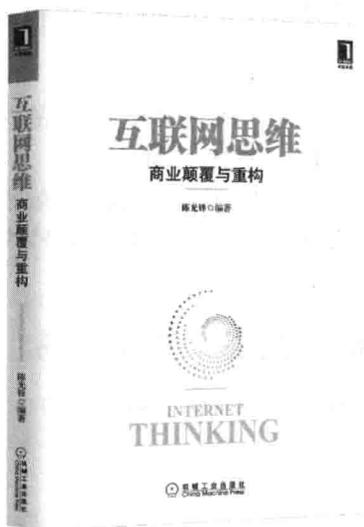
本书编写人员

张熠、冯华君、刘博、刘燕、叶青、张军、张立娟、张艺、彭涛、徐磊、戎伟、朱毅、李佳、李玉涵、杨利润、杨春娇、武鹏、潘中强、王丹、王宁、王西莉、石淑珍、程彩红、邵毅、郑丹丹、郑海平、顾旭光。

编者

2014 年 2 月

推荐阅读



互联网思维

第1本系统化阐述互联网思维的力作!

深度揭秘12大核心互联网思维!

瞬间掌握互联网思维精髓!

即刻改变未来=一本可以影响个人与企业命运的著作!

本书以雷军互联网七字诀“专注、极致、口碑、快”为核心精髓，结合马化腾在腾讯15周年“WE大会”上发表的“马七条”讲话精神，系统化提炼出互联网12大核心思维：标签思维、简约思维、NO.1思维、产品思维、痛点思维、尖叫点思维、屌丝思维、粉丝思维、爆点思维、迭代思维、流量思维、整合思维。

本书产品经理可以读读，从而认识对于产品来说，以哪些为产品素质的考核，又需要对产品做出怎样的调整；数据如何进行解读，又有哪些数据能够帮助我们认识用户的要求；如何在产品的制作中深挖用户的“痛点”，又如何利用“快速迭代”来完成产品的焕然一新。

运营经理也可以读读，在粉丝经济时代，粉丝只是一弯水中月，看得到捞不着。如何把握粉丝的参与感，又如何在恰当的时候激励粉丝，保持敏感的思维和触觉，随时感知到可能存在的引爆点，互联网思维的运营应该往这个方向去。

企业管理者也可以读读，关于如何提高产品的水平，同时又能有效避免人力成本和时间成本的浪费，在KPI和ROI的结合下，实现最大的效益。通过行业上下游合作，提高自身产品的能力，又如何将企业资源整合，成为产业链中不可或缺的平台。

本书是企业成长的必备参考书，适合每一位致力于企业快速成长及提升自身综合实力的职场人士阅读，对初创企业、在互联网方面涉足不深，以及处在发展缓慢状态的公司都有极大的启示作用。

目 录

前言

第一篇 HTML基础知识

第 1 章 认识 HTML 语言	1
1.1 HTML 与网页	1
1.1.1 认识 HTML 语言	1
1.1.2 静态网页	2
1.1.3 动态网页	2
1.2 HTML 标记的组成	4
1.2.1 单标记	4
1.2.2 双标记	4
1.3 创建第一张网页	5
1.4 浏览 HTML 的浏览器兼容	7
1.5 编写 HTML 的工具	8
1.5.1 Aptana 代码提示工具	8
1.5.2 Dreamweaver 网页开发工具	9
1.6 小结	11
1.7 习题	11
第 2 章 常用 HTML 标记和格式	13
2.1 设置正文的标题	13
2.1.1 标题设置的语法	13
2.1.2 标题设置经典案例——字号由大到小	13
2.2 设置段落	14
2.2.1 段落标记	14
2.2.2 段落中的空格	15
2.2.3 强制换行	16
2.3 给代码添加注释	17
2.4 文字的修饰	18

2.4.1	粗体与斜体	18
2.4.2	下划线与删除线	19
2.4.3	上标与下标	20
2.5	HTML 标记和格式实例	20
2.5.1	标题用 h1	21
2.5.2	设置每一行为一段落	21
2.5.3	粗体和删除线	22
2.5.4	加上化学元素和数学平方根	22
2.6	小结	23
2.7	习题	23
第 3 章	字体标记	25
3.1	设置字体大小	25
3.2	设置字体颜色	26
3.3	设置标题字体样式	27
3.3.1	设置标题的字体大小	27
3.3.2	设置字体的大小和颜色	28
3.4	物理字体和逻辑字体	29
3.4.1	物理字体	29
3.4.2	逻辑字体	30
3.5	字体标记实例	31
3.5.1	插入字体 	31
3.5.2	将字体颜色设置为红色	31
3.5.3	将文字设置为标题 h1	32
3.5.4	将文字设置为斜体并加删除线	32
3.5.5	设置字体为 <big>	32
3.6	小结	33
3.7	习题	33
第 4 章	文字布局	35
4.1	行的控制	35
4.1.1	换行控制	35
4.1.2	不换行控制	36
4.2	文字对齐	37
4.3	段落设置	38
4.4	列表	39
4.4.1	无序列表	39
4.4.2	有序列表	40

4.5 用其他方式修饰文本	41
4.5.1 欲格式化文本	41
4.5.2 计算机输出格式	43
4.6 文字布局实例	44
4.6.1 设置文字换行与不换行	44
4.6.2 设置文字对齐	45
4.6.3 显示列表	45
4.6.4 在网页中显示源代码	46
4.7 制作滚动文字	47
4.7.1 设置文字滚动	47
4.7.2 设置文字滚动的方向	48
4.7.3 设置文字滚动的速度和形式	48
4.8 小结	49
4.9 习题	49
第5章 图像和链接	51
5.1 图像的基本语法	51
5.1.1 在网页中插入图像	51
5.1.2 图像无法显示时的提示信息	52
5.1.3 控制图像的大小	53
5.2 图像和文字对齐	53
5.3 图像的边框	54
5.4 图像映射图	55
5.5 图像的应用实例	55
5.5.1 在网页上引入图像标记	55
5.5.2 设置图像参数 alt	56
5.5.3 给图像设置超链接	56
5.5.4 设置图像的边框	56
5.5.5 设置文本与图像对齐	57
5.5.6 设置图像的映射图	57
5.6 文字链接	58
5.6.1 最简单的文字链接	58
5.6.2 超链接网页的打开方式	59
5.6.3 链接的注释	60
5.7 图片链接	60
5.8 锚点链接	61
5.9 邮箱地址链接	62
5.10 相对路径与绝对路径	63

5.10.1 相对路径	63
5.10.2 绝对路径	64
5.11 链接应用实例	64
5.11.1 在网页中做文字链接	64
5.11.2 给图片设置超链接	65
5.11.3 设置锚点链接	66
5.11.4 邮箱链接	67
5.12 小结	67
5.13 习题	67
第6章 表单	69
6.1 表单概述	69
6.1.1 表单的用途	69
6.1.2 表单的结构	69
6.1.3 表单的基本语法	70
6.1.4 表单的数据传递方式	70
6.2 文本框和密码框	71
6.2.1 文本框	71
6.2.2 密码框	72
6.3 单选框和复选框	73
6.3.1 单选框	73
6.3.2 复选框	74
6.4 下拉列表与文本域	75
6.4.1 下拉列表	75
6.4.2 文本域	76
6.5 按钮	77
6.6 图像域、文件域和隐藏域	78
6.7 表单实例	79
6.7.1 插入表单	79
6.7.2 添加文本框和密码框	79
6.7.3 添加单选框和复选框	80
6.7.4 添加下拉列表和文本域	81
6.7.5 在表单中添加提交按钮	82
6.8 小结	84
6.9 习题	84
第7章 表格	86
7.1 页面中的表格	86
7.1.1 表格的外观	86

7.1.2 表格的基本语法	86
7.2 跨多行、多列表元	87
7.2.1 跨多行表元	88
7.2.2 跨多列表元	89
7.3 设置表格大小	90
7.3.1 设置表格边框	90
7.3.2 设置表格高度和宽度	90
7.4 表格内文字对齐	91
7.5 表格在网页中对齐	92
7.6 表格实例	93
7.6.1 表格中嵌入表格	93
7.6.2 设置跨行和跨列	94
7.7 小结	95
7.8 习题	95
第8章 框架	97
8.1 用框架灵活地组织网页	97
8.1.1 框架在网站中的作用	97
8.1.2 框架的结构分析	97
8.1.3 框架的基本语法	97
8.2 框架分栏	99
8.2.1 框架垂直分栏	99
8.2.2 框架水平分栏	100
8.3 框架的常用属性	100
8.3.1 设置不可调节框架大小	100
8.3.2 浏览器不支持框架	101
8.3.3 设置框架边框	102
8.3.4 设置滚动条	103
8.4 框架链接	103
8.4.1 导航框架	103
8.4.2 内联框架	104
8.5 框架实例	105
8.5.1 在网页中插入框架	105
8.5.2 设置框架的滚动条和边框	106
8.5.3 设置不支持框架文字	106
8.5.4 设置框架导航链接	107
8.6 小结	108
8.7 习题	108

第 9 章 网页中的动态图文和多媒体	110
9.1 移动属性的基本语法	110
9.2 文字的移动	111
9.3 图片的移动	112
9.4 设置对象的移动	113
9.4.1 文本移动的方向	113
9.4.2 文本的滚动循环	113
9.4.3 文本的移动速度	113
9.4.4 应用文本的全部移动标记	114
9.5 移动实例——电影结束效果	115
9.6 网页多媒体	117
9.6.1 网页多媒体的基本语法	117
9.6.2 设置自动播放	118
9.6.3 设置多媒体循环播放	119
9.6.4 设置面板的隐藏	120
9.6.5 设置面板的大小	121
9.6.6 设置对齐方式	122
9.7 网页多媒体实例——在网页中播放视频	123
9.7.1 引入视频文件	123
9.7.2 设置多媒体为非自动播放	123
9.7.3 设置多媒体无循环播放	124
9.7.4 设置控制面板的大小	125
9.8 小结	125
9.9 习题	125

第二篇 DIV + CSS 布局

第 10 章 用 DIV 布局	127
10.1 DIV 与 TABLE 布局比较	127
10.1.1 DIV 简介	127
10.1.2 DIV 与 TABLE 的区别	128
10.2 如何用 DIV 布局	131
10.2.1 设置 DIV 选择符	131
10.2.2 设置 CSS 修饰 DIV 效果	132
10.2.3 设置 DIV 的背景颜色	134
10.3 使用 DIV 注意事项	135
10.4 认识 SPAN	136
10.4.1 设置 DIV 选择符	136

10.4.2 何时用 SPAN	136
10.5 DIV 与 SPAN 的区别	137
10.5.1 默认宽度不同	138
10.5.2 块元素与行内元素	138
10.6 DIV 与 SPAN 实例——多层布局	139
10.7 小结	141
10.8 习题	142
第 11 章 DIV 与 CSS 结合	143
11.1 用 CSS 控制 DIV	143
11.2 CSS 文件链接	144
11.2.1 内联定义	144
11.2.2 链接内部 CSS	145
11.2.3 链接外部 CSS	147
11.3 DIV+CSS 布局实例	148
11.3.1 确定布局	148
11.3.2 定义 body 样式	148
11.3.3 定义主要的 DIV	149
11.3.4 设置层居中	151
11.4 小结	152
11.5 习题	152

第三篇 CSS部分

第 12 章 CSS 的基础知识	153
12.1 CSS 简介	153
12.2 CSS 文件链接方式	153
12.3 选择符	154
12.4 伪类	155
12.5 注解	155
12.6 单位	156
12.7 编写 CSS 的工具	157
12.7.1 打开 CSS 工具软件	157
12.7.2 新建 CSS 文件	157
12.7.3 另存为 CSS 文件	158
12.7.4 打开 Aptana 工具	158
12.7.5 在菜单栏中新建 CSS 文件	159
12.8 CSS 与浏览器的兼容性	159

12.9 CSS 基础实例	160
12.9.1 在 Aptana 中新建网页	160
12.9.2 在 Aptana 中新建 CSS 文件	160
12.9.3 链接 CSS 文件到 HTML 页面中	161
12.9.4 用不同的浏览器测试页面	162
12.10 小结	162
12.11 习题	162
第 13 章 字体设置和文本设置	164
13.1 文字的字体	164
13.1.1 字体的复合属性	164
13.1.2 字体颜色	166
13.1.3 字体大小	167
13.1.4 字体重量	168
13.1.5 字体类型	170
13.2 行距	171
13.3 设置文本间距与间隔	172
13.3.1 文本间距	172
13.3.2 文本间隔	172
13.4 字体实例——字体花样	174
13.4.1 加入字体名字	174
13.4.2 加入 CSS 控制字体	175
13.4.3 设置行距	176
13.4.4 设置文本间隔	177
13.5 字母大小写转换	178
13.6 文本修饰	180
13.7 空格处理方式	182
13.8 设置文本垂直与水平对齐	184
13.8.1 文本垂直对齐	184
13.8.2 文本水平对齐	186
13.9 文本缩进	188
13.10 文本设置——给段落设置缩进	189
13.10.1 对 DIV 布局	189
13.10.2 设置文本首行缩进	189
13.11 小结	190
13.12 习题	190
第 14 章 设置背景和尺寸	192
14.1 背景的基本语法	192

14.2	设置背景颜色	194
14.3	设置背景图像	195
14.3.1	设置背景图像滚动	196
14.3.2	设置背景图像的位置	197
14.3.3	设置背景图像的铺排	199
14.4	背景实例	200
14.4.1	设置层背景为绿色	200
14.4.2	设置层背景图像	201
14.4.3	设置背景为不铺排	202
14.4.4	设置背景图像位置	202
14.5	设置尺寸	203
14.5.1	设置宽度	204
14.5.2	设置高度	205
14.6	尺寸实例——设置层居中	206
14.6.1	尺寸实例手把手	206
14.6.2	设置层居中	209
14.7	小结	211
14.8	习题	211
第 15 章	设置外边距和内边距	214
15.1	外边距的基本语法	214
15.2	外边距的四边设置	216
15.2.1	顶端外边距的用法	216
15.2.2	右端外边距的用法	217
15.2.3	底端外边距的用法	218
15.2.4	左端外边距的用法	219
15.3	外边距实例——设置层与层的距离	220
15.3.1	层与层的距离效果	220
15.3.2	对层布局	222
15.3.3	设置层的尺寸	223
15.3.4	设置层漂移	225
15.3.5	设置层与层之间的间隔	226
15.4	内边距的基本语法	228
15.5	内边距的四边属性	229
15.5.1	顶端内边距的用法	230
15.5.2	右端内边距的用法	231
15.5.3	底端内边距的用法	231
15.5.4	左端内边距的用法	232



15.6 内边距实例	233
15.6.1 设置顶部内边距	234
15.6.2 设置其他位置的内边距	235
15.6.3 用复合属性 padding 设置内边距	237
15.7 小结	237
15.8 习题	238
第 16 章 设置对象边框	240
16.1 边框的基本语法	240
16.2 边框颜色	242
16.3 边框样式	243
16.4 边框宽度	245
16.5 对象的四个边框	248
16.5.1 设置顶端边框的属性	248
16.5.2 设置右端边框的属性	249
16.5.3 设置底端边框的属性	250
16.5.4 设置左端边框的属性	251
16.6 边框实例——多层边框	252
16.6.1 设置单层边框	252
16.6.2 设置多层边框	255
16.7 小结	257
16.8 习题	257
第 17 章 定位	259
17.1 定位的基本语法	259
17.2 设置对象的层叠顺序	261
17.3 设置对象的边框位置	262
17.4 定位实例——圆角制作	264
17.4.1 相对定位实例	264
17.4.2 绝对定位实例	265
17.4.3 实例——圆角制作	267
17.5 小结	272
17.6 习题	272
第 18 章 用 CSS 控制列表和表格	274
18.1 列表的基本语法	274
18.2 设置列表的图像	276
18.3 设置列表的文本排列	277
18.4 列表的预设标记	278

18.5	列表实例	280
18.5.1	设置列表图像文字内显示	280
18.5.2	设置列表样式为空心圆	281
18.5.3	设置文本外自定义列表图像	282
18.6	控制表格的基本语法	283
18.7	设置表格边框独立或合并	285
18.8	设置表格布局的算法	287
18.9	CSS 控制表格实例	288
18.9.1	布局好表格	288
18.9.2	表格边框独立和合并	289
18.9.3	设置表格算法为自动和固定	291
18.10	小结	292
18.11	习题	292
第 19 章	滚动条和布局	294
19.1	滚动条	294
19.1.1	滚动条的基本语法	294
19.1.2	设置滚动条亮边框颜色	295
19.1.3	设置 3D 界面亮边颜色和表面颜色	296
19.1.4	设置滚动条方向箭头颜色	297
19.1.5	设置滚动条 3D 界面暗边颜色和暗边框颜色	298
19.1.6	设置滚动条基准颜色和拖动区域颜色	300
19.2	层的漂移	301
19.2.1	设置层的漂移	301
19.2.2	清除层的漂移范围	303
19.3	设置层的可视区域	305
19.4	内容超过层大小时	307
19.5	鼠标指针的不同表示	308
19.6	设置对象可见性	311
19.6.1	设置对象是否隐藏	311
19.6.2	设置对象是否显示	313
19.7	小结	315
19.8	习题	315
第 20 章	选择符	317
20.1	通配选择符	317
20.2	类型选择符	318
20.3	包含选择符	319
20.4	布局常用选择符	320

20.4.1 ID 选择符	321
20.4.2 类选择符	322
20.5 分组选择符	323
20.6 伪类	324
20.7 伪类的四种状态	326
20.7.1 访问链接的状态	326
20.7.2 鼠标光标置于其上的状态	327
20.7.3 获得焦点的状态	328
20.7.4 访问过后的状态	329
20.8 CSS 中的单位	331
20.8.1 长度单位	331
20.8.2 颜色单位	333
20.9 小结	334
20.10 习题	335

第四篇 实践部分

第 21 章 游戏网站布局	337
21.1 网站主页规划	337
21.2 用 DIV+CSS 布局	338
21.2.1 实现页面头部	338
21.2.2 实现中间部分	341
21.2.3 实现页面底部	347
21.3 加入文字	348
21.4 调整全页面的兼容性	351
21.5 小结	352
第 22 章 企业网站布局	353
22.1 企业主页规划	353
22.2 对网页布局	354
22.2.1 将页面用 DIV 布局出框图	355
22.2.2 用 CSS 实现页面头部	356
22.2.3 用 CSS 实现页面体	358
22.3 添加文本等信息	362
22.4 调整全页面和浏览器兼容	363
22.5 其他二级页面模块制作	364
22.6 小结	364

第 23 章 博客主要页面布局	366
23.1 博客网站主页规划	366
23.1.1 分析主页需求	367
23.1.2 确定内容模块	367
23.1.3 画出效果图	367
23.2 用 DIV 布局框图	368
23.2.1 页眉	368
23.2.2 实现内容模块	369
23.2.3 页脚	372
23.3 CSS 实现框图	373
23.3.1 实现内容模块	373
23.3.2 实现整页效果	378
23.4 添加图片	379
23.5 加入文字	380
23.6 调整全页面	381
23.7 其他页面布局	381
23.8 小结	382
第 24 章 面试题	383
面试题 1 DIV 与 SPAN 的区别	383
面试题 2 HTML 的框架是什么	383
面试题 3 CSS 引入的方式有哪些	383
面试题 4 前端页面有哪三层构成, 各层的作用是什么	384
面试题 5 常见的浏览器的内核有哪些	384
面试题 6 img 元素的 alt text 属性和 title 属性有什么不同	385
面试题 7 document.body 和 document.documentElement 有什么不同	385
面试题 8 XHTML 和 DHTML 有什么区别	385
面试题 9 写出实现效果所需的 CSS 代码	385
面试题 10 写出一段左边固定、右边随电脑屏幕浮动而变动的 HTML+CSS 代码	386
面试题 11 写一段两个 DIV 交叉重叠的 CSS 样式	386
面试题 12 怎样使一个层垂直居中位于浏览器中	387
面试题 13 Firefox 嵌套 DIV 标签居中问题的处理方法	387
面试题 14 用纯 CSS 实现带下拉菜单的二级横向导航菜单	387
面试题 15 在网页制作中为什么要使用 CSS 技术	388
面试题 16 CSS+DIV 开发 Web 页面的优势有哪些	389
面试题 17 实现两列高度自适应的 CSS 代码	389

第一篇

HTML基础知识

第 1 章 认识 HTML 语言

本章主要介绍 HTML 语言，通过介绍，让读者对 HTML 语言有初步的认识。学好 HTML 是网页前台制作的基础，从而为以后做网页打好基础。本章先介绍 HTML 语言最基础的部分，如什么是 HTML，用什么来写 HTML 代码，怎么查看写完代码后的效果等。本章将要介绍的知识点如下：

- HTML 标记概念；
- 怎么创建网页；
- HTML 页面运行环境；
- 如何在 Windows 系统安装用于编写 HTML 的软件；
- HTML 最基本的代码是什么；
- HTML 有哪些浏览器支持。

1.1 HTML 与网页

HTML (Hypertext Markup Language, 超文本标记语言) 是一种用来制作超文本文档的简单标记语言。用 HTML 编写的超文本文档称为 HTML 文档。自 1990 年以来, HTML 就一直被当做万维网 (World Wide Web) 上的信息表示语言, 用于描述网页的格式设计和它与万维网上其他页面的连接信息。

1.1.1 认识 HTML 语言

HTML 语言利用各种标记 (tag) 来标识文档的结构以及标识超链接 (hyperlink) 的信息。但有时因浏览器不同会显示不同的效果, 这就是为什么同一文档在不同的浏览器中展示的效果会不一样。

目前 HTML 语言的版本是 5.0, 它是从 SGML (Standard Generalized Markup Language, 标准广义标记语言, 一套用来描述数字化文档的结构并管理其内容的复杂规范) 中的一个子集演变而来的。

现在很多工具都支持“所见即所得”。“所见即所得”是指在工具上直接编辑页面效果，可以不用管理 HTML 代码。换言之，在不知道 HTML 的前提下，有了“所见即所得”的工具也可以做网页。但需要提醒读者的是，那些“所见即所得”工具的确给网页制作带来很大方便，供很多人使用，但其原理离不开 HTML 语言，只不过它用另一种方式帮用户写好了这些代码。要注意，万变不离其宗，学好了 HTML 语言，在使用那些“所见即所得”的工具时，可以不使用“所见即所得”的功能。

在以后的学习过程中，读者将会慢慢对 HTML 有清晰的认识。

说明 Dreamweaver CS5 是一个“所见即所得”的好工具，它给网页制作者带来很大方便，实现“所见即所得”完全依赖于 Dreamweaver CS5 的内置浏览器，使得制作者可以直接在 Dreamweaver CS5 里看到 HTML 代码的效果。

1.1.2 静态网页

在网站设计中，纯粹的 HTML 格式网页通常称为“静态网页”，早期的网站一般都是由静态网页组成的。静态网页的特点是这个网页不论何时何地被浏览，都将显示相同的形式和内容，且仅能供浏览，无法与网站进行互动。也就是说，无法提供信息给网站，让网站响应使用者的需求。如下代码是一个静态页面的全部代码。

```
01 <html>
02 <head>
03 <title>静态页面</title>
04 </head>
05 <body>
06 <h1>这是一个静态页面</h1>
07 <h3>您只能浏览，不能进行交互。</h3>
08 </body>
09 </html>
```

静态页面使用 HTML 编写，通常扩展名一般为 .htm、.html、.shtml、.xml 等。静态网页只能单纯地在网页中展示文字与图片，听起来似乎功能简单，但它是所有网页的基础要素，其重要性自然也不言而喻。

在静态网页中，整个网页的主要结构与网页的显示控制都必须利用 HTML 实现。在 HTML 格式的网页上，可以出现各种动态的效果，如 GIF 格式的动画、FLASH、滚动字母等。这些“动态效果”只是视觉上的，它们与动态网页是不同的概念。

1.1.3 动态网页

与静态网页相对应的是动态网页。动态网页指网页的内容可以根据某种条件的改变而自动改变。如腾讯公司的 Qzone 空间里，常常会有一些使用者嵌入一个小小的计数器功能，当有人单击设计者的网页时，计数器的值会自动增加，这个计数器就是动态的。再比如，目前网络流行的论坛、社区网中用户的注册页面，当用户输入正确的用户名和密码后会成功登录，如果输入的用户名或者密码错误，页面会提示用户错误信息，这个也是典型的动态页面。

同时，与静态网页的扩展名不同，动态网页是以.asp、.jsp、.php、.perl、.cgi等形式为扩展名，并且在动态网页网址中有一个标志性的符号“?”。一个典型的动态网页的URL（Uniform Resource Locator，统一资源定位器）形式为：http://www.sina.cn/ip/index.asp?id=1。那么，动态网页与网页上的各种动画、滚动字幕等视觉上的动态效果为什么不是一个概念呢？

动态网页可以是纯文字内容的，也可以是包含各种动画的内容。这些只是网页具体内容的表现形式，并不是动态技术生成的页面，它不能根据用户的要求来更新页面。而一个网页，无论是否具有动态效果，除非是采用动态网站技术生成的网页，才可以称为动态网页。

注意

分清楚静态网页和动态网页的区别很重要，概念清晰才能进一步理解哪些是HTML语言可以做到的，哪些是不允许的，避免在学习的过程中钻牛角尖。

在了解了动态网页的概念之后，下面通过一个动态网页的例子来说明动态页面是如何和用户互动的。图 1-1 是一个动态页面的第一个页面。当输入一个已注册用户，如用户名为 appleing，密码为 1234567 时，系统检查到用户名和密码是正确的，即跳转到用户页面。图 1-2 所示为一个用户页面。



图 1-1 userLogin.jsp 页面



图 1-2 用户页面

注意 这种功能是静态页面无法做到的。或者说，仅仅依靠HTML代码是无法实现这种效果的。

图 1-2 所示的页面可以显示该用户可以使用的功能，包括用户邮件服务和用户短消息服务。这种针对使用者的设计是动态页面典型的功能标志。而如果用户输入错误的用户名和密码，比如用户名为 bupt，密码为 bupt，则系统会检查到用户名和密码是错误的，会显示错误信息，显示登录失败。图 1-3 所示为登录失败的页面。

以上是动态页面的一个典型例子，用户输入不同的信息，页面就会显示不同的结果。动态页面比静态页面显得更智能，更人性化，自然也是互联网发展的趋势。而理解 HTML 页面是学习动态页面的必备基础知识。



图 1-3 登录报错页面

1.2 HTML 标记的组成

HTML 用于描述功能的符号称为“标记”，如“HTML”、“BODY”、“TABLE”等。标记在使用时必须用尖括号“< >”括起来，而且是成对出现，无斜杠的标记表示该标记的作用开始，有斜杠的标记表示该标记的作用结束。

HTML 标记由左尖括号 (<) 和右尖括号 (>) 组成，浏览器会对尖括号里面的标记进行解释。如在 HTML 代码中出现<p></p>标记对时，浏览器在解释该标记对的过程中，会把它们看成段落并以段落的样式显示在网页中，同理，<table></table>标记对被浏览器看做表格来解析。在标记中，可分成单标记和双标记，而双标记即刚刚提到的标记对。

1.2.1 单标记

有些标记能完整地表达标记里的意思，只须在尖括号中输入标记名即可，这类标记叫单标记。一种单标记表示一种功能，即没有范围之分。单标记的语法形式如下。

```
<标记>
```

尖括号中的“标记”是指英文字母，这种字母由 HTML 提供，具有特殊功能，如<hr>表示画一条横线，再如最常用的换行标记
，表示在标记处换行，
标记告诉浏览器让其解析
时强制换行，这些单标记可以在网页中任意嵌套。

1.2.2 双标记

双标记与单标记刚好相反，这种标记有头有尾，且前面的标记与后面的标记保持一致，但在后面的标记前有斜杠，所以叫双标记。其中开始标记告诉浏览器从本处开始标记所表达的功能，再由结束标记告诉浏览器结束，构成结束标记，即在左尖括号后面加斜杠。其语法形式如下。

```
<标记>内容</标记>
```

其中，<标记>指的是字母关键字组成的标记，如<div></div>组成一个层，<table></table>组成一个表格，而“内容”是被标记修饰的部分。从标记对开始到结束的内容都被标记对修饰，如<div>这是一个层</div>中的文本文字“这是一个层”被<div>标记对修饰，表示一个层的内容，再如<p>段落内容</p>中的“段落内容”被<p>标记对修饰，表示一个段落等。双标记可以多层嵌套，如在<p></p>标记对中嵌套<div></div>标记对或其他标记对。在多个嵌套中，一个标记对的结束标记与最近一个标记对的开始标记配对。值得注意的是，标记对不能交叉。

标记对的使用比单标记更频繁，被修饰的内容是有范围限制的，即由开始标记到结束标记这段内容表示被修饰的范围，如文章主题修饰。

```
<h1>这是主题</h1>主题外
```

在双标记<h1></h1>中，标记内容“这是主题”表示成<h1>的样式，即显示效果上比正常文字大了，而且标记对外面的内容“主题外”没有被<h1>修饰，所以效果是正常的字体。

标记和内容组成网页上的信息，再加上 HTML 基本的标记就构成正式的网页源代码，可以放在 Web 服务器上让用户浏览。网页的最基本标记对为<html></html>，表示 HTML 语言。其下面又分成<head></head>标记对和<body></body>标记对，把头部信息放在<head>标记对中，如主题、字

体编码设置等，而正文等都放在<body>网页体中，其中网页体表现出来的效果就是日常看到的网页内容，如新闻、BBS 上的内容等。代码 1-1 所表示的是 HTML 基本标记。

代码 1-1 HTML 基本标记

```
-----文件名: HTML基本标记.html-----
01 <html>                                <!--网页的开始标记-->
02   <head>                               <!--网页的头部信息开始标记-->
03     <title>html基本标记</title>       <!--网页的主题开始标记-->
04   </head>                              <!--网页的头部信息结束标记-->
05   <body>                                <!--网页的主体开始标记-->
06     <!--注解:下面是网页体的正文-->
07     <p>这里由内容加标记组成</p>       <!--网页的段落标记-->
08   </body>                              <!--网页的主体结束标记-->
09 </html>
```

【代码解析】从代码 1-1 中得知，在这些 HTML 基本标记文档中，其中<html></html>表示 HTML 语言，在<html></html>里面包含头部（<head></head>）和内容体（<body></body>）。而主题标记是在头部<head></head>里面的。<p></p>在<body></body>里面，标记对<p>表示段落。HTML 标记的结构如图 1-4 所示。

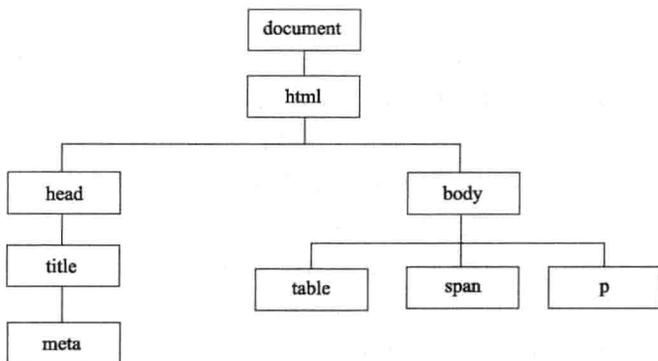


图 1-4 HTML 标记的构成

HTML 中的标记对不能随便放，如代码 1-1 所示，<title></title> 标记对不能放在 <body></body> 标记对中，同理，<p></p> 也不能放在 <head></head> 中，分清楚网页头部与网页体就不会混淆。

1.3 创建第一张网页

通过前面的学习，知道了网页的基本结构由网页头部和网页体组成，所以每一张网页都离不开 HTML 的基本结构。为了证实网页头和网页体组成网页的效果，下面创建第一张网页。本节将展示如何完成一张简单的网页，从基本标记一步一步完成。

(1) 用系统自带的记事本编写最基本的 HTML 代码，在电脑硬盘上任何一处右击，新建文本文档，这样就在当前位置出现一个文本文件，其名字为“新建文本文档.txt”（如果看不到 txt，说明用户电脑文件扩展名隐藏），该文件正处于重命名状态，按 Enter 键打开文档。

(2) 打开记事本后, 光标不停闪烁, 说明文档处于编辑状态, 这时可以输入 HTML 代码。根据 1.2 节学习的 HTML 基本结构, 可先输入最基本的 HTML 结构代码, 核对无误后再选择“文件”|“保存”命令, 如代码 1-2 所示。

代码 1-2 HTML 基本结构

```
<html>
  <head>
    <title></title>
  </head>
  <body><!-- 注释内容-->
  </body>
</html>
```

【代码解析】 在基本结构中, 在代码第 2 行<head>下的标记对<title></title>中放置网页主题, 而在代码第 5~6 行网页体<body></body>中放置内容。其中<!-- -->作为注解符, 不会在网页中显示, 但会出现在网页的 HTML 代码中。

对于初学者, 先输入 HTML 基本结构并不是坏事 (在有的编写 HTML 的工具中, 新建网页就包含基本结构), 这样就对 HTML 代码有了结构性的认识。下一步需要做的是对 HTML 基本结构添加标记和内容。

(3) 输入 HTML 标记和文本内容。在 1.2 节的“注意”中说明了页面头部不能放在页面体中的标记, 如<title></title>只能放在<head></head>中, 表示该页面的标题 (页面头部还有更多标记对), 页面的标题标记对不能在<body></body>中出现, 反之亦然。所以在页面头部和页面体应加上相应的内容或标记, 在代码 1-2 中添加内容或标记的代码如代码 1-3 所示。

代码 1-3 文本文件的 HTML 代码

```
-----文件名: 我的第一张网页.html-----
01  <html>
02    <head>
03      <title>创建第一张网页</title>
04    </head>
05    <body>
06      <!-- 第一个段落-->
07      <p>第一段落<br>                                <!--p为段落开始符, br为换行符-->
08          这里是第一段落第一行<br>
09          第二行
10      </p>
11      <!-- 第二个段落-->
12      <p>
13          第二段落<br>
14          这里是第二段落第二行<br>
15          第二行
16      </p>                                <!--p为段落结束符-->
17    </body>
18  </html>
```

【代码解析】 在代码 1-3 中, 在代码第 3 行对页面头部设置了<title></title>标记对表示页面标题, 在页面体中添加了两个段落, 即<p>标记对, 在代码第 7 行同时还嵌入了单标记
来换行, 这样第一张网页代码就创建好了, 接下来就需要转换成 HTML 文件了。

(4) HTML 代码是在记事本中编写的, 如何才能将其转换成 HTML 文件呢? 要想从文本文件

转换成 HTML 文件，可选择文本编辑器的“文件”|“另存为”命令，这时会打开“另存为”对话框，在“文件名”中输入要命名的名称，然后在英文输入状态下输入“.”，再加上 html 或 htm 为扩展名，选择保存类型为“所有文件”，单击“保存”按钮，即可输出 HTML 文件，到保存的路径上可查看 HTML 文件。对话框设置如图 1-5 所示。

(5) 打开 HTML 文件，浏览网页。通过保存的设置，在硬盘中多了一个 HTML 文件，双击该 HTML 文件，将会用默认的浏览器（一台电脑上可以装多个浏览器，其中有一个是默认的）打开。代码 1-3 中的标记被浏览器解析后，效果如图 1-6 所示。

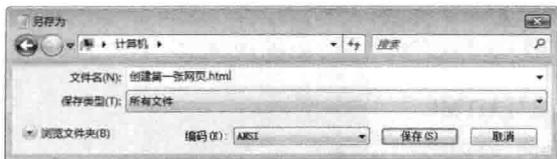


图 1-5 用“另存为”保存页面的对话框

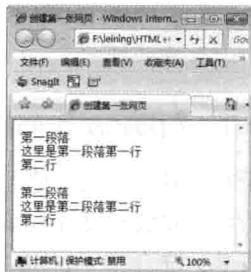


图 1-6 创建第一张网页界面

说明

创建的第一张网页就完成了，从 HTML 最基本结构到添加标记和文本内容，最终显示的是 HTML 代码，输入代码的方式很多，上面只是用记事本新建页面的一种方式，还有更多的方式，希望读者动手试试。对于双标记，别忘了输入结束标记来组成标记对，结束标记内还包含斜杠。

1.4 浏览 HTML 的浏览器兼容

目前主流浏览器有 IE、火狐 (Firefox) 和谷歌的 Chrome 等。不管编写 HTML 还是 CSS 代码，所显示的网页效果都要在这几种浏览器中调试，因为这些浏览器的厂商对 HTML 和 CSS 支持有所差异，所表现的效果当然会不一样。作为一个网页设计者，制作出来的网页至少要兼容这三家浏览器（还有 Safari、Opera，在市场上占有率很少）。

对上一节介绍的第一张网页，分别用 IE 和火狐浏览器打开代码 1-3，并观察二者同一页面的显示效果，IE 显示的效果如图 1-7 所示，火狐 (Firefox) 显示的效果如图 1-8 所示。



图 1-7 用 IE 打开 HTML 文件

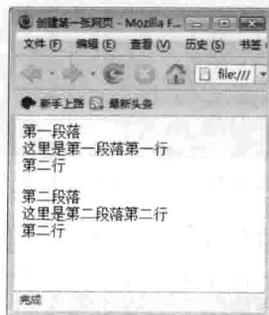


图 1-8 用 Firefox 打开 HTML 文件

1.5 编写 HTML 的工具

写 HTML 语言的工具很多，如 1.2 节所介绍的实例使用的是记事本，这也是最简单的一种，最常用的有 Dreamweaver 系列的，当然还有其他的很多工具。但本节介绍的两个编写软件，可以满足不同读者的需求。第一个软件——Aptana 有强大的代码提示工具，它是开源的并且有智能代码提示，可到官方网站下载（www.apтана.com），无论是编写现在学的 HTML，还是以后要学到的 CSS 都会更方便，非常有用。因为它是国外开源软件，所以界面是英语版的。第二个编写软件是 Dreamweaver CS5，由 Adobe 公司开发，也是最常用的软件，可到官方网站（<http://www.adobe.com>）下载，但它是商业软件。

1.5.1 Aptana 代码提示工具

对于初学者，知道下载地址后，应将安装文件下载到本地。如何安装和设置其参数？下面介绍编写 HTML 软件的安装。先介绍 Aptana，然后介绍 Adobe 公司的 Dreamweaver CS5。

(1) 到 www.apтана.com 网站下载 Aptana 软件，下载到本地后，双击“安装包”进行安装。图 1-9 所示为正在对安装包解压缩。



图 1-9 双击下载好的 Aptana 安装程序

(2) 读取包里的内容后，开始出现安装 Aptana 的安装设置，如图 1-10 所示，在此单击 Next 按钮。

(3) 出现安装协议界面。协议是软件商对用户的一些说明，单击 I Agree（同意协议）按钮即可。协议界面如图 1-11 所示。



图 1-10 开始安装前设置

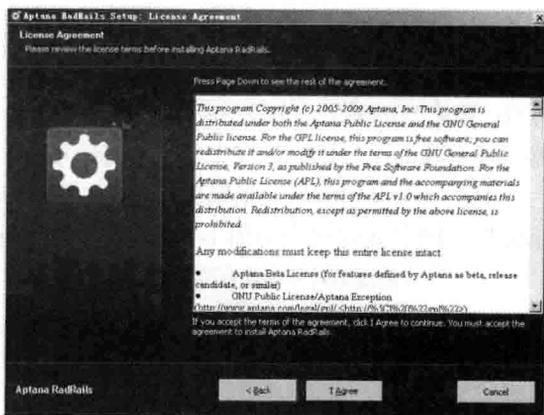


图 1-11 单击 I Agree 按钮进一步安装

(4) 选择软件的安装路径。如果安装路径保持默认，直接单击 Next 按钮到下一步，如想安装在其他盘符，单击 Browse 按钮选择路径，选择好后单击 OK 按钮，即选择了将要安装的路径，如图 1-12 所示。

(5) 开始正式安装界面。选择好路径后正式开始安装,在其界面中,单击 Show details 按钮将显示安装时详细文件的生成过程,如图 1-13 所示。

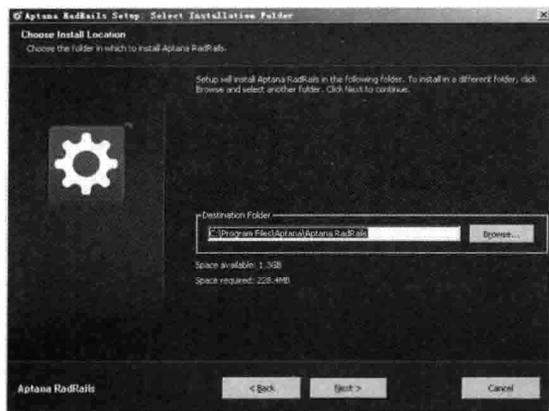


图 1-12 选择安装路径



图 1-13 正在安装

(6) 安装完毕后,界面会提示已完成 Aptana 的安装,如图 1-14 所示。只需要单击 Close 按钮,关闭界面,即可使用该软件。

(7) 安装完成后,会在桌面上显示 Aptana 图标,双击桌面图标,打开“工作界面”,将会出现英文的工作界面,如图 1-15 所示。



图 1-14 安装完成

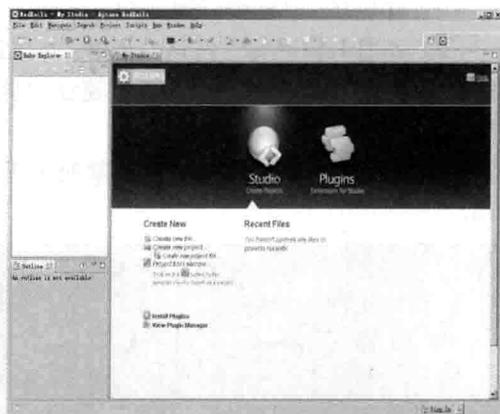


图 1-15 Aptana 工作界面

Aptana 安装完成。在以后的学习中就可以用 Aptana 作为编辑器了。在实际应用中,基本代码都会在 Aptana 界面里面产生,即在实际学习中通过代码提示、对 IE 和火狐浏览器支持的提示(写 CSS 文件时会提示),就会感觉到 Aptana 软件给网页代码编写带来多大的方便和价值。

1.5.2 Dreamweaver 网页开发工具

开源软件 Aptana 安装成功后,就可以使用该软件来编写 HTML 代码了。并不是所有的用户都喜欢使用它,不喜欢 Aptana 的用户或许喜欢 Adobe Dreamweaver CS5 这个软件,这也是初学者的首选工具。不管是对于代码编写还是网页效果预览它都是非常方便和实用的,除了对 HTML 的编

写外，还可以对动态网页编写代码，如 PHP 和 JSP 等页面，Adobe Dreamweaver CS5 也值得一用。为了讲解其安装过程，现在通过几个步骤演示如何安装 Adobe Dreamweaver CS5。

(1) 下载 Adobe Dreamweaver CS5 后，在本地硬盘找到 Adobe Dreamweaver CS5 的安装软件，双击安装软件，开始安装。下载该软件可以到官方网站 <http://www.adobe.com>，也可以到国内华军、天空等大型下载网站上下载。双击“安装软件”后的效果如图 1-16 所示。

(2) 解压缩后，开始安装前会出现安装提示，如欢迎安装 Adobe Dreamweaver CS5 软件等，告诉用户正要安装的软件名称，列出使用条款，这时一般单击“接受”按钮进入下一步，如图 1-17 所示。

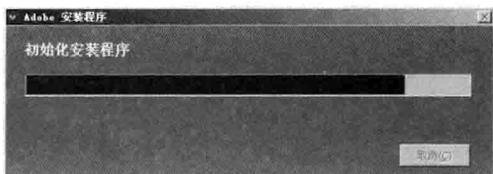


图 1-16 开始解压缩安装包

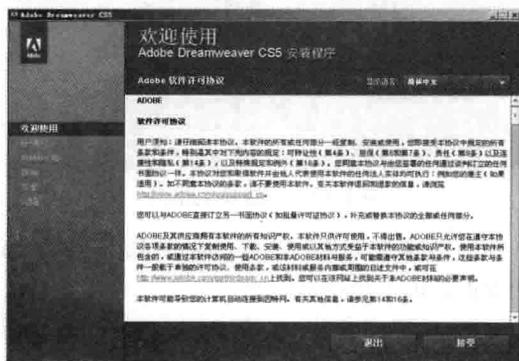


图 1-17 提示“接受”信息

(3) 如图 1-18 所示，在这一步中，可以选择安装的目标文件夹位置，单击路径文本框旁边的文件夹按钮，即可选择目标文件夹。建议初学者完全安装，选择“所有组件”复选框，单击“下一步”按钮开始安装。

(4) 当选项全部设置完成时，就处于正式安装状态，需要几分钟时间，安装过程如图 1-19 所示。



图 1-18 选择目标文件夹

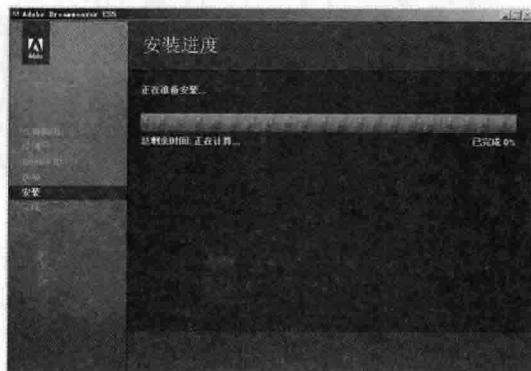


图 1-19 正在安装

(5) 安装完成后，出现如图 1-20 所示的界面。单击“完成”按钮，或者选择“观看视频教程”文字链接，了解 Adobe Dreamweaver CS5 的基本使用方法。

(6) Adobe Dreamweaver CS5 安装完成后，可以通过鼠标双击打开（要在步骤 (3) 中选择“在

桌面创建快捷方式”复选框)，也可以通过“开始”|“程序”|Adobe Dreamweaver CS5 选项打开软件，打开后工作环境如图 1-21 所示。

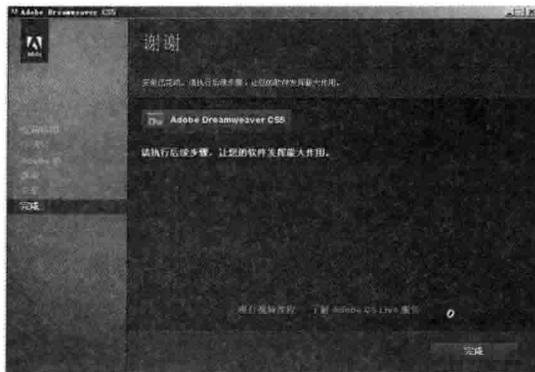


图 1-20 安装完成



图 1-21 Dreamweaver CS5 的工作环境

用 Aptana 还是用 Dreamweaver CS5? 以前编者写 CSS 代码时，常常用 Aptana，其中 Aptana 也可以作为插件在 Eclipse（访问 www.eclipse.org 查看更多内容）中使用，在新建的网页中，为 <head></head> 标记对中 <meta> 标记设置字符时，设置 charset 为 gb2312，即用国际 2312 编码支持汉字（如果按默认值在有汉字时会出现乱码）。

技巧 本软件需要序列号，可以到网上查找相应的序列号，否则只能使用30天。

说明

对于这两种编写环境，读者可根据爱好安装，若两种同时安装也不会冲突，选择合适的环境很重要。

1.6 小结

本章介绍了 HTML 的基础知识，并对 HTML 做了简单的介绍。HTML 由单标记和双标记组成，二者都可以嵌套，但双标记不能交叉。HTML 文件在 <html></html> 标记对中包含 <head></head> 页面头部和 <body></body> 页面体，然而，头部和页面嵌套的标记对也不同，这是网页基本结构。编写 HTML 代码可以用记事本，另存为“.html”文件即可转换为网页文件，也可以用 Aptana 和 Adobe Dreamweaver CS5 来编写，最后一节分别介绍了 Aptana 和 Adobe Dreamweaver CS5 的安装步骤。

1.7 习题

一、填空题

1. HTML 的英文全称是_____，中文全称是_____。
2. 静态页面使用 HTML 编写，通常扩展名一般为_____、_____、_____等。
3. 有些标记能完整地表达标记里的意思，只须在尖括号中输入标记名即可，这类标记叫_____。

二、上机实践

请问下面的标记有哪些错误?

```
<html>
  <head>
    <title>练习
    <p></p>
  </head>
  <body>
    </title>
    2<sup>2<+1=<u>5</u>
  </body>
</html>
```

【提示】注意标记对出现的位置，还有单标记和标记对的问题。

第 2 章 常用 HTML 标记和格式

通过学习第 1 章，可以对 HTML 有一定的认识。HTML 由标记组成，其中有单标记和双标记之分，大部分以双标记出现，即标记对。而本章要介绍的是应用中常用的 HTML 标记和格式，这些知识点在网页中会经常用到，也是必须得掌握的知识点之一。在讲解过程中分两部分：标记和格式。学完本章，应掌握以下内容：

- 标题标记；
- 段落标记；
- 换行与注释；
- 粗体与斜体；
- 删除线；
- 上标与下标。

2.1 设置正文的标题

现今网络媒体在新闻传播中起的作用越来越大，其传播速度和内容丰富程度都超越了传统媒体。如从现在的门户网站新闻点击率上可以看出网民直接在网上看新闻的热情。在新闻的格式中，有新闻主题和新闻内容之分，往往新闻主题的字号比正文内容大，同时也会看到字号很大或比较小的标题。如果读者细心，会发现标题从大逐渐变小是有规律的。

2.1.1 标题设置的语法

本节介绍标题的设置。标题的作用是用精简的文字概括整篇文章的主题，故文字不能太长，应从外表上突出标题与内容。标题的语法很简单，由双标记组成，在尖括号对里输入 h 与从 1~6 中的任意一个数字即可，数字不同，所表达的标题大小效果也不一样。其完整标题语法如下。

```
<h#>主题文字</h#>
```

其中“#”是标题里数字 1~6 中的任意一个，系统一共提供 6 种的字号标题，从数字 1 到数字 6，标题的字号按由大到小排列，即表示不同字号的标题。换言之，数字越大，标题字号就越小。

2.1.2 标题设置经典案例——字号由大到小

理解了标题是由字母 h 与从 1 到 6 的数字组成的后，下面用实例来演示这 6 种标题的效果，代码 2-1 表示从 h1 到 h6 的标题。

代码 2-1 6 种不同的标题标记

```

-----文件名：标题标记.html-----
01  <html>
02    <head>
03      <title>标题标记</title>    <!--网页标题标记对-->
04    </head>
05    <body>                        <!--网页主体标记对-->
06      <!--下面是标题标记对-->
07      <h1>标题h1</h1>            <!--主体内容标题标记对-->
08      <h2>标题h2</h2>
09      <h3>标题h3</h3>
10      <h4>标题h4</h4>
11      <h5>标题h5</h5>
12      <h6>标题h6</h6>
13    </body>
14  </html>

```

【代码解析】代码 2-1 中，从第 5 行开始的 `<body></body>` 标记对中输入了 6 种字号不同的标题，标题从 h1 到 h6，标题字号由大变小，效果如图 2-1 所示。

从图 2-1 中可以看到，代码 2-1 中第 7 行 `<h1></h1>` 双标记里显示最大字号的标题，同理，`<h6></h6>` 双标记里显示最小字号的标题。根据实际需要，在不同正文内容的主题下选择字号不同的标题标记，注意只选择一种做标题即可，如标题 h3 是文章的标题，其他标题不必在代码中出现了，如下面的代码所示。

```

<h3>这是标题h3</h3>
这是文章正文字。

```

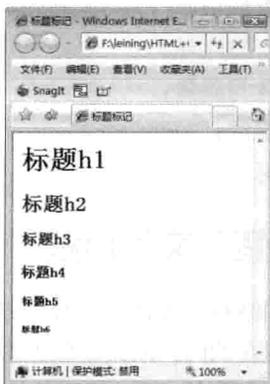


图 2-1 6 种标题标记

说明 不只是标题大小能明显突出主题重要性，在具体应用中选择不同标题也会达到意想不到的效果，甚至还可以加上标题字体的颜色，可以设置标题的字号和字体类型等，这些内容以后将会学习到。

2.2 设置段落

一篇文章由标题和正文组成，正文又是由多个段落组成的，所以可以说段落是构成文章的主体。既然一篇文章是由多个段落组成的，那么又如何区分段落呢？段落与段落之间有什么不同呢？带着这些问题来学习本节的段落标记。

2.2.1 段落标记

在学习段落与段落的区别前，先了解段落的基本语法。段落语法代码由尖括号对里放置 `p` 的双

标记来表示，在标记对里面放置的内容就构成了一个段落。段落的完整语法表示如下。

```
<p>这里表示段落</p>
```

在<p></p>标记对中就表示一个段落，那么怎样在一篇文章中存放多个段落呢？很简单，每个标记对就表示一个段落，有多少段落就有多少<p></p>，也就是从代码上区别段落与段落的不同。同理，从网页表现出的效果来看，段落与段落间有一行空格不显示任何内容，也就是常常看到新闻中的段落间有空行。那么可以利用 HTML 的段落标记来显示出文章结构的美观效果，使用户对文章理解有层次感。段落不管是在新闻还是书籍中都会出现，下面来看段落在网页中如何表现其效果，如代码 2-2 所示。

代码 2-2 多段落的表示

```
-----文件名：段落.html-----
01 <html>
02   <head>
03     <title>段落</title>
04   </head>
05   <body>
06     <!--这是注解，不会在网页中显示，下面表示有4个段落-->
07     <p>这里表示段落--</p>           <!--段落的标记对-->
08     <p>段落二</p>
09     <p>段落三</p>
10     <p>段落四</p>
11   </body>
12 </html>
```

【代码解析】在代码 2-2 中存在一行注解和 4 个 <p></p> 标记对。注解在 HTML 标记中用 <!--注解内容--> 表示，浏览解释注解行时，不会在网页中显示出来，只起注解作用，让网页制作者更好地理解 HTML 代码，而 4 个段落表示由 4 个 <p></p> 标记对组成的段落，每个段落都有不同的内容。当然，在浏览器中显示出来的效果就是表示 4 个段落，段落与段落间的空行与正常的换行是不相同的，各段落的格式和段落间的不同如图 2-2 所示。



图 2-2 多段落效果

注意 段落之间要空一行，从上面的效果中可以看出，四个段落中间有三行空出来的，这只是段落与段落间的间隔。

2.2.2 段落中的空格

在正规格式的文本中每一段落的开头会空两格，而当浏览器解释 HTML 文件的多个空格时，会自动去掉多余的空格，只显示出一个空格。即不管 HTML 中加上多少空格，都被看做一个空格。如果想在网页中出现空格的效果，必须加上 HTML 代码“ ”；想要显示多个空格，可以加上多个“ ”（注意后面的分号）。如下面的代码：

```
01 <html>
```


说明 代码
用于告诉浏览器强制换行，但并不在网页中显示。图2-3所示的是自动换行。

2.3 给代码添加注释

制作大型网站的时候，每个网页程序员所做的工作都会不同，但却相互依赖其他程序员写的代码。为了在同一个项目中让其他程序员看得明白，常常用注释来说明代码功能。注释代码的内容不会在网页中显示，只起到注释作用，如在一个网页头部用注释来说明该页面的作用和创建日期以及本网页所展示的功能等，让其他程序员或学习者看到注释就知道代码的功能。

在前面的代码中，多次出现了注解符，而且在注解内容中，编者都在前面标记“注解：”，这是为了告诉初学者代码的意义所采用的一种解说代码方法，并不会在网页中显示注解内的内容。注释语法比较容易，可以在单行中存在注释，同时也可以多行一起注释，单行注释的完整语法如下。

```
<!--这里放注释-->
```

在一行中，在尖括号中放置感叹号“!”和两个“-”，接下来放置注释内容，最后一定要结束注释，用两个“-”和另一半尖括号结束注释，即上面的代码表示单行注释。如果网页代码中只需要注释一行，可以用上面的注释。如果有多行同时需要注释，那是不是每行都用尖括号包围呢？答案是不需要的。其实还有一种方法可以注释多行，请看下面的代码。

```
<!--
    这里放注释
    第二行注释
    第三行注释
-->
```

即在注释处以“<!--”开始，在多行后用“-->”结束，中间的内容就是注释内容，不会在网页中显示。多行注释与单行注释类似，只不过中间多次换行，值得注意的是，注释标记对的结束标记一定不要忘记。

下面用实例来演示在 HTML 代码中包含注释，包括单行的和多行的，然后通过内容的显示了解被注释的内容是否在网页中显示。代码 2-4 表示在两行代码里包含注释，请读者观察能不能从显示的网页中看到注释。

代码 2-4 注释

```
-----文件名：注释.html-----
01 <html>
02   <head>
03     <title>注释</title>
04   </head>
05   <body>
06     注释代码是看不到的!
07     <!--                               注释标记，这里是多行注释
08                                     这里放注释
```

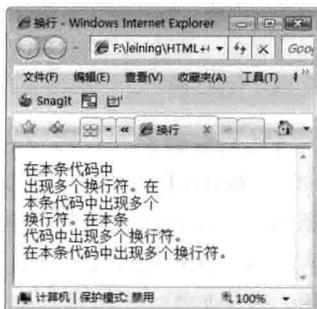


图 2-4 用多个换行符来换行

```

09                                     第二行注释
10                                     第三行注释
11         -->
12         <br>
13         上面有注释，你看到了吗？
14     </body>
15 </html>

```

【代码解析】在代码 2-4 中从第 5 行开始的 <body></body> 标签内输入的文字有多行，基本都被注释掉了，没有被注释的已显示，而被注释的代码没有显示，这就是注释标记在 HTML 代码中所起的作用，效果如图 2-5 所示。

注意

注释的尖括号不能交叉，不要忘记注解符后面的结束符。

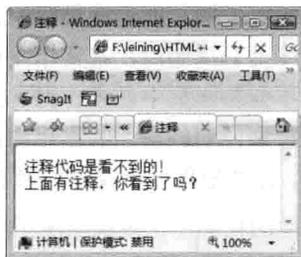


图 2-5 带有注释的代码

2.4 文字的修饰

本节从文字的细节修饰入手，使读者能把握 HTML 的各种文字在样式上的变化，如粗体、斜体等，以便制作出更为专业的网页。

2.4.1 粗体与斜体

粗体与斜体在语法结构上差不多，应用在文字文本中，且都是用双标记对来表示。在学习本节内容前，先比较二者的区别。

粗体标记的语法由 标记对表示，标记对里的文字在网页中显示的字体为粗体，语法简单。语法代码如下。

```
<b>粗体，这里显示的是粗体</b>
```

同理，斜体 <i> 里面显示的字体为斜体，语法代码如下。

```
<i>斜体，这里显示的是斜体</i>
```

可以这样理解，粗体用 括起来，斜体用 <i> 括起来。代码 2-5 演示二者代码和效果的不同之处。

代码 2-5 粗体与斜体

```

-----文件名：粗体与斜体.html-----
01 <html>
02     <head>
03         <title>粗体与斜体</title>
04     </head>
05     <body>
06         <!--下面表示粗体的用法-->
07         <b>粗体，这里显示的是粗体</b>     <!--粗体标记对-->
08     <br>
09         <!--下面表示斜体的用法-->
10         <i>斜体，这里显示的是斜体</i>     <!--斜体标记对-->

```

```
11     </body>
12 </html>
```

【代码解析】在代码 2-5 的<body></body>标记对里，用粗体标记对表示粗体，用换行标记给文字换行，还用斜体标记对文本设置斜体，粗体与斜体代表着文本不同的样式。效果如图 2-6 所示。

技巧 学习了粗体、斜体，可以联想相似的知识，如下一节要学习的<u></u>表示下划线等。

2.4.2 下划线与删除线

为了使部分文字特别显示，可以用下划线或删除线，这种字体往往引人注目，从而达到想要的效果。在网页中也可以使用下划线与删除线。

下划线就是在文字下面有一条线，也是用双标记对包含文字，尖括号里面用 u 表示。其语法代码如下。

```
<u>下划线</u>
```

类似地，删除线也是双标记对尖括号形式，里面用 s 表示。其语法代码如下。

```
<s>删除线</s>
```

二者比较容易区分，u 表示下划线，s 表示删除线，中间是要表达的内容。代码 2-6 所示为二者比较的示例。

代码 2-6 下划线与删除线

```
-----文件名: 下划线与删除线.html-----
01 <html>
02   <head>
03     <title>下划线与删除线</title>
04   </head>
05   <body>
06     <!--下面代码表示下划线-->
07     <u>这是下划线</u>           <!--下划线标记对-->
08     <br>                         <!--换行符-->
09     <!--下面代码表示删除线-->
10     <s>这是删除线</s>         <!--删除线标记对-->
11   </body>
12 </html>
```

【代码解析】代码 2-6 比较了下划线与删除线二者的区别，中间用换行符
是为了在网页上对齐比较。在第 7 行代码中下划线用<u></u>标记对表示，而在第 10 行代码中删除线用<s></s>表示，当浏览器解释这些标记对时，效果如图 2-7 所示。



图 2-6 粗体与斜体比较



图 2-7 下划线与删除线

说明 试着比较<s></s>与<strike></strike>的区别。

2.4.3 上标与下标

上标与下标多应用于文字或数字，如表示化学中的水元素、数学方程式，在数学中用平方、对数等。这是比较特殊的标记。

上标是位于文字或数字的右上角位置的标记。如 10 的平方，平方符号就显示在 10 的右上位置。稍后给出实例分析，这里先看语法代码，上标语法代码表示如下。

```
<sup>上标</sup>
```

同理，下标是在文字或数字的右下角位置的标记，应用也比较广泛，如化学元素中的水元素。下标语法代码表示如下。

```
<sub>下标</sub>
```

为了比较上标与下标的用法所显示出的不同效果，下面用实例来显示二者的效果，如代码 2-7 所示。

代码 2-7 上标与下标

```
-----文件名：上标与下标.html-----
01 <html>
02   <head>
03     <title>上标与下标</title>
04   </head>
05   <body>
06     <!--下面表示下标用法 -->
07     这是下标<sub>下标</sub>下标      <!--下标标识符 -->
08     <br>
09     <!--下面表示上标用法-->
10     这是上标<sup>上标</sup>上标
11     <br>
12     <!--用下标表示水的化学表达式-->
13     H<sub>2</sub>O
14     <br>
15     <!--用上标表示10的3次方-->
16     10<sup>3</sup>                    <!--上标标识符 -->
17   </body>
18 </html>
```

【代码解析】上面用代码在文字中体现了上、下标的位置。基本上，在第 10 行代码中上标用表示，在第 16 行代码中下标用表示，这些标记对可以表示特殊的文本应用，如在化学和数学上的应用等。效果如图 2-8 所示。

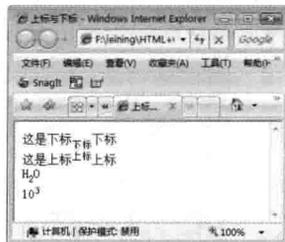


图 2-8 上标与下标

2.5 HTML 标记和格式实例

本章只介绍了 HTML 常用的标记和格式，这些知识点以后会常用到，毕竟网页只由少数标记

组成。学好常用标记和格式在以后学习中会容易很多。现在对前面所学的内容进行练习，知识点通过实例阐述。

2.5.1 标题用 h1

网页代码的基本组成在前一章学过，输入文章的关键放在<body></body>里面。现在的文章就是这样输入进去的，如代码 2-8 所示。

代码 2-8 HTML 标记和格式

```
-----文件名: HTML标记和格式实例手把手.html-----
01  <html>
02    <head>
03      <title>HTML标记和格式实例手把手</title>
04    </head>
05    <body>
06      <!--下面这行表示标题h1-->
07      <h1>HTML+CSS完全自学手册</h1>           <!--主体内容标题标记对-->
08      <!--下面表示文本行并在代码中换行-->
09      HTML+CSS完全自学手册
10
11
12      HTML+CSS完全自学手册
13
14
15      HTML+CSS完全自学手册
16    </body>
17  </html>
```

【代码解析】在第 7 行显示出标题 h1，其他的 3 行文字在代码中都有换行，效果如图 2-9 所示。

2.5.2 设置每一行为一段落

从图 2-9 中可以看到，有两条一样的文字显示在一行中。这与代码中显示的文字不同，那是因为代码会去掉多余的空格，只显示一个空格。要想添加段落必须有<p></p>标记对，给上面的代码加上<p></p>标记对后，部分代码如下。



图 2-9 标题 h1 和正文

```
01  <h1>HTML+CSS完全自学手册</h1>
02  <!--下面在每行文本中，用段落标记对包围-->
03  <p>                                     <!--段落的标记对-->
04  HTML+CSS完全自学手册
05  </p>
06  <p>
07  HTML+CSS完全自学手册HTML+CSS完全自学手册
08  </p>
09  <p>
10  HTML+CSS完全自学手册
11  </p>
```

【代码解析】上述代码是放在<body></body>中的，每个文本都加上<p>标记对，其他标记没

变化，效果如图 2-10 所示。

2.5.3 粗体和删除线

把第一段文字设置为粗体，在第二段文字上加删除线。部分代码如下。

```

01 <!--下面一行表示用标题的标记对-->
02 <h1>HTML+CSS完全自学手册</h1>
03 <p>                                <!--段落的标记对-->
04     <!--表示在文本中用粗体-->
05     <b>HTML+CSS完全自学手册</b>    <!--粗体标记对-->
06 </p>
07 <p>
08     <!--下面一行表示用删除线-->
09     <s>HTML+CSS完全自学手册</s>    <!--删除线标记对-->
10 </p>
11 <p>
12     <!--下面一行只是被包围在段落中-->
13     HTML+CSS完全自学手册
14 </p>

```

【代码解析】在这段代码里，比前面多了两个标记对，即****和**<s></s>**，其他部分都还是一样的，所表示的效果却不同，如图 2-11 所示。



图 2-10 设置段落



图 2-11 粗体和删除线

技巧

从上面显示的删除线中可以看出，后来在代码中加上标记对**<s></s>**即可达到效果，同理，设置文本的粗体用****标记对，请尝试使用下划线的方式。

2.5.4 加上化学元素和数学平方根

在前面的学习中用****表示上标，****表示下标，下面用上标和下标表示数学、化学等特殊符号，完整代码如代码 2-9 所示。

代码 2-9 HTML 标记和格式

```

-----文件名：HTML标记和格式实例手把手.html-----
01 <html>
02     <head>
03         <title>HTML标记和格式实例手把手</title>
04     </head>
05     <body>

```

```

06      <!--下面一行表示用标题的标记对-->
07      <h1>HTML+CSS完全自学手册</h1>
08      <p>
09      <!--表示在文本中用粗体-->
10      <b>HTML+CSS完全自学手册</b>
11      </p>
12      <p>
13      <!--下面一行表示用删除线-->
14      <s>HTML+CSS完全自学手册</s>
15      </p>
16      <p>
17      <!--下面一行只是被包围在段落中-->
18      HTML+CSS完全自学手册
19      </p>
20      <!--下面一行表示水的化学元素-->           H<sub>2</sub>S
21      <!--下标标识符的标记对-->
22      <br>
23      <!--下面一行表示10的2次方-->
24      10<sup>2</sup>                                <!--上标标识符的标记对-->
25      </body>
26      </html>

```

【代码解析】代码 2-9 中,从第 5 行开始的<body></body>标记对中,存在标题<h1>(第 7 行)标记对,粗体(第 10 行)标记对,删除线<s>(第 14 行)标记时,段落<p>(第 8 行)标记对等,用实例演示了前面各知识点的用法,在浏览器中运行后的效果如图 2-12 所示。

注意

在以后的学习中,还会遇到上面讲解过的知识,常用的标记和样式在实际应用中是不同缺少的,请读者注意观察实例代码中所运用的学过的知识,并总结其用法。



图 2-12 添加上标和下标标记

2.6 小结

本章所学的知识是网页编写过程中所用到的部分标记和格式,语法简单易学。由标题<h1></h1>到段落<p></p>再到换行
、注释<!-- -->都是常用的标记,下划线<u></u>、删除线<s></s>、粗体、斜体<i></i>,还有上标和下标都是重要的格式,也是在日常网页工作中所常用到的。最后通过实例,在 HTML 代码中演示了这些常用的标记和格式的用法,希望读者能举一反三,以便在实际应用中开发出更多的实例。

2.7 习题

一、填空题

1. 如果想出现多个空格的效果,必须加上 HTML 代码“_____”,想要多个空格,

可以加上多个“_____”，注意后面的分号。

2. 一个_____标记表示换一行，如果在一行文字中出现多次_____, 就会换行多次。

3. 粗体标记是_____, 斜体标记是_____, 下划线标记是_____。

二、上机实践

1. 在网页中编辑如下文字。

我的第一堂化学课

今天我第一次上化学课，心情比较兴奋，听爸爸妈妈说：“化学是很有趣的一门学科，可以做很多有趣的实验”。

课堂上，老师向我们解释了身边哪些是化学现象，例如，发光、发热等。还跟我们解说了水分子（H₂O）的组成。

【提示】

<code><p></code>	<code><!--p为段落开始符--></code>
<code><h3></code>	<code><!--主体标识符--></code>
<code></code>	<code><!--加粗标识符--></code>
<code><i></code>	<code><!--斜体标识符--></code>
<code>&nbsp;</code>	<code><!--空格标识符--></code>

2. 在网页输出 $2^2 + 1 = 5$ 。

【提示】

<code>< sup ></code>	<code><!--sup为上标开始符--></code>
<code>< u ></code>	<code><!--u为下划线开始符--></code>

第3章 字体标记

字体标记在网页中常常会用到，如一个标题字体需要设置特大字号且颜色是红色，字体类别使用宋体或楷体等。在 HTML 中用其字体标记来修饰文字字体方面的属性，使文字显示更丰富。本章所学习的正好是应用非常广的字体标记。本章主要学习下列内容。

- 设置字体大小；
- 设置字体颜色；
- 设置标题字体样式；
- 物理字体；
- 逻辑字体。

3.1 设置字体大小

在网页中常常可以看到，有的新闻网站为了满足不同年龄的读者浏览，有三种不同大小的字体供选择：12 像素、14 像素和 16 像素，默认的是 14 像素大小的字体。这样看起来比较舒服，适合大部分读者浏览，如是老年人可以选择大点的字体。现在来学习如何设置字体大小。

合理的字体大小能为网页增加不少装饰，更是一种美的享受。字体的语法如下。

```
<font>要放的文字</font>
```

这只是在字体上的标记，以后的学习则是在中加一部分属性，即可达到对字体设置的效果。下面介绍字体大小的设置。设置字体大小的语法代码如下。

```
<font size="1">HTML+CSS完全自学手册</font>
```

关键部分就是 size，用来表示大小。等号(=)后面的参数是有规定的，即从 1~7 的任意一个数字，数字越大，字体也就越大。现在将字体大小从 1~7 排列，如代码 3-1 所示。

代码 3-1 字体大小编排

```
-----文件名：字体大小编排.html-----
01  <html>
02    <head>
03      <title>标题字体大小</title>
04    </head>
05    <body>
06      <!--注解：以下是不同的文本字体的大小-->
07      <!--下一行表示文本字体为1-->
08      <font size="1">HTML+CSS完全自学手册</font>
09      <br>                                <!--换行-->
10     <!--下一行表示文本字体为2-->
```

```

11      <font size="2">HTML+CSS完全自学手册</font>
12      <br>
13      <font size="3">HTML+CSS完全自学手册</font>
14      <br>
15      <font size="4">HTML+CSS完全自学手册</font>
16      <br>
17      <font size="5">HTML+CSS完全自学手册</font>
18      <br>
19      <font size="6">HTML+CSS完全自学手册</font>
20      <br>
21      <!--下一行表示文本字体为7-->
22      <font size="7">HTML+CSS完全自学手册</font>
23      </body>
24 </html>

```

【代码解析】以上代码是从字体最小的递增到字体最大的，其中文字内容都一样。从代码 3-1 可以发现，每设置一行字体后都会带
，以便比较。在页面中的显示效果如图 3-1 所示。

读者可尝试在 size 值前面加上“+”、“-”字符，更灵活地设置文字尺寸。

3.2 设置字体颜色

通常在网页中看到的五彩缤纷的字体，有清新一点的、有暗淡一点的，颜色搭配好了，用户从视觉上认可该网页，这样才有可能继续读下去。字体颜色的语法简单易学，由 color 来表示，语法形式如下。

```
<font color=""></font>
```

标记中的参数 color 后面可以接 RGB 颜色值，如#000000。也可以用颜色的单词表示，如 red。完整的实例代码如代码 3-2 所示。

代码 3-2 颜色值表示的两种方式

```

-----文件名：字体颜色.html-----
01 <html>
02   <head>
03     <title>字体颜色</title>
04   </head>
05   <body>
06     <!--下一行表示文本颜色为红色-->
07     <font color="red">html+css完全自学手册</font>
08     <br>
09     <!--下一行表示文本颜色为#cccccc-->
10     <font color="#cccccc">html+css完全自学手册</font>
11   </body>
12 </html>

```

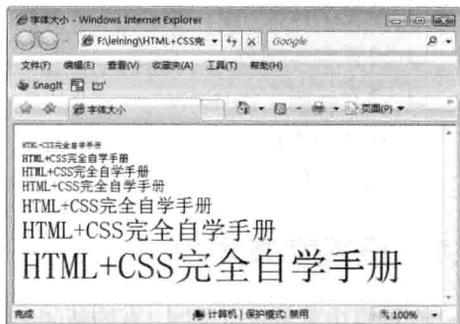


图 3-1 标题字体大小

【代码解析】在第7行代码的字体标记中,设置 color 参数值分别为红色和#cccccc,在两行文字中表示两种不同颜色。同理,如果想设置更多文字颜色,可以在标记中包围文字,并设置参数 color 值,被包含的文字正是字体颜色设置的区域。代码 3-2 的运行效果如图 3-2 所示。

技巧

根据W3C标准,建议读者设置字体颜色时,尽量用以后会学到的CSS的color属性。



图 3-2 字体颜色

3.3 设置标题字体样式

关于标题,在前一章中已讲过如何设置标题大小,但往往有的标题文字需要用颜色来表示其效果,在这里主要在前面的基础上学习标题属性,如标题的颜色和大小。标题的颜色和大小设置要用到标题标记<h#>(#代表不同大小的标题的数字)和前面学习的字体颜色。

3.3.1 设置标题的字体大小

本节再次讲解标题 h1~h6,然后在这些标题中设置不同的颜色和样式。标题用<h#>表示,#代表数字,数字1~6表示标题字体大小从大到小,完整语法表述如下所示。

```
<h1>设置标题h1</h1>
<h2>设置标题h2</h2>
<h3>设置标题h3</h3>
<h4>设置标题h4</h4>
<h5>设置标题h5</h5>
<h6>设置标题h6</h6>
```

由系统提供的标题大小只有这6种,其中h1字体最大,相应地,h6是最小的。为了体现这些字体的大小,代码3-3演示了各个标题的大小,设置标题字体只需要在标题标记内输入文字即可。

代码 3-3 标题字体设置

```
-----文件名: 标题字体设置.html-----
01 <html>
02 <head>
03 <title>标题字体设置</title>
04 </head>
05 <body>
06 <!--下面的行表示标题用法-->
07 <h1>html+css完全自学手册--h1</h1>
08 <h2>html+css完全自学手册--h2</h2>
09 <h3>html+css完全自学手册--h3</h3>
10 <h4>html+css完全自学手册--h4</h4>
11 <h5>html+css完全自学手册--h5</h5>
12 <h6>html+css完全自学手册--h6</h6>
13 </body>
14 </html>
```

【代码解析】代码3-3表示了标题用法,代码第7~12行分别是h1~h6,即标题从大到小,每个标题都会占用一行(代码中没有换行符
),效果如图3-3所示。

3.3.2 设置字体的大小和颜色

上一节的代码显示出标题字体的大小与颜色。用标记 `font` 可以设置文本对象的大小，同时还可以用标题标记设置其大小或颜色，也可以设置标题带下划线、删除线等样式。代码 3-4 即表示在标题中设置字体的大小与颜色。

代码 3-4 标题字体的大小与颜色

```
-----文件名：字体的大小与颜色.html1-----
01 <html>
02 <head>
03 <title>标题字体设置</title>
04 </head>
05 <body>
06 <!--下面的行表示在标题中，设置文本字体样式-->
07 <!--下一行表示对h1标题设置红色字体-->
08 <h1><font color="red">html+css完全自学手册--h1</font></h1>
09 <!--下一行表示对标题h2设置删除线-->
10 <h2><s>html+css完全自学手册--h2</s></h2>
11 <!--下一行表示对标题h3设置粗体-->
12 <h3><b>html+css完全自学手册--h3</b></h3>
13 <!--下一行表示对标题h4设置斜体-->
14 <h4><i>html+css完全自学手册--h4</i></h4>
15 <!--下一行表示对标题h5设置下划线-->
16 <h5><u>html+css完全自学手册--h5</u></h5>
17 <!--下一行表示对标题h6设置字体大小和字体颜色-->
18 <h6><font size="3" color="blue">html+css完全自学手册--h6</font></h6>
19 </body>
20 </html>
```

【代码解析】在代码 3-4 中，第 6~17 行设置了标题字体的颜色、大小，还对标题的字体设置了下划线、删除线等。这说明除了标题默认的字体大小和颜色外，还可以用 `` 或其他标记对标题设置不同的样式，效果如图 3-4 所示。

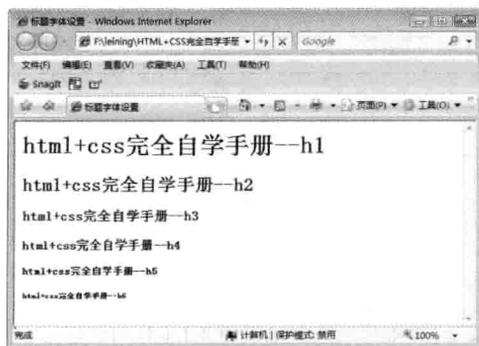


图 3-3 标题字体



图 3-4 标题字体的颜色与大小

说明 在HTML标记中，标题标记主要应用于文章标题，对正文等其他内容不太合适，这些内容应该用 `` 标记中的参数 `size` 来设置文字大小。同时，针对下划线、粗体、颜色等样式都可以进行设置。

3.4 物理字体和逻辑字体

第2章介绍了常用的HTML标记，这些标记也是本节要介绍的字体标记，字体分为物理字体和逻辑字体，本节就介绍它们。

3.4.1 物理字体

物理字体在工作中常会用到。尽管在前面章节已介绍过部分标记，但现在还有必要罗列出来，与其他标记一起比较，从而达到提高学习效率的目的。

以下是所有物理字体列表，中间的省略符代表放置的文本内容，通过标记对其进行修饰，后面的文字用来说明其功能。

```
<B>...</B>           //将字符设置成粗体
<I>...</I>           //将字符设置成斜体
<U>...</U>           //给字符增加下划线
<S>...</S>           //给字符增加删除线
<TT>...</TT>         //将字符设置成打字机字体
<SUP>...</SUP>        //将字符设置成上标字体
<SUB>...</SUB>        //将字符设置成下标字体
<strike>的意义<s>，在XHTML 1.0中已不推荐使用。
```

若对“文字”设置不同的物理字体，可以把“文字”包含在物理字体标记内，其包含的标记对如下所示。

```
<b>文字</b>
<i>文字</i>
<u>文字</u>
<sup>文字</sup>
<sub>文字</sub>
<s>文字</s>
<strike>文字</strike>
```

在标记对里放入文字就可以表现出标记里的效果，在应用中发现，物理字体并不被标记包含着。物理字体可以单独用来表现文字。代码3-5表示各物理字体标记的区别。

代码3-5 物理字体

```
-----文件名: 物理字体.html-----
01  <html>
02    <head>
03      <title>物理字体</title>
04    </head>
05    <body>
06      <!--下面表示物理字体用法-->
07      <b>文字--粗体</b>
08      <br>
09      <i>文字--斜体</i>
10      <br>
11      <u>文字--下划线</u>
12      <br>
13      <sup>文字--上标</sup>
14      <br>
```

```

15      <sub>文字--下标</sub>
16      <br>
17      <s>文字--删除线</s>
18      <br>
19      <strike>文字--删除线</strike>
20      </body>
21 </html>

```

【代码解析】在代码 3-5 中，上面的物理字体用法在前面已介绍过，并用实例做演示，本节对这些标记进行汇总，效果如图 3-5 所示。

注意 使用物理字体的标记对时不要忘记写上标记对结束符。

3.4.2 逻辑字体

双标记对的逻辑字体和物理字体很相似，都由标记对组成，但它们所体现的重点不同，逻辑字体所显示出来的效果更美观。

接下来介绍逻辑字体，逻辑字体可能很少见到，但作用很大。如在网页上表现代码形式的字符等。先看一下<code>，其语法形式如下。

```
<code>代码</code>
```

说明 先大概了解逻辑字体的语法，实例会在下一章讲解。

以下是所有逻辑字体列表，中间的省略符代表放置的文本内容，通过标记对修饰，后面的文字用来说明其功能。

```

<EM>...</EM>           //强调文字
<STRONG>...</STRONG>   //字体加重
<CODE>...</CODE>       //显示编程代码
<SAMP>...</SAMP>       //显示示例文字
<KBD>...</KBD>        //显示键盘按键文字
<SMALL>...</SMALL>    //缩小文字
<BIG>...</BIG>        //放大文字

```

以上逻辑字体语法形式与上一节中的一样，都是标记对。代码 3-6 罗列出了部分逻辑字体。

代码 3-6 逻辑字体

```

<em>文字</em>
<strong>文字</strong>
<small>文字</small>
<big>文字</big>

```

试比较以上逻辑字体的区别，先看示例代码。

```

-----文件名: 逻辑字体.html-----
01 <html>
02   <head>
03     <title>逻辑字体</title>
04   </head>
05   <body>

```

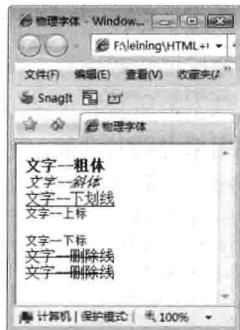


图 3-5 物理字体

```

06      <!--下面表示逻辑字体-->
07      <em>文字 --斜了</em>
08      <br>
09      <strong>文字--加重</strong>
10      <br>
11      <small>文字--小的</small>
12      <br>
13      <big>文字--小的</big>
14  </body>
15 </html>

```

【代码解析】在从第5行代码开始的<body></body>标记中插入4行文字，这些文本都被不同的逻辑字体修饰。第7行强调其内部文本，第9行加重其内部文本，第11行使其内部文本变小，第13行标记对里面的文字进行变大显示。本实例通过逻辑字体的设置，使文本内容效果发生变化，效果如图3-6所示。

在网页中包含计算机代码，较短的计算机代码可以直接用在页面中，如多行代码需要在页面中显示，在编写代码时会用到<pre>标记。

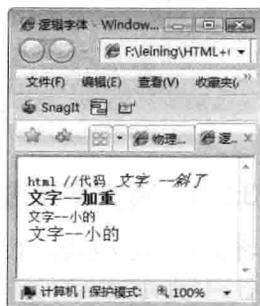


图 3-6 常用逻辑字体

3.5 字体标记实例

本章主要讲解字体标记的用法，字体分为物理字体和逻辑字体，并可以设置字体的大小和颜色等，下面的实例从字体入手，对字体大小、颜色等样式进行演示，以下按步骤讲解。

3.5.1 插入字体

在HTML基本结构代码中，输入字体标记，标记对内的内容为“HTML+CSS完全自学手册”，代表控制这行文本的字体，代码3-7表示文字在内。

代码 3-7 字体实例

```

-----文件名：字体实例.html-----
01 <html>
02   <head>
03     <title>字体实例</title>
04   </head>
05   <body>
06     <!--下面一行表示字体标记-->
07     <font>HTML+CSS完全自学手册</font>    <!--下面都是在这一行添加标记对-->
08   </body>
09 </html>

```

【代码解析】本实例在网页体中输入字体标记，即插入一行文本，代码第7行中的文本被字体标记修饰，在没有设置其他参数的条件下，各参数都为默认值，效果如图3-7所示。

3.5.2 将字体颜色设置为红色

将字体颜色设置为红色，代码变化部分只在内。

```
<!--下面一行表示对字体颜色的设置-->
<font color="red">HTML+CSS完全自学手册</font>
```

上面代码是修改部分，其他部分没变。效果如图 3-8 所示。



图 3-7 插入字体



图 3-8 设为红色字体

3.5.3 将文字设置为标题 h1

将文字设置为标题 h1，代码如下所示。

```
<!--下面一行表示字体加入标题h1-->
<h1><font color="red">HTML+CSS完全自学手册</font></h1>
```

加上标题后的效果如图 3-9 所示。

3.5.4 将文字设置为斜体并加删除线

将文字设置为斜体并加删除线的代码如下所示。

```
<!--下面一行表示在标题字体中设置斜体和删除线-->
<h1><font color="red"><i><s>HTML+CSS完全自学手册</s></i></font></h1>
```

设为斜体并添加删除线后的效果如图 3-10 所示。



图 3-9 设文字为标题



图 3-10 将文字设置为斜体并加删除线

3.5.5 设置字体为<big>

设置字体为<big>的代码如下所示。

```
<!--下面一行表示在原来的代码基本上把字体设置成大号-->
<h1><font color="red"><i><s><big>HTML+CSS完全自学手册</big></s></i></font></h1>
```

字体变大后的效果如图 3-11 所示。

注意 标记对可以包含其他标记，但不能交叉。

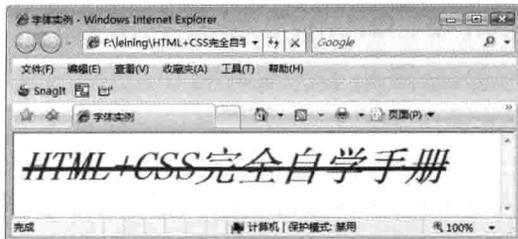


图 3-11 设置字体为 big

3.6 小结

本章所学的都是字体方面的属性，分别介绍了字体的大小、颜色、标题，还有物理字体和逻辑字体。通过对标题的学习，可以把物理字体的标记用在标题标记上，使标题字体更丰富。最后通过实例演示了字体标记的各知识点的用法，共 5 个步骤，一步一个知识点，最后完成的是一个综合效果。学好本章对以后从事网页设计和制作很有帮助，在后面的章节中也常会涉及本章内容。

3.7 习题

一、填空题

1. 标记中的参数 color 后面可以接_____值，如#000000。也可以用_____表示，如 red。
2. 标题用_____表示，_____代表数字，数字 1~6 表示标题大小从大到小。

二、上机实践

1. 按要求在网页中编辑如下文字。

要求：（1）标题要用 3 号大小，倾斜、加粗、用红色。

（2）正文用蓝色显示。

我的第一堂化学课

今天我第一次上化学课，心情比较兴奋，听爸爸妈妈说：“化学课是很有趣的一门学科，可以做很多有趣的实验”。

课堂上，老师向我们解释了身边哪些是化学现象，例如，发光、发热就是化学现象的一种。还跟我们解说了水分子（H₂O）的组成。

【提示】

<p>	<!--p为段落开始符-->
<h3>	<!--主体标识符-->
	<!--加粗标识符-->
<i>	<!--斜体标识符-->
 	<!--空格标识符-->
	<!--设置字体颜色-->

2. 从以下代码中找出错误，更正错误后说明文字最终的显示效果。

```
01 <html>
02 <head>
03 <title>字体实例</title>
04 </head>
05 <body>
06 <h1><font color="red"><b><i><s><big>我爱网页设计 </big></i></s></font></h1><b>
07 </body>
08 </html>
```

【提示】 标记对不能交叉使用。

第4章 文字布局

文字布局指对文字进行处理并设置样式。文字布局在网页中是必不可少的，主要涉及文章的段落、列表等。一篇文章在网页中的美观与文字的布局有很大关系。本章主要对 HTML 的布局标记作讲解。学完本章内容，应掌握以下知识。

- 行的控制；
- 文字对齐；
- 段落设置；
- 列表；
- 文字布局实例。

4.1 行的控制

有时在页面中要对文字的行进行控制或处理，如设置其在某个位置换行、文本对齐等，要对这行的文本设置不同的样式，学习行的控制是很重要的。行的控制主要分为换行控制和不换行控制。以下分别介绍。

4.1.1 换行控制

仔细想想，在前面学习标记是否与行有联系呢？例如空一行出来。是的，就是前面学习过的 `<p></p>` 标记对。在这里告诉读者，`<p>` 也可以单独使用，语法代码如下。

文字一<p>文字二

代码的意思是文字一后面有一对 `<p></p>`，简写为 `<p>`，从网页上看，两行文字中间空出一行，也就是 `<p>` 的作用。代码 4-1 说明了行的控制。

代码 4-1 p 的用法

```
-----文件名: p的用法.html-----
01 <html>
02   <head>
03     <title>p的用法</title>
04   </head>
05   <body>
06     <!--下面一行表示在文本中插入标记<p>-->
07     文字一<p>文字二
08   </body>
09 </html>
```

【代码解析】代码 4-1 的第 7 行说明在文字中间插入标记 `<p>`，使文本到此处中断本行，另起

一段，效果如图 4-1 所示。

在前面也学过一个标记，
——强制换行符。标准的写法如下。

```
<!--下面一行表示换行标记-->
<br />
```

当然，很多人喜欢写成
，这在语法上没有错，但最好是用标准写法。如果存在多个
，会出现多行空行。请读者思考为什么。代码 4-2 说明
的用法。

代码 4-2 br 的用法

```
-----文件名: br的用法.html-----
01 <html>
02   <head>
03     <title><br>的用法</title>
04   </head>
05   <body>
06     文字一
07     <!--下面一行表示多个换行符-->
08     <br>
09     <br>
10     <br>
11     <br>
12     文字二
13   </body>
14 </html>
```

【代码解析】在第 6~12 代码中的两行文字中间加了 4 个换行符，即出现 3 个空行（其中第 1 个换行标记表示在第 1 行中换行，其他的表示在空行中换行）。强制换行后的效果如图 4-2 所示。



图 4-1 <p>的用法



图 4-2 强制换行

4.1.2 不换行控制

有读者可能会问，能不能不换行一直显示下去，回答是肯定的。如需要不换行控制，用标记对<no>br></no>br>表示，标记对中的内容在页面中一直都不会换行，本标记对并不会经常用到。下面列出不换行控制的完整语法。

```
<no>br></no>br>
```

使用本标记时，只要文字在标记里面，不管有多长，都在一行里，在状态栏上会出现左右移

动的框，说明在<noBr><noBr>标记里显示的内容超过了正常宽度。代码 4-3 说明不换行符<noBr>的用法。

代码 4-3 不换行

```
-----文件名：不换行.html-----
01 <html>
02   <head>
03     <title>不换行</title>
04   </head>
05   <body>
06     <!--下面一行表示不换行符-->
07     <noBr>
08       <!--下面一行表示文本被包围在不换行标记中-->
09       HTML+CSS完成自学手册HTML+CSS完成自学手册
10       HTML+CSS完成自学手册HTML+CSS完成自学手册
11       HTML+CSS完成自学手册HTML+CSS完成自学手册
12       HTML+CSS完成自学手册HTML+CSS完成自学手册
13     </noBr>
14   </body>
15 </html>
```

【代码解析】第 7~13 行代码的多个“HTML+CSS 完成自学手册”文本，被标记对<noBr><noBr>包围着，尽管在代码中有 4 行文本，但在网页中出现的效果会不会像代码中那样呢？代码 4-3 的效果如图 4-3 所示。

从代码中可以看到，这与代码<noBr></noBr>中有没有换行和长度大小都没关系。所以，想对行进行控制，可以使用<p>、
、<noBr></noBr>等标记符。

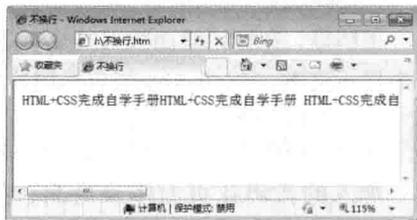


图 4-3 用<noBr>一直没换行

<p />就是<p></p>的简写，只不过<p></p>里面没有内容，所以出现空格。请读者注意，**注意** 如果想空两行是不是就用两个<p>呢？不是的。<p>后面输入再多<p>也是空一行。这与
不同。

4.2 文字对齐

文字对齐也很重要，当对一个段落或层里面的文字作对齐布局时，就会用到 align 标记。对齐标记不会单独使用，往往与某些标记配合使用，如在<p>标记、<table>标记中都会用到，对齐的语法如下。

```
align="#"
```

其中#代码表示方位，可选的有：left（向左）、right（向右）、center（居中）。align 可以放在<p></p>和<div></div>（稍后介绍）里，表示文字在标记中向哪个方面偏移，可以包括文字与图片，也可以全是文字。代码 4-4 表示文字向右对齐。

代码 4-4 文字对齐

```
-----文件名: 文字对齐.html-----
01 <html>
02   <head>
03     <title>文字对齐</title>
04   </head>
05   <body>
06     <!--下面一行表示段落-->
07     <p align="right">HTML+CSS完全自学手册</p>
08   </body>
09 </html>
```

【代码解析】在从第 7 行代码开始的标记对<p></p>中有一行文字，文字在<p></p>中设置为向右偏移，即 align=right，效果如图 4-4 所示。

同理，可以在 align 中设置其他对齐方式，如向左用 align=left，居中用 align=center，位置两侧的引号可有可无。为了表示真正的意思，最好是用引号包围关键字，其他的属性设置都一样。

有一点比较有意思，居中显示也可以不用设置 align 为 center，可以换个方式来居中，其语法如下。

```
<!--下面一行表示居中标记对-->
<center>居中</center>
```

很明显，在标记里面放入的文字是居中的，现在在图 4-4 的代码中加入下面的居中代码，表示居中显示文本，在</body>结束标记前加入以下代码。

```
<!--下面一行表示对文本设置居中-->
<center>HTML+CSS完全自学手册 ---居中</center>
```

加入的代码在页面体最后面，故显示在代码 4-4 中的文本后，向右对齐后面的文本用<center></center>修饰，处于居中状态，效果如图 4-5 所示。



图 4-4 文字向右偏移

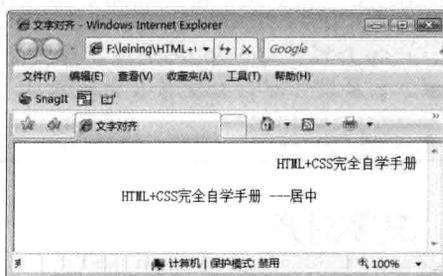


图 4-5 center 居中

注意

对本节的内容如果读者还不清楚的话，在下一章的文字与图片对齐中也会讲到相关内容，包括垂直对齐。

4.3 段落设置

段落设置其实很容易，就是在一张网页上用到段落的地方，会对其布局或字体进行设置，网页中的文章都会有多个段落，所以在段落上要注意。

多个段落有多个段落标记，如下代码表示两个段落。

```
<p>段落一</p>
<p>段落二</p>
```

有多少个段落就有多少个<p>标记，然后对每个段落进行必要的设置。现在用以前学过的知识来练习段落设置，如代码 4-5 所示。

代码 4-5 段落设置

```
-----文件名：段落设置.html-----
01 <html>
02   <head>
03     <title>段落设置</title>
04   </head>
05   <body>
06     <!--下面一行表示给不同段落设置不同的样式-->
07     <p><font color="red">段落一:HTML+CSS完全自学手册</font></p>
08     <p><b>段落二:HTML+CSS完全自学手册</b></p>
09     <p><s>段落三:HTML+CSS完全自学手册</s></p>
10     <p><h1>段落四:HTML+CSS完全自学手册</h1></p>
11   </body>
12 </html>
```

【代码解析】在第 7~10 行代码有 4 个段落，每个段落都设置了不同的样式，如代码第 7 行的段落一颜色设置为红色，代码第 8 行的段落二设置为粗体，代码第 9 行的段落三设置为添加删除线，代码第 10 行的段落四的文本设置为标题的样式。这 4 个段落的效果对比如图 4-6 所示。

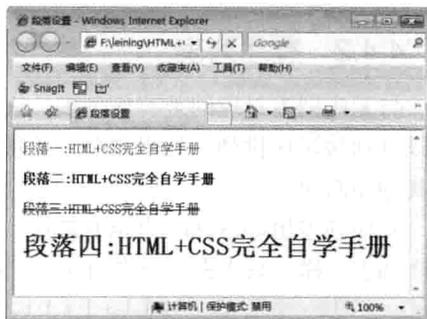


图 4-6 段落设置

4.4 列表

列表分为有序列表和无序列表，在什么情况下用到这两个列表呢？读者在门户网站上浏览新闻时，在网站的新闻列表中，新闻标题前会出现一个小圆点，而有的的是有序号的数字，前者表示的是无序列表，后者表示的是有序列表。本节介绍列表的表示，通过代码可以表现出列表是有序列表还是无序列表。

4.4.1 无序列表

所谓无序列表是指在列表中没有顺序可言，表里的每项都是相同的。列表的语法分两部分。

```
<!--下面一行表示无序列表-->
<ul>
  <li>表项一</li>
  <li>表项二</li>
  <li>表项...</li>
</ul>
```

分析上面的代码，决定无序的是，而只是里面的一列表项。如果想列出更多的表项，那么在里加表项即可。代码 4-6 表示无序列表的用法。

代码 4-6 无序列表

```
-----文件名：无序列表.html-----
01 <html>
02   <head>
03     <title>无序列表</title>
04   </head>
05   <body>
06     <!--下面一行表示无序列表-->
07     <ul>
08       <!--下面一行表示表项-->
09       <li>HTML+CSS完全自学手册</li>
10       <li>HTML+CSS完全自学手册</li>
11       <li>HTML+CSS完全自学手册</li>
12       <li>HTML+CSS完全自学手册</li>
13     </ul>
14   </body>
15 </html>
```

【代码解析】第 7~13 行代码在无序列表符中间添加了 4 行带有列表项的文字，默认状态下无序列表所显示的效果在文本前用圆点表示，效果如图 4-7 所示。

4.4.2 有序列表

与无序列表对应的是有序列表，表项里不用设置就可以自动按顺序排列，初学者乍一看很神奇，现在来揭开其神秘面纱。

有序列表用表示有顺序，里面的表项符与无序列表一样，只代表一个表项而已，在多个表项中，系统自动按顺序排列，语法代码如下。

```
<!--下面一行表示有序列表-->
<ol>
  <li>表项一</li>
  <li>表项二</li>
  <li>表项...</li>
</ol>
```

有序列表与无序列表只相差在上。下面示例的内容与无序列表是一样的（表项都一样），不同的是，用标记对取代标记对。代码 4-7 表示有序列表的用法。

代码 4-7 有序列表

```
-----文件名：有序列表.html-----
01 <html>
02   <head>
03     <title>有序列表</title>
04   </head>
05   <body>
06     <!--下面一行表示有序列表-->
07     <ol>
08       <li>HTML+CSS完全自学手册</li>
09       <li>HTML+CSS完全自学手册</li>
```

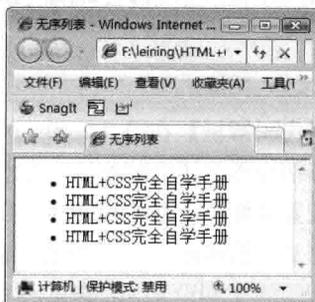


图 4-7 无序列表

```

10         <li>HTML+CSS完全自学手册</li>
11         <li>HTML+CSS完全自学手册</li>
12     </ol>
13 </body>
14 </html>

```

【代码解析】与无序列表不相同的是，有序列表在项前面用数字序号表示，从第1项开始，每加一项序号也加1，效果如图4-8所示。

有序与无序只相差一个字母，但在区别上有明显不同。请读者在合适的位置上使用。在有序列表中还可以用其他有序符取代数字，`<ol type="#">`中的顺序可以表示为A、a、i、I、1等。



图4-8 有序列表

4.5 用其他方式修饰文本

本节对文本的欲格式化、计算机输出、块引用、闪烁等进行讲解。这几种方式在网页中很少用到，但学会了也是一种很好的技能。换言之，在必要的时候会用到。

4.5.1 欲格式化文本

欲格式化文本在网页中输出的是在代码中的样式，因为如果代码没有欲格式化，就会去掉里面的空格，其语法如下。

```

<!--下面一行表示欲格式化标记对-->
<pre>
    文字
</pre>

```

这样，在网页中就会输入文字里面的格式。里面有空格就会输出空格，有换行也会输出换行。代码4-8说明欲格式化文本的用法。

代码4-8 欲格式化文本

```

-----文件名：欲格式化文本.html-----
01 <html>
02   <head>
03     <title>欲格式化文本</title>
04   </head>
05   <body>
06     <!--下面一行表示欲格式化的用法-->
07     <pre>
08         HTML+CSS完全自学手册           HTML+CSS完全自
09 学手册
10     </pre>
11   </body>
12 </html>

```

【代码解析】按正常的格式，在网页中不会出现多个空格，但应用`<pre></pre>`标记对后的效

果就不一样了，如第 7~10 行代码所示，在一行文字里的出现很多的空格，然后再出现文本内容，效果如图 4-9 所示。

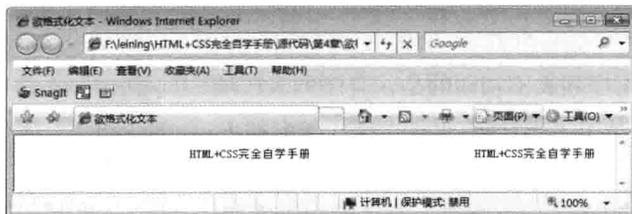


图 4-9 欲格式化 pre

另外一个标记 `xmp` 与 `pre` 的效果很相似，语法形式也是双标记的。在标记对 `<xmp></xmp>` 里面放文本或代码，语法形式如下。

```
<!--下面一行表示欲格式化标记对-->
<xmp>
    文本
</xmp>
```

`xmp` 与 `pre` 的重要区别是前者把 HTML 代码看做文本，而后者会解释 HTML 代码。现在比较二者的区别，代码 4-9 表示把在 `pre` 的代码中加入的 HTML 标记分别输入到两个标记对内。

代码 4-9 欲格式化文本行

```
-----文件名：欲格式化文本行.html-----
01 <html>
02   <head>
03     <title>欲格式化文本</title>
04   </head>
05   <body>
06     <!--下面一行表示pre与xmp的用法比较-->
07     下面是pre代码：
08     <pre>
09       <ol>
10         <li>HTML+CSS完全自学手册      HTML+CSS完全自学手册</li>
11         <br>
12         <li>HTML+CSS完全自学手册      HTML+CSS完全自学手册</li>
13       </ol>
14     </pre>
15     下面是xmp代码：
16     <xmp>
17       <ol>
18         <li>HTML+CSS完全自学手册      HTML+CSS完全自学手册</li>
19         <br>
20         <li>HTML+CSS完全自学手册      HTML+CSS完全自学手册</li>
21       </ol>
22     </xmp>
23   </body>
24 </html>
```

【代码解析】第 8~22 行代码比较 `pre` 与 `xmp` 中文本的区别，效果如图 4-10 所示。



图 4-10 比较 pre 与 xmp

4.5.2 计算机输出格式

现在 IT 技术网站越来越多，要在网页中显示计算机代码的需求也增多了，所以初学者学习计算机输出格式很有必要。那么什么样才是计算机的格式呢？请看下面的语法代码。

```

<!--下面一行表示计算机输出格式标记对-->
<code>
    代码样式小型固定宽度字体显示的文本
</code>
<kbd>
    代码样式固定宽度字体渲染
</kbd>
<tt>
    代码样式固定宽度字体渲染文本
</tt>
<var>
    代码样式斜体字渲染
</var>

```

下面用上面的语法来看看效果，是不是语法里面所说的样式呢？代码 4-10 表示计算机输出格式。

代码 4-10 计算机输出格式

```

-----文件名：计算机输出格式.html-----
01 <html>
02   <head>
03     <title>计算机输出格式</title>
04   </head>
05   <body>
06     <code>
07       <!--以下代码把HTML嵌入到<body>中-->
08       <html>
09         <head>
10           <title>网页标题</title>
11         </head>

```

```

12         <body>
13             正文--这是用html标记表示的效果
14         </body>
15     </html>
16 </code>
17 <br>
18 <kbd>HTML+CSS完全自学手册</kbd>
19 <br>
20 <tt>HTML+CSS完全自学手册</tt>
21 <br>
22 <samp>HTML+CSS完全自学手册</samp>
23 <br>
24 <var>HTML+CSS完全自学手册</var>
25 </body>
26 </html>

```

【代码解析】在代码第 6~16 行的 `<code></code>` 中, 放置了基本的 HTML 结构, 在 `<code></code>` 内所显示的效果是该网页体在 `<code></code>` 范围中所显示的, 计算机输出格式的效果如图 4-11 所示。

4.6 文字布局实例

前面几节讲述了文字布局的知识, 接下来把前面所讲解的知识用实例表现出来。希望通过一个完整的实例演示, 读者会对 HTML 中的文字布局标记更有印象, 学完后能举出更多例子。



图 4-11 计算机输出格式

4.6.1 设置文字换行与不换行

下面输入文字并设置文字换行与不换行。代码 4-11 设置了文字换行与不换行, 其中换行标记用 `
`, 而不换行的标记用 `<no>` 表示。

代码 4-11 文字布局

```

-----文件名: 文字布局手把手.html-----
01 <html>
02     <head>
03         <title>文字布局手把手</title>
04     </head>
05     <body>
06         <p>HTML+CSS完全自学手册</p>
07         <br>
08         <no>
09         <p>HTML+CSS完全自学手册HTML+CSS完全自学手册
10             HTML+CSS完全自学手册HTML+CSS完全自学手册</p>
11         </no>
12     </body>
13 </html>

```

【代码解析】第 7 行代码中的标记 `
` 表示在网页中换行, 而在第 8 行代码中用 `<no>` 标记时, 不管代码中是否出现换行, 在网页中都不会换行, 直到把 `<no></no>` 标记对中的对象显示完,

效果如图 4-12 所示。

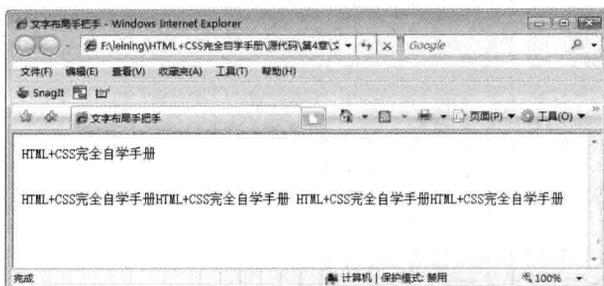


图 4-12 换行与不换行

4.6.2 设置文字对齐

本节分别设置左对齐、居中对齐。以下代码表示文字对齐的方式。

```
01      <p align="right">HTML+CSS完全自学手册</p>
02          <br>
03          <nobr>
04      <p align="center">HTML+CSS完全自学手册HTML+CSS完全自学手册
05      HTML+CSS完全自学手册HTML+CSS完全自学手册</p>
06      </nobr>
```

【代码解析】在没有改变文本的情况下，在第 1 行代码的<p></p>标记对中设置右对齐，在第 4 行代码的<p></p>标记对中设置居中对齐，效果如图 4-13 所示。

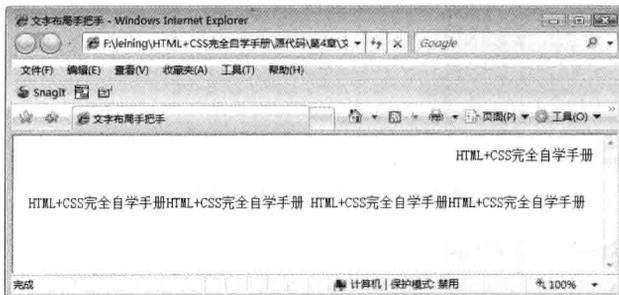


图 4-13 文字对齐

4.6.3 显示列表

在代码 4-10 的基础上，加入有序列表和无序列表，完整的代码如代码 4-12 所示。

代码 4-12 文字布局

```
-----文件名: 文字布局手把手.html-----
01  <html>
02  <head>
03  <title>文字布局手把手</title>
04  </head>
05  <body>
06  <p align="right">HTML+CSS完全自学手册</p>
07  <br>
```

```

08     <nobr>
09     <p align="center">HTML+CSS完全自学手册HTML+CSS完全自学手册
10 HTML+CSS完全自学手册HTML+CSS完全自学手册</p>
11     </nobr>
12     <ol>
13         <li>HTML+CSS完全自学手册</li>
14         <li>HTML+CSS完全自学手册</li>
15         <li>HTML+CSS完全自学手册</li>
16     </ol>
17     <ul>
18         <li>HTML+CSS完全自学手册</li>
19         <li>HTML+CSS完全自学手册</li>
20         <li>HTML+CSS完全自学手册</li>
21     </ul>
22 </body>
23 </html>

```

【代码解析】上面代码在原有代码的基础上加入有序列表和无序列表效果(第12~21行代码),在网页中的效果如图4-14所示。

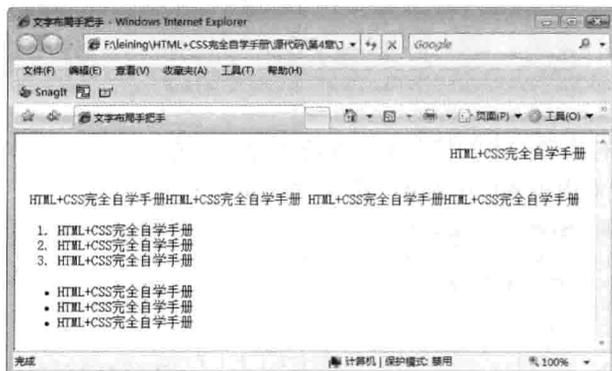


图 4-14 有序列表和无序列表

4.6.4 在网页中显示源代码

输入源代码4-13,以显示计算机格式。

代码 4-13 文字布局

```

-----文件名:文字布局手把手.html-----
01 <html>
02     <head>
03         <title>文字布局手把手</title>
04     </head>
05     <body>
06         <P align="right">HTML+CSS完全自学手册</P>
07         <br>
08         <nobr>
09         <p align="center">HTML+CSS完全自学手册HTML+CSS完全自学手册
10 HTML+CSS完全自学手册HTML+CSS完全自学手册</p>
11     </nobr>
12     <ol>

```

```

13         <li>HTML+CSS完全自学手册</li>
14         <li>HTML+CSS完全自学手册</li>
15         <li>HTML+CSS完全自学手册</li>
16     </ol>
17     <ul>
18         <li>HTML+CSS完全自学手册</li>
19         <li>HTML+CSS完全自学手册</li>
20         <li>HTML+CSS完全自学手册</li>
21     </ul>
22     <pre>
23         int a;
24         int b;
25         float f;
26         f = a/b;
27     </pre>
28 </body>
29 </html>

```

【代码解析】在第22~27行代码的标记对<pre></pre>中，输入源代码，其中每行代码后都有换行，在网页中显示的效果如图4-15所示。

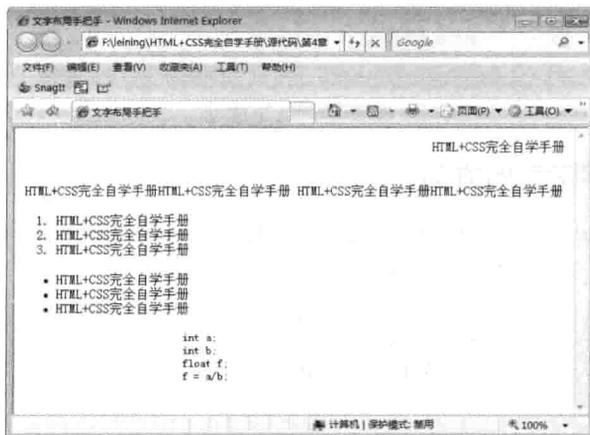


图 4-15 源代码

说明 对于欲格式化标记，在计算机技术方面的网站放置代码时应用较多。

4.7 制作滚动文字

通过前面的学习，读者已经能够很好地控制各种段落文字的显示方式，不过无论怎么设置，文字都是静态的。本节编者讲述 HTML 代码中比较特殊的标签，它能使网页中的文字滚动，并且可以控制其滚动的属性。

4.7.1 设置文字滚动

HTML 技术中使文字滚动的方法是使用双标签<marquee></marquee>。在 HTML 代码中可使其作用区文字滚动，默认为从右到左，循环滚动。在 D:\web\目录下创建网页文件，命名为 mar.htm，

编写代码如 4-14 所示。

代码 4-14 文字滚动

```

01 <html>
02 <head>
03   <title>文字滚动的设置</title>
04 </head>
05 <body>
06 <font size="5" color="#cc0000">
07 文字滚动示例（默认）：<marquee>做人要厚道</marquee>
08 </font>
09 </body>
10 </html>

```

本例在网页中显示的效果如图 4-16 所示。

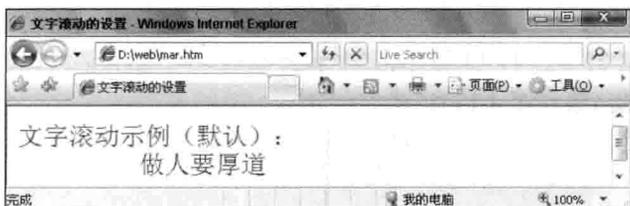


图 4-16 文字滚动

4.7.2 设置文字滚动的方向

`<marquee></marquee>` 标签的 `direction` 属性用于设置内容的滚动方向，属性值有 `left`、`right`、`up`、`down`，分别代表向左、向右、向上、向下，例如以下代码。

```

<marquee direction="left">做人要厚道</marquee>
<marquee direction="right">做人要厚道</marquee>
<marquee direction="up">做人要厚道</marquee>
<marquee direction="down">做人要厚道</marquee>

```

4.7.3 设置文字滚动的速度和形式

设置文字滚动使用 `<marquee></marquee>` 标签，其属性说明如下。

- `<marquee></marquee>` 标签的 `scrollamount` 属性用于设置内容滚动速度。
- `<marquee></marquee>` 标签的 `behavior` 属性用于设置内容滚动方式，默认为 `scroll`，即循环滚动。当其值为 `alternate` 时，内容将来回循环滚动；当其值为 `slide` 时，内容滚动一次即停止，不会循环。还有一个 `loop` 属性可设置其滚动循环次数，默认为没有限制。
- `<marquee></marquee>` 标签的 `scrolldelay` 属性用于设置内容滚动的时间间隔。
- `<marquee></marquee>` 标签的 `bgcolor` 属性用于设置内容滚动背景色（类似于 `body` 的背景色设置）。
- `<marquee></marquee>` 标签的 `width` 属性用于设置内容滚动背景宽度。
- `<marquee></marquee>` 标签的 `height` 属性用于设置内容滚动背景高度。

修改前面文字滚动的网页，编写代码如 4-15 所示。

代码 4-15 文字滚动设置

```

01 <html>
02 <head>
03   <title>文字滚动的设置</title>
04 </head>
05 <body>
06 <font size="3" color="#cc0000">
07 文字滚动示例（默认）：<marquee>做人要厚道</marquee>
08 文字滚动示例（向右）：<marquee direction="right" scrolldelay="500">做人要厚道</marquee>
09 文字滚动示例（向下，滚动方式为slide，速度为10）：
10 <marquee scrollamount="10" behavior="slide">做人要厚道</marquee>
11 文字滚动示例（默认方向，滚动方式为alternate，循环3次，速度为2）：
12 <marquee scrollamount="2" behavior="alternate" loop="3">做人要厚道</marquee>
13 文字滚动示例（向上，背景色为#CCFF66，设置了背景宽度和高度）：
14 <marquee direction="up" bgcolor="#CCFF66" width="250" height="55">做人要厚道
15   </marquee>
16 </font>
17 </body>
18 </html>

```

在浏览器中的浏览效果如图 4-17 所示。

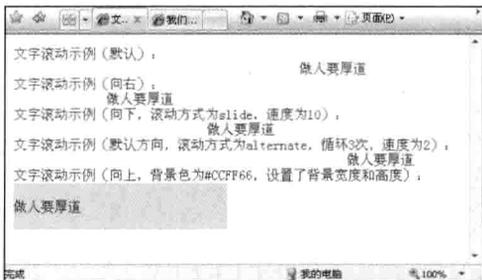


图 4-17 文字滚动设置后

4.8 小结

本章所讲的知识点都以文字为中心，文字中的行、文本对齐、段落和列表，以及对行的控制有换行与不换行的用法，其中列表的应用在布局中非常常见的，如果用 DIV 和 CSS 来取代表格布局，那么就会用到列表标记。这些知识点对网页布局帮助很大，这方面的技巧性比较强，可以用对齐达到美观效果，段落体现层次化，列表显示出内容有条理等。在网页中运用这些内容也是一种艺术，努力达到让页面从感官上就能吸引浏览者的效果。

4.9 习题

一、填空题

1. HTML 技术中使文字滚动的方法是使用双标签_____。
2. 换行控制符有_____、_____，不换行控制符有_____。
3. 无序列表的标签是_____，有序列表的标签是_____。

二、上机实践

1. 按以下文本样式实现网页。

“爸爸，对不起，我也来了。”

首先出来的白发老者以惊异的目光看着小女孩，友善的脸庞变得严肃，嘴巴动着，不知他在说什么。

“你怎么也来了？外星探索有多危险知道吗？！”

“我不怕危险，我也有和您一样的梦想啊。”（注意，这里用蓝色）

“好了，你还是个孩子，在情况未明朗前先回飞船上去，这里太危险了。”

【提示】

```
<p>-----段落标记  
<br>-----换行标记  
<font color>-----字体颜色  
<b>-----加粗  
<pre>-----欲格式化标记对
```

2. 按以下文本样式实现网页。

写作要求如下。

(1) 卷面要求

A. 字迹清楚

B. 卷面干净

(2) 文章要求

A. 语句通顺

B. 没有错别字

【提示】

```
<ol type="#"></ol>-----有序列表
```

第 5 章 图像和链接

图像在网页中应用很广泛，新闻中有图片新闻，其中的相片就是在网页中放置图像的结果。在网络中，光有文字和图像还不够，互联网因有超链接才如此丰富，使网络不限于特定的地理位置，只要鼠标一点，就可以到达全球任意一个站点。超链接人们常说成链接，在以后的学习中以链接形式给出。对于超链接，可以在文字中指定链接，同时也可以图片中设置超链接，而链接的地址可以是网页（静态页面 html 或动态页面 jsp、php、aspx 等），还可以是图片（jpg、gif 等）或其他文件。网上的下载功能就用到了超链接，其链接到将要下载的文件上。本章主要介绍以下内容。

- 图像的基本语法；
- 图像和文字对齐；
- 文字链接；
- 图片链接；
- 锚点的使用；
- 相对路径与绝对路径；
- 图像客户端映射图。

5.1 图像的基本语法

本节学习图像的基本语法，在网页中放入图像很简单，只需在<body></body>标记对中插入图像标记，其中图像标记用来链接图像地址，同时还可以设置其他参数，如宽度、高度等，只要会设置标记的语法点就可以放入漂亮的图片了。不过想控制它还要学几个属性，如宽度 width、对齐 align 等。本节会详细讲解图像的基本操作。

5.1.1 在网页中插入图像

插入图像用到的属性是 src，正确格式，其中#表示图像 url 路径，如下面的代码表示链接一个图像。

```

```

上述代码表示在一个标记中，插入一张名为 cn.jpg 的图片。应当注意，标记是放在<body></body>里面的，大部分标记都是在网页体中放置。代码 5-1 表示插入图像的完整代码。

代码 5-1 插入图像

```
-----文件名：插入图像.html-----
01 <html>
02   <head>
03     <title>插入图像</title>
```

```

04     </head>
05     <body>
06         <!--以下是插入图像-->
07         
08     </body>
09 </html>

```

【代码解析】已在网页上成功插入一张图片，在学习其他参数设置时，只需在上面代码第7行插入图像标记，同理可设置其他参数。在默认条件下，显示的是图像文件的原图，网页效果如图5-1所示。

5.1.2 图像无法显示时的提示信息

在浏览网页时，会出现不能正常显示图像的问题，如图像地址不存在等，此时图像右上角有叉显示，表示未正常显示图像。那么有没有这种情况，当图像不能正常显示时，而又要在第一时间让用户知道这图像表达的信息？答案是肯定的。这就是将要讲的第二个参数alt。alt表示鼠标指针在图片上面时显示交互文本信息，当图像非正常显示时，在图像的区域显示alt中的文本，即使没有看到图像，也可以给用户该图像的主题。

在实现应用的时候，为了避免出现图像不能正常显示的情况，要设置alt参数，首先，在正常情况下显示图片，当鼠标指针在图像的位置时显示alt设置中引号里面的文字。其次，在图像未正常显示时，也给用户提供图像的信息，对信息的设置刚好是alt的设置。下面代码用来描述图片的信息。

```

```

对图像cn.jpg设置alt参数，在里面设置alt为“风景”，当显示图像时，会有提示。同理，当图像显示不正常时，会出现什么样的效果，用过alt参数就知道答案了。代码5-2表示alt的用法。

代码 5-2 alt 用法

```

-----文件名: alt用法.html-----
01 <html>
02   <head>
03     <title>alt用法</title>
04   </head>
05   <body>
06     <!--下面一行表示在图像标记中设置alt-->
07     
08   </body>
09 </html>

```

【代码解析】当出现正常的图像时，效果如图5-2所示；当由于其他原因使图像不能显示时，有alt的设置效果如图5-3所示。



图 5-1 插入图像



图 5-2 图像正常显示

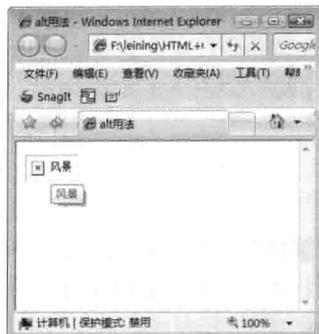


图 5-3 图像非正常显示

5.1.3 控制图像的大小

当图像不是想要的大小时,或者要给不同图像设置同一个宽度时,就得在图像标记中设置对象的大小。在前面提到过设置图像的大小,这里的大小包括宽度和高度,这样在做网页时就可以控制图像在特定位置上的大小了。图像大小的设置非常容易,就是在上面的标记中,用 width 代表宽度, height 代表高度,下图是 200*150 像素大小的图像,把它设置成 400*300 像素的图后,大小变化了,效果也不一样。以下代码是在代码 5-2 中的标记中加入“width=200px”和“height=150px”的设置,px 表示像素,属于长度单位。

```
<!--下面一行表示对图像标记设置宽度和高度-->
```

```

```

在标记中设置了宽度 (width) 和高度 (height) 后,显示的图像大小变化了,效果如图 5-4 所示。

图像的基本语法先学这么多,是在网页中常用的标记。在实践中多多体会其用法,更多的信息可以查阅相关书籍或上网查询。总之,动手实践是学 HTML 的最佳方法。



图 5-4 控制图像大小

5.2 图像和文字对齐

制作网页时,会遇到有图像和文字的情况。有时要把图像放在左边,文本在右边显示。有时要求文本在图像的中间等。那怎么来处理这些情况呢?还好,用里面的 align 来决定是左还是在右,是上还是下。知道了这个,就可以将图片放置在理想位置了。

align 后面可以是 top (顶)、right (右)、bottom (低)、left (左),还有 middle (中间)。图像与文本 top 对齐方式的设置如代码 5-3 所示。

代码 5-3 图像 top 对齐

```
-----文件名: 图像top对齐.html-----
01 <html>
02 <head>
```

```

03         <title>图像top对齐</title>
04     </head>
05     <body>
06         <!--下面一行表示图像与文本对齐顶部-->
07         图像与文本top对齐
08     </body>
09 </html>

```

【代码解析】顶部对齐代码如下所示，第7行代码表示文本与图像顶端对齐，即文本出现在与图像顶端平行处，起作用的代码如下所示，效果如图5-5所示。

```

<!--下面一行表示对齐顶部-->
top对齐

```

与代码5-3一样，居中对齐的代码如下所示。区别就是 align 的设置不同，下面的代码在代码5-3的基础上，设置 align 为 middle 后，效果如图5-6所示。

```

<!--下面一行表示图像与文本居中对齐-->
图像与文本middle对齐

```



图 5-5 top 对齐



图 5-6 居中对齐

类似地，可以设置左对齐、右对齐等不同的对齐。对参数的设置不同，其效果也不同。记住，默认的是底部对齐，所以当用 align="bottom"设置对齐底部时，即是默认设置，此时，设置与否效果都是一样的。

5.3 图像的边框

有时在图像四周加边框可以达到美化效果。这样可以确定该图像的实现大小。也有时候，不想让图像边框显示。所以如何处理图像边框呢？下面来看看 border——边框。border 在上可以用到，在以后会学到的 CSS 中，它也可以控制层的边框。

在有超链接时默认是有边框的，也可以将其设置成没有超链接。

```

<!--下面一行表示去掉超链接后的图像边框-->
<a href="http://www.baidu.com"></a>

```

上面的代码把链接后面有边框的图像设置成无边框，其关键代码 border="0px"起作用，当鼠标指针放到图像上面时，呈手形，表示有链接，与默认链接不同的是，该图像的边框设置为 0，如图5-7所示。

现在，使用不带链接的效果得到边框，即去掉里面的超链接，用边框参数 border 设置图像的

边框，在标记中加入 border 为 10px 的边框效果，代码如下。

```
<!--下面一行表示设置图像边框-->

```

当在图像标记中设置 border 时，对单位设置很重要，本例中设置为 10px，即边框为 10 像素，效果如图 5-8 所示。



图 5-7 用 border 去掉默认边框



图 5-8 border 为 10 像素的边框

5.4 图像映射图

有时，单击同一图像中不同位置，会出现不同的链接，有矩形、圆形、多边形等。这些分别用 rect、circle 和 poly 实现，结果是图像相同，单击位置不同就会使链接不同。

在里放的是 usermap，后面接地图名。在声明地图时，用<map></map>包围。<map>里放 name 表示它的名称，以便在其他地方调用。在<map></map>里有区域，由<area>属性指定用什么形状，后面紧跟 Shape。Shape 后面接 rect、circle 和 poly 中的一个，coords 里面是像素，href 是要链接的地址。具体实例请看图像实例。

5.5 图像的应用实例

在前面几节中用到常用的知识点，也学会了在图像操作中的部分控制。为了让读者更好地理解图像，以一个系统的实例一步一步地阐述相关知识。

下面从插入图像到最后完成图像整个工作，一步一步给大家演示。首先到网上找一张 600*450 像素的风景图。现在开始第一步。

5.5.1 在网页上引入图像标记

先在网页上输入的标记，表明用的是图像属性，其后所接的为图像链接地址，别忘记了还有其他标记。以后每一步都在中操作，即其他标记不会变化，如代码 5-4 所示。

代码 5-4 图像实例

```
-----文件名：图像实例手把手.html-----
01 <html>
02   <head>
03     <title>图像实例手把手</title>
```

```

04     </head>
05     <body>
06         <!-- 下面一行表示插入图像-->
07         
08     </body>
09 </html>

```

【代码解析】 在新建的网页中，第 5~8 行代码在标记对<body></body>中引入标记，通过图像标记中的参数 src 引入图像名称，本实例中在同目录下的图像名为 html.jpg。插入图像后的效果如图 5-9 所示。

5.5.2 设置图像参数 alt

标记中的 alt 参数，描述图像的含义。本例中的图像为风景图，故设置 alt 为“风景中的桥”，当鼠标指针放置到图像上时，会显示出这行文字。同理，设置图像的大小用参数 width 表示宽度，height 表示高度。本实例说明图像是风景中的桥并设置图像宽为 120 像素，高为 90 像素，效果如图 5-10 所示。

```

<!-- 下面一行表示设置图像的alt和标记大小-->


```



图 5-9 插入图像



图 5-10 带有交互文本的缩小图像

5.5.3 给图像设置超链接

给图像设置一个超链接指向百度，代码如下，效果如图 5-11 所示。

```

<!-- 下面一行表示设置图像超链接-->
<a href="http://www.baidu.com"></a>

```

5.5.4 设置图像的边框

去掉超链接，并设置边框为 10 像素，在原来代码的基础上加上边框标记前，设置边框值为 10 像素，代码如下所示。

```

<!-- 下面一行表示去掉超链接-->


```

给图像设置边框，即在标记中，设置参数 border 就表示边框，边框值设置为长度单位，

如本实例中是 10 像素，表示边框为 10 像素，数值越大，图像的边框也就越宽，效果如图 5-12 所示。



图 5-11 给图像设置超链接



图 5-12 设置边框

5.5.5 设置文本与图像对齐

加入一部分文字，对文字设置上、中、下对齐。联想一下，图样放在左边，就在图像右边输入文字，反之在左边输入文字。

```
<!--下面代码表示在段落中设置对齐-->
01 <p>文字在底部显示文字在
02 底部显示</p>
03 <p>文字在中间显示文字在
04 中间显示</p>
05 <p>文字在顶端显示文字在
06 顶端显示</p>
```

【代码解析】在上面代码中，对图像与文本进行了设置，说明图像与文本是可以对齐的，通过设置 align 的参数，改变图像与文本的对齐方式，效果如图 5-13 所示（图 5-13 展示了三种对齐方式，分别为：底部对齐、居中对齐和顶部对齐）。

5.5.6 设置图像的映射图

为图像设置多个映射图，代码如下所示。

```
<!--下面一行表示对图像设置映射图-->
01 
02     <map name="Face">
03         <area shape="rect" href="page.html" coords="140,20,280,60">
04         <area shape="poly" href="image.html" coords="100,100,180,80,200,140">
05         <area shape="circle" href="new.html" coords="80,100,60">
06     </map>
```

【代码解析】在<body></body>标记对中加上以上第 2~6 行代码，表示在图像 html.jpg 中设置映射图，其中有三个热点区域，即 coords 中的坐标范围表示的区域，同时这些区域中都设置了超链接，分别链接到不同的页面，在图 5-14 中，当鼠标指针放在图像上面时，在热点区域会出现手形状态，表示可链接到其他页面。



图 5-13 图像与文字对齐



图 5-14 映射图

技巧 热点区域可以用Dreamweaver CS5画出来,即选择图像后,在下面属性栏中的左边有圆形、多边形和四边形的按钮,单击其中一个后,在图像上可以画出对应的区域,在后面链接中设置目标地址即可设置映射图,免去了检测图像的位置。

5.6 文字链接

前面学过的文字都是以文本形式出现的,可能很多读者想知道为什么看新闻的时候,单击新闻上的标题就跳到另一网页中。这是因为文字上加了超链接,所以链接到了其他页面中。现在开始学习文字链接。

5.6.1 最简单的文字链接

文字链接就是为文本加上超链接,以达到跳转网页效果。文字链接的语法如下。

```
<!--下一行表示设置文本链接-->
<a href="">文字链接</a>
```

其中在引号里面输入要链接到的网页地址,文字链接被<a>标记对包围着,代表超链接。里面的 href 属性说明文字链接的地址。属性里面可以是 url 地址、图片地址或影视文件等。

代码 5-5 演示了文字链接的用法。

代码 5-5 文字链接

```
-----文件名: 文字链接.html-----
01 <html>
02     <head>
03         <title>文字链接</title>
04     </head>
05     <body>
06         <!--下一行表示给"163网站"设置链接并指向163首页-->
07         <a href="http://www.163.com">163网站</a>
```

```
08     </body>
09 </html>
```

【代码解析】 代码第7行在163网站的文本中设置超链接，链接地址是 <http://www.163.com>，链接后的文本颜色默认是蓝色，同时在链接的文本下面有一条下划线，当鼠标指针移动到文本上面时，形状就变成手形，如图5-15所示。

说明

超链接的文字默认是蓝色，鼠标指针放上去后变成手形。



图5-15 文字链接

5.6.2 超链接网页的打开方式

知道了<a>是链接代码且 href 表示路径地址后，再来看一个重要属性——target 属性，它也是放在起始标记里面，表示所链接的网页在浏览器中的打开方式，后面接一个参数，其值可以是 _blank、_parent、_self 或 _top，这些参数代表的含义如下所示。

- **_blank**：在新浏览器窗口中打开网页。比较常用。
- **_parent**：将要链接的文件载入含有该链接框架的父框架集或父窗口中。如果含有该链接的框架不是嵌套的，则在浏览器全屏窗口中载入链接的文件，就像_self 参数一样。
- **_self**：在同一框架或窗口中打开所链接的文档。此参数为默认值，通常不用指定。
- **_top**：在当前的整个浏览器窗口中打开所链接的文档，因而会删除所有框架。

代码5-6分别用上面四个知识点设置四个链接。

代码5-6 target 用法

```
-----文件名: target用法.html-----
01 <html>
02     <head>
03         <title>文字链接</title>
04     </head>
05     <body>
06         <!--下一行表示设置链接参数_blank值-->
07         <a href="http://www.163.com" target="_blank">163网站--新浏览器打开
08     </a><br>
09         <!--下一行表示设置链接_parent值-->
10         <a href="http://www.163.com" target="_parent">163网站--父窗口中打开
11     </a><br>
12         <!--下一行表示设置链接_self值-->
13         <a href="http://www.163.com" target="_self">163网站--同一框架或窗口中打开
14     </a>
15     </body>
16 </html>
```

【代码解析】 代码5-2在3行文本中都设置了文本链接，指向 www.163.com 网站，但设置的 target 值不同，使得单击这些链接时，打开窗口的方式不同。在代码第7行设置为 _blank，表示在浏览器中打开新的窗口；代码第10行的文本链接设置为 _parent，表示在父窗口中打开；代码第13行，即最后一个文本链接中，设置 target 值为 _self，表示在同一窗口中打开网页。三种方式的效果如图5-16所示。

说明 target的设置多试几次就很明白了，仔细观察开启新网页的不同形式。

5.6.3 链接的注释

除了前面介绍的链接属性之外，还有一种链接属性（即注释）。当鼠标指针放在文本链接上时，稍后会出现一行文字注释，关键字为title，语法形式如下。

```
<!--下一行表示对链接设置注释-->
<a href="http://www.163.com" title="网易网站" >163网站</a>
```

title的值就是链接的注释，当鼠标指针在文字上面，它就会出现在鼠标指针后面。代码5-7演示了title的用法。

代码 5-7 title 用法

```
-----文件名: title用法.html-----
01 <html>
02   <head>
03     <title>文字链接</title>
04   </head>
05   <body>
06     <!--下一行表示对“163网站”链接设置title注释-->
07     <a href="http://www.163.com" title="网易网站" >163网站</a>
08   </body>
09 </html>
```

【代码解析】代码第7行除对文本“163网站”设置超链接外，还设置了title值，即为本链接添加注释，当鼠标指针移动到文字上面时，注释内容即在鼠标指针后面显示。这种效果和使用图像时的alt参数不同的是，这是文本链接，而且用title表示注释，而图像在中用alt表示。上述代码的效果如图5-17所示。

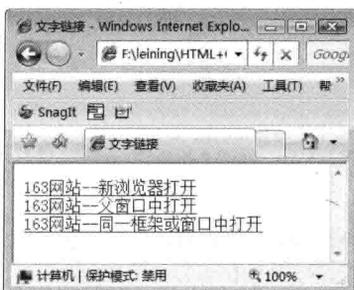


图 5-16 target 用法



图 5-17 title 用法

注意

除了文本可以表达出信息外，还可以在title里添加类似信息表示注释，使主题更丰富，更容易理解。

5.7 图片链接

文本链接是相对文本内容而言的，同理，图像链接是相对图片而言的。对图片设置链接在前面

已介绍，也有相关实例，本节将系统地讲解图片链接内容。单击图片后，通过链接也可以指向全球任意位置的站点，目标可以是网页，也可以是文件。如果文件不能被浏览器解析成网页和图片等，可能会用于下载；在网页中，为了表示下载链接的效果，通常用图片来展现下载的图像效果，然后链接文件。当单击链接时，就下载文件，此时将会出现下载选项。

图片链接顾名思义就是对图片设置超链接。通过上一节的学习，知道了文字链接是在标记对里插入文字，同理，图片链接就是在链接标记对里放入图片地址。单击图片同样有超链接的效果。语法形式如下。

```
<!--下一行代码表示设置图片链接-->
<a href=""><img src=""></a>
```

在第4章学了图片的插入，上面的代码是在插入的图片中设置超链接，图片默认有边框。代码5-8说明了图片链接。

代码 5-8 图片链接

```
-----文件名：图片链接.html-----
01 <html>
02   <head>
03     <title>图片链接</title>
04   </head>
05   <body>
06     <!--下一行代码表示设置html.jpg链接到163网站-->
07     <a href="http://www.163.com"></a>
08   </body>
09 </html>
```

【代码解析】对图片（或图像）链接，即在网页中插入图片的同时，第7行代码在标记

技巧

也可以在图片链接中用title属性，在图片标记中用alt，表示对该图片注释，不管有没有链接都会出现。



图 5-18 图片链接

5.8 锚点链接

锚点常常在很长的文章中用到，以便链接到页面中的具体位置。锚点由name属性来创建一个命名的锚，使用命名锚后，单击带有命名锚的地址直接跳转到一个页面中的锚点位置。这样打开网页后直接指向某一段落，从而不用从上到下慢慢找。

命名锚的语法如下。

```
<!--下一行代码表示在链接时用name对文本设置锚点-->
<a name="锚点名">文章中的文字</a>
```

定义命名锚后，链接中要指定这个命名锚，在其名前加上“#”。使用name属性，要设置两方面：一是设定name的名称；二是设置一个href指向这个name。代码5-9演示了锚的用法。

代码 5-9 锚点链接

```
-----文件名: 锚点链接.html-----
01 <html>
02   <head>
03     <title>锚点链接</title>
04   </head>
05   <body>
06     <a href="#html">点击到锚点链接</a>
07     <br>
08     <br>
09     <a name="html">锚点链接</a>
10   </body>
11 </html>
```

【代码解析】第 9 行代码定义了命名锚，第 6 行代码的文字链接指向这个锚，当链接接入本网页中，在 URL（即网页地址）后面的 # 字符串链接到本页中的字符串。当然，字符串要存在才可以指定到字符串的位置。代码的浏览效果如图 5-19 所示。

注意 在文章内容比较多时，用锚点链接来指定位置，即指定到中间一段落，在网页中时常用到。



图 5-19 锚点链接

5.9 邮箱地址链接

邮箱地址在网页上也常用到，网站工作人员为了与用户交流互动，会在网页中设置邮箱地址链接，只要单击链接就会自动弹出 Outlook Express 的邮箱发送页面。这种方式在国内很少用，国内用户往往会复制邮箱地址直接从邮箱里面发送。不管怎样，学好邮箱链接对在网页上放置邮箱地址是很有用的。

邮箱链接表示给被链接的邮箱发邮件时，单击该邮箱地址，将出现向该地址发邮件的页面，其语法如下。

```
<!--下一行代码表示设置邮箱链接-->
<a href="mailto:leiningcn@163.com">作者的邮箱</a>
```

在链接标记内 href 属性中加入 mailto，然后加冒号，后面接上邮箱地址即可。这样一个邮箱链接就设置好了。与其他链接的区别是多了“mailto”，别忘记后面有冒号。代码 5-10 演示邮箱链接的用法。

代码 5-10 邮箱地址链接

```
-----文件名: 邮箱地址链接.html-----
01 <html>
02   <head>
03     <title>邮箱链接</title>
04   </head>
05   <body>
06     <!--下一行代码表示设置邮件链接-->
07     <a href="mailto:leiningcn@163.com">作者的邮箱</a>
08   </body>
09 </html>
```

【代码解析】在代码 5-10 中，对文本“作者的邮箱”设置链接。与文本链接和图片链接不同的是，邮箱链接在<a>标记对中的 href 后要加上 mailto，然后加冒号，再接上链接的地址。效果如图 5-20 所示。

技巧

邮箱链接的文字表达可以多样化，如可以直接显示邮箱地址，或者可以用“联系我们”等文本取代。



图 5-20 邮箱链接

5.10 相对路径与绝对路径

路径在网页中经常出现，除了用到超链接外，还会用到背景音乐地址、CSS 文件等。在引用文件时（如加入超链接，或者插入图片等），如果使用了错误的文件路径，就会导致引用失效（无法浏览链接文件，或无法显示插入的图片等）。为了避免这些错误，正确地引用文件，需要学习一下 HTML 路径。

HTML 有两种路径：相对路径和绝对路径。下面分别对两者进行介绍，通过知识的讲解，使读者对相对路径和绝对路径有更深刻的了解。

5.10.1 相对路径

相对路径是指这个文件所在的位置与其他文件或文件夹的关系，也可以说是相对位置。如 HTML 同一个目录的文件引用 html 超链接的代码如下所示。

```
c:\Inetpub
```

如果源文件和引用文件在同一个目录里，直接写引用文件名即可。现在先创建一个源文件 info.html，在 info.html 里要引用 index.html 文件作为超链接。为表示源文件所在目录的上一级目录，假设 info.html 的路径是 c:\Inetpub\wwwroot\sites\blabla\info.html，假设 index.html 的路径是 c:\Inetpub\wwwroot\sites\blabla\index.html，在 info.html 加入 index.html 超链接的代码如下所示。

```
<a href = "index.html">index.html</a>
```

为了表示上级目录，用../表示源文件所在目录的上一级目录，用../..表示源文件所在目录的上级目录，依次类推。假设 info.html 的路径是 c:\Inetpub\wwwroot\sites\blabla\info.html，假设 index.html 的路径是 c:\Inetpub\wwwroot\sites\index.html，在 info.html 加入 index.html 超链接的代码如下所示。

```
<a href = "../index.html">index.html</a>
```

假设 info.html 的路径是 c:\Inetpub\wwwroot\sites\blabla\info.html，假设 index.html 的路径是 c:\Inetpub\wwwroot\index.html，在 info.html 加入 index.html 超链接的代码如下所示。

```
<a href = "../../index.html">index.html</a>
```

假设 info.html 的路径是 c:\Inetpub\wwwroot\sites\blabla\info.html，假设 index.html 的路径是 c:\Inetpub\wwwroot\sites\wowstory\index.html，在 info.html 加入 index.html 超链接的代码如下所示。

```
<a href = "../wowstory/index.html">index.html</a>
```

为了表示下级目录，引用下级目录的文件，直接写下级目录文件的路径即可。假设 info.html 的路径是 c:\Inetpub\wwwroot\sites\blabla\info.html，假设 index.html 的路径是 c:\inetpub\wwwroot\

sites\blabla\html\index.html, 在 info.html 加入 index.html 超链接的代码如下所示。

```
<a href = "html/index.html">index.html</a>
```

假设 info.html 的路径是 c:\inetpub\wwwroot\sites\blabla\info.html, 假设 index.html 的路径是 c:\inetpub\wwwroot\sites\blabla\html\tutorials\index.html, 在 info.html 加入 index.html 超链接的代码如下所示。

```
<a href = "html/tutorials/index.html">index.html</a>
```

注意

注意区别路径中的上一级、同一级和下一级, 以及它们的用法, 这些方法在各种编程语言中都会用到。

5.10.2 绝对路径

HTML 绝对路径指带域名文件的完整路径。在描述程序路径时, 用../表示源文件所在目录的上级目录, 依次类推。要正确地引用文件, 需要学习 HTML 路径。

假设用户注册了域名 www.blabla.cn, 并申请了虚拟主机, 虚拟主机提供商会给你一个目录, 例如 www, 这个 www 就是网站的根目录。

假设在 www 根目录下放了一个文件 index.html, 这个文件的绝对路径就是 http://www.blabla.cn/index.html。

假设在 www 根目录下创建了一个目录(叫 html_tutorials), 然后在该目录下放了一个文件 index.html, 这个文件的绝对路径就是 http://www.blabla.cn/html_tutorials/index.html。

通过上面的讲解, 对相对路径和绝对路径就有了清楚的认识, 下一小节的实例将会演示两者的用法。

注意 绝对路径就是地址的一条全部路径, 如一个网站, 或本地电脑的一个文本路径。

5.11 链接应用实例

下面设置文字、图片、邮箱等链接, 通过实例来加深对链接的印象。在网页中链接作用很大, 但有的细节也不可忽视。以下实例是在一张网页中设置各种链接效果, 请看下面的步骤。

5.11.1 在网页中做文字链接

在网页中设置文字链接, 并设置成在新网页中弹出和链接注释。首先输入文字, 然后对部分文字设置链接。代码 5-11 是一个链接实例。

代码 5-11 链接实例 1

```
-----文件名: 链接实例手把手.html-----
01 <html>
02 <head>
03 <title>链接实例手把手</title>
04 </head>
05 <body>
06 <!--设置链接且有title-->
07 <p>HTML+CSS完全自学手册--<a href="http://www.163.com" target="_blank"
```

```

08         title="链接到163网站">网易</a></p>
09     <!--设置不同的链接-->
10     <p>HTML+CSS完全自学手册--<a href="http://www.sina.com.cn" target="_blank"
11         title="链接到新浪网站">新浪</a></p>
12 </body>
13 </html>

```

【代码解析】第7~10行代码中在说明文字里面设置链接,并设置在新窗口中打开网页且对链接添加注释,效果如图5-21所示。

5.11.2 给图片设置超链接

将一张图片插入到代码中设置链接指向网易首页,并用alt说明图片的内容,如代码5-12所示。

代码 5-12 链接实例 2

```

-----文件名: 链接实例手把手.html-----
01 <html>
02 <head>
03 <title>链接实例手把手</title>
04 </head>
05 <body>
06     <!--设置文本链接-->
07     <p>HTML+CSS完全自学手册--<a href="http://www.163.com" target="_blank" title="链接
08 到163网站">网易</a></p>
09     <p>HTML+CSS完全自学手册--<a href="http://www.sina.com.cn" target="_blank" title="
10 链接到新浪网站">新浪</a></p>
11     <p>下面的图片是一座桥,点图片可链接到网易首页。</p>
12     <!--设置图片链接,并设置图片的alt参数-->
13     <p><a href="http://www.163.com"></a></p>
15 </body>
16 </html>

```

【代码解析】第13行代码加一行文字和一张图片的链接,在图片链接中用alt说明图片会链接到www.163.com。浏览效果如图5-22所示。



图 5-21 文字链接



图 5-22 图片链接

注意

与文字链接一样，图片链接也有 `_blank`、`_self` 等对 `target` 属性值。若想去掉图片的边框，用 `` 表示边框为 0。

5.11.3 设置锚点链接

在网页最下面设置一个锚点链接。有时在一张网页中会出现很长的段落，用锚点链接可以直接链接到网页中某一处，如代码 5-13 所示。

代码 5-13 设置锚点链接

```
-----文件名：设置锚点链接.html-----
01 <html>
02 <head>
03     <title>链接实例手把手</title>
04 </head>
05 <body>
06 <a href="#htmlbook">单击到锚点处</a>
07     <!--设置文本链接-->
08     <p>HTML+CSS完全自学手册--<a href="http://www.163.com" target="_blank" title="链接
09 到163网站">网易</a></p>
10 <p>HTML+CSS完全自学手册--<a href="http://www.sina.com.cn" target="_blank" title="
11 链接到新浪网站">新浪</a></p>
12     <p>下面的图片是一座桥，点图片可链接到网易首页。</p>
13     <!--设置图像链接-->
14     <p><a href="http://www.163.com"></a></p>
16 <!--设置多个段落-->
17     <p>HTML+CSS完全自学手册</p>
18     <p>HTML+CSS完全自学手册</p>
19     <p>HTML+CSS完全自学手册</p>
20     <p>HTML+CSS完全自学手册</p>
21     <p>HTML+CSS完全自学手册</p>
22     <a name="htmlbook">在这里设置一个锚点。在网页第一页做锚点链接。</a>
23     <p>HTML+CSS完全自学手册</p>
24     <p>HTML+CSS完全自学手册</p>
25     <p>HTML+CSS完全自学手册</p>
26     <p>HTML+CSS完全自学手册</p>
27     <p>HTML+CSS完全自学手册</p>
28     <p>HTML+CSS完全自学手册</p>
29     <p>HTML+CSS完全自学手册</p>
30     <p>HTML+CSS完全自学手册</p>
31     <p>HTML+CSS完全自学手册</p>
32     <p>HTML+CSS完全自学手册</p>
33     <p>HTML+CSS完全自学手册</p>
34     <p>HTML+CSS完全自学手册</p>
35 </body>
36 </html>
```

【代码解析】 在第 22 行代码中，锚点在简单段落的中间（为了突出锚点在中间，作者故意加了很多段落），锚点链接在第 6 行代码中。图 5-23 所示为锚点链接前的效果，图 5-24 所示为锚点链接后的效果。

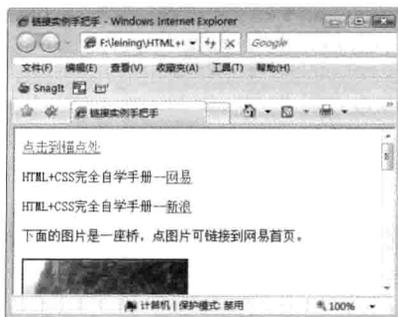


图 5-23 单击锚点链接前

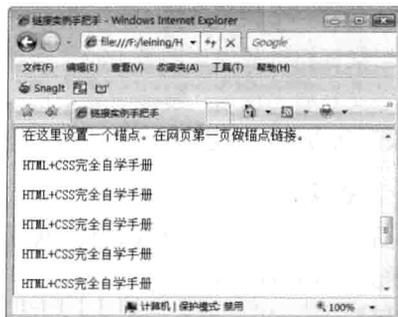


图 5-24 单击锚点链接后

技巧

在同一网页中可以有很多个锚点和锚点链接, 链接也可以设置在本网页以外, 在 URL 地址后加“#”后再加锚点名称。

5.11.4 邮箱链接

设置邮箱链接很容易, 在<a>标记中添加一个 mailto, 可将邮箱链接添加在网页中的任意位置。在代码中添加邮箱链接如下所示。

```
<!--设置邮箱链接-->
```

```
<a href="mailto:bnining.com@163.com" title="邮箱链接">单击这给作者发邮件</a>
```

以上代码是为文本“单击这给作者发邮件”设置链接, 本链接与其他链接不同的是, 该文本的链接地址为邮箱地址, 用到的关键字为“mailto”, 其后所接为邮件地址, 当然, 也可以用 title 表示标记, 效果如图 5-25 所示。

注意 现在很多邮件链接用#代替@, 主要为了防止被网络中专门收集邮件的软件找到该邮箱地址, 以免收到垃圾邮件或广告邮件。

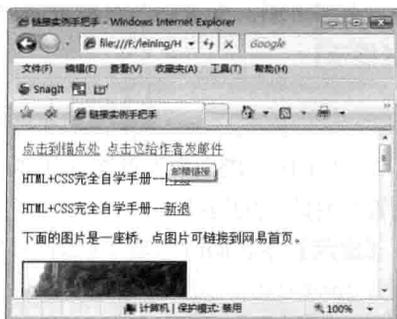


图 5-25 邮箱链接

5.12 小结

本章首先介绍图像的用法, 从图像的语法开始, 讲述如何在 HTML 代码中使用图像, 如何在图像中设置其他参数, 如宽度、高度和边框等, 这些参数都是在标记中设置的。然后又介绍了链接的用法, 从常用的文字链接到图片链接, 再到锚点链接。最后介绍了相对地址与绝对地址, 这些在文件目录中常用到(如 URL 地址、网页和文件设置链接时指向的地址)。

5.13 习题

一、填空题

1. HTML 有两种路径表示形式: _____ 和 _____。

2. 随意写一个邮箱链接：_____。
3. 当鼠标指针放置到图像上时，会显示文字的话用_____标签。
4. 定义好命名锚后，链接中要指定这个命名锚，在其名前加上“_____”。使用 name 属性，要设置两方面：一是_____；二是_____。

二、上机实践

1. 插入一张图片，标题信息为：网页制作。图片宽度为 120 像素，高度为 90 像素。设置一个超级链接到 www.baidu.com。

【提示】

```
src-----图片源
alt-----标题信息
href-----链接标记
width-----宽度
height-----高度
```

2. 给上题中的图片增加 5 像素的边框，设置一个圆形映射对象，文字位于图片顶端。

【提示】

```
align-----文本对齐方式
border-----边框
map-----区域对象映射
```

3. 在网页中，给一张图片和一行文本设置超链接为 www.baidu.com，图片的链接打开方式为打开一个新页面，文本则为在原页面打开链接页面。

【提示】

```
<a href>-----链接标记
<target>-----链接打开方式标记
```

4. 在下面一段话中设置锚点链接，在页面底部添加一个邮箱地址，此页面上层路径为 a.html，请设置相对路径的链接。

【提示】 本题同时考察了相对路径和绝对路径的知识。

```
<a href="#"
<a name="-----
a href="mailto:-----
```

第 6 章 表 单

网页中的表单可以用来收集用户填写的信息，如某网站的问卷调查、网站的注册会员等都会用到表单。表单可以说是个容器，里面元素的类型不同，所表示的功能也不同。表单容器里面的表单元素有文本框、密码框、单选框和复选框等，通过

- 表单和表单元素；
- 文本框和密码框；
- 表单；
- 单选框和复选框；
- 按钮；
- 下拉列表和文本域。

6.1 表单概述

本节介绍表单的基本知识，让读者了解表单的概念，从整体上分析其实质。不过，表单元素在 HTML 网页中并不属于动态技术，只是一种数据提交的方法。

6.1.1 表单的用途

表单不是表格，既不用来显示数据，也不用来布局网页，甚至其外观都不是很重要。表单提供一个界面，一个入口，便于浏览者把数据提交给后台程序处理。

HTML 代码中的表单是双标签<form></form>。类似于表格、列表等 HTML 元素，表单元素由很多表单子元素组成。表单子元素用于提供不同类型的容器，记录浏览者输入的数据。

浏览者完成表单数据输入后，表单将把数据提交到后台程序页面，剩下的事情由后台程序完成。页面中可以有多个表单，但要保证一个表单只能提交一次数据。表单最常见的应用有用户登录界面、用户注册界面、用户添加留言界面、网页问卷调查等。表单的用途如图 6-1 所示。

6.1.2 表单的结构

表单结构比较简单，其子元素组合比较灵活。可以添加在表单结构内多个子元素，也可以只在表单结构内添加一个子元素。表单的结构如图 6-2 所示。

说明 表单子元素也叫表单控件。



图 6-1 表单的作用



图 6-2 表单结构组成

在表单结构中，同一种表单控件可以添加多个。表单控件有一个 name 属性，必须赋予不同的名字，用于标识不同的控件以保证唯一性，并且便于程序处理。

6.1.3 表单的基本语法

表单是一个能够包含表单元素的容器，表单元素是能够让用户在表单中输入信息的元素，可以说，表单包含表单元素。表单在动态网页中常用来收集用户的信息，保存到数据库中。表单语法由一对<form>标记对组成。

```
<!--设置表单标记对-->
<form></form>
```

在上面的表单中就可以包含表单元素，用表单元素来添加信息，不同的表单元素所表现的功能不一样，如 input 输入类型有 text、password 等，分别代表的是文本和密码，下一节会讲到表单元素。代码 6-1 是表单的基本语法。

代码 6-1 表单基本语法

```
-----文件名：表单基本语法.html-----
01 <html>
02   <head>
03     <title>表单基本语法</title>
04   </head>
05   <body>
06     <!--设置空的表单-->
07     <form></form>
08   </body>
09 </html>
```

【代码解析】在第 7 行代码中，<form></form>就是表单的语法标记。

6.1.4 表单的数据传递方式

表单有两个属性会常用到，就是 action 和 method，form 中的 action 表示表单提交后发送的 URL 地址，发送的方式用 method 表示，可选择的参数有 GET 和 POST。GET 传输量比较小，而 POST 传输值比较大，这是二者最基本的区别。除了要用表单元素外，一个表单还会用到这两个属性来传递值，值是传递给 URL 地址的后台，处理后保存到数据库中，本书只讲解前台页面的知识。代码 6-2 是表单的完整实例。

代码 6-2 表单两属性

```

-----文件名：表单两属性.html-----
01 <html>
02   <head>
03     <title>表单两属性</title>
04   </head>
05   <body>
06     <!--设置表单，并在表单中输入参数-->
07     <form action="show.aspx" method="get">
08       <!--设置文本框-->
09       <input type="text">
10       <br>
11       <!--设置密码框-->
12       <input type="password">
13       <br>
14       <!--设置按钮提交-->
15       <input type="button" value="提交">
16     </form>
17   </body>
18 </html>

```

【代码解析】第9~15行代码有两个输入框和一个按钮，当单击“提交”按钮后，表单中的信息通过表单元素收集值，然后通过代码第7行的 get 方法传到 show.aspx（本页面是.net 的动态页）页面。图 6-3 为上述代码的页面效果。

注意

action 后的值因动态语言不同而扩展名不一样。method 的值只能是 GET 和 POST 其中的一种。

在上面的实例中可以看到，在表单里用到 <input> 语法，表示在表单中输入选项，完整的语法如下。

```
<input type="#">
```

其中#代码可选项有 text、password、checkbox、radio、hidden、submit 和 reset，分别代表文本、密码、多选框、单选框、隐藏、提交按钮和重置按钮，这就是上面常讲到的表单元素。有关 <input> 的部分选项值，稍后会做出详细讲解。

注意

<form> 与以 php、aspx、jsp 等为扩展名的动态网页结合，所用的动态语言会不同，但用的表单和表单元素都是相同的。



图 6-3 表单 action 和 method

6.2 文本框和密码框

通过上一节的学习，知道了表单元素，现在开始学习表单元素 <input> 的部分类型。在网页中常常看到用户输入账号和密码之类的信息，本节介绍如何定义文本框和密码框。

6.2.1 文本框

在 <input> 中，用 type 来表示要选择的类型，文本框是用来输入文本信息的，可以输入汉字、

数字和字母等，完整的语法如下。

```
<input type="text">
```

以上代码就代表一行输入文本的输入框，供用户输入信息，在 type 后面选择 text，代表文本，从页面表现的效果正是用户在登录时的那些输入格式。代码 6-3 所表示的是文本框的完整代码。

代码 6-3 文本框表示

```
-----文件名: 文本框.html-----
01 <html>
02   <head>
03     <title>文本框</title>
04   </head>
05   <body>
06     <!--在一个表单中, 只有文本框-->
07     <form>
08       <!--设置文本框-->
09       文本框: <input type="text">
10     </form>
11   </body>
12 </html>
```

【代码解析】第 9 行代码的文本框标记为<input>，参数 type 表示框中的类型，在此输入的是 text，表示文本框。代码的浏览效果如图 6-4 所示。

注意 文本框的上限是255个字符。在<input>里面还有很多属性，本节不再详述，现在只是先让读者对<input>有个初步的认识。



图 6-4 文本框

6.2.2 密码框

密码框是为了隐藏用户密码信息的输入框，语法形式与文本框差不多，不同的是类型为 password。可能的读者会问，在密码框里输入的密码信息会不会显示出来，如果这样的话密码就不安全了。其实这个不用担心，当在密码框中输入密码时，所输入的字符等全用“*”替代，这样从页面显示上隐藏了密码信息，密码框在表单上的语法形式如下。

```
<input type="password">
```

密码框表示一个用来输入密码的框并且会用“*”替代输入的信息，与文本框一样，<input>里面也有属性可以选择，这将在以后进行讲解。代码 6-4 演示了密码框的使用。

代码 6-4 密码框表示

```
-----文件名: 密码框.html-----
01 <html>
02   <head>
03     <title>密码框</title>
04   </head>
05   <body>
06     <!--在表单中只有密码框-->
```

```

07         <form>
08             <!--设置密码框-->
09             密码框: <input type="password">
10         </form>
11     </body>
12 </html>

```

【代码解析】第9行代码中，表示密码框的关键代码是：`<input type="password">`。

密码框的设置与文本框一样，在`<input>`标记下设置`type`参数为`password`表示密码，其中密码框中输入的文本被小圆点代替，为了隐藏密码，起到安全作用，当刷新页面时，密码会被及时清除，效果如图6-5所示。



图 6-5 密码框

注意

密码框是为了保证密码安全性，当在制作相应的表单和密码框的网页时，也要考虑表单或密码的安全性。

6.3 单选框和复选框

表单的功能就是收集用户填写的信息。当出现多个选项而且只能选择一个选项时，当可以选择两个或两个以上选项时，就会分别用到单选框和复选框。通过本节的学习，读者应掌握如何在网页中要求用户填表单时是进行单选还是复选，单选在选项中只能选择一项，而复选可以选择多项，即本节介绍单选框和复选框。

6.3.1 单选框

单选框是在表单中有多个选项，且只能选一个的情况下使用。和文本框与密码框一样，都是设置在`<input>`里面的`type`属性，表示单选的是`radio`。单选框的语法代码如下所示。

```

<!--设置单选框-->
<input type="radio">

```

在表单中，对于单选框是存在多个选项的，当从多选项中选择一个选项后，为了区别这几个选项是属于哪个单选框，用属性`name`表示。如果想指定某个选项处于选中状态，用`checked`在`<input>`为`radio`类型里表示选中状态，从第一个`<input>`结束到下一个`<input>`开始前表示第一个`<input>`的选项值。代码6-5演示了单选框的用法。

代码 6-5 单选框表示

```

-----文件名：单选框.html-----
01 <html>
02     <head>
03         <title>单选框</title>
04     </head>
05     <body>
06         <!--在表单中出现表单，并有多多个单选框-->
07         <form>
08             下面是单选框: <p>
09             <!--设置3个单选框，且都设置name为book-->

```

```

10      <input type="radio" name="book"> DIV
11      <!--设置name为book，表示选项在book中选一个-->
12      <input type="radio" name=book checked > HTML
13      <input type="radio" name=book >CSS
14      </form>
15  </body>
16 </html>

```

【代码解析】第9~13行代码为三个单选框的网页代码，效果如图6-6所示。

radio在**注意** 选项的结束是下一个



图 6-6 单选表示

6.3.2 复选框

顾名思义，复选框是可以选多个选项的选框。与单选框不同的是复选框可以选取多个选项，而且也可以默认几个选项都处于选中状态，其语法与单选框相同。

```

<!--设置复选框-->
<input type="checkbox" >

```

复选框的 type 值是 checkbox。在需要选项是多选时用到复选框。与单选框一样，可以用 checked 表示选中状态。代码6-6表示复选框的用法。

代码 6-6 复选框表示

```

-----文件名：复选框.html-----
01 <html>
02   <head>
03     <title>复选框</title>
04   </head>
05   <body>
06     <!--在表单中设置复选框-->
07     <form>
08       下面是复选框：<p>
09       <!--设置复选框且name为book-->
10       <input type="checkbox" name=book>DIV知识
11       <!--设置name为book的都在一个选项集内-->
12       <input type="checkbox" name=book>HTML知识
13       <input type="checkbox" name=book checked> CSS知识
14       <input type="checkbox" name=book checked>HTML+CSS知识
15     </form>
16   </body>
17 </html>

```

【代码解析】在第10~14行代码的复选框中，多个选项组成选项集，它们的 name 都为同一名称，这时可以选择多个选项，选中的选项，在复选框中用“√”表示，可以选一个或一个以上的选项。复选框的浏览效果如图6-7所示。

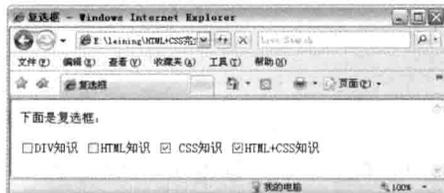


图 6-7 复选框的表示

说明 单选框与复选框的区别就是被选项是选择一个还是至少一个。

6.4 下拉列表与文本域

本节将介绍下拉列表与文本域。下拉列表是可以选择的列表；文本域是可以输入多行文本的区域，在表单中常用到。当要选择出生年月时，就会用到下拉列表。同理，选择省份时也会用到。文本域在填写多行文本时会用到。下拉列表与文本框在网页中应用非常广，下面就学习它们。

6.4.1 下拉列表

下拉列表是可以选择的列表，当在列表中选择要选的选项时使用，且只能选其中一个选项（通过设置也可以选几项）。语法如下。

```
<!--设置下拉列表-->
01 <select>
02     <!--设置下拉列表选项-->
03     <option>选项一</option>
04     <option>选项二</option>
05     <option selected>选项三</option>
06 </select>
```

下拉列表用<select>标记对表示，<option>标记对里面的文字是下拉列表的选项，其中它的selected属性表示已选中状态。代码6-7演示了下拉列表的用法。

代码 6-7 下拉列表

```
-----文件名：下拉列表.html-----
01 <html>
02     <head>
03         <title>下拉列表</title>
04     </head>
05     <body>
06         <!--设置表单，并包含下拉列表-->
07         <form>
08             <!--设置下拉表-->
09             <select>
10                 <!--设置选项-->
11                 <option>HTML</option>
12                 <option>CSS</option>
13                 <!--设置选项并用selected表示被选中-->
14                 <option selected>DIV</option>
15             </select>
```

```

16         </form>
17     </body>
18 </html>

```

【代码解析】在第9~15行代码的下拉列表中，用 select 标记对表示下拉列表，且设置选中的选项时，在<option>内加上 selected 表示选中，效果如图 6-8 所示。

技巧 <select>里的属性可以由name代表这个下拉列表的名称，size表示下拉列表可视选项的数目，multiple表示可以多选。

6.4.2 文本域

文本域在要输入多行文本、填写大量文字时用到。与前面不同的是，语法没有用到<input>，而是用<textarea>标记对包括着。代码 6-8 为默认的空文本域，表示文本域的用法。

代码 6-8 文本域

```

-----文件名：文本域.html-----
01 <html>
02     <head>
03         <title>文本域</title>
04     </head>
05     <body>
06         <!--设置文本域-->
07         <textarea>
08             <!--文本域内容-->
09         </textarea>
10     </body>
11 </html>

```

【代码解析】图 6-9 表示默认文本区域的大小，文本域中可以输入多行文本，右边的滚动条因文本多少而显示。



图 6-8 下拉列表



图 6-9 文本域

多行文本域通过在文本域上加上两个属性来表示：rows 表示显示的行数，cols 表示显示的列数。通过设置行数和列数来改变文本框的大小，以便与页面统一布局。代码 6-9 为多行文本域。

代码 6-9 多行文本域

```

-----文件名：多行文本域.html-----
01 <html>
02     <head>
03         <title>多行文本域</title>
04     </head>

```

```

05     <body>
06         <!--设置文本域和宽高-->
07         <textarea name=book rows=6 cols=60></textarea>
08     </body>
09 </html>

```

【代码解析】第7行代码表示文本域在网页中的用法。其中，文本域标记为 `textarea`，且为标记对，设置了参数 `name` 为 `book`，同时设置行数为6，列数为60。图6-10演示了多行文本域。

说明 在设置文本域大小时，将根据文本域内容的多少而设定，文本域大小与表单大小所占比例要适中（两者要有一定美观性）。



图6-10 多行文本域

6.5 按钮

按钮在表单中是最重要的，当输完表单信息时，需要用按钮提交，当输入错误信息时，这时用清除按钮更方便。

按钮是 `<input>` 中类型为 `submit` 的标单元素，表示提交，用来提交表单数据给 `action` 的网页上。其完整语法代码如下。

```
<input type="submit" value="按钮提交">
```

在把 `<input>` 里面的 `type` 设置成 `submit` 后，还要给 `value` 设置按钮名称，名称是显示在按钮上的文字和字母。

按钮在表单中起提交或重置作用，在标记 `input` 中设置为 `submit` 表示提交本表单，可以把前台填写的数据添加到网页后台，然后再保存到数据库中。当然，进一步操作超过了网页布局的范围，不属于本书的范围，但需要了解和掌握表单布局与设置。代码6-10表示按钮在表单的用法。

代码6-10 按钮

```

-----文件名：按钮.html-----
01 <html>
02     <head>
03         <title>按钮</title>
04     </head>
05     <body>
06         <!--设置表单-->
07         <form>
08             <!--设置文本框-->
09             账户：<input type="text"><p>
10             <!--设置密码框-->
11             密码：<input type="password"><p>
12             <!--设置复选框-->
13             <input type="checkbox" checked>复选框
14             <input type="checkbox">复选框二
15             <input type="checkbox">复选框三
16         <p>

```

```

17         <!--设置按钮-->
18         <input type="submit" value="按钮提交">
19     </form>
20 </body>
21 </html>

```

【代码解析】 在一个简单的表单中，第 9 行代码用文本框表示账户，第 11 行代码用密码框表示密码的输入框，第 13~15 行代码设置复选框，最后的按钮表示提交整个表单，表单内容如何提交到后台不属于本书的范围，如果读者想知道其原理，请查阅相关书籍。运行后的效果如图 6-11 所示。

注意 按钮有 submit 和 reset 之分，前者用于提交表单，后者用于重置表单，即还原成表单的默认状态。



图 6-11 按钮提交

6.6 图像域、文件域和隐藏域

表单提供了图像域以替代提交按钮。但是图像域在单击时，可返回单击位置在图片上的 x、y 坐标，从而使程序可以根据图片不同区域的坐标做出不同的处理（和 的热区链接原理相似）。其编写格式如下。

```
<input type="image" src="图片路径" />
```

浏览器如果需要上传文件到服务器，可以采用文件域。其编写格式如下。

```
<input type="file" />
```

很多时候传给程序的数据不一定需要浏览者填写，这时可以用隐藏域传数据。其编写格式如下。

```
<input type="hidden" value="数据" />
```

创建 img 目录，放入一张 jpg 格式的图片，并命名为 fifa.jpg，然后编写代码如 6-11 所示。

代码 6-11 图像域

```

01 <html>
02 <head>
03     <title>表单控件6</title>
04 </head>
05 <body>
06 <center>
07     <form name="form_set" method="post" action="form_rec.asp">
08         图像域（单击图片类似于单击提交按钮）：<br />
09         <input type="image" src="img/fifa.jpg" /><hr />
10         文件域（单击“浏览”按钮可以浏览本机文件）：<br />
11         <input type="file" /><hr />
12         隐藏域（在页面中不可见，但是可以装载须传输数据）：<br />
13         <input type="hidden" name="txt" value="我是隐藏域中的值" />
14     </form>
15 </center>
16 </body>
17 </html>

```

【代码解析】读者单击图像域的图片，表单将提交到 form_rec.asp 页面程序处理，页面将显示隐藏域的值。本例的效果如图 6-12 所示。

6.7 表单实例

通过学习本章，知道了表单的用法以及表单元素的表示。本节用实例介绍了注册页面的表单制作，其中包括表单、输入文本框、密码、单选框等。希望通过学习，读者能举一反三，从而对表单有更深层次的理解。

本实例从 HTML 标记入手，只讲解网页前台部分（因为表单提交后，通过后台保存到数据库中），以学习这些标记为主。以下是表单实例手把手的步骤。



图 6-12 图像域、文件域和隐藏域

6.7.1 插入表单

在新建网页中插入表单，表单的标记为 `<form></form>`，放在标记对 `<body></body>` 中，表示在网页中存在表单。当然，在一个网页中可以存在多个表单，即有多少表单就有多少个 `<form></form>`，其中表单中包括单选框、多选框等。最后用提交按钮把信息提交出去，即表单相当于一个容器。代码 6-12 表示表单容器。

代码 6-12 表单实例

```
----- 文件名: 表单实例手把手.html -----
01  <html>
02  <head>
03  <title>表单实例手把手</title>
04  </head>
05  <!--设置表单-->
06  <form>
07  <!--设置表单中的段落-->
08  <p>用户注册页面表单实例</p>
09  </form>
10 <body>
11 </body>
12 </html>
```

【代码解析】在第 6~9 行代码中创建了表单容器，并在表单里输入一行文字，其中存在段落标记对 `<p></p>`，效果如图 6-13 所示。

6.7.2 添加文本框和密码框

在表单中显示用户名输入，并有两次密码输入。注册页面都有用户名和密码，而且密码在创立之初，要输入两次。显然，第二次输入用来确定密码是否一致。代码 6-13 表示文本框用来输入用户名，密码框用来输入密码。



图 6-13 表单容器

代码 6-13 用户名和密码

```

-----文件名：表单实例手把手.html-----
01 <html>
02 <head>
03 <title>表单实例手把手</title>
04 </head>
05 <!--设置表单-->
06 <form>
07 <p>用户注册页面表单实例<p>
08 <!--设置表单中的文本框-->
09 用户名: <input type="text">
10 <br>
11 <!--设置表单密码框-->
12 密码: <input type="password">
13 <br>
14 <!--设置表单中的确认密码框-->
15 确认密码: <input type="password">
16 </form>
17 <body>
18 </body>
19 </html>

```

【代码解析】以上代码中,第9~15行中

6.7.3 添加单选框和复选框

添加部分选项实现单选框和复选框。注册页面时,会提供部分选项要用户选择,有的只能选一个,如年龄层次的选项;有的可以选多个,如个人爱好有很多可以选择。代码6-14是单选项和复选项的用法。



图 6-14 用户名和密码表示

代码 6-14 单选项和复选项

```

-----文件名：表单实例手把手.html-----
01 <html>
02 <head>
03 <title>表单实例手把手</title>
04 </head>
05 <!--设置表单-->
06 <form>
07 <p>用户注册页面表单实例<p>
08 用户名: <input type="text">
09 <br>
10 密码: <input type="password">
11 <br>
12 确认密码: <input type="password">
13 <!--设置表单中的单选框-->
14 <p>您的年龄层是? <br>
15 <input type="radio" name="reg">16周岁以下

```

```

16     <input type="radio" name=reg>17~22周岁
17     <input type="radio" name=reg>23~28周岁
18     <input type="radio" name=reg>29周岁以上
19     </p>
20     <!--设置表单中的复选框-->
21     <p>您个人的爱好有: <br>
22     <input type="checkbox" name=reg>交友
23     <input type="checkbox" name=reg>上网
24     <input type="checkbox" name=reg>看书
25     <input type="checkbox" name=reg>篮球
26     <input type="checkbox" name=reg>游戏
27     <input type="checkbox" name=reg>其他
28     </p>
29     </form>
30 </body>
31 </body>
32 </html>

```

【代码解析】在文本框和密码框的基础上，第15~27行代码用于在表单中加入了单选框和复选框，效果如图6-15所示。

6.7.4 添加下拉列表和文本域

下拉列表是下拉的列表，用户在下拉的列表中选择里面的选项；文本域就是可以输入多行的文本区域，即可以实现回车换行的区域。代码6-15表示下拉列表与文本域的用法。

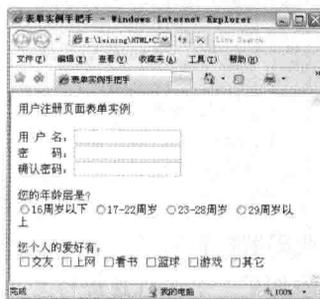


图 6-15 单选项和复选项表示

代码 6-15 下拉列表与文本域表示

```

-----文件名: 表单实例手把手.html-----
01 <html>
02 <head>
03 <title>表单实例手把手</title>
04 </head>
05     <form>
06     <p>用户注册页面表单实例<p>
07     用户名: <input type="text">
08     <br>
09     密码: <input type="password">
10     <br>
11     确认密码: <input type="password">
12     <!--设置表单的单选框-->
13     <p>您的年龄层是? <br>
14     <input type="radio" name=reg>16周岁以下
15     <input type="radio" name=reg>17~22周岁
16     <input type="radio" name=reg>23~28周岁
17     <input type="radio" name=reg>29周岁以上
18     </p>
19     <p>您个人的爱好有: <br>
20     <!--设置表单的复选框-->

```

```

21     <input type="checkbox" name=reg>交友
22     <input type="checkbox" name=reg>上网
23     <input type="checkbox" name=reg>看书
24     <input type="checkbox" name=reg>篮球
25     <input type="checkbox" name=reg>游戏
26     <input type="checkbox" name=reg>其他
27 </p>
28 <p>您来自哪里? <br>
29 <!--设置表单下拉框-->
30 <select name=come>
31     <option>北京</option>
32     <option selected>广东</option>
33     <option>上海</option>
34     <option>其他城市</option>
35 </select>
36 </p>
37 <p>自我介绍: <br>
38 <!--设置表单文本域-->
39     <textarea cols=60 rows=5>
40     </textarea>
41 </p>
42 </form>
43 <body>
44 </body>
45 </html>

```

【代码解析】第 30 行代码把下拉列表命名为 come，且选项“广东”为默认选中状态；第 39 行代码添加了一个 5 行且每行为 60 字大小的文本可视区域，效果如图 6-16 所示。

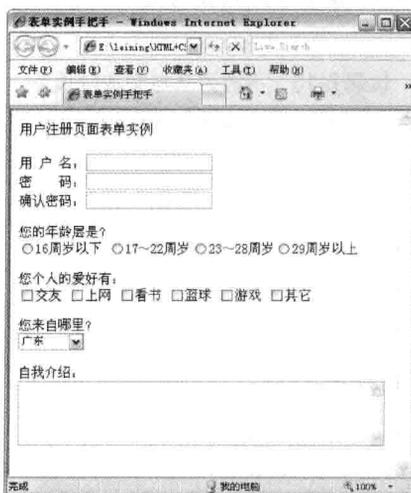


图 6-16 下拉列表和文本域表示

6.7.5 在表单中添加提交按钮

一个表单必须提交后才能保存数据到数据库中，所以提交按钮是不能少的。代码 6-16 用于添加一个提交按钮和重置按钮。

代码 6-16 提交和重置按钮

```

-----文件名：表单实例手把手.html-----
01 <html>
02 <head>
03 <title>表单实例手把手</title>
04 </head>
05 <!--设置表单-->
06 <form>
07 <p>用户注册页面表单实例<p>
08 用户名: <input type="text">
09 <br>
10 密码: <input type="password">
11 <br>
12 确认密码: <input type="password">
13 <p>您的年龄层是? <br>
14 <input type="radio" name=reg>16周岁以下
15 <input type="radio" name=reg>17~22周岁
16 <input type="radio" name=reg>23~28周岁
17 <input type="radio" name=reg>29周岁以上
18 </p>
19 <p>您个人的爱好有: <br>
20 <input type="checkbox" name=reg>交友
21 <input type="checkbox" name=reg>上网
22 <input type="checkbox" name=reg>看书
23 <input type="checkbox" name=reg>篮球
24 <input type="checkbox" name=reg>游戏
25 <input type="checkbox" name=reg>其他
26 </p>
27 <p>您来自哪里? <br>
28 <select name=come>
29 <option>北京</option>
30 <option selected>广东</option>
31 <option>上海</option>
32 <option>其他城市</option>
33 </select>
34 </p>
35 <p>自我介绍: <br>
36 <textarea cols=60 rows=5>
37 </textarea>
38 </p>
39 <!--设置表单提交按钮-->
40 <p><input type="submit" value="注册"> <input type="reset" value="重置"></p>
41 </form>
42 </body>
43 </body>
44 </html>

```

【代码解析】在第 40 行代码中，按钮的值用<input>中的 value 属性表示，此时整个注册网页就完成了。本实例只讲解表单和表单元素的用法，所以对显示效果没做处理，效果并不是最终网页效果，以后学到 CSS 部分就可以对表单进行布局控制，制作出美观的网页。本实例最终的网页效果如图 6-17 所示。

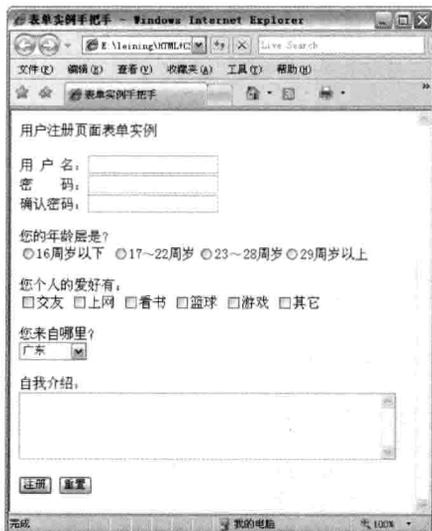


图 6-17 添加按钮后整个注册页效果

注意

一张表单中的<form>必须加action和method才可以提交，本实例并不涉及动态网页和后台数据处理，所以没给出网页中的action和method的方法。

6.8 小结

本章主要介绍了网页中的表单，并详细讲解了前台设计的表单元素。其中常用的表单元素有文本框和密码框、单选框和复选框、下拉列表和文本域，还有按钮。表单相当于容器，包含表单元素，通过不同的表单元素体现的不同功能，使网页能够收集不同信息。最后通过实例再次巩固了表单和表单元素的使用法。

6.9 习题

一、填空题

1. 写出下列 input 类型对应的中文名称：text、password、checkbox、radio、hidden、submit 和 reset，分别代表：_____、_____、_____、_____、_____、_____、_____。

2. 表单的数据传递方式有两种：_____、_____。

3. 多行文本域是在文本域上加上两个属性来表示：_____表示显示的行数；_____表示显示的列数。

二、上机实践

1. 按以下版式做一个注册网页。

用户名：

密 码：

确认密码:

性别: 男 女-----单选框实现

备注:

【提示】

```
input type = "text"-----文本框
input type = "password"-----密码框
input type = "radio"-----单选框
input type = "textarea"-----文本域
```

2. 按以下版式制作一个网页调查问卷。

用户名:

性别: -----下拉列表实现

喜欢哪些城市:

北京 上海 杭州 南京 西安

理由:

【提示】

```
input type = "text" -----文本框
select -----下拉列表
checkbox-----复选框
textarea-----文本域
```

第 7 章 表 格

表格的用途很广，可以用来显示表格型数据、图片等，也可以用来布局。在前面讲述的布局中常会用到表格，现在的网页更趋向 W3C 标准方向发展，用 DIV+CSS 布局可取代表格。但表格不仅可以用来布局，还有其他功能，所以学习表格很重要。虽然表格布局不常用，但表格的其他用法不能用 DIV+CSS 取代。学完本章后，读者会掌握以下内容。

- 标记 table 及 tr、td 用法；
- 跨多行、多列的表格；
- 表格内文字对齐；
- 表格在网页中对齐；
- 表格间隔。

7.1 页面中的表格

表格在日常生活中随处可见，如工资表、考勤表、各种报表等。本节重点帮助读者了解网页中的表格形式。

7.1.1 表格的外观

表格看上去虽然由一个一个小格子组成，但是，在 HTML 中制作表格，远远不是看上去的那么直接。表格涉及的属性很多，因为人们在表述表格的时候，不是说“某某表格左上角的那个格子”，而是通过描述某一行和某一列来定位某个单元格的位置，这里就已经描述了 3 个属性了。一个普通的表格在页面中如图 7-1 所示。

这是一个 3×3 的表格，横排为行，竖排为列，每一个格子称为单元格。在 HTML 页面中不仅可以定义表格行列的数量，还可以设定表格的边框的样式、单元格的长宽高以及单元格彼此之间的距离。所以在 HTML 页面中，编辑表格是一项不困难却很烦琐的工程。



图 7-1 一个普通的表格

7.1.2 表格的基本语法

表格在网页中可以表现出 Word 中的表格效果，即在 Word 中要表现的表格效果可以在网页中显示，但需要使用 HTML 中的表格标记。表格标记不仅用于表现表格中的效果，还可以用表格来给网页布局，布局中的表格是不需要表格中的边框的，故需要对表格进行设置。

表格是 HTML 常用的标记，表示在网页中插入一张表格。表格用 table 标记对表示，其语法形式如下。

```
<!--设置表格标记 -->
<table></table>
```

这只是表示一张表格而已，但表格常常是有行和列的，那么，如何在表格内表示行和列呢？这又要另外两个标记：<tr>标记对表示表行，每出现一个<tr></tr>代表表格的一行；<th>标记对表示表头，表头是在表格上显示下面列的；<td>标记对表示表元，表元就是在表格中显示的每一方格。代码 7-1 说明表格的基本语法。

代码 7-1 表格的基本语法

```
-----文件名：表格的基本语法.html-----
01 <html>
02 <head>
03 <title>表格的基本语法</title>
04 </head>
05 <body>
06 <!--设置表格和边框-->
07 <table border="1">
08 <!--设置表格中的表头-->
09 <tr>
10 <!--设置表格中的表项-->
11 <th>表头一</th>
12 <th>表头二</th>
13 </tr>
14 <tr>
15 <td>表元二</td>
16 <td>表元二</td>
17 </tr>
18 </table>
19 </body>
20 </html>
```

【代码解析】第 7~18 行代码中的<table>标记对中包括两个行标识符：第 9 行和第 14 行代码中的<tr>标记对表示两行。在第一个行标识符中包括了表头符，在第二个行标识符中包括了表元标识符，为了显示出效果，在表格属性中加入了 border 边框，效果如图 7-2 所示。



图 7-2 表格的基本语法

注意

在<table>中的<tr>、<th>、<td>是常用的，<th>可以省略，三者都在<table></table>中，不能交叉。

7.2 跨多行、多列表元

当表格中的单元格有相同的内容或者需要设置相同的内容时，需要设置单元格合并，合并可以为不同行的单元格，也可以为不同列的单元格。在网页中可以设置不同的单元格合并，所表现的效果与 Word 中的表格一样。

在表格中有时会合并单元格,所以对表格要进行跨行或跨列处理,使多个表元合成一个单元格。跨表元分跨多行和跨多列,包括跨表头 th 和表元 td,跨行用 rowspan 表示,同理,跨列用 colspan 表示,下面分别对它们进行详细介绍。

7.2.1 跨多行表元

跨多行表元应用在合并多行单元格时,跨多行表元语法是在 th 和 td 上加 rowspan 表示, rowspan 后面的值是数字,数字表示跨多少行表元,如表示跨 3 行表元的代码如下。

```
<!--设置表格中的跨多行-->
<th rowspan=3></th>
```

或者如下表示。

```
<!--设置表格中td的跨多行-->
<td rowspan=3></td>
```

这就表示跨越 3 行显示。为了表达出效果是跨行的,可从多行多列表格中实践跨多行表元用法。代码 7-2 表示的是跨多行表元用法。

代码 7-2 跨多行表元

```
-----文件名: 跨多行表元.html-----
01 <html>
02 <head>
03 <title>跨多行表元</title>
04 </head>
05 <body >
06 <!--设置表格-->
07 <table border="1" width="300">
08 <!--设置表格的行-->
09 <tr>
10 <!--设置表格中的表项和跨多行-->
11 <td rowspan=3>1</td><td>2</td><td>3</td><td>4</td>
12 </tr>
13 <!--以下都是设置表格行和表项-->
14 <tr>
15 <td>2</td><td>3</td><td>4</td>
16 </tr>
17 <tr>
18 <td>2</td><td>3</td><td>4</td>
19 </tr>
20 <tr>
21 <td>1</td><td>2</td><td>3</td><td>4</td>
22 </tr>
23 </table>
24 </body>
25 </html>
```

【代码解析】代码 7-2 表示一个 4 行 4 列的表格,第 11 行代码是第一列,设置 rowspan 为 3,表示在本列跨 3 行,其他的表元往后推,效果如图 7-3 所示。

注意

当出现 rowspan 时,所包含的列数并不需要与 rowspan 所在列数一致,如果一致,会导致列往右移。

7.2.2 跨多列表元

与跨多行一样，跨多列也放在 和 中，表示在一行中跨多列表元。语法用colspan表示，完整语法如下。 | |

```
<!--设置表格中跨多列-->
<th colspan=3></th>
```

或

```
<td colspan=3></td>
```

表示跨多列，相应的有多少列占用多少个单元格。代码7-3表示跨多列表元。

代码 7-3 跨多列表元

```
-----文件名：跨多列表元.html-----
01 <html>
02 <head>
03 <title>跨多列表元</title>
04 </head>
05 <body >
06 <!--设置表格-->
07 <table border="1" width="300">
08 <!--设置表格跨多列-->
09 <tr>
10 <!--设置表格中表项-->
11 <td colspan=3>1</td><td>2</td>
12 </tr>
13 <tr>
14 <td>1</td><td>2</td><td>3</td><td>4</td>
15 </tr>
16 <tr>
17 <td>1</td><td>2</td><td>3</td><td>4</td>
18 </tr>
19 <tr>
20 <td>1</td><td>2</td><td>3</td><td>4</td>
21 </tr>
22 </table>
23 </body>
24 </html>
```

【代码解析】代码7-3中的第11行表示在表格中的第一行跨多列，colspan的值为3，所以跨3列，效果如图7-4所示。



图 7-3 跨多行表元



图 7-4 跨多列表元

注意 colspan与rowspan要分清楚，表示跨多行和跨多列在表格中的单元格布局不一样。

7.3 设置表格大小

表格可以设置宽度和高度，因为默认无边框（边框为0），边框也可以设置宽度。本节首先介绍边框属性，然后介绍表格的高度和宽度。

7.3.1 设置表格边框

边框是用 border 表示嵌入在 table 标记里面的属性，表示表格边框的宽度。在学习表格大小前先讲解表格的边框，表格边框在表格标记里用 border 表示，语法形式如下。

```
<!--设置表格的边框-->
<table border=1></table>
```

标记 border 后面的值是宽度值，表示对象的边框宽度，数值越大，宽度就越大。代码 7-4 用实例演示了表格边框的宽度。

代码 7-4 表格边框的宽度

```
-----文件名：表格边框宽度.html-----
01 <html>
02 <head>
03     <title>表格边框宽度</title>
04 </head>
05 <body>
06 <table border=5>
07     <tr>
08         <td>表格宽度</td>
09         <td>表格宽度二</td>
10     </tr>
11     <tr>
12         <td>第二行表元</td>
13         <td>第二行表元二</td>
14     </tr>
15 </table>
16 </body>
17 </html>
```

【代码解析】第 6~15 行代码表示一个两行两列且边框为 5 的表格，在显示效果时，每个表项的边框还是默认的 1，而表格外边框的宽度变成 5（第 6 行代码），设置的数值越大，表格边框也越宽，效果如图 7-5 所示。



图 7-5 设置表格边框的宽度

7.3.2 设置表格高度和宽度

除了对表格边框设置大小外，还可以对整个表格设置大小。

与其他参数一样，在标记 table 中，设置宽度用 width 表示，高度用 height 表示。在网页中的表格按设置的值来显示，如下代码表示设置表格的高和宽，即在表格标记<table>中，设置 width 为 300，

设置 height 为 200，默认单位值为像素，如代码 7-5 所示。

代码 7-5 设置表格大小

```
-----文件名：设置表格大小.html-----
01 <html>
02 <head>
03 <title>设置表格大小</title>
04 </head>
05 <body>
06 <table border=1 width=300 height=200>
07 <tr>
08 <td>表格宽度</td>
09 <td>表格宽度二</td>
10 </tr>
11 <tr>
12 <td>第二行表元</td>
13 <td>第二行表元二</td>
14 </tr>
15 </table>
16 </body>
17 </html>
```

【代码解析】第 6 行代码设置表格的宽度为 300，高度为 200，且其单位默认的是像素，故所表示的是一个宽为 300 像素，高为 200 像素的表格，该表格又存在两行两列，且两行两列的大小刚好是表格大小，效果如图 7-6 所示。

7.4 表格内文字对齐

在表格内，文字对齐有在 tr、th 和 td 中对齐，在 tr 中设置对齐方式后，tr 内的 th 和 td 都按设置的要求对齐，同理在多行中可以设置不同的 tr 对齐方式，th 和 td 都可以设置其里面的文字或图片的对齐。

对齐语法用 align 表示，后面接的值是 left、center 和 right，分别代表向左、居中和向右对齐，语法形式如下。

```
<!--设置表格对齐-->
<tr align=#></tr>
<th align=#></th>
<td valign=#></td>
```

以上代码代表在 tr、th 和 td 中对齐，# 表示 align 后面接的值，即 left、center 和 right 中的一个。代码 7-6 表示在表格内文字对齐。

代码 7-6 表格内文字对齐

```
-----文件名：表格内文字对齐.html-----
01 <html>
02 <head>
03 <title>对齐</title>
```

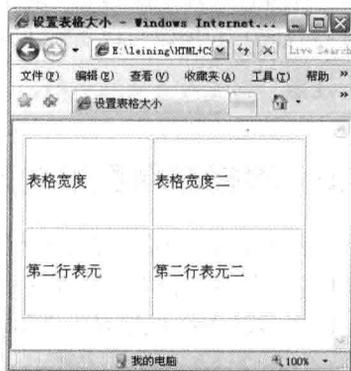


图 7-6 设置表格大小

```

04 </head>
05 <body >
06 <!--设置表格边框和宽度-->
07 <table border="1" width="300">
08   <!--设置表格的行居中对齐-->
09   <tr align="center">
10     <td>1</td><td>2</td><td>3</td><td>4</td>
11   </tr>
12   <!--设置表格行向右对齐-->
13   <tr align="right">
14     <td>1</td><td>2</td><td>3</td><td>4</td>
15   </tr>
16   <!--设置表格对齐为默认值，即向左-->
17   <tr>
18     <td>1</td><td>2</td><td>3</td><td>4</td>
19   </tr>
20 </table>
21 </body>
22 </html>

```

【代码解析】表格中第一行用居中对齐（第9行代码），第二行向右对齐（第13行代码），第三行为默认向左对齐（第18行代码），所以可以省略不写，效果如图7-7所示。

注意 向左对齐是默认的，写不写align=left都一样。



图7-7 表格内文字对齐

7.5 表格在网页中对齐

表格是有大小的，有时在网页中需要调节表格位置，如放在网页中间或网页右边，这时需要对表格进行设置。设置表格在网页中的对齐与上一节介绍的一样，都会用到 align 语法，但不同的是在<table></table>标记对中表格在网页中的对齐形式。

```
<table align=#></table>
```

#的表示与上一节也相同，就是 left、center 和 right。代码7-7表示表格在网页的右边对齐显示。

代码7-7 表格在网页右边

```

-----文件名：表格在网页右边.html-----
01 <html>
02 <head>
03 <title>表格在网页中的对齐</title>
04 </head>
05 <body >
06 <!--设置表格在网页右边对齐-->
07 <table align=right border="1" width="300">
08   <!--设置表格的行-->
09   <tr>
10     <!--设置表格的列，即表项-->
11     <td>1</td><td>2</td><td>3</td><td>4</td>
12   </tr>
13   <tr>
14     <td>1</td><td>2</td><td>3</td><td>4</td>

```

```

15     </tr>
16     <tr>
17         <td>1</td><td>2</td><td>3</td><td>4</td>
18     </tr>
19 </table>
20 </body>
21 </html>

```

【代码解析】上述代码表示表格在网页中向右对齐(第7行代码),同理可以设置 align 为 center (居中),设置 align 为 left (向左)对齐。需要注意的是,默认都是向左对齐,而本实例演示表格在网页中向右对齐,效果如图 7-8 所示。

在table中的对齐主要是对表格在网页的位置对齐,而在tr、th、td中的对齐是相对文字的对齐方式。

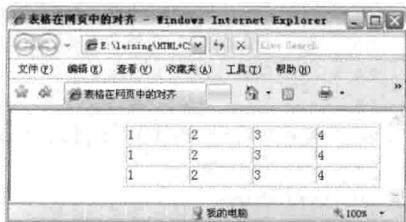


图 7-8 表格在网页中向右对齐

7.6 表格实例

表格标记 (table) 在网页中用来表示一张表格,其内可以用来布局,也可以用来显示表格中的内容。插入表格的方法为在标记对<body></body>中放入表格标记对<table></table>,其中表格标记对中还需要用一些标记表示表格中的内容,如用 th 表示表头,td 表示表元等,这些标记都包含在表格标记中。

本实例用表格做一张课程表,通过表格知识来实践网页中的表格表示,首先确定表格中的行数和列数,然后确定有没有必要做表头,步骤如下。

7.6.1 表格中嵌入表格

确定表格行数和列数并写上表头。课程在一星期中只有 5 天,因此确定有 5 列,如果按一天上 4 节课,分上午和下午,则确定有 4 行,如代码 7-8 所示。

代码 7-8 表格实例手把手

```

-----文件名: 字体颜色.html-----
01 <html>
02 <head>
03 <title>表格实例手把手</title>
04 </head>
05 <body>
06 <!--设置表格-->
07 <table border="1">
08     <!--设置表格的行-->
09     <tr>
10         <!--设置表格的列-->
11         <th>星期一</th><th>星期二</th><th>星期三</th><th>星期四</th><th>星期五12
10 </th>
12     <tr>
13     <tr>

```

```

14         <td>HTML基础</td><td>计算机图形学</td><td>C语言程序设计</td><td>计算机
15  导论</td><td>计算机图形学</td>
16     </tr>
17 </tr>
18         <td>C语言程序设计</td><td>计算机导论</td><td>C语言程序设计</td><td>法律
19  基础</td><td>HTML基础</td>
20     </tr>
21 </tr>
22         <td>计算机图形学</td><td>C语言程序设计</td><td>计算机导论</td><td>法律基
23  础</td><td>计算机导论</td>
24     </tr>
25         <td>C语言程序设计</td><td>计算机图形学</td><td>HTML基础</td><td>计算机
26  图形学</td><td>HTML基础</td>
27     </tr>
28 </table>
29 </body>
30 </html>

```

【代码解析】第7~28行代码设置了5行文本，且每行有5列，默认向左对齐，其中第一行为表头，用th表示，效果如图7-9所示。

7.6.2 设置跨行和跨列

上一节设置了表格并添加了表格中的课程，在多行和多列的表格中，需要对相同的课程进行合并，即所谓的对部分重复的课程跨行或跨列，即对这些表元进行跨行和跨列设置，如代码7-9所示。



图 7-9 表格的行和列

代码 7-9 表格实例

```

-----文件名：实例手把手.html-----
01 <html>
02 <head>
03 <title>表格实例手把手</title>
04 </head>
05 <body>
06 <table border="1">
07     <tr>
08         <th>星期一</th><th>星期二</th><th>星期三</th><th>星期四</th><th>星期五</th>
09     </tr>
10 </tr>
11         <td>HTML基础</td><td>计算机图形学</td><td rowspan="2">C语言程序设计
12 </td><td>计算机导论</td><td>计算机图形学</td>
13     </tr>
14 </tr>
15         <td>C语言程序设计</td><td>计算机导论</td><td rowspan="2">法律基础
16 </td></tr>
17 </tr>
18         <td>计算机图形学</td><td>C语言程序设计</td><td>计算机导论</td><td>计算机
19  导论</td><td>HTML基础</td>

```

```

20      <tr>
21          <td>C语言程序设计</td><td>计算机图形学</td><td>HTML基础</td><td>计算机图
22 形学</td><td>HTML基础</td>
23      </tr>
24  </table>
25  </body>
</html>

```

【代码解析】在第 11、15 行代码设置了单元格中的跨行，即将第 3 列的第 2、3 行合并成一行，同时，将第 3 行的第 4、5 列并成一列，课程表中相同的内容在表元中通过跨多行和跨多列表现出来，效果如图 7-10 所示。



图 7-10 跨多行和跨多列

综上所述，现实生活中对表格的应用远远多于本实例，有的在布局中用到，还有在网页中插入表格等。对 table 的理解从 tr、th 和 td 这三个标记上入手是比较快的，通过实例分析更容易掌握。

7.7 小结

本章讲解了表格的应用，首先从表格基本知识入手，对语法进行介绍，然后通过对跨多行跨多列的学习，理解一张表格中为什么很多单元格都被合并。在表格中对齐文字，可以设置表格内的文字往哪边对齐，同时对表格在网页中对齐也做了介绍，这点在布局中非常重要。最后通过实例全面地应用表格的用法。

7.8 习题

一、填空题

1. 表格由一个一个的小格子组成，这个小格子称为_____。
2. 制作完整的表格需要至少由_____、_____和_____ 3 个标签组成。
3. 表格的高度属性和宽度属性分别是_____、_____。

二、上机实践

1. 按下面的版式制作一个学籍表。

姓名	性别	专业	学制	班主任
HTML张三	男	计算机科学与技术	4	维维
HTML张三	男	计算机科学与技术	4	

【提示】

tr-----行
th-----表元
td-----表头
rowspan-----行合并

2. 按下面的版式制作一个成绩表。

姓名	学号	语文	数学	总分
张三	2011001	90	95	185
张三	2011002	95	80	175

【提示】

tr-----行
th-----表元
td-----表头
align-----样式标记

第 8 章 框 架

网页中框架 (frame) 的主要作用是将浏览器窗口分割成几个相对独立的小窗口, 浏览器可以将不同网页文件同时传送到这几个小窗口, 这样就可以同时浏览不同网页的内容。使用框架, 可以在一个浏览器中显示多张 HTML 文档, 这种 HTML 文档称为框架页。各张框架页都可以不同, 所以各页面之间是独立的。通过本章的学习, 可以了解到以下框架知识。

- frameset 标记用法;
- frame 标记用法;
- 框架分栏;
- 框架重要属性;
- 框架链接;
- 框架实例。

8.1 用框架灵活地组织网页

本节介绍框架页面的作用以及总体结构, 内容难度不大, 需要读者多进行实践操作。

8.1.1 框架在网站中的作用

框架页面可用于向浏览器窗口中装载多个 HTML 网页文件, 每个 frame 里的网页相互独立。如果网站的每个网页某部分是不变的, 那就很适合使用框架页面, 如图 8-1 所示。

如果做成框架页面, 图 8-1 所示的导航区域是一个单独的 HTML 网页文件, 下面的内容区域也是一个单独的 HTML 网页文件。

8.1.2 框架的结构分析

框架页面上有多个帧 (frame), 每个 HTML 文件占据一个帧, 而多个帧可以同时显示在同一个浏览器窗口中, 它们组成了一个最大的帧, 也即是一个包含多个 HTML 文档的 HTML 文件 (称为框架集页面)。帧通常的使用方法是在一个帧中导航链接, 然后将需要变化的内容 HTML 文件显示在另一个帧中。常用的框架页面结构如图 8-2 所示。

图 8-2 中一共有 4 个 HTML 网页文件, 当然, 根据需要的不同, 框架集页面可以通过帧链接更多的页面。

8.1.3 框架的基本语法

框架就是在一个浏览器中显示多个 HTML 页面, 使用框架后, 在一张网页中可以用来放置多

个不同的网页。在新闻类的网站可以看到有的广告专用一张网页来显示，每次更新广告的时候，只需要对那张网页做修改。本节主要介绍框架的基本语法，如 frameset、frame 和框架的部分属性等。



图 8-1 导航区

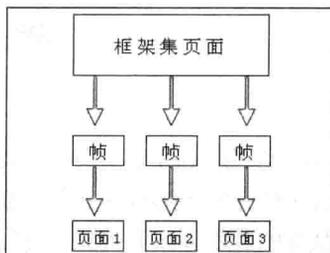


图 8-2 框架页面结构

如果一个网页的左边导航菜单是固定的，而页面中间的信息可以上下移动，这就可以认为该网页是一个框架型网页。此外，一些框架型站点的模板在其页面上方放置了公司的 LOGO 或图片，不过它们也是位置固定的。而页面的其他部分则可以上下左右移动。有的框架型站点模板还会在其固定区域中放入链接或导航按钮。

框架的基本语法是由<frameset></frameset>标记对表示，标记对放在<html></html>标记对下的<head></head>后面，即与<head></head>标记对并列。框架的实现是用<frameset></frameset>来设置，其内又包含<frame></frame>。用<frame></frame>标记对来表示在框架内放入什么文件。代码 8-1 表示框架的基本语法。

代码 8-1 框架基本语法

```
-----文件名：框架基本语法.html-----
01 <html>
02 <head>
03 <title>框架基本语法</title>
04 </head>
05 <!--设置框架-->
06 <frameset cols="25%,50%,25%">
07 <frame src="http://www.hao123.com/">
08 <frame src="http://www.sina.com.cn/">
09 <frame src="http://www.163.com/">
10 </frameset>
11 </html>
```

【代码解析】第 6~10 行代码在 html 中插入了框架，框架包括 3 个网页，分别表示了网址之家、新浪和网易的首页，在<frameset>中的 cols 参数代表列，在代码 8-1 中，表示在框架集中有 3 个框架且链接不同的页面。效果如图 8-3 所示。

这样，三张网页就插入到了这个网页的框架中，框架中的每张网页都是独立的。设置<frameset></frameset>时，必须有行和列的表示，不能空格。



图 8-3 框架基本语法

`<frameset></frameset>`在`<html></html>`标记对内,不能与`<body></body>`并列,所以在网页中先去掉`<body></body>`,再写`<frameset></frameset>`标记对,以免不知道在哪儿出现错误。`<frame>`是在`<frameset></frameset>`标记对中的,且是单标记。

8.2 框架分栏

在网页中,有时要插入多张网页到框架中,那么对网页的分栏也很重要。如何操作才能使网页看起来更自然呢?

框架分栏有垂直分栏和水平分栏两种。框架分栏是在网页中插入多个框架的结果,多个框架放在同一个网页中,必然会出现分栏状态,如何使框架分栏在网页中达到更好的视觉效果?本节介绍框架分栏的两种分栏样式。

8.2.1 框架垂直分栏

垂直分栏是对框架里的网页文档垂直排列,每张网页文档占用一列显示。垂直分栏是在`<frameset>`中用`cols`表示,后面的数字表示列宽,语法形式如下。

```
<frameset cols="50,50">
<frame src="http://www.163.com">
<frame src="http://www.sina.com.cn">
</frameset>
```

垂直分栏在`<frameset>`中包含两张网页文档,每张网页占50%,第二张网页用`<frame>`来引进网页文档路径。代码8-2表示垂直分栏。

代码 8-2 框架垂直分栏

-----文件名: 框架垂直分栏.html-----

```
01 <html>
02 <head>
03 <title>框架垂直分栏</title>
04 </head>
05 <frameset cols="50,50">
06 <frame src="http://www.163.com">
07 <frame src="http://www.sina.com.cn">
08 </frameset>
09 </html>
```

【代码解析】第5行代码表示在框架中包含两个垂直的网页文档,并设置了`cols`参数为50和50,表示两个网页各占一半显示,效果如图8-4所示。

框架的垂直分栏把整个网页分成多列,如`cols`中有两个数字,表示两个列。如果想用三列显示网页文档那要怎么样呢?可以在后面加逗号再加一个数字就可以了。注意,三个数字加起来不能超过100%,请读者想想为什么。



图 8-4 框架垂直分栏

注意 路径可以是网址，也可以是同目录下的网页文档。

8.2.2 框架水平分栏

框架可以垂直分栏，同样也可以水平分栏。水平分栏就是把网页文档分成多行显示。水平分栏的语法用 rows 表示。与垂直分栏一样，水平分栏也放在<frameset>上一个框架不能同时出现水平和垂直分栏，即不能同时出现 cols 和 rows。代码 8-3 表示框架水平分栏。

代码 8-3 框架水平分栏

```
-----文件名：框架水平分栏.html-----
01 <html>
02 <head>
03 <title>框架水平分栏</title>
04 </head>
05 <frameset rows="40,60">
06     <frame src="http://www.sina.com.cn">
07     <frame src="http://www.163.com">
08 </frameset>
09 </html>
```

【代码解析】第 5 行代码表示两个网页文档，一个占用 40%，另一个占用 60%，分上下两行显示，上面显示的是新浪，而下面显示的是网易。效果如图 8-5 所示。

在使用框架的时候，根据网页美观性来选择框架是水平分栏还是垂直分栏。再通过其他属性的运用，可以做出很漂亮的页面样式，稍后介绍框架常用的属性。

注意

当框架里网页文档超过两个时，必须出现 cols 或 rows，如果只有一张网页文档，可以省掉。在<frameset></frameset>中也可以再嵌入<frameset></frameset>标记对。



图 8-5 框架的水平分栏

8.3 框架的常用属性

只使用框架基本语法显示网页文档是不够的，如果对特殊的环境处理不当，会影响效果，而且也不好控制框架在整个网页上的样式。框架的常用属性有不可调节框架大小、不支持框架等，下面分别对其进行介绍。

8.3.1 设置不可调节框架大小

上面所述的框架都是可以调节大小的，有时网页制作需将框架设置成不能调节大小，这时在该网页文档中用 noresize="noresize"表示，在网页效果中就不需要用户去调节大小。代码 8-4 表示用 noresize 设置不可调节框架大小。

代码 8-4 用 noresize 表示不能调节框架大小

```

-----文件名: 用noresize表示不能调节框架大小.html-----
01 <html>
02 <head>
03 <title>不能调节框架大小</title>
04 </head>
05 <frameset cols="50,30,20">
06     <frame noresize="noresize" src="http://www.163.com" />
07     <frame src="http://www.sina.com.cn" />
08     <frame src=http://www.hao123.com>
09 </frameset>
10 </html>

```

【代码解析】第5~9行代码表示框架的3个网页文档中，第一个占用50%的宽度，显示网易首页内容，且不能调节框架大小，而第二个和第三个网页文档合起来占用50%的宽度，而且是默认的可以调节框架大小。效果如图8-6所示。

图8-6中第一个网页文档中的边框不能调节大小，鼠标指针放上去没有显示出可以拉伸的箭头，而放在新浪网首页的网页文档右边框处能显示出可以调节大小的箭头，这说明 noresize 对第一个即网易所在的网页文档不起作用。



图 8-6 不可调节框架大小

8.3.2 浏览器不支持框架

在以前的旧浏览器中不支持框架，为了给这些不支持框架的浏览器用户友好提醒，可以用 `<noframes></noframes>` 标记对显示文字，说明该浏览器不支持框架，支持框架的浏览器不再显示 `<noframes></noframes>` 里的内容。代码8-5说明浏览器不支持框架。

代码 8-5 浏览器不支持框架

```

-----文件名: 浏览器不支持框架.html-----
01 <html>
02 <head>
03 <title>不支持框架</title>
04 </head>
05 <frameset rows="40,60">
06     <frame src="http://www.sina.com.cn">
07     <frame src="http://www.163.com">
08 </frameset>
09 <noframes>对不起，您的浏览器不支持框架</noframes>
10 </html>

```

【代码解析】可以在每个网页的框架里面加上 `<noframes></noframes>` 标记对（第9行代码），当用以前的旧浏览器浏览网页中的框架时，会显示 `<noframes></noframes>` 的内容，而忽略

<frameset></frameset>标记对，所以浏览器显示出<frameset></frameset>上的内容时，表明该浏览器支持框架。效果如图 8-7 所示。

为了让不支持框架的浏览器访问带有框架的网页，建议每次都带上<noframes></noframes>标记对。



图 8-7 noframes 用法

8.3.3 设置框架边框

当存在框架时，有很多个网页文档在里面，如果按默认有边框的话，框架看起来很不美观。在出现多个网页文档时，可以设置框架边框，在<frameset>中用 frameborder 属性表示。

```
<frameset frameborder=#>
```

通过对 frameborder 的设置，就可以控制框架的边框，其中#代表设置的值，有两种表示方法。

- 第一种方法用英语表示，yes 表示有边框，默认时也是有的，用 no 代表无边框。
- 第二种方法是用数字 1 表示有边框，0 表示无边框。

下面用代码 8-6 表示无边框，注意，只在设置 frameborder 时与有边框不同而已。

代码 8-6 框架边框隐藏

```
-----文件名：框架边框隐藏.html-----
01 <html>
02 <head>
03 <meta>
04 <title>设置框架边框</title>
05 </head>
06 <frameset cols="40%,*" frameborder="no">
07     <frame src="a.html" />
08     <frame src="b.html" />
09 </frameset>
10 </html>
```

【代码解析】第 6~9 行代码用一个网页表示两个网页文档，框架边框设置为无，效果如图 8-8 所示，图 8-9 是设置为有边框的效果，即框架默认的边框。



图 8-8 框架无边框



图 8-9 框架有边框

有时为了隐藏框架边框，通过设置 frameborder 的值可以达到美观效果，是不是比有边框的网页清秀点？当有多个框架页面存在时，隐藏框架边框更显得重要。

注意 边框与滚动条不同，如果出现滚动条，比较难判断有没有边框，请读者仔细观察。

8.3.4 设置滚动条

当网页文档中的高度超过当前浏览器的高度时，滚动条会自动出现，滚动条的作用是方便拖拉网页窗口，以便查看显示不到的网页部分。

滚动条比较方便查看显示不到的部分，不过有时出现滚动条也会带来不便，所以要设置框架滚动条使其不显示。设置滚动条是在<frame>里面，用 scrolling 属性，后面的值可以选择 auto、yes、no，分别表示自动滚动条、有滚动条和无滚动条。代码 8-7 对框架滚动条进行设置。

代码 8-7 设置滚动条

```
-----文件名：设置滚动条.html-----
01 <html>
02 <head>
03 <meta>
04 <title>设置框架滚动条</title>
05 </head>
06 <frameset cols="40%,*">
07     <frame src="http://www.sina.com.cn" scrolling="no" />
08     <frame src="http://www.163.com" />
09 </frameset>
10 </html>
```

【代码解析】第 6~9 行代码表示一个网页的两个框架文档：第一个文档设置了滚动条为无（第 7 行代码）；第二个文档没设置，即为默认的 auto。效果如图 8-10 所示。

8.4 框架链接

框架链接有导航框架和内联框架，导航框架是在<frameset></frameset>基础上的链接，而内联框架指在网页内部的框架链接其他网页，重要的是内联框架包括在<body></body>中的框架。

8.4.1 导航框架

导航框架是在网页框架的<frame>中加入 name 属性，表示该<frame>的名称，然后通过用<a>标记的链接，并用 target 等于<frame>的名称，那所得到的链接地址网页会显示在该<frame>中。代码 8-8 表示网页中的导航框架。

代码 8-8 导航框架

```
-----文件名：导航框架.html-----
01 <html>
02 <head>
03 <title>导航框架</title>
04 </head>
05 <frameset cols="20%,*">
```



图 8-10 设置滚动条

```

06 <frame src="导航框架链接.html">
07 <frame src="http://www.163.com" name="myframe">
08 </frameset>
09 </html>

```

【代码解析】代码 8-8 说明在网页的框架中，第一个网页文档占 20% 的宽度，且显示的是“导航框架链接.html”页面（第 6 行代码），剩下的是第二个网页文档宽度，显示的是网易首页页面（第 7 行代码）。代码 8-9 表示导航框架链接网页源代码。

代码 8-9 导航链接

```

-----文件名：导航链接.html-----
01 <html>
02 <head>
03 <title>导航框架</title>
04 </head>
05 <a href ="http://www.sina.com.cn" target ="myframe">导航链接</a>
06 </html>

```

【代码解析】代码 8-9 表示一个导航框架的链接。与其他链接不同的是，target 后面接的是 <frame> 定义过的名称（第 5 行代码），表示内容在该 <frame> 内显示。图 8-11 所示的是网页刚载入的效果。当单击导航链接后，右边会显示新浪的网页文档，而左边的内容不变，效果如图 8-12 所示。

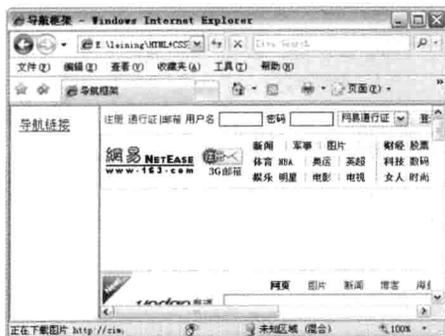


图 8-11 导航框架链接前



图 8-12 导航框架链接后

说明 导航框架很有用，读者观察一下有哪些网站利用了这个技术。

8.4.2 内联框架

内联框架存在于 <body></body> 的单个 HTML 文件中，可以链接其他网页并显示它，即在一个页面中嵌入一个框架窗口来显示另一个页面的内容，叫做浮动框架，也称内联框架。其语法是用 <iframe></iframe> 标记对表示，或者可以用 <iframe /> 表示，后面的斜线是 iframe 的结束符。代码 8-10 表示内联框架。

代码 8-10 内联框架

```

-----文件名：内联框架.html-----
01 <html>
02 <head>

```

```

03 <title>内联框架</title>
04 </head>
05 <body>
06 <iframe src="http://www.sina.com.cn" />
07 </body>
08 </html>

```

【代码解析】第6行代码为<iframe></iframe>的使用方法。效果如图8-13所示。

技巧

<iframe src="URL" width="x" height="x" scrolling="[OPTION]" frameborder="x" allowtransparency style="background-color: "></iframe>,可以设置<iframe>的属性值,这与其他设置都一样。



图8-13 内联框架

8.5 框架实例

框架可以给网页带来很大的方便,框架在网页中经常用到,如在一张网页中设置不同的部分,可以对经常变动的部分用框架表示。更新页面只对其中的框架进行更新即可,所以框架可以将多张网页组成一个网页,在这张网页中来显示这些网页的信息。

框架最常用在网页的页头和页脚,页头指的是网页的头部或上面部分,相反,页脚指的是网页的尾部。在正常的网站中,网页的头部和尾部是不需要变化的,即使要变化也是对整个网站的全部网页统一更新。这时,框架就有用了,在这些网页中插入两个框架网页,分别放在网页的头部和尾部,以后网站更新,只需要对框架页面更新,即可更新全部的网页头部和尾部。

本实例演示框架的用法。在框架集中,列出6个框架,通过这6个框架来表示框架的特性,如设置框架的 noresize 后不能调节框架大小,还可以设置框架的链接,同时可以在不同的框架中显示,以下是框架的用法实例步骤。

8.5.1 在网页中插入框架

在网页中插入两行三列的框架,分别显示不同的框架文档。在网页中建框架,设置 cols 为“50%,50%”,表示两行各占一半,而 rows 要设置为显示三列的数字,所以输入“30%,30%,40%”,即表示两行三列,如代码8-11所示。

代码8-11 框架实例手把手

```

-----文件名: 框架实例手把手.html-----
01 <html>
02 <head>
03 <title>框架实例手把手</title>
04 </head>
05 <frameset rows="50%,50%" cols="30%,30%,40%">
06     <frame src="a.html">
07     <frame src="a.html">
08     <frame src="a.html">
09     <frame src="a.html">
10     <frame src="a.html">

```

```

11     <frame src="a.html">
12 </frameset>
13 </html>

```

【代码解析】第 5~12 行代码表示两行三列，显示 6 个框架文档，所以用 6 个<frame>设置每个框架文档的页面，效果如图 8-14 所示。

8.5.2 设置框架的滚动条和边框

左边的框架文档设置成不可调节大小状态，且设置中间的文档为没有滚动条和框架边框。要想将左边的文档设置成不可调节大小状态，就对第 1 个和第 4 个<frame>的 noresize 进行设置；在中间的框架文档设置没有滚动条和框架边框，就对第 2 个与第 5 个<frame>的 scrolling 和 frameborder 进行设置。代码 8-12 完成上面的设置。

代码 8-12 设置滚动条和边框

```

-----文件名：框架实例手把手.html-----
01 <html>
02 <head>
03 <title>框架实例手把手</title>
04 </head>
05 <frameset rows="50%,50%" cols="30% ,30%,40%">
06     <frame src="a.html" noresize="noresize">
07     <frame src="http://www.sina.com.cn" scrolling="no" frameborder="no">
08     <frame src="a.html">
09     <frame src="a.html" noresize="noresize">
10     <frame src="http://www.163.com" scrolling="no" frameborder="0">
11     <frame src="a.html">
12 </frameset>
13 </html>

```

【代码解析】第 6 行、第 9 行代码对第 1 个和第 4 个文档设置了不可调节大小状态，为了让第 3 个和第 6 个文档显示更多内容，不做任何设置，保持默认条件，这样当内容多时就会出现滚动条（第 8、11 行代码）。再设置第 2 个和第 5 个框架为没有滚动条与边框的状态（第 7、10 行代码），其他的文档保存不变。效果如图 8-15 所示。



图 8-14 用 cols 和 rows 划分两行三列



图 8-15 设置滚动条和边框

8.5.3 设置不支持框架文字

如果不支持框架的话，在<frameset></frameset>后面加上<noframes></noframes>标记对即可，如代码 8-13 所示。

代码 8-13 设置不支持框架文字

```
-----文件名：框架实例手把手.html-----
01 <html>
02 <head>
03 <title>框架实例手把手</title>
04 </head>
05 <frameset rows="50%,50%" cols="30%,30%,40%">
06     <frame src="a.html" noresize="noresize">
07     <frame src="http://www.sina.com.cn" scrolling="no" frameborder="no">
08     <frame src="a.html">
09     <frame src="a.html" noresize="noresize">
10     <frame src="http://www.163.com" scrolling="no" frameborder="0">
11     <frame src="a.html">
12 </frameset>
13 <noframes>
14 <body>
15 <p>对不起，您的浏览器不支持框架</p>
16 </body>
17 </noframes>
18 </html>
```

【代码解析】代码 8-13 的第 13~17 行代码对框架设置了<noframes>标记对，标记对中有 1 行文字，此属性表示当用户的浏览器不支持框架时，不会显示框架中的内容。相反，会显示<noframes></noframes>标记对中的内容，说明用户的浏览器不支持框架。当然，用户的浏览器支持框架时不会显示出<noframes></noframes>标记对中的内容。

注意

为了兼顾有的浏览器不支持框架，建议在做框架的网页布局时，设置<noframes> </noframes>标记对。

8.5.4 设置框架导航链接

为第 1 个文档设置链接并给第 3 个和第 6 个文档的<frame>设置名称，单击不同的链接，在它们的区域内显示内容。代码 8-14 表示导航链接。

代码 8-14 设置导航链接

```
-----文件名：框架实例手把手.html-----
01 <html>
02 <head>
03 <title>框架实例手把手</title>
04 </head>
05 <frameset rows="50%,50%" cols="30%,30%,40%">
06     <frame src="c.html" noresize="noresize">
07     <frame src="http://www.sina.com.cn" scrolling="no" frameborder="no">
08     <frame src="a.html" name="myframe1">
09     <frame src="a.html" noresize="noresize">
10     <frame src="http://www.163.com" scrolling="no" frameborder="0">
11     <frame src="a.html" name="myframe2">
12 </frameset>
13 <noframes>
14 <body>
15 <p>对不起，您的浏览器不支持框架</p>
```

```

16 </body>
17 </noframes>
18 </html>

```

【代码解析】上面代码从原来的基础上做了部分修正，首先在第一个<frame>中链接了 c.html 网页（第 6 行代码），在 c.html 做了链接，并设置 target 为本网页设置好的<frame>名称。图 8-16 所示为单击前的初始状态；图 8-17 所示为分别单击两个链接后，显示的最后结果。

在图 8-16 中，在第 1 个框架中有两个链接，而第 3 个和第 6 个框架为框架的默认状态。当单击第 1 个框架中的第一行文本链接时，在第 3 个框架中会显示链接后的效果，同理，当单击第 1 个框架中的第二行文本链接时，在第 6 个框架中会显示出链接后的效果，因为在<frame>标记中，设置了 name 参数，链接后的效果如图 8-17 所示。



图 8-16 单击第一个文档的文字链接前的状态



图 8-17 单击第一个文档的文字链接后的状态

技巧

现在的系统都用框架，如进入邮箱后，保持一个网页不动；如网易邮箱中的页面进去后，会自动播放音乐；如发送邮件或收取邮件时也应用到了框架。

8.6 小结

本章介绍了框架的用法。框架在网页中比较重要，有时可以用来显示多个独立文档，又要相互不影响，可以显示多行多列。通过 cols 和 rows 设置，用<frameset>的属性设置框架的边框是否可调节框架的大小、滚动条等。在<frameset>中还可以包括<iframe>，显示文档还可以嵌入 HTML 文档，用框架导航链接可以在一张网页中操作不同的网页页面。最后通过实例来演示框架的知识点用法。

8.7 习题

一、填空题

1. <frameset></frameset>在<html></html>标记对内，不能与_____标记并列。
2. 在出现多个网页文档时，可以设置框架边框，在<frameset>中用_____属性表示。
3. 框架分栏有_____和_____两种形式。

二、上机实践

1. 按下面的样式编写框架。



【提示】

frameset-----框架标识
 frame-----框架内容标识
 rows-----框架分行属性
 cols-----框架分列属性

2. 按下面的样式编写框架网页。



【提示】

frameset-----框架标识
 frame-----框架内容标识
 rows-----框架分行属性
 cols-----框架分列属性
 target-----制定目标链接在哪个框架展开
 frameborder-----框架边框属性
 noresize-----对应框架不可变动

第9章 网页中的动态图文和多媒体

不管是新闻网站还是电子商务网站,对及时新闻的显示或广告的滚动播出,都会用到移动显示。所谓的移动就是让对象动起来,对象包括文字和图片。移动属性语法用 `marquee` 表示,是双标记对,标记对里放置的是将要移动的对象,对象可以是图片或文本等。

人们上网的机会越来越多,网上的内容也越来越丰富,很多人在看电视与玩电脑之间选择了后者。通过电脑除了可以浏览网页新闻外,也可以进视频网站看视频新闻(从电视上录制并上传到网上),还可以看电影等。网页多媒体的出现,使网页的功能更丰富,很多人除了上网聊 QQ、玩游戏外,还可以进视频网站看视频,而这种视频正是网页多媒体的应用。

学习本章,将会了解以下内容。

- 图片和文字的移动;
- 移动实例;
- 自动播放多媒体;
- 设置多媒体循环播放;
- 隐藏多媒体面板;
- 设置多媒体面板的大小和外观。

9.1 移动属性的基本语法

在网页中,可以让文字或图片在指定的区域动起来,可以由上到下或者由下到上移动,还可以左右移动,也可以设置移动的次数。这些效果往往在电子商城中对卖得很火的产品设置移动,有的公告也可以移动,本节分析和讲解这样的效果。

移动属性是双标记的,其语法用 `marquee` 标记对表示,里面放的内容就是要移动的对象。完整语法如下。

```
<marquee>
    将要移动的文字或图片对象
</marquee>
```

属性 `marquee` 默认的是向左循环移动,对象移动是在其所在的父对象的有效区域内。代码 9-1 表示 `marquee` 基本语法的用法。

代码 9-1 移动的基本语法

```
-----文件名: 移动的基本语法.html-----
01  <html>
02  <head>
03      <title>移动的基本语法</title>
```

```

04 </head>
05 <body>
06 <!--移动标记对, 里面的文本将会移动-->
07 <marquee>
08     HTML+CSS完全自学手册---正在移动
09 </marquee>
10 </body>
11 </html>

```

【代码解析】第5~10行代码在<body></body>中放入属性 marquee 标记对, 然后输入要移动的文字, 这是最简单的移动表示(还有几个可选参数没设置, 不设置就是默认的)。移动前的效果如图 9-1 所示, 移动后的效果如图 9-2 所示。



图 9-1 对象移动前



图 9-2 对象移动后

在设置 marquee 时, 文本就在默认的区域从右向左移动, 当在不同时间查看网页时, 文本在网页的位置也不一样, 图 9-2 正处于从右边向中间移动的文本效果, 其中速度也是默认的。

说明 其实图9-1和图9-2二者都有移动, 在本书上只用两张网页移动位置不同来比较, 要想看到真正的效果, 请读者自己编写代码或直接在随书光盘的源代码第9章中查看, 稍后介绍对移动属性设置参数值。

9.2 文字的移动

由 9.1 节可以知道, 文字的移动就是用属性 marquee 标记内对文字进行操作, 不过在本小节中除了要用到上一节中介绍的语法外, 还要对本行的文字设置不同的移动方向。在实际应用中文字并非按默认向左移动, 可能需要向不同的方向移动才能表达出效果, 这就需要控制对象移动的方向。

设置移动的方向是在移动属性 marquee 中。设置对象移动的方向其语法用 direction 表示, 可接 left 表示向左, 也是默认值, 还可以接 right, 表示向右移动。完整语法如下。

```
<marquee direction=#>
```

在属性 marquee 中, 设置方向值 direction, 其值#可以选择 left 或 right, 表示对象向左还是向右移动。当不设置 direction 值时, 系统默认向左, 与设置 direction 值为 left 是一样的效果。在代码 9-2 中, 一行文字表示向左移动, 另一行文字表示向右移动。

代码 9-2 文字的移动方向

```
-----文件名: 文字的移动方向.html-----
```

```

01 <html>
02 <head>

```

```

03 <title>文字的移动方向</title>
04 </head>
05 <body>
06 <!--当不设置direction 值时表示默认向左-->
07 <marquee>
08     HTML+CSS完全自学手册---正在向左移动
09 </marquee>
10 <!--设置direction 值为右-->
11 <marquee direction="right">
12     HTML+CSS完全自学手册---正在向左移动
13 </marquee>
14 </body>
15 </html>

```

【代码解析】第7~13行代码用了两个 marquee 来移动两行文字，第一行表示默认的方向（没有写出 direction 值，按系统默认方向移动，第7行代码），而第二行将 direction 设置为 right（第11行代码），即向右移动，移动前的效果如图9-3所示，移动后的效果如图9-4所示。

比较两图，在不同的时刻文本的位置在变化，正是由于文本移动产生的作用，而文本的流向是由右和向左，速度也是默认的（流向和速度等都可以设置），文本移动后的效果如图9-4所示。

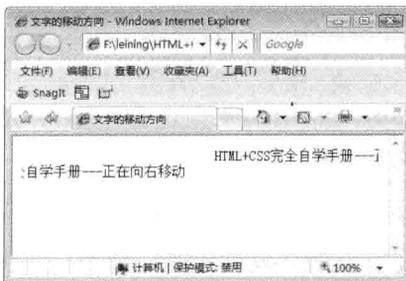


图 9-3 对象往各方向移动前

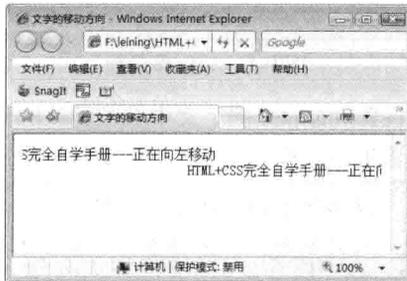


图 9-4 对象往各方向移动后

注意 如果设置 direction 为 left，与默认参数一样，可以省去，但要注意该参数的用法和设置。

9.3 图片的移动

如今的电子商城为了展示商品，都通过移动商品图片来表现出效果：把所有的商品图片放到 marquee 中，然后通过处理使图片移动，达到在很小的空间展示更多的内容的效果，而且加强了用户的体验感。

下面的代码表示一张图片在标记 <marquee> 中移动，而且设置了对象的宽为 600 像素，高为 200 像素，背景颜色为 #666666，如代码 9-3 所示。

代码 9-3 图片的移动

```

-----文件名: 图片的移动.html-----
01 <html>
02 <head>
03 <title>图片的移动</title>
04 </head>
05 <body>

```

```

06 <!--设置移动区域大小-->
07 <marquee width="600px" height="200px" bgcolor="#666666">
08     图片的大小在marquee中移动
09 </marquee>
10 </body>
11 </html>

```

【代码解析】第7~9行代码设置了图片的移动，使用图片和文本从右往左移动，而且是不停地循环移动，这时图片和文本属于同一个对象，文本与图片默认对齐，即图片与文本底部对齐。效果如图9-5所示。

技巧 设置对象的移动时，可以在有限的区域内设置更多内容，所移动的对象只在这个区域中。

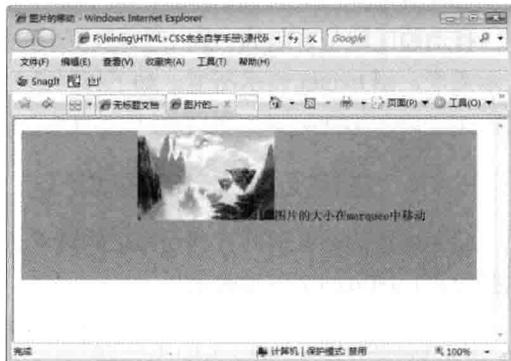


图 9-5 图片的移动

9.4 设置对象的移动

当引入 marquee 后，如按默认值，表现的效果并不是当初想要的，还需要进行设置，如移动的方向、区域的大小、是否循环移动等，这些都在标记 <marquee> 中设置。

下面从移动方向、移动方式、移动速度、延时、宽度与高度来介绍标记的设置。从上面的实例中可以知道，默认的移动方向是从右向左，而在标记中可以设置从左向右或从上到下等方向移动。

9.4.1 文本移动的方向

移动的方向用参数 direction (direction 的中文意思就是方向) 表示。这个参数的取值有四个：left (左)、right (右)、up (上)、down (下)。通过设置方向值就可以设置对象往哪个方向移动，代码如下。

```
<marquee direction=up>向上移动</marquee>
```

代码表示文本“向上移动”，即从下往上移动，而且是一次又一次不停地移动着。

9.4.2 文本的滚动循环

如果想设置对象移动的方式，例如滚动循环，语法用 behavior 表示，其后可选择参数为：scroll (循环移动)、slide (只移动一个回合)、alternate (来回移动)，如下面代码表示来回移动。

```
<marquee direction=left behavior=alternate>哈哈，我来回走！</marquee>
```

以上代码设置了文本来回移动，当然还可以设置文本在 <marquee> 标记中循环的次数，设置循环用 loop，其后可接上具体数值表示循环的次数，如下所示设置只循环三次。

```
<marquee loop=3>我只走三次哦</marquee>
```

9.4.3 文本的移动速度

还可以用 scrollamount 来设置移动的速度，可以设置移动很快，也可以设置移动很慢，其速度

通过数值来设置，值取正整数，数值越大，速度越快。如下代码表示文本在标记中移动的速度。

```
<marquee scrollamount=25>看，我走多快！</marquee>
```

设置好移动的速度后，还可以设置对象的延时。延时用 `scrolldelay` 表示，以时间为单位，用毫秒表示，1000 毫秒表示 1 秒。下面代码表示延时 1 秒。

```
<marquee scrolldelay=1000>我走一走，停一停</marquee>
```

移动的区域是有限的，那么想得到更大的区域可以设置移动的区域。与其他属性一样，用 `width` 表示宽度，用 `height` 表示高度。如下代码设置了区域的宽和高。

```
<marquee width=200 height=200 bgcolor=pink direction=down>这个面积不够我移动！</marquee>
```

9.4.4 应用文本的全部移动标记

前面讲述的是常用的对移动标记的设置，下面用代码来演示上述所有功能。在实践中可以结合这些设置让网页的字体动起来。

代码 9-4 设置对象移动

```
-----文件名：设置对象移动.html-----
01 <html>
02 <head>
03 <title>设置对象的移动</title>
04 </head>
05 <body>
06 <!--设置不同对象的移动-->
07 <marquee direction=up>向上移动</marquee>
08 <marquee direction=left behavior=alternate>哈哈，我来回走！</marquee>
09 <marquee loop=3>我只走三次哦</marquee>
10 <marquee scrollamount=25>看，我走多快！</marquee>
11 <marquee scrolldelay=1000>我走一走，停一停</marquee>
12 <marquee width=200 height=200 bgcolor=pink direction=down>这个面积不够我移动！
13 </marquee>
14 </body>
15 </html>
```

【代码解析】在代码 9-4 中，通过设置不同对象的移动，来表现标记 `marquee` 的用法。在第 1 个 `marquee` 标记中，设置了向上移动的文本文对象（第 7 行代码）；在第 2 个 `marquee` 标记中，设置对象为来回移动，设置参数 `behavior` 的值为 `alternate`，使对象在区域中来回移动（第 8 行代码）；在第 3 个移动标记中，设置参数 `loop` 为 3，表示只移动 3 次（第 9 行代码）；在第 4 个移动标记中，设置参数 `scrollamount` 的值为 25，表示移动的速度（第 10 行代码）；同理，第 5 个标记中，设置参数 `scrolldelay` 的值为 1000，数值单位默认为毫秒，表示移动延迟为 1000 毫秒，刚好为 1 秒钟（第 11 行代码）；在最后的移动标记中，设置了移动的区域为 200 像素的宽和高，背景为 `pink`（粉色），移动的方向为 `down`（向下）（第 12 行代码）。效果如图 9-6 所示。

注意

在设置移动标记的参数时，可以在标记内设置多个参数，综合这些参数在网页中表现出效果。

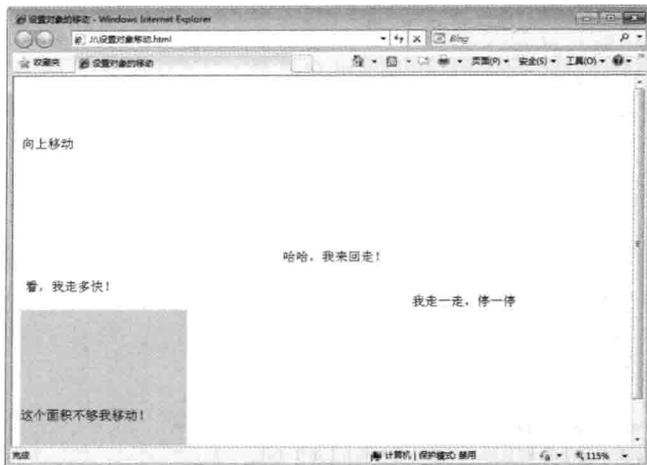


图 9-6 图片移动

9.5 移动实例——电影结束效果

本实例演示电影播放完后出现演员表的效果。演员表等文字是向上移动的，而且上下两部分的背景黑边用两个层设置定位。下面用 HTML 标记在网页中来实现这种样式。

移动对象需要用到移动标记对 `<marquee></marquee>`，标记对内的对象即是移动的对象，默认是从右向左移动，而且是不停地移动。如要改变这些默认设置，就要设置标记 `<marquee>` 中的参数。

实例部分主要列出其代码，不再按步骤讲解，因为还没讲到 CSS 知识，很多内容读者还不太明白，但只要明白这种效果是用 `marquee` 标记演示出的就行，通过对以后内容的学习会慢慢理解本节中的用法。下面通过 CSS 与 HTML 标记结合，来演示电影结束时的效果，如代码 9-5 所示。

代码 9-5 移动实例手把手——让图片像电影动起来

```
-----文件名：移动实例手把手——让图片像电影动起来.html-----
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
02 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
03 <html>
04 <head>
05     <title>移动实例手把手——让图片像电影动起来</title>
06 <style type="text/css">
07 #movie
08 {
09     /*CSS代码：设置容器层，margin设置auto表示对象居中*/
10     margin:auto;
11     /*设置宽度和高度*/
12     width:350px;
13     height:280px;
14     /*设置边框*/
15     border:1px solid #000000;
```

```
16      /*设置容器层为相对定位*/
17      position:relative;
18      /*字体大小为15像素*/
19      font-size:15px;
20      /*文本居中*/
21      text-align:center;
22  }
23  #movieup
24  {
25      /*容器层上面的黑层,宽为100%,高为50像素*/
26      width:100%;
27      height:50px;
28      /*绝对定位*/
29      position:absolute;
30      /*与顶端的距离为0*/
31      top:0px;
32      /*设置背景为黑色*/
33      background:#000000;
34  }
35  #moviedown
36  {
37      /*容器层的下面黑色层,与上面的层一样。*/
38      width:100%;
39      height:50px;
40      position:absolute;
41      bottom:0px;
42      left:0;
43      background:#000000;
44  }
45  </style>
46  </head>
47  <body>
48  <div id="movie">
49      <div id="movieup"></div>
50      <marquee direction="up">
51          <p>
52              HTML+CSS完全自学手册
53          </p>
54          <p>
55              HTML+CSS完全自学手册
56          </p>
57          <p>
58              HTML+CSS完全自学手册
59          </p>
60          <p>
61              HTML+CSS完全自学手册
62          </p>
63      </marquee>
64  </div id="moviedown"></div>
```

```

65 </div>
66 </body>
67 </html>

```

【代码解析】在代码中，设置了一个容器层，然后在第 33 行、第 43 行代码用定位（position）把容器层的上下两处设置为黑色背景的，像电影一样，而将文字设置为往上移动，类似于电影的结束画面。效果如图 9-7 所示。

标记marquee的其他用法在前面中已介绍过，
注意 如读者还要添加其他效果，可以直接在<marquee>中添加参数。

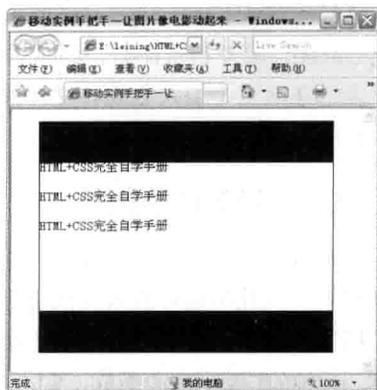


图 9-7 电影结束效果

9.6 网页多媒体

网页多媒体可以播放 flash、mid、wav、mp3 等格式的多媒体文件，最简单的用法是直接标记中插入文件。其中多媒体标记用 embed 表示，embed 标记的参数比较多，参数不同，对多媒体的控制不一样，如 autostart 表示自动播放，loop 表示循环播放等。这些都建立在多媒体标记 embed 中。

9.6.1 网页多媒体的基本语法

多媒体标记的使用使网页内容从文字显示扩大到播放动画、视频和音乐等内容，首页在标记后，必需引入多媒体文件，保证多媒体路径正确，才可能在网页中正确显示，其语法代码如下所示。

```
<embed src=url>
```

其中“=”后的 url 表示多媒体的路径，可以是相对路径也可以是绝对路径。要说明的是，网页可以支持多种多媒体样式，且都是通过 embed 标记的 src 来插入各种多媒体，格式可以是 midi、wav、aiff、au、mp3 等。

最基本的样式是通过 embed 引入一个多媒体文件，在指定的区域显示。代码 9-6 表示在网页中引用一个 MP3 文件，使其在网页中播放出来。播放的前提是系统已安装 Windows Media Player 播放器，因为多媒体文件的处理都是通过此播放器，本实例中的多媒体文件播放是由系统自带的 Windows Media Player11 播放器所支持，Windows Media Player 播放器在安装系统时就存在了，只是因系统不同而版本不一样。在网页中播放多媒体文件的效果如图 9-6 所示。

代码 9-6 多媒体的基本语法

```

-----文件名：多媒体的基本语法.html-----
01 <html>
02 <head>
03 <title>多媒体的基本语法</title>
04 </head>
05 <body>
06     下面是mp3文件：
07     <br>
08     <!--多媒体标记-->

```

```

09     <embed src="smak slow.mp3">
10  </body>
11  </html>

```

【代码解析】在<body></body>标记中，加入了多媒体标记 embed（第 9 行代码），同时引入同目录下的 mp3 文件，表示在网页中引入多媒体文件并可以播放，在网页中表现出的效果如图 9-8 所示。

9.6.2 设置自动播放

网页载入后，文件立刻播放，从而听到动听的音乐或看到视频，但并不是每载入完网页后就播放，可以设置进入是否是暂停状态。

标记<embed>默认的是自动播放，所以要设置成载入页面时不自动播放，需要对参数 autostart 设置。autostart 表示自动播放，其后可接 true 和 false，true 表示打开自动播放；也是默认值，而 false 表示没有打开自动播放。当标记 embed 设置 autostart 时，且网页完全载入，多媒体不会自动播放，这样就达到上面的效果了。参数 autostart 可接两个值。其自动播放的完整语法如下。

```

<!-- 自动播放设置-->
autostart=true|false

```

参数 autostart 的功能表示音频或视频等多媒体文件是否在载入完之后就自动播放。值 true 表示音乐等文件在下载完之后自动播放；另一个值 false 表示音乐文件在下载完之后不自动播放，默认值是 true。

与参数 src 一样，参数 autostart 所接的值放在标记<embed>中，用来控制多媒体在网页中的播放。如下代码表示网页载入时自动播放。

```

<!--设置为自动播放-->
<embed src="your.mid" autostart=true>

```

以上代码表示网页载入多媒体文件时，不会自动播放，只有当用户单击面板（播放时的功能菜单）中的“开始”按钮才开始播放。

```

<!--设置为非自动播放-->
<embed src="your.mid" autostart=false>

```

并不是所有的多媒体文件都需要自动播放，这时需要设置<embed>标记的自动播放参数。为了使网页完全载入后不立即自动播放，则设置 autostart 为 false；要想播放，只有手动单击“开始”按钮。代码 9-7 表示网页多媒体载入时的多媒体 mp3 文件，载入后，不会自动播放。

代码 9-7 多媒体的自动播放

```

-----文件名：多媒体的自动播放.html-----
01  <html>
02  <head>
03  <title>多媒体的自动播放</title>
04  </head>
05  <body>
06      下面是mp3文件,设置自动播放为false:
07      <br>
08      <embed src="smak slow.mp3" autostart="false">

```



图 9-8 多媒体的基本语法

```
09 </body>
10 </html>
```

【代码解析】在第8行代码网页引入多媒体文件 smak slow.mp3，当整个网页完全载入时，即多媒体控制面板出现时，播放进度条不会动，因为已设置 autostart 为 false，所以网页并没有声音，效果如图 9-9 所示。

9.6.3 设置多媒体循环播放

在博客中，为了使用户轻松地浏览文章，设置一首好听的歌曲是很有必要的。在 9.2 节中，只能设置在网页完全载入时，会不会自动播放的功能，当这首歌播放完成后，又该如何处理？是停止播放，还是再播放一次？

对多媒体文件的播放次数可通过<embed>标记设置，即设置文件可播放一次或无数次。所谓无数次就是设置多媒体循环无数次，用参数 loop 来设置。与自动播放一样，loop 也是用 true 和 false 来表示，当 loop 设置为 true 时，多媒体文件播放无数次，直到单击“停止”按钮或“暂停”按钮为止；同理，设置 loop 为 false 时，只播放一次就停止。下面代码表示 loop 在标记中的语法。

```
<embed src=url loop="true|false">
```

与其他参数一样，在引入一个多媒体文件的同时，设置多媒体循环播放，该参数规定音频或视频文件是否循环及循环次数。属性值为正整数值时，音频或视频文件的循环次数与正整数值相同；属性值为 true 时，音频或视频文件循环；属性值为 false 时，音频或视频文件不循环。如下代码所示。

```
<embed src="your.mid" autostart=true loop=2>
```

以上代码表示在引入多媒体文件 your.mid 时，同时设置了自动播放，且循环播放两次，两次播放完后，停止播放。如以下代码所示。

```
<embed src="your.mid" autostart=true loop=true>
```

在上面代码中设置多媒体文件为自动播放，且播放是无限循环，只有当外界干预时才会停止，如单击关闭网页、刷新后等。下面代码刚好相反，设置 loop 值为 false，表示不出现循环播放。

```
<embed src="your.mid" autostart=true loop=false>
```

在设置循环时，与自动播放没直接联系，即使网页载入完成后不是自动播放状态，但单击“开始”按钮后，循环播放仍是 loop 设置的值。在代码 9-8 中表示在<embed>标记，设置多媒体文件循环播放。

代码 9-8 多媒体循环播放

```
-----文件名：多媒体循环播放.html-----
01 <html>
02 <head>
03 <title>多媒体循环播放</title>
04 </head>
05 <body>
06     下面是mp3文件；
07     <br>
```



图 9-9 多媒体的自动播放

```

08     <embed src="smak slow.mp3" loop="true">
09     <br>
10     给多媒体文件设置循环播放
11 </body>
12 </html>

```

【代码解析】第8行代码设置了多媒体播放时为循环播放，将参数 loop 设置为 true，即无限次的循环播放，如图 9-10 所示。

9.6.4 设置面板的隐藏

在网页中引入多媒体文件播放，必然存在多媒体播放区域，用来告诉浏览者该区域中有多媒体播放。多媒体所在的区域称为多媒体控制面板，其内部有开始/暂停/声音等调节按钮。有了面板后，用户在网页中控制播放更方便了。但有时只要一载入网页就开始播放多媒体，而且不需要面板显示，这时可以把面板隐藏。

多媒体控制面板隐藏后，用户看不到网页中有控制面板，也就不需要设置开始、暂停等。设置隐藏面板的参数还需要依赖于标记 <embed>。与其他参数一样，可以用 true 和 false 来控制多媒体是否隐藏。隐藏多媒体面板用 hidden 表示，其后所接的参数值为 true 或 false，当设置 hidden 值为 true 时，表示多媒体面板隐藏，相反，设置参数值为 false 时，不会隐藏控制多媒体面板，这也是默认值。参数 hidden 在标记 <embed> 中的语法如下。

```
<embed src="url" hidden=true|false>
```

在标记 <embed> 中，引入路径为 url 的多媒体文件，然后设置面板是否隐藏，即设置 hidden 的参数值为 true 表示隐藏，相反，设置为 false 是不隐藏，让多媒体面板显示，即默认值。

同时参数值 false 也可以用 no 取代，即可表示成 true:隐藏面板，no 表示显示面板，同时也是默认值。隐藏面板的代码如下。

```
<embed src="your.mid" hidden=true>
```

以上代码表示在 <embed> 标记中，引入多媒体文件 your.mid，并设置面板隐藏，即 hidden 设置为 true，当在网页中显示时，面板将不会出现，同理，以下代码表示在标记 <embed> 中设置不隐藏面板。

```
<embed src="your.mid" hidden=no>
```

同理，要设置面板显示可以设置 hidden 参数值为 no，也可以用 false 取代，效果与不设置 hidden 参数都是一样的，因为没有设置则代表默认值。代码 9-9 演示了设置隐藏面板的用法。

代码 9-9 隐藏面板

```

-----文件名：隐藏面板.html-----
01 <html>
02 <head>
03     <title>隐藏面板</title>
04 </head>
05 <body>
06     下面是mp3文件：
07     <br>

```



图 9-10 多媒体循环播放

```

08      <embed src="smak slow.mp3" hidden="true">
09      多媒体面板是隐藏的，打开网页自动播放。
10     </body>
11     </html>

```

【代码解析】第8行代码是在网页中引入多媒体文件，然后设置面板隐藏，当打开网页时，立刻能听到多媒体播放出来的声音，却看不到面板，这是因为已将 hidden 设置为 true，表示多媒体面板隐藏。效果如图 9-11 所示。

9.6.5 设置面板的大小

面板虽然可以隐藏，但很多时候却是显示的，而且显示的面板也不一定按默认的大小。那么如何来确定自己定义的面板大小？这时就需要手动设置了。这就要用到多媒体标记来设置面板容器大小，即面板的宽度和高度。

与其他属性设置一样，设置面板的宽度和高度时，用 width 表示宽度，height 表示高度。高度和宽度是距离值，设置其值时还要带上单位，最常用的单位 px (pixel)、en，二者在相同的数值下距离不同，但在此参数是用 px，请注意区别。下面代码表示在标记 <embed> 中设置其宽度和高度的语法。

```
<embed src="url" height=# width=#>
```

从前面的讲解中可以知道，以上代码中，url 表示将引入的多媒体路径，height 表示高度，width 表示宽度。当然，它们出现的顺序没限定，其中取值可以为正整数或百分数，单位为像素。该参数规定控制面板的高度和宽度，如以下代码所示。

```
<embed src="your.mid" height=200 width=200>
```

以上代码表示在引入的多媒体中，设置宽为 200，且高也为 200，默认单位为像素 (px)。代码 9-10 演示了面板大小的设置。

代码 9-10 设置面板的大小

```

-----文件名：设置面板的大小.html-----
01  <html>
02  <head>
03  <title>设置面板大小</title>
04  </head>
05  <body>
06      下面是mp3文件：
07      <br>
08      <embed src="smak slow.mp3" width="400px" height="80px" bgcolor = black>
09      <br>
10      设置了面板宽为400像素，高为80像素
11  </body>
12  </html>

```

【代码解析】第8行代码设置了面板的宽度为 400 像素，高度为 80 像素。在此区域中，载入开始面板时，用黑色背景填充设置的区域，在实际应用中，要合理地设置面板大小。本实例的效

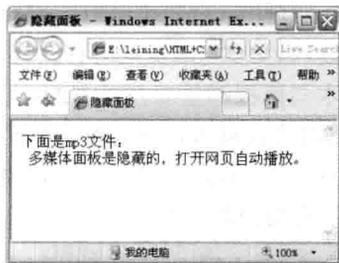


图 9-11 隐藏面板

果如图 9-12 所示。

9.6.6 设置对齐方式

在网页中，可能要用到层（DIV 标记）来包含多媒体面板，当文本与面板同时存在时，按默认的对齐方式，即文本与面板都是向左对齐的，所以出现的效果不会好，要改变这种方式就要对其设置对齐。与其他属性设置对齐一样，面板对齐的语法也用 align 表示对齐的方式，其参数值如下面表示。



图 9-12 设置面板的大小

```
align=top、bottom、center、baseline、left、right、texttop、middle、absmiddle、absbottom
```

参数 align 在标记<embed>中表示该参数控制面板和当前行中的对象的对齐方式。各参数值表示如下。

- center: 控制面板居中。
- left: 控制面板居左。
- right: 控制面板居右。
- top: 控制面板的顶部与当前行中的最高对象的顶部对齐。
- bottom: 控制面板的底部与当前行中的对象的基线对齐。
- baseline: 控制面板的底部与文本的基线对齐。
- texttop: 控制面板的顶部与当前行中的最高的文字顶部对齐。
- middle: 控制面板的中间与当前行的基线对齐。
- absmiddle: 控制面板的中间与当前文本或对象的中间对齐。
- absbottom: 控制面板的底部与文字的底部对齐。

通过学习上面的参数表示，可用下面代码表示面板与其他对象（如文字或图片）的对齐方式。

```
<embed src="your.mid" align=top>
```

以上代码表示引入 your.mid 这个多媒体文件，且控制面板与网页顶端对齐，又如下面的代码所示。

```
<embed src="your.mid" align=center>
```

上述代码表示引入 your.mid 多媒体文件的面板居中对齐。在代码 9-11 中用实例表示参数 align 在多媒体标记中的用法。

代码 9-11 多媒体面板的对齐

```
-----文件名：多媒体面板的对齐.html-----
01 <html>
02 <head>
03 <title>多媒体面板的对齐</title>
04 </head>
05 <body>
06     下面是mp3文件:
07     <br>
08     <embed src="smak slow.mp3" align="right">
09     <br>
```

```

10     多媒体面板的对齐
11 </body>
12 </html>

```

【代码解析】第8行代码引入多媒体文件的面板，同时面板设置为向右对齐，文字和面板默认条件是向左对齐，所以目前是文字在左边多媒体面板在右边。效果如图9-13所示。

说明 多媒体标记`<embed>`除了以上这些参数外，还有常用的如ID、CLASS等，这些内容将会在学习CSS内容时详解。



图9-13 多媒体面板的对齐

9.7 网页多媒体实例——在网页中播放视频

在前面的几节中，主要讲解了网页多媒体的标记及其参数的设置，其中多媒体文件只针对mp3这种格式，其他样式的多媒体文件与上面实例的用法一样。下面在网页多媒体实例中，通过在网页中播放视频来进行讲解。

9.7.1 引入视频文件

准备好视频资料，引用到网页。本实例准备了“swf”格式（Flash文件的格式）的视频文件，新建网页保存在与多媒体文件同一目录下，通过href参数链接视频文件。代码9-12表示在网页中引入视频文件。

代码9-12 多媒体实例手把手——在网页中放视频

```

-----文件名：多媒体实例手把手——在网页中放视频.html-----
01 <html>
02 <head>
03     <title>多媒体实例手把手——在网页中放视频</title>
04 </head>
05 <body>
06     <embed src="刻舟求剑.swf">
07 </body>
08 </html>

```

【代码解析】第6行代码给`<body></body>`标记对中放入`<embed>`标记，在多媒体标记中，引入了“刻舟求剑.swf”文件，其他的参数目前都是默认的，所以载入网页界面就可以看到flash文件正在播放，效果如图9-14所示。

9.7.2 设置多媒体为非自动播放

当网页载入时，默认情况是自动播放的，在本步骤中，设置为非自动播放，如代码9-13所示。



图9-14 载入多媒体文件时

代码 9-13 多媒体实例手把手—在网页中放视频

```
-----文件名：多媒体实例手把手—在网页中放视频.html-----
01 <html>
02 <head>
03     <title>多媒体实例手把手—在网页中放视频</title>
04 </head>
05 <body>
06     <embed src="刻舟求剑.swf" autostart="false">
07 </body>
08 </html>
```

【代码解析】第6行代码设置多媒体载入时为非自动播放，即网页载入时，会暂停播放，直到激活其为止（flash 文件与 mp3 等文件一样，都是播放器支持播放的格式），效果如图 9-15 所示。

9.7.3 设置多媒体无循环播放

设置多媒体循环播放时，如果是默认值，表现的是视频文件不停地播放，而当设置 loop 为 false 时，表示只播放一次就停止。代码 9-14 中，表示了多媒体无循环播放标记 loop 的用法。

代码 9-14 多媒体实例手把手—在网页中放视频

```
-----文件名：多媒体实例手把手—在网页中放视频.html-----
01 <html>
02 <head>
03     <title>多媒体实例手把手—在网页中放视频</title>
04 </head>
05 <body>
06     <embed src="刻舟求剑.swf" loop="false">
07 </body>
08 </html>
```

【代码解析】在代码 9-14 中的第 6 行代码中，设置了多媒体为无循环状态，当多媒体播放完成时，不会重新开始播放，因本实例文件默认的是循环播放，但设置 loop 为 false 后，文件只播放一次，这次播放完成时，视频处于最后一帧不动。效果如图 9-16 所示。



图 9-15 设置为非自动播放



图 9-16 设置为不循环

9.7.4 设置控制面板的大小

为调节面板大小，可在实例中设置与原面板大小不一样的面板容器大小，代码 9-15 表示设置控制面板的大小。

代码 9-15 多媒体实例手把手—在网页中放视频

```
-----文件名：多媒体实例手把手—在网页中放视频.html-----
01 <html>
02 <head>
03     <title>多媒体实例手把手—在网页中放视频</title>
04 </head>
05 <body>
06     <embed src="刻舟求剑.swf" width="400" height="300">
07 </body>
08 </html>
```

【代码解析】因文件是按原始大小显示的，当用户需要设置不同大小时，可通过 width 和 height 来确定面板的高和宽，代码 9-15 中第 6 行设置了宽为 400 像素，高为 300 像素。设置后显示画面也变大了。效果如图 9-17 所示。



图 9-17 设置面板大小后

9.8 小结

本章首先介绍了移动的文字和图片，属性值为 marquee，其语法中又对各属性值进行了说明，在实例中演示了其用法。然后介绍了网页中的多媒体，以前学习过的标记是插入图片，显示出图片的效果；还有另一个标记可以插入 flash、mp3、mid 和一些视频文件，可以直接播放相应的文件内容。

9.9 习题

一、填空题

1. 移动属性是双标记的，其语法用_____标记对表示，里面放的内容就是要移动的对象。
2. 移动的方向用参数_____表示，这个参数的取值有四个：_____、_____、_____和_____。
3. 当标记 embed 设置为_____时，网页完全载入后多媒体不会自动播放。
4. 如果想设置对象移动的方式，例如滚动循环，语法用_____表示，其后可选择参数为：_____、_____和_____。

二、上机实践

1. 用记事本编辑一个网页，要求 IE 标题栏显示“欢迎大家光临！”，网页内容是：“我爱网页制作技术！”“我一定能制作出精美的网页！”让第 1 行文字从左向右移动，第 2 行文字在网页

中来回快速移动。

【提示】

marquee-----对象移动标记
direction-----对象移动方向
behavior-----对象移动方式
alternate-----对象来回移动

2. 在第 1 题的基础上, 给第 1 行滚动的文字加上黄色背景, 给第 2 行滚动的文字加上红色前景。

bgcolor-----背景设置参数
font color-----字体设置参数

3. 在网页中引入一段音乐, 循环播放, 媒体面板宽为 200 像素, 高为 200 像素。

【提示】

embed-----媒体标记
width-----宽度标记
height-----高度标记
loop-----循环标记

4. 在网页中引入一段音乐, 循环播放, 媒体面板宽为 200 像素, 高为 200 像素, 居右对齐, 不隐藏媒体面板, 不自动播放。

【提示】

autostart-----是否自动播放标记
hidden-----是否隐藏标记

第二篇

DIV + CSS布局

第 10 章 用 DIV 布局

DIV 是 HTML 中的标记，也称作层，用 DIV 布局也可说成用层布局。DIV 标记对是用来布局的，结合层叠样式表设计出完美的网页，这样做比以前的 TABLE 布局有更多好处，而且更接近标准。

DIV 标记主要用来布局，在布局中，可以作为容器使用，里面可以包含其他对象，如表格、表单、文本和图像等。DIV 标记中还可以嵌套 DIV 标记，而且层数没有限制，每个 DIV 标记与最近的 DIV 结束标记符（</div>）相配对，且每个 DIV 必须有一个 DIV 结束符与之对应。DIV 和 SPAN 元素最大的特点是默认的都没有对元素内的对象进行任何格式化渲染，主要用于应用样式表。两者最明显的区别在于 DIV 是块元素，而 SPAN 是行内元素。

学习本章，将会了解以下内容。

- DIV 与 TABLE 布局的不同；
- 如何用 DIV 布局；
- 使用 DIV 的注意事项；
- DIV 布局实例；
- DIV 与 SPAN 的区别；
- DIV 与 SPAN 在布局中的应用。

10.1 DIV 与 TABLE 布局比较

大部分网站都往 W3C 标准靠近，对于初学者，先学好基础知识，然后再结合 W3C 标准，使网页质量更高。本节除了介绍基本的 DIV 外，还比较了 DIV 与 TABLE 的区别。

10.1.1 DIV 简介

DIV 是标记对，表示一个层，里面包含 HTML 代码，也可以被 HTML 标记包围。DIV 的属性很多，都是用来控制 DIV 的。下面代码表示就是一个 DIV。

```
<div>  
  <p>我是一个DIV! </p>  
</div>
```

以上代码在 DIV 标记对中包含了一个段落，这时的效果和在与<body></body>中输入一段落没什么不一样，因为对<div>没有设置，稍后会讲到如何设置 DIV。DIV 标记对可以嵌套，嵌套的深度没有限制，但必须每个<div>都能对应一个</div>，而且<div>与最近的一个</div>相匹配，有人把外围的 DIV 称为 DIV 容器，包含很多层。代码 10-1 表示层与层之间的嵌套。

代码 10-1 层嵌套

```
01 <div>
02     <div>这是第一子层</div>
03     <div>这是第二子层，我们都被第一个DIV包含着</div>
04 </div>
```

【代码解析】代码 10-1 中有 3 个层，第一个是层容器，包含着两个子层，默认(layer)占用的宽度是 100%，所以两个子层上的内容占两行。效果如图 10-1 所示。

10.1.2 DIV 与 TABLE 的区别

DIV 层可以用来布局，使所表现的网页更清楚、细腻，而且 HTML 代码少，所以越来越多的网页制作者采用 DIV 布局取代 TABLE 布局。TABLE 表现表格式布局更方便，很多初学者习惯用 TABLE，而对学习 DIV 布局难以入手。其实这是一种误解，在对 DIV 布局的学习时，控制好 DIV 层就容易学好了。记住一点，有时利用 DIV 是很方便的，但也有用到 TABLE 的时候，请读者注意。

在用 TABLE 布局时，可以再嵌套 TABLE，这在前面学习过。TABLE 可以表示表头、表元等，每一行要有<tr></tr>标记对，然后再放入<td></td>标记对，所以布局下来标记对非常多，HTML 代码所占用的空间也比 DIV 多。代码 10-2 表示三行三列的布局模式，没有用到 CSS 来控制这些 DIV。

代码 10-2 TABLE 布局

```
-----文件名：TABLE布局.html-----
01 <html>
02 <head>
03     <title>TABLE布局</title>
04 </head>
05 <body>
06 <!--表格布局-->
07 <table border=1 width=100%>
08     <tr height=100 bgcolor=blue>
09         <td colspan="3">网页头部</td>
10     </tr>
11     <tr height=250>
12         <td bgcolor=orange>网页体部左边</td>
13         <td bgcolor=yellow>网页体部中间</td>
14         <td bgcolor=green>网页体部右边</td>
15     </tr>
16     <tr height=80 bgcolor=gray>
17         <td colspan="3">网页底部</td>
18     </tr>
19 </table>
20 </body>
21 </html>
```

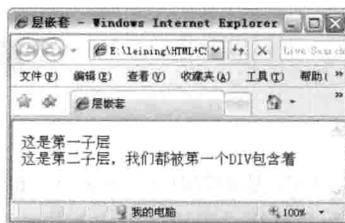


图 10-1 层嵌套

【代码解析】 在第 7~19 行代码，用 table 标记对网页的各部分进行布局，<tr>表示行，<td>表面数据单元，参数的设置都在这些标记中，如设置高度、背景颜色等，通过不同背景颜色体现了网页各个部分。效果如图 10-2 所示。

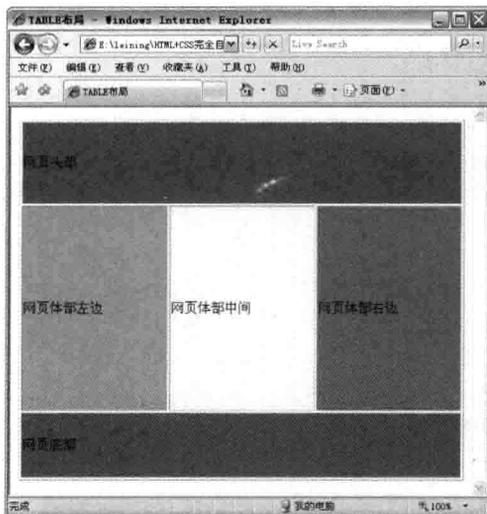


图 10-2 TABLE 布局演示

下面看 DIV 布局显示相同的效果。上面的效果先用 HTML 标记设置好，然后使用 CSS 代码控制它。代码 10-3 全是用 DIV 加上样式属性布局。

代码 10-3 DIV 布局

```
-----文件名: DIV布局.html-----
01 <html>
02 <head>
03 <title>DIV布局</title>
04 </head>
05 <body>
06 <!--层对象的设置-->
07 <div style="background:blue;height:100px;">头部</div>
08 <div id="content">
09 <div style="width:33%;height:250px;float:right;background:green;">网页体右边
10 </div>
11 <div style="width:33%;height:250px;float:left;background:orange;">网页体左边
12 </div>
13 <div style="width:33%;height:250px;float:left;background:yellow;">网页中间
14 </div>
15 </div>
16 <div style="height:80px;background:gray;">网页底部</div>
17 </body>
18 </html>
```

【代码解析】 第 7~16 行代码用 DIV 表示，有的 DIV 被样式属性修饰，如高度、背景等。上述需编写的代码比较大，有一种更好的方法，把重复的样式表示成样式文件，再链接进来（本节中提到样式链接以后会学到）。

代码 10-4 提取样式到一个样式文件中的代码。

代码 10-4 用 DIV 布局

-----文件名: DIV布局样式.html-----

```
01 <html>
02 <head>
03 <title>DIV布局</title>
04 <style type="text/css">
05 /*设置层的样式*/
06 #header
07 {
08     background:blue;
09     height:100px;
10 }
11 #content{}
12 #right
13 {
14     width:33%;
15     height:250px;
16     float:right;
17     background:green;
18 }
19 #left
20 {
21     width:33%;
22     height:250px;
23     float:left;
24     background:orange;
25 }
26 #middle
27 {
28     width:33%;
29     height:250px;
30     float:left;
31     background:yellow;
32 }
33 #bottom
34 {
35     height:80px;
36     background:gray;
37     clear:both;
38 }
39 </style>
40 </head>
41 <body>
42 <div id="header">头部</div>
43 <div id="content">
44     <div id="right">网页体右边</div>
45     <div id="left">网页体左边</div>
46     <div id="middle">网页中间</div>
47 </div>
48 <div id="bottom">网页底部</div>
49 </body>
50 </html>
```

【代码解析】代码 10-4 从代码 10-3 中的<div>中提取了 style 的内容，用<style></style>标记表示样式文件，对应的<div>中被一个 ID 取代，代码 10-3 中的第 42~48 行 DIV 中的 style 内容被放入到对应的 ID 里面，然后在<div id=#></div>链接即可，代码<body></body>上的内容减少到如下代码。

```
<div id="header">头部</div>
<div id="content">
  <div id="right">网页体右边</div>
  <div id="left">网页体左边</div>
  <div id="middle">网页中间</div>
</div>
<div id="bottom">网页底部</div>
```

第一个层是头部部分，用一个层表示；第二个层中包含了三个子层，即左边、中间和右边；最后一个层表示底部的层，内容非常少，效果由样式文件控制着，非常方便。

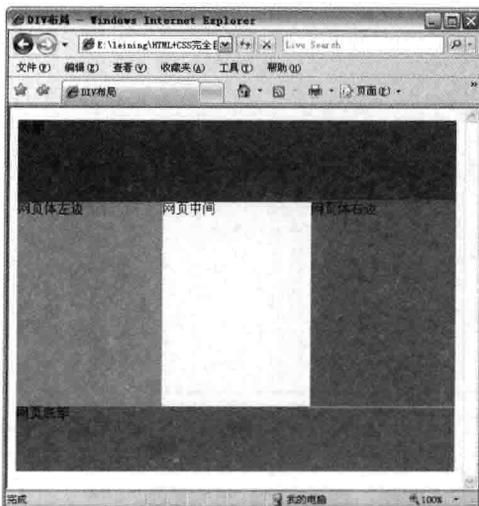


图 10-3 DIV 布局效果

注意

一个网页样式文件不多，样式内容可以重用，内容体上代码量比TABLE少得多，也是DIV布局成为标准的重要因素。

10.2 如何用 DIV 布局

通过前一节的学习，大体知道了用 DIV 布局的方式，现在再总结 DIV 布局是如何进行的。DIV 布局先利用层把内容框架打好，再利用样式表控制。下面通过实例讲解如何进行 DIV 布局，实例所完成的是门户网站的内容浏览新闻页的布局。

10.2.1 设置 DIV 选择符

首先在<body></body>中定好网站的 DIV，对页面的头部、导航条、内容等各个模块用 DIV 列出并对选择符命名，并在样式表中列出，完成后如代码 10-5 所示。

代码 10-5 如何用 DIV 布局

```

-----文件名：如何用DIV布局.html-----
01 <html>-
02 <head>
03 <title>如何用DIV布局</title>
04 <style type="text/css">
05 /*设置CSS选择符*/
06 #header{}
07 #logo{}
08 .ad468{}
09 #banner{}
10 #content{}
11 #newsad{}
12 #news{}
13 #bottom{}
14 </style>
15 </head>
16 <body>
17 <div id="header">
18     <div id="logo">logo--网站图标</div><div class="ad468">468像素广告</div>
19 </div>
20 <div id="banner">导航条</div>
21 <div id="content">
22     <div id="newsad">新闻右边的广告</div>
23     <div id="news">新闻内容</div>
24 </div>
25 <div id="bottom">网页底部</div>
26 </body>
27 </html>

```

【代码解析】在 HTML 中的 DIV 标记中设置了 ID 选择符和类选择符，在 CSS 样式中，以“#”开头的选择符为 ID 选择符，与 DIV 中的 ID 设置关联，同理，以“.”开头的为类选择符，与 DIV 中的参数 class 设置关联。在上述代码中，设置的选择符都为空，即在大括号中没有 CSS 代码，所表现的效果如图 10-4 所示。



图 10-4 没有 CSS 代码时的效果

10.2.2 设置 CSS 修饰 DIV 效果

用样式表修饰内容。样式表就是用 CSS 来控制 HTML 标记，使标记达到预定的效果。代码 10-6 表示在 CSS 代码中，添加 CSS 样式来控制选择符关系的 DIV 标记，层表现出的效果即为 CSS 的功能。

代码 10-6 如何用 DIV 布局

```

-----文件名：如何用DIV布局.html-----
01 <html>
02 <head>
03 <title>如何用DIV布局</title>
04 <style type="text/css">
05 html,body{margin:0px;padding:0px;}

```

```
06 #header
07 {
08     width:778px;
09     margin:auto;
10     border:1px solid #bbbbbb;
11 }
12 #logo
13 {
14     width:180px;
15     height:100px;
16     float:left;
17     border:1px solid #ccc;
18 }
19 .ad468
20 {
21     width:468px;
22     height:60px;
23     float:left;
24     margin:20px 0px 0px 100px;
25     border:1px solid #ccc;
26 }
27 #banner
28 {
29     width:778px;
30     height:30px;
31     margin:5px 0px;
32     text-align:center;
33     border:1px solid #ccc;
34 }
35 #content
36 {
37     width:778px;
38     margin:auto;
39     border:1px solid #ccc;
40 }
41 #newsad
42 {
43     width:260px;
44     height:260px;
45     margin:5px;
46     float:right;
47     border:1px solid #ccc;
48 }
49 #news
50 {
51     width:480px;
52     height:260px;
53     margin:5px;
54     border:1px solid #ccc;
55 }
56 #bottom
57 {
58     width:778px;
```

```

59     margin:auto;
60     height:80px;
61     margin-top:10px;
62     border:1px solid #ccc;
63 }
64 </style>
65 </head>
66 <body>
67 <div id="header">
68     <div id="logo">logo--网站图标</div><div class="ad468">468像素广告</div>
69 </div>
70 <div id="banner">导航条</div>
71 <div id="content">
72     <div id="newsad">新闻右边的广告</div>
73     <div id="news">新闻内容</div>
74 </div>
75 <div id="bottom">网页底部</div>
76 </body>
77 </html>

```

【代码解析】本代码用样式控制了层的外观，第 35~40 行代码的内容层用 1 像素的边框表示，显示出大小，以方便比较。对于固定宽度的层，可以用 float 进行漂移定位，如向右漂移就设置成 float:right。两个层之间的空格用 margin 表示，表示层的外围距离，同理，层的内部填充用 padding 表示。这些常用的参数会在以后的章节中讲解到。加上边框代表层区域的效果如图 10-5 所示。

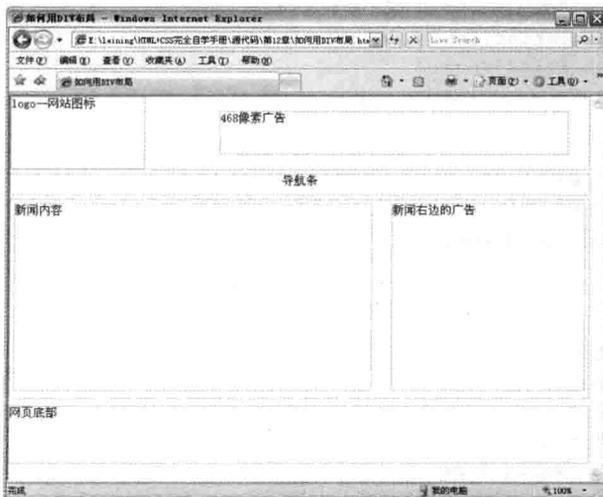


图 10-5 用样式修饰内容

每个区域用边框表示显得更直观，下面用效果区分各区域。

10.2.3 设置 DIV 的背景颜色

本实例只表示布局，里面内容如何设置在以后会讲解到，对样式表代码不了解也没有关系，先对样式表有个初步的认识，在脑海形成有对布局的意识。本步骤对四个大层设置了背景颜色，即对 header、banner、content、bottom 四层的 background 设置不同的颜色，颜色值表示方法在前面学过，

可以用字母表示，也可以用 RGB 形式表示。在 RGB 形式中注意一点：如代码中所看到的 #ccc 与 #cccccc 颜色值为三位时表示这三个值都有重叠的值，第一个 c 代表 cc 值，同理，第二个 c 也代表 cc 值，所以三个值刚好代表 6 个 c 的值，再如，三个不同的值也是重叠的，如 #cba 表示 #ccbbaa，效果如图 10-6 所示。

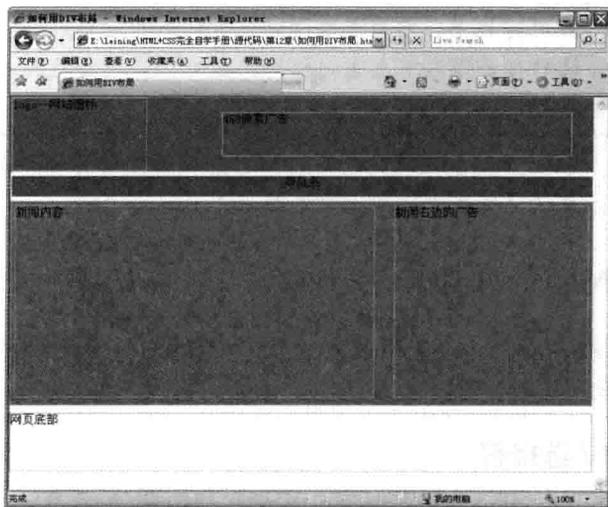


图 10-6 颜色区分不同区域

技巧

网站的网页布局可以选择DIV+CSS的方式，方法如上面所述。CSS文本可以单独保存成以CSS为扩展名的文件，且在网页是链接进来即可。

10.3 使用 DIV 注意事项

在使用 DIV 时，要注意 ID 与类 (class) 选择符的应用。ID 是在一张网页中单独的 HTML 关联 CSS 的名称，而类可以在 HTML 中重复出现，可以定义一个名称在同一网页多处应用（当然也可以在多个网页中应用）。

用 DIV 布局时，好多初学者都不太熟悉 DIV+CSS 布局的使用，而在控制标记时存在比较难以控制的情况。出现这种情况应注意以下几点。

(1) 使用 DIV 时要确定好层对象的宽度。很多初学者想把层控制到一个方面，但层默认的是 100%，在不确定宽度情况下不能移动。

(2) 当确定宽度后，使用属性 float 进行漂移，如在一个位置要想两个层并排，可以设置两层向一个方向漂移（如设置 float 为 left），还有一种方法设置一个向左一个向右，但是中间有间隔，合理地调节这个间隔就可以了。

(3) 当用 float 设置过后的层，在接下来用到的层受空间漂移后的影响，也会出现漂移情况，但这时的漂移并不是当初想要的，所以还需要用到属性 clear 来设置对象不往某方向漂移。当然，clear 主要针对 float 属性应用。

(4) 在设置高度时，如果对层对象设置了高度，而显示内容超过其区域，在 IE 中可以自动延

长对象的区域，但在 FF 中，会出现固定的区域（设置边框可以看到层对象的边框），其他的内容超过对象还会显示，要解决这个问题可以去掉高度属性的（height）设置。

(5) 在层内部要设置绝对定位时，需要在父对象中先设置相对定位，否则绝对定位是以 <body></body> 父对象进行定位。

说明 在使用 HTML+CSS 的过程中，需要长期总结，有的问题在现在遇到了，但在下次还可能会遇到，一个合格的设计师在下次遇到相同问题时，会很好地解决。因为在前面看到过的问题，印象会非常深刻。

用 CSS 和 DIV，除了注意上面的事项外，还有很多地方是读者需要注意的，这些问题都从实际项目中总结，所以对 CSS 的属性要多练习，多实现才能很好地理解。

10.4 认识 SPAN

与 DIV 标记块级元素不同，SPAN 标记指行内元素，主要用于文本的容器。相对 DIV 容器，SPAN 容器相对于文本或图片要小些。

10.4.1 设置 DIV 选择符

与 DIV 一样，SPAN 也是一个标记，包含着一个容器，里面放置的可以是文本或图片等。SPAN 的宽度依照包围着对象的宽度而定，故不能用宽度属性 width 设置 SPAN 标记内对象的宽度，但可以设置 SPAN 的 margin（外边距）值，设置 SPAN 中的对象位置与父对象的距离。

SPAN 的语法与 DIV 一样，其参数可以接 ID 和类（class）等。代码 10-7 表示 SPAN 的用法。

代码 10-7 span 基本语法 1

```
-----文件名：span基本语法1.html-----
01 <html>
02 <head>
03 <title>span基本语法</title>
04 </head>
05 <body>
06 <!--在body标记对里加入span标记-->
07 <span>
08     SPAN基本语法是文本的容器。
09 </span>
10 </body>
11 </html>
```

【代码解析】第 7~9 行代码用到 SPAN 标记。因 SPAN 标记对对象无宽度设置，且默认值与没有 SPAN 标记对是一样的，所以看起来的效果好像没有任何变化，好像只输入了一行文字，效果如图 10-7 所示。

10.4.2 何时用 SPAN

上一节讲解了 SPAN 的语法，也初步了解了 SPAN



图 10-7 SPAN 的基本语法

的用法，那么什么时候才可以用 SPAN 标记呢？用 SPAN 标记时可以用 DIV 取代吗？现在针对这些问题进行探讨并做出解答。

从前面的讲解中可以知道，一个 DIV 内容不超过一行，所占的就是一行文本内容，因为 DIV 默认宽度是父对象的 100%，而 SPAN 标记中的内容宽度是指其内容所占的宽度，即不能设置 SPAN 的宽度，所以在一行文本中，想对其中的文字设置样式，而又不使该行文本换行的条件下，可以选择 SPAN 标记，同理，在图片中也如此应用。

代码 10-8 中显示的是同一行文字在不同文字中需要不同样式，而且又不影响该行换行的效果。这时用 SPAN 来做文字的容器，然后在 SPAN 中设置 ID，在样式表中设置不同 ID 中不同的字体颜色和字体大小。这样，在一行文本显示出不同的样式，如代码 10-8 所示。

代码 10-8 何时该用 SPAN

```
-----文件名：何时该用SPAN.html-----
01 <html>
02 <head>
03 <title>何时该用SPAN</title>
04 <style type="text/css">
05 #span1
06 {
07     /*设置颜色值*/
08     color:blue;
09 }
10 #span2
11 {
12     /*设置字体大小和字体重量*/
13     font-size:15px;
14     font-weight:600;
15 }
16 </style>
17 </head>
18 <body>
19 <span id ="span1">HTML+CSS</span>完全<span id ="span2">自学</span>手册
20 </body>
21 </html>
```

【代码解析】代码第 19 行用到了 SPAN 做容器，其中 HTML+CSS 用 SPAN 标记设置字体颜色为蓝色，而对“自学”二字设置粗体，且大小是 15 像素。范例在同一行文本中，设置了不同的样式，实现了不换行的情况下，一行文本不同的部分可以设置不同的样式。在实际应用中，对段落的同一行设置不同的样式，还保证了段落中的不换行。效果如图 10-8 所示。

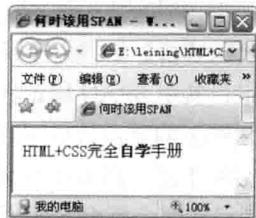


图 10-8 何时该用 SPAN

10.5 DIV 与 SPAN 的区别

DIV 与 SPAN 都可以用来布局，而且在布局用 SPAN 时，DIV 难以取代。相反，用 DIV 的时候，SPAN 也不能取代之。那么这两者的区别在哪里呢？怎么样使用才可以给布局带来方便呢？下

面讲解 DIV 标记与 SPAN 标记的区别。注意，两者的相同点都是 HTML 标记。

虽然 DIV 与 SPAN 用法类似，但还是有区别的，在实际应用中应当注意。下面来比较 DIV 与 SPAN 的区别。

10.5.1 默认宽度不同

SPAN 和 DIV 的区别在于，DIV 是一个块级元素，可以包含段落、标题、表格，乃至章节、摘要和备注等；而 SPAN 是行内元素，SPAN 的前后是不会换行的，它没有结构的意义，纯粹是应用样式，当其他行内元素都不合适时，可以使用 SPAN。

代码 10-9 DIV 与 SPAN 区别一

```
-----文件名: DIV与SPAN区别一.html-----
01 <html>
02 <head>
03 <title>DIV与SPAN区别一</title>
04 </head>
05 <body>
06 /*3个span和3个div*/
07 <span>
08     <!--这里是SPAN容器内容-->
09     这是SPAN容器一
10 </span>
11 <span>
12     这是SPAN容器二
13 </span>
14 <div>
15     <!--这里是DIV容器内容-->
16     这是DIV容器一
17 </div>
18 <div>
19     这是DIV容器二
20 </div>
21 </body>
22 </html>
```

【代码解析】第 7~20 行设置了两个 SPAN 标记和 DIV 标记，其中 SPAN 标记在前面，而 DIV 标记在后面。由 SPAN 标记的显示效果可以知道，SPAN 对象的宽度为文本的实际宽度，而 DIV 中的宽度为默认的 100%，所以 SPAN 标记在同一行中显示两个 SPAN 标记，而两个 DIV 在两行中显示。效果如图 10-9 所示。



图 10-9 DIV 与 SPAN 区别一

10.5.2 块元素与行内元素

块元素和行内元素也不是一成不变的，通过定义 CSS 的 display 属性值可以互相转化。代码 10-10 表示了 DIV 与 SPAN 的区别，即设置块元素与行内元素时，用 display 属性可以对其转化。

代码 10-10 DIV 与 SPAN 区别二

```

-----文件名: DIV与SPAN区别二.html-----
01 <html>
02 <head>
03 <title>DIV与SPAN区别二</title>
04 <style type="text/css">
05 #div1
06 {
07     /*设置内联对象*/
08     display:inherit;
09 }
10 #span1
11 {
12     /*设置块对象*/
13     display:block;
14 }
15 </style>
16 </head>
17 <body>
18 1.这在容器外
19 <div id="div1">
20 这里会另起一行。样式设置display:inherit
21 </div>
22 <span id="span1">这里也会另起一行。样式设置display:block</span>
23 </body>
24 </html>

```

【代码解析】第 19~21 行用 DIV 设置了样式 display 属性值为 inherit，而在代码第 22 行的 SPAN 标记中，其 CSS 属性 display 被设置为 block。在实际效果中，块元素与行内元素之间的变化如图 10-10 所示。

注意 如果不对 DIV 元素设置任何 CSS 属性，那么显示的效果将与 P 元素的效果一样。这也就是本实例的含义。



图 10-10 DIV 与 SPAN 区别二

10.6 DIV 与 SPAN 实例——多层布局

在本节中，用实例来演示 DIV 与 SPAN 的用法，以及它们之间的区别。通过实例讲解来介绍多层布局。所谓多层布局，就是多个层用 DIV 和 SPAN 来设置，然后通过 CSS 代码控制效果。因 DIV 与 SPAN 的应用也有不同，所以在本实例中的多层中，对 DIV 与 SPAN 的应用环境也做了演示，若以后当这种环境出现时，不能用 DIV 来做容器的，可能用 SPAN 会更简单。

本节布局实例没有步骤，主要说明从 CSS 代码做出注解，请读者认真看 CSS 代码对标记的作用。详细的代码请参考代码 10-11 所示。

代码 10-11 DIV 与 SPAN 实例手把手—多层布局

```
-----文件名：DIV与SPAN实例手把手—多层布局.html-----
01 <html>
02 <head>
03 <title> DIV与SPAN实例手把手—多层布局</title>
04 <style type="text/css">
05 /*SPAN标记对应的CSS样式关联*/
06 #span1
07 {
08     /*高为20像素*/
09     height:20px;
10     /*边框为1像素红色实线*/
11     border:1px solid red;
12 }
13 #span2
14 {
15     /*高为30像素,边框颜色为蓝色*/
16     height:30px;
17     border:1px solid blue;
18 }
19 #span3
20 {
21     /*高为40像素,边框颜色为橙色*/
22     height:40px;
23     border:1px solid orange;
24 }
25 #div1
26 {
27     /*DIV关联,20像素红色边框*/
28     height:20px;
29     border:1px solid red;
30 }
31 #div2
32 {
33     height:30px;
34     border:1px solid blue;
35 }
36 #div3
37 {
38     height:40px;
39     border:1px solid orange;
40 }
41 </style>
42 </head>
43 <body>
44 <!--3个SPAN标记-->
45 <span id="span1">
46     这是SPAN容器一,高20像素
47 </span>
48 <span id="span2">
49     这是SPAN容器二,高30像素
50 </span>
```

```

51 <span id="span3">
52     这是SPAN容器二,高40像素
53 </span>
54 <!--3个DIV标记-->
55 <div id="div1">
56     这是DIV容器一,高20像素
57 </div>
58 <div id="div2">
59     这是DIV容器二,高30像素
60 </div>
61 <div id="div3">
62     这是DIV容器三,高40像素
63 </div>
64 </body>
65 </html>

```

【代码解析】第 45~63 行用 3 个 SPAN 标记和两个层进行比较。每个对象都有边框，SPAN 的对象按其内容的宽度来显示的，代码中设置 3 个 SPAN 标记的边框用不同颜色表示，且高度也不一样（SPAN 可以设置高度，但不能设置宽度），出现在同一行中。而 DIV 设置其高度不同，边框颜色也不一样。但两个 DIV 层每个层中的内容只有 10 个字，却占了一整行，两个 DIV 占用了两个不同高度的行。效果如图 10-11 所示。

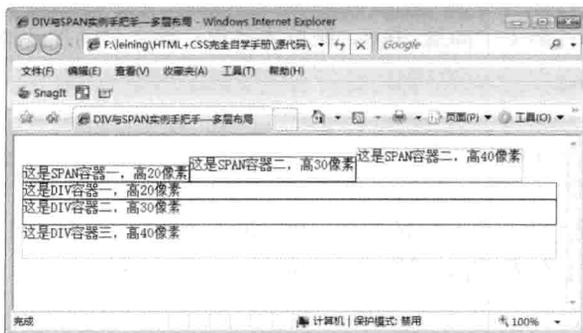


图 10-11 DIV 与 SPAN 实例手把手—多层布局

注意

因为DIV与SPAN元素的特殊性，所以一般用于应用样式表，比如用CSS定义为层，而需要分清的是块元素和行内元素的区别，还有两者间的相互转化。

10.7 小结

本章从为什么会选择 DIV 来布局和比较 DIV+CSS 与 TABLE 布局的方式进行同一效果对比，更突出 DIV 布局的优势；在 DIV+CSS 布局中，看到了好多不认识的属性和样式的表现形式，这给读者一个对布局的初步认识，以后会继续学习这方面的知识；最后还介绍了 SPAN 的语法和其用法，对何时使用 SPAN 作了详细讲解。SPAN 用法与 DIV 类似，但也有区别，往往在实际应用中 DIV 和 SPAN 会一起用，而且都可以用 CSS 对它们进行控制。

10.8 习题

一、填空题

1. 在 CSS 样式中, 以“#”开头的选择符为_____, 与 DIV 中的 ID 设置关联, 同理, 以“.”开头的为_____, 与 DIV 中的参数 class 设置关联。
2. 当用 float 设置过后的层, 在接下来用到的层受空间漂移后的影响, 也会出现漂移情况, 但这时的漂移并不是当初想要的, 所以还需要用到属性_____来设置对象不往某方向漂移。

二、上机实践

1. 做一个 DIV 层, 要求: 居中、高度为 100px、底色为红色。

【提示】

```
div-----层标记
height-----高度
float-----分布标记
background-----背景标记
```

2. 在第 1 题的基础上, 在层里体现“我爱网页制作”字样, 同时设置文字居中。

【提示】

```
text-align:center-----文字居中
```

3. 设置 3 个 SPAN: 第 1 个高度 20px, 颜色为蓝色; 第 2 个高度 30px, 颜色为红色; 第 3 个高度 40px, 颜色为绿色。

【提示】

```
height-----高度
color-----颜色
```

4. 在第 1 题的基础上为 3 个 SPAN 增加边框: 第 1 个为红色, 第 2 个为蓝色, 第 3 个为橙色; 同时改变字体大小, 第 1 个为 20px, 第 2 个 30px, 第 3 个为 40px。

【提示】

```
border-----边框
font-size-----字体大小
```

第 11 章 DIV 与 CSS 结合

本章介绍 DIV+CSS 布局方法, DIV 只不过是个层而已, 并不会给网页添加什么。CSS (Cascading Style Sheet) 是控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。DIV 通过与 CSS 的结合, 使 CSS 能够控制 DIV 的外观、大小等。在网页中通过外接方式链接 CSS 文件, 也可以通过内接方式写 CSS 文件, 还可以在<div>中加入 style 方式结合样式文件。

学习本章, 将会了解以下内容。

- 用 CSS 控制 DIV;
- CSS 文件链接;
- 内联样式;
- 内部样式表示;
- 链接外部样式;
- DIV+CSS 实例。

11.1 用 CSS 控制 DIV

HTML 中的<div></div>标记对是放在<body></body>中的标记对, 每个<div></div>标记对代表一个层, <div>里面有很多属性, 有 id、class、align 和 style 等, 对层进一步描述, 这几个属性都比较常用。

用 CSS 来控制 DIV, 即用样式表来控制 DIV, 在前面章节提到过, 最基本的是在<div>标记里用 style 表示, 用来描述里面的样式, 语法形式如下。

```
<div style=#></div>
```

“#”代表样式的内容, 每个样式都以名值对(一个属性名称对应一个值, 在 CSS 代码中都是这种形式)出现, 中间用分号间隔开来。语法形式如下。

```
<div style="width:760px;height:200px;">这表示一个层</div>
```

以上代码表示一个层, 层中被样式表控制着, 样式表内容是宽度为 760 像素(px), 高度为 200 像素, 其中宽度和高度都是名称, 数值表示数据再加单位, 分号代表名值对结束, 所以一个名值对表示样式表的一个动作或效果。代码 11-1 表示了设置一个层的边框、背景颜色和高度。

代码 11-1 用 CSS 控制 DIV

```
-----文件名: 用CSS控制DIV.html-----
01 <html>
02 <head>
03     <title>CSS控制DIV</title>
04 </head>
```

```

05 <body>
06 <!--设置内联样式-->
07 <div style="background:blue;border:1px solid #cccccc;height:100px;">这是一个
08 层! </div>
09 </body>
10 </html>

```

【代码解析】在代码第 7 行的<div>中有样式属性，即用 style 表示，双引号里面是名值对形式的样式表内容，background 表示背景，为蓝色，即 blue（也可以用 #0000ff，前面提到过颜色值的两种表示方法）。效果如图 11-1 所示。

在<div>的 style 中，可以包含很多样式名值对，样式内容越多，所对层的控制越多，那么写起来的长度也越长，为了解决这个问题，可以把所有的样式表内容写到一个文件中，详情请看 11.2 节。

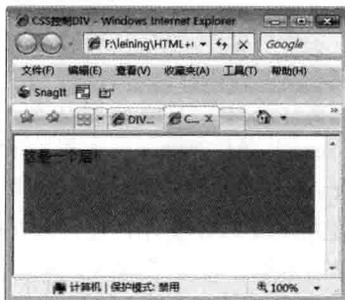


图 11-1 用 CSS 控制 DIV

注意 style 中的样式中有分号，这个分号是不能省略的，代表本语句（名值对）结束。

11.2 CSS 文件链接

CSS 文件的存在使网页节省很多空间，如 11.1 节中的代码，每个<div>都由一个 style 控制，如果有很多层的话，就要写很多的样式内容了，这是不现实的。所以有两种方式可以把 CSS 文件专写在一起，那就是 CSS 的外链接和内链接。

CSS 文本的链接方式有三种，分别为内联定义、链入内部 CSS 和链接外部 CSS，下面对三种不同的链接形式进行讲解。

11.2.1 内联定义

内联定义即在对象的标记内，使用对象的 style 属性定义适用的样式表属性。用这种方式主要是对特定的层或标记设置样式，如设置一个层的边框，即只代表这个层的样式，对其他层或标记无效。

内联定义的使用有一个好处就是可以灵活地设置对象的样式，缺点是样式扩展性差，即不能让其他的对象享用它的样式。虽然其他的样式链接可以取代内联定义，但内联定义还是对某些方面是有用的，如下面代码 11-2 所示。

代码 11-2 内联定义样式

```

-----文件名：内联定义样式.html-----
01 <html>
02 <head>
03 <title>内联定义样式</title>
04 </head>
05 <body>
06 <!--在标记对<p>中设置内联样式-->
07 <p style="border:2px solid #000000">HTML+CSS完全自学手册</p>

```

```

08      <!--设置本对象字体颜色为红色-->
09      <p style="color:red">HTML+CSS完全自学手册</p>
10      <!--设置字体大小为12像素-->
11      <p style="font-size:12px">HTML+CSS完全自学手册</p>
12  </body>
13  </html>

```

【代码解析】在第 6~11 行代码的标记对<p></p>中设置了内联定义，第 7 行代码为第 1 个段落，设置了标记的边框样式，2 像素实线；第 9 行代码为第 2 个段落，将字体设置为红色；第 11 行代码为第 3 个段落，设置字体的大小为 12 像素。效果如图 11-2 所示。

11.2.2 链接内部 CSS

链接内部 CSS 样式比内联样式的修饰的范围要大许多，与内联样式范围在本对象上不一样的是，链接内部样式所在的范围在整个网页中，这个形式在有的大型网站采用比较多，加载整个网页更快。

CSS 内链接是由<style></style>标记对放在<head></head>中，在<style>中有一个类型属性 type，后面接 text/css，表示 CSS 文本。完整语法如下。

```

<style type="text/css">
/*这里写CSS内容*/
</style>

```

在<head></head>中放入<style></style>文件中本网页的样式，如果使每张网页的层都由各网页中的<style></style>修饰。

在<div>中，可以对<div>的 ID 和 CLASS 进行修饰。ID 表示层的名称，CLASS 表示类，ID 在每个网页中只能用一次，而 CLASS 可以重复使用，不限次数。同理，在<div>中的名称或类如何在 CSS 中定义呢？下面来回答这个问题。ID 在 CSS 中用#加 ID 名再接一对大括号组成，大括号里面用来写样式表内容，同理，CSS 中的可以表示 CLASS 的样式，与 ID 不同的是 CLASS 用“.”表示，点后面接名称和大括号，下面是其语法形式。

```

<style type="text/css">
#myid
{
    /*以下是样式代码，被style标记对包围着*/
    width:200px;
    height:300px;
    /*设置字体颜色*/
    color:#cccccc;
}
.myclass
{
    width:200px;
    height:300px;
    /*设置字体颜色*/
    color:#cccccc;
}
</style>

```

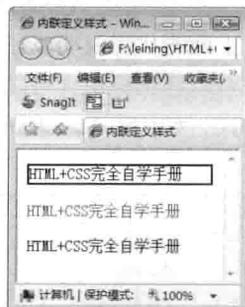


图 11-2 内联定义样式

在<style></style>标记对中,用#myid 表示层中的 ID,用.myclass 表示层中的 class 类,对应的层如下。

```
<!--设置层之间关联-->
<div id="myid">这里是ID,一张网威页面只能出现一次</div>
<div class="myclass">这里是class,可以出现多次。</div>
<div class="myclass">又出现一次。</div>
```

上面的代码说明了在 CSS 控制中,可以通过对层中的 ID 或 CLASS 来控制。用内部链接 CSS 方式可以修饰标记,可以用 CSS 控制标记的效果。下面介绍将内联定义的代码改成链入内部 CSS 的方式,如代码 11-3 所示。

代码 11-3 链入内部 CSS 样式

```
-----文件名:链入内部CSS样式.html-----
01 <html>
02 <head>
03 <title>链入内部CSS样式</title>
04 <style type="text/css">
05 /*内部CSS样式*/
06 #p1
07 {
08     /*设置边框*/
09     border:2px solid #000000;
10 }
11 #p2
12 {
13     /*设置字体颜色*/
14     color:red;
15 }
16 .p3
17 {
18     /*设置字体大小*/
19     font-size:12px;
20 }
21 </style>
22 </head>
23 <body>
24     <!--设置段落的标记-->
25     <p id="p1">HTML+CSS完全自学手册</p>
26     <p id="p2">HTML+CSS完全自学手册</p>
27     <p class="p3">HTML+CSS完全自学手册 </p>
28 </body>
29 </html>
```

【代码解析】在第 23~28 行代码<body></body>标记内的对象(<p></p>),其内用 ID 和 CLASS 关联了样式中的代码,其中 ID 对应样式名称前用“#”表示,CLASS 对应样式名前用“.”表示,效果与图 11-2 所示一样。

注意

网页由<html></html>总标记对组成,里面包含<head></head>和<body></body>,CSS内链接的<style></style>就放在<head></head>里。

11.2.3 链接外部 CSS

CSS 外链接是把 CSS 文件放在网页外面，通过链接使 CSS 文件对本网页的样式有效，这样的链接称为外链接。

由<link>开始，里面的属性有 type 和 rel。其中 type 和上一节学的一样，设置 type 为 text/css，表示 CSS 文本；rel 是样式，即 stylesheet。最重要的是地址，地址用 src 表示。下面代码是一个完整外链接，前提是 CSS 文件和页面是同一目录。

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

通过标记<link>即可把外面的样式文本链接到网页，这也是绝大多数网站所采用的方法，这样更有效率，只完成一张样式表即可控制所有的页面效果。当然，在修改样式表的同时，链接了该样式表的所有网页都会受到影响。这样对整个项目效率会提高，这也是为什么网站都采用这个样式的原因。

下面通过把内联定义的效果，把代码转换为链接外部 CSS 形式。首先用上面的<link>标记把样式文件链接进来，设置 type 为样式文本 (text/css)，别忘记了 rel="stylesheet"，这个参数很关键，<link>标记是放在<head></head>中的，即在头部链接外面的样式文件。代码 11-4 是通过代码 11-2 转换的 HTML 代码。

代码 11-4 链接外部 CSS 样式

```
-----文件名：链接外部CSS样式.html-----
01 <html>
02 <head>
03 <title>链接外部CSS样式</title>
04 <!--引入外面链接-->
05 <link type="text/css" rel="stylesheet" href="style.css" />
06 </head>
07 <body>
08     <p id="p1">HTML+CSS完全自学手册</p>
09     <p id="p2">HTML+CSS完全自学手册</p>
10     <p class="p3">HTML+CSS完全自学手册</p>
11 </body>
12 </html>
```

【代码解析】在第 5 行代码用<link>把同目录的 CSS 文件 (即 style.css 以 css 为扩展名的文件) 链接进来，进一步修饰这个网页。样式代码如代码 11-5 所示。

代码 11-5 style.css

```
-----文件名：style.html-----
01 /*以下是css样式,保存为单独文件*/
02 #p1
03 {
04     border:2px solid #000000;
05 }
06 #p2
07 {
08     color:red;
09 }
10 .p3
```

```

11 {
12     font-size:12px;
13 }

```

【代码解析】代码 11-5 实际就是把内部链接 CSS 样式修饰代码放到了一个文件里。与链入内部样式一样，链接外面样式中代码所表示的效果如图 11-2 所示，即用 3 种不同的形式表示了同一种效果。

注意

当链接外部样式文件时，在<link>标记中，一定不要忘记将rel设置成stylesheet，否则，网页链接失败。

11.3 DIV+CSS 布局实例

接下来讲解真正的设计布局。和一般的方法一样，首先要在脑海中有大致的轮廓构想，可能看到有关 Web 标准的站点大都很朴素，因为 Web 标准更关注结构和内容，实际上它与网页的美观没有根本冲突，想怎么设计就怎么设计，用传统表格方法实现的布局，用 DIV 也可以实现。技术有一个成熟的过程，把 DIV 看成和 TABLE 一样的工具，如何运用就靠发挥想象力了。

注意

在实际应用过程中，DIV在有些地方的确不如表格方便，例如背景色的定义。但任何事情都有得有失，取舍于个人的价值判断。

11.3.1 确定布局

W3C 的最初设计草图采用的表格一般是上中下三行布局，用 DIV+CSS 使用三列来布局，先定义好层和样式名称，部分代码如下。

```

01 <!--设置层布局-->
02 <!--层布局中的头部-->
03 <div id="header"></div>
04 <!--设置容器层-->
05 <div id="mainbox">
06     <!--设置子层-->
07     <div id="menu"></div>
08     <div id="sidebar"></div>
09     <div id="content"></div>
10     <div id="left">页面左列</div>
11     <div id="middle">页面中列</div>
12     <div id="right">页面右列</div>
13 </div>
14 <!--设置底部-->
15 <div id="footer"> </div>

```

【代码解析】第 5~13 行代码只是从<body></body>中定义了其层和样式类型的名称，通过 DIV 层的设置，把页面分成头部 (header) 和体部 (mainbox) 和底部 (footer)。

11.3.2 定义 body 样式

先定义整个页面的 body 的样式，代码如下。

```

01  /*<body></body>标记的样式*/
02  body {
03      /*设置内外边距*/
04      margin: 0px;
05      padding: 0px;
06      /*设置背景颜色*/
07      background:#FEFEFE no-repeat right bottom;
08      /*设置字体类别*/
09      font-family:'Lucida Grande','Lucida Sans Unicode','宋体',arial,verdana,sans-serif;
10      /*设置文本颜色*/
11      color:#666;
12      /*设置字体大小*/
13      font-size:12px;
14      /*设置行高*/
15      line-height:150%;
16  }

```

【代码解析】第 4 行代码定义了边框边距为 0；第 7 行定义背景颜色为 #FEFEFE；第 11 行定义字体颜色为 #666；第 13 行定义了字体尺寸为 12px；第 15 行定义行高 150%。这是对整个 <body></body> 的修饰，同时，还有其他的表示，如 img 下也可以设置图像的样式，又如 p 的样式也可以用 p{} 表示。

11.3.3 定义主要的 DIV

初次使用 CSS 布局，采用固定宽度的三列布局（比自适应分辨率的设计简单），分别定义左、中、右的宽度为 200：300：280，在 CSS 中定义如下。

```

#mainbox{
    /*设置本对象居中*/
    MARGIN:auto;
    BORDER: 0px;
    /*设置背景颜色*/
    BACKGROUND: #fff;
    /*设置宽度*/
    WIDTH: 778px;
}
#left{
    /*下面是设置宽高内外边距*/
    width: 200px;
    height:400px;
    margin: 0px;
    padding: 0px;
    /*设置向左漂移*/
    float:left;
    /*设置背景颜色*/
    background:#CDCDCD;
}
#middle {
    padding:0px;
    float:left;
    width:200px;
    height:400px;
}

```

```

        background: #DADADA;
    }
    #right{
        width:378px;
        float:left;
        height:400px;
        background: #cccccc;
    }
    #header{
        MARGIN: auto;
        BORDER: 0px;
        BACKGROUND: #ccd2de;
        WIDTH: 778px;
        HEIGHT:60px;
        border:1px solid #000000;
    }
    #footer{
        /*清除左右的漂移*/
        CLEAR: both;
        /*设置对象居中*/
        MARGIN: auto;
        /*设置内边距上下为5像素,其他边为0*/
        PADDING: 5px 0px 5px 0px;
        BACKGROUND: #ccd2de;
        WIDTH: 778px;
        HEIGHT:40px;
    }
}

```

注意 定义中列和右列div, 都采用了POSITION: absolute;, 然后分别定义了LEFT:200px;TOP:0px;和LEFT:500px;TOP:0px; 这是这个布局的关键, 采用了层的绝对定位。定义中间列距离页面左边框200px, 距离顶部0px; 定义右列距离页面左边框500px, 距离顶部0px;。

这时整个页面的代码如代码 11-6 所示。

代码 11-6 用 CSS+DIV 布局

```

-----文件名: CSS控制DIV.html-----
01  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
02  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
03  <html>
04  <head>
05  <title>DIV+CSS布局</title>
06  <link type="text/css" rel="stylesheet" href="div.css">
07  </head>
08  <body>
09  <!--头部层-->
10  <div id = "header">
11      头部
12  </div>
13  <!--中间层-->
14  <div id = "mainbox">
15      <div id = "left">页面左列</div>
16      <div id ="middle">页面中列</div>

```

```

17     <div id ="right">页面右列</div>
18 </div>
19 <!--底部层-->
20 <div id = "footer">
21 </div>
22 </body>
23 </html>

```

【代码解析】第 10~21 行代码运行效果仅为 3 个并列的灰色矩形，但在中间部分设置了 3 个层，用 CSS 来布局可以把 HTML 代码调协成不同的效果，这完全由 CSS 控制。想全屏显示，那该怎么办呢？只需设置宽度为 100% 即可。全屏在 DIV 中的效果如图 11-3 所示。

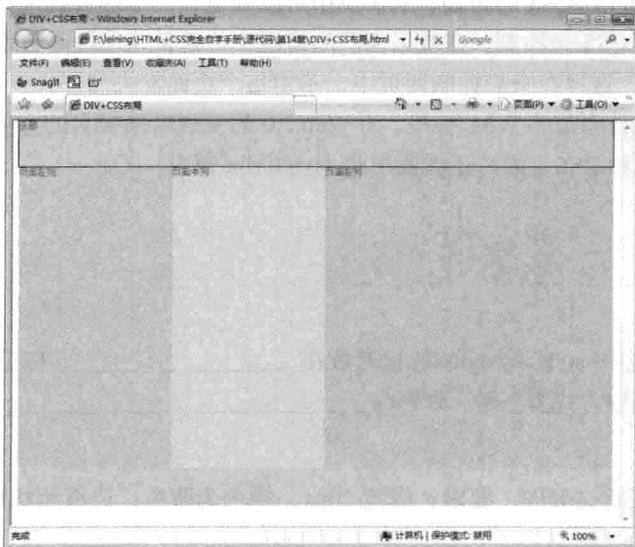


图 11-3 用 DIV+CSS 布局

11.3.4 设置层居中

如果遇到必须对齐三列底部的问题，在 Table 布局中，用大表格嵌套小表格的方法，可以很方便地对齐三列；而用 DIV 布局，三列独立分散，内容高低不同，就很难对齐。其实完全可以嵌套 DIV，把三列放进一个 DIV 中，就做到了底部对齐。

具体样式表都写在相应的版块里了。重点是 #menu、#sidebar、#content 三个层。当 #content 的内容增加，#content 的高度就会增高，同时 #mainbox 的高度也会改变。#footer 层就自动下移。这样就实现了高度的自适应。

注意

#menu 和 #content 都是浮动在页面右面 “FLOAT: right;”，#sidebar 是浮动在 #menu 层的左面 “FLOAT: left”，这是浮动定位，还可以采用绝对定位来实现这样的效果。

这个办法存在另一个问题，就是侧列 #sidebar 的背景无法百分之百显示。一般的解决办法就是用 body 的背景色来填充满。（不能使用 #mainbox 的背景色，因为在 Mozilla 的浏览器中，#mainbox 的背景色失效。）

在代码中，用 margin 可以设置层居中，在保证下面代码存在的条件下，可以用 margin 设置层

居中。

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

除了以上代码外，还要设置 margin 为 auto，表示居中，即左右居中对齐。

注意 上面的DOCTYPE声明对IE和FF的表现不同。IE中需要该DOCTYPE声明，而FF中则相反。

11.4 小结

本章主要介绍了 DIV+CSS 的布局，从实例中可以知道，一个网页效果可以用内联样式、链接内部 CSS 和链接外面 CSS 来控制。在效果相同的情况下，常常为了提高效率，一般采用第三种方式，即链接外部样式，这样在全局控制网页中，不用每个页面都去写样式表。

DIV+CSS 布局也是当前的 W3C 标准，为 Web 2.0 的发展带来很大的方便，如以后用到的 Ajax 技术，就得用到 DIV 和 CSS，同时还需要用到 JavaScript 语言。

11.5 习题

一、填空题

1. CSS 内链接是由<style></style>标记对放在_____标记对中。
2. CSS 文本的链接方式有三种，分别为_____、_____和_____。

二、上机实践

1. 设置一个内联定义 DIV，要求：高度 20px，颜色为蓝色，边框为红色，字体大小为 20px。

【提示】

```
style-----DIV样式属性
```

2. 设置一个内部链接 DIV，要求：高度 20px，颜色为蓝色，边框为红色，字体大小为 20px。

【提示】

```
内部链接DIV在<head></head>中
```

3. 设置一个外部链接 DIV，要求：高度 20px，颜色为蓝色，边框为红色，字体大小为 20px。

【提示】

```
外部链接DIV在<head></head>中
```

第 12 章 CSS 的基础知识

CSS 是 Cascading Style Sheet 的缩写，翻译成中文的意思是层叠样式表单，之前所用到的样式表就是本章要学习的 CSS。本章先介绍其基础知识，通过前面的认识和现在的学习，掌握 CSS 的基本知识。在本章中，只进行初步介绍，更为详细的内容将会在以后的章节中介绍。

学习本章，将会更清楚地了解以下内容。

- CSS 编写环境；
- CSS 文件链接方式；
- CSS 与浏览器的兼容性；
- 选择符；
- 伪元素；
- CSS 常用单位；
- CSS 的实例。

12.1 CSS 简介

对于一个网页设计者来说，对 HTML 语言一定不会感到陌生，因为它是所有网页制作的基础。但是如果希望网页能够美观、大方，并且升级方便、维护轻松，仅仅靠 HTML 是不够的，CSS 在这中间扮演着重要的角色。本章从 CSS 的基本概念出发，介绍 CSS 语言的特点，以及如何在网页中引入 CSS，并对 CSS 进行初步的体验。

CSS 是控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。CSS 通常在网页制作中起重要作用，而且还可以减少网页代码量，更能提高网络访问网页的速度，而且更容易学习和实践，所以 CSS 越来越被更多的网页工作者采用。

CSS 常常与 HTML 标记配合使用，通过链接方式（下一节会讲到）对 HTML 标记控制，可以使网页达到更美观的效果。

12.2 CSS 文件链接方式

CSS 文件链接分成内链接、链接内部 CSS 和链接外部 CSS 三种方式，该知识点在 11.2 节中已

详细详解，本节中不再赘述。

在网站中，网页设计者都先新建一个 CSS 文件，然后在网页中链接此 CSS 文件，当控制 HTML 标记时，都在这个 CSS 文件中写入 CSS 代码，当要修改所有网页的风格时，只要对 CSS 样式文件进行修改即可。每个网站几乎都有 CSS 样式，当然链入多个外部样式，则需要多个<link>。

链接 CSS 样式是 HTML 与 CSS 结合的基础，做好这一步，才可以用 CSS 中的属性控制 HTML 标记，使网页表达出更好的效果。

注意 如何对 CSS 文件链接请查看 11.2 节。

12.3 选择符

CSS 选择符就是 CSS 样式的名字，选择符的命名规则可以使用英文字母的大写与小写、数字、连字号、下划线、冒号、句号。CSS 选择符只能以字母开头，选择符在 CSS 中可分成多种，本节只介绍常用的 CSS 选择符：HTML 标记选择符、ID 选择符和 CLASS 选择符（类选择符）。

HTML 标记选择符是指在 CSS 中，用 HTML 的标记来代表网页中这些标记中的样式集合。用 CSS 控制这些标记，会改变标记的默认样式，如下代码所示。

```
P
{
    /*设置段落的外边距*/
    Margin:0px;
}
```

在 HTML 标记中的<p></p>，表示一个段落，当这个段落被上面的 CSS 样式控制时，默认的段落间隔就没有了，即段落间的间隔为 0（margin 控制的效果）。同理，这些标记除了 p 外，还有 div、td、tr、body 等。

ID 选择符是唯一性选择符，例如#dreamdured{color:red;}，就是在名字前增加了一个#，id 选择符在一个页面中只能出现一次，在整个网站中也最好出现一次（这样有利于程序员控制此元素，有多个一样名称的元素，就无法区别，不好控制了）。

ID 选择符在 HTML 代码中，对 ID 符只引用一次，如果多次引用，只有第一个有效，其他的没有效果。一个外部的样式文件定义一个 ID 样式，可以在多个网页中使用，但不能在同一页面使用两次或两次以上。下面代码表示 ID 选择符的用法。

```
01 #mydiv
02 {
03     width:200px;
04     height:100px;
05     border:2px solid red;
06 }
```

如上所述，ID 选择符以#开头，接上 ID 名，再由 CSS 代码组成，CSS 样式语法是由选择符加属性和属性值构成的，其中属性和属性值由冒号(:)间隔开，多个属性定义用分号(;)分开，使用方便简单。

CLASS 选择符是多重选择符，例如.dreamdubue{color:blue;}，就是在名字前增加了一个“.”，CLASS 选择符在一个页面中可以出现多次（注意下面的示例中 CLASS 选择符的用法）。下面是

CSS 表示 CLASS 选择符语法的完整代码。

```
01 .mydiv
02 {
03     width:200px;
04     height:100px;
05     border:2px solid red;
06 }
```

以上代码中代表名称为 mydiv 的类选择符，在大括号里面包含着属性和属性值的名值对，在多个名值对中用分号分开。即使只有一个属性名值对也建议写好分号，方便日后添加其他名值。

每个选择符都包含大括号，所以有很多选择符时，也同时包含着很多的括号，里面是该选择符的 CSS 代码。

注意

ID 选择符在网页代码中不能重复，而 CLASS 选择符在网页代码中可以重复，而且次数不限。

12.4 伪类

在前面实例中用到选择符如 div, p 等标记类型的样式设置，称为类型选择符。同理，在标记中设置 ID 或 CLASS 称为 ID 选择符和 CLASS 选择符。除了类型选择符、ID 选择符和 CLASS 选择符之外，CSS 也允许使用伪类选择符和伪元素选择符。

伪类选择符是基于一组预定义性质的选择符，HTML 元素可以占有这些预定义性质。实际上这些性质与 class 属性的功能是相同的，因此在 CSS 术语中，它们被称作伪类。在对应这些伪类的标记中，不存在真正的 class 属性，相反，它们代表应用到这些元素的某些方面，或者是相对于该元素的浏览器用户界面的状态。

伪选择符标识虚拟元素，即在标记中不存在，但是浏览器可以推理并应用样式的元素。在伪类中，没有对应伪元素的标记。

以下是四个可以安全使用的用在连接上的伪类。

- link：用在对象处于的连接状态上。
- visited：用在已经访问过的链接上。
- active：用于获得焦点（比如被单击）的链接上。
- hover：用于鼠标光标置于其上的链接。

默认条件时，已链接文本颜色是蓝色的，已访问的链接文本是紫色的。然而，伴随着 CSS 的广泛使用，默认的颜色看起来比较陈旧了，也不再假设链接必须是蓝色或紫色的，第 20 章将详细讲解伪类。

12.5 注解

注解在 CSS 中常用到，用来注释 CSS 代码功能等，使自己或他人以后看到某代码就知道该内容的功能等信息，使程序可读性更强。CSS 注解有两种方式：第一种是单行注解，第二种是多行注解。

单行注解是指在 CSS 代码中，只对单行做注解，浏览器解释到该注解时不会执行，只起标示作用。单行注解语法如下表示。

```
/*这是单行注解*/
```

单行注解是用斜线与星号组成的，其中星号在内，里面所包含的文本是被注解的信息，而多行文本注解与单行文本一样，在 CSS 中，用斜线和星号表示注解的开始，然后是注解的内容，注解结束用星号和斜线表示。在代码编辑器中，注解的文本颜色与代码不同。下面表示多行注解。

```
/*
    这是注解第一行。
    这是注解第二行。
    . . . .
*/
```

多行注解就是用斜线包含其注解内容，分布在多行文本中，需注意的是，这些注解符不能交叉，且不能没有结束符。

12.6 单位

长度的值由可选的正号“+”或负号“-”接一个数字及标明单位的两个字母组成。在一个长度的值中是没有空格的，例如，1.3 em 就不是一个有效的长度的值，但 1.3em 就是有效的。一个为零的长度不需要两个字母的单位声明。

无论是相对值还是绝对值长度，CSS 都支持。相对值单位确定一个相对于另一长度属性的长度，因为它能更好地适应不同的媒体，所以是首选的。以下是有效的相对单位。

- em: em, 元素的字体的高度。
- ex: x-height, 字母“x”的高度。
- px: 像素, 相对于屏幕的分辨率。

绝对长度单位视输出介质而定，所以逊色于相对单位。以下是有效的绝对单位。

- in: 英寸, 1in=2.54cm。
- cm: 厘米, 1cm=10mm。
- mm: 毫米, 1m=1000mm。
- pt: 点, 1pt=1/72in。
- pc: 帕, 1pc=12pt。

一个百分比值由可选的正号“+”或负号“-”接一个数字和百分号“%”组成。在一个百分比值中是没有空格的。

百分比值相对于其他数值，同样可以用于定义每个属性。最经常使用的百分比值是相对于元素的字体大小而言的。

颜色值是一个关键字或一个 RGB 格式的数字。Windows VGA (视频图像阵列) 形成了 16 个关键字: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white 和 yellow。

RGB 颜色可以有如下四种形式。

- #rrggbb, 如#00cc00。

- #rgb, 如#0c0。
 - rgb (x, x, x), x 是一个介于 0~255 的整数, 如 rgb (0, 204, 0)。
 - rgb (y%, y%, y%), y 是一个介于 0.0~100.0 的整数, 如 rgb (0%, 80%, 0%)。
- 上述的例子用不同的方式指定同一颜色。

注意 绝对长度在网页的单位中不常用。

12.7 编写 CSS 的工具

编写 CSS 的环境很多, 可以在文本文件中, 在文本文件编写 CSS 内容是外部链接 CSS 方式, 需要注意的是在保存文件时, 扩展名改成.css, 点前面是 CSS 文件名; 也可以在网页中<html>与<body>之间直接加入<style type="text/css"></style>样式标记对, 里面放内容即可。虽然在文本文件和网页本身都可以编写 CSS 代码, 但是可利用的编写工具很多。下面介绍常用的 Dreamweaver CS5 和开源的 Aptana。

Dreamweaver CS5 是常用的工具, 可以直接在这里面新建 CSS 文件, 然后直接写 CSS 代码即可 (如果在<html>与<body>间插入 CSS 的话还要写上<style type="text/css"></style>)。为了能让初学者更好地理解编写环境, 下面演示新建 Dreamweaver CS5 中的 CSS 文件过程。

12.7.1 打开 CSS 工具软件

安装好程序后, 可以在桌面打开 Dreamweaver CS5 软件, 启动程序后, 将会出现导航界面, 如“打开最近项目”、“创建新项目”等, 还可以看到菜单栏和中间位置的部分信息, 如图 12-1 所示。

12.7.2 新建 CSS 文件

选择菜单栏的“文件”|“新建”命令, 弹出“新建文档”对话框, 在“常规”选项卡中选择“基本页”, 在“基本页”中选择 CSS, 单击“创建”按钮, 即可进入一张 CSS 编辑界面, 如图 12-2 所示。



图 12-1 打开 Dreamweaver CS5 的环境界面时

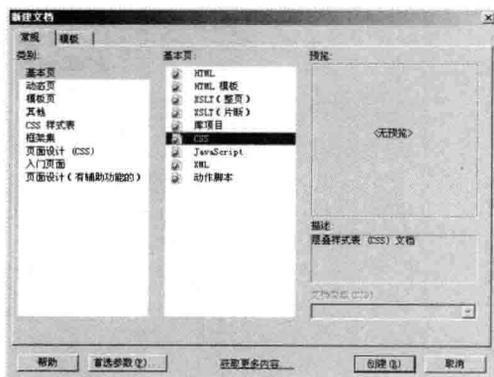


图 12-2 “新建文档”对话框

12.7.3 另存为 CSS 文件

新建好 CSS 文档后,先不急于写代码,在菜单栏中选择“文件”|“保存”按钮,当第一次保存时会出现“另存为”对话框,选择好保存路径和 CSS 文件名,单击“保存”按钮,即可保存为 CSS 文件,如图 12-3 所示。

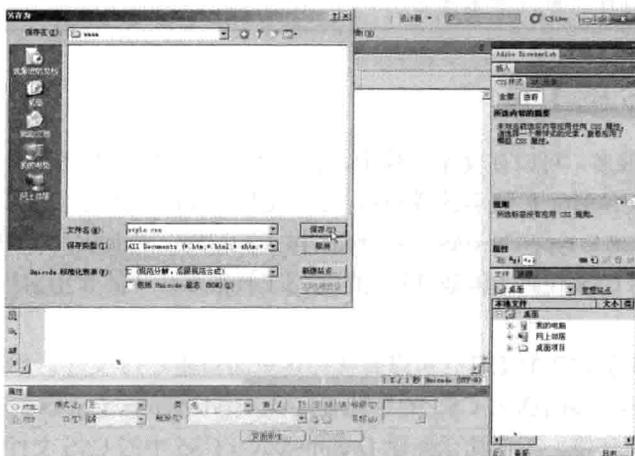


图 12-3 “另存为”对话框

这样就新建好了一个 CSS 文件,除了之前写好的 CSS 代码外,在下次打开 CSS 文件时可继续编写 CSS 代码。

上面讲解的是在 Dreamweaver CS5 中如何新建 CSS 文件,之前编者推荐过使用开源的 Aptana 工具,下面用示图的方式演示在 Aptana 中如何新建 CSS 文件。

12.7.4 打开 Aptana 工具

首先在桌面或通过选择“开始”|“程序”|“Aptana 启动”命令,打开 Aptana 软件,将会显示 Aptana 的环境界面。与其他工具软件一样,它也有菜单栏等其他的工具栏,这里的 Aptana 是英文版的,如图 12-4 所示。

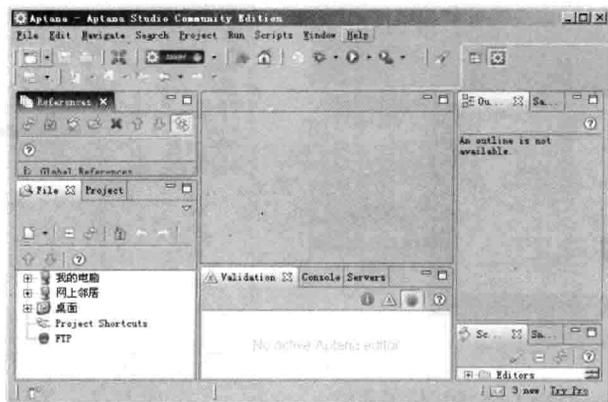


图 12-4 Aptana 工具环境界面

12.7.5 在菜单栏中新建 CSS 文件

在菜单栏中，选择“File”|“New”|“Untitled CSS File”命令，表示新建一个未命名的 CSS 文件，此时打开一个 CSS 编辑区，这时即可输入 CSS 代码，完成后在菜单栏中选择“File”|“Save”命令，这时会弹出“另存为”对话框，选择好路径并添加好文件名，单击“保存”按钮即可，如图 12-5 所示。

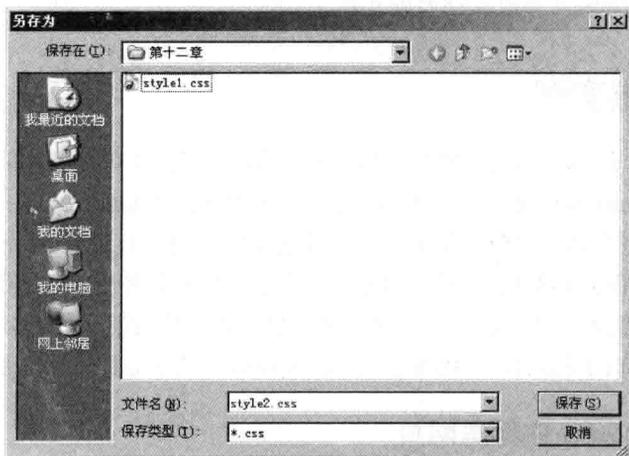


图 12-5 Aptana 中 CSS 文件“另存为”对话框

在两种工具中都可以新建 CSS 文件，写好的 CSS 代码通过 CSS 文件保存后，直接可以在网页中用外部链接 CSS 方式链接到网页，这样就可以在 CSS 文件中控制网页了。需要注意的是，一个 CSS 文件可以给不限量的网页链接的，只要 CSS 文件代码一变，所有使用该 CSS 文件的页面都会因代码变化而变化。

注意 内部链接 CSS 和内联方式不需要新建 CSS 文件。

12.8 CSS 与浏览器的兼容性

现在浏览器市场的主流浏览器主要是 Internet Explorer (简称 IE) 和 Mozilla Firefox (简称 FF)，网页中的 CSS 因两个浏览器支持不同，所以对同一代码会显示不同的效果，那么为了达到网页的统一风格，有时要注意浏览器对要写的 CSS 属性支持如何。如果某一属性只有一个浏览器支持，那另一浏览器对这属性代码就不起作用。

现在浏览器对 CSS 的支持主要有 CSS 标准，有这样的标准才可以约束浏览器厂商对 CSS 的支持度。

在 Aptana 中编写 CSS 代码，会出现自动提示功能，当输入属性后，系统自动出现相关内容供选择，通过对浏览器图标的颜色深度深浅来判断是否支持该浏览器，如图 12-6 所示。

从图 12-6 中可以看到，height 这个高度属性有 IE 和 FF 支持，而下面的 ime-mode 只有 IE 支持，而 FF 的图标呈灰色，表示不支持本属性。

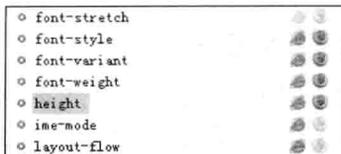


图 12-6 在 Aptana 中用图标表示浏览器对属性支持

注意 IE、FF和Chrome是目前主流的浏览器。

12.9 CSS 基础实例

在上面的图文中，演示了编写 HTML 和 CSS 工具的使用，本节用其中的 Aptana 环境来演示一个实例，主要通过 Aptana 来创建一个基本的网页。对于编写 HTML 或 CSS 的环境，可以选择 Aptana 或 Dreamweaver CS5，当然还可以用记事本，但最后保存的格式为.html 或.htm 为网页格式，如.css 表示 CSS 文件。本实例分别新建一个网页和 CSS 文件来演示如何用 CSS 控制 HTML 标记。

本实例介绍如何在网页中新建网页，并链接外部 CSS，然后把这个 CSS 文件引入到网页中，即可在 CSS 中看到网页中的内容，用到的工具是 Aptana，以下是具体操作步骤。

12.9.1 在 Aptana 中新建网页

在 Aptana 中新建网页，选择好 HTML 文件后，会出现图 12-7 所示的界面，当保存时，会弹出“另存为”对话框，只在其中输入 HTML 的文件名和保存路径即可。写好标题并保存为.html 的格式后，效果如图 12-7 所示。

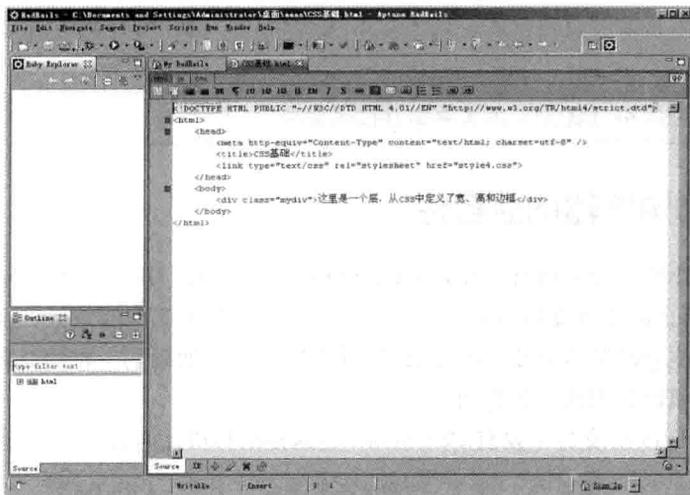


图 12-7 新建的网页

12.9.2 在 Aptana 中新建 CSS 文件

新建 CSS 文件并写入层的边框代码。首先建一个 CSS 文件，然后写入代码后保存为.css 的格式，如图 12-8 所示。

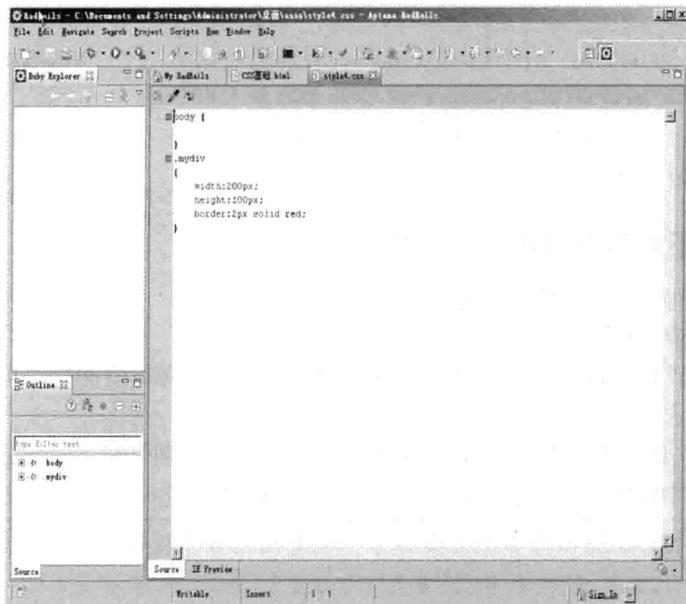


图 12-8 新建的 CSS 文件

12.9.3 链接 CSS 文件到 HTML 页面中

在网页文件中链接新建好的 CSS 文件。对于外部链接，是在<html>和<body>之间用<link>标记。代码 12-1 表示已加入外部链接的网页源码。

代码 12-1 CSS 基础

```
-----文件名: CSS基础.html-----
01 <html>
02   <head>
03     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
04     <title>CSS基础</title>
05     <link type="text/css" rel="stylesheet" href="style4.css">
06   </head>
07   <body>
08     <div class="mydiv">这里是一个层，从CSS中定义了宽、高和边框</div>
09   </body>
10 </html>
```

【代码解析】在 Aptana 中，每新建一个网页，系统都会自动加载好网页基本标记，但对英语版的工具来说，字体的默认是如下代码。

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

上面代码是 iso-8859-1 的，对于中文简体文字出现的是乱码，所以要设置 charset 为 utf-8（代码第 3 行），完成后代码如下。

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

这样才可以正常显示中文。代码 12-1 在<title></title>下面插入了外部链接，下面语句是外部链接 CSS 的。

```
<link type="text/css" rel="stylesheet" href="style4.css">
```

在代码 12-1 中,在 HTML 页面用标记<link>把外部的 CSS 文件引入到本页面中,而且可以在外部的 CSS 文件中控制本页面中的 HTML 标记,如设置 DIV 的选择符后,即可在 CSS 中控制。显示出来的效果如图 12-9 所示。

12.9.4 用不同的浏览器测试页面

对浏览器兼容问题进行调整,通过编写 CSS 代码,并用不同浏览器浏览页面后,才能发现哪个层或哪个区别出现不正常显示现象,然后再找到代码中的 CSS 属性名称,这样对照效果图判断问题,再从 CSS 中修正,最后达到多浏览器显示一样的效果。

现在的浏览器主要以 IE 和 FF 为主,所设计的网页必须要在两种环境下运行才可以放到网上,因为有时在 IE 中显示正常的效果,可能在 FF 中就会出现不正常错误,如过分的靠左或图片与文本重叠等。



图 12-9 用 CSS 控制 HTML 内容

说明

CSS是HTML+CSS布局中的主要部分,也是比较难掌握的,特别是在设置网页时,用到CSS来控制DIV。通过学习CSS属性后,对CSS控制DIV等标记有了一定的认识,那么对DIV等对象的控制就比较容易了,因为浏览器支持的CSS不同、故有的属性在多个浏览器下运行的效果不一样。

12.10 小结

本章主要介绍了 CSS 基本概念、CSS 的基本语法及在网页中如何配合 HTML 标记。CSS 文件链接方式有三种:内联方式、内部链接 CSS 和外部链接 CSS 方式。其中外部链接 CSS 要在外部建立好新的样式后链接才会成功。编写的 CSS 代码并不是所有浏览器都会支持,所以 CSS 属性的使用效果要与浏览器支持有关,这样做出来的网页在多种浏览器中才能正常显示,满足不同的浏览器用户浏览网页的需求。

12.11 习题

一、填空题

- _____是唯一性选择符。
- CSS 注解有两种方式:第一种是_____,第二种是_____。
- ID 选择符在 HTML 代码中,对 ID 符只引用一次,如果多次引用,只有_____有效。

二、上机实践

- 下面代码中的错误是什么?

```
<html>
<head>
<title> DIV与SPAN实例手把手—多层布局</title>
<style type="text/css">
/*span标记对应的css样式关联*/
```

```
.span1
{
    /*高为20像素*/
    height:20px;
    /*边框为1像素红色实线*/
    border:1px solid red;
        font-size:20px;
        color:blue;
}
</style>
</head>
<body>
<!--3个SPAN标记-->
<link type="text/css" rel="stylesheet" href="style4.css">
<div id="span1">
    高20像素,蓝色,字体大小20
</div>
</body>
</html>
```

【提示】注意外部链接 CSS 的定义语义的位置, CSS 的 ID 符号的定义。

2. 设置一个外部链接 DIV, 要求: 高度为 30px, 颜色为绿色, 边框为蓝色, 字体大小为 20px。

【提示】外部链接 DIV 在<head></head>中。

第 13 章 字体设置和文本设置

本章开始讲解 CSS 属性方面的知识，学好这些知识，才可以很灵活地应用 CSS，以便对页面的控制了如指掌，更早地实现网页制作的工作。本章讲解的是 CSS 中的字体属性和文本属性。

学完本章，将会了解以下内容。

- 字体和字体颜色；
- 字体类型；
- 字体大小和字体重量；
- 行距、间距与间隔；
- 文本省略标记；
- 文本空格处理方式；
- 文本水平对齐和垂直对齐；
- 文本缩进。

13.1 文字的字體

网页中的一段文字展现在用户面前的，不仅仅是一段内容，还会涉及好多外观，如文字的颜色、大小、字体类型等。本节就介绍有关字体的一些知识。

13.1.1 字体的复合属性

字体是给对象设置文本特性的属性，而且是复合属性。所谓复合属性是指该属性是由多种属性组合而成，在 CSS 中字体用 `font` 表示，字体 `font` 的语法形式如下。

```
font:font-style || font-variant || font-weight || font-size || line-height || font-family
```

`font` 代表字体的意思，冒号“:”后面接的就是 `font` 的参数，而复合属性的参数刚好是其他属性，这样在 `font` 中，只输入这些单个属性值就行，且每个参数只允许出现一个值，省略某些属性系统会按其默认值设置。下面是对各个属性作为字体的参数介绍，其中 `normal` 作为参数值的一般都是默认值。

- `font` 属性中的第一个参数 `font-style` 是设置对象中字体的样式，可选参数为 `normal`、`italic` 和 `oblique` 三种，分别代表正常字体、斜体和倾斜的字体，其中 `normal` 是默认值，以下都相同。
- `font-variant` 表示对象中的文本是否为小型的大写字母，也有两个可选参数 `normal` 和 `small-caps`，分别表示正常字体和小型大写字母的字体。
- `font-weight` 设置对象文本的粗细，也称字重，详情请看 13.1.4。

- font-size 设置对象文本字体大小, 详情请看 13.1.3。
- font-family 设置对象文本字体名称序列, 详情请看 13.1.5。

这些属性组成的复合属性中, 默认值为: normal normal normal medium normal "Times New Roman"。为了证实这些单独属性组成复合属性, 下面对以上知识做实例说明, 代码 13-1 演示字体复合特征。

代码 13-1 字体设置

```
-----文件名: 字体设置.html-----
01 <html>
02 <head>
03 <title>字体设置</title>
04 <style type="text/css">
05 #font1
06 {
07     /*设置字体复合属性*/
08     font:normal normal normal medium normal "Times New Roman";
09 }
10 #font2
11 {
12     /*设置字体为隶书*/
13     font:oblique normal 900 18px 隶书;
14 }
15 #font3
16 {
17     /*设置字体为黑体*/
18     font:italic small-caps 600 14px 黑体;
19 }
20 </style>
21 </head>
22 <body>
23 <div id="font1">
24     HTML+CSS完全自学手册--默认值
25 </div>
26 <div id="font2">
27     HTML+CSS完全自学手册--18像素隶书倾斜粗体
28 </div>
29 <div id="font3">
30     HTML+CSS完全自学手册--14像素黑体斜体粗体
31 </div>
32 </body>
33 </html>
```

【代码解析】代码 13-1 中用到字体的复合属性, 在<body></body>中插入 3 层, 第 1 层是字体默认的参数值 (第 8 行代码); 在第 2 层中, 设置了对象层的 18 像素隶书倾斜粗样式 (第 13 行代码); 与第 2 层一样, 第 3 层对复合属性字体设置成为 14 像素黑体斜体粗体 (第 18 行代码), 因斜体方式不一样, 粗体程度不同, 所以在第 2 层和第 3 层中, 表示的效果也不一样。值得注意的是, 在第 1 层的 CSS 样式中, 有没有 font 设置都是一样的, 因为默认值与没有设置任何字体属性值的效果是一样的。当然, 本实例是为了学习默认值而设置的默认值, 整体效果如图 13-1 所示。

注意

复合属性可以拆开使用,如只对字体设置粗体,并不一定需要设置font,可以用font-weight 取代。

13.1.2 字体颜色

在网页中会呈现五颜六色的文字,这都由字体颜色决定。字体颜色在 HTML 也介绍过,那只是 HTML 中的 font 标记,而本节要学习的是用 CSS 来控制字体颜色,达到轻量级代码重量级的效果。

字体颜色是对对象的字体设置颜色,用 color 表示,无默认值,所接的值是颜色值,颜色值的表示方法在本书中也有介绍,但常用的是#RRGGBB,本书的实例大部分采用此方式。其语法形式如下。

```
/*CSS代码:设置字体颜色*/
Color:颜色值
```

颜色值表示方法有三种:#RRGGBB、rgb(R,G,B)和颜色名。建议使用#RRGGBB,所有浏览器都支持这种颜色值。代码 13-2 表示了一个层中的字体,其颜色是红色。

代码 13-2 字体颜色

```
-----文件名:字体颜色.html-----
01 <html>
02 <head>
03 <title>字体颜色</title>
04 <style type="text/css">
05 /*设置所有标记DIV的字体颜色*/
06 Div
07 {
08     color:#ff0000;
09 }
10 </style>
11 </head>
12 <body>
13 <div>HTML+CSS完全自学手册</div>
14 </body>
15 </html>
```

【代码解析】用 CSS 设置字体颜色,首先在<html>和<body>中插入<style>标记对,然后在里面设置所有 DIV 的字体颜色值为 ff0000 (第 8 行代码),即红色。效果如图 13-2 所示。



图 13-1 复合字体属性

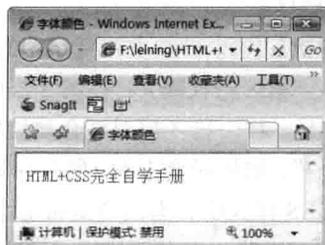


图 13-2 字体颜色

技巧

字体颜色设置可以对不同的标记，如上面的DIV，也可以是对p标记或h1等不同的对象可以设置不同的颜色。

13.1.3 字体大小

设置字体大小和设置颜色一样，都是对对象的文本设置的。字体大小用 font-size 表示，font 和 size 中间是横线，而不是下划线。font-size 后面可接的参数值有：xx-small、x-small、small、medium、large、x-large、xx-large、larger、smaller、length，这些参数代表字体尺寸的值。

参数 xx-small 指绝对字体尺寸，是最小的；而 x-small 是较小，但比 xx-small 稍大点；而参数 small 是小的值；medium 是默认值，是对象字体正常大小值；比 medium 大点的是参数 large，绝对字体尺寸中大的值；同理，x-large 是较大值；xx-large 是最大的值。

要注意的是，参数 larger 是相对字体的尺寸，相对于父对象中字体尺寸相对增大，使用成比例的 em 单位计算。与之相对应的是 smaller，相对于父对象中对字体尺寸相对减小，也是使用成比例的 em 单位计算。最后的 length 是百分数、浮点数字和单位标识符组成的长度，但不能为负值。百分比取值是基于父对象中字体的尺寸。下面用实例比较各个字体的大小，代码 13-3 演示了字体大小的用法。

代码 13-3 字体大小

```
-----文件名：字体大小.html-----
01 <html>
02 <head>
03 <title>字体大小</title>
04 <style type="text/css">
05 #xxsmall
06 {
07     /*设置字体大小为最小*/
08     font-size:xx-small;
09 }
10 #xsmall
11 {
12     /*设置字体大小为较小*/
13     font-size:x-small;
14 }
15 #small
16 {
17     /*设置字体为小型字体*/
18     font-size:small;
19 }
20 #large
21 {
22     /*设置字体为大型字体*/
23     font-size:large;
24 }
25 #xlarge
26 {
27     font-size:x-large;
28 }
```

```

29 #xxlarge
30 {
31     font-size:xx-large;
32 }
33 </style>
34 </head>
35 <body>
36 <p id="xxsmall">HTML+CSS完全自学手册</p>
37 <p id="xsmall">HTML+CSS完全自学手册</p>
38 <p id="small">HTML+CSS完全自学手册</p>
39 <p id="large">HTML+CSS完全自学手册</p>
40 <p id="xlarge">HTML+CSS完全自学手册</p>
41 <p id="xxlarge">HTML+CSS完全自学手册</p>
42 </body>
43 </html>

```

【代码解析】在代码 13-3 中，分别用绝对字体尺寸表示 6 种不同的字体，第 1 个段落设置了文本对象为最小（第 36 行代码）；第 2 个段落设置的文本字体为较小（第 37 行代码）；第 3 个段落设置对象字体为小型字体（第 38 行代码）；而在第 4 个段落中，设置字体为大型字体（第 39 行代码）；在第 5 个段落中，设置的字体为超大型字体（第 40 行代码）；最后在第 6 个段落中，设置字体为最大（第 41 行代码）。效果如图 13-3 所示。

字体大小除了上面的绝对字体尺寸、相对字体尺寸外还有用 length 表示的，此种方法可以写上具体的字体大小，如下代码用不同大小单位表示。

```

/*设置不同单位的字体*/
font-size:12px;
font-size:12pt;
font-size:9;

```

上面三行代码表示不同的单位值，所表示的字体大小也不一样，所以读者在设置字体大小时，在确定数值时，还要确定其单位是否需要设置。

13.1.4 字体重量

字体重量也就是字体粗细，其语法用 font-weight 表示，作用由用户端系统安装的字体的特定字体变量映射决定，系统选择最近的匹配。也就是说，用户可能看不到不同值之间的差异。

字重 font-weight 有多个参数可选择，除字母表示不同的字重外，还可以用数字表示，数字越大，字体越粗，但数值也是有限的选择。下面是表示字体重量（font-weight）的完整语法形式。

```
font-weight:normal|bold|bolder|lighter|100|200|300|400|500|600|700|800|900
```

参数 normal 表示正常大的字体，相当于用数字 400 表示，而 bold 是粗体格式，相当于 700，或者用 标记对表示的字体重量。

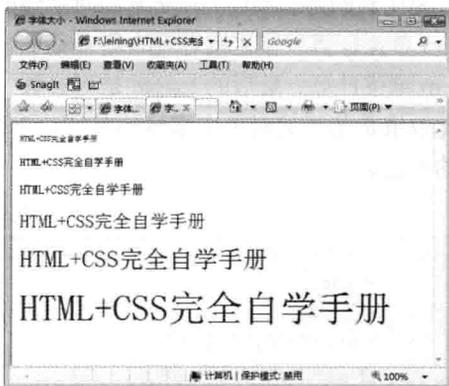


图 13-3 绝对字体大小

bolder 和 **lighter** 表示不是一个具体值，前者比正常值要粗，而后者比正常值要细，数字为从 100~900，字体粗细程度由细到粗。代码 13-4 表示字体重量。

代码 13-4 字体重量

```
-----文件名: 字体重量.html-----
01 <html>
02 <head>
03 <title>字体重量</title>
04 <style type="text/css">
05 #bold
06 {
07     /*以下是设置字体重量*/
08     font-weight:bold;
09 }
10 #bolder
11 {
12     /*设置为粗体*/
13     font-weight:bolder;
14 }
15 #lighter
16 {
17     /*设置字体重量,字体表现比较细*/
18     font-weight:lighter;
19 }
20 #num100
21 {
22     font-weight:100;
23 }
24 #num300
25 {
26     font-weight:300;
27 }
28 #num500
29 {
30     font-weight:500;
31 }
32 #num700
33 {
34     font-weight:700;
35 }
36 #num900
37 {
38     font-weight:900;
39 }
40 </style>
41 </head>
42 <body>
43 <p>HTML+CSS完全自学手册--正常</p>
44 <p id="bold">HTML+CSS完全自学手册--bold</p>
45 <p id="bolder">HTML+CSS完全自学手册--bolder</p>
46 <p id="lighter">HTML+CSS完全自学手册--lighter</p>
47 <p id="num100">HTML+CSS完全自学手册--100</p>
```

```

48 <p id="num300">HTML+CSS完全自学手册--300</p>
49 <p id="num500">HTML+CSS完全自学手册--500</p>
50 <p id="num700">HTML+CSS完全自学手册--700</p>
51 <p id="num900">HTML+CSS完全自学手册--900</p>
52 </body>
53 </html>

```

【代码解析】代码 13-4 表示了字体重量的不同参数，在多个段落中，设置不同的字体重量，用 bold 等关键字和 100~900 中的数字都可以表示字体重量（第 44~51 行代码），在网页中不同的字体重量表示出不同的文本粗细。效果如图 13-4 所示。

注意

字体的粗细可以由 bold 和 lighter 决定，也可以由数字 100~900 表示。

13.1.5 字体类型

字体类型是对象中文本的字体名称序列，序列可包含嵌入字体，因浏览器不同而默认的字体会不同。设置字体类型时，可以选择多种类型，用逗号隔开。字体类型的语法形式为 font-family，后面接字体的类型，用冒号隔开。其他语法表示如下。

```

/*CSS代码：设置字体类型*/
Font-family:fontname;

```

其中 fontname 是指字体类型，如宋体、Courier、Courier New。对于汉字，字体类型首选为中文类型字体，同时可以放多种字体。代码 13-5 表示字体类型的用法。

代码 13-5 字体类型

```

-----文件名：字体类型.html-----
01 <html>
02 <head>
03 <title>字体类型</title>
04 <styletype="text/css">
05 #fontclass1
06 {
07     /*以下是设置字体类型*/
08     font-family:"宋体";
09 }
10 #fontclass2
11 {
12     /*设置字体类型为多个*/
13     font-family:"Courier New",Courier,monospace;
14 }
15 #fontclass3
16 {
17     font-family:黑体;
18 }
19 </style>

```

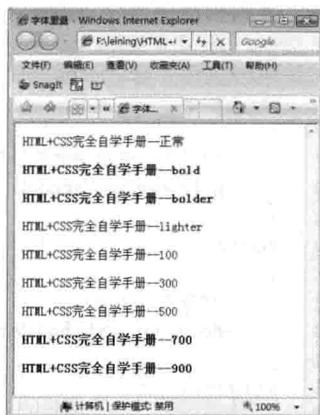


图 13-4 字体重量

```

20 </head>
21 <body>
22 <div="fontclass1">
23     html+css完全自学手册宋体
24 </div>
25 <div="fontclass2">
26     html+css完全自学手册--"Courier New",Courier,monospace;
27 </div>
28 <div="fontclass3">
29     html+css完全自学手册-黑体
30 </div>
31 </body>
32 </html>

```

【代码解析】第 22~30 行代码对 3 个层设置不同的字体类型。第 1 层中，设置为宋体（第 22 行代码），宋体只对汉字起作用，HTML+CSS 还是默认的英文字体；同理，第 2 层中，设置了 Courier New、Courier 和 monospace 三种字体（第 25 行代码），在显示字体类型时，系统先用第 1 种字体显示，如果第 1 种字体在系统中没有，那么显示第 2 种字体类型，直到最后一种设置的类型；在第 3 层中设置了中文为黑体。效果如图 13-5 所示。



图 13-5 设置字体类型

13.2 行距

行距指的是行与行之间的距离。在正文中，行距能让文本有足够的文本视觉空间，使用户浏览时感到更舒服、更有层次感。如果正文内容多，按默认的行距看，文本有点拥挤，这样很容易让用户感觉浏览困难，影响用户体验，因此需要调整行距。

在 CSS 中有行距的属性，用英文 line-height 表示，其后设置距离值。完整语法如下。

```
line-height: #;
```

其中#代表距离值，注意要加上单位，不同的单位表达的效果也不同，行距属性简单且容易理解。下面用实例来加深对行距 line-height 用法的理解，如代码 13-6 所示。

代码 13-6 设置行距

```

-----文件名：设置行距.html-----
01 <html>
02 <head>
03 <title>设置行距</title>
04 <styletype="text/css">
05 #font5
06 {
07     /*设置宽度和边框*/
08     width: 230px;
09     border: 1px solid #000000;
10     /*设置行距为30像素*/
11     line-height: 30px;

```

```

12 }
13 </style>
14 </head>
15 <div id="font5">
16     html+css完全自学手册html+css完全自学手册html+css完全自学手册html+css完全自
17     学手册html+css完全自学手册
18 </div>
19 </body>
20 </html>

```

【代码解析】第 11 行代码设置了层对象中的行距为 30 像素，即第 1 行与第 2 行间的距离为 30 像素，同理第 2 行与第 3 行也如此，设置行距后，网页高度拉长了，增强了用户体验。效果如图 13-6 所示。

13.3 设置文本间距与间隔

本节来介绍如何设置文本间距与间隔。首先要区别间距与间隔的含义，文本的间距是指单词（包括汉字与字母）之间插入的空格数，最小单位是单词，也就是说距离差只是单词与单词的，与单词内部没关系，间距语法用英语表示为 `word-spacing`，其后接值为 `normal` 或具体的值，也可以接负数。文本的间隔是指文本中字符、单词的间隔，即文字间隔和字母间隔。文本的间隔设置包括设定字与字之间，字母与字母之间的距离包括文字间隔和字母间隔的设置。换言之，间隔就是指单词内部各文字（包含英语单词间各字母）间的距离值，其语法与 `word-spacing` 相似，用 `letter-spacing` 表示，默认值也为 `normal`，表示正常的文本间隔，当然，也可以用负值表示。

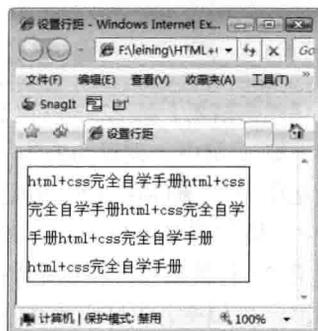


图 13-6 设置行距

13.3.1 文本间距

文本间距用 `word-spacing` 表示，说明文本间的距离值，其后接的值的完整语法如下所示。

```
word-spacing: normal | 长度单位
```

如果该参数使用 `normal` 值，将按照浏览器的默认设置显示，相反直接设置其值（可以是负值），其间距就是设置的值的距离，值越大间距就越宽。

13.3.2 文本间隔

与 `word-spacing` 接的值一样，`letter-spacing` 也可以接 `normal` 或者数值表示间隔，文本间隔的完整代码如下。

```
letter-spacing: normal | length
```

与间距一样，间隔的值也是默认 `normal`，且也可以为负值。下面用实例来演示间距与间隔的区别。代码 13-7 所表示的是 `word-spacing` 与 `letter-spacing` 在设置文本时的用法，所表现的效果也不同。

代码 13-7 间距与间隔

```

-----文件名：间距与间隔.html-----
01 <html>
02 <head>
03 <title>间距与间隔</title>
04 <styletype="text/css">
05 #word1
06 {
07     /*设置间距*/
08     word-spacing:normal;
09 }
10 #word2
11 {
12     /*设置间距*/
13     word-spacing:-5px;
14 }
15 #word3
16 {
17     /*设置间距*/
18     word-spacing:5px;
19 }
20 #letter1
21 {
22     /*设置间隔*/
23     letter-spacing:normal;
24 }
25 #letter2
26 {
27     /*设置间隔*/
28     letter-spacing:-5px;
29 }
30 #letter3
31 {
32     /*设置间隔*/
33     letter-spacing:5px;
34 }
35 </style>
36 </head>
37 <body>
38 下面是间距：
39 <divid="word1">
40     间距的表示--HTML+CSS完全自学手册word-spacing,todayismonday.
41 </div>
42 <divid="word2">
43     间距的表示--HTML+CSS完全自学手册word-spacing,todayismonday.
44 </div>
45 <divid="word3">
46     间距的表示--HTML+CSS完全自学手册word-spacing,todayismonday.
47 </div>
48 下面是间隔：
49 <divid="letter1">
50     间隔的表示--HTML+CSS完全自学手册letter-spacing

```

```

51 </div>
52 <div id="letter2">
53     间隔的表示--HTML+CSS完全自学手册letter-spacing
54 </div>
55 <div id="letter3">
56     间隔的表示--HTML+CSS完全自学手册letter-spacing
57 </div>
58 </body>
59 </html>

```

【代码解析】第 39~55 行代码展示了间距与间隔的用法，用比较法，比较二者的区别，读者在学习间距与间隔时，用效果图学习比较更容易理解其区别和用法。效果如图 13-7 所示。

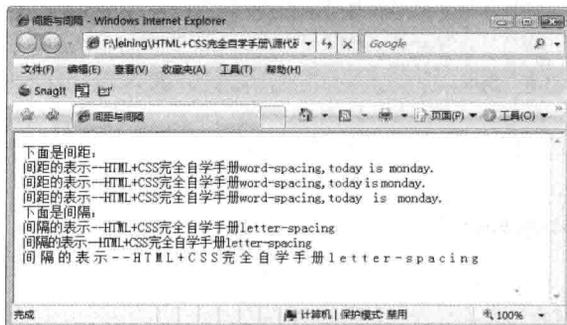


图 13-7 设置间距与间隔

注意

间隔与间距的区别。这两者对中文与英语所表示的效果不一样，对实际项目进行设置其值。

13.4 字体实例——字体花样

字体在网页中的作用前面章节已介绍，在汉字中常用的字体有宋体、黑体等。其实网页中也可以设置字体，而且也可以设置字体的大小、颜色等，还可以设置行与行之间的行距和文字的间距等。

本实例分四步来讲解字体花样，通过 CSS 中的字体属性来表现出字体的丰富性，字体是复合属性，由一些与字体关联的属性组成，如字体大小、字体类型等，都可以设置字体的样式。下面通过实例来演示字体的用法。

13.4.1 加入字体名字

先在网页中输入不同字体的名字，这些名字由不同的层组成，分成 4 个层，分别设置 ID 为 font1、font2、font3 和 font4。在没有 CSS 的控制下，这些层中的文本字体还是原来的默认字体类型和字体大小。可以用 CSS 字体属性改变字体的样式。代码 13-8 显示了字体文本在没有 CSS 控制下的效果。

代码 13-8 字体实例手把手——字体花样

```

-----文件名: 字体实例手把手——字体花样.html-----
01 <html>
02 <head>

```

```

03 <title>字体实例手把手——字体花样</title>
04 </head>
05 <body>
06 <!-- 布局DIV标记-->
07 <div id="font1">
08     宋体
09 </div>
10 <div id="font2">
11     黑体
12 </div>
13 <div id="font3">
14     隶书
15 </div>
16 <div id="font4">
17     粗体
18 </div>
19 </body>
20 </html>

```

【代码解析】因只有 HTML 代码，没有用 CSS 控制，所定的层没起到修饰效果，所以显示的文本还是原始状态。而 DIV 中的层本来就是占一行的宽度，故在每个层中都表示单独的行，且其内对象为默认的字体类型和字体大小，分别表示“宋体”、“黑体”、“隶书”和“粗体”。效果如图 13-8 所示。

13.4.2 加入 CSS 控制字体

对上一节中的字体进行样式设置，需要在<style></style>标记对中加入 CSS 代码，并用上面设置的 ID 选择符表示样式的名称，如对 ID 为 font1 的层，在 CSS 中设置了字体类型 font-family 为“宋体”；在 ID 为 font2 的层中，设置其内对象为“黑体”的字体类型；在 ID 为 font3 的选择符中，设置的字体类型为“隶书”；而在最后一个层中，设置的字体重量 (font-weight) 为粗体。因正文内容不变，只对样式进行设置，代码如下所示。

```

01 <style type="text/css">
02 #font1
03 {
04     /*设置字体类型*/
05     font-family:"宋体";
06 }
07 #font2
08 {
09     font-family:"黑体";
10 }
11 #font3
12 {
13     font-family:"隶书";
14 }
15 #font4

```



图 13-8 字体文本

```

16 {
17     /*设置字体重量*/
18     font-weight:bold;
19 }
20 </style>

```

【代码解析】第 5、9、13、18 行代码设置与之关联的 DIV 中的 ID 选择符样式,前 3 个层表示汉字不同的字体类型,而在最后一个层,也就是第 4 层中,设置层内对象为粗体,在网页中表现为第 1 个层是宋体的文字;第 2 层是黑体的文字;同理,第 3 层是隶书;而在最后一层中,为粗体的文本,且字体类型为默认状态。效果如图 13-9 所示。

13.4.3 设置行距

输入多行文字设置其行距。再新建一个层,然后输入文字,再用 CSS 控制其行距。这时为了区别行距,设置为 30 像素。完整代码如代码 13-9 所示。



图 13-9 CSS 控制字体样式

代码 13-9 字体实例手把手—字体花样

```

-----文件名: 字体实例手把手—字体花样.html-----
01 <html>
02 <head>
03 <title>字体实例手把手—字体花样</title>
04 <styletype="text/css">
05 #font1
06 {
07     /*设置字体类型为宋体*/
08     font-family:"宋体";
09 }
10 #font2
11 {
12     /*设置字体类型为黑体*/
13     font-family:"黑体";
14 }
15 #font3
16 {
17     /*设置字体类型为隶书*/
18     font-family:"隶书";
19 }
20 #font4
21 {
22     /*设置字体为粗体*/
23     font-weight:bold;
24 }
25 #font5
26 {
27     width:150px;
28     border:1pxsolid#000000;

```

```

29     line-height:30px;
30 }
31 </style>
32 </head>
33 <body>
34 <div id="font1">
35     宋体
36 </div>
37 <div id="font2">
38     黑体
39 </div>
40 <div id="font3">
41     隶书
42 </div>
43 <div id="font4">
44     粗体
45 </div>
46 <div id="font5">
47     HTML+CSS完全自学手册HTML+CSS完全自学手册HTML+CSS完全自学手册
48 HTML+CSS完全自学手册HTML+CSS完全自学手册
49 </div>
50 </body>
51 </html>

```

【代码解析】 在第 2 步的基础上加入文本和代码，第 25~30 行代码设置了对对象的行距。在多行中，行与行之间的距离设置为 30 像素，且对象层中有边框显示。效果如图 13-10 所示。

13.4.4 设置文本间隔

可以设置行中文本间距和间隔。在#font5 的基本上设置对象的间距和间隔，与第 2 步一样，在本步骤只列出样式部分代码。

```

01 #font5
02 {
03     width:150px;
04     border:1pxsolid#000000;
05     line-height:30px;
06     /*设置字体间距*/
07     word-spacing:10px;
08     /*设置字体间隔*/
09     letter-spacing:5px;
10 }

```

【代码解析】 本步骤只在#font5 的基础上加上间距与行距的 CSS 代码，其他代码没变。第 3~9 行代码设置层 ID 为 font5 的样式，宽度为 150 像素，边框用 1 像素的黑色实线表示，其内都是层 font5 的范围，设置的行高为 30 像素，字体间距为 10 像素，而间隔为 5 像素。网页中表现的字体效果如图 13-11 所示。

说明

本实例讲解了字体在不同的 CSS 属性控制下所表现出来的效果，主要对 CSS 如何控制字体做实例演示。

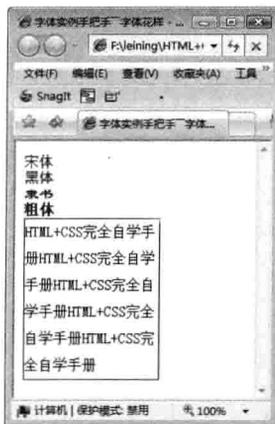


图 13-10 设置行距

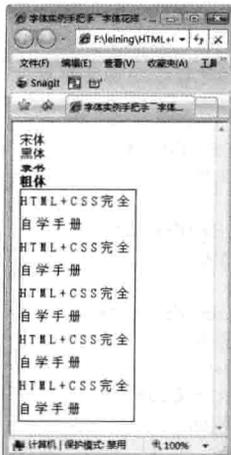


图 13-11 整体效果图

13.5 字母大小写转换

文本转换属性仅仅被用于表达某种格式的要求，它有四个属性，允许通过其中的一个来转换文本。在 CSS 中通过 `text-transform` 属性可以使网页设计者很容易控制字母的大小写，即可以用该属性来转换字母文本的大小写。其语法格式如下。

```
text-transform: capitalize | uppercase | lowercase | none
```

属性 `text-transform` 的参数如上所示，在所表示的语法中，对各参数值的意义说明如表 13-1 所示。

表 13-1 text-transform 属性各参数值说明

参 数 值	说 明
capitalize	使每个词的首字母大写
uppercase	使每个字的所有字母大写
lowercase	使每个字的所有字母小写
none	字母以正常形式显示

表 13-1 中为 `text-transform` 属性控制字母的大小写的参数列表。下面用实例来演示字母文本的大小写转换，如代码 13-10 所示。

代码 13-10 字母大小写转换

```
-----文件名: 字母大小写转换.html-----
01 <html>
02 <head>
03 <title>text-transform属性</title>
04 <style type="text/css">
05 .mydiv
06 {
```

```
07     width:auto;
08     margin:5px;
09     padding:5px;
10     border:1px solid #000000;
11 }
12 /*设置文本修饰*/
13 #tran1
14 {
15     text-transform:none;
16 }
17 #tran2
18 {
19     /*转为大写*/
20     text-transform:capitalize;
21 }
22 #tran3
23 {
24     text-transform:uppercase;
25 }
26 #tran4
27 {
28     /*转为小写*/
29     text-transform:lowercase;
30 }
31 </style>
32 </head>
33 <body>
34 <div class="mydiv">
35     输入小写的: html+css<br>
36     <input type="text" id="tran1">
37     值为none: 无转换发生。
38 </div>
39 <div class="mydiv">
40     输入小写的: html+css<br>
41     <input type="text" id="tran2">
42     值为capitalize: 每个词的首字母大写, 只有为小写输入状态转换才发生。
43 </div>
44 <div class="mydiv">
45     输入小写的: html+css<br>
46     <input type="text" id="tran3">
47     uppercase : 转换成大写。
48 </div>
49 <div class="mydiv">
50     输入大写的: HTML+CSS<br>
51     <input type="text" id="tran4">
52     lowercase : 转换成小写。
53 </div>
54 </body>
55 </html>
```

【代码解析】 代码 13-10 演示了 text-transform 属性的使用方法。在网页中存在 4 个层，每个层中都有可以输入值。第 1 层，设置 text-transform 值为 none，即不发生转换（第 36 行代码）；第 2 层设置 text-transform 值为 capitalize，让其首字母为大写，在英语正文中较常用（第 41 行代码）；第 3 层设置 text-transform 值为 uppercase，就是把小写转换成大写，当在此层输入小写字母后，显示出来的就是大写（第 46 行代码）；第 4 层刚好相反，设置 text-transform 值 lowercase，就是把大写转换成小写（第 51 行代码）。应当注意，如果从大写转换成小写时，输入小写，显示出来的还是小写，小写转换成大写样式也是一样的。其效果如图 13-12 所示。

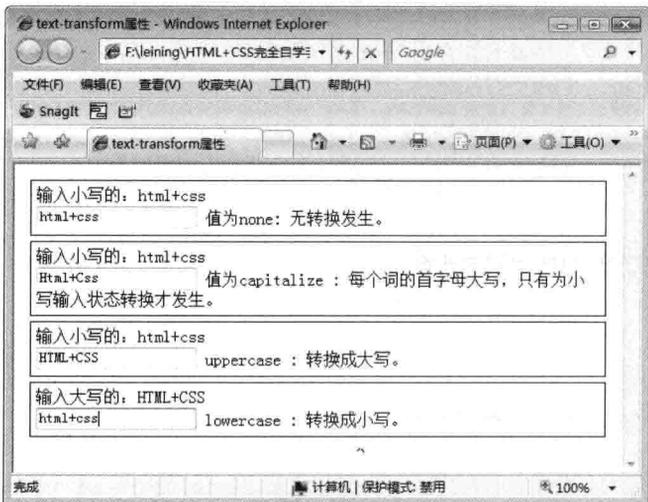


图 13-12 文本转换属性

13.6 文本修饰

文本修饰属性允许通过五个属性中的一个来设置文本的某种效果，在进行文本修饰时，如加上下划线、上划线、删除线、闪烁或者缺省地使用。其语法格式如下。

```
text-decoration: underline | overline | line-through | blink | none
```

语法中各参数值的意义说明如表 13-2 所示。

表 13-2 text-decoration 参数值说明

参 数 值	说 明
underline	给文字加下划线
overline	给文字加上划线
line-through	给文字加删除线
blink	文字在闪烁
none	缺省使用无

下面使用 text-decoration 属性对文本修饰，其中 none 参数也非常有用，它可以使链接的文字不以下划线的形式显示。例如，取消链接时带下划线的形式。为了说明文本修饰属性 text-decoration

的用法，下面用实例来演示 text-decoration 的各参数用法，通过比较来学习文本修饰所给网页添加的美观元素。代码 13-11 中用文本修饰属性在不同的段落设置不同的参数值，比较所表示的效果有什么不一样。

代码 13-11 文本修饰

```
-----文件名: 文本修饰.html-----
01 <html>
02 <head>
03 <title>文本修饰属性text-decoration </title>
04 </head>
05 <style type = "text/css">
06 <!--
07 #p1
08 {
09     /*设置文本修饰属性*/
10     text-decoration: underline;
11 }
12 #p2
13 {
14     text-decoration: overline;
15 }
16 #p3
17 {
18     /*设置文本修饰属性*/
19     text-decoration: line-through;
20 }
21 #p4
22 {
23     text-decoration: blink;
24 }
25 -->
26 </style>
27 <body bgcolor = lightyellow>
28 <center>
29 <h1>text-decoration属性的应用效果</h1>
30 </center>
31 <p id = p1>这段的文本修饰属性(text-decoration)值是underline。</p>
32 <p id = p2>这段的文本修饰属性(text-decoration)值是overline。</p>
33 <p id = p3>这段的文本修饰属性(text-decoration)值是line-through。</p>
34 <p id = p4>这段的文本修饰属性(text-decoration)值是line-blink </p>
35 </body>
36 </html>
```

【代码解析】代码 13-11 演示了 text-decoration 属性的使用方法，在 4 个段落中，使用文本修饰属性展现了 4 个不同的效果（由第 31~34 行代码实现），效果如图 13-13 所示。

注意 要正确地应用文本修饰，才可以给网页添加效果。



图 13-13 文本修饰属性

13.7 空格处理方式

空格处理应用在给对象处理文本间的空格时,如要使文本在一行中显示或者在计算机编程中的代码,因代码常常用缩进和转行的方式以表示代码的美观性,故在网页中显示要忽视这些空间(正常情况下,网页中的代码中有很多空格符只接收一个空格符),这就需要对空格进行处理。

空格处理属性是由 white-space 表示,其后接的值有默认的 normal 和预先的模式,还有只在一行中显示。其完整用法如下。

```
white-space:normal | pre | nowrap
```

- normal: 默认处理方式。
- pre: 用等宽字体显示预先格式化的文本。不合并字间的空白距离和进行两端对齐。参阅 pre 对象。
- nowrap: 强制在同一行内显示所有文本,直到文本结束或者遭遇 br 对象。参阅 td、div 等对象的 nowrap 属性(特性)。
- 代码 13-12 表示空格处理。

代码 13-12 空格处理

```
-----文件名: 空格处理.html-----
01 <html>
02 <head>
03 <title>空格处理</title>
04 <style type="text/css">
05 #ws1
06 {
07     width:200px;
08     border:1px solid #000000;
09     /*设置空格处理样式为正常*/
10     white-space:normal;
11 }
12 #ws2
13 {
14     width:200px;
15     border:1px solid #000000;
16     /*设置空格处理样式为预处理*/
17     white-space:pre;
18 }
```

```

19 #ws3
20 {
21     width:150px;
22     border:1px solid #000000;
23     /*设置空格处理样式为强制一行显示*/
24     white-space:nowrap;
25     padding:10px;
26 }
27 </style>
28 </head>
29 <body>
30 <div id="ws1">
31     HTML+CSS完全自学手册HTML+CSS完全自学手册
32 </div>
33 <div id="ws2">
34     HTML+CSS完全自学手册
35     以下是本对象的CSS代码:
36     #ws2
37     {
38         width:100px;
39         border:1px solid #000000;
40         white-space:pre;
41     }
42 </div>
43 <div id="ws3">
44     HTML+CSS完全自学手册HTML+CSS完全自学手册
45 </div>
46 </body>
47 </html>

```

【代码解析】上述代码中，3个层用固定宽度表示，第1层中正常的处理，在超过层宽度时自动换行显示（第30~32行代码）；第2层是预处理方式，即代码中是怎么显示，在网页中也跟着显示，常常用于计算机代码（第33~42行代码）；第3层是强制性处理，在一行中显示完才会换行，不管是否超出对象层的宽度（第43~45行代码）。其效果如图13-14所示。

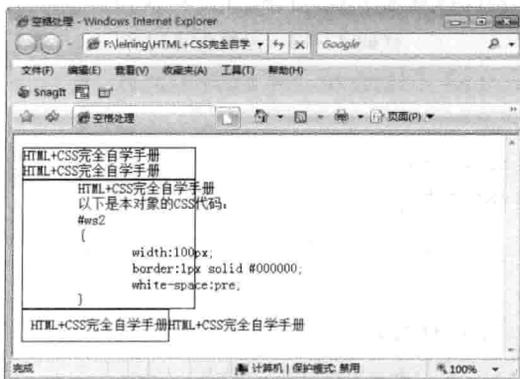


图 13-14 空格处理

注意 要修改预处理方式的文本空格，可直接在代码中修改。

13.8 设置文本垂直与水平对齐

垂直对齐属性可以用作一个内部元素的纵向位置，相对于它的上级元素或相对于元素行（一个内部元素是没有行在其前后断开的，例如，在 HTML 中的 EM、A、IMG），它的值可以是一个相对于元素行高属性的百分比，它会在上级基线上增高基线的指定数量。

13.8.1 文本垂直对齐

文本垂直对齐默认是向底部对齐，如果需要居中对齐或向顶部对齐就需要设置垂直对齐的方式，需用到 `vertical-align` 属性，设置层对象。

文本垂直也允许使用负值，该值也可以是一个方位关键字。其语法格式如下。

```
vertical-align: baseline | sub | super | top | text-top | middle | bottom | text-bottom
```

为了说明其参数的用法，以下用表 13-3 列出，并用文字说明其用法。

表 13-3 vertical-align 参数值说明

参 数 值	说 明
baseline	使元素和上级元素的基线对齐
sub	使对象以下标的形式显示
super	使对象以上标的形式显示
top	使元素和行中最高的元素向上对齐
text-top	使元素和上级元素的字体向上对齐
middle	纵向对齐元素基线加上上级元素的 x-高度
bottom	使元素和行中最高的元素向下对齐
text-bottom	使元素和上级元素的字体向下对齐

下面使用 `vertical-align` 属性设置文字的垂直对齐，如代码 13-13 所示。

代码 13-13 设置垂直对齐

```
-----文件名：文字垂直对齐.html-----
<html>
01 <head>
02 <title>设置垂直对齐</title>
03 <style type = "text/css">
04 <!--
05 div
06 {
07     height:40px;
08     border:1px solid #000000;
09 }
10 #base{
11     /*此为默认值*/
12     vertical-align:baseline;
13 }
14 #sub{
```

```

15      /*设置垂直对齐方式为下标*/
16      vertical-align:sub;
17      font-size:0.5em;
18  }
19  #sup{
20      /*设置垂直对齐方式为上标*/
21      vertical-align:super;
22      font-size:0.5em;
23  }
24  #top{
25      /*设置垂直对齐方式为向上对齐*/
26      vertical-align:top;
27  }
28  #t-top{
29      /*设置为文字垂直向上对齐*/
30      vertical-align:text-top;
31  }
32  #mid{
33      /*设置垂直对齐方式为居中对齐*/
34      vertical-align:middle;
35  }
36  #bot{
37      /*设置垂直对齐方式为向下对齐*/
38      vertical-align:bottom;
39  }
40  #b-bot{
41      /*设置为文字垂直向下对齐*/
42      vertical-align:text-bottom;
43  }
44  -->
45  </style>
46  </head>
47  <body>
48  <center>
49  <font size = 5 color = red> vertical-align垂直对齐属性的应用效果</font>
50  <table border = 4 height = 200>
51  <tr bgcolor = lightgreen align = center>
52  <td>设置vertical-align属性</td>
53  <td>显示结果</td>
54  <td>设置vertical-align属性</td>
55  <td>显示结果</td>
56  </tr>
57  <tr>
58  <td >vertical-align:sub</td>
59  <td>c+o<font id = sub>2</font>→co<font id = sub>2</font></td>
60  <!--在font标签内利用id选择符, 引用sub样式-->
61  <td>vertical-align:super</td>
62  <td>a<font id = sup>2</font>+b<font id = sup>2</font>=c<font id = sup>2</font></td>
63  <!--在font标签内利用id选择符, 引用sub样式-->
64  </tr>
65  <tr>
66  <td id = top>vertical-align:top</td>
67  <td id = top></td>
68  <!--在td标签内利用id选择符, 引用top样式-->
69  <td id = t-top>vertical-align:text-top</td>
70  <td id = t-top><img src ="logo.jpg"></td>
71  <!--在td标签内利用id选择符, 引用t-top样式-->

```

```

72 </tr>
73
74 <tr>
75 <td id = mid>vertical-align:middle</td>
76 <td id = mid><img src = "logo.jpg"></td>
77 <!--在td标签内利用id选择符, 引用mid样式-->
78 <td id = base>vertical-align:baseline</td>
79 <td id = base ><img src = "logo.jpg"></td>
80 <!--在td标签内利用id选择符, 引用base样式-->
81 </tr>
82 <tr>
83 <td id = bot>vertical-align:bottom </td>
84 <td id = bot><img src = "logo.jpg"></td>
85 <!--在td标签内利用id选择符, 引用bot样式-->
86 <td id = t-bot>vertical-align:text-bottom</td>
87 <td id = t-bot><img src = "logo.jpg"></td>
88 <!--在td标签内利用id选择符, 引用t-bot样式-->
89 </tr>
90 </table>
91 </center>
92 </body>
93 </html>
    
```

【代码解析】代码 13-13 中，用表格布局在单元格中显示出垂直对齐，分别表示了上顶端、中间和底部对齐，演示了 vertical-align 属性的使用方法（第 49~87 行代码）。效果如图 13-15 所示。

13.8.2 文本水平对齐

文本水平对齐属性可以应用于块级元素（P、H1 等），使元素文本得到排列。水平对齐是个很常用的属性，与 HTML 中的对齐不同的是，水平对齐是用 CSS 来控制，且在 CSS 中水平对齐属性用 text-align 表示；这个属性的功能类似于 HTML 的段、标题和部分的 ALIGN 属性。其语法格式如下。

```
text-align: left | right | center | justify
```

语法中各参数值的意义说明如表 13-4 所示。

表 13-4 text-align 参数值说明

参 数 值	说 明
left	浏览器默认的对齐方式，左对齐
right	右对齐
center	居中
justify	左右对齐



图 13-15 设置文字垂直对齐

各参数的用法与前面介绍 HTML 时的用法差不多，但在 CSS 中如何来控制文本水平对齐呢？下面使用 text-align 属性控制文本水平对齐，在段落中设置不同的对齐参数，显示不同的对齐方式，如代码 13-14 所示。

代码 13-14 文本排列

```
-----文件名：文本排列.html-----
01 <html>
02 <head>
03 <title>text-align属性</title>
04 <style type="text/css">
05 <!--
06 #h1 {
07     /*此为居中对齐*/
08     text-align: center;
09 }
10 #h2 {
11     /*此为默认值*/
12     text-align: left;
13 }
14 #h3 {
15     /*此为右对齐*/
16     text-align: right;
17 }
18 -->
19 </style>
20 </head>
21 <body >
22 <h1>text-align属性的应用效果</h1>
23 </center>
24 <p id = h1>此行文字居中对齐</p>
25 <p id = h2>此行文字左对齐</p>
26 <p id = h3>此行文字右对齐</p>
27 </body>
28 </html>
```

【代码解析】 代码 13-14 的第 24~26 行演示了 text-align 属性的使用方法。效果如图 13-16 所示。

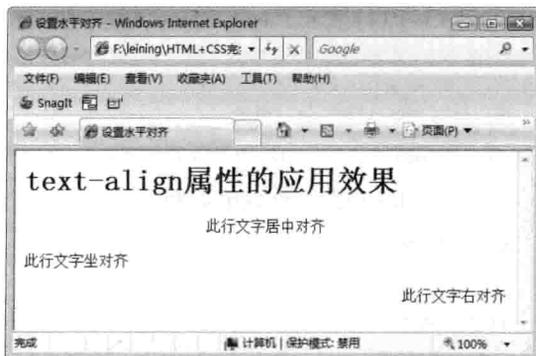


图 13-16 文本水平对齐

注意 垂直对齐比水平对齐用得少，要注意参数的设置。

13.9 文本缩进

文本缩进属性可以应用于块级元素(P、H1 等),以定义该元素第一行可以接受的缩进的数量。该值必须是一个长度或一个百分比。若是百分比则视上级元素的宽度而定。通用的文本缩进用法适用于段的缩进。其语法格式如下。

```
text-indent: <长度> | <百分比>
```

属性 text-indent 表示文本的缩进,属性值有两种表示方式:第一种直接接长度值,表示文本缩进的具体长度;第二种方式用百分数表示,表示该文本缩进在对象中占有的百分数。代码 13-15 使用 text-indent 属性控制文本的排列。

代码 13-15 文本缩进

```
-----文件名: 文本缩进.html-----
01 <html>
02 <head>
03 <title>首行缩进</title>
04 <style type="text/css">
05 <!--
06 #indent1 {
07     text-indent:20px;
08 }
09 -->
10 </style>
11 </head>
12 <body>
13 <h1>text-indent属性的应用效果</h1>
14 <h3>首行不缩进效果: </h3>
15 <p >HTML+CSS完全自学手册<br>
16     HTML+CSS完全自学手册HTML+CSS完全自学手册</p>
17 <h3>首行缩进30像素: </h3>
18 <p id = indent1>HTML+CSS完全自学手册<br>
19     HTML+CSS完全自学手册HTML+CSS完全自学手册</p>
20 </body>
21 </html>
```

【代码解析】代码 13-15 的第 13~19 行演示了 text-indent 属性的使用方法,效果如图 13-17 所示。

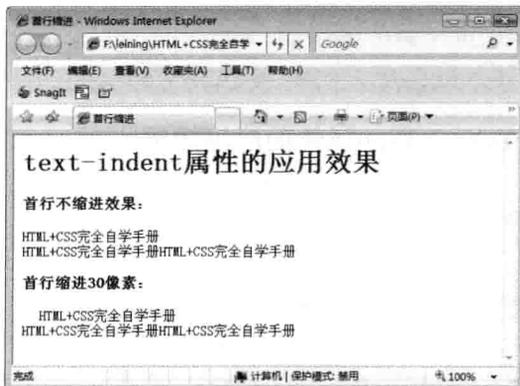


图 13-17 文本缩进属性

13.10 文本设置——给段落设置缩进

正文往往在一个层中显示，所以设置好这个层的样式后，再对段落设置缩进，系统会对每个段落设置其缩进，这样的效果更适合用户阅读。下面用实例来演示其用法。

13.10.1 对 DIV 布局

输入文本，定好层的 ID。先新建网页，然后在<body></body>中输入层和 ID，显示的是原始状态。

代码 13-16 文本设置—给段落设置缩进

```
-----文件名：文本设置手把手-给段落设置缩进.html-----
01 <html>
02 <head>
03 <title>文本设置手把手-给段落设置缩进</title>
04 </head>
05 <div id="indent1">
06 <p>文本设置手把手-给段落设置缩进文本设置手把手-给段落设置缩进文本</p>设置手把手
07 -给段落设置缩进文<p>本设置手把手-给段落设置缩进
08 文本设置手</p>把手-给段落设置<p>缩进文本设置手把手-给段落设置缩进</p>
09 </div>
10 </body>
11 </html>
```

【代码解析】上面代码设置了一行，一行里分了5段，在网页中表现的效果如图 13-18 所示。

13.10.2 设置文本首行缩进

在段落默认情况下，首行缩进在段落中默认的第一行没有缩进，但在日常文章或其他的段落中，需要首行缩进。要想设置首行缩进，需用到 CSS 属性 text-indent，其所接的值表示缩进的距离。代码 13-17 表示用 CSS 代码设置文本的首行缩进。

代码 13-17 文本设置手—给段落设置缩进

```
-----文件名：文本设置手把手-给段落设置缩进.html-----
01 <html>
02 <head>
03 <title>文本设置手把手-给段落设置缩进</title>
04 <style type="text/css">
05 #indent1
06 {
07     width:200px;
08     border:1px solid #000000;
09     padding:5px;
10     /*设置首行缩进*/
11     text-indent:18px;
12     /*设置垂直对齐*/
13     vertical-align:top;
14 }
15 </style>
16 </head>
```

```

17 <body>
18 <div id="indent1">
19 <p>文本设置手把手—给段落设置缩进文本设置手把手—给段落设置缩进文本</p><p>设置手把
20 手—给段落设置缩进文</p><p>本设置手把手—给段落设置缩进</p>
21 <p>文本设置手</p><p>把手—给段落设置</p><p>缩进文本设置手把手—给段落设置缩进</p>
22 </div>
23 </body>
24 </html>

```

【代码解析】代码 13-17 的第 11 行通过 text-indent，让段落缩进 18 像素。效果如图 13-19 所示。

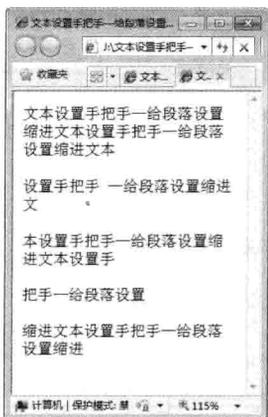


图 13-18 在段落中输入文本

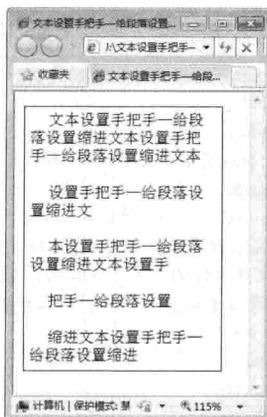


图 13-19 有缩进后的效果

13.11 小结

本章主要介绍了字体和文本方面的属性。字体在网页中占很重要的地位，因为有了字体，内容显得更丰富。字体是由字体和字体颜色、字体类型、字体大小、字体重量、行距和间距等组成，通过 CSS 设置这些属性，使对象的字体更丰富。文本属性在网页布局中很重要，也是影响网页美观的属性中的几种，如文字修饰 text-decoration 属性、垂直对齐 vertical-align 属性、水平对齐 text-align 属性及文本缩进 text-indent 属性。文本缩进属性很有必要在正文对象层中设置，使每个段落的第一行都有缩进，使正文合乎正常阅读习惯。

13.12 习题

一、填空题

- font 属性中的第一个参数 font-style 是设置对象中字体的样式，可选参数为：_____、_____和_____三种。
- 文本间距用_____表示，说明文本间的距离值。
- 文本修饰属性允许通过五个属性中的一个来设置文本的某种效果，这五个属性是_____、_____、_____和_____。

4. 文本转换属性仅仅被用于表达某种格式的要求,它有四个属性,分别是_____、_____、_____和_____。

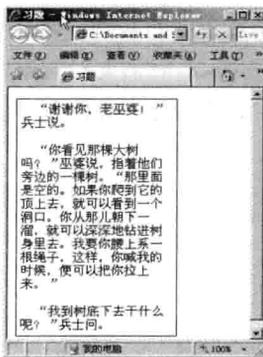
二、上机实践

1. 通过 CSS 对网页中“我爱网页制作,我在努力学习”的文本进行设置。要求:分别设置为“宋体”、“黑体”、“隶书”。

【提示】

font-family-----字体类型标识

2. 按以下图示编辑一个网页。



【提示】

p-----段标识
text-indent-----行缩进
vertical-align----对齐方式

3. 按以下图示编辑一张网页。



【提示】

text-decoration-----文本修饰属性
underline-----上划线
line-through-----删除线
underline-----下划线

第 14 章 设置背景和尺寸

本章所讲的背景是 CSS 中背景属性，背景属性也是复合属性，由一系列与背景相关的属性组成，如背景颜色、背景图像等，与 HTML 中的背景不同的是，在 CSS 中背景属性用 `background` 表示。背景在网页布局中非常重要，如游戏类型的网站，里面包含大量图片，而且在有背景颜色的条件下还可能显示部分文字。在网页布局中，有时需要对对象设置尺寸，根据对象的实际情况来设置对象的宽度（`width`）和高度（`height`），以固定对象的尺寸，使对象的大小在设置的范围内。

通过学习，读者会了解以下内容。

- 如何设置背景颜色；
- 如何设置图像的滚动；
- 如何设置背景图像的位置；
- 如何铺排背景图像；
- 怎样使用背景属性；
- 如何设置对象宽度；
- 如何设置对象高度；
- 宽度和高度实例。

14.1 背景的基本语法

背景是指对象在网页中显示的背景颜色或图像，如何设置背景是本节的重点，设置背景和前面学习的设置字体一样，也是复合属性，其语法是用 `background` 表示，后面接的参数如下。

```
<!--背景颜色-->
```

```
Background:background-color | background-image|background-repeat|  
background-attachment | background-position
```

复合属性 `background` 冒号接的这些参数分别表示为背景颜色、背景图像、背景图像的铺排、背景图像滚动和背景图像的定位。这些值可以是单个的也可以是多个，如果是单个的值，其他参数的值就是默认值。`background` 的默认值分别如下。

```
Background:transparent、none、repeat、scroll、0%、0%
```

背景的对象可以是 HTML 标记，也可以是 CSS 中的 ID 或 CLASS 对象。下面介绍如何设置对象的背景。

```
/*CSS样式，设置标记样式*/
```

```
01 div { background: blue no-repeat scroll 50% 60%;}  
02 body { background: url("images/bg1.gif") repeat-y; }  
03 #myid { background: url("images/bg2.gif") top; }
```

上述代码在 CSS 中对不同的对象设置不同的背景。第 1 行代码表示对所有的 DIV 层设置背景为蓝色，背景图像没有重复排铺，且有滚动，显示横坐标的 50%和纵坐标的 60%；第 2 行代码表示对<body>，即整个网页的背景由 images 文件夹下的 bg1.gif 图像显示，且只重复显示 Y 轴，其他没出现的值为默认；第 3 行代码表示层中的 ID 为 myid，其背景设置用的是 images 文件夹下的 bg2.gif 图像，顶点对齐显示图像。代码 14-1 表示了 3 种不同的背景。

代码 14-1 背景的语法

```
-----文件名：背景的语法.html-----
01 <html>
02 <head>
03 <title>背景的语法</title>
04 <style type="text/css">
05 body
06 {
07     /*设置整个网页的背景颜色*/
08     background:#ccc;
09 }
10 #box
11 {
12     /*设置宽度和高度*/
13     width:200px;
14     height:100px;
15     /*调协边框*/
16     border:1px solid red;
17     /*设置字体颜色*/
18     color:#ffffff;
19     /*设置背景颜色，引入photo.gif为背景，重复X轴*/
20     background:url("photo.gif") repeat-x;
21 }
22 </style>
23 </head>
24 <body>
25 <!--设置层-->
26 <div id="box">
27     HTML+CSS完全自学手册
28 </div>
29 </body>
30 </html>
```

【代码解析】上述代码设置两个对象的背景，首先设置整个网页（body）的背景颜色值为 CCCCCC（第 5~9 行代码），然后对层 ID 为 box 的对象（第 26 行代码），设置背景是同目录下的 photo.gif 图像按 X 轴重复显示，效果如图 14-1 所示。

注意 background 里面的参数并不要求全部设置，那些没有设置的即为默认值，如 background 为:#ccc，背景颜色为 CCCCCC。



图 14-1 设置背景

14.2 设置背景颜色

背景颜色是对象中背景的颜色值，除了可以用复合属性设置外，还可以单独设置背景颜色的属性。背景颜色的值用 `background-color` 表示，中间是一条横线，后面的单词表示颜色，结合前面的背景，即这个属性是用来设置背景颜色的，其完整语法如下。

```
background-color:transparent | color
```

背景颜色 `background-color` 有两个参数值，前者表示背景色透明，后者直接接颜色值（还记得颜色值有几种表示方法吗？前面章节介绍过）。代码 14-2 用 `background-color` 来设置不同层的不同颜色值。

代码 14-2 设置背景颜色

```
-----文件名：设置背景颜色.html-----
01 <html>
02 <head>
03 <title>设置背景颜色</title>
04 <style type="text/css">
05 body
06 {
07     /*设置网页的背景颜色*/
08     background:#bbbbbb;
09 }
10 #id1
11 {
12     /*设置id1的背景色为透明*/
13     background-color:transparent;
14 }
15 #id2
16 {
17     /*设置为一种颜色*/
18     background-color:#cccccc;
19 }
20 #id3
21 {
22     /*设置为颜色名的背景颜色*/
23     background-color:blue;
24 }
25 </style>
26 </head>
27 <body>
28 <!--层设置-->
29 <div id="id1">
30     HTML+CSS完全自学手册
31 </div>
32 <div id="id2">
33     HTML+CSS完全自学手册
34 <div>
35 <div id="id3">
36     HTML+CSS完全自学手册
```

```

37     </div>
38 </body>
39 </html>

```

【代码解析】第 8 行代码用 #bbbbbb 表示整个网页中的背景,里面 3 个层用 3 个样式表示,每个样式的背景颜色值不同: id1 的背景颜色为 transparent (透明),即默认值(第 13 行代码); id2 的背景颜色值设置为 #cccccc (第 18 行代码); id3 的颜色值设置为 blue (第 23 行代码)。效果如图 14-2 所示。

14.3 设置背景图像

上面两节讲到了背景的语法形式,也知道了背景的设置并不需要对每项都设置,可以对背景的单项属性进行设置,本节讲的是设置背景图像。首先讲的是背景颜色属性用来设置对象的背景颜色,其语法用 background-color 表示,整个名值对表示如下。

```
background-color:none | url (url)
```

background-color 表示背景颜色, none 是默认值,表示设置成没有颜色的背景;另一值用 url 表示,括号内写上图像的路径,表示以 url 里面的图像为背景显示在对象中,两值中选其中一个表示背景颜色。代码 14-3 表示背景颜色的设置。

代码 14-3 设置背景颜色

```

-----文件名: 设置背景颜色.html-----
01 <html>
02 <head>
03 <title>设置背景颜色</title>
04 <style type="text/css">
05 #id1
06 {
07     /*内边距为20像素,没有设置背景颜色*/
08     padding:20px;
09     background-color:none;
10 }
11 #id2
12 {
13     /*内边距为20像素,背景颜色为blue*/
14     padding:20px;
15     background-color:url("blue.gif");
16 }
17 #id3
18 {
19     /*背景颜色为shan.gif*/
20     padding:20px;
21     background-color:url("shan.gif");
22 }
23 </style>
24 </head>
25 <body>
26 <!--层设置-->

```



图 14-2 设置背景颜色

```

27 <div id="id1">
28     HTML+CSS完全自学手册
29 </div>
30 <div id="id2">
31     HTML+CSS完全自学手册
32 </div>
33 <div id="id3">
34     HTML+CSS完全自学手册
35 </div>
36 </body>
37 </html>

```

【代码解析】代码 14-3 表示了在 3 个层对象中分别设置不同的背景图像：ID 为 id1 的层设置的是默认值，即 none，表示没有背景图像（第 9 行代码）；ID 为 id2 的层，设置背景图像的 url 为同目录下的 blue.gif，该图像是纯色的（第 15 行代码）；ID 为 id3 的层，设置了一张有图案的图像为背景（第 21 行代码）。比较 3 个层的背景效果如图 14-3 所示。

属性 background-image 表示设置背景的图像，下面接着学习三个属性，这三个属性是对 background-image 的背景图像的补充，在不同的场合有不同的应用。

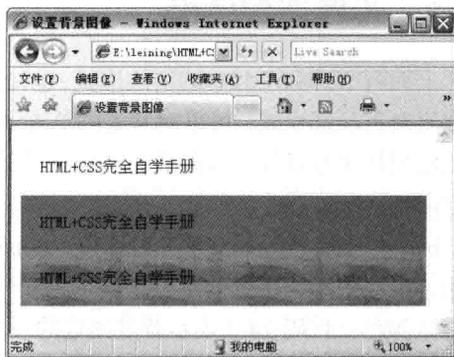


图 14-3 设置背景图像

14.3.1 设置背景图像滚动

背景图像滚动是指背景图像是随对象内容滚动还是让背景图像固定。本属性常常在博客中看到，如滑动鼠标滚轮使网页内容往下翻，而有时看到的背景图像不动，这种效果就利用了图像的滚动。

图像的滚动用 background-attachment 表示，后面接参数时的完整语法如下。

```

/*CSS样式：设置图像滚动*/
background-attachment:scroll|fixed

```

background-attachment 后面的参数 scroll 为默认值，表示图像因内容滚动而滚动；而参数 fixed 是固定的，即背景图像固定在网页中。代码 14-4 表示在背景中设置背景图像的滚动。

代码 14-4 设置背景图像滚动

```

-----文件名：设置背景图像滚动.html-----
01 <html>
02 <head>
03 <title>设置背景图像滚动</title>
04 <style type="text/css">
05 #idDiv{
06     /*设置层样式*/
07     width:160px;
08     height:100px;
09     color:#FFFFFF;
10     background:#ccc url(html.jpg) no-repeat;
11     background-attachment:fixed;

```

```

12  }
13  </style>
14  </head>
15  <body>
16  <textarea id=idDiv>
17      1111<br>
18      2222<br>
19      33333<br>
20      44444<br>
21  </textarea>
22  </body>
23  </html>

```

【代码解析】背景图像滚动前如图 14-4 所示，滚动后的效果如图 14-5 所示。



图 14-4 背景图像滚动前



图 14-5 背景图像滚动后

技巧 在博客布局中，可以设置背景图像为固定，使文本在滚动时，背景图像不会滚动。

14.3.2 设置背景图像的位置

当对象的背景图像用 `background-image` 引入时，根据背景图像大小不同，所在页面显示的位置也会不一样，现在学习的是如何设置背景图像的位置，这样就容易控制图像在对象背景中的位置了。

当然，在设置背景图像的位置前要先由 `background-image` 引入背景图像，方可控制图像位置，其语法用 `background-position` 表示，后面可接百分数或用数值加单位表示长度值，同时也可以接方位单词，如 `left`、`right` 等，两种形式表示如下。

```

background-position:length | length
background-position:position | position

```

第一种表示即 `length`，后面可接百分数或者由浮点数字和单位标识符组成的长度值。可写两个值，分别表示 X 轴和 Y 轴，如果只有一个值，该值表示 X 轴，而 Y 轴默认为 50%。

第二种表示即 `position` 后面可接 `top`、`center`、`bottom`、`left`、`right` 等方位值，分别表示的位置为：顶端、居中、底端、左边、右边。

虽然有两种表示方法，但这两种都可以表示 X 轴或 Y 轴，也可以混合使用。这两种用法如代码 14-5 所示。

代码 14-5 设置背景图像位置

-----文件名：设置背景图像位置.html-----

```
01 <html>
02 <head>
03 <title>设置背景图像位置</title>
04 <style type="text/css">
05 .box
06 {
07     /*设置高度*/
08     height:70px;
09     /*外边距顶端10像素*/
10     margin-top:10px;
11     /*设置边框*/
12     border:1px solid red;
13     /*设置背景图像*/
14     background-image:url("photo.gif");
15     /*设置背景图像不重复*/
16     background-repeat:no-repeat;
17 }
18 #id1
19 {
20     /*设置背景定位:顶端 右边*/
21     background-position:top right;
22 }
23 #id2
24 {
25     /*设置背景定位:底端 中间*/
26     background-position:bottom center;
27 }
28 #id3
29 {
30     background-position:center center;
31 }
32 #id4
33 {
34     background-position:left 50%;
35 }
36 #id5
37 {
38     background-position:30px 20%;
39 }
40 </style>
41 </head>
42 <body>
43 <!--层设置-->
44 <div class="box" id="id1"></div>
45 <div class="box" id="id2"></div>
46 <div class="box" id="id3"></div>
```

```

47 <div class="box" id="id4"></div>
48 <div class="box" id="id5"></div>
49 </body>
50 </html>

```

【代码解析】第 14 行代码用 background-image 引入一个图像，用 box 表示成一个类，可以给多个层共用，还可以把 ID 和 CLASS 放在一个层中（第 44~48 行代码），给每个层配一个 ID 控制背景位置，效果如图 14-6 所示。

设置背景图像的位置，除了设置 top、right、left 和 bottom 外，还可以设置为居中，即用 center 表示。

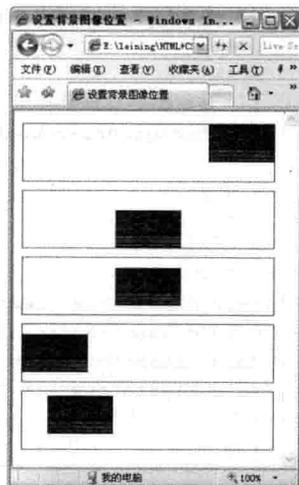


图 14-6 设置背景图像的位置

14.3.3 设置背景图像的铺排

从前面小节中可以看到，当用 background-image 引入背景图像时，默认的是重复铺排，如果在网页中不想出现重复的现象，那该怎么办呢？要想设置背景图像如何铺排，要用到 background-repeat 属性，其语法形式如下。

```

/*CSS样式：背景铺排*/
background-repeat : repeat | no-repeat | repeat-x | repeat-y

```

属性 background-repeat 后面接的值分别表示如下。

- Repeat: 默认值。背景图像在纵向和横向上平铺。
- no-repeat: 背景图像不平铺。
- repeat-x: 背景图像仅在横向上平铺。
- repeat-y: 背景图像仅在纵向上平铺。

这样就可以控制在对象中的背景如何铺排，使背景效果更好、更直观。代码 14-6 表示如何对背景图像进行铺排。

代码 14-6 设置背景图像铺排

```

-----文件名：设置背景图像铺排.html-----
01 <html>
02 <head>
03 <title>设置背景图像铺排</title>
04 <style type="text/css">
05 .box
06 {
07     /*设置层的高度等样式*/
08     height:60px;
09     margin-top:10px;
10     border:1px solid red;
11     background-image:url("photo.gif");
12 }
13 #id1
14 {

```

```

15     background-repeat: no-repeat;
16 }
17 #id2
18 {
19     /*设置背景和铺排*/
20     background-repeat: repeat-x;
21 }
22 #id3
23 {
24     background-repeat: repeat-y;
25 }
26 </style>
27 </head>
28 <body>
29 <!--层设置-->
30 <div class="box" id="id1"></div>
31 <div class="box"></div>
32 <div class="box" id="id2"></div>
33 <div class="box" id="id3"></div>
34 </body>
35 </html>

```

【代码解析】在代码 14-6 中有 4 个层：第 1 层被设置为 no-repeat，表示没有任何铺排（第 30 行代码）；第 2 层是默认的值 repeat，有铺排，即 X 轴和 Y 轴都会有（第 31 行代码）；第 3 层设置为 repeat-x，只对 X 轴铺排，其他的保持原型（第 32 行代码）；相反，第 4 层 repeat-y 只对 Y 轴铺排，即 X 轴保持不变（第 33 行代码）。效果如图 14-7 所示。

14.4 背景实例

设置背景的时候，可以用 background 复合属性对对象的背景进行设置，这样就直接接参数即可，也可以单项设置，但前提是设置不同的属性要写出其全称，并不是写 background，如设置背景图像：background-image:url("photo.gif")。

用 CSS 来控制背景颜色、背景图像，对于已引入的背景图像，如何设置其他位置，如何设置背景铺排，下面用实例一步步实现。

注意 复合属性是由多个属性组合而成的，对单个属性设置时要写出这个属性的全称。

14.4.1 设置层背景为绿色

在本实例的第一步，首先在新建的网页中，设置层对象的选择符，然后内部链接 CSS 样式，在样式中，设置好与 HTML 层标记相关联的选择符类 box，在 box 中，设置背景为绿色。下面代码表示新建一个以绿色为背景的层，如代码 14-7 所示。

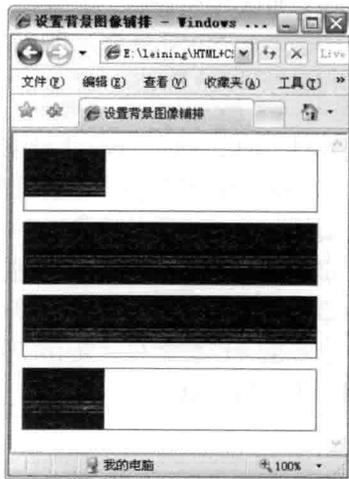


图 14-7 设置背景图像的铺排

代码 14-7 背景实例

```
-----文件名：背景实例手把手.html-----
01 <html>
02 <head>
03 <title>背景实例手把手</title>
04 <style type="text/css">
05 .box
06 {
07     /*设置高度和边框*/
08     height:100px;
09     border:1px solid red;
10     background:green;
11 }
12 </style>
13 </head>
14 <body>
15     <div class="box"></div>
16 </body>
17 </html>
```

【代码解析】本代码表示了一个高为 100 像素的层中，其背景颜色设置为绿色，外面用红色边框显示，效果如图 14-8 所示。

14.4.2 设置层背景图像

下面设置层背景图像。对于本操作，只需要在 box 的类代码后面加入 background-image 即可，用 url 表示路径。添加成功后 box 类选择符的样式代码如下。

```
01 .box
02 {
03     height:100px;
04     border:1px solid red;
05     background:green;
06     background-image:url("html.jpg");
07 }
```

【代码解析】在上述代码中，在原有的背景颜色为绿色的情况下，又设置了背景图像，其中，背景图像用 background-image 表示，url 值表示链接图像的地址，地址用引号包围（第 6 行代码）。引入背景图像的效果如图 14-9 所示。



图 14-8 背景颜色



图 14-9 引入背景图像后

14.4.3 设置背景为不铺排

下面设置背景图像不铺排。对 CSS 中的 box 操作后，多了一行代码，为 background-repeat，然后设置其值为 no-repeat，表示不向任何方向铺排。box 的代码如下。

```
01 .box
02 {
03     height:100px;
04     border:1px solid red;
05     background:green;
06     /*设置背景图像*/
07     background-image:url("html.jpg");
08     /*设置背景不铺排*/
09     background-repeat:no-repeat;
10 }
```

【代码解析】在上述代码中，原来的图像按层大小全部重复显示，即铺排效果，通过 background-repeat 来设置图像是否铺排，本步中，设置其值为 no-repeat，表示不铺排（第 9 行代码）。在网页中的效果如图 14-10 所示。

14.4.4 设置背景图像位置

背景图像在层背景中是顶部向左对齐的，为默认值，但有时为了表现出背景在对象层中的效果，使背景图像为顶部向右对齐，就得用 background-position 来设置，其后所接的值为方位值。两个值中，第一个值表示 X 轴，第二个值表示 Y 轴。如本实例中用 top right 表示顶部向右对齐，即设置了背景图位置顶端向右的位置。对 CSS 中的 box 操作后代码如下所示。

```
01 .box
02 {
03     height:100px;
04     border:1px solid red;
05     background:green;
06     background-image:url("html.jpg");
07     background-repeat:no-repeat;
08     background-position:top right;
09 }
```

【代码解析】在上述代码中，在原来的 CSS 代码基础上，添加了 background-position 的属性，表示设置背景的位置（第 8 行代码），在本实例中，设置了背景位置 X 轴为 top，Y 轴为 right，表示顶部向右的位置。效果如图 14-11 所示。



图 14-10 设置背景图像不铺排



图 14-11 设置背景图像顶部向右

上面介绍了背景的知识，在 box 中用了好几行的代码，其他内容在 14.1 节学过，background 是复合属性，也就是说用 background 可以把所有的属性值包括进去，可以表现同样的效果。图 14-11 的最终效果可以用代码 14-8 表示。

代码 14-8 背景实例

```
-----文件名：背景实例手把手.html-----
01 <html>
02 <head>
03 <title>背景实例手把手</title>
04 <style type="text/css">
05 .box
06 {
07     height:100px;
08     border:1px solid red;
09     /*设置背景和样式*/
10     background:green;
11     background-image:url("html.jpg");
12     background-repeat:no-repeat;
13     background-position:top right;
14 }
15 .otherbox
16 {
17     height:100px;
18     margin-top:10px;
19     border:1px solid red;
20     /*背景复合属性*/
21     background:green url("html.jpg") no-repeat top right;
22 }
23 </style>
24 </head>
25 <body>
26 <div class="box"></div>
27 <div class="otherbox"></div>
28 </body>
29 </html>
```

【代码解析】代码中的两个层用两个不同的 CSS 类控制，区别是前一个 box 是属性，后一个 otherbox 用复合属性 background 表示（第 21 行代码），但效果都是一样的，如图 14-12 所示。

技巧

在正式设计网页时，用 background 可以取代其他属性，这样不仅可以节省时间，还可以节省代码量。

14.5 设置尺寸

对象的尺寸都是由宽度和高度构成，本节就来学习如何在 CSS 中设置对象的宽度和高度。



图 14-12 复合属性与属性表示同样背景效果

14.5.1 设置宽度

宽度是由 width 表示，可在 CSS 中设置对象宽度，也可以在标记中使用，如仅表示宽度属性，其 height 值将根据图片源尺寸等比例缩放。宽度的语法形式如下。

```
width:auto | length
```

属性 width 有两个可选值，第一个值是 auto，根据浏览器窗口调节宽度，也是默认值；第二个值按具体的值加单位标识符表示，可以是百分比。值得注意的是，百分比是基于父对象的宽度，且不能为负值。代码 14-9 表示对象层的宽度。

代码 14-9 设置宽度

```
-----文件名：设置宽度.html-----
01 <html>
02 <head>
03 <title>设置宽度</title>
04 <style type="text/css">
05 .box
06 {
07     /*设置宽度和边框*/
08     width:200px;
09     /*边框红色1像素实线型*/
10     border:1px solid red;
11 }
12 .box2
13 {
14     /*设置宽度为100像素*/
15     width:100px;
16     /*设置边框*/
17     border:1px solid red;
18     /*设置对象与上一对象距离为10像素*/
19     margin-top:10px;
20 }
21 .box3
22 {
23     /*设置宽度为50像素*/
24     width:50px;
25     /*设置边框*/
26     border:1px solid red;
27     /*设置对象的外边距，与上面对象的距离值*/
28     margin-top:10px;
29 }
30 </style>
31 </head>
32 <body>
33 <div class="box">
34     这是第一个层，宽度为200px。
35 </div>
36 <div class="box2">
37     这是第二个层，宽度为100px。
38 </div>
39 <div class="box3">
```

```

40     这是第三个层，宽度为50px。
41 </div>
42 </body>
43 </html>

```

【代码解析】在上面代码中，存在有 3 个层，每个层的宽度不一样。为了让效果更明显，每个层的边框为红色：第 1 层设置的宽度是 200 像素（第 33 行代码），第 2 层设置的宽度是 100 像素（第 36 行代码），而第 3 层设置的宽度只有 50 像素（第 39 行代码）。最终效果如图 14-13 所示。

注意 宽度用百分数表示就不用单位标识符了，如 width:20%，表示占父对象的20%的宽度。

14.5.2 设置高度

与宽度相联系的是对象的高度，属性高度在 CSS 中用 height 表示，在中也可以用 height 表示，如仅表示 height 值时，其 width 值将根据图片源尺寸等比例缩放。设置高度的语法形式如下。

```

/*CSS代码：设置高度语法*/
height:auto | length

```

属性 height 有两个可选值，第一个值是 auto，也是默认值，根据浏览器窗口调节高度；第二个值是用具体的值加单位标识符表示，可以是百分数，百分数也是基于父对象的，不能为负值，宽度 width 可以和高度 height 一起使用来确定对象尺寸。代码 14-10 表示层的高度。

代码 14-10 设置高度

```

-----文件名：设置高度.html-----
01 <html>
02 <head>
03 <title>设置高度</title>
04 <style type="text/css">
05 .box
06 {
07     /*设宽度和边框*/
08     height:30px;
09     /*设置边框*/
10     border:1px solid red;
11 }
12 .box2
13 {
14     /*设置高度为60像素*/
15     height:60px;
16     border:1px solid red;
17     /*设置顶部外边距 */
18     margin-top:10px;
19 }
20 .box3

```



图 14-13 层的宽度对比

```

21 {
22     /*设置高度为50%*/
23     height:50%;
24     /*设置边框*/
25     border:1px solid red;
26     margin-top:10px;
27 }
28 </style>
29 </head>
30 <body>
31 <div class="box">
32     这是第一个层，高度为30px。
33 </div>
34 <div class="box2">
35     这是第二个层，宽度为60px。
36 </div>
37 <div class="box3">
38     这是第三个层，宽度为50%。
39 </div>
40 </body>
41 </html>

```

【代码解析】代码 14-10 为 3 个层设置不同的高度，前两个层设置高度为 30 和 60 像素（第 8、15 行代码），第 3 个层设置的高度为 50%（第 23 行代码），即该层高度为浏览器当前高度的 50%。设置后的效果如图 14-14 所示。

14.6 尺寸实例——设置层居中

本节介绍设置尺寸的实例——如何设置对象的宽和高。本示例中设置层的宽和高，然后对比多个层来体现层与层之间尺寸的不同。

本节通过一个实例演示本章中所学过的知识，而另外的实例用来设置在实际中所常见的问题：设置层的居中。通过这两个实例的演示，更能体现尺寸的用法。

14.6.1 尺寸实例手把手

通过下面的步骤，亲自动手来实现界面尺寸的设计。

(1) 在网页中布局多个层，并在每个层中写上文字。在没有设置层的宽度前，默认为 100%，在父对象中，有多宽就显示多宽。故需要对确定大小的层设置其尺寸。

尺寸就是由宽度和高度组成，在网页布局中，有的层不需要设置高度，由其对象中的内容决定高度。下面的代码表示在多个层中设置不同的大小，宽度用 width 表示，如代码 14-11 所示。

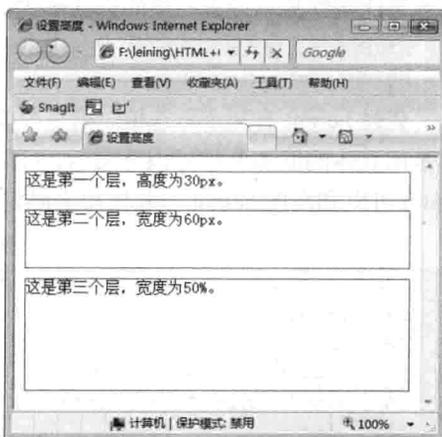


图 14-14 比较层高度

代码 14-11 尺寸实例手把手

```

-----文件名: 尺寸实例手把手.html-----
01 <html>
02 <head>
03 <title>尺寸实例手把手</title>
04 <style type="text/css">
05 /*这里是CSS代码*/
06 </style>
07 </head>
08 <body>
09 <div>
10     HTML+CSS完全自学手册---层一
11 </div>
12 <div>
13     HTML+CSS完全自学手册---层二
14 </div>
15 <div>
16     HTML+CSS完全自学手册---层三
17 </div>
18 <div>
19     HTML+CSS完全自学手册---层四
20 </div>
21 </body>
22 </html>

```

【代码解析】在页面布局中，本步只针对层进行设置（第 9~20 行代码），且 CSS 代码中没有任务代码，并没有将 CSS 与层相联系。效果如图 14-15 所示。

(2) 设置层的宽度，使每个层按 50 像素增加。确定宽度后，有时因内容太少，所体现的高度没有那么多高，这时就需要单独设置对象的高度，设置对象的高度用 `height` 表示，后面接的值表示对象的高度值。代码 14-12 表示不同对象按一定高度增加，而不同的层显示出的高度不一样。

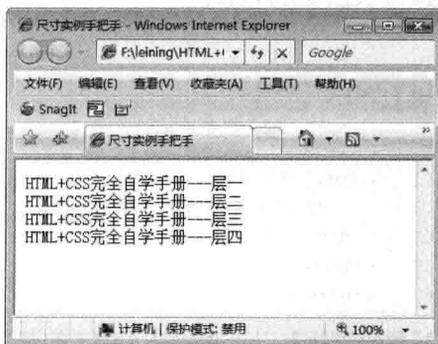


图 14-15 没有 CSS 控制的层

代码 14-12 尺寸实例手把手

```

-----文件名: 尺寸实例手把手.html-----
01 <html>
02 <head>
03 <title>尺寸实例手把手</title>
04 <style type="text/css">
05 #id1
06 {
07     /*设置宽度和边框*/
08     width:100px;
09     border:1px solid red;
10 }
11 #id2

```

```
12 {
13     width:150px;
14     border:1px solid red;
15     margin-top:10px;
16 }
17 #id3
18 {
19     /*设置宽度、边框和向顶的外边距*/
20     width:200px;
21     border:1px solid red;
22     margin-top:10px;
23 }
24 #id4
25 {
26     /*设置宽度、边框和向顶的外边距*/
27     width:250px;
28     border:1px solid red;
29     margin-top:10px;
30 }
31 </style>
32 </head>
33 <body>
34 <div id="id1">
35     HTML+CSS完全自学手册---层一
36 </div>
37 <div id="id2">
38     HTML+CSS完全自学手册---层二
39 </div>
40 <div id="id3">
41     HTML+CSS完全自学手册---层三
42 </div>
43 <div id="id4">
44     HTML+CSS完全自学手册---层四
45 </div>
46 </body>
47 </html>
```

【代码解析】以上代码在 14-11 的基础上，分别在第 8、13、20、27 行代码设置了 width（宽度）的值。

(3) 给每个层添加一个 ID，然后在 CSS 中对 ID 名称进行控制，每个层之间的宽度就显示出来了，效果如图 14-16 所示。

(4) 设置各层的高度。高度值就用 height 添加进去，在 <body></body> 中的内容不变，而变化后的样式代码如下。

```
01 <style type="text/css">
02 #id1
03 {
04     /*下面代码就是代码14-4中所解析的*/
05     width:100px;
06     border:1px solid red;
07     height:100px;
08 }
```

```

09 #id2
10 {
11     width:150px;
12     border:1px solid red;
13     margin-top:10px;
14     height:75px;
15 }
16 #id3
17 {
18     /*设置尺寸*/
19     width:200px;
20     border:1px solid red;
21     margin-top:10px;
22     height:50px;
23 }
24 #id4
25 {
26     width:250px;
27     border:1px solid red;
28     margin-top:10px;
29     height:25px;
30 }
31 </style>

```

【代码解析】每个层对象中都设置了不同的尺寸，各层以 50px 递增，上述代码在网页中表现的效果如图 14-17 所示。



图 14-16 设置层宽度

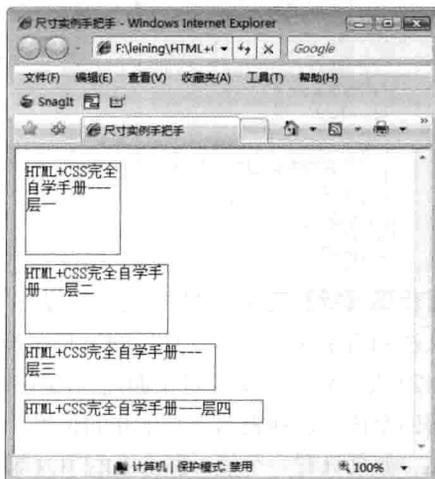


图 14-17 设置层高度

技巧 对象的宽度和高度是网页布局中常用到的属性，用来确定对象的尺寸，宽度默认的是 auto。

14.6.2 设置层居中

层居中并不同于用 align 设置 TABLE 布局的居中，前面也提到过，TABLE 布局已不是当今所选择的了，而 DIV+CSS 才是被广大网页制作者喜爱的（往往有很多用 TABLE 布局的人不太习惯

用 DIV+CSS，因为 DIV 不容易控制），但也会遇到很多问题，如层默认都是向左对齐的，但因实际需要，要设置成层居中，这时得动动脑筋了。正如初学 DIV+CSS 布局者所困惑的，如何设置才可以让层居中，本实例除介绍了设置尺寸外，还对很多初学者如何设置层居中做解析。

层的默认大小是按父对象宽度的 100%，即该层所在的范围有多宽层就显示多宽，当层在确定宽度时，也是按左对齐的，所以默认时层不会居中。为了让层居中，就得改变层的默认值，即要确定层的大小，调整层的外边距（外边距起关键作用，也是初学者不注意的地方，下一章会介绍），这样层就可以在网页中间显示了，超过层大小的，在两边用默认的白色填充，下面用步骤一步一步演示给读者。

(1) 新建层并设置大小和边框以区别层大小。

代码 14-13 设置层居中

```
-----文件名: 设置层居中.html-----
01 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
02 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
03 <html>
04 <head>
05 <title>设置层居中</title>
06 <style type="text/css">
07 #mydiv
08 {
09     /*设置尺寸大小*/
10     width:400px;
11     height:300px;
12     border:1px solid red;
13 }
14 </style>
15 </head>
16 <body>
17 <div id="mydiv">
18     HTML+CSS完全自学手册
19 </div>
20 </body>
21 </html>
```

【代码解析】第 10、11 行代码只设置了层的宽度和边框，从红色边框里可以看出层包含了文字，且在网页中居左显示，且层的大小已确定，效果如图 14-18 所示。

(2) 设置外边距使层对上面左右的距离为自动。外边距用 margin 表示，后面接的值表示距离，如果接四个值，分别表示上右下左的位置；如果只有两个值，第一个表示上下距离，第二个表示左右距离；如果只有一个值，那么上右下左都是以这个值为距离。当然还可以设置成 auto，这样表示上下左右都是自动距离，层居中的技巧也就于此。下面代码是在代码 14-12 中的 #mydiv 上加一行 margin:auto 所得，就是要使层四个边框自动对齐。

```
01 #mydiv
02 {
03     /*设置尺寸*/
04     width:400px;
05     height:300px;
06     /*设置边框*/
07     border:1px solid red;
```

```

08     /*层居中*/
09     margin:auto;
10 }

```

【代码解析】第 9 行代码解决了居中问题，这往往是初学者不注意的地方。效果如图 14-19 所示。

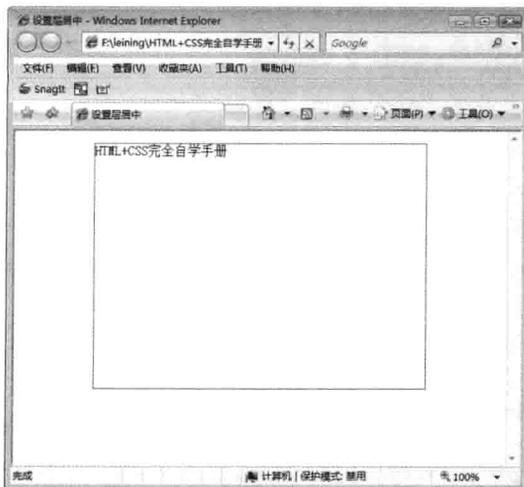


图 14-18 设置层大小与边框



图 14-19 margin 设置层居中

有时设置了 `margin:auto` 也不能居中，那要查看 `<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">` 第一行代码有没有输入。可以试着去掉这行，层就向左边移了。

14.7 小结

本章首先讲述了背景的设置，从背景的复合属性 `background` 入手，讲解了该复合属性可以包括背景的所有功能，也可以设置里面的部分参数值，让其他值保持默认，同样的效果也可以单独设置，如设置背景颜色用 `background-color`、设置背景图像用 `background-images` 等。然后讲解了如何对对象设置尺寸，尺寸包括对象的宽度 (`width`) 和高度 (`height`)，在确定对象尺寸时，就要控制其宽度与高度，当没有设置宽度时，系统自动默认浏览器宽度就是对象的宽度，同理，在没有设置高度时，对象高度就是实际占用多高的高度。

14.8 习题

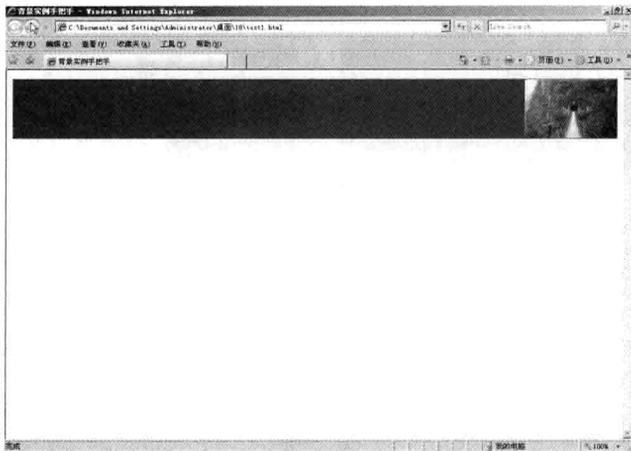
一、填空题

1. 当对象的背景图像用_____引入时，根据背景图像大小不同，所在页面显示的位置也会不一样。
2. 属性 `width` 有两个可选值：第一个值是_____，根据浏览器窗口调节宽度；第二个值是_____，表示按具体的值加单位标识符表示，可以是百分数。

3. 通过_____来设置图像是否铺排, 如果设置为_____, 表示不铺排。

二、选择题

1. 按以下网页样板做一个网页, 上只允许使用 background 属性设置背景图片。



【提示】

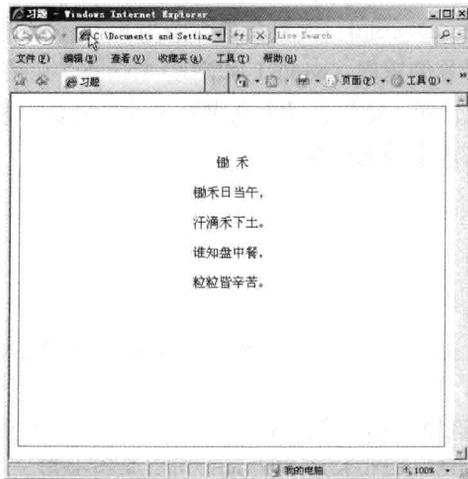
background:background-color|background-image|background-repeat|background-attachment|background-position-----背景属性设置。

2. 按第 1 题的图片样板做一个网页, 背景图片必须用 background、background-image、background-repeat 和 background-position 设置。

【提示】

background-----背景属性
 background-image-----背景图片
 background-repeat-----图片是否铺排
 background-position-----图片位置

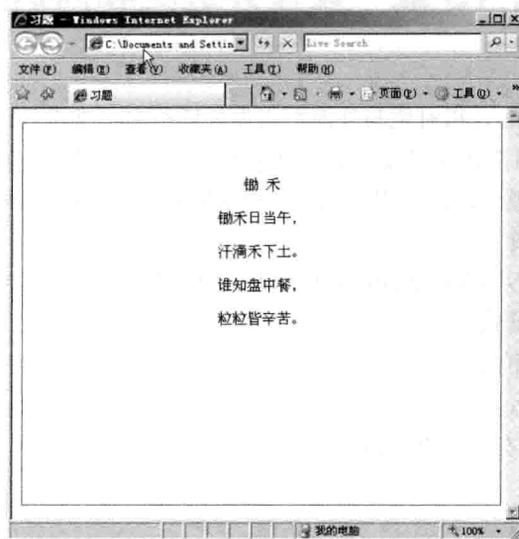
3. 按下图编辑一个网页。



【提示】

width-----宽度属性
 height-----高度属性
 border-----边框属性

4. 在第 1 题的基础上, 修改成如下图所示的网页。

**【提示】**

margin-----外边距

第 15 章 设置外边距和内边距

外边距的英文是 `margin`，指对象与外面距离或外延距离。常常可以看到网页中的层与层之间有明显的间隔，这就是网页制作人员设置了层对象中的外边距，这样两层间的距离就是设置的外延值。内边距表示对象内部填充距离，用 `padding` 表示复合属性，如 `padding-top` 表示顶端内边距、`padding-left` 表示左端内边距等。

外边距和内边距在 DIV+CSS 布局中是不可缺少的属性，有了它才可以使网页如此丰富，使网页设计者灵活地控制层的样式。

学完本章，会了解以下内容：

- 外边距的基本语法；
- 外边距的上、下、左、右外延设置；
- 外边距实例；
- 内边距的基本语法；
- 内边距顶、右、底和左端的内边距的用法；
- 内边距综合实例。

15.1 外边距的基本语法

外边距用 `margin` 表示，后面接的参数可以是数值或百分比，表示具体的精确宽度。内联对象要用 `margin` 的话，必须先设定对象的 `height` 或 `width` 属性，或者设定 `position` 属性为 `absolute`。参数还可以用字母 `auto` 表示，那么它的值是相对边的值，其语法形式如下。

```
/*CSS代码：设置外边距*/  
margin:auto | length
```

属性 `margin` 可以用数值或百分数表示，而且表示方式有四种：

(1) 后面接四个数值或百分数值，表示上右下左，即按顺时针顺序表示外延边的值，如 `margin:10px 20px 25px 30px;`，该语句表示对象的外边距到上端外延距离为 10 像素，到右端外延距离为 20 像素，到底端外延距离为 25 像素，最后到左端外延距离值为 30 像素。

(2) 后面只接一个数值或百分数值，表示的是四周的外延值。如 `margin:20px;`，该语句表示对象的上、右、下、左的外延值都是 20 像素。

(3) 后面只接两个数值或百分数时，前一数值或百分数表示上下的外延值，后一数值或百分数表示左右的外延值。这种情况在上下和左右值相同时使用方便些。

(4) 后面只接三个数值或百分数时，第一个数值或百分数表示上端的外延值，而第二个数值或百分数表示左右的外延值，最后一个表示下端的外延值。

通过四种方式表示 margin 的参数值，在实际应用中就带来很大的方便，特别是只接一个数值或百分数表示外延值时。为了实践上面四种方式，代码 15-1 表示了这四种方式的区别，请仔细观察代码间的不同。

代码 15-1 margin 基本语法

```
-----文件名: 字体颜色.html-----
01 <html>
02 <head>
03 <title>margin基本语法</title>
04 <style type="text/css">
05 html,body
06 {
07     /*设置外边距为0*/
08     margin:0px;
09     /*设置内边距为0*/
10     padding:0px;
11 }
12 div
13 {
14     /*设置宽度和边框*/
15     width:200px;
16     border:1px solid red;
17 }
18 #m1
19 {
20     /*设置外边距四边为20像素*/
21     margin:20px;
22 }
23 #m2
24 {
25     /*设置外边距上下为35像素,左右为40像素*/
26     margin:35px 40px;
27 }
28 #m3
29 {
30     /*设置外边距上为20像素,左右为30像素,下为40像素*/
31     margin:20px 30px 40px;
32 }
33 #m4
34 {
35     /*设置上、右、下、左的外边距*/
36     margin:20px 10px 5px -20px;
37 }
38 </style>
39 </head>
40 <body>
41     <div id="m1">margin为1个值时</div>
42     <div id="m2">margin为2个值时</div>
43     <div id="m3">margin为3个值时</div>
44     <div id="m4">margin为4个值时</div>
45 </body>
46 </html>
```

【代码解析】代码 15-1 中，用 4 个 ID 表示不同的 4 个层，在样式表中控制着这 4 个 ID：第 1 个 ID 对象 m1 属性 margin 只有一个值，表示四个外延距离为 20 像素（第 21、41 行代码）；第 2 个 ID 对象 m2 属性 margin 有 2 个值，前面的值 35 像素表示上和下的外延距离为 35 像素，而后面的值 40 像素表示左右的外延距离为 40 像素（第 26、42 行代码）；第 3 个 ID 对象 m3 属性 margin 有 3 个值，前面的值表示上面的外延距离为 20 像素，中间的值表示左右外延值为 30 像素，后面的值是下面的外延距离为 40 像素（第 31、43 行代码）；第 4 个 ID 对象 m4 的 4 个值分别表示上、右、下、左的值分别为 20 像素、10 像素、5 像素和 -20 像素。注意，左边的值是负值。在这里也可以用负值表示（第 36、44 行代码）。基本语法的演示效果如图 15-1 所示。



图 15-1 margin 的基本语法

注意

负值可以超出父对象的尺寸范围，如图 15-1 中的第 4 层在父对象 body 中向左移了 20 像素，而超过部分刚好被网页隐藏掉了。

15.2 外边距的四边设置

尽管一个 margin 可表示四个边的外延值，但别忘了还有其他属性可以表示里面的各个边的外延值，如上端可以用 margin-top 表示，右端可用 margin-right 表示等。与 margin 不同的是，这些不是复合属性，后接的只是一个参数值，表示对应的外延值。下面从这四个不同方向的外边距来介绍其属性。

15.2.1 顶端外边距的用法

属性 margin 是复合属性，其值可以表示 margin-top 的值，二者效果是一样的，但有时只表示上面的外延距离时，可以只用 margin-top 属性表示，所以学好 margin-top 是用好 margin 的基础。

从字母中可以看到，margin-top 表示的是顶端外边距，也称上端外延距离，是对象与父对象的上端距离。完整的语法形式如下。

```
/*CSS代码：设置顶部外延距离*/
margin-top:auto|length
```

margin-top 的值可接 auto，表示相对边的值，另一值可以是数值或百分比，表示一个数值的距离，百分比是基于父对象的高度。代码 15-2 表示 margin-top 的用法。

代码 15-2 margin-top 用法

```
-----文件名：margin-top用法.html-----
01 <html>
02 <head>
03 <title>margin-top用法</title>
04 <style type="text/css">
05 #m1
```

```

06  {
07      /*设置宽度和边框*/
08      width:200px;
09      border:1px solid red;
10      margin-top:100px;
11  }
12  </style>
13  </head>
14  <body>
15      <div id="m1">margin-top用法</div>
16  </body>
17  </html>

```

【代码解析】第 10 行代码对 ID 为 m1 的层设置了 margin-top 值，即外边距顶部距离为 100 像素，其他方位的距离是默认值，所以层对象与顶端的距离为 100 像素，且用边框显示出效果，如图 15-2 所示。

15.2.2 右端外边距的用法

与 margin-top 属性一样，margin-right 的属性也是在其后接距离值，表示右边的外延距离，距离值可以为正值或负值。完整的语法如下。

```

/*CSS代码：设置右边的外延距离*/
margin-right:auto|length

```

参数值 auto 表示自适应，length 的值是精确值，表示右边的外延值距离，可以为正值或负值。代码 15-3 表示 margin-right 的用法。

代码 15-3 margin-right 的用法

```

-----文件名：margin-right用法.html-----
01  <html>
02  <head>
03  <title>margin-right用法</title>
04  <style type="text/css">
05  #box
06  {
07      width:300px;
08      margin:auto;
09      border:1px solid blue;
10  }
11  #m1
12  {
13      /*设置边框*/
14      border:1px solid red;
15      margin-right:100px;
16  }
17  </style>

```



图 15-2 margin-top 用法

```

18 </head>
19 <body>
20 <div id="box">
21 这里用层box包含了层m1<br>
22 <div id="m1">margin-right用法</div>
23 上面是层m1右外延距离的效果，margin-right的值为100像素。
24 </div>
25 </body>
26 </html>

```

【代码解析】第5~10行代码用一个容器层 box 包含了 m1 层，层 box 的宽度是 300 像素，而 m1 层的宽度没有设置，margin-right 设置为 100 像素（第 15 行代码），表示层 m1 的右端到父对象 box 层的右端距离值，即右外延距离是 100 像素，效果如图 15-3 所示。

15.2.3 底端外边距的用法

属性 margin-bottom 也是 margin 复合属性之一，表示底部的外延距离。与其他属性一样，其语法表示如下。

```

/*CSS代码：设置底部的外延距离*/
margin-bottom:auto|length

```

参数值 auto 表示自适应，length 的值是精确值，表示底部的外延值距离，可以为正值或负值。代码 15-4 表示 margin-bottom 的用法。

代码 15-4 margin-bottom 用法

```

-----文件名：margin-bottom用法.html-----
01 <html>
02 <head>
03 <title>margin-bottom用法</title>
04 <style type="text/css">
05 #box
06 {
07     /*设置高度和边框*/
08     height:200px;
09     border:1px solid blue;
10 }
11 #m1
12 {
13     width:200px;
14     border:1px solid red;
15     margin-bottom:100px;
16 }
17 </style>
18 </head>
19 <body>
20 <div id="box">
21     <p>下面是margin-bottom用法<br>margin-bottom是m1到下端的距离</p>

```

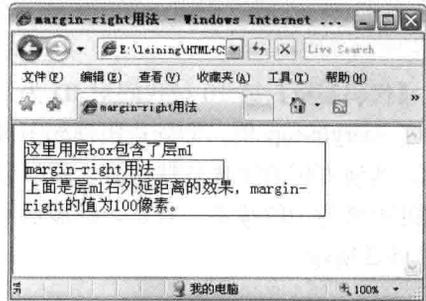


图 15-3 margin-right 用法

```

22     <div id="m1">margin-bottom用法</div>
23 </div>
24 </body>
25 </html>

```

【代码解析】代码 15-4 中有两个层。一个是容器层 box，而层 m1 是被 box 包围着的。层 box 高度为 200 像素(第 8 行代码)，层 m1 的 margin-bottom 值为 100 像素(第 15 行代码)，效果如图 15-4 所示。

15.2.4 左端外边距的用法

属性 margin-left 也是 margin 复合属性之一，表示左边的外延距离。与其他属性一样，其语法表示如下。

```

/*CSS代码：设置左边的外延距离*/
margin-left:auto|length

```

参数值 auto 表示自适应，length 的值是精确值，表示左边的外延值距离，可以为正值或负值。代码 15-5 表示 margin-left 的用法。

代码 15-5 margin-left 的用法

```

-----文件名:margin-left用法.html-----
01 <html>
02 <head>
03 <title>margin-left用法</title>
04 <style type="text/css">
05 #box
06 {
07     /*设置宽度为300像素*/
08     width:300px;
09     /*设置外边距*/
10     margin:auto;
11     /*设置边框样式*/
12     border:1px solid blue;
13 }
14 #m1
15 {
16     border:1px solid red;
17     margin-left:100px;
18 }
19 </style>
20 </head>
21 <body>
22 <div id="box">
23 这里用层box包含了层m1<br>
24     <div id="m1">margin-left用法</div>

```

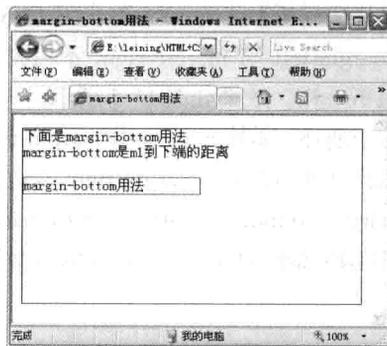


图 15-4 margin-bottom 用法

```

25     上面是层m1右外延距离的效果，margin-left的值为100像素。
26 </div>
27 </body>
28 </html>

```

【代码解析】代码 15-5 与代码 15-3 相似，只是 margin-left 与 margin-right 不相同，所以效果也不一样。效果如图 15-5 所示。

综上所述，属性 margin 由 margin-top 顶部外延距离（外边距）、margin-right 右边外延距离、margin-bottom 底部外延距离和 margin-left 左边外延距离组成的，且其后面的参数设置都相同。



图 15-5 margin-left 的用法

15.3 外边距实例——设置层与层的距离

本节通过实例来说明外边距 margin 的用法。本属性在网页中经常用到，对网页布局起着非常重要的作用。若要掌握本属性的用法，多实践就会有很好的认识，从而能够更好地使用。

下面的实例演示外边距的各知识点用法，从不同的对象比较各个属性的区别。实例中以红色边框为边界，以区分各个对象。这种效果常常在网页中用来间隔层对象，使各个层之间有完美的距离，浏览起来有层次感。

15.3.1 层与层的距离效果

下面列出整体效果代码和效果图片。为了体验外边距的用法，本实例先列出网页代码，然后再一步步分析其代码内容。代码 15-6 是本实例的完整代码。

代码 15-6 margin 实例—设置层与层的距离

```

-----文件名：margin实例—设置层与层的距离.html-----
01 <html>
02 <head>
03 <title>margin实例—设置层与层的距离</title>
04 <style type="text/css">
05 #mybody
06 {
07     /*设置网页中的层容器*/
08     /*宽为778像素*/
09     width:778px;
10     /*容器层上、右、下、左的外边距为自动*/
11     margin:auto;
12     /*字体大小为14像素*/
13     font-size:14px;
14     /*设置容器层边框为0*/
15     border:0;
16 }
17 div
18 {

```

```
19      /*设置所有的层为1像素为红色实线*/
20      border:1px solid red;
21  }
22  #header
23  {
24      /*设置头部层属性*/
25      /*设置上、右、下、左外边距都为5像素*/
26      margin: auto;
27      /*设置宽度为100%*/
28      width:100%;
29      /*设置高度为80像素*/
30      height:80px;
31  }
32  #contentleft
33  {
34      /*设置尺寸大小*/
35      width:180px;
36      height:450px;
37      /*设置层向左漂移*/
38      float:left;
39  }
40  #contentcenter
41  {
42      width:410px;
43      height:450px;
44      /*设置层向左漂移*/
45      float:left;
46      margin-left:6px;
47  }
48  #contentright
49  {
50      width:170px;
51      height:450px;
52      float:right;
53      margin-left:6px;
54  }
55  #bottom
56  {
57      width:100%;
58      /*设置外边距上下为5像素，左右为自动*/
59      margin:5px auto;
60      height:60px;
61  }
62  </style>
63  </head>
64  <body>
65  <div id="mybody">
66      <div id="header">
67          这是网页头部，放一些logo，宣传栏之类
68      </div>
69      <div id="contentright">
```

```

70     内容右栏
71     </div>
72     <div id="contentleft">
73     网页内容左栏
74     </div>
75     <div id="contentcenter">
76     内容中间栏
77     </div>
78     <div id="bottom">
79     网页底部
80     </div>
81 </div>
82 </body>
83 </html>

```

【代码解析】在代码 15-6 中，用网页的布局风格设置了各层之间的外边距，即用 margin 表示，使各层之间的距离间隔一致，体现出各层之间的层次性，其中设置 header 层对象外边距 margin 值为 auto，表示上、右、下、左的外延距离都是 auto（第 26 行代码），在网页中表现出来的是居中显示。效果如图 15-6 所示。

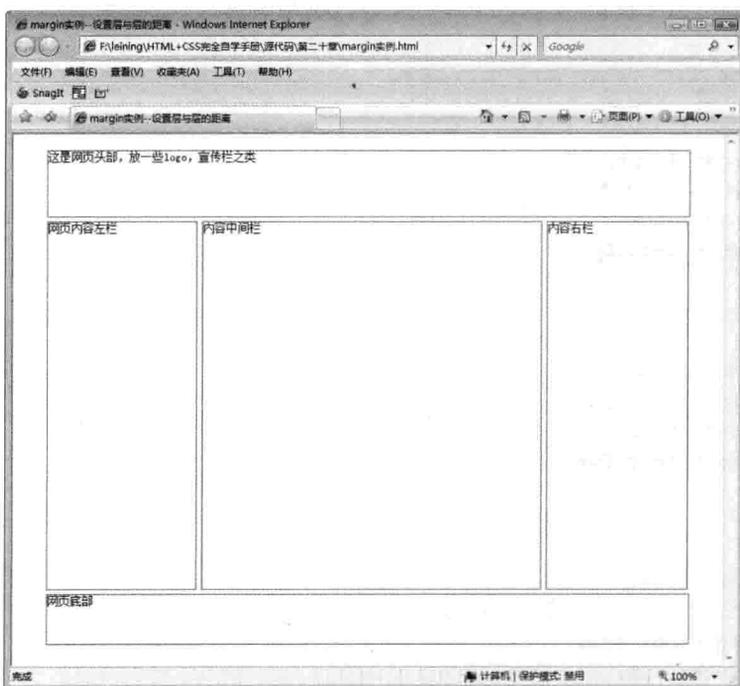


图 15-6 整体效果图

15.3.2 对层布局

先定好五个层，分别表示头部、中部、左栏、右栏和底部。新建一张网页，然后定好层的 ID 和文字内容，以下内容只起显示作用，先确定有哪些层，然后再对层进行分步布局，如代码 15-7 表示。

代码 15-7 margin 实例—设置层与层的距离

```

-----文件名: margin实例—设置层与层的距离.html-----
01 <html>
02 <head>
03 <title>margin实例—设置层与层的距离</title>
04 </head>
05 <body>
06 <div id="mybody">
07     <div id="header">
08         这是网页头部, 放一些logo, 宣传栏之类
09     </div>
10     <div id="contentright">
11         内容右栏
12     </div>
13     <div id="contentleft">
14         网页内容左栏
15     </div>
16     <div id="contentcenter">
17         内容中间栏
18     </div>
19     <div id="bottom">
20         网页底部
21     </div>
22 </div>
23 </body>
24 </html>

```

【代码解析】代码 15-7 中, 通过给层设置文字和 ID, 在没有 CSS 控制下, 显示原型, 这时的 ID 不会起作用, 因为根本没有 CSS 可以控制, 效果如图 15-7 所示。

15.3.3 设置层的尺寸

下面设置各层的尺寸。这时层的内容和文字都确定了, 每一步只改变 CSS 样式即可。如何设置好各层的尺寸, 就要用到前面学习的 width 和 height 属性。为了突出层的大小, 设置层为红色边框。代码 15-8 表示设置好各层的大小属性, 只表示样式代码。

代码 15-8 margin 实例—设置层与层的距离

```

01 <style type="text/css">
02 #mybody
03 {
04     /*设置网页中层容器*/
05     /*宽为778像素*/
06     width:778px;
07     /*容器层上右下左的外边距为自动*/
08     margin:auto;
09     /*字体大小为14像素*/

```

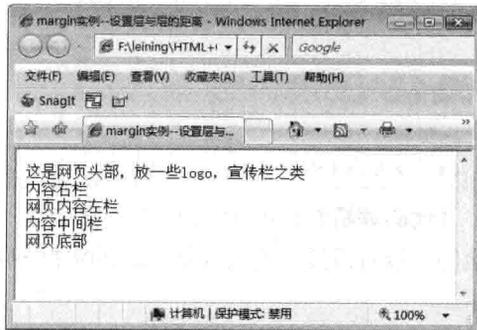


图 15-7 设置层 ID

```

10     font-size:14px;
11     /*设置容器层边框为0*/
12     border:0;
13 }
14 div
15 {
16     /*设置所有的层为1像素为红色实线*/
17     border:1px solid red;
18 }
19 #header
20 {
21     /*设置头部层属性*/
22     /*设置宽度为100%*/
23     width:100%;
24 }
25 #contentleft
26 {
27     /*设置中间部分的左边*/
28     width:180px;
29 }
30 #contentcenter
31 {
32     width:410px;
33 }
34 #contentright
35 {
36     width:170px;
37 }
38 #bottom
39 {
40     width:100%;
41     height:60px;
42 }
43 </style>

```

【代码解析】代码 15-8 设置了层布局，各层在网页中，CSS 只是对宽度和高度进行约束，对其他的并没有设置，所表现的为层的宽度不同方式的设置，体现出层与层之间的布局，如图 15-8 所示。



图 15-8 设置各层尺寸

15.3.4 设置层漂移

下面设置中间部分都往左漂移,使中间部分的各层之间能平行布局。在层与层之间,不管是否确定层的大小,默认情况下,层与层之间不会平行对齐。换言之,要想设置两层对齐,得用 float 设置层与层之间的漂移。代码 15-9 在前面代码的基础上,设置了中间部分层的漂移。

代码 15-9 margin 实例一设置层与层的距离

```
01 <style type="text/css">
02 #mybody
03 {
04     /*设置网页中的层容器*/
05     /*宽为778像素*/
06     width:778px;
07     /*容器层上、右、下、左的外边距为自动*/
08     margin:auto;
09     /*字体大小为14像素*/
10     font-size:14px;
11     /*设置容器层边框为0*/
12     border:0;
13 }
14 div
15 {
16     /*设置所有的层为1像素为红色实线*/
17     border:1px solid red;
18 }
19 #header
20 {
21     /*设置头部层属性*/
22     /*设置上、右、下、左外边距都为5像素*/
23     margin: auto;
24     /*设置宽度为100%*/
25     width:100%;
26     /*设置高度为80像素*/
27     height:80px;
28 }
29 #contentleft
30 {
31     /*设置层宽度为180像素,高度为450像素*/
32     width:180px;
33     height:450px;
34     float:left;
35 }
36 #contentcenter
37 {
38     width:410px;
39     height:450px;
40     float:left;
41 }
42 #contentright
43 {
44     width:170px;
```

```

45     height:450px;
46     float:right;
47 }
48 #bottom
49 {
50     width:100%;
51     height:60px;
52 }
53 </style>

```

【代码解析】用 float 漂移设置中间部分层流向，第 34 行代码使 contentleft 层往左漂移，同理，第 40 行代码设置 contentcenter 层也往左漂移。和第 42~47 行代码的 contentright 层不同的是，contentright 层是向右漂移的，而在 HTML 代码中，contentright 在 contentleft 和 contentcenter 前面。漂移后的效果如图 15-9 所示。



图 15-9 设置中间部分的高和漂移

15.3.5 设置层与层之间的间隔

设置关键一步，使各层之间有 5 像素的间隔。因为层与层之间默认时，中间是没有间隔的，如在两个边框为 1 像素的层中，默认情况下，层与层交界处是没有空隙的。从图 15-9 中可以看到，中间部分的层与页头和页脚的间隔为 0，所以看起来是 2 像素，其实不然，这是因为两个层的边框靠到一起。

如果设置层与层之间的间隔，就会使得层布局有模块感，更美观。设置层与层的间隔或者说是距离会用到本章所学的外边距。外边距用 margin 表示，其后所接的属性值表示对象的上、下、左、右的外边距，从效果上看表示上、下、左、右的间隔值。代码 15-10 设置了层与层之间的间隔。

代码 15-10 margin 实例—设置层与层的距离

```
01 #mybody
02 {
03     /*设置网页中的层容器*/
04     /*宽为778像素*/
05     width:778px;
06     /*容器层上、右、下、左的外边距为自动*/
07     margin:auto;
08     /*字体大小为14像素*/
09     font-size:14px;
10     /*设置容器层边框为0*/
11     border:0;
12 }
13 div
14 {
15     /*设置所有的层为1像素为红色实线*/
16     border:1px solid red;
17 }
18 #header
19 {
20     /*设置头部层属性*/
21     /*设置上、右、下、左外边距都为5像素*/
22     margin: auto;
23     /*设置宽度为100%*/
24     width:100%;
25     /*设置高度为80像素*/
26     height:80px;
27     margin:5px 0px;
28 }
29 #contentleft
30 {
31     /*设置尺寸*/
32     width:180px;
33     height:450px;
34     float:left;
35 }
36 #contentcenter
37 {
38     width:410px;
39     height:450px;
40     float:left;
41     margin-left:6px;
42 }
43 #contentright
44 {
45     width:170px;
46     height:450px;
47     float:right;
48     margin-left:6px;
49 }
50 #bottom
51 {
```

```

52     width:100%;
53     margin:5px auto;
54     height:60px;
55 }
56 </style>

```

【代码解析】第 53 行代码设置底部的顶端外延距离值和头部的下端外延距离值，使对象与其他层对象体现出外延距离，效果如图 15-10 所示。

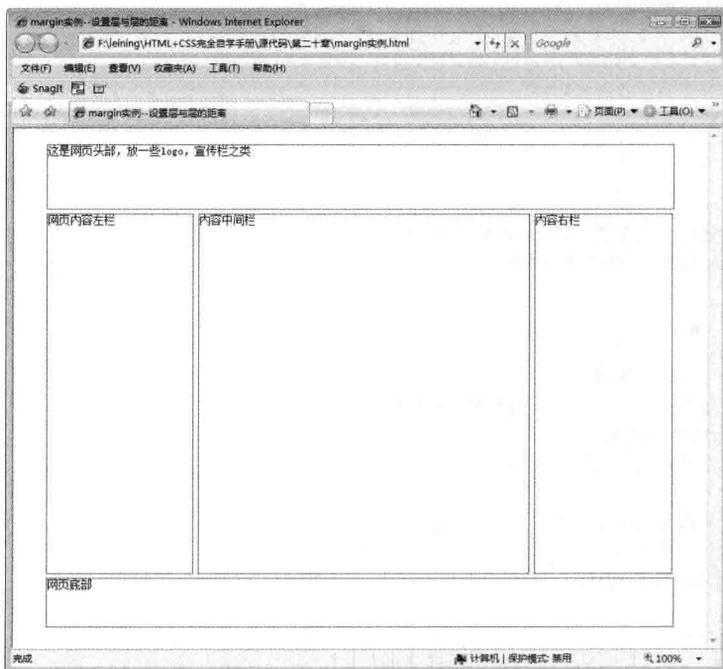


图 15-10 整体效果显示

15.4 内边距的基本语法

内边距用在对象内部，表示对象中的内部填充距离，与外边距的对象外部外延距离相反，但都是表示距离值。内边距语法用 `padding` 表示，是复合属性，其完整语法形式如下。

```
padding:length
```

参数 `length` 表示百分比或数值，百分比基于父对象的宽度，不能为负值，且不能为 `auto`，这点与 `margin` 不同（`margin` 的参数中可以是负值和 `auto`）。下面通过背景颜色来演示内边距 `padding` 的用法，代码 15-11 表示 `padding` 复合属性的用法。

代码 15-11 padding 基本语法和用法

```

-----文件名: padding基本语法和用法.html-----
01 <html>
02 <head>
03 <title>padding基本语法和用法</title>
04 <style type="text/css">
05 #m1

```

```

06  {
07      /*设置对象宽度和高度*/
08      width:200px;
09      height:150px;
10      /*设置背景颜色*/
11      background:#ccc;
12      /*设置内边距*/
13      padding:20px;
14  }
15  #m2
16  {
17      /*设置高度和背景*/
18      height:100%;
19      background:green;
20  }
21  </style>
22  </head>
23  <body>
24  <div id="m1">
25      <div id="m2">padding用法: <br>
26      绿色以外的颜色是表示内边距区域。
27      </div>
28  </div>
29  </body>
30  </html>

```

【代码解析】代码 15-11 中有两个层对象，m1 是外围的层，背景颜色是#cccccc，其内边距 padding 被设置为 20 像素（第 13 行代码），当 m1 层对象中有其他对象时，内边距的效果就显示出来了，刚好里面的层对象 m2 背景为绿色，当 m2 被包含在父对象 m1 中时，内边距的距离就是#cccccc 的区域，如图 15-11 所示。

15.5 内边距的四边属性

上一节介绍了复合属性 padding 的语法和用法，还有一种重要的用法没有介绍到，就是组成复合属性中的各个单独属性。padding 由顶、右、底和左端组成，表示内边距的四边。它们的完整形式如下。

```

/*内边距的各分属性值*/
padding-top:length;
padding-right:length;
padding-bottom:length;
padding-left:length;

```

各属性的参数值与 padding 的一样，是百分数或数值（细节请看上一节），但表示的是各个方向位置：padding-top 表示顶端内边距，对象内部顶端到子对象的距离（如层中再包括层）；padding-right 表示右端内边距，对象内部右端到子对象的距离；padding-bottom 表示底端内边距，对象内部底端到子对象的距离；padding-left 表示左端内边距，对象内部左端到子对象的距离。

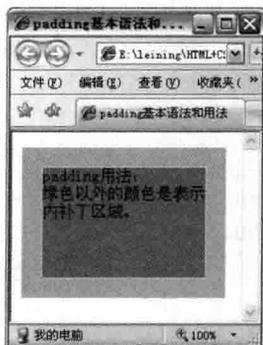


图 15-11 padding 基本语法和用法

虽然一个 padding 复合属性可以包含上面四个属性，但是还有必要学好这四个属性的用法，下面分别介绍这些属性。

15.5.1 顶端内边距的用法

当知道了 padding 属性后，学习 padding-top 就容易了，padding-top 是表示顶端内边距，使用这个属性对其他单独属性并没有影响，它只表示顶端这个方向的距离，其语法如代码 15-12 所示。

代码 15-12 padding-top 用法

```
-----文件名: padding-top用法.html-----
01 <html>
02 <head>
03 <title>padding-top用法</title>
04 <style type="text/css">
05 #m1
06 {
07     width:200px;
08     height:150px;
09     /*设置内边距顶端值*/
10     padding-top:30px;
11     background:red;
12 }
13 #m2
14 {
15     /*设置背景和高度*/
16     background:green;
17     height:100%;
18 }
19 </style>
20 </head>
21 <body>
22 <div id="m1">
23     <div id="m2">padding-top用法</div>
24 </div>
25 </body>
26 </html>
```

【代码解析】在代码 15-12 中，有两个层 m1 和 m2，m1 是容器层包含首子层 m2，对容器层设置的顶端内边距为 30 像素（第 10 行代码），且确定了层的宽和高，背景为红色，而子层 m2 的高度是 100%，由父对象 m1 的高度决定的，由于 m1 的高度是 150 像素（第 8 行代码），故 m2 的 100% 也为 150 像素（第 17 行代码），背景为绿色，红色的区域就是要得到的顶端内边距距离。效果如图 15-12 所示。

注意

padding-top 中间不是下划线，如果写错则在网页中不会出现所要的效果，务必要检查输入是否正确。



图 15-12 padding-top 用法

15.5.2 右端内边距的用法

与 padding-top 相同的是，padding-right 也表示内边距，只不过方向是右端，其语法形式如下。

```
/*CSS样式：设置右端内边距值*/
padding-right : length
```

参数形式在上一节介绍过，在这里不再重复讲解，下面通过 padding-right 的实例来演示右端内边距的用法，如代码 15-13 所示。

代码 15-13 padding-right 用法

```
-----文件名：padding-right用法.html-----
01 <html>
02 <head>
03 <title>padding-right用法</title>
04 <style type="text/css">
05 #m1
06 {
07     width:200px;
08     height:150px;
09     padding-right:30px;    /*设置内边距右端值*/
10     background:red;
11 }
12 #m2
13 {
14     background:green;
15     height:100%;
16 }
17 </style>
18 </head>
19 <body>
20 <div id="m1">
21     <div id="m2">padding-right用法</div>
22 </div>
23 </body>
24 </html>
```

【代码解析】在背景为红色且右端内边距为 30 像素（第 9 行代码）的父对象中，存在着以绿色为背景的子层，高度为父对象的高度，效果如图 15-13 所示。

注意

各分属性的设置与padding一样，只不过padding-right所表达的是右端的内边距，同时，其他分属性都一样。

15.5.3 底端内边距的用法

与其他属性一样，padding-bottom 表示的是底部内边距，用来设置底部的内边距，而其他方位的属性不受影响，其语法形式如下。

```
/*CSS样式：设置底端内边距值*/
padding-bottom : length
```



图 15-13 padding-right 用法

padding-bottom 的用法与其他属性相同，但所表达的是不同的方位。代码 15-14 演示了 padding-bottom 的用法。

代码 15-14 padding-bottom 用法

```
-----文件名: padding-bottom用法.html-----
01 <html>
02 <head>
03 <title>padding-bottom用法</title>
04 <style type="text/css">
05 #m1
06 {
07     width:200px;
08     height:150px;
09     padding-bottom:30px; /*设置内边距底端值*/
10     background:red;
11 }
12 #m2
13 {
14     background:green; /*设置背景为绿色*/
15     height:100%;
16 }
17 </style>
18 </head>
19 <body>
20 <div id="m1">
21     <div id="m2">padding-bottom用法</div>
22 </div>
23 </body>
24 </html>
```

【代码解析】在代码 15-14 中，padding-bottom 表示了对象内边距的位置是底部（第 9 行代码），设置背景层的颜色为红色，所以红色出现的位置刚好是内边距的处理位置，证明 padding-bottom 只对底部起作用，效果如图 15-14 所示。

15.5.4 左端内边距的用法

与其他属性一样，padding-left 表达的是左边内边距，用来设置左边的内边距，而其他方位的属性不受影响，其语法形式如下。

```
/*CSS样式：设置左边内边距值*/
padding-left:length
```

padding-left 的用法与其他属性相同，但所表达的是不同的方位。代码 15-15 演示了 padding-left 在网页中的用法。



图 15-14 padding-bottom 用法

代码 15-15 padding-left 用法

```

-----文件名: padding-left用法.html-----
01 <html>
02 <head>
03 <title>padding-left用法</title>
04 <style type="text/css">
05 #m1
06 {
07     /*设置宽度和高度*/
08     width:200px;
09     height:150px;
10     /*设置左内边距*/
11     padding-left:30px;    /*设置内边距左端值*/
12     /*设置背景颜色*/
13     background:red;
14 }
15 #m2
16 {
17     background:green;
18     /*设置高度为100%*/
19     height:100%;
20 }
21 </style>
22 </head>
23 <body>
24 <div id="m1">
25     <div id="m2">padding-left用法</div>
26 </div>
27 </body>
28 </html>

```

【代码解析】padding-left 表示了对象内边距的位置是左边（第 11 行代码），设置背景层的颜色为红色，所以红色出现的位置刚好是内边距的处理位置，证明 padding-left 只对左边位置起作用，效果如图 15-15 所示。

15.6 内边距实例

内边距用 padding 表示，表示对象的内部距离值，属性为复合属性。该复合属性由顶端内边距 padding-top、右端内边距 padding-right、下端内边距 padding-bottom 和左端内边距 padding-left 组成，不管是复合属性还是表示方位的属性，其所接的值都是一样的，即接数值表示内边距的距离值，数值越大，距离就越远。

在实际应用中，主要在层对象中设置内边距。为了体现层边框与文本的距离值，即设置该层的内边距，这样就表示层边框与文本间有一定距离的间隔，这间隔就是内边距的距离值。本实例演示内边距的用法，通过步骤来说明单属性、复合属性的用法，请读者注意各代码的作用和区别。

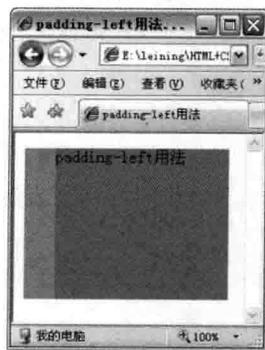


图 15-15 padding-left 用法

15.6.1 设置顶部内边距

本节从内边距的顶部入手，设置单独顶部的内边距值。顶部的内边距设置并不影响其他方面的内边距，因为顶部内边距 `padding-top` 只对顶部起作用。

在输入好网页的基本语句后，在网页内部即 `<head></head>` 中插入样式内容，定好各个层和样式文件。代码 15-16 表示实例的顶部内边距。

代码 15-16 内边距实例

```
-----文件名：内边距实例手把手.html-----
01 <html>
02 <head>
03 <title>内边距padding实例</title>
04 <style type="text/css">
05 /*设置层标记的样式*/
06 div
07 {
08     /*设置层的边框*/
09     border:1px solid red;
10     width:200px;
11     /*设置层标记的外边距*/
12     margin-top:5px;
13 }
14 #p1
15 {
16     /*设置内边距值为20像素*/
17     padding-top:20px;
18     /*设置高度和背景颜色*/
19     height:20px;
20     background:blue;
21 }
22 #pp1
23 {
24     /*设置背景颜色和高度*/
25     background:orange;
26     height:30px;
27 }
28 </style>
29 </head>
30 <body>
31 <div id="p1">
32     <div id="pp1">padding-top设置</div>
33 </div>
34 </body>
35 </html>
```

【代码解析】第 22~27 行代码设置了层 `p1` 包含子层 `pp1`，且容器层的背景颜色为 `blue`（第 20 行代码），而子层的背景颜色为 `orange`（第 25 行代码），但在容器层中设置的顶端内边距

为 20 像素（第 17 行代码），从所展现的效果可以看出，顶部出现的背景刚好是 padding-top 设置的值，如图 15-16 所示。

15.6.2 设置其他位置的内边距

在上一节中，通过设置顶部内边距的值（顶部内边距用 padding-top 表示），表示顶部边缘与对象内容的距离，在 15.5.1 节中，设置过的顶部内边距在网页中显示的是顶部的距离值，对其他方位的距离没有影响。在本节中，完成其他三个方位的设置，分别是 padding-right、padding-bottom 和 padding-left 的值。通过这三个方位的内边距的设置，就可以在对象中表现出用四个属性设置的四个方位的内边距。为了让层与对象之间容易区分，各层还是用颜色区别开来，如代码 15-17 所示。



图 15-16 设置 padding-top

代码 15-17 内边距实例

```
-----文件名：内边距实例手把.html-----
01 <html>
02 <head>
03 <title>内边距padding实例</title>
04 <style type="text/css">
05 /*设置标记DIV样式*/
06 div
07 {
08     border:1px solid red;
09     width:200px;
10 }
11 #p1
12 {
13     /*顶端内边距设置为20像素*/
14     padding-top:20px;
15     background:blue;
16 }
17 #pp1
18 {
19     /*设置背景颜色和高度*/
20     background:orange;
21     height:50px;
22 }
23 #p2
24 {
25     /*设置右内边距为20像素*/
26     padding-right:20px;
27     background:blue;
28 }
29 #pp2
30 {
31     background:yellow;
32     height:50px;
```

```
33 }
34 #p3
35 {
36     padding-bottom:20px;
37     background:blue;
38 }
39 #pp3
40 {
41     background:red;
42     height:50px;
43 }
44 #p4
45 {
46     padding-left:20px;
47     background:blue;
48 }
49 #pp4
50 {
51     /*设置背景颜色和高度*/
52     background:gray;
53     height:50px;
54 }
55 </style>
56 </head>
57 <body>
58 <div id="p1">
59     <div id="pp1">padding-top设置</div>
60 </div>
61 <br>
62 <div id="p2">
63     <div id="pp2">padding-right设置</div>
64 </div>
65 <br>
66 <div id="p3">
67     <div id="pp3">padding-bottom设置</div>
68 </div>
69 <br>
70 <div id="p4">
71     <div id="pp4">padding-left设置</div>
72 </div>
73 </body>
74 </html>
```

【代码解析】在代码 15-17 中，用 4 个不同的层表示 padding 的各分属性。在第 2 层中，用 padding-right 设置了向右内边距的值（第 26 行代码），同时第 3 层中，设置了 padding-bottom 的值（第 36 行代码），最后在第 4 层中将 padding-left 的值设置为与其他值相同的数值（第 46 行代码），效果如图 15-17 所示。

15.6.3 用复合属性 padding 设置内边距

在前两节中，分别用上、右、下、左的方位表示了对象的内边距用法，并展现其效果。实际上，这四个方位的单独属性组合成一个复合属性，两者表达出的效果一样，而且所设置的代码会更少。

用复合属性 padding 表示上面的四边。如果用 padding 表示的话，只要一行代码就可以把四个边的效果表达出来。代码 15-17 表示用 padding 表达四种效果，在样式代码中加入如下 CSS 代码。

```

01 #p5
02 {
03     padding: 20px;
04     background: blue;
05 }
06 #pp5
07 {
08     background: green;
09     height: 50px;
10 }

```

【代码解析】在代码第 3 行中，表达了上、右、下、左的内边距值为 20 像素。

同时，在 HTML 代码中，也要添加两个层：一个容器层和一个子层。通过上面的 ID 设置，达到 padding 的效果，HTML 代码中的 p5 层如下所示。

```

<div id="p5">
    <div id="pp5">padding设置,综合上面4个属性
</div>

```

上面代码放入样式代码中，表示有两个新的 ID，容器层用蓝色背景，子层用绿色背景，效果中出现的蓝色刚好是 padding 设置的内边距，效果如图 15-18 所示。



图 15-17 设置其他单属性

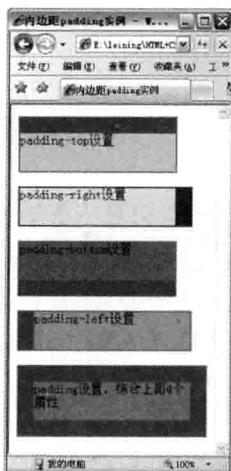


图 15-18 设置 padding

15.7 小结

本章首先介绍了外边距的基本语法，以及外边距的上、下、左、右边框的外延边距的设置。外边距用 margin 表示，同理，margin-top 表示顶部的外边距，即对象与上端位置的距离；margin-right

表示对象与右边的位置的距离；margin-bottom 表示对象与下端位置的距离；margin-left 表示对象与左端位置的距离。如果直接用 margin 表示，其后面接的四个数值分别代表上、右、下、左的外延距离。然后讲解了内边距的基本语法和四个方向的属性用法，内边距用 padding 表示，其后的值表示对象内部填充边距，复合属性接的四个数值分别表示 padding-top、padding-right、padding-bottom 和 padding-left 这四个方位，同时这四个单属性也可以单独表示其方位的内边距。

15.8 习题

一、填空题

1. 内边距用_____表示，是复合属性，该复合属性由顶端内边距_____、右端内边距_____、下端内边距_____和左端内边距_____组成。
2. 外边距用 margin 表示，后面接的参数可以是_____或_____，表示具体的精确宽度。

二、上机实践

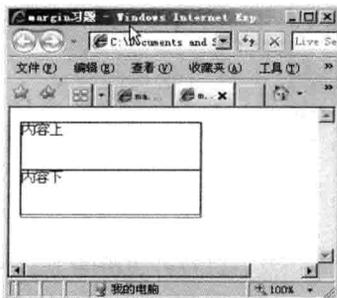
1. 按以下 DIV 布局制作网页。



【提示】

height-----DIV的高属性
width-----DIV的宽属性
float-----DIV的漂移属性
border-----DIV的边框属性

2. 按以下 DIV 布局制作网页。



【提示】

height-----DIV的高属性
width-----DIV的宽属性
float-----DIV的漂移属性
border-----DIV的边框属性

3. 按下图制作一个网页，要求只能用内边距 padding 属性。

**【提示】**

padding : length-----内边距语法

4. 按下图制作一个网页，要求不用内边距 padding 属性。

**【提示】**

padding-top:-----顶端内边距
padding-right-----右端内边距
padding-bottom-----底端内边距
padding-left-----左端内边距

第 16 章 设置对象边框

现在网页布局都喜欢用 DIV 和 CSS 结合，而通过 DIV 可以设置容器层的边框，使层对象的大小直观地表现出来。目前的网页都是用边框把内容包围起来，使页面更美观，更有模块感。

细心的读者会发现，边框在前面章节中的实例中已用到了，不过前面用到的是边框的复合属性 border，本章中除了介绍复合属性外，还会介绍边框的其他单属性。

学完本章，可以了解以下知识点：

- 边框的基本语法；
- 边框颜色表示；
- 边框样式表示；
- 边框宽度表示；
- 边框的四边设置；
- 边框的应用。

16.1 边框的基本语法

边框是对象的边界框图，用来设置对象的边框样式，语法用 border 表示，是复合属性，由颜色\样式和宽度组成，其语法完整表示如下：

```
border: border-width || border-style || border-color
```

从以往学习的复合属性可以知道，没有声明是在哪个方位的条件下，都代表着上、右、下、左的方位，边框属性也不例外，即 border 表示的是四边的边框，参数 border-width 表示边框的宽度，border-style 表示边框用什么样式表达（16.3 节介绍），border-color 表示边框的颜色。边框属性的参数默认值为 medium none，参数 medium 指 border-width 的值，参数 none 指 border-style 的值，参数 border-color 的默认值将采用文本颜色，所以在默认条件下，对象是没有边框的。代码 16-1 表示有 border 属性设置的层对象和没有 border 属性（即默认的）的层对象比较。

代码 16-1 border 基本用法

```
-----文件名: border基本用法.html-----
01 <html>
02 <head>
03 <title>border基本语法</title>
04 <style type="text/css">
05 #bb1
06 {
07     /*设置宽度*/
08     width:200px;
```

```

09      /*设置高度*/
10      height:80px;
11  }
12  #b1
13  {
14      /*设置边框5像素红色实线*/
15      border:5px solid red;
16  }
17  </style>
18  </head>
19  <body>
20  <div id="bb1">
21      这个层对象没有border,即为默认值
22      宽为200像素,高为80像素。
23  </div>
24  <div id="b1">
25      border是对象的边框<br>
26      宽度为5像素,实线红色
27  </div>
28  </body>
29  </html>

```

【代码解析】 在第 15 行代码设置了边框宽度为 5 像素,且为红色实线,而没有 border 属性的层只设置了宽和高,以确定层的大小,对比效果如图 16-1 所示。



图 16-1 border 基本用法

除了属性 border 有宽度、样式和颜色这三个参数外, border 四边的属性也具有这些参数,表示各个方位的边框样式,完整的语法形式如下。

```

/*CSS代码:边框的四个方位属性*/
border-top:border-width || border-style || border-color
border-right:border-width || border-style || border-color
border-bottom:border-width || border-style || border-color
border-left:border-width || border-style || border-color

```

上面的属性参数与 border 复合属性参数一样,都是由宽度、样式和颜色组成的,其默认值也是如此,它们的用法稍后一一介绍。

技巧

如果存在 border 属性和 border-top 属性对同一对象设置,其效果是最后面设置的值。即前面的设置被后面的设置覆盖, border-left 等也如此。

16.2 边框颜色

通过上一节对 border 属性的介绍，对其参数也有所了解，本节就对参数之一的边框颜色做讲解。从前面介绍可以知道，边框颜色由 border-color 表示，后面接颜色值，表示设置边框的颜色，完整代码如下。

```
border-color : color
```

从前面表示颜色的属性可以知道，color 的表示是不同的方式，在此不打算再做介绍。但要注意的是，颜色值可以接多个，在前面实例中都只有一种颜色，表示四边框的颜色，如果提供全部四个参数值，将按上、右、下、左的顺序作用于四个边框；如果只提供一个，将用于全部的四条边；如果提供两个，第一个用于上、下，第二个用于左、右；如果提供三个，第 1 个用于上，第 2 个用于左、右，第 3 个用于下。

当为对象设置边框颜色时，代码 16-2 在不同的层中表现不一样的颜色，下面看看如何设置边框颜色。

代码 16-2 设置边框颜色

-----文件名：设置边框颜色.html-----

```
01 <html>
02 <head>
03 <title>设置边框颜色</title>
04 <style type="text/css">
05 div
06 {
07     /*外边距顶端5像素*/
08     margin-top:5px;
09     /*边框样式为实型*/
10     border-style:solid;
11 }
12 #b1
13 {
14     /*设置边框颜色*/
15     border-color:#bbbbbb;
16 }
17 #b2
18 {
19     /*设置边框颜色*/
20     border-color:blue yellow;
21 }
22 #b3
23 {
24     设置边框综合属性
25     border-color:#bbbbbb red orange green;
26 }
27 #b4
28 {
29     /*设置为无边框*/
30     border:0;
31     /*设置边框底端为5像素红色实线*/
32     border-bottom:5px solid red;
```

```

33  }
34  </style>
35  </head>
36  <body>
37  <div id="b1">
38  HTML+CSS完全自学手册
39  <br>
40  这里4个边颜色一样
41  </div>
42  <div id="b2">
43  HTML+CSS完全自学手册
44  <br>
45  这里上下2个边颜色一样
46  </div>
47  <div id="b3">
48  HTML+CSS完全自学手册
49  <br>
50  这里4个边颜色都不一样
51  </div>
52  <div id="b4">
53  HTML+CSS完全自学手册
54  <br>
55  这里只有一个边框颜色
56  </div>
57  </body>
58  </html>

```

【代码解析】在代码 16-2 中，用不同的层对象表示不同的颜色，不过条件是当边框样式存在时（不为 none 时，本实例设置样式为 solid），最后一个层只有一个边框且颜色为红色（第 32 行代码），在样式中设置复合属性为 0，即设置整个边框为 none，然后对底端边框进行设置，注意，第 32 行代码通过 border-bottom 的设置，改变了 border 对底端边框的设置，效果如图 16-2 所示。

16.3 边框样式

边框样式用来表示对象边框不同的效果，不同的对象包括层、图像等，可以为其设置各式各样的效果，如常用的实线，还可以设置为虚线、点线等。设置边框的样式用 border-style 表示，其后所接的值为边框的不同样式，即多个参数中可选择一个来表示对象的边框样式，这些内容稍后讲解。

与边框颜色一样，边框样式也是边框属性的参数之一，由 16.1 节可知，边框复合属性由边框颜色、边框样式和边框宽度组成，而本节讲解的边框样式是组成边框复合属性的重要属性，用来表示对象的边框样式，参数只是其中的一个值，下面代码表示边框样式属性的完整语法。

```
border-style : none | dotted | dashed | solid | double | groove | ridge | inset | outset
```

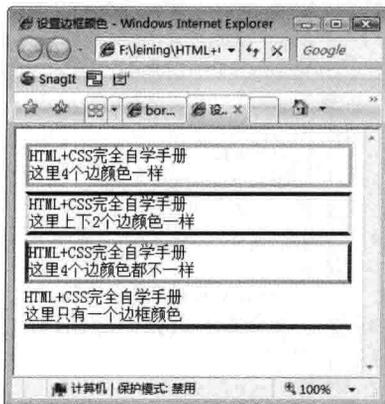


图 16-2 设置边框颜色

边框样式参数默认的是 none，表示默认条件下，边框样式是没有值的。在设置边框样式时，可在上面列出的属性值中能选择一个参数，不同的参数值表示不同的样式。以下是各属性值的含义。

- none：表示无边框，默认值。
- dotted：表示点线，由一各个点组成的边框。
- dashed：表示虚线，边框用虚线表示。
- solid：表示实线，边框用实线表示，常用的参数。
- double：双线边框，两条单线加个它们的间隔即是边框宽度，宽度越宽，间隔距离越大。
- groove：立体形式的凹槽。
- ridge：立体形式的凸槽。
- inset：立体形式的凹边。
- outset：立体形式的凸边。

由这些样式表示的边框样式是不同的。常用的边框样式为 solid，表示边框样式用实线表示，与之相反的参数值是 dashed，表示虚线样式，即设置对象的边框为虚线，同时还有 dotted 等。

边框样式在设置边框时用到，如设置了边框的宽度、颜色时，再通过设置边框样式来体现对象的边框效果，如设置边框样式为 none 时，不管边框颜色为何值，都不会出现任何边框，因为没有边框样式。代码 16-3 表示边框样式的不同参数值用法，通过设置不同的参数来表示不同的效果。

代码 16-3 边框样式用法

-----文件名：边框样式用法.html-----

```
01 <html>
02 <head>
03 <title>边框样式用法</title>
04 <style type="text/css">
05 div
06 {
07     /*设置层的顶部外边距*/
08     margin-top:15px;
09 }
10 #b1
11 {
12     /*设置边框样式*/
13     border-style:dashed;
14 }
15 #b2
16 {
17     border-style:dotted;
18 }
19 #b3
20 {
21     border-style:double;
22 }
23 #b4
24 {
25     border-style:groove;
26 }
27 #b5
```

```

28 {
29     border-style:inset;
30 }
31 #b6
32 {
33     border-style:outset;
34 }
35 #b7
36 {
37     border-style:ridge;
38 }
39 #b8
40 {
41     border-style:solid;
42 }
43 </style>
44 </head>
45 <body>
46 <div>none是默认的,表示无边框</div>
47 <div id="b1">dashed的边框样式</div>
48 <div id="b2">dotted的边框样式</div>
49 <div id="b3">double的边框样式</div>
50 <div id="b4">groove的边框样式</div>
51 <div id="b5">inset的边框样式</div>
52 <div id="b6">outset的边框样式</div>
53 <div id="b7">ridge的边框样式</div>
54 <div id="b8">solid的边框样式</div>
55 </body>
56 </html>

```

【代码解析】第 46 行代码对第一个层表示边框的样式设置为 none，也是边框样式的默认值；在第二层中，设置边框样式为 dashed（第 13 行代码），表示层对象边框为虚线的样式，与 dashed 效果类似的是参数值 dotted，表示点线的边框，即层对象的边框由点组成；在以后的层中设置了 double、groove、inset、outset 和 ridge 的参数值；而在最后一层，将边框设置为最常用的实线边框样式，参数值为 solid。整个效果如图 16-3 所示。

注意

如果设置 border-style 为 none，其他两个属性设置都无效。因为 none 表示无边框的样式。

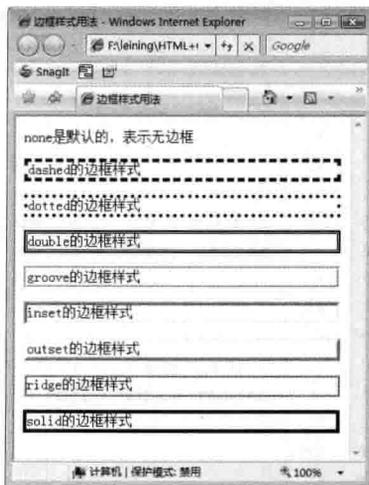


图 16-3 边框样式参数的用法

16.4 边框宽度

前面几节介绍了边框复合属性的语法以及其组成的属性，上一节介绍了边框复合属性中的两个属性：边框颜色和边框样式，其中边框颜色用 border-color 表示，而边框样式用 border-style 表示。本节来介绍 border 复合属性的最后一个属性用法，即边框宽度。

边框宽度语法用 `border-width` 表示，用来设置对象边框中的宽度值，边框宽度的属性值有两种表示方法：最常用的为直接接数值，然后接上单位；第二种方法为接表示宽度的词，通过三个单词来表示边框为宽、中等宽和窄。与其他属性设置属性一样，只能在两种表示方法中选择一个值，下面是其完整的语法形式。

```
border-width : medium | thin | thick | length
```

- `medium`：默认值，表示默认宽度。
- `thin`：表示小于默认宽度。
- `thick`：表示大于默认宽度。
- `length`：直接表示边框宽度的值。

由以上可知，属性值 `medium` 表示宽度为中等的边框宽度，也是默认的宽度，而当设置 `border-width` 为 `thin` 时，表示边框的宽度小于默认的宽度，与之相反的是，属性值 `thick` 表示大于默认宽度值的宽度，上面三种参数值并不是非选不可的，设置 `border-width` 的值为具体的数值时，表示边框的真实宽度。需要注意的是，设置参数 `length` 要注意单位。代码 16-4 演示了边框宽度的用法。

代码 16-4 边框宽度设置

```
-----文件名：边框宽度设置.html-----
01 <html>
02 <head>
03 <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
04 <title>边框宽度设置</title>
05 <style type="text/css">
06 div
07 {
08     /*设置顶部外边距为15像素*/
09     margin-top:15px;
10     /*设置边框样式为实线*/
11     border-style:solid;
12 }
13 #b1
14 {
15     /*设置边框宽度为medium*/
16     border-width:medium;
17 }
18 #b2
19 {
20     /*设置边框宽度为thin*/
21     border-width:thin;
22 }
23 #b3
24 {
25     border-width:thick;
26 }
27 #b4
28 {
29     border-width:1px;
30 }
```

```

31 #b5
32 {
33     border-width:10px;
34 }
35 </style>
36 </head>
37
38 <body>
39 <div id="b1">
40     宽度为medium时，默认宽度。
41 </div>
42 <div id="b2">
43     宽度为thin时，大于默认宽。
44 </div>
45 <div id="b3">
46     宽度为thick时，大于默认宽度。
47 </div>
48 <div id="b4">
49     宽度用数值1px时。
50 </div>
51 <div id="b5">
52     宽度用数值10px时。
53 </div>
54 </body>
55 </html>

```

【代码解析】在代码 16-4 中，用实例表示了边框宽度 border-width 的用法。第 16、21、25 行代码分别设置了边框宽度为 medium、thin、thick，表示边框的默认值、小于默认值、大于默认值，而在第 4 层中，也就是第 29 行代码设置了边框宽度为 1 像素，在最后一个边框中，即第 33 行代码设置了层对象的边框为 10 像素。效果如图 16-4 所示。

学完本节，知道了边框复合属性 (border) 是由边框颜色 (border-color)、边框样式 (border-style) 和边框宽度 (border-width) 组成。要设置一个边框的属性，除了这三个属性设置外，还可以单独用复合属性设置，如本书常设置边框值如下。

```
border:1px solid red;
```

以上代码表示设置对象的边框值为 1 像素的实线，且为红色的边框，是通过复合属性表示的。如用三个属性表示复合属性的效果如下。

```
border-width:1px;
border-style:solid;
border-color:red;
```

以上三行代码与复合属性的效果一样。可见，复合属性所表达的效果比用其属性表示的效果效率方面更高、更方便，这也正是在 CSS 中常用复合属性的原因。



图 16-4 边框宽度设置

除了复合属性用三个属性组成外，对边框的四个角的边框也可以进行设置，如单独设置边框中的某一方向的边框值，边框的方向只有顶端、右端、底端和左端。关于这方面的知识，稍后再讲解。

说明 边框宽度最常用的是直接用数值表示其宽度。

16.5 对象的四个边框

从前面的章节中可以知道边框（border）是复合属性，这个复合属性由边框颜色、边框样式和边框宽度组成，所表示的边框为顶端、右端、底端和左端的边框值，而且是同一个效果的值。

边框属性是复合属性，是由上端、右端、下端和左端边框属性组成的复合属性，也可以单独表示的某一方向的边框，如只设置对象顶端的边框，以区分其他的边框。为了实现边框在不同方向设置边框属性，需要对不同边框方面的属性进行设置。

在网页中表示的边框都是矩形的，即由上、下、左、右的边框围成对象，在本书中也称为顶端、右端、底端和左端的边框。通过边框方向属性可以设置边框在不同位置的边框值，各属性如下所示。

- border-top: 表示顶端的边框。
- border-right: 表示右端的边框。
- border-bottom: 表示底端的边框。
- border-left: 表示左端的边框。

通过上面的属性可以设置不同位置的边框值。本节分别介绍顶端、右端、底端和左端的属性值以及用法。

16.5.1 设置顶端边框的属性

顶端边框的属性专门用来设置对象的顶部边框。设置本属性，对其他方向的边框值没有影响，同时，属性值所接的值与复合属性一样，由边框宽度、边框颜色和边框样式组成。代码如下。

```
border-top : border-width || border-style || border-color
```

顶端的边框属性用 border-top 表示，其属性值有三个参数：第一个 border-width 表示顶部边框的宽度，第二个 border-style 表示顶部边框的样式，第三个 border-color 表示顶部边框的颜色。需要注意的是，对这三个参数值的顺序没有要求，即可以以任意顺序出现。

设置了顶端边框后，就确定了顶部的边框效果。代码 16-5 演示了 border-top 在对象层中的用法。

代码 16-5 border-top 用法

```
-----文件名: border-top用法.html-----
01 <html>
02 <head>
03 <title>border-top用法</title>
04 <style type="text/css">
05 #btop
06 {
07     width:200px;
```

```

08     height:150px;
09     /*设置顶部边框为5像素红色实线*/
10     border-top:5px solid red;
11 }
12 </style>
13 </head>
14 <body>
15 <div id="btop">
16 在这个层里面，只有顶端存在边框<br>
17 宽为200px，高为150。
18 </div>
19 上面的层高为150px。
20 </body>
21 </html>

```

【代码解析】第 15 行代码设置 ID 为 btop 的层对象的大小为宽 200 像素，高 150 像素。第 5~11 行代码设置顶部边框为 5 像素，且为红色实线，对其他方向和边框没有设置，即设置 border-top 属性，不会影响其他方面和边框，效果如图 16-5 所示。

16.5.2 设置右端边框的属性

与顶端的边框一样，右端的边框属性也是边框四个方向属性成员之一，右端的边框属性用 border-right 表示，其后所接的值与 border 复合属性一样，由边框宽度、边框颜色和边框样式组成的。完整代码如下。

```
border-right : border-width || border-style || border-color
```

本属性默认值参数为 medium none，而 border-color 的默认值将采用文本颜色。当然，这是表示右端的边框属性。与顶端的边框一样，设置本属性对其他方向的属性没有影响，只对该设置的方面产生影响。

代码 16-6 演示了 border-right 的用法。

代码 16-6 border-right 用法

```

-----文件名: border-right用法.html-----
01 <html>
02 <head>
03 <title>border-right用法</title>
04 <style type="text/css">
05 #btop
06 {
07     /*设置尺寸*/
08     width:200px;
09     height:150px;
10     border-right:5px solid red;
11 }
12 </style>
13 </head>

```

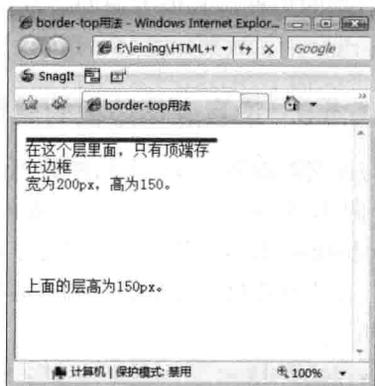


图 16-5 border-top 用法

```

14 <body>
15 <div id="btop">
16 在这个层里面，只有右端存在边框<br>
17 宽为200px，高为150。
18 </div>
19 上面的层高为150px。
20 </body>
21 </html>

```

【代码解析】在第 5~11 行代码的层对象中，设置宽为 200 像素，高为 150 像素，而设置右端边框 `border-right` 为 5 像素红色实线，即边框表示的只在右端，其他方向的边框没有设置，即没有边框值，同时设置右端的边框对其他方向的边框没有影响。设置 `border-right` 后，效果如图 16-6 所示。

16.5.3 设置底端边框的属性

与顶端的边框一样，底端的边框属性也是边框四个方向属性成员之一，底端的边框属性用 `border-bottom` 表示，其后所接的值与 `border` 复合属性一样，由边框宽度、边框颜色和边框样式组成。完整代码如下。

```
border-bottom : border-width || border-style || border-color
```

本属性默认值参数为 `medium none`，而 `border-color` 的默认值将采用文本颜色。当然，这是表示底端的边框属性。与顶端的边框一样，设置了本属性对其他方向的属性没有影响，只对该设置的方面影响。

代码 16-7 演示了 `border-bottom` 的用法。

代码 16-7 border-bottom 用法

```

-----文件名: border-bottom用法.html-----
01 <html>
02 <head>
03 <title>border-bottom用法</title>
04 <style type="text/css">
05 #btop
06 {
07     /*设置宽度和高度*/
08     width:200px;
09     height:150px;
10     /*设置底端边框为5像素红色实线*/
11     border-bottom:5px solid red;
12 }
13 </style>
14 </head>
15 <body>
16 <div id="btop">

```

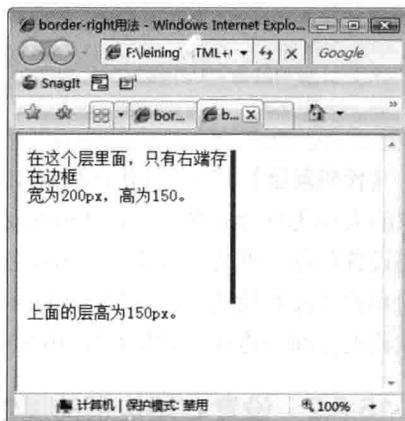


图 16-6 border-right 用法

```

17     在这个层里面，只有底端存大边框<br>
18     宽为200px，高为150。
19 </div>
20 上面的层高为150px。
21 </body>
22 </html>

```

【代码解析】第 16 行代码将 btop 层的宽设置为 200 像素，高设置为 150 像素，只对该层的底端设置了边框，且设置的边框宽度为 5 像素，用红色实线表示。btop 层外还有一行文本，通过边框把层与下面的文本分开，效果如图 16-7 所示。

16.5.4 设置左端边框的属性

与顶端的边框一样，左端的边框属性也是边框四个方向属性成员之一，左端的边框属性用 border-left 表示，其后所接的值与 border 复合属性一样，由边框宽度、边框颜色和边框样式组成。完整代码如下。

```
border-left : border-width || border-style || border-color
```

本属性默认值参数为：medium none，而 border-color 的默认值将采用文本颜色。当然，这是表示左端的边框属性，与顶端的边框一样，设置了本属性对其他方向的属性没有影响，只对该设置的方面影响。

代码 16-8 演示了 border-left 的用法。

代码 16-8 border-left 用法

```

-----文件名：border-left用法.html-----
01 <html>
02 <head>
03 <title>border-left用法</title>
04 <style type="text/css">
05 #btop
06 {
07     width:200px;
08     height:150px;
09     /*设置左问边框为5像素红色实线*/
10     border-left:5px solid red;
11 }
12 </style>
13 </head>
14 <body>
15 <div id="btop">
16     在这个层里面，只有左端存大边框<br>
17     宽为200px，高为150。
18 </div>
19 上面的层高为150px。
20 </body>
21 </html>

```



图 16-7 border-bottom 用法

【代码解析】与设置其他方向的属性一样，第 10 行代码在一个宽为 200 像素，高为 150 像素的层中，设置 border-left 的值为 5px solid red，表示 5 像素的红色实线边框，且在对象的左端显示，效果如图 16-8 所示。

说明

不管在 border 中还是在 border-right 等四个方向的边框属性中，对 border-width、border-style 和 border-color 的顺序都可以用任意顺序，按自己的习惯书写即可，注意中间用空格分开。



图 16-8 border-left 用法

16.6 边框实例——多层边框

本实例手把手来完成两件事。第一，用完整的实例来演示边框的所有属性以及用法，总结本章学习的边框知识。第二，用实例来设置多层边框，通过 CSS 属性来设置双边框的效果。

16.6.1 设置单层边框

本实例对不同的层对象设置边框。其属性用本章所学的边框知识作为演示的对象，在本小节的实例中，将要完成的是边框的四个方位属性在网页中的用法，通过以下步骤来实现这些用法。

(1) 设置多个层不同的样式。在本步骤中，先设置好 HTML 代码中的 DIV，然后再关联对应的 CSS 样式，如代码 16-9 所示。

代码 16-9 边框实例

```
-----文件名：边框实例手把手.html-----
01 <html>
02 <head>
03 <title>边框实例手把手</title>
04 <style type="text/css">
05 div
06 {
07     width:200px;
08     height:50px;
09     margin-top:5px;
10 }
11 #b1
12 {
13     /*设置边框为3像素红色点线*/
14     border:3px dotted red;
15 }
16 #b2
17 {
18     /*设置边框为5像素绿色虚线*/
19     border:5px dashed green;
20 }
21 #b3
22 {
```

```

23      /*设置边框为8像素蓝色立体凸槽*/
24      border:8px ridge blue;
25  }
26  </style>
27  </head>
28  <body>
29  <div id="b1">
30      边框的样式为点线, 颜色为红色, 宽度为3px。
31  </div>
32  <div id="b2">
33      边框的样式为虚线, 颜色为绿色, 宽度为5px。
34  </div>
35  <div id="b3">
36      边框样式为立体凸槽, 颜色为蓝色, 宽度为8px。
37  </div>
38  </body>
39  </html>

```

【代码解析】在 16-9 代码中, 通过实例演示了边框的用法。第 29 行代码在 ID 选择符为 b1 的样式中, 设置了边框为 3 像素红色点线; 第 32 行代码在 ID 选择符为 b2 的样式中, 设置层边框为 5 像素绿色虚线; 同理, 第 35 行代码在 ID 选择符为 b3 的样式中, 设置该层边框为 8 像素蓝色立体凸槽。各层边框表示的效果如图 16-9 所示。

(2) 给不同的层设置不同的边框颜色。在上面的代码中, 设置了三个层对象的边框, 使每个边框的宽度、样式和颜色都不一样, 在本步骤中, 再添加两个层以及与层关联的 CSS 样式, 在样式中设置边框的属性。

通过添加这些样式, 在原来的基础上增添层标记和 CSS 代码, 在 CSS 样式表中加入 ID 选择符 b4 和 b5, 分别设置不同颜色的边框样式, 添加后的 CSS 代码如下。

```

01  #b4
02  {
03      border:1px solid #666666;
04  }
05  #b5
06  {
07      border:1px solid orange;
08  }

```

【代码解析】以上代码新增两个层样式, 第 1~4 行代码的 b4 层边框的宽度为 1px, 实线, 颜色为 #666666; 第 5~8 行代码的 b5 层边框的宽度为 1px, 实线, 颜色为橙色。

通过上面的 CSS 代码设置, 在 <body></body> 中加入的层对象需要关联上面的 CSS 选择符, 即设置添加的第一个层 ID 为 b4, 同理, 对第二个层设置 ID 为 b5, 添加后, 对文件进行保存。

```

01  <div id="b4">
02      边框样式为实线, 颜色为#666666, 宽度为1px。
03  </div>
04  <div id="b5">
05      边框样式为实线, 颜色为橙色, 宽度为1px。
06  </div>

```

【代码解析】第 1 行代码设置了边框样式为实线, 颜色为 #666666, 宽度为 1px, 而第 4 行代码设置了边框样式为实线, 颜色为橙色, 宽度为 1px, 把上述代码加入原有代码的结尾, 效果如图 16-10 所示。



图 16-9 设置不同层的边框样式

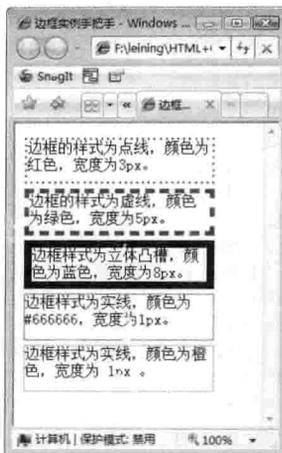


图 16-10 加入两个不同颜色的层

(3) 设置对象四边边框为不同颜色和宽度。在本步骤中，设置不同的边框样式和不同的边框宽度，用 border 属性来设置复合值和单独的边框样式，如代码 16-10 所示。

代码 16-10 边框实例

```

-----文件名：边框实例手把手.html-----
01 <html>
02 <head>
03 <title>边框实例手把手</title>
04 <style type="text/css">
05 div
06 {
07     /*设置宽度和高度*/
08     width:200px;
09     height:50px;
10     /*设置层的顶部外边距*/
11     margin-top:5px;
12 }
13 #b1
14 {
15     /*点线，颜色为红色，宽度为3px*/
16     border:3px dotted red;
17 }
18 #b2
19 {
20     border:5px dashed green;
21 }
22 #b3
23 {
24     border:8px ridge blue;
25 }
26 #b4
27 {
28     border:1px solid #666666;
29 }
    
```

```

30 #b5
31 {
32     border:1px solid orange;
33 }
34 #b6
35 {
36     border-top:4px double red;
37     border-right:6px groove green;
38     border-bottom:8px outset yellow;
39     border-left:2px dashed orange;
40 }
41 </style>
42 </head>
43 <body>
44 <div id="b1">
45     边框的样式为点线，颜色为红色，宽度为3px。
46 </div>
47 <div id="b2">
48     边框的样式为虚线，颜色为绿色，宽度为5px。
49 </div>
50 <div id="b3">
51     边框样式为立体凸槽，颜色为蓝色，宽度为8px。
52 </div>
53 <div id="b4">
54     边框样式为实线，颜色为#666666，宽度为1px。
55 </div>
56 <div id="b5">
57     边框样式为实线，颜色为橙色，宽度为 1px 。
58 </div>
59 <div id="b6">
60     在这个层里面，四个边的颜色，边框和样式都不同。
61 </div>
62 </body>
63 </html>

```

【代码解析】代码 16-10 中，对不同的层对象设置了不同的边框属性，第 44 行代码为第 1 层，设置的边框为点线，颜色为红色，宽度为 3px；第 47 行代码为第 2 层，设置的是虚线，颜色为绿色，宽度为 5px；第 50 行代码为第 3 层，设置的是立体凸槽，颜色为蓝色，宽度为 8px；第 53 行代码为第 4 层，设置的是实线，颜色为 #666666，宽度为 1px；第 56 行代码为第 5 层，设置的是实线，颜色为橙色，宽度为 1px；第 59 行代码为第 6 层，设置的是四个不同的边框样式和不同的颜色。效果如图 16-11 所示。

16.6.2 设置多层边框

有时单条边框不能表达出边框的效果，换言之，

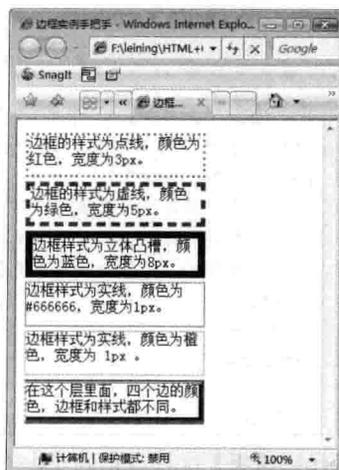


图 16-11 层中四个不同的边框

有更好的方法可以在某些应用比单边框效果会好些。有一种用法，两个层中，大层背景与小层的背景相区别，然后在大层中加上不同颜色的边框，这样就会出现三种颜色，看起来有立体效果，如百度知道的边框就是利用这种方式布局，只不过百度用到了边框圆角（圆角在以后的实例中也会讲解）。

本节介绍在布局中实用的多边框用法，在多个边框中，用到不同颜色的边框组成一个对象的边框效果，从外观给人的感觉有层次感，能够表现出有阴影的 3D 效果，如代码 16-11 所示。代码 16-11 是如何布局类似的边框，但不会按步骤讲解，请多看代码中的解析。

代码 16-11 设置多层边框

```
-----文件名：设置多层边框.html-----
01 <html>
02 <head>
03 <title>多层边框</title>
04 <style type="text/css">
05 #myborder
06 {
07     /*设置宽度*/
08     width:300px;
09     /*设置背景颜色*/
10     background:#bbb;
11     /*设置边框，但颜色与背景不同*/
12     border:1px solid #aaa;
13 }
14 #mybg
15 {
16     /*上、右、下、左的外边距为3像素*/
17     margin:3px;
18     /*上、右、下、左的内边距为2像素*/
19     padding:2px;
20     /*设置与上面两种都不同的颜色*/
21     background:#ccc;
22 }
23 </style>
24 </head>
25 <div id="myborder">
26 <div id="mybg">
27     这里面是用背景填充，类似于边框<br />
28     效果类似于边框，但这种可以定颜色值<br>
29     能体现立体性，加上圆角会更好。
30 </div>
31 </div>
32 <body>
33 </body>
34 </html>
```

【代码解析】这样就创建了一个对象层。通过三种不同的颜色来设置边框增强了立体感，不过要注意的事，颜色搭配也是一种艺术。本实例中的多层边框是用两个深点儿的颜色表示边框，然后

用另一种颜色表示背景，效果如图 16-12 所示。

说明 颜色搭配在网页中很重要，要想好好地学习，并不是看看书就能够提高的，要多看、多练习和多交流。能用好网页中的颜色，会起事半功倍的效果，这需要多参阅这方面的书籍。

16.7 小结

本章主要介绍了边框 border 的属性。border 是复合属性，参数包括 border-color、border-width 和 border-style，分别表示边框颜色、边框宽度和边框样式。边框的四个边都可以用颜色、宽度和样式这三个属性表示，同时也是复合属性，如 border-top:1px solid red 表示 border-top-width 为 1 像素、border-top-style 为 solid，即实线、border-top-color 为红色。同理，其他方向边框也可这样表示。最后用实例再次演示了边框必须要理解的知识，同时用实例对多层边框进行了演示。

16.8 习题

一、填空题

1. 边框是对象的边界框图，用来设置对象的边框样式，其语法用_____表示，是_____属性。
2. 边框复合属性由_____、_____和_____组成。
3. 顶端的边框属性用_____表示，其属性值有三个参数：第一个_____表示顶部边框的宽度，第二个_____表示顶部边框的样式，第三个_____表示顶部边框的颜色。

二、上机实践

1. 按下面的样式编辑一张网页。

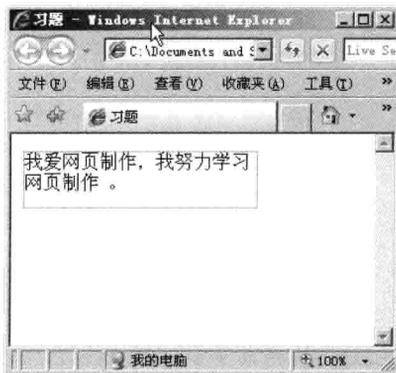


图 16-12 设置多层边框

【提示】

`border : border-width || border-style || border-color-----边框基本语法`

2. 按下面的样式编辑一张网页。



【提示】

`border-top-----顶端边框标识`
`border-left-----左端边框标识`
`border-bottom-----底端边框标识`
`border-right-----右端边框标识`

第 17 章 定 位

定位经常在网页中用到,如设置对象的绝对定位或设置图像特殊位置的背景,都需要用到定位,以达到特殊的效果。

本章讲解对象的定位方式,即定位属性,还会介绍一个对象的层叠顺序 `z-index` 及对象与父对象的相关边框位置设置。通过实例对相对定位和绝对定位进行解释,相信读者对定位的理解会达到一个新的层次。

学完本章,应理解以下内容:

- 定位的基本语法;
- 对象的层叠顺序;
- 对象的边框位置;
- 相对定位用法;
- 绝对定位用法。

17.1 定位的基本语法

定位属性由 `position` 表示,用来设置对象的定位方式,有绝对定位和相对定位之分,完整语法形式如下所述。

```
position:static | absolute | relative
```

属性 `position` 后面接的值分别表示如下。

- `static`: 默认值。无特殊定位。
- `absolute`: 将对象从文档流中流出,使用 `left`、`right`、`top`、`bottom` 等属性表示最接近的一个父对象的绝对定位的方位。
- `relative`: 对象不可层叠,但将依据 `left`、`right`、`top` 和 `bottom` 等属性在正常文档流中偏移位置。

设置定位属性值为 `absolute` 时,会将对象拖离出正常的文档流的绝对定位,而不考虑它周围内容的布局。假如其他具有不同 `z-index` 属性(下一节讲解)的对象已经占据了给定的位置,它们之间不会相互影响,而会在同一位置层叠。此时对象不具有外边距(`margin`),但仍有内边距(`padding`)和边框(`border`)。

要激活对象的绝对定位(`absolute`),必须指定 `left`、`right`、`top` 和 `bottom` 属性中的至少一个,并且设置此属性值为 `absolute`。

设置此属性值为 `relative` 会保持对象在正常的 HTML 流中,但是它的位置可以根据它的前一个对象进行偏移。在相对(`relative`)定位对象之后的文本或对象占有它们自己的空间而不会覆盖被

定位对象的自然空间。与此不同的是，在绝对（absolute）定位对象之后的文本或对象在被定位对象拖离正常文档流之前会占有它的自然空间。放置绝对（absolute）定位对象在可视区域之外会导致滚动条出现。而放置相对（relative）定位对象在可视区域之外，滚动条不会出现。代码 17-1 表示 position 的用法。

代码 17-1 position 的用法

```
-----文件名: position的用法.html-----
01 <html>
02 <head>
03 <title>position的用法</title>
04 <style type="text/css">
05   div
06   {
07     border:1px solid red;
08     margin-top:5px;
09   }
10   #p1
11   {
12     /*设置为绝对定位*/
13     position:absolute;
14     /*距左端20像素*/
15     left:20px;
16     /*距顶端100像素*/
17     top:100px;
18   }
19   #p2
20   {
21     /*设置为相对定位*/
22     position:relative;
23     /*距左端100像素*/
24     left:100px;
25   }
26 </style>
27 </head>
28 <body>
29 <div id="p3">
30   这是默认值。static
31 </div>
32 <div id="p1">
33   这是absolute的值--HTML+CSS完全自学手册<br>
34   left设置为20px;top设置为100px.
35 </div>
36 <div id="p2">
37   这是relative的值--HTML+CSS完全自学手册<br>
38   left设置为100px.
39 </div>
40 </body>
41 </html>
```

【代码解析】在代码 17-1 中，有 3 个层：第 29 行代码将第 1 层的 position 属性，设置为默认值 static；第 32 行代码将第 2 层的 position 属性，设置为 absolute，即绝对定位，同时设置了 left

的值为 20 像素，top 的值为 100 像素；第 36 行代码将第 3 层的 position 属性设置为 relative，即相对定位，同时还设置了 left 的值为 100 像素。3 个层在一张网页中，方便比较 position 属性的各参数区别，效果如图 17-1 所示。

设置了相对定位 (relative) 或绝对定位 (absolute) 后，至少要设置 left、top、right 和 bottom 中的一种。

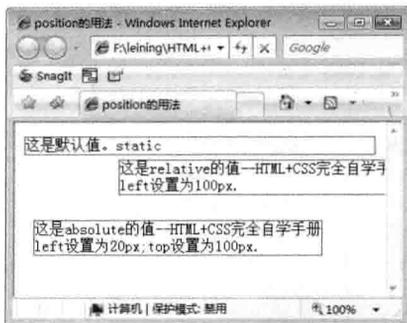


图 17-1 position 的用法

17.2 设置对象的层叠顺序

学习了定位语法之后，对下一个属性学习是很有必要的，这就是对象的层叠顺序。层叠属性类似于三维的 Z 轴，以对象为中心，前后移动，语法用 z-index 表示，后面可取值，值较大的元素将叠加在 z-index 值较小的元素之上。以下是层叠对象的完整语法。

```
z-index : auto | number
```

层性 z-index 后面接的值分别表示如下。

- auto：默认值，由其父对象的定位决定。
- number：无单位的整数值，可为负数。

属性的默认值为 auto，如果设置是数字，可正可负。如果为正，最大数的对象将在前面，数值越大，越往前端显示；如果为负值，将会出现对象之后。在代码 17-2 中，用 z-index 设置了三个不同层的值，以演示 z-index 的用法。

代码 17-2 z-index 用法

```
-----文件名：z-index用法.html-----
01 <html>
02 <head>
03 <title>z-index用法</title>
04 <style type="text/css">
05 #p1
06 {
07     /*设置为100*/
08     z-index:100;
09     /*设置为相对定位*/
10     position:relative;
11     /*距左端10像素*/
12     left:10px;
13     /*用不同背景区别各层*/
14     background:#ccc;
15 }
16 #p2
17 {
18     /*设置为200，比100大，将会显示在设置z-index值对象的前面*/
19     z-index:200;
20     position:relative;
21     left:30px;
22     top:-5px;
```

```

23     background:#888;
24 }
25 #p3
26 {
27     /*设置为300比前面两个对象都大*/
28     z-index:300;
29     position:relative;
30     left:60px;
31     top:-10px;
32     background:#666;
33 }
34 </style>
35 </head>
36 <body>
37 <div id="p1">
38     这是第一个层, z-index设置为100。
39 </div>
40 <div id="p2">
41     这是第二个层, z-index设置为200。
42 </div>
43 <div id="p3">
44     这是第三个层, z-index设置为300。
45 </div>
46 </body>
47 </html>

```

【代码解析】在代码 17-2 中,存在 3 个不同的层:第 37 行代码为第 1 层,设置的 z-index 值为 100,背景颜色设为 #ccc;第 40 行代码为第 2 层,设置的 z-index 值为 200,且背景颜色值用 #888 表示;第 43 行代码为第 3 层,设置的 z-index 为 300,比前两个层的值都要高,同时将背景颜色设置为 #666。通过不同的 z-index 和不同的背景颜色来表现各个层,效果如图 17-2 所示。



图 17-2 属性 z-index 用法

17.3 设置对象的边框位置

设置对象的边框位置用方向属性,方向属性就是表达上、右、下、左方向的属性,即 top 表示顶端、right 表示右端、bottom 表示底端和 left 表示左端。在属性 position 中,不管是相对定位还是绝对定位,都可以设置对象与父对象边框的位置距离。如果没有 position 属性,系统会忽略方向属性值。

对方向属性这四个属性语法,其后可接数值或 auto,其中 auto 表示默认值,数值是要带距离单位的,完整语法形式如下。

```

top : auto | length
right : auto | length
bottom : auto | length
left: auto | length

```

方向属性与其他属性参数设置的一样，不同的是应用在什么样的属性上面，在这里是配合 position 属性时所用，正如前面所讲，没有定位属性，系统会忽略这些方向属性。代码 17-3 表示如何在定位属性中使用方向属性，来给对象设置边框位置。

代码 17-3 设置对象的边框位置

```
-----文件名：设置对象的边框位置.html-----
01 <html>
02 <head>
03 <title>设置对象的边框位置</title>
04 <style type="text/css">
05 div
06 {
07     width:100px;
08     height:50px;
09 }
10 #p1
11 {
12     /*设置为相对定位*/
13     position:relative;
14     left:20px;
15     top:5px;
16     background:#ccc;
17 }
18 #p2
19 {
20     /*设置为绝对定位*/
21     position:absolute;
22     right:20px;
23     top:45px;
24     background:#888;
25 }
26 #p3
27 {
28     /*设置为绝对定位*/
29     position:absolute;
30     bottom:20px;
31     background:#666;
32 }
33 </style>
34 </head>
35 <body>
36 <div id="p1">
37     这是第一个层, left:20px;top:5px;
38 </div>
39 <div id="p2">
40     这是第二个层
41 <br>right:20px;top:45px;。
42 </div>
43 <div id="p3">
44     这是第三个层, bottom:20px;。
45 </div>
46 </body>
47 </html>
```

【代码解析】在代码 17-3 中，设置了 3 个不同定位属性的层：第 36 行代码为第 1 层，设置为相对定位，left 为 20 像素，top 为 5 像素，即对象与父对象的左边距离为 20 像素，到顶端距离是 5 像素；第 39 行代码为第 2 层，设置为绝对定位，同时设置 right 的值为 20 像素，top 的值为 45 像素；第 43 行代码为第 3 层，设置的也是绝对定位，只不过方向属性只设置 bottom 值为 20 像素，即对象到底端的距离为 20 像素。效果如图 17-3 所示。



图 17-3 设置对象的边框位置

17.4 定位实例——圆角制作

定位 (position) 在网页布局中占很重要的比例，而且该属性也很关键。学会了定位，为以后用 HTML+CSS 模式布局网页会带来很大的方便，也会表现出其他属性无法表现的效果。本节实例中，把定位分成相对定位和绝对定位来演示。

17.4.1 相对定位实例

从前面学习中可知道相对定位的对象不可层叠，但会依据 left、right、top、bottom 等属性在正常文档流中偏移位置。代码 17-4 演示了相对定位的用法。

代码 17-4 相对定位实例

```
-----文件名：相对定位实例.html-----
01 <html>
02 <head>
03 <title>相对定位实例</title>
04 <style type="text/css">
05 #p1
06 {
07     width:250px;
08     border:1px solid red;
09 }
10 #pp1
11 {
12     position:relative;           /*相对定位*/
13     right:-10px;                 /*距右10像素*/
14     bottom:10px;                /*距底10像素*/
15     width:120px;                /*宽为120像素*/
16     height:80px;               /*高为80像素*/
17     border:1px solid blue;      /*用不同的背景颜色区别各层*/
18 }
19 #pp2
20 {
21     position:relative;
22     width:120px;
23     height:80px;
```

```

24     left:10px;
25     top:10px;
26     border:1px solid orange;
27 }
28 #pp3
29 {
30     position:relative;
31     width:120px;
32     height:80px;
33     border:1px solid green;
34 }
35 </style>
36 </head>
37 <body>
38 <div id="p1">
39     <div id="pp1">
40     </div>
41     <div id="pp2">
42     </div>
43     <div id="pp3">
44     </div>
45 </div>
46 </body>
47 </html>

```

【代码解析】在代码 17-4 中，一个大容器层包含 3 个小层，这 3 个小层都设置为相对定位，即这 3 个层为不可叠层，但依据 left、top 等属性会产生偏移。第 39 行代码为第 1 层，设置了 right 为 10 像素且 bottom 也是 10 像素，边框颜色值为 blue，这样层就会在原来的位置上产生偏移；第 41 行代码为第 2 层，设置的都是相对定位，所以第 2 层在第 1 层下方，因设置了 left 为 10 像素，top 为 10 像素，边框颜色值为 orange，所以也会产生偏移；第 43 行代码为第 3 层，没有设置方向属性，只是设置边框颜色值为 green，这些参数都是默认值 auto。偏移后的效果如图 17-4 所示。

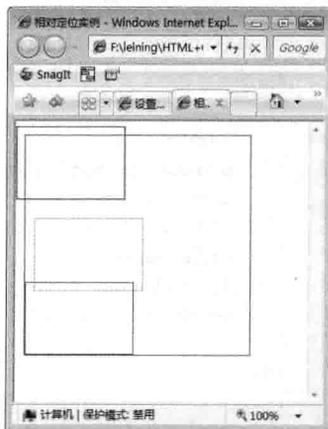


图 17-4 相对定位实例

17.4.2 绝对定位实例

从前面的讲解可知道，绝对定位将对象从文档流中拖出。使用 left、right、top 和 bottom 等属性，对相对于其最接近的一个最有定位设置的父对象进行绝对定位。与相对定位不同的是，绝对定位的层是可以重叠的，如果多个层都不设置方向属性（如 left、top 等）为默认值，所有的层都重叠。

本实例主要对绝对定位用法进行演示，代码 17-5 表示在一个父层中存在三个不同设置的绝对定位层，通过设置不同的方向属性，表现出不同的效果。

代码 17-5 绝对定位实例

```
-----文件名：绝对定位实例.html-----
01 <html>
02 <head>
03 <title>绝对定位实例</title>
04 <style type="text/css">
05 #p1
06 {
07     width:250px;
08     height:200px;
09     /*红色为容器层*/
10     border:1px solid red;
11     /*设置为相对定位*/
12     position:relative;
13 }
14 #pp1
15 {
16     /*绝对定位*/
17     position:absolute;
18     /*往右离开对象10像素*/
19     right:-10px;
20     /*距底10像素*/
21     bottom:10px;
22     width:120px;
23     height:80px;
24     border:1px solid blue;
25 }
26 #pp2
27 {
28     /*绝对定位*/
29     position:absolute;
30     width:120px;
31     height:80px;
32     left:-10px;
33     top:50px;
34     border:1px solid orange;
35 }
36 #pp3
37 {
38     /*绝对定位*/
39     position:absolute;
40     width:120px;
41     height:80px;
42     border:1px solid green;
43 }
44 </style>
45 </head>
46 <body>
47 <div id="p1">
48     <div id="pp1">
49         第一个层
50     </div>
```

```

51     <div id="pp2">
52         第二个层
53     </div>
54     <div id="pp3">
55         第三个层
56     </div>
57 </div>
58 </body>
59 </html>

```

【代码解析】在代码 17-5 中，p1 表示属性 position 为 relative 的父层（为什么是父层，请读者来思考），其子层中都设置 position 属性为 absolute，然后这 3 个子层对方向属性设置不同，来定位不同的位置。第 48 行代码为第 1 层，边框颜色值为 blue，right 为 10 像素且 bottom 也为 10 像素，这样就在层 p1 中产生了偏移；第 51 行代码为第 2 层，其边框颜色为 orange，且 left 和 top 值都是 50 像素；第 54 行代码为第 3 层，边框颜色为 green，方向属性值都是默认值。整体效果如图 17-5 所示。

技巧

如果在一个层中对作为子层的父层来绝对定位，那么一定要设置父层属性 position 为 relative，然后设置子层属性 position 为 absolute。否则，设置绝对定位的层以整个网页为父对象。



图 17-5 绝对定位实例

17.4.3 实例——圆角制作

在网页中，各个模块都用圆角作为背景，这样看上去给人的感觉更舒服。本节用实例来讲解如何用定位属性来做圆角效果，同时还会用到前面章节学习的背景，原理就是用背景图来覆盖角的位置。选择好四个角的图片，现在就开始利用定位来做圆角。

(1) 提供效果图，并准备好素材。下面列出圆角效果图和其完整代码，读者先看代码 17-6，然后分析下面步骤是如何实现效果的。

代码 17-6 实例——圆角制作

```

-----文件名：实例—圆角制作.html-----
01 <html>
02 <head>
03 <title>实例—圆角制作</title>
04 <style type="text/css">
05 body
06 {
07     /*整个背景为黄色*/
08     background:yellow;
09 }
10 #p
11 {
12     width:300px;

```

```
13     height:120px;
14     /*设置为相对定位*/
15     position:relative;
16     /*设置4边边框*/
17     border:1px solid #a7a5a5;
18     /*对象背景为黑色*/
19     background:#000000;
20     /*字体为白色*/
21     color:#ffffff;
22 }
23 #plefttop
24 {
25     /*确定该对象的大小*/
26     width:8px;
27     height:8px;
28     /*设置为绝对定位*/
29     position:absolute;
30     /*设置对象向左移1像素*/
31     left:-1px;
32     /*设置对象向上移1像素*/
33     top:-1px;
34     /*背景属性 取背景图片的位置并不重复*/
35     background:url(bg.gif) no-repeat -2px -3px;
36 }
37 /*下面的注解与#plefttop相似,故不做注解*/
38 #prighttop
39 {
40     width:8px;
41     height:8px;
42     position:absolute;
43     right:-1px;
44     top:-1px;
45     background:url(bg.gif) no-repeat -2px -32px;
46 }
47 #pleftbottom
48 {
49     width:8px;
50     height:8px;
51     position:absolute;
52     left:-1px;
53     bottom:-1px;
54     background:url(bg.gif) no-repeat -2px -18px;
55 }
56 #prightbottom
57 {
58     width:8px;
59     height:8px;
60     position:absolute;
61     right:-1px;
62     bottom:-1px;
63     background:url(bg.gif) no-repeat -3px -49px;
```

```

64  }
65  </style>
66  </head>
67  <body>
68  <div id="p">
69      <div id="plefttop"></div><div id="prighttop"></div>
70      <p>
71          这个层是要设置成圆角，文字为白色，边框为黑色。用到的是这个圆角素材
73      </p>
74      <div id="pleftbottom"></div><div id="prightbottom"></div>
75  </div>
76  </body>
77  </html>

```

【代码解析】在代码 17-6 中，用一个层包含 4 个带有背景图片的小层，代码第 19 行设置容器层的背景为黑色，字体为白色，层中显示的图片就是下面用 background 对其操作的素材，最终效果如图 17-6 所示。

(2) 在基本代码中插入层，并设置边框以区分层大小。本步骤操作的是在网页中建层对象，并设置边框和部分文字。

现在从新建网页开始。从前面学的知识可知，插入样式代码即可在<head></head>中放入<style></style>标记对，然后配合<body></body>中的标记进行操作。

由于步骤 (1) 提供了完整代码，因此以后步骤只对关键代码列出并讲解。下面是对样式的操作。

```

01  <style type="text/css">
02  body
03  {
04      /*设置黄色背景*/
05      background:yellow;
06  }
07  #p
08  {
09      /*设置对象宽和高，定位是相对定位，还有1像素的边框*/
10      width:300px;
11      height:100px;
12      position:relative;
13      border:1px solid #a7a5a5;
14  }
15  </style>

```

【代码解析】第 7~14 行代码为 p 的样式设置。

在标记中放入一层，设置 ID 为 p，输入文字，再插入一张要用到的图片素材，图片由圆角组成，共四个。按效果都要覆盖层 p 中的四个角，代码如下。

```

01  <body>
02  <div id="p">

```

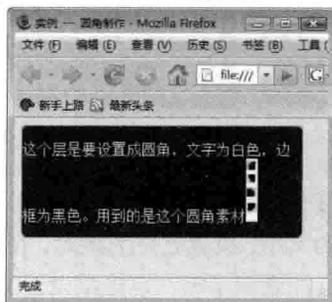


图 17-6 实例——圆角制作完效果

```

03     <p>
04     这个层是要设置成圆角，文字为白色，边框为黑色。用到的是这个圆角素材
06     </p>
07 </div>
08 </body>

```

本步骤的效果如图 17-7 所示。

(3) 用不同的层设置定位属性来覆盖容器层的四个角，使背景为各层的背景效果，这样就达到了圆角效果。具体实现要在层对象中加入四个层，然后在 CSS 中来设置各层的定位属性。下面先对左上角的圆角设置，下面的样式是新加入的，本代码效果只控制层中 ID 设置为 plefttop 的层。

```

01 #plefttop
02 {
03     width:8px;
04     height:8px;
05     position:absolute;
06     left:-1px;
07     top:-1px;
08     background:url(bg.gif) no-repeat -2px -3px;
09 }

```

【代码解析】分析这个样式，第 3~4 行代码设置宽和高都为 8 像素，表示该层要在容器层中覆盖的大小，设置定位为绝对，在代码第 5 行，要对本容器层起作用则需要在容器层对象设置相对定位（注意到了吗？容器层设置为是相对定位），然后在第 6、7 行代码，使该层往左和上端各移动 1 像素，刚好把边框的那部分遮住，用到的是背景图片 bg.gif，其中 -2 和 -3 表示对背景图片的取坐标位置。设置完 CSS 代码后，HTML 中的 ID 选择符就被 CSS 关联起来了，其代码如下。

```

<div id="p">
  <div id="plefttop"></div>
  <p>
    这个层是要设置成圆角，文字为白色，边框为黑色。用到的是这个圆角素材
  </p>
</div>

```

上面只对一个角设置，其他角的设置方法类似。设置效果如图 17-8 所示。

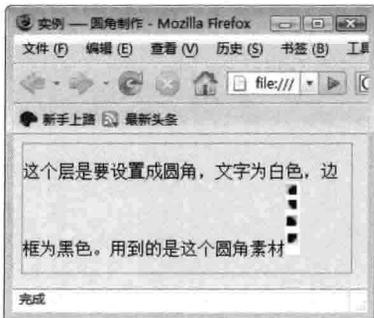


图 17-7 文字和边框

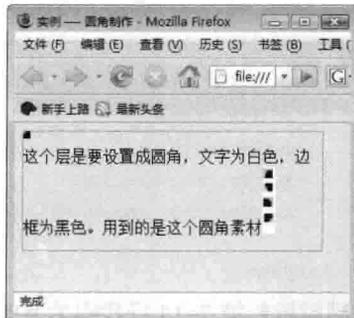


图 17-8 对左上端圆角设置

(4) 调节边框与圆角背景，使其看上去更加美观，并设置层内背景。由步骤 (3) 可知，其他角设置只对定位方向进行设置，如对象 prighttop 的是向上面右端位置，同理负值往不同方

向移动 1 像素遮住边框。同理，其他设置只对方向和背景取值不同。样式代码如下。

```

01 #prighttop
02 {
03     width:8px;
04     height:8px;
05     position:absolute;
06     right:-1px;
07     top:-1px;
08     background:url(bg.gif) no-repeat -2px -32px;
09 }
10 #pleftbottom
11 {
12     width:8px;
13     height:8px;
14     position:absolute;
15     left:-1px;
16     bottom:-1px;
17     background:url(bg.gif) no-repeat -2px -18px;
18 }
19 #prightbottom
20 {
21     width:8px;
22     height:8px;
23     position:absolute;
24     right:-1px;
25     bottom:-1px;
26     background:url(bg.gif) no-repeat -3px -49px;
27 }

```

代码 HTML 的<body></body>中，与步骤（3）的区别就是多了几个层，代码如下。

```

<div id="p">
  <div id="plefttop"></div><div id="prighttop"></div>
  <p>
    这个层是要设置成圆角，文字为白色，边框为黑色。用到的是这个圆角素材
  </p>
  <div id="pleftbottom"></div><div id="prightbottom"></div>
</div>

```

这样就完成了四个角的设置，有点圆角的感觉了，效果如图 17-9 所示。

（5）完成圆角制作。在容器层中设置背景为黑色，相反字体为白色，这样效果就表示出来了。

CSS 代码如下。

```

01 #p
02 {
03     width:300px;
04     height:120px;
05     position:relative;
06     /*设置了边框颜色*/
07     border:1px solid #a7a5a5;
08     /*背景颜色*/

```

```

09     background: #000000;
10     /*字体颜色*/
11     color: #ffffff;
12 }

```

这样就完成了整个效果。本实例主要讲解圆角的制作，除了要求对代码的掌握外还得要求有美工（专给网站设置效果图的工作人员）的图片，再用图片来实现网页的效果。本实例最终效果如图 17-10 所示。

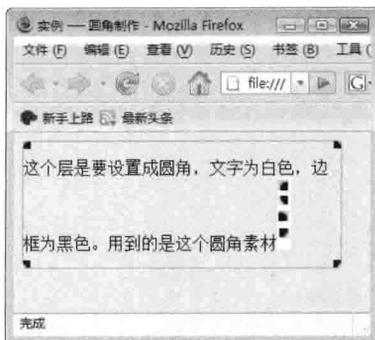


图 17-9 4 个圆角设置

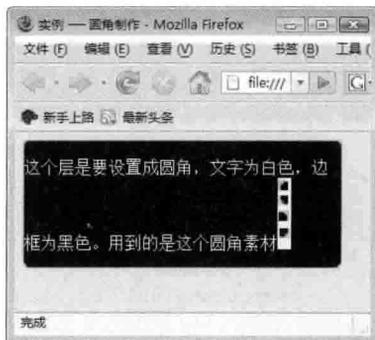


图 17-10 圆角最终效果

说明 用程序来实现圆角只是用四个不同的层，去覆盖四个角的位置，使视觉上达到圆角效果。

17.5 小结

本章主要介绍了对象的定位、定位的基本语法和层叠顺序设置，最后用实例演示了各个知识点。属性定位由 position 表示，除了可以用 z-index 控制层的层叠顺序外，还可以用 top、right、bottom 和 left 设置对象在父对象的定位。应注意一点，其后可以接负值。用实例对相对定位和绝对定位讲解。特别地对现在网页常用的圆角制作进行了详细讲解，更方便读者理解定位的属性知识。

17.6 习题

一、填空题

1. 定位属性由_____表示，用来设置对象的定位方式，有_____和_____之分。
2. 层叠属性类似于三维的 Z 轴，以对象为中心，前后移动，语法用_____表示。
3. 设置对象的边框位置用方向属性，方向属性就是表达上、右、下、左方向的属性，即_____表示顶端、_____表示右端、_____表示底端和_____表示左端。

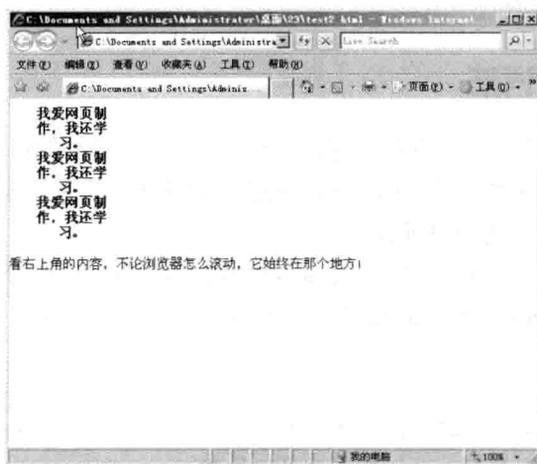
二、上机实践

1. 按下图制作一张网页，体会 absolute 的功能。

**【提示】**

position-----定位属性
absolute-----绝对定位

2. 按下图制作一张网页, 体会 relative 的功能。

**【提示】**

position-----定位属性
relative-----相对定位

第 18 章 用 CSS 控制列表和表格

读者或许已经对列表和表格这俩词有点儿印象，本章不再介绍列表和表格的 HTML 知识，而是讲解如何用 CSS 对列表和表格产生影响，以达到网页想表达的效果。

本章主要学习以下内容：

- 列表的基本语法；
- 设置列表中的图像；
- 列表中的文本排列；
- 列表的预设标记；
- 控制表格的基本语法；
- 设置表格边框独立或合并；
- 设置表格布局的算法；
- 用 CSS 控制表格实例。

18.1 列表的基本语法

本章所介绍的列表与本书第一篇中的列表是不一样的，本章不会从 HTML 角度介绍如何设置列表，而是用 CSS 属性来控制列表外观，通过样式对列表进行设置。但是有一点，本章中的列表是在 HTML 中的列表基础上进行设置的。

属性列表在 CSS 中用 `list-style` 表示，是复合属性，其下包含列表的图像 (`list-style-image`)、列表文本排列 (`list-style-position`) 和列表预设标记类型 (`list-style-type`)。属性 `list-style` 的完整语法表示如下。

```
/*列表属性*/  
list-style:list-style-image || list-style-position || list-style-type
```

列表属性后面接三个单属性，它们的作用如下所述。

- `list-style-image`：用来设置列表的图像。
- `list-style-position`：用来设置列表项标记如何根据文本排列。
- `list-style-type`：用来设置列表项所使用的预设标记。

上面的参数都是属性，在这些参数后面接的值，稍后作详细介绍，默认值为 `disc outside none`。在本节中只用实例来演示复合属性 `list-style` 如何使用，让读者对本复合属性有一个初步的认识。代码 18-1 表示了复合属性 `list-style` 的用法。

代码 18-1 list-style 用法

```
-----文件名：list-style用法.html-----  
01 <html>  
02 <head>
```

```

03 <title>list-style用法</title>
04 <style type="text/css">
05 #l1
06 {
07     /*设置边框*/
08     border:1px solid red;
09 }
10 #l2
11 {
12     /*设置边框和列表*/
13     border:1px solid red;
14     margin-top:5px;
15     /*列表设置*/
16     list-style:url("lsit1.gif") inside none;
17 }
18 </style>
19 </head>
20 <body>
21 <!--层中包含列表-->
22 <div id="l1">
23     这个层表示列表默认状态:
24     <ul>
25         <li>表项一</li>
26         <li>表项二</li>
27         <li>表项三</li>
28     </ul>
29 </div>
30 <div id="l2">
31     这个层表示设置列表后的状态:
32     <ul>
33         <li>表项一</li>
34         <li>表项二</li>
35         <li>表项三</li>
36     </ul>
37 </div>
38 </body>
39 </html>

```

【代码解析】在代码 18-1 中，用两个层表示了列表的用法。为了对比，第 19 行代码为第 1 层，是默认的 list-style 值（没有存在 list-style 属性，系统按默认值显示）；第 27 行代码为第 2 层，设置了 list-style，第 1 个属性 list-style-image 用自己的图像来显示，这样第 3 个属性的类型就设置成 none（由于对 list-style-image 为非 none，优先权低的 list-style-type 属性失效），第 2 个属性 list-style-position 设置成 inside，表示列表图像放置在文本以内，且环绕文本根据标记对齐。效果如图 18-1 所示。



图 18-1 list-style 用法

18.2 设置列表的图像

上一节介绍了列表 `list-style` 复合属性的语法，读者对 `list-style-image` 有了大概了解，现在正式来介绍这个可以自定义的列表图像的属性，其语法形式如下。

```
/*设置列表图像*/
list-style-image : none | url (url)
```

本属性有两个参数，其区别如下所述。

- `none`：表示属性不设置列表图像，是默认值。
- `url`：使用绝对或相对地址指定背景图像，这样就可以显示自定义的图像了，要注意图像大小，如太大会影响列表的美观（图像是按图像原始大小显示的）。图像链接地址要用括号括起来，然后用引号引起（也可以不用引号），注意括号前面有关键字 `url`。

在输入图像地址的时候，要检察是否为有效地址，如果图像不能正确显示，系统会按 `none` 处理，同时，若 `list-style-image` 属性为 `none` 或指定图像不可用时，`list-style-type` 属性将发生作用。代码 18-2 表示如何设置列表的图像。

代码 18-2 设置列表的图像

```
-----文件名：设置列表图像.html-----
01 <html>
02 <head>
03 <title>设置列表图像</title>
04 <style type="text/css">
05 #i1
06 {
07     /*设置边框*/
08     border:1px solid red;
09 }
10 #ii2
11 {
12     /*设置列表图像，引用list1.gif*/
13     list-style-image:url("list1.gif");
14 }
15 #ii3
16 {
17     /*设置列表图像，引用list2.gif*/
18     list-style-image:url(list2.gif);
19 }
20 </style>
21 </head>
22 <body>
23 <div id="i1">
24 <!--设置列表 -->
25 <ul>
26     <!--设置列表项-->
27     <li id="ii1">表列图像默认值none</li>
28     <li id="ii2">设置列表图像1</li>
```

```

29     <li id="ii3">设置列表图像2</li>
30 </ul>
31 </div>
32 </body>
33 </html>

```

【代码解析】 在一个列表中，有 3 个表项，分别设置了它们的 `list-style-image` 属性，第 27 行代码为表项一，表示默认值；第 28 行代码为表项二，设置属性 `list-style-image` 的值为图像 `list1.gif`，就与本网页同一目录；第 29 行代码为表项三，与表项二相差不多，只是图像文件不同，这样在一个列表中显示了 3 个不同的列表图像。效果如图 18-2 所示。

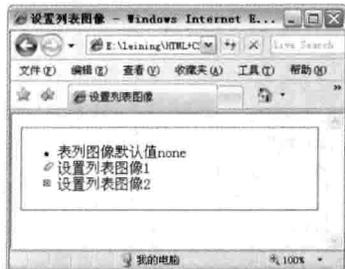


图 18-2 设置列表的图像

注意

复合属性 `list-style` 可以表示 `list-style-image` 的效果，只要设置参数 `url` 即可，其他两个参数都是默认值，系统会按默认值表示。

18.3 设置列表的文本排列

下面介绍复合属性第二个参数是如何设置列表的文本排列，即表示对象的列表标记如何根据文本排列。其语法如下，

```

/*CSS样式：设置列表文本排列*/
list-style-position : outside | inside

```

列表的文本排列表示在列表的表项是否在表项范围内显示，参数值设置有 `outside` 和 `inside`，下面对这两个参数值进行说明。

- `outside`：默认值。列表项目标记放置在文本以外，且环绕文本不根据标记对齐。
- `inside`：列表项目标记放置在文本以内，且环绕文本根据标记对齐。

下面用实例来演示列表的文本排列。在 HTML 标记中，用 `ul` 设置无序排列以及其表项中设置不同的内容，当关联样式设置 `list-style-position` 时，在网页中就可表现出文本排列的效果，如代码 18-3 所示。

代码 18-3 设置列表文本排列

```

-----文件名：设置列表文本排列.html-----
01 <html>
02 <head>
03 <title>设置列表文本排列</title>
04 <style type="text/css">
05 #i1
06 {
07     /*设置为默认的文本排列*/
08     border:1px solid red;
09 }
10 #ii2
11 {

```

```

12      /*设置文本内包含表项标记*/
13      list-style-position:outside;
14  }
15  #ii3
16  {
17      /*设置表项标记在文本外*/
18      list-style-position:inside;
19  }
20 </style>
21 </head>
22 <body>
23 <div id="i1">
24 <!--设置列表为无序列表-->
25 <ul>
26     <li id="ii1">列表标记在文本以外, 默认值outside</li>
27     <li id="ii2">列表标记在文本以内, 设置值为inside</li>
28     <li id="ii3">列表标记在文本以内, 设置值为inside</li>
29 </ul>
30 </div>
31 </body>
32 </html>

```

【代码解析】在 HTML 中用无序列表表示了 3 个不同的表项。当设置文本排列时，第 26 行代码是第一个表项，为默认值；第 44 行代码为第二个表项，设置 list-style-position 为 outside，效果与表项一一样；但表项三，在第 7 行代码设置 list-style-position 的参数值为 inside，表示在文本内显示表项标记。效果如图 18-3 所示。

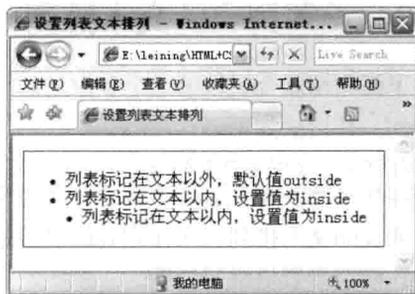


图 18-3 设置列表文本排列

18.4 列表的预设标记

预设标记是指表项中前面出现的标记，因为默认的是实心圆，所以在前面看到的表项标记都是实心圆，要想自定义标记效果，就要用 list-style-type 来设置。下面是表示各预设标记的参数表。

```

/*预设标记参数*/
list-style-type : disc | circle | square | decimal | lower-roman | upper-roman | lower-alpha
| upper-alpha | none

```

有了 list-style-type 的参数值，就可以设置自己想要的表项标记，而且还可以设置自定义图片作为预设标记使用。下面是各参数的用法。

- disc：默认值，表示实心圆。
- circle：表示空心圆。
- square：表示实心方块。
- decimal：表示阿拉伯数字。
- lower-roman：表示小写罗马数字。
- upper-roman：表示大写罗马数字。

- lower-alpha: 表示小写英文字母。
- upper-alpha: 表示大写英文字母。
- none: 表示不使用项目符号。

标记的使用根据实际项目而定,学好这些标记值是以后对其灵活应用的基础,特别是需要这些标记时,就可以用到项目上。代码 18-4 用实例演示了属性 list-style 中属性值的用法以及在网页中显示的效果。

代码 18-4 列表的预设标记用法

```
-----文件名:列表的预设标记用法.html-----
01 <html>
02 <head>
03 <title>列表的预设标记用法</title>
04 <style type="text/css">
05 #i1
06 {
07     /*设置边框*/
08     border:1px solid red;
09 }
10 #ii1
11 {
12     /*不使用标记符号*/
13     list-style-type :none;
14 }
15 #ii2
16 {
17     /*标记为空心圆*/
18     list-style-type :circle;
19 }
20 #ii3
21 {
22     /*实心方块*/
23     list-style-type :square;
24 }
25 #ii4
26 {
27     /*阿拉伯数字*/
28     list-style-type :decimal;
29 }
30 #ii5
31 {
32     /*小写罗马数字*/
33     list-style-type :lower-roman;
34 }
35 #ii6
36 {
37     /*大写罗马数字*/
38     list-style-type :upper-roman;
39 }
40 #ii7
41 {
42     /*小写英文字母*/
```

```

43     list-style-type :lower-alpha;
44 }
45 #ii8
46 {
47     /*大写英文字母*/
48     list-style-type :upper-alpha;
49 }
50 </style>
51 </head>
52 <body>
53 <div id="i1">
54 <ul>
55     <li>默认值disc, 实心圆</li>
56     <li id="ii1">设置为none时, 无符号显示</li>
57     <li id="ii2">设置为circle时, 空心圆</li>
58     <li id="ii3">设置为square时, 实心方块</li>
59     <li id="ii4">设置为decimal时, 阿拉伯数字</li>
60     <li id="ii5">设置为lower-roman时, 小写罗马数字</li>
61     <li id="ii6">设置为upper-roman时, 大写罗马数字</li>
62     <li id="ii7">设置为lower-alpha时, 小写英文字母</li>
63     <li id="ii8">设置为upper-alpha时, 大写英文字母</li>
64 </ul>
65 </div>
66 </body>
67 </html>

```

【代码解析】第 5~49 行代码用不同的样式来表现预设标记，这些标记都是 CSS 提供的属性值，在实际应用中，每个 list-style 的值只能接受一个属性值，效果如图 18-4 所示。

注意

列表的标记在参数值中选择一个值，作为列表的表项标记，取代默认列表标记。



图 18-4 列表的预设标记用法

18.5 列表实例

前面学习了列表的语法、列表中的图像以及列表的文本排列等，在 CSS 布局中，主要使用列表来布局，取代了以往的 TABLE 布局。而列表不管在 TABLE 或 DIV 布局中都可以用到，现在学的是 DIV 或 TABLE 中嵌套列表时所用到的 CSS 控制样式。

列表在 CSS 中是属性，表示为 list-style，其后所接的值表示列表的样式。在学习了列表的用法后，下面用实例来演示这些知识的用法，通过设置列表的标记和图像等来说明。下面用步骤实现每个知识点的介绍。

18.5.1 设置列表图像文字内显示

下面设置表项一为列表图像且文字内显示。首先新建一张网页，在 HTML 中把列表标记放到

标记<body></body>中，其中 HTML 中列表标记包含无序列表和有序列表之分，无序列表用标记对表示，有序列表用标记对表示，里面所包含的为列表项。不管是无序列表还是有序列表，列表项都用标记对表示。

当列表标记在 HTML 代码中布局好后，现在设置其默认的样式。下面代码在网页中表示无序列表，只有一个表项，且文本为“设置自定义图像，文字内显示”，然后能通过 CSS 代码控制，如代码 18-5 所示。

代码 18-5 设置列表图像且文字内显示

```
-----文件名：设置列表图像且文字内显示.html-----
01 <html>
02 <head>
03 <title>列表实例手把手</title>
04 <style type="text/css">
05 #l1
06 {
07     /*设置列表图像显示在文本内*/
08     list-style:url(list1.gif) inside;
09 }
10 </style>
11 </head>
12 <body>
13 <ul>
14     <li id="l1">
15         设置自定义图像,文字内显示
16     </li>
17 </ul>
18 </body>
19 </html>
```

【代码解析】在 HTML 标记中设置了无序列表，在第 13~17 行代码的列表中有一列表项，且在列表项中设置了 ID 选择符，表示用列表图像设置了表项的自定义图像，只有一个表项，效果如图 18-5 所示。

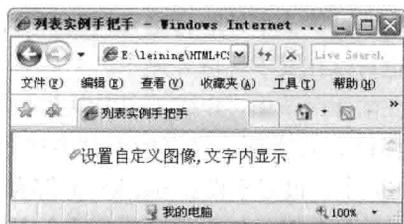


图 18-5 设置图像且文字内显示

18.5.2 设置列表样式为空心圆

在上一节的步骤中，设置了列表项中的图像在文本内显示，而图像是自定义图像。在本节中设置列表项为文本外显示，且列表项图像为空心圆，其中，空心圆是由 CSS 提供的参数。

在文字外显示空心圆，需要设置 list-style 的属性值为空心圆形。代码 18-6 表示空心圆的用法，在上一节的代码基础上，添加了一列表项，通过设置 list-style 属性来表现列表项的空心圆。

代码 18-6 文字外且空心圆显示

```
-----文件名：文字外且空心圆显示.html-----
01 <html>
02 <head>
03 <title>列表实例手把手</title>
04 <style type="text/css">
```

```

05 #11
06 {
07     list-style:url(list1.gif) inside;
08 }
09 #12
10 {
11     /*设置列表图像为空心圆*/
12     list-style:circle;
13 }
14 </style>
15 </head>
16 <body>
17 <ul>
18     <li id="11">
19     设置自定义图像,文字内显示</li>
20     <li id="12">
21     文字外显示,空心圆显示
22     </li>
23 </ul>
24 </body>
25 </html>

```

【代码解析】第 12 行代码将 list-style 设置为 circle, 表示空心圆, 在来自定义基础上加上另一表项, 与列表项一比较。第 20 行代码为第 2 个列表项在前面显示, 即在文本外显示, 而列表项一为文本内显示。效果如图 18-6 所示。

18.5.3 设置文本外自定义列表图像

本节设置文本外的自定义列表图像, 对其他值保持默认。列表项图像可以在文本内的区域内显示, 也可以在文本区域外显示。

在前面两节中, 介绍了列表中文本内图像和文本外图像。在本节中, 设置了第三个表项的自定义的图像, 对表项一设置的是文本排列为文本外的图像。文本外的设置, 即为默认值, 在属性 list-style 中, 设置不设置 outside 值都表示文本外。代码 18-7 中表示设置表项文本外的图像, 用属性 list-style 的默认值表示表项中的自定义图像。

代码 18-7 只设置图像

```

-----文件名: 只设置图像.html-----
01 <html>
02 <head>
03 <title>列表实例手把手</title>
04 <style type="text/css">
05 #11
06 {
07     /*设置文本排列*/
08     list-style:url(list1.gif) inside;
09 }
10 #12

```

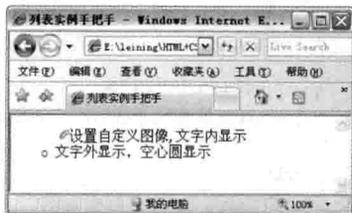


图 18-6 文字外且空心圆显示

```

11  {
12      /*设置为空心圆*/
13      list-style:circle;
14  }
15  #l3
16  {
17      /* 设置自定义图像*/
18      list-style:url(list2.gif);
19  }
20  </style>
21  </head>
22  <body>
23  <ul>
24      <li id="l1">
25          设置自定义图像,文字内显示</li>
26      <li id="l2">
27          文字外显示,空心圆显示
28      </li>
29      <li id="l3">
30          设置图像,其他默认
31      </li>
32  </ul>
33  </body>
34  </html>

```

【代码解析】在代码 18-7 中，表项一设置的还是原来的样式，但在表项三中，添加了自定义标记（第 18 行代码），同时表项一的文本排列为内置到文本（第 8 行代码），其他两表项都是文本外显示，效果如图 18-7 所示。



图 18-7 只设置图像

18.6 控制表格的基本语法

为了与表格的语法区别开，本节介绍的是用 CSS 代码控制表格的 CSS 语法。这些语法可以控制表格的行与单元格边框是独立还是合并以及表格的布局算法，这样就可以通过 CSS 来控制表格了。如果会 JavaScript 语言的话，也可以用 js（JavaScript 的简写）来控制 CSS，再从网页上显示效果。

下面是控制表格样式的属性，语法用 border-collapse 表示，其后接的值在 separate 和 collapse 中选择一个，默认的是前者，表现出的效果正如在 HTML 中的表格所表现的效果一样，以下是其语法的完整表达。

```
border-collapse : separate | collapse
```

属性 border-collapse 的第一个值 separate，是本属性的默认值，表示边框独立；而后面的值为 collapse，表示相邻边被合并。前者表示有边框，后者表示合并了边框，很容易区分其功能。

下面再介绍一个属性——table-layout，表示布局固定的算法，也是控制表格属性常用的属性之一，如下表示其完整语法。

```
table-layout : auto | fixed
```

属性 `table-layout` 后面可接两值：前者为 `auto`，默认的自动算法，布局将基于各单元格的内容，表格在每一单元格内所有内容读取计算之后才会显示出来；后者为 `fixed`，固定布局的算法。在这种算法中，表格和列的宽度取决于 `col` 对象的宽度总和。假如没有指定，则会取决于第一行每个单元格的宽度。假如表格没有指定宽度（`width`）属性，则表格被呈递的默认宽度为 100%。

下面用实例来演示上面两属性的用法，通过这个演示，要读者在实践中多总结其用法，如代码 18-8 所示。

代码 18-8 控制表格的基本语法

```
-----文件名：控制表格的基本语法.html-----
01 <html>
02 <head>
03 <title>控制表格的基本语法</title>
04 <style type="text/css">
05 #t1
06 {
07     /*设置边框为合并*/
08     border-collapse:collapse ;
09 }
10 #t2
11 {
12     /*设置边框合并和算法固定*/
13     border-collapse:collapse;
14     table-layout:fixed;
15 }
16 #t3
17 {
18     /*设置边框颜色独立与算法自动*/
19     border-collapse:separate;
20     table-layout:auto;
21 }
22 </style>
23 </head>
24 <body>
25 <table id="t3" bgcolor="red" >
26     <tr>
27         <td bgcolor="#cccccc">单元格一</td><td bgcolor="#cccccc">单元格二</td>
28     </tr>
29     <tr>
30         <td bgcolor="#cccccc">单元格三</td><td bgcolor="#cccccc">单元格四</td>
31     </tr>
32 </table>
33 <br>
34 <table id="t1" bgcolor="red" >
35     <tr>
36         <td bgcolor="#cccccc">单元格一</td><td bgcolor="#cccccc">单元格二</td>
37     </tr>
38     <tr>
39         <td bgcolor="#cccccc">单元格三</td><td bgcolor="#cccccc">单元格四</td>
40     </tr>
41 </table>
42 <br>
```

```

43 <table id="t2" bgcolor="red" >
44     <tr>
45         <td bgcolor="#cccccc">单元格一</td><td bgcolor="#cccccc">单元格二</td>
46     </tr>
47     <tr>
48         <td bgcolor="#cccccc">单元格三</td><td bgcolor="#cccccc">单元格四</td>
49     </tr>
50 </table>
51 </body>
52 </html>

```

【代码解析】在代码 18-8 中，显示了 3 个表格。第 25 行代码为第一个表格，存在独立的边框，其他两个表格是合并边框的，但在布局固定上，第 34 行代码为表格二，设置 table-layout 属性为 fixed，而第 43 行代码为表格三，设置 table-layout 为 auto，效果如图 18-8 所示。

技巧

使用 CSS 对表格进行控制时，可以直接在样式文件中写 TABLE，这时代表在本样式表有效果的区域中所有表格 (table) 都以此样式为准，代码 18-1 在 table 标记中定义了 ID 表示只控制该 ID 的表格样式。在前面的实例中，编者常用 DIV 控制所有的 DIV 标记样式。

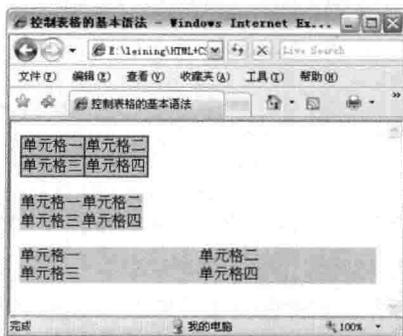


图 18-8 控制表格的基本语法

18.7 设置表格边框独立或合并

在上一节中简单地介绍了表格中的两个重要属性，其中一个就是本节要详细介绍的表格边框独立与合并。与前面介绍的内容一样，本属性应用在表格是否显示边框，如果要显示其边框，就设置为 separate；否则，设置为 collapse。参数设置如下。

```
border-collapse : separate | collapse
```

属性 border-collapse 的应用主要是表格上，即 HTML 代码中出现表格标记对，然后用 CSS 控制表格边框是否合并。同时，如果边框为独立显示时，也根据边框的宽度来显示，即边框宽度为 5 时，在表格中也显示为 5。边框宽度越大，显示也越宽。相反，如果设置的是 collapse，则边框不管设置何值，都没有显示，即合并了边框的宽度。代码 18-9 表示表格边框独立与合并的用法。

代码 18-9 设置表格边框独立或合并

```

-----文件名：设置表格边框独立或合并.html-----
01 <html>
02 <head>
03 <title>设置表格边框独立或合并</title>
04 <style type="text/css">
05 #t1
06 {
07     /*设置为独立显示*/
08     border-collapse: separate;

```

```

09  }
10  #t2
11  {
12      /*设置为合并显示*/
13      border-collapse:collapse;
14  }
15  </style>
16  </head>
17  <body>
18  border-collapse属性设置separate时，为默认值。
19  <table id="t1" bgcolor="red" >
20      <tr>
21          <td bgcolor="#cccccc">单元格一;HTML+CSS完全自学手册</td><td
22  bgcolor="#cccccc">单元格二;HTML+CSS完全自学手册</td>
23      </tr>
24      <tr>
25          <td bgcolor="#cccccc">单元格三;HTML+CSS完全自学手册</td><td
26  bgcolor="#cccccc">单元格四;HTML+CSS完全自学手册</td>
27      </tr>
28      <tr>
29          <td bgcolor="#cccccc">单元格五;HTML+CSS完全自学手册</td><td
30  bgcolor="#cccccc">单元格六;HTML+CSS完全自学手册</td>
31      </tr>
32  </table>
33  当border-collapse属性设置为collapse时:
34  <table id="t2" bgcolor="red" >
35      <tr>
36          <td bgcolor="#cccccc">单元格一;HTML+CSS完全自学手册</td><td
37  bgcolor="#cccccc">单元格二;HTML+CSS完全自学手册</td>
38      </tr>
39      <tr>
40          <td bgcolor="#cccccc">单元格三;HTML+CSS完全自学手册</td><td
41  bgcolor="#cccccc">单元格四;HTML+CSS完全自学手册</td>
42      </tr>
43      <tr>
44          <td bgcolor="#cccccc">单元格五;HTML+CSS完全自学手册</td><td
45  bgcolor="#cccccc">单元格六;HTML+CSS完全自学手册</td>
46      </tr>
47  </table>
48  </body>
49  </html>

```

【代码解析】在代码 18-9 中，用两个表格表示 border-collapse 的用法。其中第 19~32 行代码为表格一，设置值为 separate，在网页中效果显示红色边框，这是在整个表格的背景，而表格中单元格的填充色为 #cccccc，使边框背景显现出来；同理，在第 34~47 行代码的表格二中，背景颜色同样设置为红色，表格中的数据单元格设置为 collapse，表达的是合并单元格的边框，在网页中显示为无边框。效果如图 18-9 所示。

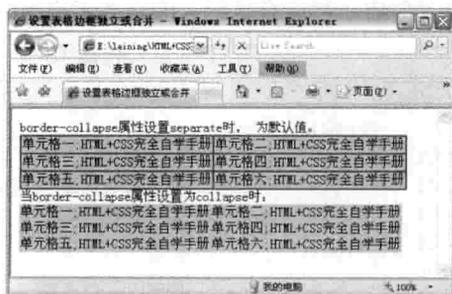


图 18-9 设置表格边框独立或合并

18.8 设置表格布局的算法

设置表格的布局算法在网页应用中比较重要，特别是用表格来布局时，表格的宽度是否要设置为当前浏览器的宽度。要设置好表格的布局算法，将用到布局算法的属性，表格布局算法用 `table-layout` 表示，其后接 `auto` 值，表示默认，或者接 `fixed`。其完整语法如下表示。

```
table-layout : auto | fixed
```

参数 `auto` 为默认值，为默认的自动算法，布局将基于各单元格的内容。表格在每一单元格内所有内容读取计算之后才会显示出来。而另一参数值 `fixed` 表示固定布局的算法，在这种算法中，表格和列的宽度取决于 `col` 对象的宽度总和，假如没有指定，则会取决于第一行每个单元格的宽度，假如表格没有指定宽度 (`width`) 属性，则表格被呈递的默认宽度为 100%。

利用布局算法来设置表格的宽度，是用 TABLE 布局网页的重要技巧，能表现出布局的宽度自适应性。下面用实例来比较其用法，表格一设置 `table-layout` 为 `auto`，表格二设置其为 `fixed` 参数，如代码 18-10 所示。

代码 18-10 设置表格布局的算法

```
-----文件名: 设置表格布局的算法.html-----
01 <html>
02 <head>
03 <title>设置表格布局的算法</title>
04 <style type="text/css">
05 #t1
06 {
07     /*设置表格布局算法auto*/
08     table-layout: auto;
09 }
10 #t2
11 {
12     /*设置表格布局算法fixed*/
13     table-layout: fixed;
14 }
15 </style>
16 </head>
17 <body>
18 table-layout属性设置为auto时,为默认值。
19 <table id="t1" bgcolor="red" >
20     <tr>
21         <td bgcolor="#cccccc">单元格一;HTML+CSS完全自学手册</td><td
22 bgcolor="#cccccc">单元格二;HTML+CSS完全自学手册</td>
23     </tr>
24     <tr>
25         <td bgcolor="#cccccc">单元格三;HTML+CSS完全自学手册</td><td
26 bgcolor="#cccccc">单元格四;HTML+CSS完全自学手册</td>
27     </tr>
28     <tr>
29         <td bgcolor="#cccccc">单元格五;HTML+CSS完全自学手册</td><td
30 bgcolor="#cccccc">单元格六;HTML+CSS完全自学手册</td>
31     </tr>
32 </table>
```

```

33  当table-layout属性设置为fixed时:
34  <table id="t2" bgcolor="red" >
35      <tr>
36          <td bgcolor="#cccccc">单元格一;HTML+CSS完全自学手册</td><td
37  bgcolor="#cccccc">单元格二;HTML+CSS完全自学手册</td>
38      </tr>
39      <tr>
40          <td bgcolor="#cccccc">单元格三;HTML+CSS完全自学手册</td><td
41  bgcolor="#cccccc">单元格四;HTML+CSS完全自学手册</td>
42      </tr>
43      <tr>
44          <td bgcolor="#cccccc">单元格五;HTML+CSS完全自学手册</td><td 45
45  bgcolor="#cccccc">单元格六;HTML+CSS完全自学手册</td>
46      </tr>
47  </table>
48  </body>
49  </html>

```

【代码解析】在上述代码中设置了表格布局的算法，其中第 19~32 行代码为第 1 个表格中设置 table-layout 属性值为默认值 auto，而第 34~47 行代码为第 2 个表格中设置属性值为 fixed。上面代码布局表格算法在网页中的效果如图 18-10 所示。



图 18-10 设置表格布局的算法

18.9 CSS 控制表格实例

经过前面各节的学习，理解了用 CSS 控制表格的基本语法，然后分别介绍了 border-collapse 属性和 table-layout 的用法，本节再来总结两属性的用法，通过实例比较这两个属性在表格中的应用。

本实例分三步，每个步骤演示出不同的属性和功能。从 HTML 开始设置表格，再到 CSS 控制表格，步骤如下所示。

18.9.1 布局好表格

新建一个网页，在标记对 <body></body> 中，插入表格标记对 <table></table>，并在表格中设置多行多列表格数据，设置表格中的背景颜色为 blue，设置表格中的表元背景颜色值为 #333333，这样通过用不同的背景颜色来区别单元格和边框，即颜色为 blue 的为边框的颜色，如代码 18-11 所示。

代码 18-11 区别单元格和边框

```

-----文件名：区别单元格和边框.html-----
01 <html>
02 <head>
03 <title>CSS控制表格实例手把手</title>
04 <style type="text/css">
05 table
06 {
07     /*设置表格中文本颜色*/
08     color:red;
09 }
10 </style>
11 </head>
12 <body>
13 <table bgcolor="blue">
14     <tr>
15         <td bgcolor="#333333">CSS样式控制表格1</td><td bgcolor="#333333">CSS样
16 式控制表格2</td><td bgcolor="#333333">CSS样式控制表格3</td>
17     </tr>
18     <tr>
19         <td bgcolor="#333333">CSS样式控制表格4</td><td bgcolor="#333333">CSS样
20 式控制表格5</td><td bgcolor="#333333">CSS样式控制表格6</td>
21     </tr>
22     <tr>
23         <td bgcolor="#333333">CSS样式控制表格7</td><td bgcolor="#333333">CSS样
24 式控制表格8</td><td bgcolor="#333333">CSS样式控制表格9</td>
25     </tr>
26 </table>
27 </body>
28 </html>

```

【代码解析】在上述代码中，第 13~26 行代码列出了表格一，且背景颜色为 blue，通过三行三列的单元格表示，且输入不同文本的内容，用颜色#333333 来填充单元格，所看到的颜色呈现出边框效果，如图 18-11 所示。

18.9.2 表格边框独立和合并

本步骤介绍对表格的边框设置不同的表格边框独立和合并。通过前面的介绍可以看到表格边框的效果，本步骤再添加一个表格标记，且文本内容设置为一样，但设置表格的样式不一样。

在第一个表格中，设置表格属性 border-collapse 为 separate，表示表格的独立边框，在网页中可以看到各表格的表元用边框隔开；第二个表格用同样的表格文本，在设置表格中的 border-collapse 属性为 collapse 时，两者效果就不一样了。在第二个表格中，所表示的是表格的边框合并，即表格中的表元之间没有边框显示。代码 18-12 演示了边框独立与合并的用法。



图 18-11 区别单元格和边框

代码 18-12 设置不同的表格边框独立和合并

```
-----文件名：设置不同的表格边框独立和合并.html-----
01 <html>
02 <head>
03 <title>CSS控制表格实例手把手</title>
04 <style type="text/css">
05 table
06 {
07     /*表格中的字体颜色*/
08     color:red;
09 }
10 #t1
11 {
12     /*边框独立*/
13     border-collapse:separate;
14 }
15 #t2
16 {
17     /*边框合并*/
18     border-collapse:collapse
19 }
20 </style>
21 </head>
22 <body>
23 <table bgcolor="blue" id="t1">
24     <tr>
25         <td bgcolor="#333333">CSS样式控制表格1</td><td bgcolor="#333333">CSS样
26 式控制表格2</td><td bgcolor="#333333">CSS样式控制表格3</td>
27     </tr>
28     <tr>
29         <td bgcolor="#333333">CSS样式控制表格4</td><td bgcolor="#333333">CSS样
30 式控制表格5</td><td bgcolor="#333333">CSS样式控制表格6</td>
31     </tr>
32     <tr>
33         <td bgcolor="#333333">CSS样式控制表格7</td><td bgcolor="#333333">CSS样
34 式控制表格8</td><td bgcolor="#333333">CSS样式控制表格9</td>
35     </tr>
36 </table>
37 <br>
38 <table bgcolor="blue" id="t2">
39     <tr>
40         <td bgcolor="#333333">CSS样式控制表格1</td><td bgcolor="#333333">CSS样
41 式控制表格2</td><td bgcolor="#333333">CSS样式控制表格3</td>
42     </tr>
43     <tr>
44         <td bgcolor="#333333">CSS样式控制表格4</td><td bgcolor="#333333">CSS样
45 式控制表格5</td><td bgcolor="#333333">CSS样式控制表格6</td>
46     </tr>
47     <tr>
```

```

48      <td bgcolor="#333333">CSS样式控制表格7</td><td bgcolor="#333333">CSS样
49 式控制表格8</td><td bgcolor="#333333">CSS样式控制表格9</td>
50    </tr>
51  </table>
52 </body>
53 </html>

```

【代码解析】在代码 18-12 中，第 38~51 行代码添加了一个与表格一内容相同的表格，但 border-collapse 的值设置为 collapse 后，与表格一的效果就区别开了，通过设置 separate 的属性值表示边框的独立，而设置 collapse 的属性值表示边框的合并，即由表格二所表现的效果。两者表现出来的效果如图 18-12 所示。

18.9.3 设置表格算法为自动和固定

下面介绍设置表格布局的算法为自动和固定。通过前面知识的介绍，用 CSS 控制表格算法为自动在 table-layout 属性后接 auto，而设置 table-layout 属性的属性值为 fixed 时，表示表格算法的固定。

在 t1 中加入布局算法为自动，而在 t2 中加入布局算法为固定。下面代码表示在样式表的基本上进行修改，其他 HTML 标记都没变化，只有样式加上 table-layout 属性。通过修改样式文件后的样式代码如下所示。

```

01 #t1
02 {
03     border-collapse:separate;
04     table-layout:auto;
05 }
06 #t2
07 {
08     border-collapse:collapse ;
09     table-layout:fixed;
10 }

```

【代码解析】在代码 18-12 的基础上，添加上面的代码，主要从 CSS 中对样式进行修改。第 4 行代码对于表格一设置 table-layout 的值为 auto，表示默认值，而第 9 行代码对于表格二设置 table-layout 的值为 fixed，表示对布局算法的固定。通添加样式后，对 HTML 中的两个表格算法控制效果如图 18-13 所示。



图 18-12 设置不同的表格边框独立和合并



图 18-13 设置表格布局的算法为自动和固定

综上所述,应用好 border-collapse 和 table-layout 对表格处理非常有用且非常方便,能根据实际应用设置好表格的样式。

18.10 小结

本章讲解了 CSS 中列表的用法,首先介绍列表的所有属性语法,其次详细介绍列表各属性并用实例演示了其用法,如列表的图像主要演示了对象的列表项标记的图像,列表的文本排列演示了对象的列表项标记如何根据文本排列,列表的预设标记主要演示了对象的列表项所使用的预设标记。

本章后半部分介绍了如何用 CSS 来控制表格,在表格标记中,可以用 border-collapse 来控制单元格边框与行的独立或合并,同时还可以由 table-layout 来设置表格布局,除了设置自动外还可以为 fixed,水平布局是仅仅基于表格的宽度、表格边框的宽度、单元格间距、列的宽度为自适应性等。

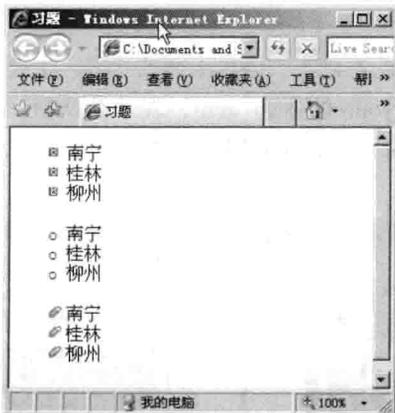
18.11 习题

一、填空题

1. 属性列表在 CSS 中用_____表示,是复合属性,其下包含列表的图像_____、列表文本排列_____和列表预设标记类型_____。
2. 属性 table-layout 后面可接两个值:前者为_____,默认的自动算法,布局将基于各单元格的内容;后者为_____,固定布局的算法。

二、上机实践

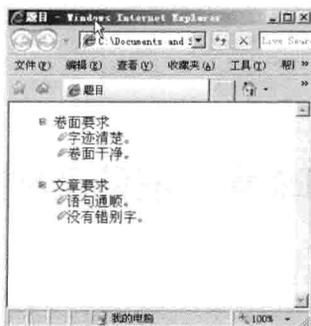
1. 按下图编辑一个网页。



【代码解析】

list-style:url -----列表样式自定义图片

2. 按下图编辑一个网页。

**【代码解析】**

Inside-----列表缩进

3. 按下图编辑一个网页。

**【提示】**

border-collapse:-----表格的边框是否被合并属性

table-layout:-----表格布局属性

4. 按下图编辑一个网页。

**【提示】**

border-collapse:-----表格的边框是否被合并属性

table-layout:-----表格布局属性

第 19 章 滚动条和布局

滚动条属性通过 CSS 的控制，可以让用户体验滚动效果。注意，这些属性只在 IE 中使用（IE 是 Internet Explorer 的简称，Mozilla 的 Firefox 不支持滚动属性）。用 CSS 布局是目前主流布局方式，也是 W3C 标准支持的布局方式，学好 CSS 布局对网站的设计、制作有非常大的好处。本章介绍布局中常用到的一些属性，这些属性非常重要，也是在布局中常用到的几个属性，这些属性使网页增添不少色彩。

本章主要涉及以下知识点：

- 设置滚动条亮边框颜色；
- 设置滚动条 3D 界面；
- 设置滚动条的拖动区域颜色；
- 设置层的漂移；
- 清除层的漂移范围；
- 设置层的可视区域；
- 鼠标指针的不同表示；
- 设置对象可见性。

19.1 滚动条

滚动条在布局时应用比较多，当然，当网页的高度超过可视高度时，网页中自动会出现滚动条，这是默认状态。现在也可以模拟出这种效果，即在层中设置宽度和高度，当其内容超过其高度时，这时会出现滚动条，不管是自动的滚动条还是层设置的，都可以对滚动条的效果进行设置。本章学习滚动条中的特效属性，对这些效果进行总结，但只是应用在 IE 浏览器中。

19.1.1 滚动条的基本语法

从总体来讲，对滚动条的设置主要在颜色上，如设置滚动条亮边框和暗边框、拖动区颜色等。下面将要学习的滚动条属性，其属性值是颜色值，从以前的学习中可知道，颜色值可分成不同的表达方式。滚动条属性列表如下所述。

```
scrollbar-3dlight-color:color  
scrollbar-highlight-color:color  
scrollbar-face-color:color  
scrollbar-arrow-color:color  
scrollbar-shadow-color:color  
scrollbar-darkshadow-color:color
```

```
scrollbar-base-color:color
scrollbar-track-color:color
```

属性各参数的含义如下所述。

- scrollbar-3dlight-color: 滚动条亮边框颜色。
- scrollbar-highlight-color: 滚动条 3D 界面的亮边颜色。
- scrollbar-face-color: 滚动条 3D 表面的颜色。
- scrollbar-arrow-color: 滚动条方向箭头的颜色。
- scrollbar-shadow-color: 滚动条 3D 界面的暗边框颜色。
- scrollbar-darkshadow-color: 滚动条暗边框颜色。
- scrollbar-base-color: 滚动条基准颜色。
- scrollbar-track-color: 滚动条的拖动区域颜色。

上面的属性表示滚动条的样式，主要设置滚动条的颜色，通过颜色的不同，表示出滚动条的风格，同时也可以设置滚动条是否显示。

上面表示的是滚动条的语法，下面通过各个语法点来介绍表示滚动条的语法属性的用法，给出实例，并注意各属性的区别。

19.1.2 设置滚动条亮边框颜色

当出现滚动条时，即可通过滚动条的 CSS 属性对滚动条进行控制。通过上面的知识点介绍可知，滚动条的设置主要出现在有滚动条出现的情况，而且在 IE 下才支持，最关键的是在滚动条的效果上，对滚动条的调协主要部分是用过颜色来表现。可见，在滚动条中可以显示出不同的颜色效果。

滚动条亮边框颜色表示滚动条的边框颜色，是用在 IE 中的属性，属性值为颜色值，其属性在语法上表示如下。

```
scrollbar-3dlight-color:color
```

其值所接的是表示滚动条亮边框颜色，颜色值有三种表示方法，在前面章节已介绍过，下面实例所设置的颜色值为颜色名称的形式，如代码 19-1 所表示的是设置滚动条亮边框颜色为红色。

代码 19-1 设置滚动条亮边框颜色

```
-----文件名：设置滚动条亮边框颜色.html-----
01 <html>
02 <head>
03 <title>设置滚动条亮边框颜色</title>
04 <style type="text/css">
05 #s1
06 {
07     /*设置宽度和高度*/
08     width:200px;
09     height:100px;
10     /*超过区域时显示滚动条*/
11     overflow:scroll;
12     /*设置滚动条亮边颜色*/
13     scrollbar-3dlight-color:red;
14 }
```

```

15 </style>
16 </head>
17 <body>
18 <div id="s1">
19     HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手
20 册.HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手册
21 </div>
22 </body>
23 </html>

```

【代码解析】第 13 行代码设置好滚动条亮边框颜色为红色时，只需要在层中设置 scrollbar-3dlight-color 为红色 (red) 就可以在网页中看到滚动条的边框颜色，颜色只出现在边框上，效果如图 19-1 所示。

19.1.3 设置 3D 界面亮边颜色和表面颜色

滚动条的效果可以用 3D 效果表现出来，还可以在其中设置不同的颜色，而且还有亮边框和表面等颜色的设置，本节介绍上述内容。

设置 3D 界面亮边颜色和表面颜色表示滚动条 3D 界面的亮边 (ThreedHighlight) 颜色，其语法形式如下。

```
scrollbar-highlight-color:color
```

以上代码表示 3D 界面的亮边颜色，其后的属性值为颜色值。与亮边属性一样，3D 界面表面颜色属性所接的值也是颜色值，其语法形式如下。

```
scrollbar-face-color:color
```

其属性值接颜色值，表示设置 3D 界面亮边颜色和表面颜色。代码 19-2 表示属性 scrollbar-highlight-color 的用法。通过设置 scrollbar-highlight-color 的颜色与 scrollbar-face-color 的颜色来表示两者的用法。

代码 19-2 设置 3D 界面亮边颜色和表面颜色

```

-----文件名：设置3D界面亮边颜色和表面颜色.html-----
01 <html>
02 <head>
03 <title>设置3D界面亮边颜色和表面颜色</title>
04 <style type="text/css">
05 div
06 {
07     /*设置层边框*/
08     border:1px solid blue;
09 }
10 #s1
11 {
12     /*设置层高度和宽度还有滚动条*/
13     width:200px;
14     height:100px;
15     overflow:scroll;

```

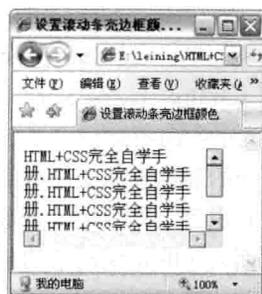


图 19-1 设置滚动条亮边框颜色

```

16      /*设置3D亮边颜色*/
17      scrollbar-highlight-color:red;
18  }
19  #s2
20  {
21      width:200px;
22      height:100px;
23      overflow:scroll;
24      /*设置滚动条表面颜色*/
25      scrollbar-face-color :red;
26  }
27  </style>
28  </head>
29
30  <body>
31  下面是3D界面亮边颜色:
32  <div id="s1">
33      HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手
34  册.HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手册
35  </div><br>
36  下面是3D表面颜色
37  <div id="s2">
38      HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手
39  册.HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手册
40  </div>
41  </body>
42  </html>

```

【代码解析】 属性 scrollbar-highlight-color 表示 3D 界面亮边颜色，在第 17 行代码设置的值为红色，而属性 scrollbar-face-color 表示 3D 界面表面颜色，在第 25 行代码设置的值也为红色，但用不同的两个滚动条表示，如图 19-2 所示。

注意 以上设置在 FF 浏览器中不支持。

19.1.4 设置滚动条方向箭头颜色

滚动条有表示方向的箭头，当单击箭头时，文档将会向箭头方面移动。通过 CSS 属性可以对箭头设置颜色，当然，只有滚动条出现时才会有颜色的出现。下面介绍滚动条方向箭头的颜色。

设置滚动条方向箭头颜色首先要在 IE 的效果中设置出现滚动条的方向箭头，其次如果层中有滚动条，就会出现设置好颜色的滚动条方向箭头，否则，不会出现箭头。滚动条方向箭头颜色语法形式如下。

```
scrollbar-arrow-color:color
```

属性 scrollbar-arrow-color 表示滚动条方向箭头的颜色，其后接的值为颜色值。代码 19-3 演示



图 19-2 设置 3D 界面亮边颜色和表面颜色

了其属性的用法。

代码 19-3 设置滚动条方向箭头的颜色

```
-----文件名: 设置滚动条方向箭头的颜色.html-----
01 <html>
02 <head>
03 <title>设置滚动条方向箭头的颜色</title>
04 <style type="text/css">
05 #s1
06 {
07     width:200px;
08     height:100px;
09     overflow:scroll;
10     /*设置滚动条的箭头颜色*/
11     scrollbar-arrow-color:red;
12 }
13 </style>
14 </head>
15
16 <body>
17 <div id="s1">
18     HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手
19     册.HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手册
20 </div>
21 </body>
22 </html>
```

【代码解析】在第 11 行代码设置了 scrollbar-arrow-color 的值，表示设置滚动条方向箭头的颜色，当层中出现滚动条时，那么两端的箭头就显示出来，且颜色是在 CSS 代码中设置好的颜色值，效果如图 19-3 所示。

注意

在图 19-3 中，因左右两边没有出现滚动条，故不会出现方向箭头的颜色。



图 19-3 设置滚动条方向箭头的颜色

19.1.5 设置滚动条 3D 界面暗边颜色和暗边框颜色

除了上面介绍的亮边外，还可以设置滚动条的暗边颜色，与之相区别的是，暗边框颜色与暗边颜色不同，下面来介绍。

本节介绍设置滚动条 3D 界面暗边颜色和暗边框颜色。滚动条 3D 界面暗边颜色与暗边框颜色表示方法不同，滚动条 3D 界面暗边颜色语法表示如下。

```
scrollbar-shadow-color:color
```

上述代码表示滚动条 3D 界面暗边颜色，接的值为颜色值。与之一样的是滚动条 3D 界面暗边框颜色，表示暗边框中的颜色，其语法形式如下表示。

```
scrollbar-darkshadow-color:color
```

该属性表示暗边框颜色，两者对比，区别不是很大。

在代码 19-4 中，设置滚动条 3D 界面暗边颜色和 3D 界面暗边框颜色的用法。

代码 19-4 设置滚动条 3D 界面暗边颜色和暗边框颜色

```
-----文件名：设置滚动条3D界面暗边颜色和暗边框颜色.html-----
01 <html>
02 <head>
03 <title>设置滚动条3D界面暗边颜色和暗边框颜色</title>
04 <style type="text/css">
05 div
06 {
07     border:1px solid blue;
08 }
09 #s1
10 {
11     width:200px;
12     height:100px;
13     overflow:scroll;
14     /*设置暗边颜色*/
15     scrollbar-shadow-color:red;
16 }
17 #s2
18 {
19     width:200px;
20     height:100px;
21     overflow:scroll;
22     /*设置暗边框颜色*/
23     scrollbar-darkshadow-color :red;
24 }
25 </style>
26 </head>
27
28 <body>
29     下面是3D界面暗边框颜色：
30 <div id="s1">
31     HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手
32 册.HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手册
33 </div><br>
34     下面是暗边框颜色
35 <div id="s2">
36     HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手
37 册.HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手册
38 </div>
39 </body>
40 </html>
```

【代码解析】第 15、23 行代码分别表示了滚动条 3D 界面暗边颜色和暗边框颜色，在第 1 层

中用 `scrollbar-shadow-color` 表示滚动条 3D 界面暗边颜色, 而第 2 层表示的是 `scrollbar-darkshadow-color` 属性, 其表示为暗边框颜色, 效果如图 19-4 所示。

19.1.6 设置滚动条基准颜色和拖动区域颜色

滚动条的基准颜色是指滚动条中全部区域的颜色, 而拖动区域的颜色是指滚动条能拖动的区域的颜色, 通过设置不同颜色值来表示。下面介绍这两种属性的用法。

本节介绍滚动条基准颜色和拖动区域颜色, 两者的区别在于一个是滚动条基准部分, 而另一个表示拖动区部分。滚动条基准颜色的语法如下表示。

```
scrollbar-base-color: color
```

属性 `scrollbar-base-color` 表示滚动条基准颜色。滚动条的拖动区域颜色语法形式如下表示。

```
scrollbar-track-color: color
```

属性 `scrollbar-track-color` 表示滚动条的拖动区域颜色。代码 19-5 表示滚动条基准颜色和拖动区域颜色的用法。

代码 19-5 设置滚动条基准颜色和拖动区域颜色

```
-----文件名: 设置滚动条基准颜色和拖动区域颜色.html-----
01 <html>
02 <head>
03 <title>设置滚动条基准颜色和拖动区域颜色</title>
04 <style type="text/css">
05 div
06 {
07     border:1px solid blue;
08 }
09 #s1
10 {
11     width:200px;
12     height:100px;
13     overflow:scroll;
14     /*设置滚动条的基准颜色*/
15     scrollbar-base-color:red;
16 }
17 #s2
18 {
19     width:200px;
20     height:100px;
21     overflow:scroll;
22     /*设置滚动条拖动区域颜色*/
23     scrollbar-track-color:red;
24 }
25 </style>
26 </head>
27
```



图 19-4 设置滚动条 3D 界面暗边颜色和暗边框颜色

```

28 <body>
29 下面是滚动条基准颜色:
30 <div id="s1">
31     HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手
32 册.HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手册
33 </div><br>
34 下面是拖动区域颜色:
35 <div id="s2">
36     HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手
37 册.HTML+CSS完全自学手册.HTML+CSS完全自学手册.HTML+CSS完全自学手册
38 </div>
39 </body>
40 </html>

```

【代码解析】第 15 行代码用 scrollbar-base-color 表示滚动条基准颜色，第 23 行代码用 scrollbar-track-color 表示滚动条拖动区域颜色，效果如图 19-5 表示。

说明

为了在多种平台中能有一致的效果，建议读者不要使用滚动条的效果，而且设置滚动条的效果后不仅仅不能在 FF 中浏览，颜色设置不当反而不好看，并且更容易在滚动条中出现问题。

19.2 层的漂移

设置层的漂移时，层与层之间在默认情况下不在同一水平线上显示，只有垂直显示。在网页设计中，需要多个层在同一水平线上时，按默认的值是不能达到多层在同一位置上的。下面介绍如何设置层的漂移，使单个或多个层在同一水平线上显示。

19.2.1 设置层的漂移

要设置多个层处于同一水平线上平行，需要对层设置成为向左或者向右漂移，当层确定宽度后，多个层总宽度没有超过父对象的宽度时，会出现层漂移。设置层漂移用 float 表示，其后接 left 或 right，默认值是 none。下面是其完整语法。

```
float:none|left|right
```

属性 float 表示对象的漂移方向，有三个参数，默认的是 none，表示不漂移，其他两个参数各有一个方位，其介绍如下所述。

- none: 默认值。对象不漂移。
- left: 文本流向对象的右边。
- right: 文本流向对象的左边。

需要注意的是，在使用 float 属性时要确定对象的宽度，默认情况下层对象的宽度为 100%，即使使用了 float 也不会漂移，因为它没有多余的空间给下一个对象漂移，而设置对象的宽度后，剩下的宽度可以让给其他对象漂移到此位置，前提是有足够的空间。



图 19-5 设置滚动条基准颜色和拖动区域颜色

漂移的方向可以是左边也可以是右边，如果设置了向右漂移，在编写 HTML 代码中，写列表向右漂移的对象，总之，向右漂移的 HTML 对象在向左漂移的对象前面。代码 19-6 演示了多个层往不同方向的漂移。

代码 19-6 float 的用法

```
-----文件名: float的用法.html-----
01 <html>
02 <head>
03 <title>float的用法</title>
04 <style type="text/css">
05 #float1
06 {
07     width:200px;
08     height:80px;
09     border:1px solid red;
10     /*设置向右漂移*/
11     float:right;
12 }
13 #float2
14 {
15     width:200px;
16     height:80px;
17     border:1px solid red;
18     /*设置向右漂移*/
19     float:right;
20 }
21 #float3
22 {
23     width:200px;
24     height:80px;
25     border:1px solid red;
26     /*设置不漂移*/
27     float:none;
28 }
29 #float4
30 {
31     width:200px;
32     height:80px;
33     border:1px solid red;
34     /*设置向左漂移*/
35     float:left;
36 }
37 </style>
38 </head>
39 <body>
40 <div id="float1">
41     向右漂移float1。
42 </div>
43 <div id="float2">
44     向右漂移float2。
45 </div>
46 <div id="float3">
```

```

47     没有漂移float3。
48 </div>
49 <div id="float4">
50     向左漂移float4。
51 </div>
52 </body>
53 </html>

```

【代码解析】在代码 19-6 中，网页中使用了四个层对象，用来表示层对象的漂移。第 05~12 行代码为第一层，设置的宽度为 200 像素，且向右漂移；第 13~20 行代码为第二层，设置的漂移方向也为向右；同时，在第 21~28 行代码的第三层中，设置漂移为默认状态，即为 none；而在第 29~36 行代码的第四层中，设置的漂移方向为向左，在 HTML 代码中，设置为向右的对象写在向左对象的前面。效果如图 19-6 所示。



图 19-6 float 的用法

技巧

在HTML中向右漂移对象必须在向左漂移对象的前面，否则，达不到要设置漂移的目的，如果出现这种问题，只需要将向右漂移的对象写在向左漂移对象的前面即可。

19.2.2 清除层的漂移范围

若使用 float 不当会出现网页不对齐，过多向左或向右漂移，这点也是让很多使用 TABLE 布局的设计者头疼的地方。针对这种问题，需要了解 clear 的用法，用来清除不同方位的漂移，clear 用法是相对 float 而言的。

```
clear : none | left | right | both
```

从以上代码可知，语法 clear 表示控制 float 漂移，可以设置 float 的漂移方向。如下是各属性值的用法。

- none: 默认值。允许两边都可以有浮动对象。
- left: 不允许左边有浮动对象。
- right: 不允许右边有浮动对象。
- both: 不允许有浮动对象。

因为 float 比较难控制，所以很多初学者不喜欢使用 CSS，其实，如果对 CSS 关键的问题搞清

楚了,那么 CSS 就容易学了,而且也会带来很大的方便,例如 clear 的用法,能使 float 漂移时,设置不需要 float 漂移的位置。代码 19-7 表示了用 clear 来改变属性 float 的漂移效果。

代码 19-7 clear 的用法

```
-----文件名: clear的用法.html-----
01 <html>
02 <head>
03 <title>clear的用法</title>
04 <style type="text/css">
05 #box
06 {
07     /*设置宽度为100%*/
08     width:100%;
09 }
10 #clear1
11 {
12     width:200px;
13     height:80px;
14     border:1px solid red;
15     /*设置向左漂移*/
16     float:left;
17 }
18 #clear2
19 {
20     width:200px;
21     height:80px;
22     margin:3px;
23     /*设置向左漂移*/
24     float:left;
25     border:1px solid red;
26 }
27 </style>
28 </head>
29 <body>
30 <div id="box">
31     <div id="clear1">
32         HTML+CSS完全自学手册1111
33     </div>
34     <div id="clear2">
35         HTML+CSS完全自学手册2222
36     </div>
37 </div>
38 </body>
39 </html>
```

【代码解析】在代码 19-7 中,第 16、24 行代码使用 float 对两个层设置在容器层中向左漂移,且确定两层之间的宽度都为 200 像素,漂移效果如图 19-7 所示。

若 float 设置为 left,图 19-7 中的两层都向左漂移,现在用 clear 属性对第二层进行控制,使第二层在整体效果中发生变化。对第二层设置的 CSS 代码只要加上一行代码,设置 clear 为 both。其第二层完整样式代码如下。

```

/*第二层的样式文件*/
01 #clear2
02 {
03     width:200px;
04     height:80px;
05     margin:3px;
06     /*设置向左漂移*/
07     float:left;
08     border:1px solid red;
09     /*设置对象不能漂移*/
10     clear:both;
11 }

```

【代码解析】只对第二层设置 clear 为 both（代码第 10 行），其 float 的漂移就按 clear 设置的显示，本处设置的是 both，即不向两边漂移，效果如图 19-8 所示。



图 19-7 没有设置 clear 时



图 19-8 设置 clear 为 both 后

19.3 设置层的可视区域

属性 clip 用来设置对象的可视区域，而区域外的部分是透明的。属性 clip 其后可接两个参数，默认值为 auto，表示对象无剪切，而参数 rect 表示坐标的位置，有四个数值。其语法如下所示。

```
clip : auto | rect ( number number number number )
```

这个属性很有意思，特别是在网页中做特效时所表达出的功能不同。下面是对这两个参数的解释。

- auto: 默认值，对象无剪切。
- rect: 依据上、右、下、左的顺序提供自对象左上角为 (0, 0) 坐标计算的四个偏移数值，其中任一数值都可用 auto 替换，即此边不剪切。

将两个文字相同而色彩不同的文字重合在一起，分别给其加 clip 属性，使上面和下面的文字被剪切位置不同，从而产生两种不同的色彩。代码 19-8 演示了如何将两行文字剪切成一行特效文本。

代码 19-8 clip 用法

```

-----文件名: clip用法.html-----
01 <html>
02 <head>
03 <title>clip用法</title>

```

```
04 <style type="text/css">
05 .textBottom {
06     color: #333333;
07     /*设置绝对定位*/
08     position: absolute;
09     /*左端距离*/
10     left: 3em;
11     /*顶部距离*/
12     top: 1em;
13     /*字体属性*/
14     font: 26px Century Gothic,Arial, Helvetica, sans-serif;
15     clip: rect(18px auto auto auto);
16 }
17 .textTop {
18     color: #CC0000;
19     /*绝对定位*/
20     position: absolute;
21     left: 3em;
22     top: 1em;
23     font: 26px Century Gothic,Arial, Helvetica, sans-serif;
24     clip: rect(0 auto 18px 0);
25 }
26 .container {
27     width: 28em;
28     height: 5em;
29     margin: 1em auto;
30     /*相对定位*/
31     position: relative;
32     background: #F6F6F6;
33 }
34 </style>
35 </head>
36 <body>
37 <div class="container">
38     <a href="#" class="textTop">HTML+CSS完全自学手册</a>
39     <a href="#" class="textBottom">HTML+CSS完全自学手册</a>
40 </div>
41 </body>
42 </html>
```

【代码解析】在代码 19-8 中，用定位 position 和 clip 表示出了文本特效，在 HTML 代码中出现两行相同的文本，在网页中却显示出一行文本，而且字体颜色变成上半部分与下半部分不一样。其原理是用 clip 把部分剪切掉了，第 15 行代码截取了第一行文字的下半部分，第 24 行代码截取了第一行文字的上半部分，用定位把两行文本放在同一位置上。效果如图 19-9 所示。



图 19-9 用 clip 设置文本特效

19.4 内容超过层大小时

本节介绍布局时常用的属性之一——`overflow`，它表示内容超过其指定高度及宽度时如何管理内容。在实际项目中，有的文本超过对象的宽度，或超过对象的高度，这时设置对象的 `overflow` 属性来管理是否显示滚动条。下面是 `overflow` 的完整语法。

```
overflow : visible | auto | hidden | scroll
```

属性 `overflow` 接有 4 个参数，分别为 `visible`、`auto`、`hidden` 和 `scroll`。下面是相应的解释。

- ❑ `visible`：默认值。不剪切内容也不添加滚动条。假如显式声明此默认值，对象将以包含对象的 `window` 或 `frame` 的尺寸裁切，并且 `clip` 属性设置将失效。
- ❑ `auto`：在必需时对象内容才会被裁切或显示滚动条。
- ❑ `hidden`：不显示超过对象尺寸的内容。
- ❑ `scroll`：总是显示滚动条。

代码 19-9 overflow 用法

```
-----文件名：overflow用法.html-----
01 <html xmlns="http://www.w3.org/1999/xhtml">
02 <head>
03 <title>overflow用法</title>
04 <style type="text/css">
05 #overflow1
06 {
07     width:180px;
08     height:80px;
09     /*设置为可视*/
10     overflow:visible ;
11 }
12 #overflow2
13 {
14     width:180px;
15     height:80px;
16     overflow:auto;
17 }
18 #overflow3
19 {
20     width:180px;
21     height:80px;
22     /*设置为隐藏*/
23     overflow:hidden;
24 }
25 #overflow4
26 {
27     width:180px;
28     height:80px;
29     /*设置为滚动*/
30     overflow:scroll;
31 }
32 </style>
33 </head>
```

```

34 <body>
35 下面图片是第一层: overflow设置为visible<br>
36 <div id="overflow1">
37     
38 </div>
39 下面图片是第二层: overflow设置为auto<br>
40 <div id="overflow2">
41     
42 </div>
43 下面图片是第三层: overflow设置为hidden<br>
44 <div id="overflow3">
45     
46 </div>
47 下面图片是第四层: overflow设置为scroll<br>
48 <div id="overflow4">
49     
50 </div>
51 </body>
52 </html>

```

【代码解析】在代码 19-9 中,对 4 个层设置不同的参数:首先第 10 行代码对第一层设置为默认值,即为 visible,表示不添加滚动条;第 16 行代码为第二层设置的 overflow 为 auto,在必需时显示了滚动条;第 23 行代码为第三层设置的为 hidden,表示隐藏,当内容超过对象的范围时,多余的部分会隐藏掉;而在第 30 行代码的第四层中,设置的是 scroll,表示总是显示滚动条。效果如图 19-10 所示。

19.5 鼠标指针的不同表示

Cursor 属性在网页中应用也非常广泛,表示在对象上移动的鼠标指针采用的光标形状。在对象中,设置不同的参数所显示的光标状态不同。

下面是其完整语法形式。

```

cursor : auto | all-scroll | col-resize | crosshair | default | hand | move | help | no-drop
| not-allowed | pointer | progress | row-resize | text | vertical-text | wait | *-resize
| url ( url )

```

本属性的参数非常多,为了表示各参数的用法,下面分别说明各参数的用法。

- ❑ auto: 默认值。浏览器根据当前情况自动确定鼠标光标类型。
- ❑ all-scroll: IE6.0 有上、下、左、右四个箭头,中间有一个圆点的光标,用于标示页面可以向上、下、左、右任何方向滚动。
- ❑ col-resize: IE6.0 有左、右两个箭头,中间是由竖线分隔开的光标,用于标示项目或标题栏可以被水平改变尺寸。



图 19-10 overflow 的用法

- ❑ `crosshair`: 简单的十字线光标。
- ❑ `default`: 客户端平台的默认光标, 通常是一个箭头。
- ❑ `hand`: 竖起一只手指的手形光标, 就像通常将光标移到超链接上时那样。
- ❑ `move`: 十字箭头光标, 用于标示对象可被移动。
- ❑ `help`: 带有问号标记的箭头, 用于标示有帮助信息存在。
- ❑ `no-drop`: IE6.0 带有一个被斜线贯穿的圆圈的手形光标, 用于标示被拖起的对象不允许在光标的当前位置被放下。
- ❑ `not-allowed`: IE6.0 禁止标记 (一个被斜线贯穿的圆圈) 光标, 用于标示请求的操作不允许被执行。
- ❑ `pointer`: IE6.0 和 `hand` 一样, 竖起一只手指的手形光标, 就像通常用户将光标移到超链接上时那样。
- ❑ `progress`: IE6.0 带有沙漏标记的箭头光标, 用于标示一个进程正在后台运行。
- ❑ `row-resize`: IE6.0 有上下两个箭头, 中间由横线分隔开的光标, 用于标示项目或标题栏可以被垂直改变尺寸。
- ❑ `text`: 用于标示可编辑的水平文本的光标, 通常是大写字母 I 的形状。
- ❑ `vertical-text`: IE6.0 用于标示可编辑的垂直文本的光标, 通常是大写字母 I 旋转 90° 的形状。
- ❑ `wait`: 用于标示程序忙, 需要等待的光标, 通常是沙漏或手表的形状。
- ❑ `url`: IE6.0 用户自定义光标。使用绝对或相对 url 地址指定光标文件 (扩展名为 `.cur` 或者 `.ani`)。

下面用实例来演示光标的用法。每个段落表示一种光标用法, 当光标上移在对象上面时, 光标就变化了, 如代码 19-10 所示。

代码 19-10 cursor 用法

```
-----文件名: cursor用法.html-----
01 <html>
02 <head>
03 <title>cursor用法</title>
04 <style type="text/css">
05 div
06 {
07     width:150px;
08     height:20px;
09     margin:5px;
10     border:1px solid red;
11 }
12 #cursor0
13 {
14     /*设置指针*/
15     cursor:auto;
16 }
17 #cursor1
18 {
19     cursor:all-scroll;
20 }
21 #cursor4
22 {
```

```
23     cursor:default;
24 }
25 #cursor5
26 {
27     cursor:hand;
28 }
29 #cursor6
30 {
31     cursor:move;
32 }
33 #cursor7
34 {
35     cursor:help;
36 }
37 #cursor9
38 {
39     cursor:pointer;
40 }
41 #cursor10
42 {
43     cursor:text;
44 }
45 #cursor11
46 {
47     cursor:wait;
48 }
49 </style>
50 </head>
51 <body>
52 <div id="cursor0">
53     cursor设置为: auto, 是默认值
54 </div>
55 <div id="cursor3">
56     cursor设置为: crosshair
57 </div>
58 <div id="cursor4">
59     cursor设置为: default
60 </div>
61 <div id="cursor5">
62     cursor设置为: hand
63 </div>
64 <div id="cursor6">
65     cursor设置为: move
66 </div>
67 <div id="cursor7">
68     cursor设置为: help
69 </div>
70 <div id="cursor9">
71     cursor设置为: pointer
72 </div>
73 <div id="cursor10">
74     cursor设置为: text
75 </div>
```

```

76 <div id="cursor11">
77     cursor设置为: wait
78 </div>
79 </body>
80 </html>

```

【代码解析】第 12~16 行代码为第 1 个段落，表示了 cursor 属性的 auto 参数，是默认值；第 17~20 行代码为第 2 个段落，用到参数 crosshair，表示简单的十字线光标；第 21~24 行代码为第 3 个段落，用 default 表示，通常是一个箭头；第 25~28 行代码为第 4 个段落，表示 hand 参数，显示的是手形，超链接就是这种手形；第 29~32 行代码为第 5 个段落，用到参数 move，十字箭头光标，用于标记对象可被移动的效果；第 67~69 行代码为第 6 个段落，用到 help，表示带有问号的箭头；第 70~72 行代码为第 7 个段落，用 pointer，效果与 hand 一样，也是手形；第 73~75 行代码为第 8 个段落，用到 text，标记可编辑的水平文本光标；第 76~78 行代码为最后一个段落，用到 wait，表示需要等待的状态，这些光标在网页中显示，整体效果如图 19-11 所示。



图 19-11 cursor 的用法

19.6 设置对象可见性

在布局中，有两种方法设置对象是否可见：第一种，用属性 display 设置对象是否隐藏；第二种，用 visibility 属性设置对象是否显示。两者的用法不同，所表现的效果也有区别，下面分别介绍。

19.6.1 设置对象是否隐藏

在设置某些层时，可能需要对层进行隐藏；当链接需要放大显示时，可能需要设置块元素来表示该区域。要实现这些功能，用 display 属性，不管是隐藏还是块元素，都能得以实现。下面是其完整语法。

```
display : block | none | inline | inline-block list-item
```

各参数含义如下。

- ❑ block：块对象的默认值。将对象强制作为块对象呈递，为对象之后添加新行。
- ❑ none：隐藏对象。与 visibility 属性的 hidden 值不同，其不为被隐藏的对象保留物理空间。
- ❑ inline：内联对象的默认值。将对象强制作为内联对象呈递，从对象中删除行。
- ❑ inline-block：将对象呈递为内联对象，但是对象的内容作为块对象呈递。旁边的内联对象会被呈递在同一行内。
- ❑ list-item：将块对象指定为列表项目，并可以添加可选项目标志。

下面用实例来演示上面参数的用法，如代码 19-11 所示。

代码 19-11 display 用法

```
-----文件名: display用法.html-----
01 <html>
02 <head>
03 <title>display用法</title>
04 <style type="text/css">
05 .box
06 {
07     border:1px solid red;
08     margin:5px;
09 }
10 #display1
11 {
12     /*块显示*/
13     display:block;
14 }
15 #display2
16 {
17     display:none;
18 }
19 #display3
20 {
21     display:inline;
22 }
23 #display4
24 {
25     display:inline-block;
26 }
27 #display5
28 {
29     display:list-item;
30 }
31 </style>
32 </head>
33 <body>
34 <div class="box">
35     这是一本<span id="display1">HTML+CSS完全自学手册</span>参考书
36 </div>
37 <div class="box">
38     这是一本<span id="display2">HTML+CSS完全自学手册</span>参考书
39 </div>
40 <div class="box">
41     这是一本<div id="display3">HTML+CSS完全自学手册</div>参考书
42 </div>
43 <div class="box">
44     这是一本<div id="display4">HTML+CSS完全自学手册</div>参考书
45 </div>
46 <div class="box">
47     这是一本<div id="display5">HTML+CSS完全自学手册</div>参考书
48 </div>
49 </body>
50 </html>
```

【代码解析】第 10~14 行代码为第 1 层，表示了块元素的用法；第 15~18 行代码为第 2 层，设置 display 值为 none，表示隐藏对象；第 19~22 行代码为第 3 层，表示内联对象的默认值；第 23~26 行代码为第 4 层，表示将对象呈递为内联对象；第 27~30 行代码为第 5 层，表示块对象指定为列表项目。效果如图 19-12 所示。

19.6.2 设置对象是否显示

在布局时，有时需要有的对象不显示，但需要占用对象的空间，当达到一定条件时，该对象才显示出来。与之相关的还有其他需求：设置对象是否显示或刚开始不显示，等条件达到后才显示出来。设置对象是否显示用 visibility 表示，其后接参数来表示对象是否显示。

属性 visibility 用来设置是否显示对象，其后接的值效果与 display 有部分相似。其完整语法形式如下。

```
visibility : inherit | visible | hidden
```

各参数的含义如下。

- inherit: 默认值。继承父对象的可见性。
- visible: 表示对象可视。
- hidden: 表示对象隐藏。

本属性在应用中是比较常用的，如设置层对象显示。而本参数的隐藏与 display 的隐藏效果稍有区别，一个表示物理上隐藏，一个表示完全的隐藏。从页面的效果中可以看到，visibility 表示的对象在网页中是存在一个空间的，不管它是否显示；而 display 表示的对象是在网页中显示就有空间存在，而隐藏时，网页中的空间也随之消失。这是两者的关键区别。

属性 visibility 与 display 的参数值表示也不同，visibility 的值用 visible 来表示对象可见，而用 hidden 来表示对象不可见，即隐藏，但空间还存在于对象上。代码 19-12 表示 visibility 的用法。

代码 19-12 visibility 用法

```
-----文件名: visibility用法.html-----
```

```
01 <html>
02 <head>
03 <title>visibility用法</title>
04 <style type="text/css">
05 #vis1
06 {
07     /*设置visibility的默认值*/
08     visibility:inherit;
09     border:1px solid red;
10 }
11 #vis2
12 {
13     /*设置对象可见*/
14     visibility:visible;
15     border:1px solid red;
```

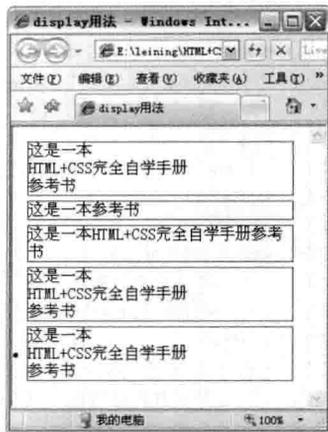


图 19-12 display 用法

```
16 }
17 #vis3
18 {
19     /*设置对象不可见*/
20     visibility:hidden;
21     border:1px solid red;
22 }
23 </style>
24 </head>
25 <body>
26 <div id="vis1">
27     可以看到这张图片吗? visibility设置为inherit
28     
29 </div>
30 <div id="vis2">
31     可以看到这张图片吗? visibility设置为visible
32     
33 </div>
34 <div id="vis3">
35     可以看到这张图片吗? visibility设置为hidden
36     
37 </div>
38 </body>
39 </html>
```

【代码解析】在代码 19-7 中，对 visibility 设置不同的参数，第 6~10 行代码为第 1 层设置为 inherit，而在第 11~16 行代码的第 2 层中，设置 visibility 的值为 visible，表示显示对象，相反，在第 17~22 行代码的第 3 层中，设置对象的 visibility 为 hidden，在网页上占用热物理空间，但不显示出来，从右边的滚动条是可以看出，空间上还是存在这个层对象的，效果如图 19-13 所示。



图 19-13 visibility 用法

注意 理解 visibility 与 display 的区别,除了效果上表现出来的不同外,两属性的参数值也不一样。

19.7 小结

本章首先介绍了滚动条的相关属性,这些属性都以滚动条的颜色来设置,而且只有 IE 支持;然后介绍了用 CSS 布局的常用属性,这些属性在网页制作中非常重要。本章中比较重要的是清除属性 clear、浮动属性 float 和显示属性 display,有了这些属性才可以在网页中对对象有所控制,网页才可以达到所看到的那样漂亮的效果。

这些布局都是平时布局中经常用到的,也是不可缺少的属性,有了这些属性可以用 CSS 很好地控制层对象。

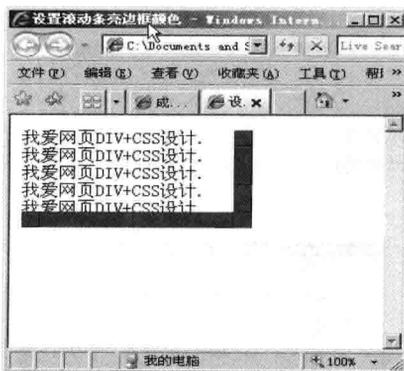
19.8 习题

一、填空题

1. 属性_____表示滚动条方向箭头的颜色,其接的值为颜色值。
2. 设置层漂移用_____表示,其后接三个参数,分别是:_____、_____和_____。
3. 在布局中,有两种方法设置对象是否可见。第一,_____;第二,_____。

二、上机实践

1. 按照下图编辑一个网页。



【提示】

scrollbar-3dlight-color	-----	滚动条亮边框颜色属性
scrollbar-highlight-color	-----	3D界面亮边颜色属性
scrollbar-arrow-color	-----	滚动条方向箭头的颜色属性
scrollbar-shadow-color	-----	滚动条3D界面暗边颜色属性
scrollbar-base-color	-----	滚动条基准颜色属性

2. 按照下图编辑一个网页。



【提示】

scrollbar-face-color-----滚动条表面颜色属性
 scrollbar-darkshadow-color-----滚动条暗边框颜色属性
 scrollbar-track-color:red-----滚动条拖动区域颜色属性

3. 按下图编辑一个网页，同时为每个框各定一个鼠标样式。



【代码解析】

float-----层漂移属性
 cursor-----鼠标样式属性

4. 按下图编辑一个网页。



【代码解析】

clip-----可视区域属性

第 20 章 选 择 符

在 CSS 中，有了选择符，可以与 HTML 的标记联系，这样网页中的标记就被样式控制着，显现出来的就是完美效果。选择符分成很多种，如通配选择符、类型选择符、ID 选择符等，这些选择符在本章中都要讲解。而伪类是选择符的螺栓，用来指定链接或者与其相关的选择符的状态。它是一种特殊的类选择符，是能被支持 CSS 的浏览器自动所识别的特殊选择符。伪类最常用的用法是表示对链接对象的不同状态下定义不同的样式效果。

在以前的章节中，常常在代码中使用单位，如表示层对象的宽度和高度、背景和边框的颜色等，这样不同的单位值表达了属性的不同效果。下面就来介绍单位，配合属性值使用。

本章主要涉及的内容如下：

- 通配选择符；
- 类型选择符；
- 包含选择符；
- ID 选择符；
- 类选择符；
- 选择符分组；
- “:link”、“:hover”、“:active”和“:visited”；
- 长度单位；
- 颜色单位。

20.1 通配选择符

通配选择符指选定文档对象模型 (DOM) 中所有类型的单个对象。正如通配选择字面意思一样，其语法形式用 (*) 表示，假如通配对象选择符不是单一选择符中的唯一组成，则通配选择符 (*) 可以省略。

```
/*CSS代码：表示选择符规则*/  
* | CSS代码规则 |
```

如用星号表示，可以代表所有类型，包含 body、div、td 等。代码 20-1 演示通配选择符的用法。

代码 20-1 通配选择符

```
-----文件名：通配选择符.html-----  
01 <html>  
02 <head>  
03 <title>通配选择符</title>  
04 <style type="text/css">
```

```

05  /*星号代表通配选择符*/
06  * {
07      /*对象背景颜色*/
08      background:#666666;
09      /*对象边框为1像素红色实线*/
10      border:1px solid red;
11  }
12  </style>
13  </head>
14  <body>
15  <!--以下在一个层里面包含了段落-->
16  <div>
17      <p>这里是在一个层的段落里面</p>
18  </div>
19  </body>
20  </html>

```

【代码解析】在第 14~19 行代码的<body></body>中存在两个对象，即层和段落，而样式中用通配选择符设置对象为 666666 的背景颜色和 1 像素红色实线边框，其对层对象也有效，同时对段落也是起作用的，效果如图 20-1 所示。

技巧 通配选择符常常在网页布局设置背景颜色和补丁时所用，其中可以设置内外边距为 0，即 margin:0 和 padding:0 代表对整个文档起作用。



图 20-1 通配选择符

20.2 类型选择符

类型选择符是指以文档对象模型 (DOM) 作为选择符，即选择某个 HTML 标记为对象，设置其样式规则。以下代码表示类型选择符。

```

01  p{
02      width:200px;
03      height:100px;
04      border:1px solid red;
05  }
06  div
07  {
08      margin-top:10px;
09  }

```

【代码解析】第 1~5 行代码和第 6~9 行代码分别是段落和层的类型选择符的定义，类型选择符的对象名是 DOM 类型，样式代码与其他的一样，都是用来控制对象的规则。以上代码就是对段和层做了样式修饰，其完整的语法形式如下。

```

E
{
    /*CSS规则*/
}

```

其中 E 就是代码 DOM 类型，如 td、p 等都是 DOM 类型。为了说明类型选择符的作用，下面用实例演示，代码 20-2 是对不同对象设置类型选择符。

代码 20-2 类型选择符

```
-----文件名：类型选择符.html-----
01 <html>
02 <head>
03 <title>类型选择符</title>
04 <style type="text/css">
05 /*对td元素设置大小*/
06 td
07 {
08     /*宽为200像素*/
09     width:200px;
10     /*高为100像素*/
11     height:50px;
12 }
13 /*对段落标记设置边框*/
14 p
15 {
16     /*边框为1像素红色实线*/
17     border:1px solid red;
18 }
19 </style>
20 </head>
21 <body>
22 <table border="1">
23     <tr>
24         <td>类型选择符</td><td><p>这里一个段落</p></td>
25     </tr>
26     <tr>
27         <td><p>这里是另一个段落</p></td><td>CSS</td>
28     </tr>
29 </table>
30 </body>
31 </html>
```

【代码解析】第 22~29 行通过运用表格，在数据单元里放入段落，然后在第 6~18 行代码用类型选择符对 td 和 p 标记设置样式规则，对文档所有的 td 和 p 都按样式规则进行，不管文档中有多少个标记，效果如图 20-2 所示。

技巧

还可以用类型选择符控制链接颜色等样式，下一章会对此进行讲解。

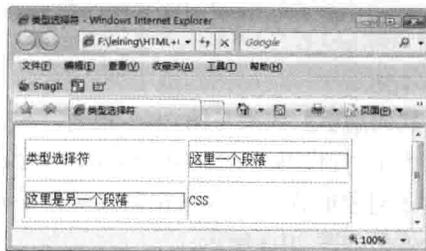


图 20-2 类型选择符的用法

20.3 包含选择符

包含选择符在样式中会常常用到，因布局中常常用到容器层和其包含的子层，如果用到包含选

选择符就可以对某个容器层的子层控制,使其他同名的对象不受该规则影响。这样在语法上表示与前两个选择符不一样,包含选择符对象要依次选择出对象,从大到小,即从容器层到子层。其完整的语法如下。

```
E1 E2
|
|
| /*对子层控制规则*/
|
|
```

其中 E1 指父层对象, E2 指小点的层或者子层对象,即 E1 对象包含 E2 对象。下面用实例来演示包含选择符的用法,具体代码如代码 20-3 所示。

代码 20-3 包含选择符

```
-----文件名: 包含选择符.html-----
01 <html>
02 <head>
03 <title>包含选择符</title>
04 <style type="text/css">
05 table td
06 {
07 width:200px;
08 height:50px;
09 }
10 td p
11 {
12 border:1px solid red;
13 }
14 </style>
15 </head>
16 <body>
17 <table border="1">
18 <tr>
19 <td>类型选择符</td><td><p>这里一个段落</p></td>
20 </tr>
21 <tr>
22 <td><p>这里是另一个段落</p></td><td>CSS</td>
23 </tr>
24 </table>
25 </body>
26 </html>
```

【代码解析】效果与图 20-2 相同,在表格中,第 5~13 行代码在 table td、td p 的包含选择符上,子对象的样式只有在相关父对象里才会生效,效果如图 20-3 所示。

20.4 布局常用选择符

本书主要讲解 DIV+CSS 布局,而 DIV 是 HTML 中的标记,CSS 是控制 DIV 等其他标记

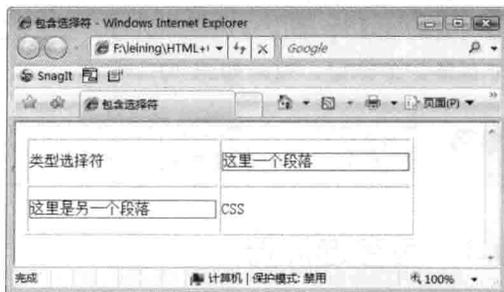


图 20-3 包含选择符

的语句，要使两者相关联，就得使用一定的协议，使两者之间能够“通信”。从前面学习中可以知道，起“通信”作用的是这些选择符。

在 HTML 中的 DIV 标记中，有参数 ID 和 CLASS 用来设置关联 CSS 样式，其中 ID 所设置的标识符在 CSS 中与 ID 选择符相关联，而 CLASS 设置的标识符与 CLASS 选择符相关联，这样 CSS 就可以控制 DIV 中的效果。在布局中，ID 选择符和 CLASS 选择符是最常用的，下面分别介绍。

20.4.1 ID 选择符

ID 选择符是指在 DOM 中用为唯一标识符的 ID 为选择符。ID 标识符以“#”打头，其后接 ID，注意 ID 标识符不能在样式中出现两次或两次以上，如果出现，系统会以第一次出现的为准，其他的被忽视掉。其完整的语法如下。

```
#ID标识符
{
    /*样式规则*/
}
```

以上代码表示 ID 选择符的语法，ID 选择符与 HTML 标记中的 DIV 设置的 ID 标识符一样的，即表示该 ID 的样式，代码 20-4 演示了 HTML 中的 ID 与 CSS 代码中的 ID 选择符的用法。

代码 20-4 ID 选择符

```
-----文件名: ID选择符.html-----
01 <html>
02 <head>
03 <title>ID选择符</title>
04 <style type="text/css">
05 #id1
06 {
07     width:100px;
08     height:50px;
09     border:1px solid red;
10 }
11 #id2
12 {
13     width:200px;
14     height:60px;
15     border:1px solid blue;
16 }
17 </style>
18 </head>
19 <body>
20 <div id="id1">
21     ID1的层
22 </div>
23 <div id="id2">
24     ID2的层
25 </div>
26 </body>
27 </html>
```

【代码解析】第 20、23 行代码用两个 DIV 对象设置两个 ID 值，在 CSS 中用 ID 选择符与之相关联。在第 20 行代码 ID 为 ID1 的 ID 选择符控制 HTML 中的 ID1 层，而在第 23 行代码 ID 为 ID2 的 ID 选择符控制 HTML 中的 ID2 层，用 CSS 设置不同对象的不同边框颜色。ID 选择符在网页中的效果如图 20-4 所示。

说明

正常时为 div#标识符，表示 ID 选择符的标识，其后接的标识符是 HTML 中的对象名称。

20.4.2 类选择符

除了 ID 选择符外，类选择符也是与布局联系紧密的选择符。类选择符也可以叫做 CLASS 选择符。类选择符与 ID 选择符相似，但是类选择符可以表示一个类，即可以在文档中使用多次。

类选择符与 HTML 中的标记 CLASS 所设置的标识符关联，可以用 CSS 来控制这些标记，如 DIV 中的 CLASS 值、TABLE 中的 CLASS 值、P 标记中的 CLASS 值等。下面用实例演示类选择符的用法，如代码 20-5 所示。

代码 20-5 类选择符

```
-----文件名：类选择符.html-----
01 <html>
02 <head>
03 <title>类选择符</title>
04 <style type="text/css">
05 /*类选择符一*/
06 .class1
07 {
08     /*设置宽度、高度和边框*/
09     width:100px;
10     height:50px;
11     border:1px solid red;
12 }
13 /*类选择符二*/
14 .class2
15 {
16     /*设置宽度、高度和边框*/
17     width:200px;
18     height:60px;
19     border:1px solid blue;
20 }
21 </style>
22 </head>
23 <body>
24 <!--HTML代码中设置好类选择符的标识符-->
25 <div class="class1">
26     class1的层
27 </div>
28 <div class="class2">
```



图 20-4 ID 选择符的用法

```

29     class2的层
30 </div>
31 </body>
32 </html>

```

【代码解析】 在第 6~20 行代码设置了两个层，分别用类选择符表示为 class1 和 class2，其中 class1 在 CSS 中设置的宽度为 100 像素，高度为 50 像素，边框颜色为红色，而 class2 中设置的宽度和高度分别为 200 像素和 60 像素，边框颜色为蓝色，即宽度比 class1 宽了一倍，如图 20-5 所示。

类选择符与类型选择符不同，类型是指 HTML 中的标记。注意 ID 选择符与类选择符的区别。



图 20-5 类选择符

20.5 分组选择符

分组选择符指的是对多个标记设置同样的样式，在不同的类型中，表示同样的样式。本书常用的样式如下。

```

html, body
{
    /*CSS代码*/
}

```

上述代码表示在 html 标记和 body 标记中，都有一样的样式，主要是兼容 IE 和 FF 浏览器。分组选择符的语法如下。

```

E1, E2, E3
{
    /*CSS代码*/
}

```

将同样的定义应用于多个选择符，可以将选择符以逗号分隔的方式并为组。语法中，E1、E2、E3 表示不同的标记类型，代码表示 CSS 样式来控制这些标记类型。再如以下代码所示。

```

td, div, a, body
{
    font-size: 14px;
}

```

上面表示对 td、div、a 和 body 标记的样式设置成字体大小为 14 像素。对选择符分组的实例如代码 20-6 所示。

代码 20-6 选择符分组

```

-----文件名: 选择符分组.html-----
01 <html>
02 <head>
03 <title>选择符分组</title>
04 <style type="text/css">

```

```

05  html,body
06  {
07      margin:0px;
08      padding:0px;
09      background:#cccccc;
10  }
11  div,p
12  {
13      width:200px;
14      height:100px;
15      margin-top:5px;
16      border:1px outset red;
17  }
18  </style>
19  </head>
20  <body>
21  <div>
22      这里一个层。
23  </div>
24  <div>
25      这也是一个层。
26  </div>
27  <p>而这里是一个段落，与div的样式一样的。</p>
28  </body>
29  </html>

```

【代码解析】第 6~10 行代码表示了 html 和 body 有一样的样式，同时第 11~17 行代码对 div 与 p 标记也设置了一样的样式，边框都设置成红色，宽度和高度也相同，同时外边距顶端为 5 像素，效果如图 20-6 所示。

选择符 div、p 与 div p 表示的内容是不同的，
技巧 前者是表示 div 和 p 两种标记，而 div p 表示的是在 div 标记下的 p 标记。

20.6 伪类

伪类可以看做是一种特殊的类选择符，是能被支持 CSS 的浏览器所自动识别的特殊选择符。伪类最大的用处是对链接在不同状态下定义不同的样式。伪类的语法是在原有的语法里加上一个伪类，在选择符中用冒号接上伪类的类型，然后在大括号中输入 CSS 样式来修饰页面。伪类的作用很大，最关键的是链接前后过程中的状态，其语法形式如下。

```

selector:pseudo-class
{
    property: value;
}

```

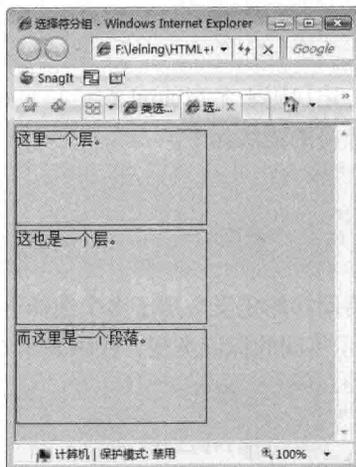


图 20-6 选择符分组

伪类的表示（选择符:伪类 {属性: 值}）中，选择符如链接的

伪类和类不同，伪类是 CSS 已经定义好的，不能像类选择符一样随意用别的名字，根据上面的语法可以解释为对象（选择符）在某个特殊状态下（伪类）的样式。类选择符及其他选择符也同样可以和伪类混用。

如给链接设置伪类，链接的标记是

代码 20-7 伪类的类型用法

-----文件名：伪类的类型用法.html-----

```

01 <html>
02 <head>
03 <title>伪类的类型用法</title>
04 <style type="text/css">
05  /*设置伪类的链接状态*/
06  a:link
07  {
08      /*链接颜色*/
09      color:red;
10  }
11  /*鼠标光标置于其上的状态*/
12  a:hover
13  {
14      /*鼠标悬停时*/
15      color:blue;
16  }
17  /*获得焦点的状态*/
18  a:active
19  {
20      /*被点击时颜色*/
21      color:orange;
22  }
23  /*访问过后的状态*/
24  a:visited
25  {
26      /*访问过后的颜色*/
27      color:yellow;
28  }
29 </style>
30 </head>
31 <body>
32     <a href="#">HTML+CSS完全自学手册</a>
33 </body>
34 </html>

```

【代码解析】第 32 行代码对链接（``表示链接地址为本页）的伪类类型设置成不同的颜色，其中第 6~10 行代码的 `a:link` 表示链接的颜色；第 12~16 行代码的 `a:hover` 表示鼠标指针放到链接处时的颜色；第 18~22 行代码的 `a:active` 表示单击时那时的颜色，当新的网页打开后或以前曾访问过该链接时，效果是 `a:visited` 设置的颜色（第 24~28 行代码），效果如图 20-7 所示。

注意

伪类的四种状态都是在链接前后所体现的效果，前提是对文本或图片先做好链接。

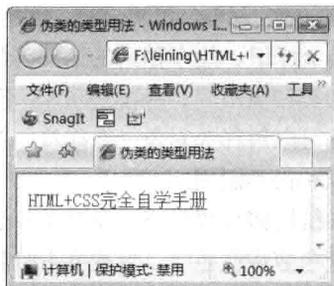


图 20-7 伪类的类型用法

20.7 伪类的四种状态

伪类的使用使网页效果更丰富，特别是在链接状态下，整个互联网因有了链接而不用记地址就可以直接到达目标地址，通过链接完成。

对链接的控制是伪类的主要功能，如刚设置文本的链接时，文本对象会呈蓝颜色，而单击链接时颜色会变化，等访问过后，颜色又会不同。除了默认的连接显示颜色外，还可以通过伪类来改变链接的状态，使不同的状态下颜色或其他样式不同。下面介绍伪类的四种状态。

20.7.1 访问链接的状态

在做好链接时，链接默认状态下都会变成蓝色，而且如果是文本链接则会出现下划线，是图片链接会出现边框，当然边框的默认颜色也为蓝色。为了更好地体现网页外观效果，有的网页设计师不喜欢默认的风格，那么怎么样来设置成用户喜欢的颜色呢？

要改变默认状态，才能使访问链接的状态有不一样的样式，不一样的效果。设置这些效果需用到伪类选择器。选择器 (selector) 是 CSS 中很重要的概念，所有 HTML 语言中的标记都是通过不同的 CSS 选择器进行控制的。只需要通过选择器，即可对不同的 HTML 标记进行控制，并赋予各种样式声明，即可实现各种效果。

而访问链接的状态是伪类选择器的一种，用 `:link` 表示，表示当文本或图片处于链接状态时的样式。`link` 的语法如下。

```
a:link {
    /*property: value;*/
}
```

说明

设置 `a` 对象在未被访问前的样式，默认值由浏览器决定。对于无 `href` 属性 (特性) 的 `a` 对象，此伪类不发生作用。代码 20-8 表示 `link` 的应用。

代码 20-8 link 应用

```
-----文件名: link应用.html-----
01 <html>
02 <head>
03 <title>:link 属性</title>
```

```

04 <style type="text/css">
05 /*设置访问链接状态*/
06 a:link
07 {
08     color:blue;
09     padding:10px;
10     border:1px solid red;
11 }
12 </style>
13 </head>
14 <body>
15 <a href="#">床前明月光</a><br>
16 疑是地上霜<br>
17 举头望明月<br>
18 低头思故乡
19 </div>
20 </body>
21 </html>

```

【代码解析】第 6~11 行代码演示了:link 属性的使用方法,其效果如图 20-8 所示。当“床前明月光”处于链接时,文本边框为红色,内边距为 10 像素,且字体颜色为蓝色。

20.7.2 鼠标光标置于其上的状态

与 link 语法一样,鼠标光标置于其上的状态也属于伪类的一种,其语法标记用 hover 表示,在链接标记符 a 后面接冒号,然后接上 hover。用大括号把 CSS 样式包围起,则表示鼠标光标放置到链接对象时出现的效果。其语法形式如下。

```

a:hover{
    /*property: value;*/
}

```

以上表示的是设置对象在其鼠标光标悬停时的样式。在 CSS1 中此伪类仅可用于 a 对象。对于无 href 属性(特性)的 a 对象,此伪类不发生作用。在 CSS2 中此伪类可以应用于任何对象。

当没有设置 hover 状态时,按默认状态鼠标指针放到链接对象中,指针立刻变成手形,而且颜色也不是蓝色,如果想修改默认的状态,用 hover 表示。代码 20-9 表示 hover 在网页链接中的用法。

代码 20-9 hover 的应用

```

-----文件名: hover的应用.html-----
01 <html>
02 <head>
03 <title>: hover属性</title>
04 <style>
05 a.demo:hover{
06     background:#FFC080;
07     color:#000000;
08 }
09 #idCodeDiv{

```

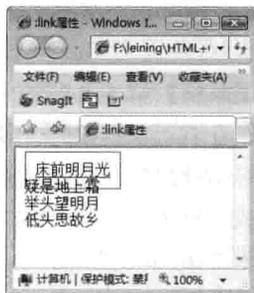


图 20-8 :link 属性

```

10     width:100%;
11     padding:4px;
12     font-family:verdana,tahoma;
13     margin:12px 0px 0px 0px;
14     background-color:#EEEEEE;
15     font-weight:bold;
16 }
17 </style>
18 </head>
19 <body bgcolor = lightyellow>
20 <center>
21     <a class=demo href="none.html" onclick="return false;">床前明月光;疑是地上霜。举头
22 望明月;低头思故乡。<br></a>
23 </center>
24 <br>
25 <br>
26     <div id=idCodeDiv> a :hover{background:#FFC080;color:#000000;}
27 </div>
28 </body>
29 </html>

```

【代码解析】第 5~8 行演示了: hover 属性的使用方法,在有对象链接时,设置 hover 的样式后,当鼠标指针放置在链接对象上时,对象的样式就会变成设置 hover 中的样式,而且指针为手形,其效果如图 20-9 所示。

20.7.3 获得焦点的状态

获得焦点的状态也是伪类的一种,表示鼠标单击链接时,获得焦点时的状态,其语法用 active 表示。其完整的语法如下。

```

a:active{
    /*property: value;*/
}

```

说明

设置对象在被激活(在鼠标单击与释放之间发生的事件)时的样式;在 CSS1 中此伪类仅可用于 a 对象,对于无 href 属性(特性)的 a 对象,此伪类不发生作用。在 CSS2 中此伪类可以应用于任何对象,并且: active 可以和: link 以及: visited 状态同时发生。

获得焦点的状态是链接对象,当链接被获得焦点时,这时的效果就会是 active 设置的效果。代码 20-10 表示 active 的用法。

代码 20-10 active 应用

```

-----文件名: active应用.html-----
01 <html>
02 <head>
03 <title>: active属性</title>
04 <style>
05 /*设置链接的active状态*/

```



图 20-9 :hover 属性的用法

```

06 a.demo:active{
07     background:#CCFFFF;
08     color:#000000;
09 }
< 10 /*设置层的样式*/
11 #idCodeDiv{
12     width:100%;
13     padding:4px;
14     /*设置字体*/
15     font-family:verdana,tahoma;
16     margin:12px 0px 0px 0px;
17     background-color:#EEEEEE;
18     /*设置字体为粗体*/
19     font-weight:bold;
20 }
21 </style>
22 </head>
23 <body bgcolor = lightyellow>
24 <center>
25     <a class=demo href="none.html" onclick="return false;">床前明月光；疑是地上霜。举头
26     望明月；低头思故乡。<br></a>
27 </center>
28 <br>
29 <br>
30     <div id=idCodeDiv> a :active{background:#CCFFFF;color:#000000;}
31 </div>
32 </body>
33 </html>

```

【代码解析】第 6~9 行代码演示了:active 属性的使用方法，当对象获得焦点时，样式效果就是 active 所设置的那样，如图 20-10 所示。

20.7.4 访问过后的状态

在浏览网页中，发现有的链接颜色发生了变化了，变化后的颜色刚好是访问过后的链接，那些都是默认的访问过后的状态，从默认情况上看，访问过后的颜色不是很好看。要想改变访问过后的状态，用 visited 可以设置访问过后的状态，其完整语法如下。

```

a:visited{
    /*property: value;*/
}

```

设置 a 对象在其链接地址已被访问过的样式。IE 将:link 伪类的样式表属性作用于:visited 伪类，默认值由浏览器决定。定义网页过期时间或用户清空历史记录将影响此伪类的作用，对于无 href 属性（特性）的 a 对象，此伪类不发生作用。请参阅 body 对象的 vlink 属性（特性）和 document 对象的 vlinkColor 特性。



图 20-10 : active 属性

代码 20-11 表示 visited 的用法。访问过后的链接样式就是 visited 的样式，当然，可以设置成与 link 一样。

代码 20-11 visited 应用

```
-----文件名: visited应用.html-----
01 <html>
02 <head>
03 <title>:visited属性</title>
04 <style>
05 /*设置链接访问后的状态*/
06 a.demo:visited {
07     background:#CCCCFF;
08     color:#000000;
09 }
10 #idCodeDiv{
11     width:100%;
12     padding:4px;
13     font-family:verdana,tahoma;
14     margin:12px 0px 0px 0px;
15     background-color:#EEEEEE;
16     font-weight:bold;
17 }
18 </style>
19 </head>
20 <body bgcolor = lightyellow>
21 <center>
22     <a class=demo href="none.html" onclick="return false;">床前明月光;疑是地上霜。举头
23     望明月;低头思故乡。<br></a>
24 </center>
25 <br>
26 <br>
27     <div id=idCodeDiv> a :visited{background:#FF3300;color:#00000
28 </div>
29 </body>
30 </html>
```

【代码解析】第 6~9 行代码演示了:visited 属性的使用方法，访问过后状态会变化，而不是默认的值，其效果如图 20-11 所示。

说明

伪类的使用主要是对链接的样式进行控制，大部分网页中的应用都建立在链接标记<a>基础上，其中在上面四个状态中，访问链接时的link和鼠标指针放置链接对象上的hover应用机会更多、更常用，所表示的效果也不一样。



图 20-11 :visited 属性

20.8 CSS 中的单位

单位不管在计算机还是化学、物理中都存在，而且计算机中的单位也不少，但在本书中，只介绍在页面布局时常使用的长度和颜色单位。

20.8.1 长度单位

长度单位用来表示对象的尺寸、大小等距离值。在计算机中，对长度有不同的长度单位表示，每个单位所包含的尺寸不一样。距离属性可以配合长度单位使用，如可以用数值加长度单位 px 表示对象的宽度，同理也可以用 em 表示。在计算机中，长度单位有 px、em、ex、pt、pc、in、mm 和 cm。以下是各长度单位的用法说明。

- px: 像素。像素是相对于显示器屏幕分辨率而言的。
- em: 相对于当前对象内文本的字体尺寸，如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。
- ex: 相对于字符“x”的高度，此高度通常为字体尺寸的一半。如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。
- pt: 点。
- pc: 派卡 (Pica)，相当于我国新四号铅字的尺寸。
- in: 英寸。
- mm: 毫米。
- cm: 厘米。

上述长度单位中，最常用到的是 px 和 em。为了方便理解这些长度单位，代码 20-12 对三个层对象都设置数值为 10，高和宽的长度单位不同。

代码 20-12 长度单位

```
-----文件名: 长度单位.html-----
01 <html>
02 <head>
03 <title>长度单位</title>
04 <style type="text/css">
05 /*对div标记设置*/
06 div
07 {
08     /*设置本页面所有div外边距为5像素*/
09     margin:5px;
10     /*边框为1像素红色实线*/
11     border:1px solid red;
12     /*层对象向左漂移*/
13     float:left;
14 }
15 #length1
16 {
17     /*对象的宽为10像素*/
18     width:10px;
19     /*对象的高为10em*/
```

```

20     height:10em;
21 }
22 #length2
23 {
24     /*对象宽为10ex*/
25     width:10ex;
26     /*对象高为10点*/
27     height:10pt;
28 }
29 #length3
30 {
31     /*对象宽为10派卡*/
32     width:10pc;
33     /*对象高为10厘米*/
34     height:10cm;
35 }
36 </style>
37 </head>
38 <body>
39     <div id="length1">设置了10px和10em</div>
40     <div id="length2">设置了10ex和10pt</div>
41     <div id="length3">设置了10pc和10cm</div>
42 </body>
43 </html>

```

【代码解析】第 6~14 行代码对 3 个层对象都设置一样的边框，且往左漂移。第 15~21 行代码为第 1 层，设置为宽 10 像素，高 10em；而第 22~28 行代码为第 2 层，数值也是 10，但宽为 10ex，高为 10 点，与之前设置的长度单位都不一样；第 29~35 行代码的第 3 层，宽和高分别为 10 派卡和 10 厘米。这样 3 个层的数值都为 10，因不同的长度单位所表达的层大小也不同，效果如图 20-12 所示。



图 20-12 长度单位

说明 只要了解px、em和pt就可以了，但最常用的是px，表示多少像素。

20.8.2 颜色单位

在前面学习的内容中，也常涉及颜色值的表示，本书最常用的表示边框的颜色为 red，红色，通过颜色名来表示。类似的，还有其他表示方式一样可以表示颜色值。在计算机中用来表示颜色单位的有：RGB 表示法 rgb (RGB)，颜色名称表示法和 #RRGGBB 十六进制正整数表示法。以下是各表示法的说明。

RGB 表示法如下。

R: 红色值。正整数 | 百分数
G: 绿色值。正整数 | 百分数
B: 蓝色值。正整数 | 百分数

以上三个参数，正整数值的取值范围为：0~255，百分数值的取值范围为：0.0%~100.0%，超出范围的数值将截取其最接近的取值极限。如 rgb (10%, 20%, 30%) 表示一种颜色。

颜色名称表示法直接用颜色表示，很直观，但表示的颜色有限。如红色用 red，白色用 white，绿色用 green。下面是能用颜色名称表示的颜色。

Black, White, Red, Yellow, Lime, Aqua, Blue, Fuchsia, Gray, Silver, Maroon, Olive, Green, Teal, Navy Purple.

#RRGGBB 十六进制整数表示法如下。

□ RR: 红色值。十六进制正整数
□ GG: 绿色值。十六进制正整数
□ BB: 蓝色值。十六进制正整数

以上三个参数，取值范围为：00~FF。参数必须是两位数，对于只有一位的，应在前面补零。代码 20-13 表示颜色单位在网页中的应用。

代码 20-13 颜色单位

```
-----文件名：颜色单位.html-----
01 <html>
02 <head>
03 <title>颜色单位</title>
04 <style type="text/css">
05 div
06 {
07     /*设置层标记的宽、高和漂移*/
08     /*所有层宽为100像素*/
09     width:100px;
10     /*所有层高为100像素*/
11     height:100px;
12     /*外边距为5像素*/
13     margin:5px;
14     /*向左漂移*/
15     float:left;
16 }
17 #color1
18 {
19     /*用rgb来表示颜色值*/
20     background:rgb(12%,120,120);
```

```

21     border:1px solid rgb(152,50%,20%);
22 }
23 #color2
24 {
25     /*用颜色名称来表示颜色值*/
26     background:blue;
27     border:1px solid red;
28 }
29 #color3
30 {
31     /*用16进制表示颜色值*/
32     background:#cccccc;
33     border:1px solid #bb00dd;
34 }
35 </style>
36 </head>
37 <body>
38 <div id="color1">
39     这里背景的颜色为rgb(12%,120,120)，边框为rgb(152,50%,20%)。
40 </div>
41 <div id="color2">
42     这里背景颜色为blue，边框颜色为red。
43 </div>
44 <div id="color3">
45     这里背景为#cccccc，边框为#bb00dd。
46 </div>
47 </body>
48 </html>

```

【代码解析】第 38~46 行代码用 3 个大小相同的层表示不同的颜色单位用法，设置参数不同所表达的颜色不一样。第 17~22 行代码为第 1 层，用 rgb 表示法，背景颜色设置为 (12%，120，120)，而边框也用 rgb 表示，其值为 (152，50%，20%)；第 29~34 行代码为第 2 层，用颜色值表示法，背景颜色用 blue，而边框颜色设置为红色；最后，第 23~28 行代码为第 3 层，设置的背景颜色为 #cccccc，而边框颜色值为 #BB00DD。这些层的效果如图 20-13 所示。



图 20-13 颜色单位

注意 用#RRGGBB表示时，前面有“#”，而用颜色名称表示时，没有前面的“#”，同时用RGB表示时，别忘了写rgb。

20.9 小结

本章首先介绍了选择符的用法，选择符分成通配选择符、类型选择符、包含选择符、ID 选择符、类选择符；然后介绍了伪类，伪类可以看做是一种特殊的类选择符，最大的用法是对文本或图

像链接在不同状态下定义不同的样式；最后介绍了之前在布局中常用到的单位，有长度单位和颜色单位，其中长度主要用对象的长度或宽度时的尺寸表示，也包括对对象细节性的调节，如内外边距 (margin, padding) 的表示，颜色最多用于对象的背景和边框还有文字等方面，可以用 RGB 表示法，用红、绿、蓝三种颜色来表示一个颜色值，也可以直接用颜色，如红色 red、黄色 orange，但用的最多的是 #RRGGBB 表示法，这种用十六进制正整数表示更受欢迎，因此最常用。

20.10 习题

一、填空题

1. 选择符分成_____、_____、_____、_____和_____。

2. RGB 表示法，用_____、_____和_____三种颜色来表示一个颜色值。

3. ID 选择符是指在 DOM 中用为唯一标识符的 ID 为选择符，ID 标识符以“_____”打头。

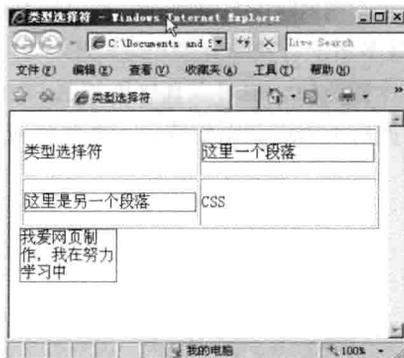
二、上机实践

1. 找出下面代码的错误。

```
<html>
<head>
<title>类选择符</title>
<style type="text/css">
.class1
{
    width:100px;
    height:50px;
    border:1px solid red;
}
#class2
{
    width:200px;
    height:60px;
    border:1px solid blue;
}
</style>
</head>
<body>
<div id="class1">
    class1的层
</div>
<div class="class2">
    class2的层
</div>
</body>
</html>
```

【提示】 注意 ID 选择符与类选择符的区别。

2. 按下图编辑一个网页，注意用到了哪些选择符。



【提示】

#ID-----ID标识符
E1 E2-----包含选择符

3. 做一个红色的访问链接，鼠标在其上时显示为蓝色，背景为绿色。

【提示】

link-----链接样式
hover-----鼠标光标置于其上的样式

4. 做一个链接，单击链接为红色，单击后状态为蓝色，背景为绿色。

【提示】

active-----单击样式
visited-----单击后的样式

5. 用不同的单位设置 6 个层的高度，高度数值都为 10，宽度为 100%。

【提示】

em-----相对于当前对象内文本的字体尺寸
pt-----点
cm-----厘米
px-----像素
pc-----派卡
ex-----此高度通常为字体尺寸的一半

6. 设置一个层，边框为红色，高度数值都为 10px，宽度为 100%。

【提示】

border : border-width || border-style || border-color-----边框属性

第四篇

实践部分

第 21 章 游戏网站布局

本章演示如何制作一个游戏风格的主页面。为了表现特殊的效果，网页中用到了很多图片，网站的制作流程是先做好页面的美术，如果自己没有美术功底，可以请专业的美术设计人员设置网页风格，通过 Photoshop 等软件绘出网页的效果图，然后将效果图放置到一个专用的保存图像的文件夹中，如本实例的 images 文件夹，其他的事情由网页设计师完成。

在保存文件时，要注意将网页中的图片专门放置在一个文件夹中，取名与文件夹内容相关，如 image。再如，网站需要上传文件时，可以设置文件夹为 upload。CSS 文件比较多，也需要专门新建 CSS 文件夹。经常需要的文件夹有 CSS、图片和脚本（JavaScript）。其他的文件夹根据页面二级目录而定，如游戏网站的新闻系统，与新闻相关的页面将被放置在 news 文件夹下面，同时，网页访问时也会带上这个目录。

在学习操作本章实例前，先准备网页中需要的效果图片，然后通过切片技术，将图片切成网页中需要的效果图片，来弥补 HTML 或 CSS 不能表现出来的效果，如圆角、按钮等，也可以将整个图片拿来作背景图像。

学完本章，可以了解到以下内容：

- 网站主页规划；
- 添加图片；
- 加入文字；
- 调整全页面兼容性。

21.1 网站主页规划

本实例将围绕一个游戏主题类的站点布局来展开，包含的内容有公司信息、游戏信息、游戏下载、商务合作等。实例中只针对站点主页布局，其他的页面可以根据主页面的布局来修改成要实现的页面，这样就不必重复地进行布局操作，从而也可以更好地保持整个网站的风格。

这个游戏站点以中国风为主题，网页顶部是中国古代建筑风格，将 LOGO 放在中间，其中顶端还有两个小柱子，中间部分为左边链接栏和右边的内容为显示区。在样式方面，链接栏的按钮用

到带有花边的图片,中间的文本把默认字体修改了,而不是默认的宋体,下一排文本公告区域的背景图片用扇形样式呈现。在右边内容显示区中,边框样式用到古代建筑风格的花边。与下端一样,中间的心形都是对着内容的,不同的是下面的花边整条都在网页上,而右边部分又分成两块,上部内容用移动的效果表现出来,下部内容通过四个角用隶书字体表示,然后加上一些友情链接,所表现出的效果如图 21-1 所示,该图也是该主页的整体效果图,需要对该图进行切片处理。



图 21-1 主页效果图

本实例布局采用从上到下,从左到右的顺序进行。所谓从上到下是指从上面布局设置开始,然后往下面内容进行布局设置。如果在某一部分,存在左、右分栏或者左、中、右的分栏时,通过布局顺序从左往右进行布局设置。这样确定好布局的顺序后,网页设置将要通过准备好的图片素材来进行网页布局。

21.2 用 DIV+CSS 布局

当确定好效果图以及对其切片处理后,就需要对这个页面布局了。除了上面提到的用顺序布局外,还将要用到布局的核心知识,即用 DIV 和 CSS 结合控制 HTML 效果。CSS 样式将会单独出来一个文件,通过外部链接的方式把该文件引用进来,然后用顺序布局的方式进行布局。下面分别对不同顺序的内容布局进行介绍。

21.2.1 实现页面头部

确定好页面布局方式和方法后,现在开始对页面的头部部分进行布局。从构思和图片切图(先

由美术工作人员对网页设置效果图，然后根据需要对外效果图切图，提供给编写 HTML 的工作人员使用) 中可以知道，头部只由一张图片组成，其中 LOGO 标志也被包含在头部的图片中。

布局的关键部分是将图片放置在 DIV 层中，且设置头部的层为居中对齐，然后通过 CSS 样式控制层的选择符。对于 CSS 样式设置，本实例将样式表单独保存为一个以 CSS 为扩展名的文件，然后通过外部链接引入到页面中来，除了可以节省本张网页的代码空间外，还可以链接到其他页面中，即需要新建其他网页时，只需在该新建的网页上链接保存的样式文件即可。样式文件的链接如下所示。

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

外部链接 CSS 样式由标记 <link> 完成，该标记没有结束标记对，需要在结束尖括号前输入斜线表示结束，参数 type 表示链接的类型，上面代码代表样式文本，rel 表示为样式表，这个参数在 <link> 标记中很重要，如没有设置本参数，网页与样式文件不能成功关联。最后需要链接样式文件时，用到 href 参数，表示文件的地址，实例中的样式文件在网页的同一目录，用到的是相对路径。

链接好样式表后，在 <body></body> 标记对中设置层布局，即 DIV 布局，然后确定好选择符。如果选择符在本网页中只出现一次，即可以设置为 ID 选择符；如果需要复合出现同一个样式，设置类 (CLASS) 选择符效率更高。下面是对 DIV 的布局设置，头部表示一张图片，用到 标记把图片引用进来，如代码 21-1 所示。

代码 21-1 游戏网站布局

```
-----文件名：游戏网站布局.html-----
01 <html>
02 <head>
03 <title>游戏网站布局</title>
04 <link type="text/css" rel="stylesheet" href="style.css" />
05 </head>
06 <body>
07 <!--设置头部，引用图片-->
08 <div class="box"> </div>
09 <!--以下是其他位置的布局-->
10 </body>
11 </html>
```

【代码解析】第 8 行代码设置层的类 (CLASS) 选择符为 “box”，表示为容器层，在布局的其他位置也可以用到该类选择符。设置好 DIV 布局并确定选择符后，需要在 CSS 样式中，来控制这些标记，首先需要用 CSS 对整个网页做一些设置，如网页的背景颜色、网页中的字体大小、<body> 标记的内外边距等。在 CSS 中，对上面的类 (CLASS) 选择符设置宽度属性和层居中对齐，在设置好宽度后才可以让层居中，而且不能缺少网页的标准声名，在 <html> 中输入以下代码：

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

以上代码表示网页在 IE 浏览器中所声明的网页文档类型，而在火狐 (FF) 浏览器中可以不加，但要兼容这两个主流浏览器，所以本实例添加上面的代码。

综上所述，如果设置层居中需要三个条件：第一为对 doctype 的声明；第二为层宽度的确定；第三也是关键一步，用 CSS 设置 margin 属性值为 auto。下面是 CSS 代码对上面完整 HTML 标记控制的代码。

```

/*CSS代码：控制头部*/
/*设置全局样式*/
01  body
02  {
03      /*设置整个网页的内外边距*/
04      margin:0;
05      padding:0;
06      /*设置字体大小*/
07      font-size:12px;
08      /*设置背景颜色*/
09      background:#a3a3a3;
10  }
11  /*设置容器层，主要功能为层居中*/
12  .box
13  {
14      /*确定宽度*/
15      width:955px;
16      /*设置外边距为auto*/
17      margin:auto;
18  }

```

【代码解析】以上代码是对网页主体的设置。第15行代码设置了宽度为995px，第17行代码设置了外边距margin为auto。

通过上面的设置，所表现出来的效果为一张图片在一个层中，这张图片正好是网页中的头部信息，宽为995像素。现在网页的标准布局宽度有778像素和1002两种，这两种会经常用到。在分辨率为800*600情况下，网页宽度保持在778像素以内合适，而分辨率为1024*768或者更高时，网页宽度保持在1002以内最合适。本实例的网页相对1024*768分辨率的用户，宽为995像素，显示器分辨率更高的用户在浏览该主页时，浏览器的左右两边还会出现空白空间，效果如图21-2所示。



图 21-2 设置网页的头部

注意

在HTML标记中，用<center></center>可以设置对象居中，而上面所讲的居中是用CSS设置DIV层的居中。

21.2.2 实现中间部分

头部的 HTML 与 CSS 布局部分比较简单,本节演示的是页面的中间部分,中间部分分成两栏:左栏内容和右栏内容。其中,左栏内容主要是一些链接信息,由一些表示该链接信息的图片组成,对图像设置链接;右栏内容为内容显示区,将显示左栏中的链接页面的信息,在样式设置时 CSS 内容比较多,如四角有中国风的边框,顶部有条形花纹显示这都需要大量的样式设置。

在中间部分先对左栏区域设置,先查看图片的大小,取得图片大小后设置左栏的宽度为图片的宽度,然后在层中用标记

在 DIV 布局设置中,先插入一个容器层,在这个容器层中已设置好层的居中对齐,对左栏的选择符设置 ID 为 left,把将要链接的图片放在本层下面。需要注意的是,图像标记之间空格也会影响整个效果的显示。HTML 设置如代码 21-2 所示。

代码 21-2 游戏网站布局

```
-----文件名: 游戏网站布局.html-----
01 <html>
02 <head>
03 <title>游戏网站布局</title>
04 <link type="text/css" rel="stylesheet" href="style.css" />
05 </head>
06 <body>
07     <div class="box"></div>
08     <!--中间布局部分设置-->
09     <div class="box">
10         <!--设置左栏的层 -->
11         <div id="left">
12             <!--插入图像-->
13                
16             </div>
17         </div>
18 </body>
19 </html>
```

【代码解析】第 11~16 行代码对左栏区域设置了层,然后把图像插入到层中,标记设置完后,还得设置 CSS 样式,样式的设置在原来样式基础上,从尾部添加新的选择符名称和代码即可,如是 ID 在名称前用“#”。类似地,类选择符名称用“.”表示,同时也可以对代码进行注解,注解用/**/表示。

下面代码表示在原来的样式文件后面添加新的样式代码,对左栏的层和其内的图像样式进行设置,添加后的代码如下。

```
01 #left
02 {
03     /*设置左栏的宽度和高度*/
04     width:300px;
05     height:565px;
06     /*设置层向左漂移*/
07     float:left;
```

```

08      /*背景用图片显示*/
09      background:url(images/line1.gif);
10  }
11  /*对left下的图像进行设置*/
12  #left img
13  {
14      /*设置为无边框*/
15      border:0;
16      /*左边的外边距为7像素*/
17      margin-left:7px;
18  }

```

【代码解析】第 12~18 行代码为包含选择符，left 的选择符里的图片样式为无边框，距离父对象左边 7 像素。

设置好左栏的标记和样式后，先从浏览器中查看效果，在 CSS 布局过程中，常常在修改后要及时查看页面显示效果，如效果不满意，再修改 CSS 样式，然后再来浏览页面效果，直到满意为止。通过上面添加 HTML 标记和 CSS 样式，对左栏区域布局才完成，显示出的左栏的效果如图 21-3 所示。

注意

属性 float 的设置需要在确定对象宽度情况下进行，否则，没有漂移的效果。

在设置好左栏区域后，对右栏区域的内容显示也要布局设置。先对右栏区域设置层布局，设置好宽度值，且向左漂移，使之与左栏的层显示在同一水平线上。设置容器层，容器层在右边显示，里面包含一个小层，设置这个层为相对定位，里面包含四个层，每个层表示一个角，且设置为绝对定位确定好位置。即右栏的布局先为容器层包含一个层，而每个层里面又包含四个角的样式层。设置 HTML 标记的代码如下。

```

<!--设置右边栏的标记-->
01  <div id= "right">
02      <!--包含的层-->
03      <div id="rightbox">
04          <!--设置两柱子-->
05          <div id="floatleft"></div><div id="floatright"></div>
06          <!--设置四个角-->
07          <div id="cornortl"></div><div id="cornortr"></div><div id="cornortbl"></div><div
08  ^ id="cornorbr"></div>
09      </div>
10  </div>

```

【代码解析】以上代码添加了右栏布局层，容器层样式选择符为 right，第 5 行代码设置柱子的层，第 7~8 行代码是四个角的设置。



图 21-3 左栏设置效果

设置好标记后,需要用 CSS 样式控制这些标记,然而,在右边栏的设置之初,到层下面再添加层时,样式可能还需要调动。通过上述分析,在添加代码时,只在原来 CSS 代码后面添加以下代码。

```
/*设置右边栏容器层*/
01 #right
02 {
03     /*宽度与左栏刚好与整体对齐*/
04     width:655px;
05     /*向左漂移*/
06     float:left;
07     /*容器层相对定位*/
08     position:relative;
09     /*背景颜色为白*/
10     background:#ffffff;
11 }
12 /*设置左边柱子的定位*/
13 #floatleft
14 {
15     /*为绝对定位*/
16     position:absolute;
17     /*设置宽度和高度*/
18     width:53px;
19     height:31px;
20     /*与顶点的距离为0*/
21     top:0px;
22     /*负值表示超过左边48像素*/
23     left:-48px;
24     /*把柱子图片设置为背景图片*/
25     background:url(images/float1.gif) no-repeat;
26 }
27 /*设置右边的柱子*/
28 #floatright
29 {
30     /*设置绝对定位*/
31     position:absolute;
32     /*设置宽度和高度*/
33     width:53px;
34     height:31px;
35     /*与顶点的距离为0*/
36     top:0px;
37     /*距离左边边框227像素*/
38     left:277px;
39     /*把柱子图片设置为背景图片*/
40     background:url(images/float2.gif) no-repeat;
41 }
42 /*设置右边的内容区*/
43 #rightbox
44 {
45     /*设置层与外面的距离值*/
46     margin:12px 0px 12px 12px;
47     /*边框样式*/
48     border:1px solid #42605e;
```

```
49     /*设置本层的宽度和高度*/
50     width:591px;
51     height:500px;
52     /*设置内边距*/
53     padding:20px;
54 }
55 /*设置顶部右边的角*/
56 #cornortr
57 {
58     /*引入图片*/
59     background:url(images/t_r.gif) no-repeat;
60     width:56px;
61     height:56px;
62     /*设置了绝对定位*/
63     position:absolute;
64     /*设置定位位置,距右边框6像素*/
65     right:6px;
66     /*到顶部14像素*/
67     top:14px;
68 }
69 /*设置底部左边的角*/
70 #cornortl
71 {
72     background:url(images/t_l.gif) no-repeat;
73     width:56px;
74     height:56px;
75     position:absolute;
76     left:9px;
77     top:13px;
78 }
79 /*设置底部左边的角*/
80 #cornortbl
81 {
82     background:url(images/b_l.gif) no-repeat;
83     width:56px;
84     height:56px;
85     position:absolute;
86     left:9px;
87     bottom:9px;
88 }
89 /*设置底部右边的角*/
90 #cornorbr
91 {
92     background:url(images/b_r.gif) no-repeat;
93     width:56px;
94     height:56px;
95     position:absolute;
96     right:6px;
97     bottom:10px;
98 }
```

【代码解析】以上代码为右边栏层的样式设置,第8行代码把容器层设置为相对位置,以此来约束其在网页中的位置。容器层里面的其他层为绝对位置,表示根据容器层显示的位置来确定。

通过上面的 CSS 样式设置，漂移向右边的层为容器层，其中容器层又包含了两个小柱子所在的层和一个层中的四个角所在的层。要注意的是，绝对定位的设置应根据层与层之间的距离而设置，表现出来的效果如图 21-4 所示。

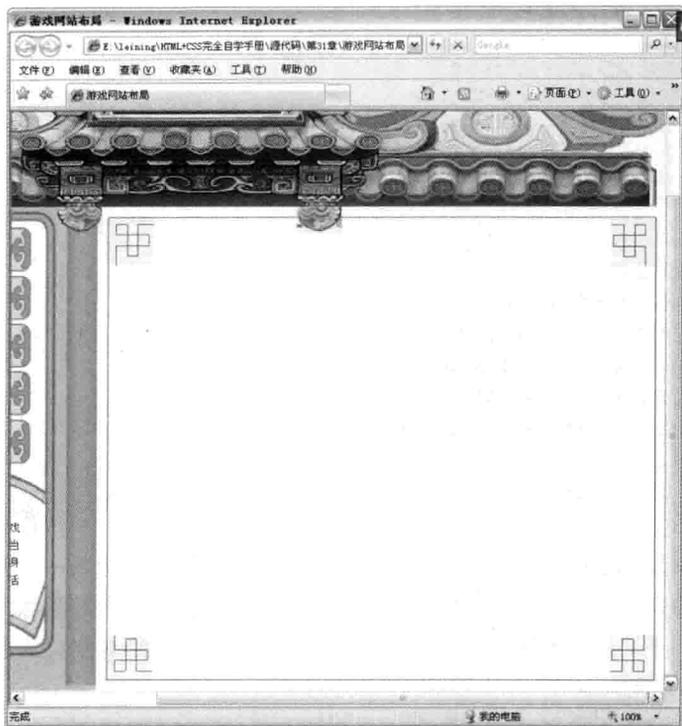


图 21-4 右边栏容器层

右边容器层设置好后，需要对右边容器层中设置一个小层，这个小层包含的信息是设置可移动的对象，且在层的顶层设置中国风的花纹，边框宽度和颜色设置与上面容器层设置一样，本层与上面层按顺序显示，其中两个柱子的层需要搬移到这个层来，因为本层在容器层上面。标记设置代码如下。

```

<!--右边的大容器层 -->
01 <div id= "right">
02     <!--添加右边上面层包含柱子-->
03     <div id="rightup">
04         <!--两柱子的层-->
05         <div id="floatleft"></div><div id="floatright"></div>
06     </div>
07     <!--上面介绍过的层，里面包含四个角-->
08     <div id="rightbox">
09         <!--4个角-->
10         <div id="cornortl"></div> <div id="cornortr"></div><div id="cornortbl"> </div> <div
11 id="cornorbr"></div>
12     </div>
13 </div>

```

【代码解析】代码第 3 行在原有的层中添加一个层 ID 为 rightup 的选择符，然后把原来两个柱子所在的层 rightbox 中的 floatleft 和 floatright 移到 rightup 中，且把 rightup 层设置成像一个花边框，整个层作为一个表框通过在层的背景中添加图片实现，且从 X 轴重复显示即可。这样的变换要改动 CSS 的样式，样式设置代码如下。

```
/*设置右边上面的层*/
01 #rightup
02 {
03     /*设置宽度*/
04     width:619px;
05     /*设置外边距*/
06     margin:12px;
07     /*设置内边距*/
08     padding:20px 0px 0px 10px;
09     /*设置边框与上面的层一样*/
10     border:1px solid #42605e;
11     /*设置背景图片，且x轴循环*/
12     background:url(images/p.gif) repeat-x;
13     /*相对定位*/
14     position:relative;
15 }
```

【代码解析】添加 rightup 层后，在显示效果上，把两个柱子包含其中，然后放在层 rightbox 前面，两层之间为并列关系。在设置好层之间的关系后，只需要对层内部添加内容即可，同时对将要遇到的问题准备一下，如文本放置进去后，可能太靠近边框，被绝对定位的层覆盖掉等。添加层后的效果如图 21-5 所示。

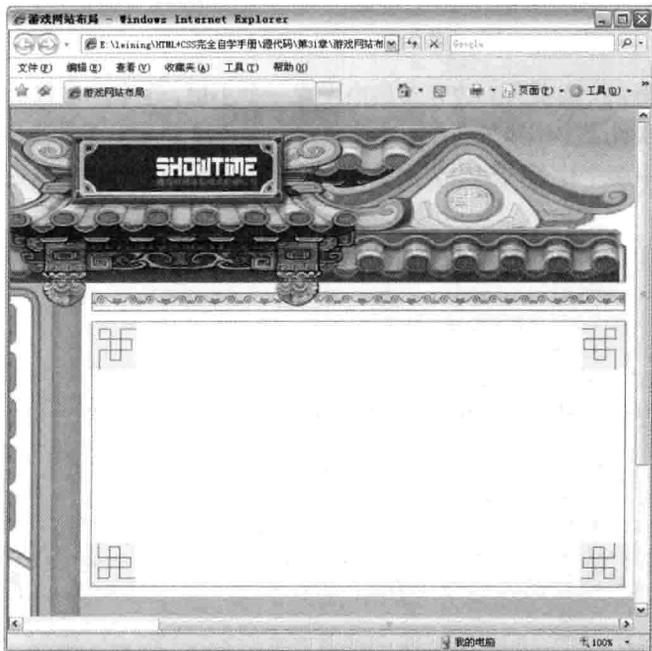


图 21-5 右栏添加层

通过上面步骤，右边栏区域就设置好了，该区域布局的关键是对柱子、四个边框角、花框图等布局效果，主要用到绝对定位来确定对象的定位。在 CSS 中设置背景时，链接入背景图片使层中显示该图片。

21.2.3 实现页面底部

底部的设置很简单，但需要设置底部中国风的花边框，然后用一条渐变图像作为背景，里面的文本就是底部信息，其中层布局是指在前面定好的 box 容器层中设置 ID 为层，然后用设置 ID 为 bottom 的选择符来表示页脚信息，这些文本信息设置为居中，HTML 代码如下。

```
<!--页脚信息-->
01     <div class="box">
02         <div id="b">
03             </div>
04             <div id="bottom">
05                 页脚
06             </div>
07     </div>
```

【代码解析】第 1 行代码定义容器层的样式为 box，第 4~6 行代码为页脚层。

设置 HTML 标记后，需要用 CSS 来控制这些标记。在标记中，box 是前面已定好的类选择符，可以在一个页面中重复使用，而且表示居中对齐，对其他两个层设置主要用来表达网页底部的样式。其代码如下。

```
/*CSS样式：表示页脚信息*/
01 #bottom
02 {
03     /*设置背景图片*/
04     background:url(images/xia.gif);
05     /*文本居中*/
06     text-align:center;
07     /*设置高度，以免层过高*/
08     height:62px;
09 }
10 #b
11 {
12     /*设置高度，高度刚好是背景图片的高度*/
13     height:24px;
14     /*清除两边的漂移*/
15     clear:both;
16     /*设置背景图片*/
17     background:url(images/b.gif) repeat-x;
18 }
```

【代码解析】第 10~18 行代码为底层 b 的设置，图片 b.gif 按横轴平铺一层，bottom 在 b 层中，对文字进行约束。

上面 HTML 和 CSS 的代码对底部的层进行了设置，包括设置页脚层的顶端为花边框，其下层的背景为一条渐变颜色，而文本是属于居中的，每条文本都会在中部显示，效果如图 21-6 所示。



图 21-6 设置底部

21.3 加入文字

网页中层的布局主要是显示文本信息，来表达站点的信息内容，在添加层中的文本内容后，文本在 CSS 里面的默认效果是不美观的，为了改变这些值，需要对 CSS 文本属性再次设置。本实例在中间部分的右边添加文本，文本内容根据页面的性质决定。下面是对游戏公司信息的简单介绍。文本添加分成两步：第一步为中间右边栏只有一个层的效果；第二步为添加完层 `rightup` 后的效果。代码如下所示。

```

01 <div id="right">
02     <div id="rightup">
03         <div id="floatleft"></div><div id="floatright"></div>
04         <marquee direction="up" height="185px">
05             <h4>2007年</h4>
06             <p>5月，X' Zone游戏平台项目启动，项目团队为禧添公司前身。</p>
07             <p>7月，X' Zone项目前期市场调研工作完成，开始一期研发工程。</p>
08
09             <h4>2008年</h4>
10             <p>1月，X' Zone项目团队入驻无锡新区创新创意园区。</p>
11             <p>3月，无锡禧添（SHOWTIME）网络信息技术有限公司正式成立。
12 </p>
13         </marquee>
14     </div>
15     <div id="rightbox">
16         <p>2008年3月5日，一群专注于互联网互动娱乐内容的年青人汇聚在无锡，将各自的梦想
17 合而为一，再炼成龙骨、铸成蒙皮、锻成甲板、编成风帆，造出一艘源于梦想承载梦想的船，名为“禧添”！
18 </p>
19         <p><strong>无锡禧添（SHOWTIME）网络信息技术有限公司</strong>，开始启航！
20 </p>
21         <p>禧添公司是无锡第一家由无锡人自己创办的专业互联网互动娱乐公司，在成立时即得到了
22 无锡市政府相关部门和新区信软办的大力支持。公司注册资本100万，主要从事互联网互动娱乐软件的开发
23 和运营，以及互联网增值服务和人才培养等业务。 </p>
24         <p>目前，禧添公司落户在无锡新区创新创意园区，各项业务正在迅速开展中。在未来，禧添将
25 秉持“激情、专心、细致”的方针，以最大的努力，向中国广大的互联网用户提供最优质的服务。 </p>

```

```

26 <div id="cornortl"></div><div id="cornortr"></div>
27 <div id="cornortbl"></div><div id="cornorbr"></div>
28 </div>
29 </div>

```

【代码解析】对 rightbox 层加入文本后，显示的内容被全局设置的 body 所控制，如字体大小为 12 像素，字体为默认的宋体等；在层 rightup 中，设置了文本向上移动，用到标记<marquee>（第 4~13 行代码）。效果如图 21-7 所示。



图 21-7 加入文本后

对文本的设置还是默认值，效果如图 21-7 所示。可以看出，对文本布局还没有把握好，文本看起来较小，而且字体也不好看。要对这些字体等文本属性进行调整，需要对两个包含这些文本层的 CSS 设置。设置代码如下。

```

/*设置右边的内容区，上面的层*/
01 #rightup
02 {
03     width:620px;
04     margin:12px;
05     padding:20px 0px 0px 10px;
06     border:1px solid #42605e;
07     background:url(images/p.gif) repeat-x;
08     position:relative;
09     /*设置文本字体*/
10     font-family:"隶书";

```

```

11      /*设置文本大小*/
12      font-size:14px;
13  }
14  /*设置右边的内容区, 在rightup下面的层*/
15  #rightbox
16  {
17      margin:12px 0px 12px 12px;
18      border:1px solid #42605e;
19      width:551px;
20      height:240px;
21      position:relative;
22      padding:40px;
23      /*设置文本字体*/
24      font-family:"隶书";
25      /*设置文本大小*/
26      font-size:15px;
27      /*设置字体颜色*/
28      color:#42605e;
29  }

```

【代码解析】通过设置这两个层的属性,使其内容的文本发生了变化,文本字体变了,不是默认的宋体,变成有中国风的隶书(第10、24行代码),字体大小不再是在body中设置的12像素,分别设置为14像素(第12行代码)和15像素(第26行代码),而对rightbox的字体颜色设置为#42605e(第28行代码),与边框颜色一样。变化后的内容区效果如图21-8所示。



图 21-8 修改后的文本效果

对字体文本的设置可以定义一个类选择符,因有的文本不在同一段落,但设置的效果是一样的,这样可以提高布局的效率。

对定位的设置需要注意的是,在一个容器层中设置相对定位,然后在其层中包含绝对定位的层,即在相对定位中包含绝对定位,没有设置相对定位的话,直接用绝对定位时,位置是相对 body 移动的,即所设置的位置从页面的左顶部开始。在本实例中,中间部分的左边和右边都设置 float 为 left,向左漂移,当在底部开始布局时,所设置的层往上面移动,这是不合理的设置,也是初学者很难把握的一点。在本实例中解决的方法是对漂移到上面的层设置 clear,即本实例中的 ID 为 b 的层设置了 clear 值为 both,表示不向两边漂移。

21.4 调整全页面的兼容性

网页的布局通过 HTML 和 CSS 样式,完成了初期版本,需要修改的地方还有很多,而且最关键的是看在 FF(火狐浏览器)里能否运行,如出现层显示不当,或者漂移到不正确的位置,以及层与层之间的位置差不合理等现象。毕竟 IE 和 FF 支持的标准不同,所显示的效果也有差别,网页设计者应把这些差别减到最少。

为了让页面在 FF 里正常运行,先打开 FF 浏览器,然后把网页文件拖到 FF 浏览器中,或者在网页文件中右击,选择“打开方式”中的“Firefox”,即用 FF 浏览器打开页面,前提是在电脑上已安装该浏览器才可以打开,没有安装的用户可以到 <http://www.mozilla.com/> 下载最新版本。

在本实例中,游戏网站主页在 IE 中运行正常,为了使它在 FF 下也运行正常,需要在 FF 中打开,然后查看网页有哪些不正常的地方,如果有,先查找该错误的地方的层,通过修改 CSS 样式来改变网页中的效果,因为浏览器对 HTML 的支持都是一样的,不同的只是 CSS 中的属性或者属性值。本实例在 FF 中的运行效果如图 21-9 所示。



图 21-9 实例页面在 FF 中运行正常

21.5 小结

本章主要介绍了游戏主题类网站的主页布局，游戏主题类网站主要由图片组成，对部分细节要做细致处理，如圆角、中国风的边框效果等，其中部分文字需要用 CSS 控制。本章对布局中各个步骤做了仔细的讲解，理解了这类主题网站的风格布局和设置模块要点，还要根据不同的浏览器对网页的兼容做出调整。

布局中，首先对页面分成页面头部、页面中部和页面底部三个部分。头部的效果由一张图片组成，其中包含 LOGO。比较难的是中间部分，中间部分呈两块，即左边区域和右边区域，左边区域主要是网站的功能性链接，也是由图片组成，这些图片根据美术切图所定，加上链接是设置各张图片的超链接。而对右边的设置，是本实例的难点，操作起来比较麻烦，特别是在一个层中设置四个角的布局时，用到绝对定位，而且需在上面加上一个层用来移动显示内容，同时在页面头部的两个柱子上，也用到了定位，把两个柱子与网页头部“接”上。对页脚中的图案设置成一个层的背景 X 轴循环，然后设置成一个宽度为 955 像素且居中的页脚，同时，下面的背景图片用到宽为 1 像素的渐变图片，与上面的图案一样，对 X 轴重复显示渐变图片就可以看到背景效果。

完成的页面在 IE 中运行正常，且没有出现 CSS 的误差。为了在 FF 浏览器中也能正确显示，本实例在设置移动的标记 (marquee) 参数时，FF 中要加上长度单位才可以移动，若去掉 px，在 FF 下则不移动，请读者注意这些细节。

第 22 章 企业网站布局

本章开始演示企业主题的网站布局。本示例是一个企业网站主页的布局。在企业网站的组成结构中，网页数量比较少，而且页面风格都是相同的。可以把本实例看成一个模板页，对其他页面的布局，只需在“模板页”上添加或删除部分文字即可，把这些页面放到一个目录下，就可以组成一个企业网页了。

企业网站主要表达的是一个企业的信息，而企业信息都分成几个模块，如公司简介、联系我们、产品展示等。本实例主要以互联网上最常用的公司网站布局做介绍，如想制作更漂亮的网站，美术方面一定要细致，然后通过 HTML+CSS 在页面中表现出来。

学习完本章，将会学到以下内容：

- 企业主页的规划；
- 用 DIV 布局出标记；
- CSS 实现页面体；
- 调整网页中的 CSS 代码兼容不同浏览器；
- 其他网页的制作。

22.1 企业主页规划

企业主页用来表达企业信息，这是企业网站的主要目标。如何做到在一个网页中可以表达更多企业信息，如何在页面内容和外观上吸引用户，使之浏览更多内容，这都需要在页面的规划下功夫。本节讲解企业网站的规划，从功能模块上分，主要分成以下几个模块：

- 首页；
- 公司简介；
- 管理团队；
- 游戏产品；
- 招纳贤才；
- 法律声明；
- 联系我们。

上面的七个模块是企业网站常包含的功能模块，当然，还有的模块并没有列出，但网站的页面或多或少都会存在一个模板页面，也是确定该网站的风格页面。本实例就来制作这样的模板页，对于其他页面，在前面介绍过，只需要添加或删除部分文本或图片即可。

确定了页面模块，需要对网页的构思和风格花点儿工夫，即对网页的布局，首先考虑到这是企业性质的网站。按现在流行的网站风格来讲，有上、中、下模式，上指的是页面的头部；中是指是

页面的中间部分；下就是网页的下部信息部分。

对页面的构思可以按上、中、下的模式进行，然后分别在上、中、下模式上设置内容。如页面的头部分主要放置公司的 LOGO，而且 LOGO 放在左边位置，右边可以设置 468*60 标准的广告宣传栏，当然这不是为了做广告，而是公司自己的宣传栏。为了表现页面头部的美观性，在页面头部右边放几个文字链接，如中文版|英文版之类的链接。

页面的中部也是很重要的一个模块。在本实例中的中间部分，模块分成左右两部分，其中左边显示企业网站的各功能模块，如首页、公司简介等，右边是左边功能模块的对应内容，如公司简介放置的内容是企业的信息，在布局方面，左边的宽度明显会小于右边。左边只是表现出按钮的效果，而右边显示其内容，对内容的长度没有限制。

底部就是页面的页脚，都是放置一些公司信息，不太常用的链接和国家信息产业部的备案号等，这部分每个网页都是一样的。在做模板页时可以确定其内容，以便网页修改。页面效果如图 22-1 所示。

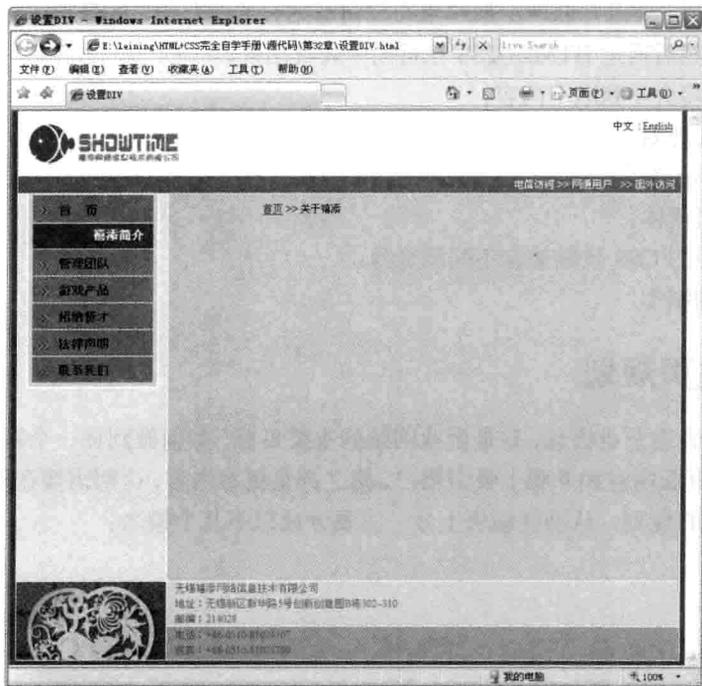


图 22-1 企业网站布局效果

技巧 也可以在页面头部和页脚单独做一个网页，然后在其他网页链接该页面时就可以表示页面头部和脚部了。修改一处可以应用到所有的网页，这样更方便，更容易管理，门户上的广告多采用这种方式。

22.2 对网页布局

在本节中，主要对网页的头部和中部以及网页页脚部分进行布局，首先在网页 HTML 标记中


```

29         <li><a href="law.htm">法律声明</a></li>
30         <li><a href="connectus.htm">联系我们</a></li>
31     </ul>
32 </div>
33 </div>
34 </div>
35     <!-- 网页的页脚-->
36 <div id="ender">
37 <!-- 用不同的层包含不同的内容-->
38 <div class="enderbox2">
39     <p>无锡禧添网络信息技术有限公司</p>
40     <p>地址: 无锡新区新华路5号创新创意园B栋302~310<p>邮编: 214028</p>
41 </div>
42 <div class="enderbox1"></div>
43 <div class="enderbox3">
44     <p>电话: +86-0510-81024707</p>
45     <p>传真: +86-0510-81024709</p>
46 </div>

```

【代码解析】第 1~8 行代码为页面布局的头部，第 9~35 行代码为页面的中间部分，剩下的层就是页面的底部。中间又分为左边栏和右边栏，左边栏主要为导航信息的链接列表（第 24~30 行代码）。

在以上代码中，页面的层设置只是布局第一步，先了解清楚页面的层构架后，才能更好地利用 CSS 来控制 HTML 标记，对第一个模块的功能，主要表现为页面的层设置和内容功能实现，所控制的范围才有明确的目标。以上代码在网页中，即为原始的 HTML 代码包含 DIV 层，没有任何对层进行设置的代码，效果如图 22-2 所示。

22.2.2 用 CSS 实现页面头部

通过设置好 HTML 代码的层标记且与之关联的选择符后，需要用 CSS 来控制这些标记，要注意的是在 DIV 中设置 ID 选择符时，需要在 CSS 代码中用“#”开头，且后面接的是该名称，表示 ID 选择符。同理，如果是类选择符时，同样在 CSS 代码中把该名称加上，但类选择符名前用“.”表示，后面的 CSS 代码都用大括号包含起来。

本节主要设置网页的整体页面和头部样式的控制。所谓整体网页控制指的是对<body></body>标记进行控制，表示对网页全局布局进行设置。这种表示方法在 CSS 中叫做类型选择符（与类选择符不同）。除了 body 外还有 p、img 等标记类型。本实例中先清除 body、img 标记的边框，然后设置头部的样式，如下代码所示。

```

/*设置网页的全局*/
01 html,body{
02     margin:0;
03     padding:0;
04     height:100%;

```

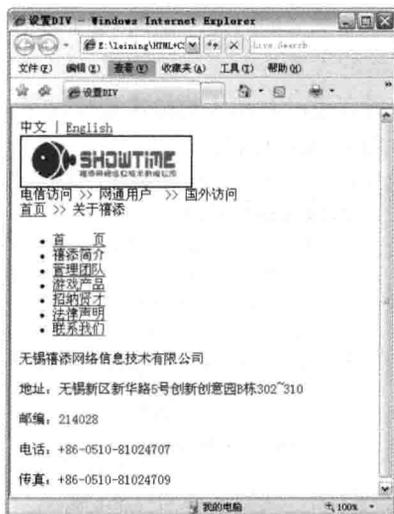


图 22-2 设置 DIV 布局

```

05     /*设置字体类型 */
06     font-family:"Times New Roman";
07     font-size:12px;
08     background:url(images/bg3.jpg);
09     }
10     /*设置头部容器层*/
11     #header
12     {
13         width:776px;
14         margin:auto;
15         position:relative;
16         height:15%;
17         background:#ffffff;
18         border-left:1px solid #ffffff;
19         border-right:1px solid #ffffff;
20     }
21     /**设置图片标记/
22     img
23     {
24         border:0;
25     }
26     /*设置层LOGO*/
27     #logo
28     {
29         margin-top:10px;
30         height:85%;
31     }
32     #banner
33     {
34         width:100%;
35         padding:2px 0px;
36         position:absolute;
37         bottom:0px;
38         /*文本右对齐*/
39         text-align:right;
40         color:#fff;
41         background:#666;
42         border-bottom:1px solid #ffffff;
43         border-top:1px solid #ffffff;
44     }
45     .showright
46     {
47         margin:10px 10px 0px 0px;
48         float:right;
49     }

```

【代码解析】在上述代码中，第 3~12 行只对头部的标记进行设置，通过对头部的设置，头部控制的效果表现出来了。在第 1~9 行代码对全局 body 控制时，字体大小表示为 12 像素，引用一张图片作为页面的背景图像，设置内边距和外边距为 0，头部变成白色背景，且中文与英文的文本都漂移到页面的右边，对“电信访问”、“网通用户”等文本也出现灰颜色背景且文本右对齐，效果如图 22-3 所示。

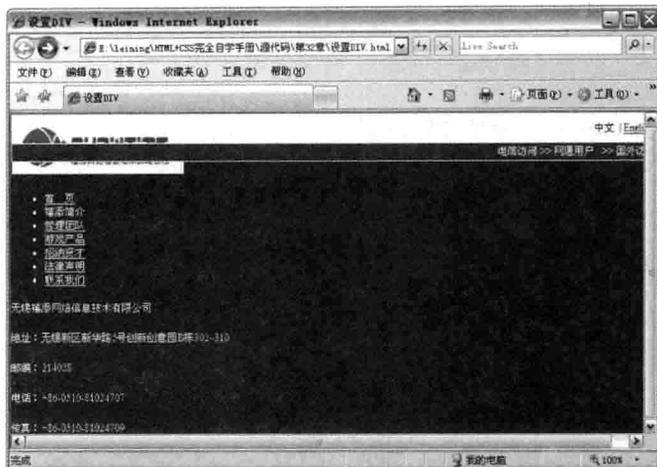


图 22-3 设置头部样式

说明

以上内容是对页面头部进行控制，整个宽度比较宽，具体页面请浏览光盘中源代码\第22章页面内容。

22.2.3 用 CSS 实现页面体

通过设置头部样式可以控制网页头部效果。本节来控制网页体的效果，即中间部分的效果。在 HTML 代码中，设置了 DIV 的标记，且列出各模块的内容，而在右边的层是空内容，只有标记存在，下面对网页的中间部分 CSS 代码进行控制。为了让中间部分与头部区别开来，通过下面的代码来实现。

```

/*CSS代码：设置内容区的样式*/
01  #content
02  {
03      /*设置宽度、高度和对齐方式*/
04      width:778px;
05      margin:auto;
06      height:70%;
07  }
08  /**页脚样式*/
09  #ender
10  {
11      width:776px;
12      margin:auto;
13      height:14%;
14      border:1px solid #ffffff;
15  }
16      /*设置页脚中的标记p*/
17  #ender p
18  {
19      margin:2px 0px 0px 10px;
20      color:#666666;
21  }
22      /*设置中间区域的左边样式*

```

```
23 #left
24 {
25     width:175px;
26     height:100%;
27     float:left;
28     /*设置边框信息*/
29     border-left:1px solid #ffffff;
30     border-right:1px solid #ffffff;
31     background:#e6e6e6;
32 }
33 /*设置中间区域的右边样式*/
34 #right
35 {
36     width:600px;
37     height:100%;
38     float:right;
39     background:#cccccc;
40     border-right:1px solid #ffffff;
41 }
42 #rightbanner
43 {
44     width:120px;
45     float:left;
46 }
47 /*中间部分右边内容显示*/
48 #rightbox
49 {
50     width:480px;
51     height:100%;
52     float:right;
53     padding:10px;
54     background:#cfcfcf;
55 }
56 /*不同的层设置不同的样式*/
57 .enderbox1
58 {
59     width:175px;
60     height:100%;
61     float:left;
62     background:url(images/bg12.jpg);
63     border-right:1px solid #ffffff;
64 }
65 .enderbox2
66 {
67     width:600px;
68     height:57%;
69     float:right;
70     background:#dadada;
71 }
72 .enderbox3
73 {
74     width:600px;
75     height:42%;
```

```
76     float:right;
77     background:#acacac;
78     border-top:1px solid #ffffff;
79 }
80     /*设置页脚中的列表层*/
81 #menu
82 {
83     margin:0px 10px 10px;
84     padding:0px 5px 5px 5px;
85     border:2px solid #ffffff;
86     border-top:0px;
87     background:#aleffb;
88 }
89     /*无序列表的UL设置*/
90 #menu ul
91 {
92     margin:0;
93     padding:0;
94     /*边框设置*/
95     border:medium none;
96     /*去掉列表中的前面标记*/
97     list-style-type:none;
98 }
99     /*列表中的各项*/
100 #menu li
101 {
102     /*设置边框颜色和字体大小*/
103     border-top:1px solid #fff;
104     background:#08aeba;
105     font-size:14px;
106     font-weight:800;
107 }
108     /*设置表项中的链接样式*/
109 #menu li a
110 {
111     /*设置链接区域为链接块*/
112     display:block;
113     /*设置字体类型*/
114     font-family:Times New Roman,ARIAL,Courier;
115     font-weight:blod;
116     padding:10px 0px 5px 31px;
117     width:110px;
118     color:#020202;
119     /*背景图片*/
120     background:url(images/dot.gif) transparent no-repeat 8px 14px;
121     text-decoration:none;
122 }
123     /*当处于链接时的状态*/
124 #menu li a:hover
125 {
126     background:#0d5472;
127 }
128 .current
```

```

129     {
130         padding:7px 10px;
131         background:#0d5472;
132         text-align:right;
133         color:white;
134     }

```

【代码解析】通过上面的 CSS 代码控制，表示 HTML 标记用 CSS 的修饰，表现出从无序显示到有条理地显示，从页面头部中可以看到，LOGO 的图标出现在左边，而且与页面头部层的背景相同，但整个网页的背景为红色，这显示出了 LOGO 与头部的关系，同时移动到右边的文本中，链接到不同的网页版本，在布局中使体现出页头的内容对称性，使页面更美观。

页面中部的布局，先从左边对 DIV 分析，左边的层表现白色的背景（第 31 行代码），而各链接块的颜色与该页整体颜色深度不同，从颜色上很容易区别。在页面体中，左边区域的文本向右显示时，表示正处于该模块在右边区域显示状态，其他模块向左显示，当鼠标置于文本上时，会出现手形，因为是处于链接状态。

网页中间部分的右边区域，分成两个不同颜色的部分。在第一部分中，在左栏区域中间出现一个空层；在另一部分中，用来显示“首页>>关于禧添”，该位置是页面将要显示内容的区域。该区域与中间部分的背景颜色有点不一样，中间部分的颜色更深，也可以从颜色上区分不同的区域。

同时对于页脚的布局，左边是用中国风的图案表示，因正处于过中国年时的站点都喜欢带上过年的喜庆，这也是本实例用红色背景的原因。在页脚部分的右边，显示的是企业的信息，如地址与联系电话等。国内网站都需要到信息产业部的网站上备案才可以放到互联网上，所以备案号也需要在页脚出现并链接指向信息产业部备案网站。

本实例的风格除了过年喜庆的红色外，还以白色和灰色为主，且灰色在不同的层中深度不同，使网页中每个部分看起来有所不同，效果如图 22-4 所示。

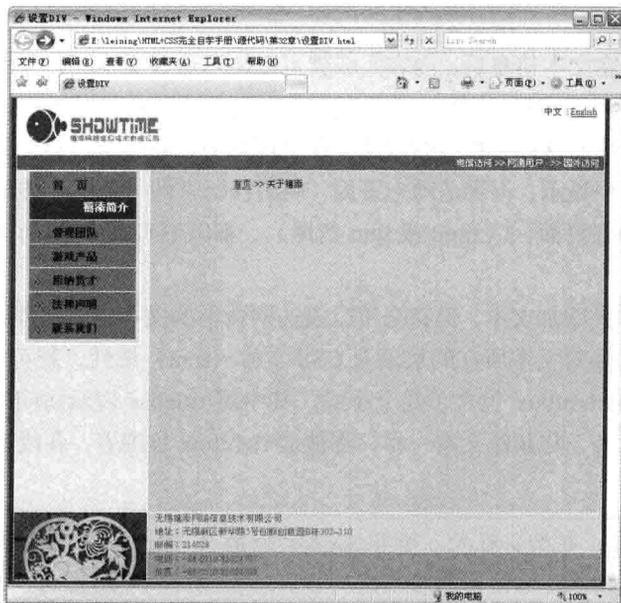


图 22-4 用 CSS 实现标记

22.3 添加文本等信息

在做好层的布局和 CSS 代码后，整个页面框架就形成了，如头部信息的 LOGO 和版本链接都在页面头部，中间部分的左边显示出链接的效果，当选中这些链接时，会在右边区域显示内容。

网站内容显示区的代码如下：

```
<!--HTML中的内容显示区域-->
01 <div id="rightbox">
02     <div class="titlebar"><a href="index.htm">首页</a>&nbsp;&nbsp;>>&nbsp;&nbsp;>关于禧添</div>
03     <p>
04         2008年3月5日，一群专注于互联网互动娱乐内容的年青人汇聚在无锡，将各自的梦想合而为一，再炼成
05 龙骨、铸成蒙皮、锻成甲板、编成风帆，造出一艘源于梦想承载梦想的船，
06 名为“禧添”！
07     </p>
08     <p>
09         无锡禧添（SHOWTIME）网络信息技术有限公司，开始启航！
10     </p>
11     <p>
12         禧添公司是无锡第一家由无锡人自己创办的专业互联网互动娱乐公司，在成立时即得到了
13 无锡市政府相关部门和新区信软办的大力支持。公司注册资本100万，主要从事互联网互动娱
14 乐软件的开发和运营，以及互联网增值服务和人才培养等业务。
15     </p>
16     <p>
17         目前，禧添公司落户在无锡新区创新创意园区，各项业务正在迅速开展中。在未来，禧添将秉持“激情、
18 专心、细致”的方针，以最大的努力，向中国广大的互联网用户提供最优质的
19 服务。
20     </p>
21 </div>
```

【代码解析】添加完文本信息后，内容区域就可以显示这些文本了。当然，在实际的企业网站中，并不是用编辑器打开 HTML 文件，再来修改这些信息来插入和更新这些内容。而是在管理界面中，通过表单的形式填写这些资料，然后保存在数据库中，最后通过网页的浏览，调用数据中内容后，这些内容在页面中显示，不管是动态页面（用编程语言调用数据库传给页面显示）还是静态页面（本书讲的都是静态页面，以.html 或.htm 结尾），都以 HTML 标记为基础，换言之，都建立在静态页面上。

本页面是以手工方式添加文本，但完全可以表达网页中的文本信息操作，如设置字体大小、类型或首行缩进等。只需要对文本所在的层设置 CSS 字体（font）属性，就可以对字体控制。

在以上代码中，层 rightbox 包含了几个小层，其中层 titlebar 表示访问站点的文本位置（第 2 行代码），文本向左对齐，与其他文本一样，都被层 rightbox 包围着，各段落之间都有间隔，效果如图 22-5 所示。

对字体的设置，有很多形式，如用，是HTML标记中的一种，而通过CSS对层设置font属性也可以达到相应效果，但W3C标准采用CSS方式，因此应尽量用CSS属性来控制这些文本。

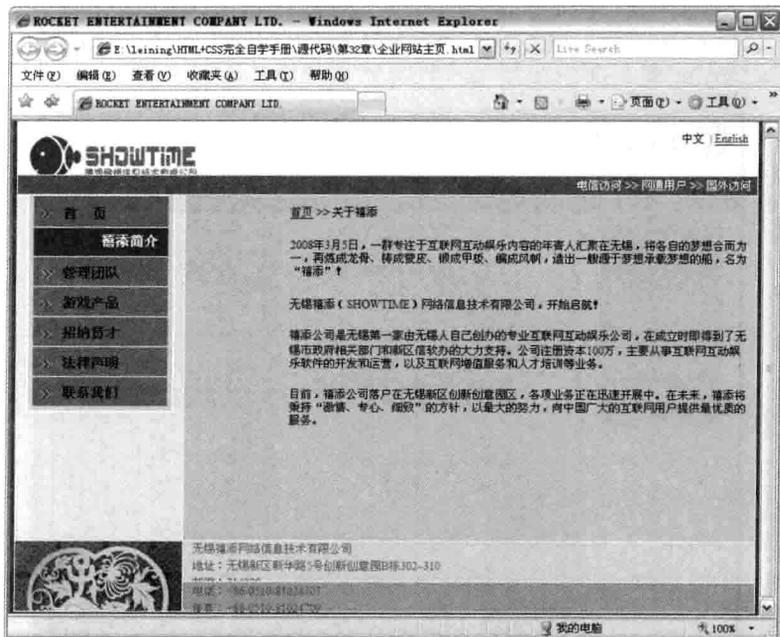


图 22-5 添加文本信息

22.4 调整全页面和浏览器兼容

网页最后的效果是给用户浏览，而用户分布在全球不同的位置，使用的浏览器也不同，但浏览器市场上用 IE 和 FF 非常多，正常条件下，只兼容这两种就行了，所以在完成网页布局时，记得在 FF（很多网页设计师在 IE 中运行，因为操作系统默认安装了 IE）下运行一次，对不合格的层进行调整，使在 IE 下的效果与 FF 中的效果差减少到最小。

调整网页效果的过程是一项重复的工作，如设置一个层是否与上面的层对齐，因重新设置了 margin 或 padding 值，使整个层宽度变化了，那么需要重新设置层的 width 值等。有的页面部分属性在 IE 下运行很好，而到 FF 下就不能正常运行，这些都应当属于浏览器兼容问题。

本实例介绍的页面，经在 IE 中调试多次后，也需要到 FF 中运行，因 FF 与 IE 对 CSS 的支持不同，所以在运行时会出现不同的效果，这就要求设计者去调试这些页面。设置的方法如下。

- 更换 CSS 属性；
- 对属性值修改；
- 换不同浏览器。

有更多的方法可以调整网页中不正常的页面。只有使页面在不同的浏览器中运行正常，才可以合成一个网站。对网站页面整合好后，再放到 Web 服务器上，供用户浏览这些页面，那么网站就完成了，效果如图 22-6 所示。

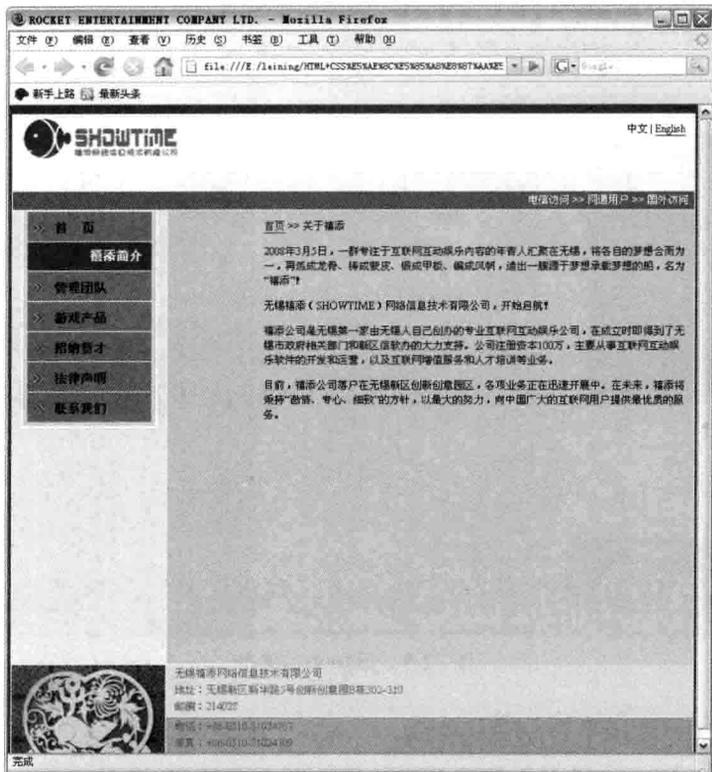


图 22-6 完成的网站

22.5 其他二级页面模块制作

本实例只演示了企业网站的模板页制作，通过页面的制作步骤，清楚了网页的各个组成部分，即每个网页都存在页头、页体和页脚，根据不同应用设置不同的页面风格。本实例是企业类网站，没有花哨的界面，没有太多的功能。

企业网站的页面大致统一为一种风格，所以通过本实例的演示，其他网页可以此为模板，对于新的页面，在“模板页”的基础上添加或修改内容信息，换不同的标题，就差不多完成了该页面，做完几个页面整合后就是一个企业网站了。页面与页面之间的相互链接才使用户从主页进来，更快地浏览其他信息，使网站的可读性更高。

二级页面完成后，通过整合成为一个新的企业网站，测试没问题后，上传到互联网上的服务器，通过域名绑定服务器的 IP 地址，在浏览器地址栏中输入绑定好的服务器的域名，就可以正常访问页面了。

22.6 小结

本章主要介绍企业网站的布局，实例演示的是企业网站的模板页，先对网站进行规划、构思，

得到网站的需求，然后在 HTML 中设置好 DIV 和关联 CSS 的选择符，通过外部链接的方法，设置好 CSS 代码。之前先用层表示区域，即设置好层的宽度和高度。为了区分层大小，给层设置边框并用颜色显示。

在调节画面细节时，在完成 HTML 代码和内容的条件下，只对 CSS 代码设置即可对整个页面效果进行控制。因浏览器对 CSS 属性支持不同，所以在页面完成前还要在不同的浏览器上运行、调试，直到最终效果在不同浏览器上达到正常效果即可。对于其他的页面，只需复制这些代码，再根据需求修改主题和正文文本，在功能链接模块上做好链接，最后加上新页面的内容，页面就新建好了，这样效率也能提高不少。

第 23 章 博客主要页面布局

现在的博客网站流行已超过以前的 BBS，一个人拥有几个博客空间早已是正常的事情，如 QQ 空间、51 空间等。特别是学生，都喜欢个性化的界面，所以每个博客的定位人群都不一样，如 QQ、163 等博客，年轻人用得比较多，个性化界面的模板，网上到处都有。在博客构架方面，代码与界面模板都是分离的，这样用户可以自由地更换自己喜欢的博客界面风格，而且操作简单。

本章介绍如何布局博客主要页面，如个人博客的主页。主页往往包含博客中所有模块的内容，也是博客总信息的汇总。本章是实践部分，从布局代码中体现出布局的要点。学完本章，会深深地体会到博客主页充分利用了以前学习过的知识，除了学习过的 HTML 和 CSS 方面的知识外，还会涉及本书没有介绍过的知识，如 JavaScript 脚本（js 文件）和一个优秀的 JS 框架——jQuery。本章的内容用 jQuery 做出隐藏显示的动画效果，利用这个框架对网页前台设计师的帮助非常大。

学完本章，主要了解的内容如下：

- 博客网站主页规划；
- 用 DIV 布局博客主页的层；
- CSS 实现博客各模块；
- 给主页添加图片；
- 给主页添加文本文字；
- 从全局角度调整整个页面；
- 其他页面布局。

23.1 博客网站主页规划

博客网站的个人主页都包含有首页、相册、个人信息、日志、好友、好友留言和评论等功能模块，即在主页中显示所有模块最新内容。当然，显示的内容也是有规律的，如好友访问该博客后，是按最近访问的顺序显示，同理，给博客主人（简称博主）留言的也是从时间上往后推的排列方式。

除了如何设置各个模块的显示问题外，还得对模块位置进行设置，如相册放在博客首页的中间位置，个人信息放左上角，留言内容放置底部等，这样的布局 and 用户的习惯相关。本实例介绍如何布局博客主页。对主页的规划首先要分析好主页的需求，如本博客主要用户群是哪个年龄阶段的，还要确定内容的模块（所谓模块，是指个人信息和相册、日志等内容区域），最后依照这些需求设置网站的效果图。

下面关于页面的规划，包括分析主页需求、确定内容模块和画出效果图这三个部分。

23.1.1 分析主页需求

主页的需求即该博客主页面主要对哪个领域或行业开放的,即博客面向社会中的哪部分人。如技术性网站博客园以文章、留言为主,因为博客园定位的是技术性,主要用户都是由于技术需要来博客中探讨问题,这时就不需要相册和关于博主的星座等信息,往往技术人员在撰写技术文章时,文章功能和留言功能就够用了,文章功能主要用来写技术类文字(主要写技术性文字),当网友访问该文章时可能会通过留言的功能来评论文章的好坏或错误部分等。

同理,如果定位于学生、青少年这些年轻的群体,对相册、留言互动、界面外部个性等方面的花样要求就高得多,网易的博客模板就有 1000 多个不同的版本,QQ 空间的模板更多。当然也得到了喜欢“非主流”人们的喜爱,这更是博客商的准确定位。

当分析博客的主页需求时,也要分析该博客主要针对哪些人群,他们有什么样的爱好或习惯。对目标群进行准确的分析,才能对博客网站有更准定位,这样的博客才会有用户注册。本章实例需求以年轻人为主的游戏类博客,根据这种人群的习惯特征在博客上设置不同的模块。

23.1.2 确定内容模块

当分析好博客主页需求后,了解了博客的需求,同时也明白了让用户注册博客的目的,对内容模块就能做出准确划分。

本实例做一个游戏类博客,分析玩游戏的人群大部分是年轻人,因这部分人的生活环境影响,他们喜欢潮流的、有个性的并带有游戏元素等特征的博客界面,同时再对模块的设置做如下分析:

(1)“个人信息”可列出博客主人的网名、性别、年龄、所喜欢的游戏(因为一起玩同一个游戏的人有共同点,他们之间也会交流),即个人信息用简单的文本描绘更多的信息。

(2)“我的好友”模块用来显示一起玩游戏或在网上认识的网友的博客地址,加在好友列表中。

(3)“最近访客”模块中,可以显示到目前为止的这段时间内,有哪些用户访问过这个博客。博主看到他们的朋友访问自己博客,是一件很高兴的事情。

(4)当好友访问博客时,顺便还留言了,这样好友与博客主人之间互动起来,使他们沟通的距离又更近了一步,同时也体现出该功能的用途。

(5)好友过来,一定要有些吸引的东西给他们看,如“我的日志”上写了最近对某个游戏的玩法,当好友访问这篇文章时,感到收获不少,也吸引了他们在下次内容更新时回访。

(6)与“我的日志”模块一样,相片的吸引效果更直接,在当多年没见的同学在博客上看到“我的相册”上的照片时,心里有说不出的激动。

这些模块根据需求来定,日志的模块差不多也是这几种,并不一定是每种模块都加到博客就是好的。

23.1.3 画出效果图

与做网站一样,博客也需要美术的支持,通过美术人员的设计,把博客需求和模块转化成效果图,通过网页效果图来看整体网页。如果效果图做得好,那么用代码(指 HTML 代码写出来的语言,动态网页也离不开 HTML)和通过美术人员切好的图来实现 HTML 标记,让最终效果在网页

中显示出来。

有了效果图，就有了博客的最终目标，用 HTML 代码来实现。本章实例中以 DIV+CSS 方式布局博客主页，因本书不是讲解动态网页内容，所以不会涉及动态页面的内容（动态页面由 HTML 组成，网页源代码显示出来的全是 HTML 标记）。

实例最终效果如图 23-1 所示，从博客主页效果图中切出来的图片放在专门放置图片的文件夹中，命名为 images。

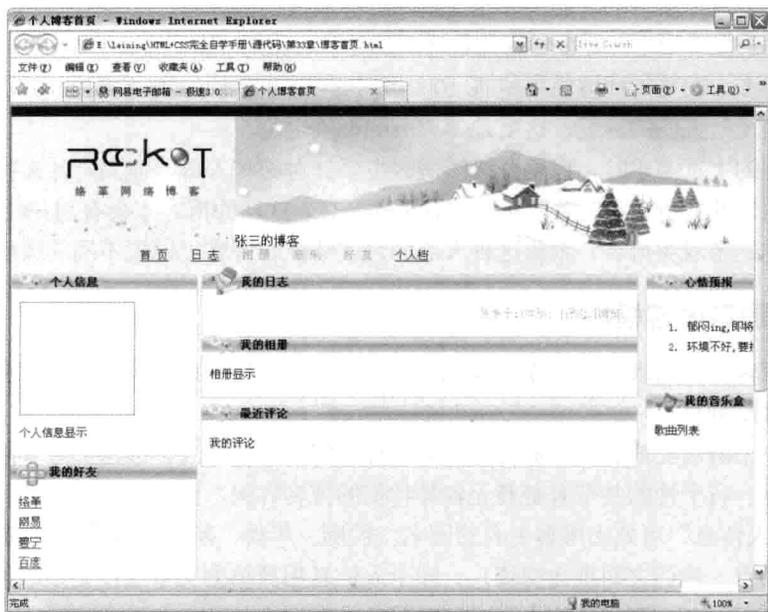


图 23-1 博客页面效果图

23.2 用 DIV 布局框图

本节通过一个空网页的编写，慢慢实现博客主页面的布局。博客主页面分成页眉、内容模块和页脚。在博客中，页眉用来标明博客的名称和对博客的描述，往往用背景图作为名称的背景，该图可以体现博客的风格。在页眉下面有导航栏，用来链接主要模块的地址，如相册、日志等。具体的内容因模块不同放置在主页面中的位置也不同，如个人信息放在中部左上角，相片封面在中间显示，留言在中部的下面显示。最后在页脚声明博客版权、ICP 备案号等信息。

23.2.1 页眉

页眉是博客主页的头部，也是显示博客主题的位置，绝大部分博客都在这个位置用背景和几行简要文字叙述博客属性，如“张三的博客”为第一行，主要是博客名称，第二行为对博客的一种描述，如张三这样写——学习 HTML+CSS 的技术博客。这样从页眉就可以看出，这博客是张三的，而且他喜欢写 HTML，通常用 CSS 来布局。

现在用 DIV 来描述头部的层。分析一下，头部由两块组成。首先，在头部上面用一条黑色的框显示，然后下面用一张图片做 DIV 层的背景，最后放置两行文本，即博客名和对博客的描述。


```

<!--左边栏-->
01     <div class="myblogleft">
02         <div class="mybox">
03             <div class="myblogtitle">
04                 个人信息
05             </div>
06             <div id="index3">
07                 <div class="myblogcon">
08                     <div class="myphoto"></div>
09                 </div>
10             </div>
11         </div>
12         <div class="mybox">
13             <div class="myblogtitle" >
14                 我的好友
15             </div>
16             <div id="index4">
17                 <div class="myblogcon">
18
19                 </div>
20             </div>
21         </div>
22         <div class="mybox">
23             <div class="myblogtitle">
24                 最近访客
25             </div>
26             <div id="index5">
27                 <div class="myblogcon">
28                 </div>
29             </div>
30         </div>
31         <div class="mybox">
32             <div class="myblogtitle">
33                 最新留言
34             </div>
35             <div id="index6">
36                 <div class="myblogcon">
37                     <!--设置内容 -->
38                 </div>
39             </div>
40         </div>
41     </div>

```

【代码解析】以上代码主要在左边栏设置了4个层：第1~11行代码为第1层，显示内容为“个人信息”；第12~21行代码为第2层，显示“我的好友”信息；第22~30行代码为第3层，是“最近访客”信息；第31~41行代码为第4层，显示“最新留言”信息。

与向左漂移相同的是，中间栏的表现也需要设置对象的float为left，和上面代码一样，表示向左漂移，代码如下所示。

```

01     <!--中间栏-->
02     <div class="myblogcenter">

```

```

03         <div class="mybox">
04             <div class="myblogtitle">
05                 我的日志
06             </div>
07         <div id="index7">
08             <div class="myblogcon">
09                 <!--设置内容区域 -->
10                 <div class="logtitle">
11                 </div>
12                 <div class="floatright">
13                 </div><br />
14             </div>
15         </div>
16     </div>
17     <div class="mybox">
18         <div class="myblogtitle">
19             我的相册
20         </div>
21         <div id="index8">
22             <div class="myblogcon"></div>
23         </div>
24     </div>
25     <div class="mybox">
26         <div class="myblogtitle">
27             最近评论
28         </div>
29         <div id="index9">
30             <div class="myblogcon">
31             </div>
32         </div>
33     </div>
34 </div>
35 </div>
36 </div>

```

【代码解析】以上代码主要设置中间边栏的 3 层：第 1~16 行代码为第 1 层，显示“我的日志”信息；第 17~24 行代码为第 2 层，显示“我的相册”信息；第 26~33 行代码为第 3 层，显示“最近评论”信息。

注意

在设置 float 时，要确定对象的宽度，如上面代码两个层都向左漂移，即需要确定这两个层的宽度。

对模块的设置，先用容器层包含两小层：第一层表示模块主题，而第二层用来表现该模块的内容，如图片或文字等，其中的 ID 选择符主要是为了控制该层的显示与隐藏特效。模块采用以下代码形式。

```

01 <div class="mybox">
02     <div class="myblogtitle">
03         最近评论
04     </div>
05     <div id="index1">
06         <div class="myblogcon">

```

```

07         </div>
08     </div>
09 </div>

```

【代码解析】在上述代码中，层 mybox 中作为容器层包围着 myblogtitle、myblogcon 等类选择符和一个 ID 为 index1 的 ID 选择符（第 1~9 行代码），myblogtitle 层表示模块主题层（第 2~4 行代码），同理，myblogcon 表示内容的层（第 6~7 行代码），而 index1 层目的是隐藏内容层。同时，这些模块又被左边、中间和右边分成 3 部分，其他层的设置代码如下。

```

<!--右边栏-->
01 <div class="myblogright">
02     <div class="myblogtitle">
03         </div>
04 </div>
05 <!--左边栏-->
06 <div class="myblogleft">
07     <div class="myblogtitle">
08         </div>
09 </div>
10 <!--中间栏-->
11 <div class="myblogcenter">
12     <div class="myblogtitle">
13         </div>
14 </div>

```

【代码解析】以上代码就是将网页的中间部分划分成左、中、右三栏。代码第 1~4 行是右边栏的定义；代码第 5~9 行是左边栏的定义；代码第 11~14 行是中间栏的定义。

注意

上面的代码都是在一个网页中的部分代码，本实例的HTML代码按区域列出，CSS代码按各模块来表示。

23.2.3 页脚

页脚的部分是整个网页的结束符，一般都放一些公司相关信息的链接，以及版权、备案 ICP 证号等。

在本项目中的页脚，用无序列表表示各个链接的文字，然后再做上链接，每个标记对都有文本和链接。通过两个层来表示，footer 表示页脚的容器层，aboutus 里面包含着无序列表。其代码如下。

```

01 <div class="footer">
02     <div class="aboutus">
03         <ul>
04             <li> <a href=" ../aboutus.html">关于我们</a></li>
05             <li> <a href=" ../reg.aspx">注册会员</a></li>
06             <li> <a href=" ../pay/index.aspx">在线充值</a></li>
07             <li> <a href=" ../help/index.aspx">游戏客服</a></li>
08             <li> <a href=" ../map.html">网站地图</a></li>
09             <li> <a href=" ../law.html">法律声明</a></li>
10             <li> <a href=" ../zhaopin.html">招贤纳士</a></li>
11             <li> <a href=" ../cooperate.html">合作伙伴</a></li>
12             <li> <a href=" ../connectus.html">联系我们</a></li>

```

```

13         </ul>
14     </div>
15     <p><br />
16     实例作品</p>
17     <p>版权所有</p>
18 </div>

```

【代码解析】第 4~12 行代码通过<a href>为每个序列做了链接。

至此所有的层设置都完成了，这些代码都只出现在 HTML 代码的<body></body>中，即没有出现 CSS 代码的控制，其效果还是原始状态。为了让这些标记中的内容变成想要的效果，下面需要 CSS 来控制，接下来是以上代码的样式代码，其中，包含<div>中的 class 或 id 是对应的代码。

注意 以上代码都在<body></body>中。

23.3 CSS 实现框图

在 23.2 节中设置好了 DIV，并命名好 CLASS 和 ID 的选择符，接下来设置 CSS 的样式。CSS 对网页是非常重要的，所以如何用 CSS 实现效果也是需要技巧的。同时，写好 CSS 后也要对 CSS 代码进行优化，这样网页才有更好的效果。

23.3.1 实现内容模块

上面的层中设置了 ID 和 CLASS 选择符，下面是全部的样式表，对于比较难理解的样式做了注解（CSS 代码用/* */注解，而 HTML 标记用<!-- -->注解），由于存在很多 CSS 代码的情况下，需要单独新建一个样式表，然后保存为以“.css”为扩展名的文件，使用时要链接到网页中即可，博客主页的样式内容代码如下。

代码 23-1 style.css

```

/*整个网页的样式设置*/
01 body
02 {
03     margin:0px;
04     padding:0px;
05     background:#e1eff1;
06 }
07 /*头部层样式*/
08 .blogheader
09 {
10     width:998px;
11     height:161px;
12     position:relative;
13     background:url("images/headerbg.gif");
14     margin:0 auto;
15 }
16 /*导航栏样式设置*/
17 .blogheaderindex
18 {
19     width:998px;

```

```
20     height:161px;
21     position:relative;
22     margin:0 auto;
23 }
24 .myblogbanner
25 {
26     width:600px;
27     margin:0 auto;
28     font-size:14px;
29     font-family:宋体;
30     color:Orange;
31     padding:5px 20px;
32 }
33 /*设置博客个人信息相片*/
34 .myphoto
35 {
36     width:140px;
37     height:140px;
38     margin:10px 0px 10px 40px;
39     border:1px solid #666;
40 }
41 /*设置小相片*/
42 .mysmallphoto
43 {
44     width:60px;
45     height:60px;
46     margin:5px;
47     float:left;
48     border:1px solid #ccc;
49 }
50 .titlebox
51 {
52     width:78px;
53     height:41px;
54     margin-left:3px;
55     float:left;
56     font-family:黑体;
57     line-height:22px;
58     font-size:12px;
59     text-align:center;
60     background:url("images/titlebg2.gif");
61 }
62 .titleboxblack
63 {
64     width:78px;
65     height:41px;
66     margin-left:3px;
67     font-family:黑体;
68     line-height:22px;
69     float:left;
70     color:White;
```

```
71     font-size:12px;
72     text-align:center;
73     background:url("images/titlebg1.gif");
74
75 }
76 .blogbanner
77 {
78     width:680px;
79     position:absolute;
80     right:10px;
81     bottom:5px;
82 }
83 .blogbody
84 {
85     width:998px;
86     overflow:hidden;
87     clear:both;
88     background:#fff;
89     margin:0 auto;
90 }
91 /*首页左边flash*/
92 .flashbox
93 {
94     width:300px;
95     height:200px;
96     position:relative;
97     margin:5px 0px 0px 5px;
98     float:left;
99 }
100 .flash1
101 {
102     width:13px;
103     height:42px;
104     position:absolute;
105     left:0px;
106     top:0px;
107     background:url("images/flash1.gif");
108 }
109 .flash2
110 {
111     width:32px;
112     height:48px;
113     position:absolute;
114     right:0px;
115     top:0px;
116     background:url("images/flash2.gif");
117 }
118 .flash3
119 {
120     width:12px;
```

```
121     height:34px;
122     position:absolute;
123     left:0px;
124     bottom:0px;
125     background:url("images/flash3.gif");
126 }
127 .flash4
128 {
129     width:17px;
130     height:30px;
131     position:absolute;
132     right:10px;
133     bottom:0px;
134     background:url("images/flash4.gif");
135 }
136 .flashbody
137 {
138
139     height:190px;
140     padding:10px 0px 0px 15px;
141     background:url("images/18.gif") repeat-x;
142 }
```

【代码解析】第 1~6 行代码为类型 body 的类型选择符，其他都是类选择符。

下面的样式表是首页右边的导航条，代码如下。

```
/*首页右边的导航条*/
01 .titlebanner
02 {
03     width:675px;
04     margin:5px 0px 0px 15px;
05     float:left;
06     background:url("images/titlebannercenter.gif") repeat-x;
07 }
08 .keytitle
09 {
10     width:672px;
11     float:left;
12     margin:5px 0px 0px 15px;
13     border:1px solid #ffccb4;
14 }
15 .bgup
16 {
17     height:12px;
18     background:#000000;
19 }
20 .bgcolor
21 {
22     clear:both;
23     height:4px;
24     background:#333;
25 }
```

```
26 .bgcolored
27 {
28     width:18%;
29     height:4px;
30     float:left;
31     background:#cc0000;
32 }
33 .bgcolorblue
34 {
35     width:15%;
36     height:4px;
37     float:left;
38     background:#669900;
39 }
40 .bgcolorgreen
41 {
42     width:30%;
43     float:left;
44     height:4px;
45     background:#009999;
46 }
47 .bgcolorgray
48 {
49     width:12%;
50     float:left;
51     height:2px;
52     background:#333;
53 }
```

【代码解析】 以上代码为右边导航条的样式，每个样式都有自己的背景色，而且都通过 width,height 设置了范围，而且基本都是左漂移。

下面的代码是对博客的功能进行设置。

```
01 .blogcon
02 {
03     margin:5px;
04     font-size:14px;
05     clear:both;
06 }
07 .blogright
08 {
09     width:185px;
10     height:200px;
11     margin:5px 5px 0px 0px;
12     float:right;
13     border:1px solid #ffccb4;
14 }
15 .blogleft
16 {
17     width:250px;
18     margin:5px 0px 0px 5px;
19     float:left;
```

```

20     border:1px solid #ffccb4;
21   }
22   .blogcenter
23   {
24     width:546px;
25     height:200px;
26     margin:5px 0px 0px 5px;
27     float:left;
28     border:1px solid #ffccb4;
29   }

```

【代码解析】以上代码主要为中间栏的内容，每一层都设置了外边距，向左漂移。

以上 CSS 代码实现了博客的主要功能结构，通过 CSS 设置可以控制 HTML 标记的样式。当内容模块实现后，就要来实现页面效果了，下一节介绍对页面的操作。

23.3.2 实现整页效果

以上演示的是博客系统通过 HTML 标记和 CSS 代码的结合，使 HTML 代码与 CSS 样式关联起来，用网页表现博客的效果。

因 CSS 代码保存为“boss.css”，为单独的样式文件，但需要链接到网页中才可以显示出效果，HTML 设置的层标记也在上面的代码中，且 CSS 也保存了文件，那么需要用前面学习的<link>标记把样式文件引进来，设置如下。

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

这条语句指导 HTML 标记与 CSS 文件联系起来，其中<link>表示链接的意思，而 type 表示类型，该类型指的是样式文本，即“text/css”，“rel”是 Relations 的缩写，指关联到一个 stylesheet（样式表单），然后“href”表示样式的链接地址，这样就把一个样式文件链接进来了。当然，要保证样式文件的正确性，而且在 HTML 对象标记中，用 ID 或类设置了对的选择符，这样才可以正常控制标记并显示出来。

关联后的网页代码如下所示。

```

01 <html>
02   <head>
03     <title>个人博客首页</title>
04     <!--表示链接样式文件-->
05     <link type="text/css" rel="stylesheet" href="blog.css">
06   </head>
07   <body>
08     <!--这里放置上面的DIV标记-->
09   </body>
10 </html>

```

【代码解析】在<head></head>中加入<link>标记使样式文件链接到本页面中（第 5 行代码），而且在<body></body>中设置了头部、中间和底部的层标记，一个博客首页的构图就形成了，如图 23-2 所示。

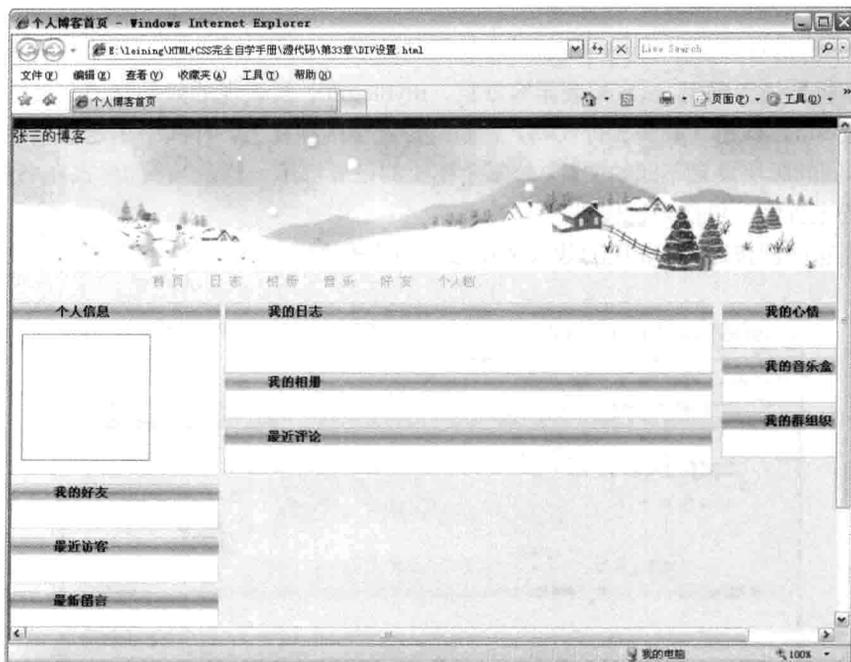


图 23-2 博客首页效果图

23.4 添加图片

博客的首页构图完成后，下面来设置博客的图片，其中的图片都放在 `images` 文件夹下，本实例需要的图片有 LOGO 图和每个功能模块的小图标等。

添加图片，可以直接从 `IMG` 标记中引入，同时更多的小图片用 CSS 设置背景图像，如各功能模块中的图标等。为了将这些图标加入到功能模块的主题条上，需要对原来的 CSS 文件进行设置。

因这些标记只在主页中使用，故可以用内联的方式设置样式，如左边的第一个区域是个人信息，而可以设置其他背景样式，代码如下。

```

01 <div class="myblogtitle">
02 
03 
04 
05 个人信息
06 </div>

```

【代码解析】在上述代码中，使用了类 `myblogtitle`，并在其内部引入了标记 `` 同时设置该标记的样式（第 2~4 行代码），使用其出现在左上角，同时其他的功能模块上，可以设置一个类选择符，相同的样式可以用该类选择符，代码如下。

```

01 <div class="mybox">
02 <div class="myblogtitle">
03 
04 
05 
06 我的相册

```

```
07     </div>
08 </div>
```

【代码解析】该代码用 mybox 表示容器层，myblogtitle 层表现的效果与“个人信息”一样，都在其内设置

当设置完这些图像后，所表现的效果是在每个功能模块上都有一个不同的图像，效果如图 23-3 所示。

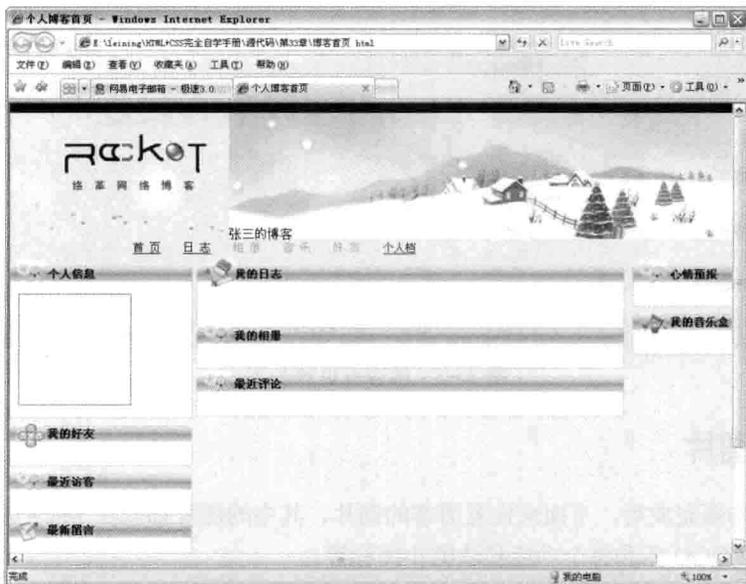


图 23-3 添加图片效果

23.5 加入文字

文本文字在网页中很关键，也是博客中所表现出来的内容，在博客系统中，文本都是从数据库中取出，显示在布局好的页面上，本书只介绍布局方面的知识，能手动地模拟些文本让网页显示出来，以方便初学者在此基础上再做更多的布局，使内容更丰富。

本实例中对文本进行简单的添加，其中功能模块上显示的信息正是其下所要显示的内容，如在左边顶部的“个人信息”模块，先显示博客主人的个性头像，接着显示性别和重要的信息，左边中间的模块是“我的好友”，好友的名字和链接在其下。要注意的是，在博客显示的只是一部分，如需全部显示要再建页面用来显示链接，而且还要分页。

主页的中间部分有“我的日志”、“我的相册”和“最近评论”，其中日志显示博客主人最近发表的文章，相册每次显示一张，而且会自动切换，要实现这种功能需要用到 flash 和 js (JavaScript) 知识，而“最近评论”显示以时间排序的评论。在右边显示的是心情和音乐方面的信息，这些信息都需要从数据库中取出，更新更快，但布局是实现这从数据库中取出内容的基础，效果如图 23-4 所示。

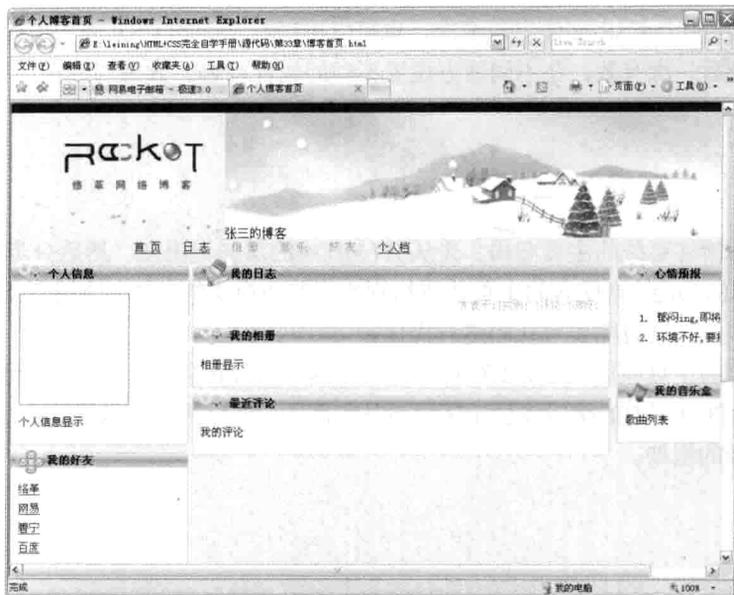


图 23-4 添加文本后效果

23.6 调整全页面

页面布局类似于盖高楼，首先要规范好楼房的各个细节，而网页布局的每个细节也是需要注意的，同时要对不同的环境有所适应，如在 IE 中运行很好的网页不一定会在 FF 中显示正常，那么需要在代码中调整使其能在 FF 中正常显示。

调整文字标题的第一个词组。在每个加粗或文字标题的第一个词组比其他位置的词组对用户的刺激和影响会大很多，看看网页每个标题的第一个词组，还有没有比它更有影响力的词组可以打动用户呢？有些标题的重点关键字在短句的较后位置，可以通过调整句子的表达方法把有影响力的关键字提前，这有利于“震撼”用户，从而提高网站转化率。

最后网页能正常运行了，再接着做下一个页面。

23.7 其他页面布局

其他页面的布局与本实例风格一样，但实现的布局根据需要设置，如相册、日志以及评论的布局页面是不同的，但博客风格还是统一的。

不管博客中的各模块功能如何实现，布局的差别也不会太大，如博客各页面的页头和页脚显示的内容总不会变。所以布局的样式是一样的，通过引入同一个 CSS 文件，来实现不同页面的同一个页头和页脚。

需要注意一下页体中对不同的功能实现，往往这些页面的布局要单独设置，在 CSS 编写中，可以单独放置，同时也可以原有的 CSS 文件中，但每个 CSS 代码不要放置太多代码，即每个网页可以链入多个样式文件。

在布局中，除了可以链入多个样式文件外，还可以使同一个选择符出现多次，如 ID 为 mybox

的 ID 选择符可以在一个 CSS 文件出现多次，也可以出现在不同的 CSS 样式文件中，但对重复设置的属性，只对最后一次有效，在不同地方设置不同的属性内容，效果与放在一个 mybox 的样式中是一样的。

23.8 小结

本章主要讲解博客系统的主页布局。先从 HTML 中的层标记出发，然后分步骤对 CSS 代码列出，且说明其表示的功能。本博客的主页从布局到实现其他模块的功能，使用 CSS 代码比较多（实际项目的代码更多），这时需要对代码进行优化。

做好一个博客的主页面后，还需要对其他各功能模块进行布局，其他页面用到的页头和页脚与主页的一样，同时部分页面的按钮或显示区域也是相同的，所以博客的风格就会统一起来，本实例只演示了博客布局的框架。

第 24 章 面 试 题

虽然从事 HTML+CSS 的人很多，但是在网络上很少看到有关这个方向的面试题。不管是网络编辑、网站美工、网站设计人员，都需要懂一些 HTML+CSS 的知识。本章通过一些常见的面试题，对读者的学习情况进行考核，希望读者能有所收获。

面试题 1 DIV 与 SPAN 的区别

DIV 只是一个分块的标签，它可以将网页分成几个区块。DIV 里面可能包含一个标题，一个段落，也可能包含图片在内的很多元素，甚至 DIV 也可以再包含 DIV。而 SPAN 是行级元素（行内标签），通常情况下它都用来定义一小段文字的样式。它们的另一个区别就是 DIV 会造成换行，而 SPAN 则不会。

面试题 2 HTML 的框架是什么

HTML 的框架就是将页面分成几部分，其中有两个标签实现这种框架。

1. iframe 标签

iframe 标签是成对出现的，以<iframe>开始，</iframe>结束。使用实例如下所示。

```
<iframe src="http://www.sina.com/" width="20%" height="50%">
    这里是iframe标签设置的框架
</iframe>
```

2. frameset 标签

frameset 标签作用是指定一个框架集，用于组织多个框架和嵌套框架集。frameset 元素是 frame 元素的容器。HTML 文档可包含 frameset 元素或 body 元素之一，而不能同时包含两者。使用实例如下所示。

```
<frameset rows="*,*" cols="*,*">
<frame name="top_left" src="top_left.html">
<frame name="top_right" src="">
<frame name="bottom_left" src="">
<frame name="bottom_right" src="">
</frameset>
```

面试题 3 CSS 引入的方式有哪些

CSS 引入的方式有如下三种。

1. 定义内部对象

可以在 HTML 文档的<HTML>和<BODY>标记之间插入一个<style>...</ style >块对象。

2. 内联定义

在对象的标记内使用对象的 style 属性定义适用其的样式表属性。

3. 层叠样式

调用外部样式表，这是最常使用样式信息与网页内容分离的深层运用。

可根据不同需求和页面环境配合使用以上几种方法，个人认为没有限定的写法，只有最优的做法。

面试题 4 前端页面有哪三层构成，各层的作用是什么

前端页面由结构层、表示层和行为层构成。

1. 结构层

结构层 (structural layer) 由 HTML 或 XHTML 之类的标记语言负责创建。那些出现在尖括号里的单词称为标签。例如，<p> 标签表示这是一个文本段。

2. 表示层

表示层 (presentation layer) 由 CSS 负责创建。CSS 对“如何显示有关内容”的问题做出了回答。

3. 行为层

行为层 (behavior layer) 负责回答“内容应该如何对事件做出反应”这一问题。这是 JavaScript 语言和 DOM 主宰的领域。

面试题 5 常见的浏览器内核有哪些

常见的浏览器内核有四种：Trident、Geckos、Presto 和 Webkit。

1. Trident

这是微软开发的一种排版引擎。该内核程序在 1997 年的 IE4 中首次被采用，是微软在 Mosaic 代码的基础之上修改而来的。Trident 实际上是一款开放的内核，其接口内核设计得相当成熟，因此才有许多采用 IE 内核而非 IE 的浏览器涌现（如 Maxthon、The World、TT、GreenBrowser、AvantBrowser 等）。此外，为了方便也有很多人直接简称其为 IE 内核。

2. Geckos

Geckos 是套开放源代码的、以 C++ 编写的网页排版引擎。这个软件原本是由网景通讯公司开发的，Netscape 6 开始采用该内核，后来的 Mozilla Firefox 也采用了该内核。Geckos 的特点是代码完全公开，因此，其可开发程度很高，全世界的程序员都可以为其编写代码，增加功能。Geckos 现在由 Mozilla 基金会维护。

3. Presto

Presto 是一个由 Opera Software 开发的浏览器排版引擎。该内核在 2003 年的 Opera7 中首次被

使用。该款引擎的特点就是渲染速度的优化达到了极致，也是目前公认的网页浏览速度最快的浏览器内核。

4. Webkit

Webkit 是苹果公司自己的内核，也是苹果的 Safari 浏览器使用的内核。Webkit 引擎包含 WebCore 排版引擎及 JavaScriptCore 解析引擎，均是从 KDE 的 KHTML 及 KJS 引擎衍生而来的。它们都是自由软件，在 GPL 条约下授权，同时支持 BSD 系统的开发。所以 Webkit 也是自由软件，同时开放源代码。在安全方面不受 IE、Firefox 的制约，所以 Safari 浏览器在国内还是很安全的。

使用 Trident 内核的浏览器：IE、Maxthon、TT、The World 等。

使用 Geckos 内核的浏览器：Netscape 6 及以上版本的浏览器、FireFox、MozillaSuite/SeaMonkey。

使用 Presto 内核的浏览器：Opera7 及以上版本的浏览器。

使用 Webkit 内核的浏览器：Safari 和 Chrome。

面试题 6 img 元素的 alt text 属性和 title 属性有什么不同

正确地使用 alt text 属性和 title 属性不仅可以提高图片的搜索能力，而且对用户体验也很有帮助。替换文字 (alt text) 是为了给那些不能看到文档中图像的浏览者提供文字说明。title 是对图片的说明和额外补充，如果需要在鼠标经过图片时出现文字提示应该用属性 title。title 属性的优先级高于 alt text。两属性的区别主要如下。

- (1) alt text 和 title 对访问者都很重要 (alt text 对搜索引擎会更重要一些)，所以在定义 img 对象时，最好将 alt text 和 title 属性都写全，保证在各种浏览器中都能正常使用。
- (2) 在 alt text 和 title 中包含关键字，并且二者的内容最好不一样。
- (3) 不要在 alt text 和 title 中堆积关键字，否则可能会导致搜索引擎惩罚。

面试题 7 document.body 和 document.documentElement 有什么不同

document.body 是 DOM 中 Document 对象里的 body 节点，document.documentElement 是文档对象根节点 (html) 的引用。

面试题 8 XHTML 和 DHTML 有什么区别

DHTML 与 XHTML 是一种语言的不同阶段，有点类似于文言文和白话文之间的关系。在 XHTML 中标记名称、属性名称必须小写，标记必须严格嵌套，必须封闭。DHTML 是一种制作网页的概念，实际上没有一个组织或机构推出过所谓的 DHTML 标准或技术规范之类的。DHTML 不是一种技术、标准或规范，DHTML 只是一种将目前已有的网页技术、语言标准整合运用，制作出在下载后仍然能实时变换页面元素效果的网页的设计概念。

面试题 9 写出实现效果所需的 CSS 代码

实现以下效果，一个 class 为 f 的 DIV 中包含 4 个子 DIV。父 DIV 宽 460px，高 100px，子 DIV 宽高都为 100px，子 DIV 之间的间距为 20px，4 个子 DIV 只能有一个 class 名。写出实现效果所需的代码。

【提示】

HTML 部分为:

```
<div class="f">
    <div></div>
    <div></div>
    <div></div>
    <div class="f_sp"></div>
</div>
```

CSS 部分为:

```
.f{width:460px;height:100px;background-color:#000;}
.f div{width:100px;height:100px;background-color:#ccc;float:left;margin:0 20px 0 0;}
.f .f_sp{margin:0;}
```

面试题 10 写出一段左边固定、右边随电脑屏幕浮动而变动的 HTML+CSS 代码

【提示】

HTML 部分为:

```
<div class="out">
    <div class="part1"></div>
    <div class="part2"></div>
</div>
```

CSS 部分为:

```
body,div{margin:0;padding:0;}
.out{width:980px;height:300px;overflow:hidden;}
.part1{width:280px;height:300px;background-color:green;float:left;position:relative;}
.part2{width:700px;height:300px;background-color:red;float:left;}
```

面试题 11 写一段两个 DIV 交叉重叠的 CSS 样式

写一段 CSS 样式,要求将两个 DIV 交叉重叠在一起,下面的 DIV 背景色为黑色,上面的背景色为白色。

【提示】

```
<!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>交叉重叠层</title>
<style type="text/css">
html{ color:#f00;}
#top{width:200px;height:300px; background-color:#fff;position:absolute;top:50px;
left:50px;}
#bottom{width:300px;height:400px; background-color:#000; z-index:-100;position:
absolute;top:0;left:0;}
</style>
</head>
<body>
```

```
<div id="top">上面层</div>
<div id="bottom">下面层</div>
</body>
</html>
```

面试题 12 怎样使一个层垂直居中位于浏览器中

【提示】

```
<style type="text/css">
<!--
div {
    position:absolute;
    top:50%;
    left:50%;
    margin:-100px 0 0 -100px;
    width:200px;
    height:200px;
    border:1px solid red;
}
-->
</style>
```

面试题 13 Firefox 嵌套 DIV 标签居中问题的处理方法

假定有如下情况:

```
<div id="a">
  <div id="b"> </div>
</div>
```

如果要实现 b 在 a 中居中放置,一般只需用 CSS 设置 a 的 text-align 属性为 center 即可。这样的方法在 IE 里看起来一切正常,但是在 Firefox 中 b 却是居左的。处理办法就是设置 b 的横向 margin 为 auto。例如设置 b 的 CSS 样式为: margin:0 auto;。

面试题 14 用纯 CSS 实现带下拉菜单的二级横向导航菜单

面试者碰到这种手写代码的问题会感到发憊,所以平时应该多练习手写代码。

【提示】

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

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
  <title>CSS下拉菜单</title>
  <style type="text/css">
    *{margin:0;padding:0;}
    .menu{font-size:12px;position:relative;z-index:100;}
    .menu ul{list-style:none;}
    .menu li {float:left;position:relative;}
    .menu ul ul {visibility:hidden;position:absolute;left:3px;top:23px;}
  </style>
</head>
</html>
```

```

.menu table {position:absolute; top:0; left:0;}
.menu ul a:hover ul{visibility:visible;}
.menu a{display:block;border:1px solid #000;background:#8192A6;padding:2px
10px;margin:3px;color:#fff;text-decoration:none;}
.menu a:hover{background: #0FF;color:#f00;border:1px solid #00F;}
.menu ul ul li {clear:both;text-align:left;font-size:12px;}
.menu ul ul li a{display:block;width:100px;height:13px;margin:0;border:
0;border-bottom:1px solid red;}
.menu ul ul li a:hover{border:0;background:#f2cdb0;border-bottom:1px solid #00F;}
</style>
</head>

<body>
<div class="menu">
<ul>
<li><a href="#">网站首页
<!--[if IE 7]><!--></a><!--<![endif]-->
<!--[if lte IE 6]><table><tr><td><![endif]-->
<ul>
<li><a href="#">最新内容</a></li>
<li><a href="#">最受关注</a></li>
<li><a href="#">分类导航</a></li>
</ul>
<!--[if lte IE 6]></td></tr></table></a><![endif]-->
</li>
<li><a href="#">网页特效
<!--[if IE 7]><!--></a><!--<![endif]-->
<!--[if lte IE 6]><table><tr><td><![endif]-->
<ul>
<li><a href="#">CSS菜单</a></li>
<li><a href="#">层和布局</a></li>
</ul>
<!--[if lte IE 6]></td></tr></table></a><![endif]-->
</li>
<li><a href="#">源码下载
<!--[if IE 7]><!--></a><!--<![endif]-->
<!--[if lte IE 6]><table><tr><td><![endif]-->
<ul>
<li><a href="#">ASP源码</a></li>
<li><a href="#">PHP源码</a></li>
</ul>
<!--[if lte IE 6]></td></tr></table></a><![endif]-->
</li>
<li><a href="#">素材模板</a></li>
<li><a href="#">网站留言</a></li>
</ul>
</div><br>
IE7、8存在兼容问题……
</body>

</html>

```

面试题 15 在网页制作中为什么要使用 CSS 技术

CSS (Cascading Style Sheets) 也叫层叠样式表, 用它来控制网页中一个或多个元素的外观。

在网页制作中,使用样式表可以减少很多具有相同外观元素的重复设置工作,从而给网页的设计和维持带来了很大的方便。例如,改变样式表中元素显示样式的定义,则在使用该样式表的所有网页中相应元素的显示样式都会自动发生改变。

面试题 16 CSS+DIV 开发 Web 页面的优势有哪些

【提示】

(1) 在 CSS+DIV 网页设计模式中, DIV 承担了网页的内容, CSS 承担了网页的样式。这样就使网页的内容和样式分离开来,有利于页面的维护升级。

(2) 有助于提高搜索引擎亲和力(快速找到需要的数据,而不是像在 Table 中一层层地查找)。

(3) 有助于页面的重构(如给 Blog 换皮肤,直接套用另外一套样式就可以实现,而不用改动网页脚本)。

面试题 17 实现两列高度自适应的 CSS 代码

两列高度自适应,不用 JavaScript,也不用 padding、margin 折合的方式,要求用纯 CSS+DIV 布局方法,不能用 padding-bottom:1000px;margin-bottom:-1000px;这类的办法。

【提示】

```
<style>
    .row { display:table-row; overflow:hidden; clear:both; zoom:1; }
    .row div { display:table-cell; width:30%; margin-bottom:-500px; padding:1.6%;
*padding-bottom:510px; *float:left; }
    #column1 { background:#f00; width:200px; }
    #column2 { background:#000; width:200px; }
    #wrap { width:300px; height:100px; background:#0f0; }
</style>

<div class="row">
<div id="column1">这<br />是第<br /><br /><br /><br /><br />一列</div>
<div id="column2">这<br /><br /><br /><br />是第二列</div></div>
<div id="wrap"></div>
```