



# 前言

## 我们的HTML

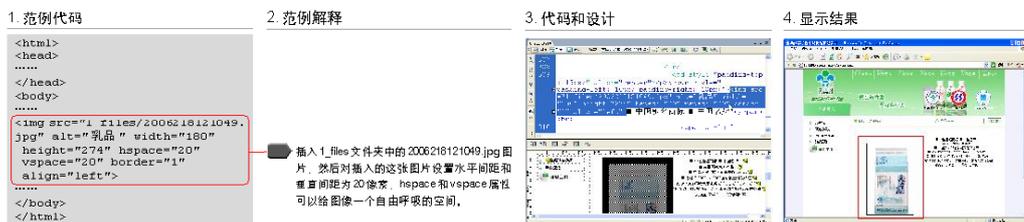
HTML 给人更多的感觉是代码，不过它与其他的程序语言有着很大的不同，它是 Hypertext Markup Language，也就是“超文本标记语言”，请大家不要忽略“标记”，它才是HTML的精华所在，也就是<>中的内容。每个标记都有它特定的含义，精彩的Web页面就是靠它们描述出来的。大多数标记成对出现，中间是属性等实质性内容，后面的标记加上一个“/”，表示结束部分。这样看起来你会觉得HTML也不是什么高难的语言，仅此“标记组合”而已。

本书弥补了以往HTML图书的不足，将真实的网页设计案例融入到培训专家的讲义中，铸造了一本真正意义上从入门到精通的专家级HTML参考书。

## 本书主要特色

全书包括13章及附录，每个章节都配有完整的制作实例，使读者能够在真实的学习环境中掌握HTML语言。

- 起点较低，收获丰富——真正将HTML代码编写放入Dreamweaver 8中结合讲解，读者能直接体验从入门到精通的学习过程。
- 上手容易，轻松学习——100%网页设计真实案例讲解，让学习代码像学习可视化操作一样简单、容易。
- 代码注解，时时提示——添加注释，对每段代码中的词汇进行细致的解释，避免理解歧义的发生，让读者真正领会代码的所有含义。
- 四位一体，全面讲解——范例代码、范例解释、代码和设计、显示结果四位一体的讲解过程，把代码和制作效果直接联系在一起。



- 公开代码，方便查阅——所有实例代码及效果文件放在附赠光盘中，读者可以随时学习使用、借鉴发挥。
- 配送丰富，物超所值——16进制RGB值与命名对照表、HTML元素速查表、HTML字符集识别等内容随附在光盘中。

## 看后可以学会

HTML 语言是任何高级网页制作技术的核心与基础，本书通过几十个真实的网站实例讲解了上百个语法知识点的应用。主要包括：

- HTML 语言的基本概念和特点，对 HTML 语言有一个初步的理解。
- 使用记事本和 Dreamweaver 编写 HTML 页面，迅速了解到网页的制作流程。
- HTML 网页的结构、头部信息、页面的整体设置。
- 通过 HTML 语言对字体、大小、颜色、底纹、边框等设置文本的属性。
- 使用 HTML 语言制作页面中的超级链接。
- 图片、音频、视频、动画、滚动字幕等多媒体效果在网页中的应用。
- 表单、表格、框架结构的使用，让网页更加完美。
- CSS 样式表、JavaScript 脚本的使用，是动静 Web 设计的最佳选择。

## 光盘附赠

1. 全书所有实例的原始文件以及完成文件。
2. 8款Dreamweaver中使用的源代码相关插件。
3. 8款优秀源代码编辑器。
4. HTML 语言电子学习手册。
5. HTML 语言授课教学幻灯片。
6. JavaScript特效代码集。
7. 网页颜色表。
8. 语法简明速查表。

## 作者要说

作者在3个月的写作时间里，一直坚持不懈地投入很多精力，可以看到本书在内容的原创性、体例的特殊性和版式的创新性上下了很大的功夫，目的是希望为学习网页制作的读者提供一本讲述HTML语言、CSS样式表的经典图书。从这里开始，HTML将不再是高不可攀的代码技术，而是正在变得愈加和蔼可亲起来。

作 者  
2007年1月



# 目 录 CONTENTS

## Chapter 1

初识HTML语言 .....	1
WWW 及其发展现状 .....	2
WWW 的基本概念 .....	2
WWW 的工作机制 .....	2
WWW 的应用 .....	3
WWW 的站点建设 .....	4
HTML 简介 .....	5
HTML 基本概念 .....	5
HTML 的发展历史 .....	6
HTML 4.0 的特点 .....	6

## Chapter 2

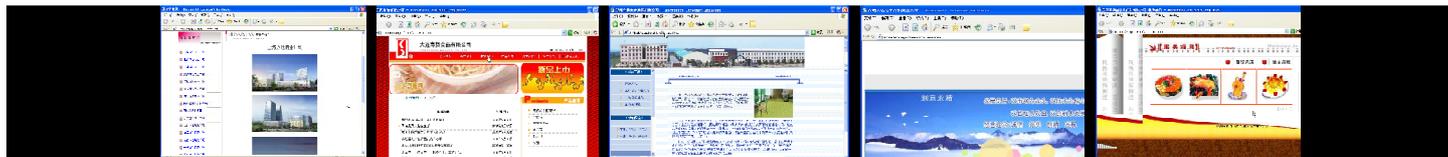
快速编写HTML文件 .....	9
使用记事本编写HTML页面 .....	10
使用编辑器编写HTML页面 .....	11
HTML 文件的基本结构 .....	14
HTML 中的标签 .....	15
HTML 中的属性 .....	16

## Chapter 3

文档的顶级信息 .....	19
根元素 .....	20
Html 标签 .....	20
Head 标签 .....	22
Body 标签 .....	23
头部元素 .....	29
Title 标签 .....	29
Base 标签 .....	30
Basefont 标签 .....	32
Meta 标签 .....	33
实例制作 .....	41

## Chapter 4

文字和段落 .....	43
文字内容 .....	44
空格 .....	44
特殊字符 .....	45
注释 .....	46



## Chapter 5

显示HTML代码标签	47
文字标记	48
Font标签	48
Hn标签	51
文字的修饰	53
B和Strong标签	54
斜体字标签	55
Sup标签	56
Sub标签	57
Big标签	58
Small标签	59
U标签	59
S和Strike标签	60
Address标签	61
Code和Samp标签	63
Kbd标签	64
Tt标签	64
Var标签	65
Rt和Ruby标签	66
段落的排布	67
P标签	68
Br标签	71
Nobr标签	72
Div标签	73
Span标签	76
Center标签	77
Blockquote标签	79
实例制作	80
<b>水平线和列表</b>	<b>83</b>
水平线	84
Hr标签	84
列表	89
无序列表	89
有序列表	93
目录列表	98
定义列表	99
菜单列表	101

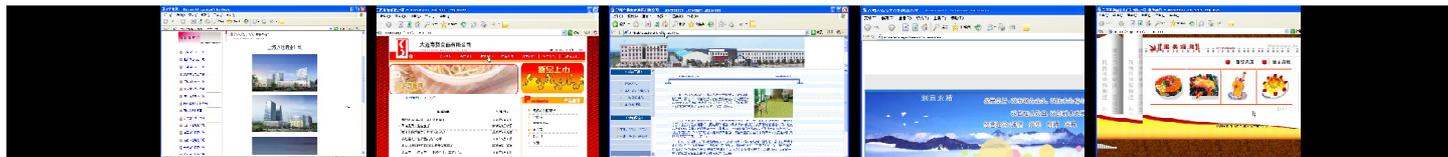


# Chapter 6

# Chapter 7

# Chapter 8

实例制作 .....	103
<b>插入图片 .....</b>	<b>107</b>
网页图片的基本格式 .....	108
GIF图像 .....	108
JPEG 图像 .....	109
插入图片 .....	109
<img>标签 .....	109
实例制作 .....	116
<b>页面中的超级链接 .....</b>	<b>119</b>
关于超级链接 .....	120
URL .....	120
制作超级链接 .....	120
外部链接 .....	121
访问News 新闻组 .....	127
发送Email .....	128
内部链接 .....	129
书签链接 .....	130
空链接 .....	132
脚本链接 .....	132
制作图像映射 .....	133
实例制作 .....	137
<b>多媒体效果 .....</b>	<b>141</b>
页面中的背景音乐 .....	142
Bgsound 标签 .....	142
插入图片 .....	143
Embed 标签 .....	143
嵌入Flash 动画 .....	144
嵌入Mp3 音乐 .....	146
嵌入Mpg 视频 .....	146
利用Img 标签插入AVI 视频 .....	147
插入AVI 视频 .....	147
制作滚动字幕 .....	150
Marquee 标签 .....	150
其他嵌入式内容 .....	158
使用Object 标签 .....	158
使用Applet 标签 .....	159



## Chapter 9

## Chapter 10

## Chapter 11

使用Param标签 .....	160
实例制作 .....	163
<b>表单元素 .....</b>	<b>167</b>
什么是表单 .....	168
Form标签 .....	168
表单元素 .....	171
Input标签 .....	171
Textarea标签 .....	184
Select标签 .....	186
Option标签 .....	187
Optgroup标签 .....	190
表单元素分组 .....	192
Label 标签 .....	192
Fieldset 标签 .....	194
Legend 标签 .....	194
实例制作 .....	197
<b>使用表格排版网页 .....</b>	<b>203</b>
表格的整体结构 .....	204
Table 标签 .....	204
Tr 标签 .....	205
Td 标签 .....	206
Th 标签 .....	207
Caption 标签 .....	209
表格的属性 .....	211
Table 标签的属性 .....	211
Tr 标签的Valign 属性 .....	225
Td 和 Th 标签的属性 .....	226
Caption 标签的属性 .....	231
高级表格标签 .....	234
Thead 标签 .....	234
Tbody 标签 .....	237
Tfoot 标签 .....	239
实例制作 .....	241
<b>框架结构页面 .....</b>	<b>249</b>
框架的整体结构 .....	250
Frameset 标签 .....	250



# Chapter 12

# Chapter 13

Frame标签 .....	251
Noframes标签 .....	252
框架的属性 .....	255
Frame标签的属性 .....	255
Frameset标签的属性 .....	263
制作内联框架 .....	266
Iframe标签 .....	266
Iframe标签的属性 .....	268
实例制作 .....	275
<b>使用CSS样式表将结构和样式分开 .....</b>	<b>279</b>
什么是样式表 .....	280
样式表的基本语法 .....	280
样式表类型 .....	280
样式表语法 .....	284
样式表的属性 .....	289
字体属性 .....	290
颜色和背景属性 .....	296
文本属性 .....	303
边框属性 .....	311
方框属性 .....	315
列表属性 .....	323
滤镜属性 .....	327
鼠标滤镜 .....	340
实例制作 .....	341
<b>在页面中使用JavaScript脚本 .....</b>	<b>345</b>
什么是JavaScript脚本 .....	346
在页面中加入JavaScript脚本 .....	346
Script标签 .....	346
使用内在事件调用 .....	348
脚本链接 .....	349
JavaScript内在事件 .....	350
onBlur事件 .....	351
onChange事件 .....	352
onClick事件 .....	352
onFocus事件 .....	353
onLoad事件 .....	354
onMouseOver事件 .....	355



# 附录

onMouseOut 事件 .....	356
onReset 事件 .....	357
onSelect 事件 .....	358
onSubmit 事件 .....	359
onUnload 事件 .....	360
实例制作 .....	361
<b>附录 .....</b>	<b>365</b>
附录1 .....	366
附录2 .....	367
附录3 .....	378
附录4 .....	381



# Chapter

# 1

## 初识HTML语言

HTML（超文本标识语言）是能生成活动文档的代码系统，活动文档就是使用户和别人能够进行交互作用的文档。HTML的普通应用带来了超文本技术——从一个主题转入另一主题，而不是线性地寻找和阅读信息。如今人们不再需要逐页翻阅信息，而可以直接单击链接取得所要的信息，通过E-mail与别人联系，填写表单并联机提交，以及访问巨大的数据库和信息资源等。而这些正是HTML使文档的访问、阅读及使用变得如此快捷和有效。

```
meta http-equiv="Content-Type" content="text/html" charset="gb2312">
<link href="1_files/css.css" rel="stylesheet" type="text/css">
```



## 1 WWW 及其发展现状

作为一种新的信息传播与处理技术，WWW 极大地推动了 Internet 的发展及信息的交流与共享。

### WWW 的基本概念

WWW (World Wide Web) 最初起源于 CERN (欧洲粒子物理实验室) Tim Berners-Lee 提出的一项名为“分布式超媒体系统”工程，其最初目的是使科学家们能容易地查阅同行们的文章。后来人们发现这项工程中的超文本和多媒体技术非常适于信息的发布与共享，于是被广泛应用到 Internet 上，并由此成为 Internet 发展的推动器。后来在 MIT 专门成立了一个 W3C 组织，负责 WWW 的技术研讨、标准发布以及应用推广等工作。

在 1990 年 WWW 出现以前，Internet 上的应用大多采用文字终端，其使用对象为少数计算机专业人员。自从 WWW 出现后，Internet 开始摆脱 UNIX 的局限和实验室环境，走进寻常百姓家。在 1993 年 2 月，以 NCSA 开发的 WWW 浏览器 Mosaic 为标志，WWW 初具规模并走向成熟，Internet 上的应用开始转向图形用户界面并提供多媒体功能。

WWW 主要由一些应用软件以及一系列协议和约定组成，它使用超文本和多媒体技术在计算机网络上进行信息的发布和浏览。从概念上来说，WWW 可以应用在所有的计算机网络上，而并非仅 Internet：可以是基于 TCP/IP 协议的 Intranet，也可以是基于其他网络协议（如 IPX/SPX）的网络。对基于不同网络协议的网络，WWW 的实现需要相应的应用软件来支挂。目前 WWW 主要应用在基于 TCP/IP 协议的网络上。

### WWW 的工作机制

WWW 服务器端有大量的原始数据（如文本、图像、视频、音频等）和超文本文档（用于组织管理原始数据），这些信息在存放方式、访问模式、数据格式等方面差异很大。如何将这信息有效地组织起来，让用户方便地访问是 HTML 语言研究的重点之一。WWW 通过 HTML 文件来组织各类原始数据，这种文档可以包含各种数据类型（经 MIME 定义）的数据。其实 HTML 文档中本身并不包含这些数据，而仅仅包含这些数据文件的链接指针。这些链接指针分别指向存放在不同地方的、不同类型的数据。WWW 管理和传输的数据一般是 HTML 文档和原始数据。这些数据通过 HTTP 协议来进行传输。

WWW 采用 Client/Server 模式，WWW 服务器软件（可能还包括其他相关软件，如网关、Proxy Server、扩展应用软件）负责处理客户端的请求，并将有关数据（HTML 文档和原始数据）通过网络返回给客户端。客户端软件（一般是 WWW 浏览器）负责初始化用户的请求后提交给 WWW 服务器，并对服务器返回的数据进行处理和显示（有时还需要相关的软件协助处理）。因此对 HTML 文档的解释处理、对各种网络数据的获取、对各种多媒体数据的处理，都是通过 WWW 浏览器来完成的。浏览器是 WWW 实现过程中的重要组成部分。目前主要有两种 WWW 浏览器：一是 Netscape 公司的 Navigator，一是 Microsoft 的 IE。

在技术方面, WWW 主要由 HTML (超文本标识语言)、HTTP (超文本传输协议)、CGI (通用网关接口)、URL (统一资源定位符)、MIME 等技术组成。其中 HTTP 协议是 WWW 的基础协议, WWW 服务器和客户端软件都是基于 HTTP 协议实现的; WWW 通过 HTTP 协议传输管理的数据主要是超本文档信息, 超文本文件是用 HTML 语言编写的; 在超本文档中, 可以包含许多指向不同类型的网络资源的超文本链接, 这些链接用 URL 表示; CGI 是基于 WWW 服务器的一种标准扩展技术; MIME 是用来定义 Internet 数据类型的一种标准。除了 HTTP 协议外, WWW 还涉及其他一些相关协议, 如 FTP 和 Telnet 等, WWW 能兼容这些协议所具有的功能。

## WWW 的应用

应该说 WWW 是基于 Internet 的一种应用, 只不过已成为 Internet 上最主要的应用, 实际上它已经涵盖了大部分传统的标准应用。其实人们平时所说的 Internet, 主要指的是其硬件设施和网络本身提供的基本通信服务方面 (从学术上说, 它处于 TCP/IP 协议组的应用层以下, 如物理层、链路层、IP 层和 TCP 层); 而基于它的网络应用则很多。这种网络应用一般分为标准应用 (一般有相应的 RFC 推荐标准) 和专业领域应用 (一般有行业标准或自定义协议标准)。标准应用一般包括文件传输、终端仿真、电子邮件、WWW、Gopher 等应用, 专业应用则随具体的应用需求而异, 在实际应用中, 它们常互为补充。

WWW 之所以被广泛接受, 主要在于其实施简单 (可充分利用以前的系统设备)、管理和维护方便 (可远程维护管理并且采用统一的多媒体界面)、高效率 and 多媒体功能。

WWW 为网络用户提供了一种全新的信息查询、信息浏览和信息发布手段。作为一种信息处理技术, 其应用领域可谓无所不及。当然, 它最主要的应用还在于 Internet/Intranet。这些站点提供的信息和服务涉及到社会生活的各个领域, 常见的应用如下。

### 1. 信息发布

借助于 Internet 遍布全球的优势, 在网上作广告、宣传、电子出版等。例如宣传公司的产品、图文并茂的风景名胜、人文地理、投资环境介绍等。



### 2. 数据查询

通过搜索引擎, 如 Yahoo、Infoseek, 查询网上的各种信息等。





### 3. 实况转播

将最新信息实时地在网上传播。在网络传输速率较高的地区，可以实现有限的含图像和声音的转播，例如通过Internet进行围棋、桥牌等转播。



### 4. 电子商务

通过 WWW 平台进行远程购物、电子商场、电子货币、电子银行等商务活动，是目前 WWW 发展的一个主要方向。



### 5. 专业应用

通过 WWW 平台进行各个领域的专业应用，是许多行业建立 WWW 站点的主要因素。



WWW的应用不仅在Internet，也广泛用于Intranet（基于TCP/IP的企业内部网）。在Intranet上也可以实现与Internet相同的应用，只是它对信息的传播与访问进行了一定的限制。传统的企业MIS大多版本各异、维护管理困难，但是借助于WWW技术，我们可以建立基于Web的企业MIS系统，并且更加灵活有效。在实际的应用系统中，WWW本身提供的功能是有限的，例如动态交互，就需要根据实际需求开发应用软件，与WWW集成在一起提供服务。开发专门的WWW扩展系统可以借助于Java、CGI等技术来实现。

## WWW的站点建设

建设WWW站点首先需要以下相关设备和条件。

- 硬件设施：WWW服务器的硬件平台一般选择PC服务器或工作站，相对而言前者具有较好的性能价格比。但如果对处理效率、网络吞吐量等方面有较高要求，则宜选择工作站。
- 软件平台：主要是服务器操作系统和WWW服务器软件。可选的操作系统主要有UNIX和Windows等；可选的WWW服务器软件很多，如IIS、Apache等。此外为方便管理与调

试, 还应有 WWW 浏览器、主页编辑器、图像处理等辅助性工具。UNIX 和 Windows 各有优势, 从本身实现机制和性能方面来看, UNIX 的要好一些; 但 Windows 与普通用户的距离似乎更近一些。

- 网络平台: 如果尚无网络平台, 还需要建立网络环境(可能需要购买路由器、访问服务器以及集线器等网络设备)。此外, 如果建立的 WWW 站点要提供 Internet 访问, 申请 IP 地址是必须的。IP 地址可以从电信局或拥有剩余 IP 地址的相关公司申请。申请 IP 地址的数量取决于实际的需求, 建立一个 WWW 站点最好拥有 3 个以上的 IP 地址。
- 维护和开发人员: 这方面人员的数量和技术水平应根据应用需要决定。如果建立的 WWW 站点要提供大量的经常变化的信息, 则必须拥有一定数量的维护人员, 负责网页的制作和各种原始数据的处理。如果站点还要提供 HTML 等描述性语言(包括 JavaScript 等语言)不能实现的服务, 如股市行情、新闻信息发布检索系统以及网络沙龙等需要动态数据交换的系统, 则需要采用 ASP、Java 等技术进行专门的应用开发。
- 信息采集与处理: WWW 站点需要提供大量的信息, 有些信息可能还需要特定的系统进行处理。因此, 信息的采集与处理工作很重要。这是站点建立后比较主要的工作。
- 站点安全管理: 如果 WWW 站点与 Internet 相连, 那么安全对某些用户来说就显得很重要了。安全管理不仅涉及到系统硬件和软件方面, 还涉及到公司机密等信息的安全管理。目前, WWW 在技术和应用方面已比较成熟, 大部分企业内部网都引入了 WWW 信息处理机制。对于 ISP 来说, WWW 站点的建设成为其重要任务。对于软件开发商而言, 存在两大方面的机会: 将传统的信息系统改造移植到基于 Web 的应用上来; 另外一个则是开发基于或扩展 WWW 的应用, 例如新闻信息动态出版与浏览系统、基于 WWW 的虚拟现实会议系统、基于 Web 的全文检索系统、数据库网关接口以及传输层网关等中间件等。

## 2

## HTML 简介

Internet 的飞速发展导致了成千上万互联网站的创建, 我们浏览这些网站的时候, 看到的是丰富的影像、文字、图片, 而这些内容都是通过当 HTML 语言表现出来的。对于网页设计和制作人员, 尤其是开发动态网站的编程人员来讲, 制作网页的时候, 如果不涉及 HTML 语言, 几乎是不可能的。

## HTML 基本概念

要把信息发布到全球, 就必须使用能够被大众接受的语言, 也就是使用一种大多数计算机能够识别的发布语言。在 WWW 上, 通常使用的发布语言是 HTML, 即超文本标识语言。HTML 语言(HyperText Markup Language, 中文通常称作超文本标识语言或超文本标记语言)是一种解释执行的文本类标记语言, 它是 Internet 上用于编写网页的主要语言。用 HTML 编写的超文本文件称为 HTML 文件。

HTML 语言是一种简易的文件交换标准, 有别于物理的文件结构, 它旨在定义文件内的对象和描述文件的逻辑结构, 而并不定义文件的显示。由于 HTML 所描述的文件具有极高的适应性, 因此特别适合于 WWW 的发布环境。





HTML 是纯文本类型的语言，使用 HTML 编写的网页文件也是标准的纯文本文件。我们可以用任何文本编辑器，例如 Windows 的“记事本”程序打开它，并查看其中的 HTML 源代码，也可以在浏览器打开网页时，通过单击相应的“查看>源文件”命令查看网页中的 HTML 代码。HTML 文件可以直接由浏览器解释执行，而无需编译。当用浏览器打开网页时，浏览器读取网页中的 HTML 代码，分析其语法结构，然后根据解释的结果显示网页内容，正是因为如此，网页显示的速度同网页代码的质量有很大的关系，保持精简和高效的 HTML 源代码十分重要。

## HTML 的发展历史

HTML 最初由 Tim Berners-Lee 开发——当他在 CERN 时，因为 NCSA 开发的 Mosaic 浏览器而流行起来；在 1990 年由于 Web 的爆炸性增长而得到成功。这期间，HTML 以多种不同方式被扩充，并且网页作者和零售商共享 HTML，因此这就促进了他们合作制定 HTML 的规范。在 Internet Engineering Task Force (IETF) 的支持下，在 1994 年末通用的规则被制定为 HTML 2.0（1995 年 11 月制定）。HTML+（1993 年）和 HTML 3.0（1995 年）在内容上做了进一步的丰富。尽管从来没有收到标准讨论的一致同意，但是这些草稿导致很多新特征的采用。WWW 协会的 HTML 工作组在 1996 年的努力结果是 HTML 3.2（1997 年 1 月）。时至今日，HTML 已经发展到了比较成熟的 HTML 4.0 版本，在这个版本的语言中，规范更加统一，浏览器之间的兼容性也更加完好了。HTML 已经发展成各种方式的设备都可以利用网上的信息：蜂窝电话、手持设备、快速输入 / 输出设备、计算机等。

## HTML 4.0 的特点

HTML 4.0 通过以下机制扩展了 HTML：样式表、脚本、框架、嵌入对象、改进的从右到左的支持和混合方向文本、功能更强的表格、增强的表单，并且为残疾人提供了访问的可能。

**国际化：**这个版本的 HTML 在设计过程中得到了国际化领域专家的帮助，因此可以用各种语言写文档并易于在世界范围内发布。设计过程中的一个重要步骤是采用了 ISO/IEC: 10646 标准作为 HTML 的字符集。它是处理国际化字符、文本方向、发音和其他语言问题的最具包容性的标准。HTML 文档内提供对不同人类语言的强大的支持。这就为搜索引擎提供了更有效的文档索引功能、更高质量的排版格式、更好的文本到发音的转换、更好的连字符连接等。

**可访问性：**网络社区在增长，而它的成员的能力和技术各不相同，那么底层技术满足一些特殊需要是很重要的。HTML 已经被设计成可以被有身体缺陷的人访问的网页。HTML 4.0 便于访问的特性包括以下几个方面。

- 文档的结构和表现更好地区分开来，这就鼓励使用样式表而不用 HTML 表现元素和属性。
- 更好的表单，包括附加的存取键、按语意分类表单控件、Select 选项和活动标签。
- 标记作为内嵌对象的文本描述的能力（使用 Object 元素）。
- 一个新的客户端图像映射机制（MAP），允许把图像和文本链接集成在一起。
- 替代文字与图像（包含在 Img 元素内）和图像映射（包含在 Area 元素内）一起出现。

- 支持所有元素的 Title 和 Lang 属性。
- 支持 Abbr 和 Acronym 元素。
- 与样式表一起使用的更多的目标媒体 (Tty、 Braille 等)。
- 更好的表格，包括标题、列组和实现不可见渲染的机制。
- 表格、图像、框架等的长名描述。

在设计中考虑到可访问性因素不仅可以方便访问者，还便于设计更容易地采用新技术。

**表格：**通过表格，设计者可以更好地控制结构和布局。定义好列宽度后可以逐渐地显示表格，而不必等到全部表格绘制完。

**混合文档：**HTML 提供了在 HTML 文档中嵌入常见媒体对象和应用程序的标准机制。Object 元素（与 Img 和 Applet 一起）提供了在文档内包含图像、视频、声音、数学、特殊应用程序和其他对象的功能。

**样式表：**简化了 HTML 标记并且减轻了 HTML 对表现形式的责任。它们使设计者和用户都可以控制文档的表现形式（字体信息、对齐方式、颜色等）。样式信息可以为单独的元素或一组元素定制。样式信息可以定制在 HTML 文档内或以外部样式表的形式出现。

把样式表与文档联系起来的机制独立于样式表语言。在样式表出现之前，制作者很少能控制页面绘制过程。HTML 3.2 包括一些属性和元素，它们提供了对对齐方式、字体大小和文本颜色的控制。制作者也使用表格和图像进行页面布局。用户需要较长时间才能升级他们的浏览器，意味着对这些特征的使用还要持续一段时间。然而，既然样式表提供更强大的表现机制，WWW 协会将逐渐淘汰很多 HTML 的表现元素和属性。

**脚本：**通过脚本，制作者可以制作动态网页，例如使用 HTML 建立网络应用程序。把脚本包含在 HTML 文档中的机制独立于脚本语言。

**打印：**有时，制作者希望阅读者不仅是只打印当前文档。当文档成为一个工程的一部分时，它们之间的关系可以被描述为使用 HTML Link 元素或使用 W3C 的资源描述语言 (Resource Description Language, RDF)。





# Chapter

# 2

## 快速编写HTML文件

在编写第一个HTML文档之前，网页设计者不需要花费太多的时间去学习详细的语法，因为简单的HTML页面非常容易编写。学会本章的基本编写方法后，会增强读者的学习信心，这种信心来自于很短的时间内就可以创建出优秀的网页页面。

```
meta http-equiv="content-type"
charset="gb2312"
<link href="1_files/css.css" rel="stylesheet" type="text/css">
```



# 1 使用记事本编写HTML 页面

创建第一个HTML 文档，只需要两个基本工具：HTML 编辑器和Web 浏览器。

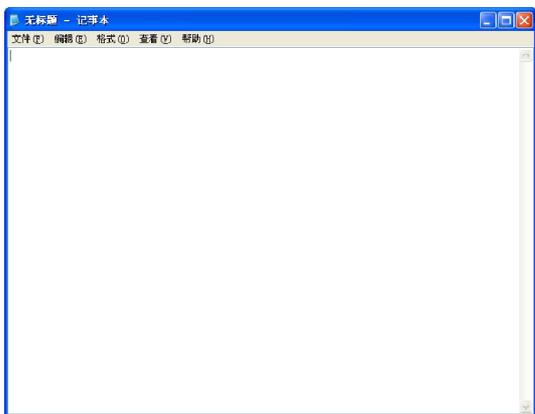
- HTML 编辑器就是用于生成和保存HTML 文档的程序。
- Web 浏览器就是用于浏览和测试HTML 文档的程序。

一般而言，HTML 编辑器分为两类：基于文本或代码的编辑器，在创建文档时只能看到HTML 代码；所见即所得编辑器，在格式化文档时即显示出类似于浏览器窗口的结果。

由于HTML 语言编写的文件是标准的ASCII 文本文件，因此可以使用任何文本编辑器来打开并编写HTML 文件，如Windows 系统中自带的记事本。下面就使用记事本来快速编写第一个HTML 文件。

范例文件	Sample\Ch02\1\1.htm
学习要点	使用记事本编写页面

1 单击“开始>程序>附件>记事本”命令，打开记事本程序。



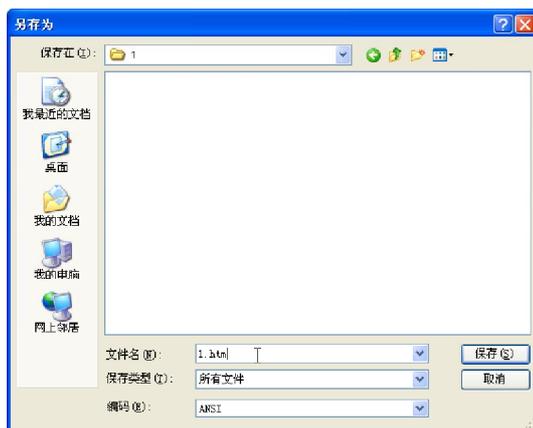
2 在记事本中直接输入下面的HTML 代码（不区分大小写）：

```
<html >
<head>
<title>使用记事本编写html 文件</
title>
</head>
<body>
<p align="center">
快速编写html 文件
</p>
</body>
</html >
```

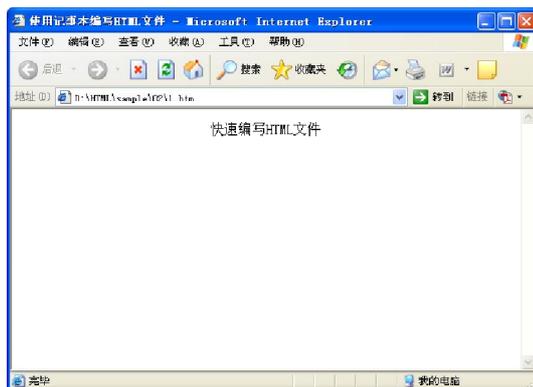
3 输入了代码后，记事本中显示出了代码的内容。



- 4 单击记事本菜单栏中的“文件>保存”命令，弹出“另存为”对话框。
- 5 在对话框中选择存盘的文件夹，然后在“保存类型”下拉列表中选择“所有文件”，在“编码”下拉列表中选择“ANSI”，这里将“文件名”设置为1.htm，然后单击“保存”按钮。



- 6 关闭记事本，回到存盘的文件夹，双击1.htm文件，可以在浏览器中看到最终的页面效果。



## 2

## 使用编辑器编写HTML页面

FrontPage 和 Dreamweaver 作为可视化网页编辑软件的代表，在网页制作的过程中得到了广泛应用。下面以 Macromedia（现 Adobe）公司的 Dreamweaver 8 中文版为例，说明使用可视化网页编辑软件制作页面的方法。

范例文件	Sample\Ch02\2\1.htm
学习要点	使用编辑器编写页面

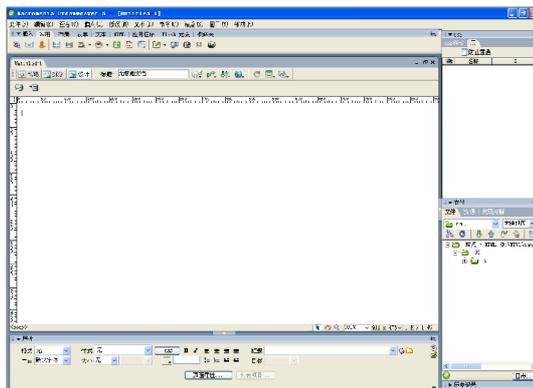


2

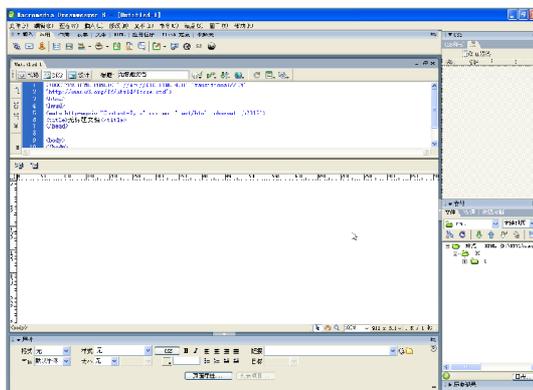
快速编写HTML文件



1 单击“开始>所有程序>Macromedia>Macromedia Dreamweaver 8”命令，启动软件的主程序。

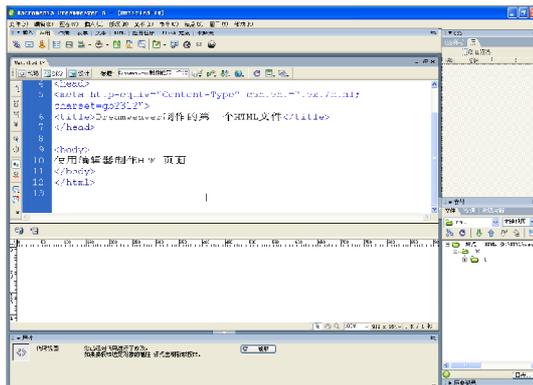
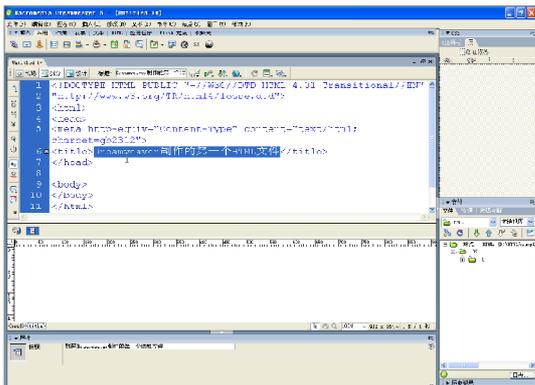


2 在这个软件的主编辑环境中，分为“代码视图”、“代码视图和设计视图”、“设计视图”这3种。单击“拆分”按钮，切换到“代码视图和设计视图”环境。在软件的上半部分显示出代码视图的内容，下半部分显示出设计视图的内容。这两部分是互相联系、密不可分的，在代码视图中所做的任何修改都会影响设计视图。



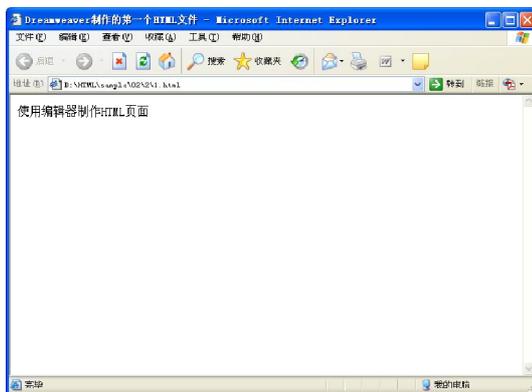
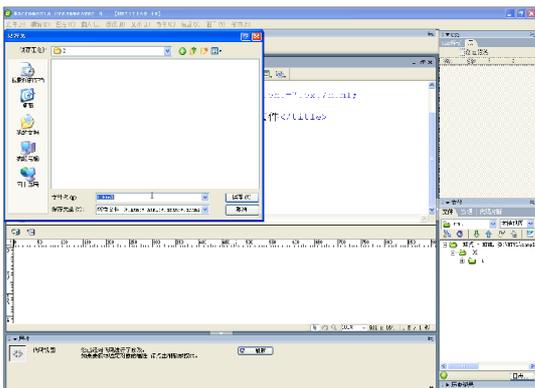
3 在<Title>标签后删除“无标题文档”，然后输入“Dreamweaver 制作的第一个HTML文件”，作为页面的标题。

4 在<Body>标签后输入“使用编辑器制作HTML页面”，作为页面的正文。



5 单击“文件>保存”命令，在弹出的“另存为”对话框中选择存盘的位置，将文件命名成1.html，然后单击“保存”按钮。

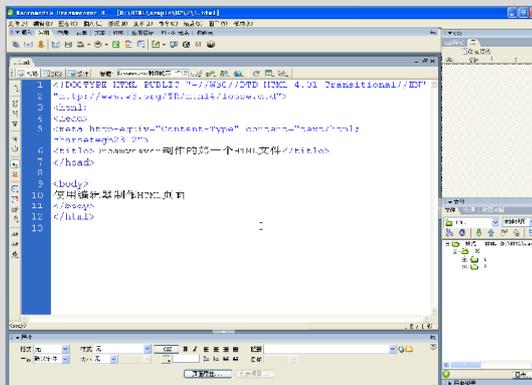
6 按下F12快捷键，可以在浏览器中看到页面的效果。



## 提示

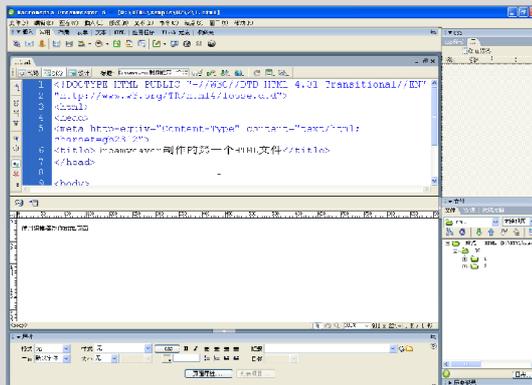
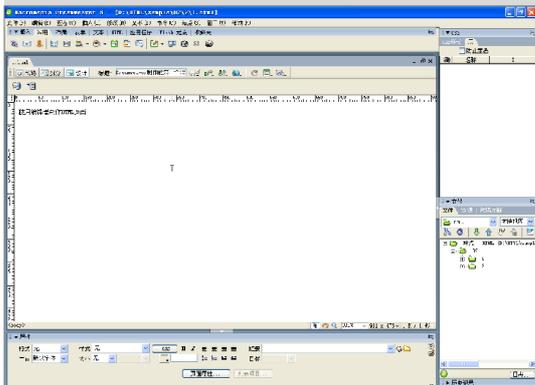
## Dreamweaver 的 3 种视图

- 代码视图：在可视化网页的背后，可以控制网页的源代码，如果想查看或编辑源代码，可以进入该视图。



- 设计视图：Dreamweaver 是可视化的网页编辑软件，所以设计视图是最常用的。设计视图中看到的网页外观和浏览器中看到的基本上是一致的。

- 代码视图和设计视图：在这种视图下，编辑窗口被分割成了上下两部分，上面显示的是源代码视图，下面是可视化视图，这样可以在选择和编辑源代码的时候即时在可视化视图中看到效果。



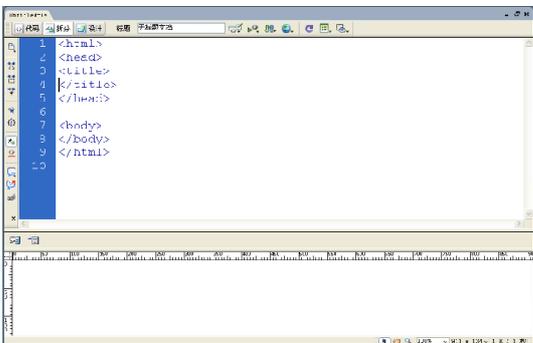


### 3 HTML文件的基本结构

编写HTML文件的时候，必须遵循HTML的语法规则。一个完整的HTML文件由标题、段落、列表、表格、单词等嵌入的各种对象组成。这些逻辑上统一的对象称为元素，HTML使用标签来分割并描述这些元素。实际上整个HTML文件就是由元素与标签组成的。

HTML文档的组成包括定义文档内容的文本和定义文档结构及外观的标签。HTML文档的结构很简单，是由最外层的<HTML>标签组成，里面是文档的头部和主体。

<code>&lt;html &gt;</code>	HTML 文件开始
<code>&lt;head&gt;</code>	HTML 文件的头部开始
<code>...</code>	HTML 文件的头部内容
<code>&lt;/head&gt;</code>	HTML 文件的头部结束
<code>&lt;body&gt;</code>	HTML 文件的主体开始
<code>...</code>	HTML 文件的主体内容
<code>&lt;/body&gt;</code>	HTML 文件的主体结束
<code>&lt;/html &gt;</code>	HTML 文件结束



每一个文档都包括头部(head)和主体(body)，它们分别由<head>标签和<body>标签分隔开来。头部就是放置文档标题的地方，同时也可以在这个地方指明浏览器在显示文档时可能会用到的其他参数。主体是放置文档实际内容的地方。其中包括要显示的文本和文档的控制标记(标签)，这些标记将告诉浏览器该如何显示那些文本。标签还可以引用特殊效果的文件，包括图形和声音，并指出将文档链接到其他文档的热点(超链接和锚)。

#### <html>标签

<html>标签表示该文档为HTML文档。技术上，这个标签在<!doctype>标签之后显得多余，但对于不支持<!doctype>标签的旧式浏览器来说，这个标签是必要的。它还能帮助人们阅读HTML代码，因此最好还是包括这个标签，以便其他工具尤其普通文字处理工具能够识别出文档是HTML文档。<html>开始和结束标签的存在，可以保证用户不会无意中删掉文档的开始或者结束部分。

## <head>标签

<head>标签中包含文档的标题、文档使用的脚本、样式定义和文档名信息，并不是所有浏览器都要这个标签，但大多数浏览器都希望在<head>标签中找到关于文档的补充信息。此外<head>标签还可以包含搜索工具和索引程序需要的其他信息的标签。

## <title>标签

<title>标签是 HTML 3.2 和 HTML 4 规范所要求的，它包含文档的标题。标题并不出现在浏览器窗口中，而是显示在浏览器标题栏中。在起始和结束标签之间，可以放置简述文档内容的标题。

## <body>标签

<body>标签放置要在访问者浏览器中显示信息的所有标签和属性。后面涉及的绝大多数内容都在<body>标签中。

## HTML 中的标签

标签很容易理解和使用，因为它们是由普通词语、缩写词和注释构成的。例如<body>和</body>标签显示包含在它们之间的字符。

## 标签语法

所有标签都有一个标签名称，有些标签后面还有一个可选的属性列表，所有这些都放在开始和结束括号之间（“<”和“>”）。最简单的标签是用括号括起来的一个名称，例如<head>和<i>。更复杂的标签则具有一个或者多个属性，用来指定或者修改标签的行为。

根据 HTML 标准，标签和属性名不区分大小写。实际上，在<head>、<Head>、<HEAD>，甚至<HeaD>之间没有任何区别，它们的作用都是一样的。但赋给特定属性的值可能会区分大小写，这取决于所使用的浏览器和服务器。尤其是对文件位置和名称的引用，以及统一资源定位符（URL），这些都要区分大小写。

如果标签有属性，那么这些属性将跟随在标签名称的后面，每个属性都由一个或者多个制表符、空格或者回车符分开。它们出现的顺序没有区别。

标签的属性值放在标签属性名称后面的等号（=）之后。等号的两边可以放置空格，这样一来 width=6、width = 6、width =6 和 width= 6 的效果就是完全一样的。但为了具有更好的可读性，建议在等号两边不使用空格。这样就可以很容易地在非常长的标签中把属性值找出来。

在 HTML 中，如果属性值是一个单独的词或数字（没有空格），那么直接将该值放在等号的后面即可。所有其他的值都必须加上单引号或双引号，尤其是那些含有用空格分开的多个词的值。





## 开始标签和结束标签

在前面提到过大多数标签都有一个开始标签和一个结束标签，而标签影响的范围就是它们之间包含的内容。内容包括的部分可能很大，也可能很小，从一个单独的文本字符、一个音节或者一个词，到像<html>这样包括整个文档的标签。对应的结束标签只是在标签名称前面加一条斜线，它没有属性。

根据HTML标准，少数标签没有结束标签。实际上，这个标准禁止对一些特殊标签使用结束标签，但是大多数浏览器都能够容忍和忽略那些错误的结束标签。例如<br>标签将产生换行；它对文档随后的部分没有任何影响，因此也不需要结束标签。

在HTML中没有相应的结束标签的标签有：

```
<area><base><basefont><br><col><frame><hr><img><input><isindex><link>
<meta><param>
```

## 标签的嵌套

标签可以放在另外一个标签（嵌套的）所影响的片段中，以实现对其一段文档的多重标签效果。例如，下面这段文字中的某部分既被加粗了，又是被<a>标签定义的一个锚：

```
<a href = "http://www.21books.com">21books<b>中国青年出版社</b></a>
```

根据HTML标准，必须先结束最靠近嵌套标签内容的标签，然后按照由内及外的顺序依次进行。例如在前面的示例中，先结束粗体标签（</b>），然后再结束链接标签（</a>），因为是以相反的顺序开始嵌套它们的：先是<a>标签，然后是<b>标签。虽然多数浏览器并不会绝对按照这个标准去做，但最好遵守它。不合理的嵌套可能在一个甚至所有现在使用的浏览器中通过，但如果某个浏览器的新版本不再允许这种违反标准的做法，那么改动源HTML文档的工作量将会非常巨大。

## HTML中的属性

属性可对标签进行补充说明。例如在一行文本中采用标题标签<h1>，代码如下。

```
<h1>中国青年出版社</h1>
```

接着可以用属性进一步指定标题要居中，代码如下。

```
<h1 Align="center">中国青年出版社</h1>
```

所有属性都放在起始标签的尖括号内并相互用空格分开。有些属性要求用引号，有些则不要。作为一般原则，大多数属性（只包括字母、数字、连字符和点号的）都不需要引号。例如，可以输入Align=Center或Align="Center"，所有浏览器都能用相同方式显示。

但是包括空格、%、#等其他字符的属性值则需要引号。例如用Width属性表示文档窗口大小的百分数，则要输入Width="100%"。

在标签中可以放上多个属性，相互间用空格分开，例如：

```
<h1 align="center" size="#FF0000">中国青年出版社</h1>
```

## 标签属性范例

下面是一些具有属性的标签。

### 1. 范例代码

```
<a href = "http://www.21books.com">
```

```
<ul compact>
```

```
<link title="Table of Content">
```

```
<input type=text name=filename  
size=24 maxlength=80>
```

### 2. 范例解释

01 指向中国青年出版社的网站首页,<a> 标签只有一个属性: href。这个属性后面紧跟着网址,也就是它的URL。

02 它的作用是把文本按照无序项目列表的形式进行格式化。它的惟一属性 compact 限定了列表项之间的距离,该属性无需赋值。

03 显示了当属性的值多于一个词时,应该怎样给它们加双引号。

04 显示了一个具有多个属性的HTML 标签,其中每一个属性都有一个值,但是不需要加引号。

在上面这些示例中,没有马上明确说明的是,虽然在HTML 中属性名称不区分大小写(在HTML 中 href、HREF 和 HreF 的效果都一样),但大多数属性的值却是区分大小写的,例如在<input>标签中,Name 属性的值 filename 就不同于 Filename。





# Chapter

# 3

## 文档的顶级信息

顶级信息向浏览器提供关于文档特性的信息，例如所用HTML的版本、文档的介绍性信息和标题等。结构标志虽然也是HTML文档的一部分，但大部分不显示在浏览器窗口中，而是在后台工作，指示浏览器要放上哪些元素和如何显示这些元素。这些标签并不能在Web页面中产生动人的效果，而是告诉浏览器如何解析文档的重要信息。

```
meta http-equiv="content-type"
charset="gb2312"
<link href="1_files/css.css" rel="stylesheet" type="text/css">
```



## 1 根元素

所有HTML文档都应当遵循HTML SGML DTD，即定义HTML标准的正式文档类型定义(Document Type Definition, DTD)。这个DTD定义了用来创建HTML文档的标签和语法。通过在文档的第一行放置一个特殊的SGML(Standard Generalized Markup Language, 标准通用标记语言)命令，便可以告诉浏览器文档遵循的是哪种DTD。

<!doctype>标签向浏览器(和验证服务)说明文档遵循的HTML版本，HTML 3.2和4规范都要求文档具备这个标签，因此应将其放在所有文档中。一般将其放在文档的开头，例如：

```
<!doctype HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

<!doctype>标签的关键部分是DTD(文档类型定义)元素，它向浏览器(和验证服务)说明文档遵循的HTML版本和签发规范的组织。上段代码说明，文档将遵从由万维网联盟(World Wide Web, W3C)定义的HTML 4.01的最终DTD。DTD的其他版本定义了局限性更多的HTML标准的版本，因此并不是所有浏览器都支持HTML DTD的所有版本。实际上，如果指定了其他文档类型，那么浏览器可能会错误地显示文档。

### Html 标签

<html>和</html>标签限定了文档的开始和结束点。

基本语法	<html>	
	包含<head>、<body>、<frameset>等标签	
	</html>	
功能	界定一个完整的HTML文档	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Version	定义当前文档使用的DTD信息

dir 属性指定了浏览器该用什么方向来显示包含在元素中的文本。将它用于<html>标签中时，即决定了文本在整个文档中将以什么方向显示。将它用在其他标签中时，则只决定那个标签中内容的显示方向。这个属性的默认值是ltr，意味着文本将会从左到右显示给用户。如果用其他值，rtl 则从右到左显示文本，如古中文和希伯来文。当然，显示的结果还要看文档的内容和浏览器对HTML 4的支持程度。Netscape Navigator 和Internet Explorer 4.0及以前的版本都会忽略dir 属性。

lang 属性如果包含在<html>标签中，那么就可以指定整个文档所使用的语言。如果用在其他标签中，则此属性将指出那个标签内容中所使用的语言。理想情况下，浏览器会使用lang 属性将文本更好地显示给用户。用户可以把lang 属性的值设置成一个ISO-639标准的双字节语言代码；还可以通过带有短线和子码名称的ISO语言代码，来指定这个标准中的语言。例如，en 在ISO语言代码中代表英语，zh 代表中文。

Version 属性定义了用于创作文档的 HTML 标准的版本。对于 HTML 4.01 版本来说，它的值的准确表示如下：

```
version="-//W3C//DTD HTML 4.01//EN"
```

一般来说，<html> 标签内的版本信息所带来的麻烦比好处还多，这个属性在 HTML 4 中已经被废弃了。严谨的设计者应该使用 SGML 的 <!doctype> 标签来代替它，并把这个标签放在文档的最开始，例如下面的这段代码。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

范例文件	Sample\Ch03\1\1.html
学习要点	使用<html>标签

### 1. 范例代码

```
<!DOCTYPE HTML PUBLIC "-//W3C//
DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/
loose.dtd">
<html>
<head>
  <!-- 文档头部 -->
</head>
<body>
  文档主体
</body>
</html>
```

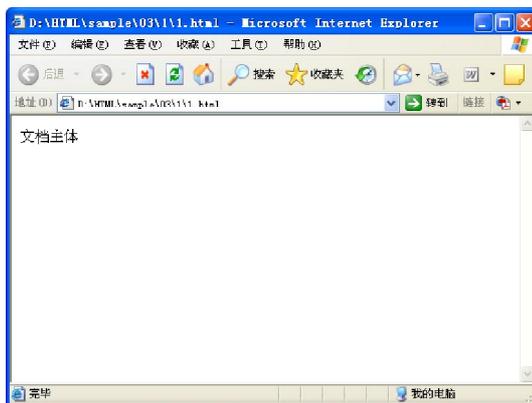
### 2. 范例解释

- 01 声明 HTML 版本为 W3C 制订的 4.01 版。
- 02 声明 HTML 文件的开始。
- 03 声明 HTML 文件的结束。

### 3. 代码和设计



### 4. 显示结果





## Head 标签

<head> 标签用来封装其他位于文档头部的标签。把该标签放在文档的开始处，紧跟在 <html> 标签后面，并位于 <body> 标签或 <frameset> 标签之前。不论是 <head> 标签还是相应的结束标签 </head>，都可以清楚地被浏览器识别出来，因此使用 HTML 时，可以在文档中安全地省略它们。但是建议读者还是把它们包含在文档中，这样更有利于提高文档的可读性，并支持文档的自动化操作。

基本语法	<head>	
	包含头部内容	
	</head>	
功能	定义文档的头部	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Profile	一个或多个文档元数据信息的地址

正如在关于 <html> 标签属性一节中提到的那样，Dir 和 Lang 属性有助于将 HTML 展示给全世界的浏览者。

文档的头部经常会包含一些 <meta> 标签，用来告诉浏览器关于文档的附加信息。未来设计者们可能会利用预先定义好的标准文档的元数据 (metadata) 配置文件 (profile)，以便更好地描述文档。Profile 属性提供了与当前文档相关联的配置文件的 URL。配置文件的格式以及浏览器使用它的方式都还没有进行定义，这个属性主要是为将来的开发而保留的占位符。

范例文件	Sample\Ch03\1\2.html
学习要点	使用 <head> 标签

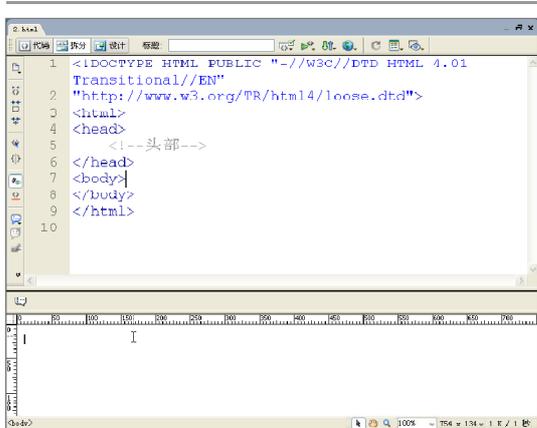
### 1. 范例代码

```
<!DOCTYPE HTML PUBLIC "-//W3C//
DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/
loose.dtd">
<html>
<head>
  <!-- 头部 -->
</head>
<body>
  </body>
</html>
```

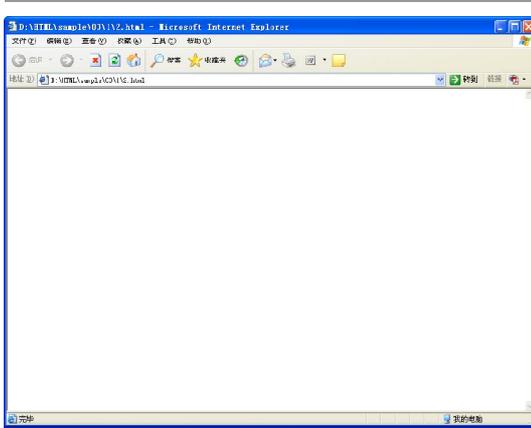
### 2. 范例解释

- 01 声明头部内容的开始。
- 02 声明头部内容的结束，两行代码之间放置头部的具体内容。

### 3. 代码和设计



### 4. 显示结果



## Body 标签

`<body>` 标签界定了文档的主体。在 `<body>` 标签和其结束标签 `</body>` 之间的所有部分都被称作主体内容。最简单的文档可能仅在 `<body>` 标签中有一系列文本段。复杂一些的文档可能会有格式非常繁杂的文本、图形图表、表格和其他各种特殊效果等。在 HTML 4 中, `<body>` 标签有很多属性来控制文档的颜色和背景等。不同的浏览器有其各自不同的扩展标签, 从而能够更好地控制文档的外观。

基本语法	<code>&lt;body attribute=" " &gt;</code>	
	包含主体内容	
功能	<code>&lt;/body&gt;</code>	
	定义文档的主体	
属性及说明	属性	说明
	Text	页面文字的颜色
	Bgcolor	页面背景的颜色
	Background	页面的背景图像
	Bgproperties	页面的背景图像是否固定
	Link	页面默认的连接颜色
	Alink	鼠标正在单击时候的连接颜色
	Vlink	访问过后的连接颜色
	Topmargin	页面上边距
Leftmargin	页面左边距	

### 背景颜色

将浏览器窗口的默认背景颜色改为另一个色调时, 一种标准但不推荐使用的方法是利用 `<body>` 标签的 `Bgcolor` 属性。 `Bgcolor` 属性需要的值可以用下面两种方法来表示: 所需颜色的红、绿、蓝 (RGB) 成分, 或者一个标准颜色名。本书附录 1 中详细列出了关于 RGB 颜色编码方面的内容, 包括 `bgcolor` 属性可以使用的十六进制 RGB 值和颜色。





原始文件	Sample\Ch03\1\3.html
最终文件	Sample\Ch03\1\3-end.html
学习要点	使用<body>标签的bgcolor属性

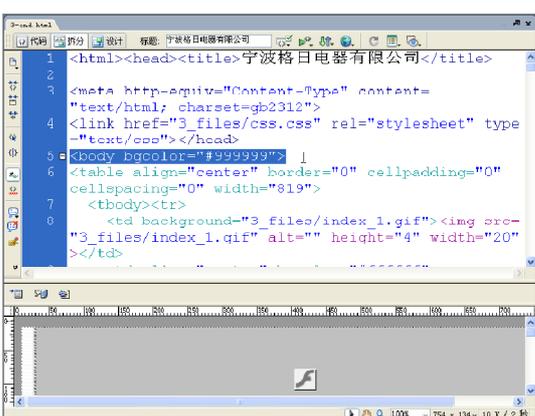
### 1. 范例代码

```
<html >
<head>
...
</head>
<body bgcolor="#999999">
...
</body>
</html >
```

### 2. 范例解释

设置背景颜色为 #999999 的灰色。999999 为十六进制颜色代码，前面加 # 号标明十六进制色彩。

### 3. 代码和设计



### 4. 显示结果



### 背景图像

如果单纯一种颜色不足以烘托效果，还可以通过设置<body>标签的 Background 属性在文档背景中放置一个图像。Background 属性需要的值是一个图像的 URL。浏览器会自动在水平和垂直方向上重复该图像（平铺）来填满整个窗口。通常应该选择小而暗淡的图像来创建一个有趣的背景图案，这样不会显得突兀。除此之外，小的图片在网络中浏览时，比错综复杂的全屏图片要快得多。

原始文件	Sample\Ch03\1\4.html
最终文件	Sample\Ch03\1\4-end.html
学习要点	使用<body>标签的background属性

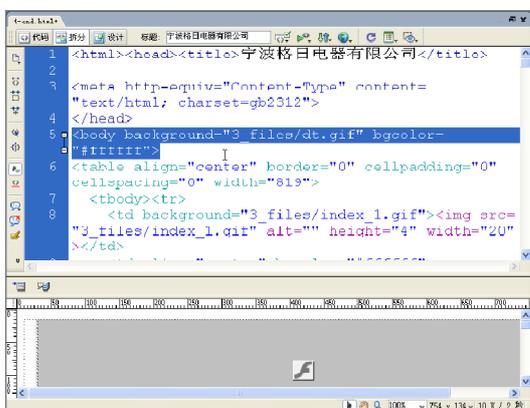
## 1. 范例代码

```
<html >
<head >
...
</head >
<body background="3_files/dt.gif"
bgcolor="#ffffff">
...
</body >
</html >
```

## 2. 范例解释

设置背景图像为“3\_files”文件夹下的dt.gif文件，默认情况下，该背景图像在浏览器中会以平铺的形式显示，在代码最后设置背景颜色为白色，这时使用十六进制值表示。

## 3. 代码和设计



## 4. 显示结果



## 背景图像固定

Internet Explorer和Netscape Navigator都支持<body>标签的Bgproperties属性，但它只有与Background属性一起使用时才有效。Bgproperties属性只有一个值fixed，它的作用是把背景图像冻结在浏览器窗口中。

原始文件	Sample\Ch03\1\5.html
最终文件	Sample\Ch03\1\5-end.html
学习要点	使用<body>标签的bgproperties属性

## 1. 范例代码

```
<html >
<head >
...
</head >
<body background="3_files/dt.gif"
bgcolor="#ffffff" bgproperties
="fixed">
...
</body >
</html >
```

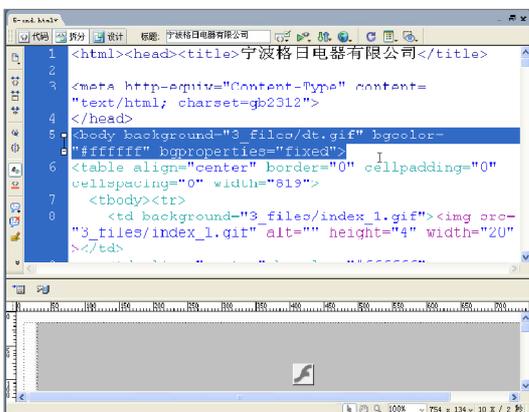
## 2. 范例解释

设置背景图像为3\_files文件夹下的dt.gif文件，并设置背景颜色为白色，然后通过设置Fixed属性值设置背景图像为固定，这样它就不会随着窗口内容的滚动而滚动了。





### 3. 代码和设计



### 4. 显示结果



#### 正文颜色

一旦改变了文档的背景颜色或添加了背景图像，还可能调整文本颜色才能确保用户可以阅读文本。<body>标签在HTML 4 中的Text（文本）标准属性是这样设置的：它设置整个文档中所有无链接文本的颜色。用与指定背景颜色相同的格式来为Text 属性指定一个颜色值，也就是一个RGB 组合或颜色名称，例如要制作一个黄褐色背景、蓝色文本的文档，可以用下列代码：

```
<body bgcolor="#777700" text="blue">
```

当然，最好选择与背景颜色或图像对比鲜明的文本颜色。HTML 4 中不推荐使用Text 属性，因为用样式表也可以得到同样的效果。

原始文件	Sample\Ch03\1\6.html
最终文件	Sample\Ch03\1\6-end.html
学习要点	使用<body>标签的text属性

#### 1. 范例代码

```

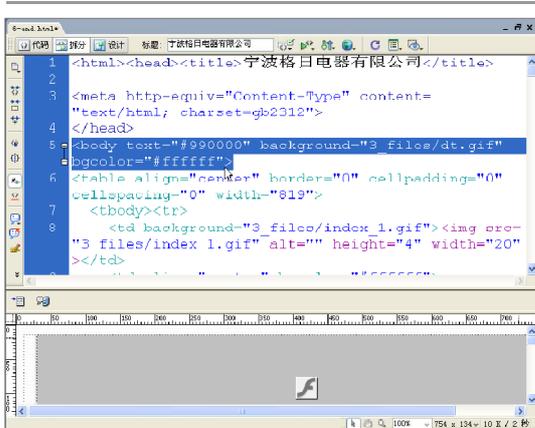
<html>
<head>
...
</head>
<body text="#990000" background="3_files/dt.gif" bgcolor="#ffffff">
...
</body>
</html>

```

#### 2. 范例解释

设置文字颜色为#990000的红色，然后设置背景图像为3\_files文件夹下的dt.gif文件，并在代码最后设置背景颜色为白色。

### 3. 代码和设计



### 4. 显示结果



#### 链接颜色

<body>标签的 Link、Vlink 和 Alink 属性控制着文档中超链接文本（<a>标签）的颜色。所有这3种属性与 Text 和 Bgcolor 属性一样，都接受将颜色指定为一个 RGB 组合或颜色名的值。Link 属性决定用户还没有单击过的所有超链接的颜色；Vlink 属性设置用户已经单击过的所有链接的颜色；Alink 属性则定义激活链接文本时的颜色，例如用户目前选择的链接，以及被鼠标按钮单击过鼠标指针下的链接。和文本颜色一样，也应该小心地选择链接相对应的颜色。此外，链接颜色应该与常规文本不同，并且链接之间的颜色也应该不同。HTML 4 中不推荐使用这些属性，因为用样式表也可以得到同样的效果。

原始文件	Sample\Ch03\1\7.html
最终文件	Sample\Ch03\1\7-end.html
学习要点	使用<body>标签的link、alink和vlink属性

#### 1. 范例代码

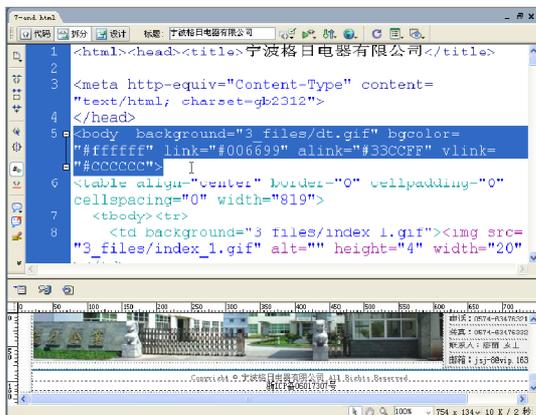
```
<html >
<head >
...
</head >
<body background="3_files/dt.gif" bgcolor="#ffffff" link
="#006699" alink="#33CCFF" vlink
="#CCCCCC">
...
</body >
</html >
```

#### 2. 范例解释

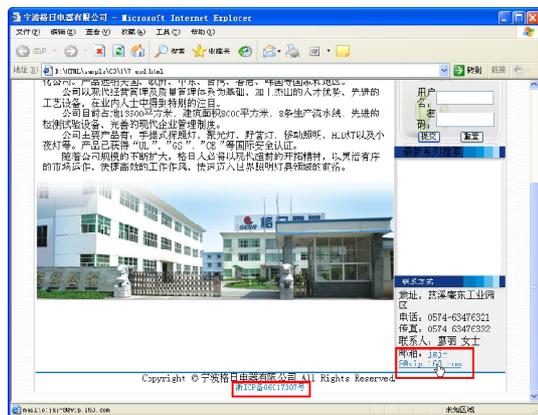
首先设置背景图像为3\_files文件夹下的dt.gif文件，并设置背景颜色为白色，最后分别设置默认的链接颜色为#006699的深蓝色，激活状态的链接颜色为#33CCFF的浅蓝色，访问过后的链接颜色为#CCCCCC的灰色。



### 3. 代码和设计



### 4. 显示结果



#### 边距

Leftmargin和Topmargin是Internet Explorer的专有属性,<body>标签的Leftmargin属性允许相对于浏览器窗口的左边边缘缩进至左边的边界,如一张纸的边界一样,其他浏览器则会忽略这个属性,并且通常紧接着文档窗口的左边将主体内容左对齐。Leftmargin属性值是左边边界缩进的像素的整数值,0为默认值。而边界则是用背景颜色或图像填充的。

和Leftmargin一样,目前Topmargin属性专属于Internet Explorer,它可以包含在<body>标签中,用来设定文档顶部的边界。边界空间是用文档的背景颜色或图像填充的,主体内容在Topmargin指定的像素数目以下的整数位置开始显示,0为默认值。

原始文件	Sample\Ch03\1\8.html
最终文件	Sample\Ch03\1\8-end.html
学习要点	使用<body>标签的leftmargin和topmargin属性

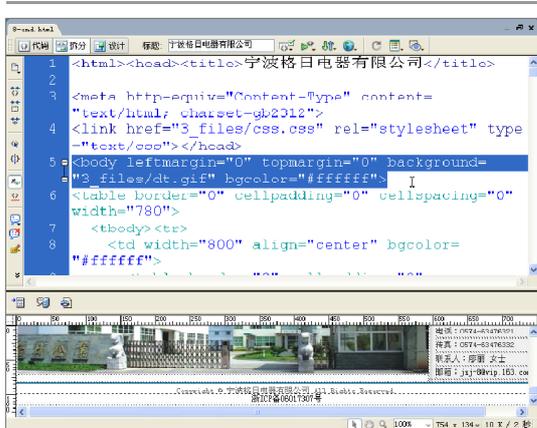
#### 1. 范例代码

```
<html >
<head >
...
</head >
<body leftmargin="0" topmargin="0" background="3_files/dt.gif" bgcolor="#ffffff">
...
</body >
</html >
```

#### 2. 范例解释

设置左边距为0,上边距为0,页面将贴紧浏览器的边缘,然后设置背景图像为3\_files文件夹下的dt.gif文件,最后设置背景颜色为白色。

### 3. 代码和设计



### 4. 显示结果



## 2 头部元素

头部元素<head>标签中可以包含一些用来定义和管理文档内容的其他标签。它包括（可以任意次序出现）：<base>、<basefont>、<i si ndex>、<l ink>、<meta>、<next id>、<scri pt>、<style>和<ti tle>，如下表所示。

标签	描述
<base>	当前文档的URL 全称（基底网址）
<basefont>	设定基准的文字字体、字号和颜色
<title>	设定显示在浏览器左上方的标题内容
<meta>	有关文档本身的元信息，例如用于查询的关键词等
<style>	设定CSS层叠样式表的内容
<l ink>	设定外部文件的链接
<scri pt>	设定页面中程序脚本的内容

### Title标签

<title>标签的功能是在其开始和结束标签之间定义文档的标题。

<title>标签是<head>标签中惟一要求包含的东西。由于<head>标签本身，甚至是<html>标签都可能被安全地忽略，因此<title>标签可能是一个合法HTML 文档的第一行。更进一步来说，多数浏览器都会为缺少<title>标签的文档自动指定一个通用标题例如文档的文件名等，所以用户甚至可以不用提供标题。浏览器不会特意对标题的文本进行格式化处理，它们将忽略在标题开始标签和结束标签之间除文本以外的内容，例如忽略图像或者到其他文档的链接。



3

文档的顶级信息



基本语法	<code>&lt;title&gt;</code>	
	包含标题内容	
	<code>&lt;/title&gt;</code>	
功能	定义文档标题	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息

正如在关于<html>标签属性一节中提到的那样，Dir 和 Lang 属性有助于将 HTML 展示给全世界的浏览者。

原始文件	Sample\Ch03\2\1.html
最终文件	Sample\Ch03\2\1-end.html
学习要点	使用<title>标签

### 1. 范例代码

```

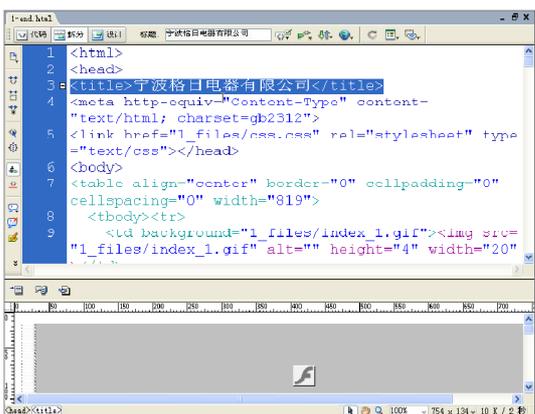
<html>
<head>
<title>宁波格日电器有限公司</title>
</head>
<body>
...
</body>
</html>

```

### 2. 范例解释

设置页面标题为“宁波格日电器有限公司”。浏览器会以特殊的方式来使用这个标题：通常把它放置在浏览器窗口的标题栏上；当把文档加入用户的链接列表、收藏夹或书签列表中时，标题将成为该文档链接的默认名称。

### 3. 代码和设计



### 4. 显示结果



## Base 标签

文档中的 URL 既可以是绝对地址（由制作者明确提供 URL 的所有元素），也可以是相对地址（URL 的某些元素被省略，而浏览器会提供这些被省略的元素）。通常情况下，浏览器会从当前文档的 URL 中提取相应的元素来填写相对地址中的空白。使用<base>标签可以改变

这一点, <base> 标签应该只出现在文档的头部, 而不应该出现在文档的主体内容中。浏览器随后将不再使用当前文档的 URL, 而使用指定的基本 URL 来解析所有的相对地址, 这其中包括 <a>、<img>、<link>、<form> 标签中的 URL。它同时还定义了用于在带有 <isindex> 标签的可搜索文档中解析查询的 URL。

基本语法	<basehref="URL" target="window_name">	
功能	为文档中的其他锚定义基本URL	
属性及说明	属性	说明
	Href	指定文档的基础URL地址
	Target	定义打开页面的窗口

href 属性必须有一个有效的 URL 作为它的值, 浏览器随后会使用该值定义基于文档中出现的相对地址的绝对地址。例如, 下列代码是在 HTML 文档头部的 <base> 标签。

```
<head>
<base href="http://www.21books.com">
</head>
```

这段代码将告知浏览器, 此文档中的所有相对地址都和 [www.21books.com](http://www.21books.com) 上的顶级文档目录相关, 无论用户是从什么地址和目录中检索到当前文档。

基本 URL 也可以制定为相对地址而不是绝对地址。浏览器应该 (但并不总是) 用文档本身的 URL 来填上相对地址中缺少的部分, 从而形成绝对的基本 URL。这种属性具有很多优点。例如, 在下面的 HTML 示例中:

```
<head>
<base href="/news/">
</head>
```

浏览器让 <base> 标签中的 URL 成为与服务器的 “/news” 目录相对的 URL, 而该目录可能并不是当前文档的同一目录。<base> 标签不仅可以帮助设计者缩短文档中同一根目录下的 URL 的长度, 还允许设计者限定从其中检索相对引用的目录, 而无须将文档绑定到特定的服务器上。在使用框架内的文档时, <a> 标签中的 Target 属性确保被引用的 URL 能够加载到正确的框架中。同样, 对于浏览器要显示的重定向的超链接文档, <base> 标签中的 Target 属性允许为显示它们的某个框架或窗口创建默认名称。如果框架中的超链接没有其他默认目标, 就应该考虑使用 <base target=top>, 这就确保了在链接没有特别指向某个框架或窗口的情况下, 随后会被加载到顶级浏览器窗口中。这样就不会出现在自己页面的框架中引用了其他站点的页面, 而不是显示自己的页面这种尴尬的局面和常见错误了。虽然这只是 HTML 中的一个很小的技巧, 但它确实可以为阅读者带来更多的方便。Target 的属性值如下表所示。

属性值	描述
_parent	在上一级窗口中打开。一般使用分帧的框架页会经常使用
_blank	在新窗口中打开
_self	在同一个帧或窗口中打开, 这项一般不用设置
_top	在浏览器的整个窗口中打开, 忽略任何框架





## Basefont 标签

<basefont>标签可以设定基准的文字字体、字号和颜色，一旦遇到页面中其他相关标签未定义文字或段落的样式时，将套用基准的文字样式。

基本语法	<basefont face="font_name, font_name, ..." size="value" color="value">	
功能	定义基准文字样式	
属性及说明	属性	说明
	Face	字体
	Size	字号
	Color	颜色

Face 定义了字体，不同的字体可以定义多次，字体之间使用“，”分开，如果第一种字体系统中不存在，就显示第二种字体，如果字体都不存在，就显示默认的字体。Size 属性定义了字号，Value 取值范围为从 1 到 7 或者从 +1 到 +7、从 -1 到 -7。1 是最小的字号，7 是最大的字号。Color 定义了颜色，Value 定义颜色的名称或者十六进制代码。

原始文件	Sample\Ch03\2\2.html
最终文件	Sample\Ch03\2\2-end.html
学习要点	使用<basefont>标签

### 1. 范例代码

```

<html >
<head>
...
<basefont face=" 黑体" color
="#0066FF" size="3">
...
</head>
<body>
...
宁波格日电器有限公司是一家集科研、开发
及生产灯具和小家电产品的专业化公司.....
...
</body>
</html >

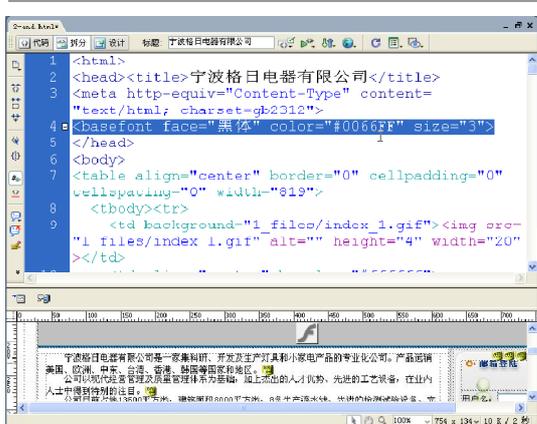
```

### 2. 范例解释

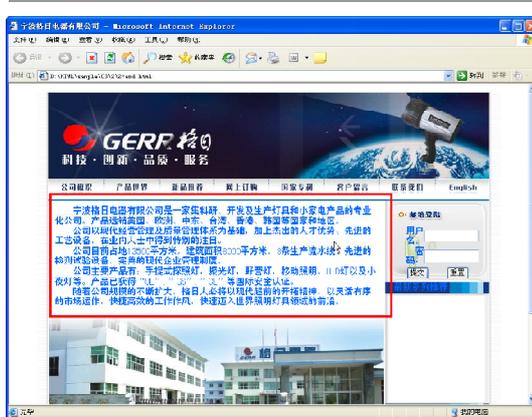
01 设定的基准文字是黑体、3 号字、#0066FF 色。

02 这是未设置样式的文字，由于基准文字字体、字号和颜色的影响，这段未定义文字或段落的样式将套用基准的文字样式。

### 3. 代码和设计



### 4. 显示结果



## Meta 标签

设计者希望能够提供有关他们宝贵文档的更多信息，其中包括浏览器、源文档或文档索引工具软件等可能要使用到的各种信息。<meta>标签正好符合这类设计者的需求。<meta>标签放置在文档头部，不包含任何内容。<meta>标签的属性定义了与文档相关联的名称/值对。在某些情况下，这些值被 Web 服务器用于提供文档，以进一步向浏览器定义文档的内容类型。

基本语法	<meta name="GENERATOR" content="VALUE" Http-equiv="VALUE">	
功能	提供文档的附加信息	
属性及说明	属性	说明
	Http-Equiv	生成一个Http标题域，它的取值与Content属性相同，例如Http-Equiv=Expires，实际取值由Content确定
	Name	如果元数据是以关键字/取值的形式出现的，则NAME表示关键字，如AUTHOR或ID
	Content	关键字/取值的内容

Name 属性提供了由<meta>标签定义的名称/值对中的名称。HTML 标准没有指定任何预先定义的<meta>名称。通常情况下，可以自由使用对自己和源文档的阅读者来说都富有意义的名称。经常使用的一个名称是 keywords，它为文档定义了一组关键字。当 Web 上任何常用的搜索引擎遇到关键字时，都会用这些关键字对文档进行分类。如果希望自己的文档可以进入搜索引擎的索引中，就要考虑在每个文档的<head>中放置<meta>标签，例如下面的代码。

```
<meta name="keywords" content=" 图书, 培训, 设计 ">
```

如果没有提供 Name 属性，那么名称/值对中的名称会采用 Http-equiv 属性的值。Content 属性提供了名称/值对中的值。该值可以是任何有效的字符串（如果值中包含空格，就要用引号括起来）。Content 属性始终要和 Name 或 Http-equiv 属性一起使用。例如可以将设计者的名字放在文档中：

```
<meta name="Authors" content="LaoHu">
```



Http-equiv属性为名称/值对提供了名称,并指示服务器在发送实际的文档之前先要在要传送给浏览器的MIME文档头部包含名称/值对。当服务器向浏览器发送文档时,会先发送许多名称/值对。虽然有些服务器会发送许多这种名称/值对,但是所有服务器都至少要发送一个名称/值对:

```
content-type: text/html
```

这将告知浏览器准备接收一个HTML文档。使用带有Http-equiv属性的<meta>标签时,服务器将把名称/值对添加到发送给浏览器的内容头部中。例如添加如下代码:

```
<meta http-equiv="charset" content="iso-8859-1">
<meta http-equiv="expires" content="31 Dec 08">
```

这样发送到浏览器的头部就应该包含:

```
Content-type: text/html
charset: iso 8859 1
expires: 31 Dec 08
```

当然,只有在浏览器接受这些附加的头部字段,并且能以适当的方式使用它们时,这些字段才有意义。

范例文件	Sample\Ch03\2\3.html
学习要点	使用<meta>标签定义编辑器名称

使用FrontPage、Dreamweaver等多种编辑工具可以制作网页,在源代码中可以设定网页编辑器的名称,该名称不会显示在浏览器中。

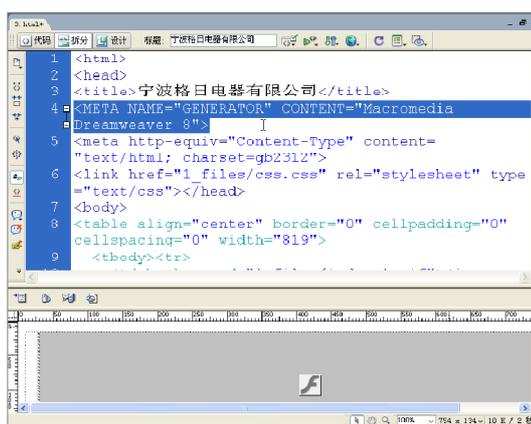
## 1. 范例代码与解释

```
<html>
<head>
...
<META NAME="GENERATOR" CONTENT="Macromedia Dreamweaver 8">
...
</head>
<body>
...
</body>
</html>
```

GENERATOR为编辑器名称,CONTENT中定义编辑器的名称。

范例文件	Sample\Ch03\2\4.html
学习要点	使用<meta>标签定义关键字

## 2. 代码和设计



关键字是给搜索引擎提供的，例如一个电器网站，为了提高在搜索引擎中被搜索到的几率，可以设定多个和电器主题相关的关键字以便搜索。需要注意的是，大多数搜索引擎检索时都会限制关键字的数量，有时关键字过多该网页会在检索中被忽略。

## 1. 范例代码与解释

```
<html >
<head >
...
<META NAME="KEYWORDS" CONTENT="
电器, 产品, 家电 ">
...
</head >
<body >
...
</body >
</html >
```

KEYWORDS 为关键字定义，CONTENT 中定义关键字的内容。

范例文件	Sample\Ch03\2\5.html
学习要点	使用<meta>标签定义说明

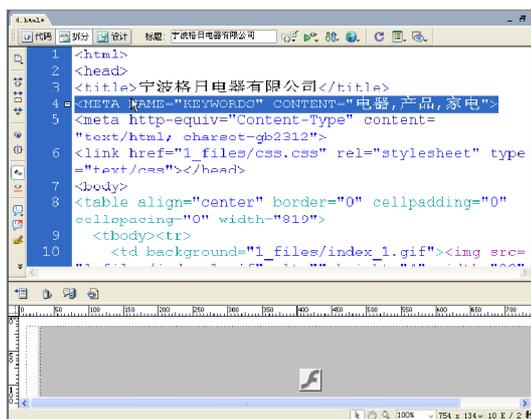
对于一个网站的页面，可以在源代码中添加说明语句，以便将网站的主题描述清晰，这就是描述语句的作用。这个描述语句内容也不会显示在浏览器中。说明文字可供搜索引擎寻找网页，还可存储在搜索引擎的服务器中以便浏览者搜索时随时调用，还可以在检索到网页时作为检索结果返给浏览者。搜索引擎同样限制说明文字的字数，所以内容应尽量简明扼要。

## 1. 范例代码与解释

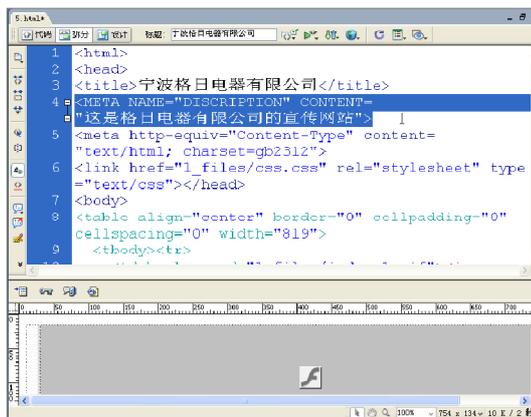
```
<html >
<head >
...
<META NAME="DISCRIPTION" CONTENT
=" 这是格日电器有限公司的宣传网站 ">
...
</head >
<body >
...
</body >
</html >
```

DISCRIPTION 为定义描述，CONTENT 中定义描述的内容。

## 2. 代码和设计



## 2. 代码和设计





范例文件	Sample\Ch03\2\6.html
学习要点	使用<meta>标签定义设计者

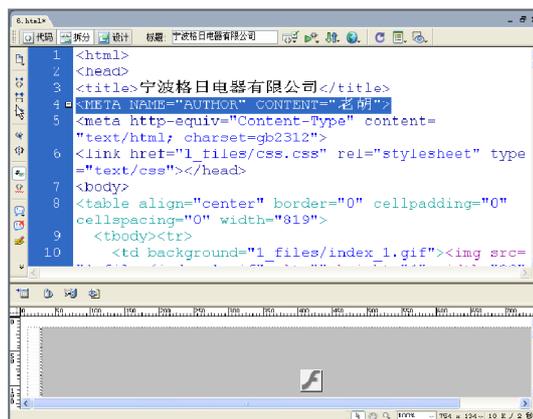
在页面的源代码中，可以显示页面设计者的姓名及个人信息。这可以在源代码中保留设计者希望保留的信息。

## 1. 范例代码与解释

```
<html >
<head>
...
<META NAME="AUTHOR" CONTENT="老胡" >
...
</head>
<body>
...
</body>
</html >
```

AUTHOR 为设计者定义，CONTENT 中定义设计者的个人信息。

## 2. 代码和设计



范例文件	Sample\Ch03\2\7.html
学习要点	使用<meta>标签定义字符集

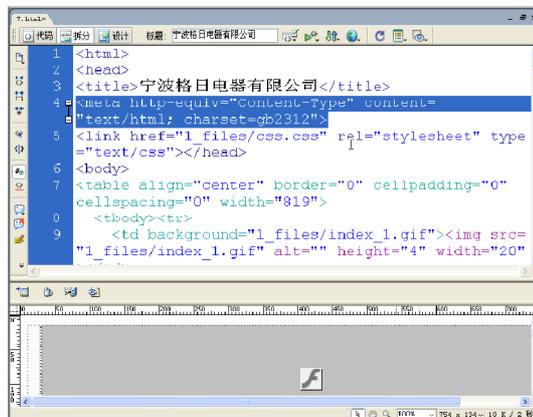
HTML 页面的内容可以用不同的字符集来显示，如中国常用的 GB 码（简体中文），欧洲常用的 ISO8859-1（英文）等。如果浏览器不能显示某个字符集，则浏览器中显示的都是乱码。这时就需要用 HTML 语言来定义页面的字符集，告诉浏览器以相应的内码显示页面内容。

## 1. 范例代码与解释

```
<html >
<head>
...
<META HTTP-EQUIV="CONTENT-TYPE"
CONTENT="TEXT/HTML; CHARSET=gb2312" >
...
</head>
<body>
...
</body>
</html >
```

HTTP-EQUIV 传送 HTTP 通信协议的标头，CONTENT 中定义页面的内码，在 CHARSET 中可以设置内码的语系。

## 2. 代码和设计



范例文件	Sample\Ch03\2\8.html
学习要点	使用<meta>标签定义刷新

使用Http-Equiv属性中的Refresh属性值能够设定页面的自动刷新,就是每隔几秒的时间后刷新页面的内容,例如常用的互联网现场图文直播、论坛消息的自动更新等。

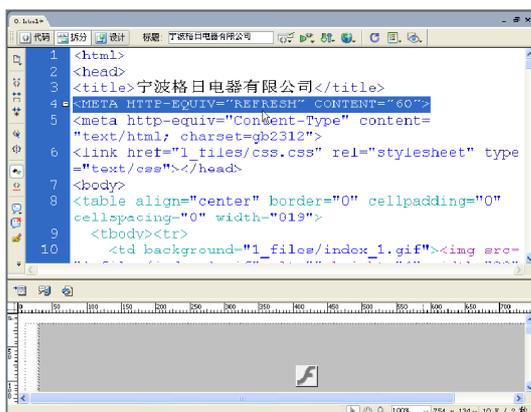
### 1. 范例代码

```
<html >
<head>
...
<META HTTP-EQUIV="REFRESH" CON-
TENT = "60">
...
</head>
<body>
...
</body>
</html >
```

### 2. 范例解释

HTTP-EQUIV 传送 HTTP 通信协议的标头, REFRESH 代表刷新, CONTENT 中定义刷新间隔的秒数为 60 秒。

### 3. 代码和设计



### 4. 显示结果



范例文件	Sample\Ch03\2\9.html
学习要点	使用<meta>标签定义跳转

使用Http-Equiv属性中的Refresh不仅能够完成页面自身的自动刷新,也可以实现页面之间的跳转过程。希望在当前的页面中等待几秒钟之后就自动跳转到新的网站地址,这就可以通过设定跳转时间和地址来实现。自动跳转特性目前已经被越来越多的网页所使用。





### 1. 范例代码与解释

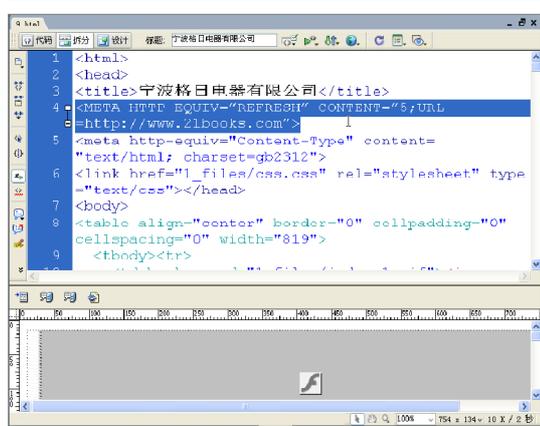
```

<html >
<head>
<title>宁波格日电器有限公司</title>
<META HTTP-EQUIV="REFRESH" CONTENT="5;URL=http://www.21books.com">
...
</head>
<body>
...
</body>
</html >

```

HTTP-EQUIV 传送 HTTP 通信协议的标头，REFRESH 代表刷新，CONTENT 中定义跳转间隔的秒数以及跳转后打开的文件地址。这里设定页面的跳转时间为 5 秒，跳转的地址是 <http://www.21books.com>。

### 2. 代码和设计



### 3. 显示结果



范例文件	Sample\Ch03\2\10.html
学习要点	使用<meta>标签定义有效期

通过设置Http-equiv属性的Expires属性值可以控制网页的有效期限。

## 1. 范例代码与解释

```

<html >
<head >
...
<META HTTP-EQUIV="expires" con-
tent ="Sun, 31 Dec 2006 23 59 59
GMT">
...
</head >
<body >
...
</body >
</html >

```

expires为定义网页有效期限, content中定义到期时间, 其格式为“星期, 日 月 年 时 分 秒 GMT”, 使用英文和数字进行设定。

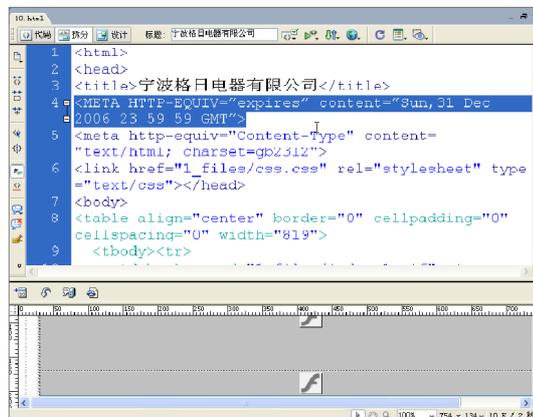
范例文件

Sample\Ch03\2\11.html

学习要点

使用&lt;meta&gt;标签定义网页过渡效果

## 2. 代码和设计



3

文档的顶级信息

网页过渡是指当进入或离开网站时, 页面具有不同的切换效果。添加此功能可以使网页看起来更具有动感。

基本语法	<meta http-equiv="event" content="reveal trans(duration=value, transition=number)">	
功能	定义网页过渡效果	
Event属性值及说明	属性值	说明
	Page-Enter	表示进入网页时有网页过渡效果
	Page-Exit	表示退出网页时有网页过渡效果
Transition属性值及说明	属性值	说明
	0	盒状收缩
	1	盒状展开
	2	圆形收缩
	3	圆形展开
	4	向上擦除
	5	向下擦除
	6	向左擦除
	7	向右擦除
	8	垂直百页窗
	9	水平百页窗
	10	横向棋盘式
	11	纵向棋盘式
	12	溶解
	13	左右向中部收缩
14	中部向左右展开	



(续表)

	属性值	说明
Transition属性值及说明	15	上下向中部收缩
	16	中部向上下展开
	17	阶梯状向左下展开
	18	阶梯状向左上展开
	19	阶梯状向右下展开
	20	阶梯状向右上展开
	21	随机水平线
	22	随机垂直线
	23	随机

Duration中定义网页过渡效果的延续时间,单位为秒,Transition中定义过渡效果的方式编号。

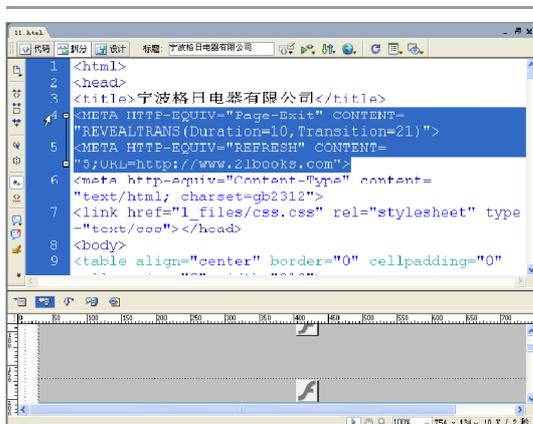
### 1. 范例代码

```
<html >
<head>
<title>宁波格日电器有限公司</title>
<META HTTP-EQUIV="Page-Exit" CON-
TENT ="REVEALTRANS(Duration=10,
Transition=21)" >
<META HTTP-EQUIV="REFRESH" CON-
TENT ="5;URL=http://www.21books.
com">
...
</head>
<body>
...
</body>
</html >
```

### 2. 范例解释

- 01 设定了退出页面时产生的切换转场效果,经过10秒钟的时间,使用编号为21的随机水平线效果。
- 02 设置了跳转语句,用来测试页面过渡产生的效果,即等待10秒钟后,页面将以随机水平线效果的转场形式,跳转到http://www.21books.com的中国青年出版社的网站上去。

### 3. 代码和设计



### 4. 显示结果



## 3 实例制作

下面通过一个实例练习页面头部元信息的设置，以及页面主体部分整体属性的设置。

原始文件	Sample\Ch03\3\1.html
最终文件	Sample\Ch03\3\1-end.html
学习要点	使用头部信息和整体设置



原始效果



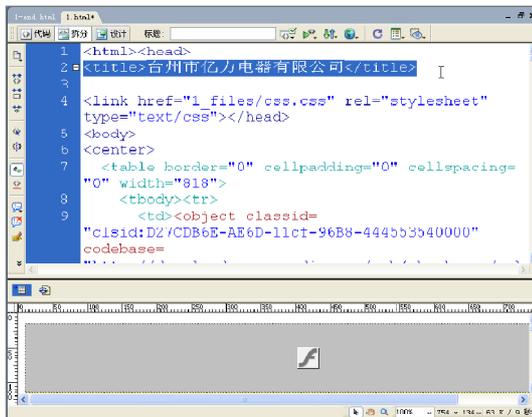
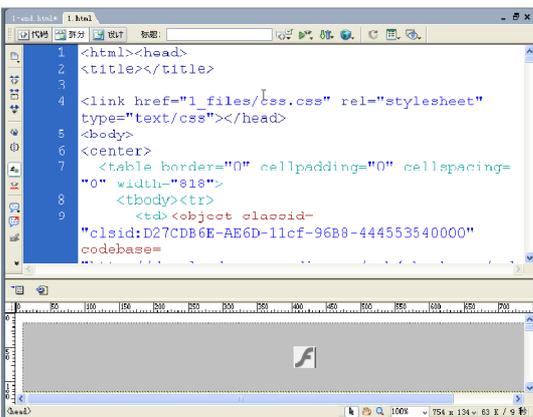
最终效果

1 在 Dreamweaver 中打开提供的页面，单击“拆分”按钮，切换到“代码视图”和设计视图”环境。

2 将第2行的<title></title>标记修改为以下代码：

```
<title>台州市亿力电器有限公司</title >
```

● 定义页面标题为“台州市亿力电器有限公司”。





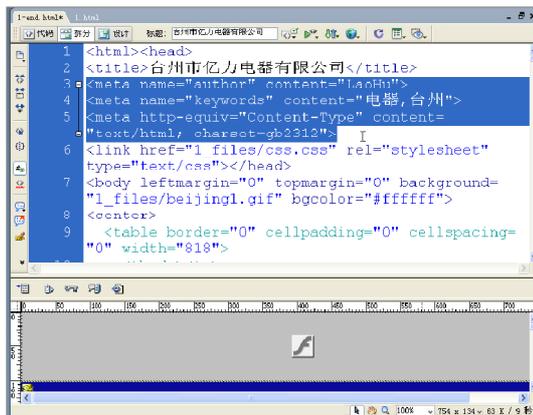
3 在第3行输入以下代码:

```

<meta name="author" content="LaoHu">
<meta name="keywords" content="电器, 台州">
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">

```

- 1 定义作者为LaoHu。
- 2 定义关键字为“电器，台州”。
- 3 定义字符集为简体中文。



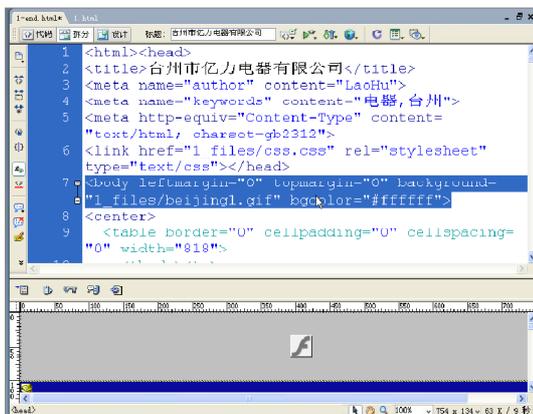
4 将<body>标签修改为以下代码:

```

<body leftmargin="0" topmargin="0" background="1_files/beijing1.gif" bgcolor="#ffffff">

```

- 1 定义左边距和顶边距为0，背景颜色为白色，背景图像为1\_files文件夹下的beijing1.gif文件。



5 按下F12快捷键，可以在浏览器中看到页面的效果。





# Chapter

# 4

## 文字和段落

一个优秀的网页设计应该把它的文本组织成一个有吸引力而且有效的文档，这正是HTML的优点。直接使用HTML标记是格式化文字的最快捷方法，例如可以使用标题2标记<h2>将一行文本设置为标题2格式。

meta http-equiv="content-type" charset="gb2312">  
<link href="1\_files/css.css" rel="stylesheet" type="text/css">



# 1 文字内容

在浏览器中显示的文字内容编写在<body>标签中，这些内容包括普通的文字、空格符号和特殊符号以及页面的注释语句。

## 空格

HTML 页面中空格符号是通过代码控制的。

基本语法	&nbsp;
功能	定义空格

原始文件	Sample\Ch04\1\1.html
最终文件	Sample\Ch04\1\1-end.html
学习要点	使用空格

### 1. 范例代码

```

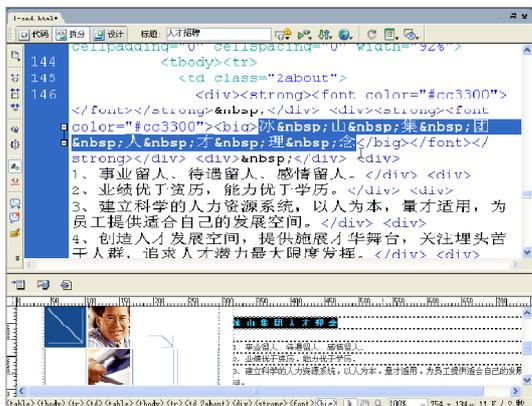
<html>
<head>
...
</head>
<body>
...
冰 &nbsp;&nbsp;山 &nbsp;&nbsp;集 &nbsp;&nbsp;团 &nbsp;&nbsp;
人 &nbsp;&nbsp;才 &nbsp;&nbsp;理 &nbsp;&nbsp;念
...
</body>
</html>

```

### 2. 范例解释

设置在每两个文字之间出现一个空格。

### 3. 代码和设计



### 4. 显示结果



## 特殊字符

文档中大部分不属于某标签的字符都会在浏览器中显示出来。但有些字符因为具有特殊的含义，而不能够直接显示，另外还有一些字符不能通过普通键盘直接输入到源文档中。特殊字符需要用特殊的名称或者数字字符来进行编码，从而使其能够包含在源文档中。

源文档中有 3 种字符具有特殊的含义：小于号 (<)、大于号 (>) 和 “&” 符号。这些字符界定了标签的范围和对特殊字符的引用。如果将它们随便放在文档中或者标签语法不正确，会使浏览器解析困难。所以得找出一种特殊的编码方式，把它们实际含义的符号包括在文档中。另外在需要包括一个键盘上没有的特殊字符，但却存在于 ISO Latin-1 字符集中并得到大多数浏览器支持的特殊字符时，也需要特殊的编码方式。

如果想把特殊字符包含在文档中，必须把字符的标准实体名称或者 “#” 符号加上它在 Latin-1 标准字符集里面的位置编号，并包含在一个 “&” 符号和分号之间，而且中间没有任何空格。常用的特殊字符如下表所示。

特殊符号	符号码	十进制字符编号
"	&quot;	&#34;
&	&amp;	&#38;
<	&lt;	&#60;
>	&gt;	&#62;
©	&copy;	&#169;
®	&reg;	&#174;
±	&plusmn;	&#177;
×	&times;	&#215;
§	&sect;	&#167;
¢	&cent;	&#162;
¥	&yen;	&#165;
•	&middot	&#183;
€	&euro;	&#8364;
£	&pound;	&#163;
™	&trade;	&#8482;

往网页中输入特殊字符时，需在 HTML 代码中加入以 “&” 开头的字母组合或以 “&#” 开头的十进制字符编号数字。例如，版权标志符 “©” 即是用 “©” 表示或 “&#169” 来表示。

原始文件	Sample\Ch04\1\2.html
最终文件	Sample\Ch04\1\2-end.html
学习要点	使用特殊字符





### 1. 范例代码

```

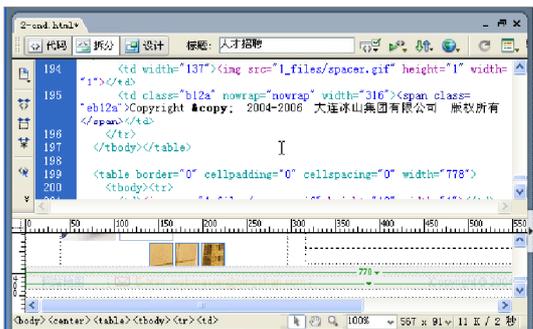
<html >
<head>
...
</head>
<body>
...
Copyright &#169; 2004-2006 大连
冰山集团有限公司 版权所有
...
</body>
</html >

```

### 2. 范例解释

设置在Copyright和2004-2006文字之间出现版权符号，编号为#169，包含在一个“&”符号和分号之间。

### 3. 代码和设计



### 4. 显示结果



## 注释

注释是另外一种文本内容，它出现在HTML源文档中，但浏览器并不显示它们。注释都放在特殊的“<!--”和“-->”标记元素中间。浏览器将忽略这段注释文本。下面是一段关于注释的例子：

```
<!-- 注释语句 -->
```

对于注释来说，最大的区别就是HTML不允许对它进行嵌套。Internet Explorer 允许把注释放在特殊的<comment>标签中。放在<comment>和</comment>标签之间的所有内容都会被Internet Explorer忽略，但是其他浏览器都将显示这些内容。由于浏览器可能出现这种不可预知的行为，因此不推荐用<comment>标签来标识注释，建议把注释放在“<!--”和“-->”中间。

基本语法	<!--...-->
功能	定义注释

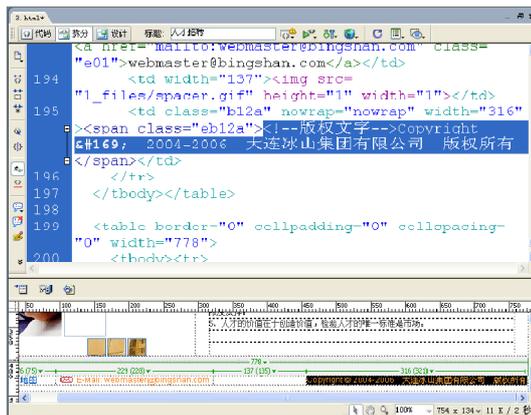
范例文件	Sample\Ch04\1\3.html
学习要点	使用注释

## 1. 范例代码与解释

```
<html >
<head >
...
</head >
<body >
...
<!-- 版权文字 -->Copyright &#169;
2004-2006 大连冰山集团有限公司 版
权所有
...
</body >
</html >
```

设置注释为“版权文字”。

## 2. 代码和设计



## 显示HTML代码标签

在HTML语言中加入`<plaintext>`或`<xmp>`标签，可以使该标签后的HTML标签失去其标识作用，而直接显示在网页页面中。

基本语法	<code>&lt;plaintext&gt;</code> <code>&lt;xmp&gt;...&lt;/xmp&gt;</code>
功能	将标签后或标签中的内容（包括HTML代码）原封不动地显示在浏览器页面中

原始文件	Sample\Ch04\1\4.html
最终文件	Sample\Ch04\1\4-end.html
学习要点	使用 <code>&lt;xmp&gt;</code> 标签

## 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<xmp><span class="eb12a"><!-- 版
权文字 -->Copyright &#169; 2004-
2006 大连冰山集团有限公司 版权所有</
span></xmp>
...
</body >
</html >
```

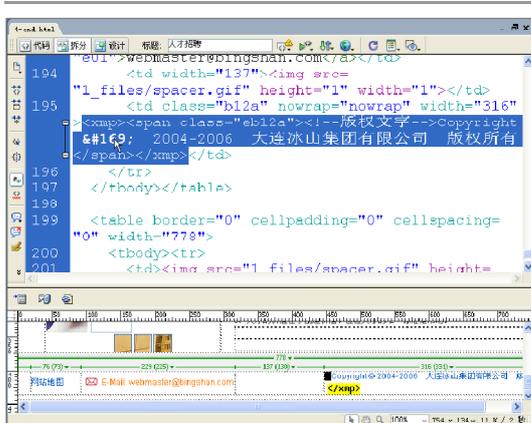
## 2. 范例解释

设定了忽略HTML标签的标记`<xmp>`，这时，标签内的“`<span class="eb12a"><!-- 版权文字 -->Copyright &#169; 2004-2006 大连冰山集团有限公司 版权所有</span>`”文字将被原封不动地显示在浏览器中。





### 3. 代码和设计



### 4. 显示结果



## 2 文字标记

如果希望更改页面中的字体、字号和颜色，普通的文字可以使用<font>标记，标题字可以使用<h>(n=1~6) 标记。

### Font 标签

<font>字体标签用于控制文字的字体、大小与颜色。控制的方式是利用属性设置实现。

基本语法	<font si ze="val ue" face="font_name, font_name, . . ." col or="val ue">	
	文字内容	
	</font>	
功能	定义文字的字体、字号和颜色。	
属性及说明	属性	说明
	Face	字体
	Si ze	字号
	Col or	颜色

原始文件	Sampl e\Ch04\2\1. html
最终文件	Sampl e\Ch04\2\1-end. html
学习要点	使用<font>标签的face属性

一般来说，不应该在网页中使用过于特殊的字体，因为对于浏览网页的人来说，其计算机中不一定会安装这些特殊字体，如果浏览到带有这些特殊字体格式的字符，只能以普通的默认字体来显示。对于中文网页来说，应该尽量对汉字使用宋体或黑体，因为大多数计算机中都默认安装有这两种字体。<font>标签中的 Face 属性定义了字体，不同的字体可以定义多次，字体之间使用“，”分开，如果第一种字体系统中不存在，就显示第二种字体，如果字体都不存在，就显示默认的字體。

## 1. 范例代码

```

<html >
<head >
...
</head >
<body >
...
<font face="隶书">冰山集团人才理念
</font >
...
</body >
</html >

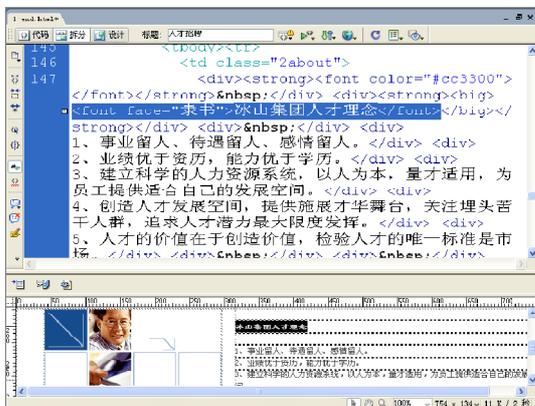
```

## 2. 范例解释

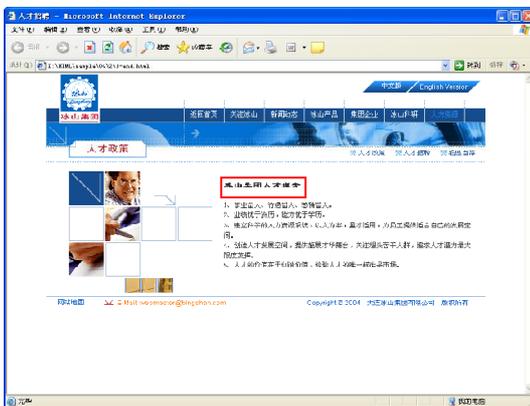
01 设置“冰山集团人才理念”文字使用隶书显示。

02 结束字体标签。

## 3. 代码和设计



## 4. 显示结果



原始文件	Sample\Ch04\2\2.html
最终文件	Sample\Ch04\2\2-end.html
学习要点	使用<font>标签的color属性

HTML 页面中的文字可以使用不同的颜色表现。丰富的字符颜色毫无疑问能够极大地增强文档的表现力。<font>标签中的Color属性用来定义文字颜色，Value属性用来定义颜色的名称或者十六进制值。

<font>标签的Color属性与<text>标签的Color属性不同，它设置的是特定文字的颜色，而<text>标签的Color属性设置的是整个页面的颜色。





## 1. 范例代码

```

<html >
<head>
...
</head>
<body>
...
<font face="隶书" color=" #CC0000">冰山集团人才理念
</font>
...
</body>
</html >

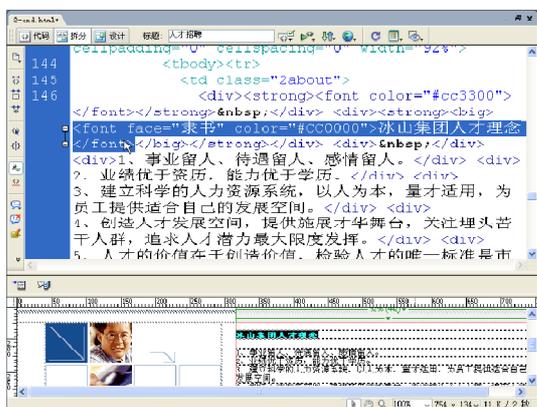
```

## 2. 范例解释

01 设置“冰山集团人才理念”文字使用隶书显示，并设置“冰山集团人才理念”文字的颜色为#CC0000的红色。

02 结束字体标签。

## 3. 代码和设计



## 4. 显示结果



原始文件	Sample\Ch04\2\3.html
最终文件	Sample\Ch04\2\3-end.html
学习要点	使用<font>标签的size属性

HTML 页面中的文字可以使用不同的字号表现。HTML 共有 7 种字号，1 号最小，7 号最大，默认字号为 3。字号指的是字体的大小，它没有一个绝对的大小标准，其大小只是相对于默认字体而言。例如，1 号和 2 号字比默认字体要小一些，而 4 号和 5 号字比默认字体要大一些。<font>标签中的 size 属性定义了字体，Value 取值范围为从 +1 到 +7 或者从 -1 到 -7。其中 1 是最小的字号，7 是最大的字号。

设置字号有两种办法，一是设置绝对字号，例如<font size=字号>; 另一种是设置文本的相对字号，例如<font size= ±n>。用第二种方法时“+”号表示字体变大，“-”号表示字体变小。

## 1. 范例代码

```

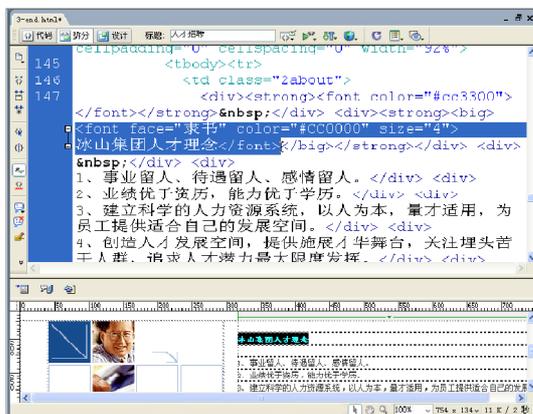
<html >
<head >
...
</head >
<body >
...
<font face="隶书" color="#CC0000" size="4">冰山集团人才理念
</font >
...
</body >
</html >

```

## 2. 范例解释

- 01 设置“冰山集团人才理念”文字使用隶书显示，并设置“冰山集团人才理念”文字的颜色为#CC0000的红色，最后设置“冰山集团人才理念”文字的字号为4。
- 02 结束字体标签。

## 3. 代码和设计



## 4. 显示结果



## Hn 标签

用户在读取屏幕上显示的信息时会很费劲。通篇文章如果没有用标题、副标题和其他标题区分开，这些信息就会麻木了浏览者的头脑，而要想在文档里找到某个特定的主题，那几乎就是不可能的事情。因此设计者应该用一个或者多个标题来把文本分割成不同的部分。HTML 的标题一共有 6 个级别，合理运用它们可以调整文档的结构，使其变得更易读，而且更容易管理。这 6 个标题标签分别是<h1>、<h2>、<h3>、<h4>、<h5>和<h6>，它们表示的是标题在文档中从最高（<h1>）到最低（<h6>）的优先顺序。



基本语法	<code>&lt;h n &gt;</code>	
	文字内容	
功能	<code>&lt;/h n &gt;</code>	
	定义6个级别标题中的一级	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Align	对齐方式
	Class	用一个名称来标记标题，该标记名称指向一个预定义类，而该类是在文档级声明的或者在外部定义的样式表
	Id	为段落创建一个标记，将来超链接可以用这个标记来明确地引用该段落，以便作为样式表选择器和使用其他应用程序来执行自动搜索
	Style	创建标题内容的内联样式
Title	给标题加上说明性的文字	

原始文件	Sample\Ch04\2\4.html
最终文件	Sample\Ch04\2\4-end.html
学习要点	使用<h n >标签

## 1. 范例代码

```

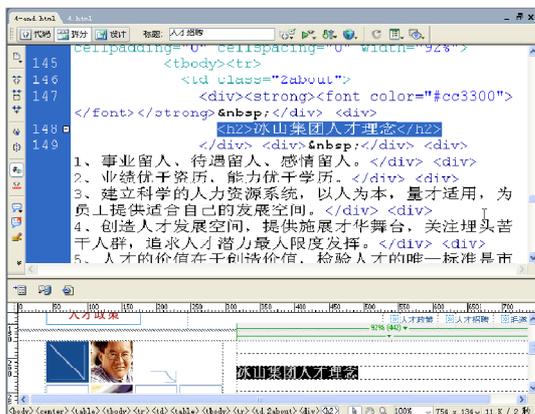
<html >
<head >
...
</head >
<body >
...
<h2>冰山集团人才理念</h2>
...
</body >
</html >

```

## 2. 范例解释

设置“冰山集团人才理念”文字为二级标题，将使用大一些的字号，并且以粗体显示文字。

## 3. 代码和设计



## 4. 显示结果



对大多数浏览器来说，标题的默认对齐（Align）属性是左对齐（Left），也可以把Align属性的对齐方式改为Center、Right，如下表所示。

Align属性值及说明	Align属性值	说明
	Left	标题居左对齐
	Right	标题居右对齐
	Center	标题居中对齐

原始文件	Sample\Ch04\2\5.html
最终文件	Sample\Ch04\2\5-end.html
学习要点	使用<Hn>标签的align属性



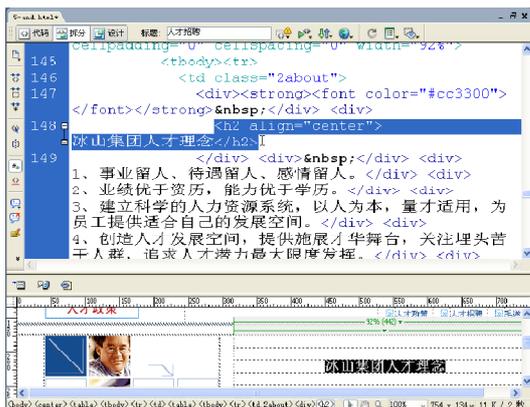
### 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<h2 align="center">冰山集团人才理念
</h2>
...
</body >
</html >
```

### 2. 范例解释

设置“冰山集团人才理念”文字为二级标题，然后设置align属性的值为center，可以让“冰山集团人才理念”文字水平居中对齐。

### 3. 代码和设计



### 4. 显示结果



在HTML文件中，可以加入多种文字的修饰标签，具体如下表所示。



标签	描述
<b>, <strong>	粗体
<i>, <em>, <cite>	斜体
<sup>	上标
<sub>	下标
<big>	大字号
<small>	小字号
<u>	下划线
<s>, <strike>	删除线
<address>	地址
<tt>	打字机文字
<code>, <samp>	等宽
<kbd>	键盘输入文字
<var>	声明变量
<rt>, <ruby>	标注说明

## B和Strong标签

<b>和<strong>标签可以明确地将包括在它们及其结束标签</b>和</strong>之间的字符或者文本设置成粗体。

<strong>是加重语气标签，<b>是粗体标签。大多数情况下，它们表现出来的效果也大致相同，所不同的是，<b>标签是一个实体标签，它所包围的字符将被设为Bold（粗体），而<strong>标签是逻辑标签，它能加强语气，而这又是通过将字符设置为Bold（粗体）来实现的。实际应用中，为了优化网页效果和代码，建议用<b> 标签。

原始文件	Sample\Ch04\3\1.html
最终文件	Sample\Ch04\3\1-end.html
学习要点	使用<b>标签

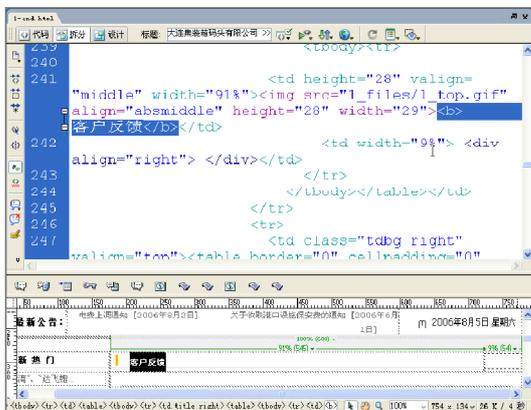
### 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<b>客户反馈</b>
...
</body >
</html >
```

### 2. 范例解释

设置“客户反馈”文字为粗体，如果加粗的字体不可用，则浏览器将使用一些其他的表现方法，例如反相显示或者加下划线等。

### 3. 代码和设计



### 4. 显示结果



## 斜体字标签

对于需要强调的英文内容，一般可以使用斜体的效果。这种方式同样可以适用于中文文字中。现在`<em>`标签修饰的内容都是用斜体字来显示的，这些内容有强调的含义，也许以后浏览器会使用其他特殊效果来显示强调的文本。除强调以外，当引入新的术语或在引用特定类型的术语或概念作为固定样式的时候，也可以考虑使用`<em>`标签。如果只想使用斜体字来显示文本的话，可使用`<i>`标签。除此之外，文档中还可以包括用来改变文本显示的级联样式定义。`<cite>`和`</cite>`用来输出引用方式的字体，通常是斜体。

原始文件	Sample\Ch04\3\2.html
最终文件	Sample\Ch04\3\2-end.html
学习要点	使用 <code>&lt;em&gt;</code> 标签

#### 1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<em>客户反馈</em>
...
</body>
</html>

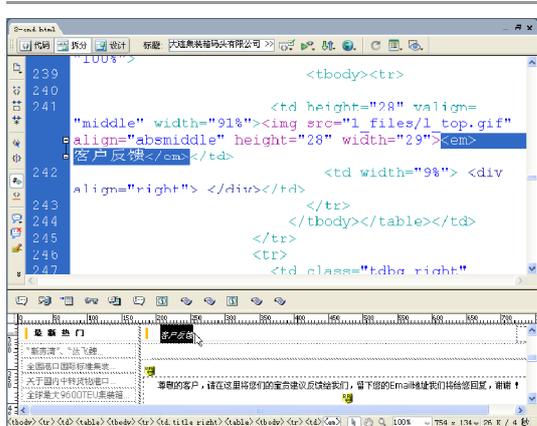
```

#### 2. 范例解释

通过`<em>`标签设置“客户反馈”文字为斜体。`<em>`标签可以把文字设置成斜体以与其他文字区别开来。



### 3. 代码和设计



### 4. 显示结果



## Sup 标签

在常见的数学表达式中可以将一段文字以小字体的方式显示在另一段文字的右上角，这就是上标。<sup>标签和其结束标签</sup>可以把它们包含的内容变成上标；它以当前文本流中字符高度的一半来显示，但是与当前文本流中文字的字体和字号都一样。<sup>标签在向文档添加脚注以及表示方程中的指数值时经常要用到。

原始文件	Sample\Ch04\3\3.html
最终文件	Sample\Ch04\3\3-end.html
学习要点	使用<sup>标签

#### 1. 范例代码

```

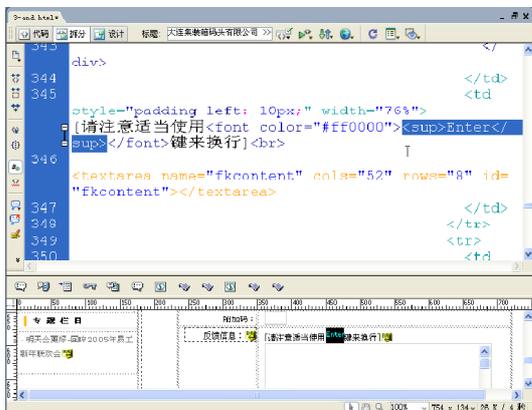
<html>
<head>
...
</head>
<body>
...
[ 请注意适当使用<font color="#ff00-00"><sup>Enter</sup></font>键来换行 ]
...
</body>
</html>

```

#### 2. 范例解释

将Enter 文字作为“请注意适当使用”文字的上标，并且通过<font>标签，将Enter 文字的颜色单独设置为#ff0000的红色。

### 3. 代码和设计



### 4. 显示结果



## Sub 标签

和<sup>标签相反,<sub>标签将一段文字以小字体的方式显示在另一段文字的右下角,这就是下标。<sub>标签和其结束标签</sub>可以把它们包含的内容变为下标以当前文本流中字符高度的一半来显示,但是与当前文本流中的字体和字号都是一样的。无论是<sub>标签还是和它对应的<sup>标签,在数学等式、科学符号和化学公式中都经常要用到。

原始文件	Sample\Ch04\3\4.html
最终文件	Sample\Ch04\3\4-end.html
学习要点	使用<sub>标签

#### 1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
[请注意适当使用<font color="#ff0000"><sub>Enter</sub></font>键来换行]
...
</body>
</html>

```

#### 2. 范例解释

将Enter文字作为“请注意适当使用”文字的下标,并且通过<font>标签,将Enter文字颜色单独设置为#ff0000的红色。



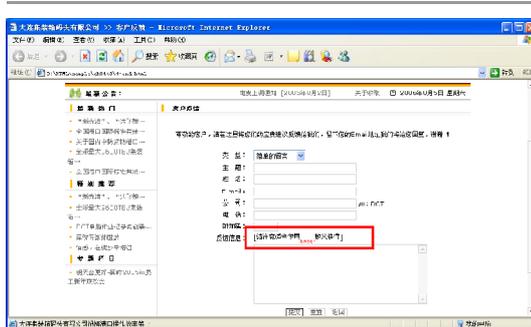
### 3. 代码和设计

```

344 </td>
345 </tr>
346 <tbody>
347 <tr>
348 <td style="padding-left: 10px; width: 76%;>
349 <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">
350 <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">

```

### 4. 显示结果



## Big 标签

使用<big>标签可以很容易地放大字体，但如果文字已经是最大号字体，则<big>标签将不起任何作用。

原始文件	Sample\Ch04\3\5.html
最终文件	Sample\Ch04\3\5-end.html
学习要点	使用<big>标签

#### 1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<big>客户反馈</big>
...
</body>
</html>

```

#### 2. 范例解释

设置“客户反馈”文字为大字体，浏览器显示包含在<big>标签和其相应的</big>标签之间的“客户反馈”文字时，其字体比周围的文字要大一号。

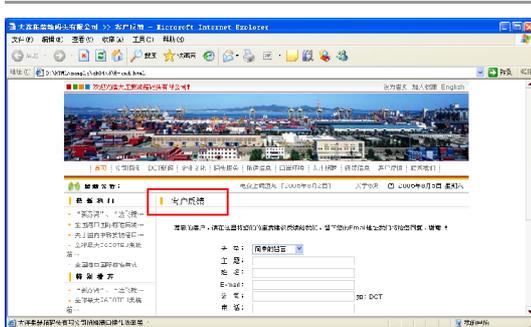
### 3. 代码和设计

```

239 <tbody>
240 <tr>
241 <td style="width: 50%; vertical-align: top; padding: 5px;">
242 <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">
243 <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">
244 </div>
245 </td>
246 <td style="width: 50%; vertical-align: top; padding: 5px;">
247 <div style="border: 1px solid #ccc; padding: 5px; min-height: 100px;">

```

### 4. 显示结果



## Small 标签

<small>标签和它所对应的<big>标签类似（见前面的描述），但它是缩小字体而不是放大。如果被包含的字体已经是字体模型所支持的最小字号，那么<small>标签将不起任何作用。与<big>标签类似，<small>标签也可以嵌套，从而连续地把文字缩小。每个<small>标签都将把文本的字体变小一号，直到达到下限的一号字。

原始文件	Sample\Ch04\3\6.html
最终文件	Sample\Ch04\3\6-end.html
学习要点	使用<small>标签

### 1. 范例代码

```

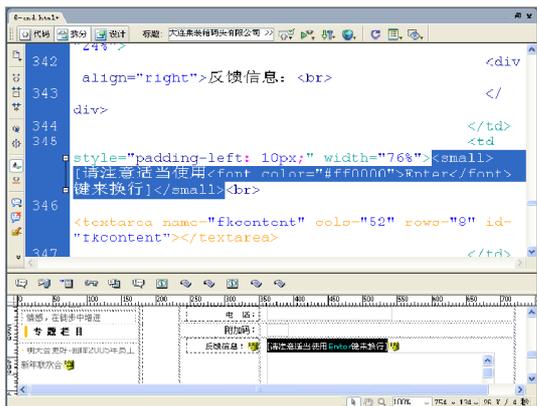
<html >
<head >
...
</head >
<body >
...
<small >[ 请注意适当使用<font color=
"#ff0000">Enter</font>键来换行]</
small >
...
</body >
</html >

```

### 2. 范例解释

将“[请注意适当使用Enter键来换行]”文字设置为小字体，浏览器显示包含在<small>标签和其相应的</small>标签之间的“客户反馈”文字时，其字体比周围的文字要小一号，并且通过<font>标签，将Enter文字的颜色单独设置为#ff0000的红色。

### 3. 代码和设计



### 4. 显示结果



## U 标签

<u>标签告诉浏览器要把包含在<u>和相应的</u>标签之间的文本加上下划线。后面介绍的样式表可以达到比<u>标签更好的效果。





原始文件	Sample\Ch04\3\7.html
最终文件	Sample\Ch04\3\7-end.html
学习要点	使用<u>标签

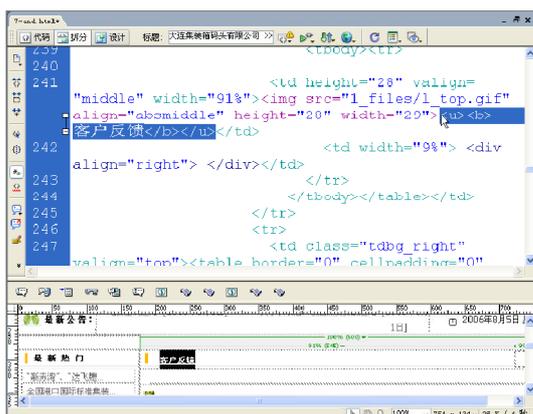
## 1. 范例代码

```
<html >
<head>
...
</head>
<body>
...
<u><b>客户反馈</b></u>
...
</body>
</html >
```

## 2. 范例解释

为加粗的“客户反馈”文字添加下划线，这种添加下划线的技术非常简单，它只是在空格、标点符号以及文本的下面划一条横线。

## 3. 代码和设计



## 4. 显示结果



## S和Strike标签

大多数浏览器都会在<strike>标签和其结束标签</strike>之间的文字中间划一条线(删除线)。而<s>标签是<strike>标签的缩写形式。

原始文件	Sample\Ch04\3\8.html
最终文件	Sample\Ch04\3\8-end.html
学习要点	使用<s>标签

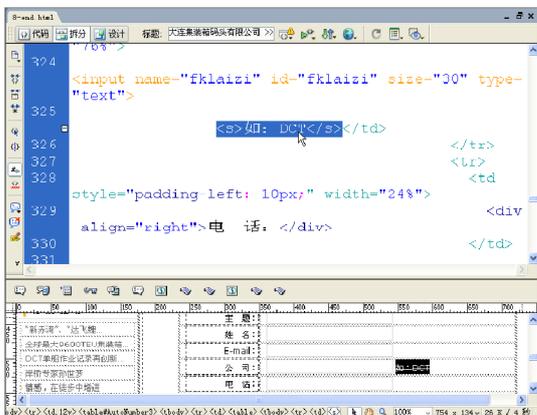
## 1. 范例代码

```
<html >
<head>
...
</head>
<body>
...
<s>如: DCT</s>
...
</body>
</html >
```

## 2. 范例解释

为“如: DCT”文字添加删除线, 它是一种编辑标识, 告诉读者可以忽略这段文字。

## 3. 代码和设计



## 4. 显示结果



## Address 标签

`<address>` 标签用于显示 Email、地址等文字内容, 主要用于英文字体的显示中, 当然也可以用于显示中文字体。在网页设计中, 可以方便浏览者与网站之间联系, 因此它所显示的信息是相当重要的。

原始文件	Sample\Ch04\3\9.html
最终文件	Sample\Ch04\3\9-end.html
学习要点	使用 <code>&lt;address&gt;</code> 标签



4

文字和段落



### 1. 范例代码

```

<html >
<head>
...
</head>
<body>
...
<address>
<font color="#666666"> 版权所有
Copyright&#169; 2004-2005</font>
<a href="http://www.dct.com.cn/"
  title="大连集装箱码头有限公司"
  target="blank">
<u>大连集装箱码头有限公司</u>
</a>
&nbsp; &nbsp; &nbsp; <br>
<font color="#666666">地址：大连
市开发区大窑湾港区 &nbsp; &nbsp; &nbsp; 邮编：
116601</font>
&nbsp; &nbsp; &nbsp; <font color="#666666"
>网站信箱：</font>
<a href="mailto:webmaster@dct.
com.cn">
<u>webmaster@dct.com.cn</u>
</a>
</address>
...
</body>
</html >

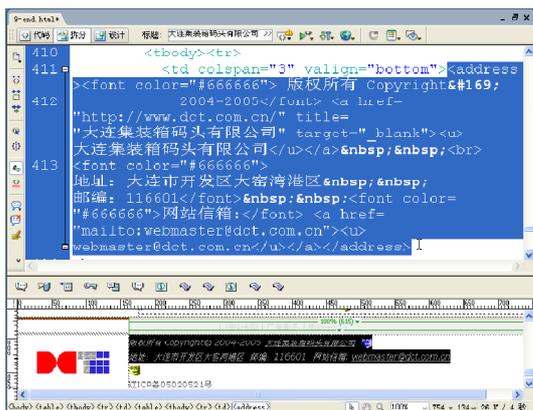
```

### 2. 范例解释

01 使用<address>标签使“版权所有 Copyright@ 2004-2005大连集装箱码头有限公司 地址：大连市开发区大窑湾港区 邮编：116601 网站信箱：webmaster@dct.com.cn”文字实现地址文字的斜体效果。

02 使用</address>标签结束地址文字。

### 3. 代码和设计



### 4. 显示结果



## Code 和 Samp 标签

软件代码的编写者已经习惯了编写源程序代码时文本表现的特殊样式。<code>标签就是为方便他们的工作习惯而设计的。

<samp>标签并不经常使用。只有在要从正常的上下文中将某些短字符序列提取出来，并对它们加以强调的极少数情况下，才使用这个标签。

<code>和<samp>标签实现的效果从外观上看是相同的。

原始文件	Sample\Ch04\3\10.html
最终文件	Sample\Ch04\3\10-end.html
学习要点	使用<code>标签

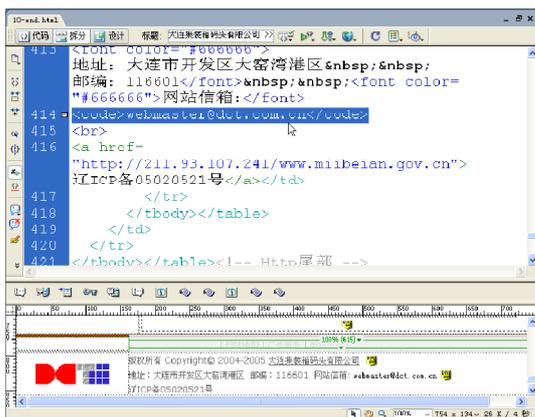
## 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<code>webmaster@dct.com.cn</code>
...
</body >
</html >
```

## 2. 范例解释

通过对“webmaster@dct.com.cn”文字使用<code>标签产生等宽文字效果。包含在该标签内的文本将用等宽、类似电传打字机样式的字体（Courier）显示出来，对于大多数程序员来说，这应该是十分熟悉的。

## 3. 代码和设计



## 4. 显示结果





## Kbd 标签

说到技术概念上的特殊样式时，就要提到<kbd>标签。它用来表示文本是从键盘输入的。浏览器通常用键盘输入字体来显示该标签中包含的文本。<kbd>标签经常用在与计算机相关的文档和手册中。

原始文件	Sample\Ch04\3\11.html
最终文件	Sample\Ch04\3\11-end.html
学习要点	使用<kbd>标签

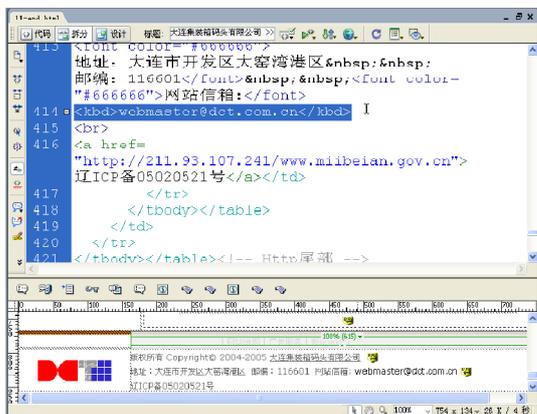
### 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<kbd>webmaster@dct.com.cn</kbd>
...
</body >
</html >
```

### 2. 范例解释

通过对“webmaster@dct.com.cn”文字使用<kbd>标签产生键盘输入文字效果。

### 3. 代码和设计



### 4. 显示结果



## Tt 标签

与<code>和<kbd>标签一样，<tt>标签和它必需的</tt>结束标签将告诉浏览器要把其中包含的文本显示为等宽字体。对于那些已经使用了等宽字体的浏览器来说，这个标签在文本的显示上就没有什么特殊效果了。

原始文件	Sample\Ch04\3\12.html
最终文件	Sample\Ch04\3\12-end.html
学习要点	使用<tt>标签

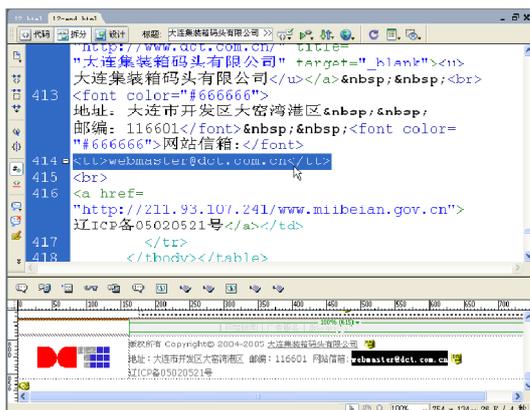
### 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<tt>webmaster@dct.com.cn</tt>
...
</body >
</html >
```

### 2. 范例解释

通过对“webmaster@dct.com.cn”文字使用<tt>标签产生打字机效果，把这段文本显示为等宽字体。

### 3. 代码和设计



### 4. 显示结果



## Var 标签

<var>标签是计算机文档中应用的另一个小窍门，它表示变量的名称，或者由用户提供的值。<var>标签不仅能让用户更容易理解和浏览文档，而且某些自动系统还可以合理地利用该标签，从文档中提取信息以及文档中提到的有用参数。

原始文件	Sample\Ch04\3\13.html
最终文件	Sample\Ch04\3\13-end.html
学习要点	使用<var>标签





### 1. 范例代码

```

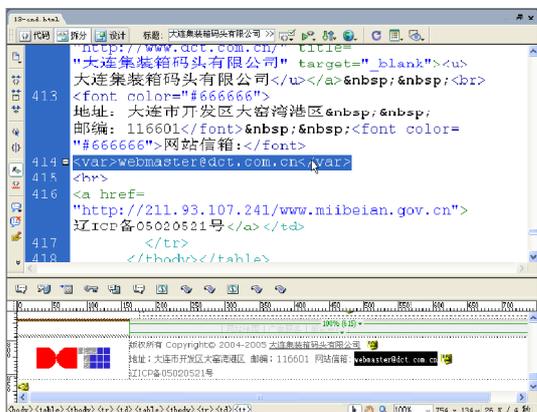
<html >
<head >
...
</head >
<body >
...
<var>webmaster@dct.com.cn</var>
...
</body >
</html >

```

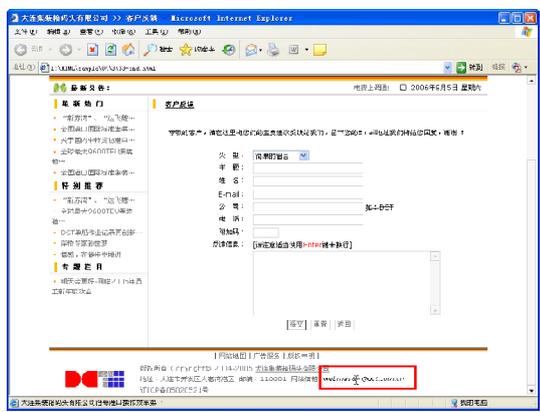
### 2. 范例解释

通过对“webmaster@dct.com.cn”文字使用<var>标签产生声明变量效果。用<var>标签标记的这段文本通常显示为斜体，就像其他与计算机编程和文档相关的标签一样。

### 3. 代码和设计



### 4. 显示结果



## Rt和Ruby标签

使用<ruby>标签及其结束标签</ruby>可以对汉字加注拼音；而使用<rt>标签及其结束标签</rt>则可以对其包含的说明文字以缩小字号的形式显示。

基本语法	<ruby>
	被标识的文字
	<rt>
	说明文字
功能	</rt>
	</ruby>
	被<rt>标签标识的文字，将以缩小字号的形式显示在以<ruby>标签标识的文字上方，用来说明这段文字

原始文件	Sample\Ch04\3\14.html
最终文件	Sample\Ch04\3\14-end.html
学习要点	使用<rt>和<ruby>标签





## P 标签

<p>标签表示了一个段落的开始。我们熟悉的大多数文字处理程序都只使用一种特殊字符（如回车符等）来标记段落的结束，而在HTML中，每一个段落都必须由一个<p>开始，再由一个相对应的</p>结束。虽然可以在文字处理程序中使用一系列回车符来产生一些新的空白段落，但是除了第一个段落标签以外，浏览器通常会忽略其他所有标签。

在实际应用中，当处理HTML文档时，可以忽略第一段开始的<p>标签，以及每一段末尾的</p>标签：浏览器可以根据文档中的其他标签推断出这两个标签来，因此这种省略是非常安全的。

基本语法	< p >	
	段落文字	
	</p>（可省略）	
功能	定义文本的段落	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Align	对齐方式
	Class	用一个名称来标记段落，该标记名称指向一个预定义的种类，而该类是在文档级声明的或者在外部定义的样式表
	Id	为段落创建一个标记，将来超链接可以用这个标记来明确地引用该段落，以便作为样式表选择器和使用其他应用程序来执行自动搜索
	Style	创建段落内容的内联样式
Title	给段落加上说明性的文字	

原始文件	Sample\Ch04\4\1.html
最终文件	Sample\Ch04\4\1-end.html
学习要点	使用<p>标签

当遇到新的段落标签（<p>）时，浏览器通常会在开始一个新的段落前，插入一个空行并增加行间距；然后浏览器把所有词语和内联图像（如果有的话）都收集到一起，组成新的段落，并忽略开头和结尾的所有空格（当然不包括文字中间的空格）及文本中的回车符。之后浏览器软件就把这个最终的文字和图像的序列流到显示屏幕上生成一个新的段落，并自动加入适当的换行，以适应窗口显示的宽度。

## 1. 范例代码

```

<html >
<head >
...
</head >
<body >
...
<p>中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率，成为成都市脱颖而出的著名楼盘。在随后的5天时间里，第一批次房源基本销售完毕，演绎了成都房地产行业2006年的一个奇迹。</p>
<p>在4月20日的成都春季房交会上，中大·君悦金沙销售依然高潮迭起，新推出的房源再次受到购房者的青睐，成了成都购房者最为关注的热点楼盘之一。距离第一次开盘5天后又推出新的房源，这在成都房地产的销售史上也是极为罕见的。</p>
...
</body >
</html >

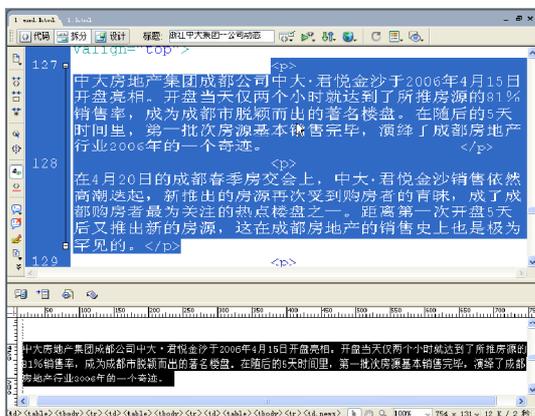
```

## 2. 范例解释

01 设置“中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率，成为成都市脱颖而出的著名楼盘。在随后的5天时间里，第一批次房源基本销售完毕，演绎了成都房地产行业2006年的一个奇迹。”文字使用段落格式。

02 将“在4月20日的成都春季房交会上，中大·君悦金沙销售依然高潮迭起，新推出的房源再次受到购房者的青睐，成了成都购房者最为关注的热点楼盘之一。距离第一次开盘5天后又推出新的房源，这在成都房地产的销售史上也是极为罕见的。”文字设置成新的段落，实现自动换行。

## 3. 代码和设计



## 4. 显示结果



对于大多数浏览器来说，段落默认的对齐方式是左对齐，但用户也可以把 `Align` 属性的对齐方式改为 `Center` 或 `Right`，如下表所示。

Align属性值及说明	Align属性值	说明
	Left	段落居左对齐
	Right	段落居右对齐
	Center	段落居中对齐



原始文件	Sample\Ch04\4\2.html
最终文件	Sample\Ch04\4\2-end.html
学习要点	使用<p>标签的align属性

## 1. 范例代码

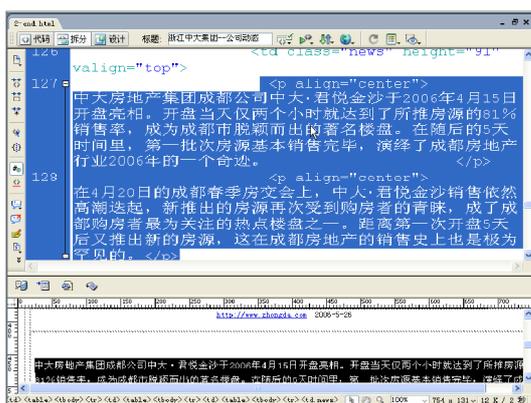
```
<html>
<head>
...
</head>
<body>
...
<p align="center">中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率，成为成都市脱颖而出的著名楼盘。在随后的5天时间里，第一批次房源基本销售完毕，演绎了成都房地产行业2006年的一个奇迹。</p>
<p align="center">在4月20日的成都春季房交会上，中大·君悦金沙销售依然高潮迭起，新推出的房源再次受到购房者的青睐，成了成都购房者最为关注的热点楼盘之一。距离第一次开盘5天后又推出新的房源，这在成都房地产的销售史上也是极为罕见的。</p>
...
</body>
</html>
```

## 2. 范例解释

**01** 将“中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率，成为成都市脱颖而出的著名楼盘。在随后的5天时间里，第一批次房源基本销售完毕，演绎了成都房地产行业2006年的一个奇迹。”段落文字设置为水平居中对齐。

**02** 将“在4月20日的成都春季房交会上，中大·君悦金沙销售依然高潮迭起，新推出的房源再次受到购房者的青睐，成了成都购房者最为关注的热点楼盘之一。距离第一次开盘5天后又推出新的房源，这在成都房地产的销售史上也是极为罕见的。”段落文字设置为水平居中对齐。

## 3. 代码和设计



## 4. 显示结果



## Br 标签

<br>标签将打断HTML文档中正常段落的行间距和换行。在HTML中它没有结束标签，只不过简单地指出在文本中从哪里开始新的一行。大多数浏览器会停止在本行中继续添加文字和图像，转移到下一行的开始处，也就是该行的最左边，然后再继续添加间距和换行。

基本语法	 	
功能	在文本中插入一个换行	
属性及说明	属性	说明
	Class	用一个名称来标记换行，该标记名称指向一个预定义的种类，而该类是在文档级声明的或者在外部定义的样式表为换行创建一个标记，将来超链接可以用这个标记来明确地引用该行，以便作为样式表选择器和使用其他应用程序来执行自动搜索
	Id	创建换行内容的内联样式
	Style	给换行加上说明性的文字
	Title	

原始文件	Sample\Ch04\4\3.html
最终文件	Sample\Ch04\4\3-end.html
学习要点	使用 标签

## 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
中大房地产集团成都公司中大·君悦金沙于
2006年4月15日开盘亮相。开盘当天仅两
个小时就达到了所推房源的81%销售率，
成为成都市脱颖而出的著名楼盘。在随后
的5天时间里，第一批次房源基本销售完
毕，演绎了成都房地产行业2006年的一个
奇迹。<br >
在4月20日的成都春季房交会上，中大·
君悦金沙销售依然高潮迭起，新推出的房
源再次受到购房者的青睐，成了成都购房
者最为关注的热点楼盘之一。距离第一
次开盘5天后又推出新的房源，这在成
都房地产的销售史上也是极为罕见的。<br >
...
</body >
</html >
```

## 2. 范例解释

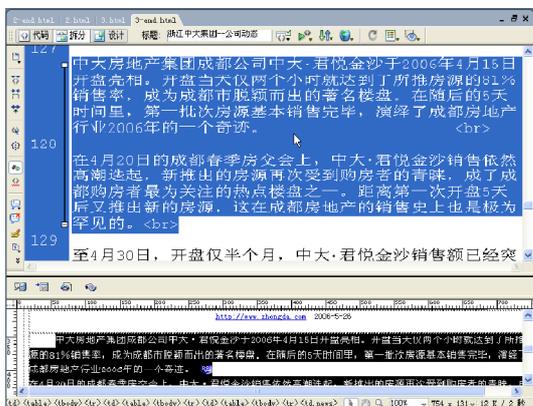
01 设置“中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率，成为成都市脱颖而出的著名楼盘。在随后的5天时间里，第一批次房源基本销售完毕，演绎了成都房地产行业2006年的一个奇迹。”段落文字使用换行。

02 “在4月20日的成都春季房交会上，中大·君悦金沙销售依然高潮迭起，新推出的房源再次受到购房者的青睐，成了成都购房者最为关注的热点楼盘之一。距离第一次开盘5天后又推出新的房源，这在成都房地产的销售史上也是极为罕见的。”文字开始新的换行。





### 3. 代码和设计



### 4. 显示结果



## Noobr 标签

有时设计者可能希望某个短语在浏览器窗口的一行中出现，即使该短语可能已经超出窗口的显示区域，也不会被打断。计算机命令就是一个很好的例子。一般来讲，应在一行中输入计算机指令，尽管它可能含有很多个单词。因为无法预计用户窗口中每一行可以显示多少个单词，所以计算机指令可能会被分开为两行或者更多行文本。这些指令语法本身就已经非常复杂了，因此没有必要把它分开在两行里，以免更加混乱。

使用<noobr>标签就可以保证文本在同一行中显示，显示的样式也和普通文本一样。<noobr>标签的效果就是使浏览器像对待一个不可分割的单词那样对待这个标签中的内容。标签内容保持了当前的字体样式，当然也可以在标签中将其改变成另外一种样式。

基本语法	<noobr>
	文字
	</noobr>
功能	创建一个不换行文本区域

原始文件	Sample\Ch04\4.html
最终文件	Sample\Ch04\4-end.html
学习要点	使用<noobr>标签

### 1. 范例代码

```

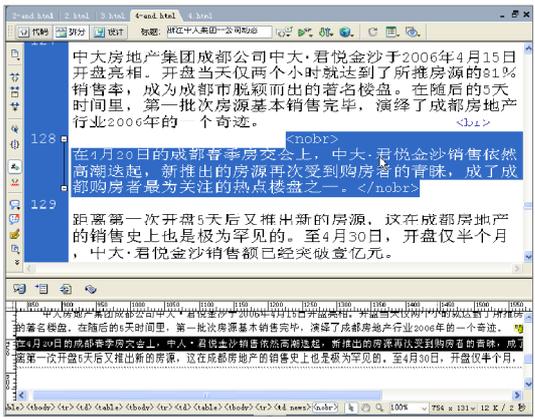
<html >
<head >
...
</head >
<body >
...
<nobr>在 4 月 20 日的成都春季房交会上，
中大·君悦金沙销售依然高潮迭起，新推出
的房源再次受到购房者的青睐，成了成都购
房者最为关注的热点楼盘之一。</nobr >
...
</body >
</html >

```

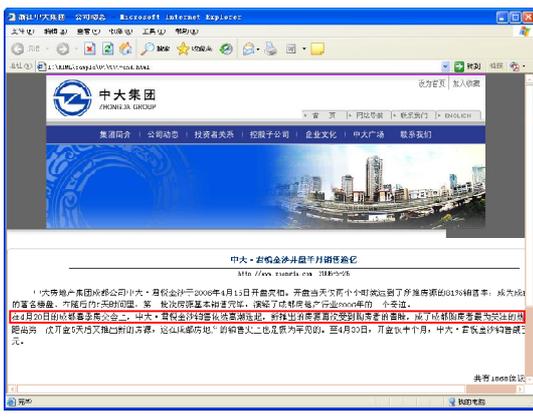
### 2. 范例解释

设置“在 4 月 20 日的成都春季房交会上，中大·君悦金沙销售依然高潮迭起，新推出的房源再次受到购房者的青睐，成了成都购房者最为关注的热点楼盘之一。”文字不使用换行。

### 3. 代码和设计



### 4. 显示结果



4  
文字和段落

## Div 标签

前面介绍过，设计者可以使用<br>标签来明确地强制普通文本进行换行；<p>标签虽然也有换行的作用，但是在意义和效果上却不仅是换行那么简单；<div>标签则略有不同。<div>标签最早是在 HTML 3.2 标准中定义的，当初将它包括在内，只是为了让它成为一种简单的组织工具，也就是用来把文档分成各个分离的部分，因此很少有人使用这个似乎有些迟钝的功能。但是最近的一些改进（对齐、样式以及用于文档引用和自动化的 id 属性等），允许用户更加清楚地标记和定义文档的每一个部分，还允许控制这些部分的排列和外观。这些特征为<div>标签注入了实际应用的活力意义。

通过使用<div id=name class=name>标签和属性，可以给文档的各个部分都关联一个id和class名称（也可以用其他标签进行该项操作，如使用<p>标签），这样一来，不仅为以后超链接对文档的引用和文档自动化处理及管理（例如把所有参考文献部分都整理起来）标记了这些部分，而且还可以为这些部分定义不同的、独特的显示样式，例如可以将文档第一部分的类名称定义为摘要（例如<div class=abstract>），另外一部分定义为主体，第三部分定义为结论，最后一部分定义为参考文献（例如<div class=bi bio>）。然后每个类都可以赋予一



个文档级或者外部样式表级的不同显示定义,例如摘要用缩进的斜体字表示(如

abstract {font-style: italic}); 主体使用左对齐的罗马(Roman)字体,结论与摘要类似,参考文献则会自动加上编号,并适当进行格式化。

与HTML 4.01标准中定义的一样,<div>标签可以将文档分割成独立的、不同的部分。它可以用作严格的组织工具,而不使用任何格式化,但是用Id和Class属性来标记部分时,该标签的作用会变得更加有效。<div>标签还可以和Align属性结合起来,控制整个部分中文档内容显示时的对齐方式,还可以和许多编程属性结合起来实现用户交互。

基本语法	<div>	
	分块文字	
	</div> (可省略)	
功能	定义文本的段落	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Align	对齐方式
	Class	用一个名称来标记分块,该标记名称指向一个预定义的种类,而该类是在文档级声明的或者在外部定义的样式表为分块创建一个标记,应用超链接时可以用这个标记来明确地引用该分块,以便作为样式表选择器或使用其他应用程序来执行自动搜索
	Id	创建分块内容的内联样式
	Style	给分块加上说明性的文字
	Nowrap	强制不换行

原始文件	Sample\Ch04\4\5.html
最终文件	Sample\Ch04\4\5-end.html
学习要点	使用<div>标签

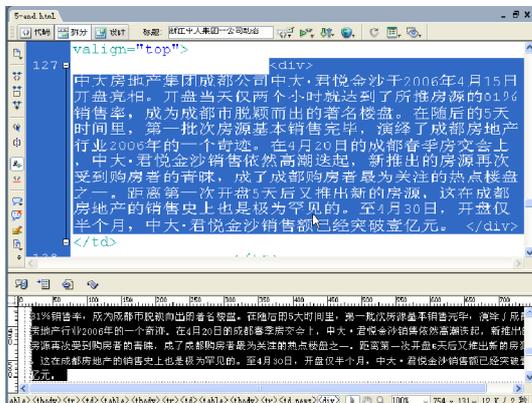
## 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<div >中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率,成为成都市脱颖而出的著名楼盘…… </div >
...
</body >
</html >
```

## 2. 范例解释

设置“中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率,成为成都市脱颖而出的著名楼盘……”文字使用分块。

### 3. 代码和设计



### 4. 显示结果



`<div>`的 `Align` 属性指定了标签中包含的文档内容在显示时是左对齐 (Left, 默认设置)、右对齐 (Right), 还是居中对齐 (Center)。另外还可以指定 `Justify` (两端对齐) 来把文本的左边和右边同时对齐。`<div>` 标签可以嵌套, 嵌套的 `<div>` 标签的排列优先于包含它的 `<div>` 标签。像其他标签的 `Align` 属性一样, 在 HTML 标准中, 为了遵守基于样式表的布局控制, 现在已经不推荐再使用这个属性。 `Align` 属性值如下表所示。

Align属性值及说明	Align属性值	说明
	Left	分块居左对齐
	Right	分块居右对齐
	Center	分块居中对齐
	Justify	分块两端对齐

原始文件	Sample\Ch04\4\6.html
最终文件	Sample\Ch04\4\6-end.html
学习要点	使用 <code>&lt;div&gt;</code> 标签的 <code>align</code> 属性

#### 1. 范例代码

```

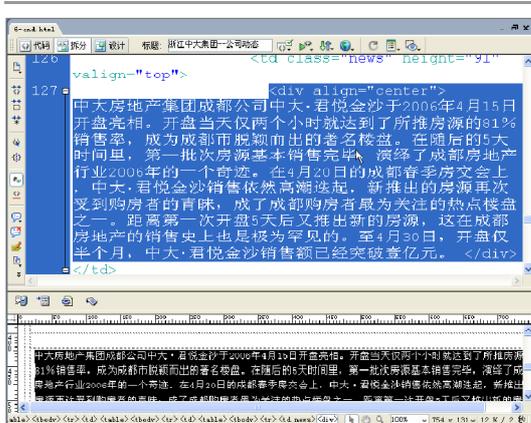
<html>
<head>
...
</head>
<body>
...
<div align="center">中大房地产集团
成都公司中大·君悦金沙于2006年4月15
日开盘亮相。开盘当天仅两个小时就达到了
所推房源的81%销售率，成为成都市脱颖
而出的著名楼盘…… </div>
...
</body>
</html>
    
```

#### 2. 范例解释

设置“中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率，成为成都市脱颖而出的著名楼盘……”分块为水平居中对齐。



## 3. 代码和设计



## 4. 显示结果



## Span 标签

`<span>` 标签只是界定了内容的一部分。某些情况下，可能只需要改变标签内容中一部分（通常是文本）的外观，这时可用 `<span>` 标签来指定这些特殊段。浏览器把 `<span>` 标签当作另一个自然或基于内容的样式标签。`<span>` 标签默认不管文本样式。在 HTML 的未来版本中，`<span>` 标签可能还有其他功能，目前可以将它应用到样式、显示和事件管理或文档内容的任意部分。用户可以为 `<span>` 标签定义一个样式。

基本语法	<code>&lt;span&gt;</code>	
	行内样式定义文字	
	<code>&lt;/span&gt;</code>	
	任意界定文本	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记分块，该标记名称指向一个预定义的种类，而该类是在文档级声明的或者在外部定义的样式表为行创建一个标记，应用超链接时可以用这个标记来明确地引用该行，以便作为样式表选择器或使用其他应用程序来执行自动搜索
	Id	创建行内容的内联样式
	Title	给行加上说明性的文字

原始文件	Sample\Ch04\4\7.html
最终文件	Sample\Ch04\4\7-end.html
学习要点	使用 <code>&lt;span&gt;</code> 标签

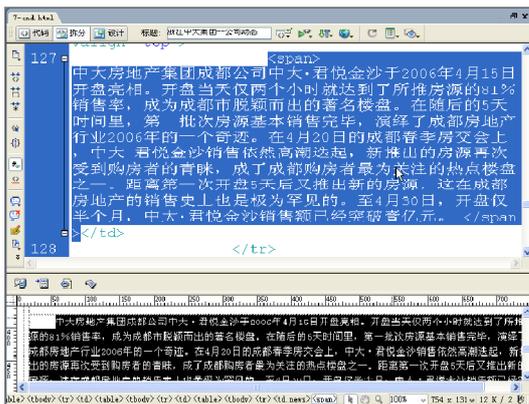
## 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<span>中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率，成为成都市脱颖而出的著名楼盘……</span>
...
</body >
</html >
```

## 2. 范例解释

设置“中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率，成为成都市脱颖而出的著名楼盘……”使用行内样式定义文字。

## 3. 代码和设计



## 4. 显示结果



## Center 标签

`<center>` 标签是另一个效果非常明显的标签，其内容（包括文本、图形、表格等）都会在浏览器的显示窗口中水平居中排列。对于文本来说，它意味着当在文本流中添加了间距并进行了换行之后，每一行都居中显示。`<center>` 标签将其包含的文本显示为水平居中对齐效果，直到被 `</center>` 结束标签终止为止。

基本语法	<code>&lt;center&gt;</code>	
	文字	
功能	<code>&lt;/center&gt;</code>	
	将文本居中	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息





(续表)

属性	说明
Class	用一个名称来标记文字，该标记名称指向一个预定义的一类，而该类是在文档级声明的或者在外部定义的样式表
Id	为文字创建一个标记，应用超链接时可以用这个标记来明确地引用该文字，以便作为样式表选择器或使用其他应用程序来执行自动搜索
Style	创建文字内容的内联样式
Title	给文字加上说明性的文字

原始文件	Sample\Ch04\4\8.html
最终文件	Sample\Ch04\4\8-end.html
学习要点	使用<center>标签

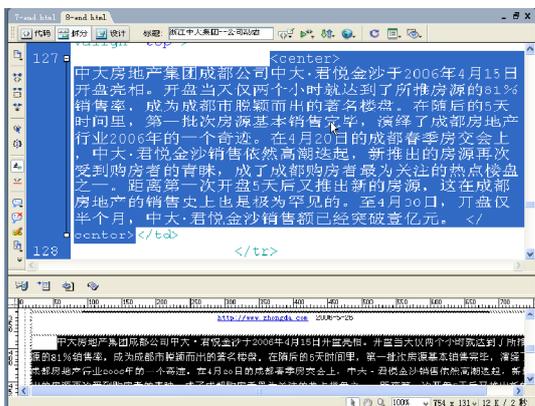
## 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<center>中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率，成为成都市脱颖而出的著名楼盘…… </center>
...
</body >
</html >
```

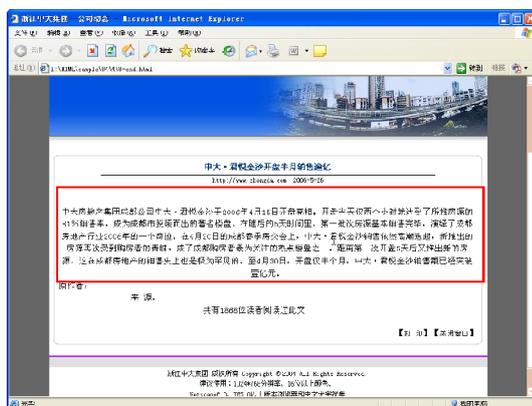
## 2. 范例解释

设置“中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率，成为成都市脱颖而出的著名楼盘……”文字水平居中对齐。

## 3. 代码和设计



## 4. 显示结果



## Blockquote 标签

在<blockquote>和</blockquote>标签之间的所有文本都会从常规文档中分离出来,经常会被浏览器显示为左、右两边缩进的效果,有时也会显示为斜体。当然,实际的显示效果会因浏览器的不同而各不相同。HTML 标准允许在<blockquote>中使用任何标签。

基本语法	<blockquote>	
	文字	
	</blockquote>	
功能	定义一个块引用	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记文字,该标记名称指向一个预定义的种类,而该类是在文档级声明的或者在外部定义的样式表为文字创建一个标记,应用超链接时可以用这个标记来明确地引用该文字,以便作为样式表选择器或使用其他应用程序来执行自动搜索
	Id	创建文字内容的内联样式
	Style	给文字加上说明性的文字
	Title	

原始文件	Sample\Ch04\4\9.html
最终文件	Sample\Ch04\4\9-end.html
学习要点	使用<blockquote>标签

## 1. 范例代码

```
<html >
<head>
...
</head>
<body>
...
<blockquote>
<p>
中大房地产集团成都公司中大·君悦金沙于
2006年4月15日开盘亮相。开盘当天仅两
个小时就达到了所推房源的81%销售率,
成为成都市脱颖而出的著名楼盘。在随后
的5天时间里,第一批次房源基本销售完
毕,演绎了成都房地产行业2006年的一个
奇迹.....
</p>
</blockquote>
...
</body>
</html >
```

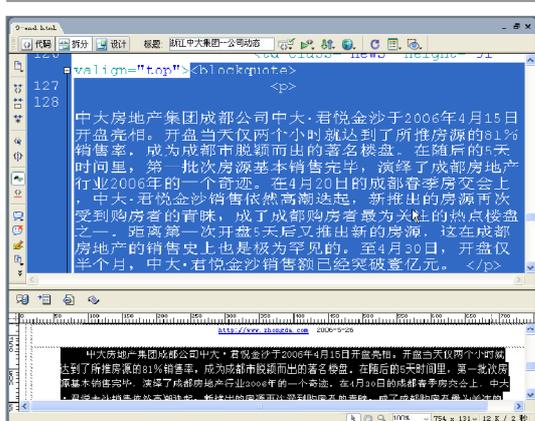
## 2. 范例解释

设置“中大房地产集团成都公司中大·君悦金沙于2006年4月15日开盘亮相。开盘当天仅两个小时就达到了所推房源的81%销售率,成为成都市脱颖而出的著名楼盘。在随后的5天时间里,第一批次房源基本销售完毕,演绎了成都房地产行业2006年的一个奇迹……”段落文字为左右两边缩进排列。

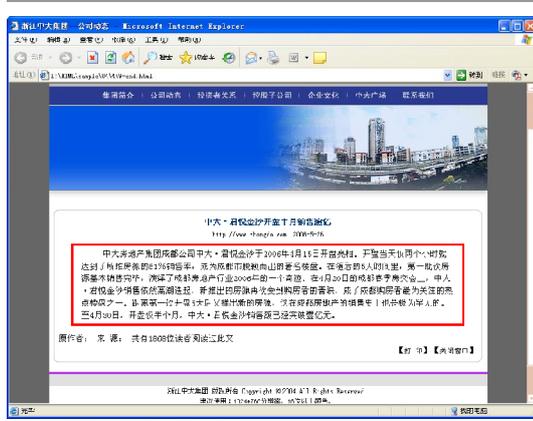




### 3. 代码和设计



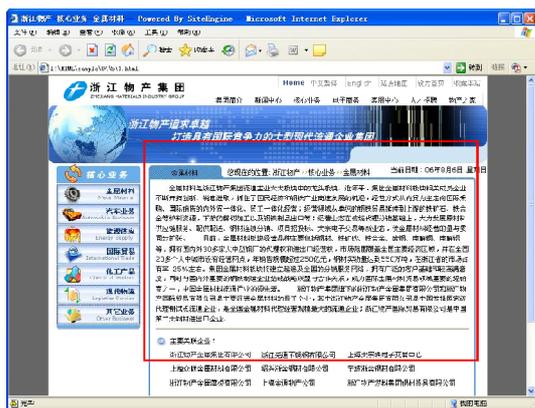
### 4. 显示结果



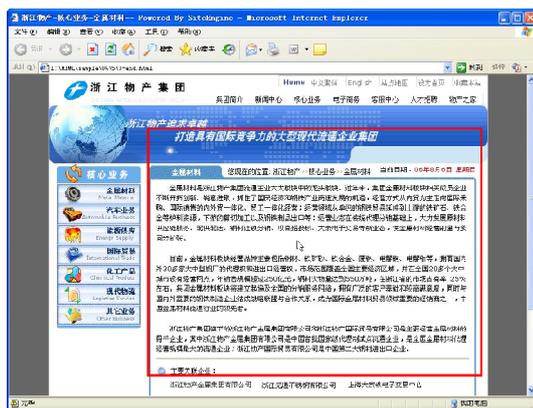
## 5 实例制作

下面将通过一个实例练习文字页面的制作，设计者需要用到文字标签、文字修饰、段落排列等HTML标签。读者可以先观察一下原始效果图与最终效果图的不同，再在编写代码的过程中体会各种标签的用法。

原始文件	Sample\Ch04\5\1.html
最终文件	Sample\Ch04\5\1-end.html
学习要点	使用文字标签、文字修饰标签和段落排列标签



原始效果



最终效果





4 将光标定位到“金属材料是浙江物产集团流通主业……”的位置，在代码视图中将其修改为以下代码：

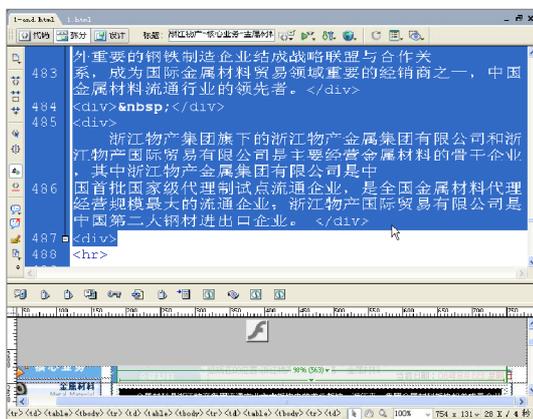
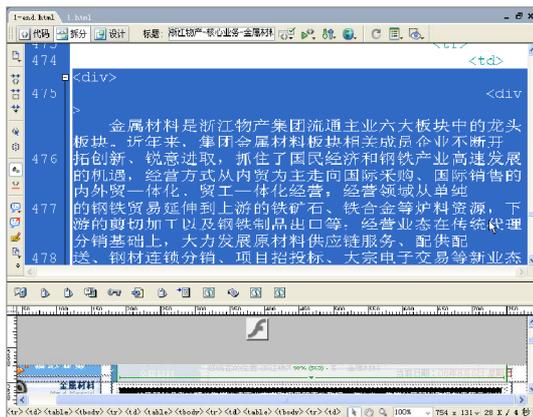
```

<div>
    <div> 金属材料是浙江物产集团流通主业六大板块中的龙头板块。……大宗电子交易等新业态，使金属材料经营的量与质同步扩张。</div>
<div>&nbsp; </div>
<div> 目前，金属材料板块经营品种主要包括钢材、铁矿砂、铁合金、废钢、电解铜、电解铝等，……成为国际金属材料贸易领域重要的经销商之一，中国金属材料流通行业的领先者。</div>
<div>&nbsp; </div>
<div> 浙江物产集团旗下的浙江物产金属集团有限公司和浙江物产国际贸易有限公司是主要经营金属材料的骨干企业，……浙江物产国际贸易有限公司是中国第二大钢材进出口企业。 </div>
<div>

```

- ❶ 设置3段分块文字。
- ❷ 设置4个空白的分块行。

5 按下F12快捷键，即可在浏览器中看到页面的效果。





# Chapter

# 5

## 水平线和列表

水平线用于段落与段落之间的分隔，它可以使文档的结构清晰明白，文字的编排整齐美观。水平线有很多的属性，如宽度、高度、颜色、排列对齐等。水平线在HTML文档中经常用到，合理地使用水平线可以获得非常好的视觉效果。

在HTML页面中，列表可以起到提纲挈领的作用。列表分为两种类型，一是有序列表，一是无序列表。前者使用项目符号来标记无序的项目，后者则使用编号来记录项目的顺序。

```
meta http-equiv="content-type"
charset=gb2312"
<link href="1_files/css.css" rel="
"text/css">
```



## 1 水平线

水平分隔线可以从视觉上将文档分隔成各个不同的部分。这样可以给浏览者一个整洁而明确的视觉指示，告诉他们文档一个部分已经结束，而另一个部分即将开始。水平分隔线有效地将文本分隔成小块，并界定了文档的页首（header）和页尾（footer），另外还突出了对文档内标题的显示。

### Hr 标签

<hr>标签告诉浏览器要插入一条横跨整个显示窗口的水平分隔线。在HTML中，该标签没有相应的结束标签。和<br>标签一样，<hr>标签也强制执行一个简单的换行；但是与<br>标签不同的是，<hr>标签将导致段落的对齐重新回到默认值设置（左对齐）。浏览器会立即在当前行的下面放置一条分隔线，然后再从分隔线下面的一行继续文本流的输出。

基本语法	<hr>	
功能	插入一条水平分隔线	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记水平线，该标记名称指向一个预定义的种类，而该类是在文档级声明的或者在外部定义的样式表
	Id	为水平线创建一个标记，应用超链接时可以用这个标记来明确地引用该水平线，以便作为样式表选择器或使用其他应用程序来执行自动搜索
	Style	创建水平线内容的内联样式
	Title	给水平线加上说明性的文字
	Align	水平线对齐方式
	Color	水平线颜色
	Noshade	使水平线不出现阴影
	Size	水平线高度
Width	水平线宽度	

原始文件	Sample\Ch05\1\1.html
最终文件	Sample\Ch05\1\1-end.html
学习要点	使用<hr>标签

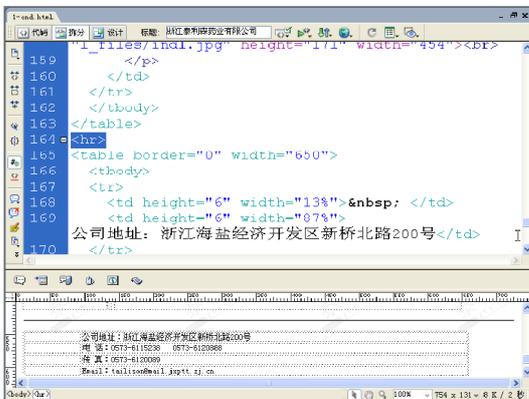
## 1. 范例代码

```
<html >
<head>
...
</head>
<body>
...
<hr>
...
</body>
</html >
```

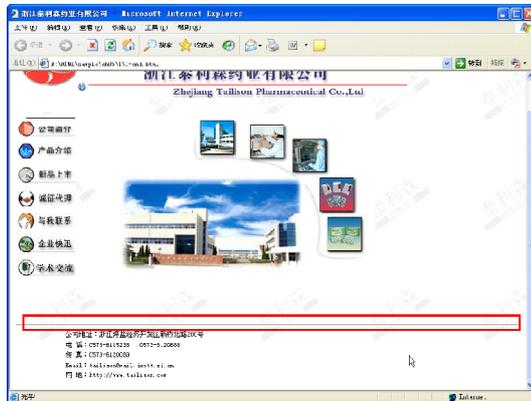
## 2. 范例解释

设置水平线，通常情况下，它会横跨整个文档。浏览器可能会以凹陷或者凸起的效果来显示水平分隔线。在水平分隔线的上面或下面没有其他空白。如果希望把它和周围的文本分隔开，那么就必须明确地将分隔线放在一个新的段落中，后面再跟另外一个包含随后文本的段落。

## 3. 代码和设计



## 4. 显示结果



通过设置<hr>标签的Size属性可以加粗分隔线。它的值表示的是以像素为单位的分隔线的粗细。

原始文件	Sample\Ch05\1\2.html
最终文件	Sample\Ch05\1\2-end.html
学习要点	使用<hr>标签的size属性

## 1. 范例代码

```
<html >
<head>
...
</head>
<body>
...
<hr size="4">
...
</body>
</html >
```

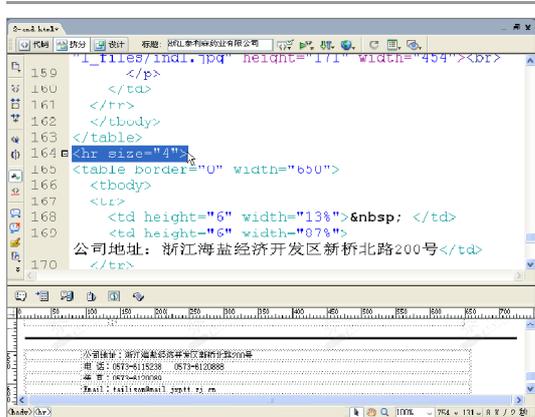
## 2. 范例解释

设置水平线的厚度为4像素。通常情况下，浏览器以2到3个像素的粗细和凹陷的3D效果来显示水平分隔线，这样分隔线看起来就好像镶嵌在文档中一样。

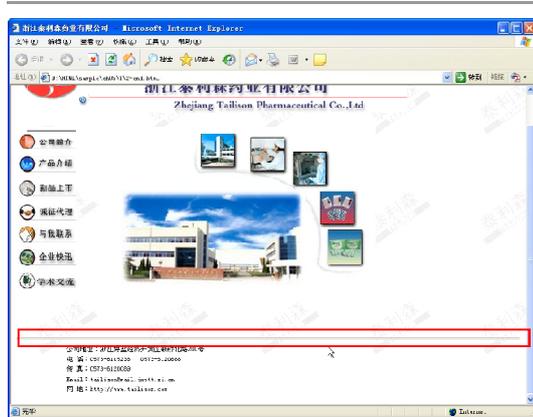




### 3. 代码和设计



### 4. 显示结果



有的用户可能不喜欢 3D 效果的分隔线，而希望使用扁平的 2D 分隔线，那么只需把 Noshade 属性加到<hr>标签中，就可以去掉 3D 效果。在 HTML 语法中并不要求为它赋值。

原始文件	Sample\Ch05\1\3.html
最终文件	Sample\Ch05\1\3-end.html
学习要点	使用<hr>标签的noshade属性

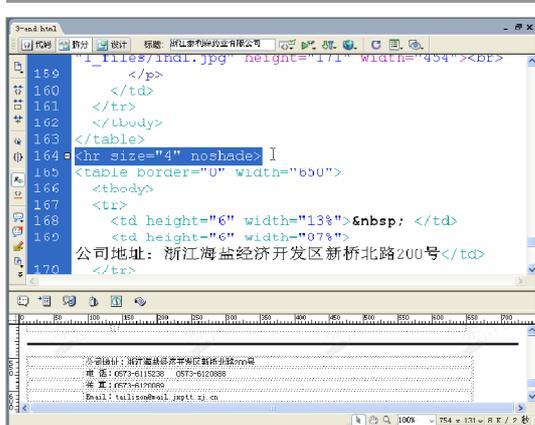
### 1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<hr size="4" noshade>
...
</body>
</html>
```

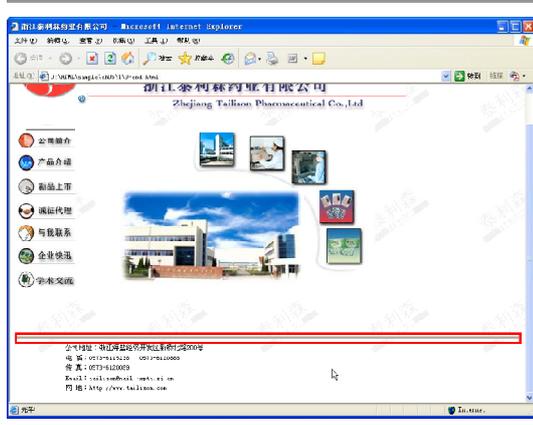
### 2. 范例解释

设置 4 像素粗细的水平线没有阴影，这就将实现扁平的 2D 分隔线的效果。

### 3. 代码和设计



### 4. 显示结果



默认的分隔线会横跨整个显示窗口。可以通过设置<hr>标签的Width属性来缩短或者加长分隔线的长度，从而创建特定宽度的分隔线，大多数浏览器会自动居中显示非整屏宽度的分隔线，可参阅Align属性来创建左对齐或者右对齐的水平分隔线。

原始文件	Sample\Ch05\1\4.html
最终文件	Sample\Ch05\1\4-end.html
学习要点	使用<hr>标签的width属性

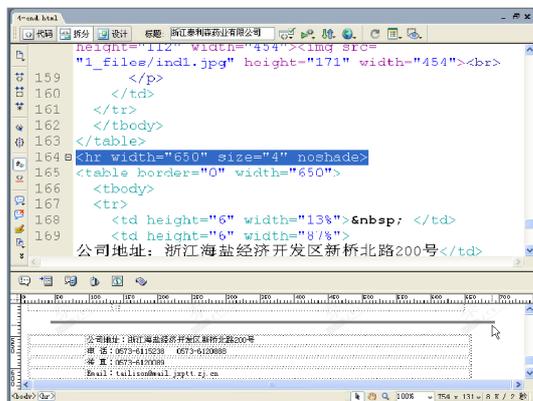
### 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<hr width="650" size="4" noshade >
...
</body >
</html >
```

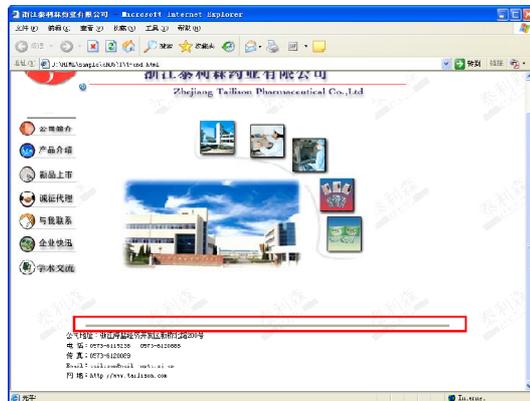
### 2. 范例解释

设置4像素粗细没有阴影的水平线宽度为650像素，需要注意的是，这里也可以设置当前文本流宽度的一定百分比的分隔线。

### 3. 代码和设计



### 4. 显示结果



水平分隔线的Align属性可能会是下面3个值中的一个：Left、Center或者Right，那些宽度小于当前文本流宽度的分隔线会相对于窗口边界进行定位。默认的对齐方式是Center（居中）。

原始文件	Sample\Ch05\1\5.html
最终文件	Sample\Ch05\1\5-end.html
学习要点	使用<hr>标签的align属性





### 1. 范例代码

```

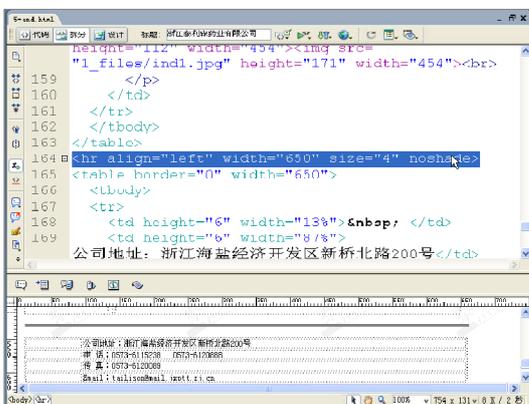
<html >
<head>
...
</head>
<body>
...
<hr align="left" width="650" size
="4" noshade>
...
</body>
</html >

```

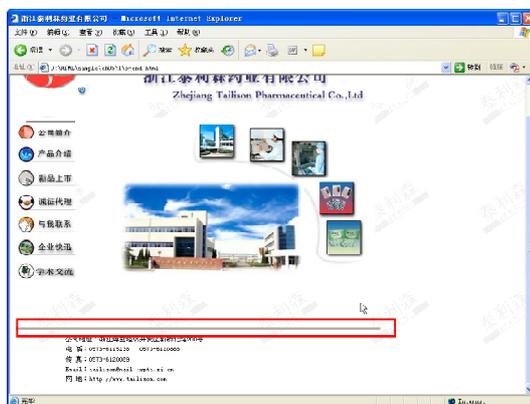
### 2. 范例解释

设置4像素粗细、没有阴影、宽度为650像素的水平线为居左对齐。设计师经常通过各种对齐方式的分隔线来作为部分分隔符。

### 3. 代码和设计



### 4. 显示结果



<hr>标签的Color属性可以为分隔线设置颜色。默认情况下，分隔线的颜色与文档背景的颜色相同，只是凹陷的边缘部分比背景色略深或略浅；如果指定其他颜色，就会失去这种3D效果。

原始文件	Sample\Ch05\1\6.html
最终文件	Sample\Ch05\1\6-end.html
学习要点	使用<hr>标签的color属性

### 1. 范例代码

```

<html >
<head>
...
</head>
<body>
...
<hr align="left" width="650" size
="4" noshade color="#CC0000">
...
</body>
</html >

```

### 2. 范例解释

设置4像素粗细、没有阴影、宽度为650像素、排列为居左对齐的水平线颜色为#CC0000的红色，这里使用十六进制值来表示。指定颜色后，水平线就失去了3D效果。





基本语法	<code>&lt;ul &gt;</code>	
	<code>&lt;li&gt;项目一&lt;/li&gt;</code> (可省略)	
	<code>&lt;li&gt;项目二&lt;/li&gt;</code> (可省略)	
	<code>&lt;li&gt;项目三&lt;/li&gt;</code> (可省略)	
	...	
	<code>&lt;/ul &gt;</code>	
功能	插入一张图像	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Type	项目符号类型
	Class	用一个名称来标记无序列表，该标记名称指向一个预定义的类，而该类是在文档级声明的或者在外部定义的样式表为无序列表创建一个标记，将来超链接可以用这个标记来明确地引用该无序列表，以便作为样式表选择器和使用其他应用程序来执行自动搜索
	Id	为无序列表创建一个标记，将来超链接可以用这个标记来明确地引用该无序列表，以便作为样式表选择器和使用其他应用程序来执行自动搜索
	Style	创建无序列表内容的内联样式
	Title	给无序列表加上说明性的文字
Compact	紧凑无序列表	

`<ul >`标签向浏览器表示下面的内容，即以`</ul >`标签结束的内容是一个无序的条目列表。在此无序列表中的每个条目都由一个前导的`<li >`标签进行定义。另外几乎所有的元素，包括其他列表、文本和多媒体元素都会以HTML格式显示。

因为列表元素的末尾通常可以通过周围的文档结构推断出来，所以大多数作者会在使用HTML时省略`</li >`标签。这样做是很有意义的，因为这样可以更容易地在列表中添加、删除和移动元素。虽然`<li >`标签是通用的，但是在不同的列表类型中使用`<li >`标签还是会有一定的差别和限制。在无序和有序列表中，`<li >`标签后面几乎可以跟随任何内容，包括其他列表和许多段落等。一般情况下，如果浏览器能够处理缩进，那么它将连续缩进嵌套的列表，并且这些条目的内容会调整到与最里面的缩进边界对齐。

一般情况下，浏览器会在每个条目前添加一个项目符号，并让其占用一行，每行会针对文档的左边界缩进一定距离。在各种流行的浏览器中，无序列表的实际显示效果都很相似，但是标准并没有对此进行规定。

原始文件	Sample\Ch05\2\1.html
最终文件	Sample\Ch05\2\1-end.html
学习要点	使用无序列表

## 1. 范例代码

```

<html >
<head >
...
</head >
<body >
...
<ul >
<li><font color="#cc3300">E-mail :
xxxxxxx@chinadownfeather.com</
font></li >
<li><font color="#cc3300">
电话：0571-xxxxxxx</font></li >
<li><font color="#cc3300">传真：
0571-xxxxxxx </font></li >
</ul >
...
</body >
</html >

```

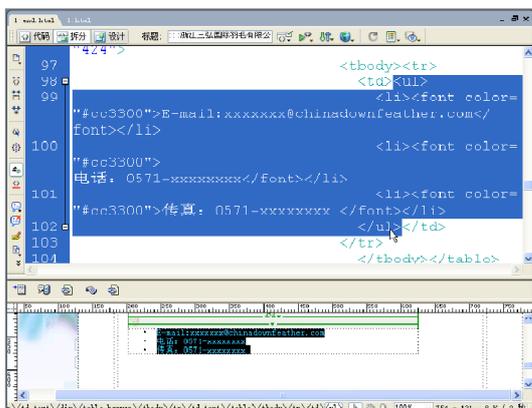
## 2. 范例解释

01 设置无序列表的开始。

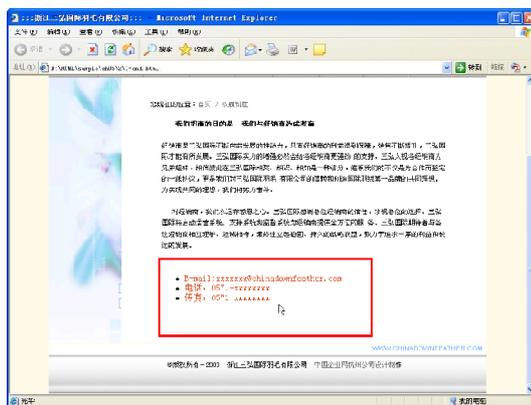
02 设置无序列表的内容分别为“E-mail : xxxxxx@chinadownfeather.com”、“电话：0571-xxxxxxx”、“传真：0571-xxxxxxx”。

03 设置无序列表的结束。

## 3. 代码和设计



## 4. 显示结果



浏览器会自动为每个使用<li>标签的无序列表的条目添加项目符号,例如Netscape Navigator和Internet Explorer都使用实心的圆点作为项目符号。支持HTML 3.2及其后继版本(包括4.0和4.01)的浏览器,都允许用Type属性来指定出现在无序列表条目前的项目符号,此属性的值包括disc、circle和square,使用Type属性后,列表中的所有条目都将使用指定的项目符号,除非有个别条目忽略了列表项目符号类型。Type的属性值如下表所示。

Type属性值及说明	Type属性值	说明
	Disc	●
	Circle	○
	Square	■





原始文件	Sample\Ch05\2\2.html
最终文件	Sample\Ch05\2\2-end.html
学习要点	使用<ul>标签的type属性

### 1. 范例代码

```

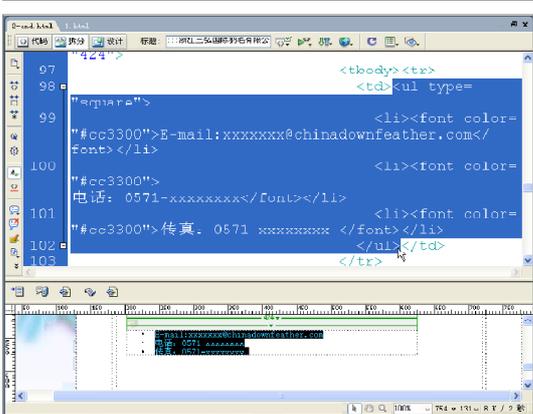
<html>
<head>
...
</head>
<body>
...
<ul type="square">
<li><font color="#cc3300">E-mail:
xxxxxxx@chinadownfeather.com</fo-
nt></li>
<li><font color="#cc3300">
电话: 0571-xxxxxxx</font></li>
<li><font color="#cc3300">传真:
0571-xxxxxxx </font></li>
</ul>
...
</body>
</html>

```

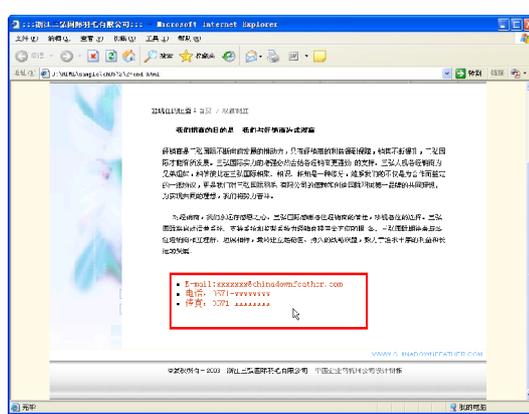
### 2. 范例解释

- 01 设置无序列表的开始，并设置类型为方形。
- 02 设置无序列表的内容，分别为“E-mail: xxxxxxx@chinadownfeather.com”、“电话: 0571-xxxxxxx”和“传真: 0571-xxxxxxx”。
- 03 设置无序列表的结束。

### 3. 代码和设计



### 4. 显示结果



Compact 属性会告诉浏览器要将无序列表压缩成更小、更紧凑的文本块，一般来说，此时浏览器会减少列表条目之间的行距。如果浏览器还要进行缩进，那么还有可能会减少列表条目之间的缩进。

原始文件	Sample\Ch05\2\3.html
最终文件	Sample\Ch05\2\3-end.html
学习要点	使用<ul>标签的compact属性

## 1. 范例代码

```

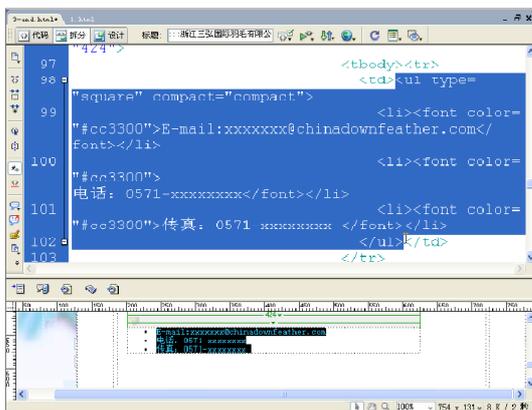
<html>
<head>
...
</head>
<body>
...
<ul type="square" compact="compact">
<li><font color="#cc3300">E-mail :
xxxxxxx@chinadownfeather.com</
font></li>
<li><font color="#cc3300">
电话: 0571-xxxxxxx</font></li>
<li><font color="#cc3300">传真:
0571-xxxxxxx </font></li>
</ul>
...
</body>
</html>

```

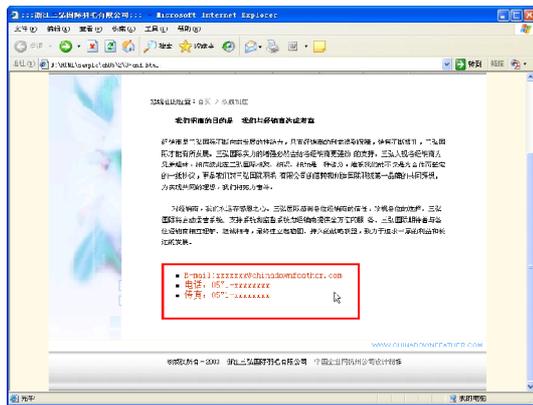
## 2. 范例解释

- 01 设置无序列表的开始，类型为方形，并设置无序列表为紧凑列表。
- 02 设置无序列表的内容，分别为“E-mail : xxxxxx@chinadownfeather.com”、“电话: 0571-xxxxxxx”和“传真: 0571-xxxxxxx”。
- 03 设置无序列表的结束。

## 3. 代码和设计



## 4. 显示结果



## 有序列表

如果列表条目的顺序非常重要，这时就应该使用有序列表。指令列表就是一个很好的例子，还有目录和文档脚注或尾注列表也是很好的有序列表的例子。





基本语法	<code>&lt;ol &gt;</code>	
	<code>&lt;li&gt;项目一&lt;/li&gt;</code> (可省略)	
	<code>&lt;li&gt;项目二&lt;/li&gt;</code> (可省略)	
	<code>&lt;li&gt;项目三&lt;/li&gt;</code> (可省略)	
	...	
	<code>&lt;/ol &gt;</code>	
功能	定义一个有序列表	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Type	项目符号类型
	Class	用一个名称来标记有序列表，该标记名称指向一个预定义的类，而该类是在文档级声明的或者在外部定义的样式表
	Id	为有序列表创建一个标记，应用超链接时可以用这个标记来明确地引用该有序列表，以便作为样式表选择器或使用其他应用程序来执行自动搜索
	Style	创建有序列表内容的内联样式
	Title	给有序列表加上说明性的文字
	Compact	紧凑有序列表
Start	起始数字	

浏览器一般都会将有序列表内容的格式设置成与无序列表类型相似的外观，惟一的不同在于列表条目前面添加的是编号，而不是项目符号。编号从第一个条目开始依次向后递增。

原始文件	Sample\Ch05\2\4.html
最终文件	Sample\Ch05\2\4-end.html
学习要点	使用有序列表

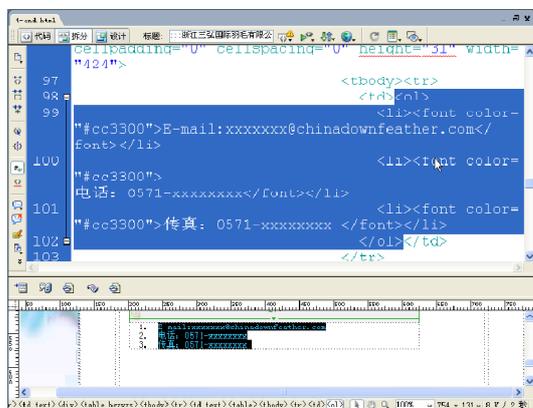
## 1. 范例代码

```
<html >
<head>
...
</head>
<body>
...
<ol >
<li><font color="#cc3300">E-mail :
xxxxxxx@chinadownfeather.com</
font></li >
<li><font color="#cc3300">
电话：0571-xxxxxxx</font></li >
<li><font color="#cc3300">传真：
0571-xxxxxxx </font></li >
</ol >
...
</body>
</html >
```

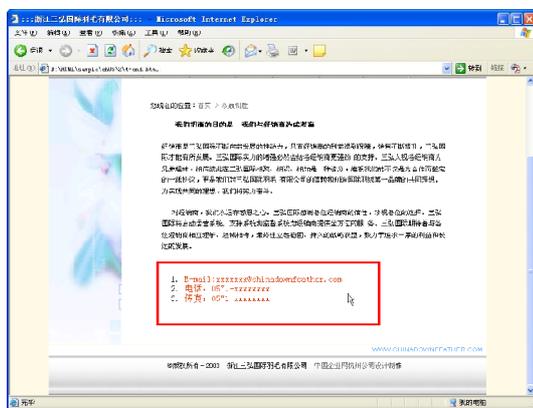
## 2. 范例解释

- 01 设置有序列表的开始。
- 02 设置有序列表的内容分别为“E-mail : xxxxxx@chinadownfeather.com”、“电话：0571-xxxxxxx”和“传真：0571-xxxxxxx”。
- 03 设置有序列表的结束。

## 3. 代码和设计



## 4. 显示结果



在HTML4 中可以改变列表的开始值，并可以在5 种不同的编号样式中选择。默认情况下，浏览器使用阿拉伯数字编号。在<ol>标签中，type 属性值A 代表使用大写字母编号，a 代表使用小写字母编号，I 代表使用大写罗马数字编号，i 代表使用小写罗马数字编号，1 代表使用阿拉伯数字编号，具体属性值如下表所示。

Type属性值及说明	Type属性值	说明
	1	数字1、2、3.....
	a	小写字母a、b、c.....
	A	大写字母A、B、C.....
	i	小写罗马数字、ii、iii.....
	I	大写罗马数字、II、III.....

原始文件	Sample\Ch05\2\5.html
最终文件	Sample\Ch05\2\5-end.html
学习要点	使用<ol>标签的type属性

## 1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<ol type="I">
<li><font color="#cc3300">E-mail:
xxxxxxx@chinadownfeather.com</
font></li>
<li><font color="#cc3300">
电话: 0571-xxxxxxx</font></li>
<li><font color="#cc3300">传真:
0571-xxxxxxx </font></li>
</ol>
...
</body>
</html>

```

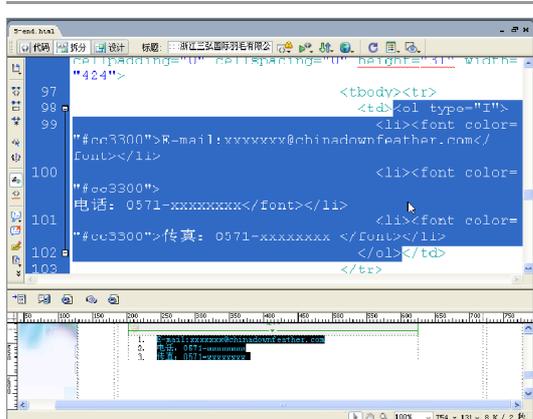
## 2. 范例解释

- 01 设置有序列表的开始，并设置编号样式为大写罗马字母。
- 02 设置有序列表的内容，分别为“E-mail: xxxxxx@chinadownfeather.com”、“电话: 0571-xxxxxxx”和“传真: 0571-xxxxxxx”。
- 03 设置有序列表的结束。

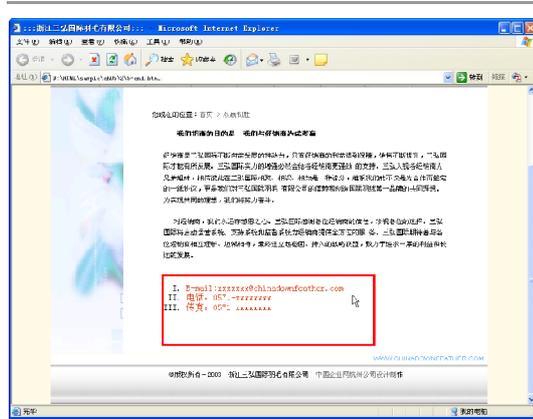




### 3. 代码和设计



### 4. 显示结果



通常情况下，浏览器会从阿拉伯数字“1”开始自动进行编号。<ol>标签的 start 属性允许用户改变开始值。start和type扩展属性需要按前后顺序使用。Start属性用于设置有序列表条目的整数计数器的开始值；type 属性则用于设置实际的编号样式。下面给出的有序列表的条目编号是从 3 开始的，但是因为编号样式被设置为 I，所以第一个编号为大写的罗马数字“III”。随后的条目将以相同的样式进行编号，并以 1 为增量来递增编号。

原始文件	Sample\Ch05\2\6.html
最终文件	Sample\Ch05\2\6-end.html
学习要点	使用有序列表的start属性

#### 1. 范例代码

```

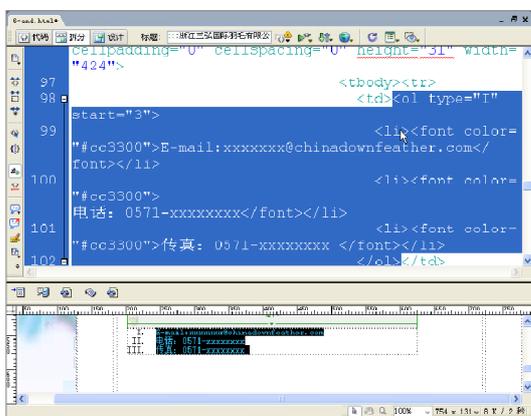
<html>
<head>
...
</head>
<body>
...
<ol type="I" start="3">
<li><font color="#cc3300">E-mail:
xxxxxxx@chinadownfeather.com</
font></li>
<li><font color="#cc3300">
电话: 0571-xxxxxxx</font></li>
<li><font color="#cc3300">传真:
0571-xxxxxxx </font></li>
</ol>
...
</body>
</html>

```

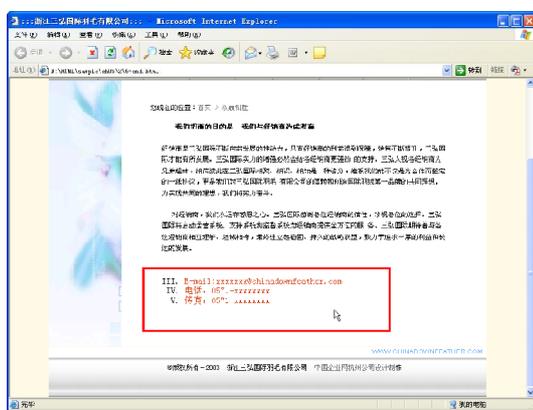
#### 2. 范例解释

- 01 设置有序列表的开始，类型为大写罗马数字，起始数字从 3 开始。
- 02 设置有序列表的内容，分别为“E-mail: xxxxxx@chinadownfeather.com”、“电话: 0571-xxxxxxx”和“传真: 0571-xxxxxxx”。
- 03 设置有序列表的结束。

## 3. 代码和设计



## 4. 显示结果



和无序列表的<ul>标签一样，有序列表的<ol>标签也有可选的 Compact 属性。

原始文件	Sample\Ch05\2\7.html
最终文件	Sample\Ch05\2\7-end.html
学习要点	使用有序列表的compact属性

## 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...
<ol type="I" start="3"
compact="compact">
<li><font color="#cc3300">E-mail:
xxxxxxx@chinadownfeather.com</
font></li >
<li><font color="#cc3300">
电话: 0571-xxxxxxx</font></li >
<li><font color="#cc3300">传真:
0571-xxxxxxx </font></li >
</ol >
...
</body >
</html >
```

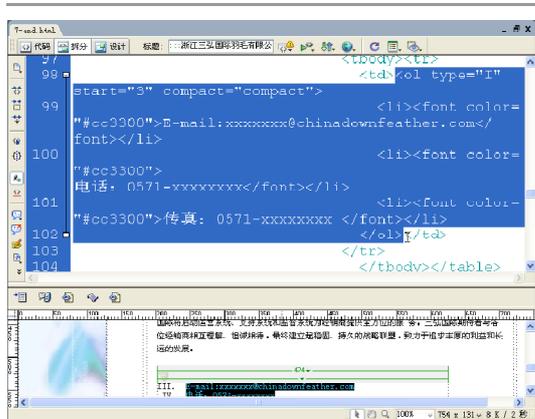
## 2. 范例解释

- 01 设置有序列表的开始，类型为大写罗马数字，起始数字从3开始，并设置为紧凑有序列表。
- 02 设置有序列表的内容，分别为“E-mail: xxxxxxx@chinadownfeather.com”、“电话: 0571-xxxxxxx”和“传真: 0571-xxxxxxx”。
- 03 设置有序列表的结束。

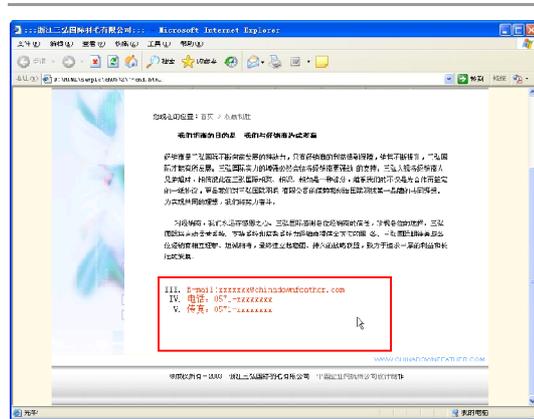




### 3. 代码和设计



### 4. 显示结果



## 目录列表

目录列表是无序列表的一种特殊形式。HTML 的设计者最初创建<dir>标签时，是希望将其应用于显示文件列表，同时，浏览器中的各种列表元素能够尽量短小，最好不要超过 20 个字符。有些浏览器会以多列格式显示元素，而可能不使用前导的项目符号。

基本语法	<dir>	
	<li>项目一</li> (可省略)	
	<li>项目二</li> (可省略)	
	<li>项目三</li> (可省略)	
	...	
</dir>		
定义一个目录列表		
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记目录列表，该标记名称指向一个预定义的类，而该类是在文档级声明的或者在外部定义的样式表为目录列表创建一个标记，应用超链接时可以用这个标记来明确地引用该目录列表，以便作为样式表选择器或使用其他应用程序来执行自动搜索
	Id	创建目录列表内容的内联样式
	Title	给目录列表加上说明性的文字

<li>标签可以在列表中定义条目，正如在前面所讨论的那样，<li>标签既可以用于定义有序列表 (<ol>) 条目，也可以用于定义无序列表 (<ul>) 条目。不仅如此，<li>标签还可以用于定义目录列表 (<dir>)。但是在目录列表中使用<li>标签时，<li>标签中不能包含任何块元素，例如段落、其他列表、预先格式化的文本或者表单等。

原始文件	Sample\Ch05\2\8.html
最终文件	Sample\Ch05\2\8-end.html
学习要点	使用目录列表

## 1. 范例代码

```

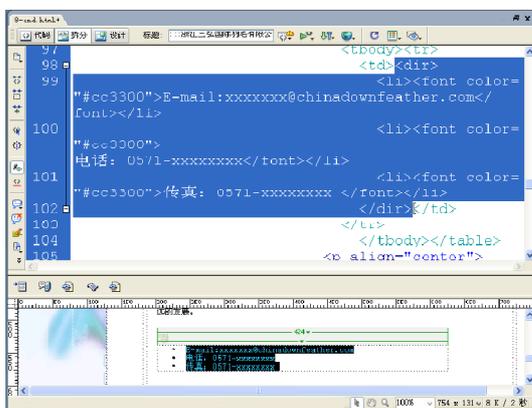
<html>
<head>
...
</head>
<body>
...
<dir>
<li><font color="#cc3300">E-mail:
xxxxxxx@chinadownfeather.com</
font></li>
<li><font color="#cc3300">
电话: 0571-xxxxxxx</font></li>
<li><font color="#cc3300">传真:
0571-xxxxxxx </font></li>
</dir>
...
</body>
</html>

```

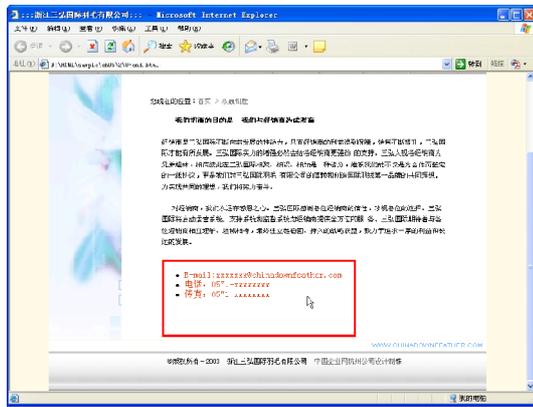
## 2. 范例解释

- 01 设置目录列表的开始。
- 02 设置目录列表的内容，分别为“E-mail: xxxxxxx@chinadownfeather.com”、“电话: 0571-xxxxxxx”和“传真: 0571-xxxxxxx”。
- 03 设置目录列表的结束。

## 3. 代码和设计



## 4. 显示结果



## 定义列表

HTML 还支持一种完全不同于前面所讨论的有序列表和无序列表的列表样式——定义列表。与字典或百科全书中的条目一样，定义列表的条目也可以带有文本、图片或其他多媒体元素，因此定义列表是表现术语表、名次列表或其他名称/值样式列表的理想表现方式。





基本语法	<code>&lt;dl &gt;</code>	
	<code>&lt;dt &gt;</code> 名词一 <code>&lt;dd &gt;</code> 解释一	
	<code>&lt;dt &gt;</code> 名词二 <code>&lt;dd &gt;</code> 解释二	
	<code>&lt;dt &gt;</code> 名词三 <code>&lt;dd &gt;</code> 解释三	
	...	
	<code>&lt;/dl &gt;</code>	
功能	定义一个定义列表	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记定义列表，该标记名称指向一个预定义的类，而该类是在文档级声明的或者在外部定义的样式表
	Id	为定义列表创建一个标记，应用超链接时可以用这个标记来明确地引用该定义列表，以便作为样式表选择器或使用其他应用程序来执行自动搜索
	Style	创建定义列表内容的内联样式
	Title	给定义列表加上说明性的文字
	Compact	紧凑定义列表

定义列表用`<dl >`和`</dl >`标签包围。在标签中，定义列表的每个条目都由两部分组成：术语及其定义或解释。`<dl >`列表中每一项的名称不再使用`<li >`标签，而是用`<dt >`标签来标记，后面跟随着用`<dd >`标签标记的条目的定义或解释。

`<dt >`标签定义了组成定义列表的术语部分，此标签只在定义（`<dl >`）列表中使用时才有效。它出现在定义列表的术语或条目前，其后跟随着`<dd >`标签标记的条目的定义或解释。一般来讲，`<dt >`标签后面的定义术语都很短小，一般只有一个或几个词语；但从技术角度讲，它可以是任意长度的。如果很长，浏览器可能会将部分条目延伸出去，或转到下一行，从定义开始的位置开始显示。由于`<dt >`标签的结尾正好对应着`<dd >`标签的开始，所以`<dt >`标签的结尾位置很明确，因此HTML文档中不要求必须有`</dt >`结束标签。

`<dd >`标签用于标记定义列表中条目的定义或解释部分的开始处。按照HTML标准的规定，`<dd >`标签仅在定义（`<dl >`）列表中使用，并紧跟着`<dt >`标签和术语，其后是定义或解释。除非使用样式表规则来改变显示属性，否则浏览器一般会在左边界显示条目或术语的名称，并在下一行缩进显示其定义或解释。如果定义术语非常短（一般少于3个字符），那么浏览器可能会将定义的第一部分与该术语放在同一行中。

与其他类型的列表一样，在定义列表的条目之间也可以添加更多的空间，方法是在定义列表的条目之间插入段落`<p >`标签，或者为各个标签定义具有更大空间的样式。

原始文件	Sample\Ch05\2\9.html
最终文件	Sample\Ch05\2\9-end.html
学习要点	使用定义列表

## 1. 范例代码

```

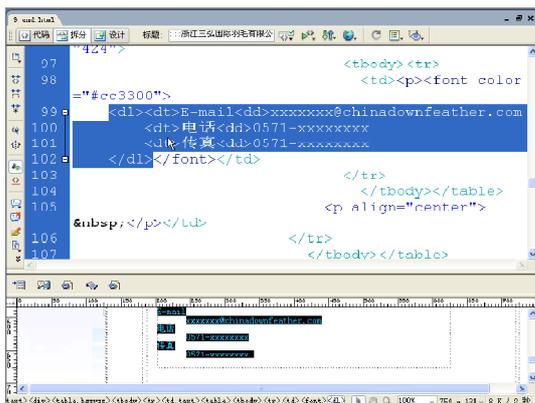
<html >
<head >
...
</head >
<body >
...
<dl >
<dt>E-mail
<dd>xxxxxxx@chinadownfeather.com
<dt>电话
<dd>0571-xxxxxxx
<dt>传真
<dd>0571-xxxxxxx
</dl >
...
</body >
</html >

```

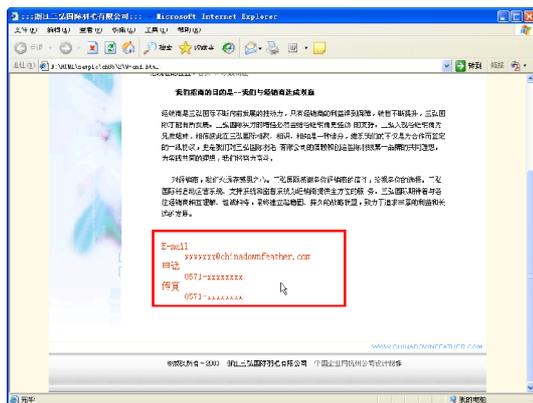
## 2. 范例解释

- 01 设置定义列表的开始。
- 02 第8、10、12行设置定义列表的名词，分别为“E-mail”、“电话”、“传真”。
- 03 第9、11、13行设置定义列表的解释，分别为“xxxxxxx@chinadownfeather.com”、“0571-xxxxxxx”和“0571-xxxxxxx”。
- 04 设置定义列表的结束。

## 3. 代码和设计



## 4. 显示结果



## 菜单列表

菜单列表是无序列表的另一种特殊形式。菜单列表用于显示菜单内容，设计单列的菜单等，菜单列表中的项目一般比较简短，用户浏览时非常方便，并且它还可以链接其他菜单。

基本语法	<menu>
	<li>项目一（可省略）
	<li>项目二
	<li>项目三
	...
	</menu>



(续表)

功能	定义一个菜单列表	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记菜单列表，该标记名称指向一个预定义的类，而该类是在文档级声明的或者在外部定义的样式表为菜单列表创建一个标记，应用超链接时可以用这个标记来明确地引用该菜单列表，以便作为样式表选择器或使用其他应用程序来执行自动搜索
	Id	创建菜单列表内容的内联样式
	Style	给菜单列表加上说明性的文字
	Title	

`<menu>`标签可以显示一个简短选项的列表。与普通的无序列表相比，浏览器可能会对菜单列表中的条目使用特殊的外观（一般会更加紧凑），或者使用某种图形化下拉菜单来显示。如果列表条目短到一定程度，浏览器甚至可能会以多列格式来显示，并且不会在条目前添加项目符号。

与无序列表相似，定义菜单列表条目也要使用`<li>`标签。在菜单列表中使用`<li>`标签时，`<li>`标签中不能包含任何块元素，包括段落、其他列表、预先格式化的文本或者表单等。

原始文件	Sample\Ch05\2\10.html
最终文件	Sample\Ch05\2\10-end.html
学习要点	使用菜单列表

## 1. 范例代码

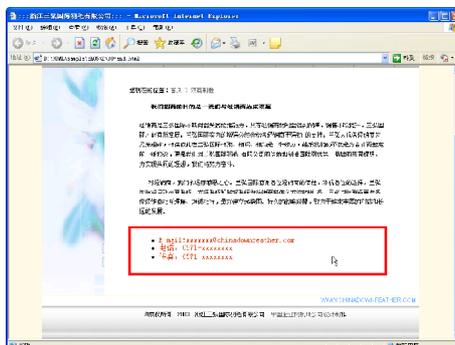
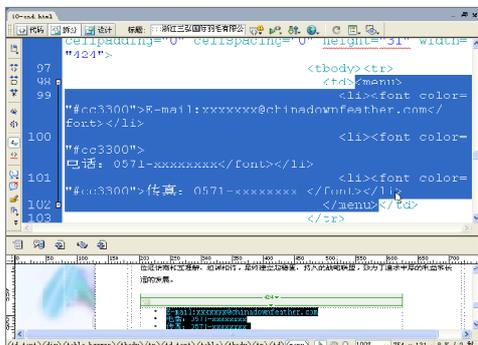
```
<html>
<head>
...
</head>
<body>
...
<menu>
<li><font color="#cc3300">E-mail:
xxxxxxx@chinadownfeather.com</font></li>
<li><font color="#cc3300">
电话: 0571-xxxxxxx</font></li>
<li><font color="#cc3300">传真:
0571-xxxxxxx </font></li>
</menu>
...
</body>
</html>
```

## 2. 范例解释

- 01 设置菜单列表的开始。
- 02 设置菜单列表的内容分别为“E-mail: xxxxxx@chinadownfeather.com”、“电话: 0571-xxxxxxx”和“传真: 0571-xxxxxxx”。
- 03 设置菜单列表的结束。

3. 代码和设计

4. 显示结果



3 实例制作

下面通过一个实例练习页面中不同类型的列表的定义以及水平线的使用。

原始文件	Sample\Ch05\3\1.html
最终文件	Sample\Ch05\3\1-end.html
学习要点	使用水平线标签和列表标签

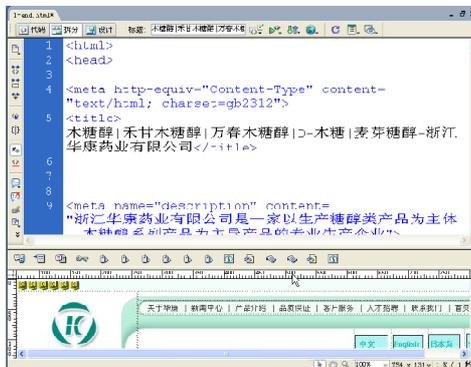


原始效果



最终效果

1 在 Dreamweaver 中打开提供的页面，单击“拆分”按钮，切换到“代码视图”和设计视图”环境。





2 将光标定位到“经营理念：”的位置，在代码视图中将其改为以下代码：

```
<DL>
<DT>
<span class="hang14"><strong>
<font color="#fd4b05">经营理念：</font></strong></span>
<p>
<DD>以其专业的精神追求品质、环保和适用性，服务于百姓健康生活的需求。
</p>
```

- ❶ 声明定义列表。
- ❷ 声明“经营理念”文字为名词。
- ❸ 声明“以其专业的精神追求品质、环保和适用性，服务于百姓健康生活的需求。”为解释。

3 将光标定位到“企业特质：”的位置，在代码视图中将其修改如下：

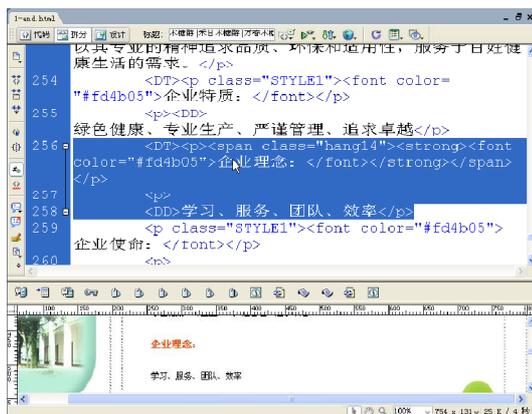
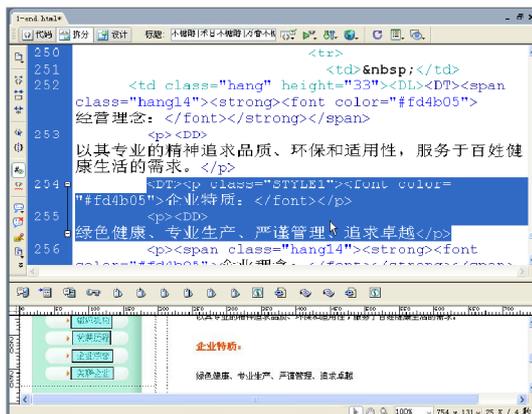
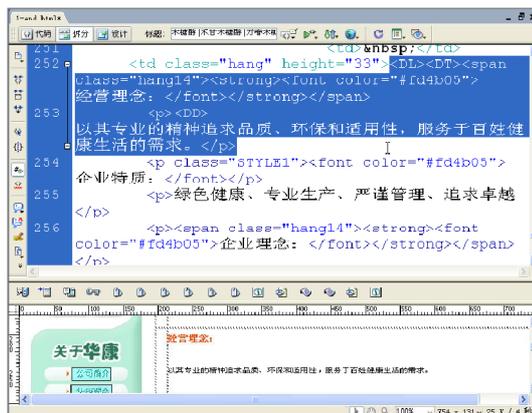
```
<DT>
<p class="STYLE1"><font color="#fd4b05">企业特质：</font>
</p>
<DD>绿色健康、专业生产、严谨管理、追求卓越
</p>
```

- ❶ 声明“企业特质”文字为名词。
- ❷ 声明“绿色健康、专业生产、严谨管理、追求卓越”为解释。

4 将光标定位到“企业理念：”的位置，在代码视图中将其修改为以下代码：

```
<DT>
<p><span class="hang14"><strong>
<font color="#fd4b05">企业理念：</font></strong></span></p>
<DD>学习、服务、团队、效率
</p>
```

- ❶ 声明“企业理念：”文字为名词。
- ❷ 声明“学习、服务、团队、效率”为解释。



5 将光标定位到“企业使命：”的位置，在代码视图中将其修改为以下代码。

```
<DT>
<p class="STYLE1"><font
color="#fd4b05">企业使命：</font>
</p>
<p>
<DD>以高度的团队凝聚力、行动力和自主
的学习进取能力为人类提供健康食品中间
体。 </p>
</DL>
```

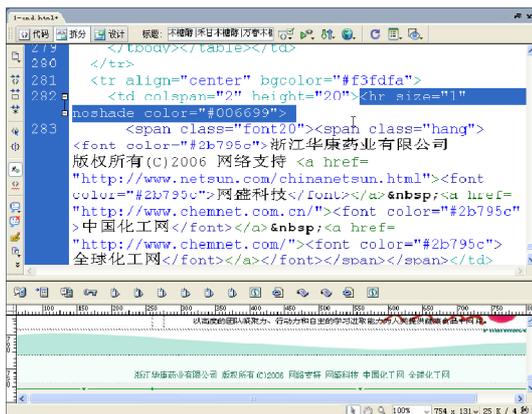
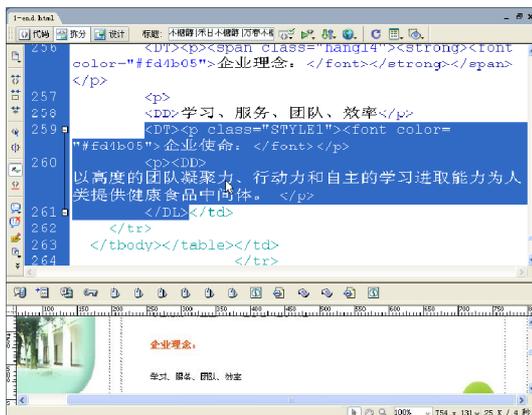
- ❶ 声明“企业使命：”文字为名词。
- ❷ 声明“以高度的团队凝聚力、行动力和自主的学习进取能力为人类提供健康食品中间体。”为解释。
- ❸ 声明定义列表结束。

6 将光标定位到“浙江华康药业有限公司”的位置，在代码视图中其前面添加以下代码：

```
<hr size="1" noshade color
="#006699">
```

- ❶ 声明粗细为 1 像素、没有阴影、颜色为 #006699 的水平线。

7 按下 F12 快捷键，可以在浏览器中看到页面的效果。





# Chapter

# 6

## 插入图片

HTML引人注目的特性之一，就是能够在文档的文本中包括图像，既可以把图像作为文档的内在对象（联机图像），也可以将其作为通过超链接下载的单独文档，或者将其作为文档的背景。合理地在文档主体内容中插入图片（静态的或者具有动画效果的图标、照片、说明、绘画等）时，会使文档变得更加生动活泼和引人入胜，并且视觉上更加专业、美观，方便浏览者阅读，更有利于信息的传播。

```
meta http-equiv="content-type" content="text/html; charset=gb2312"
<link href="1_files/css.css" rel="stylesheet" type="text/css">
```



## 1

## 网页图片的基本格式

在HTML页面中可以插入图片，网页常用的图片格式有JPEG和GIF这两种。尽管HTML没有规定图像的官方格式，然而流行的浏览器却规定了一定的图片格式：通常情况下是这两种格式。由于其他多媒体格式大多数都需要特殊的辅助应用程序，每个浏览者都需要去下载、安装并正确地操作这些应用程序，这样才能看到这些特殊的文件；因此，GIF和JPEG成为图像在Web上的实际标准也就不令人奇怪了。在Web出现以前，这两种图像格式已经得到了广泛使用，所以有大量支持软件可以帮助我们以这两种格式创建图像。这两种格式各自有其优缺点，有些浏览器会利用其特性来实现特殊的显示效果。

## GIF图像

图形交换格式(Graphics Interchange Format, GIF)最初是CompuServe为其在线服务用户传输图像而开发的。这种格式有很多特性，因此在HTML中十分普及。它的编码技术在很多平台上都可以使用，GIF的第二个主要特性是，它采用了一种特殊的压缩技术，可以显著减小图像文件的大小，从而得以在网络上更快地进行传输。而且GIF压缩是“无损”压缩，也就是说，图像中原来的数据都不会发生改变或丢失，所以解压缩并解码后的图像与原来的图像完全一样。同时，GIF图像还非常容易实现动画效果。尽管GIF图像文件都用.gif（或者是.GIF）作为文件名后缀，实际上却有两个GIF版本：原始的GIF87和扩展的GIF89a，后者支持很多新特性，包括透明背景、交叉存储和动画等，这些特性在Web创作者中使用得十分普及。现在流行的浏览器都支持这两种GIF版本，它们都是通过同一种方案来把8位的像素值映射到一个颜色表当中，这样每个图像最多可以有256种颜色。大多数GIF图像实际颜色的数目更少，有些特殊工具可以在更精细的图像中简化这些颜色，通过简化GIF图像，可以创建一个更小的颜色映像并且强化像素冗余，来使文件压缩得更多，从而使下载速度更快。然而，由于颜色数目有限，用GIF编码的图像并不是任何时候都适用，尤其是对那些要求具有照片一样逼真效果的图片来说。GIF可以用来创建非常好看的图标和颜色不多的图像及图画。因为大多数图形浏览器都明确地支持GIF格式，因此它现在是Web上接受面最广泛的图像编码格式。

GIF89a图像有3种特殊的技巧：隔行扫描、透明性和动画。通过隔行扫描，GIF89a图像可以在屏幕上一下子显现出来，而不是从上到下逐步地显示。一般情况下，用GIF编码的图像是像素数据从图像的顶部到底部顺次、逐行排列的一个序列。所以，普通的GIF图像在屏幕上显示时，就好像一下子拉开窗帘；而具有隔行扫描效果的GIF89a在显示时，则像卷起百叶窗一样。GIF89a图像另外一种常见的效果，就是它可以让图像的一部分变成透明效果，这样图像下面的内容（通常是浏览器的窗口背景）就可以透过透明部分显示出来了。透明的GIF图像在它的颜色映射里专门用一种颜色作为背景颜色，从而让显示窗口的背景透过来。通过仔细地剪切图像的大小和选择一种接近纯色的背景颜色，透明图像可以制作成看上去好像完全镶嵌在网页中，或者是浮动在上面的效果。GIF89a格式图像的第3个独特之处是，它可以实现简单的逐帧动画。通过使用特殊的GIF动画工具软件，就可以把一系列GIF

图像放在一个单独的 GIF98a 文件中，浏览器会相继显示文件中的每个图像。在 GIF 文件中，每个图像之间都具有特殊的控制部分，可以用来设置浏览器从头至尾显示整个序列（循环）的次数，每两个图像之间停顿的时间，以及在浏览器显示后面一个图像之前是否从背景中抹去图像空间等。

## JPEG 图像

联合图像专家组(Joint Photographic Experts Group, JPEG)是开发我们现在所使用的 JPEG 图像编码格式的标准化组织。和 GIF 一样，JPEG 图像也是独立于平台的；为了能通过数字通信技术来高速传播，而专门对它进行了压缩处理。与 GIF 不一样的是，JPEG 支持数以万计的颜色，可以显示更加精细甚至像照片一样逼真的数字图像。JPEG 使用的是特殊的压缩算法，从而可以实现非常高的压缩比。例如，它可以把 200KB 的 GIF 图像压缩到只有 30 KB 的 JPEG 图像，这样惊人的压缩率导致 JPEG 要损失一些图像数据。但是通过专门的 JPEG 工具便可以调整这个“损失率”，所以尽管解压缩后的图像和原来的图像并不完全一样，但它们非常接近，以至于大多数人都无法分辨出其中的差别。尽管 JPEG 对照片来说是一个不错的选择，但对插图来说就不那么合适了。JPEG 使用的压缩和解压缩算法使得它在处理大范围的颜色块时，会留下很明显的人工痕迹。因此如果想显示用线条描绘的图画，GIF 也许更适合一些。JPEG 格式通常用 .JPEG（或者 .JPG）文件名来结尾，所有浏览器都可以识别这种格式。



6

插入图片

## 2 插入图片

在页面中插入图片可以起到美化的作用。插入图片的标签只有一个，那就是 `<img>` 标签。`<img>` 标签允许在文档当前的文本流中引用或者插入图形图像。

## `<img>` 标签

基本语法	<code>&lt;img src="file_name"&gt;</code>	
功能	插入一张图像	
属性及说明	属性	说明
	Src	图像的源文件
	Alt	提示文字
	Width、height	宽度、高度
	Border	边框
	Vspace	垂直间距
	Hspace	水平间距
	Dynsrc	设定 AVI 文件的播放
Loop	设定 AVI 文件循环播放次数	



(续表)

属性	说明
Loopdelay	设定AVI文件循环播放延迟
Start	设定AVI文件播放方
Lowsrc	设定低分辨率图片
Usemap	映像地图
Dir	文本方向
Lang	语言信息
Align	对齐方式
Class	用一个名称来标记图片，该标记名称指向一个预定义的种类，而该类是在文档级声明的或者在外部定义的样式表
Id	为图片创建一个标记，应用超链接时可以用这个标记来明确地引用该图片，以便作为样式表选择器或使用其他应用程序来执行自动搜索
Style	创建图片内容的内联样式
Title	给图片加上说明性的文字

<img>标签的Src属性是必需的（除非在基于IE的浏览器中使用Dysrc）在（img）标签中，除Src是不可缺的外，其他属性都是可选的，注意只有当图像文件和HTML文档在同一目录时，才可采取相对地址来标识图像文件的位置。Src的值是图像文件的URL，也就是引用该图像文件的绝对地址或相对地址。

原始文件	Sample\Ch06\1\1.html
最终文件	Sample\Ch06\1\1-end.html
学习要点	使用<img>标签的src属性

## 1. 范例代码

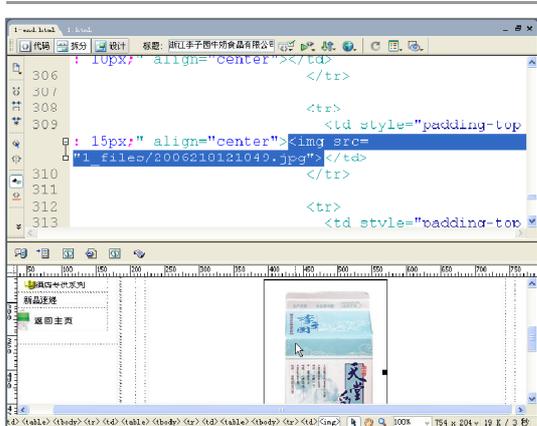
```
<html>
<head>
...
</head>
<body>
...

...
</body>
</html>
```

## 2. 范例解释

插入了1\_files文件夹中的2006218121049.jpg图片，在<img>标签的前后没有设定任何换行符或者段落分隔符，所以图像可以真正地“内联”在文本或其他内容当中。

### 3. 代码和设计



### 4. 显示结果



Alt 属性指定了替代文本，用于在图像无法显示或者用户禁用图像显示时，代替图像显示在浏览器中。这是一个可选项，但强烈推荐在文档的每个图像中都使用这个属性。这是因为即使图像无法显示，用户还是可以看到丢失了的一些信息。此外，当用户把光标移到图像上方时，新版本的浏览器会在一个文本框中显示描述性文本。相应地，可以在文档中嵌入一些简短的说明性文字，当用户把光标移动到一个小的内联图标上时，就会弹出这段文字用于提示浏览者。

原始文件	Sample\Ch06\1\2.html
最终文件	Sample\Ch06\1\2-end.html
学习要点	使用 <img&gt;标签的alt属性< td=""> </img&gt;标签的alt属性<>

#### 1. 范例代码

```
<html >
<head >
...
</head >
<body >
...

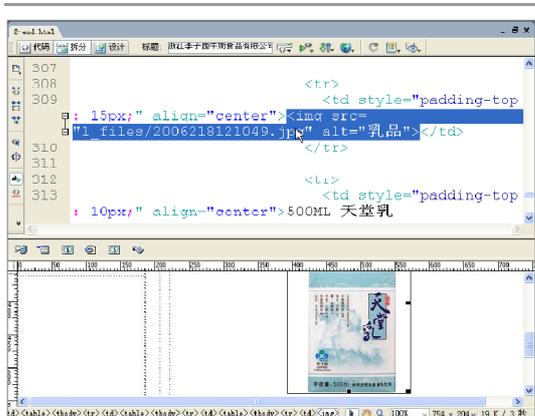
...
</body >
</html >
```

#### 2. 范例解释

插入了1\_files文件夹中的2006218121049.jpg图片，然后对插入的这张图片使用“乳品”作为提示文字。当用户把光标移动到图片上时，就会弹出这段文字，或者在图像无法显示或者用户禁止图像显示时，代替图像显示在浏览器中的提示内容为“乳品”。



### 3. 代码和设计



### 4. 显示结果



<img>标签的 Height 和 Width 属性用来指定图像的尺寸。这两个属性都要求是整数，并且以像素为单位来表示图像尺寸，它们在<img>标签中出现的次序并不重要。使用 Height 和 Width 属性的另一种技巧就是可以非常容易地实现对页面区域的填充，同时还可以改善文档的性能。设想一下，如果想在文档中放置一个彩色的横条，不需要创建一个具有完整尺寸的图像，相反，只要创建一个宽度和高度都为 1 像素的图像，并把自己希望使用的颜色赋给它，然后使用 Height 和 Width 属性把它扩展到更大的尺寸。

原始文件	Sample\Ch06\1\3.html
最终文件	Sample\Ch06\1\3-end.html
学习要点	使用<img>标签的width和height属性

#### 1. 范例代码

```

<html>
<head>
...
</head>
<body>
...

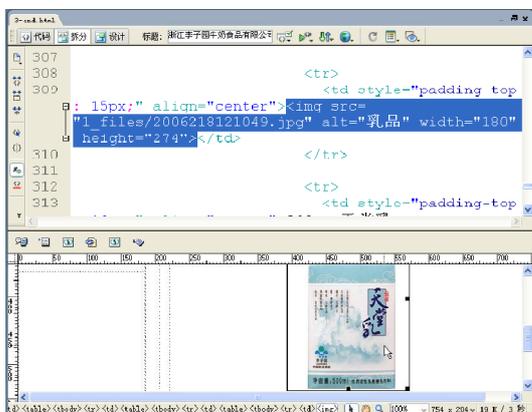
...
</body>
</html>

```

#### 2. 范例解释

插入了1\_files文件夹中的2006218121049.jpg 图片，然后设置插入图片的宽度为 180 像素，高度为 274 像素。这样设置使得浏览器在下载图像之前就为其预留出了位置，从而可以加速文档的显示，而且还可以避免文档内容的移动。

### 3. 代码和设计



### 4. 显示结果



浏览器通常会把代表超链接的图像显示在两个像素宽的边框里面，以提示浏览者可以通过选择这个图像来访问相关联的文档。使用Border 属性和一个用像素标识的宽度值就可以去掉或者加宽图像的边框。通过在

原始文件	Sample\Ch06\1\4.html
最终文件	Sample\Ch06\1\4-end.html
学习要点	使用<img>标签的border属性

#### 1. 范例代码

```
<html >
<head>
...
</head>
<body>
...

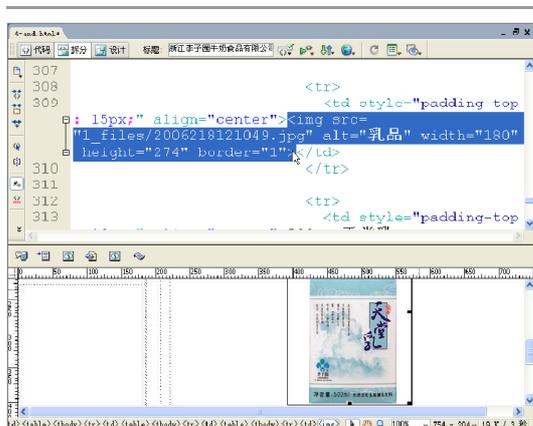
...
</body>
</html >
```

#### 2. 范例解释

插入了1\_files文件夹中的2006218121049.jpg 图片，然后将插入的这张图片的边框设置为 1 像素。



### 3. 代码和设计



### 4. 显示结果



HTML 图像在行中出现时通常只伴有一行文字。而通常的印刷媒体，例如杂志则将文字在图像的周围进行环绕，这样就会有更多行文字与图像相邻，而不只是一行。设计者可以通过 `<img>` 标签的 `Align` 属性来控制带有文字环绕的图像的对齐方式。

Align属性值及说明	Align属性值	说明
	Top	文字的中间线位于图片上方
	Middle	文字的中间线位于图片中间
	Bottom	文字的中间线位于图片底部
	Left	图片在文字的左侧
	Right	图片在文字的右侧
	Absbottom	文字的底线位于图片底部
	Absmiddle	文字的底线位于图片中间
	Baseline	英文文字基准线对齐
	Texttop	英文文字上边线对齐
Top	文字的中间线位于图片上方	

**Top:** 图像的顶部将与本行文字中最高一项的顶部对齐，如果本行中没有其他图像，那么这张图像的上沿将与文字的上沿对齐。

**Middle:** 把图像的正中央与当前行中最高元素的正中央对齐，这个元素可能是文字，也可能是图像。

**Absmiddle:** 如果把 `<img>` 标签的 `Align` 属性设置为 `Absmiddle` 的话，那么浏览器将把图像的绝对中间与当前行的绝对中间对齐。

**Center:** `center` 图像对齐值在浏览器中可得到与 `Absmiddle` 同样的效果。

**Bottom** 和 **Baseline:** 浏览器把图像的底部和文本的底部进行对齐。在 IE 中，`Bottom` 和 `Absbottom` 的效果是一样的。

**Absbottom:** “`align=absbottom`” 属性会告诉浏览器要把图像的底部和当前文本行真正的底部相对齐。这个真正的底部考虑了字母向下延伸的部分，也就是文本的最低点，即使当前行中可能没有这样向下延伸的字母。

**Texttop:** “`align=texttop`” 属性和值告诉浏览器要把图像的上部与当前行中最高文本项的上部对齐。这个选项与 `Top` 选项不同，因为 `Top` 选项会将图像与本行中任何最高的项（图

像或文字)的上部对齐。如果当前行中没有其他图像超过文本最上沿的高度,那么Texttop与Top的效果相同。

图像的Left和Right对齐方式将告诉浏览器,要分别把图像紧贴着当前文本流的最左边或者最右边的边界摆放,然后浏览器会在与图像相邻的地方显示文本流中其余文档的内容,最终的效果就是靠近图像的文档内容在图像周围形成了环绕。

原始文件	Sample\Ch06\1\5.html
最终文件	Sample\Ch06\1\5-end.html
学习要点	使用 <img&gt;标签的align属性< td=""> </img&gt;标签的align属性<>

## 1. 范例代码

```
<html>
<head>
...
</head>
<body>
...

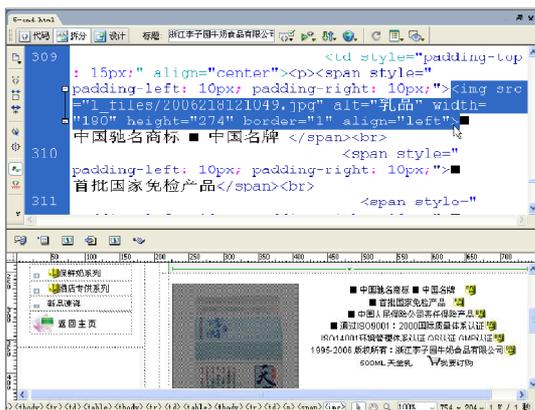
■ 中国驰名商标 ■ 中国名牌……
...
</body>
</html>
```

## 2. 范例解释

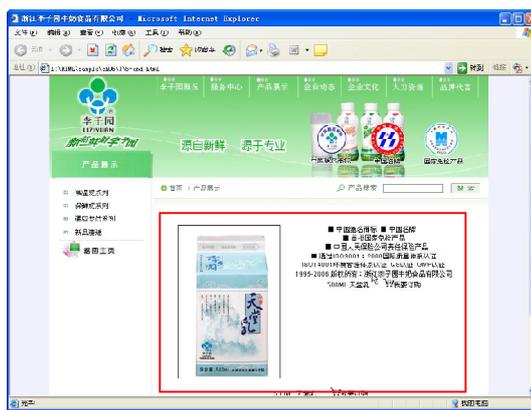
01 插入了1\_files文件夹中的2006218121049.jpg图片,然后将插入的这张图片的对齐方式设置为居左。

02 “■ 中国驰名商标 ■ 中国名牌……”的文字会环绕在图像的右侧。

## 3. 代码和设计



## 4. 显示结果



通常图形浏览器不会在图像和其周围的文字之间留出很多空间。除非创建一个透明的图像边框来扩大这些间距,否则图像和其周围文字之间默认仅有两个像素的距离,对于大多数设计者来说太近了。如果把图像放在超链接中,特殊颜色的边框会把留出的所有间距都占满了,这会使人们注意到文字与图像之间太近了。通过设置Hspace属性,可以以像素为单位指定图



6

插入图片



像左边或右边的文字与图像之间的间距；而 Vspace 属性值则是上面或下面的文字与图像之间距离的像素值。

原始文件	Sample\Ch06\1\6.html
最终文件	Sample\Ch06\1\6-end.html
学习要点	使用<i mg>标签的hspace和vspace属性

### 1. 范例代码

```

<html >
<head >
...
</head >
<body >
...

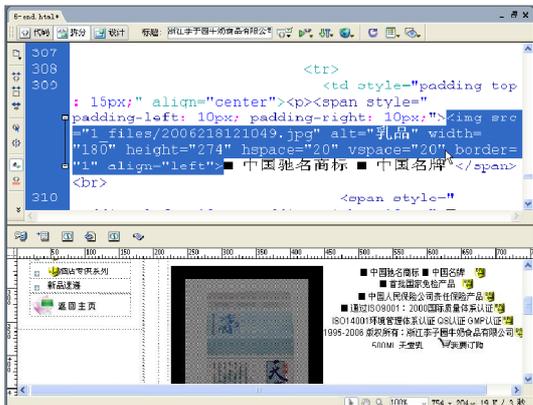
...
</body >
</html >

```

### 2. 范例解释

插入了1\_files文件夹中的2006218121049.jpg 图片，然后设置插入的这张图片的水平间距和垂直间距为 20 像素，hspace 和 vspace 属性可以给图像一个自由呼吸的空间。

### 3. 代码和设计



### 4. 显示结果



## 3 实例制作

下面通过一个实例练习制作页面中图文混排的效果。

原始文件	Sample\Ch06\2\1.html
最终文件	Sample\Ch06\2\1-end.html
学习要点	使用图片标签及相关属性

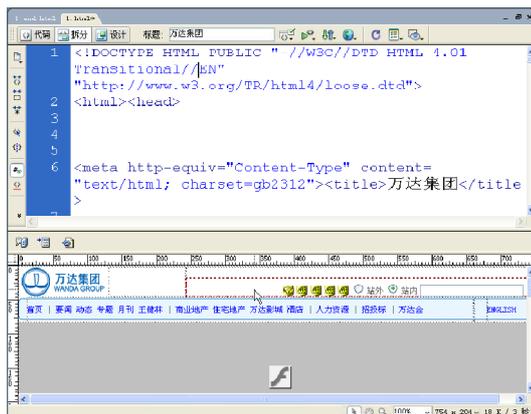


原始效果



最终效果

- 1 在 Dreamweaver 中打开提供的页面，单击“拆分”按钮，切换到“代码视图”和设计视图”环境。



- 2 将光标定位到“上海万达商业广场”下面空白的单元格中如下代码的位置：

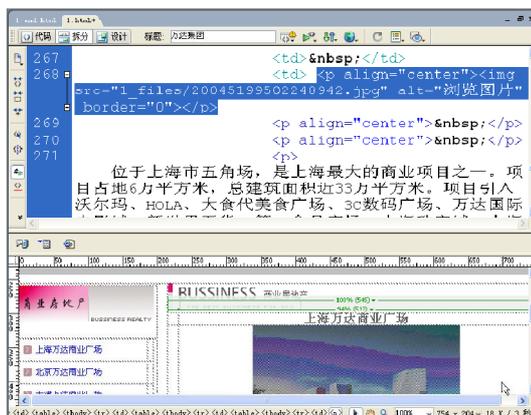
```
<p align="center">&nbsp;&nbsp;&nbsp;</p>
```

在代码视图中将其修改为以下代码：

```
<p align="center">

</p>
```

- 1 插入了1\_files文件夹下的20045199502248942.jpg 图片，设置边框为0，提示文字为“浏览图片”。



6

插入图片



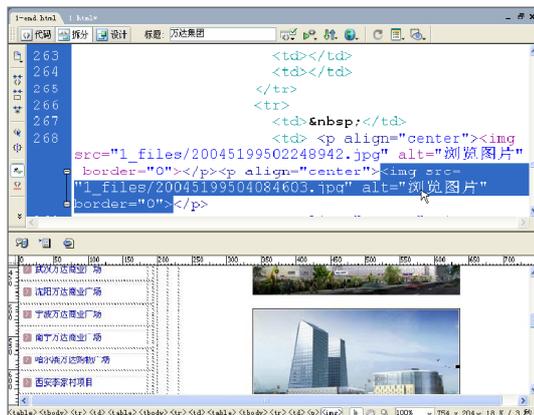
3 将光标定位到刚才插入的图片下方的如下代码的位置:

```
<p align="center">&nbsp;&nbsp;&nbsp;</p>
```

在代码视图中将其修改为以下代码:

```
<p align="center">  
  
</p>
```

- 1 插入了1\_files文件夹下的20045199504084603.jpg图片,设置边框为0,提示文字为“浏览图片”。



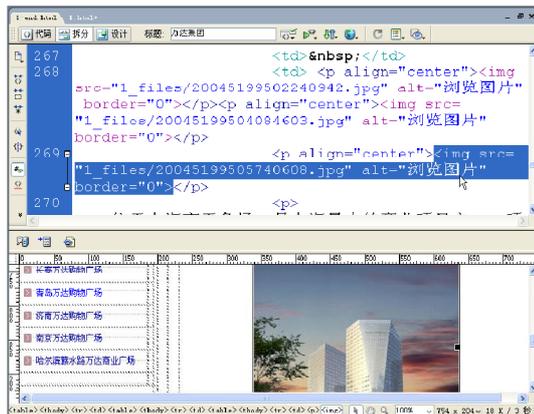
4 将光标定位到刚才插入的图片下方的如下代码的位置:

```
<p align="center">&nbsp;&nbsp;&nbsp;</p>
```

在代码视图中将其修改为以下代码:

```
<p align="center">  
  
</p>
```

- 1 插入了1\_files文件夹下的20045199505740608.jpg图片,设置边框为0,提示文字为“浏览图片”。



5 按下F12快捷键,可以在浏览器中看到页面的效果。

