



前言

我们的 HTML

HTML 给人更多的感觉是代码,不过它与其他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。

1. 范例代码

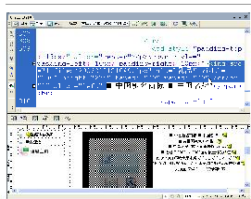
```
<html>
<head>
.....
</head>
<body>
.....

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

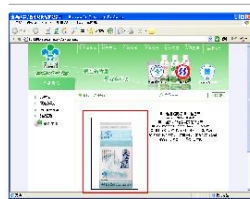
2. 范例解释

插入 1_files 文件夹中的 2006218121049.jpg 图片,然后对插入的这张图片设置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



- 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



Chapter 1

Chapter 2

Chapter 3

Chapter 4

目 录 CONTENTS

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

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

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

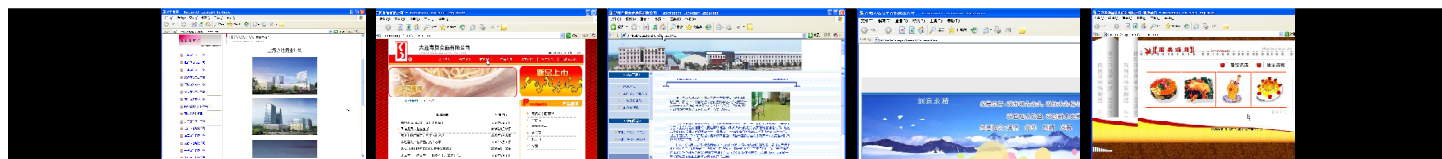
文字和段落 43

文字内容 44

 空格 44

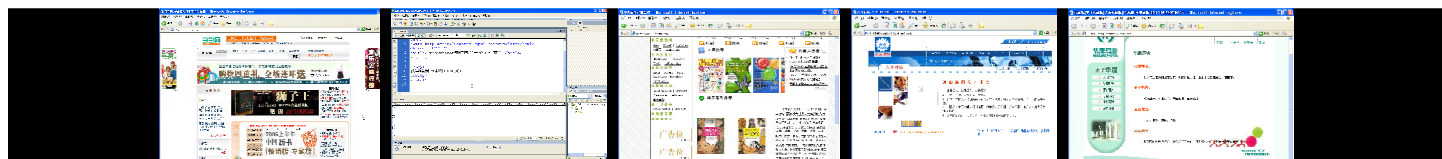
 特殊字符 45

 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
水平线和列表	83
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

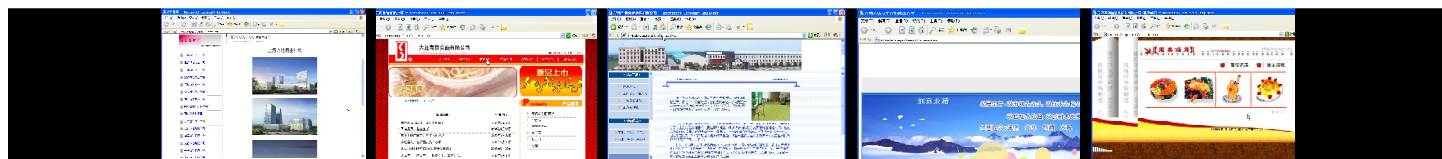


Chapter 6

Chapter 7

Chapter 8

实例制作	103
插入图片	107
网页图片的基本格式	108
GIF图像	108
JPEG 图像	109
插入图片	109
标签	109
实例制作	116
页面中的超级链接	119
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
多媒体效果	141
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

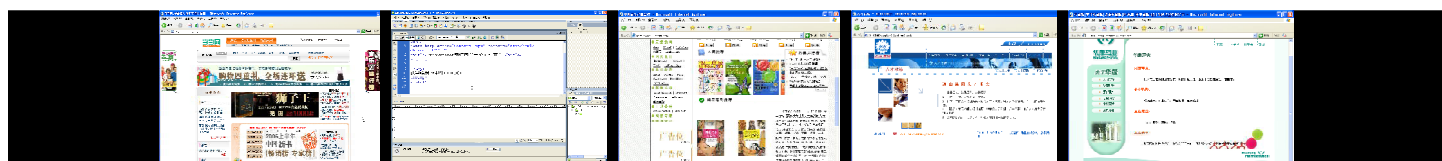


Chapter 9

Chapter 10

Chapter 11

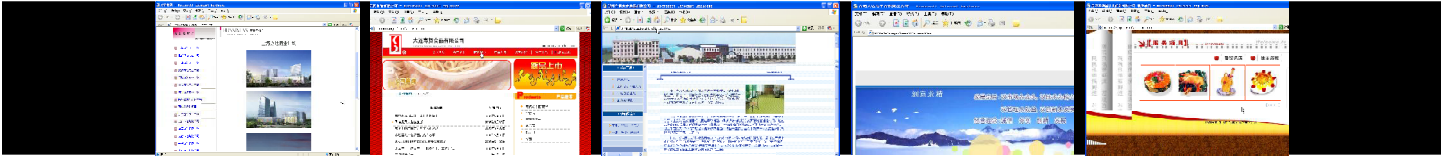
使用Param标签	160
实例制作	163
表单元素	167
什么是表单	168
Form标签	168
表单元素	171
Input标签	171
Textarea标签	184
Select标签	186
Option标签	187
Optgroup标签	190
表单元素分组	192
Label 标签	192
Fieldset标签	194
Legend标签	194
实例制作	197
使用表格排版网页	203
表格的整体结构	204
Table标签	204
Tr标签	205
Td标签	206
Th标签	207
Caption标签	209
表格的属性	211
Table标签的属性	211
Tr标签的Valign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
Ttfoot标签	239
实例制作	241
框架结构页面	249
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
使用CSS样式表将结构和样式分开	279
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
在页面中使用JavaScript脚本	345
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
	onSel ect 事件	358
	onSubmi t 事件	359
	onUnl oad 事件	360
	实例制作	361
	附录	365
	附录1	366
	附录2	367
	附录3	378
	附录4	381



Chapter

7

页面中的超级链接

到目前为止，我们都是将HTML文档作为独立的实体来看待的，只关注了那些用于构建结构和格式化工作的语言元素。而标记语言的真正威力在于它的收集能力，它可以将收集来的文档组合成一个完整的信息库，并且可以将文档库与世界上其他的文档集合链接起来。这样来浏览者不仅可以完全控制文档在屏幕上的显示内容，还可以控制浏览信息的顺序。这就是HTML中的超级链接功能，也就是它可以将整个Web网络连接起来。

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



1

关于超级链接

超文本的基本特征就是可以超链接文档；浏览者可以指向其他位置，这个位置可以在当前文档中、其他文档中，也可以在因特网上任何地理位置的文档中，这些文档组成了一个庞杂的信息网。目标文档通常与其来源有某些关联，且丰富了来源。来源中的链接元素则将这种关系传递给浏览者。超链接可以用于各种效果，可以用在目录和主题列表中。浏览者可以在浏览器窗口中单击鼠标或按下按键，从而选择并自动跳转到自己感兴趣的那个主页，或跳转到世界上某处完全不同的集合中的某个文档。

URL

正如我们前面所讨论的那样，Web 上的所有文档都有一个惟一的地址。文档的地址也就是它的统一资源定位符（Uniform Resource Locator，URL）。

书写 URL 时要使用 US-ASCII 字符集中可以显示的字符，例如一个简单的 URL 可以写作“<http://www.21books.com>”。

设计者可以用两种方式书写 URL：绝对方式或相对方式。

绝对 URL 是资源的完整地址，包含系统在 Web 上寻找文档及其所在服务器时所需的所有信息。绝对 URL 至少要包含方案和 URL 所需的全部元素。绝对路径为文件提供完全的路径，包括适用的协议例如 http、ftp 等。

而相对 URL 只是提供一个简写的文档地址，系统会自动将其与“基本地址”合并起来，生成一个完整的文档地址。URL 的任何部分在相对 URL 中都可以省略，浏览器会使用基本 URL 中相对应的元素自动填写相对 URL 中缺少的部分。基本 URL 一般是包含相对 URL 文档的 URL，但也可能是<base>标签中指定的其他文档的 URL。相对路径适合网站的内部链接：只要属于同一网站之下，即使不在同一个目录下，相对链接也非常适合。文件相对地址是书写内部链接的理想形式：只要处于站点文件夹之内，相对地址可以自由地在文件之间构建链接，这种地址形式利用的是构建链接的两个文件之间的相对关系，不受站点文件夹所处服务器位置的影响。因此这种书写形式省略了绝对地址中的相同部分，这样做有它的好处：当站点文件夹所在的服务器地址发生改变时，文件夹的所有内部链接（如果采用此种地址形式）都不会出问题。

2

制作超级链接

链接标记虽然在网站设计制作中占有不可替代的地位，但是其标记只有一个，那就是<a>标记。通常使用<a>标签的 Href 属性来创建超文本链接（或叫超链接），以链接到同一文档的其他位置或其他文档中。在这种情况下，当前文档便是链接的源；Href 属性的值（即 URL）便是目标。另一种使用<a>标签的方式是使用 Name 属性在文档中标记超链接目标或文档中的片段标识符。

在<a>标签和它的结束标签之间可以插入常规文本、换行符、图像和标题。浏览器会按常规显示这些元素，并会为它们添加一些特殊效果，以表示它们是可以链接到其他文档的超链接。

基本语法	< a >	
	链接内容	
	< / a >	
功能	定义超级链接	
属性及说明	属性	说明
	Href	指定链接地址
	Name	给链接命名
	Target	指定链接的目标窗口
	Accesskey	链接热键
	Dir	文本方向
	Lang	语言信息
	Align	对齐方式
	Class	用一个名称来标记超级链接，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	为超级链接创建一个标记，应用超链接时可以用这个标记来明确地引用该超级链接，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	创建超级链接内容的内联样式
	Title	给超级链接加上说明性的文字
	Charset	指定作为目标文档中所使用的字符编码
	Rel	指定从源文档到目标文档的关系
	Rev	指定从目标文档到源文档的关系
	Type	指定调用的源的内容类型
	TabIndex	对浏览器窗口中的对象重新排序



外部链接

所谓外部链接，指的是跳转到当前网站外部、其他网站中的页面或其他元素之间的链接关系。这种链接在一般情况下需要书写绝对的链接地址。制作外部链接的时候，使用URL统一资源定位符来定位万维网信息，这种方式可以简洁、明了、准确地描述信息所在的地点。常用的外部链接有如下格式。

URL 格式	描述
http://	进入万维网站点
ftp://	进入文件传输服务器
news://	启动新闻讨论组
telnet://	启动Telnet方式
gopher://	访问Gopher服务器
mailto://	启动邮件



进入万维网站点

到目前为止，链接到外部网站的 HTTP URL 是最普通的形式。HTTP URL 用来访问 Web 服务器上的文档，HTTP URL 的格式非常简单，具体如下。

```
http://server/path
```

上面的 URL 指定了惟一的服务器、目录路径和文档名称。

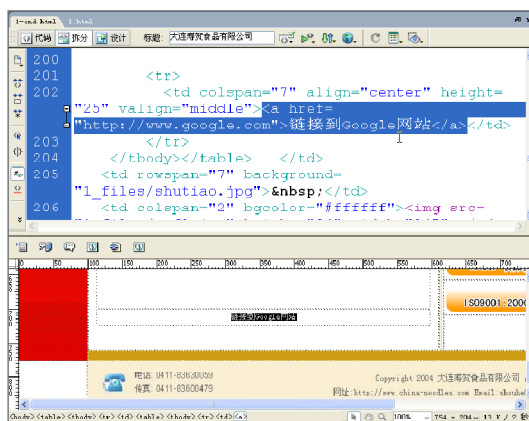
原始文件	Sample\Ch07\1\1.html
最终文件	Sample\Ch07\1\1-end.html
学习要点	链接到外部网站

1. 范例代码与解释

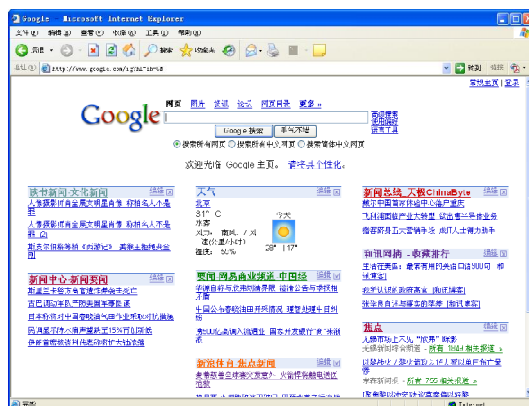
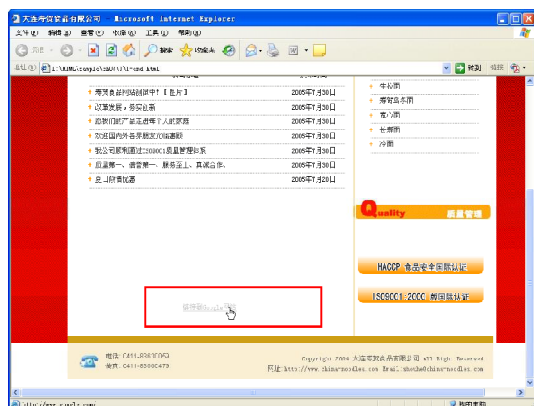
```
<html>
<head>
...
</head>
<body>
...
<a href="http://www.google.com">
  链接到 Google 网站</a>
...
</body>
</html>
```

定义“链接到 Google 网站”的链接地址为 <http://www.google.com>。

2. 代码和设计



3. 显示结果



单击超链接后，默认的浏览器窗口是原有窗口，可以指定这个超链接打开的目标窗口，如新窗口等。这需要通过设置 Target 属性实现，它的属性值如下表所示。

Target属性值及说明	Target属性值		说明
	_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

原始文件	Sample\Ch07\1\2. html
最终文件	Sample\Ch07\1\2-end. html
学习要点	使用target属性

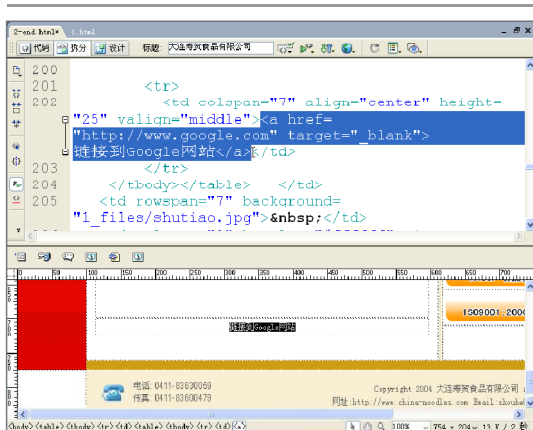
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<a href="http://www.google.com"
target="_blank">链接到Google网站</a>
...
</body>
</html>
```

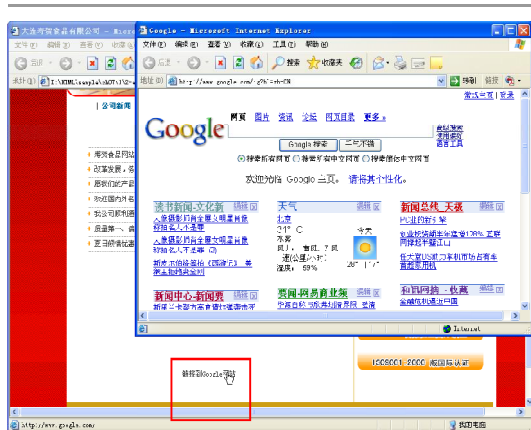
2. 范例解释

定义了“链接到Google网站”的http://www.google.com链接在新窗口中打开。

3. 代码和设计



4. 显示结果



进入文件传输服务器

Internet 上的资源非常丰富，通过文件传输协议FTP，就可以足不出户地获得各种免费软件和其他文件。FTP 即“文件传输协议”（协议是使计算机与计算机之间能够相互通讯的语言），FTP 使文件和文件夹能够在 Internet 上公开传输。在某些情况下，浏览者需要从网络计算机管理员处获得权限才能登录并访问计算机上的文件。但是通常浏览者会发现可以使





用 FTP 访问某个网络或服务器，而不必是该计算机的账户，也不必是授权的密码持有人。这些匿名 FTP 服务器存储着能够通过 FTP 获得的公开数据。

文件 URL 是第二种最常用的格式，但是 Web 用户尤其是 Web 设计者很难识别出来。它用于指向存储在计算机上的文件，而不指向用于检索文件的协议。从这点上来说，使用文件 URL 时只能限于联网环境中。文件 URL 可以加载和显示本地存储的文档，在引用那些个人的 HTML 文档集时尤为有用，例如那些“正在建设中”的页面、还不准备发布的文档、或

```
ftp://server/path
```

CD-ROM 上的文档集合等。文件 URL 使用的是下列格式。

与前面讲述的 HTTP 服务器一样，文件服务器也必需有因特网域名或包含要检索文件的计算机的 IP 地址。但是，HTTP 服务器要求使用 TCP/IP 网络协议，文件服务器则不需要，它还可能是个人网络上惟一的计算机名称，或是同一台计算机上的存储设备、（如 CD-ROM），或者是从另一台联网的计算机上映射来的名称。它们都没有规定浏览器会采用何种方式联系机器以获取文件；浏览器应该会进行一些连接来获得文件，该连接可能是通过网络文件系统或 FTP 进行的。

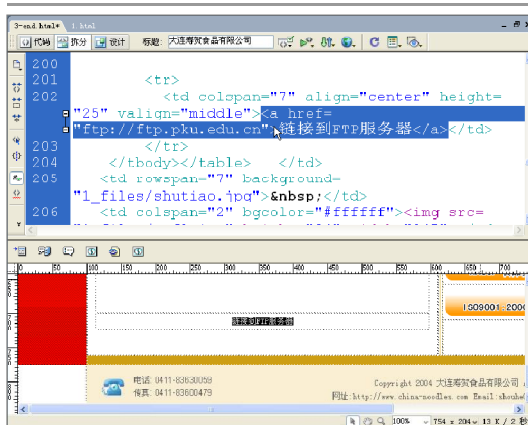
原始文件	Sample\Ch07\1\3.html
最终文件	Sample\Ch07\1\3-end.html
学习要点	链接到FTP

1. 范例代码与解释

```
<html>
<head>
...
</head>
<body>
...
<a href="ftp://ftp.pku.edu.cn">
链接到FTP服务器</a>
...
</body>
</html>
```

定义了“链接到FTP服务器”的链接地址是 `ftp://ftp.pku.edu.cn`。

2. 代码和设计



3. 显示结果



启动Telnet方式

Telnet URL 会打开与某个服务器之间的交互会话，并允许用户登录和使用该服务器。一般情况下，与该服务器的连接会自动开始对用户的特殊服务；在其他情况下，用户必须知道该输入何种命令来使用该系统。Telnet URL 的格式如下：

```
telnet://user:password@server:port /
```

Telnet URL 中用户和密码的用法与 Ftp URL 中的完全相同，应用 Telnet 服务必需要知道一台 Telnet 服务器的因特网域名或 IP 地址；而端口是指服务器侦听请求时所使用的端口。

原始文件	Sample\Ch07\1\4.html
最终文件	Sample\Ch07\1\4-end.html
学习要点	链接到Telnet

1. 范例代码

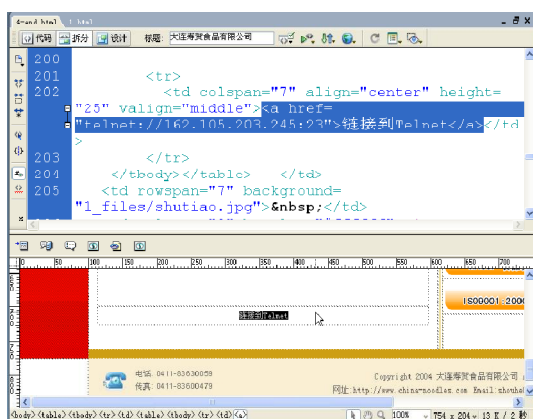
```
<html>
<head>
...
</head>
<body>
...
<a href="telnet://162.105.203.245:23">链接到Telnet</a>
...
</body>
</html>
```

2. 范例解释

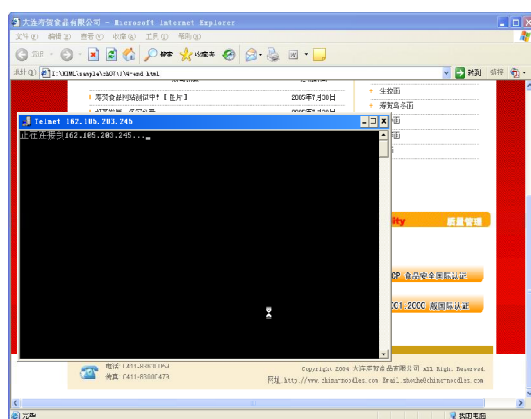
定义了“链接到Telnet”的链接地址是telnet://162.105.203.245:23,它表示Telnet服务器是一台IP地址为162.105.203.245的服务器 而端口23 是指服务器侦听请求时所使用的端口。



3. 代码和设计



4. 显示结果



访问Gopher服务器

Gopher 是一种类似 Web 的文档检索系统。在 Web 出现之前，Gopher 在因特网上还是比较流行的。但随着 Web 的流行，Gopher 就逐渐被废弃了。但是现在仍然有一些 gopher 服务器在提供服务，因此可以使用 Gopher URL 访问 Gopher 文档。Gopher URL 的格式如下：

```
gopher://server:port/path
```

Gopher 服务器必需要知道 Gopher 服务器的因特网域名或 IP 地址；而端口则是指服务器侦听请求时所使用的端口。

原始文件	Sample\Ch07\1\5.html
最终文件	Sample\Ch07\1\5-end.html
学习要点	链接到gopher

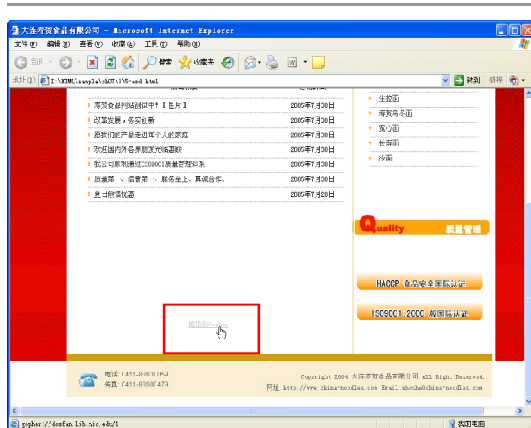
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<a href="gopher://dosfan.lib.nic.edu/1">链接到Gopher</a>
...
</body>
</html>
```

2. 范例解释

定义了“链接到Gopher”的链接地址是 gopher://dosfan.lib.nic.edu/1, 它表示Gopher 服务器是一台因特网域名为dosfan.lib.nic.edu 的服务器。

4. 显示结果



页面中的超级链接

```
news: newsgroup
news: message id
```

新闻 URL 的一个缺点是用户无法指定新闻服务器。相反，用户只能在浏览器属性设置中指定新闻服务器。以前因特网新闻组曾一度采用全球发布的形式：所有新闻服务器都带有完全相同的新闻组和它们各自的文章，因此一台新闻服务器就可以替代所有的新闻服务器了。现在，由于每天存储新闻组活动时需要耗费大量无保存意义的磁盘空间，所以单个新闻服务器经常会禁止存储所有内容，还会使用本地新闻组的审查制度。

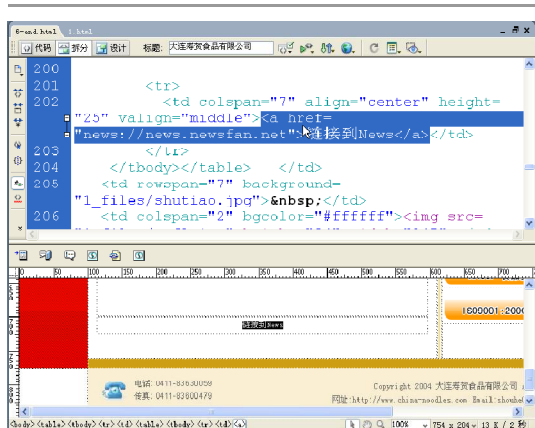
原始文件	Sample\Ch07\1\6.html
最终文件	Sample\Ch07\1\6-end.html
学习要点	链接到新闻组

2. 范例解释

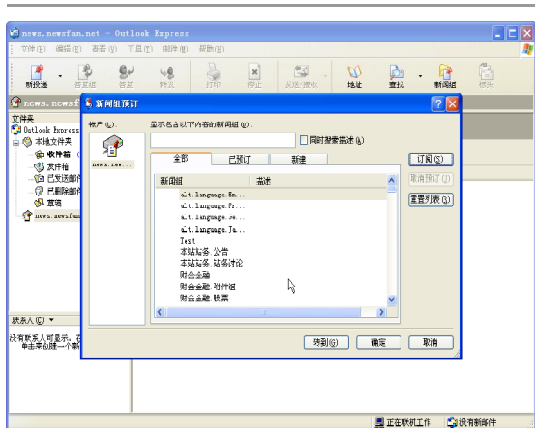
定义了“链接到 News”的链接地址是 news://news.newsfan.net。



3. 代码和设计



4. 显示结果



发送邮件

在HTML页面中，可以建立Email链接。当浏览者单击链接后，系统会启动默认的电子邮件软件进行Email的发送。在Windows系统中，如果用户设置了邮件软件，如Outlook、Outlook Express等，在浏览器中单击Email链接会自动地打开新邮件窗口，地址栏会自动添加Email链接中的邮箱地址，具体格式如下：

```
mailto://a@b.c
mailto://a@b.c?subject=content
mailto://a@b.c?cc=a@b.c
mailto://a@b.c?bcc=a@b.c
```

其中a@b.c为邮件地址，后面的参数如下表所示。

参数	描述
Subject	电子邮件主题
C C	抄送收件人
B C C	暗送收件人

原始文件	Sample\Ch07\1\7.html
最终文件	Sample\Ch07\1\7-end.html
学习要点	发送邮件

1. 范例代码

```

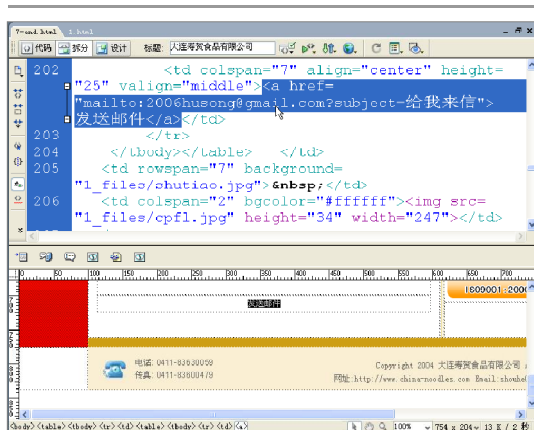
<html>
<head>
...
</head>
<body>
...
<a href="mailto:2006husong@gmail.com?subject=给我来信">发送邮件</a>
...
</body>
</html>

```

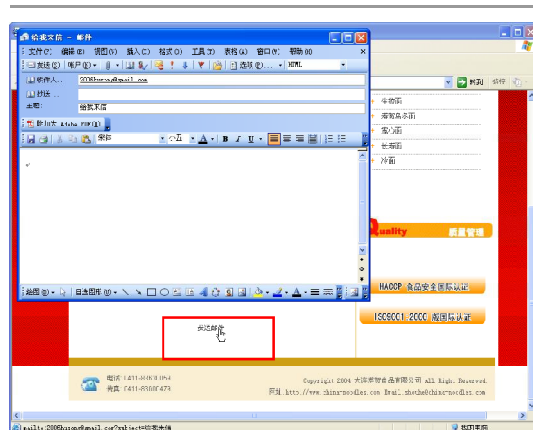
2. 范例解释

定义了“发送邮件”的链接地址是mailto:2006husong@gmail.com, 邮件主题为“给我来信”。

3. 代码和设计



4. 显示结果



内部链接

所谓内部链接指的是在同一个网站内部, 不同的HTML 页面之间的链接关系。在建立网站内部链接的时候, 要考虑到如何使链接具有清晰的导航结构, 使用户方便地找到所需内容的HTML 文件。

原始文件	Sample\Ch07\1\8.html、8-2.html
最终文件	Sample\Ch07\1\8-end.html
学习要点	链接到其他页面

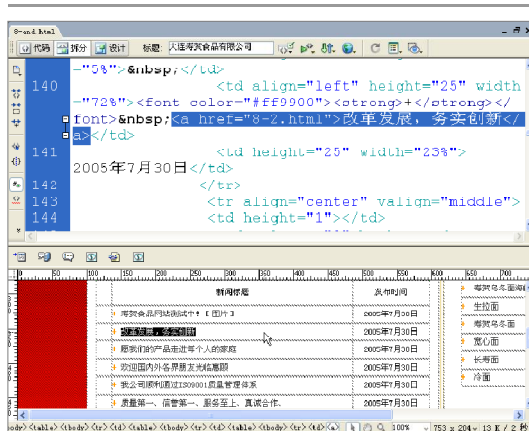


1. 范例代码与解释

```
<html>
<head>
...
</head>
<body>
...
<a href="8-2.html">改革发展，务实
创新</a>
...
</body>
</html>
```

定义了“改革发展，务实创新”的链接地址是8-2.html。

2. 代码与设计



3. 显示结果



书签链接

在浏览页面的时候，如果页面的内容较多，页面过长，就需要不断上下滑动滚动条，很不方便；如果要寻找特定的内容，就更不方便。这时就可以在该网页或另外一个页面上建立目录，浏览者只需单击目录上的项目就能自动跳到相应的位置进行阅读。建立书签首先需要命名书签，具体如下：

```
<a name="name">文字</a>
```

链接到这个书签时，使用如下代码：

```
<a href="#name">文字链接</a>
```

原始文件	Sample\Ch07\1\9.html
最终文件	Sample\Ch07\1\9-end.html
学习要点	链接到书签

1. 范例代码

```

<html>
<head>
...
</head>
<body>
<a name="top"></a>
...
<a href="#top">返回页首</a>
...
</body>
</html>

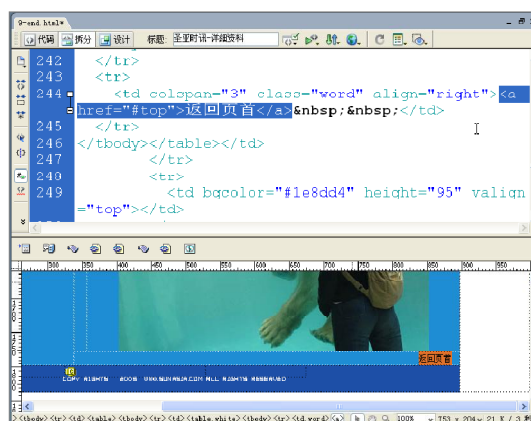
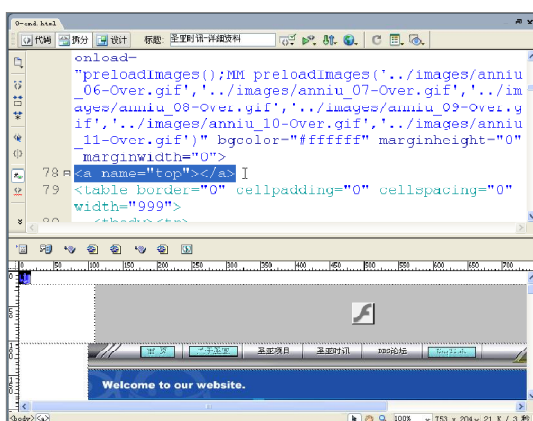
```

2. 范例解释

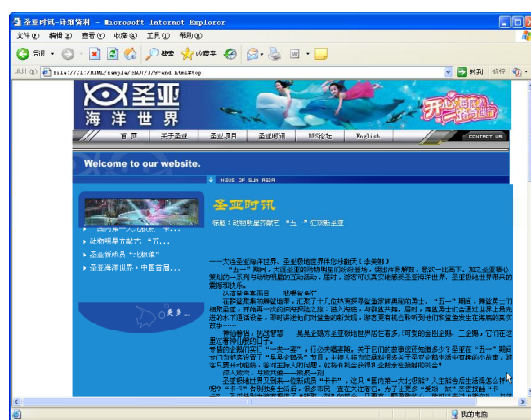
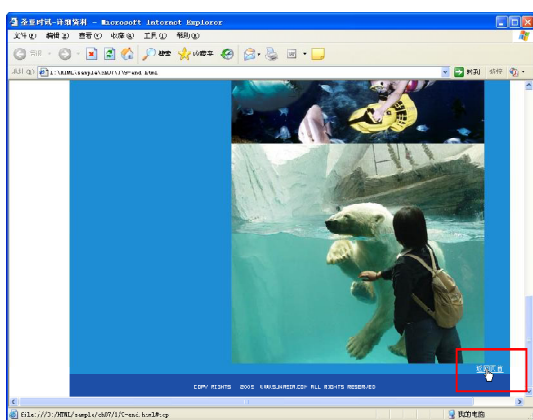
01 定义了名为 top 的书签。

02 定义了链接到 top 书签的“返回页首”文字。

3. 代码和设计



4. 显示结果





空链接

在链接中可以通过#符号实现空链接。所谓空链接是指指向链接后，光标变成手形；但单击链接后，仍然停留在当前页面，具体代码如下。

```
<a href=#>文字链接</a>
```

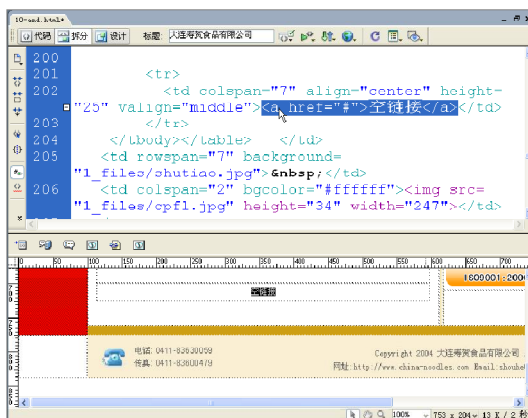
原始文件	Sample\Ch07\1\10.html
最终文件	Sample\Ch07\1\10-end.html
学习要点	制作空链接

1. 范例代码与解释

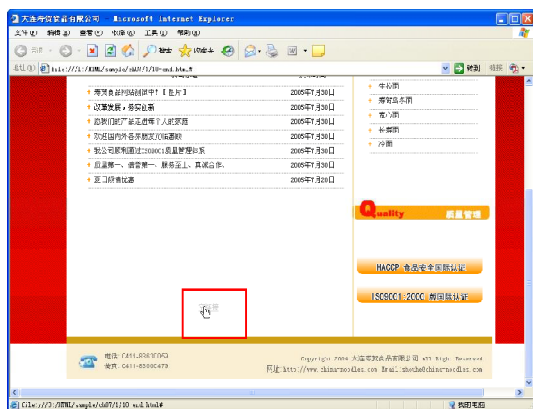
```
<html>
<head>
...
</head>
<body>
...
<a href="#">空链接</a>
...
</body>
</html>
```

定义了“空链接”的链接地址是#。

2. 代码和设计



3. 显示结果



脚本链接

用户可以在事件发生时执行一个或多个 JavaScript 命令和函数。通过使用<a>标签，可以将 JavaScript 代码和许多与鼠标或键盘相关的事件关联起来。事件处理器的值是一个或一系列以分号隔开的 JavaScript 表达式、方法和调用函数，并且要用引号引起来。当事件发生时，浏

览器就会执行这些代码，具体如下：

```
<a href="javascript:...">文字链接</a>
```

原始文件	Sample\Ch07\1\11.html
最终文件	Sample\Ch07\1\11-end.html
学习要点	制作脚本链接

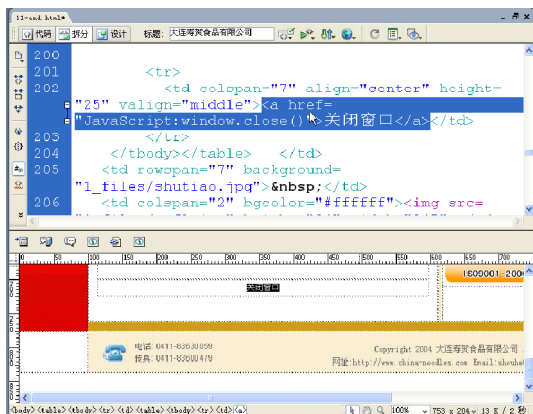
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<a href="JavaScript:window.close()">关闭窗口</a>
...
</body>
</html>
```

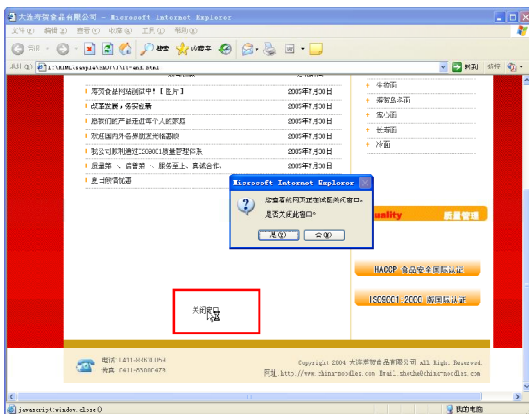
2. 范例解释

定义了“关闭窗口”的链接地址是调用 JavaScript 的关闭窗口代码，这是一个 JavaScript 的表达式，调用 Window 窗口对象，然后将其关闭。

3. 代码和设计



4. 显示结果



3

制作图像映射

一般情况下，放置在锚中的图像只是作为锚内容的一部分。浏览器可能会以某种特殊的方式来改变图像（通常是使用特殊的边框），以提醒浏览者该图像是一个超链接。但是用户单击图像的方式并没有改变，还和单击文本超链接时一样。HTML 标准提供了一种特性，允许在同一幅图像中嵌入多个不同的链接。单击图像的不同区域可以使浏览器链接到不同的目标文档。这种对光标位置敏感的图像被称为图像映射，它开创了多种创建链接的样式。

创建图像映射的方式是用标签的 Usemap 属性创建的，它要和对应的<map>和<area>标签





同时使用。

为了让客户端图像映射能够正常工作，必须在文档的某处包含一组坐标及 URL，用它们来定义映射的光标敏感区域和每个区域相应的超链接，以便用户可以单击或选择。可以将这些坐标和链接作为常规标签或特殊的<area>标签的属性值。<area>说明集合或<a>标签都要包含在<map>及其结束标签</map>之间。<map>代码段可以出现在文档主体的任何位置。

基本语法	<map>	
	</map>	
功能	定义图像映射说明	
属性及说明	属性	说明
	Name	给链接命名
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记图像映射，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
	Style	创建图像映射内容的内联样式
	Id	为图像映射创建一个标记，应用超链接时可以用这个标记来明确地引用该图像映射，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
	Title	给图像映射加上说明性的文字

<map>标签中 Name 属性的值是标签中 Usemap 属性所使用的名称，该值用于定位图像映射的命名。Name 属性的值必须是惟一的，不能被文档中的其他<map>标签使用，但是可以有不只一个图像映射引用同样的<map>说明。

基本语法	<area>	
功能	为图像映射的某个区域定义坐标和链接	
属性及说明	属性	说明
	Href	指定链接地址
	Target	指定链接的目标窗口
	Accesskey	链接热键
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记图像映射的区域，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	为图像映射的区域创建一个标记，应用超链接时可以用这个标记来明确地引用该图像映射区域，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	创建图像映射区域内容的内联样式
	Title	给图像映射区域加上说明性的文字
	TabIndex	对浏览器窗口中的对象重新排序
	Coords	图像映射中对光标敏感区域的坐标
	Shape	图像映射区域的形状
	Alt	为图像附加一个文本标记

客户端图像映射的重要内容是映射段中的<area>标签。这些<area>标签定义了每个对光标敏感的区域，以及当用户在图像映射中选取了某个区域时，浏览器应该采取的操作。用<area>标签定义的区域和其他任何超链接的功能一样：当用户将鼠标指针移到图像区域上时，指针图标会发生改变，一般是变为手形；然后浏览器可能会在浏览器窗口底部的状态框中显示相关超链接的 URL。从未在任何一个<area>标签中定义的客户端图像映射区域不会对光标敏感。<area>标签中必需的 Coords 属性定义了客户端图像映射中对光标敏感区域的坐标。坐标的数字及其含义取决于 Shape 属性中决定的区域形状，Shape 属性可以将客户端图像映射中的超链接区域定义为矩形、圆形或多边形，具体如下表所示。

Shape属性值及说明	Shape属性值	说明
	Rect	矩形区域
	Circle	椭圆形区域
	Poly	多边形区域

Shape 属性的值会影响浏览器对 Coords 属性的解释。如果没有使用 Shape 属性，那么会假设使用值 Default。依照标准，Default 意味着该区域覆盖整个图像。在实际中，浏览器默认使用矩形区域，并期望能找到 4 个 Coords 值。如果没有指定形状，而且在标签中也没有包括 4 个坐标，那么浏览器会忽略整个区域。

与<a>标签中的Href属性一样，<area>标签的Href属性也定义了单击相关联的图像映射区域中希望链接的 URL。Href 属性值可以是任何有效的相对 URL 或绝对 URL，包括 JavaScript 代码。单击超链接后，默认的浏览器窗口是原有窗口，当然也可以指定这个超链接要打开的目标窗口，如新开窗口等。这需要通过 Target 属性实现，属性值如下表所示。

Target属性值及说明	Target属性值	说明
	_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

原始文件	Sample\Ch07\2\1.html
最终文件	Sample\Ch07\2\1-end.html
学习要点	图像映射

1. 范例代码

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

```

2. 范例解释

01 定义了1_files文件夹下的anni u.jpg图片使用名为 Map 的图像映射。



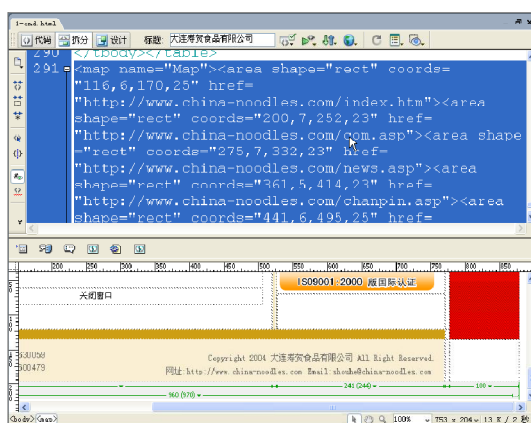
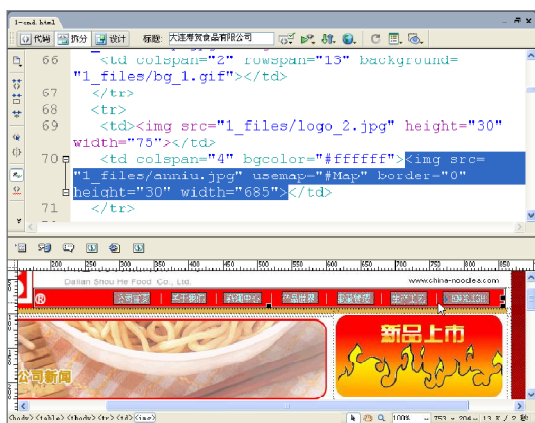
```
<map name="Map">
<area shape="rect" coords="116,
6,170,25" href="http://www.china-
noodles.com/index.htm">
<area shape="rect" coords="200,
7,252,23" href="http://www.china-
noodles.com/com.asp">
<area shape="rect" coords="275,
7,332,23" href="http://www.china-
noodles.com/news.asp">
<area shape="rect" coords="361,
5,414,23" href="http://www.china-
noodles.com/champion.asp">
<area shape="rect" coords="441,
6,495,25" href="http://www.china-
noodles.com/wsdg.asp">
<area shape="rect" coords="518,
5,572,26" href="http://www.china-
noodles.com/lxwm.asp">
<area shape="rect" coords="594,
5,662,25" href="http://www.china-
noodles.com/en/index.asp">
</map>
</body>
</html>
```

02 定义图像映射名称为Map。

03 定义了7个矩形区域，并为每一个区域设置了坐标和链接地址。

04 结束图像映射。

3. 代码和设计



4. 显示结果



4

实例制作

下面通过一个实例练习在页面中制作不同类型的超级链接。读者可通过编写该实例体会内部链接、书签链接和外部链接的用法。

原始文件	Sample\Ch07\3\1.html、1-2.html
最终文件	Sample\Ch07\3\1-end.html
学习要点	使用超级链接标签及相关属性



原始效果



最终效果

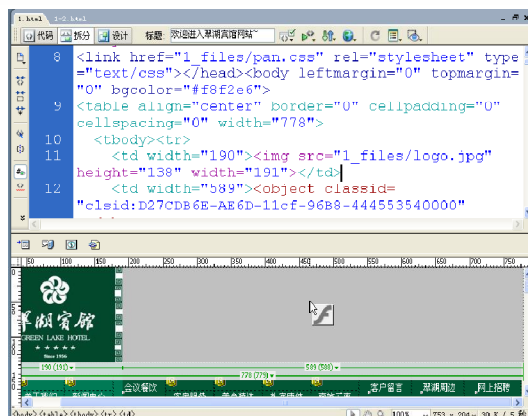


7

页面中的超级链接



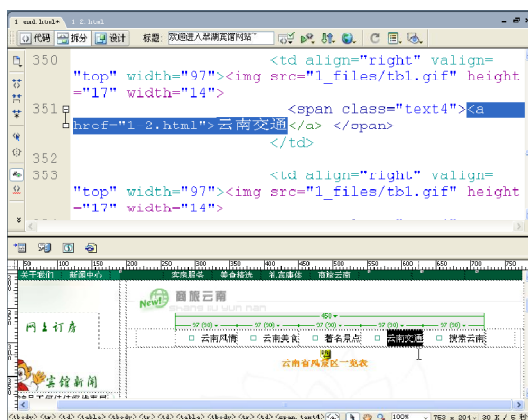
- 1 在Dreamweaver中打开提供的页面，单击“拆分”按钮，切换到“代码视图”和“设计视图”环境，如右图所示。



- 2 将光标定位到“云南交通”文字所在的位置，然后声明将其链接到“1-2.html”文件上，再结束链接，可在代码视图中将其修改为以下代码：

```
<a href="1-2.html">  
云南交通  
</a>
```

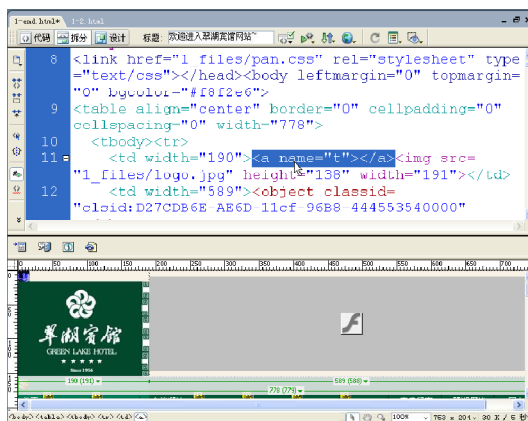
- 1 声明链接到1-2.html。
- 2 结束链接。



- 3 将光标定位到左上角“翠湖宾馆”图片前面所在的位置，并声明名称为t的书签，再结束声明，可在代码视图中插入如下代码：

```
<a name="t">  
</a>
```

- 1 声明书签，名称为t。
- 2 结束声明。



- 4 选择页面右下角的文字 Top，并声明将其链接到书签 t，可在代码视图中插入如下代码：

```
<a href="#t">
Top
</a>
```

- ❶ 声明链接到书签 t。
- ❷ 结束声明。

- 5 将光标定位到“搜索云南”文字所在的位置，声明将其链接到 <http://www.baidu.com> 站点，可在代码视图中将其修改为以下代码：

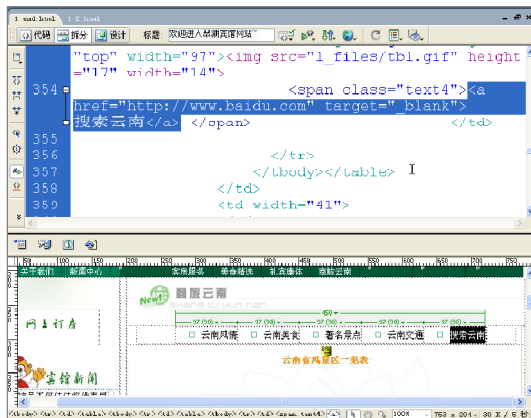
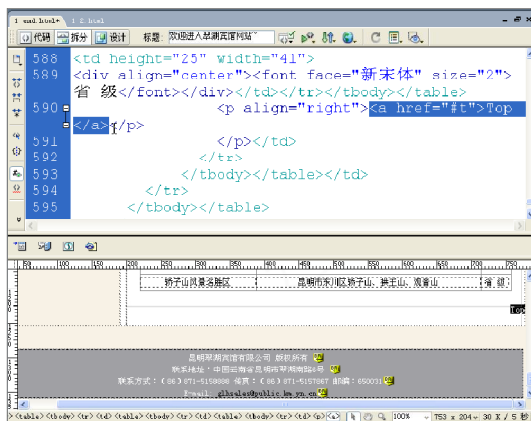
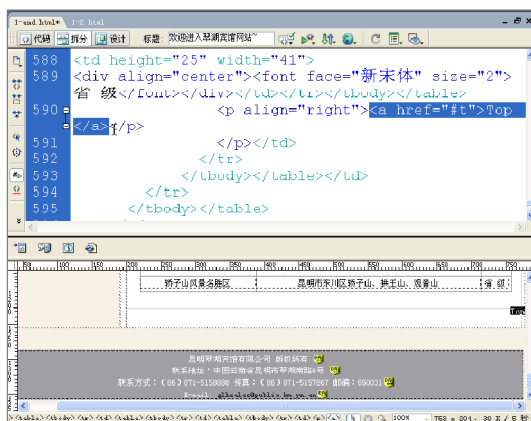
```
<a href="http://www.baidu.com">
搜索云南
</a>
```

- ❶ 声明链接到 <http://www.baidu.com>。

- 6 为了使链接能在新窗口中打开，继续将代码修改为：

```
<a href="http://www.baidu.com"
target="_blank">
搜索云南
</a>
```

- ❶ 使链接在新窗口中打开。

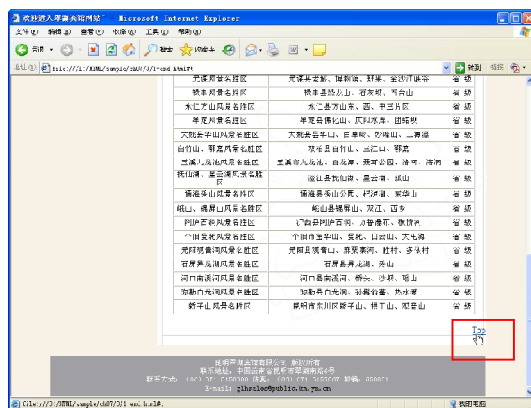




7 按下F12 快捷键，就可以在浏览器中看到页面的效果。



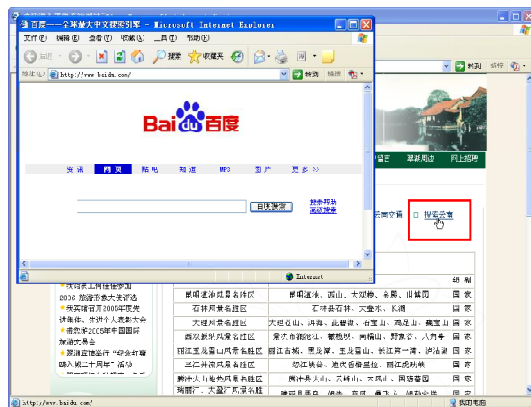
原始页面



页面下方的Top 链接



“云南交通”链接



“搜索云南”链接



Chapter

8

多媒体效果

在HTML语言中，可以在页面中嵌入声音、视频、制作滚动效果等。这些多媒体效果极大地丰富了页面的表现力，给浏览者带来了阅读上的享受和对网站的强烈印象。

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



1 页面中的背景音乐

Web 用户经常可以体验到内联多媒体——音频 (Audio)。浏览器把音频多媒体当作独立的文档,并由特殊的辅助应用程序、applet 或插件下载和显示。另一方面,IE 还自带了一个内置的声音解码器,支持特殊标签 (<bgsound>),该标签可以让用户把声音文件集成到文档中,在后台播放以作为页面的背景音乐。

网页中常用的声音格式包括以下几种。

- **WAV**: 具有较好的声音品质,许多浏览器都支持它并且不要求插件。用户可以从 CD、磁带、麦克风等录制自己的 WAV 文件。但是文件大小严格限制了可以在 Web 页面上使用的声音剪辑的长度。
- **MP3**: 最大的特点就是能以较小的比特率、较大的压缩比达到近乎完美的 CD 音质。要知道 CD 是以 1.4MB/s 的数据流量来表现其优异的音质的,而 MP3 仅需要 112 或 128Kbit/s 就可以达到逼真的 CD 音质。所以,可以用 MP3 格式对 WAV 音乐文件进行压缩,既可以保证效果,又可以减少数据存储量。
- **MIDI**: 这种格式用于器乐。许多浏览器都支持 MIDI 文件并且不要求插件。尽管其声音品质非常好,但由于访问者的声卡不同,声音效果也会有所不同。很小的 MIDI 文件也可以提供较长时间的声音剪辑。MIDI 文件不能被录制并且必须使用特殊的硬件和软件在计算机上合成。
- **AIF (音频交换文件格式或 AIFF) 格式**: 与 WAV 格式类似,也具有较好的声音品质,大多数浏览器都可以播放它并且不要求插件;也可以从 CD、磁带、麦克风等录制 AIFF 文件。但是文件大小严格限制了可以在 Web 页面上使用的声音剪辑的长度。
- **rm 格式**: 具有非常高的压缩程度,文件大小要小于 MP3。全部歌曲文件可以在合理的时间范围内下载。因为可以在普通的 Web 服务器上对这些文件进行“流式处理”,所以访问者在文件完全下载完之前即可听到声音。访问者必须下载并安装 RealPlayer 辅助应用程序或插件才可以播放这些文件。

Bgsound 标签

在页面中的背景音乐的效果可通过<bgsound>标签实现。当用户第一次下载和显示主文档时,这个标签会下载并播放音频文件。当用户刷新浏览器显示时,还会重新播放背景声音文件。

基本语法	<bgsound src="file_url">	
功能	实现页面中的背景音乐	
属性及说明	属性	说明
	Src	背景音乐的源文件
	Loop	循环方式

Src 属性是<bgsound>标签必需的。它的值指向相关声音文件的 URL。Loop 属性允许将背景声音重新播放一定的次数(或一直播放),至少到用户转移到另一个页面或者关闭浏览器时才停止播放。Loop 属性的值是音频文件重播的整数次数,或者设为“-1”无限次重放声音。

原始文件	Sample\Ch08\1\1.html
最终文件	Sample\Ch08\1\1-end.html
学习要点	使用<bgsound>标签

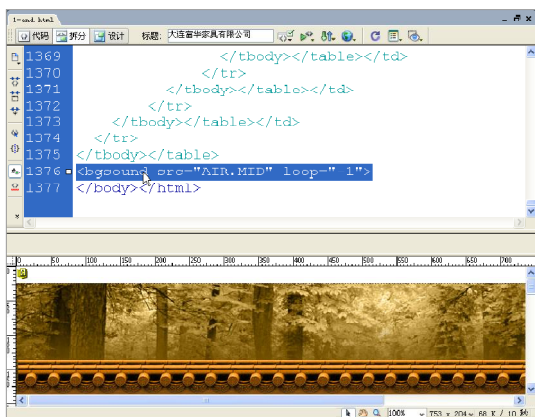
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<bgsound src="AIR.MID" loop="-1">
</body>
</html>
```

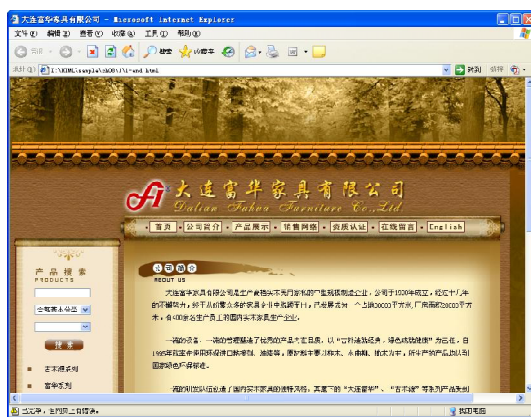
2. 范例解释

设置页面的背景音乐为 air.mid，无限次重复播放。

3. 代码和设计



4. 显示结果



2

插入图片

Web 可以传递所有类型的动画、声音和视频文件。但是，要想显示这些文件，客户端浏览器需要安装并执行某种辅助应用程序，而浏览器也支持插件辅助软件。

Embed 标签

使用<embed>可以在网页中放置如 MP3 音乐、电影、SWF 动画等多种多媒体内容。



8

多媒体效果



基本语法	<embed src="file_url">	
功能	实现页面中的内嵌多媒体	
属性及说明	属性	说明
	Src	背景音乐的源文件
	Loop	循环方式
	Width	宽度
	Height	高度
	Hidden	是否隐藏内嵌播放器
	Autostart	设置自动播放
	Dir	文本方向
	Lang	语言信息
	Align	对齐方式
	Class	用一个名称来标记内嵌多媒体，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	为内嵌多媒体创建一个标记，应用超链接时可以用这个标记来明确地引用该内嵌多媒体，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	创建内嵌多媒体内容的内联样式
	Title	给内嵌多媒体加上说明性的文字
	Type	嵌入多媒体的类型

Src 属性是<embed>标签必需的。它的值是指向相关声音文件的 URL。Loop 属性允许将背景声音重新播放一定的次数（或一直播放），直到用户转移到另一个页面或者关闭浏览器时才停止播放。Loop 属性的值是音频文件重播的整数次数，而设置为“-1”时则无限次重放声音。Hidden 用于控制播放面板的显示和隐藏，其属性值如下表所示。

Hidden属性值及说明	Hidden 属性值	说明
	True	隐藏面板
	No	显示面板

Autostart 用于控制多媒体内容是否自动播放，其属性值如下表所示。

Autostart属性值及说明	Autostart属性值	说明
	True	自动播放
	No	不自动播放

嵌入Flash动画

SWF 是 Flash 软件产生的动画，具有丰富的视频动画效果。由于网络带宽的限制，在页面中放置过大的文件是不现实的；而 Flash 动画基于矢量，确保了文件精简的尺寸。另一方面，由于 Flash 软件功能强大，其生成的 SWF 动画早已成为互联网中矢量动画的标准。现在任何一个版本的浏览器只要安装好插件，就可以观看 Flash 动画了。嵌入动画的属性除了上面列出的之外，还包括如下表所示的属性。

属性及说明	属性	说明
	Quality	动画的播放质量
	Pluginpage	Flash动画播放的插件所在位置
	Wmode	设置Flash动画播放的窗口模式

原始文件	Sample\Ch08\2\1.html
最终文件	Sample\Ch08\2\1-end.html
学习要点	使用<embed>标签嵌入Flash动画

1. 范例代码

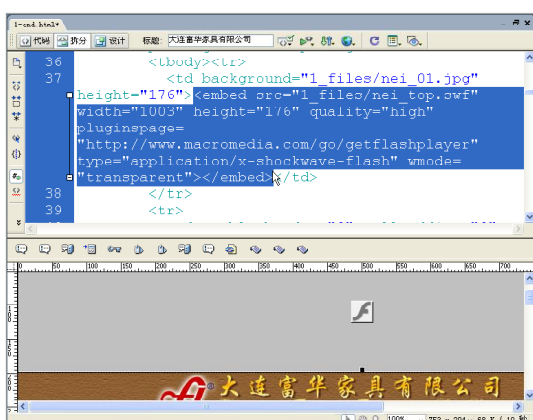
```
<h<h<html>
<head>
...
</head>
<body>
...
<embed src="1_files/nei_top.swf"
width="1003" height="176"
quality="high"
pluginpage="http://www.macromedia.
com/go/getflashplayer
" type="application/x-shockwave-
flash" wmode="transparent">
</embed>
...
</body>
</html>
```

2. 范例解释

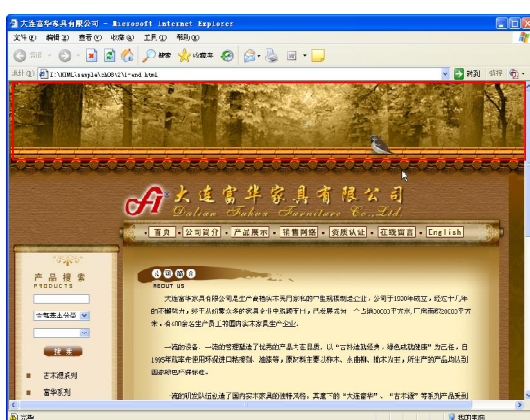
01 设置嵌入的文件为1_files文件夹下的nei_top.swf动画，其宽度为1003像素，高度为176像素，高质量，透明显示，下载Flash播放插件的地址是<http://www.macromedia.com/go/getflashplayer>。

02 结束嵌入标记。

3. 代码和设计



4. 显示结果



8

多媒体效果



嵌入Mp3 音乐

MP3 是 MPEG Layer3 的简称,也就是一种数字音频格式。在一张容量为 640MB 的普通 CD 中能够存储大约 70 分钟的声音文件,而 MP3 由于采用了高比率的数字压缩技术(压缩比率可以达到 12:1),经过 MP3 编码软件进行编码后,在音质上几乎与 CD 音乐没有什么明显的区别。这就使得它在因特网上广泛发布和传输歌曲成为可能。

原始文件	Sample\Ch08\2\2.html
最终文件	Sample\Ch08\2\2-end.html
学习要点	使用<embed>标签嵌入Mp3 音乐

1. 范例代码

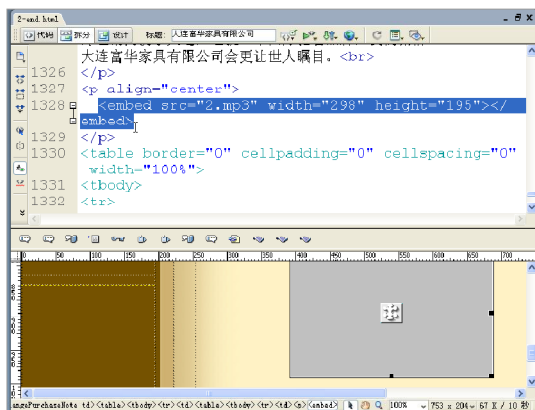
```
<html>
<head>
...
</head>
<body>
...
<embed src="2.mp3" width="298"
height="195">
</embed>
...
</body>
</html>
```

2. 范例解释

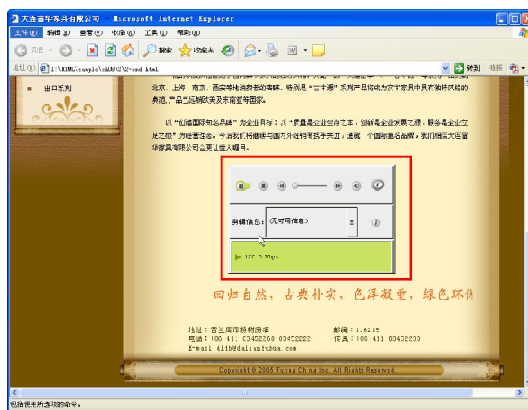
01 设置嵌入的文件为 2.mp3,宽度为 298 像素,高度为 195 像素。只要用户的计算机安装有播放 MP3 的软件,就可以欣赏这个嵌有 MP3 的网页了。

02 结束嵌入标记。

3. 代码和设计



4. 显示结果



嵌入Mpg 视频

MPEG 的全称是 Moving Pictures Experts Group (动态图像专家组),由 ISO (International Organization for standardization, 国际标准化组织)与 IEC (International Electronic Committee, 国际电 2 委员会)于 1988 年联合成立,致力于运动图像(MPEG 视频)及其伴音编码(MPEG 音频)标准化工作。它在数字电视、动态图像、互联网、实时多媒体监

控、移动多媒体通信、Internet/Intranet 上的视频服务与可视游戏、DVD 上的交互多媒体应用等方面大显身手。

原始文件	Sample\Ch08\2\3. html
最终文件	Sample\Ch08\2\3-end. html
学习要点	使用<embed>标签嵌入Mpg 视频

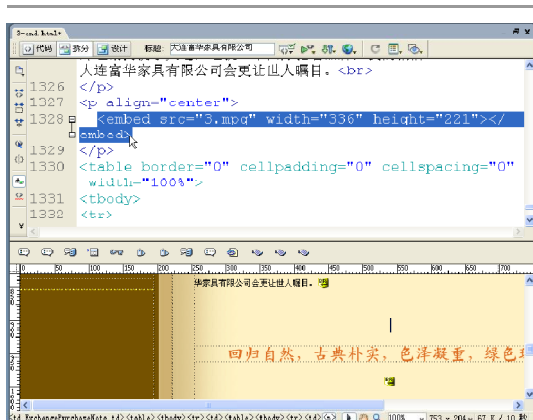
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<embed src="3.mpg" width="336"
height="221">
</embed>
...
</body>
</html>
```

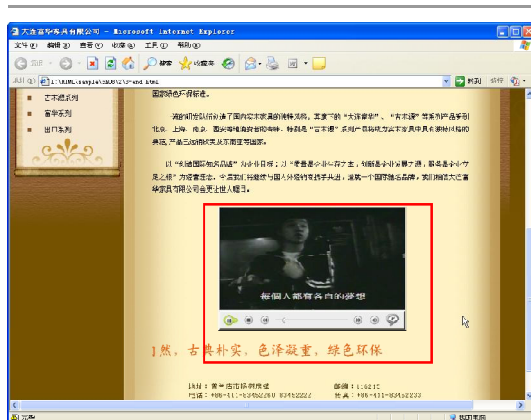
2. 范例解释

- 01 设置嵌入的文件为 3.mpg，宽度为 336 像素，高度为 221 像素，浏览器的计算机只要安装有默认的 Windows Media Player，就可以播放这段 mpg 影片。
- 02 结束嵌入标记。

3. 代码和设计



4. 显示结果



3

利用I mg标签插入AVI 视频

AVI 即 Audio Video Interleaved，是微软公司推出的视频格式文件，它应用广泛，是目前视频文件的主流。这种格式的文件随处可见，例如一些游戏、教育软件的片头等都会应用到 AVI。

插入 AVI 视频

使用<i mg>标签的 Controls、Dynsrc、Loop 或 Start 等特殊的属性允许将一个内联电影嵌入到主体内容中。



8

多媒体效果



基本语法		
功能	插入 AVI 视频	
属性及说明	属性	说明
	Dynsrc	设定 AVI 文件的播放
	Loop	设定 AVI 文件循环播放次数
	Loopdelay	设定 AVI 文件循环播放延迟
	Start	设定 AVI 文件播放方式

在标签内使用 Dynsrc 属性扩展, 可使浏览器显示一段 AVI 电影。它需要的值是电影文件的 URL。

浏览器通常在下载后会从头至尾将电影片段播放一次。电影标签的 Loop 属性使用户能够多次重复播放这个片段, 具体重复次数由属性值来设定; 如果该值为“-1”, 就会一直播放下去。

原始文件	Sample\Ch08\3\1.html
最终文件	Sample\Ch08\3\1-end.html
学习要点	使用标签插入 AVI 视频

1. 范例代码

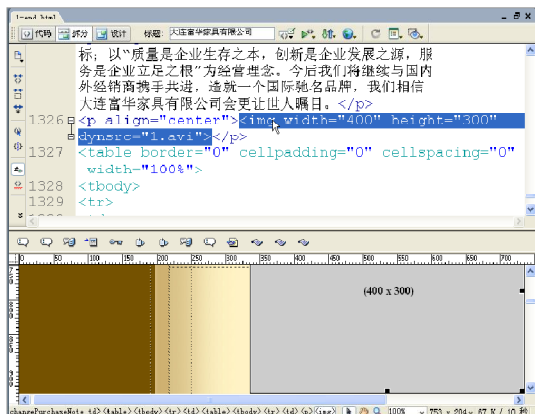
```
<html>
<head>
...
</head>
<body>
...

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

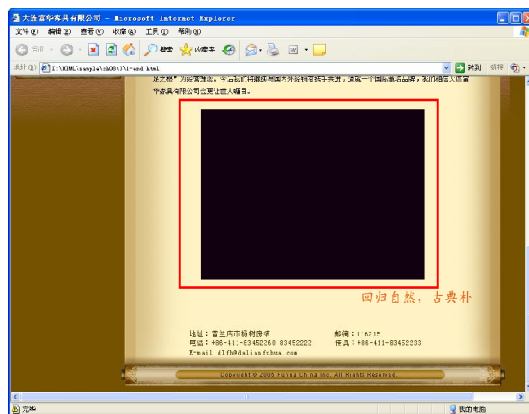
2. 范例解释

插入 1.avi 文件, 宽度为 400 像素, 高度为 300 像素, 浏览器在 HTML 显示窗口中留出一个视频播放的窗口来播放电影, 如果这段电影有声音, 那么在播放视频的同时也会播放声音。

3. 代码和设计



4. 显示结果



用电影的标签的Start 属性可以修改开始时间。通过将它的值设定为 Mouseover, 可以拖延至当用户将鼠标指针移到电影视窗中时才播放。另一个有效的Start属性值是Fileopen, 它是默认值: 下载之后马上播放。这两个值可以在 Start 属性中结合起来使用, 这样电影在下载后就会自动播放, 当用户把光标移到电影视窗中时还可以重新播放电影。当结合使用 Start 属性值时, 应在两个值之间加入逗号, 但不要加空格, 或者把它们放在引号里。Start 的属性值如下表所示。

Start属性值及说明	Start属性值	说明
	Fileopen	网页打开时即播放
	Mouseover	当鼠标滑动到 AVI 文件上方后播放

原始文件	Sample\Ch08\3\2. html
最终文件	Sample\Ch08\3\2-end. html
学习要点	使用标签的start属性

1. 范例代码

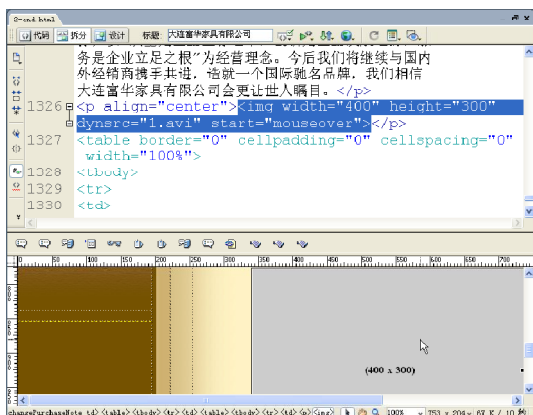
```
<html>
<head>
...
</head>
<body>
...

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

2. 范例解释

插入宽度为400像素, 高度为300像素的1.avi文件, 设定当鼠标上滚到视频上方时播放视频, 当用户把鼠标移到电影视窗中时还可以重新播放电影。

3. 代码和设计



4. 显示结果



8

多媒体效果



4

制作滚动字幕

在HTML页面中，可以实现如字幕一般的滚动文字效果。在一个排版整齐的页面中，添加适当的滚动文字可以起到让页面生动活泼的效果，在移动静止文档中的标语（banner）和其他元素时很有效。

Marquee标签

<marquee>标签定义了浏览器的显示屏幕中滚动的文字。<marquee>标签和其必需的</marquee>结束标签之间的文字将滚动显示在屏幕上。不同的标签属性控制了显示区域的大小、外观、与周围文字的对齐方式以及滚动速度等。

基本语法	<marquee>	
	滚动文字	
	</marquee>	
功能	实现页面中的滚动文字	
属性及说明	属性	说明
	Align	排列方式
	Behavior	滚动方式
	Bgcolor	滚动的背景颜色
	Class	用一个名称来标记滚动内容，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
	Direction	滚动方向
	Width	滚动宽度
	Height	滚动高度
	Hspace	滚动内容的水平间距
	Vspace	滚动内容的垂直间距
	Style	创建滚动内容的内联样式
	Loop	滚动的循环次数
	Scrollamount	滚动速度
	Scrolldelay	滚动延迟

原始文件	Sample\Ch08\4\1.html
最终文件	Sample\Ch08\4\1-end.html
学习要点	使用<marquee>标签

1. 范例代码

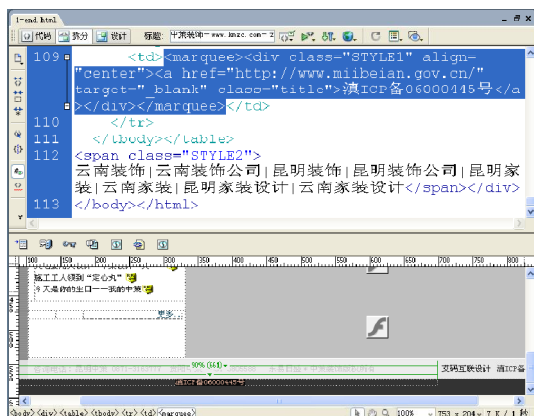
```
<html>
<head>
...
</head>
<body>
...
<marquee>
<div class="STYLE1"
align="center">
<a href="http://www.miibeian.gov.
cn/" target="_blank"
class="title">滇 ICP 备 06000445 号
</a>
</div>
</marquee>
...
</body>
</html>
```

2. 范例解释

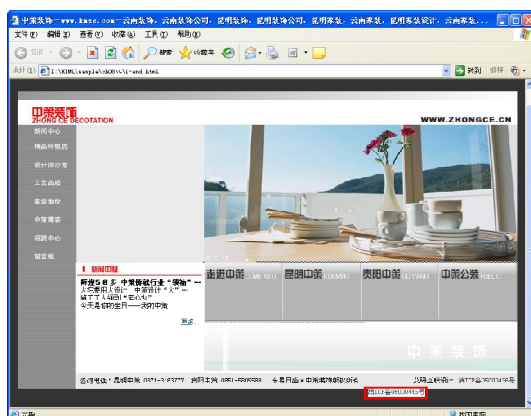
01 声明滚动文字“滇 ICP 备 06000445 号”开始，这段居中的、带有到<http://www.miibeian.gov.cn/> 链接的文字将会以默认方式从右向左滚动。

02 声明滚动文字结束。

3. 代码和设计



4. 显示结果



可以通过 Direction 设定文字滚动的方向，分别为向上、向下、向左、向右，使滚动的文字具有更多的变化，Direction 的属性值如下表所示。

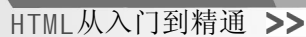
Direction 属性值及说明	Direction 属性值	说明
	Up	滚动文字向上
	Down	滚动文字向下
	Left	滚动文字向左
	Right	滚动文字向右

原始文件	Sample\Ch08\4\2.html
最终文件	Sample\Ch08\4\2-end.html
学习要点	使用<marquee>标签的direction属性



8

多媒体效果

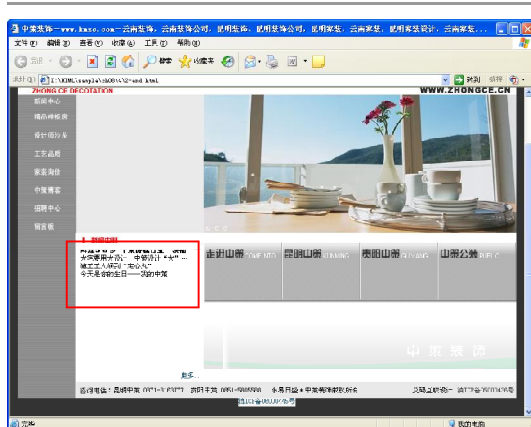


2. 范例解释

01 声明滚动文字“辉煌50步中策铸就行业‘领袖’…大宅要用大设计中策设计‘大’…施工工人领到‘定心丸’今天是你的生日--我的中策“开始，滚动方向向上。

02 声明滚动文字结束。

4. 显示结果



Behavi or属性值及说明	Behavi or 属性值	说明
	Scroll	循环往复
	Slide	只走一次滚动
	Alternate	交替进行滚动



1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<marquee direction="up"
behavior="alternate"
scrollamount="3">
<b>辉煌50步 &nbsp;&nbsp;中策铸就行业“领袖”...</b><br>
大宅要用大设计 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;中策设计
“大”...<br>
施工工人领到“定心丸”<br>
今天是你的生日——我的中策<br>
</marquee>
...
</body>
</html>

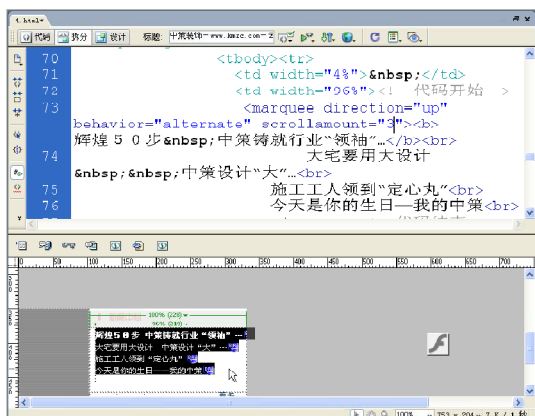
```

2. 范例解释

01 声明滚动文字“辉煌50步 中策铸就行业‘领袖’... 大宅要用大设计中策设计‘大’... 施工工人领到‘定心丸’今天是你的生日--我的中策”开始，滚动方向为向上，方式为往复滚动，滚动速度为3。

02 声明滚动文字结束。

3. 代码和设计



4. 显示结果



Scroll delay属性用来设置滚动过程中文字相邻两次移动之间的时间间隔，以毫秒为单位。这个值越小，滚动得越快。设计者可以使用较小的Scroll delay值来减少一个很小但滚动很平滑的Scroll amount的延迟时间。

原始文件	Sample\Ch08\4\5. html
最终文件	Sample\Ch08\4\5-end. html
学习要点	使用<marquee>标签的scroll delay属性

1. 范例代码

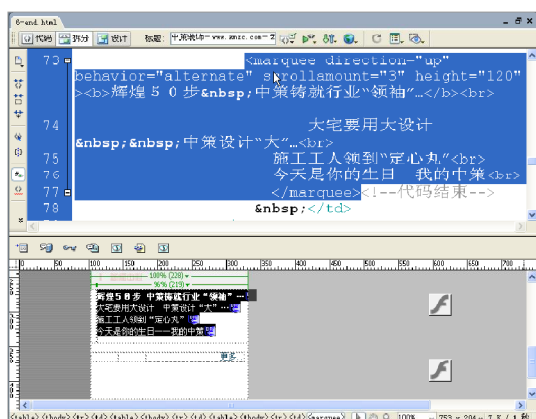
[illegible]

2. 范例解释

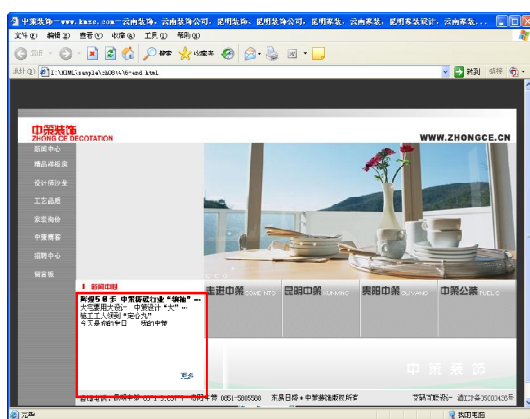
01 声明滚动文字“辉煌 50 步中策铸就行业‘领袖’… 大宅要用大设计中策设计‘大’… 施工工人领到‘定心丸’今天是你的生日——我的中策”开始，滚动方向向上，方式为往复滚动，滚动速度为 3，滚动高度为 120 像素。

02 声明滚动文字结束。

3. 代码和设计



4. 显示结果



Bgcolor 属性允许改变字幕区域的背景颜色。它的值可以是 RGB 颜色或者标准颜色名。关于这两种颜色规范方法的完整讨论可参阅附录。

原始文件	Sample\Ch08\4\7. html
最终文件	Sample\Ch08\4\7-end. html
学习要点	使用<marquee>标签的bgcolor属性

1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<marquee direction="up"
scrollamount="3" height="120"
bgcolor="#EEEEEE">
<b>辉煌 5 0 步 &nbsp;&nbsp;中策铸就行业“领袖”...</b><br>
大宅要用大设计 &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;中策设计
“大”...<br>
施工工人领到“定心丸”<br>
今天是你的生日——我的中策<br>
</marquee>
...
</body>
</html>

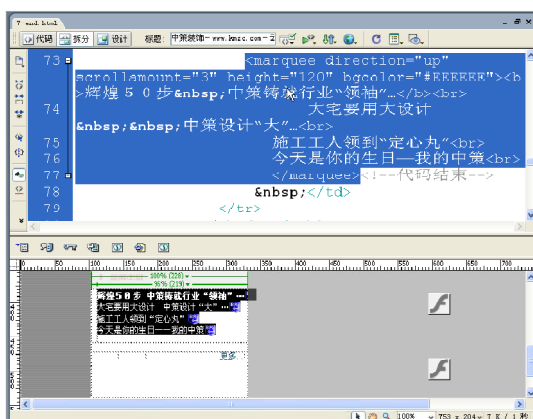
```

2. 范例解释

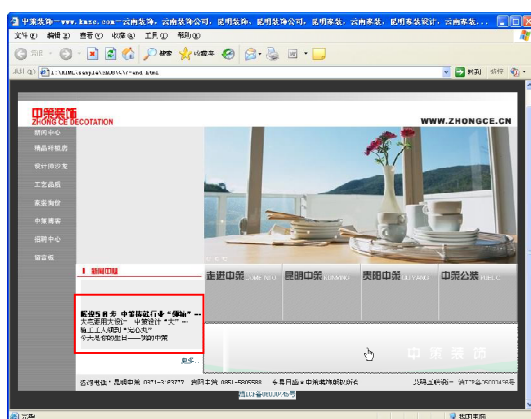
01 声明滚动文字“辉煌 50 步中策铸就行业‘领袖’... 大宅要用大设计中策设计‘大’... 施工工人领到‘定心丸’今天是你的生日——我的中策”开始，滚动方向向上，滚动速度为 3，滚动高度为 120 像素，背景颜色为 #EEEEEE。

02 声明滚动文字结束。

3. 代码和设计



4. 显示结果



Hspace 和 Vspace 属性可用来在字幕和周围文字之间留出间距。这样字幕可以相对于周围文字而体现出来。这两个属性都要求使用整数值，并且用像素值来指定间距。Hspace 属性控制着字幕左右两侧的间距；Vspace 属性控制着字幕上下两侧的间距。

原始文件	Sample\Ch08\4\8.html
最终文件	Sample\Ch08\4\8-end.html
学习要点	使用<marquee>标签的hspace和vspace属性



8

多媒体效果

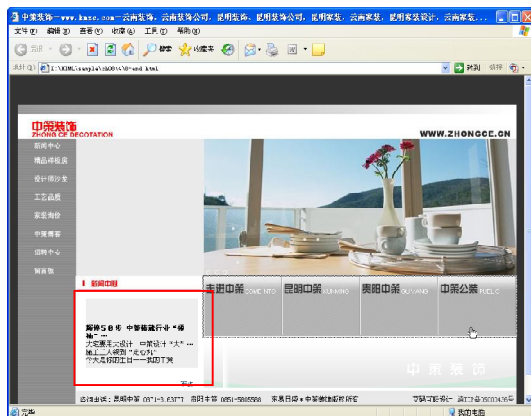


2. 范例解释

01 声明滚动文字“辉煌50步中策铸就行业‘领袖’…大宅要用大设计中策设计‘大’…施工工人领到‘定心丸’今天是你的生日--我的中策”开始，方向为向上，滚动速度为3，滚动高度为120像素，背景颜色为#EEEEEE，水平间距为10像素，垂直间距为10像素。

02 声明滚动文字结束。

4. 显示结果



5 其他嵌入式内容

其他常用的嵌入标签包括<object>、<applet>、<param>等。

使用Object标签

<object>标签最初是Microsoft用来支持ActiveX Applet,不久以后,Microsoft又添加了对Java的支持。同样,Netscape开始是为包含对象而支持<Embed>和<Applet>标签,后来对<object>也提供了有限的支持。

基本语法	<object>	
	...	
	</object>	
功能	在文档中嵌入对象	
属性及说明	属性	说明
	Classid	指定包含对象的位置
	Archive	URL 列表，每个URL都指向一个在显示或执行对象之前浏览器需要加载的档案文件
	Border	对象边框
	Codebase	提供一个可选的URL，浏览器从这个URL中获取对
	Data	指定需要对象处理的数据文件
	Width	对象宽度
	Height	对象高度
	Hspace	对象水平间距
	Vspace	对象垂直间距
	Name	对象名称
	Type	指定调用的源内容类型
	TabIndex	对浏览器窗口中的对象重新排序
	Dir	文本方向
	Lang	语言信息
	Align	对齐方式
	Class	用一个名称来标记对象，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	为对象创建一个标记，应用超链接时可以用这个标记来明确地引用该对象，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	创建对象内容的内联样式
	Title	给对象加上说明性的文字



<object>标签的内容可以是任何有效的 HTML 或者 XHTML 内容，以及将参数传递给 applet 的 <param>标签。如果浏览器可以得到所要求的对象，并且通过执行 applet 或者用插件（辅助）应用程序成功处理该对象，那么<object>标签中除<param>标签以外的内容都会被忽略。如果在获取和处理对象的过程中出了什么问题，浏览器不会把对象插入到文档中；相反它将会显示<object>标签中除<param>标签以外的内容。简而言之，要考虑浏览器无法处理<object>标签或者对象无法成功加载的情况，并提供替代的解决方式。

使用 Applet 标签

在文档中使用<applet>标签可下载和执行 applet，另外，还可用这个标签在文档中定义一个显示区域。设计者可以在<applet>标签中提供可选的内容，以显示在不支持 applet 的浏览器中。大多数 applet 都需要在文档中提供一个或者多个参数来控制它们的执行，并将这些参数用<param>标签放置在<applet>标签和对应的</applet>结束标签之间。浏览器将这些文档规定的参数在执行时传递给 applet。



基本语法	<applet>	
	...	
	</applet>	
功能	插入applet小程序	
属性及说明	属性	说明
	Alt	替换文字
	Code	指定需要浏览器执行的j ava类的文件名称
	Codebase	提供一个可选的URL，浏览器从这个URL中获取小程序
	Width	小程序宽度
	Height	小程序高度
	Hspace	小程序水平间距
	Vspace	小程序垂直间距
	Name	小程序名称
	Align	对齐方式
	Class	用一个名称来标记小程序，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	为小程序创建一个标记，应用超链接时可以用这个标记来明确地引用该小程序，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	创建小程序内容的内联样式
	Title	给小程序加上说明性的文字

使用Param标签

<param>标签为一个包含它的<object>或者<applet>标签提供参数。<param>标签没有内容，并且在HTML中没有结束标签。它只出现在<object>或者<applet>标签及其结束标签之间，有时和其他<param>标签一起使用。使用<param>标签可以将参数传递给嵌入的对象，这些参数是对象正常运行所需要的。

基本语法	<param>	
功能	给嵌入对象提供参数	
属性及说明	属性	说明
	Type	指定调用的源的内容类型
	Name	参数名称
	Id	为参数创建一个标记，超链接可以用这个标记来明确地引用该嵌入对象，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
	Value	参数的值

<param>标签有两个必需的属性：Name 或者 Id，以及 Value。在前面的表单中已经应用了。它们合在一起定义了浏览器传递给 applet 的名称 / 值（Name/Value）对。

原始文件	Sample\Ch08\5\1.html
最终文件	Sample\Ch08\5\1-end.html
学习要点	使用<object>标签和<param>标签插入Flash动画

1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<object classid="clsid:D27CDB6E-
AE6D-11cf-96B8-444553540000"
codebase="http://download.
macromedia.com/pub/shockwave/
cabs/flash/swflash.cab#version=
6,0,29,0"
height="240" width="558">
<param name="movie"
value="1_files/main.swf">
<param name="quality"
value="high">
</object>
...
</body>
</html>

```

2. 范例解释

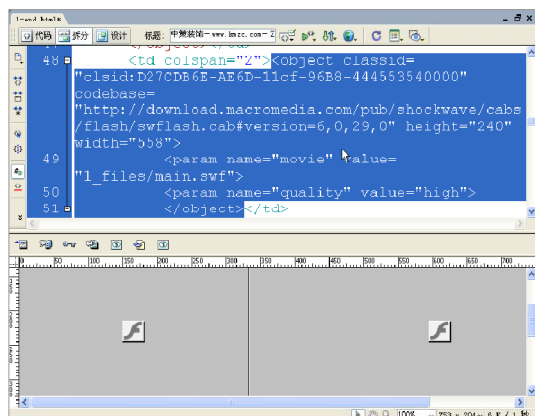
- 01 声明包含对象的位置，插件的位置为 `http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,29,0`，并设置其宽度为 240 像素，高度为 558 像素。
- 02 声明参数名称为 `movie`，值为 `1_files` 文件夹下的 `main.swf`。
- 03 声明参数名称为 `quality`，值为 `high`。
- 04 声明对象的结束。



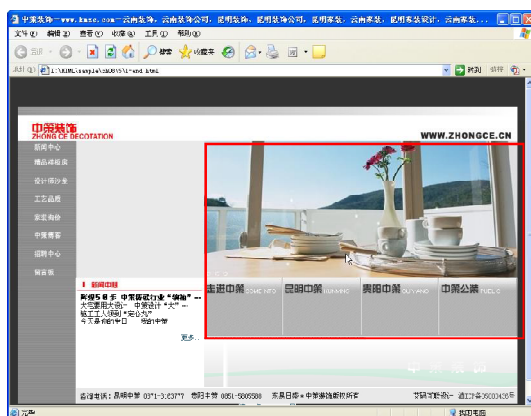
8

多媒体效果

3. 代码和设计



4. 显示结果



原始文件	Sample\Ch08\5\2.html
最终文件	Sample\Ch08\5\2-end.html
学习要点	使用<applet>标签和<param>标签插入Java小程序



1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<applet archive="ansnow.jar"
code="ansnow.class" width=404
height=288>
<param name=credits value="Applet
by Fabio Ciucci (www.anfyjava.com)">
<param name=regcode value="NO">
<param name=reglink value="NO">
<paramname=regnewframevalue="YES">
<paramname=regframenamvalue="_blank">
<paramname=statusmsgvalue="Snow">
<param name=flakes1 value="600">
<param name=flakes2 value="300">
<param name=flakes3 value="100">
<param name=flakes4 value="100">
<param name=windmax value="1">
<paramname=windvariation value="7">
<param name=speed value="4">
<paramname=backimage value="1.jpg">
<paramname=bgcolorvalue="000133">
<param name=overtex value="NO">
<param name=overimg value="NO">
<param name=overimgX value="0">
<param name=overimgY value="0">
<paramname=texttype value="vertical">
<paramname=textvspace value="10">
<paramname=textminfont value="8">
<paramname=textmaxfont value="172">
<paramname=textoffset value="120">
<param name=textspeed value="1">
<paramname=textfont value="Times-
Roman">
<param name=textbold value="YES">
<paramname=textitalic value="YES">
<param name=textsize value="18">
<paramname=textshadow value="YES">
<param name=TextColR value="255">
<param name=TextColG value="255">
<param name=TextColB value="255">
<param name=TextSColR value="0">
```

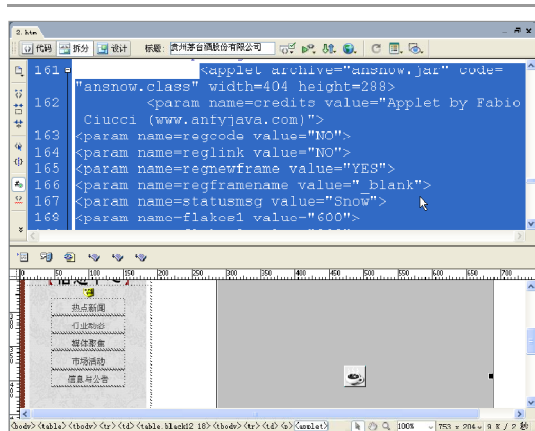
2. 范例解释

01 声明包含对象的位置, java 压缩包为 ansnow.jar, 引用的 java 类文件为 ansnow.class, 宽度为 404 像素, 高度为 288 像素。

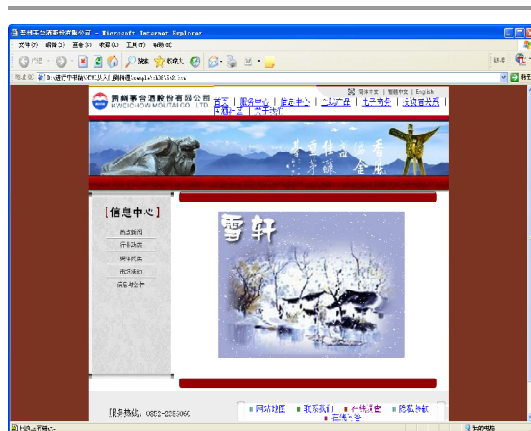
02 此行及后面部分各种参数的声明。

```
<param name=TextSColG value="0">
<param name=TextSColB value="0">
<paramname=TextJumpAmp value="0">
<paramname=TextJumpSpd value="0">
<paramname=TextSi neAmp value="0">
<paramname=TextSi neSpd value="0">
<paramname=TextSi neAngle value="3">
<paramname=memdelay value="1000">
<param name=priority value="3">
<param name=MinSYNC value="10">
</applet>
...
</body>
</html>
```

3. 代码和设计



4. 显示结果



6

实例制作

下面通过这个实例练习页面中不同类型的多媒体文件效果的制作。读者可通过编写该实例来体会在页面中插入Flash动画和滚动文字的用法。其中插入Flash动画稍麻烦一些，设置显示质量图片大小、插件位置等。

原始文件	Sample\Ch08\6\1. html
最终文件	Sample\Ch08\6\1-end. html
学习要点	使用多媒体标签



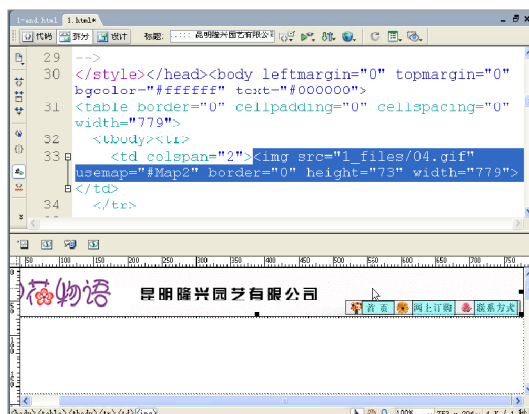
原始效果



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。

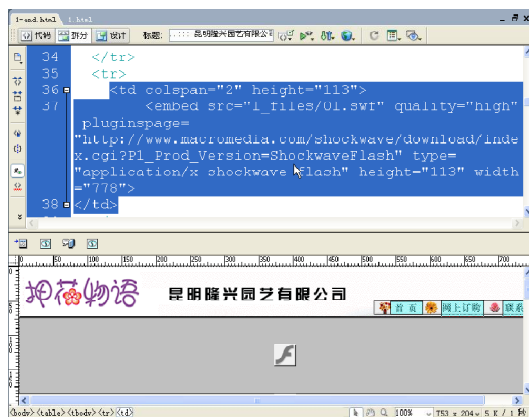


- 2 将光标定位到页面上方空白的单元格中如下代码的位置：

```
<td colspan="2" height="113">
    &nbsp;
</td>
```

在代码视图中将其修改为以下代码：

```
<td colspan="2" height="113">
    <embed src="1_files/01.swf"
    quality="high" pluginspage="http:
    //www.macromedia.com/shockwave/down
    load/index.cgi?
    P1_Prod-Version=ShockwaveFlash"
    type="application/x-shockwave-
    flash" height="113" width="778">
</td>
```



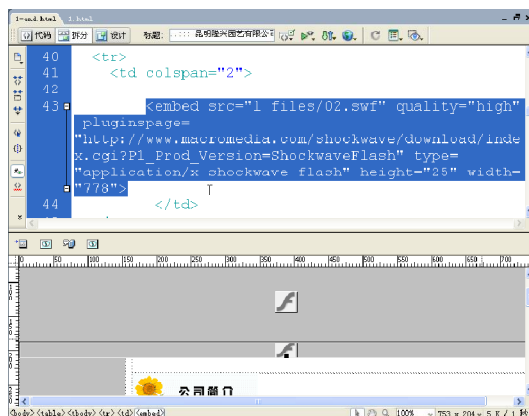
插入1_files文件夹下的01.swf图片，高质量显示，并且宽度为778像素、高度为113像素，然后定义插件的位置。

- 3 将光标定位到插入的Flash下方的空白单元格中如下代码的位置：

```
<td colspan="2">
    &nbsp;
</td>
```

在代码视图中将其修改为以下代码：

```
<td colspan="2">
    <embed src="1_files/02.swf"
    quality="high" pluginspage="http:
    //www.macromedia.com/shockwave/
    download/index.cgi?
    P1_Prod-Version=ShockwaveFlash"
    type="application/x-shockwave-
    flash" height="25" width="778">
</td>
```



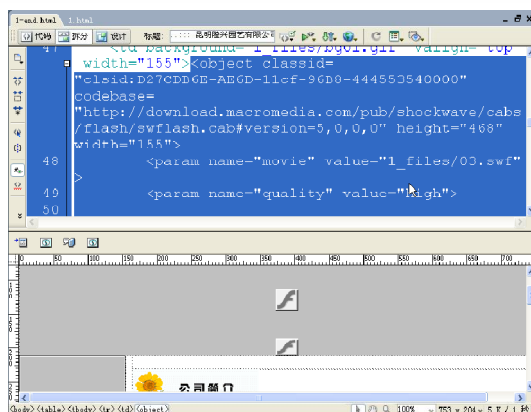
插入1_files文件夹下的02.swf图片，高质量显示，并且宽度为778像素，高度为25像素，然后定义插件的位置。

4 将光标定位到插入的Flash下方左侧的空白单元格中如下代码的位置:

```
<td background="1_files/bg01.gif"
align="top" width="155">&nbsp;
</td>
```

在代码视图中将其修改为以下代码:

```
<td background="1_files/bg01.gif" valign="top" width="155">
<object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=5,0,0,0" height="468" width="155">
<param name="movie" value="1_files/03.swf">
<param name="quality" value="high">
</object>
</td>
```



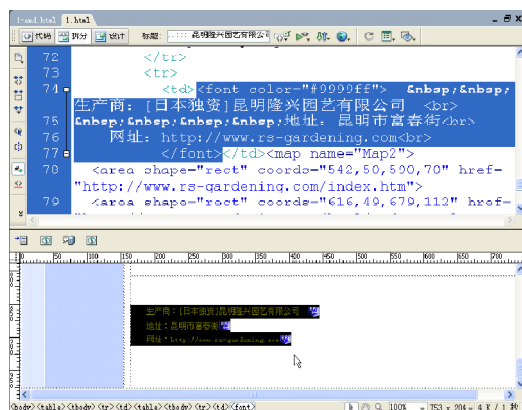
01 声明包含对象的位置，插件的具体位置，并设置宽度为468像素，高度为155像素。

02 声明参数名称为movie, 值为1_files文件夹下的03.swf。

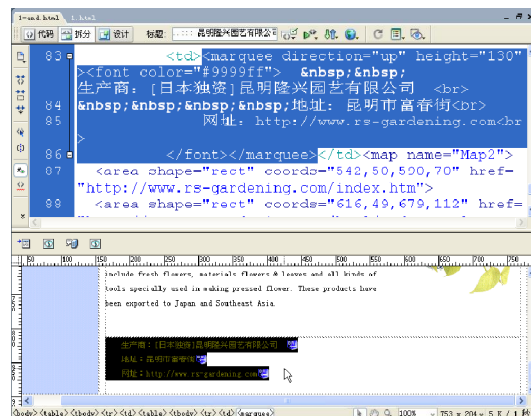
03 声明参数名称为`qual i ty`, 值为`hi gh`。

04 声明对象的结束。

5 将光标定位到页面下方如下代码的位置:

[illegible]

在代码视图中将其修改为以下代码:

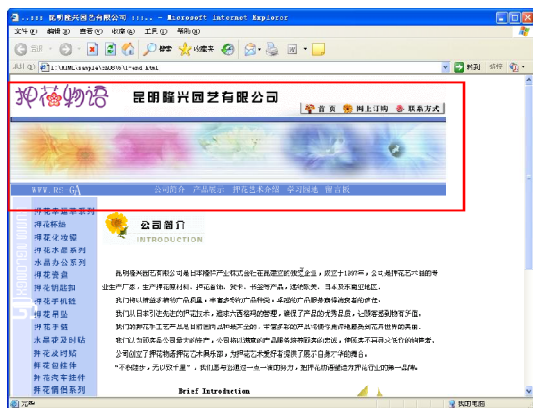
[illegible]

❶ 插入滚动标签，方向向上，高度为130。

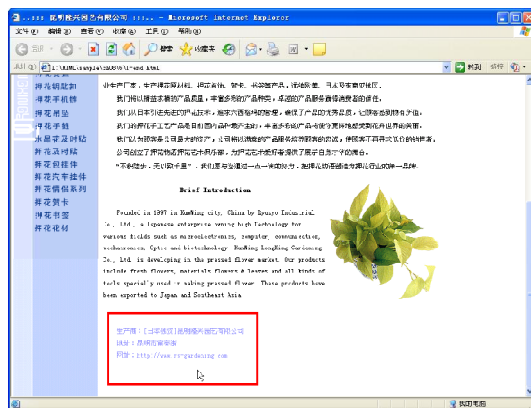
● 结束滚动。



6 按下F12 快捷键，可以在浏览器中看到页面的效果。



页面中的Flash 动画



页面下方的滚动文字



Chapter

9

表单元素

表单是实现动态网页的一种主要的外在形式，使用HTML可以轻松制作它。通过表单，可以与用户实现信息交互。一个表单至少应包括说明性文字、用户填写的表格、提交和重填按钮等内容。

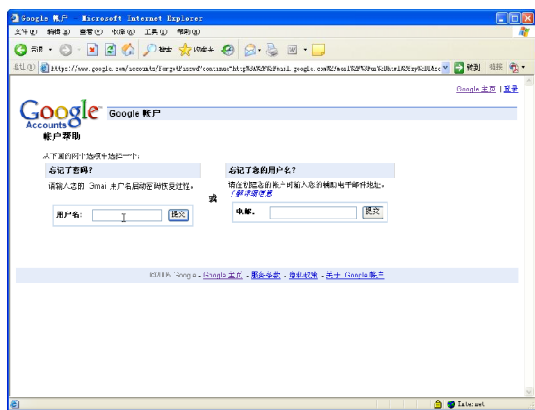
```
<meta http-equiv="Content-Type" content="text/html" charset="gb2312">
<link href="1_files/css.css" rel="stylesheet" type="text/css">
```



1

什么是表单

表单(form)是由一个或多个文本输入框、可单击的按钮、多选框、下拉菜单和图像按钮等组成的,所有这些都在<form>标签中。一个文档中可以包含多个表单,而且在每个表单中可以放置通常的主体内容(包括文字和图像在内)。文字可以为用户提供表单元素标记以及各种提示和指示,告诉他们该如何填写表单。另外,在不同的表单元素中,还可以用JavaScript事件处理器来实现多种效果,例如检验和证实表单内容及计算流量等。用户首先要填写表单中的不同字段,然后单击特殊的“提交(Submit)”按钮(或者按Enter键)将表单提交给服务器。浏览器会将用户提交的数值和选项数据打包,并将它们发送给某台服务器或某个电子邮件地址。服务器通常会将信息传递给一个能够处理这些信息的支持程序或应用程序,并产生一个HTML形式的回应。这个回应可能提示用户如何正确填写表单中的内容,或是提示还有哪些字段没有填写。然后服务器将回应送交浏览器客户端,而浏览器又会将结果呈现给用户。如果是电子邮件式的表单,信息将会直接发送到某人的电子信箱中。



Form 标签

设计者可以将表单放在文档主体中的任何位置,但前提是将表单的元素都放在<form>标签和</form>结束标签之间。

基本语法	<form>	
	包含表单元素	
功能	</form>	
	定义表单	
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	用一个名称来标记表单,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

(续表)

属性及说明	Id	为表单创建一个标记,将来超链接可以用这个标记明确地引用该表单,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	创建表单内容的内联样式
	Title	给表单加上说明性的文字
	Name	表单的名称
	Method	定义表单结果从浏览器传送到服务器的方法,一般有两种方法: get、post。
	Action	用来定义表单处理程序(一个ASP、CGI等程序)的位置(相对地址或绝对地址)。
	Enctype	设置表单资料的编码方式
	Target	设置返回信息的显示方式

<form>标签中必需的 Action (动作) 属性说明了接收和处理表单数据的应用程序的 URL。大多数 Web 管理员都把他们的表单处理应用程序放在 Web 服务器中某个特定的目录下,这个目录的名称通常为 cgi-bin,代表的是公共网关接口(Common Gateway Interface, CGI)的二进制文件。在一个目录中保存这些特殊的表单处理程序将更加便于管理,同时也可以确保服务器的安全。

浏览器在把数据传递给服务器之前,会专门对表单的数据进行编码,这样在传输过程中数据才不会被打乱或破坏。然后由服务器来决定是将参数解码,还是将编码形式的参数传递给应用程序。标准的编码格式是因特网媒体格式。使用<form>标签中可选的 Enctype 属性可以改变编码格式,它目前支持的编码格式只有“Mul ti part/Form-data (多部分表单数据)”和“text/plain (纯文本)”两种。对用户上传时需要选择字段的表单来说,一般要采用多部分表单数据格式(Mul ti part/Form-data)。纯文本格式(Text/Plain)应和 Action 属性中的 mail to URL 一起使用,从而将表单发送给某个电子邮件地址而不是服务器。Enctype 的属性列表如下。

Enctype属性值及说明	Enctype属性值	说明
	Text/Plain	以纯文本形式传送信息
	Appl i cation/x-www-Form-url encoded	默认的编码形式
	Mul ti part/Form-data	使用MINE编码

除了 Action 之外,<form>标签中另一个必需的属性是设置方法: post 方法和 get 方法,而浏览器则用这些方法将表单中的数据传送给服务器进行处理。如果采用 post 方法,浏览器将会按下面两步来发送数据。首先,浏览器将与 Action 属性中指定的表单处理服务器建立联系,一旦建立起连接之后,浏览器就会按分段传输的方法将数据发送给服务器。在服务器端,一旦 post 样式的应用程序开始执行,就会从一个标准位置读取参数。而一旦读到参数,在应用程序能够使用这些表单值以前,还要对这些参数进行解码。用户特定的服务器会明确指定应用程序该如何接收这些参数。另一种情况是采用 get 方法,这时浏览器会先与表单处理服务器建立连接,然后直接在一个传输步骤中发送所有的表单数据:浏览器会将数据直接附在表单的 Action URL 之后,这两者之间用问号来进行分隔。

一般浏览器通过上述任何一种方法都可以传输表单信息,而有些服务器只接受其中一种方法





提供的数据。设计者可以在<form>标签的method（方法）属性中指明表单处理服务器要用哪种方法。

随着框架概念的提出，有可能将表单的结果重新指向另一个窗口或框架。只要简单地在<form>标签中加入target（目标）属性，并提供要接收结果的窗口或是框架的名称即可。Target的属性值如下表所示。

Target属性值及说明	Target属性值		说明
	Blank		将返回信息显示在新开的浏览器窗口中
	Parent		将返回信息显示在父级浏览器窗口中
	Self		将返回信息显示在当前浏览器窗口中
	Top		将返回信息显示在顶级浏览器窗口中

原始文件	Sample\Ch09\1\1.html
学习要点	使用<form>标签

1. 范例代码

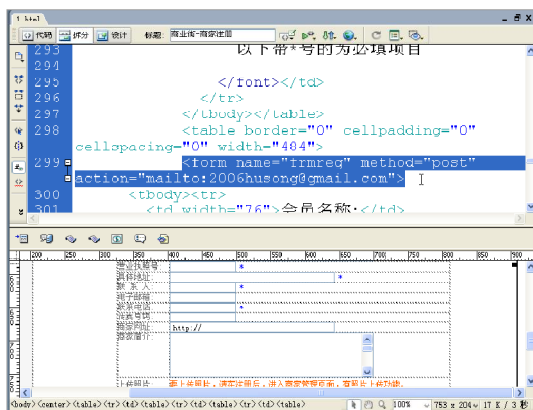
```
<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
</form>
...
</body>
</html>
```

2. 范例解释

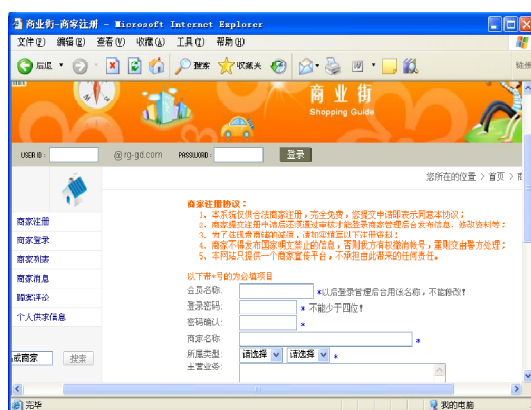
01 声明名称为frmreg的表单，使用post方法，提交到2006husong@gmail.com的邮箱中。

02 结束表单。

3. 代码和设计



4. 显示效果



2

表单元素

在<form>标签中，可以包含以下 5 个标签。

标签	描述
<input>	表单输入标记
<select>	菜单和列表标记
<option>	菜单和列表项目标记
<textarea>	文字域标记
<optgroup>	子菜单标记

各种表单元素基本可以满足网站收集信息的要求。如果要求浏览者输入文字信息，如“姓名”、“留言”等，可以使用文本字段；如果要求浏览者在固定的范围内作出选择，可以选择单选按钮或者多选按钮，在实际应用中“性别”、“籍贯”、“爱好”等常采用这种方法；如果还有让浏览者提交文件的需求，可以使用文件域，例如某游戏选拔女主人公原型，要求申请者提交照片等。

Input 标签

HTML 4.0 标准中引入了表单元素，对于大量通常的表单元素，可以使用<input>标签来进行定义，其中包括文本字段、多选列表、可单击的图像和提交按钮等。虽然<input>标签中有许多属性，但对每个元素来说，只有 Type 和 Name 属性是必需的。正如后面详细说明的那样，每种 input 元素都仅有一个其允许设置的属性的子集。当然，根据指定的表单元素类型，也可以设置一些其他的<input>属性。

基本语法	<input name="field_name" type="type_name">	
功能	定义表单输入元素	
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	用一个名称来标记输入元素，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	为输入元素创建一个 标记，应用超链接时可以用这标记来明确地引用该输入元素，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	创建输入元素内容的内联样式
	Title	给输入元素加上说明性的文字
	Name	输入元素的名称
	Type	输入元素的类型
	Alt	图像的替换文字
	Accesskey	链接热键





(续表)

属性及说明	属性	说明
	Accept	文件上传的MIME类型列表
	Checked	已选中
	Disabled	禁用某个输入元素
	MaxLength	输入元素的最大输入字符数
	Size	输入元素的宽度(以字符为单位)
	Src	输入元素的图像源文件
	TabIndex	对浏览器窗口中的对象重新排序
	Value	输入元素的默认值

可以用标签中的Name属性来为字段命名(表单提交处理过程中使用的字段,这种字段的说明可参阅前面的介绍)。从技术角度来讲,Name属性的值是任意的一个字符串,但是我们建议最好采用没有嵌入空格或标点的字符串。标签中必需的Type属性可以用来选择控件的类型,其属性值如下表所示。

Type属性值及说明	Type属性值	说明
	Text	文字域
	Password	密码域
	File	文件域
	Checkbox	复选框
	Radio	单选框
	Button	普通按钮
	Submit	提交按钮
	Reset	重置按钮
	Hidden	隐藏域
	Image	图像域(图像提交按钮)

文字域

最有用也是最常见的表单输入元素是文字域,在浏览器窗口中显示为一行的空框,它接收用户输入的一行信息,当用户将表单提交给服务器时,这些信息就变成了元素的值。如果要在文档的表单里创建一个文字域,将表单元素的type属性设为text就可以了;但同时要包括一个必需的name属性。

在不同的浏览器中,一行文本的组成成分也有所不同。HTML提供了一种解决方法,就是采用Size和MaxLength属性来分别规定文本输入显示框的长度(按字符的数目计算),以及能从用户那里接收的总字符数。这两个属性的值允许用户在字段内看到和填写字符的最大数量。如果MaxLength大于Size,那么文本会在文本输入框内来回滚动。如果MaxLength小于Size,那么文本输入框内会有一些多余的空格来填补这两个属性之间的差异。Size的默认值和浏览器的设置有关;MaxLength的默认值则不受限制。调整Size属性可以使文本输入框不会超过通常浏览器窗口的右边界。除非用户在文本字段内输入一些内容,否则文本字段通常都是空的;但是也可以利用Value属性为字段设置一个初始的默认值,当然用户也可以修改这个默认值。

原始文件	Sample\Ch09\2\1.html
最终文件	Sample\Ch09\2\1-end.html
学习要点	使用 <input/> 标签的type属性的text值

1. 范例代码

```

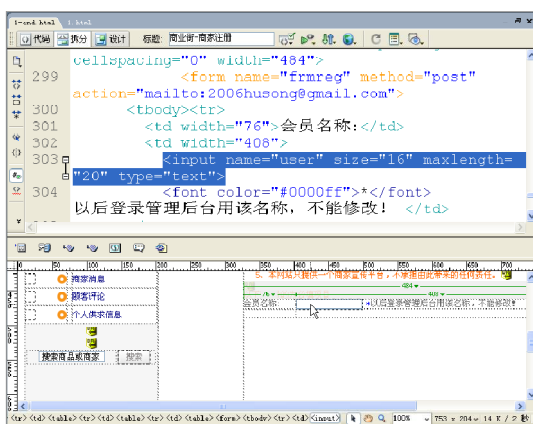
<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail
.com">
...
<input name="user" size="16"
maxlength="20" type="text">
...
</form>
...
</body>
</html>

```

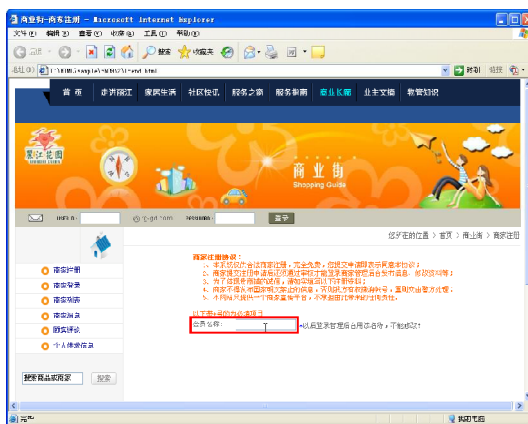
2. 范例解释

- 01 声明名称为 frmreg 的表单，使用 post 方法，提交到 2006husong@gmail.com 的邮箱中。
- 02 声明名称为 user、16 个字符宽度、最大输入字符宽度为 20 的单行文本域。
- 03 结束表单。

3. 代码和设计



4. 显示结果



密码域

将 Type 属性的值设为 Password，就可以创建一个密码域。传统文本域的其他属性和语义也可应用于密码域。应用密码域时必需指定一个字段的名称，这样就可以为该字段设置 Size 和 Maxlength 属性，并可以设置一个初始的 Value。





原始文件	Sample\Ch09\2\2.html
最终文件	Sample\Ch09\2\2-end.html
学习要点	使用 <input/> 标签的type属性的password值

1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
  action="mailto:
2006husong@gmail.com">
...
  登录密码:
  ...
  <input name="pass1" size="12"
maxlength="20" type="password">
  <font color="#0000ff">*</font>
  不能少于四位!
  ...
  密码确认:
  ...
  <input name="pass2" size="12"
maxlength="20" type="password">
  <font color="#0000ff">*</font>
  ...
</form>
...
</body>
</html>
```

2. 范例解释

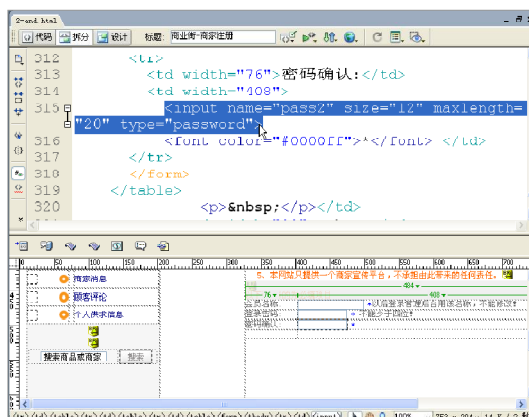
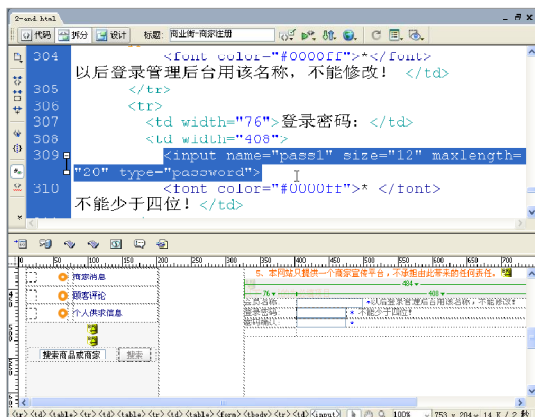
01 声明名称为 frmreg 的表单，使用 post 方法，并提交到 2006husong@gmail.com 的邮箱中。

02 声明名称为 pass1、12 个字符宽度、最大输入字符宽度为 20 的密码域。

03 声明名称为 pass2、12 个字符宽度、最大输入字符宽度为 20 的密码域。

04 结束表单。

3. 代码和设计



4. 显示结果



文件域

顾名思义，文件域能够让用户选择存储在本地计算机上的文件，并且在用户提交表单时将它发送给服务器。和其他文本字段一样，浏览器也为用户提供了文件输入框，在该框的右侧通常会有一个 Browse（浏览）按钮。用户可以在字段内直接输入文本来指明文件的路径名；也可以利用浏览功能，从系统指定的对话框中选择。

通过将 Type 属性的值设置为 File，即可创建一个文件域。和其他文本域一样，文件域的 Size 和 Maxlength 属性应设为适当的值，如果不另行指定，浏览器会默认生成一个 20 个字符宽的字段。由于文件和路径名会因系统的差异而千变万化，因此无法为这个元素提供一个默认值。因此，Value 属性不能用于这种文本元素。“浏览”按钮可以打开特定平台上的文件选择对话框，这个对话框允许用户为该字段选择一个值。在这种情况下，即使所选文件的路径名长度超出了指定的 Maxlength，还是可以将完整的路径名称放进这个字段中。

使用 Accept 属性可约束浏览器允许用户选择的文件类型。其值是一个用逗号分隔的 MIME 编码列表；用户可以仅选择那些与列表类型相匹配的文件。例如，要将文件选择的类型限制为图像，则应在文件域的 <input> 标签中加上 “accept=“image/*””。

与其他表单输入元素不同，文件域只有在特定的表单数据编码方式和传输方法下才能正常工作。如果要在表单中包括一个或多个文件域，必需把 <form> 标签的 Enctype 属性设置为 Multipart/Form-Data，并把 <form> 标签的 method 属性设为 post。否则，这个文件选择字段的行为就会像普通的文本字段一样，把它的值（文件的路径名称）传输给服务器，而不是传输文件本身的内容。

原始文件	Sample\Ch09\2\3. html
最终文件	Sample\Ch09\2\3-end. html
学习要点	使用 <input> 标签的 type 属性的 file 值





1. 范例代码

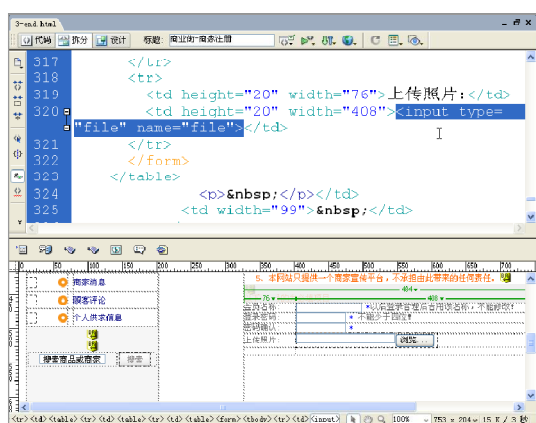
```
<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
上传照片:
...
<input type="file" name="file">
...
</form>
...
</body>
</html>
```

2. 范例解释

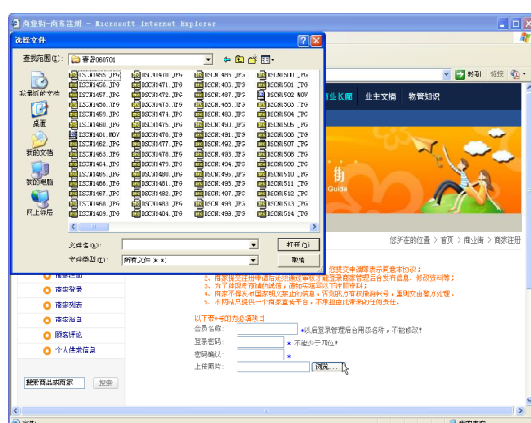
01 声明名称为 frmreg 的表单，使用 post 方法，并提交到 2006husong@gmail.com 的邮箱中。

02 声明名称为 file 的文件域。

3. 代码和设计



4. 显示结果



复选框

复选框 (checkbox) 表单控件为用户提供了一种在表单中选择或取消选择某个条目的快捷方式。复选框也可以集中在一起而产生一组选择，用户可以选择或取消选择组中的每一个选项。通过把每个标签中的 Type 属性都设置为 checkbox，就可以生成单独的复选框，其中要包括必需的 Name 和 Value 属性。如果用户选择了某项，在提交表单时，它就要给出一个值。如果用户没有选择该项，该元素就不会给出任何值。如果用户没有用鼠标单击某个复选框取消选择，那么可选的 Checked 属性将告诉浏览器要显示一个选中状态的复选框，并且告诉浏览器向服务器提交表单时要包含一个值。在将表单提交给服务器时，浏览器会包含选中的复选框的值和其他表单参数的值。选中的复选框的这些值是在必需的 Value 属性中指定的文本字符串。

原始文件	Sample\Ch09\2\4. html
最终文件	Sample\Ch09\2\4-end. html
学习要点	使用 <input/> 标签的type属性的checkbox值

1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
选择分类:
...
<input type="checkbox"
name="checkbox"
value="checkbox" checked>电器
<input type="checkbox"
name="checkbox2"
value="checkbox">家居
<input type="checkbox"
name="checkbox3"
value="checkbox">饰品
<input type="checkbox"
name="checkbox4"
value="checkbox">服装
...
</form>
...
</body>
</html>

```

2. 范例解释

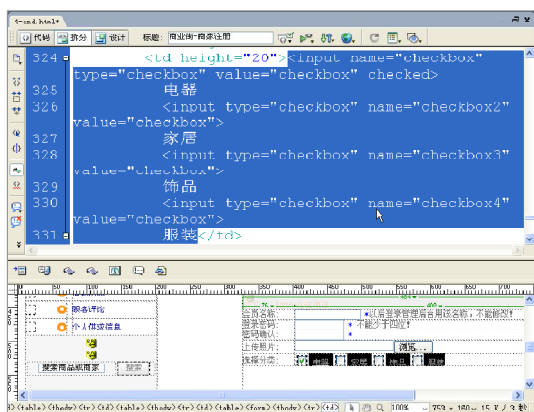
- 01 声明名称为 checkbox、值为 checkbox 的复选框默认被选中。
- 02 声明名称为 checkbox2、值为 checkbox 的复选框。
- 03 声明名称为 checkbox3、值为 checkbox 的复选框。
- 04 声明名称为 checkbox4、值为 checkbox 的复选框。



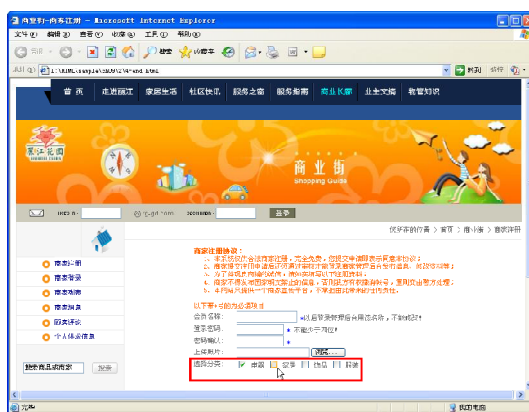
9

表单元素

3. 代码和设计



4. 显示结果





单选按钮

单选按钮（radio button）表单控件同复选框的行为非常相似，但在的一组单选按钮中用户只能选择其中一个。通过把的Type属性设置为radio，就可以创建一个单选按钮，如同复选框一样，每个单选按钮都需要一个Name和Value属性。具有相同名称的单选按钮会在同一个组中。如果在Checked属性中设置了该组中的某个元素，就意味着该按钮在开始时处于选中状态；如果没有选中这一组中的元素，浏览器会自动选定这一组中的第一个元素。设计者应该为每个单选按钮元素都设置一个不同的值，这样在提交表单之后，表单处理服务器就可以为它们自动分类。

原始文件	Sample\Ch09\2\5.html
最终文件	Sample\Ch09\2\5-end.html
学习要点	使用 <input/> 标签的type属性的radio值

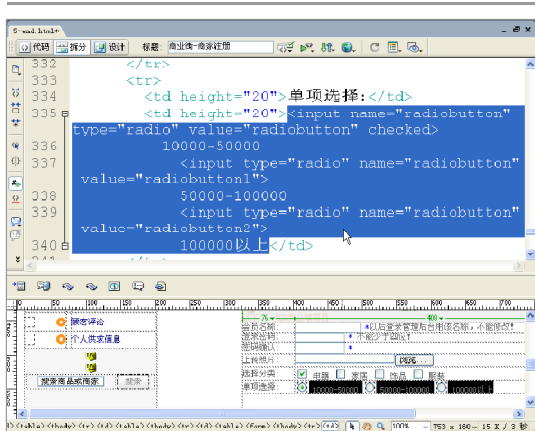
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
单项选择:
...
<input name="radiobutton"
type="radio" value="radiobutton"
checked>10000-50000
<input type="radio"
name="radiobutton"
value="radiobutton1">50000-
100000
<input type="radio"
name="radiobutton"
value="radiobutton2">100000 以上
...
</form>
...
</body>
</html>
```

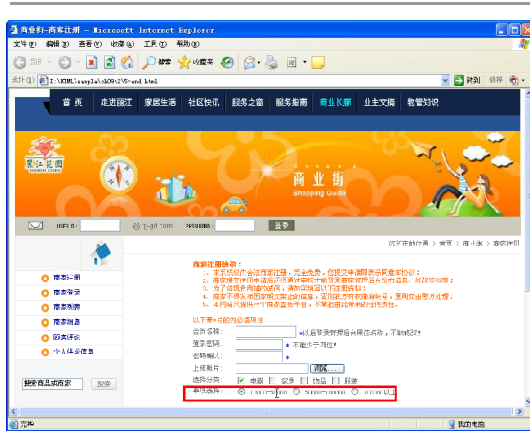
2. 范例解释

- 01 声明名称为radiobutton、值为radiobutton的单选按钮，默认被选中。
- 02 声明名称为radiobutton、值为radiobutton1的单选按钮。
- 03 声明名称为radiobutton、值为radiobutton2的单选按钮。

3. 代码和设计



4. 显示结果



普通按钮

使用标签可以生成一个供用户单击的按钮,但这个按钮不能提交或重置表单,可以用Value 属性来设置按钮上的标记,如果指定了Name 属性,就会把提供的值传递给表单处理程序。

普通按钮与提交按钮和重置按钮都有共同的属性和方法,也有各自不同的属性和方法。它们都可以在<Input>标签中通过Type="..." 来创建;与提交按钮和重置按钮不同的是普通按钮没有默认动作,需要通过使用onCl ick 事件句柄,在此事件触发时调用函数。

原始文件	Sample\Ch09\2\6.html
最终文件	Sample\Ch09\2\6-end.html
学习要点	使用 <input type="button"/> 标签的type属性的button值

1. 范例代码

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

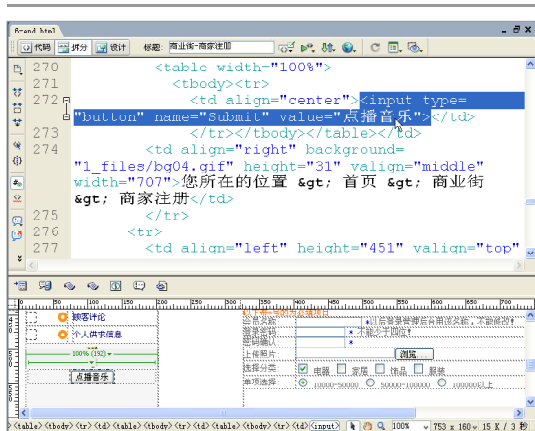

```

2. 范例解释

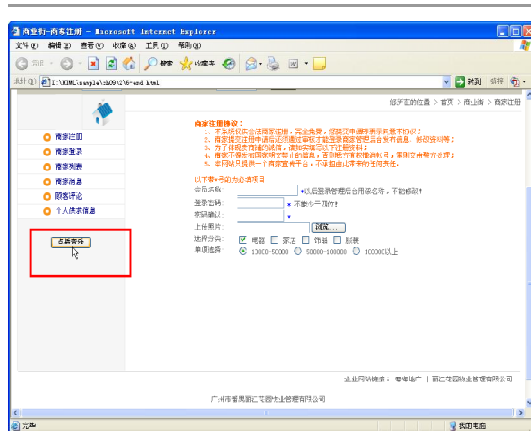
声明名称为Submit、值为“点播音乐”的按钮,因为指定了Name 属性,所以会把提供的值Submit 传递给表单处理程序。



3. 代码和设计



4. 显示结果



提交按钮

提交按钮 (submit button) 会启动将表单数据从浏览器发送给服务器上的提交过程。一个表单中可以有多个提交按钮，也可以利用表单按钮的提交类型设置 Name 和 Value 属性。对于最简单的提交按钮 (这个按钮不包括 Name 或 Value 属性) 来说，浏览器会显示一个小的长方形或椭圆形，上面有默认的标记 Submit (提交)。其他情况下，浏览器会用标签的 Value 属性设置的文本来标记按钮。如果给出了一个 name 属性，当浏览器将表单信息发送给服务器时，也会将提交按钮的 Value 属性的值添加到参数列表中。

原始文件	Sample\Ch09\2\7.html
最终文件	Sample\Ch09\2\7-end.html
学习要点	使用 <input/> 标签的type属性的submit值

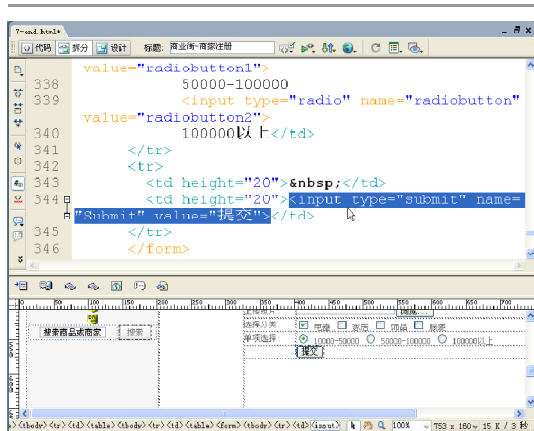
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
<input type="submit"
name="Submit" value="提交">
...
</form>
...
</body>
</html>
```

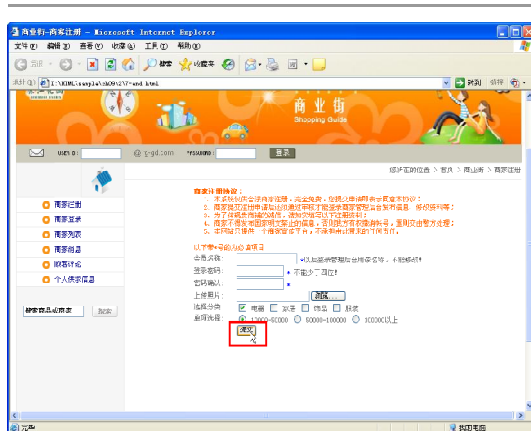
2. 范例解释

- 01 声明名称为 frmreg 的表单，使用 post 方法，提交到 2006husong@gmail.com 的邮箱中。
- 02 声明名称为 Submit、值为“提交”的提交按钮，当单击这个按钮后，就将把表单内容提交到 2006husong@gmail.com 的邮箱中。
- 03 结束表单。

3. 代码和设计



4. 显示结果



重置按钮

`<input>` 表单按钮的重置 (reset) 类型几乎是不言而喻的：它允许用户重置表单中的所有元素，也就是清除或设置某些默认值。与其他按钮不同，重置按钮不会激活表单处理程序。相反，浏览器将完成所有重置表单元素的工作。默认情况下，浏览器会显示一个标记为 Reset (重置) 的重置按钮。设计者可以在 Value 属性中指定自己的按钮标记，或改变默认设置。

原始文件	Sample\Ch09\2\8.html
最终文件	Sample\Ch09\2\8-end.html
学习要点	使用 <code><input></code> 标签的 type 属性的 reset 值

1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.com">
...
<input type="reset"
name="Submit2" value="重置">
...
</form>
...
</body>
</html>

```

2. 范例解释

- 01 声明名称为 frmreg 的表单，使用 post 方法，提交到 2006husong@gmail.com 的邮箱中。
- 02 声明名称为 Submit2、值为“重置”的重置按钮，当单击这个按钮后，就可以把名称为 frmreg 的表单内容清除。
- 03 结束表单。

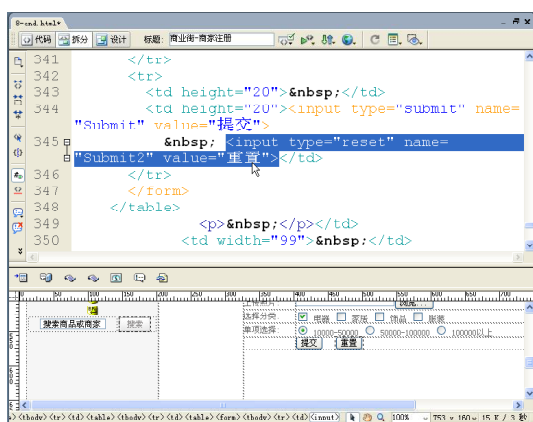


9

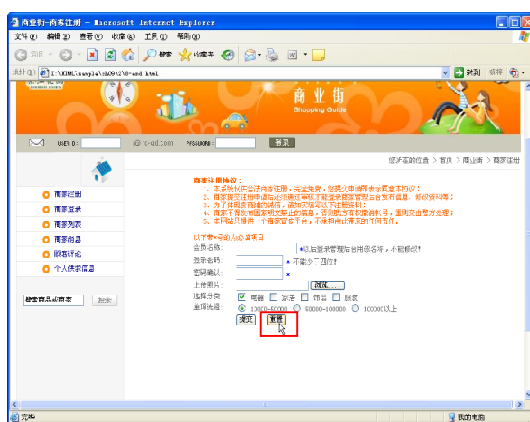
表单元素



3. 代码和设计



4. 显示结果



图像提交按钮

`<input>` 表单元素的 `image` 类型会创建一个定制按钮，这个按钮是个可单击的图像。这是一个用特定图像生成的特殊按钮，当用户单击该按钮后，浏览器会将表单提交给服务器。除图像文件的 URL 之外，图像按钮还需要一个 `Src` 属性，另外还包括一个 `Name` 属性和一个针对非图形浏览器的描述性 `Alt` 属性。

原始文件	Sample\Ch09\2\9.html
最终文件	Sample\Ch09\2\9-end.html
学习要点	使用 <code><input></code> 标签的 <code>type</code> 属性的 <code>image</code> 值

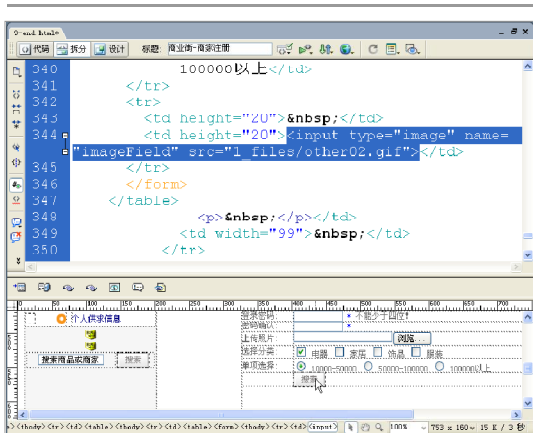
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.com">
...
<input type="image"
name="imageField" src="1_files/
other02.gif">
...
</form>
...
</body>
</html>
```

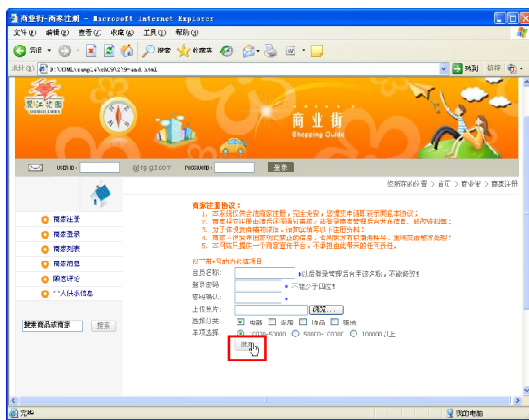
2. 范例解释

- 01 声明名称为 `frmreg` 的表单，使用 `post` 方法，提交到 `2006husong@gmail.com` 的邮箱中。
- 02 声明名称为 `imageField`、源文件为 `1_files` 文件夹下的 `other02.gif` 的图像提交按钮，当单击这个按钮后，就会把表单内容提交到 `2006husong@gmail.com` 的邮箱中。
- 03 结束表单。

3. 代码和设计



4. 显示结果



隐藏域

`<input>` 表单的隐藏控件从视觉上是无法看到的。这是一种向表单中嵌入信息的方法，以便这些信息不会被忽略或改变。不仅如此，在 `<input type="hidden">` 标签中必需的 Name 和 Value 属性，还会自动包括在提交的表单参数列表中。这些属性可用来给表单做标记，并且可以用来区分表单。

隐藏字段的另一个作用是管理用户和服务器的交互操作。例如它可以帮助服务器知道，当前的表单来自于一个几分钟前发出类似请求的人。通常情况下，服务器不会保留这种信息，并且每台服务器和用户之间的处理过程与其他事务无关。例如，用户提交的第一个表单也许需要一些基本信息（用户名字和住址等信息）；基于这些初始信息，服务器可能会创建第二个表单，向用户询问一些更详细的问题；重新输入第一个表单中的基本信息对用户来说太烦了，因此可以对服务器端进行编程，将这些值直接保存在第二个表单的隐藏字段中；当返回第二个表单时，这两个表单中的所有重要信息都保存下来了，如果必要，还可以查看第二个表单中的值是否和第一个表单相匹配。

原始文件	Sample\Ch09\2\10.html
最终文件	Sample\Ch09\2\10-end.html
学习要点	使用 <code><input></code> 标签的 type 属性的 hidden 值

1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
  action="mailto:
2006husong@gmail.com">
...
```

2. 范例解释

01 声明名称为 frmreg 的表单，使用 post 方法，提交到 2006husong@gmail.com 的邮箱中。

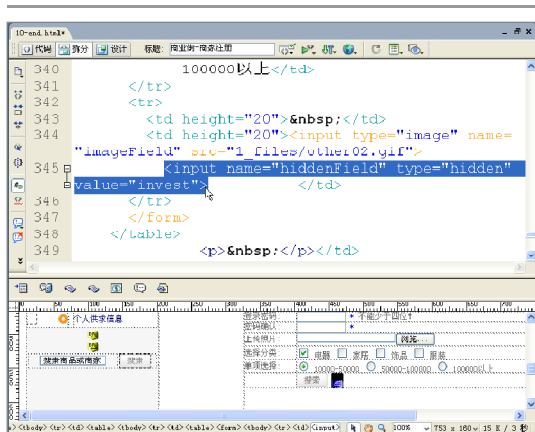


```
<input name="hiddenField"
type="hidden" value="invest">
...
</form>
...
</body>
</html>
```

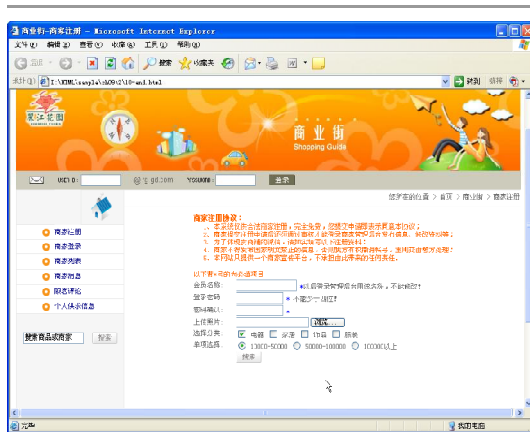
02 声明名称为hiddenField、值为invest的隐藏域在页面中不显示出来。

03 结束表单。

3. 代码和设计



4. 显示结果



Textarea标签

作为表单的一部分，<textarea>标签可在用户浏览器中创建一个多行文字域。在此文本输入区内，用户几乎可以输入无限行的文字。提交表单之后，浏览器将把所有行中的文字都收集起来，行间用回车 / 换行符分隔，并将它们作为表单元素的值发送给服务器，这个值必需使用Name属性中指定的名称。

基本语法	<pre><textarea name="name" rows=alue cols=alue value="value"> </textarea></pre>	
功能	定义表单多行文字域	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记多行文字域，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	为多行文字域创建一个标记，应用超链接时可以用这个标记来明确地引用该多行文字域，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	创建多行文字域内容的内联样式
	Title	给多行文字域加上说明性的文字

(续表)

属性及说明	属性	说明
	Name	输入元素的名称
	Accesskey	链接热键
	Di sabled	禁用多行文字域
	Tabindex	对浏览器窗口中的对象重新排序
	Rows	多行文字域的行数
	Col s	多行文字域的列数
	Wrap	多行文字域的换行

在<textarea>标签及其结束标签(</textarea>)之间可以包括纯文本,它的默认文本必需是纯文本(没有标签或其他特殊元素)。用户可以修改其中的内容,如果用户按下表单中的“重置”按钮,浏览器会使用修改过的文本作为默认值。

多行文本输入区在屏幕上独立存在的:文本主体内容可以在它的上面和下面显示,但是不会环绕它。然而,通过定义可视矩形区域的Col s和Rows属性可以控制其维数,这个矩形区域是浏览器专门用来显示多行输入的区域。这两个属性都需要用整数值来表示以字符为单位的维数大小。通常浏览器都会为<textarea>输入内容设置一个最小的可读区域,而且用户无法更改它的大小。浏览器会自动滚动显示那些超出设定维数的文本。

通常情况下,当用户在文本输入区中输入文本后,浏览器会将它们输入时的状态发送给服务器,只在用户按下Enter键的地方换行,但用户通常并不希望这样做,因此可以在文本输入区内启动自动换行功能。当用户输入的一行文字长于文本区的宽度时,浏览器会自动将多余的文字挪到下一行,在文字中最近的那一点换行。如果将Wrap属性设为Virtual,文字将在文本区内实行换行,以便显示给用户;但在传输给服务器时,文字只在用户按了Enter键的地方进行了换行,其他地方则没有换行的效果。如果将Wrap属性设为physical,文本将在文本输入区内实行换行,并以这种形式传送给服务器。因为文本要以用户在文本区内看到的方式传输给服务器,因此使用自动换行是非常有用的方法。如果设置为Wrap=Off,文本区会包括一行,用户必须将光标移动到右边才能看到全部文本,这时将把一行文本传送给服务器。Wrap的属性值如下表所示:

Wrap属性值及说明	Wrap 属性值	说明
	Virtual	虚拟换行
	Physical	物理换行
	Off	不换行

原始文件	Sample\Ch09\2\11.html
最终文件	Sample\Ch09\2\11-end.html
学习要点	使用<textarea>标签



9

表单元素



1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
主营业务:
...
<textarea rows="3" name="dobuy"
cols="40">
</textarea>
<font color="#0000ff">*</font>
...
</form>
...
</body>
</html>

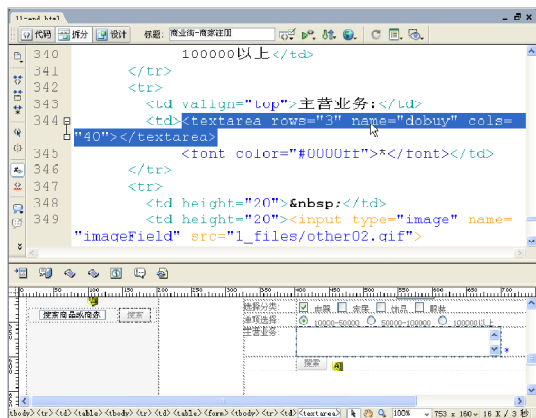
```

2. 范例解释

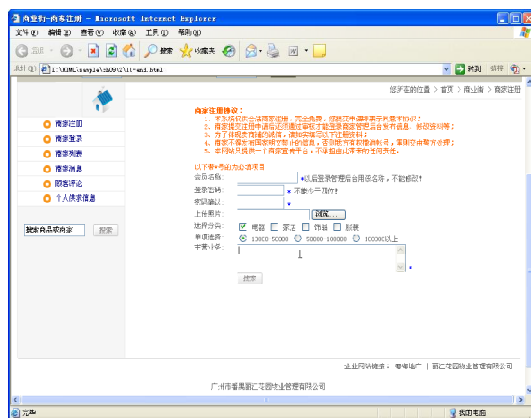
01 声明3行40个字符宽度、名称为dobuy的多行文字域。

02 结束多行文字域。

3. 代码和设计



4. 显示结果



Select标签

复选框和单选按钮为设计者们提供了强大的功能，便于创建多项选择的问题和答案。但是它们可能会是很长的表单，编写起来也许非常麻烦，而且在屏幕上会显得杂乱无章。因此，<select>标签提供了两种简洁的选择：下拉菜单和下拉列表。

将一列用<option>标签标记的条目放到表单的<select>标签中，这样就可创建一个选项的下拉式菜单。与其他表单标签一样，Name属性在这里也是必需的，在将<select>选项提交给服务器时，浏览器将使用它。与单选按钮不同，这里不会预先选定任何选项，因此如果没有选定值的话，提交表单时也就不会有值发送给服务器。而选定值后浏览器会提交选定的选项，或者收集用逗号分隔的多个选项，将其合成为一个单独的参数列表，并且在将<select>表单数据提交给服务器时包含Name属性。

基本语法	<select name="name" size=value>	
	选项	
	</select>	
功能	定义菜单或列表	
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或列表, 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	为菜单或列表创建一个标记, 应用超链接时可以用这个标记来明确地引用该菜单或列表, 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	创建菜单或列表内容的内联样式
	Title	给菜单或列表加上说明性的文字
	Name	菜单或列表的名称
	Disabled	禁用某个菜单或列表
	Size	菜单或列表的高度
	TabIndex	对浏览器窗口中的对象重新排序
	Multiple	列表中的多选题目

如果希望一次允许选择多个选项的话, 可以在<select>标签中加入Multiple属性, 这样可以让<select>元素像☐元素那样起作用。如果没有指定multiple, 那么一次只能选定一个选项, 如同单选按钮组那样。Size (大小) 属性决定了用户一次可以看到多少个选项。Size 的值应该是一个正整数, 没有指定Size 时, 默认值是1。当Size=1 时, 如果没有指定Multiple, 浏览器通常会将<select>列表显示成一个弹出式菜单; Size 的值超过1或者是指定了Multiple 属性, <select>会显示为一个滚动列表。

Option 标签

使用<option>标签可以定义一个<select>表单控件中的每个条目。浏览器将<option>标签中的内容作为<select>标签的菜单或是下拉列表中的一个元素显示, 这样其内容只能是纯文本。使用Value 属性可以为每个选项设置一个值, 当用户选中该选项时, 浏览器会将其发送给服务器。

基本语法	<option>	
	选项	
	</option> (可省略)	
功能	定义菜单或列表中的可选项	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记可选项, 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	为可选项创建一个标记, 将来超链接可以用这个标记来明确地引用该可选项, 以便作为样式表选择器





(续表)

属性及说明	属性	说明
	Id	或使用其他应用程序来执行自动搜索
	Style	创建可选项内容的内联样式
	Title	给可选项加上说明性的文字
	Disabled	禁用某个可选项
	Value	可选项的值
	Selected	默认被选中的可选项

默认情况下,所有多选的<select>标签中的选项都是未选中状态,因此当客户端向服务器提交表单时,这些选项不会包括在参数列表中。在<option>标签中包括Selected属性后,就可以预先选定一个或多个选项,随后用户可以再取消选择。Selected属性的HTML版本没有值,如果没有任何选项被预先选定,那么单选的<select>标签会预先选中第一个选项。

原始文件	Sample\Ch09\2\12.html
最终文件	Sample\Ch09\2\12-end.html
学习要点	使用<select>标签和<option>标签制作下拉列表

1. 范例代码

2. 范例解释

```
<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
所属类型:
...
<select name="typesup" size="1">
<option value=""
selected="selected">请选择</option>
<option value=" 购物 ">购物</option>
<option value=" 吃喝 ">吃喝</option>
<option value=" 保健 ">保健</option>
<option value=" 健身 ">健身</option>
<option value=" 教育 ">教育</option>
<option value=" 服务 ">服务</option>
<option value=" 娱乐 ">娱乐</option>
<option value=" 其他 ">其他</option>
</select>
<font color="#0000ff">*</font>
.....
</form>
.....
</body>
</html>
```

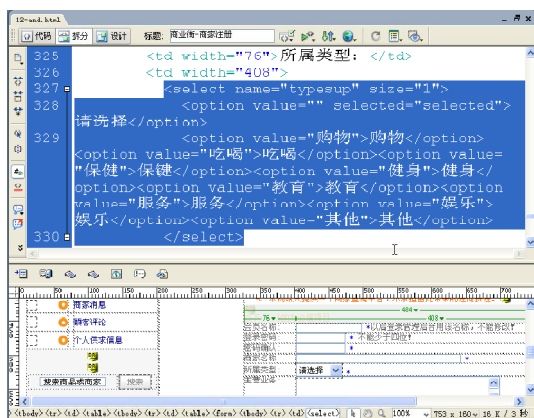
01 声明名称为 typesup, 高度为 1 的下拉列表。

02 声明“请选择”选项, 默认被选中。

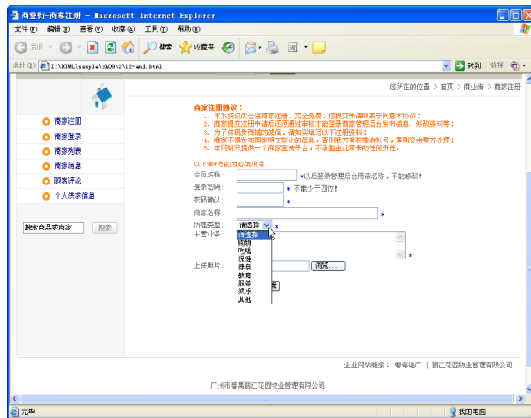
03 声明 8 个选项, 分别为购物、吃喝、保健、健身、教育、服务、娱乐、其他。

04 结束下拉列表。

3. 代码和设计



4. 显示结果



原始文件	Sample\Ch09\2\13.html
最终文件	Sample\Ch09\2\13-end.html
学习要点	使用<select>标签和<option>标签制作列表框

1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
所属类型:
...
<select name="typesup" size="9">
<option value=""
selected="selected">请选择</option>
<option value="购物">购物</option>
<option value="吃喝">吃喝</option>
<option value="保健">保健</option>
<option value="健身">健身</option>
<option value="教育">教育</option>
<option value="服务">服务</option>
<option value="娱乐">娱乐</option>
<option value="其他">其他</option>
</select>
<font color="#0000ff">*</font>
...
</form>
...
</body>
</html>

```

2. 范例解释

- 01 声明名称为 typesup，高度为 9 的列表框。
- 02 声明“请选择”选项，默认被选中。
- 03 声明 8 个选项，分别为购物、吃喝、保健、健身、教育、服务、娱乐、其他。
- 04 结束列表。

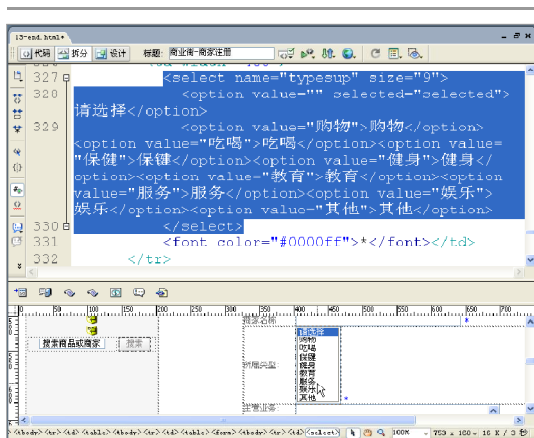


9

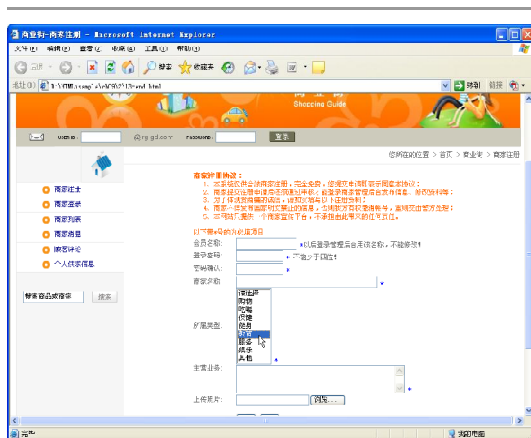
表单元素



3. 代码和设计



4. 显示结果



Optgroup标签

表单中的选项菜单有可能非常长，这使得它们很难显示和使用。在这种情况下，可以将一些相关的选项分组，然后将这些组作为一组嵌套的层叠式菜单呈现给用户。<optgroup>在HTML 4.0中是新引入的标签，虽然它的使用方式有限，但是它为HTML的表单提供了嵌套层叠菜单功能。只能在<select>标签中使用<optgroup>标签，并且它只能包含<option>标签。浏览器会为<select>主菜单中的每个<optgroup>标签都生成子菜单。

基本语法	<optgroup label="value">	
	选项	
	</optgroup> (可省略)	
功能	将菜单或列表中的相关可选项组织在一起	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记子菜单，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	为子菜单创建一个标记，应用超链接时可以用这个标记来明确地引用该子菜单，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	创建子菜单内容的内联样式
	Title	给子菜单加上说明性的文字
	Disabled	禁用某个子菜单
	Label	定义子菜单的标题
	Start	设定AVI文件播放方式

使用Label属性可以定义一个<optgroup>子菜单的标题以显示给用户。其标记应该简短，这样才能保证菜单在不同的显示情况下都能显示出来。

原始文件	Sample\Ch09\2\14.html
最终文件	Sample\Ch09\2\14-end.html
学习要点	使用<optgroup>标签

1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
所属类型:
...
<select name="typesup">
<option value=""
selected="selected">请选择</
option>
<optgroup label="选项组 1">
<option value="购物">购物</option>
<option value="吃喝">吃喝</option>
<option value="保健">保健</option>
<option value="健身">健身</option>
</optgroup>
<optgroup label="选项组 2">
<option value="教育">教育</option>
<option value="服务">服务</option>
<option value="娱乐">娱乐</option>
<option value="其他">其他</option>
</optgroup>
</select>
<font color="#0000ff">*</font>
...
</form>
...
</body>
</html>

```

2. 范例解释

01 声明标签为“选项组 1”的子选项组。

02 声明“选项组 1”的子选项组结束。

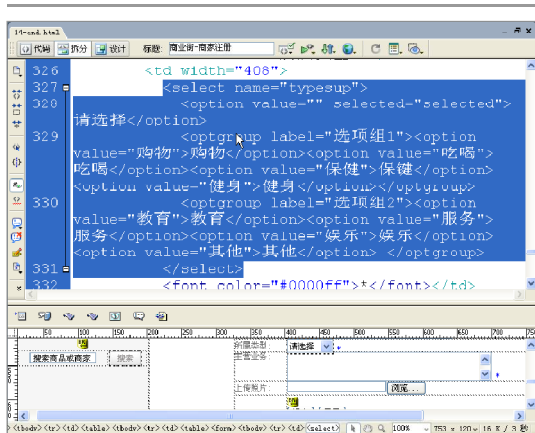
03 声明标签为“选项组 2”的子选项组。

04 声明“选项组 2”的子选项组结束。

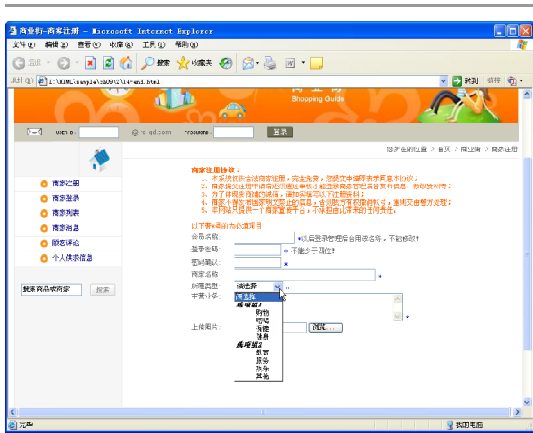




3. 代码和设计



4. 显示结果



3

表单元素分组

HTML 4.0 标准引入了 3 个标签，它们为用户提供了更简便的对表单的引导。它们能够将表单分组并加上标题，也可以对每一个表单控件分别进行标记。如果浏览器能够对它们进行特殊处理（譬如可以用语言生成器提供这种服务，同时也可以特殊显示出来），还可以很容易地从用户的键盘那里获得信息。它们分别如下表所示。

<label>	标签元素标记
<fieldset>	元素区域分组标记
<legend>	分组区域的标题显示标记

Label 标签

使用 label 标签可以定义表单控制间的关系（例如一个文本输入字段和一个或多个文本标记之间的关系）。根据最新的标准，标记中的文本可以得到浏览器的特殊支持。浏览器可以为这个标签选择一种特殊的显示样式（也可以使用样式表），并且当用户选择该标签时，浏览器会自动将焦点转到和标签相关的表单元素上。

基本语法	<label for="value">	
	元素	
	</label>	
功能	为表单元素创建标签	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Class	用一个名称来标记标签，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	为子菜单创建一个标签，应用超链接时可以用这个标记来明确地引用该标签，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

(续表)

属性及说明	属性	说明
	Style	创建标签内容的内联样式
	Title	给标签加上说明性的文字
	For	命名一个目标表单Id
	Accesskey	链接热键

原始文件	Sample\Ch09\3\1.html
最终文件	Sample\Ch09\3\1-end.html
学习要点	使用<label>标签



9

表单元素

1. 范例代码

2. 范例解释

```

<html>
<head>
...
</head>
<body>
...
<form name="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
  单项选择:
  ...
  <label for="a">
  <input name="radiobutton"
  type="radio" value="radiobutton"
  checked id="a">      10000-50000
  </label>
  <label for="b">
  <input type="radio"
  name="radiobutton"
  value="radiobutton1" id="b">
  50000-100000
  </label>
  <label for="c">
  <input type="radio"
  name="radiobutton"
  value="radiobutton2" id="c">
  100000 以上
  </label>
  ...
</form>
...
</body>
</html>

```

01 为id为a的单选按钮及10000-50000文字声明标签。

02 结束a 标签。

03 为id为b的单选按钮及文字50000-100000声明标签。

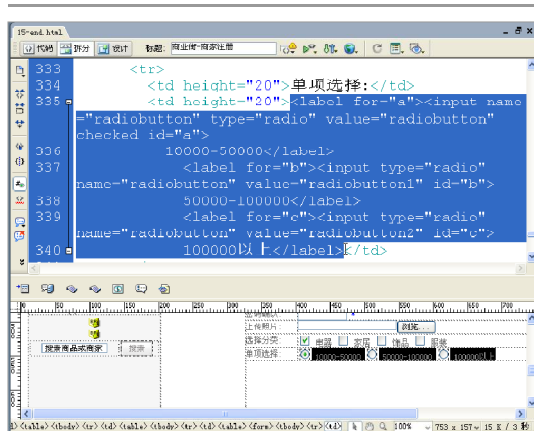
04 结束b 标签。

05 为id为c的单选按钮及文字100000以上声明标签。

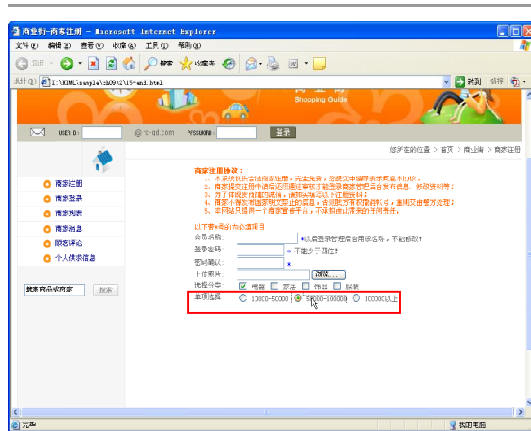
06 结束c 标签。



3. 代码和设计



4. 显示结果



Fieldset 标签

除了单独的标记外,也可以将一群表单元素生成一个组,并用<fieldset>和<legend>标签来标记这个组。<fieldset>标签将表单内容的一部分打包,生成一组相关表单字段。<fieldset>没有必需的或是惟一的属性。当一组表单元素放到<fieldset>标签内时,浏览器会以特殊方式来显示它们,例如特殊的边界、3D 效果,甚至一个子表单。

基本语法	<fieldset>	
	元素	
	</fieldset>	
功能	将表单的相关元素分组	
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组,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为分组创建一个标签,应用超链接时可以用这个标记来明确地引用该分组,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	为分组创建一个标签,应用超链接时可以用这个标记来明确地引用该分组,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	创建分组内容的内联样式
	Title	给分组加上说明性的文字

Legend 标签

使用<legend>标签可以为表单中的一个字段集合生成图标符号。这个标签可能仅能在<fieldset>里显示。与<label>类似,当<legend>内容被选定时,焦点会转移到相关的表单元素上,也可以用来提高用户对<fieldset>的控制。<legend>标签也支持Accesskey和Align属性。Align的值可以是Top、Bottom、Left 或 Right,向浏览器说明图标符号放在字段集中的位置。

基本语法	<legend>	
	图标	
	</legend>	
功能	为分组的表单元素创建符号	
属性及说明	属性	说明
	Cir	文本方向
	Lang	语言信息
	Class	用一个名称来标记分组，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	为分组创建一个标签，应用超链接时可以用这个标记来明确地引用该标签，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	创建分组内容的内联样式
	Title	给分组加上说明性的文字
	Accesskey	链接热键
	Align	排列方式

原始文件	Sample\Ch09\3\2. html
最终文件	Sample\Ch09\3\2-end. html
学习要点	使用<fieldset>标签和<legend>标签



1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<formname="frmreg" method="post"
action="mailto:2006husong@gmail.
com">
...
<fieldset>
<legend>普通</legend>
<fieldset>
<legend>基本信息</legend>
<input name="user" size="16"
maxlength="20" type="text">
<br>
<input name="pass1" size="12"
maxlength="20" type="password">
<br>
<input name="pass2" size="12"
maxlength="20" type="password">
<br>
```

2. 范例解释

- 01 声明表单元素分组。
- 02 声明这个组的符号文字为“普通”。
- 03 声明表单元素的子组。
- 04 声明这个子组的符号文字为“基本信息”。



```
<input name="name" size="40"
maxlength="50" type="text">
<br>
</fieldset>
<fieldset>
<legend>扩展信息</legend>
<select name="typesup">
<option value=""
selected="selected">请选择
</option>
<optgroup label=" 选项组 1">
<option value=" 购物 ">购物</option>
<option value=" 吃喝 ">吃喝</option>
<option value=" 保健 ">保健</option>
<option value=" 健身 ">健身</option>
</optgroup>
<optgroup label=" 选项组 2">
<option value=" 教育 ">教育</option>
<option value=" 服务 ">服务</option>
<option value=" 娱乐 ">娱乐</option>
<option value=" 其他 ">其他</option>
</optgroup>
</select>
<br>
<textarea rows="3" name="dobuy"
cols="40"></textarea>
<br>
<input type="file" name="file">
<br>
</fieldset>
</fieldset>
<input value=" 提交 " style="font-
size: 9pt; " name="B1"
onClick="ValidateInput()"
type="button">
<input name="Submit2" value=" 重置 "
style="font-size: 9pt; "
type="reset">
...
</form>
...
</body>
</html>
```

05 结束这个子组。

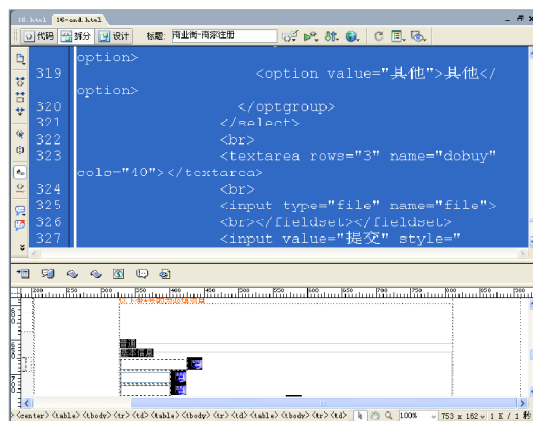
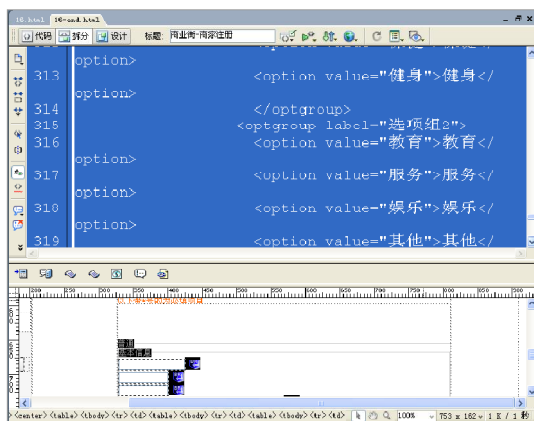
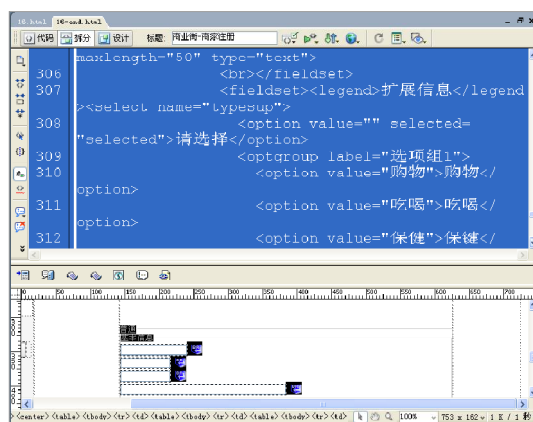
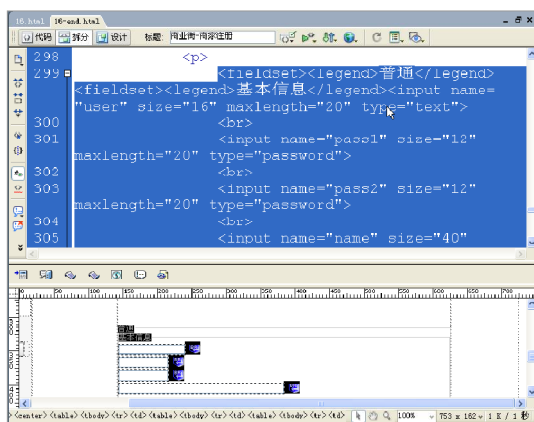
06 声明新的子组。

07 声明这个新子组的符号文字为“扩展信息”。

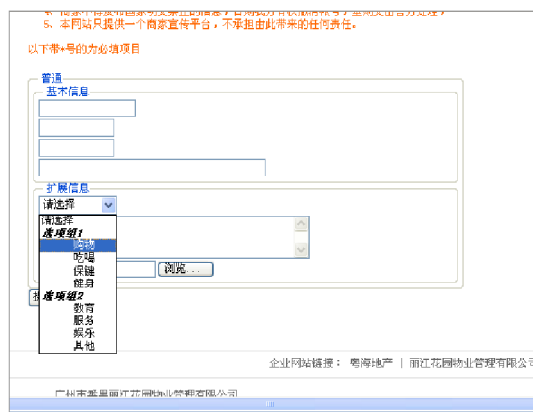
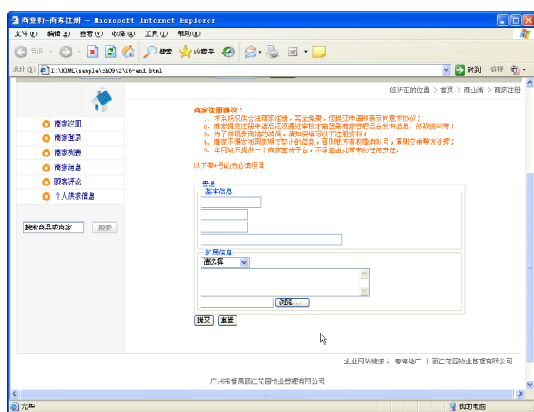
08 结束这个新子组。

09 结束整个表单的组。

3. 代码和设计



4. 显示结果



4

实例制作

下面通过一个实例来练习页面中不同类型的表单元素的实现。



原始文件	Sample\Ch09\4\1.html
最终文件	Sample\Ch09\4\1-end.html
学习要点	使用表单标记

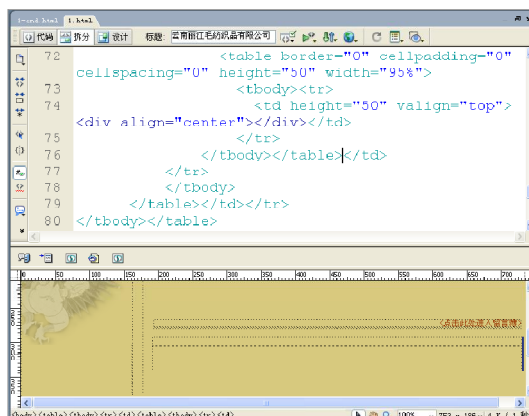


原始效果



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。



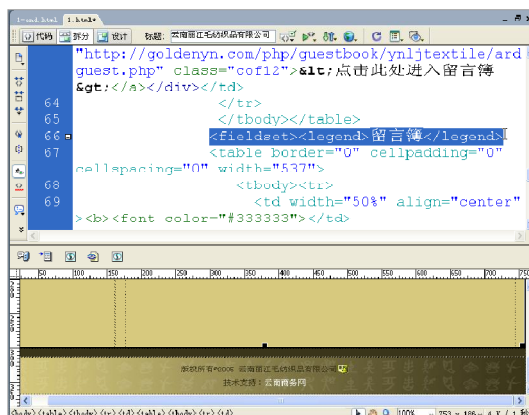
- 2 将光标定位到页面中间空白的单元格中如下代码的前面：

```
<table border="0" cellpadding="0"
cellspacing="0" width="537">
```

在代码视图中插入以下代码：

```
<fieldset>
<legend>
留言簿
</legend>
```

- 声明表单元素分组。
- 声明组的符号文字为“留言簿”。



- 3 将光标定位到页面中间左侧空白的单元格中如下代码的前面：

```
<td width="50%" align="center">
<b><font color="#333333">
```

在代码视图中插入以下代码：

```
<fieldset>
<legend>直接联系我们</legend>
```

- 声明表单元素再分一个子组。
- 声明子组的符号文字为“直接联系我们”。

- 4 将光标定位到刚插入的代码后面，插入如下代码：

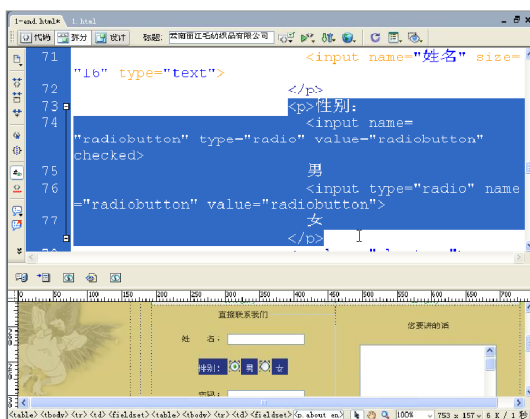
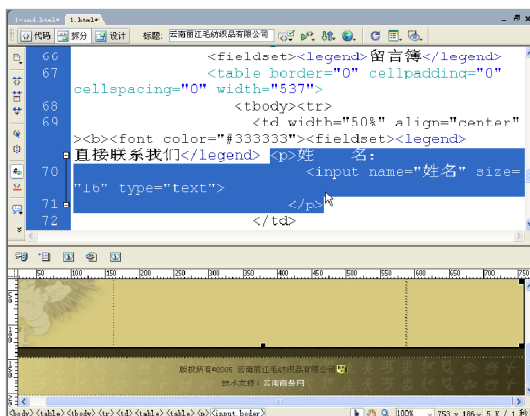
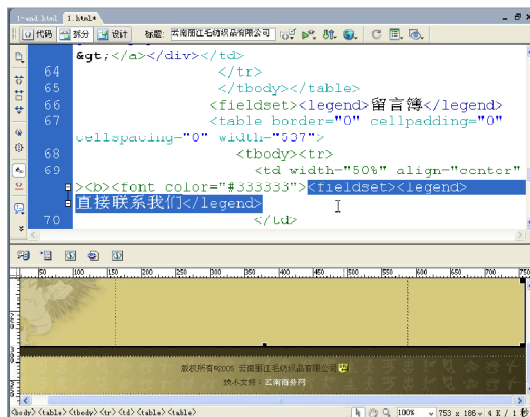
```
<p>姓 名:
<input name="姓名" size="16"
type="text">
</p>
```

- 声明名称为“姓名”、宽度为16个字符的文本框。

- 5 将光标定位到刚才插入的代码的后面，并插入如下代码：

```
<p>性别:
<input name="radiobutton"
type="radio" value="radiobutton"
checked>男
<input name="radiobutton"
type="radio" value="radiobutton">女
</p>
```

- 声明名称为radiobutton、值为radiobutton的单选按钮，并默认为选中。
- 声明名称为radiobutton、值为radiobutton的单选按钮。





- 6 将光标定位到刚才插入的代码的后面，并插入如下代码：

```
<p>密码:
<input name="联系电话" size="16"
type="text">
</p>
```

- ① 声明名称为“联系电话”、宽度为16个字符的文本框。

- 7 将光标定位到刚才插入的代码的后面，并插入如下代码：

```
<p>E-Mail:
<input name="email" size="16"
type="text">
</p>
```

- ① 声明名称为email、宽度为16个字符的文本框。

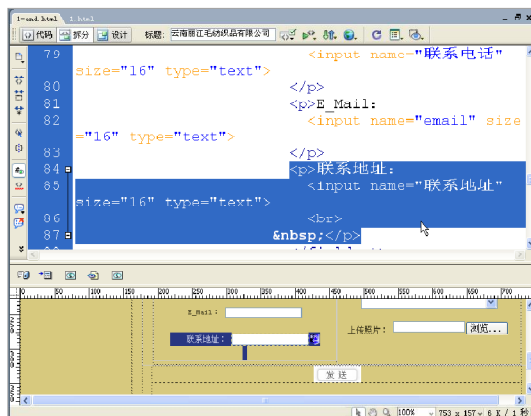
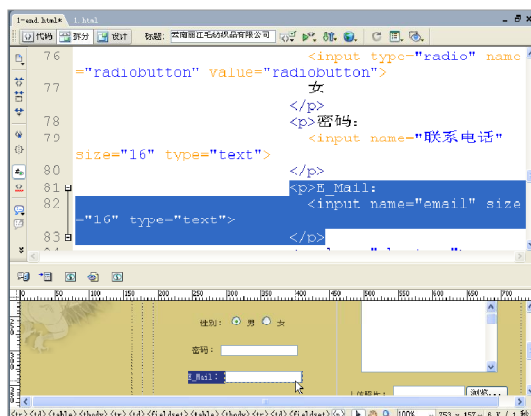
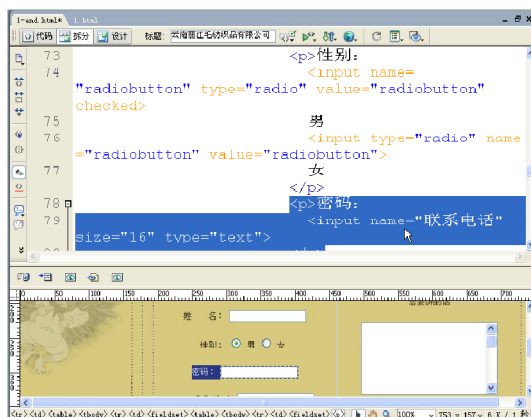
- 8 将光标定位到刚才插入的代码的后面，并插入如下代码：

```
<p>联系地址:
<input name="联系地址" size="16"
type="text">
<br>&nbsp;
</p>
</fieldset>
```

- ① 声明名称为“联系地址”、宽度为16个字符的文本框。
② 结束子组。

- 9 将光标放到刚才插入的代码的下面的如下代码后面：

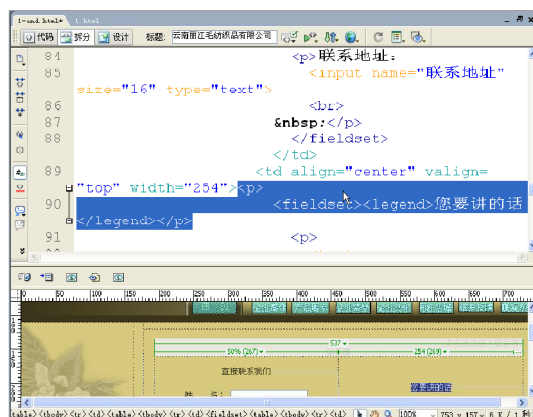
```
</td>
<td align="center" valign="top"
width="254">
```



10 插入如下代码:

```
<p>
<fieldset>
<legend>您要讲的话</legend>
</p>
```

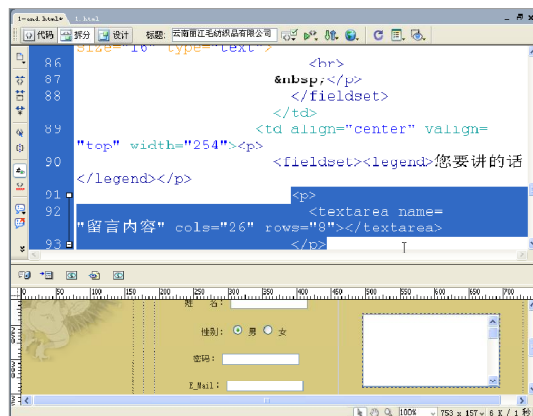
- ❶ 声明表单元元素的一个新子组。
- ❷ 声明这个子组的符号文字为“您要讲的话”。



11 将光标定位到刚才插入的代码的后面，并插入如下代码:

```
<p>
<textarea name="留言内容"
cols="26" rows="8">
</textarea>
</p>
```

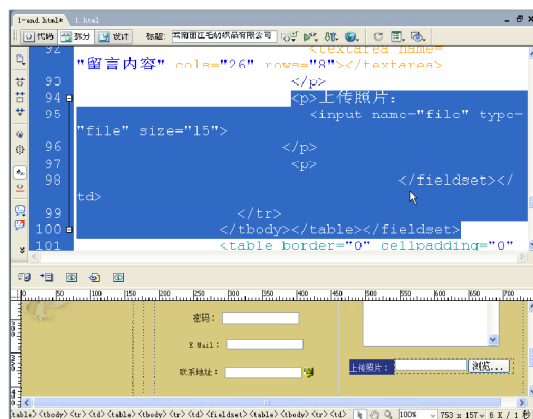
- ❶ 声明名称为“留言内容”、宽度为26个字符、8行的多行文字框。
- ❷ 结束多行文字框。



12 将光标定位到刚才插入的代码的后面，插入如下代码:

```
<p>上传照片:
<input name="file" type="file"
size="15">
</p>
</fieldset>
</td>
</tr>
</tbody>
</table>
</fieldset>
```

- ❶ 声明名称为file、宽度为15个字符的文件域”。
- ❷ 结束前面声明的子组。
- ❸ 行结束表单组。





- 13 将光标定位到页面中间下方空白的单元格中如下代码的后面：

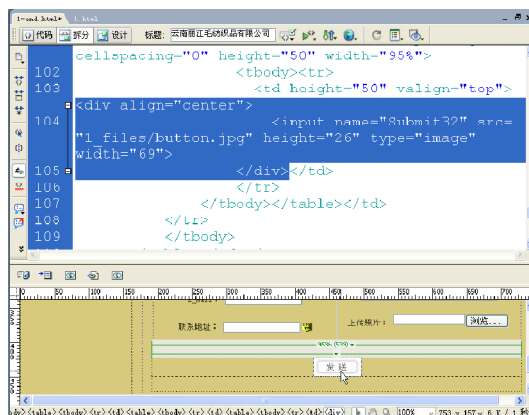
```
<td height="50" valign="top">
```

在代码视图中插入以下代码：

```
<div align="center">
<input name="Submit32"
src="1_files/button.jpg"
height="26" type="image"
width="69">
</div>
```

- 声明一个名称为submit32、源文件为1_files文件夹下的button.jpg，它是一个高度为26、宽度为69的图像提交按钮。

- 14 按下F12快捷键，可以在浏览器中看到页面的效果。





Chapter

10

使用表格排版网页

表格可以用于通常的表格化数据的显示，它在管理文件布局方面有着非常重要的作用。表格是用于排列内容的最佳方式，在HTML页面中，绝大多数页面都是使用表格进行排版的。

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



1

表格的整体结构

定义一个表格时,在<table>标签和</table>结束标签中间将包括所有元素。表格元素包括数据项、一行和列的表头、标题,它们都有自己的修饰标签;按照从上到下、从左到右的顺序,可以为表格中的每列定义表头和数据,然后一行接一行下去。

设计者可以将任何元素放进HTML的表格单元格中,包括图像、表单,分隔线,表头,甚至是另一个表格。浏览器将每个单元格作为一个窗口处理,让单元格的内容填满空间,当然在这个过程中会有一些特殊的格式规定和范围。

设计者只用5个标签就可以生成一个样式很复杂的表格:<table>标签在文档主体内容中封闭表格及其元素;<tr>标签定义表格中的一行;<td>标签定义数据单元格;<th>标签定义表头;<caption>标签定义表格标题。具体如下表所示。

标签	描述
<table>	表格
<tr>	行
<td>	单元格
<th>	表头
<caption>	表格标题

Table 标签

基本语法	<table>	
	表格内容	
	</table>	
功能	定义表格	
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	用一个名称来标记表格,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	为表格创建一个标记,应用超链接时可以用这个标记来明确地引用该表格,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	创建表格内容的内联样式
	Title	给表格加上说明性的文字
	Name	表格的名称
	Bgcolor	表格的背景颜色
	Background	表格的背景图像
	Bordercolor	表格的边框颜色

(续表)

属性及说明	属性	说明
	Bordercolorlight	表格的亮边框颜色
	Bordercolordark	表格的暗边框颜色
	Width	表格的宽度
	Height	表格的高度
	Border	表格的边框
	Cellpadding	单元格边距
	Cellspacing	单元格间距
	Frame	表格边框的可见方式
	Nowrap	禁止换行
	Rules	行与列之间边框显示的方式
	Summary	整个表格的基本概要描述

<table>标签和</table>结束标签在文档主体中定义一个表格。除非用样式表、段落、部分或是其他对齐选项在浏览器窗口中放置表格，浏览器会在当前的文本流停下来并换行，在新一行开始的地方插入表格，然后重新在表格后的新行开始文本流。

Tr 标签

使用<tr>标签可在表格中创建新的一行；在<tr>标签中可以放置一个或多个单元格（单元格包括由<td>标签定义的数据）。<tr>标签可以设置一定的特殊属性，然后和表格的通常属性一起来控制显示效果。

基本语法	<tr>	
	行内容	
	</tr>	
	定义表格的行	
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	用一个名称来标记表格的行，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	为表格的行创建一个标记，应用超链接时可以用这个标记来明确地引用该表格行，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
	Style	创建表格的行内容的内联样式
	Title	给表格的行加上说明性的文字
	Bgcolor	表格的行的背景颜色
	Background	表格的行的背景图像
	Bordercolor	表格的行的边框颜色
	Bordercolorlight	表格的行的亮边框颜色



10

使用表格排版网页



(续表)

属性及说明	属性	说明
	Bordercolordark	表格的行的暗边框颜色
	Valign	表格的行的垂直对齐方式

Td标签

<tr>标签内的<td>标签会在一行中创建单元格及其内容。数据通常会默认左对齐，和表格行标签（<tr>）中的其他标签一样，单元格标签支持很丰富的样式和内容对齐属性，这样可以将它们应用于单个数据格；这些属性会覆盖掉原来当前行的默认值。一些特殊的属性可以控制单元格在表格中跨越的行或列的数值；<td>标签也支持那些通常的表格属性。

<td>标签中的内容可以是放置到文档主体中的任何元素，如文字、图像、表单等，甚至另一个表格，浏览器会自动创建一个足够空间的表格，用来显示所有单元格的内容。

基本语法	<td>	
	单元格内容	
	</td>	
功能	定义表格的单元格	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Idalign	对齐方式
	Class	用一个名称来标记表格的单元格，该标记名称指向一个定义的类，而该类是在文档级声明的或者在外部定义的样式表
	Id	为表格的单元格创建一个标记，应用超链接时可以用这个标来明确地引用该表格的单元格，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	创建表格的单元格内容的内联样式
	Title	给表格的单元格加上说明性的文字
	Bgcolor	表格的单元格的背景颜色
	background	表格的单元格的背景图像
	Bordercolor	表格的单元格的边框颜色
	Bordercolorlight	表格的单元格的亮边框颜色
	Bordercolordark	表格的单元格的暗边框颜色
	Valign	表格的单元格的垂直对齐方式
	Width	表格的单元格的宽度
	Height	表格的单元格的高度
	Abbr	单元格的简称或缩写
	Axis	用逗号分割的目录名列表
	Rowspan	单元格的跨行数
	Colspan	单元格的跨列数
	Headers	标题单元格的列表
	Nowrap	禁止换行
	Scope	指定一组单元格提供标题信息

原始文件	Sample\Ch10\1\1.html
最终文件	Sample\Ch10\1\1-end.html
学习要点	使用<table>、<tr>、<td>标签

1. 范例代码

```

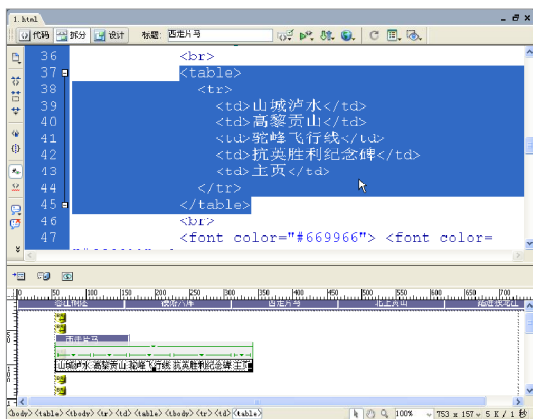
<html>
<head>
...
</head>
<body>
...
<table>
<tr>
<td>山城泸水</td>
<td>高黎贡山</td>
<td>驼峰飞行线</td>
<td>抗英胜利纪念碑</td>
<td>主页</td>
</tr>
</table>
...
</body>
</html>

```

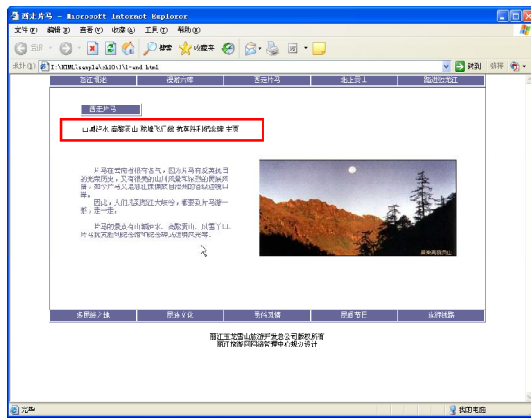
2. 范例解释

- 01 声明表格开始。
- 02 声明行开始。
- 03 声明第 1 个单元格，内容为“山城泸水”。
- 04 声明第 2 个单元格，内容为“高黎贡山”。
- 05 声明第 3 个单元格，内容为“驼峰飞行线”。
- 06 声明第 4 个单元格，内容为“抗英胜利纪念碑”。
- 07 声明第 5 个单元格，内容为“主页”。
- 08 声明行结束。
- 09 声明表格结束。

3. 代码和设计



4. 显示结果



Th 标签

<tr>标签内的<td>标签会在一行中创建表头。表头用粗体样式标记，另外内容的默认对齐方式也可能和数据的对齐方式不同。数据通常会默认左对齐，可是文本表头会在中间对齐。和表格行标签(<tr>)中的其他标签一样，表头标签支持很丰富的样式和内容对齐属性，这



10

使用表格排版网页



样可以将它们用于表头单元格。这些属性会覆盖掉原来当前行的默认值。和<td>标签一样，<th>标签中的内容可以是放置到文档主体中的任何元素，如文字、图像、表单等，甚至另一个表格。

基本语法	<th>	
	表头内容	
	</th>	
功能	定义表格的表头	
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	用一个名称来标记表格的表头，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	为表格的表头创建一个标记，应用超链接时可以用这个标记来明确地引用该表头，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	创建表格的表头内容的内联样式
	Title	给表格的表头加上说明性的文字
	Bgcolor	表格的表头的背景颜色
	Background	表格的表头的背景图像
	Bordercolor	表格的表头的边框颜色
	Bordercolorlight	表格的表头的亮边框颜色
	Bordercolordark	表格的表头的暗边框颜色
	Valign	表格的表头的垂直对齐方式
	Width	表格的表头的宽度
	Height	表格的表头的高度
	Abbr	表头的简称或缩写
	Axis	逗号分割的目录名列表
	Rowspan	表头的跨行数
	Colspan	表头的跨列数
	Headers	标题单元格的列表
	Nowrap	禁止换行
	Scope	指定一组表头提供标题信息

原始文件	Sample\Ch10\1\2. html
最终文件	Sample\Ch10\1\2-end. html
学习要点	使用<th>标签

1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<table>
<tr>
<th>山城泸水</th>
<th>高黎贡山</th>
<th>驼峰飞行线</th>
<th>抗英胜利纪念碑</th>
<th>主页</th>
</tr>
<tr>
<td>怒江概述</td>
<td>漫游六库</td>
<td>西走片马</td>
<td>北上贡山</td>
<td>踏进独龙江</td>
</tr>
</table>
...
</body>
</html>

```

2. 范例解释

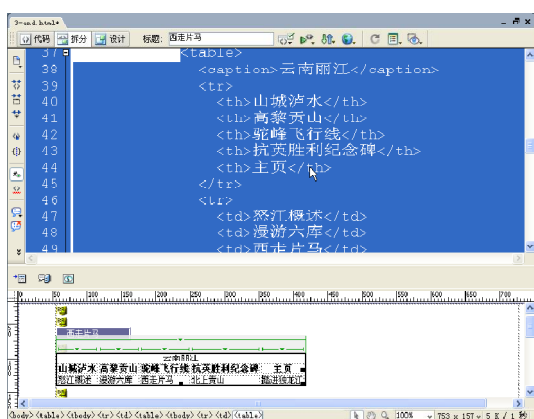
- 01 声明表格开始。
- 02 声明第1行开始。
- 03 声明第1个表头，内容为“山城泸水”。
- 04 声明第2个表头，内容为“高黎贡山”。
- 05 声明第3个表头，内容为“驼峰飞行线”。
- 06 声明第4个表头，内容为“抗英胜利纪念碑”。
- 07 声明第5个表头，内容为“主页”。
- 08 声明第1行结束。
- 09 声明表格的第2行及其单元格和内容。
- 10 声明表格结束。



10

使用表格排版网页

3. 代码和设计



4. 显示结果



Caption 标签

通常情况下，表格需要一个标题来说明它的内容。一般浏览器都提供了一个表格标题标签。设计者通常在<table>标签后立即加入<caption>标签及其内容，但是<caption>标签也可以放在表格和行标签之间的任何地方。标题可以包括任何主体内容，这一点和表格中的单元格一样。



基本语法	<caption>	
	表格标题内容	
	</caption>	
功能	定义表格的标题	
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	用一个名称来标记表格的标题，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	为表格的标题创建一个标记，应用超链接时可以用这个标记来明确地引用该表格的标题，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	创建表格的标题内容的内联样式
	Title	给表格的标题加上说明性的文字
	Valign	表格的标题的垂直对齐方式

原始文件	Sample\Ch10\1\3. html
最终文件	Sample\Ch10\1\3-end. html
学习要点	使用<caption>标签

1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<table>
<caption>云南丽江</caption>
<tr>
<th>山城泸水</th>
<th>高黎贡山</th>
<th>驼峰飞行线</th>
<th>抗英胜利纪念碑</th>
<th>主页</th>
</tr>
<tr>
<td>怒江概述</td>
<td>漫游六库</td>
<td>西走片马</td>
<td>北上贡山</td>
<td>踏进独龙江</td>
</tr>
```

2. 范例解释

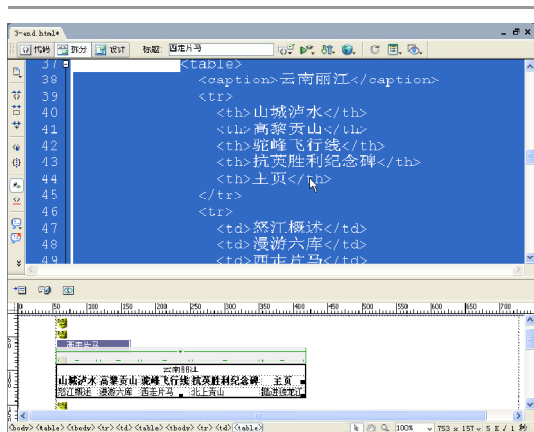
- 01 声明表格开始。
- 02 声明表格标题为“云南丽江”。
- 03 声明第1行开始。
- 04 声明第1个表头，内容为“山城泸水”。
- 05 声明第2个表头，内容为“高黎贡山”。
- 06 声明第3个表头，内容为“驼峰飞行线”。
- 07 声明第4个表头，内容为“抗英胜利纪念碑”。
- 08 声明第4个表头，内容为“主页”。
- 09 声明第1行结束。
- 10 声明表格的第2行及其单元格和内容。

```
</table>
```

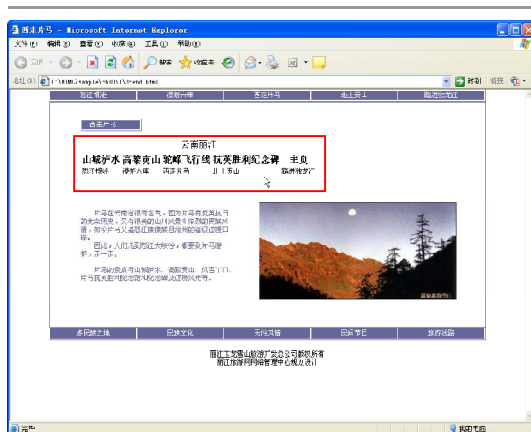
11 声明表格结束。

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

3. 代码和设计



4. 显示结果



2

表格的属性

表格的属性非常丰富，如表格本身、表头、行、单元格乃至表格标题等都具有各自相同或不同的属性。

Table 标签的属性

Border属性

默认情况下，表格的边框为 0；当然也可以为表格设置各种边框线。

原始文件	Sample\Ch10\2\1. html
最终文件	Sample\Ch10\2\1-end. html
学习要点	使用<table>标签的border属性



10

使用表格排版网页



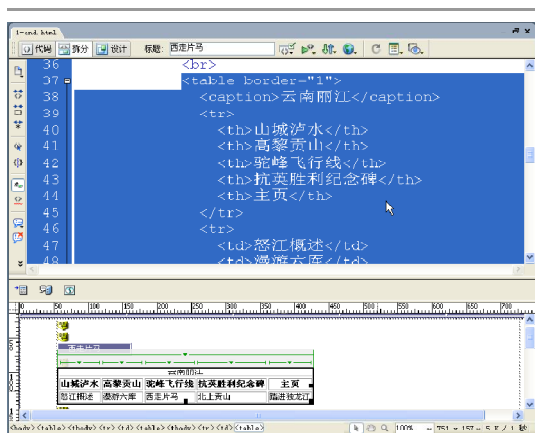
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<table border="1">
<caption>云南丽江</caption>
<tr>
<th>山城泸水</th>
<th>高黎贡山</th>
<th>驼峰飞行线</th>
<th>抗英胜利纪念碑</th>
<th>主页</th>
</tr>
...
</table>
...
</body>
</html>
```

2. 范例解释

- 01 定义表格的边框为1。
- 02 声明表格标题为“云南丽江”。
- 03 声明第1个表头，内容为“高黎贡山”。
- 04 声明第2个表头，内容为“驼峰飞行线”。
- 05 声明第3个表头，内容为“抗英胜利纪念碑”。
- 06 声明第4个表头，内容为“抗英胜利纪念碑”。
- 07 声明第5个表头，内容为“主页”。

3. 代码和设计



4. 显示结果



Width和Height属性

默认情况下，表格的宽度和高度根据内容自动调整，我们也可以手动设置表格的宽度和高度。通过Width属性定义表格的宽度，Height属性定义表格的高度，单位为像素或百分比。如果是百分比，则可分为两种情况：如果不是嵌套表格，那么百分比是相对于浏览器窗口而言；如果是嵌套表格，则百分比相对的是嵌套表格所在的单元格宽度。

原始文件	Sample\Ch10\2\2.html
最终文件	Sample\Ch10\2\2-end.html
学习要点	使用<table>标签的Width和Height属性

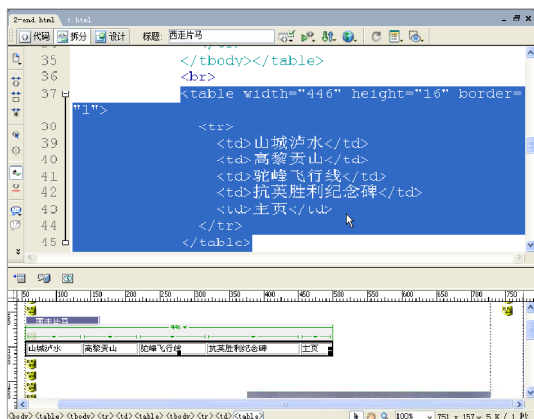
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<table width="446" height="16"
border="1">
<tr>
<td>山城泸水</td>
<td>高黎贡山</td>
<td>驼峰飞行线</td>
<td>抗英胜利纪念碑</td>
<td>主页</td>
</tr>
</table>
...
</body>
</html>
```

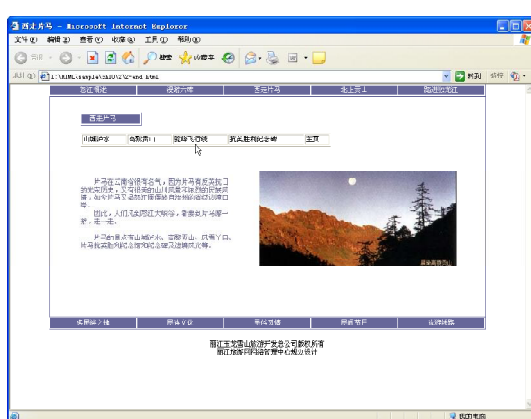
2. 范例解释

- 01 定义边框为1像素的表格的宽度为446像素，高度为16像素。
- 02 声明第1个单元格，内容为“山城泸水”。
- 03 声明第2个单元格，内容为“高黎贡山”。
- 04 声明第3个单元格，内容为“驼峰飞行线”。
- 05 声明第4个单元格，内容为“抗英胜利纪念碑”。
- 06 声明第5个单元格，内容为“主页”。

3. 代码和设计



4. 显示结果



Bordercolor属性

为了美化表格，可以为表格设定不同的边框颜色。定义颜色的时候，可以使用英文颜色名称或十六进制颜色值。



10

使用表格排版网页



原始文件	Sample\Ch10\2\3.html
最终文件	Sample\Ch10\2\3-end.html
学习要点	使用<table>标签的bordercolor属性

1. 范例代码

```

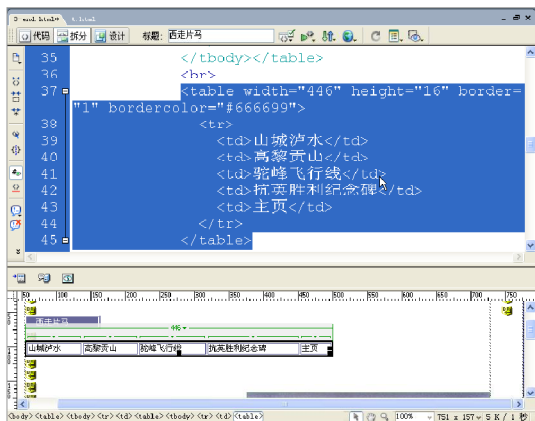
<html>
<head>
...
</head>
<body>
...
<table width="446" height="16"
border="1" bordercolor="#666699">
<tr>
<td>山城泸水</td>
<td>高黎贡山</td>
<td>驼峰飞行线</td>
<td>抗英胜利纪念碑</td>
<td>主页</td>
</tr>
</table>
...
</body>
</html>

```

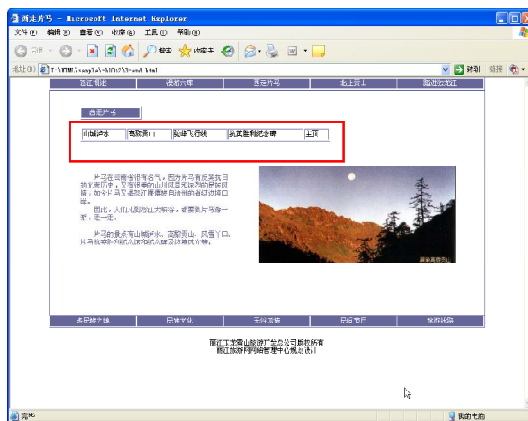
2. 范例解释

- 01 定义边框为1像素、宽度为446像素，高度为16像素的表格，边框色为#666699的蓝色。
- 02 声明第1个单元格，内容为“山城泸水”。
- 03 声明第2个单元格，内容为“高黎贡山”。
- 04 声明第3个单元格，内容为“驼峰飞行线”。
- 05 声明第4个单元格，内容为“抗英胜利纪念碑”。
- 06 声明第5个单元格，内容为“主页”。

3. 代码和设计



4. 显示结果



Bordercolorlight、Bordercolordark属性

在表格中，可以单独定义单元格中左上边框和右下边框的颜色，这两个位置的边框色我们称之为亮边框色；也可以单独定义单元格中右下边框和左上边框的颜色，这两个位置的边框色我们称之为暗边框色。定义颜色的时候，可以使用英文颜色名称或十六进制颜色值。

原始文件	Sample\Ch10\2\4.html
最终文件	Sample\Ch10\2\4-end.html
学习要点	使用<table>标签的bordercolorlight和bordercolordark属性

1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<table width="446" height="16"
border="1"
bordercolordark="#003399"
bordercolorlight="#FFCCCC">
<tr>
<td>山城泸水</td>
<td>高黎贡山</td>
<td>驼峰飞行线</td>
<td>抗英胜利纪念碑</td>
<td>主页</td>
</tr>
</table>
...
</body>
</html>
</html>

```

2. 范例解释

01 定义边框为1像素、宽度为446像素，高度为16像素的表格，亮边框色为#FFCCCC的红色，暗边框色为#003399的蓝色。

02 声明第1个单元格，内容为“山城泸水”。

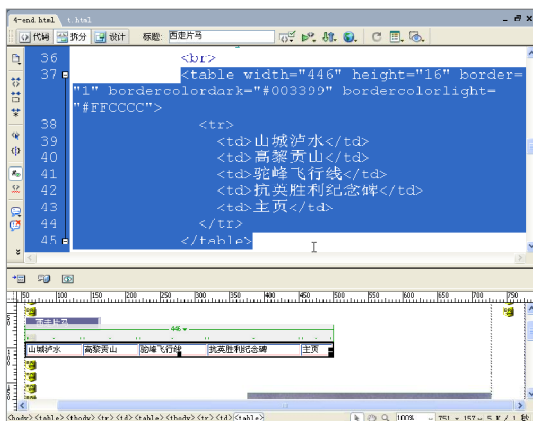
03 声明第2个单元格，内容为“高黎贡山”。

04 声明第3个单元格，内容为“驼峰飞行线”。

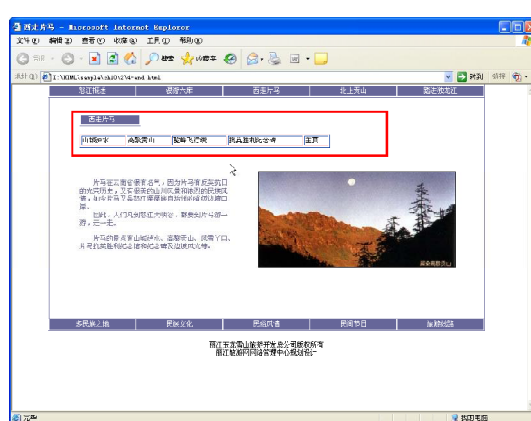
05 声明第4个单元格，内容为“抗英胜利纪念碑”。

06 声明第5个单元格，内容为“主页”。

3. 代码和设计



4. 显示结果



Bgcolor属性

通过Bgcolor属性，可以设定表格的背景颜色。定义颜色的时候，可以使用英文颜色名称或十六进制颜色值。



10

使用表格排版网页



原始文件	Sample\Ch10\2\5.html
最终文件	Sample\Ch10\2\5-end.html
学习要点	使用<table>标签的bgcolor属性

1. 范例代码

```

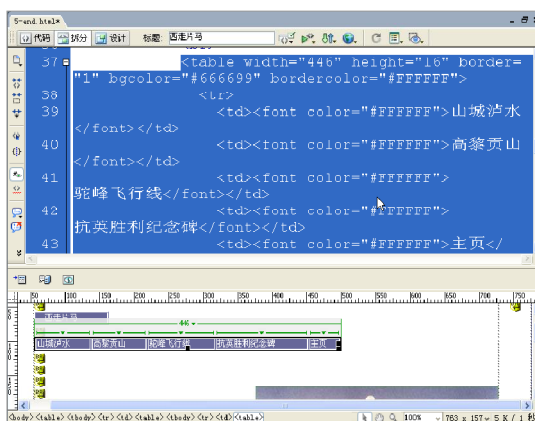
<html>
<head>
...
</head>
<body>
...
<table width="446" height="16"
border="1" bgcolor="#666699"
bordercolor="#FFFFFF">
<tr>
<td><font color="#FFFFFF">山城泸水
</font></td>
<td><font color="#FFFFFF">高黎贡山
</font></td>
<td><font color="#FFFFFF">驼峰飞行线
</font></td>
<td><font color="#FFFFFF">抗英胜利纪念碑
</font></td>
<td><font color="#FFFFFF">主页<font>
</td>
</tr>
</table>
...
</body>
</html>

```

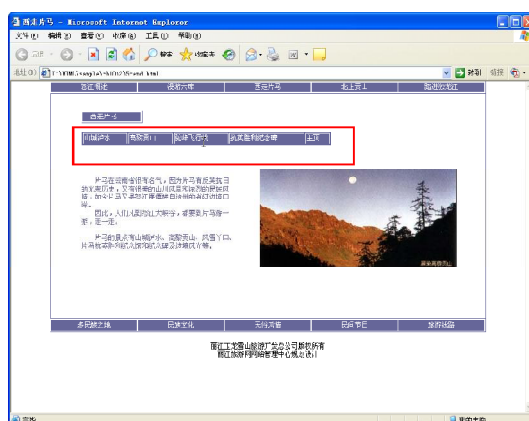
2. 范例解释

- 01 定义边框为1像素、宽度为446像素，高度为16像素、边框色为白色的表格，背景颜色为#666699的蓝色。
- 02 声明第1个单元格，内容为“山城泸水”。
- 03 声明第2个单元格，内容为“高黎贡山”。
- 04 声明第3个单元格，内容为“驼峰飞行线”。
- 05 声明第4个单元格，内容为“抗英胜利纪念碑”。
- 06 声明第5个单元格，内容为“主页”。

3. 代码和设计



4. 显示结果



Background属性

除了背景颜色之外，我们还可以为表格设置背景图像，可以使用任何 GIF 或者 JPEG 图片文件。

原始文件	Sample\Ch10\2\6. html
最终文件	Sample\Ch10\2\6-end. html
学习要点	使用<table>标签的background属性

1. 范例代码

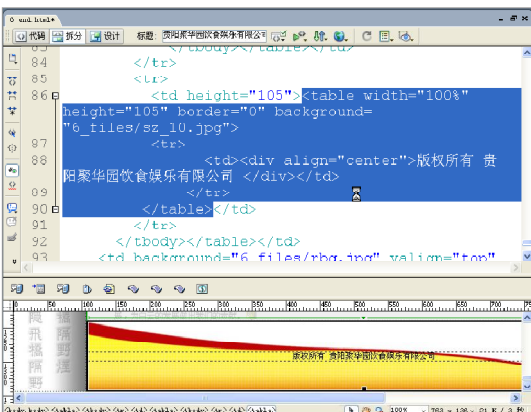
```
<html>
<head>
...
</head>
<body>
...
<table width="100%" height="105"
  border="0" background="6_files/
  sz_10.jpg">
<tr>
<td>
<div align="center">版权所有 贵阳
聚华园饮食娱乐有限公司 </div>
</td>
</tr>
</table>
...
</body>
</html>
```

2. 范例解释

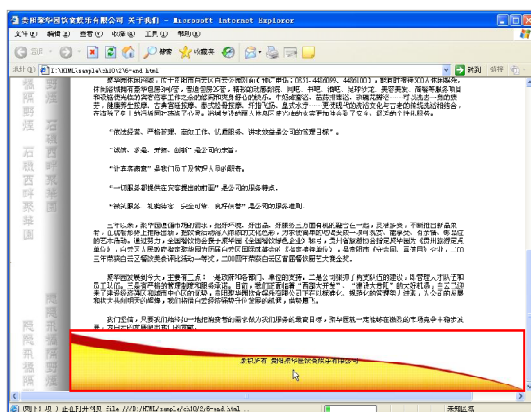
01 定义边框为0像素、宽度为100%，高度为105的表格的背景图像为6_files文件夹下的sz_10.jpg。

02 声明第1个单元格，内容为“版权所有 贵阳聚华园饮食娱乐有限公司”，文字分块居中对齐。

3. 代码和设计



4. 显示结果



10

使用表格排版网页



Cellspacing属性

Cellspacing 属性控制表格中相邻单元格的间距，以及单元格外边沿和表格边沿之间的间距。浏览器通常在单元格之间和表格外边沿之间留下两个像素的间距。如果

原始文件	Sample\Ch10\2\7.html
最终文件	Sample\Ch10\2\7-end.html
学习要点	使用<table>标签的cellspacing属性

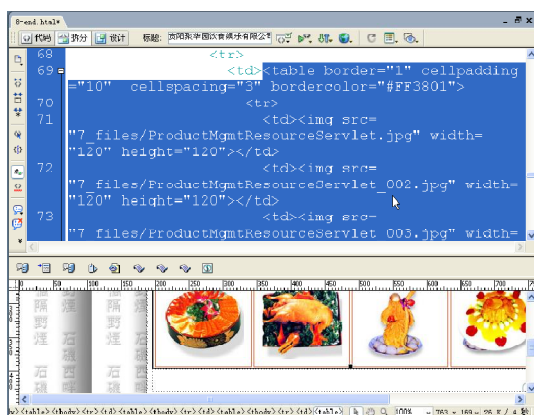
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<tableborder="1"  cellspacing="3"
bordercolor="#FF3801">
<tr>
<td></td>
<td><td>
<td></td>
<td>
</td>
</tr>
</table>
...
</body>
</html>
```

2. 范例解释

- 01 定义边框为 1 像素、背景色为 #FF3801 的表格的单元格间距为 3 像素。
- 02 声明第 1 个单元格，内容为“7_files/ProductMgmtResourceServlet.jpg”图片。
- 03 声明第 2 个单元格，内容为“7_files/ProductMgmtResourceServlet_002.jpg”图片。
- 04 声明第 3 个单元格，内容为“7_files/ProductMgmtResourceServlet_003.jpg”图片。
- 05 声明第 4 个单元格，内容为“7_files/ProductMgmtResourceServlet_004.jpg”图片。

3. 代码和设计



4. 显示结果



Cellpadding属性

Cellpadding 属性控制单元格的边沿和它的内容之间的距离，它的默认值是1个像素。可以在表格标签中用Cellpadding=0来让表格中的单元格内容和它们相对的单元格边沿紧凑在一起；也可以将它的值设为大于1，来增加Cellpadding的距离。

原始文件	Sample\Ch10\2\8.html
最终文件	Sample\Ch10\2\8-end.html
学习要点	使用<table>标签的cellpadding属性

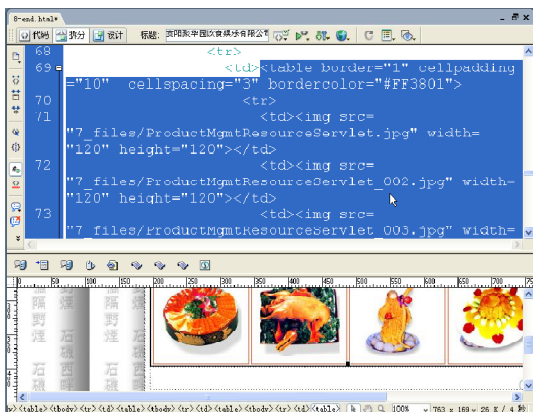
1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<table border="1"
cellpadding="10" cellspacing="3"
bordercolor="#FF3801">
<tr>
<td></td>
<td><td>
<td><td>
<td></td>
</tr>
</table>
...
</body>
</html>
```

05 声明第 4 个单元格，内容为“7_files/ProductMgmtResourceServlet_004.jpg”图片。

3. 代码和设计



4. 显示结果



Align属性

在水平方向上，可以设定表格的对齐方式，它有居左、居中、居右 3 种方式。Align 的属性值如下表所示：

Align属性及说明	Align属性值	说明
	Left	居左
	Right	居右
	Center	居中

原始文件	Sample\Ch10\2\9.html
最终文件	Sample\Ch10\2\9-end.html
学习要点	使用<table>标签的align属性

1. 范例代码

```

<html>
<head>
...
</head>
<body>
...
<table border="1" align="center"
cellpadding="10" cellspacing="3"
bordercolor="#FF3801">
<tr>
<td></td>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
</table>
...
</body>
</html>

```

2. 范例解释

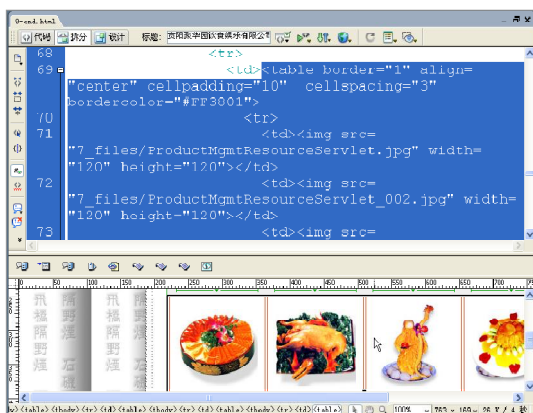
- 01 定义边框为1像素、背景色为#FF3801、单元格间距为3像素、单元格边距为10像素的表格水平居中。
- 02 声明第1个单元格，内容为“7_files/ProductMgmtResourceServlet.jpg”图片。
- 03 声明第2个单元格，内容为“7_files/ProductMgmtResourceServlet_002.jpg”图片。
- 04 声明第3个单元格，内容为“7_files/ProductMgmtResourceServlet_003.jpg”图片。
- 05 声明第4个单元格，内容为“7_files/ProductMgmtResourceServlet_004.jpg”图片。



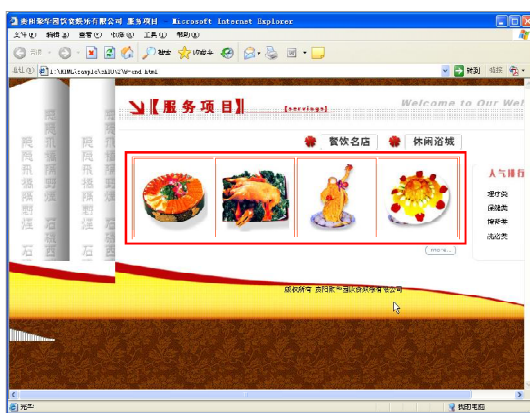
10

使用表格排版网页

3. 代码和设计



4. 显示结果





Frame属性

标准的Frame属性可以为表格周围的行修改边框效果，它的默认值是Box，它告诉浏览器在表格周围画上全部4条线。Border值和box的作用一样；Void值会将所有Frame的4条线删除。Hsides值会在表格的顶部和底部（水平方向）画上边框；vsides会在表格的左边和右边（垂直方向）画上边框。Frame值为above、below、lhs和rhs时，浏览器会分别在表格的顶部、底部、左边和右边画上不同的边框线。Frame的属性值如下表所示：

Frame属性值及说明	Frame属性值	说明
	Above	显示上边框
	Below	显示下边框
	Border	显示上下左右边框
	Box	显示上下左右边框
	Hsides	显示上下边框
	Vsides	显示左右边框
	Lhs	显示右边框
	Rhs	不显示边框
	Void	显示左右边框

原始文件	Sample\Ch10\2\10.html
最终文件	Sample\Ch10\2\10-end.html
学习要点	使用<table>标签的frame属性

1. 范例代码

2. 范例解释

```
<html>
<head>
-
</head>
<body>
-
<table border="1" align="center"
cellpadding="10" cellspacing="3"
bordercolor="#FF3801"
frame="hsides">
<tr>
<td></td>
<td></td>
```

01 定义边框为1像素、背景色为#FF3801、单元格间距为3像素、单元格边距为10像素、水平居中的表格仅显示上下边框。

02 声明第1个单元格，内容为“7_files/ProductMgmtResourceServlet.jpg”图片。

03 声明第2个单元格，内容为“7_files/ProductMgmtResourceServlet_002.jpg”图片。


```

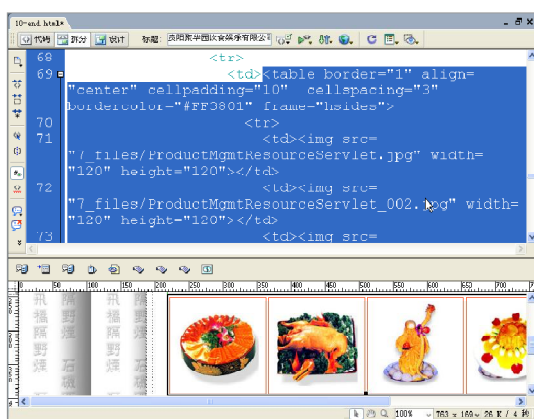
<td><td>
<td><td>
</tr>
</table>
...
</body>
</html>

```

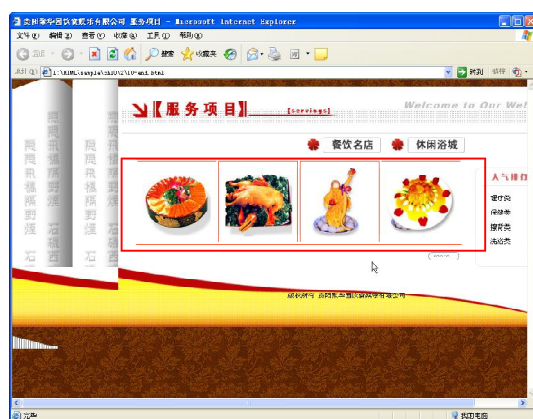
04 声明第 3 个单元格，内容为“7_files/ProductMgmtResourceServlet_003.jpg”图片。

05 声明第 4 个单元格，内容为“7_files/ProductMgmtResourceServlet_004.jpg”图片。

3. 代码和设计



4. 显示结果



Rules属性

Rules 属性可以用来控制表格内部单元格边框的宽度。它的默认值为 all，是在所有单元格周围画边框。如果指定它的值为 groups，可以在行和列组间设置更宽的边框；设置值为 rows 或 cols，可以只在每行或每列中间画边框；而指定值为 none 可以取消表格中每个单元格周围的边框。Rules 的属性值如下表所示。

Rules 属性值及说明	Rules 属性值	说明
	All	显示所有的内部边框
	Cols	仅显示列边框
	Groups	显示介于行列间的边框
	None	不显示内部边框
	Rows	仅显示行边框

原始文件	Sample\Ch10\2\11.html
最终文件	Sample\Ch10\2\11-end.html
学习要点	使用<table>标签的rules属性



10

使用表格排版网页



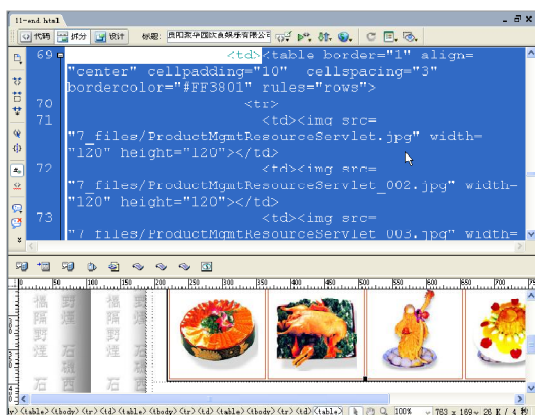
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<table border="1" align="center"
cellpadding="10" cellspacing="3"
bordercolor="#FF3801"
rules="rows">
<tr>
<td></td>
<td><td>
<td><td>
<td><td>
</tr>
</table>
...
</body>
</html>
```

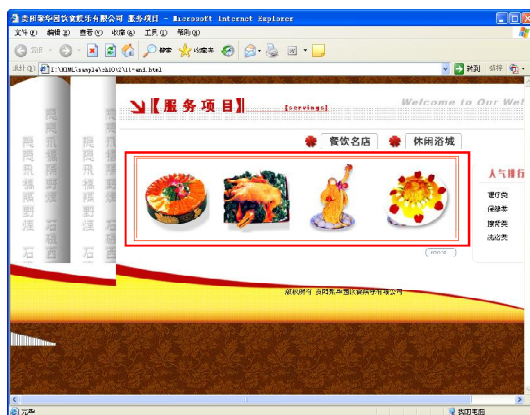
2. 范例解释

- 01 定义边框为1像素、背景色为#FF3801、单元格间距为3像素、单元格边距为10像素、水平居中的表格仅显示行边框。
- 02 声明第1个单元格，内容为“7_files/ProductMgmtResourceServlet.jpg”图片。
- 03 声明第2个单元格，内容为“7_files/ProductMgmtResourceServlet_002.jpg”图片。
- 04 声明第3个单元格，内容为“7_files/ProductMgmtResourceServlet_003.jpg”图片。
- 05 声明第4个单元格，内容为“7_files/ProductMgmtResourceServlet_004.jpg”图片。

3. 代码和设计



4. 显示结果



Tr 标签的 Valign 属性

在垂直方向上，设计者可以设定行的对齐方式，分别有居上、居中、居下、基线对齐 4 种方式。使用 Valign 属性可以改变行内容的垂直默认对齐方式。通常情况下，浏览器会垂直居中地生成单元格内容。在<tr>标签中，Valign 属性的 top、bottom 或 baseline 值，可以告诉浏览器如何放置表格行的内容，是靠单元格的顶部、底部，还是对准本行中其他单元格中首行文本的基线，具体属性值如下表所示。

Valign 属性值及说明	Valign 属性值	说明
	Top	居顶
	Middle	居中
	Bottom	居底
	Baseline	基线对齐

原始文件	Sample\Ch10\2\12.html
最终文件	Sample\Ch10\2\12-end.html
学习要点	使用<tr>标签的valign属性



10

使用表格排版网页

1. 范例代码

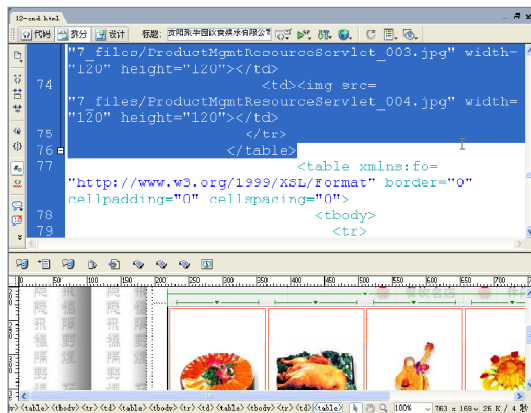
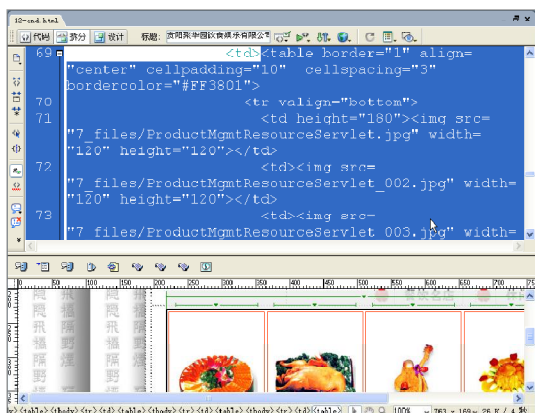
```
<html>
<head>
...
</head>
<body>
...
<table border="1" align="center"
cellpadding="10" cellspacing="3"
bordercolor="#FF3801">
<tr valign="bottom">
<td></td>
<td><td>
<td><td>
<td><imgsrc="7_files/
ProductMgmtResourceServlet_004.
jpg" width="120"height="120"><td>
</tr>
</table>
...
</body>
</html>
```

2. 范例解释

- 01 定义行的对齐方式为居底。
- 02 声明第 1 个单元格，内容为“7_files/ProductMgmtResourceServlet.jpg”图片。
- 03 声明第 2 个单元格，内容为“7_files/ProductMgmtResourceServlet_002.jpg”图片。
- 04 声明第 3 个单元格，内容为“7_files/ProductMgmtResourceServlet_003.jpg”图片。
- 05 声明第 4 个单元格，内容为“7_files/ProductMgmtResourceServlet_004.jpg”图片。



3. 代码和设计



4. 显示结果



Td和Th标签的属性

Valign属性

在垂直方向上，设计者可以设定单元格或表头的对齐方式，分别有居上、居中、居下、基线对齐 4 种方式。使用 Valign 属性可以改变单元格或表头内容的垂直默认对齐方式。通常情况下，浏览器会垂直居中地生成单元格或表头的内容。同<tr>标签类似在<td>或<th>标签中，包括值为 Valign 属性的 top、bottom 或 baseline，可以告诉浏览器如何放置表格单元格或表头的内容，是靠单元格或表头的顶部、底部，还是对准本单元格或表头中首行文本的基线具体属性值如下表所示：

Valign属性值及说明	Valign属性值	说明
	Top	居顶
	Middle	居中
	Bottom	居底
	Baseline	基线对齐

原始文件	Sample\Ch10\2\13.html
最终文件	Sample\Ch10\2\13-end.html
学习要点	使用<td>或<th>标签的valign属性

1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<table border="1" align="center"
cellpadding="10" cellspacing="3"
bordercolor="#FF3801" height="180"
><tr><td valign="bottom">
<imgsrc="7_files/
ProductMgmtResourceServlet.jpg"
width="120" height="120"></td>
<td valign="bottom"><td>
<td valign="bottom"><td>
<td valign="bottom"><td>
</tr>
</table>
...
</body>
</html>
```

2. 范例解释

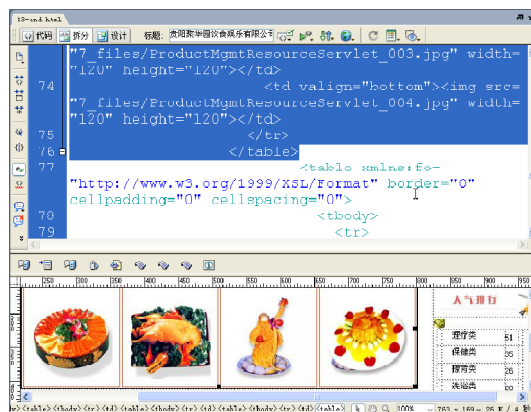
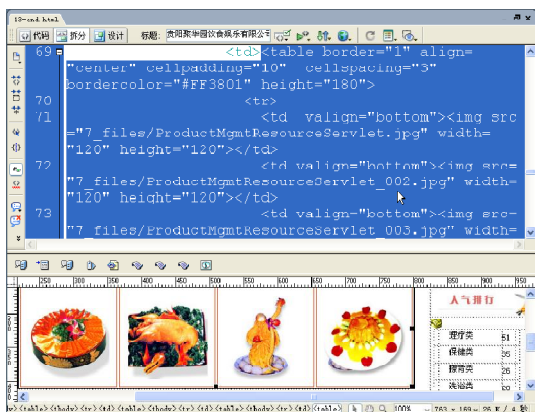
定义4个单元格的垂直对齐方式为居底，这样单元格中的图像将位于单元格底部。



10

使用表格排版网页

3. 代码和设计





4. 显示结果



Col span属性

在表格表头或数据单元格中使用Col span属性, 可以将一行中的一个表格单元格扩展为两列或更多列。

原始文件	Sample\Ch10\2\14. html
最终文件	Sample\Ch10\2\14-end. html
学习要点	使用<td>或<th>标签的col span属性

1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<table border="1" align="center"
cellpadding="10" cellspacing="3"
bordercolor="#FF3801"
height="180">
<tr>
<td colspan="4"><div align="center">
美食</div></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
```

2. 范例解释

- 01 定义第1行的“美食”所在的单元格跨4列, 并居中对齐。
- 02 声明第2行第1个单元格, 内容为“7_files/ProductMgmtResourceServlet.jpg”图片。
- 03 声明第2行第2个单元格, 内容为“7_files/ProductMgmtResourceServlet_002.jpg”图片。
- 04 声明第2行第3个单元格, 内容为“7_files/ProductMgmtResourceServlet_003.jpg”图片。

```

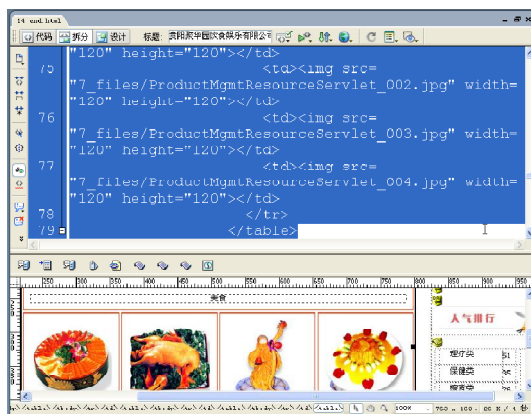
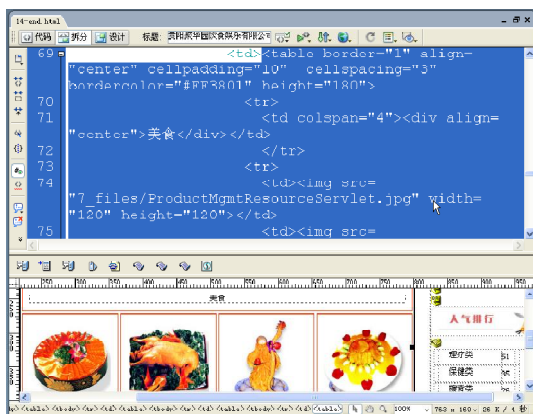
<td><imgsrc="7_files/Product
MgmtResourceServlet_004.jpg"
width="120" height="120"></td>
</tr>
</table>

...
</body>
</html>

```

05 声明第2行第4个单元格，内容为“7_files/ProductMgmtResourceServlet_004.jpg”图片。

3. 代码和设计



4. 显示结果



Rowspan属性

如同使用 Col span属性将一个表格单元格扩展为几列一样，使用 Rowspan属性也可以将一个单元格扩展到表格中的上下几行。

原始文件	Sample\Ch10\2\15. html
最终文件	Sample\Ch10\2\15-end. html
学习要点	使用<td>或<th>标签的rowspan属性



10

使用表格排版网页



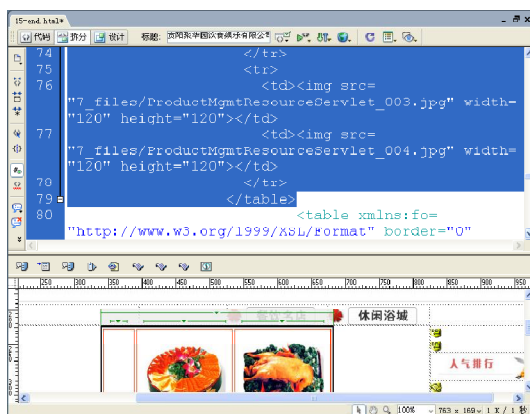
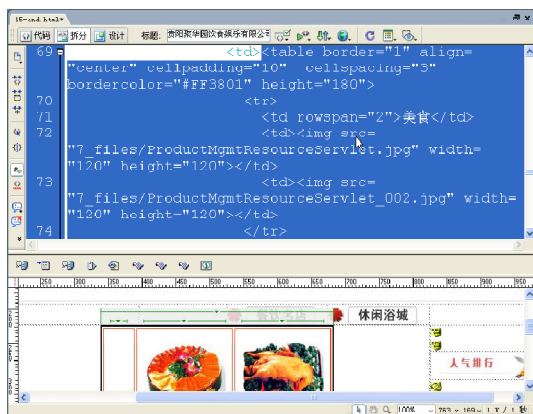
1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<table border="1" align="center"
cellpadding="10" cellspacing="3"
bordercolor="#FF3801"
height="180">
<tr>
<td rowspan="2">美食</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</table>
...
</body>
</html>
```

2. 范例解释

定义第1列的单元格跨两行，这样“美食”文字占据了两个单元格的位置，后面的7_files/ProductMgmtResourceServlet.jpg图片和7_files/ProductMgmtResourceServlet_002.jpg图片占据第1行单元格的位置；7_files/ProductMgmtResourceServlet_003.jpg图片和7_files/ProductMgmtResourceServlet_004.jpg图片占据第2行单元格的位置。

3. 代码和设计



4. 显示结果



Caption 标签的属性

Align属性

在水平方向上，可以通过 `Align` 设定表格标题的对齐方式，分别有居左、居中、居右这3种。`Align` 属性值如下表所示。

Align属性值及说明	Align属性值	说明
	Left	居左
	Center	居中
	Right	居右

原始文件	Sample\Ch10\2\16.html
最终文件	Sample\Ch10\2\16-end.html
学习要点	使用<caption>标签的align属性

1. 范例代码

```
<html>
<head>
.....
</head>
<body>
.....
<table border="1" align="center"
cellpadding="10" cellspacing="3"
bordercolor="#FF3801"
height="180">
<caption align="right">美食
</caption>
```

2. 范例解释

01 定义表格的标题为右对齐。



10

使用表格排版网页



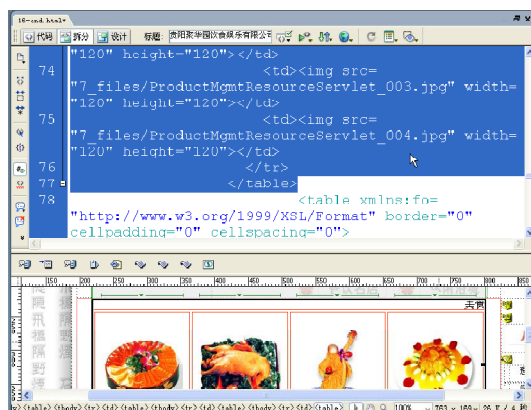
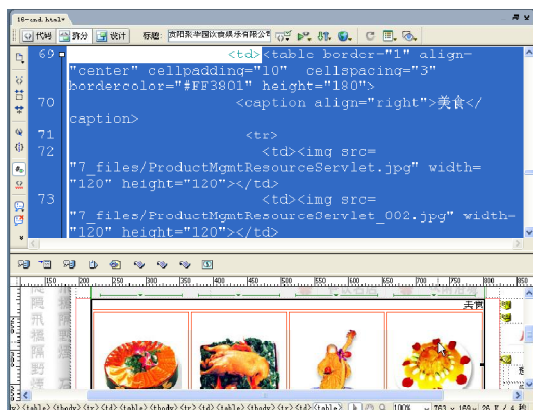
```
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</table>
...
</body>
</html>
```

02 声明第2行第1个单元格，内容为“7_files/ProductMgmtResourceServlet.jpg”图片。

03 声明第2行第2个单元格，内容为“7_files/ProductMgmtResourceServlet_002.jpg”图片。

04 声明第2行第3个单元格，内容为“7_files/ProductMgmtResourceServlet_003.jpg”图片。

05 行声明第2行第4个单元格，内容为“7_files/ProductMgmtResourceServlet_004.jpg”图片。



4. 显示结果



Valign属性

在垂直方向上，可以通过设置 Valign 属性来设定表格标题的对齐方式，分别有居上、居下两种方式。Valign 的属性值如下表所示。

valign属性值及说明	Valign属性值		说明
	Top		居上
	Bottom		居下

原始文件	Sample\Ch10\2\17.html
最终文件	Sample\Ch10\2\17-end.html
学习要点	使用<caption>标签的valign属性

1. 范例代码

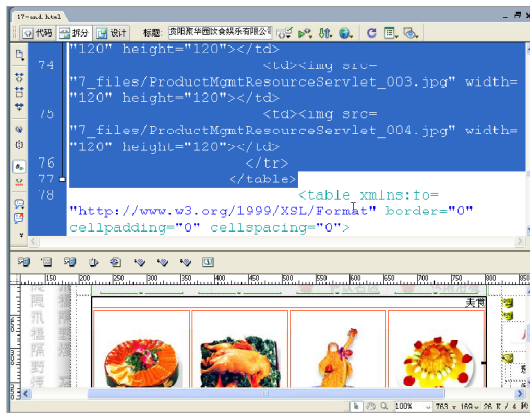
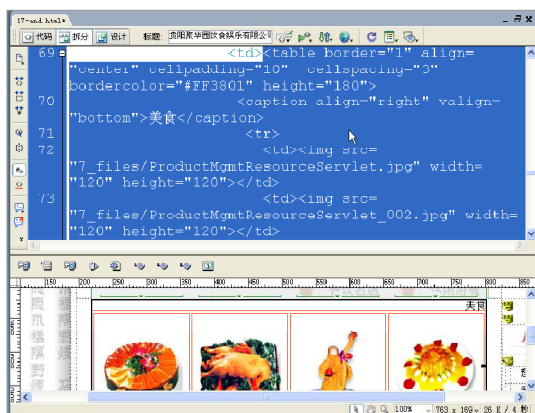
```
<html>
<head>
...
</head>
<body>
...
<table border="1" align="center"
cellpadding="10" cellspacing="3"
bordercolor="#FF3801"
height="180">
<caption align="right"
valign="bottom">美食</caption>
<tr>
<td></td>
<td></td>
<td><td>
<td></td>
</tr>
</table>
...
</body>
</html>
```

2. 范例解释

- 01 定义表格右对齐的标题为居底。
- 02 声明第 2 行第 1 个单元格，内容为 “7_files/ProductMgmtResourceServlet.jpg” 图片。
- 03 声明 2 行第 2 个单元格，内容为 “7_files/ProductMgmtResourceServlet_002.jpg” 图片。
- 04 声明第 2 行第 3 个单元格，内容为 “7_files/ProductMgmtResourceServlet_003.jpg” 图片。
- 05 声明第 2 行第 4 个单元格，内容为 “7_files/ProductMgmtResourceServlet_004.jpg” 图片。



3. 代码和设计



4. 显示结果



3

高级表格标签

为了能在源代码中清楚地区分表格结构，HTML 语言中规定了<thead>、<tbody>、<tfoot>这 3 个标记，它们分别对应于表格的表首、表主体和表尾。

Thead 标签

用<thead>标签可定义一组表头行。<thead>标签也许在<table>标签中表格的最前端就出现一次。在<thead>标签中可以放置一个或多个<tr>标签，用于定义表头中的行。当以多部分方式打印表格或显示表格时，浏览器会复制这些表头行，因此如果表格多于一页，在每个打印页上都会重复这些表头。对 HTML 来说，结束标签</thead>是可选的：由于<thead>标签只出现在当有行被指定为表格正文或表注的地方，因此当浏览器碰到<tbody>或<tfoot>标签或是当表格结束时，<thead>标签会自动关闭。

基本语法	<thead>	
	表头内容	
	</thead>（可省略）	
功能	定义表格的表头	
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	用一个名称来标记表格的表头，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	为表格的表头创建一个标记，应用超链接时可以用这个标记来明确地引用该表头，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	创建表格的表头内容的内联样式
	Title	给表格的表头加上说明性的文字
	Valign	表格表头的垂直对齐方式

<thead>标签的许多属性和前面标签的用法一样，可以取同样的值，同分别为每个<tr>条目指定一样，会影响<tr>包括的所有内容，例如Align属性接受left, right, center或justify作为它的值，相应地可以在所有表头行中控制文本的水平对齐方式。同样Valign属性接受的值有top、middle、baseline或bottom，指明了所有表头行中文本的垂直对齐方式。如果不指定一种对齐方式或样式，浏览器会将表头文本在单元格中水平和垂直对齐（相当于对每个单元格指定Align=center和Valign=middle），当然不同行和单元格或样式表可能会覆盖掉这些属性。

原始文件	Sample\Ch10\3\1.html
最终文件	Sample\Ch10\3\1-end.html
学习要点	使用<thead>标签

1. 范例代码

```
<html>
<head>
...
</head>
<body>
...
<table align="center" border="0"
cellpadding="2" cellspacing="2"
width="566">
<thead>
```

2. 范例解释

01 声明表头开始。



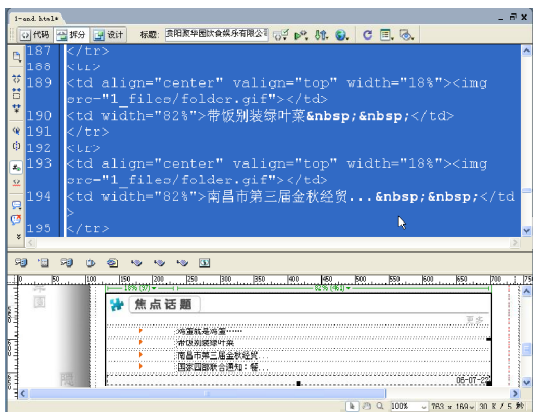


02 为表头内容。

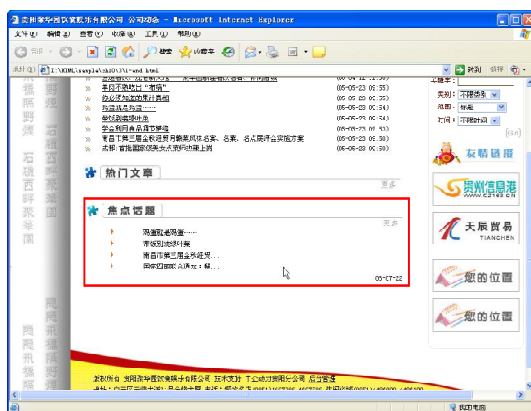
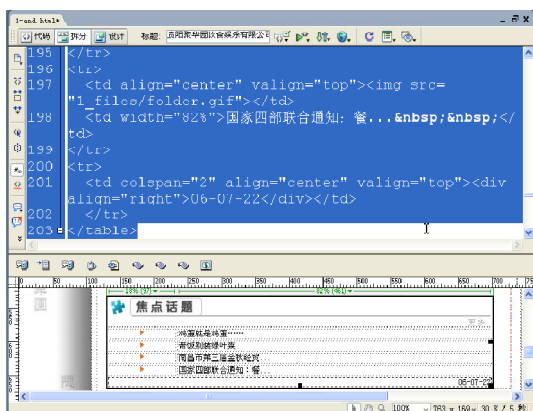
03 声明表头的结束。

[illegible]

3. 代码和设计



4. 显示结果

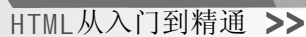


Tbody 标签

使用<tbody>标签可以将表格分成一个单独的部分，也可以将表格中的一行或几行合成一组。在<tbody>标签中，只有表格行可以用<tr>标签来定义；一旦定义，一个<tbody>标签就是表格中的一个独立的部分。在HTML中，由于<tbody>部分会在下一个<tbody>(或<tfoot>)标签处或是当表格结束时结束，因此</tbody>标签是可选的。

基本语法	<tbody>	
	表格主体内容	
	</tbody> (可省略)	
功能	定义表格的表主体	
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	用一个名称来标记表格的表主体，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	为表格主体创建一个标记，应用超链接时可以用这个标记来明确地引用该表主体，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	创建表格主体内容的内联样式
	Title	给表格主体加上说明性的文字
	Valign	表格主体的垂直对齐方式

原始文件	Sample\Ch10\3\2.html
最终文件	Sample\Ch10\3\2-end.html
学习要点	使用<tbody>标签

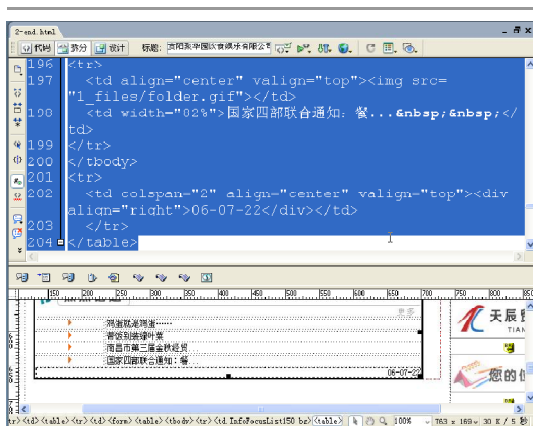
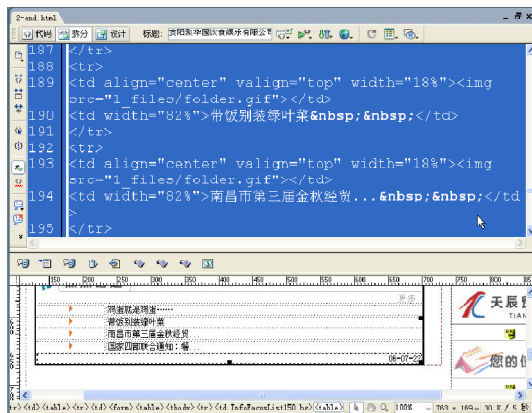
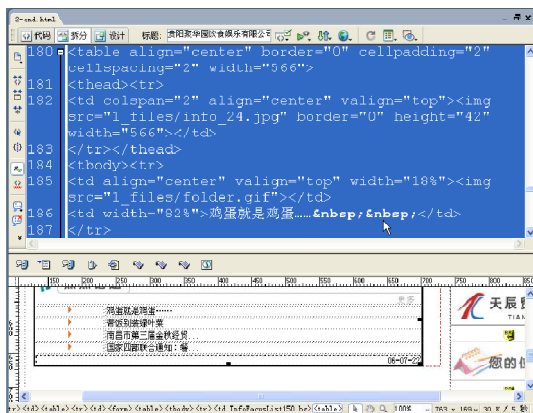


2. 范例解释

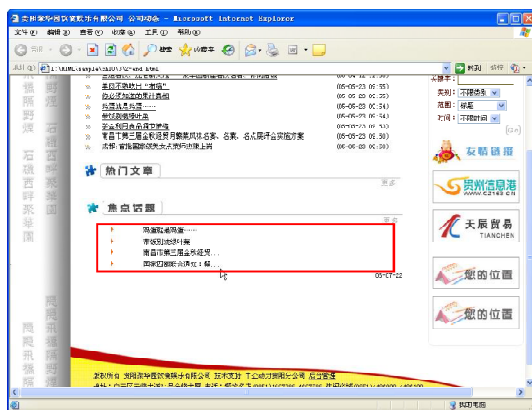
```
<tbody>
<tr>
<td align="center" valign="top"
width="18%"></td>
<td width="82%">鸡蛋就是鸡蛋.....
&nbsp; &nbsp; &nbsp; </td></tr><tr>
<td align="center" valign="top"
width="18%"></td>
<tdwidth="82%">带饭别装绿叶菜
&nbsp; &nbsp; &nbsp; </td></tr><tr>
<td align="center" valign="top"
width="18%"></td>
<td width="82%">南昌市第三届金秋经
贸... &nbsp; &nbsp; &nbsp; </td></tr><tr>
<td align="center" valign="top">
<imgsrc="1_files/folder.gif"><td>
<td width="82%">国家四部联合通知：
餐... &nbsp; &nbsp; &nbsp; </td></tr>
</tbody>
```

03 声明表格主体的结束。

3. 代码和设计



4. 显示结果



Tfoot 标签

使用<tfoot>标签可以为表格定义一个表注。<tfoot>标签可能在表格结束之前只出现一次。和<thead>类似，它可以包括一个或多个<tr>标签，这样可以定义一些行，浏览器会将这些行作为表格的表注。因此，如果表格跨越多页，浏览器会重复这些行，一般浏览器会在多页表格每个部分的底部重复这些表注。在HTML中，由于表格结束时，表注也会结束，所以</tfoot>结束标签是可选的。

基本语法	<tfoot>	
	表脚注内容	
	</tfoot>（可省略）	
功能	定义表格的表脚注	
属性及说明	属性	说明
	Dir	文本方向
	Lang	语言信息
	Align	对齐方式



(续表)

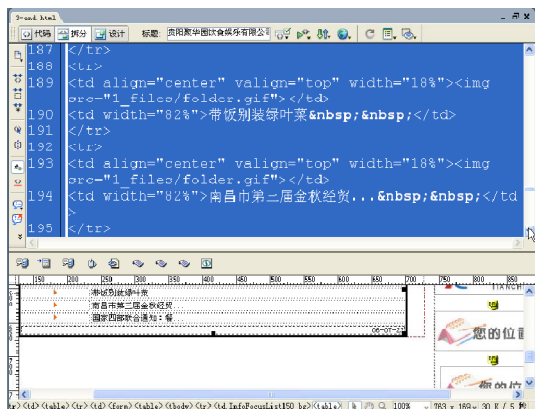
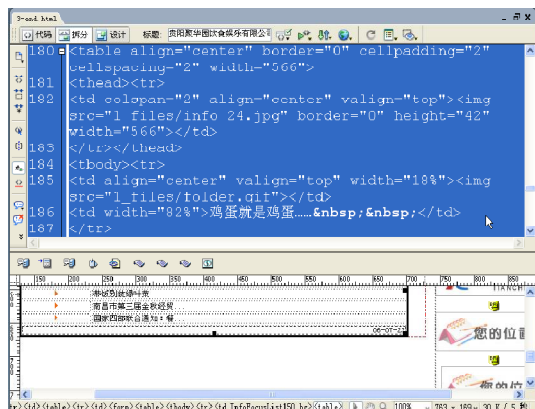
属性及说明	属性	说明
	Class	用一个名称来标记表格的表脚注, 该标记名称指向一个预定义的类, 而该类是在文档级声明的或者在外部定义的样式表
	Id	为表格的表脚注创建一个标记, 将来超链接可以用这个标记来明确地引用该表角注, 以便作为样式表选择器和其他应用程序来执行自动搜索
	Style	创建表格的表脚注内容的内联样式
	Title	给表格的表脚注加上说明性的文字
	Valign	表格的表脚注的垂直对齐方式

原始文件	Sample\Ch10\3\3.html
最终文件	Sample\Ch10\3\3-end.html
学习要点	使用<tfoot>标签

1. 范例代码与解释

```
<html>
<head>
...
</head>
<body>
...
<table align="center" border="0"
cellpadding="2" cellspacing="2"
width="566">
<thead>
<tr>
<td colspan="2" align="center"
valign="top"></td>
</tr>
</thead>
<tbody>
<tr>
<td align="center" valign="top"
width="18%"></td>
<td width="82%">鸡蛋就是鸡蛋.....
&nbsp;&nbsp;&nbsp;&nbsp;</td></tr><tr>
<td align="center" valign="top"
width="18%"></td>
<td width="82%">带饭别装绿叶菜
&nbsp;&nbsp;&nbsp;&nbsp;</td></tr><tr>
<td align="center" valign="top"
width="18%"></td>
```

2. 代码和设计



```

<td width="82%">南昌市第三届金秋经
贸...&nbsp;&nbsp;&nbsp;</td></tr><tr>
<td align="center" valign="top">

</td>
<td width="82%">国家四部联合通知:
餐...&nbsp;&nbsp;&nbsp;</td>
</tr>
</tbody>
<tfoot>
<tr>
<td colspan="2" align="center"
valign="top"><div align="right">
06-07-22</div></td>
</tr>
</tfoot>
</table>
...
</body>
</html>

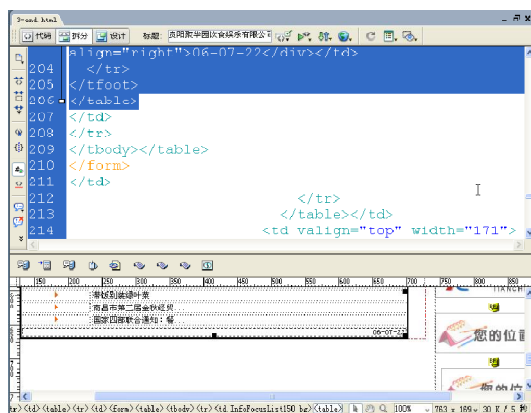
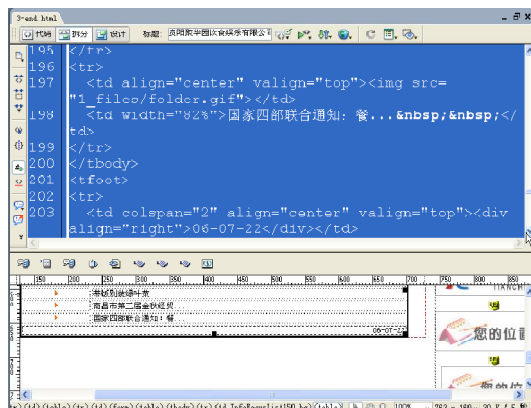
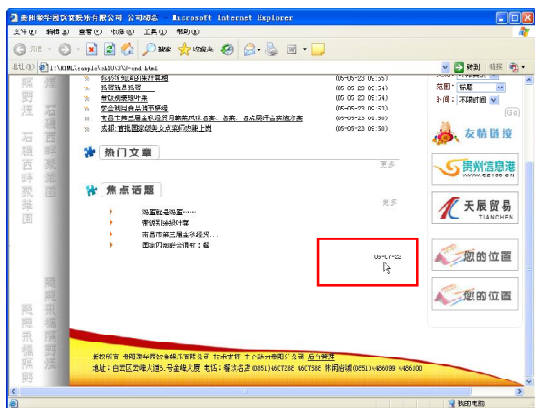
```

01 声明表脚注的开始。

02 声明表脚注的内容。

03 声明表脚注的结束。

3. 显示结果



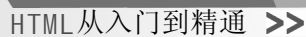
10

使用表格排版网页

4

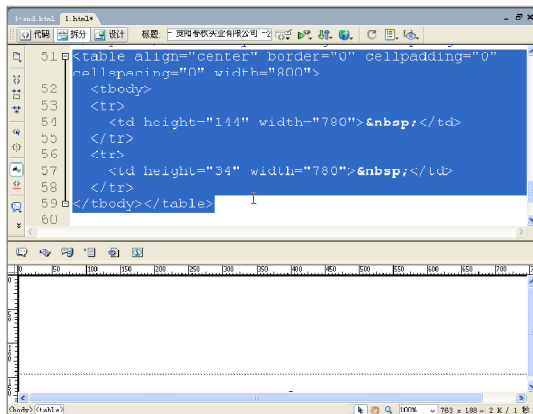
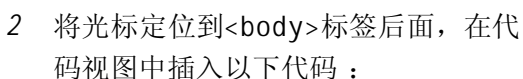
实例制作

下面通过一个实例练习页面中表格的创建以及使用表格排版页面的方法。



The screenshot shows a web browser window with the address bar displaying "http://www.chinacoffee.com.cn/". The page has a traditional Chinese aesthetic with a header featuring calligraphy and a navigation menu. The main content area is titled "【供应】2006年高茶上帮" (Supply: 2006 High Tea Upper Grade). A paragraph describes the product as a high-quality tea from the 2006 harvest, available in various grades. Below the text is a large image of a circular pile of green tea leaves. The browser's status bar at the bottom shows the file path "C:\Program Files\Internet Explorer\".

- 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。



- 242

- 3 将光标定位到第1行的<td>标签中, 插入一个Flash动画。

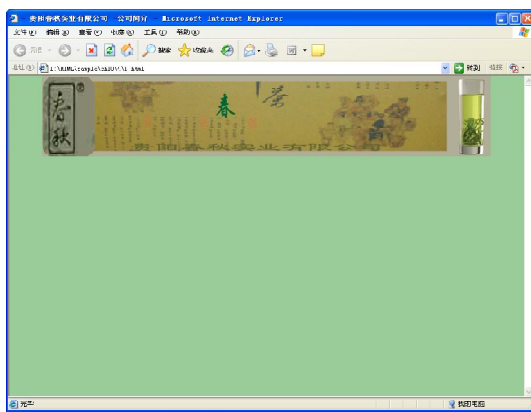
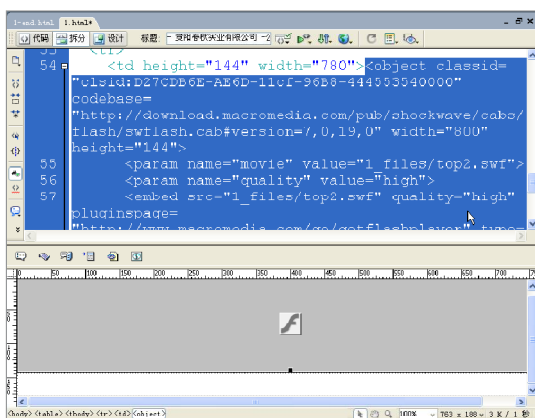
```
<object classid="clsid:D27CDB6E-
AE6D-11cf-96B8-444553540000"
  codebase="http://download.
  macromedia.com/pub/shockwave/
  cabs/flash/swflash.cab#version=
  7,0,19,0"
  width="800" height="144">
  <param name="movie"
  value="1_files/top2.swf">
  <param name="quality"
  value="high">
  <embed src="1_files/top2.swf"
  quality="high"
  pluginspage="http://www.
  macromedia.com/go/
  getflashplayer"
  type="application/x-shockwave-
  flash" width="800" height="144">
</embed>
</object>
```

01 声明Flash对象, 宽度为800, 高度为144。

02 声明源文件的路径为1_files文件夹下的top2.swf。

03 声明动画质量为高。

04 使用<embed>标记嵌入动画, 宽度为800, 高度为144。源文件的路径为1_files文件夹下的top2.swf。



按下 F12 快捷键预览页面, 此时的效果为动画插入到了页面。

- 4 将光标定位到第2行的<td>标签中, 插入一张图片。

```

```

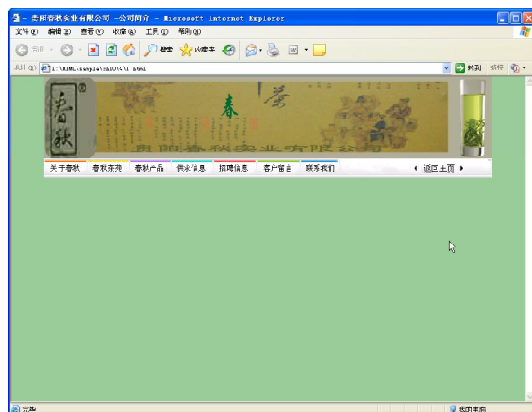
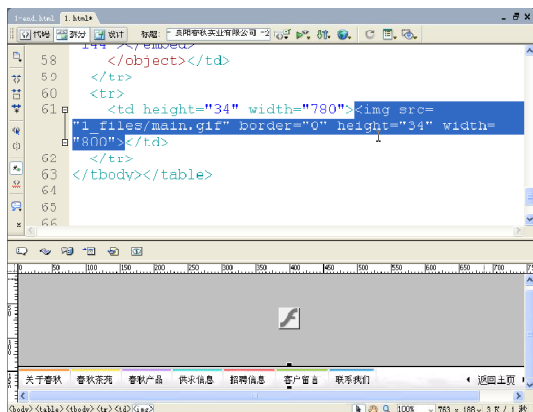
声明宽度为800, 高度为34、源文件的路径为1_files文件夹下的main.gif图片。

按下 F12 快捷键预览页面, 此时的效果为图片插入到了页面。



10

使用表格排版网页



5 将光标定位到刚才插入的表格标签的后面，在代码视图中插入以下代码：

```
<table align="center"
bgcolor="#ffffff" border="0"
cellpadding="0" cellspacing="0"
height="335" width="762">
<tbody>
<tr>
<td bgcolor="#f5f5f5"
valign="top" width="170">&nbsp;
</td>
<td bgcolor="#cccccc" width="1">
</td>
<td valign="top" width="583">
<p align="center">&nbsp;
</p>
</td>
</tr>
</tbody>
</table>
```

01 声明一个边框、单元格边距、单元格间距都为0，宽度为762像素，高度为335像素，背景色为白色，水平居中的表格。

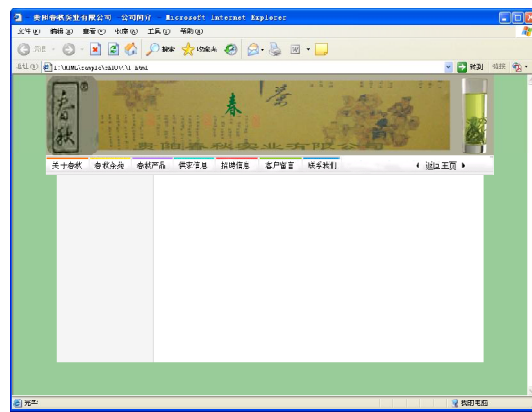
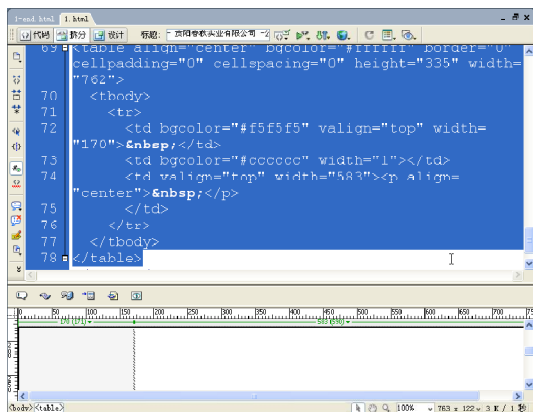
02 声明表的主体结构。

03 声明表格的第1行。

04 声明第1行的第1个单元格，宽度为170，居顶对齐，背景色为#f5f5f5。

05 声明第1行的第2个单元格，宽度为1，背景色为#cccccc。

06 声明第1行的第3个单元格，宽度为583，垂直居顶对齐，其中有一个水平居中对齐的段落。



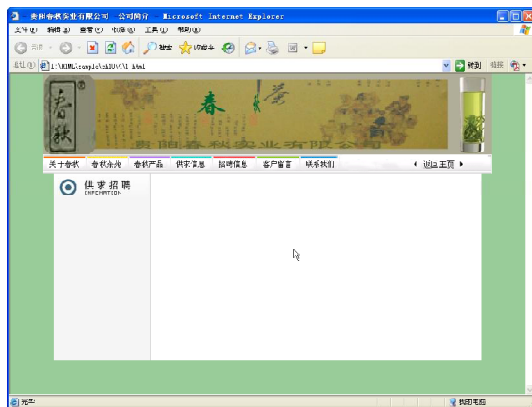
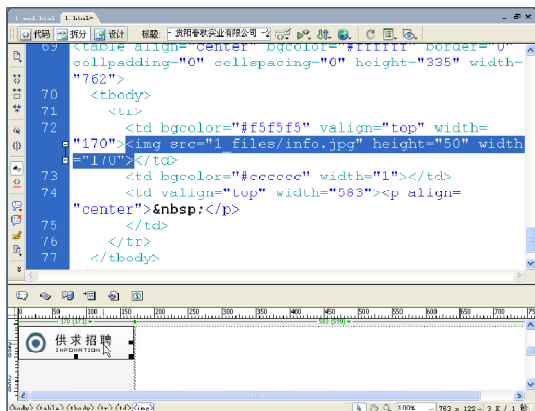
按下F12快捷键预览页面，此时的效果为表格插入到了页面。

- 6 将光标定位到刚才插入的表格的第1个单元格中，在代码视图中插入以下代码：

```

```

声明宽度为170，高度为50、源文件的路径为1_files文件夹下的info.jpg图片。



按下 F12 快捷键预览页面，此时的效果为图片插入到了页面。

- 7 将光标定位到刚才插入的表格的第3个单元格中，在代码视图中插入以下代码：

```
<div align="center">
<span class="style2"><br>
[供应]</span>
&nbsp;<span class="style1">2006
年新茶上市</span>
</div>
<p align="center"><br>
我公司生产的<font size="2">的“春秋”
牌雀舌报春，“春秋毛尖”现已上市，
欢迎新老顾客前来洽谈.</font>
</p>
<p align="center">
</p>
```

01 声明分块居中。

02 输入自定义样式的文字。

03 声明段落居中。

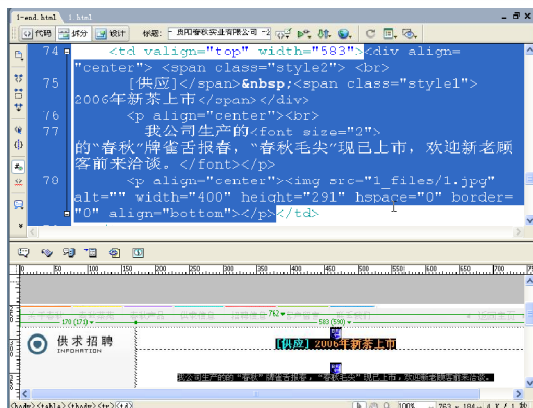
04 输入文字。

05 再次声明段落居中，并插入1_files文件夹下的1.jpg图片，宽度为400像素，高度为291像素，边框为0，垂直居底。



10

使用表格排版网页

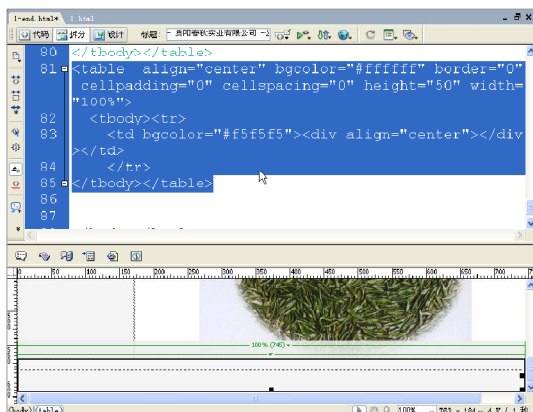


按下 F12 快捷键预览页面，此时的效果为文字和图片插入到了页面。

- 8 将光标定位到刚才插入的表格的后面，
在代码视图中插入以下代码：

```
<table align="center"
bgcolor="#ffffff" border="0"
cellpadding="0" cellspacing="0"
height="50" width="100%">
<tbody>
<tr>
<td bgcolor="#f5f5f5">
<div align="center"></div>
</td>
</tr>
</tbody>
</table>
```

- 01 声明 1 个边框、单元格边距、单元格间距都为 0、宽度为 100 百分比、高度为 50 像素、背景色为白色、水平居中的表格。
- 02 声明表的主体结构。
- 03 声明表格的第 1 行。
- 04 声明第 1 行的第 1 个单元格，背景色为 #f5f5f5。
- 05 声明一个居中的分块。



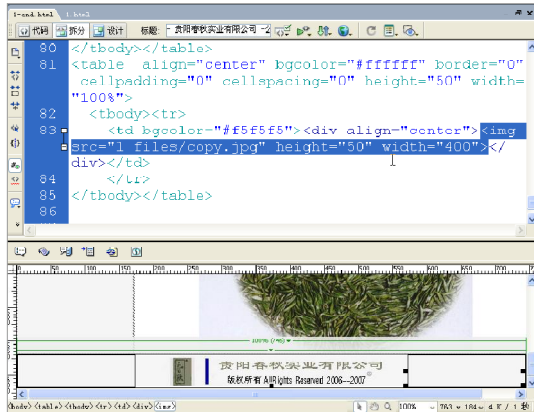
按下 F12 快捷键预览页面，此时的效果为表格插入到了页面。

9 将光标定位到刚才插入的表格的单元格中，在代码视图中插入以下代码：

```

```

● 插入1_files文件夹下的copy.jpg图片，其宽度为400，高度为50。



10 按F12快捷键，可以在浏览器中看到页面的最终效果：



10

使用表格排版网页