第1章 概　述

制作网页的基础是使用超文本置标语言(hypertext markup language,HTML),其核心思想是需要设置什么样式,就使用相应的 HTML 标记和属性。然而仅仅依靠 HTML 会遇到很多无法解决的问题,为此 HTML 逐步发展为 XHTML,与此同时 CSS 也应运而生。本章主要介绍 HTML、XHTML 和 CSS 三者之间的关系及 Web 前端开发的流程,读者应理解使用 CSS 的核心原理。

【学习目标】

- 了解 Web 前端开发的流程及 Web 标准。
- 掌握 XHTML 文档的结构。
- 会搭建 Web 前端页面开发环境。
- 会使用记事本实现简单网页的编写。

1.1 初探前端技术

1.1.1 万维网和网页技术

因特网(Internet)是一个笼统的说法，指的是庞大的全球范围的计算机网络，这些计算机彼此连接在一起，并且能够发送和接收数据，以接近光的速度在世界各地来回传送信息。万维网(World Wide Web)只是因特网的一个方面，由保存在因特网上不同的计算机上的无数文件和文档组成。这些交叉引用、彼此连接的文件和文档织成了一张世界范围的信息大网，由此得名万维网。在相对较短的历史中，Web 的发展已经远远超越了当初那种简单的文本文档的形式，同样的渠道如今承载着图像、视频和音频等多种信息，提供着引人入胜的互动体验。但就其本质而言，Web 基本上还是一种基于文本的媒体，它使用的文本编码通常为 HTML。

访问者浏览各种网站上的内容的过程，实际上就是从远程计算机中读取一些内容，然后在本地计算机上显示出来的过程。提供内容的计算机称为服务器。访问者使用浏览器程序(如集成在 Windows 操作系统中的 Internet Explorer)，通过网络取得服务器上的文件以及其他信息。

网站(web site)是由多个网页(web page)构成的具有相互联系的页面集合。一个网站少则由几个网页，多则由成百上千个网页组成，所有的信息都通过网页这个载体传递给浏览者。从是否执行程序来分，网页分为静态网页和动态网页。静态网页中没有程序代码，运行于客户端的程序、网页、插件和组件等都属于静态网页。在网络中看到的静态网页通常是以. htm 或. html 结尾的。动态网页内含有程序代码，运行于服务器端的程序、网页和组件等都属于动态网页，它们会随客户、时间及需求的不同返回不同的网页。在网络中看到的动态网页通常是以. asp、. php 或. jsp 等结尾的。

1.1.2 Web 前端技术的发展

近几年来，Web 开发技术的发展十分迅速。随着《网站重构》一书的问世，CSS 布局代替传统 table 布局之风迅速刮起，国内外大大小小的网站都纷纷加入到了这场技术革命中。Gmail(Google 的免费网络邮件服务)的上线，让 Ajax 一夜之间成为了 Web 开发领域的“明星”。随着 Ajax 的火热，DHTML(动态的 HTML)再次受到热捧，各种 JavaScript 框架如雨后春笋般涌现出来，让人应接不暇。

这种发展变化是一把双刃剑：一方面，它使得网页的表现力越来越强，人们可以做出有惊艳效果的网页；另一方面，漂亮的界面背后隐藏着的是越来越难维护的实现代码。为什么网页的维护工作会变得越来越难？原因主要来自以下三个层面。

(1)浏览器层面。浏览器的向前兼容使得前端开发中被淘汰的技术、不推荐的方法依然广为流传和应用，而新一轮的浏览器大战却愈演愈烈，除了 Firefox、Opera、Safari、Chrome 这些 IE 的挑战者外，IE 本身也同时流行着 IE 6、IE 7 和 IE 8 三个不同的版本。不同的浏览器对网页代码的解析存在着或大或小的差异。

(2)技术层面。Web 标准被重视和普遍采用的时间并不长,整个大环境对 Web 标准的理解还停留在概念层面,对“好的实现方案”仍处于摸索阶段。而不同的公司、团队、工程师对“好的实现方案”有自己的理解。理解不深,就很容易写出可维护性差的代码。

(3)团队合作层面。随着用户对使用体验要求的不断增加,对网页表现力的要求也越来越高,从而导致实现代码越来越复杂,这无疑给团队合作带来了麻烦。代码越复杂,对团队合作的要求就越高。如果团队合作不默契,很可能需要不停地打补丁,最后让代码变得千疮百孔,满是“地雷”,没有人愿意维护它。

随着维护难度的增加,网页制作对技术的要求越来越高,Web 开发领域长期以来形成的设计和制作不分的局面终于有所好转。细心的朋友会发现,在 51job 招聘网站上,前些年几乎只有“网页设计师”这个职位,而现在已经有了前端开发工程师和页面工程师的职位之分。之前既要负责设计又要负责制作的网页设计师已经分离成了两个岗位:一个专门负责设计,属于艺术类;另一个专门负责制作,属于技术类。这是一个可喜的变化,因为设计师(designer)和开发者(developer)负责的本来就是两个完全不同的方向,将两者明确分开,表示网页制作的分工向着合理成熟的方向又迈进了一大步。若专注于网页制作的技术方向,则有一种更专业的叫法——前端开发。

1.1.3 Web 标准——结构、表现和行为的分离

网页主要由三部分组成:结构(structure)、表现(presentation)和行为(behavior)。

用一本书来比喻,一本书分为篇、章、节和段落等部分,这构成了一本书的“结构”,而每个部分使用的字体、字号、颜色等是这本书的“表现”。由于传统的图书是固定的、不能变化的,因此不存在“行为”。

在一个网页中,同样可以分为若干组成部分,如各级标题、正文段落、各种列表等构成了一个网页的“结构”,而每个组成部分的字号、字体和颜色等属性则构成了“表现”。网页和传统媒体不同的一点是,它可以随时变化,而且可以和访问者互动,因此如何变化以及如何交互被称为它的“行为”。

概括来说,“结构”决定网页“是什么”,“表现”决定网页看起来是“什么样子”,而“行为”决定网页“做什么”。

“结构”“表现”和“行为”分别对应三种常用的技术,即(X)HTML、CSS 和 JavaScript。也就是说,(X)HTML 用来决定网页的结构和内容,CSS 用来设定网页的表现形式,JavaScript 用来控制网页的行为。这三个部分被明确以后,一个重点的思想随机产生,即这三者的分离。最开始的时候,HTML 同时担任着“结构”和“表现”的双重任务,从而给网站的开发、维护等工作带来了很多困难。

万维网协会(World Wide Web Consortium,W3C)是一个专门负责制定网页标准的非营利性组织,致力于结束网页制作领域混乱不堪的局面,Web 标准就是由 W3C 组织推行的。Web 标准由一系列标准组合而成,其核心理念就是将网页的结构、表现和行为分离开来,所以它分为三大部分:结构标准、样式标准和行为标准。结构标准包括 XML 标准、XHTML 标准、HTML 标准;样式标准主要是指 CSS 标准;行为标准主要包括 DOM 标准和 ECMAScript 标准。

一个符合标准的网页,标签中的标签名应该全部都是小写的,属性要加上引号,样式和

行为不再夹杂在标签中，而应该分别单独存放在样式文件和脚本文件中。理想状态下，网页源代码由三部分组成：.html 文件、.css 文件和.js 文件。

1.1.4 浏览器与 CSS

网上的浏览器各式各样，绝大多数浏览器对 CSS 都能很好地支持，因此设计者不用担心其设计的 CSS 文件不能显示。目前主要的问题在于各个浏览器对 CSS 很多细节的处理上存在差异，设计者在一种浏览器上设计的 CSS 效果，与在其他浏览器上的显示效果很可能不一样。就目前主流的两大浏览器 IE 与 Firefox 而言，在某些细节的处理上就不尽相同，IE 6 与 IE 7 对相同页面产生的浏览效果也存在一些差异。

相对于目前使用较多的 3 种浏览器 IE 6、IE 7 和 Firefox，制作网页后应该进行调整，使页面在这 3 个浏览器中都能正确显示，这样就可以保证 99%以上的访问者能够正确浏览该网页。

各个浏览器的显示效果存在差异，主要是因为各个浏览器对 CSS 样式默认值的设置不同。因此，只要严格规范 CSS 文件各个细节的编写，就能使各个浏览器达到基本相同的显示效果。

提示：使用 CSS 制作网页时，一个基本的要求就是，网页在主流浏览器中的显示效果要基本一致。通常的做法是：一边编写 XHTML 和 CSS 代码，一边在两个不同的浏览器上进行预览，及时调整各个细节。另外，Dreamweaver 的“视图”模式只能作为设计时的参考使用，绝对不能作为最终显示效果的依据，只有浏览器中的效果才是用户所看到的。

1.1.5 Web 前端技术的开发流程

Web 前端技术的开发流程必须按照步骤执行，大概分为 7 步，如图 1-1 所示，每个步骤下面列出的是该步骤可以（或可能）用到的工具。

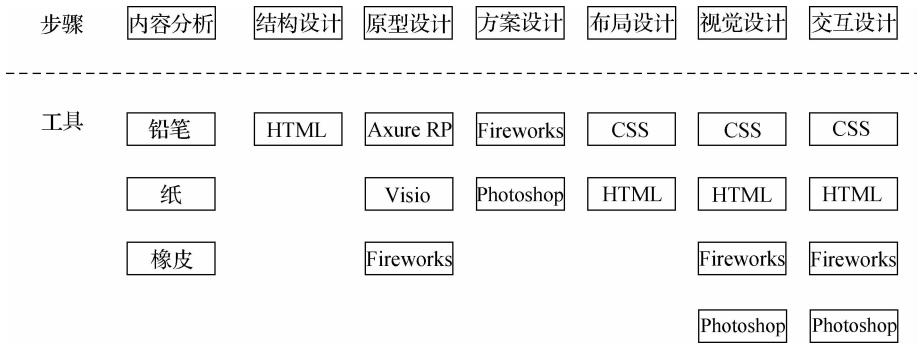


图 1-1 Web 前端技术的开发流程

（1）内容分析：仔细研究需要在网页中展现的内容，梳理其中的逻辑关系，分清层次以及重要程度。

（2）结构设计：根据内容分析的成果搭建出合理的 HTML 结构，保证在没有任何 CSS 样式的情况下在浏览器中保持高可读性。

（3）原型设计：根据网页的结构绘制出原型线框图，对页面进行合理的分区和布局。原型线框图是设计负责人与客户交流的最佳媒介。

（4）方案设计：在确定的原型线框图基础上，使用美工软件设计出具有良好视觉效果的

页面设计方法。

(5)布局设计:使用 XHTML 和 CSS 对页面进行布局。

(6)视觉设计:使用 CSS 并配合美工设计标签,完成由设计方法到网页的转化。

(7)交互设计:为网页增添交互效果,如鼠标指针经过时的一些特效等。

课堂实训 1-1　搭建 Web 前端页面环境

1. 实训内容

安装 Dreamweaver CS3 开发软件,搭建 Web 前端页面开发环境。

2. 实训目的

学会安装 Dreamweaver CS3 和搭建 Web 前端页面开发环境的方法。

3. 实训步骤

(1)将 Dreamweaver CS3 的安装光盘放入光盘驱动器,系统会自动运行 Dreamweaver CS3 的安装程序。安装程序自动弹出安装向导窗口的欢迎界面,如图 1-2 所示。如果系统中有应用程序正在运行,会提示关闭所有打开的应用程序。关闭系统中所有打开的应用程序后,单击“下一步”按钮。

图 1-2　欢迎界面

(2)安装过程中需要创建一个文件夹,用来存放 Dreamweaver CS3 的全部内容。可以将 Dreamweaver CS3 安装在默认的文件夹中,如图 1-3 所示。如果想要更改安装路径,则可以单击“浏览”按钮,选择需要安装的路径。

图 1-3　设置安装路径

(3)选择好安装的路径后单击“下一步”按钮，弹出“选择附加任务”窗口，如图 1-4 所示，提示是否创建桌面快捷方式，这里选中“创建桌面快捷方式”复选框，单击“下一步”按钮。

图 1-4 “选择附加任务”窗口

(4)此时安装程序显示已经做好了安装的准备，如图 1-5 所示，单击“安装”按钮进行安装。

图 1-5 确认信息

(5)软件开始安装，如图 1-6 所示。

图 1-6 开始安装

(6)安装完成后会显示一个安装完成界面,如图 1-7 所示,单击“完成”按钮完成 Dreamweaver CS3 的安装。

图 1-7　完成安装

1.2 Web 前端页面基本框架

1.2.1 HTML

要想把互相连接的数据文本片段“编织”成 Web,就必须有某种用于建立这种连接的技术,这就是超文本(hypertext)的基础。使用超文本可以将一个文档中的一串文字直接链接到 Web 上的另一个文档。超文本置标语言(hypertext markup language,HTML)是一种计算机语言,用于将普通文本转化为活动(active)文本,以供显示和在 Web 上使用,并为普通的无结构文本提供结构。如果没有结构,单纯的文本将会汇合在一起,以至于无法把一串文字与另一串文字区分开来。

HTML 由一些称为标签(tag)的经过编码的标记符组成,标签包围着文本片段,将其与其他部分区分开来,并且表明了所标记的文本的功能和用途。标签被直接嵌入普通文本文档中,并在计算机软件处理该文档时得以理解。标签自身不会被显示,并且会被区别对待于它们所封装的实际内容。

HTML 被刻意设计成一种简单、灵活的语言。它是一个免费、公开的标准,不需要购买许可,也不需要使用特别的软件来创建 HTML 文档,任何人都可以自由地创建和发布网页。Web 能发展成为现在这样一种强大、影响深远的媒体正是得益于这种开放性。

当使用 HTML 创建文档时,必须遵守特定的规则,因为 HTML 中的各种要素必须按特定的方式组织起来才能正确发挥作用。这些规则由 W3C 负责维护。W3C 制定了用以构建 Web 的许多公开的技术标准,它们统称为 Web 标准。

1.2.2 XHTML 编写规范

尽管目前浏览器都兼容 HTML，但为了使网页能够符合标准，设计者应该尽量使用 XHTML(可扩展超文本置标语言)规范来编写代码。下面介绍 XHTML 的编写规范。

1. XHTML 标签和属性名必须小写

对于所有标签和属性名，XHTML 文档必须小写。因为 XHTML 是对大小写敏感的，如<li>和<LI>是不同的标签。

2. XHTML 文档必须具有良好完整的排版

编排良好性(well-formedness)是 XHTML 引入的一个新概念，即 XHTML 要求必须有结束标签，或者必须以特殊方式书写，而且标签可以嵌套。

1)非空标签必须使用结束标签

XHTML 不允许忽略结束标签。除了在 DTD(一套关于标记符的语法规则)中被声明为空标签，其他标签必须有结束标签。

下面的写法带有结束标签，是正确的：

```
<p>XHTML 教程</p><p>CSS 教程</p>
```

下面的写法中没有结束标签，是错误的：

```
<p>XHTML 教程<p>CSS 教程
```

2)空标签的规定

空标签也必须有结束标签，或者起始标签必须以“/>”结束。

下面的空标签有结束标签，是正确的：

```
<br/><hr/>
```

下面的空标签没有结束标签，是错误的：

```
<br><hr>
```

3)标签可以嵌套

下面的标签嵌套的形式是正确的：

```
<p>梦之都<em>XHTML 教程</em></p>
```

下面的标签嵌套的形式是错误的：

```
<p>梦之都<em>XHTML 教程</p></em>
```

3. XHTML 属性值必须在引号中

XHTML 所有的属性值必须在引号中，即使是数字形式的属性值。

下面的写法是正确的：

```
<table rows="3">
```

下面的写法是错误的：

```
<table rows=3>
```

4. 不支持属性最小化

XHTML 不支持属性最小化，属性名与属性值必须完整、成对地出现。像 disabled、checked 这样的属性名不能在不指定属性值的情况下出现。

下面的写法是正确的：

```
<input checked="checked">
```

下面的写法是错误的：

```
<input checked>
```

1.2.3　XHTML 的文档结构

XHTML 的文档结构如下：

```
<! DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict //EN"
  "http://www.w3.org/TR/xHTML1/DTD/xHTML1-strict.dtd">
<html>
    <head>
      <title>HTML 文件标题</title>
      HTML 头信息
    </head>
    <body>
      HTML 内容信息
    </body>
</html>
```

关于 XHTML 文档结构有以下几点说明。

(1)XHTML 文档首先要声明一个文档类型。文档结构的第一行就定义了 XHTML 文档的类型为 XHTML 1.0(文档类型用于说明这个页面使用了何种 HTML 规则或者结构)。

(2)<html>和</html>是 HTML 文档的开始与结束，也是 XHTML 文档的根标签。

(3)除了文档类型外的所有页面内容都包括在 HTML 标签中。

(4)XHTML 文件主要包括头信息(head)与内容信息(body)。

(5)头信息可以容纳文档的 HTML 相关信息，如标题(title)、页面的语言与文字类型、CSS 样式、JavaScript 代码、简短描述、关键词等内容，这些是用户无法直接看到的。

(6)内容信息包括用户可以看到的全部内容，如段落、链接、表格等。

课堂实训 1-2　一个典型的 Web 应用

1. 实训内容

旅游已经成为当前比较时尚的一种休闲方式，受到越来越多人的追捧。制作这类网站可以从以下 3 个方面规划。

(1)要放置足够多的旅游景点信息，用如画的景色图片吸引访客。

(2)要为访问者提供充足的旅游线路和价目表，让访客感受到网站的价值。

(3)要提供大量不断更新的景点介绍信息，让访问者能持续不断地关注网站。

2. 实训目的

分析图 1-8 所示的旅游资讯网站的制作过程，了解 Web 前端开发流程。

图 1-8　旅游资讯网站效果图

3. 技能分析

该旅游资讯网站的制作过程如下。

(1)内容分析:研究需要在网页中展现的内容,梳理其中的逻辑关系,分清层次及重要程度,收集所需的图片、文字、音频、视频、动画等素材。

(2)结构设计:根据内容分析的结果,搭建 HTML 结构。

(3)原型设计:根据网页的结构绘制原型线框图,对页面进行合理的分区布局。

(4)方案设计:根据原型线框图使用 Photoshop 等图像软件进行页面方案设计,这一步一般是美工的任务。

(5)布局设计:使用 HTML 和 CSS 对页面进行布局。

(6)视觉设计:使用 CSS 并配合美工设计标签,完成由设计方法到网页的转化。

(7)交互设计:为网页添加交互效果,如鼠标指针经过时的一些特效等。

1.3 实战训练:手动编写第一个前端页面

在记事本中使用 HTML 标签制作一个欢迎页面,效果如图 1-9 所示。

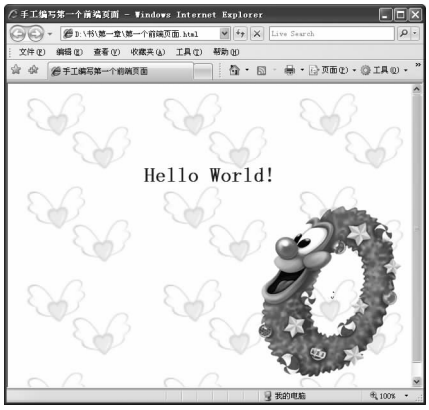


图 1-9 欢迎页面效果图

1. 任务分析

页面中包含背景图片、欢迎文字及图片，风格简单明了。

2. 任务提示

(1)利用<body>标签的 background 属性设置页面的背景图片。

(2)“Hello World!”文字可用<h1>标签，并设置其 align 属性为 center(居中显示)。

(3)利用<font>标签的 style 属性设置文字颜色。

(4)利用<img>标签插入图片并设置其样式，包括图片大小(weight、height)和对齐方式(align)。

3. 步骤提示

(1)准备好素材图片 a_1.jpg 和 a_2.gif，并存放在 images 文件夹中。

(2)运行“记事本”程序，输入代码清单 1-1 中的代码。

```
<html>
    <head>
      <title>手工编写第一个前端页面</title>
    </head>
    <body background="images/a_1.jpg">
        <p> </p>
        <p> </p>
        <p> </p>
        <h1 align="center"><font style="color:#0000CC">Hello World! </font>
        </h1>
        <img src="images/a_2.gif" align="right" width="233" height="279">
    </body>
</html>
```

代码清单 1-1 手工编写第一个前端页面的文档代码

提示：“&.nbsp;”代表一个空格。

(3)将文件保存为 html 格式。

4. 技术难点提示

文件保存为 html 格式或者 htm 格式，而不是记事本程序默认的 txt 格式。

1.4 思考与练习

一、选择题

1. 从是否执行程序来分，网页分为(　　)和(　　)两种类型。

A. 静态网页　　B. 动态网页　　C. 首页　　D. 活动网页

2. 下面以(　　)结尾的文件不是网页文件。

A. html　　B. htm　　C. jsp　　D. doc

3. (　　)不是网页的基本构成标签。

A. 文本　　B. 图像　　C. 数据表　　D. 表单

4. 超文本置标语言简写为(　　)。

A. FTP　　B. HTML　　C. IP　　D. HTTP

二、简答题

1. XHTML 的编写规范是什么？
2. XHTML 文档的基本结构由哪几部分组成？
3. 如何判断一个网站是否符合 Web 标准？

第2章

XHTML 基础知识

可扩展超文本置标语言(extensible hypertext markup language,XHTML)是一种置标语言,表现方式与 HTML 类似,不过语法上更加严格。从继承关系上讲,HTML 是一种基于标准通用置标语言(SGML)的应用,而 XHTML 是基于可扩展置标语言(XML)的,XML 是 SGML 的一个子集。本章主要介绍网页设计中各种标签元素的使用。

【学习目标】

- 掌握 XHTML 文档的结构。
- 会使用各种 XHTML 标签及其属性。
- 会使用 Dreamweaver 软件可视化的方式实现一个简单网页的设计。

2.1 XHTML 基本框架

2.1.1 XHTML 的部件

1. XHTML 简介

HTML 不是一种编程语言，而是一种标记语言。标记语言是一套标记标签(markup tag)的集合，HTML 使用标记标签描述网页。HTML 标记标签简称 HTML 标签，是由尖括号包围的关键词，如＜html＞。

与其他标记语言一样，XHTML 的关键也是标签(tag)。标签是经过编码的符号，用于分隔和区分内容的不同部分，并告知浏览器它所处理的是何种类型的内容。大多数 XHTML 标签的名字都准确描述了它们的用途和它们所标注的内容的类型，如标题、段落、列表、图像、引文等。标签是 XHTML 文档的基本组成要素。

XHTML 的标签包围在一对尖括号(＜＞)之间，以便将它们与普通文本明确区分开。“＜”标明了标签的开头，随后是特定的标签名(tag name)，最后以“＞”结束。例如，表示段落开头的 XHTML 标签是＜p＞。

注意：标签名必须使用小写，这是 XHTML 的要求。标签名在 HTML 中不区分大小写，但在 XHTML 中必须是小写，这是 XHTML 区别于 HTML 的严格规则之一。

大多数标签都配对使用：一个开标签(start tag)用于表明一个内容片段的开始，还有一个闭标签(end tag)用于表明其结束。例如，段落的开始用开标签＜p＞表示，其结束用闭标签＜/p＞表示。闭标签中“＜”后的斜线将它与开标签区别开来。一个完整的段落标记如下：

```
＜p＞Hello,world!＜/p＞
```

某些标签表示空标签，它们没有内容。空标签不需要闭标签，但需要通过在代表整个标签的一个单独的标签的尾部使用一个结尾斜线来“自闭合”。例如，标签＜br/＞就是一个空标签，其功能是在浏览器呈现文档时，强迫后面的文本转到新行上。

2. 标准属性

标签可含有属性，包括必需属性和可选属性。有些属性是通用的，可以用于大部分标签(并且基本都是可选的)，这些通用的属性称为标准属性。为避免重复，这里把标准属性单独列出，本书在后面介绍每个具体的标签时，将列举它们的必需属性、可选属性和标准属性。

标准属性分为以下几类。

1)核心属性

这类属性包含关于标签的一般性信息，可以包含在大部分标签的开标签内。

(1)class：表示特定标签所属的一个类或一组类。同属于一个类的标签可以共享一些呈现性特征，而且将标签分类对客户端脚本编程也有用。类名由字母、数字、连字符(-)和下划线(_)组成，不能包含其他标点符号或特殊符号。多个标签可以属于同一类。一个标签也可以属于不止一个类，此时属性值中的多个类名用空格分隔。

(2)id:为标签指定一个具有唯一性的标识符。id 在一个文档中必须是唯一的,多个标签不能共用一个标识符。id 标识符由字母、数字和符号组成,但不能包含除连字符和下划线之外的任何标点符号或特殊符号。其中,第一个字符必须是字母,而不能是数字或其他任何字符。

(3)style:为标签指定 CSS 属性,也称为内联样式定义。虽然 style 属性对大多数标签都有效,但应避免使用,因为它把内容和表现混在了一起。

(4)title:为标签提供一个文本标题。许多图形化浏览器将 title 属性的值显示在"工具提示"(当用户的鼠标指针停留在所呈现的标签上方时出现的浮动窗口)中。

2)国际化属性

国际化属性包含关于书写标签内容的自然语言(如英语、法语、拉丁语等)的信息。它们可以包含在大部分标签中,特别是那些包含的文本所使用的语言不同于文档其他部分的标签。

(1)dir:把文本的阅读方向设置为由值 ltr(从左到右)或 rtl(从右到左)所指定的方向。通常很少使用这个属性,因为语言的方向应该从 lang 和 xml:lang 属性推断。

(2)lang:指定文档所用语言。语言用一种缩写的语言代码表示,如 en 代表英语,es 代表西班牙语,zh 代表汉语等。

(3)xml:lang:也用于指定文档所用语言。这是 lang 属性的 XML 形式,用于 XML 文档中。XHTML 文档既是 XML,又是 HTML,这取决于服务器如何交付它,因此标签既可指定 lang 属性,也可指定 xml:lang 属性,它们的值是同样的语言代码。

3)焦点属性

当某些标签,尤其是链接和表单控件处于预激活状态时,称为拥有焦点,可以为这些标签设置下列焦点属性,以增强其用键盘在网页上导航的可用性。

(1)accesskey:为标签分配一个快捷键,以便在使用键盘导航时能更方便、快捷地访问它。该属性的值是对应访问键的字符。用于激活访问的实际按键组合因浏览器和操作系统的不同而不同。

(2)tabindex:指定标签在使用 Tab 键遍历链接和表单控件时,所形成的访问顺序中的位置。

3. 添加注释的方法

在文档中嵌入注释很有用,它们不会被浏览器显示,但在查看源代码时可以看到。注释中主要包含一些说明信息,如说明文档为何要以特定方式进行组织、文档的修改指南或改动历史记录等。XHTML 中的注释使用一种特定的标签结构,格式如下:

```
<!-- Use an h2 for subheadings -->
<h2>Adding Comments</h2>
```

一段注释始于"<!--",终于"-->"。浏览器不会显示任何出现在这些标记符号之间的内容或标签,即使注释跨越多行也不会显示。

注释也可用于在测试网页时暂时"隐藏"一部分标记代码,如下面的代码:

```
<!-- Hiding this for testing
<h2>Adding Comments</h2>
End Hiding -->
```

虽然注释不会被浏览器显示出来，但是它们依然随着其余的代码一起被发送，而且访问者查看网页的源代码时可以看到它们。所以，不要期望注释能完全保密，也不要依赖它们来永久性地删除或禁用任何重要的内容或标记代码。

2.1.2 XHTML 文档结构简介

1. 一个基本的 XHTML 文档

一个有效、规范的 XHTML 文档必须符合严格的结构，并按确定的格局安排一些必需的组成部分。代码清单 2-1 展示了一个规范文档的基本架构。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xHTML1/DTD/xHTML1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xHTML" lang="en" xml:lang="en">
    <head>
        <title>My first Web page</title>
    </head>
    <body>
        <p>XHTML is easy! </p>
    </body>
</html>
```

代码清单 2-1　一个基本的 XHTML 文档代码

尽管上述代码上看上去很简单，但这确实是个完整、有效、规范的文档。XHTML 文档开始于一个文档类型声明(document type declaration，DOCTYPE)。这是一个必需的组成部分，它声明了文档的类型及其所遵守的标准规则集。文档类型声明是一种标签，虽然也使用尖括号，但它并不是 XHTML 标签，因此并不需要一个闭标签或结尾斜线。实际上，它根本不是 XHTML 文档标记代码的一部分，它只是向用户代理传达关于文档的信息，助其判断所处理的文档的类型，并按照恰当的规则呈现页面。

2. <html>标签

实际的标记代码从<html>标签开始。<html>标签是整个文档的容器，称为根标签。<html>标签没有自己特有的属性，纯粹是一个用于定义文档的开始和结尾的容器。出现在这个标签之外的任何标签或内容(除了并非标签的文档类型声明)都将导致整个文档无效。

1)必需属性

xmlns：用于指定 XML 命名空间的 URL。对于 XHTML 文档，该属性的属性值应该是 http://www.w3.org/1999/xHTML。

2)可选属性

该标签没有可选属性。

3)标准属性

该标签的标准属性包括 dir、id、lang 和 xml:lang。

命名空间(namespace)是 XML 中规定标签和属性名称的地方。XML 是一种可扩展标记语言，允许作者自定义标签和属性。例如，对于一个关于动物的文档来说，一个具有

species属性的 animal 标签会很有用。这些自定义名称可以定义在一个特别的命名空间中。XHTML 文档中的命名空间通过＜html＞标签的 xmlns 属性声明。

标准的 lang 和 xml:lang 属性对＜html＞标签来说是可选的。但由于＜html＞标签是根标签，所有其他标签都是其后裔，这里的语言声明将会传递到文档中的所有其他标签，所以建议指定这两个属性。

3. 文档的其他部分

文档的其他部分由＜head＞和＜body＞标签组成。＜head＞标签包含关于文档自身的信息(包括必需的＜title＞标签)，而＜body＞标签则包含所有最终由浏览器呈现并供访问者浏览和使用的内容。这些标签将在下面详细介绍。

总而言之，XHTML 文档的基本结构非常简单，只需要一条文档类型声明、一个根标签、一个具有＜title＞标签的＜head＞标签和一个＜body＞标签。

4. 文档树的结构

可以把 XHTML 文档的结构想象为一棵倒置的树，这棵树发端于顶部的根标签，并向下延伸出所有其他标签。它更像是一个家谱图，出于这个原因，家谱学中的术语常被用来描述标签之间的关系。图 2-1 展示了一棵简单的文档树。

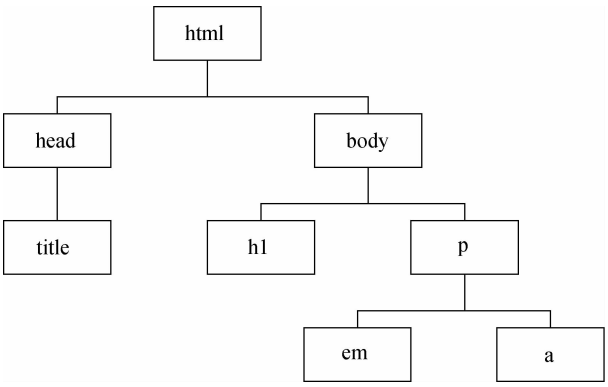


图 2-1　一棵简单的文档树

在图中，树始于根标签＜html＞。根标签有两个子(child)标签——＜head＞和＜body＞。＜body＞标签又有两个自己的子标签：代表一级标题的＜h1＞标签和代表段落的＜p＞标签。这两个标签互为同胞(sibling)，＜body＞是它们共同的父(parent)标签，它们是＜html＞标签的后代(descendant)，同时＜html＞标签也是它们的祖先(ancestor)。图中的＜p＞标签包含一个＜em＞标签和一个＜a＞标签，它们是同胞，其父(＜p＞)的孩子，传承了祖先＜body＞和＜html＞的血脉。

课堂实训 2-1　制作第一个网页——Hello World

1. 实训目的

利用 Dreamweaver 制作一个简单的页面，页面效果如图 2-2 所示。

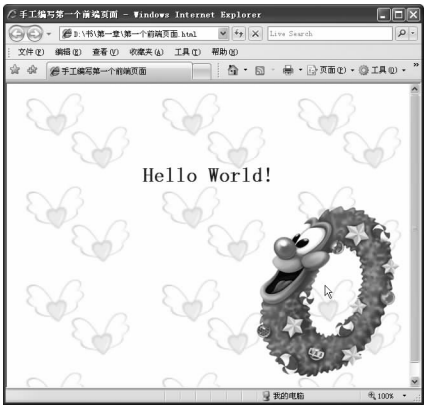


图 2-2　页面效果图

2. 技能分析

(1)利用 Dreamweaver CS3 创建基本的静态页面。

(2)掌握标签的使用方法。

3. 实训步骤

(1)利用 Dreamweaver CS3 创建静态页面 HTML,设置文档类型,操作如图 2-3 所示。

图 2-3　设置文档类型

(2)插入素材图片和文字,代码如代码清单 2-2 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xHTML1/DTD/xHTML1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xHTML">
    <head>
        <meta http-equiv="Content-Type" content="text/HTML; charset=utf-8"/>
        <title>Hello World!</title>
```

```
    </head>
    <body background="images/a_1.jpg">
        <p> </p>
        <p> </p>
        <p> </p>
        <h1 align="center"><font style="color:#0000CC">Hello World! </font>
        </h1>
        <img src="images/a_2.gif" align="right" width="233" height="279"/>
    </body>
</html>
```

代码清单 2-2　一个基本的 XHTML 文档代码

2.2 辅助的<head>标签

2.2.1 <title>标签

可以使用<title>标签为文档提供一个标题。浏览器通常将其值显示在标题栏中，并用它作为书签的默认名称。

1. 必需属性

该标签没有必需属性。

2. 可选属性

该标签没有可选属性。

3. 标准属性

该标签的标准属性包括 class、id、style、dir、lang 和 xml:lang。

要为文档添加标题，只需将要显示的文本放在<title>标签和</title>标签之间即可。<title>标签的用法如代码清单 2-3 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xHTML1/DTD/xHTML1-strict.dtd">
<html>
    <head>
        <title>This text will be displayed within the titlebar</title>
    </head>
    <body>
    </body>
</html>
```

代码清单 2-3　<title>标签的用法

图 2-4 显示了上述代码在浏览器中的显示结果。

图 2-4 ＜title＞标签在浏览器中的显示结果

提示：一般来说，应该为文档指定一个标题。因为它会显示在浏览器的标题栏中，即使其浏览器窗口未被最大化，也能让用户知道正在浏览的是哪个文档。如果不指定标题，文档将显示一个空标题。

2.2.2 ＜meta＞标签

＜meta＞标签除了提供文档字符集、使用语言、作者等信息外，还涉及对关键词和网页等级的设定。搜索引擎可以使用它们来编录文档。＜meta＞标签另一种常见的用法是使用 http-equiv 属性来让浏览器自动刷新文档。meta 一词代表元数据(metadata)，这个术语通常被解释为“关于数据的数据”。＜meta＞标签提供的正是关于文档中的数据的数据。

1. 必需属性

content：与一个名称(指 name 属性的值)或 http-equiv 关联的值。

2. 可选属性

(1)http-equiv：把 content 属性的值与一个特定的 HTTP 响应头相关联。可以使用它来要求浏览器做某件事，或在外部的信息中引用关于文档的来源。

(2)name：用于为文档赋予额外的信息。该属性的值来自 content 属性。常见的名称包括 author、keywords、description、summary。

(3)scheme：定义一种用以解释 content 属性值的格式。

3. 标准属性

该标签的标准属性包括 dir、lang 和 xml：lang。

4. 用法

通过将＜meta＞标签的 name 属性指定为 keywords，并将 content 属性指定为一个用逗号分隔的关键词列表，即可为搜索引擎提供线索。例如，可以为本书使用下列 XHTML 代码：

```
＜meta name＝"keywords" content＝"HTML,XHTML,CSS,JavaScript."/＞
```

也可以通过将 name 属性指定为 description，然后用 content 属性值为搜索引擎提供一份关于文档的简短说明。例如：

```
<meta name="description" content="This is an introduction to XHTML/HTML."/>
```

http-equiv 属性提供了很多功能，如指定其值为 refresh，可以让文档按指定的时间间隔被刷新。下面的代码实现每隔 15 s 刷新一次文档：

```
<meta http-equiv="refresh" content="15"/>
```

说明：上面的代码应谨慎使用，否则可能会惹恼网站的访问者！

2.2.3　<base>标签

<base>标签有助于把链接变得更简短，更易维护。它可用来为文档中的所有链接指定一个基础 URL。

1. 必需属性

href：指定一个 URL，用做文档中的所有链接的基础 URL。

2. 可选属性

该标签没有可选属性。

3. 标准属性

该标签没有可用的标准属性。

在向 XHTML 文档中加入一个图像时，需要说明在什么地方能找到图像。当多个图像位于同一个目录时，使用<base>标签能够让事情更轻松一些。例如，如果一个文档包含了来自同一目录的几个图像，那么可以通过使用<base>标签让它们的 URL 变得短一些。此外，当把这些图像移到新的位置时，修改那些链接很简单，只需修改<base>标签的 href 属性即可。

例如，代码清单 2-4 显示了如何利用<base>标签显示 URL 地址为 http://www.apress.com/images/logo.gif 的图像。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xHTML1/DTD/xHTML1-strict.dtd">
<html>
<head>
    <base href="http://www.apress.com/images/"/>
</head>
<body>
    <img src="logo.gif" alt="The Apress logo"/>
</body>
</html>
```

代码清单 2-4　与图像配合使用的<base>标签

当浏览器准备获取图像时，它取出<base>标签中指定的基础 URL，将其与所请求的文件相结合（严格地说，是与<img>标签的 src 属性值相结合）。当使用同一个 URL 地址下的多个图像时，使用<base>标签是一种高效的手段，因为它可以节省输入时间，并减小文件的总体占用空间。

2.2.4 ＜style＞标签

＜style＞标签唯一的用途就是为文档创建内部样式表。

1. 必需属性

type：定义样式类型。其值总是设置为 text/css，除非使用某种专属的样式语言，但这种情况是应当尽量避免的。

2. 可选属性

media：说明样式影响的是哪种媒体。常见的值包括 screen、print、tty、tv、projection、handheld、braille、aural、all。all 是未指定 media 属性时所采用的默认值。

3. 标准属性

该标签的标准属性包括 dir、lang、title 和 xml：space。

media 属性可以针对不同的输出设备使用不同的样式。例如，某人想打印一份在线报告，想让屏幕上显示的文本比打印页面上的更大一点，而且使用不同的字体。代码清单 2-5 中的代码可以实现上述功能。

```
＜style type＝"text/css"＞
    @media screen
    {
        ptext {font-size：16px}
    }
    @media print
    {
        ptext {font-size：12px}
    }
    @media screen，print
    {
        ptext{font-weight： normal}
    }
＜/style＞
```

代码清单 2-5　使用了几种不同媒体类型的＜style＞标签

代码清单 2-5 所示的样式表中，屏幕字体大小被设置为 16 px（像素），打印页面的字体被设置为 12 px，屏幕和打印媒体的字体粗细都被设置成了 normal。注意其中的@media 的规则，使用它可以在一个样式表中指定多种媒体类型。

2.2.5 ＜link＞标签

＜link＞标签用来定义两个链接在一起的文档之间的关系，常用于把外部样式表链接到当前文档。

1. 必需属性

该标签没有必需属性。

2. 可选属性

(1)charset:设置所链接的文档使用的字符集。

(2)href:指向所链接的文档的 URL。

(3)media:说明所链接的文档用于哪种媒体。常见值包括 all、braille、print、projection、screen、speech。使用 media 属性可以针对不同的媒体类型指定不同的样式表。

(4)rel:定义所链接到的文档与当前文档之间的关系。常见值包括 alternate、appendix、bookmark、chapter、contents、copyright、glossary、help、home、index、next、prev、section、start、stylesheet、subsection。

(5)rev:与 rel 相反,此属性定义当前文档与所链接到的文档之间的关系。

(6)type:指定目标 URL 的多用途网际邮件扩充协议(MIME)类型。常见值包括用于外部样式表的 text/css、用于 JavaScript 文件的 text/javascript 和用于 GIF 图像文件的 image/gif。MIME 类型告诉浏览器所下载的文件的类型是什么以及应该如何处理。

3. 标准属性

该标签的标准属性包括 class、dir、id、lang、style 和 title。

下面代码显示如何链接到一个外部样式表——这是<link>标签的一个常见用途。

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

课堂实训 2-2　了解意想不到的<head>标签效果

1. 实训内容

使用 Dreamweaver 软件创建网页,利用<meta>标签实现 Web 页的自动跳转。通常 Web 页上需要浏览者自己选择浏览的内容,如果想自动更换显示内容,可以用<meta>标签实现,更换的时间和跳转的目的地都可以自行设定。页面从(a)跳转到(b)的效果如图 2-5 所示。

(a) 跳转前的页面　　(b) 跳转后的页面

图 2-5　利用<meta>标签实现页面自动跳转的显示结果

2. 实训目的

掌握网页的基本结构及利用<meta>标签实现 Web 页的自动跳转。

3. 技能分析

(1)熟悉 XHTML 的基本结构。

(2)了解 XHTML 标签的含义。

(3)利用<meta>标签实现 Web 页的自动跳转。

4. 实训步骤

具体的代码如代码清单 2-6 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xHTML1/DTD/xHTML1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xHTML">
<head>
    <meta http-equiv="Content-Type" content="text/HTML; charset=utf-8" />
    <meta http-equiv="Refresh" content="5;url=http://www.w3school.com.cn"/>
    <title>head 标签的特殊效果</title>
</head>
<body>
    <p>对不起。我们已经搬家了。您的 URL 是
    <a href="http://www.w3school.com.cn">http://www.w3school.com.cn</a>
    </p>
    <p>您将在 5 秒内被重定向到新的地址。</p>
    <p>如果超过 5 秒后您仍然看到本消息,请单击上面的链接。</p>
</body>
</html>
```

代码清单 2-6　利用<meta>标签实现页面自动跳转

上面的代码说明如下。

(1)第 1 行代码指出了本网页的超文本类型是可扩展超文本过渡文档类型(XHTML 1.0 Transitional)。

(2)第 4 行～第 8 行代码设置了网页的“头”(head)。其中,第 5 行代码 http-equiv="Content-Type"设置的是 HTTP 的头部协议,提示浏览器网页的信息。第 6 行代码<meta http-equiv="Refresh" content="5;url=http://www.w3school.com.cn"/>利用<meta>标签实现 Web 页的自动跳转。content="5"指定刷新的时间为 5 s;url=http://www.w3school.com.cn"指定刷新跳转的目标页面。第 7 行代码指出了网页的标题(title)。

(3)第 9 行～第 15 行代码为网页的主体部分(body)。其中,第 10 行～第 14 行代码使用了段落标签<p>显示几行提示信息。第 11 行代码中还使用了<a>标签设置链接,以便由于某种原因未能实现自动跳转时,用户可单击链接文字实现页面跳转,href 属性指定跳转的目标页面的 URL 为 http://www.w3school.com.cn。

5. 实训提高

(1)如何在 Web 页上显示一段欢迎信息,隔一定时间后自动跳转到其他的 Web 页面?

提示: 在<head>与</head>标签之间加上下面的语句:

<meta http-equiv="Refresh" content="秒数;url=跳转的文件或地址"/>

(2)如何实现多个内容相似而背景不同的页面每隔一定时间自动相互跳转?

2.3 丰富的<body>标签

2.3.1 文字的变化效果

1. 段落标签

段落是文章中最基本的单位,是文章思想内容在表达时由于转折、强调、间歇等情况造成的文字停顿,具有换行的作用。制作网页时,可以使用 标签标记每一个段落的边界,以此告诉 Web 浏览器如何分段。代码清单 2-7 显示了 XHTML 中的两个段落。段落的开头用开标签 表示,结尾用闭标签 表示。标签之间的空行不是必须有的,但它们可以让标记代码在编辑时更具有可读性。段落是块级标签,只允许包含文本和行内标签。

```
<p>Spaghetti and cruft opened our doors in 1999,bring great pizza and pasta to
    the heart of the city's trendy Riverbend district. We handcraft our pizzas
    on the spot using only the best ingredients,and then we bake them to perfec-
    tion in our rustic wood-fired brick oven. We sell pizza by the slice or by the
    pie and even offer catering for any occasion all around the neighborhood.
</p>
<p>Our broad menu of pasta dishes puts a modern twist on Old Italia,served in
    heaping bowlfuls sure to satisfy any appetite(though we bet you'll want
    seconds anyway). But it's not all noodles and crust at Spaghetti and Cruft;
    we also have fresh veggie sides,an all-you-can-eat salad bar, and the best
    cannolis in town!</p>
```

代码清单 2-7　两个样例段落

图 2-6 所示为这些段落在浏览器中的显示效果。因为 标签是块级标签,所以每个段落都从一个新行开始,段落的后面跟着一个空行的空白间隔。许多 Web 设计人员会在文档中插入一些空段落( ),以在页面上添加一些额外的纵向间隔。应该避免使用这种表现性的标记代码——空段落没有任何含义。如果需要在页面布局中添加一些纵向空白间隔,应该使用 CSS。

图 2-6　浏览器把两个段落呈现为分开的块

2. 标题标签<h1> ～ <h6>

标题用于引用新一节的内容。XHTML 提供了 6 个级别的标题标签，用来表示标题的相对重要性或它在文档层次体系中的级别(及尾随其后的相关内容的重要性或级别)。可以把文档组织为一份简单的大纲，把它分成一些特定的主题或关注领域，按重要性顺序排列，每一节都包含自己的小节。

代码清单 2-8 显示了一些标记为标题和段落的内容，每个标题都说明了跟随其后的内容。不同的标题级别意味着不同的重要性层次。顶级标题引出整个节，其下的小标题则引出其中的小节。

```
<h1>Praise for Spaghetti and Cruft :Geek Pizzeria </h1>
<p>See what people are saying about us!</p>

<h2>Customer feedback </h2>
<p>Our loyal Customers love us (and we love them).</p>

<h2>Reviews</h2>
<p>Even those stuffy restaurant critics can't resit our charms.</p>
```

代码清单 2-8　标题和段落的混合举例

标题标签也是块级标签，只能包含文本或行内标签。<h1>标签用来标明顶级标题——页面上最重要的标题。因为逻辑上只能有一个“最重要的”标题，所以习惯上一个文档中<h1>只出现一次，通常用于标记网站的名称或所浏览的网页的标题。这并不是 XHTML 的要求，只是一条良好的语义经验法则。所有标题都应该保持正确的顺序，例如，<h5>不应该出现在<h2>之前，除非这样做有特殊含义。

图 2-7 所示为代码清单 2-8 的标记代码在浏览器中呈现的结果。在大多数的图形化 Web 浏览器中，标题都会自动用粗体字显示，并且不同级别的标题使用不同大小的字体，<h1>最大，<h6>最小。由于有这样一种默认样式，标题标签在过去经常被滥用。这种做

法是不可取的，标题应该按有意义的方式使用。<h2>应该是“第二重要的标题”，而不是“第二大的字体”。欲控制字体的大小，可以用 CSS 改变默认的外观大小。

图 2-7　默认不同级别的标题使用不同大小的字体显示

3. 文字标签

标签本身并没有什么意义，必须通过属性的设置后才有意义。文字的字体、大小和颜色都可以通过这个标签设置，方法是：在一段文字中将所要改变的文字利用标签声明，并在文字变化结束时加上标签。

1）文字的大小设置

标签格式如下：

```
<font size="N">…</font>
```

其中，N 的有效值范围为 1～7，即文字大小分 7 个等级。

另一种格式如下：

```
<font size=±"N">…</font>
```

size 的默认值为 N，可以用“±”来对默认值进行加减。

代码清单 2-9 显示了使用<font>标签设置文字的大小。

```
<font size=1>字体一</font>
<font size=－2>字体一</font>
<font size=2>字体二</font>
<font size=－1>字体二</font>
<font size=3>字体三</font>
<font size=＋0>字体三</font>
```

代码清单 2-9　使用<font>标签设置文字的大小

2）文字的颜色设置

标签格式如下：

```
<font color="属性值">…</font>
```

属性值可以为颜色码＃RRGGBB 或者颜色名称。＃RRGGBB 代表的是红、绿、蓝三原色，每一种颜色由两位十六进制的数值表示。

代码清单 2-10 显示了使用<font>标签设置文字的颜色。

```
<font color=FF0000>红色的字</font>
<font color=YELLOW>黄色的字</font>
<font color=BLUE>蓝色的字</font>
```

代码清单 2-10　使用<font>标签设置文字的颜色

3)文字的字体设置

标签格式如下：

```
<font face="字体">…</font>
```

字体可以是本地计算机上的任意一种字体类型，但是只有对方的计算机中装有相同的字体时，才可以在对方的浏览器中显示预先设计的风格。如果对方的计算机中无相关字体，浏览器仍会以宋体显示。

代码清单 2-11 显示了使用<font>标签设置文字的字体。

```
<font face="楷体_GB2312">楷体_GB2312</font>
<font face="黑体">黑体</font>
```

代码清单 2-11　使用<font>标签设置文字的字体

4. 其他辅助标签

1)<pre>标签

在浏览器呈现文档时，其中的空白会发生“缩合”，连续的多个空格会被缩减为一个，回车会被忽略。这时可以使用<pre>标签定义一段预先格式化过的文本，就能使其中的空白和换行按其在标记代码中的原样保留。该标签在显示计算机代码和诗歌时特别有用，因为这些内容中的换行和缩进很重要。代码清单 2-12 显示了一段用<pre>标签标记的诗句。

```
<pre>
        醉花阴
薄雾浓云
              愁永昼
        瑞脑
消金兽。
</pre>
```

代码清单 2-12　一段用<pre>标签标记的诗句

<pre>标签是块级标签，只能包含行内标签。它的内容通常被默认地呈现为等宽字体，如图 2-8 所示。

图 2-8　内容被呈现时保持了预先的格式

2)＜em＞标签

＜em＞标签强调一个词或短语。可视化 Web 浏览器往往将＜em＞标签包含的内容显示为斜体，但其他的设备可能会用不同的方式表现强调。例如，供有视力障碍的人使用的屏幕阅读软件会用一种不同的声调大声念出＜em＞标签包含的内容。

3)＜strong＞标签

有些词语或短语要求强调的程度比＜em＞标签所能提供的更高，这时可以使用＜strong＞标签对它们进行着重强调。图形化浏览器把＜strong＞标签中的文本显示为粗体，但其他的设备可能会用不同的方式表示强调。

代码清单 2-13 显示了一段文本，其中对一些短语进行了强调。为了达到更好的强调效果，在代码中组合使用了＜strong＞标签和＜em＞标签，大多数浏览器会将其中包含的内容显示为斜体加粗体的形式。图 2-9 所示为该代码在浏览器中的呈现效果。

```
＜p＞A traditional pizza is round. Not only ＜em＞should＜/em＞ a pizza be round,
But a proper pizza ＜strong＞must＜/strong＞ be round. To reiterate,
＜strong＞＜em＞real pizzas are round＜/em＞＜/strong＞. Except when they're not.＜/p＞
```

代码清单 2-13　包含一些被强调的短语的段落

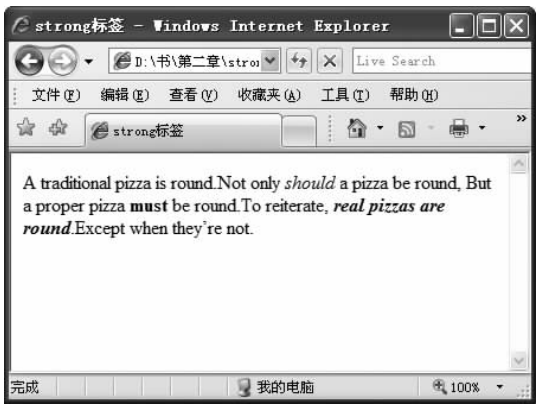


图 2-9　＜em＞标签和＜strong＞标签混合使用的效果

4)＜i＞标签和＜b＞标签

＜i＞标签可使其中的文本呈现为斜体，而＜b＞标签可使其中的文本呈现为粗体。让文本显示为斜体或粗体，目的基本上都是想进行强调。强调文本应该优先使用＜strong＞标签和＜em＞标签。

5)＜big＞标签和＜small＞标签

包含在＜big＞标签中的文本显示时，使用的字体将比其外围文本稍大一些，包含在＜small＞标签中的文本显示时，使用的字体则比其外围文本稍小些。这些标签没有什么别的语义，其表现性效果最好用更富有意义的标签辅以 CSS 样式来实现。

6)＜sup＞标签和＜sub＞标签

有时需要在文本中加入上标或下标字符，特别是所撰写的文档包含数学公式、化学符号或某种语言(如法语)本身需要上标和下标时。在此情况下，可以使用＜sup＞标签和＜sub＞标签，它们分别代表上标和下标。上标文本的位置比周围文本略高，而下标文本则比周围文本略低。代码清单 2-14 为＜sup＞标签和＜sub＞标签示例：＜sup＞标签用在直

角三角形边长的勾股定理公式中，<sub>标签用在硫酸的化学式中。图 2-10 显示了该代码在浏览器中的呈现效果。

```
<p>a<sup>2</sup>+b<sup>2</sup>=c<sup>2</sup></p>
<p>H<sub>2</sub>SO<sub>4</sub></p>
```

代码清单 2-14　<sup>标签和<sub>标签示例

图 2-10　<sup>标签和<sub>标签的使用效果

2.3.2　列表的配合使用

列表是两个或更多相关项的集合。只包含一个项的列表也完全有效，在某些情况下甚至在语义上也是正确的，但通常被一个列表组织在一起的项有多个。XHTML 中有 3 种类型的列表：无序列表、有序列表和自定义列表。

1. 无序列表

无序列表用
标签表示，用于项的顺序不太重要的列表，如食谱中的配料列表。每个列表项又由
标签定义，它们都包含在外围的
和
标签中。<ul>标签是块级标签，其子只能是<li>标签，文本和其他标签都不能出现在无序列表中，除非它们被包含在一个<li>标签中。

在无序列表和后面要介绍的有序列表中，都将使用块级标签<li>定义各个项。列表项中可以包含文本或其他标签，甚至可以包含新的列表。

代码清单 2-15 展示了配置比萨面团的原料组成的无序列表。

```
<ul>
    <li>1 cup warm water</li>
    <li>1 packet active dry yeast</li>
    <li>2 1/2 to 3 cups all-purpose flour</li>
    <li>2 tablespoons olive oil</li>
    <li>1/2 teaspoon salt</li>
</ul>
```

代码清单 2-15　烹调配料的无序列表

在图形化浏览器中，无序列表的默认显示形式为带有一点缩进，且每个列表项前都有一

个圆形符号。图 2-11 显示了该代码在浏览器中的呈现效果。

图 2-11　无序列表的浏览器显示效果

2. 有序列表

标签用于定义有序列表。有序列表中的项应该按特定顺序安排，如食谱中的操作步骤。代码清单 2-16 展示了一个有序列表的例子。下面的 XHTML 标记代码中并未对各项进行编号。

```
<ol>
    <li>做面坯</li>
    <li>放料</li>
    <li>烤制</li>
</ol>
```

代码清单 2-16　有序列表代码示例

在图形化浏览器中，有序列表的默认显示形式为带有一点缩进，且每个列表项都按顺序编号。图 2-12 显示了该代码在浏览器中的呈现效果。

图 2-12　有序列表的浏览器显示效果

3. 自定义列表

自定义列表不仅仅是项的集合，更是项及其说明的集合。不像有序列表和无序列表那

样包含列表项标签<li>，自定义列表中的项由定义术语（用<dt>标签表示）或定义说明（用<dd>标签表示）组成，或两者共同组成。一个术语可以有多条相关的说明，一条说明也可以应用于在它之前出现的一组术语。划分自定义列表各部分的方法是在一个</dd>标签之后紧接着一个<dt>标签，表示一个新的术语和说明序列的开始。

在术语和它的说明之间有一种隐含的语义联系。<dt>标签和<dd>标签彼此捆绑在一起，配对形成列表的结构。因为这种语义上的共生关系，自定义列表有时被作为特殊表现形式的标记。一系列问题及其答案、一组图像及其标题、用发言者的名字及其发言表示的一系列对话，都是自定义列表潜在的用武之地。

1）<dl>标签

<dl>标签用于创建一个自定义列表。它是块级标签，必须至少包含一条术语<dt>或一条说明<dd>。<dl>标签之子只能是<dt>标签和<dd>标签。

2）<dt>标签

<dt>标签是块级标签，只能包含文本和行内标签。它标明了一个术语或所要说明的对象。一条定义术语关联着紧随其后的每一条说明，直到有另一个<dt>标签开始新的序列（或直到列表结束于其闭标签</dl>）。

3）<dd>标签

<dd>标签包含对紧靠其前面的<dt>标签的说明。在一条术语有多条说明的情况下，每一条说明都应该包装在自己的<dd>标签中。<dd>标签是块级标签，可以包含文本、行内标签和其他块级标签。如果说明分为多个段落，应该把它们标记为段落<p>，但应该包含在一个<dd>标签中，而不是分成多个<dd>标签。一个<dd>标签的全部内容应该构成一条说明。

代码清单 2-17 展示了一个简短的自定义列表的标记代码。在此例子中，第一条术语的说明包含两个段落，而第二条术语有两条不同的说明。

```
<dl>
<dt>Pizza </dt>
<dd>
        <p> A flat,open-faced baked pie of Italian origin,consisting of a layer
        of bread dough covered with tomato sauce,cheese and a wide variety of op-
        tional toppings.</p>
        <p>Also called <em>pizza pie</em>.</p>
</dd>
<dt>Pasta</dt>
<dd>Unleavened dough that is molded into any of a variety of shapes and boiled.</dd>
<dd>A prepared dish containing pasta as its main ingredient.</dd>
</dl>
```

代码清单 2-17　包含两条术语的自定义列表

自定义列表的浏览器显示效果如图 2-13 所示。在大多数浏览器的显示结果中，<dd>标签相对于相关的<dt>标签都有少许缩进。如果<dd>标签包含有其他结构性标记代码（如段落<p>），那么嵌套标签的默认边距将会被应用。从图 2-13 中可以看到，第一条术语

的说明中的段落的上下方都有空白间隔，而第二条术语的两条说明则没有。当然，可以用 CSS 改变这种呈现效果。

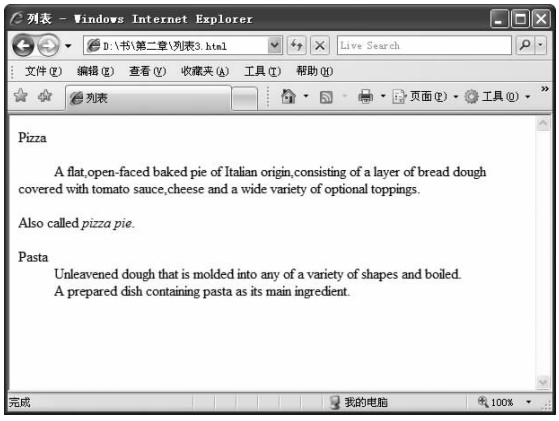


图 2-13　自定义列表的浏览器显示效果

4. 列表的树形结构及应用范围

当一个列表嵌套在另一个列表中时，默认情况下，内层列表的样式将根据其嵌套级别而有所不同。代码清单 2-18 为一个复杂的无序列表示例。

```
<h2>Specialty Pizzas</h2>
<ul>
    <li>
        <h3>Barbecue Chicken Pizza</h3>
        <p>This hearty American departure from Italian tradition is one of
    our most popular pizzas.</p>
         <ul>
           <li>Spicy barbecue sauce.</li>
           <li>Chunks of mesquite grilled chicken.</li>
           <li>Blend of three cheeses:
               <ul>
                   <li>Mozzarella</li>
                   <li>Monterey Jack</li>
                   <li>Smoked Gouda</li>
               </ul>
           </li>
           <li>Thin-sliced red onion.</li>
           <li>Roasted red peppers.</li>
        </ul>
    </li>
</ul>
```

代码清单 2-18　一个复杂的无序列表示例

图 2-14 所示为该代码在浏览器中的呈现效果。可以看到，每层嵌套的列表都相对于前面的层次缩进了一些，并且标记符号的样式也各不相同。

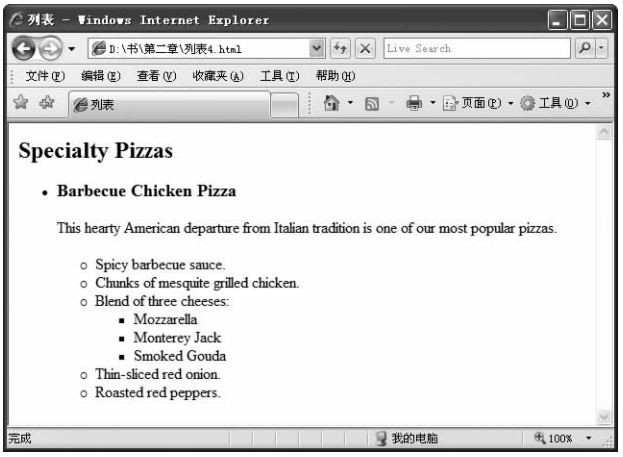


图 2-14　树形列表在浏览器中按默认样式显示的结果

2.3.3　富有感染力的图像的使用

网页的文本内容是 XHTML 文档的一部分，它们被一系列标签包围着。相比之下，图像属于外部文件，它不是文档的一部分。在 XHTML 文档中通常使用＜img＞标签（或＜object＞标签）引用图像。呈现一个包含图像的网页时往往分为两个阶段，即先下载标记代码，然后再下载外部图像。只要＜img＞标签出现在文档里，浏览器就会从 Web 服务器获取它所引用的文件，并将其呈现在页面上以替代该标签。

1. 图像标签＜img＞

图像标签＜img＞被认为是一种替换标签，因为它呈现的并不是标签本身。＜img＞标签也是没有任何文本内容的空标签，所以必须用一个结尾斜线（/＞）自闭合。

＜img＞标签需要使用 src 属性声明图像文件的来源，其值是一个指向该文件在 Web 服务器上的位置的 URL（绝对的和相对的均可）。alt 属性也是一个必需属性，它提供了与图像等价的替代性文本，即如果无法获得图像或者浏览器不能显示图像时，那么将显示它所指定的替代性文本。代码清单 2-19 显示了一个只包含 src 属性和 alt 属性的＜img＞标签，这两个属性是保证＜img＞标签有效性的基本要求。

```
＜img src="/images/pizza.jpg" alt="A pizza with sausage and olives"/＞
```

代码清单 2-19　＜img＞标签的最简单形式

1）必需属性

（1）src：指定一个 URL，它指向图像文件在 Web 服务器上的位置。

（2）alt：提供与图像等价的代替性文本。

2）可选属性

（1）width：指定图像的宽度，单位为像素。

（2）height：指定图像的高度，单位为像素。

(3)ismap:声明图像要被用于服务器端图像映射。

(4)usemap:用于明确所使用的客户端图像映射。

(5)longdesc:指定指向包含对图像的详细说明的文档的 URL。

2. 背景中的图片

使用 CSS 的 background-image 属性可以向页面中加入装饰性的图像,而且不会把表现性的因素与内容混在一起。XHTML 中几乎所有标签都能设置背景图像,标签的内容就覆盖在背景之上。在默认情况下,背景图像将在纵横两个方向平铺,从标签左上角开始,在纵向和横向反复绘制,直到填满标签拥有的区域,这就像居室地板上铺的地砖。代码清单 2-20 展示了一条为＜body＞标签设置背景图像的 CSS 规则。其中,图像通过其 URL 指定,而 URL 包含在一对括号中,前面用关键字 url 说明。

```
body{
    background-image:url(/images/background.gif);
}
```

代码清单 2-20　为＜body＞标签设置背景图像

图 2-15 所示为该代码在浏览器中的呈现效果,图像铺满了整个文档窗口。

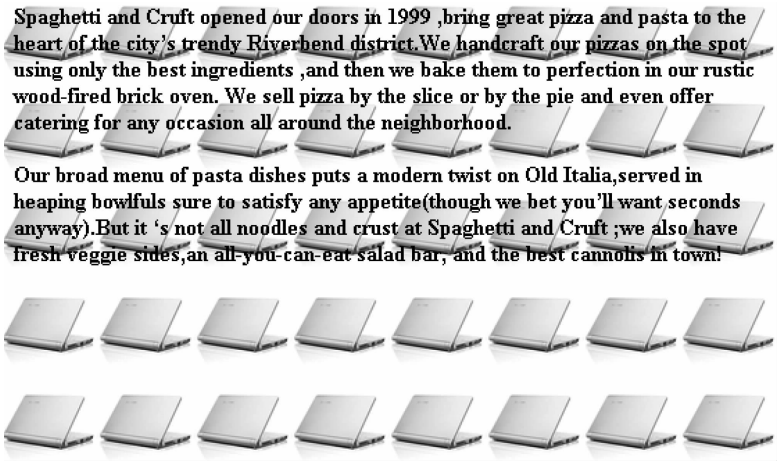


图 2-15　为＜body＞标签设置背景图像的显示效果

CSS 的 background-repeat 属性可以改变默认的平铺方式,可设置图像只在横向平铺、只在纵向平铺或根本不平铺。代码清单 2-21 扩充了前面的 CSS 规则,声明背景图像沿 *X* 轴横向平铺。

```
body{
    background-image:url(/images/background.gif);
    background-repeat:repeat-x;
}
```

代码清单 2-21　背景图像只在横向平铺

图 2-16 所示为该代码在浏览器中的呈现效果，图像只在页面的顶端重复绘制，并未向下重复绘制。

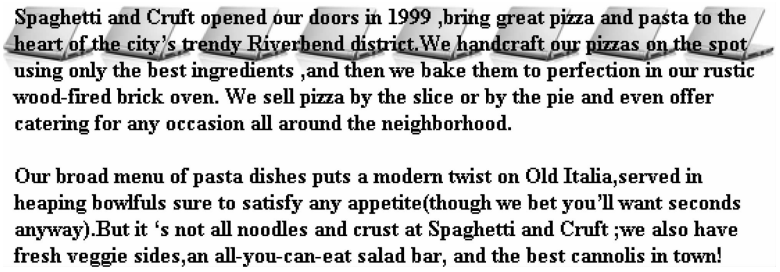


图 2-16　背景图像只在横向平铺的显示效果

类似的，设置 background-repeat 属性为 repeat-y，将使图像只在纵向平铺。background-repeat 属性的默认值是 repeat，如果有必要，可以用它覆盖别的规则中指定的其他值。也可以指定 background-repeat 属性的值为 no-repeat，从而完全禁止平铺。

如果背景图像比它所装饰的标签大，那么标签的区域就像是一个用于界定背景中可见部分的窗口。如果由于加入了更多内容、增大了文字的尺寸或者用 CSS 改变了标签的尺寸等使标签发生扩张，那么图像中将有更多的部分显露出来。

如果背景图像比较复杂，或者背景颜色与前景颜色之间的对比不够明显，那么文字会显得难以阅读。所以，在使用背景图像时要小心，不要过多地影响内容的可阅读性。

2.3.4　超链接的应用

1. 超链接

链接是超链接的缩写，英文是 hyperlink。XHTML 中的 H 就是 hypertext 的缩写，是超文本的意思。超文本链接语言（网页）的精髓就是链接，通过链接才可以把世界各地的网页链接到一起，成为互联网。链接是使万维网影响力巨大的最重要的特性之一。使用链接，用户可以从自己的文档指向其他文档、图像和程序。默认情况下，链接被显示为文档中的一串带下划线的文本。当鼠标指针停留在链接上方时，指针会从默认形状变为另一种形状，以表示这是一个链接。当单击某链接时，浏览器会导航到链接所指向的位置。

2. 超链接标签<a>

超链接的使用基于<a>标签。这个标签的属性不多，但它提供了许多功能。超链接标签<a>的基本语法如下：

显示的文字

其中，href 是链接的属性，告诉浏览器链接到的网址。url 是要链接到的网页或者文件，可以是一个绝对的网页，如 http://www. dreamdu. com/xHTML/，或者是一个相对网页。在<a>和之间的文本用于向用户显示。在可视化浏览器中，这些文本通常具有下划线；在其他类型的浏览器中，链接的提示线索有所不同，具体取决于用户代理本身。当用户单击链接时，浏览器会被导向 href 属性所指定的文档。

1)必需属性

<a>标签没有必需属性。

2)可选属性

(1)charset:说明目标 URL(所指向的文档)使用的编码字符集。

(2)coords:用于在客户端图像映射中定义一个形状的坐标。

(3)href:指定当用户单击链接时被打开的 URL,是<a>标签最常用的属性。

(4)hreflang:说明由 href 属性指定的 URL(所指向的文档)所使用的基础语言。

(5)rel:说明当前文档和目标 URL 之间的关系。可能值包括 alternate、stylesheet、start、next、prev、contents、index、glossary、copyright、chapter、section、subsection、appendix、help、bookmark。

(6)rev:说明目标 URL 和当前文档之间的关系。可能值包括 alternate、stylesheet、start、next、prev、contents、index、glossary、copyright、chapter、section、subsection、appendix、help、bookmark。

(7)shape:定义图像映射中,当前<a>标签对应的映射区域的类型。可能值包括 circle、default、poly、rect。

(8)type:说明目标 URL 的 MIME 类型。

3. 超链接的定义和应用

1)链接到其他文档

<a>标签最常见的用法就是链接到另一个文档,这种用法的示例如代码清单 2-22 所示。

```
<!DOCTYPE HTML PUBLIC"-//W3C//DTD XHTML 1.0 Strict//EN"
http://www.w3.org/TR/xHTML 1/DTD/xHTML 1-strict.dtd>
<html>
    <head>
        <title>Using Links</title>
    </head>
    <body>
        <p>
            <a href="http://www.google.com">Search</a>
        </p>
    </body>
</html>
```

代码清单 2-22　使用<a>标签

这个例子使用了一个带 href 属性的<a>标签,功能是让用户能够使用 google 进行搜索。单击开标签<a>和闭标签之间的文本或图像,则激活链接。如果当前焦点在链接标签上,按 Enter 键也可以激活链接。href 属性提供了浏览器应该导航的位置(本例中是 http://www.google.com),程序运行结果如图 2-17 所示。

图 2-17　一个简单的网页中的<a>标签

代码清单 2-22 提供了指向所请求的文档的完整 URL，该 URL 描述了一个绝对路径。使用绝对路径链接，适用于要链接到的文档与当前文档存储在不同的位置的情况，以便浏览器知道究竟该到什么地方获取这个文档。绝对路径至少应该提供域名，此外还可以包含目录和所要链接的特定文件(文档)的名称。

当要链接的文档是本域中的其他文档时，则可以使用相对路径。相对路径是一种简化的方法，不必指定想要链接到的文档的域名，也可不必指定其目录路径。相对路径基于原文档所在位置，当要链接的文档与原文档位于同一目录时，则不必包含路径；当要访问原文档所在目录的子目录下的文档时，只需在子目录名后面加一个斜线再加文档名即可，如 images/filename；当要访问当前文档所在目录的上一级目录中的文档或文件时，可以使用两个句点来代表上一级目录，如../filename。在代码清单 2-23 中，href 属性并未包含任何域名或文件路径信息。

```
<!DOCTYPE HTML PUBLIC"-//W3C//DTD XHTML 1.0 Strict//EN"
http://www.w3.org/TR/xHTML 1-strict.dtd>
<html>
    <head>
        <title>Using Links - Relative Links</title>
    </head>
    <body>
        <p>
          Options:<br/>
          <a href="home.HTML">Home</a><br/>
          <a href="news.HTML">News</a><br/>
          <a href="menu.HTML">Menu</a><br/>
          <a href="locations.HTML">Locations</a>
        </p>
    </body>
</html>
```

代码清单 2-23　使用相对路径

代码清单 2-23 的文档中添加了多个链接，为用户提供了多种选择，但是浏览器是如何找到名为 home. HTML 的文档的呢？这是由于使用了相对路径，浏览器会认为：既然没有额外提供一个位置以供查找文档，那么这个文档一定位于当前文档所在的域和文件路径。

提示：使用相对路径在网站中的文档之间导航是个好主意。这样，以后若想改变文档的位置（这会导致 URL 中的一些部分发生改变），修改起来要容易得多。

当网站规模越来越大，并且包含 XHTML、图像和 CSS 等各种类型的文件时，可能会将这些文件分门别类地放在不同的文件夹中，以增强其条理性。这时使用相对路径可能会遇到一些麻烦，因为有些文件可能存放在与文档所在目录不同的其他目录中。这种情况可以使用“../”指示浏览器在目录路径中向上移动一级，例如：

```
<a href="../home/index.HTML">Home page</a>
```

根据上面代码中指定的 href，浏览器将从当前文档所在的目录级别上移一级，然后再转到 home 文件夹中找到并加载 index. HTML 文档。

2）链接到指定位置

所谓指定位置的链接，就是直接链接到文档中一个定义好的位置。指定位置的链接通常用在文章较长、一屏无法显示的情况下，也可以直接链接到其他页面的指定位置。下面的代码为链接到指定位置的示例：

```
<h2 id="top">页面上部</h2>
<a href="#bottom">链接到页面下部</a>
<a href="#top">链接到页面上部</a>
<h2 id="bottom">页面下部</h2>
```

上述代码给 XHTML 文档顶部的标签<h2>加了一个 id 属性，又在 XHTML 文档的底部加了一个链接<a href="#top">链接到页面上部。代码运行后，只要单击这个链接，就可以直接转到具有 id="top"的标签的位置（也就是顶部）。

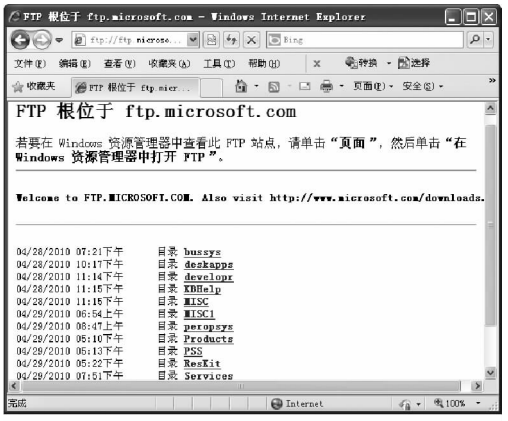
3）链接到非 XHTML 文档

可以建立到因特网上的非 XHTML 数据的链接。一种常见用法是提供到 FTP 站点的链接，以便用户下载文件。例如，下面的链接能让用户通过浏览器访问微软的 FTP 站点：

```
<a href="ftp://ftp.microsoft.com">Microsoft FTP Site</a>
```

当用户单击这个链接时，将被链接到微软的 FTP 站点，如图 2-18 所示。

(a)



(b)

图 2-18　正在访问微软的 FTP 站点的用户浏览器

这样的技术也可以用于链接因特网上的其他非 XHTML 文件。例如，如果一个.pdf 文档想通过网站提供给外界，那么只需使用一个类似下面的链接即可：

```
<a href="menu.pdf">Download Menu in PDF(500KB)</a>
```

说明：如果所链接的文档不是一个 HTML/XHTML 文档，那么把它的类型和大小明确地告诉用户是一个很好的做法。上面的例子中就说明了链接的目标是一个容量为500 KB 的.pdf 文档。这些信息便于用户判断自己的计算机是否支持这种文件以及是否有足够的空间容纳这个文件。

当用户单击 Download Menu in PDF(500KB)这个链接时，浏览器可提供下列选择中的一个或全部。

(1)把文件保存到计算机。

(2)打开文件。

如果用户选择保存文件，浏览器将提示用户指定一个保存位置，然后再下载文件，并将其保存在本地计算机上用户指定的位置。如果用户选择打开文件，浏览器将获得 menu.pdf 文件，并根据其文件扩展名决定使用何种应用程序打开它。如果浏览器不知道应该如何处理这个文件，它将会显示某种错误信息，具体取决于系统平台，它也可能会提示用户确定与这种文件相关联的应用程序。

注意：下载直接链接到某些类型的文件(如.exe 文件)时一定要小心，因为有些居心叵测的人会把计算机病毒或破坏性的文件放在网站上。

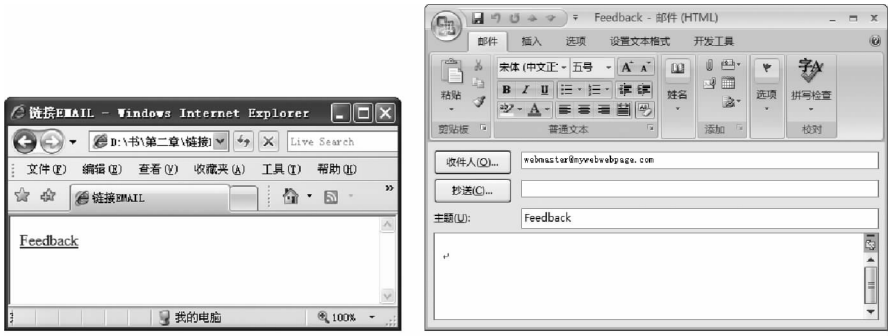
4)链接到电子邮件地址

<a>标签还有一个常见的用途，就是自动链接到电子邮件地址。这是一种确保用户的电子邮件发送到正确目的地的好办法。把<a>标签用于电子邮件的示例如下：

```
<a href=mailto:webmaster@mywebwebpage.com?subject=Feedback>Feedback</a>
```

默认情况下，当用户单击这个链接时，浏览器将打开默认电子邮件应用程序的新窗口。如果没有设置默认电子邮件应用程序，那么系统不会进行任何操作。在上述例子中，电子邮件地址 webmaster@mywebwebpage.com 会被自动插入电子邮件应用程序中的“收件人”一行。另外，如果电子邮件应用程序支持，“?subject＝Feedback”中等号后面的部分会被自动

插入电子邮件应用程序中的“主题”一行。本例中，放在“主题”行中的是 Feedback。图 2-19 所示为单击链接后打开的 Microsoft Outlook 窗口。



(a)　　(b)

图 2-19　单击链接后打开的 Microsoft Outlook 窗口

提示：链接到电子邮件地址时会带来一些问题，即如果把自己的电子邮件地址嵌入文档，那么任何阅读该文档的人都会看到这个地址。而有些人就会使用专门的程序收集这些电子邮件地址来发送垃圾邮件。如果真的希望能让用户使用电子邮件发送问题和评论，可以使用表单的方式联系，或者使用混淆技术迷惑收集电子邮件地址的程序。在决定是否应该把电子邮件地址直接嵌入文档时，应该权衡利弊。

2.3.5　简单表格和表单的创建

1. 表格的基础知识

表格在描述表格类数据时很有用。所谓表格类数据，是指最适合组织为表格格式（按行和列组织）的数据。在文档中创建一个简单的表格很容易。表格由 3 种标签组成：<table>、<tr>和<td>。代码清单 2-24 展示了表格基本的标签布局。

```
…
<table border="1">
<tr>
    <td>Row 1 Cell 1</td>
    <td>Row 1 Cell 2</td>
    <td>Row 1 Cell 3</td>
</tr>
<tr>
    <td>Row 2 Cell 1</td>
    <td>Row 2 Cell 2</td>
    <td>Row 2 Cell 3</td>
</tr>
</table>
…
```

代码清单 2-24　表格基本的标签布局

上述表格在可视化 Web 浏览器中的呈现结果如图 2-20 所示。可以看到，它生成了一组列和行，就像在电子表格程序中一样。

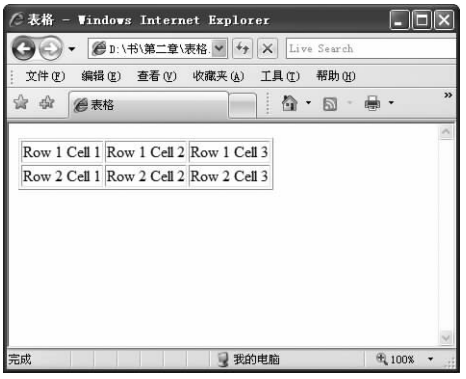


图 2-20　表格在 Web 浏览器中的效果

下面介绍用于创建表格的 3 个基本标签的属性。

1)＜table＞标签

＜table＞和＜/table＞标签标明了表格的开始和结尾，可以把表头、行、单元格和其他表格放在此标签中。＜table＞标签没有必需属性，可选属性如下。

(1)border：指定表格边框的宽度，单位为像素。

(2)cellpadding：指定单元格的四壁与其内容之间的距离，单位为像素或百分数。

(3)cellspacing：指定相邻单元格之间的距离，单位为像素或百分数。

(4)frame：说明表格的外边框应如何显示，与 border 属性配合使用。可能值包括above、below、border、box、hsides、lhs、rhs、void、vsides。

(5)rules：说明表格内部的分隔线应如何绘制，与 border 属性配合使用。可能值包含all、cols、groups、none、rows。

(6)summary：为提供语音合成和非视觉化表达能力的专用浏览器提供表格的概要说明。如果表格用于布局，则不应该使用 summary 属性，因为这将干扰使用非可视化浏览器的用户。

(7)width：指定表格的宽度，单位为像素或百分数。使用 CSS 样式中的 width 指定表格的宽度是更可取的方法。

2)＜tr＞标签

＜tr＞标签标明了表格中一个新行的开始。＜tr＞标签没有必需属性，可选属性如下。

(1)align：指定单元格中文本的对齐方式。可能值包括 center、char、justify、left、right。使用 CSS 样式中的 text-align 指定单元格内容的对齐方式是更可取的方法。

(2)char：说明文本应基于其中的哪一个字符对齐。要求同时使用 align 属性，并将其值设置为 char。

(3)charoff：指定用做对齐基准的字符在单元格中的偏移位置，单位为像素或百分数。要求同时使用 align 属性，并将其值设置为 char。

(4)valign：指定单元格中文本的垂直对齐方式。可能值包括 baseline、bottom、middle、top。

3)＜td＞标签

＜td＞标签在表格行中标明新单元格的开始。＜td＞标签没有必需属性，可选属性如下。

(1)abbr:为单元格中的内容指定一个文本形式的简短版本。对于比较长的内容,可以用这个可选属性向非可视化浏览器提供一个简短版本。

(2)align:指定单元格中文本的对齐方式。可能值包括 center、char、justify、left、right。使用 CSS 样式中的 text-align 指定单元格内容的对齐方式是更可取的方法。

(3)char:说明文本应基于其中的哪一个字符对齐。要求同时使用 align 属性,并将其值设置为 char。

(4)charoff:指定用做对齐基准的字符在单元格中的偏移位置,单位为像素或百分数。要求同时使用 align 属性,并将其值设置为 char。

(5)colspan:指定该单元格应跨越的列数。

(6)rowspan:指定该单元格应跨越的行数。

(7)scope:说明单元格提供的是否是针对同一行中其余单元格或针对同一列中其余单元格的标题信息。有效值包括 col、colgroup、row、rowgroup。

(8)valign:指定单元格内容的垂直对齐方式。可能值包括 baseline、bottom、middle、top。

2. 创建简单表格

1)＜caption＞标签

＜caption＞标签用于为表格指定一个标题。这个标题并不包含在某一行或某一单元格内,可视化浏览器会把其放置在表格上方。＜caption＞标签放在＜table＞标签后面。每个表格只能指定一个标题。

2)＜th＞标签

若想标明表格中的一些单元格是标题,而不是数据本身的一部分时,可以用＜th＞标签代替＜td＞标签将单元格标记为标题,为表格添加一个标题(header)行。在可视化浏览器中,标题单元格的内容通常呈现为粗体,并居中对齐。

＜th＞标签没有必需属性,其可选属性如下。

(1)abbr:为单元格中的内容指定一个文本形式的简短版本。

(2)align:指定单元格中文本的对齐方式。可能值包括 center、char、justify、left、right。

(3)char:说明文本应基于其中的哪一个字符对齐。要求同时使用 align 属性,并将其值设置为 char。

(4)charoff:指定用做对齐基准的字符在单元格中的偏移位置,单位为像素或百分数。要求同时使用 align 属性,并将其值设置为 char。

(5)colspan:指定该单元格应跨越的列数。

(6)headers:一组标题单元格 id,其中相邻 id 值用空格分隔。它们代表的标题单元格提供了当前数据单元格的标题信息。这个属性可以帮助非可视化用户代理表达数据单元格的标题信息。

(7)rowspan:指定该单元格应跨越的行数。

(8)scope:说明单元格提供的是否是针对同一行中其余单元格,或针对同一列中其余单元格的标题信息。有效值包括 col、colgroup、row、rowgroup。

(9)valign:指定单元格中文本的垂直对齐方式。可能值包括 baseline、bottom、middle、top。

代码清单 2-25 中创建了一个包含标题和标题单元格的表格,这个表格所列的是各种比萨饼的价格信息,并把比萨饼的价格与它们的形状和大小联系在一起。

```
＜!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xHTML1/DTD/xHTML1-strict.dtd"＞
```

```
<html xmlns="http://www.w3.org/1999/xHTML">
<head>
    <title>Using the table header attribute</title>
</head>
<body>
    <table border="1" summary="Prices for types of pizza by size">
        <caption>Our Pizza Selection </caption>
        <tr>
            <th scope="col">Pizza Type</th>
            <th scope="col">Small</th>
            <th scope="col">Medium</th>
            <th scope="col">Large</th>
        </tr>
        <tr>
            <td scope="row">Thin Crust</td>
            <td>3.99</td>
            <td>4.99</td>
            <td>6.99</td>
        </tr>
        <tr>
            <td scope="row">Deep Dish</td>
            <td>4.99</td>
            <td>6.99</td>
            <td>8.99</td>
        </tr>
        <tr>
            <td scope="row">Stuffed Crust</td>
            <td>5.99</td>
            <td>7.99</td>
            <td>9.99</td>
        </tr>
    </table>
</body>
</html>
```

代码清单 2-25　包含标题和标题单元格的表格

在上述代码中，用<table>标签标明表格的开始，然后用<tr>标签创建行。餐馆的每种比萨饼在表格中都占一行，每一行中各种价格显示在不同的单元格中。表格自动确定每个单元格的大小，默认情况下，单元格中的内容都居左对齐。代码中还使用<caption>标签为表格指定了标题，用<th>标签在表格中添加了标题单元格。

上述代码在浏览器中的显示效果如图 2-21 所示。从图中可以看出，用<th>标签标记的单元格显然是标题，因为它们的内容都用粗体显示，并且居中对齐。

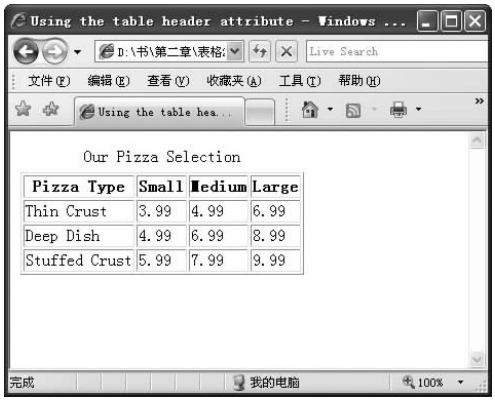


图 2-21　添加了标题及标题单元格的表格在浏览器中的效果

3. 表单的定义及工作原理

表单最简单的定义是网页上供用户输入信息的区域(有时表单标签用于信息显示,而不是收集信息)。访问者可以在空白区域中输入文字,在菜单中进行选择,然后单击按钮把这些信息发送出去以供处理。这些交互性的装置称为控件,它们的内容就是控件的值。

要修改控件的值,访问者必须先把焦点移到控件上,激活控件并使其准备接受输入。使控件获得焦点的方法是直接单击该控件,或按 Tab 键把光标从一个控件移到下一个控件。输入值的方法是输入文字或执行一些其他操作(如单击、按 Enter 键等)。

表单提交后,用户输入的信息作为一个表单数据集被传送到服务器,这个数据集由所有表单控件的名称和它们的值组成。数据集的处理工作由表单处理器负责,它是一段专为解释和使用表单提交的数据而设计的脚本或程序。许多表单处理器的作用是对输入的值进行验证,以确保需要的信息都已被输入,并且其格式符合要求。

处理提交的表单数据涉及脚本控制、编程、数据库设计和应用程序设计等复杂的事情,甚至还会涉及加密、个人隐私和安全问题,这些高级问题超出了 Web 前端设计的范围,本书不作介绍。

4. 表单的组成部分

整个表单都包装在<form>和</form>标签中。表单控件包括文本框、复选框、菜单和按钮等,访问者用它们输入信息或作出选择。提交表单时,所有控件的值都以名/值对的形式,作为数据集的一部分被发送给表单处理器。因此,每个控件都必须具有一个 name 属性,以便与其值配对。

1)<form>标签

<form>和</form>标签定义了 XHTML 文档中一个用于接受用户输入的部分。<form>标签是一个块级标签,是其他专用表单标签和其他用来建立表单的结构标签的容器,它的内容必须包含在各自的块级容器中。

<form>标签的必需属性如下。

action:指定表单处理器的 URL。表单处理器用于处理所提交的表单数据的脚本或应用程序。

<form>标签的可选属性如下。

(1)accept:包含一个用逗号分隔的 MIME 类型的列表。它表示使用表单上传文件时,可接受的文件类型。

(2)accept-charset:指定通过表单提交的数据可以使用的字符编码。如果不指定这个属性,则认为可使用的字符编码与表单所在文档的字符编码相同。

(3)enctype:指定用于提交表单的内容类型。这个属性的默认值是 application/x-www-form-urlencoded。如果所提交的表单包含用于上传文件的控件,那么这个属性应该设置为 multipart/form-data。

(4)method:指定提交表单数据时使用的发送方法,其值为 get 或 post,默认值是 get。如果 method 的值为 get,那么所提交的数据集将被作为一个由表单的所有名/值对组成的查询字符串,添加到表单处理器的 URL 的末尾。表单处理器能够解释并处理这个 URL,从查询字符串中提取控件值。如果 method 的值为 post,那么数据集将被直接发送给表单处理器,以便在服务器上进行处理。get 方法主要用来从服务器请求临时性使用的静态数据,特别是在某个 URL 可能会被重用时。post 方法主要用来把数据发送到服务器,并保存起来以供将来使用,或者用在由于安全或个人隐私等原因,而不想使用带有可见的查询字符串的 URL 的情况下。

代码清单 2-26 展示了一个简单表单的 XHTML 标记代码,其中的<form>开标签中包含了 method 和 action 属性。本示例中还包含两个<input>标签(一个文本框和一个提交按钮)和一个用<label>标签包装的文本。

```
<form method="post" action="/cgi-bin/formhandler.cgi">
    <p><label for="email">Enter your E-mail address to subscribe to our
mailing list.</label></p>
    <p><input type="text" name="email" id="email"/>
    <input type="submit" name="subscribe" value="Subscribe"/></p>
</form>
```

代码清单 2-26　创建一个简单的表单

该代码在浏览器中的显示效果如图 2-22 所示。

图 2-22　表单在浏览器中的显示效果

2)＜input＞标签

许多常见的表单控件都用行内标签＜input＞创建，其类型由 type 属性确定。因为＜input＞标签是行内标签，所以多个控件可以并排出现在一行上，但是它们都必须包含在块级容器中。＜input＞标签也是一种空标签，所以它不能包含文本内容和其他标签，并且必须用结尾斜线(/＞)闭合。浏览器呈现文档时，＜input＞标签将被功能性的表单控件替换。

＜input＞标签的必需属性如下。

name：用于标识控件，在表单被提交时将与控件值组成一个名/值对。如果漏掉该属性，标记代码验证程序可能不会为其输出错误提示，但它是成功处理表单所必需的属性。

＜input＞标签的可选属性如下。

(1)alt：指定一个替代性文字说明(只用于 input type＝"image")。

(2)accept：包含一个用逗号分隔的 MIME 类型列表，表示可接受的文件类型(只用于 input type＝"file")。

(3)accesskey：为控件提供一种快捷的键盘访问方式，它使通过键盘访问控件更方便、迅速。这个属性的值是与访问键相关的字符。实际用于激活访问键的按键组合因浏览器和操作系统而异。

(4)checked＝"checked"：设置该属性将把复选框或单选按钮的初始状态设置为选中状态(只用于 input type＝"checkedbox"和 input type＝"radio")。

(5)disabled＝"disabled"：设置该属性将禁用该控件，使其不能再获得焦点或被修改。被禁用的控件的值不会被提交。许多浏览器会用“灰化”的样式显示被禁用的控件。

(6)ismap＝"ismap"：声明控件是一个服务器端图像映射(只用于 input type＝"image")。

(7)maxlength：指定文本框中允许输入的最大字符数(只用于 input type＝"text"和 input type＝"password")。

(8)readonly＝"readonly"：说明控件只能显示值，不能被修改(只用于 input type＝"text"和 input type＝"password")。不同于 disabled 属性的是，具有该属性的控件(只读控件)仍能够获得焦点，而且其值也会随表单一起提交。

(9)size：指定文本、密码或文件控件显示的宽度(只用于 input type＝"text"，input type＝"file"和 input type＝"password")。

(10)src：指定图像文件的 URL(只用于 input type＝"image")。

(11)tabindex：指定控件在用 Tab 键遍历活动的控件的顺序中所处的位置。

(12)usemap：指定所使用的客户端图像映射的 URL(只用于 input type＝"image")。

(13)value：指定控件在被用户修改之前的初始值。

(14)type：指定该标签所创建的表单控件的类型，默认值为 text。该属性的取值及其含义如下。

①input type＝"text"：该类型的＜input＞标签用于创建一个单行文本框控件，用户可向其中输入各种文字，如名字、地址或一个问题的简短答案等。

②input type＝"password"：该类型的＜input＞标签用于创建一个密码框控件。密码框与文本框类似，是一个单行的控件，但密码框会掩饰输入的文本，通常使用一串星号(＊)或实心圆点显示输入的内容。这可以增加安全性和隐私性。

③input type="checkedbox"：该类型的<input>标签用于创建复选框控件。当有多个选项存在，且同时需要选择的选项不止一个时，可以使用复选框。

④input type="radio"：该类型的<input>标签用于创建单选按钮控件，单选按钮与复选框类似，但在一组单选按钮中只能选择一个选项。

⑤input type="file"：该类型的<input>标签用于创建一种专用的文件上传控件。该控件通常由一个文本框和一个“浏览”按钮组成，用户可以用它指定一个位于本地计算机或局域网上的文件。指定路径的方法是：在文本框中输入确切的文件路径，或通过单击“浏览”按钮打开操作系统内建的文件浏览器，用户在这个文件浏览器中找到文件所在的位置，则其路径会自动显示在控件的文本框中。递交表单时，所选择的文件将被上传到 Web 服务器。

⑥input type="submit"：该类型的<input>标签用于创建一个提交按钮控件。单击该按钮时，会提交整个表单数据集。

⑦input type="reset"：该类型的<input>标签用于创建一个重置按钮控件。该控件用于重置整个表单，清除所有输入的值，并将所有控件设为初始状态。

⑧input type="button"：该类型的<input>标签用于创建一个按钮控件。按钮控件是一种通用的按钮，没有固定的功能，可以用来触发客户端脚本。

⑨input type="image"：该类型的<input>标签用于创建一个图像按钮控件。该控件的行为基本与提交按钮一样，单击它会提交表单，但它用更具有装饰性的图像取代了标准按钮。

⑩input type="hidden"：该类型的<input>标签用于创建一个隐藏式输入控件。该控件不会被显示，只是一种在提交表单时，用来传送用户不需要看见或修改的额外数据的工具，所传送的数据由 value 属性提供。

3)<button>标签

<button>标签的工作方式类似于 type 属性值为 submit、reset、button 或 image 的<input>标签。触动<button>标签将提交或重置表单，或触发由脚本控制的响应。与<input>标签不同的是，<button>标签不是空标签，可以包含文本或其他标签。

<button>标签的必需属性如下。

type：指定该标签创建的按钮控件的类型，可以为 submit、reset 或 button。

<button>标签的可选属性如下。

(1)accesskey：为控件提供一种快捷的键盘访问方式，它使得通过键盘访问控件更方便、迅速。这个属性的值是与访问键相关的字符。实际用于激活访问键的按键组合因浏览器和操作系统而异。

(2)disabled="disabled"：设置该属性将禁用该按钮，使其不能再被触发。被禁用的控件的值不会被提交。许多浏览器会用“灰化”的样式显示被禁用的控件。

(3)tabindex：指定控件在用 Tab 键遍历活动的控件的顺序中所处的位置。

(4)value：指定将随提交的表单数据一起传送的值。

代码清单 2-27 展示了一个<button>标签的示例。示例中的<button>标签包含了一些被强调的文本和一个图像。

```
<button type="submit" name="continue">
    <strong>Continue to the next page</strong>
    <img src="next.jpg" width="28" height="20" alt=""/>
</button>
```

代码清单 2-27　包含文本和图像的<button>标签的用法

上述代码在浏览器中的呈现效果如图 2-23 所示。默认情况下，<button>标签的外观与<input>标签生成的按钮相同，这可以用 CSS 样式加以改变。

图 2-23　包含文本和图像的<button>标签的显示效果

4)<select>标签

<select>标签用于创建列表框控件，这是由待选的选项组成的菜单。根据其可选的 size 属性的设置，这种控件要么占据多行，要么显示在一行上，并且能向下展开以显示所有选项。单行列表框又称下拉式菜单，当它处于未被激活的折叠状态时，显示的是当前选择的选项。

<select>标签的必需属性如下。

name：用于标识控件，在表单被提交时将与控件值组成一个名/值对。如果漏掉该属性，标记代码验证程序可能不会为其输出错误提示，但它是成功处理表单所必需的属性。

<select>标签的可选属性如下。

(1)disabled="disabled"：设置该属性将禁用该控件，使其不能再获得焦点。被禁用的控件的值不会被提交。许多浏览器会用“灰化”的样式显示被禁用的控件。

(2)tabindex：指定控件在用 Tab 键遍历活动的控件的顺序中所处的位置。

(3)multiple="multiple"：控件将自动转变为多行列表框，表明可选择多个选项。

5)<option>标签

<select>标签中的每个选项都包含在一对<option>…</option>标签中。<option>标签是一个非空标签，但只能包含一条用于显示在菜单中的文本。<option>标签不能包含其他标签，只能包含文本。

<option>标签没有必需属性，可选属性如下。

(1)disabled="disabled"：设置该属性将禁用该控件，使其不能再被选择。被禁用的控件的值不会被提交。许多浏览器会用“灰化”的样式显示被禁用的控件。

(2)label：提供一个简短的替代文本，它将代替标签的内容被显示。

(3)value：指定随提交的表单数据一起传送的值。如果所选择的<option>标签没有设置 value 属性，那么它包含的文本内容将作为其值被传送。

(4)selected="selected":将选项的初始状态设置为选定状态。

代码清单 2-28 展示了包含 3 个<option>标签的<select>标签的示例。

```
<select name="size">
    <option value="1">Small </option>
    <option value="2">Medium </option>
    <option value="3">Large </option>
</select>
```

代码清单 2-28　包含 3 个<option>标签的<select>标签

上述代码在浏览器中的呈现效果如图 2-24 所示。

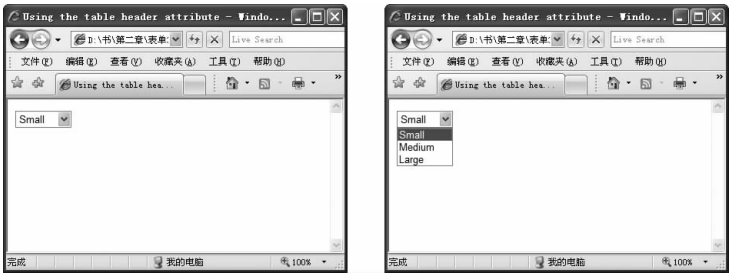


图 2-24　包含 3 个<option>标签的<select>标签的显示效果

6)<textarea>标签

<textarea>标签用于创建多行文本框,可以向其中输入多段文字。<textarea>标签是非空标签,所以需要闭标签。它只能包含文本,这些文本将作为其初始值被显示。

<textarea>标签的必需属性如下。

(1)name:用于标识控件,在表单被提交时将与控件值组成一个名/值对。如果漏掉该属性,标记代码验证程序可能不会为其输出错误提示,但它是成功处理表单所必需的属性。

(2)cols:指定一行上所能显示的字符数,即定义了所呈现的文本框的宽度。如果有必要,文本会自动换行。如果一行较长的文本中没有可用做换行位置的空格,则会出现水平滚动条。

(3)rows:指定不进行垂直滚动时所能显示的文本行数,即定义了所呈现的文本框的高度。在文本行数超过这个限制时,浏览器将自动在文本框中生成一个垂直滚动条。

<textarea>标签的可选属性如下。

(1)accesskey:为控件提供一种快捷的键盘访问方式,它使通过键盘访问控件更方便、迅速。这个属性的值是与访问键相关的字符,实际用于激活访问键的按键组合因浏览器和操作系统而异。

(2)disabled="disabled":设置该属性将禁用该控件,使其不能再获得焦点或被修改。被禁用的控件的值不会被提交。许多浏览器会用“灰化”的样式显示被禁用的控件。

(3)tabindex:指定控件在用 Tab 键遍历活动的控件的顺序中所处的位置。

(4)readonly="readonly":说明控件只能显示值,不能被修改。与 disabled 属性不同的是,具有该属性的控件(只读控件)仍能够获得焦点,而且其值也会随表单一起提交。

课堂实训 2-3　制作个人家庭网页

1. 实训内容

制作一个个人家庭网页，用来展示个人信息，表现自我个性，效果如图 2-25 所示。

(a)　(b)　(c)　(d)

图 2-25　个人家庭网页效果

2. 实训目的

通过制作个人家庭网页，掌握 XHTML 的基本结构和 XHTML 各种标签的使用方法。

3. 技能分析

(1)利用 Dreamweaver CS 创建基本的静态页面。

(2)用<a>标签实现网站内页面的链接。

(3)用<table>标签实现页面布局。

(4)用<img>标签插入图片。

4. 实训步骤

(1)建立站点根目录 myhome,并在根目录下建立存放图片的文件夹 images。

(2)利用 Dreamweaver CS3 创建站点 myhome,操作中某个重要的对话框如图 2-26 所示。

图 2-26　创建站点时某个重要的对话框

(3)创建首页文件 INDEX. html。

(4)插入段落标签<p>,设置其对齐方式为居中。在该段落中插入图片 bar1. gif,并设置其大小。

(5)插入段落标签<p>,设置其对齐方式为居中。在该段落中插入图片 ren. gif,并设置其大小、替换文本及边框宽度等属性。

(6)插入表格,并设置其文本对齐方式、宽度及边框宽度等属性。

(7)设置表格为单行 3 列,并在每个单元格中插入图片及链接文本。

(8)插入段落标签<p>,设置其对齐方式为居中。在该段落中插入邮件链接图片 email. gif,并设置其大小、替换文本及边框宽度等属性。

(9)插入段落标签<p>,设置其对齐方式为居中。在该段落中插入图片 bar2. gif,并设置其大小。

(10)至此首页文件 INDEX. html 的创建就完成了,该文件程序代码如代码清单 2-29 所示。

```
<html>
<head>
<title>我的站点</title>
<meta http-equiv="Content-Type" content="text/HTML; charset=gb2312"/>
</head>
<body text="#000000" link="#000000" vlink="#FFcc33" alink="#FFCC33">
    <p align="center"><img src="images/bar1.gif" width="512" height=
    "50"/></p>
```

```
    <p align="center"><img src="images/ren.gif" width="399" height=
    "274" alt="我的家庭照片" align="absbottom" border="0"/></p>
    <table align="center" width="75%" border="0">
     <tr>
    <td align="center">
      <p><a href="home.html"><img name="Image8" border="0"
  src="images/gal_i1_2.gif" width="64" height="64"/></a></p>
      <h2><font color="#000000"><a href="home.HTML">我的家庭</a>
  </font></h2>
    </td>
    <td align="center">
        <p><a href="HOBBY.html"><img name="Image9" border="0" src=
  "images/gal_i2_2.gif" width="64" height="64"/></a></p>
        <h2><font color="#000000"><a href="HOBBY.html">兴趣爱好
  </a></font></h2>
    </td>
    <td align="center">
      <p><a href="pet.html"><img name="Image10" border="0" src=
  "images/gal_i3_2.gif" width="64" height="64"/></a></p>
      <h2><font color="#000000"><a href="pet.html">我的宠物</a>
  </font></h2>
      </td>
     </tr>
  </table>
  <p align="center"><a href="mailto:student_ma@163.com">
  <img src="images/email.gif" width="120" height="57" border="0"
alt="与我联系"/></a></p>
  <p align="center"><img src="images/bar2.gif" width="512" height=
  "10"/></p>
</body>
</html>
```

代码清单 2-29　INDEX.html 页面代码

(11)用相同的方法创建“我的家庭”文件 home.html，文件代码如代码清单 2-30 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xHTML1/DTD/xHTML1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xHTML">
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=utf-8" />
<title>我的家庭</title>
```

```
</head>
<body text="#000000" link="#000000" vlink="#FFcc33" alink="#FFCC33">
  <p align="center"><img src="images/bar1.gif" width="512" height=
  "50"/></p>
  <h1 align="center"><font color="#000000">我的家庭</font></h1>
  <table align="center" width="75%" border="0">
    <tr>
      <td align="center"><img src="IMAGES/home5.jpg" width="163"
  height="135" /></td>
      <td align="center"><img src="IMAGES/home6.jpg" width="163"
  height="135" /></td>
    </tr>
    </table>
    <p> </p>
    <table align="center" width="75%" border="0">
    <tr>
      <td align="center"><img src="IMAGES/home1.jpg" width="170"
  height="170" /></td>
      <td align="center"><img src="IMAGES/home2.jpg" width="170"
  height="170" /></td>
      <td align="center"><img src="IMAGES/home3.jpg" width="170"
  height="170" /></td>
    </tr>
    </table>
    <p align="center"><img src="images/bar2.gif" width="512" height=
  "10"/></p>
</body>
</html>
```

代码清单 2-30　home.html 页面代码

(12)用相同的方法创建“兴趣爱好”文件 HOBBY.html，文件代码如代码清单 2-31 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xHTML1/DTD/xHTML1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xHTML">
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=utf-8"/>
<title>兴趣爱好</title>
</head>
<body text="#000000" link="#000000" vlink="#FFcc33" alink="#FFCC33">
```

```
    <p align="center"><img src="images/bar1.gif" width="512"
height="50"/></p>
    <h1 align="center"><font color="#000000">我的爱好</font>
    </h1>
    <table align="center" width="75%" border="0">
     <tr>
        <td align="center"><img src="IMAGES/sport.jpg" width="170"
height="170"/></td>
        <td align="center"><img src="IMAGES/sport1.jpg" width="170"
height="170"/></td>
        <td align="center"><img src="IMAGES/sport2.jpg" width="170"
height="170"/></td>
    </tr>
    <tr>
        <td align="center"><img src="IMAGES/sport3.jpg" width="170"
height="170"/></td>
        <td align="center"><img src="IMAGES/sport4.jpg" width="170"
height="170"/></td>
        <td align="center"><img src="IMAGES/sport5.jpg" width="170"
height="170"/></td>
    </tr>
    </table>
     <p align="center"><img src="images/bar2.gif" width="512"
height="10"/></p>
</body>
</html>
```

代码清单 2-31　HOBBY.html 页面代码

(13)用相同的方法创建“我的宠物”文件 pet.html,文件代码如代码清单 2-32 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xHTML1/DTD/xHTML1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xHTML">
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=utf-8"/>
<title>我的宠物</title>
</head>
<body text="#000000" link="#000000" vlink="#FFCC33" alink="#FFCC33">
    <p align="center"><img src="images/bar1.gif" width="512" height=
```

```
"50"/></p>
    <h1 align="center"><font color="#000000">我的宠物</font></h1>
    <table align="center" width="75%" border="0">
      <tr>
          <td align="center">
              <img src="IMAGES/1.jpg" width="170" height="170"/>
          </td>
          <td align="center">
              <img src="IMAGES/2.jpg" width="170" height="170"/>
          </td>
          <td align="center">
              <img src="IMAGES/3.jpg" width="170" height="170"/>
          </td>
      </tr>
      <tr>
          <td align="center">
              <img src="IMAGES/4.jpg" width="170" height="170"/>
          </td>
          <td align="center">
              <img src="IMAGES/5.jpg" width="170" height="170"/>
          </td>
          <td align="center">
              <img src="IMAGES/6.jpg" width="170" height="170"/>
          </td>
      </tr>
    </table>
    <p> </p>
    <p align="center"><img src="images/bar2.gif" width="512" height=
"10"/></p>
</body>
</html>
```

代码清单 2-32　pet.html 页面代码

5. 实训提高

本实例中各个页面均采用表格实现布局，这是否符合 Web 标准的结构、样式和行为分离的思想？

2.4 实战训练：制作“我的汽车”网页

利用已有知识制作“我的汽车”页面，效果如图 2-27 所示。

图 2-27　“我的汽车”页面效果

1. 任务分析

利用表格实现“我的汽车”页面布局，合理设置表格、图片及文字样式。

2. 任务提示

(1)用表格实现页面布局。

(2)为表格设置样式，如边框与内容间的距离(cellpadding)、相邻单元格之间的距离(cellspacing)、表格大小(weight、height)、对齐方式(align)等。

(3)为图片设置样式，如图片大小(weight、height)等。

(4)使用邮件链接。

3. 步骤提示

(1)准备好素材图片并存放在 images 文件夹中。

(2)用 Dreamweaver 软件设计网页元素，其中的代码如代码清单 2-33 所示。

```
<html>
<head>
<title>我的汽车</title>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
</head>
<body bgcolor="#FFFFFF" >
<table width="800" height="551" border="0" align="center" cellpadding=
```

```
    "0" cellspacing="0">
  <tr>
    <td colspan="3"><img src="images/a_01.jpg" width="641" height="349"/>
    </td>
    <td><img src="images/a_02.jpg" width="1" height="349"/></td>
    <td><img src="images/a_03.jpg" width="158" height="349"/></td>
  </tr>
  <tr>
    <td colspan="2"><img src="images/a_04.jpg" width="334" height="142"/>
      </td>
    <td colspan="3" rowspan="3">
      <img src="images/a_05.jpg" width="466" height="201"/></td>
  </tr>
  <tr>
    <td width="126" height="38" align="center" valign="middle" background=
    "images/a_06.jpg"><a href="mailto:zhangminna2006@163.com">
  <font color="#00CC00" size="4">联系我们 &gt;&gt;&gt;</font></a></td>
    <td rowspan="2"><img src="images/a_07.jpg" width="208" height="59"/></td>
  </tr>
  <tr>
    <td><img src="images/a_08.jpg" width="126" height="21"/></td>
  </tr>
  <tr>
    <td><img src="images/分隔符.gif" width="126" height="1"/></td>
    <td><img src="images/分隔符.gif" width="208" height="1"/></td>
    <td><img src="images/分隔符.gif" width="307" height="1"/></td>
    <td><img src="images/分隔符.gif" width="1" height="1" alt=""/></td>
    <td><img src="images/分隔符.gif" width="158" height="1" alt=""/></td>
  </tr>
  </table>
  </body>
  </html>
```

代码清单 2-33　“我的汽车”页面的文档代码

4. 技术难点提示

(1)<td colspan="">设置一个单元格占用的列数(缺省时值为 1),<td rowspan="">设置一个单元格占用的行数(缺省时值为 1)。

(2)使用“>”插入特殊字符“>”。

(3)设置邮件链接:<a href="mailto:zhangminna2006@163.com">。

2.5 思考与练习

一、选择题

1.(　　)用来帮助搜索引擎寻找网页。

A. 网页标题　　B. 描述　　C. 关键字　　D. 刷新

2.(　　)不是表单对象。

A. 文本区域　　B. 单选按钮　　C. 复选框　　D. 表格

3. 在网上浏览信息时经常会遇到要求填写注册资料或提供信息的情况,这些都是用(　　)来实现的。

A. 表单　　B. 表格　　C. 层　　D. 框架

4. 用于自定义列表的标签是(　　)。

A. 　　B. 　　C. 　　D. <dl>

5.(　　)不是<input>标签 type 属性的取值。

A. text　　B. file　　C. button　　D. label

二、简答题

1. 如何设置刷新功能,使浏览器在 5 s 后自动刷新网页?

2. 如何设置超链接?

3. 如何利用图像美化网页?

4. 根据个人实践经验总结,在 Dreamweaver 的“代码”视图中编写 HTML 时有哪些技巧可以提高编写代码的效率?

第3章

CSS 核心基础

使用 CSS 可以将网页的内容与内容的显示效果进行分离，这样既能够方便网页的升级和维护，也可以把页面做得更加美观、大方。本章从 CSS 的基本思想出发，介绍如何定义 CSS 的各种选择器，以及它们的应用。

【学习目标】

- 能使用各种 CSS 选择器进行页面基本设置。
- 能使用各种样式表。
- 掌握 CSS 的继承特性，能够利用继承特性控制页面的显示效果。
- 掌握 CSS 的层叠特性，能够利用层叠特性控制页面的显示效果。

3.1 CSS 的基础知识

3.1.1 CSS 的基本思想

层叠样式表单(cascading style sheets,CSS)又称风格样式表,是一种用来表现 XHTML 或 XML 等文件样式的计算机语言,用于进行网页风格设计。XHTML 和 CSS 之间是“内容结构”与“表现形式”的关系。例如,当链接字未单击时是蓝色,当鼠标指针移上去后链接字变成红色的,且有下划线,这就是一种风格。通过设置 CSS,可以统一地控制 XHTML 中各标签的显示属性。使用 CSS 可以更有效地控制网页外观,可以精确指定网页标签的位置、外观以及创建特殊效果等。

为了说明具体问题,先看一个实例。这里制作了一个 welcome 网页,效果如图 3-1 所示。

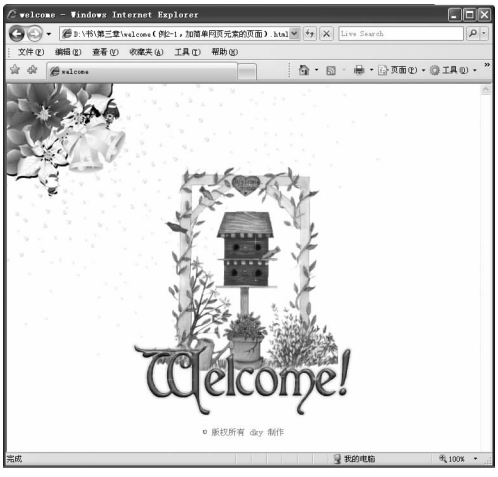


图 3-1　welcome 网页效果

welcome 网页的实现代码如代码清单 3-1 所示。

```
1.<html>
2.<head>
3.<title>welcome</title>
4.<meta http-equiv="Content-Type" content="text/HTML; charset=gb2312"/>
5.</head>
6.<body background="bg.jpg">
7.<p align="center"><img src="welcome.gif"/></p>
8.<p align="center">
9.<font size="2" color="#CB7E81">&copy; 版权所有 dky 制作</font></p>
10.</body>
11.</html>
```

代码清单 3-1　welcome 网页实现代码

提示：代码清单第 9 行“&copy”是 XHTML 编码的特殊字符，显示处理后的效果是“©”，其他字符请读者参阅相关资料。

这时会发现这个网页背景图片出现了多次，而且随着浏览器窗口的变化，背景图片也在变化。如何才能控制背景图片不变呢？这就需要利用 CSS 代码实现，具体代码如代码清单 3-2 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xHTML1/DTD/xHTML1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xHTML">
<head>
    <meta http-equiv="Content-Type" content="text/HTML; charset=gb2312" />
    <title>welcome2</title>
    <style type="text/CSS">
    body{
        background-image: url(bg.jpg);
        background-repeat: no-repeat;
    }
    </style>
</head>
<body>
    <p align="center"><img src="welcome.gif"/></p>
    <p align="center">
    <font size="2" color="#CB7E81">&copy; 版权所有 dky 制作</font></p>
</body>
</html>
```

代码清单 3-2　welcome 网页 CSS 实现代码

从上述代码可以看出，CSS 对页面的背景所进行的设置使页面背景图片只能出现一次，起到了很好的控制效果。不仅如此，CSS 还可以很好地控制整个网页的样式，其最核心的思想就是将“内容”和“表现”分别由 XHTML 和 CSS 承担。

3.1.2　CSS 基本选择器

选择器是 CSS 中重要的概念，所有 XHTML 中的标签样式都是通过不同的选择器控制的。利用 CSS 选择器可以对 XHTML 页面中的标签实现一对一、一对多或者多对一的控制。每个 CSS 选择器都包含选择器名、属性和值 3 个部分，如图 3-2 所示，其中，属性和值可以设置多个，从而实现一对多的控制。

```
选择器名 { 属性1: 值;
          属性2: 值;
          属性3: 值1 值2 值3…; }
```

图 3-2　CSS 基本选择器的结构

从图 3-2 可以看出，“{}”内的属性可以有多个，属性之间用“;”分隔，属性和值之间用“:”分隔，如果有多个值，则用空格分隔。

CSS 有多种类型的选择器，但是不同的浏览器支持的选择器不同，这里重点介绍 3 种基本选择器，分别是标签选择器、类选择器和 ID 选择器。

1. 标签选择器

一个完整的 XHTML 页面由多个不同的标签组成，而标签选择器用于决定哪些标签采用哪种 CSS 样式。一般情况下，设计整体页面的效果时使用标签选择器。例如，在 style.css 文件中对＜p＞标签样式的定义如下：

```
p{
  font-size:12px;
  background:#FF0000;
  color:#009900;
}
```

基于上面的定义，则页面中所有＜p＞标签的背景都是#FF0000(红色)，文字大小均是 12 px，字体颜色为#009900(绿色)。在后期维护中，如果想改变整个网站中＜p＞标签背景的颜色，只需要修改 background 属性就可以了，但是不能单独更改某一个标签的颜色和大小。

2. 类选择器

类选择器在使用时常在类名前面加“.”来标识，例如：

```
.tDiv{
  color:#FF0000;
}
```

然后就可以在 XHTML 中使用该类选择器了，例如：

```
＜div class="tDiv"＞
  这个区域字体颜色为红色
＜/div＞
```

3. ID 选择器

ID 选择器根据标签 ID 来选择标签，具有唯一性。ID 选择器在使用时常在选择器名前面加“#”来标识，例如：

```
#dDiv{
  color:#FF0000;
}
```

上述代码的含义是使 ID 为 dDiv 的标签中的文本的字体颜色为红色。然后就可以在页面中使用该选择器了，例如：

```
＜div id="dDiv"＞
  这个区域字体颜色为红色
＜/div＞
```

用 IE 7 浏览器浏览，可以看到区域内的颜色变成了红色。再定义一个区域：

```
<div>
    这个区域没有定义颜色
</div>
```

用 IE 7 浏览器浏览，效果与预期的一样，区域没有应用样式，所以区域中的字体颜色是默认的黑色。

提示：实际上，各种浏览器在编译 XHTML 代码时存在个别显示效果不同的现象。这里要说明的是，本书用的浏览器是 IE 7。

3.1.3 CSS 复合选择器

1. 交集选择器

交集选择器由两个选择器直接连接构成，其中，第一个必须是标签选择器，第二个可以是类选择器或 ID 选择器，两个选择器名中间不能有空格，必须连续书写。交集选择器的结果是两个选择器的交集。

下面举例说明交集选择器的用法。某个使用了交集选择器的网页的效果如图 3-3 所示。

图 3-3　使用了交集选择器的网页效果

图 3-3 所示网页的实现代码如代码清单 3-3 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=gb2312"/>
<title>交集选择器</title>
<style type="text/CSS">
p{color:blue;}
p.txtcolor{color:red;}
.txtcolor{color:green;}
</style>
</head>
<body>
    <p>直接使用 p 标签</p>
```

```
    <h1>直接使用 h1 标签</h1>
    <p class="txtcolor">使用 p 标签和 txtcolor</p>
    <h1 class="txtcolor">使用 h1 标签和 txtcolor</h1>
</body>
</html>
```

代码清单 3-3　使用交集选择器

上述代码定义了 3 个选择器，分别是标签选择器 p、类选择器. txtcolor 和交集选择器 p. txtcolor。这 3 个选择器的作用范围如图 3-4 所示。

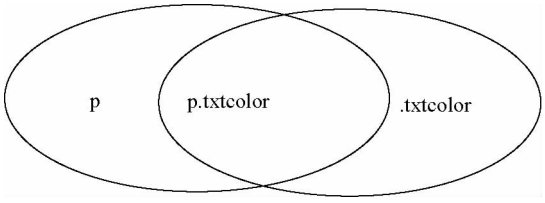


图 3-4　3 个选择器的作用范围

2. 并集选择器

并集选择器也由多个选择器构成，选择器之间由逗号连接，它的结果是多个选择器的并集。如果某几个页面选择器风格相同或者部分相同，可以利用并集选择器进行声明。

下面举例说明并集选择器的用法。某个使用了并集选择器的网页的效果如图 3-5 所示。

图 3-5　使用了并集选择器的网页效果

图 3-5 所示网页的实现代码如代码清单 3-4 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
<title>并集选择器</title>
<style type="text/css">
```

```
h2{
    color:red;
    font-size:15px;
}
p{
    color:blue;
    font-size:16px;
}
h2.txtcolor,.txtcolor{
    text-decoration:underline;
}
</style>
</head>
<body>
    <h2 class="txtcolor">h2 双重效果</h2>
    <p>p 单独效果</p>
    <p class="txtcolor">p 双重效果</p>
</body>
</html>
```

代码清单 3-4　使用并集选择器

上述代码使用了并集选择器，它们的作用范围如图 3-6 所示。

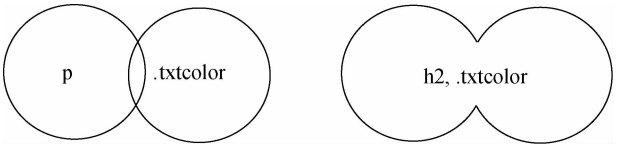


图 3-6　并集选择器的作用范围

3. 后代选择器

后代选择器通过各种选择器的嵌套实现对页面的控制，其中，内层标签是外层标签的后代。在写法上，外层标签写在前面，内层标签写在后面，中间用空格分隔。例如，对于下面的代码：

<p>我在外面<h1>我在里面</h1>我在外面<p>

<p>标签中嵌套了<h1>标签，所以<h1>标签是<p>标签的子标签，要想使两个标签显示不同的效果，就可以使用后代选择器。后代选择器的使用范围及其广泛，下面举例说明后代选择器的用法。某个使用了后代选择器的网页的效果如图 3-7 所示。

图 3-7　使用了后代选择器的网页效果

图 3-7 所示网页的实现代码如代码清单 3-5 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=gb2312"/>
<title>后代选择器</title>
<style type="text/CSS">
p b{ color:blue;}
b{ color:red;}
</style>
</head>
<body>
    <p>p 在外面<b>h2 在里面</b>外面</p>
    <b>h2 在外面<p>p 在里面</p>外面</b>
</body>
</html>
```

代码清单 3-5　使用后代选择器

3.1.4　在 XHTML 中使用 CSS 的方法

在 XHTML 中使用 CSS 的方法主要有 4 种，分别是行内样式表、嵌入式样式表、链接式样式表和导入式样式表，下面分别介绍。

1. 行内样式表

行内样式表是最直接的一种样式，它在 XHTML 中通过使用 style 属性，然后将 CSS 代码直接写入其中实现。其用法示例如代码清单 3-6 所示。

```
<html>
<head>
    <title>一个简单网页</title>
```

```
</head>
<body style="font-size:28px">
    你好！样式表！
</body>
</html>
```

代码清单 3-6　使用行内样式表

上述代码在 IE 7 浏览器中的显示效果如图 3-8 所示。

图 3-8　使用行内样式表的网页效果

在代码清单 3-6 中，<body>标签内有一条特别的语句 style="font-size：28px"。类似这样的写法称为属性，style 就是<body>标签的一个属性，这就是行内样式表的书写方式。即行内样式表出现在要控制其格式的标签内部，形式为 style=""，引号中间是样式控制的命令。

2. 嵌入式样式表

嵌入式样式表又称内嵌式样式表，它利用<style>和</style>标签进行页面控制，放在<head>和</head>之间。将代码清单 3-6 改写成嵌入式样式表，其代码如代码清单 3-7 所示。

```
<html>
<head>
    <title>一个简单网页</title>
<style type="text/CSS">
    body{font-size:28px;}
</style>
</head>
<body>
    你好！样式表！
</body>
</html>
```

代码清单 3-7　使用嵌入式样式表

3. 链接式样式表

链接式样式表属于外部样式表的一种，它将 XHTML 和 CSS 分成两个或者多个页面，把 CSS 样式存储在扩展名为. css 的文件中，利用＜link＞标签的 href 属性实现样式控制。用链接式样式表实现图 3-8 所示网页的步骤如下。

(1)创建一个 a. css 的文件，代码如下：

```
body{font-size:28px;}
```

(2)编写 XHTML 文件代码，其代码如代码清单 3-8 所示。

```
<html>
<head>
    <title>一个简单网页</title>
    <link href="a.CSS" type="text/CSS" rel="stylesheet"/>
</head>
<body>
    你好！样式表！
</body>
</html>
```

代码清单 3-8　使用链接式样式表

4. 导入式样式表

导入式样式表也属于外部样式表的一种，与链接式样式表的区别在于引入方法不同，它在 HTML 文件的＜style＞和＜/style＞标签之间引入 CSS 文件。用导入式样式表实现图 3-8 所示网页的步骤如下。

(1)创建一个 a. css 的文件，代码如下：

```
body{font-size:28px;}
```

(2)编写 XHTML 文件代码，其代码如代码清单 3-9 所示。

```
<html>
<head>
    <title>一个简单网页</title>
    <style>
    @import url(a.CSS);
    </style>
</head>
<body>
    你好！样式表！
</body>
</html>
```

代码清单 3-9　使用导入式样式表

提示：(1)如果一个页面样式表比较简单，而且只使用一次，那么可以使用行内样式表。内嵌式样式表将 CSS 代码集中在一个区域，有利于后期的维护，且网页本身会变得很清晰；

但是，如果一个网站有多个页面的风格样式都相同，使用内嵌式样式表就比较麻烦了，这时需要使用链接式样式表或者导入式样式表。

(2)如果这 4 种样式表用在同一个页面的同一个标签上，它们的优先级从高到低的顺序是：行内样式表、嵌入式样式表、外部样式表(链接式样式表和导入式样式表)。

课堂实训 3-1　制作 Good morning 网页

1. 实训内容

本实训将制作的网页是一个个人网站的首页。个人网站是一个可以发布个人信息及相关内容的网站。通俗理解，个人网站就是指网站内容是介绍自己的或是以自己的信息为中心的网站，不一定是自己做的网站，但强调的是以个人信息为中心。一个好的个人网站的首页能够清楚地表达出网站的目的。个人网站包括博客、个人论坛、个人主页等。Good morning网页的效果如图 3-9 所示，其实现代码如代码清单 3-10 所示。

图 3-9　Good morning 网页效果图

```
1.<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.
2.     w3.org/TR/xHTML1/DTD/xHTML1-transitional.dtd">
3.<html xmlns="http://www.w3.org/1999/xHTML">
4.<head>
5.<meta http-equiv="Content-Type" content="text/HTML; charset=gb2312" />
6.<title>Good morning</title>
7.<style type="text/CSS">
8.<!--
9.body{
10.     background-image:url(bg2.jpg);
11.}
12..STYLE1 {
13.     font-size:12px;
14.     color:#1A4344;
```

```
15.}
16.-->
17.</style>
18.</head>
19.<body>
20.<p> </p>
21.<p> </p>
22.<p> </p>
23.<p align="center"><img src="good morning.gif" /></p>
24.<p align="center"><font>
25.<span class="STYLE1">&copy; 版权所有 DKY 2012 制作</span></font></p>
26.</body>
27.</html>
```

代码清单 3-10　Good morning 网页实现代码

2. 实训目的

利用 Dreamweaver 软件具体分析网页，从而掌握网页的基本结构和 CSS 选择器的使用方法。

3. 技能分析

(1)XHTML 的基本结构。

(2)XHTML 中标签的含义。

(3)CSS 基本选择器的使用方法。

4. 实训步骤

(1)第 1 行代码指出了本网页的超文本类型是可扩展超文本过渡文档类型(XHTML 1.0 Transitional)。

(2)第 4 行～第 18 行设置了网页的“头”(head)。

(3)第 5 行代码 http-equiv="Content-Type"设置的是 HTTP 的头部协议，提示浏览器网页的信息；charset 的信息参数为 gb2312，说明网站采用的编码是简体中文。

(4)第 6 行代码指出了网页的标题(title)是 Good morning。

(5)第 7 行代码指出本网页使用内嵌式样式表(<style type="text/CSS">)。

(6)第 8 行和第 16 行代码为本页面的样式表加了注释(<!-- … -->)。

(7)第 9 行代码设定了一个标签选择器 body，第 19 行是它在页面中的使用。

(8)第 12 行代码设定了一个类别选择器 STYLE1，第 25 行是它在页面中的使用。

(9)第 19 行～第 26 行是网页的“身体”(body)。

5. 实训提高

(1)CSS 样式加入页面的方式有很多种，应如何使用它们?

提示：在建设网站时最好使用其中的 1～2 种，这样既方便使用，又有利于后期维护和管理。

(2)CSS 用于实现样式控制时，如果一个大的网站有很多样式，那么选用哪种方式能更好地把样式加入页面中，而且便于修改呢？

提示：导入式更方便些，这样每个相似的页面都可以应用，代码的重用性高。

课堂实训 3-2　制作汽车网页

1. 实训内容

如今，汽车已经成为不少家庭不可缺少的一部分，用于汽车展示、汽车销售、汽车论坛、汽车服务等的网站随处可见。下面就来制作一个简单的网页，用于展示甲壳虫汽车，网页效果如图 3-10 所示。

图 3-10　甲壳虫汽车展示网页效果

2. 实训目的

通过制作一个简单的甲壳虫汽车展示网页，掌握 XHTML 的基本结构和在 XHTML 中使用 CSS 的方法。

3. 技能分析

(1)利用 Dreamweaver CS 创建基本的静态页面。

(2)用 CSS 选择器设置基本样式。

①标题 h1 样式：文字的对齐方式为居中，颜色为＃333333，有下框线，框线宽度为 1 px，颜色为＃d4d0bd，且为实线。

②图片样式：宽度为 378 px，高度为 260 px，有宽度为 1 px 的边框。

③主体文字样式：大小为 14 px，行高为 24 px。

④版权文字样式：大小为 12 px，颜色为蓝色。

(3)在 XHTML 中使用 CSS。

①利用嵌入式样式表使用标题 h1 样式。
②利用行内样式表使用图片样式。
③利用嵌入式样式表使用主体文字样式。
④利用行内样式表使用版权文字样式。

4. 实训步骤

(1)利用 Dreamweaver CS3 创建静态 XHTML 页面,并设置文档类型,如图 3-11 所示。

图 3-11　设置文档类型

(2)利用 XHTML 设置页面内容,插入素材图片和文字,注意空格和版本标签的设置方法,其代码如代码清单 3-11 所示。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=gb2312"/>
<title>甲壳虫</title>
</head>
<body>
<h1>经典就在身边</h1>
<p align="center">
<img src="images/car.jpg"/></p>
<p align="left"><br/>
    提及大众旗下品牌的甲壳虫轿车,人们脑中浮现的往往都会是经典、可爱、小资、个性、时尚等许多赞美的词汇,其实“甲壳虫”作为一个名称的影响力远比作为汽车要来得更加深远。虽然甲壳虫近些年来并没有在造型方面有很大的改变,但是甲壳虫轿车作为主角多次出现在大银幕上并拥有大批的影迷就可以说明它的经典是难以被超越的。<br/>
```

```
    1933 年德国的波尔舍博士设计了一种类似甲壳虫外形的汽车。波尔舍最大限度地发挥了甲壳虫形汽车的长处，使其成为同类车中之王，“甲壳虫”也成为该车的代名词。<br/>
<br/>
<p align="center"><font size="2" color="blue">&copy; 版权所有 DKY 2012 制作</font></p>
</p>
</body>
</html>
```

代码清单 3-11　插入素材图片和文字的代码

(3)利用 CSS 设置 h1 标题样式，并在 XHTML 中使用，代码如下：

```
<style type="text/CSS">
h1{
    text-align:center;
    border-bottom:1px solid #d4d0bd;
    color:#333333
}
</style>
```

(4)利用 CSS 修饰图片，并在 XHTML 中使用，代码如下：

```
img{
    width:378;
    height:260;
    border:1;
}
```

(5)利用 CSS 修饰页面主体文字，并在 XHTML 中使用，代码如下：

```
<style type="text/CSS">
p{
    font-size:14px;
    line-height:24px;
}
</style>
```

(6)在 XHTML 中使用 CSS 修饰版权文字样式，代码如下；

```
<p align="center">
    <font style="font-size:10px,color="blue">&copy; 版权所有 DKY 2012 制作</font>
</p>
```

思考：(1)标签选择器 h1 和 p 是否可以设置为行内样式表？为什么？

提示：标签选择器 h1 可以，代码如下：

```
<h1 style="text-align:center; border-bottom:1px solid #d4d0bd;
  color:#333333">经典就在身边</h1>
```

标签选择器 p 也可以设置为行内样式表，但由于<p>标签在正文中多次出现，更改表现形式比较麻烦，故不建议使用行内样式表的形式。

(2)如果要给案例增加一个背景图片，该如何操作？

提示：首先找到背景图片的属性(background)，然后利用 URL 实现图片的定位。

5. 实训提高

如果这个网站中有多种品牌的汽车，如凯迪拉克等，要想使这些网页都有相同的展示效果，应如何设置 CSS 样式？

提示：如果一个 CSS 样式在同一站点的多个网页中使用，那么最好使用链接式样式表。首先创建一个 a.css 文件，在此文件中设置所有的样式，并存放到站点的根目录中，然后在头文件中增加如下代码：

```
<style>
@import url(a.css);
</style>
```

3.2 CSS 的特性

3.2.1 CSS 的继承特性

所谓 CSS 的继承特性，是指被包含在内部的标签将拥有外部标签的样式性质。继承特性最典型的应用通常发生在整个网页的样式预设中，需要指定为其他样式的部分，设定在个别标签里即可。这项特性可以给网页设计者提供更理想的发挥空间，但同时继承也有很多规则，下面介绍这方面的应用。

1. 继承的关系

CSS 的一个主要特性就是继承，它是依赖于祖先与后代的关系的。继承是一种机制，它允许样式不仅可以应用于某个特定的标签，还可以应用于它的后代。例如，一个<body>定义了的颜色值也会应用到段落的文本中。为了说明问题，来看代码清单 3-12 中代码的层次结构。

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=gb2312"/>
<title>继承</title>
<style type="text/CSS">
    .one{color:red;}
```

```
        .two{color：＃0099FF;}
        .three{color：＃990033;}
    </style>
    </head>
    <body>
    <ul class="one">
        <li>第 1 章 Java 开发入门</li>
        <ul class="two">
            <li>1.1 开发准备</li>
                <ul id="three">
                    <li>1.1.1 什么是 Java</li>
                    <li>1.1.2 下载和安装 JDK</li>
                    <li>1.1.3 设置 Path 和 ClassPath</li>
                </ul>
            <li>1.2 第一个 Java 程序</li>
            <ul class="three">
                <li>1.2.1 编写和编译 Java 程序</li>
                <li>1.2.2 执行 Java 程序</li>
                <li>1.2.3 为程序加入注释</li>
             </ul>
       </ul>
       <li>第 2 章 对象和类</li>
      <ul class="two">
        <li>2.1 对象</li>
        <ul class="three">
            <li>2.1.1 什么是对象</li>
            <li>2.1.2 对象的状态</li>
            <li>2.1.3 对象的行为</li>
         </ul>
      </ul>
    </ul>
    </body>
    </html>
```

代码清单 3-12　继承代码

以上是一个很简单的 XHTML 文档，它组织成一棵“树”结构，树根为<html>，它的叶子是<head>和<body>，其他标签层层嵌套，具体如图 3-12 所示。

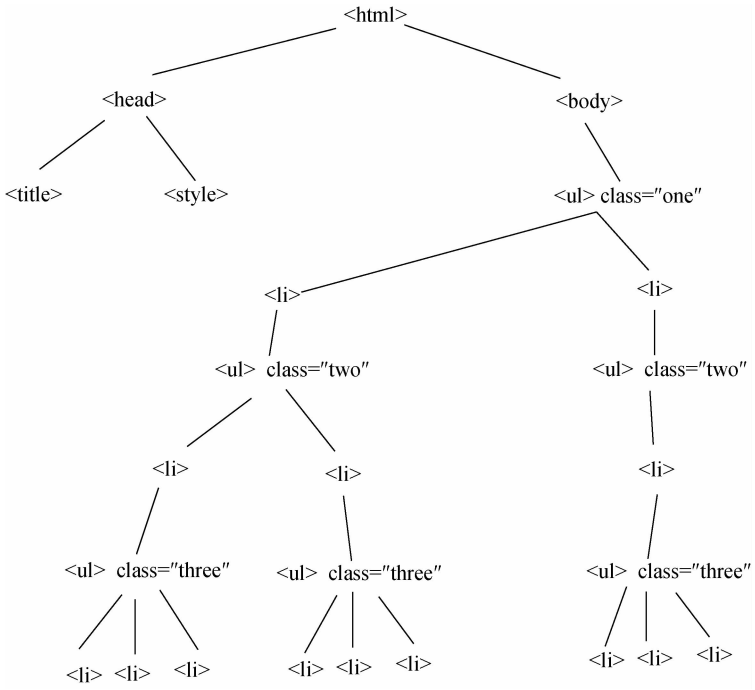


图 3-12　继承关系图

在嵌套声明中存在 CSS 的继承问题：在继承中，父标签中定义了 CSS，如果子标签中没有定义 CSS，则继承父标签的样式；如果子标签中定义了 CSS，则将覆盖父标签的样式。

在 CSS 中，继承也有其局限性，如有些属性是不能继承的。例如，border 属性是用来设置标签的边框的，它就没有继承性。多数边框类属性，如 padding（补白）、margin（边界）和背景等都不能继承。

2. 继承的优先级

在制作网页的过程中，制作者可能想要设置某个规则比其他的规则更重要，CSS 中允许这样设置，称为重要规则（important rule）。这是根据其声明的方式和它们的自然属性命名的。通过在一条规则的分号前插入“!important”来标记一条重要规则。例如：

```
P.apple{color:#red !important; background:white;}
```

由上面的代码可以看出，颜色值 #red 被标记为“!important”，而背景色 white 未被标记。如果两条规则都重要，那么每条规则都要标上“!important”。

标记为“!important”的规则具有最高的权值，也就是说它没有具体的特性值，但是比其他的权值都要大。需要注意的是，虽然设计人员定义的样式比用户定义的样式具有更高权值，但“!important”规则恰恰相反，重要的用户定义规则要比设计人员定义的样式具有更高权值，即使是标记为“!important”的重要规则也是如此。

例如下面的代码：

```
h1{color:gray !important;}
<h1 style="color:black;">看这儿!</h1>
```

“!important”规则会覆盖内联 style 属性的内容，所以结果文字是灰色，而不是黑色。

3. 继承的应用

CSS 继承是指子标签继承父标签的所有样式风格，并可以在父标签的基础上进行修改覆盖，从而产生新的样式。子标签的风格不会影响父标签。CSS 的继承贯穿整个 CSS 页面设计中，利用 CSS 继承的这种关系可以大大缩短代码的编写量，并提高程序的可读性。如果想要把代码清单 3-12 中第 3 级列表字体的显示效果进行变化，可以增加以下 CSS 样式代码：

```
<style type="text/CSS">
  li{
      font-size: 12px;
      text-decoration:none;
  }
</style>
```

修改后的显示效果如图 3-13 所示。

图 3-13　继承应用效果图 1

这时会发现所有的效果全变了，如何修改呢？一种方法是再单独设置一个类别，然后再应用，显然这样很麻烦。另一种方法是利用继承的特性，使用后代选择器实现，这样不用单独设置类别也能完成相同的任务，代码如下：

```
li li li{
    font-size:12px;
    text-decoration:underline;
}
```

修改后的显示效果如图 3-14 所示。

图 3-14　继承应用效果图 2

3.2.2　CSS 的层叠特性

如果对一个文字使用多种 CSS 样式，那么就会产生冲突，它们的优先级是怎样的呢？为了说明问题，首先来看代码清单 3-13 中的代码：

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN">
<html>
<head>
<meta http-equiv="Content-Type" content="text/HTML; charset=gb2312"/>
<title>无标题文档</title>
<style type="text/CSS">
p{color:#99CC66;}
.one{color:red;}
.two{color: #0000FF;}
#three{color: #CC9900;}
</style>
</head>
<body>
<p>《春晓》</p>
<p>作者:孟浩然</p>
<p class="one">春眠不觉晓,</p>
<p class="one" id="three">处处闻啼鸟。</p>
<p class="one two">夜来风雨声, </p>
<p style="color:black">花落知多少。</p>
</body>
</html>
```

代码清单 3-13　CSS 层叠样式代码

上述代码运行后的效果如图 3-15 所示。

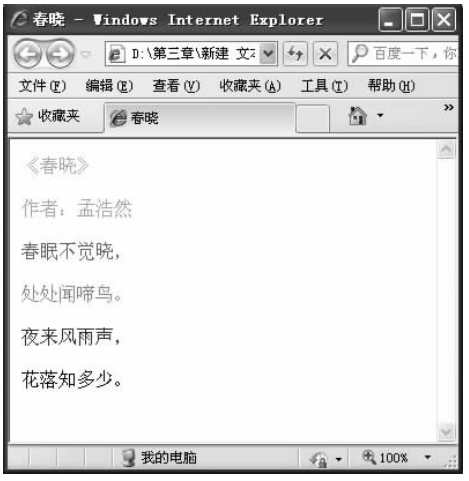


图 3-15　唐诗网页效果图

在上述代码中定义了 4 种 CSS 样式，大部分都用了两个以上的 CSS 样式，这些样式的优先级从高到低的顺序是：行内样式、ID 样式、类别样式、标签样式。也就是说，CSS 的层叠就是页面样式冲突的解决方案。

课堂实训 3-3　制作校园新闻网页

1. 实训内容

校园新闻网页是展示学校面貌的最直接的窗口，它可以包含热点新闻、学校简讯等，用于向师生提供校园内外的各种文化信息。

下面就来制作一个校园新闻网页，网页效果如图 3-16 所示。

图 3-16　校园新闻网页效果图