

第3章 脚本语言

目前,绝大多数网站都能够实现浏览器和服务器的交互功能,不再是仅实现简单连接的静态网页。JSP 可以很好地实现动态网站的设计与开发。而 HTML 语言和 CSS 是开发网页最基本的技术,将其和 JSP 结合起来就能开发出漂亮且功能强大的动态网页。本章主要介绍网页制作技术基础知识。

本章主要内容

- (1) HTML。
- (2) CSS。
- (3) JavaScript。

3.1 HTML

HTML(Hypertext Markup Language)即超文本标记语言或超文本链接语言,是目前网络上应用最广泛的标记语言,也是构成网页文档的主要语言。

3.1.1 HTML 简介

HTML 是一种用来制作超文本文档的简单标记语言,它不是一种真正的编程语言,只是一种标记符。通过一些约定的标签符号对文件的内容进行标注,指出内容的输出格式,如字体大小及颜色、背景颜色、表格形式、各部分之间逻辑关系等。当用户浏览 WWW 信息时,浏览器会自动解释这些标签的含义,并按照一定的格式在屏幕上显示这些被标记的信息。

用 HTML 编写的超文本文档称为 HTML 文档,它是一个放置了标签的 ASCII 文本文件,文件扩展名通常为 .html 或 .htm,是能独立于各种操作系统平台的、可供浏览器解释浏览的网页文件。

3.1.2 HTML 基本标签

为了了解 HTML 的基本标签,先来看一个简单的例子。在 IDE 项目中新建一个 HTML 文件或者新建记事本文档后改扩展名为 .html 或 .htm。本书所有程序都是使用集成开发工具编写的。

下面通过使用 NetBeans 7.3 新建一个 HTML 页面来了解 HTML 页面的基本构成。页面运行效果如图 3-1 所示,代码如例 3-1 所示。

【例 3-1】 HTML 页面实例(HTMLExample.html)。

```
<html>
<head>
  <title>HTML 实例</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```



图 3-1 HTML 页面实例显示效果

```

</head>
<body>
    <hr>
    <h3>欢迎光临本网站</h3>
    <hr>
    <font face="楷体_GB3212" size="2" color="blue">
        这是一个励志网站,欢迎你的访问!
    </font>
    <p>空谈误国,实干兴邦!</p>
    <br>
    <pre>
        曾经有一段美好的学习时光摆在我面前,
        我没有珍惜,
        等我失去的时候我才后悔莫及,
        人世间最痛苦的事莫过于该毕业的时候别人都找到工作而我还没有,
        如果上天能够给我一个再来一次的机会,
        我会对那个大学生说三个字:好好学。
        如果非要在学习上加上一个期限,我希望是……每一天!
    </pre>
    <!--
        儿时或者少年时代,每当我们不好好学习的时候,家里人或者初中、高中老师总会安慰
        我们说,“好好学,等考上大学就可以好好玩啦。”其实大学是最应该好好学习的时代,
        因为大学毕业就要面临工作,工作当中需要你有扎实的理论知识和较强的实践能力,
        所以大学是最应该学习的时期。请行动起来,“喊破嗓子不如甩开膀子”!
    -->
</body>
</html>

```

将上边这段代码保存在 HTMLExample.html 文件中,并在浏览器中打开它或者运行该页面,运行效果如图 3-1 所示。

在上述代码中,一些字母或单词被用“< >”括起来,如<html>、<head>等,这些称为“标签”。标签用来分割和标记网页中的元素,以形成网页的布局、格式等,通过标签可以在网页中加入文本、图片、声音、动画、影视等多媒体信息,还可以实现页面之间的跳转等。每种标签的作用均不同,当用户需要对网页某处进行修改时,就把标签放置在该处前面,浏览器就会知道用户希望下面的内容应如何显示。

接下来简单总结一下代码中出现的几种标签。

(1) 单标签。

某些标签只需单独使用就能完整地表达意思,控制网页效果,这类标签的语法如下:

<标签>

例如,
,表示换行;<meta>,描述 HTML 网页文档的属性。

(2) 双标签。

某些标签成对使用,由一个开始标签和一个结束标签构成。开始标签告诉 Web 浏览器从此处开始执行该标签所代表的功能,而结束标签告诉 Web 浏览器在这里结束该功能,结束标签的形式是在开始标签前加上一个斜杠。这类标签的语法如下:

<标签>内容</标签>

其中,“内容”就是要被这对标签施加作用的部分。

例如:

<title>HTML 实例</title>

表示将“HTML 实例”作为网页标题。

(3) 标签属性。

在单标签和双标签的开始标签里,还可以包含一些属性,以达到个性化的效果。其语法格式如下:

<标签 属性 1 属性 2 属性 3……>

各属性之间无先后次序,属性也可省略(即取默认值)。

例如:

欢迎!

使用 font 标签设置文本格式时,可以用 face 属性来指定所用的字体,使用 size 属性指定所用字号。

需要注意的是,HTML 语言不区分大小写。例如,
和
都表示换行。另外,使用 HTML 标签时,被使用标签不可以交错,即标签需正确进行嵌套,否则将造成错误。

例如:

<body><form></body></form>

应改为

<body><form></form></body>

才能正常运行。

HTML 标签有多种,下面先来了解基本标签。

1. 页面结构标签

通过上面的例子可以看出,HTML 文档分为文档头和文档体两部分。在文档头里,对这个文档进行了一些必要的定义,文档体中才是要显示在网页中的各种正文信息。通常由 3 对标签来构成一个 HTML 文档的框架。

(1) `<html></html>`。

这个标签对告诉浏览器这个文件是 HTML 文档。`<html>`用于 HTML 文档的最前边,用来标识 HTML 文档的开始;`</html>`放在 HTML 文档的最后边,用来标识 HTML 文档的结束。

(2) `<head></head>`。

这个标签对中的内容是文档的头部信息,说明一些文档的基本情况,如文档的标题等,其内容不会显示在网页中。在此标签对之间可使用`<title></title>`、`<meta>`、`<script></script>`等描述 HTML 文档相关信息的标签对。

`<meta>`标签用来描述 HTML 网页文档的属性,例如作者、日期和时间、网页描述、关键词、页面刷新等。

例如:

```
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
```

属性 `http-equiv` 用于向浏览器提供一些说明信息,从而可以根据这些说明做出相应处理。`http-equiv` 其实并不仅仅只有说明网页的字符编码这一个作用,常用的 `http-equiv` 类型还包括网页到期时间、默认脚的语言、网页自动刷新时间等。

属性 `charset` 的作用是指定了当前文档所使用的字符编码为 `gb2312`,也就是中文简体字符。根据这一行代码,浏览器就可以识别出这个网页应该用中文简体字符显示。

(3) `<body></body>`。

这个标签对中的内容是 HTML 文档的主体部分,可包含`<p></p>`、`<h1></h1>`、`<br>`、`<hr>`等标签,它们所定义的文本、图像等将会在网页中显示出来。

2. 页头标签

`<title></title>`标签对用来设定网页标题,浏览器通常会将标题显示在浏览器窗口的标题栏左边。`<title></title>`标签只能放在`<head></head>`标签对之间。

3. 标题标签

在 HTML 文档中,`<h1></h1>`标签可以定义不同显示效果的标题, n 表示标题的级数,取值范围为 $1\sim 6$, n 越小,标题字号越大。

`<h1>`可以使用属性 `align`,用于设置标题文字的对齐方式,其取值如下。

(1) `left`。左对齐。

(2) `right`。右对齐。

(3) `center`。居中对齐。

未设置该属性时,默认为左对齐。

【例 3-2】 使用标题标签(titleTag, html)。

```

<html>
  <head>
    <title>标题标签使用</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body bgcolor="pink">
    这一行文字不是标题<br>
    <h1 align="center">一级标题</h1>
    <h2>二级标题</h2>
    <h3>三级标题</h3>
    <h4>四级标题</h4>
    <h5>五级标题</h5>
    <h6>六级标题</h6>
  </body>
</html>

```

运行效果如图 3-2 所示。可以看出,每级标题的字体为“黑体”。



图 3-2 使用标题标签显示效果

4. 段落标签

`<p></p>` 标签用来创建一个新的段落,在此标签对之间加入的文本将按照段落的格式显示在浏览器上。`<p>` 表示一个段落的开始,结尾标记为 `</p>`。`<p>` 标签可以有多种属性,如 `align` 属性控制其内容的对齐方式, `clear` 属性控制图文混排方式,其取值如下。

- (1) left。下一段显示在左边界处的空白区域。
- (2) right。下一段显示在右边界处的空白区域。
- (3) center。下一段的左右两边都不许有其他内容。

为了防止文档出错,尽量不要省略结尾标记 `</p>`。

5. 预定义格式标签

在编辑文档时,如果希望将来浏览网页时仍能保留在编辑工具中已经排好的形式显示

内容,可以使用<pre></pre>标签对。使用该标签对时,默认字体为 10 磅。

【例 3-3】 使用预定格式标签(preTag.html)。

```
<html>
<head>
  <title>预定格式标签使用</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body bgcolor="pink">
  床前明月光
  疑是地上霜
  举头望明月
  低头思故乡
  <pre>
  兰陵美酒郁金香
  玉碗盛来琥珀光
  但使主人能醉客
  不知何处是他乡
  </pre>
</body>
</html>
```

运行效果如图 3-3 所示。

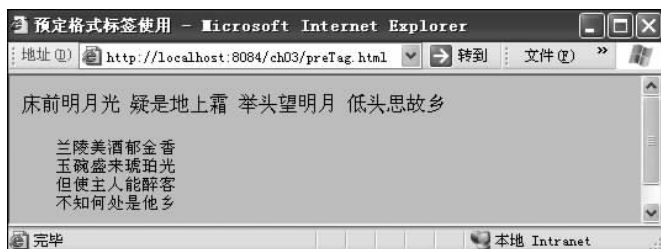


图 3-3 使用预定格式标签显示效果

6. 格式排版标签

(1)
。

该标签强制文本换行,但不会在行与行之间留下空行。如果把
加在<p></p>标签对的外边,将创建一个大的回车换行,即
前边和后边的文本的行与行之间的距离比较大;若放在<p></p>的里边则
前边和后边的文本的行与行之间的距离将比较小。

(2) <hr>。

该标签在网页中加入一条横跨网页的水平线,有多种属性可用于设置水平线的宽度、长度及显示效果等。

① size 属性。设置水平线的粗细,默认单位是像素。

② width 属性。设置水平线的宽度,默认单位是像素,也可以使用对屏幕的百分比

表示。

③ noshade 属性。该属性不用赋值,直接加入标签即可使用,用来取消水平线的阴影(不加入此属性水平线默认有阴影)。

④ align 属性。设置水平线的对齐方式。

⑤ color 属性。设置水平线的颜色。

例如:

```
<hr align="center" width="600" size="9" color="blue">
```

7. 文字格式标签

 标签对通过设置属性来控制文字的字体、大小、样式和颜色,各属性功能如下。

(1) face 属性。设置字体样式。

(2) size 属性。设置字体大小,值为整数,分为 7 个级别,默认字体大小为 3。

(3) color 属性。设置字体颜色。

8. 注释标签

在编写 HTML 文件时,为提高文件的可读性,可以使用“<!--”和“-->”标签对编写注释文字,其语法如下:

```
<!-- 注释语句 -->
```

注释内容不会在浏览器中显示。

3.1.3 列表

列表是一种规定格式的文字排列方式,用于列举内容。常用的列表分为有序列表和无序列表。

1. 有序列表

有序列表是指各列表项按一定的编号顺序显示,列表用开始,以结束,每一个列表项用标签定义,其语法如下:

```
<ol>
  <li>列表项 1</li>
  <li>列表项 2</li>
  :
</ol>
```

在中可以使用 type、start 属性。其中,type 属性用于设置编号的种类,其取值如下。

(1) 1。编号为数字,默认值。例如: 1,2,3,...

(2) A。编号为大写英文字母。例如: A,B,C,...

(3) a。编号为小写英文字母。例如: a,b,c,...

(4) I。编号为大写罗马字符。例如: I,II,III,...

(5) i。编号为小写罗马字符。例如: i,ii,iii,...

start 属性用于设置编号的开始序号,无论 type 取值是什么,start 值只能是 1、2、3 等整数,默认值为 1。

在中可以使用 type、value 属性。其中,type 属性作用与中一致,value 属性用来设定该项的编号,其后各项将以此作为起始编号而递增,其值只能是 1、2、3 等整数,没有默认值。

【例 3-4】 使用有序列表(olTag.html)。

```
<html>
  <head>
    <title>有序列表使用</title>
  </head>
  <body bgcolor="pink">
    Java 方向:
    <ol type="A">
      <li>Java 程序设计</li>
      <li>数据库技术</li>
      <li>JSP 程序设计</li>
    </ol>
  </body>
</html>
```

运行效果如图 3-4 所示。

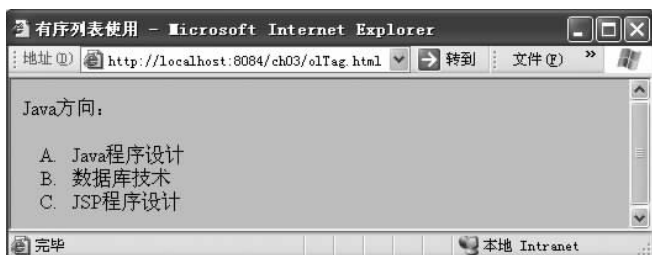


图 3-4 使用有序列表显示效果

2. 无列表表

无列表指各列表项之间没有顺序关系,列表项显示时前面有一个项目符号。无列表表用开始,以结束,每一个列表项同样也用标签定义,其语法如下:

```
<ul>
  <li>列表项 1</li>
  <li>列表项 2</li>
  :
</ul>
```

在、中都可以使用 type 属性,其中,中的 type 属性用于设置列表中所有列表项前的项目符号类型;中的 type 属性用于设置当前列表项前的项目符号类型。type 属性取值如下。

- (1) disc。实心圆点,默认值。
- (2) circle。空心圆点。
- (3) square。实心正方形。

【例 3-5】 使用无序列表(ulTag.html)。

```
<html>
  <head>
    <title>无序列表使用</title>
  </head>
  <body bgcolor="pink">
    Java 方向:
    <ul type="disc">
      <li>Java 程序设计</li>
      <li type="circle">数据库技术</li>
      <li>JSP 程序设计</li>
    </ul>
  </body>
</html>
```

运行效果如图 3-5 所示。

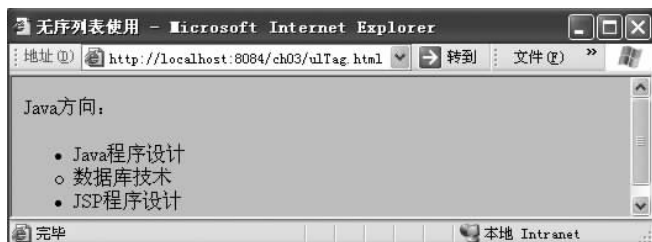


图 3-5 使用无序列表显示效果

3.1.4 多媒体和超链接

多媒体和超链接在网页中起着非常重要的作用。多媒体有图像、视频、背景音乐等多种形式,可以使网页更加丰富多彩,超链接可以使包含不同信息的网页链接在一起。

1. 插入图像

使用标签可以为网页添加.gif、.jpg、.png等格式的图像,主要属性如下。

- (1) src。指定图像的源文件路径,可以使用相对路径、绝对路径或 URL。
- (2) width。指定图像的宽度,单位为像素。
- (3) height。指定图像的高度,单位为像素。
- (4) hspace。指定图像水平方向的边沿空白,以免文字或其他图片过于贴近,单位为像素。
- (5) vspace。指定图像垂直方向的边沿空白,单位为像素。
- (6) border。指定图像边框厚度。

(7) align。当文字与图像并排放置时,指定图像与文本行的对齐方式,其属性值可取 top(与文本行顶部对齐)、center(水平居中对齐)、middle(垂直居中对齐)、bottom(底部对齐,默认值)、left(图像左对齐)、right(图像右对齐)。

(8) alt。这是用以描述该图像的文字,图像不能显示时将显示该属性值;当鼠标移至图像上时,将该属性值作为提示信息显示。

【例 3-6】 插入图像(imgTag.html)。

```
<html>
  <head>
    <title>插入图像</title>
  </head>
  <body bgcolor="pink">
    小鸭!
  </body>
</html>
```

运行效果如图 3-6 所示。

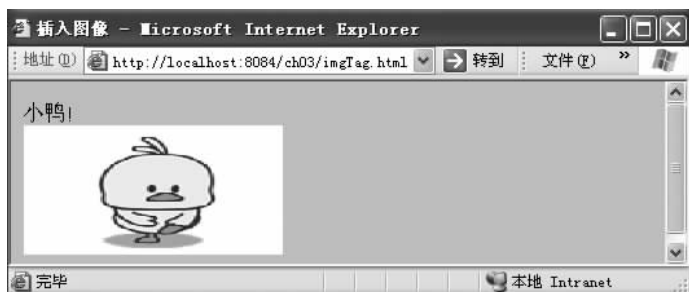


图 3-6 插入图像显示效果

2. 插入视频

使用标签可以在网页中插入视频,此时标签常用属性如下。

- (1) dynsrc。指定视频的源文件路径。
- (2) loop。指定视频循环播放次数。值为-1 或 infinite 时,表示无限次循环播放。
- (3) loopdelay。指定两次播放的时间间隔。

(4) start。指定何时开始播放视频。其属性值可取 fileopen(默认值,即在链接到含该标签的页面时开始播放)、mouseover(把鼠标移到播放区域之上时才开始播放)。

- (5) controls。指定在视频窗口下附加播放控制条。该属性直接添加,不用赋值。

【例 3-7】 插入视频(imgTag1.html)。

```
<html>
  <head>
    <title>插入视频</title>
  </head>
  <body bgcolor="pink">
```

```

        backkom 熊
    </body>
</html>

```

运行效果如图 3-7 所示。



图 3-7 插入视频显示效果

3. 插入背景音乐

使用<bgsound>标签可以在网页中添加 .wav、.mid、.mp3 等格式的背景音乐,主要属性如下。

- (1) balance。指定音乐的左右均衡。
- (2) delay。指定播放延时。
- (3) loop。指定音乐循环播放次数。值为 -1 或 infinite 时,表示无限次循环播放。
- (4) src。指定音乐源文件的路径。
- (5) volume。指定音量。

【例 3-8】 插入背景音乐(soundTag.html)。

```

<html>
    <head>
        <title>音乐无限</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    </head>
    <body bgcolor="pink">
        <br>
        <br>
        <h2 align="center">笔记</h2>
        
        <bgsound src="image/笔记.mp3" loop="1">
    </body>
</html>

```

运行效果如图 3-8 所示。



图 3-8 插入背景音乐显示效果

4. 插入超链接

创建超链接就是在当前页面与其他页面间建立链接,使用户可以从一个页面直接跳转到其他的页面、图像或服务器。在 HTML 中,文本和图像都可以用作锚点。一个超链接的基本格式如下:

```
<a href="资源地址">超链接文本或图像</a>
```

其中,<a>标签对用来创建超链接,<a>的主要属性如下。

(1) href。指定链接地址。若是链接到网站外部,必须为 URL 地址;若是链接到网站内部页面,只需指明该页面的绝对路径或相对路径。

(2) target。指定显示链接目标的窗口,其值可取_blank(浏览器总在一个新打开、未命名的窗口中载入目标文档)、_parent(目标文档载入当前窗口的父窗口中)、_self(默认值,目标文档载入并显示在当前窗口中)、_top(清除当前窗口所有被包含的框架并将目标文档载入整个浏览器窗口)。

例如:

```
<a href="http://www.zzuli.edu.cn/" target="_blank">学校网站</a>
<a href="http://www.jsp.com" target="_blank">JSP 学习网站</a>
```

3.1.5 表格

表格是一种能够有效地描述信息的组织方式,由行、列、单元格组成,可以很好地控制页面布局,所以在网页中应用非常广泛。许多网站都用多重表格来构建网站的总体布局,固定文本或图像的输出,还可以任意进行背景和前景颜色的设置。

在 HTML 中,使用<table></table>标签对来进行一个完整表格的声明,使用<tr></tr>标签对定义表格中的一行,使用<td></td>标签对定义行中的一个单元格,使用<th></th>定义表格中列的标题单元格。<tr></tr>只能放在<table></table>标签对之间使用,<td></td>、<th></th>也只有放在<tr></tr>标签对之间才是

有效的。表格定义基本格式如下：

```
<table>
  <tr>
    <th>表格第一列的标题</th>
    <th>表格第二列的标题</th>
    ⋮
  </tr>
  <tr>
    <td>表格第一行的第一个单元格内容</td>
    <td>表格第一行的第二个单元格内容</td>
    ⋮
  </tr>
  <tr>
    <td>表格第二行的第一个单元格内容</td>
    <td>表格第二行的第二个单元格内容</td>
    ⋮
  </tr>
  ⋮
</table>
```

1. <table> 常用属性

(1) border。设置表格边框的宽度,值为非负整数,若为 0 表示边框不可见,单位为像素。

(2) cellspacing。设置单元格边框到表格边框的距离,单位为像素。

(3) cellpadding。设置单元格内文字到单元格边框的距离,单位为像素。

(4) width。设置表格宽度。其值可为整数,单位为像素,如 100 表示 100 像素;也可以是相对页面宽度的百分比,如 20%表示表格宽度为整个页面宽度的 20%。

(5) height。设置表格高度,取值方式与 width 一致。

(6) bgcolor。设置表格背景色。其值可以是十六进制代码,也可以是英文单词,如 silver 为银色。

(7) bordercolor。设置表格边框颜色。

(8) align。设置表格在水平方向的对齐方式,其值可为 left、right、center。

(9) valign。设置表格在垂直方向的对齐方式,其值可为 top、middle、baseline。

2. <tr> 常用属性

(1) bordercolor。设置该行的外边框颜色。

(2) bgcolor。设置该行单元格的背景颜色。

(3) height。设置该行的高度。

(4) align。设置该行各单元格的内容在水平方向的对齐方式,其值可为 left、right、center。

(5) valign。设置该行各单元格的内容在垂直方向的对齐方式,其值可为 top、middle、bottom。

3. <td>常用属性

- (1) colspan。设置单元格所占的列数,默认值为1。
- (2) rowspan。设置单元格所占的行数,默认值为1。
- (3) background。设置单元格背景图像。
- (4) width。设置单元格宽度。

<th></th>定义的列标题的文字以粗体方式显示,其属性使用与<td>一致,在表格的定义语法中,也可以不使用<th>定义标题单元格。

【例 3-9】 创建表格(tableTag.html)。

```
<html>
<head>
  <title>创建表格</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body bgcolor="pink">
  <table border="1" width="80%" cellpadding="2">
    <tr valign="middle">
      <th colspan="2">Java 方向</th>
      <th colspan="2">软件测试</th>
      <th colspan="2">.NET 方向</th>
    </tr>
    <tr align="center">
      <td>Java 程序设计</td>
      <td>数据库技术</td>
      <td>Java 程序设计</td>
      <td>数据库技术</td>
      <td>C# 程序设计</td>
      <td>数据库技术</td>
    </tr>
    <tr align="center">
      <td>JSP 程序设计</td>
      <td>Java EE 技术</td>
      <td>JSP 程序设计</td>
      <td>软件测试理论</td>
      <td>ASP 程序设计</td>
      <td>ADO.NET</td>
    </tr>
  </table>
</body>
</html>
```

运行效果如图 3-9 所示。

3.1.6 框架

在设计网站整体结构布局时,框架也是经常被使用的一种标签,主要用来分割窗口和插

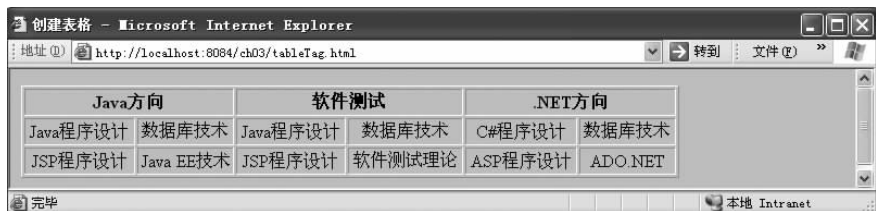


图 3-9 表格显示效果

入浮动窗口,使同一个浏览器窗口同时显示多个网页,如果能有效地运用将有助于提高网页的浏览效率。

1. 框架结构文件格式

框架将浏览器窗口分成多个子窗口,每个子窗口可以单独显示一个 HTML 文档,各个子窗口也可以相关联地显示某一个内容,例如,可以将目录放在一个子窗口,而将文件内容显示在另一个子窗口。框架结构文件格式如下:

```
<html>
<head>
    <title>...</title>
</head>
<frameset>
    <frame src="url">
    :
    <frameset>
        <frame src="url">
        :
    </frameset>
</frameset>
</html>
```

2. 框架结构基本标签

(1) `<noframes></noframes>`。

该标签对中的内容显示在不支持框架的浏览器窗口中。标签对中的内容可以是因浏览器太旧而无法支持 frame 功能时给出的提示性语句,也可以是没有 frame 语法的普通版本的 HTML 文档。这样,不支持 frame 功能的浏览器,便会自动显示没有 frame 语法的网页。

(2) `<frameset></frameset>`。

该标签对用来定义一个框架结构容器,即用来定义网页被分割成几个子窗口,各个子窗口是如何排列的。该标签对放在 HTML 文档的 `<body></body>` 标签对的外边,也可以嵌套在其他 `<frameset></frameset>` 标签对实现网页多重框架结构。`<frameset>` 常用属性包括如下。

① rows。在垂直方向将浏览器窗口分割成多个子窗口,即浏览器中所有子窗口从上到下排列,同时设置每个子窗口所占的高度。该属性值可以是百分数(子窗口高度相对页面高

度的相对值)、整数(绝对像素值)或星号(*)，其中星号代表那些未说明高度的空间，如果同一个属性中出现多个星号，则将剩下的未被说明高度的空间平均分配。各个子窗口高度之间用逗号分隔。

② cols。在水平方向将浏览器窗口分割成多个子窗口，即浏览器中所有子窗口从左到右排列，同时设置每个子窗口所占的宽度。该属性取值方式与 rows 一致。

例如：

<frameset rows="*,*,*">	在浏览器窗口垂直方向有 3 个子窗口，每个子窗口的高度占整个浏览器窗口高度的 1/3
<frameset cols="40%,*,*">	在浏览器窗口水平方向有 3 个子窗口，第一个子窗口宽度占整个浏览器窗口宽度的 40%，剩下的空间平均分配给另外两个子窗口
<frameset rows="40%,*"cols="50%,*,200">	总共有 6 个子窗口，先是在第一行中从左到右排列 3 个子窗口，然后在第二行中从左到右再排列 3 个子窗口，即两行三列

rows 属性说明框架横向分割的情况，cols 属性说明框架纵向分割的情况，所以使用<frameset></frameset>标签对时 rows 和 cols 这两个属性必须至少选择一个，否则浏览器只显示一个子窗口即一个网页内容，<frameset></frameset>标签对也就起不到任何作用了。

- ③ border。设置子窗口边框宽度。
- ④ frameborder。设置子窗口是否显示边框。
- ⑤ onload。设置框架被载入时引发的事件。
- ⑥ onunload。设置框架被卸载时引发的事件。

(3) <frame>。

<frame>标签放在<frameset></frameset>之间，用来定义框架结构中某一个具体的子窗口。常用属性如下。

- ① src。设置该子窗口中将要显示的 HTML 文件地址，取值可以是 URL 地址，也可以是相对路径或绝对路径。
 - ② name。设置子窗口的名字。
 - ③ scrolling。指定子窗口是否显示滚动条，取值可以是 yes(显示)、no(不显示)或 auto(根据窗口内容自动决定是否显示滚动条)。
 - ④ noresize。指定窗口不能调整大小，该属性直接加入标签中即可使用，不需赋值。
- src 和 name 这两个属性必须赋值。

3. target 属性

在框架结构子窗口的 HTML 文档中如果含有超链接，当用户单击该链接时，目标网页显示的位置由 target 属性指定，若没有指定则在当前子窗口打开。target 属性常用格式如下。

```
<a href="目标网页地址" target="显示目标网页的子窗口名字">超链接文字</a>
```

例如，框架中定义了一个子窗口 main，在 main 中显示 jc.htm 网页，则代码为

```
<frame src="jc.htm" name="main">，
```

若 jc.htm 中有一个超链接,在单击该链接后,网页 new.htm 将要显示在名为 main 的子窗口中,则代码为

```
<a href="new.htm" target="main">需要链接的文本</a>
```

【例 3-10】 框架结构使用(framesetTag.html)。

```
<html>
<head>
  <title>框架使用</title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<frameset cols="70%, * " frameborder="yes" border="10">
  <frameset rows="60%, * " frameborder="yes">
    <frame src="top.html" name="top" scrolling="auto" noresize>
    <frame src="bottom.html" name="bottom" scrolling="no" noresize>
  </frameset>
  <frame src="right.html" name="right" scrolling="no" noresize>
</noframes>
  对不起,您的浏览器版本太低!
</noframes>
</frameset>
</html>
```

top.html 的代码如下:

```
<html>
<head>
  <title></title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body bgcolor="pink">
  这是 top 页面
</body>
</html>
```

bottom.html 的代码如下:

```
<html>
<head>
  <title></title>
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
</head>
<body bgcolor="pink">
  这是 bottom 页面
</body>
</html>
```

right.html 的代码如下:

```

<html>
  <head>
    <title></title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body bgcolor="pink">
    这是 right 页面
  </body>
</html>

```

运行效果如图 3-10 所示。

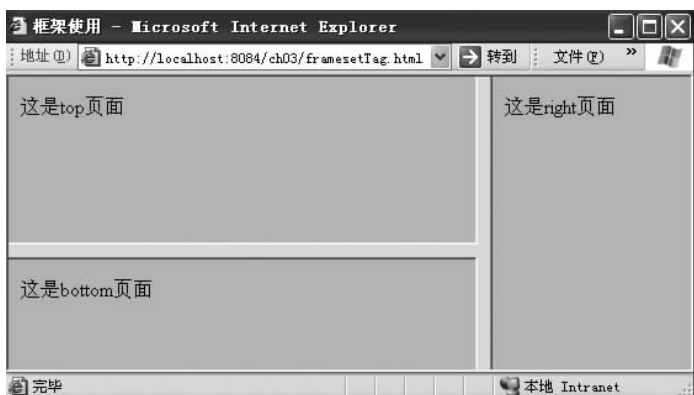


图 3-10 框架结构使用显示效果

3.1.7 表单

表单在网页中用来供用户填写信息,以实现服务器获取用户信息,使网页具有交互功能。一般是将表单设计在一个 HTML 文档中,当用户填写完信息执行提交(Submit)操作后,表单的内容就从客户端浏览器传送到服务器上,经过服务器上的处理程序处理后,再将用户所需信息传送回客户端浏览器上,这样网页就具有了交互性。

网页中的可输入项、选择项等实现数据采集功能的控件所组成的就是表单,表单一般由表单标签、表单域、表单按钮组成。表单标签包含处理表单数据所用 CGI 程序的 URL 以及数据提交到服务器的方法;表单域包含了文本框、密码框、隐藏域、多行文本框、复选框、单选框、下拉选择框和文件上传框等用于用户输入和交互的控件;表单按钮包括提交按钮、复位按钮和一般按钮,用于将数据传送给服务器上的 CGI 脚本或者取消输入,还可以用表单按钮来控制其他处理工作。

1. 表单标签

`<form></form>` 标签对用来创建一个表单,即定义表单的开始和结束位置。该标签对属于容器标签,表单里所有数据采集功能的控件需要定义在该标签对之间。表单标签基本语法结构如下:

```

<form action="url" method="get|post" name="value" onsubmit="function" onreset=
"function" target="window"></form>

```

(1) action 属性。

设置服务器上用来处理表单数据的处理程序地址,处理程序可以是 JSP 程序、CGI 程序、ASP、NET 程序等,该属性值可以是 URL 地址也可以是电子邮件地址。

例如,action="http://localhost/sample/ShopCart.jsp",表示当用户提交表单后,将调用服务器上的 JSP 页面 ShopCart.jsp 来处理用户的输入。

另外,采用电子邮件地址的格式是 action="mailto:接收用户输入信息的邮件地址"。

例如,action="mailto:youremail@zzuli.edu.cn",表示把用户的输入信息发送到电子邮件地址 youremail@zzuli.edu.cn。

(2) method 属性。

设置处理程序从表单中获得信息的方式,取值可为 get 或 post。

get 方法将表单中的输入信息作为查询字符串附加在 action 指定的地址后(中间用“?”隔开)传送到服务器。查询字符串使用 key="value"的形式定义,如果有多个域,中间用 & 隔开,如 http://localhost/sample/ShopCart.jsp? flowerid="0169"&count="16",问号后面的即为查询字符串。get 方法在浏览器的地址栏中以明文形式显示表单中各个表单域的值,对数据的长度有限制。

post 方法将表单中用户输入的数据进行包装,按照 HTTP 中的 post 方式传送到服务器,且对数据的长度基本没有限制,目前大都采用此方式。

(3) name 属性。设置表单的名字。

(4) onsubmit、onreset 属性。设置在单击了 submit 或 reset 按钮后要执行的脚本函数名。

(5) target 属性。设置显示表单内容的窗口名。

HTML 对表单的数量没有限制,但一个页面中如果有太多的表单将不易于阅读,因此需合理设置。

2. 表单域

(1) 单行输入域。

<input> 标签用来定义单行输入域,用户可在其中输入单行信息。主要属性如下。

① type 属性。设置输入域的类型,取值如表 3-1 所示。

表 3-1 单行输入域类型

type 属性取值	输入域类型
<input type="text" size="" maxlength="">	单行文本输入区域,size 与 maxlength 属性用来定义此区域显示的尺寸大小与输入的最大字符数
<input type="submit">	将表单内容提交给服务器的按钮
<input type="reset">	将表单内容全部清除、重新填写的按钮
<input type="checkbox" checked>	一个复选框,checked 属性用来设置该复选框默认状态是否被选中
<input type="hidden">	隐藏区域,用户不能在其中输入,用来预设某些要传送的信息
<input type="image" src="url">	使用图像来代替 submit 按钮,图像的源文件名由 src 属性指定,用户单击后,表单中的信息和单击位置的 X、Y 坐标一起传送给服务器

续表

type 属性取值	输入域类型
<input type="password">	输入密码的区域,当用户输入密码时,区域内将会显示 * 号代替用户输入的内容
<input type="radio" checked>	单选按钮,checked 属性用来设置该单选按钮默认状态是否被选中

- ② name 属性。设置输入域的名字。
 - ③ value 属性。设置输入域的默认值。
 - ④ align 属性。设置输入域位置,可取值 left(靠左)、right(靠右)、middle(居中)、top(靠上)、bottom(靠底)。
 - ⑤ onclick 属性。设置按下按钮后执行的脚本函数名。
- (2) 多行输入域。

<textarea></textarea> 标签对用来定义多行文本输入域,主要属性如下。

- ① name 属性。设置输入域名字。
- ② rows 属性。设置输入域的行数。
- ③ cols 属性。设置输入域的列数。
- ④ wrap 属性。设置是否自动换行,属性值可取 off(不自动换行)、hard(自动硬回车换行,换行标记一同被传送到服务器)、soft(自动软回车换行,换行标记不会被传送到服务器)。

(3) 选择域。

<select></select> 标签对用来建立一个下拉列表,<option> 标签用来定义下拉列表中的一个选项,用户可以从列表中选择一项或多项。

- ① <select></select> 的主要属性。
 - name 属性。设置下拉列表的名字。
 - size 属性。设置下拉列表中选项的个数,默认值为 1。
 - multiple 属性。表示下拉列表支持多选。
- ② <option> 的主要属性。
 - selected 属性。表示当前选项被默认选中。
 - value 属性。设置当前选项的值,在该项被选中之后,该项的值将被送到服务器。

3. 表单按钮

<button></button> 标签对用于定义提交表单内容给服务器的按钮,主要属性有 type 和 accesskey。

- (1) type 属性。设置按钮类型,属性值可取 button(一般按钮)、reset(复位按钮)、submit(提交按钮)。它们与<input>中同名的属性具有相同的功能。
- (2) accesskey 属性。设置按钮热键,即按下 Alt 键的同时按下该属性值所对应的键便可以快速定位到该按钮。

【例 3-11】 表单使用(formTag.html)。

```
<html>
<head>
```