

第 3 章 HTML 与 CSS 简介

为了设计出美观、漂亮、赏心悦目的网站,HTML 和 CSS 是必不可少的技术,将它们和 JSP 结合起来能够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的基础知识。

本章主要内容如下所示。

- (1) HTML。
- (2) CSS。

3.1 HTML 页面的基本构成

超文本标记语言(Hypertext Markup Language,HTML)是目前网络上应用最为广泛的标记语言,也是构成网页文档的主要语言。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文本文件,通常带有 html 或 htm 的文件扩展名,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。

下面通过一个 HTML 页面来了解其基本构成。页面实例运行效果如图 3-1 所示。



图 3-1 页面实例运行效果

【例 3-1】 唐诗欣赏页面实例(TangShiTML.html)。

```

<html>
  <head>
    <title>唐诗欣赏页面实例</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <h1>欢迎光临本网站</h1>
    <br>
    <hr>
    <font face="楷体_GB3212" size="2" color="blue">
      这是唐诗欣赏页面, 欢迎你的访问!
    </font>
    <p>请欣赏唐诗</p>
    <pre>
      《客中行》
      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>
    <hr>
    <!--
      《客中行》, 作者李白(701—762), 字太白, 号青莲居士, 又号“谪仙人”(贺知章评李白,
      李白亦自诩), 祖籍陇西成纪(今甘肃省静宁县), 隋朝末年, 因避乱(一说被流放)迁徙到中
      亚细亚碎叶城(今吉尔吉斯斯坦北部托克马克附近), 李白即诞生于此。他的一生, 绝大部
      分在漫游中度过。五岁时, 其家迁入绵州彰明县(今四川江油)。二十岁时只身出川, 开始
      了广泛漫游, 南到洞庭湘江, 东至吴、越。他到处游历, 希望结交朋友, 干谒社会名流, 从而
      得到引荐, 一举登上高位, 去实现政治理想和抱负。可是, 十年漫游, 却一事无成。他又继
      续北上太原、长安, 东到齐、鲁各地, 并寓居山东任城(今山东济宁)。这时他已结交了不少
      名流, 创作了大量优秀诗篇, 诗名满天下。天宝初年, 由道士吴筠推荐, 唐玄宗召他进京, 命
      他供奉翰林。不久, 因权贵的谗言, 于天宝三四年间(公元 744 或 745 年), 被排挤出京。
      唐朝文宗御封李白的诗歌、裴旻的剑舞、张旭的草书为“三绝”。李白的剑术在唐朝可
      排第二(在裴旻之下)。与李商隐、李贺三人并称唐代“三李”。集诗人、神仙家、驴友、纵横
      家为一身的伟大天才。民间流传, 高力士曾为他脱靴, 杨贵妃曾为他磨墨。杜甫赞曰: “笔
      落惊风雨, 诗成泣鬼神。”
    -->
  </body>
</html>

```

在上边这段代码中, 一些字母或单词用<>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, 把标签放置在该处前面, 浏览器就知道用户希望下面的内容应如何显示。

标签分为单标签和双标签。单标签只需单独使用就能完整地表达意思, 控制网页效果, 如<meta>、
、<hr>。双标签成对使用, 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。如<body>和</body>就是一对双标签。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,如<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)。

HTML 不区分大小写,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,如<body><form></body></form>,应改为<body><form></form></body>。

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>

这个标签告诉浏览器本文件是 HTML 文档。<html>用于 HTML 文档的最前处,用来标识 HTML 文档的开始;</html>放在 HTML 文档的最后处,用来标识 HTML 文档的结束。

2) <head></head>

这个标签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,如日期和时间、网页描述、关键词、页面刷新等。

例如:

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

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、默认的脚本语言、网页自动刷新时间等。

属性 charset 的作用是指定当前文档所使用的字符编码为 UTF-8,也就是支持中文简体字符。根据这一行代码,浏览器就可以识别出网页中的中文字符。

3) <body></body>

这个标签中的内容是 HTML 文档的主体部分,可包含<p></p>、<h1></h1>、
、<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 1~6,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,其取值如下所示。

- (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】 标题标签实例(title.html)。

```
<html>
  <head>
    <title>标题标签实例</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <hr>
    <h1 align="center">一级标题的效果</h1>
    <h2>二级标题的效果</h2>
    <h3>三级标题的效果</h3>
    <h4>四级标题的效果</h4>
    <h5>五级标题的效果</h5>
    <h6>六级标题的效果</h6>
    <hr>
  </body>
</html>
```

运行效果如图 3-2 所示。可以看出,每级标题的字体均为“黑体”。

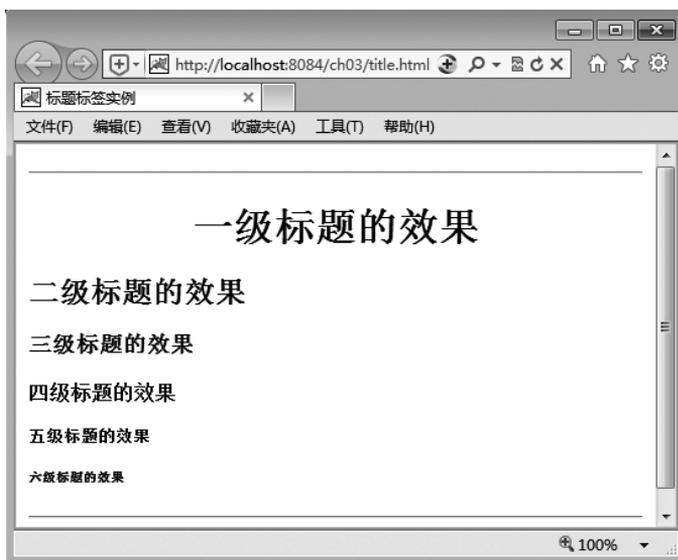


图 3-2 标题标签实例运行效果

4. 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>`

标签对的外部,将创建一个大的回车换行,即
前面和后面文本的行与行之间的距离比较大;若放在<p></p>的内部,则
前面和后面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,具有以下多种属性,用于设置水平线的宽度、长度及显示效果等。

(1) size: 设置水平线的粗细,默认单位是像素。

(2) width: 设置水平线的宽度,默认单位是像素,也可以使用相对屏幕的百分比表示。

(3) noshade: 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: 设置水平线的对齐方式。

(5) color: 设置水平线的颜色。

例如:

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

5. 文字格式标签

标签通过设置属性来控制文字的样式、大小和颜色,属性功能如下所示。

(1) face: 设置文字样式。

(2) size: 设置文字大小,值为整数,分为 7 个级别,默认文字大小为 3 磅。

(3) color: 设置文字颜色。

6. 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</p>可以省略。<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,其取值如下所示。

(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允许有其他内容。

为了防止文档出错,尽量不要省略结尾标记</p>。

7. 预定格式标签

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,可以使用<pre></pre>标签。使用该标签时,默认字号为 10 磅。

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。

3.2 HTML 常用标签

本节介绍在设计 HTML 页面时一些常用的 HTML 标签。

3.2.1 列表标签及其应用实例

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、无序列表和自定义列表。

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,其取值如下所示。

- (1) 1: 编号为数字,默认值,如 1、2、3...
- (2) A: 编号为大写英文字母,如 A、B、C...
- (3) a: 编号为小写英文字母,如 a、b、c...
- (4) I: 编号为大写罗马字符,如 I、II、III...
- (5) i: 编号为小写罗马字符,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 属性的作用与中的一致,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 3-3】 有序列表实例(olTag.html)。

```
<html>
  <head>
    <title>有序列表实例</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    Java 方向核心专业课程:
    <ol type="1">
      <li>Java 程序设计</li>
      <li>JSP 程序设计</li>
      <li>Java Web 框架技术(Struts、Spring、Hibernate)</li>
      <li>数据库技术</li>
    </ol>
```

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

运行效果如图 3-3 所示。



图 3-3 有序列表实例运行效果

2. 无序列表

无序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。无序列表用开始,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的类型;中的 type 属性用于设置当前列表项前的项目符号的类型。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 3-4】 无序列表实例(ulTag.html)。

```
<html>
  <head>
    <title>无序列表实例</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    Java 方向其他专业课程:
    <ul type="disc">
      <li>计算机网络</li>
      <li type="circle">数据结构</li>
      <li type="square">专业英语</li>
    </ul>
  </body>
</html>
```

```

    </ul>
  </body>
</html>

```

运行效果如图 3-4 所示。



图 3-4 无序列表实例运行效果

3. 自定义列表

除了上述两种列表外,在实际应用中还可以根据需求自定义列表,实现一种分两层的项目清单,其语法如下:

```

<dl>
  <dt>第一个列表项</dt>
  <dd>对第一个列表项的说明</dd>
  <dt>第二个列表项</dt>
  <dd>对第二个列表项的说明</dd>
  :
</dl>

```

自定义列表用<dl>开始,以</dl>结束,给每一个列表项加上了一段说明性文字,说明性文字独立于列表项另起一行显示。其中,<dt></dt>标签用来定义列表项;<dd></dd>标签用来对列表项进行说明。

【例 3-5】 自定义列表实例(dlTag.html)。

```

<html>
  <head>
    <title>自定义列表实例</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    Java 方向核心专业课程介绍:
    <dl>
      <dt>Java 程序设计</dt>
      <dd>该课程主要讲述...</dd>
      <dt>JSP 程序设计</dt>
      <dd>该课程主要讲述...</dd>
      <dt>Java Web 框架技术(Struts、Spring、Hibernate)</dt>
      <dd>该课程主要讲述...</dd>
    </dl>
  </body>
</html>

```

```

        <dt>数据库技术</dt>
        <dd>该课程主要讲述…</dd>
    </dl>
</body>
</html>

```

运行效果如图 3-5 所示。

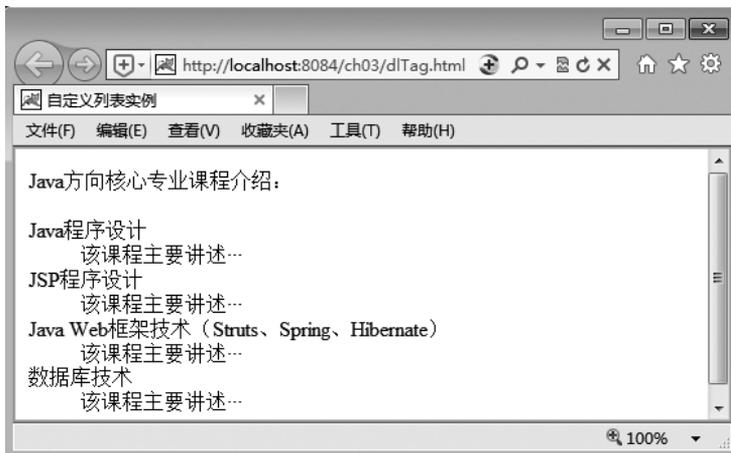


图 3-5 自定义列表实例运行效果

3.2.2 多媒体和超链接标签及其应用实例

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,可以把网页更加丰富多彩,超链接可以把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,的主要属性如下所示。

- (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: 指定图像的边框厚度。
- (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: 用于描述该图像的文字,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】 插入图像实例(imgTag.html)。

```

<html>
  <head>
    <title>插入图像实例</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    听得到的微笑!
    
  </body>
</html>

```

运行效果如图 3-6 所示。



图 3-6 插入图像实例运行效果

2. 插入视频

使用<embed>标签可以在网页中插入视频,主要属性如下所示。

- (1) src: 指定视频的源文件路径。
- (2) width: 视频宽度。
- (3) height: 视频高度。

【例 3-7】 插入视频实例(embedTag.html)。

```

<html>
  <head>
    <title>插入视频实例</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    笨笨熊的故事!
    <br>
    <embed src="image/笨笨熊.wmv" width="500" height="350"/>
  </body>

```

```
</html>
```

运行效果如图 3-7 所示。鼠标未移到播放区域之上时不播放,鼠标移到播放区域之上时将播放“笨笨熊.wmv”。

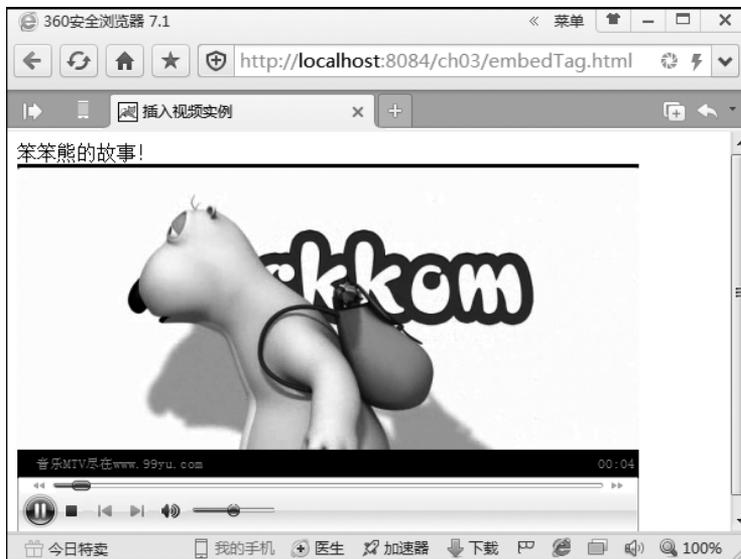


图 3-7 插入视频实例运行效果

备注: 如果播放视频时出现无法播放的情况,首先检查一下是否安装了播放器,其次可以换一下浏览器再播放。有的 IE 浏览器版本无法播放。

3. 插入背景音乐

使用<embed>标签还可以在网页中添加.wav、.mid、.mp3 等格式的背景音乐。

【例 3-8】 插入背景音乐实例(embedTag1.html)。

```
<html>
  <head>
    <title>插入背景音乐实例</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <h1 align="center">传奇</h1>
    <hr>
    
    <hr>
    <embed src="image/传奇.mp3"/>
  </body>
</html>
```

运行效果如图 3-8 所示。

4. 插入超链接

创建超链接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页面、图像或服务器。基本格式如下:



图 3-8 插入背景音乐实例运行效果

`超链接文本及图像`

其中, `<a>` 标签对用来创建超链接。 `<a>` 的主要属性如下所示。

(1) href: 指定链接地址。若是链接到网站外部页面, 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.tup.tsinghua.edu.cn/" target="_blank">清华大学出版社</a>
```

3.2.3 表格标签及其应用实例

表格是一种能够有效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, 并可以任意进行背景和前景颜色的设置。

在 HTML 中, 使用 `<table></table>` 标签来进行一个完整表格的声明, 使用 `<tr></tr>` 标签定义表格中的一行, 使用 `<th></th>` 定义表格中列标题单元格, 使用 `<td></td>` 标签定义行中的一个单元格。 `<tr></tr>` 只能放在 `<table></table>` 标签之间使用, `<td></td>`、`<th></th>` 也只有放在 `<tr></tr>` 标签之间才是有效的。表

格定义基本格式如下所示。

```
<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相对值, 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, 也可以是英文字母, 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 和 center。
- (9) valign: 设置表格在垂直方向的对齐方式, 其值可为 top、middle 和 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 和 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 和 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】 表格实例(tableTag.html)。

```
<html>
  <head>
    <title>表格实例</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <table border="1" width="90%" bordercolor="red" cellpadding="2">
      <tr height="50" valign="middle">
        <th width="33%" colspan="2">Java 方向</th>
        <th width="36%" colspan="2">软测方向</th>
        <th width="36%" colspan="2">.NET 方向</th>
      </tr>
      <tr align="center">
        <td width="16%">Java 程序设计</td>
        <td width="16%">JSP 程序设计</td>
        <td width="17%">Java 程序设计</td>
        <td width="17%">JSP 程序设计</td>
        <td width="17%">C# 程序设计</td>
        <td width="17%">ASP.NET</td>
      </tr>
      <tr align="center">
        <td width="16%">Java Web 框架技术</td>
        <td width="16%">数据库技术</td>
        <td width="17%">软件测试理论</td>
        <td width="17%">软件测试技术</td>
        <td width="17%">.NET Framework 技术</td>
        <td width="17%">数据库技术</td>
      </tr>
    </table>
  </body>
</html>
```

运行效果如图 3-9 所示。

3.2.4 表单标签及其应用实例

表单在网页中用来供用户填写信息,以实现服务器获得用户信息,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,再将用户所需要的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



图 3-9 表格实例运行效果

表单标签、表单域、表单按钮组成。表单标签包含了处理表单数据所用 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。表单标签的基本语法结构如下：

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

(1) `action`：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:8080/ch03/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:youreemail@163.com"`，表示把用户的输入信息发送到电子邮件地址 `youreemail@163.com`。

(2) `method`：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:8080/ch03/ShopCart.jsp?flowerid="0169"&.count="16"`，问号后面的内容即为查询字符串。`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: 设置表单的名字。
- (4) onsubmit、onreset: 设置在单击 submit 或 reset 按钮后要执行的脚本函数名。
- (5) target: 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,因此需要合理设置。

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。主要属性如下所示。

- (1) type: 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
<input type="password">	输入密码的区域,当用户输入密码时,区域内将会用 * 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: 设置输入域的名字。
- (3) value: 设置输入域的默认值。
- (4) align: 设置输入域位置,可取值为 left(靠左)、right(靠右)、middle(居中)、top(靠上)和 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,主要属性如下所示。

- (1) name: 设置输入域名字。
- (2) rows: 设置输入域的行数。
- (3) cols: 设置输入域的列数。

(4) wrap: 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> 标签的主要属性如下所示。

- ① name: 设置下拉列表的名字。
- ② size: 设置下拉列表中选项的个数,默认值为 1。
- ③ multiple: 表示下拉列表支持多选。

(2) <option> 的主要属性如下所示。

- ① selected: 表示当前选项被默认选中。
- ② value: 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: 设置按钮的类型,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: 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 3-10】 表单实例(formTag.html)。

```
<html>
  <head>
    <title>表单实例</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <h1 align="left">用户注册</h1>
    <form name="" action="" method="post">
      <table border="1">
        <tr>
          <td>用户名: </td>
          <td><input type="text" name="userName"></td>
        </tr>
        <tr>
          <td>密 码: </td>
          <td><input type="password" name="userPassword"></td>
        </tr>
        <tr>
          <td>确认密码: </td>
          <td><input type="password" name="userPassword1"></td>
        </tr>
        <tr>
          <td>密码提示问题: </td>
          <td><input type="text" name="passwordHint"></td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```
<tr>
  <td>真实姓名: </td>
  <td><input type="text" name="name"></td>
</tr>
<tr>
  <td>性 别: </td>
  <td>
    <input type="radio" name="sex" value="男">男
    <input type="radio" name="sex1" value="女">女
  </td>
</tr>
<tr>
  <td>出生日期: </td>
  <td>
    <select name="select" size="1">
      <option selected>1988</option>
      <option>1989</option>
      <option>1990</option>
      <option>1991</option>
      <option>1992</option>
      <option>1993</option>
      <option>1994</option>
      <option>1995</option>
      <option>1996</option>
      <option>1997</option>
    </select>
    年<select name="select" size="1">
      <option selected>1</option>
      <option>2</option>
      <option>3</option>
      <option>4</option>
      <option>5</option>
      <option>6</option>
      <option>7</option>
      <option>8</option>
      <option>9</option>
      <option>10</option>
      <option>11</option>
      <option>12</option>
    </select>
  </td>
</tr>
<tr>
  <td>证件类型: </td>
  <td>
    <select name="select">
      <option value="xsx">学生证
      <option value="sfz" selected>身份证
      <option value="jgz">军官证
    </select>
  </td>
</tr>
```

```

        </tr>
        <tr>
            <td>证件号码: </td>
            <td><input type="text" name="userID"></td>
        </tr>
        <tr>
            <td><input type="submit" name="submit" value="提交"></td>
            <td><input type="reset" name="reset" value="取消"></td>
        </tr>
    </table>
</form>
</body>
</html>

```

运行效果如图 3-10 所示。

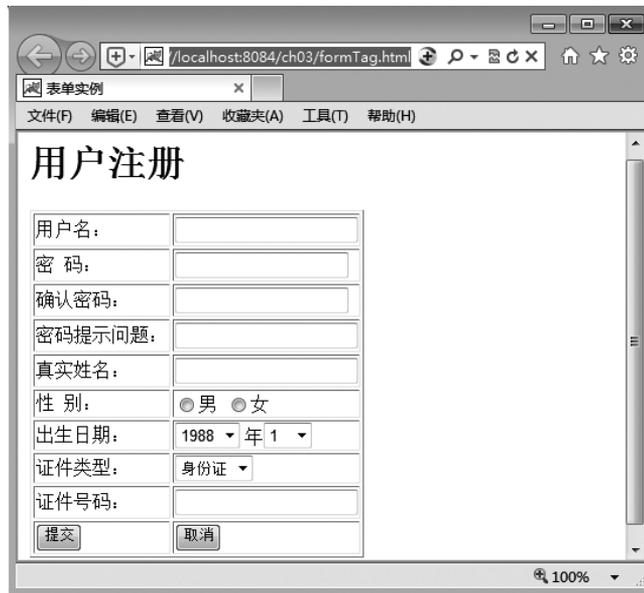


图 3-10 表单实例运行效果

3.2.5 框架标签及其应用实例

在进行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,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所示。

```

<html>
    <head>

```

```

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

```

2. 框架结构基本标签

1) <frameset></frameset>

该标签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。可以嵌套在其他<frameset></frameset>标签对中实现网页多重框架结构。<frameset>常用属性如下。

(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百分比)、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>标签时 rows 和 cols 这两个属性至少必须选择一个,否则浏览器只显示一个子窗口,即一个网页内容,<frameset></frameset>标签也就没有起任何作用。

(3) border: 设置子窗口的边框宽度。

(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。

2) <frame>

<frame> 标签放在<frameset></frameset>之间,用来定义框架结构中某一个具体