

# 第 3 章 掌握必要的 HTML 语言

网页的本质是超文本标记语句，通过结合与使用其他 Web 技术，如脚本语言、公共网关接口、组件等，可以创造出功能更为强大的网页。因而，超文本标记语言是万维网（Web）编程的基础。本章我们就来学习 HTML 的语法格式及使用方法。

本章主要内容：

- HTML 的语法结构
- 认识 HTML 的基本标记
- HTML5 简介

## 3.1 认识 HTML

HTML 又称为超文本标记语言（HyperText Markup Language）是为“网页创建和其他可在网页浏览器中看到的信息”设计的一种标记语言。HTML 被用来结构化信息，例如标识标题、段落和列表等，也可用来在一定程度上描述文档的外观和语义。

### 3.1.1 HTML 语言简介

---

HTML，即超文本标记语言，是目前因特网上用于编写网页的主要语言。之所以称为超文本，是因为文本中包含了所谓的“超链接”点。这也是 HTML 获得广泛应用的最重要的原因之一。但它并不是一种程序设计语言，而是一种规范，一种标准，通过标记符号来标记网页中要显示的各个部分。网页文件本身是一种文本文件，通过在文本文件中添加标记符，来告诉浏览器如何显示其中的内容（如：文字如何处理，画面如何安排，图像如何显示等）。

HTML 文件是一种可以用任何文本编辑软件创建的 ASCII 码文件。常见的文本编辑软件如“记事本”“写字板”等都可以编写 HTML 文件，在保存时以 .htm 或 .html 作为文件的扩展名。当使用浏览器打开这些文件时，浏览器对文件进行解释，浏览者就可以从浏览器窗口中看到网页要显示的内容。

浏览器按顺序阅读网页文件，然后根据标记符解释和显示其标记的内容；对书写出错的标记将不指出其错误，且继续执行解释过程。网页制作者只能通过显示效果来分析出错原因和出错位置。需要注意的是，不同的浏览器对同一标记符可能会有不完全相同的解释，因而可能会有不同的显示结果。

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，HTML 就是使用标记标签来描述网页的。下面介绍它的语法结构。

## 3.1.2 HTML 的基本语法结构

### 1. 标记和属性

HTML 文件由标记和被标记的内容组成。标记被封装在“<”和“>”所构成的一对尖括号中，如<P>，在 HTML 中标识段落。标记分为单标记和双标记。双标记用一对标记对所标识的内容进行控制，包括开始标记符和结束标记符；而单标记则不需要标记符成对出现。这两种标记的格式如下。

单标记格式：<标记>内容

双标记格式：<标记>内容</标记>

标记规定的是信息内容，但标记中的文本、图像等信息内容将怎样显示，还需要在标记后面加上相关的属性来指定。标记的属性用来描述对象的特征，控制标记内容的显示和输出格式。标记通常都有一系列属性。设置属性的一般语法结构为：

<标记 属性 1=属性值 属性 2=属性值……>内容</标记>

例如，要将页面中段落文字的颜色设置为红色，则应设置其 color 属性的值为 red，具体格式是：

```
<p color=red>内容</p>
```

要说明的是，并不是所有的标记都有属性，例如换行标记<br/>就没有属性。一个标记可以有多个属性，在实际使用时可根据需要设置其中一个或多个属性，这些属性之间没有先后顺序之分。

### 2. HTML文件结构

HTML 文件必须以<html>标记开始，以</html>标记结束，其他标记都包含在这里面。在这两个标记之间，主要包括 HTML 文件的文件头和文件体两个部分。下面以一个简单的网页为例来说明。

打开“记事本”程序，输入以下内容。输入后的结果如图 3-1 所示。

```
<html>
<head>
<title>欢迎光临我的小站! </title>
</head>

<body>
<center> <font size= 5 color= blue > 借一场花开，躲一场人海 </font> </center>
</body>
</html>
```

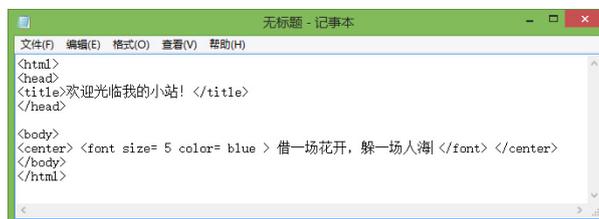


图 3-1 添加代码

然后执行“文件”→“另存为”命令，这时会弹出“另存为”对话框，在“保存类型”右边的下拉菜单里选择“所有文件”。这一点很重要，否则，它将被保存为纯文本文件，而不是HTML文件。将输入的内容保存成名为“index.htm”的HTML文件，也可以命令为“index.html”，然后设置保存路径并保存。双击保存的HTML文件，在浏览器中打开，如图3-2所示。

可以看到，整个文档内容包含在HTML标记中。<html>和</html>成对出现，<html>处于文件的第一行，表示文档的开始，</html>位于文件最后一行，表示文档的结束。

文件头部分用<head>标记标识，处于第二层。<head>和</head>成对出现，包含在<html>和</html>中。<head>和</head>之间包含的是文件标题标记，它处于第三层。网页的标题内容“欢迎光临我的网站”写在<title>与</title>之间。文件头部分用于对网页信息进行说明，在文件头部分定义的内容通常不在浏览器窗口中出现。

文件体部分用<body>标记标识，它也处于第二层，包含在<html>标记内，在层次上和文件头标记并列。网页的内容，如文字、图片、动画等就写在<body>和</body>之间，它是网页的核心。从图3-2中可以看到，浏览器顶端标题栏中显示的文字就是网页的标题，是<title>和</title>之间的内容；而源代码<body>和</body>间的内容显示在浏览器窗口中。

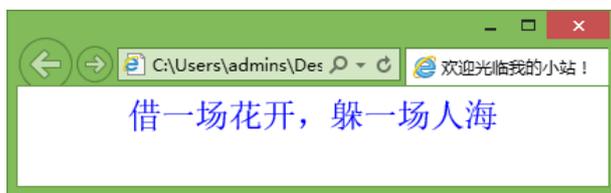


图3-2 打开制作的网页

## 3.2 HTML的基本标记

在上一节中介绍了HTML文件的基本结构及所涉及的标记，下面将介绍一些基本的HTML标记。网页中常见的标记有文本标记、图像标记、表格标记和超链接标记等。

### 3.2.1 文本标记

在HTML中，使用<h>标记来标识文档中的标题和副标题，n代表从1~6的数字，数字越大所标记的标题字越小。

用<h>标记设置标题的示例代码如下。

```
<html>
<head>
<title>网页设计</title>
</head>
```

```

<body>
<h1>标题文字 h1</h1>
<h2>标题文字 h2</h2>
<h3>标题文字 h3</h3>
<h4>标题文字 h4</h4>
<h5>标题文字 h5</h5>
<h6>标题文字 h6</h6>
</body>
</html>

```

在浏览器中的显示效果如图 3-3 所示。

### 3.2.2 段落标记

段落文本使用<p>标记定义，文本内容写在开始标记<p>和结束标记</p>之间。属性 align 可以用来设置段落文本的对齐方式，其属性值有 3 个，分别是 left（左对齐）、center（居中对齐）、right（右对齐）。当没有设置 align 属性时，默认为左对齐。

用<p>标记设置段落文本的示例代码如下。

```

<html>
<head>
<title>段落文字的对齐方式</title>
</head>

<body>
<p >段落文本</p>
<p align="left">段落文本</p>
<p align="center">段落文本</p>
<p align="right">段落文本</p>
</body>
</html>

```

在浏览器中的显示效果如图 3-4 所示。

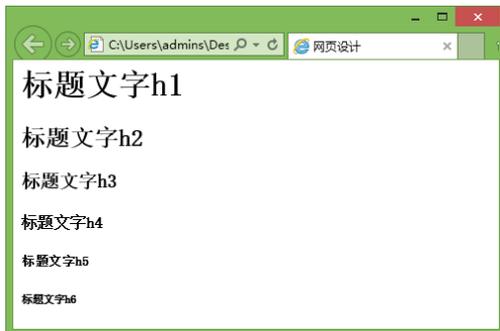


图 3-3 标题文字标记运行效果



图 3-4 段落文本标记运行效果

可以用来进行段落处理的还有强制换行标记<br/>。<br/>放在一行的末尾，可以使后面的文字、图片、表格等显示于下一行。它和<p>标记的区别是，用<br/>分开的两行之间不会有空行，而<p>却会有空行。

例如输入以下代码：

```
<html>
<head>
<title>强制换行标记</title>
</head>

<body>
<p>段落文本</p>
<p>段落文本</p>
强制换行标记<br />强制换行标记
</body>
</html>
```

在浏览器中的显示效果如图 3-5 所示。

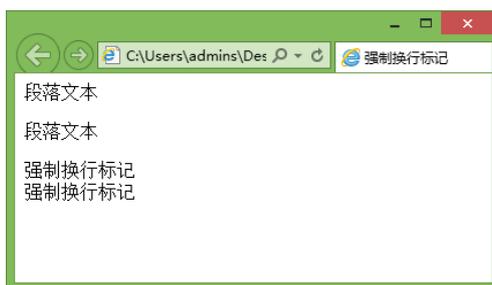


图 3-5 强制换行标记运行效果

### 3.2.3 文本格式标记

文本显示的格式使用<font>标记来标识。<font>标记常用的属性有3个，size用来设置文本字号大小，取值是0~7；color用来设置文本颜色，取值是十六进制表示的RGB颜色；face用来设置字体，取值可以是宋体、黑体等。

用<font>标记设置文本格式的示例代码如下。

```
<html>
<head>
<title>文本格式标记</title>
</head>

<body>
<font size="3">这是 size="3"的文本</font><br />
<font size="6">这是 size="6"的文本</font><br />
<font color="#000000">这是 color="#000000"的文本</font><br />
<font color="red">这是 color="red"的文本</font><br />
<font face="黑体">这是 face="黑体"的文本</font><br />
<font face="宋体">这是 face="宋体"的文本</font><br />
</body>
</html>
```

在浏览器中的显示效果如图 3-6 所示。

为了让文字格式有变化，或者为了强调某部分文字，可以设置其他的文本格式标记，有以下几种。

<code>&lt;b&gt; &lt;/b&gt;</code>	文本以加粗形式显示
<code>&lt;i&gt; &lt;/i&gt;</code>	文本以斜体形式显示
<code>&lt;u&gt; &lt;/u&gt;</code>	文本加下划线显示
<code>&lt;strong&gt; &lt;/strong&gt;</code>	文本加重显示（通常是黑体加粗）

用文本格式标记来设置文本格式的示例代码如下。

```
<html>
<head>
<title>文本格式标记</title>
</head>

<body>
<b>加粗字</b><br />
<i>斜体字</i><br />
<u>加下划线</u><br />
<strong>强调文本</strong>
</body>
</html>
```

在浏览器中的显示效果如图 3-7 所示。

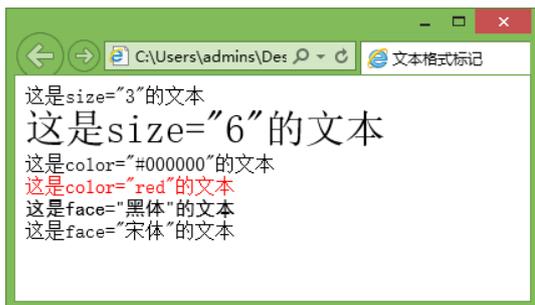


图 3-6 文本格式标记运行效果

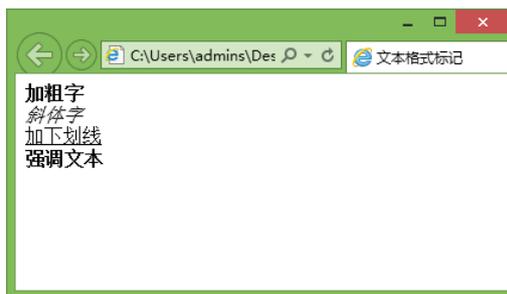


图 3-7 文本格式标记运行效果

### 3.2.4 图像标记

在页面中插入图像使用

图像标记的使用示例，代码如下。

```
<html>
<head>
<title>图像标记</title>
```

```
</head>

<body>

</body>
</html>
```

在浏览器中的显示效果如图 3-8 所示。



图 3-8 图像标记运行效果

### 3.2.5 超链接标记

超链接是指从一个页面跳转到另一个页面，或者是从页面的一个位置跳转到另一个位置的链接关系，它是 HTML 的关键技术。链接的目标除了页面还可以是图片、多媒体、电子邮件等。有了超链接，各个孤立的页面才可以相互联系起来。

#### 1. 页面链接

在 HTML 中创建超链接需要使用<a>标记，具体格式是：

```
<a href="URL" target="_blank">链接</a>
```

href 属性控制链接到的文件地址，target 属性控制目标窗口。target="blank"表示在新窗口打开链接的文件，如果不设置 target 属性则表示在原窗口打开链接文件。在<a>和</a>之间可以用任何可单击的对象作为超链接的源，如文字或图像。

常见的超链接是指向其他网页的超链接，如果超链接的目标网页位于同一站点，可以使用相对 URL；如果超链接的目标网页位于其他位置，则需要使用绝对 URL。例如，以下的 HTML 代码分别显示了创建绝对超链接和相对超链接的方法。

```
<a href="http://www.baidu.com" >百度搜索</a>
<a href="test2.htm" >网页 test2</a>
```

#### 2. 锚记链接

如果要对同一网页的不同部分进行链接，需要建立锚记链接。

设置锚记链接，首先应为页面中要跳转到的位置命名。命名时使用<a>标记的 `name` 属性。此处<a>与</a>之间可以包含内容，也可以不包含内容。

例如，在页面开始处用以下语句进行标记。

```
<a name="top" >顶部</a>
```

对页面进行标记后，可以用<a>标记设置指向这些标记位置的超链接。如果在页面开始处标记了“top”，则可以用以下语句进行链接。

```
<a href="#top" >返回顶部</a>
```

这样设置后，当用户在浏览器中单击文字“返回顶部”时，将显示“顶部”文字所在的页面内容。

要注意的是，应用锚记链接时要将其 `href` 的值指定为符号#接锚记名称的形式。如果将 `href` 的值指定为一个单独的#，则表示空链接，不做任何跳转。

### 3. 电子邮件链接

如果将 `href` 属性的值指定为“mailto:电子邮件地址”，则可以获得指向电子邮件的超链接。例如，使用以下 HTML 代码可以设置电子邮件超链接。

```
<a href="mailto:it_book@126.com" >it_book 的邮箱</a>
```

当浏览用户单击该超链接后，系统将自动启动邮件客户程序，并将指定的邮件地址填写到“收件人”栏中，此时用户可以编辑并发送邮件。

## 3.2.6 列表标记

列表分为有序列表、无序列表和定义列表。有序列表是指带有序号标志（如数字）的列表；没有序号标志的列表为无序列表；定义列表则用于对列表项做出解释。

### 1. 有序列表

有序列表的标记是<ol>，其列表项标记是<li>，具体格式是：

```
<ol type="序号类型">
  <li>列表项 1 </li>
  <li>列表项 1 </li>
  <li>列表项 1 </li>
</ol>
```

`type` 属性可取的值有以下几种。

- 1: 序号为数字。
- A: 序号为大写英文字母。
- a: 序号为小写英文字母。
- I: 序号为大写罗马字母。
- i: 序号为小写罗马字母。

使用有序列表的示例代码如下。

```
<html>
<head>
<title>有序列表</title>
</head>

<body>
<ol>
  <li>文字 </li>
  <li>图像 </li>
  <li>表格 </li>
  <li>表单 </li>
</ol>
<ol type="A">
  <li>文字 </li>
  <li>图像 </li>
  <li>表格 </li>
  <li>表单 </li>
</ol>
</body>
</html>
```

在浏览器中的显示效果如图 3-9 所示。

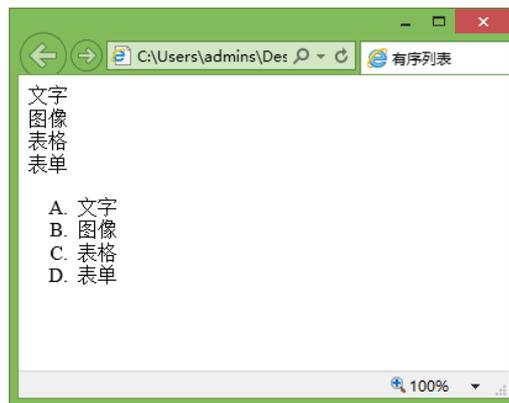


图 3-9 有序列表

## 2. 无序列表

无序列表的标记是<ul>，其列表项标记是<li>，具体格式是：

```
<ul type="符号类型">
  <li>列表项 1 </li>
  <li>列表项 1 </li>
  <li>列表项 1 </li>
</ul>
```

**type** 属性控制的是列表在排序时所使用的字符类型，可取的值有以下几种。

**disc:** 符号为实心圆。

**circle:** 符号为空心圆。

**square:** 符号为实心方点。

使用无序列表的示例代码如下。

```
<html>
<head>
<title>无序列表</title>
</head>

<body>
<ul type="circle">
  <li>文字 </li>
  <li>图像 </li>
  <li>表格 </li>
</ul>
<ul type="disc">
  <li>文字 </li>
  <li>图像 </li>
  <li>表格 </li>
</ul>
<ul type="square">
  <li>文字 </li>
  <li>图像 </li>
  <li>表格 </li>
</ul>
</body>
</html>
```

在浏览器中的显示效果如图 3-10 所示。

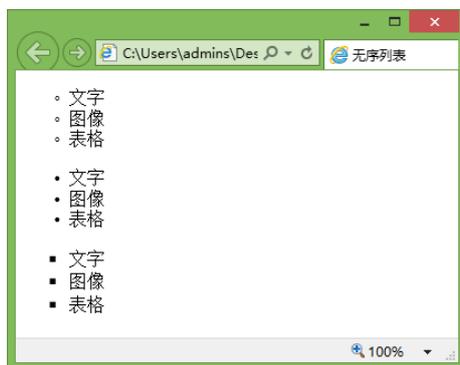


图 3-10 无序列表

### 3. 定义列表

定义型列表用在对列表项目进行简短说明的情况下，具体格式是：

```
<d1>
<dt></dt>
<dd></dd>
</d1>
```

定义列表在HTML中的标记是<dl>，列表项的标记是<dt>和<dd>。<dt>标记所包含的列表项目标识一个定义术语，<dd>标记包含的列表项目是对定义术语的定义说明。举个例子，在网页代码中插入下面的代码：

```
<dl>
  <dt>www</dt>
  <dd>World Wide Web 的缩写</dd>
  <dt>cn</dt>
  <dd>域名的后缀</dd>
</dl>
```

在浏览器中的显示效果如图 3-11 所示。



图 3-11 定义列表

### 3.2.7 表格标记

表格的主要用途是显示数据，它是进行信息管理的有效手段。表格通常由 3 部分组成，即行、列和单元格。使用表格会用到 3 个标记，即<table>、<tr>、<td>。<table>标识表格对象，<tr>标识表格中的行，<td>标识单元格，<td>必须包含在<tr>标记内。具体格式是：

```
<table>
  <tr><td>表项目 1</td>.....<td>表项目 n</td></tr>
  .....
  <tr><td>表项目 1</td>.....<td>表项目 n</td></tr>
</table>
```

表格的属性设置，如宽度、边框等包含在<table>标记内。如果要在页面中创建一个 3 行、3 列，宽度为 400，边框为 1 的表格，其示例代码如下。

```
<table width="400" border="1">
  <tr>
    <td>姓名</td>
    <td>学号</td>
    <td>成绩</td>
  </tr>
  <tr>
    <td>张三</td>
    <td>2015001</td>
    <td>98</td>
  </tr>
</table>
```



```

        <td>数学</td>
        <td>语文</td>
        <td>数学</td>
        <td>语文</td>
    </tr>
    <tr>
        <th>下午</th>
        <td>英语</td>
        <td>化学</td>
        <td>体育</td>
        <td>物理</td>
        <td>生物</td>
    </tr>
</table>
</body>
</html>

```

在浏览器中的显示效果如图 3-13 所示。



图 3-13 表格标记的运行效果

### 3.2.8 表单标记

表单在网络中的应用范围非常广，可以实现很多功能，如网站登录、账户注册等。表单是网页上的一个特定区域，这个区域由一对<form>标记定义。<form>标记声明表单，定义了采集数据的范围，也就是说，<form>与</form>里面包含的数据将被提交到服务器。组成表单的元素很多，常用的有文本框、单选按钮、复选框和按钮等。多数表单元素都由<input>标记定义，表单的构造方法则由 type 属性声明。不过下拉菜单和多行文本框这两个表单元素则是例外。常用的表单元素有下面几种。

- 文本框

文本框用来接受任何类型的文本的输入。文本框的标记为<input>，其 type 属性为 text。

- 复选框

复选框用于选择数据，它允许用户在一组选项中选择多个选项。复选框的标记也是<input>，它的 type 属性为 checkbox。

- 单选按钮

单选按钮也用于选择数据，不过在一组选项中只能选择一个选项。单选按钮的标记是<input>，它的 type 属性为 radio。

- 提交按钮

在单击提交按钮后，表单内容将被提交到服务器。提交按钮的标记是

提示：除了提交按钮，预定义的还有重置按钮。此外，制作者还可以通过自定义按钮赋予按钮其他功能。

- 多行文本框

多行文本框的标记是<textarea>，它可以创建一个对数据的量没有限制的文本框。通过 rows 属性和 cols 属性可定义多行文本框的宽和高，当输入内容超过其范围，该元素会自动出现一个滚动条。

- 下拉菜单

下拉菜单在一个滚动列表中显示选项值，用户可以从滚动列表中选择选项。下拉菜单的标记是<select>，它的选项内容用<option>标记定义。

读者学习了上面介绍的这些标记就可以创建表单，使网页能实现一些数据采集功能了。除了普通表单元素标记，还有一些很有用的表单标记，用来定义表单结构或者添加意义。

## 1. label

使用<label>标记可将文本与其他任何 HTML 对象或内部控件关联起来。无论用户单击<label>标识的文本还是 HTML 对象，引发和接收事件的行为是一致的。要使<label>标识的文本和 HTML 对象相关联，需将<label>的 for 属性设置为 HTML 对象的 ID 属性。<label>标记可以使表单组件增加可访问性。

例如，在页面中创建一个表单域，插入文本框、多行文本框和提交按钮 3 个表单元素。在每个表单元素的前面插入一个<label>标记，<label>标记内的文本为其后对应的表单元素的文字解释。设置每个<label>标记的 for 属性为对应表单元素的 ID。具体代码如下。

```
<html>
<head>
<title>表单标记</title>
</head>

<body>
<form method="post">
  <label for="name">姓名</label>
  <input name="name" type="text" id="name" /><br />
  <label for="comment">评论</label>
  <textarea cols="30" rows="5" name="comment" id="comment" >
  </textarea><br />
  <label for="submit"></label>
  <input name="submit" type="submit" id="submit" value="提交" />
</form>
</body>
</html>
```

在浏览器中的显示效果如图 3-14 所示。此时<label>标识的文本和表单元素相关联，

单击文本和单击相应的表单元素引发的事件相同。

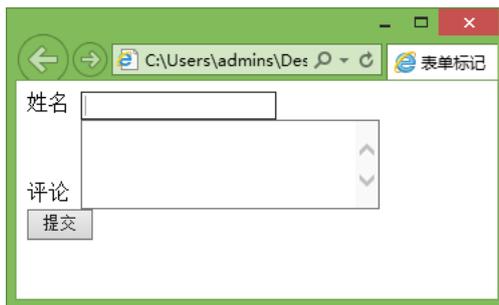


图 3-14 使表单中的文本与相应的表单元素相关联

## 2. fieldset

<fieldset>标记元素可以给<form>标记内的表单元素分组。一般情况下，在 CSS 中创建容器需要一个<div>标记，但使用<fieldset>标记可以在表单域内创建一个完美的容器。默认情况下，<fieldset>标记在内容周围画一个简单的边框，以定义分组的表单内容。

例如，在页面中创建一个表单，插入两对<fieldset>标记，将表单内容分成两组，代码如下。

```
<html>
<head>
<title>表单标记</title>
</head>

<body>
<form method="post">
<fieldset class="fieldset">
  <label for="name">姓名</label>
  <input name="name" type="text" id="name" /><br />
  <label for="comment">评论</label>
  <textarea cols="30" rows="5" name="comment" id="comment" >
  </textarea><br />
</fieldset>
<fieldset class="fieldset">
  <label for="name">姓名</label>
  <input name="name" type="text" class="text1" id="name" />
  <br /><label for="comment">评论</label>
  <textarea cols="30" rows="5" name="comment" id="comment" >
  </textarea><br />
</fieldset>
<label for="submit"></label>
  <input name="submit" type="submit" id="submit" value="提交" />
</form>
</body>
</html>
```

查看浏览效果，将如图 3-15 所示。两对<fieldset>标记将表单元素分成两组，两个组周围分别有一个简单的矩形边框。

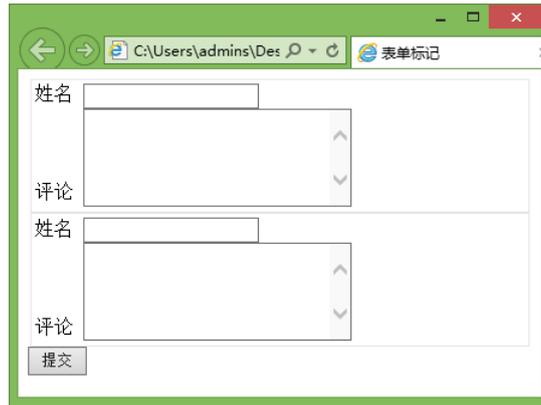


图 3-15 &lt;fieldset&gt;标记的运行效果

### 3. legend

<legend>标记的功能和表格中<caption>标记的功能相似，可用来描述它的父元素<fieldset>标记内的内容。一般情况下，浏览器会将<legend>标记内的文本放置在<fieldset>所标识对象边框的上方。

在上面例子的基础上，在两对<fieldset>标记内插入<legend>标记，并将描述文本放置在<legend>标记内。代码如下。

```
<html>
<head>
<title>表单标记</title>
</head>

<body>
<form method="post">
<fieldset class="fieldset">
<legend>评论 1</legend>
  <label for="name">姓名</label>
  <input name="name" type="text" id="name" /><br />
  <label for="comment">评论</label>
  <textarea cols="30" rows="5" name="comment" id="comment" >
  </textarea><br />
</fieldset>
<fieldset class="fieldset">
<legend>评论 2</legend>
  <label for="name">姓名</label>
  <input name="name" type="text" class="text1" id="name" />
  <br /><label for="comment">评论</label>
  <textarea cols="30" rows="5" name="comment" id="comment" >
  </textarea><br />
</fieldset>
<label for="submit"></label>
  <input name="submit" type="submit" id="submit" value="提交" />
</form>
</body>
</html>
```

查看浏览效果，如图3-16所示。两对<fieldset>标记将表单内容分成两组，<legend>标记内的描述文本放置在<fieldset>标识对象边框的上方。

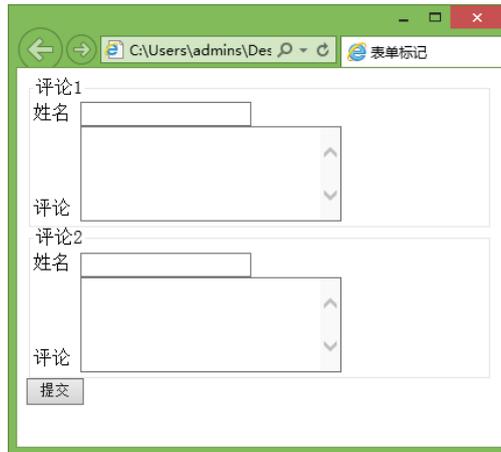


图 3-16 <legend>标记的运行效果

## 3.3 HTML5 简介

HTML5 是 HTML 的下一个主要修订版本，现在仍处于发展阶段。从广义上来说，HTML5 实际上指的是包括 HTML、CSS 和 JavaScript 在内的一套技术组合。和以前的版本不同，HTML5 并非仅仅用来标识 Web 内容，它的新使命是将 Web 带入一个成熟的应用平台。

### 3.3.1 HTML5 的语法变化

HTML5 中，语法发生了很大的变化。但是，HTML5 的“语法变化”和其他编程语言的语法变更意义有所不同。HTML 原本是通过 SGML (Standard Generalized Markup Language) 元语言来规定语法的。但是由于 SGML 的语法非常复杂，文档结构解析程序的开发也不太容易，多数 Web 浏览器不作为 SGML 解释器运行。因此，HTML 规范中虽然要求“应遵循 SGML 的语法”，但实际情况却是，对于 HTML 的执行在各浏览器之间并没有一个统一的标准。

在 HTML5 中，提高 Web 浏览器间的兼容性是 HTML5 要实现的重大目标。要确保兼容性，必须消除规范与实现的背离。因此，HTML5 需要重新定义新的 HTML 语法，即实现规范向实现靠拢。

由于文档结构解析的算法也有着详细的记载，使得 Web 浏览器厂商可以专注于遵循规范去进行实现工作。在新版本的 Firefox 和 WebKit (Nightly Builder 版) 中，已经内置了遵循 HTML5 规范的解释器。IE (Internet Explorer) 和 Opera 也为了提供兼容性更好的实现而紧锣密鼓地努力着。

### 3.3.2 HTML 5 中的标记方法

HTML 5 中的标记方法有三种。

#### 1. 内容类型(Content Type)

HTML 5 的文件扩展名与内容类型保持不变。也就是说，扩展名仍然为.html 或.htm，内容类型(Content Type)仍然为 text/html。

#### 2. DOCTYPE 声明

DOCTYPE 声明是 HTML 文件中必不可少的，它位于文件第一行。在 HTML 4 中，DOCTYPE 声明的方法如下。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

在 HTML 5 中，刻意不使用版本声明，声明文档将会适用于所有版本的 HTML。HTML 5 中的 DOCTYPE 声明方法（不区分大小写）如下。

```
<!DOCTYPE html>
```

另外，当使用工具时，也可以在 DOCTYPE 声明方式中加入 SYSTEM 识别符，声明方法如下面的代码所示。

```
<!DOCTYPE HTML SYSTEM"about: legacy-compat">
```

**提示：**在 HTML 5 中，DOCTYPE 声明方式允许不区分英文大小写，引号不区分是单引号还是双引号。

#### 3. 字符编码的设置

字符编码的设置方法有些新的变化。在以往设置 HTML 文件的字符编码时，要用到如下<meta>标记。

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

在 HTML5 中，可以使用<meta>标记的新属性 charset 来设置字符编码，如下面的代码所示。

```
<meta charset="UTF-8">
```

以上两种方法都有效。因此也可以继续使用前者的方法（通过 content 属性来设置），但要注意不能同时使用。

### 3.3.3 HTML 5 中新增加的标记

在 HTML 5 中新增了以下标记。

### 1. section

<section>标记标识页面中如章节、页眉、页脚或页面中其他部分的一个内容区块。

语法格式: <section>…</section>

示例:

```
<section> 欢迎学习 HTML 5 </section>
```

### 2. article

<article>标记标识页面中的一块与上下文不相关的独立内容, 例如博客中的一篇文章或报纸中的一篇文章。

语法格式: <article>…</article>

示例:

```
<article>HTML 5 华丽蜕变</article>
```

### 3. aside

<aside>标记用于标识<article>标记内容之外的, 并且与<article>标识内容相关的一些辅助信息。

语法格式: <aside>…</aside>

示例:

```
<aside>HTML 5 将开启一个新的时代</aside>
```

### 4. header

<header>标记标识页面中一个内容区块或整个页面的标题。

语法格式: <header>…</header>

示例:

```
<header> HTML 5 应用与开发指南</header>
```

### 5. hgroup

<hgroup>标记用于组合整个页面或页面中一个内容区块的标题。

语法格式: <hgroup>…</hgroup>

示例:

```
<hgroup>系统功能管理</hgroup>
```

### 6. footer

<footer>标记标识整个页面或页面中一个内容区块的脚注。

语法格式: <footer>…</footer>

示例:

```
<footer>李彬<br/>
    135*****1<br/>
    2011-10-1
</footer>
```

## 7. nav

<nav>标记用于标识页面中导航链接的部分。

语法格式: <nav></nav>

## 8. figure

<figure>标记标识一段独立的流内容。一般标识文档主体流内容中的一个独立单元。

语法格式: <figure>...</figure>

示例:

```
<figure>
<figcaption>HTML5</figcaption>
<p>HTML5 是当今最流行的网络应用技术之一</p>
</figure>
```

## 9. video

<video>标记用于定义视频, 例如电影片段或其他视频流。

语法格式: <video>...</video>

示例:

```
<video src="movie.ogv", controls="controls">video 标记应用示例</video>
```

## 10. audio

在 HTML 5 中, <audio>标记用于定义音频, 例如音乐或其他音频流。

语法格式: <audio>...</audio>

示例:

```
<audio src="someaudio.wav">audio 标记应用示例</audio>
```

## 11. embed

<embed>标记用来插入各种多媒体。多媒体文件的格式可以是 Midi、WAV、AIFF、AU 和 MP3 等。

语法格式: <embed/>

示例:

```
<embed src="horse.wav"/>
```

## 12. mark

<mark>标记主要用来在视觉上向用户呈现那些需要突出显示或高亮显示的文字。

语法格式: `<mark>...</mark>`

示例:

```
<mark>HTML5 技术的应用</mark>
```

### 13. progress

`<progress>`标记标识运行中的进程, 可以使用`<progress>`标记来显示 JavaScript 中耗费时间函数的进程。

语法格式: `<progress>...</progress>`

### 14. meter

`<meter>`标记定义度量衡。仅用于已知最大值和最小值的度量。

语法格式: `<meter>...</meter>`

### 15. time

`<time>`标记标识日期或时间, 也可以同时标识两者。

语法格式: `<time>...</time>`

### 16. wbr

`<wbr>`标记表示软换行。`<wbr>`标记与`<br>`标记的区别是, `<br>`标记表示此处必须换行; 而`<wbr>`标记的意思是浏览器窗口或父级元素的宽度足够宽时(没必要换行时), 不进行换行, 而当宽度不够时, 主动在此处进行换行。`<wbr>`标记对英文这类拼音型语言的作用很大, 但是对于中文却没多大用处。

语法格式: `...<wbr>...`

示例:

```
<p>To learn AJAX, you must be fami<wbr>liar with the XMLHttpRequest<wbr>Request Object. </p>
```

### 17. canvas

`<canvas>`标记用于标识图形, 例如图表和其他图像。这个标记本身没有行为, 仅提供一块画布, 但它把一个绘图 API 展现给客户端 JavaScript, 以使脚本能够把要绘制的图像绘制到画布上。

语法格式: `<canvas></canvas>`

示例:

```
<canvas id="myCanvas"width="300"height="300"></canvas>
```

### 18. command

`<command>`标记标识命令按钮, 例如单选按钮或复选框。

示例:

```
<command onclick=" cut () " label=" cut " >
```

## 19. details

`<details>`标记通常与`<summary>`标记配合使用，标识用户要求得到并且可以得到的细节信息。`<summary>`标记提供标题或图例。标题是可见的，用户单击标题时，会显示出细节信息。`<summary>`标记是`<details>`标记的第一个子标记。

语法格式：`<details>…</details>`

示例：

```
<details>
<summary>HTML 5 应用实例</summary>
本节将教您如何学习和使用 HTML 5
</details>
```

## 20. datalist

`<datalist>`标记用于标识可选数据的列表。`<datalist>`标记通常与`<input>`标记配合使用，可以制作出具有输入值的下拉列表。

语法格式：`<datalist>…</datalist>`

除了以上这些之外，还有`<datagrid>`、`<keygen>`、`<output>`、`<source>`、`<menu>`等标记，这里就不再一一讲解了，有兴趣的读者可以阅读 HTML 5 专业书籍进行学习。

### 3.3.4 HTML 5 中新增加的属性

HTML 5 新增了很多属性，下面简单介绍其中一些属性。

#### 1. 表单相关属性

新增的与表单相关的属性如下。

**autofocus** 属性。该属性可以用在`<input>`(`type=text`)、`<select>`、`<textarea>`与`<button>`标记当中。`autofocus`属性可以让标识对象在打开画面时自动获得焦点。

**placeholder** 属性。该属性可以用在`<input>`标记(`type=text`)和`<textarea>`标记当中。使用该属性时会对用户的输入进行提示，通常用于提示用户可以输入的内容。

**form** 属性。该属性用在`<input>`、`<output>`、`<select>`、`<textarea>`、`<button>`和`<fieldset>`标记当中。

**required** 属性。该属性用在`<input>`标记(`type=text`)和`<textarea>`标记当中。该属性用于在用户提交表单的时候进行检查，检查该标记对象内一定要有输入内容。

**formaction**、**formenctype**、**formmethod**、**formnovalidate** 与 **formtarget**。这些属性可以用在`<input>`标记与`<button>`标记中，用来重载`<form>`标记的 `action`、`enctype`、`method`、`novalidate` 与 `target` 属性。

**novalidate** 属性。该属性可以用在`<input>`、`<button>`和`<form>`标记中，用来取消提交时

进行的有关检查，即表单可以被无条件地提交。

## 2. 与链接相关的属性

新增的与链接相关的属性如下。

**media** 属性。该属性用在<a>与<area>标记中用来规定目标 URL 是用什么类型的媒介进行优化的。

**hreflang** 属性与 **rel** 属性。用在<area>标记中，以保持与<a>标记、<link>标记的一致性。

**sizes** 属性。用在<link>标记中，用于指定关联图标(icon 标识)的大小，通常可以与<icon>标记结合使用。

**target** 属性。用在<base>标记中，主要目的是保持与<a>标记的一致性。

## 3. 其他属性

**charset** 属性。用在<meta>标记中，用来为文档字符编码的指定提供一种良好的方式。

**type** 和 **label** 属性。用在<meta>标记中。**label** 属性为菜单定义一个可见的标注，**type** 属性可以使菜单以上下文菜单、工具条或列表菜单 3 种形式出现。

**scoped** 属性。用在<style>标记中用来规定样式的作用范围。

**async** 属性。用在<script>标记中，用于定义脚本是否异步执行。

为了方便读者学习，下面将 HTML 的标记及其含义制作成了表格，如表 3-1 所示。

表 3-1 HTML 标记及其功能描述

标 记	描 述
<!--.....-->	定义注释
<!DOCTYPE>	定义文档类型
<a>	定义超链接
<abbr>	定义缩写
<address>	定义地址元素
<area>	定义图像映射中的区域
<article>	定义 article
<aside>	定义页面内容之外的内容
<audio>	定义声音内容
<b>	定义粗体文本
<base>	定义页面中所有链接的基准 URL
<bdo>	定义文本显示的方向
<blockquote>	定义长的引用
<body>	定义 body 元素
 	插入换行符
<button>	定义按钮
<canvas>	定义图形
<caption>	定义表格标题
<cite>	定义引用
<code>	定义计算机代码文本
<col>	定义表格列的属性

续表

标 记	描 述
<colgroup>	定义表格列的分组
<command>	定义命令按钮
<datagrid>	定义树列表 (tree-list) 中的数据
<datalist>	定义下拉列表
<datatemplate>	定义数据模板
<del>	定义删除文本
<details>	定义元素的细节
<dialog>	定义对话 (会话)
<div>	定义文档中的一个部分
<dfn>	定义自定义项目
<dl>	定义自定义列表
<dt>	定义自定义的项目
<dd>	定义自定义的描述
<em>	定义强调文本
<embed>	定义外部交互内容或插件
<event-source>	为服务器发送的事件定义目标
<fieldset>	定义组合表单中的相关元素
<figure>	定义媒介内容的分组, 以及它们的标题
<footer>	定义 section 或 page 的页脚
<form>	定义表单
<h1>~<h6>	定义标题 1 到标题 6
<head>	定义关于文档的信息
<header>	定义 section 或 page 的页眉
<hr>	定义水平线
<html>	定义 HTML 文档
<i>	定义斜体文本
<iframe>	定义行内的子窗口 (框架)
<img>	定义图像
<input>	定义输入域
<ins>	定义插入文本
<kbd>	定义键盘文本
<label>	定义表单控件的标注
<legend>	定义 fieldset 中的标题
<li>	定义列表的项目
<link>	定义资源引用
<m>	定义有记号的文本
<map>	定义图像映射
<menu>	定义菜单列表
<meta>	定义元信息
<meter>	定义预定义范围内的度量
<nav>	定义导航链接
<nest>	定义数据模板中的嵌套点
<object>	定义嵌入对象

续表

标 记	描 述
<ol>	定义有序列表
<optgroup>	定义选项组
<option>	定义下拉列表中的选项
<output>	定义输出的一些类型
<p>	定义段落
<param>	定义对象参数
<pre>	定义预格式化文本
<progress>	定义任何类型的任务的进度
<q>	定义短的引用
<rule>	为升级模板定义规则
<samp>	定义样本计算机代码
<script>	定义脚本
<section>	定义 section
<select>	定义可选列表
<small>	定义小号文本
<source>	定义媒介源
<span>	定义文档中的 section
<strong>	定义强调文本
<style>	定义样式定义
<sub>	定义上标文本
<sup>	定义下标文本
<table>	定义表格
<thead>	定义表头
<tbody>	定义表格的主体
<tr>	定义表格行
<th>	定义表头
<td>	定义表格单元
<tfoot>	定义表格的脚注
<textarea>	定义 textarea
<time>	定义日期/时间
<title>	定义文档的标题
<ul>	定义无序列表
<var>	定义变量
<video>	定义视频

### 3.4 习题

1. 制作“列表”网页，页面效果如图3-17所示。
2. 新建一个网页，输入文本，进行如下设置：
  - (1) 网页背景颜色：#036876。

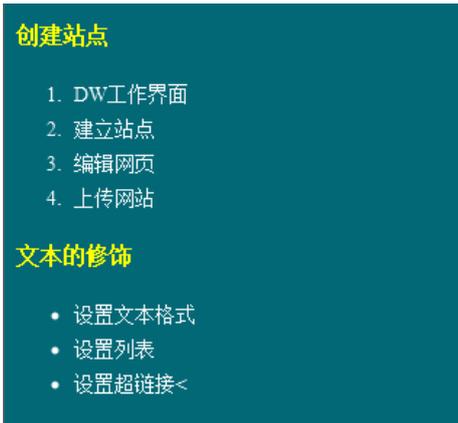


图 3-17 网页效果

(2) 网页标题：列表。

(3) 开头的两行文字设置为标题 h3，其余文本为白色，大小为 14 像素，行高为 24 像素。