第 3 章 HTML5 表格与表单

学习目标

- 掌握表格标签的用法。
- 掌握天气情况表格统计实例的实现方式。
- 掌握表单标签的用法。
- 掌握图书库存信息录入表单实例的实现方式。

在一个网页中表格与表单的应用是十分常见的。表格是由行和列组成的结构化数据集(表格数据),用于呈现数据或统计信息,可以让数据的显示变得十分规整有条理,提高数据的可读性。表单作为用户与网页之间重要的交互工具,可用于收集用户的资料信息,如网页中的用户登录、注册页面,以及一些收集用户反馈信息的调查表。表单的出现使网页从单向的信息传递发展到能够与用户进行交互对话。本章将带领读者重点学习表格与表单的制作。

3.1 HTML5 表格

在生活中,经常使用表格来统计数据和信息,以便更清晰地显示数据或信息。同理,在制作网页时,为了有条理地显示网页中的元素,可以使用表格对网页进行布局和规划,为浏览者展示大量清晰的排列数据。表格在网页中的应用是极为广泛的。

3.1.1 表格的基本结构

<table>标签用于定义一个表格。<tr>标签用于定义表格中的行,可以有一行或多行,嵌套在<table>标签中。<td>标签用于定义表格中的单元格(列),一行里可以有一个或多个单元格(列),此标签需嵌套在<tr>标签中。

1. 基本语法格式

一个最基本的表格由<table>、<tr>和<td>这 3 个标签构成,其语法格式如下。

```
<table>
  <tr>
      <td>单元格内容 1 </td>
      <td>单元格内容 2 </td>
      …
  </tr>
    …
</table>
```

除了以上 3 个标签之外,常用的表格标签还有< caption >、< th >等。< caption >标签用于定义表格的标题,< caption >标签必须紧随< table >标签之后,且每个表格只能定义一个标题,通常这个标题会被居中显示于表格之上。< th >标签用于定义表格内的表头单元格,需要在< tr >标签内部使用。

< th >和< td >是两种不同类型的单元格。< th >是表头单元格,用于存放表头信息,< th >内部的文本通常显示为居中的粗体文本。< td >是标准单元格,用于存放表格数据,< td >内部的文本通常显示为左对齐的普通文本。

2. 演示说明

中国的美食文化历史悠久,菜系可分为四大菜系:鲁菜、川菜、粤菜、淮扬菜。使用表格基本标签创建一个关于中国四大菜系的表格,具体代码如例 3-1 所示。

【例 3-1】 创建表格。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>创建表格</title>
</head>
<body>
    <!-- 定义表格,并添加边框 -->
    <table border="1">
        <!-- 定义表格的标题 -->
        <caption>中国四大菜系</caption>
        <!-- 定义表格内的行 -->
        <tr>
            <!-- 定义表格内的表头单元格 -->
            <th>编号</th>
            <th>菜系</th>
            <th>起源</th>
            <th>特点</th>
            <th>代表菜</th>
        </tr>
        <tr>
            <!-- 定义表格内的标准单元格 -->
            <td>01</td>
            <td>鲁菜</td>
            <td>山东的齐鲁风味</td>
            <td>清香、鲜嫩、味醇</td>
            <td>爆炒腰花、糖醋鲤鱼</td>
        </tr>
        <tr>
            <td>02</td>
            <td>川菜</td>
            <td>四川、重庆</td>
            <td>麻、辣、鲜、香</td>
            <td>麻婆豆腐、毛血旺</td>
        </tr>
        <!-- 此处省略雷同代码 -->
    </table>
</body>
</html>
```

使用表格基本标签创建表格，并使用 border 属性为表格添加边框，运行效果如图 3-1 所示。

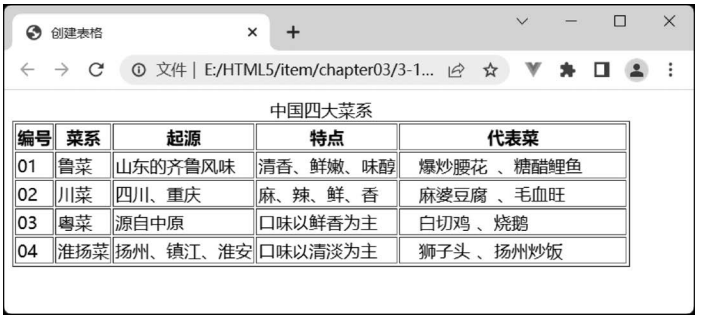


图 3-1　创建表格的运行效果

若去掉 border 属性，则可去除表格的边框，运行效果如图 3-2 所示。

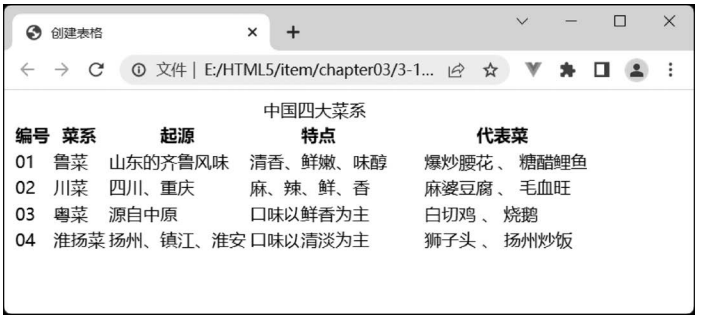


图 3-2　表格去除边框的运行效果

3.1.2　语义化标签

一个完整的表格包括< table >、< caption >、< tr >、< th >、< td >等标签。在使用表格进行布局时，为了更深入地对表格进行语义化，使网页内容更好地被搜索引擎理解，HTML 中引入了< thead >、< tbody >和< tfoot >这 3 个语义化标签。< thead >、< tbody >和< tfoot >标签将表格分别划分为头部、主体和页脚 3 部分。使用这 3 个语义化标签来定义网页中不同的内容，能够使结构更清晰，代码更具逻辑性，也更具可读性和可维护性。

表格的语义化标签及其说明如表 3-1 所示。

表 3-1　语义化标签及其说明

标　签	说　明
< thead >	用于定义表格的头部，一般包含网页的 logo 和导航等头部信息
< tbody >	用于定义表格的主体，位于< thead ></thead >标签之后，一般包含网页中除头部和底部以外的其他内容
< tfoot >	用于定义表格的页脚，位于< tbody ></tbody >标签之后，一般包含网页底部的企业信息等

演示说明

使用表格的语义化标签制作一个“语义化标签”的说明表格，具体代码如例 3-2 所示。

【例 3-2】 语义化标签。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>语义化标签</title>
</head>
<body>
    <table border="1">
        <!-- 定义表格的标题 -->
        <caption>语义化标签</caption>
        <!-- 定义表格的表头 -->
        <thead>
            <!-- 定义表格内的行 -->
            <tr>
                <!-- 定义表格内的表头单元格 -->
                <th>标签</th>
                <th>说明</th>
            </tr>
        </thead>
        <!-- 定义表格的主体 -->
        <tbody>
            <tr>
                <!-- 定义表格内的标准单元格 -->
                <td>thead</td>
                <td>用于定义表格的头部,一般包含网页的 logo 和导航等头部信息</td>
            </tr>
            <tr>
                <td>tbody</td>
                <td>用于定义表格的主体,一般包含网页中除头部和底部以外的其他内容
</td>
            </tr>
            <tr>
                <td>tfoot</td>
                <td>用于定义表格的页脚,一般包含网页底部的企业信息等</td>
            </tr>
        </tbody>
        <!-- 定义表格的页脚 -->
        <tfoot>
            <tr>
                <!-- 在表格的页脚内合并 2 个单元格 -->
                <td colspan="2">语义化标签让表格语义更加良好,结构更加清晰</td>
            </tr>
        </tfoot>
    </table>
</body>
</html>
```

语义化标签的运行效果如图 3-3 所示。

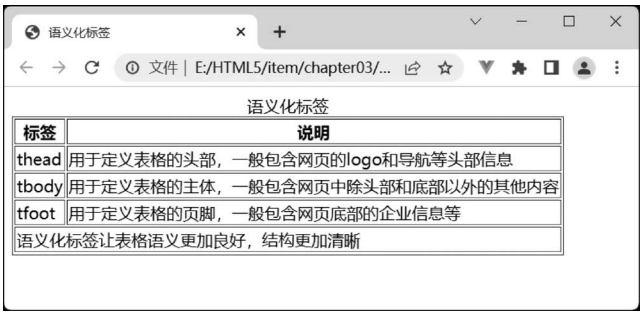


图 3-3　语义化标签的运行效果

3.1.3　单元格边距与间距

在制作一个表格时，有时需要设计表格的单元格内容与单元格边框之间的空白间距，以及单元格之间的空间，使表格更美观，这就需要使用到 cellpadding 属性和 cellspacing 属性。

1. cellpadding 属性

cellpadding 属性规定单元格内容与单元格边框之间的空白间距，即控制单元格的边距。cellpadding 属性通常使用在< table >标签中，其属性值为数值类型，常用单位是像素(px)，这个数值代表单元格内容与单元格边框之间的空白间距，即内边距，默认值为 1px。

cellpadding 属性的语法格式如下。

```
< table cellpadding = "pixels">
```

如果将 cellpadding 属性添加到例 3-1 的第 8 行代码的< table >标签中，更改的代码如下。

```
<!-- 定义表格，并添加边框、单元格的边距 -->
< table border = "1" cellpadding = "8">
```

为表格添加 cellpadding 属性，运行效果如图 3-4 所示。

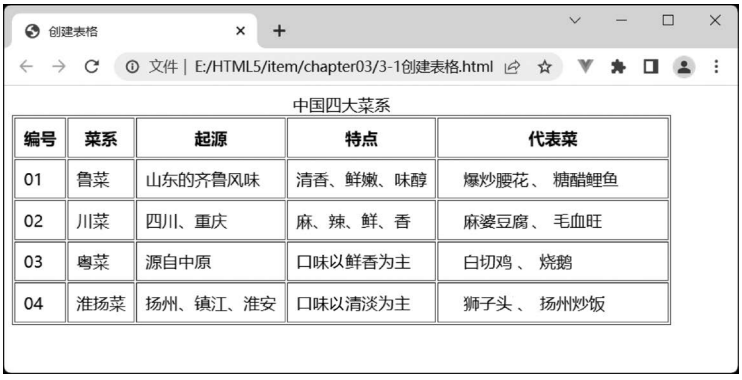


图 3-4　表格添加 cellpadding 属性的运行效果

2. cellspacing 属性

cellspacing 属性规定单元格之间的空间，即控制单元格的间距。cellspacing 属性通常使用在< table >标签中，其属性值为数值类型，常用单位是像素(px)，这个数值代表单元格之间的空白间距，即外边距，默认值为 2px。

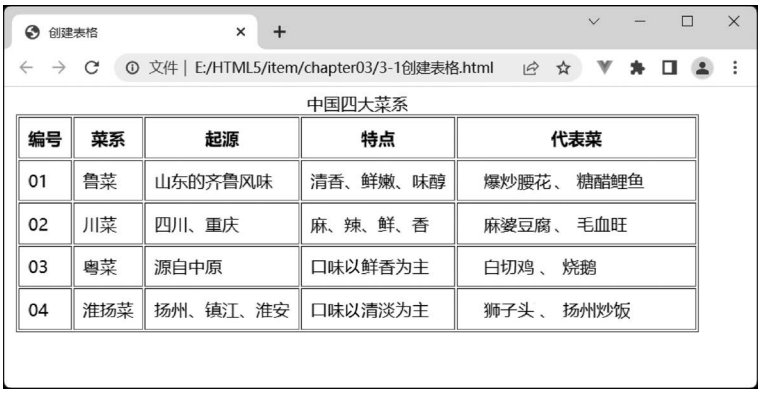
cellspacing 属性的语法格式如下。

```
< table cellspacing = "pixels">
```

如果将 cellspacing 属性添加到例 3-1 的第 8 行代码的< table >标签中,更改的代码如下。


```
<!-- 定义表格,并添加边框、单元格的边距、单元格的间距 -->
< table border = "1" cellpadding = "8"  cellspacing = "5">
```

再为表格添加 cellspacing 属性,运行效果如图 3-5 所示。



中国四大菜系

编号	菜系	起源	特点	代表菜
01	鲁菜	山东的齐鲁风味	清香、鲜嫩、味醇	爆炒腰花、 糖醋鲤鱼
02	川菜	四川、重庆	麻、辣、鲜、香	麻婆豆腐、 毛血旺
03	粤菜	源自中原	口味以鲜香为主	白切鸡 、 烧鹅
04	淮扬菜	扬州、镇江、淮安	口味以清淡为主	狮子头 、 扬州炒饭

图 3-5　表格添加 cellspacing 属性的运行效果

在< table >标签中,设置 cellspacing 属性的值为 0,表示单元格之间的空白间距为 0,即表格变为单线框,但这种方式并不推荐使用。在实际开发中,通常使用 CSS 中的 border-collapse 属性来决定表格的边框是分开还是合并。border-collapse 属性将会在 10.5.2 节中进行详细讲解和应用。

值得注意的是,请勿将 cellpadding 属性与 cellspacing 属性相混淆,需要分清它们的用途。从实用角度出发,最好不要规定 cellpadding 属性,而是使用 CSS 来添加内边距。

3.1.4　合并行与列

在制作一个表格时,有时需要对表格的单元格进行合并行或合并列的操作,把两个或多个相邻单元格合并成一个单元格,这就需要使用 rowspan 属性和 colspan 属性。

rowspan 属性和 colspan 属性合并表格行与列及其说明如表 3-2 所示。

表 3-2　合并行与列及其说明

属　性	语法格式	说　明
rowspan	< td rowspan="数值">	规定单元格可横跨的行数,即合并表格的行。rowspan 属性通常应用在< td >和< th >标签中,其属性值为数值类型,这个数值代表所要合并的单元格行数
colspan	< td colspan="数值">	规定单元格可横跨的列数,即合并表格的列。colspan 属性通常应用在< td >和< th >标签中,其属性值为数值类型,这个数值代表所要合并的单元格列数

演示说明

使用表格语义化标签以及 rowspan 属性和 colspan 属性,制作一个有关中国四大菜系分支的表格,具体代码如例 3-3 所示。

【例 3-3】 菜系分支。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>菜系分支</title>
</head>
<body>
    <!-- 定义表格,并添加边框、单元格的边距 -->
    <table border="1" cellpadding="6">
        <!-- 定义表格的标题 -->
        <caption>中国四大菜系分支</caption>
        <!-- 定义表格的表头 -->
        <thead>
            <!-- 定义表格内的行 -->
            <tr>
                <!-- 定义表格内的表头单元格 -->
                <th>编号</th>
                <th>菜系</th>
                <!-- 合并表格的列,可横跨 2 列 -->
                <th colspan="2">起源/分支</th>
                <th>特点</th>
                <th>代表菜</th>
            </tr>
        </thead>
        <!-- 定义表格的主体 -->
        <tbody>
            <tr>
                <!-- 定义表格内的标准单元格 -->
                <!-- 合并表格的行,可横跨 2 行 -->
                <td rowspan="2">01</td>
                <td rowspan="2">鲁菜</td>
                <td rowspan="2">山东的齐鲁风味</td>
                <td>济南风味菜</td>
                <td rowspan="2">清香、鲜嫩、味醇</td>
                <td rowspan="2">爆炒腰花、糖醋鲤鱼</td>
            </tr>
            <tr>
                <td>孔府菜</td>
            </tr>
            <tr>
                <!-- 合并表格的行,可横跨 3 行 -->
                <td rowspan="3">02</td>
                <td rowspan="3">川菜</td>
                <td rowspan="3">四川、重庆</td>
                <td>成都菜</td>
                <td rowspan="3">麻、辣、鲜、香</td>
                <td rowspan="3">麻婆豆腐、毛血旺</td>
            </tr>
            <tr>
                <td>自贡菜</td>
            </tr>
            <tr>
                <td>乐山菜</td>
            </tr>
            <!-- 此处省略雷同代码 -->
```

```
56          </tbody>
57          <!-- 定义表格的页脚 -->
58          <tfoot>
59              <tr>
60                  <!-- 合并表格的列,可横跨 6 列 -->
61                  <td colspan="6">四大菜系具有鲜明的地方风味特色,并为社会所公认的中国饮食的菜肴流派</td>
62              </tr>
63          </tfoot>
64      </table>
65  </body>
66  </html>
```

使用表格语义化标签以及 rowspan 属性和 colspan 属性制作表格,菜系分支的运行效果如图 3-6 所示。

图 3-6　菜系分支的运行效果

3.1.5　表格的其他属性

HTML5 为表格提供了一系列用于控制表格样式的属性,例如 border 属性、bordercolor 属性、align 属性、width 属性、bgcolor 属性、background 属性等。这些属性的具体说明如表 3-3 所示。

表 3-3　表格其他属性及其说明

属　性	说　明
border	表示是否设置边框,可以取值为 1 和 0,1 代表有边框,0 代表无边框(通常省略不写)
bordercolor	用于设置边框颜色,在< table >标签中,需配合 border 属性使用,可对表格的整体边框进行颜色的设置
align	设置单元格内容的水平对齐方式,在< tr >和< td >标签中,align 属性的默认值为左对齐(left);在< th >标签中,align 属性的默认值为居中对齐(center);而在< table >标签中,align 属性用于设置表格在网页中的水平对齐方式
valign	设置单元格内容的垂直对齐方式,默认值为居中对齐(center)

续表

属　性	说　明
width	设置单元格的宽度，当一列单元格中有不同 width 属性值时，取最大值作为这一列的宽度
height	设置单元格的高度，当一行单元格中有不同 height 属性值时，取最大值作为这一行的高度
bgcolor	规定表格的背景颜色。在 HTML 4.01 中，表格的 bgcolor 属性已废弃，HTML5 已不支持表格的 bgcolor 属性，但该属性在浏览器中仍能识别出来。当需要设置表格背景颜色时，一般在 CSS 样式中设置
background	设置表格的背景图片，属性值为一个有效的图片地址，不推荐使用。在实际开发中，通常使用 CSS 属性设置表格的背景图片

border 属性不会控制边框的样式，若需要设置边框样式，通常使用 CSS 样式设计表格边框，即通过 border 属性的连写设置边框，详细用法会在 6.1.3 节进行说明，CSS 样式设计边框的示例代码如下。

```
table{ border:1px solid #aaa; }
```

3.2　实例二：天气情况表格统计

天气现象风云变幻，奥秘无穷，时刻影响着人们的生活。华夏先民在与大自然的长期互动中，日渐适应了四时交替，阴阳变化，逐渐探索出气象变化的一般规律，掌握观天察气、看云识天的气象知识，使之成为中华民族乃至世界文明的宝贵精神财富和独特文化资源。

3.2.1　“天气情况表格统计”页面结构简图

本实例是使用 HTML 表格标签制作一份北京一周的天气情况统计表。该页面主要由表格的< table >标签、< caption >标签、< tr >标签、< td >标签以及两个语义化标签构成。“天气情况表格统计”的页面结构简图如图 3-7 所示。

图 3-7　“天气情况表格统计”的页面结构简图

3.2.2 实现“天气情况表格统计”页面效果

新建一个 HTML5 文件,使用< table >、< caption >、< tr >、< td >等标签编写相关代码,并使用表格相关属性设置表格样式,具体代码如例 3-4 所示。

【例 3-4】 天气情况表格统计。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>天气情况表格统计</title>
</head>
<body>
    <!-- 定义表格,并添加边框、单元格的边距、单元格的间距 -->
    <table border="1" cellpadding="10" cellspacing="6">
        <!-- 定义表格的标题 -->
        <caption>天气情况统计</caption>
        <!-- 定义表格的表头 -->
        <thead>
            <!-- 定义表格内的行 -->
            <tr>
                <!-- 定义表格内的表头单元格 -->
                <th>日期</th>
                <th>天气</th>
                <th>最低温度</th>
                <th>最高温度</th>
                <th>风向</th>
                <th>风力</th>
            </tr>
        </thead>
        <!-- 定义表格的主体 -->
        <tbody>
            <!-- 设置单元格内容为水平居中对齐方式 -->
            <tr align="center">
                <!-- 定义表格内的标准单元格 -->
                <td>星期一</td>
                <td>晴</td>
                <td>-7℃</td>
                <td>3℃</td>
                <td>西北风</td>
                <td>3～4 级</td>
            </tr>
            <tr align="center">
                <td>星期二</td>
                <td>晴</td>
                <td>-8℃</td>
                <td>1℃</td>
                <td>北风</td>
                <td>微风</td>
            </tr>
            <!-- 此处省略雷同代码 -->
        </tbody>
```

```
47      </table>
48 </body>
49 </html>
```

在上述代码中，首先，使用< table >标签创建表格，< caption >标签定义标题，并在< table >标签中添加 border、cellpadding、cellspacing 属性分别为表格设置边框、单元格的边距、单元格的间距；其次，使用两个语义化标签将表格划分为头部和主体两部分；最后，使用< tr >标签定义表格的行，并根据情况在< tr >标签内插入< th >表头单元格或< td >标准单元格，再在< tr >标签中添加 align＝"center"设置单元格内容为水平居中对齐。

3.3 HTML5 表单

表单是网页中常用的一种展示效果，如登录页面中的用户名和密码的输入、登录按钮等都是使用表单相关标签进行定义的。在 HTML5 中，表单是获取用户输入的手段，它的主要功能是收集用户的信息，并将这些信息传递给后台服务器，实现用户与 Web 服务器的交互。

3.3.1 表单的组成

在 HTML5 中，一个完整的表单通常由表单元素、提示信息和表单域 3 部分组成，下面将详细介绍这 3 部分。

表单元素：包含表单的具体功能项，如文本输入框、下拉列表框、复选框、密码输入框、登录按钮等。

提示信息：表单中通常还需包含一些说明性的文字，提示用户要进行的操作。

表单域：用来容纳表单元素和提示信息，可以通过它定义处理表单数据所用程序的 URL 地址，以及数据提交到服务器的方法。如果未定义表单域，表单中的数据就无法传送到后台服务器。

表单元素是表单的核心，常用的表单元素如表 3-4 所示。

表 3-4 常用的表单元素

表 单 元 素	含　义
< input >	文本输入框，可定义多种控件类型，如 text（单行文本框）、password（密码文本框）、radio（单选框）、checkbox（复选框）、button（按钮）、submit（提交按钮）、reset（重置按钮）、hidden（隐藏域）、image（图像域）、file（文件域）等
< select >	定义一个下拉列表（必须包含列表项）
< textarea >	定义多行文本框
< label >	定义表单辅助项

这里先简单了解一下常用的表单元素，本章后续小节将会对其进行详细讲解。

3.3.2 < form >标签

为了实现用户与 Web 服务器的交互，需要让表单中的数据传送给服务器，这就必须定义表单域。定义表单域与< table >标签定义表格类似，在 HTML5 中< form >标签用于定义

表单域,即创建一个表单,用来实现用户信息的收集和传递,< form ></form >标签中的所有内容都会提交给服务器。

1. 语法格式

< form >标签的语法格式如下。

```
< form action = "URL 地址" method = "数据提交方式">
    表单元素和提示信息
</form >
```

2. 标签属性

< form >标签常用的属性包括 action 属性和 method 属性,以及作为了解的 enctype 属性和 target 属性。接下来将具体介绍这几种属性。

1) action 属性

action 属性可定义表单数据的提交地址,即一个 URL 地址。HTML 表单要连接服务器,就需要在 action 属性上设置一个 URL。例如,两个人要打电话就必须要知道对方的电话号码,URL 就相当于电话号码。action 属性用于指定接收并处理表单数据的服务器的 URL 地址,示例如下。

```
< form action = "qianfeng_action.asp"
```

该示例表示当提交表单时,表单数据会传送到 qianfeng_action.asp 的页面进行处理。

action 属性值可以是相对路径或绝对路径,还可以是接收数据的 E-mail 地址,示例如下。

```
< form action = qianfeng@1000phone.com >
```

该示例表示当提交表单时,表单数据会以电子邮件的形式传递出去。

2) method 属性

method 属性可以定义表单数据的提交方式,常用的有 get(默认)和 post 这两种方式。提交方式类似于通信方式,可以打电话、发短信或发邮件。以 get 方式提交数据,数据会直接显示在浏览器的地址栏中,保密性差,且存在数据量限制。以 post 方式提交数据,保密性好,且无数据量限制,使用 method="post"可以批量提交数据。

3) enctype 属性

enctype 属性规定在发送到服务器之前应该如何对表单数据进行编码。enctype 属性的可取值为 application/x-www-form-urlencoded、multipart/form-data 和 text/plain,enctype 属性值及其说明如表 3-5 所示。

表 3-5 enctype 属性值及其说明

属 性 值	说 明
application/x-www-form-urlencoded	在发送到服务器之前,所有字符都会进行编码(空格转换为“+”加号,特殊符号转换为 ASCII HEX 值)
multipart/form-data	不对字符进行编码。当使用包含文件上传控件的表单时,必须使用该值
text/plain	空格转换为“+”号,但不对特殊字符编码

4) target 属性

target 属性定义提交地址的打开方式,常用的打开方式有_self(默认)和_blank。_self

可在当前页打开，_blank 可在新页面打开，< form >标签中的 target 属性与< a >标签中的 target 属性用法一样，此处不再赘述。

3. 演示说明

使用< form >标签及其属性创建一个用户账号信息提交表单，具体代码如例 3-5 所示。

【例 3-5】 创建表单。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>创建表单</title>
</head>
<body>
    <!-- 添加表单域，设置表单数据提交地址、数据提交方式、数据提交地址打开方式、数据提交内容方式 -->
    <form action="demo.html" method="post" target="_blank" enctype="multipart/form-data">
        用户名：<input type="text" name="user">
        密码：<input type="password" name="pass">
        <input type="submit" value="提交">
    </form>
</body>
</html>
```

使用< form >标签及其属性创建一个用户账号信息提交表单，运行效果如图 3-8 所示。

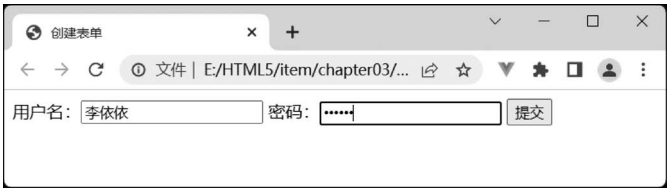


图 3-8 创建表单的运行效果

当单击“提交”按钮时，表单数据会提交到 demo.html 页面进行处理，运行效果如图 3-9 所示。

图 3-9 数据提交的运行效果

3.3.3 < input >标签

< input >标签用于搜集用户信息，是一个单标签。网页中经常会使用到单行文本框、密码文本框、单选框、提交按钮等，要想定义这些表单元素就需要使用< input >标签。

< input >标签的语法格式如下。

```
<input type="控件类型">
```


1. type 属性

<input>标签根据 type 属性的取值不同,可以展现出不同的表单控件类型,如 text 单行文本框、password 密码文本框、submit 重置按钮等。当在网页中收集用户信息时,部分信息通常会有严格的限制,不能任由用户自行输入而只能进行选择,这就需要使用到 radio 单选框或 checkbox 复选框。<input>可实现的表单控件说明如表 3-6 所示。

表 3-6 <input>可实现的表单控件说明

属 性 值	说 明
text	单行文本框。可以输入任何类型的文本,如文字、数字等,输入的内容以单行显示,如<input type="text" name="" value="">
password	密码文本框。定义密码字段,该字段中的字符被掩码,显示为“*”,如<input type="password" name="" value="">
button	普通按钮。定义可单击的按钮,如<input type="button" name="" value="">
submit	提交按钮。提交按钮会把表单数据发送到服务器,如<input type="submit" name="" value="">
image	定义图片形式的提交按钮。需要结合 src 属性和 alt 属性使用,src 属性定义图片的来源,alt 属性定义当图片无法显示时的提示文字,如<input type="image" src="图片地址" alt="提示文字">
reset	重置按钮。重置按钮会清除表单中所有的数据,如<input type="reset" name="" value="">
radio	单选框。多个 name 属性值相同的单选框控件可成为一组,让用户进行选择,一组单选框中只能选择其中 1 个选项,不可多选,如<input type="radio" name="" value="">
checkbox	复选框。多个 name 属性值相同的复选框控件可成为一组,让用户进行选择,一组复选框中允许选择多个选项。值得注意的是,在一组单选框或复选框中,name 属性值必须相同,如<input type="checkbox" name="" value="">
hidden	隐藏域。可用于隐藏向后台服务器发送的一些数据,如正在被请求或编辑的内容的 ID 名。隐藏域是一种不影响页面布局的表单控件。值得注意的是,尽量不要将重要信息上传至隐藏域,避免信息泄露,如<input type="hidden" name="">
file	文件域。可用于上传文件,用户可以选择 1 个或多个元素以提交表单的方式上传到服务器上,如文档文件上传和图片文件上传,如<input type="file" name="">

值得注意的是,使用文件域时,<form>标签的 method 属性值必须设置成 post,enctype 属性值必须设置成 multipart/form-data。

文件域不仅支持<input>元素共享的公共属性,还支持自身的一些特定属性,如 accept、capture、multiple 和 files。文件域特定属性的说明如表 3-7 所示。

表 3-7 文件域特定属性的说明

属 性	说 明
accept	文件域允许接受的文件类型,多种文件类型以逗号(,)为分隔
capture	捕获图像或视频数据的源
multiple	允许用户选择多个文件
files	列出已选择的文件

2. ＜input＞标签的其他常用属性

＜input＞标签除了 type 属性之外，还有一些常用属性，如 name 属性、placeholder 属性、readonly 属性、disabled 属性、checked 属性等。＜input＞标签的其他常用属性的说明如表 3-8 所示。

表 3-8 ＜input＞标签的其他常用属性的说明

属　性	说　明
name	规定＜input＞元素的名称，提交给服务器。name 属性值通常与 value 属性值配合成一对使用，后台服务器可通过 name 值找到对应的 value 值
value	规定＜input＞元素的值，提交给服务器
placeholder	输入框提示文本
readonly	定义元素内容为只读（不能修改编辑）
disabled	禁用。定义该元素不可用（显示为灰色），提交表单时不会被提交给服务器
checked	默认选择项。定义选择元素被默认选中的项，适用于单选框和多选框
required	必填项。具有该属性的＜input＞标签，在提交时若没有填写内容，则会提示此元素为必填项
size	宽度。设置输入框的宽度
maxlength	最大长度。设置输入框的最大长度

3.3.4 ＜label＞标签

1. 概述

＜label＞标签可用于定义＜input＞元素的标记，主要作用是辅助表单元素，可更好地提升用户体验。当用户单击＜label＞标签内的文本时，会自动将焦点转到与标签相关的表单控件上。＜label＞标签中的 for 属性用于指出当前文本标签与哪个表单控件相关联，其属性值一定要与＜input＞标签中的 id 属性值保持一致才能找到相应控件。

2. 演示说明

使用＜input＞标签及其相关控件，以及＜label＞标签创建一个基本的表单，在表单域中添加一个单行文本框、密码框、单选框、多选框和提交按钮控件，具体代码如例 3-6 所示。

【例 3-6】 表单控件。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>表单控件</title>
</head>
<body>
    <!-- 添加表单域，并在<form>标签中添加相关属性 -->
    <form action="" method="get"  target="_self">
        <!-- 为表单添加标记，for 属性关联单行文本框 -->
        <label for="use">姓名：</label>
        <!-- 添加单行文本框控件，设置相关属性，如输入框提示文本、必选项等 -->
        <input type="text" name="user" value="" id="use" placeholder="输入用户名" required>
        <br>
        <label for="word">密码：</label>
        <!-- 添加密码框控件，设置相关属性，如输入框提示文本、必选项、输入框长度等 -->
```

```
        <input type="password" name="pass" value="" id="word" placeholder="输入密码" size="20" required>
        <br>
        <!-- 添加单选框控件,name 属性值必须一致。为了避免发生漏选问题,可添加 checked 属性,此值为默认选中 -->
        性别:<input type="radio" name="gender" value="" id="man" checked>
        <label for="man">男</label>
        <input type="radio" name="gender" value="" id="woman" >
        <label for="woman">女</label>
        <br>
        <!-- 添加多选框控件 -->
        爱好:<input type="checkbox" name="hobby" value="下棋" id="chess">
        <label for="chess">下棋</label>
        <input type="checkbox" name="hobby" value="民族舞" id="dance" >
        <label for="dance">民族舞</label>
        <input type="checkbox" name="hobby" value="看书" id="read" >
        <label for="read">看书</label>
        <input type="checkbox" name="hobby" value="跑步" id="run" >
        <label for="run">跑步</label>
        <br>
        <!-- 添加提交按钮控件,将数据提交给服务器 -->
        <input type="submit" name="but" value="提交" >
    </form>
</body>
</html>
```

表单控件的运行效果如图 3-10 所示。

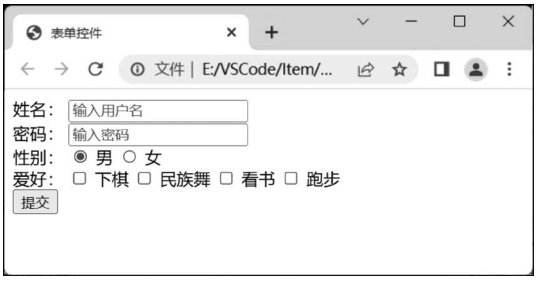


图 3-10　表单控件的运行效果

在上述代码中,单行文本框和密码框设置了 required 属性,为必填项。当密码框未填写内容时,单击“提交”按钮,会出现提示必须填写密码信息的内容,此时页面上填写的数据不会发送至服务器。

3.3.5　<select>标签

<select>标签可定义表单中的下拉列表。网页中经常会看到多个选项的下拉菜单,如选择城市、日期、科目等。在<select>标签中包含一个或多个<option>标签,<option>标签可创建选择项。<select>标签需要与<option>标签组合使用,这个特点与列表类似,如无序列表由<ul>标签和<li>标签配合使用。为了更好地理解,可将下拉列表看作一个特殊的无序列表。

1. 语法格式

<select>标签的语法格式如下。

```
<select name = "下拉列表的名称" >
    <option value = "选择项 1">选择项 1 </option>
        …
    <option value = "选择项 n">选择项 n </option>
</select>
```

值得注意的是，需要在< select >标签中设置 name 属性，每个< option >标签中也需要设置 value 属性，这样可便于服务器获取选择框，以及用户获取选择项的值。如果在< option >标签中省略 value 值，则包含的文本就是选择项的值。

2. ＜select＞标签属性

< select >标签可通过添加属性来改变下拉列表的外观显示效果。< select >标签的常用属性有 multiple 属性和 size 属性，这两种属性的说明如表 3-9 所示。

表 3-9 ＜select＞标签常用属性的说明

属　性	说　明
multiple	设置多选下拉列表。默认下拉列表只能选择一项，而设置 multiple 属性后下拉列表可选择多项(按住 Ctrl 键即可选择多项)。多选下拉列表在选择项的数目超过列表框的高度时，会显示滚动条，通过拖动滚动条可查看并选择多个选项
size	设置下拉列表可见选项的数目，取值为正整数

3. ＜option＞标签属性

< option >标签的常用属性有 value 属性、selected 属性和 disabled 属性，可用于设置下拉列表中的各个选择项。< option >标签常用属性的说明如表 3-10 所示。

表 3-10 ＜option＞标签常用属性的说明

属　性	说　明
value	定义送往服务器的选项值
selected	默认此选项(首次显示在列表中时)表现为选中状态
disabled	规定此选项应在首次加载时便被禁用

在< select >标签和< option >标签之间，可以使用< optgroup >标签对选择项进行分组，即把相关选择项组合在一起。< optgroup >标签的 label 属性可用于设置分组项的标题。

4. 演示说明

制作一个下拉列表，在表单中定义单选下拉列表和多选下拉列表，在单选下拉列表中使用 selected 属性设置默认选中值，在多选下拉列表中使用< optgroup >标签对选择项进行分组，具体代码如例 3-7 所示。

【例 3-7】 下拉列表。

```
1  <!DOCTYPE html>
2  <html lang = "en">
3  <head>
4      <meta charset = "UTF - 8">
5      <title>下拉列表</title>
6  </head>
7  <body>
8      <form>
9          <p>您目前所在的年级是
```

```
            <label for="class">
                <!-- 定义单选下拉列表 -->
                <select name="grade" id="class">
                    <!-- selected 属性将"大一"设置为默认选中值 -->
                    <option value="one" selected>大一</option>
                    <option value="two">大二</option>
                    <option value="third">大三</option>
                    <option value="four">大四</option>
                </select>
            </label>
        </p>
        <p>您目前所学科目有
            <label for="subject">
                <!-- 定义多选下拉列表,设置 multiple 和 size 属性 -->
                <select name="course" id="subject" multiple size="6">
                    <!-- 利用<optgroup>标签对选择项进行分组操作 -->
                    <optgroup label="前端">
                        <option value="html5">HTML5</option>
                        <option value="css">CSS</option>
                    </optgroup>
                    <optgroup label="后端">
                        <option value="java">Java</option>
                        <option value="python">Python</option>
                    </optgroup>
                </select>
            </label>
        </p>
    </form>
</body>
</html>
```

下拉列表的运行效果如图 3-11 所示。

图 3-11　下拉列表的运行效果

图 3-11 中,在多选下拉列表中,需要按住 Ctrl 键并单击对应选择项才可以选择多个选项。

3.3.6　<textarea>标签

<textarea>标签可定义多行文本框(文本域),用户可以在多行文本框内输入多行文本。文本区域内可容纳无限数量的文本,文本的默认字体是等宽字体(通常是 Courier)。使用 cols 和 rows 属性能够规定多行文本框的尺寸,不过更好的办法是使用 CSS 的 height 和 width 属性。

1. 语法格式

多行文本框的语法格式如下。

```
<textarea name="文本框名称" rows="文本框行数" cols="文本框列数"></textarea>
```

2. 标签属性

<textarea>标签的常用属性有 name 属性、rows 属性、cols 属性和 autofocus 属性，这些属性的具体说明如表 3-11 所示。

表 3-11　<textarea>标签常用属性的说明

属　　性	说　　明
name	定义多行文本框的名称，由于在存储文本时必须使用此项，因此 name 属性不可省略
rows	定义多行文本框的水平列，表示可显示的行数
cols	定义多行文本框的垂直列，表示可显示的列数，即一行中可容纳下的字节数
autofocus	规定在页面加载后文本区域自动获得焦点

3. 演示说明

使用<textarea>标签制作一个多行文本框，具体代码如例 3-8 所示。

【例 3-8】 多行文本框。

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>多行文本框</title>
6  </head>
7  <body>
8      <h3>留言板</h3>
9      <form action="" method="post">
10         <!-- 定义多行文本框，设置列数为 40，行数为 10 -->
11         <textarea name="content" id="content" cols="40" rows="10">请在此输入内容
   </textarea>
12     </form>
13 </body>
14 </html>
```

多行文本框的运行效果如图 3-12 所示。

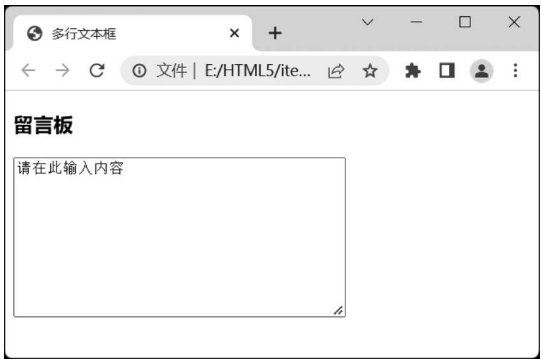


图 3-12　多行文本框的运行效果

在实际开发中,通常使用 CSS 的 height 和 width 属性来规定多行文本框的尺寸。

3.3.7 <fieldset>标签

<fieldset>标签可将表单内的相关元素进行分组,并绘制边框。<legend>标签包含于<fieldset>标签内,用于定义分组表单的标题。<fieldset>标签可以使表单域的层次更加清晰,更易于用户理解。

使用<fieldset>标签制作一个账号注册和邮箱注册的分组表单,具体代码如例 3-9 所示。

【例 3-9】 分组表单。

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>分组表单</title>
</head>
<body>
    <form action="#" method="post">
        <!-- 分组 -->
        <fieldset>
            <!-- 分组表单的标题 -->
            <legend>账号注册</legend>
            <label for="user">账户名</label>
            <input type="text" name="ming" id="user"><br>
            <label for="word">密码</label>
            <input type="password" name="pass" id="word">
        </fieldset>
        <fieldset>
            <legend>手机号注册</legend>
            <label for="phone">手机号</label>
            <input type="tel" name="tel" id="phone"><br>
            <label for="code">验证码</label>
            <input type="number" name="num" id="code">
        </fieldset>
    </form>
</body>
</html>
```

分组表单的运行效果如图 3-13 所示。

图 3-13 分组表单的运行效果

3.4 实例三：图书库存信息录入表单

“书中自有颜如玉，书中自有黄金屋”，读书能够使人明智、明理、明德。书籍以文字的形式记录人类的历史文明，使其得到延续与传承，使人类走向文明，学习到知识，是人类最宝贵的财富之一！

3.4.1 “图书库存信息录入表单”页面结构简图

本实例是制作一个“图书库存信息录入表单”的页面，该页面主要由表单中的< form >标签、< fieldset >标签、< label >标签、< select >标签和< textarea >标签，以及< input >标签中的单行文本框、单选框、数字输入框、日期输入框、文件域、提交按钮和重置按钮构成。“图书库存信息录入表单”的页面结构简图如图 3-14 所示。

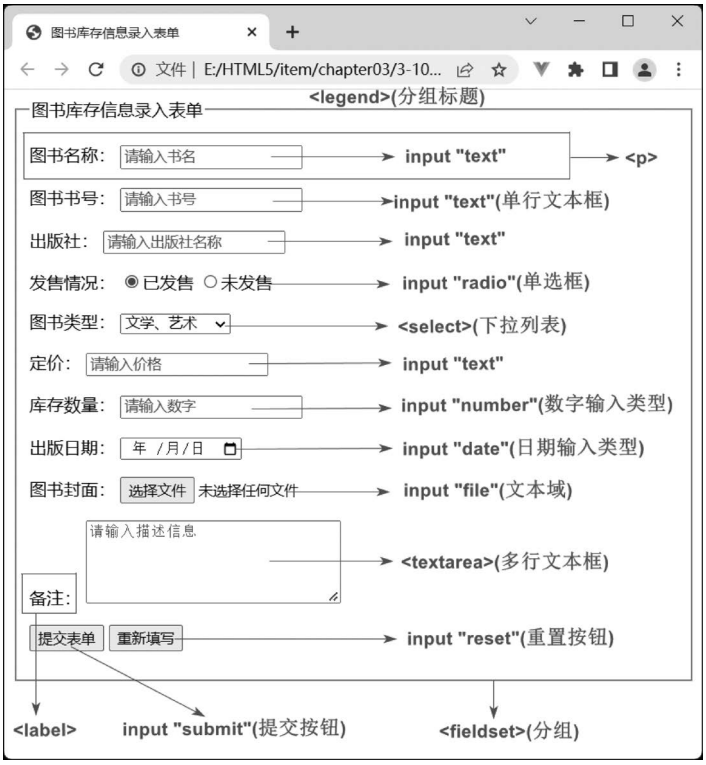


图 3-14 “图书库存信息录入表单”的页面结构简图

3.4.2 实现“图书库存信息录入表单”页面效果

新建一个 HTML5 文件，使用< form >标签创建一个表单，通过< fieldset >标签对表单进行分组，再使用表单常用标签创建各个表单控件，具体代码如例 3-10 所示。

【例 3-10】 图书库存信息录入表单。

```
1  <!DOCTYPE html>
2  <html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <title>图书库存信息录入表单</title>
</head>
<body>
    <!-- 表单域 -->
    <form action="#" method="post" enctype="multipart/form-data">
        <!-- 分组 -->
        <fieldset>
            <!-- 分组的表单标题 -->
            <legend>图书库存信息录入表单</legend>
            <p>
                <!-- 信息提示,可关联控件 -->
                <label for="ming">图书名称:</label>
                <!-- 单行文本框 -->
                <input type="text" id="ming" name="ming" placeholder="请输入书名" required>
            </p>
            <p>
                <label for="isbn">图书书号:</label>
                <!-- 单行文本框 -->
                <input type="text" id="isbn" name="isbn" placeholder="请输入书号" required>
            </p>
            <p>
                <label for="publisher">出版社:</label>
                <!-- 单行文本框 -->
                <input type="text" id="publisher" name="publisher" placeholder="请输入出版社名称" required>
            </p>
            <p>
                <label for="">发售情况:</label>
                <!-- 单选框 -->
                <input type="radio" name="sale" value="yes" checked>已发售
                <input type="radio" name="sale" value="no">未发售
            </p>
            <p>
                <label for="kind">图书类型:</label>
                <!-- 下拉列表 -->
                <select name="kind" id="kind">
                    <option value="philosophy">哲学</option>
                    <option value="Social">人文社会科学</option>
                    <option value="literature" selected>文学、艺术</option>
                    <option value="nature">自然应用科学</option>
                    <option value="history">历史、地理</option>
                </select>
            </p>
            <p>
                <label for="pricing">定价:</label>
                <input type="text" id="pricing" name="price" placeholder="请输入价格" required>
            </p>
            <p>
                <label for="amount">库存数量:</label>
```

```
53                  <!-- 数字输入类型 -->
54                  <input type="number" id="amount" name="amount" placeholder="请输
    入数字" required>
55              </p>
56              <p>
57                  <label for="time">出版日期: </label>
58                  <!-- 日期输入类型 -->
59                  <input type="date" id="time" name="time">
60              </p>
61              <p>
62                  <label for="upload">图书封面: </label>
63                  <!-- 文件域 -->
64                  <input type="file" name="file" id="upload">
65              </p>
66              <p>
67                  <label for="tip">备注: </label>
68                  <!-- 多行文本框 -->
69                  <textarea name="tip" id="tip" cols="30" rows="5" placeholder="请
    输入描述信息"></textarea>
70              </p>
71              <p>
72                  <!-- 提交按钮 -->
73                  <input type="submit" value="提交表单">
74                  <!-- 重置按钮 -->
75                  <input type="reset" value="重新填写">
76              </p>
77          </fieldset>
78      </form>
79  </body>
80  </html>
```

在上述代码中，表单中主要有 9 种控件，< select >标签和< textarea >标签分别定义下拉列表和多行文本框，而其余 7 种控件由< input >标签中的 type 属性来定义。< label >标签用来编辑信息提示文本，使其与指定控件进行关联。因此，需要为每个控件中分别定义不同的 id 属性、name 属性、value 属性等，以及一些控制控件状态的属性，如 required 属性。

3.5 本章小结

本章重点学习如何制作表格和表单，主要介绍表格和表单的相关标签与属性。表格不仅可以制作常规表格，还可用于对网页进行布局。表单能够在网页上收集用户的各类信息，增强网页与用户之间的信息交互。希望通过对本章内容的分析和讲解，读者能够熟悉表格与表单的相关标签与属性，掌握表格和表单的制作，可以编写出适应于网页页面需求的表格和表单，为后面的深入学习奠定基础。

3.6 习　　题

1. 填空题

(1) 一个最基本的表格由_____、_____、_____ 3 个标签构成。

(2) 一个完整的表单通常由_____、_____和_____3 部分构成。

(3) < th >是_____,< td >是_____。

(4) < select >标签需要与_____标签配合使用。

2. 选择题

(1) < input >标签中用于定义单选框的 type 属性值是(　　)。

A. checkbox　　B. reset　　C. radio　　D. file

(2) 下列不属于表格属性的是(　　)。

A. rowspan　　B. border　　C. cellpadding　　D. text-align

(3) 能使下拉列表选择多项的属性是(　　)。

A. selected　　B. disabled　　C. multiple　　D. size

(4) 以下标签中不能体现表格语义化的是(　　)。

A. < table ></ table >　　B. < thead ></ thead >

C. < tfoot ></ tfoot >　　D. < tbody ></ tbody >

3. 思考题

(1) 简述 3 个语义化标签的含义。

(2) 简述< label >标签的作用。