

第3章

表单



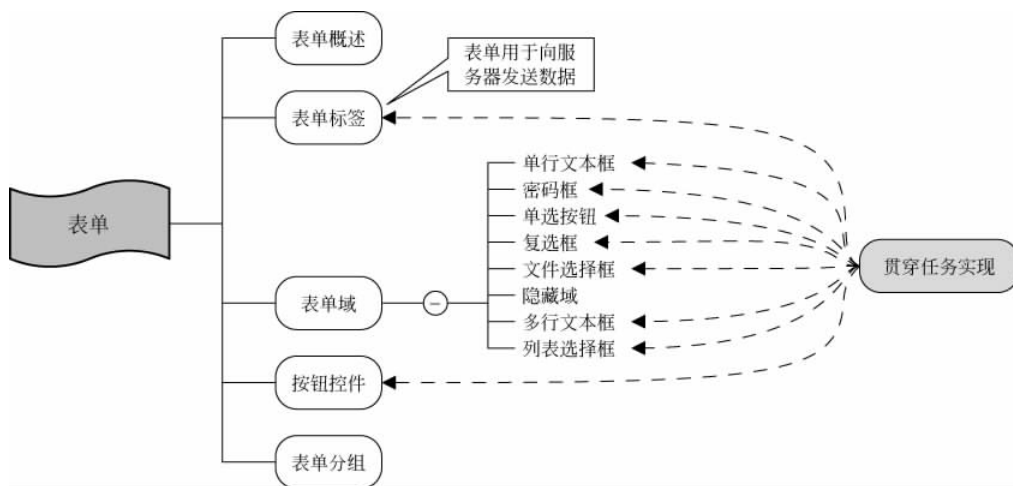
任务驱动

本章任务是完成“Q-Walking E&S 漫步时尚广场”的用户注册页面和后台的商品添加页面：

- 【任务3-1】实现“Q-Walking E&S 漫步时尚广场”的“用户注册”页面。
- 【任务3-2】实现“Q-Walking E&S 漫步时尚广场”后台管理模块——“商品添加”页面。



学习路线



本章目标

知 识 点	Listen(听)	Know(懂)	Do(做)	Revise(复习)	Master(精通)
表单概述	★	★			
表单标签	★	★	★	★	★
表单域	★	★	★	★	★
按钮控件	★	★	★	★	
表单的分组	★	★	★		

3.1 表单概述

随着 Internet 技术的迅速发展,用户对网络使用的需求也越来越高,不仅需要浏览服务器上的内容,而且还希望将用户的信息发送给服务器,实现用户与服务器之间的数据交互。例如:用户在網上订票或购物时,需要将用户的个人基本信息以及购买信息提交给服务器,此时就需要通过表单来实现。

Form 表单是 HTML 的一个重要部分,负责采集和提交用户输入的信息。表单主要分为表单标签及表单控件两大类。表单控件又可细分为表单域和按钮两部分,常见的表单域包括文本框、密码框、多行文本框、单选按钮、复选框、下拉选择框等。

在交互界面中,用户在表单域录入数据后,可通过表单的特殊控件(如提交按钮等)将数据传递给服务器端,由服务器接收表单数据并进行处理。

注意

服务器端处理表单的程序通常采用动态脚本语言,例如: JSP、PHP、ASP.NET、Node.js 等。

3.2 表单标签

表单标签是一个包含表单元素的区域,可以包含一些表单控件,还可以包含其他的 HTML 标签(如:表格、段落、标题等)。一个页面可以拥有一个或多个表单标签,标签之间相互独立,不能嵌套。虽然一个页面可以包含多个表单标签,但用户向服务器发送数据时一次只能提交一个表单中的数据;如需同时提交多个表单,则须使用 JavaScript 的异步交互方式来实现。

表单标签的基本语法格式如下:

【语法】

```
<form action = "处理数据程序的 URL 地址" method = "get|post" name = "表单名称" ... >
</form>
```

其中:

- 属性 action、method 和 name 是<form>标签常用的三个属性;
- action 属性用于指定服务器端用来处理表单的程序的 URL 地址;当提交表单时,使用 action 属性指定的服务器端程序来处理数据,多用于动态交互性网站开发;
- method 属性用于指定以何种方式向服务器端发送数据;
- name 属性指明该表单的名称。

<form>标签对应的属性如表 3-1 所示。

表 3-1 表单标签的属性

属 性	描 述
action	当提交表单时,说明向何处发送表单中的数据
accept	服务器能够处理的内容类型列表,用逗号隔开;目前主流浏览器并不支持,且在 HTML 5 中不再支持
accept-charset	服务器可处理的表单数据字符集
enctype	表单数据内容类型,说明在发送表单数据之前如何对其进行编码,取值可以为 application/x-www-form-urlencoded、multipart/form-data、text/plain
id	设置表单对象的唯一标识符
name	设置表单对象的名称
target	打开处理 URL 的目标位置(不建议使用)
method	规定向服务器端发送数据所采用的方式,取值可以为 get、post
onsubmit	向服务器提交数据之前,执行其指定的 JavaScript 脚本程序
onreset	重置表单数据之前,执行其指定的 JavaScript 脚本程序

下面对< form >标签的各个属性分别进行讲解。

1. action 属性

action 属性值是 Web 服务器上数据处理程序的 URL 地址,或者是 Email 地址。

【示例】 form 元素的 action 属性

```
< form action = "http://www.itshixun.com/web/login. jsp" ></form> <!-- 绝对 URL -->
< form action = "/web/login. jsp" ></form> <!-- 相对 URL -->
< form action = "mailto:guoqy@itshixun. com" ></form> <!-- Email 地址 -->
```

2. method 属性

浏览器使用 method 属性所设置的方式将表单中的数据传送给服务器进行处理,通常采用以下两种方式: get 方式和 post 方式。

get 方式: 将数据作为 URL 的一部分发送给服务器。URL 由地址部分和数据部分构成,两者之间用问号“?”隔开,数据以“名称=值”的方式成对出现,且数据与数据之间通过“&.”符号进行分割,格式如下。

【示例】 通过 get 方式的 URL 传递数据

```
http://www.itshixun.com/web/login. jsp?userName = guoqy&userPwd = 123456
```

get 方式的请求数据可以被缓存,能够保留在浏览器历史记录中,还能作为书签被收藏。由于 get 请求数据会出现在 URL 中,因此安全性比较低;而 URL 地址栏长度有限,最长不能超过 255 个字符,所以 get 方式有长度限制。

post 方式: 将数据隐藏在 HTTP 的数据流中进行传输;请求数据不会出现在地址栏中,安全性比 get 方式要高,并且对数据长度没有限制。

【示例】 post 方式的数据封装

```
POST /web/login. jsp HTTP/1.1
Host: itshixun. com
userName = guoqy&userPwd = 123456
```

与 get 方式不同,post 方式的请求数据不会被缓存,也不会浏览器的历史记录中保

留,更不能被收藏为书签;当单击后退或刷新按钮时,数据会被重复提交。

3. id 与 name 属性

id 属性是表单在网页中的唯一标识。在设计表单页面时,应为每个 form 表单提供一个合适的 id,方便后期的 CSS 样式表及 JavaScript 脚本对其引用。

name 属性用来设置表单元素的名称,在页面中也应尽量保持唯一。在早期版本中,name 属性使用较多,现在多用 id 属性替代。

4. enctype 属性

在表单数据提交之前,需要通过 enctype 属性说明表单中数据的编码方式。目前浏览器支持的编码方式有三种: application/x-www-form-urlencoded、multipart/form-data 和 text/plain 方式。

application/x-www-form-urlencoded 是默认编码方式,大多数表单数据会采用此种编码方式。在发送数据到服务器之前,所有字符都会进行 Unicode 编码,并对某些字符进行特殊处理。例如,遇到空格时将被转换成加号(+),其他特殊字符将转换为对应的 ASCII 格式(即“%XX”格式,由一个百分号和两位代表 ASCII 码的十六进制数字构成)。

以下面数据为例,传递的数据包括用户名(zhang san)和密码(&.)。

```
userName = zhang san
userPwd = &
```

在编码前先使用“&.”符号将数据连接起来,然后再进行编码。经过编码后,最终提交的数据格式为

```
username = zhang + san&userPwd = % 26
```

multipart/form-data 编码方式常用于表单包含文件上传控件的情况,该方式不对字符进行编码。

text/plain 编码方式遇到空格时,会将其转换为加号(+),但不对其他特殊字符进行编码。

5. target 属性

target 属性通常用于<a>标签,用于指明被连接的内容将在哪个框架或浏览器中进行加载。该属性还可以在<form>标签中使用,指明在用户提交表单时,将在哪个框架或浏览器中显示该表单的处理结果。

注意

在 HTML 4.01 中,不再推荐使用<form>表单的 target 属性;在 XHTML 1.0 Strict DTD 中,不支持该属性;在 HTML 5 中推荐使用 formtarget 属性来代替。

3.3 表单域

表单域多用于收集网站访问者的信息,一般位于<form>、</form>标签之间。表单域主要包括文本框、密码框、隐藏域、多行文本框、单选按钮、复选框、列表选择框和文件选择框

等元素。除多行文本框(< textarea >)和列表选择框(< select >)外,大部分表单域使用< input >标签来创建。

< input >标签的通用属性如表 3-2 所示。

表 3-2 < input >标签的通用属性

属 性	描 述
id	设置当前控件的唯一 ID。界面设计时,在 CSS、JavaScript 中可以引用
name	设置当前控件的名称。在向服务器发送数据时,服务器根据 name 属性获取对应表单域的值
value	设置表单域的初始值。网页加载过程中,默认显示该值
type	该属性是必需的,用来说明当前控件的类型。取值可以是 text、password、radio、checkbox、file、hidden、button、submit、reset、image 等
maxlength	设置输入到文本框的最大字符数。输入达到最大数后,用户按下更多键,也不会添加新的字符
size	设置文本输入控件的宽度,单位为字符

3.3.1 单行文本框

单行文本框通常用来输入一些简单的内容。在 HTML 中,通过将< input >标签的 type 属性设为 text 方式来创建一个单行文本框,语法格式如下。

【语法】

```
< input type = "text" name = "..." size = "..." maxlength = "..." value = "..."
      disabled = "disabled" readonly = "readonly" />
```

其中:

- type = "text" 用于指明表单域的类型是单行文本框;
- name 属性用于指定文本框的名称;
- size 属性用于指定文本框的宽度,单位是字符;
- maxlength 属性用于指定文本框的最多输入字符的个数;
- value 属性用于指定文本的初始值;
- disabled = "disabled" 用于指明文本框的禁用状态,并呈灰色显示;
- readonly = "readonly" 用于指定文本框的只读状态。

下面演示了文本框及其属性的用法。

【代码 3-1】 textDemo. html

```
< html >
< head >
  < meta http-equiv = "Content-Type" content = "text/html; charset = utf-8" />
  < title>表单域 - 文本框</title>
</head>
< body >
  < form >
    < input type = "text" /> < br/>< br/>
    < input type = "text" name = "userName" id = "userName" value = "请输入用户名" />< br/>
    < input type = "text" value = "请输入用户名" size = "25" maxlength = "10" />< br/>< br/>
    < input type = "text" value = "请输入用户名" disabled = "disabled" />< br/>< br/>
```

```
<input type="text" value="请输入用户名" readonly="readonly" />
</form>
</body>
</html>
```

上述代码中,第一个文本框是默认显示效果;第二个文本框默认显示的是 value 属性值;第三个指定文本框的宽度为 25 个字符,最大输入长度为 10 个字符,即使后面仍有空白,也无法继续输入;第四个通过 disabled 属性将文本框禁用,呈现灰色禁用状态,既不能编辑也无法获得焦点;第五个通过 readonly 属性将文本框设置为只读状态,虽然可以获得焦点,但无法对文本进行修改。在浏览器中预览效果如图 3-1 所示。



图 3-1 文本框的显示效果

3.3.2 密码框

密码框与文本框相似,但向框内输入内容时,显示的不是当前输入的内容而是掩码形式(星号“*”或其他符号),从而保护用户的输入内容不会被周围的人直接看到,其语法格式如下。

【语法】

```
<input type="password" name="..." size="..." maxlength="..." value="..." />
```

其中:

- type="password"用于指明表单域的类型是密码框;
- name 属性用于指定密码的名称;
- size 属性用于指定密码框的宽度,单位为字符;
- maxlength 属性用于指定密码框的最多输入字符的个数;
- value 属性用于指定密码框的初始值,以掩码的方式显示。

下面演示了密码框及其属性的使用。

【代码 3-2】 passwordDemo. html

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>表单域-密码框</title>
</head>
<body>
  <form>
    用户名: <input type="text" name="userName" id="userName"
      value="请输入用户名"/> <br/>
    密 码: <input type="password" name="userPwd" id="userPwd"/> <br/>
    确认密码: <input type="password" name="userRePwd" id="userRePwd" size="25"
      disabled="disabled" value="请输入确认密码"/> <br/>
  </form>
</body>
</html>
```

上述代码中,密码框也可以使用<input>标签的通用属性,第二个密码框可以被禁用,也可以提供默认值,但只能以掩码形式显示出来。在浏览器中预览效果如图 3-2 所示。



图 3-2 密码框的显示效果

注意

虽然密码框中显示的内容以掩码的方式显示,但是在数据传输过程中,仍然以明文的方式发送到服务器。为了确保数据的安全,应当采用数据加密技术对数据进行处理。

3.3.3 单选按钮

单选按钮是指在一组数据中只能选择其中一个选项。在 HTML 中,通过将<input>标签的 type 属性设为 radio 的方式来创建一个单选按钮选项,语法格式如下。

【语法】

```
<input type="radio" name="..." value="..." checked="checked" />
```

其中:

- type="radio"用于指明表单域的类型是单选按钮;
- name 属性用于指定单选按钮的名称;
- 具有相同 name 属性的单选按钮分为一组,一组只能选择一项;
- value 属性用于指定该项的值,同一组内的值不应相同;
- checked="checked"属性用于指定该项默认被选中,也可以简写成 checked 格式。

当表单中有多个单选按钮时,浏览器会根据单选按钮的 name 属性进行分组。下述代码演示了单选按钮的用法。

【代码 3-3】 radioDemo. html

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>表单域-单选按钮</title>
</head>
<body>
  <form>
    性别: <input type="radio" name="sex" value="man" checked="checked" />男
          <input type="radio" name="sex" value="woman" />女 <br/>
    专业: <input type="radio" name="major" value="computer" />计算机
          <input type="radio" name="major" value="physics" checked="" />物理
```

```
<input type="radio" name="major" value="chemical"/>化工
</form>
</body>
</html>
```

上述代码中,所有单选按钮根据 name 属性分为两组,每组只能选择其中一项。在浏览器中预览效果如图 3-3 所示。



图 3-3 单选按钮组的显示效果

注意

单选按钮的 value 属性并不在浏览器中显示;当表单验证或表单提交服务器时会使用 value 属性值;通过属性 checked="checked" 或 checked 可将当前项设为默认被选中;同一组中有多项具有 checked 属性时,最后一个为默认被选择项。

3.3.4 复选框

复选框是指在一组数据中允许用户选择一项或多项,各项之间并不互斥。在 HTML 中,通过将<input>标签的 type 属性设为 checkbox 的方式来创建一个复选框选项,语法格式如下。

【语法】

```
<input type="checkbox" name="..." value="..." checked="checked"/>
```

其中:

- type="checkbox"用于指明表单域的类型是复选框;
- name 属性用于指定复选框的名称;
- 具有相同 name 属性的复选框分为一组,组内允许多选;
- value 属性用于指定该项的值;
- checked="checked"属性用于指定该项默认被选中,也可以简写成 checked 格式。

当复选框有多项数据时,浏览器会根据复选框的 name 属性进行分组,分组前后在页面显示方面并没有差别,但在使用 JavaScript 特效或向服务器提交数据时,需要对复选框进行合理分组,以便对数据进行处理。

下述代码演示了复选框的用法。

【代码 3-4】 checkboxDemo. html

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

```

<title>表单域 - 复选框</title>
</head>
<body>
  <form>
    爱好: <input type = "checkbox" name = "hobby" value = "music" />音乐
          <input type = "checkbox" name = "hobby" value = "swimming" />游泳
          <input type = "checkbox" name = "hobby" value = "football" checked />足球 <br/>
    选修: <input type = "checkbox" name = "choice" value = "computer" checked />计算机
          <input type = "checkbox" name = "choice" value = "physics" />物理
          <input type = "checkbox" name = "choice" value = "chemical" checked />化工
  </form>
</body>
</html>

```

与单选按钮一样,checked="checked"或 checked 决定复选框在页面加载时是否被选中。当复选框没有 value 属性时,如果被选中则 value 值为 on,否则 value 值为 off。代码在浏览器中显示的效果如图 3-4 所示。



图 3-4 复选框的显示效果

3.3.5 文件选择框

用户通过表单上传文件时,需要使用文件选择框来选择上传文件。在 HTML 中,通过将<input>标签的 type 属性设为 file 方式来创建一个选择框,其语法格式如下。

【语法】

```
<input type = "file" name = "..." accept = "..." />
```

其中:

- type="file"用于指明表单域的类型是文件选择框。
- name 属性用于指定文件选择框的名称。
- accept 属性用于指定文件选择窗口的文件类型过滤。单击选择文件按钮时,会在弹出的文件选择窗口中,根据 accept 指定的类型对文件自动进行过滤,例如,图片的格式包括 image/gif、image/jpeg、image/* 等格式。

使用文件选择框时,form 表单的 enctype 属性应设为 multipart/form-data 类型,method 属性应设为 post 类型。下述代码演示了文件选择框的用法。

【代码 3-5】 fileDemo. html

```

<html>
<head>
  <meta http-equiv = "Content-Type" content = "text/html; charset = utf-8" />
  <title>表单域 - 文件选择框</title>
</head>
<body>
  <form method = "post" enctype = "multipart/form-data">
    请选择上传的头像: <input type = "file" accept = "image/*" name = "headImage" />
  </form>
</body>
</html>

```

上述代码在浏览器中显示的效果如图 3-5 所示。

其中,文件选择框由一个文本框和一个按钮组成。在设置 accept 属性为"image/*"后,单击按钮时弹出文件选择窗口,选择窗口会根据 accept 属性进行过滤,如图 3-6 所示。

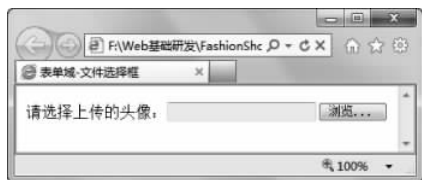


图 3-5 文本选择框的显示效果



图 3-6 文件选择窗口

注意

IE 9 及更早版本不支持 accept 属性,可以通过 IE 10+ 或 Firefox、Opera 等浏览器查看代码预览效果。当需要向服务器上传文件时,建议配合服务器端验证,以防止用户恶意操作。

3.3.6 隐藏域

在网页之间传递数据时,有些数据不希望用户在页面中看到,此时可以通过隐藏域来实现。在 HTML 中,通过将<input>标签的 type 属性设为 hidden 来创建一个隐藏域,语法格式如下。

【语法】

```
<input type="hidden" name="..." value="..." />
```

其中:

- type="hidden"用于指明表单域的类型是隐藏域,该控件不在页面中显示;
- name 属性用于指定隐藏域的名称;
- value 属性用于指定隐藏域的值。

下面演示了隐藏域的用法。

【代码 3-6】 hiddenDemo. html

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>表单域 - 隐藏域</title>
</head>
```

```

< body >
  < form >
    < input type = "hidden" name = "hiddenData" value = "不显示的数据"/>
  </form >
</body >
</html >

```

在浏览器中,隐藏域并没有显示出来。但是在网页的“源代码”中可以找到相应代码,如图 3-7 所示。



图 3-7 隐藏域的显示效果

注意

隐藏域的数据虽然不能在浏览器中直接显示,但是可以通过查看源码的方式找到。因此,不要使用隐藏域保存敏感的数据。隐藏数据也可以通过 CSS 中的 display 属性或 visibility 属性来实现。

3.3.7 多行文本框

多行文本框是用来输入较长内容的文本输入控件。在 HTML 中,通过< textarea >标签创建一个多行文本框。< textarea >与</textarea >标签之间的内容会在页面加载时显示出来,语法格式如下。

【语法】

```
< textarea name = "..." rows = "..." cols = "..." wrap = "..." > 文本内容 </textarea >
```

其中:

- 多行文本框使用< textarea >标签进行定义,文本内容放在< textarea >、</textarea >之间;
- name 属性用于指定多行文本框的名称;
- rows 属性用于指定多行文本框的行数;
- cols 属性用于指定多行文本框的宽度,单位是字符;
- wrap 属性用于指定文本内容大于文本宽度时的显示方式。

下面演示了多行文本框的用法。

【代码 3-7】 textareaDemo. html

```

< html >
< head >

```

```
<meta http-equiv = "Content - Type" content = "text/html; charset = utf - 8" />
<title>表单域 - 多行文本框</title>
</head>
<body>
  <form>
    <textarea rows = "6" cols = "40" wrap = "virtual" name = "个人介绍">您好!
    我来自青岛,我是一名信息工程大学软件工程专业的应届本科即将毕业的学员。
    我的性格偏于内向,为人坦率、热情、讲求原则;处事乐观、专心、细致、头脑清醒;富有责任心、乐于
    助人。
  </textarea>
</form>
</body>
</html>
```

上述代码在浏览器中预览效果如图 3-8 所示。

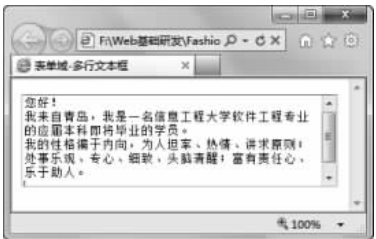


图 3-8 多行文本框的显示效果

< textarea >标签的 wrap 属性用于指定多行文本框的换行方式,取值方式如表 3-3 所示。

表 3-3 wrap 属性值表

值	描 述
off	默认值,文本域中内容足够多时,会在文本域中添加滚动条
virtual	实现文本区内的自动换行,以改善对用户的显示;但在传输给服务器时,文本只在用户按下 Enter 键的地方进行换行,其他地方没有换行的效果
physical	实现文本区内的自动换行,并以这种形式传送给服务器

注意

用户若输入更丰富的内容时,可以采用富文本框 RTE(Rich Text Editor)来实现,当前比较流行的富文本框有 ckEditor、UEditor 以及 kindEditor 等。此处不再一一介绍,读者可根据需要查阅相关资料。

3.3.8 列表选择框

列表选择框允许用户从列表中选择一项或多项。在 HTML 中,可以通过< select >和< option >标签来创建一个列表框,语法格式如下。

【语法】

```
<select name = "..." size = "..." multiple = "multiple">
  <option value = "..." selected = "selected">选项描述内容</option>
```

```
...  
</select>
```

其中：

- 列表选择框使用< select >标签进行定义,一个列表可以包含多个列表项< option >,列表项需要放在< select >、</ select >之间;
 - name 属性用于指定列表选择框的名称;
 - size 属性用于指定列表选择框显示的行数;
 - multiple="multiple"属性用于指明当前列表框是否允许按住 Ctrl 键进行多选,不设置该属性时,默认只能选择一项;
 - < option >标签的 value 属性多用于发送给服务器的选项值;
 - < option >标签的 selected="selected"用于设置当前选项默认被选中;
 - < option >和</option >标签之间的选项描述内容是显示的列表选择项。
- < select >标签对应的属性如表 3-4 所示。

表 3-4 标签< select >的属性列表

属 性	描 述
name	设置列表的名称
size	设置列表中可见选项的数目
multiple	设置列表是否可以选多个选项,可以使用 multiple 或 multiple="multiple"形式;允许选择时,需要按下 Ctrl 键选择多项
disabled	设置列表是否被禁用,可以使用 disabled 或 disabled="disabled"形式

< option >标签对应的属性如表 3-5 所示。

表 3-5 标签< option >的属性列表

属 性	描 述
value	设置该项的值,如果选中该项,该项的值就将发送给服务器
selected	设置当页面加载时,该项是否被选中。可以使用 selected 或 selected="selected"形式
disabled	该项在首次加载时被禁用,可以使用 disabled 或 disabled="disabled"形式

注意

对于具有 boolean 值的属性(如 selected、multiple、readonly、disabled 等),只写属性而不指定属性值时,表示该属性值为 true;指定该属性时,默认为 false。在早期的 HTML 版本中,boolean 类型的属性允许只写属性或“属性=属性值”的方式,而 XHTML 中要求相对严格,属性与属性值必须成对出现,到了 HTML 5 后,又允许只写属性的方式。

下面演示了选择列表框的用法。

【代码 3-8】 selectDemo. html

```
<html>  
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  
  <title>表单域-列表选择框</title>
```

```

</head>
<body>
  <form>
    请选择国家:
    < select name = "country" >
      < option>中国</option>
      < option>俄罗斯</option>
      < option>英国</option>
    </select>
    <br/><br/>请选择省份:
    < select name = "province" size = "4" multiple = "multiple">
      < option value = "SD">山东</option>
      < option value = "NMG">内蒙古</option>
      < option value = "TW" disabled>台湾</option>
      < option value = "SX">山西</option>
      < option value = "HN">湖南</option>
    </select>
    <br/><br/>请选择城市:
    < select name = "city" >
      < option value = "BJ">北京</option>
      < option value = "SH">上海</option>
      < option value = "QD" selected = "selected">青岛</option>
    </select>
  </form>
</body>
</html>

```

上述代码中,第一个列表框由于没有设置 selected 选项,因此默认显示第一项。在第二个列表框中,指定显示项的数量为 4 个,由于设置了 multiple 属性,可按住 Ctrl 键进行多选;<option>标签中的 disabled 属性使得“台湾”选项被禁用,无法选择。第三项中指定“青岛”为默认选项。在浏览器中预览效果如图 3-9 所示。

在一个列表选择框中,还可以将列表项分为多组,使描述更加清晰。HTML 中提供了<optgroup>标签,用于对列表项进行分组。该标签的 label 属性用于指定每个分组的名称,且分组名不能被选择;disabled 属性用于设置该分组是否被禁用。

【代码 3-9】 optionGroupDemo. html

```

<html>
<head>
  <meta http-equiv = "Content-Type" content = "text/html; charset = utf-8" />
  <title>表单域-列表选择框</title>
</head>
<body>
  <form>
    请选择日期:
    < select name = "day" >

```



图 3-9 列表选择框的显示效果

```

<optgroup label = " -- 工作日 -- ">
  <option value = "monday">星期一</option>
  <option value = "tuesday">星期二</option>
  <option value = "wednesday">星期三</option>
  <option value = "thursday">星期四</option>
  <option value = "friday">星期五</option>
</optgroup>
<optgroup label = " -- 休息日 -- ">
  <option value = "saturday">星期六</option>
  <option value = "sunday">星期天</option>
</optgroup>
<optgroup label = " -- 节假日 -- " disabled = "disabled" >
  <option value = "NewYear">春节</option>
  <option value = "MayDay" selected = "selected">五一</option>
  <option value = "OctoberDay">十一</option>
</optgroup>
</select>
</form>
</body>
</html>

```

上述代码中,前两个分组的名称(工作日、休息日)不能被选择,第三个分组由于使用 disabled 属性禁用而整个分组不能被选择。在浏览器中预览效果如图 3-10 所示。



图 3-10 optgroup 分组后的效果

3.4 按钮控件

表单的按钮有多种功能:可以用于提交表单,也可以用于清除或重置表单,甚至能用于触发客户端脚本程序。按钮分为提交按钮、重置按钮、图片按钮和普通按钮。可以通过 <input> 标签或 <button> 标签创建按钮,语法格式如下。

【语法】

```
<input type = "submit|reset|button|image" name = "..." src = "..." value = "..." />
```

其中:

- type 属性用于决定按钮的类型,取值为 submit、reset、button 或 image。其中, submit 表示创建一个能够提交表单的按钮; reset 表示创建一个能将表单重置为初始化状态的按钮; button 表示创建一个普通按钮,当用户单击按钮时,可以触发

JavaScript 脚本的按钮；image 表示创建一个图片按钮，单击时也可以提交表单。

- name 属性用于指定按钮的名称。
- src 属性是可选的，当 type 属性为 image 时，用来指定图片的 URL 地址。
- value 属性用于指定按钮上显示的文字内容。

下述代码演示了< input >按钮的用法。

【代码 3-10】 inputButtonDemo. html

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>表单-按钮的多样性</title>
</head>
<body>
  <form action="http://www.itshixun.com" method="post">
    <input type="submit" value="提交按钮" name="btnSubmit"/>
    <input type="reset" value="重置按钮" name="btnReset"/>
    <input type="button" value="普通按钮" name="btnNormal"
      onClick="alert('可以调用 JavaScript 脚本!')"/>
    <input type="image" src="../images/imageButton.jpg" width="100"/>
  </form>
</body>
</html>
```

上述代码在浏览器中显示的效果如图 3-11 所示。单击普通按钮时，会弹出提醒信息；单击图片按钮时，功能与 submit 按钮基本一样。

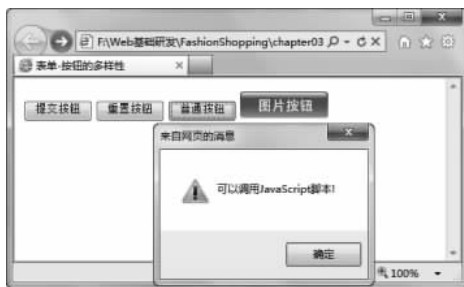


图 3-11 input 按钮显示效果

在 HTML 表单中，还可以使用< button >标签创建一个按钮，语法格式如下。

【语法】

```
<button type="submit|reset|button" value="..." name="...">正文内容</button>
```

button 按钮与 input 按钮相比，提供了更强大的功能和更丰富的内容。< button >与</button>标签之间的内容都是按钮的内容，包含任何可接受的正文内容，例如文本、图片、视频等。

下述代码演示了 button 按钮的用法。

【代码 3-11】 buttonDemo. html

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

```

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>表单 - Button 按钮</title>
</head>
<body>
  <form action="http://www.itshixun.com" method="post">
    <button type="submit" value="提交内容">提交内容 </button>
    <button type="reset" value="重置内容">重置内容 </button>
    <button type="button" value="普通按钮"
      onClick="alert('可以调用 JavaScript 脚本!')">普通按钮 </button>
    <button type="submit" value="提交内容">
      
    </button>
  </form>
</body>
</html>

```

注意

到 IE 4 和 NetScape 6 才开始支持< button>标签。IE 浏览器的默认类型是 button 类型,而其他浏览器(包括 W3C 规范)的默认值是 submit 类型。在提交表单时,IE 提交的是< button>与< button/>之间的文本内容,而其他浏览器提交的是 button 元素的 value 值。

3.5 表单分组

大型表单容易在视觉上产生混淆,通过表单的分组可以将表单上的控件在形式上进行组合,达到一目了然的效果。常见的分组标签有<fieldset>和<legend>标签。

【示例】 表单分组

```

<form>
  <fieldset>
    <legend>请选择个人爱好</legend>
    表单控件...
  </fieldset>
  ...
</form>

```

<fieldset>标签可以看作表单的一个子容器,将所包含的内容以边框环绕方式显示;而<legend>标签则是为<fieldset>边框添加相关的标题。下述代码演示了表单分组效果。

【代码 3-12】 fieldsetDemo.html

```

<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>表单分组</title>

```

```

</head>
<body>
  <form>
    <fieldset>
      <legend>请选择个人爱好</legend>
      <input type="checkbox" name="hobby" value="music" />音乐 <br/>
      <input type="checkbox" name="hobby" value="swimming" />游泳 <br/>
      <input type="checkbox" name="hobby" value="football" />足球 <br/>
    </fieldset>
    <br/>
    <fieldset>
      <legend>请选择个人课程选修情况</legend>
      <input type="checkbox" name="choice" value="computer" />计算机 <br/>
      <input type="checkbox" name="choice" value="physics" />物理 <br/>
      <input type="checkbox" name="choice" value="chemical" />化工 <br/>
    </fieldset>
  </form>
</body>
</html>

```

上述代码中,通过<fieldset>标签将表单分成两组,而<legend>标签为每个分组添加相应的标题。在浏览器中预览效果如图 3-12 所示。



图 3-12 表单分组效果图

3.6 贯穿任务实现

3.6.1 实现【任务3-1】

本小节实现“Q-Walking E&S 漫步时尚广场”贯彻项目中的【任务3-1】“用户注册”页面,如图 3-13 所示。

注册页面使用表格布局进行排版,并通过单行文本框、密码框、单选框、复选框、选择框和按钮等表单元素来实现,代码如下。

图 3-13 “用户注册”页面

【任务3-1】 register. html

```

<!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>用户注册 - Q - Walking E&S</title>
    <link href="css/register.css" rel="stylesheet" type="text/css">
</head>
<body>
<!-- 顶部区域 start -->
<table width="100%" border="0" cellspacing="0" cellpadding="0" class="top_line">
    <tr>
        <td bgcolor="#f7f7f7">
            ... <!-- 此处省略顶部区域代码 -->
        </td>
    </tr>
</table>
<!-- 顶部区域 end -->
<!-- logo 和 banner start -->
<table width="1200" border="0" cellspacing="0" cellpadding="0" align="center">
    <tr>
        <td height="95"></td>
        <td align="right"></td>
    </tr>

```

```

    </tr>
</table>
<!-- logo 和 banner end -->
<!-- 菜单导航 start -->
<table width="100%" border="0" cellspacing="0" cellpadding="0"
    bgcolor="#ce2626">
    <tr>
        <td>
            ... <!-- 此处省略菜单导航代码 -->
        </td>
    </tr>
</table>
<!-- 菜单导航 end -->
<!-- 注册部分 start -->
<table width="100%" border="0" cellspacing="0" cellpadding="0"
    bgcolor="#f8f8f8">
    <tr>
        <td>
            <table width="1000" border="0" cellspacing="0" cellpadding="0"
                bgcolor="#ffffff" align="center">
                <tr>
                    <td valign="top"><h2 align="center">用户注册</h2>
                    <hr width="90%" align="center" color="#ccc"/></td>
                    <td width="420" rowspan="2" valign="middle">
                        </td>
                </tr>
                <tr>
                    <td valign="top">
                        <form action="#" method="post" enctype="multipart/form-data">
                        <table width="90%" border="0" cellspacing="0" cellpadding="0"
                            class="reg" align="center">
                            <tr>
                                <td width="80">用户名: </td>
                                <td><input name="userName" type="text" id="userName"
                                    value="请输入用户名" /></td>
                            </tr>
                            <tr>
                                <td>邮箱地址: </td>
                                <td><input name="email" type="text" id="email"
                                    value="请输入邮箱地址" /></td>
                            </tr>
                            <tr>
                                <td>设置密码: </td>
                                <td><input name="userPwd" type="password" id="userPwd" /></td>
                            </tr>
                            <tr>
                                <td>&nbsp;</td>
                                <td class="gray12">6 - 20 个字符,由字母、数字和符号的两种以上组合。&nbsp;</td>
                            </tr>
                            <tr>
                                <td>确认密码: </td>
                                <td><input name="userRePwd" type="password" id="userRePwd" /></td>
                            </tr>
                            <tr>
                                <td>真实姓名: </td>

```

```

        <td><input name="realName" type="text" id="realName"
            value="请输入真实姓名" /></td>
    </tr>
    <tr>
        <td>您的性别: </td>
        <td><input type="radio" name="sex" value="radio" checked/>男
            <input type="radio" name="sex" value="radio" />女</td>
    </tr>
    <tr>
        <td>上传头像</td>
        <td><input type="file" name="headPic" id="headPic" /></td>
    </tr>
    <tr>
        <td>您的手机: </td>
        <td><input name="mobile" type="text" id="mobile"
            value="请输入您的手机号" /></td>
    </tr>
    <tr>
        <td>单位名称: </td>
        <td><input name="company" type="text" id="company"
            value="请输入单位名称" /></td>
    </tr>
    <tr>
        <td>单位地址: </td>
        <td><select name="province">
            <option>请选择省份</option>
            <option>北京市</option>
            <option>上海市</option>
            <option>山东省</option>
        </select>
        <select name="city">
            <option>请选择城市</option>
            <option>青岛市</option>
            <option>济南市</option>
            <option>东营市</option>
        </select>
        <select name="area">
            <option>请选择区</option>
            <option>四方区</option>
            <option>市南区</option>
            <option>市北区</option>
        </select></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><input name="address" type="text" id="address"
            value="请输入街道地址" /></td>
    </tr>
    <tr>
        <td>您的爱好: </td>
        <td><input name="hobby" type="checkbox" value="购物" />购物
            <input name="hobby" type="checkbox" value="影视" />影视
            <input name="hobby" type="checkbox" value="餐饮" />餐饮</td>
    </tr>
    <tr>

```

```

        <td>协议内容: </td>
        <td><textarea cols = "30" rows = "3"></textarea></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><input type = "submit" name = "button" value = "提交" /></td>
    </tr>
</table>
</form></td>
</tr>
</table>
<!-- 三大模块图片 -->
<table width = "1000" border = "0" cellspacing = "0" cellpadding = "0" align = "center"
    bgcolor = "#FFFFFF" class = "padding - bottom">
    <tr>
        <td align = "center"><img src = "images/shop.jpg" class = "bian"/></td>
        <td align = "center"><img src = "images/movie.jpg" class = "bian"/></td>
        <td align = "center"><img src = "images/food.jpg" class = "bian"/></td>
    </tr>
</table></td>
</tr>
</table>
<!-- 注册部分 end -->
<!-- 底部 start -->
    ... <!-- 此处省略版权部分代码 -->
<!-- 底部 end -->
</body>
</html>

```

3.6.2 实现【任务3-2】

本小节实现“Q-Walking E&S 漫步时尚广场”贯彻项目中的【任务3-2】后台管理模块中的“添加商品”页面,如图 3-14 所示。



图 3-14 后台“添加商品”页面

“添加商品”页面中包含商品名称、缩略图、类别、单价、团购价、商品数量、发布日期以及商品的描述等信息,代码如下所示。

【任务3-2】 addgoods.html

```
<!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>添加商品页面 - 后台管理系统</title>
<link href="css/layout.css" rel="stylesheet" type="text/css" />
<link href="css/add.css" rel="stylesheet" type="text/css" />
</head>
<body>
<div class="place"><span>位置:</span>
    <ul class="placeul">
        <li><a href="main.html" target="_parent">首页</a></li>
        <li><a href="#">添加商品</a></li>
    </ul>
</div>
<div class="formbody">
    <table width="100%" border="0" cellspacing="0" cellpadding="0" class="table1">
        <tr>
            <td><label>商品缩略图<b>*</b></label></td>
            <td><input name="" type="file" multiple /></td>
        </tr>
        <tr>
            <td><label>商品名称<b>*</b></label></td>
            <td><input name="" type="text" class="dfinput" value="请填写商品名称"
                style="width:518px;" /></td>
        </tr>
        <tr>
            <td><label>商品类别<b>*</b></label></td>
            <td><select class="select3">
                <option>男装</option>
                <option>女装</option>
                <option>童装</option>
                <option>运动</option>
                <option>其他</option>
            </select></td>
        </tr>
        <tr>
            <td><label>商品单价<b>*</b></label></td>
            <td><input name="goodsPrice" type="text" class="dfinput"
                style="width:100px;" />元</td>
        </tr>
        <tr>
            <td><label>团购价<b>*</b></label></td>
            <td><input name="groupPrice" type="text" class="dfinput"
                style="width:100px;" />元</td>
        </tr>
    </table>
</div>
```

```

        </tr>
        <tr>
            <td><label>商品数量<b>*</b></label></td>
            <td><input name = "amount" type = "text" class = "dfinput"
                style = "width:100px;" />件</td>
        </tr>
        <tr>
            <td><label>发布日期<b>*</b></label></td>
            <td><input name = "upTime" type = "text" class = "dfinput"
                style = "width:100px;" /></td>
        </tr>
        <tr>
            <td><label>是否审核<b>*</b></label></td>
            <td><select name = "checkup" class = "select3">
                <option>已审核</option>
                <option>未审核</option>
            </select></td>
        </tr>
        <tr>
            <td><label>商品描述<b>*</b></label></td>
            <td><textarea name = "content" rows = "3" id = "content"></textarea></td>
        </tr>
        <tr>
            <td></td>
            <td><input type = "button" class = "btn" value = "马上发布" /></td>
        </tr>
    </table>
</div>
</div>
</body>
</html>

```

本章总结

小结

- 表单(form)是 HTML 的一个重要部分,负责采集和提交用户输入的信息;
- 一个页面中可以包含多个表单,但用户一次只能向服务器发送一个表单中的数据;
- 属性 enctype 取值可以是 application/x-www-form-urlencoded、multipart/form-data 或 text/plain;
- method 属性设置的方法是将表单中的数据传送给服务器进行处理,分为 get 方式和 post 方式;
- 常见的表单域有文本框、密码框、多行文本框、单选按钮、复选框、下拉选择框等;

- 按钮主要分为提交按钮、重置按钮、图片按钮、普通按钮，具体可以通过< input >或< button >标签来实现；
- 通过表单分组的方式，可以将表单上的控件在形式上进行组合，使其一目了然。

Q&A

1. 问题：用户通过表单上传文件时，需要使用文件选择框来选择上传文件。但在设置 accept 属性为"image/ * "后，IE 浏览器在选择文件时，并没有对选择文件进行过滤，而是包括所有的文件及文件夹。
- 回答：各大浏览器采用的标准不同，对 HTML 标签及属性的解析方式也不完全相同，IE 8/9 暂不支持该属性，可切换其他浏览器查看效果。
2. 问题：在一组单选按钮中，允许多项被选中。
- 回答：在单选按钮组中，由于 name 属性值不一致，浏览器将其分为不同分组，不同分组允许选择一项，以致在视觉上产生错觉。

章节练习

习题

1. method 属性的取值可以是_____或_____，其中_____为默认值。
2. < input > 标签的 type 属性常见的取值有_____、_____、radio、_____、_____、_____、button、_____、_____和 image。
3. 列表选择框是通过_____和_____标签构成的。
4. _____标签可以对列表选项框中的选项进行分组。
5. 按钮主要分为_____、重置按钮、_____、普通按钮。
6. 在< textarea >标签中，_____属性用来设置文本输入框的宽度，_____属性用来设置文本输入框的高度。
7. 在列表选择框中，_____属性规定该列表允许多选。
8. 单选按钮分组是根据_____属性进行划分的。

上机

1. 训练目标：“用户登录”页面的设计。

培养能力	熟练使用表单及表单元素		
掌握程度	★★★★★	难度	易
代码行数	100	实施方式	编码强化
结束条件	独立编写，不出错	涉及页面	login. html
参考训练内容			
(1) 使用表格、表单及表单元素实现“用户登录”页面，如图 3-15 所示；			
(2) 使用 IE、FireFox 或 Chrome 浏览器查看页面效果			



图 3-15 “用户登录”页面

2. 训练目标：“后台系统管理”中的“添加商品”页面。

培养能力	熟练使用表单及表单元素		
掌握程度	★★★★★	难度	中
代码行数	150	实施方式	编码强化
结束条件	独立编写,不出错	涉及页面	addfood. html

参考训练内容

- (1) 使用表格、表单及表单元素实现“添加商品”页面,如图 3-16 所示;
- (2) 使用 IE、FireFox 或 Chrome 浏览器查看页面效果



图 3-16 “添加商品”页面