

## 模块五

## 表 单 页 面

**问题提出：**网页不仅可以向用户展示丰富多彩的文字、图片、音视频等内容，还能通过表单与用户进行交互。通过表单可以采集用户的信息，如姓名、性别、年龄、职业、爱好、联系方式等，还可以制作问卷调查、在线订购、留言本等，配合各类服务端程序从而获取数据以满足各种需求。表单在页面中有着非常重要的作用，那如何设计表单呢？本模块基于表单及其属性、表单元素及其属性以及表单验证等知识点来完成“登录”表单和“联系页”表单两个项目，全面学习表单的设计。

**核心概念：**表单，<fieldset>标签，<input>标签，<select>标签，<label>标签，<textarea>标签，<datalist>标签，<output>标签。

**表单：**表单是网页上的一个特定区域，由一对<form>标签定义，用来搜集用户的相关信息，有效实现用户和网站的互动。

<fieldset>标签：可将表单内容的一部分打包，生成一组相关表单的字段。

<input>标签：用于搜集用户信息。根据不同的 type 属性值，输入字段拥有很多种形式。输入字段可以是文本字段、复选框、掩码后的文本控件、单选按钮、按钮等。

<select>标签：允许访问者从选项列表中选择一项或几项，作用相当于单选框（单选时）或复选框（多选时）。

<label>标签：用于为表单 input 元素定义标签，用户单击这个标签后，可以实现将光标聚焦到对应的 input 元素上。

<textarea>标签：用来创建多行文本区域。

<datalist>标签：用于定义选项列表。与 input 元素配合使用，可以制作出输入值的下拉列表。

<output>标签：用于定义不同类型的输出，该标签必须从属于某个表单。

**学习目标：**

- 了解表单的功能与安全性问题。
- 熟悉表单的属性。
- 掌握表单<input>输入标签的多种形式以及各自的属性。
- 熟悉<select>标签、<label>标签、<textarea>标签、<datalist>标签、<output>标签的属性及使用方法。
- 理解 HTML5 的验证方法。
- 了解正则表达式。
- 能设计简洁、美观、实用的表单页面。
- 分析、归纳关于如何优化表单设计的方法，培养学生具有批判性的思维能力、缜密的逻辑推理能力以及团结协作的能力，进一步提升自身的职业素养。

# 项目 5.1 “走进 Web 前端开发”网站 “登录”表单设计

## 5.1.1 项目描述

“走进 Web 前端开发”网站“登录”表单主要包括“用户输入用户名和密码”区域、“下次自动登录”区域以及“登录”按钮、“忘记密码?”和“注册”区域,效果如图 5-1 所示。



图 5-1 “走进 Web 前端开发”网站“登录”表单效果

## 5.1.2 项目分析

### 1. “登录”表单结构

“登录”表单在页面侧边栏的最上方,其基本结构如图 5-2 所示。

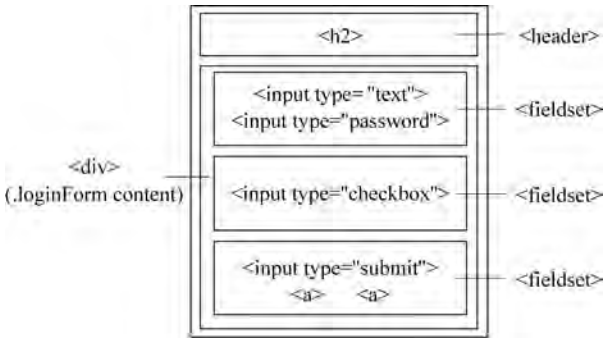


图 5-2 “走进 Web 前端开发”网站“登录”表单结构

### 2. 具体实现细节

表单由< header >和< div >组成,其中,< div >中包含三个< fieldset >区块。

- (1) < header >中包含一个< h2 >标题文字。
- (2) 所有< fieldset >无边框。
- (3) 第一个< fieldset >区块中包含单行文本输入框和密码框,两者边框都为 1px 的灰色实线。
- (4) 第二个< fieldset >区块中包含一个垂直居中的复选框。
- (5) 第三个< fieldset >区块中包含一个“登录”按钮及两个超链接,“登录”按钮背景是渐变色,鼠标指针悬停其上时背景发生变化。

### 5.1.3 项目知识点分解

通过项目分析得知,完成此表单需首先掌握创建表单、创建单行文本框、密码框、复选框、“登录”按钮等表单元素以及创建< fieldset >等知识点。

项目 5-1 知识点分解导图如图 5-3 所示。

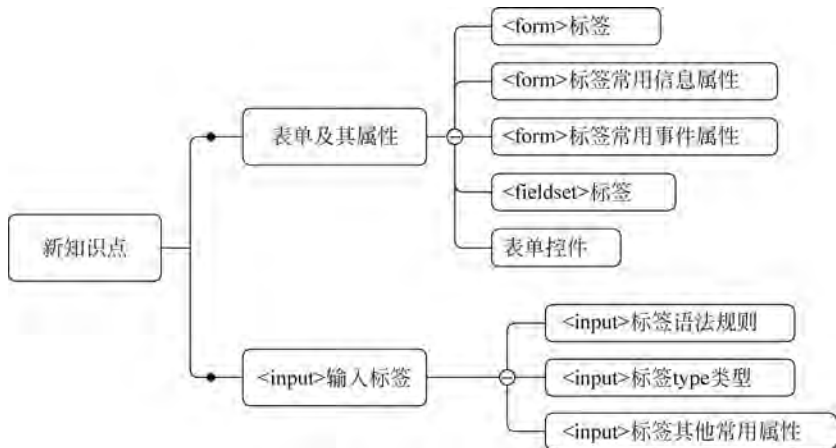


图 5-3 项目 5-1 知识点分解

### 5.1.4 知识点解析

#### 1. 表单及其属性

##### 1) < form > 标签

表单是网页上的一个特定区域,由一对< form >标签定义。< form >标签一方面可以限定表单的范围,即定义一个区域,表单各元素都要在这个区域内,单击“提交”按钮时,提交的也是这个区域内的数据;另一方面,还携带表单的相关信息,如处理表单的程序、提交表单的方法等。其基本语法如下。

```
< form name = "表单名称" action = "url" method = "提交方式">
    各种表单控件
</form>
```

##### 2) < form > 标签常用信息属性

< form > 标签的常用属性除了 name、action 和 method,还包括 enctype、novalidate、autocomplete、target 等信息属性,如表 5-1 所示。



表 5-1 < form > 标签常用信息属性及其描述

| 属 性          | 描 述                                                                                                                                                                                                                                                  |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name         | 定义表单名称,用于脚本引用                                                                                                                                                                                                                                        |
| method       | 定义表单数据从客户端传送到服务器的方法。<br>get: 默认方法,是将表单内容附加到 URL 地址后面,所以对提交信息的长度进行了限制,如信息太长,会被截去;同时不具有保密性,不能传送非 ASCII 码的字符。执行搜索操作时建议使用 get 方法。<br>post: 将用户在表单中填写的数据包含在表单的主体中,一起传送给服务器上的处理程序,无字符个数和字符类型的限制;传送的数据不会显示在浏览器的地址栏中。<br>当所传送的数据用于执行插入或更新数据库操作时,建议使用 post 方法 |
| action       | 指定处理表单的服务端程序                                                                                                                                                                                                                                         |
| enctype      | 规定在发送到服务器之前应该如何对表单数据进行编码。默认地,表单数据会编码为“application/x-www-form-urlencoded”。也就是说,在发送到服务器之前,所有字符都会进行编码(空格转换为“+”,特殊符号转换为 ASCII HEX 值)                                                                                                                     |
| target       | 规定在何处打开 action URL。可取的值和超链接<a>的 target 属性完全一样,包括_blank、_parent、_top 和 framename                                                                                                                                                                      |
| novalidate   | 指定在提交表单时取消对表单进行有效的检查。为表单设置该属性时,可以关闭整个表单的验证,这样可以使 form 内的所有表单控件不被验证                                                                                                                                                                                   |
| autocomplete | 指定表单是否有自动完成功能。“自动完成”是指将表单控件输入的内容记录下来,当再次输入时,会将输入的历史记录显示在一个下拉列表里,以实现自动完成输入。<br>on: 表单有自动完成功能,默认值。<br>off: 表单无自动完成功能                                                                                                                                   |

3) < form > 标签常用事件属性

< form > 标签常用事件属性如表 5-2 所示。

表 5-2 < form > 标签常用事件属性及其描述

| 属 性           | 描 述             |
|---------------|-----------------|
| onchange      | 在元素值被改变时运行的脚本   |
| oncontextmenu | 当上下文菜单被触发时运行的脚本 |
| onfocus       | 当元素获得焦点时运行的脚本   |
| onformchange  | 在表单改变时运行的脚本     |
| onforminput   | 当表单获得用户输入时运行的脚本 |
| oninput       | 当元素获得用户输入时运行的脚本 |
| oninvalid     | 当元素无效时运行的脚本     |
| onselect      | 在元素中文本被选中后触发    |
| onsubmit      | 在提交表单时触发        |

4) < fieldset > 标签

fieldset 元素可将表单内的相关元素分组。即利用< fieldset > 标签可将表单内容的一部分打包,生成一组相关表单的字段。当一组表单元素放到 < fieldset > 标签内时,浏览器会以特殊方式来显示它们,它们可能有特殊的边界、3D 效果,或者甚至可创建一个子表单来

处理这些元素。

<fieldset> 标签没有必须的或唯一的属性。<legend> 标签可以为 fieldset 元素定义标题。

具体使用方法参考下面的案例。

filedset.html:

```
<!doctype html >
<html >
<head >
<meta charset = "utf - 8">
<title>fieldset 标签实例</title>
</head>
<body>
<form >
  <fieldset >
    <legend>注册</legend><!-- 为 fieldset 元素定义标题 -->
    姓名: <input type = "text">
    <br >
    密码: <input type = "password">
    <br >
  </fieldset >
</form >
</body >
</html >
```

filedset.html 页面效果如图 5-4 所示。

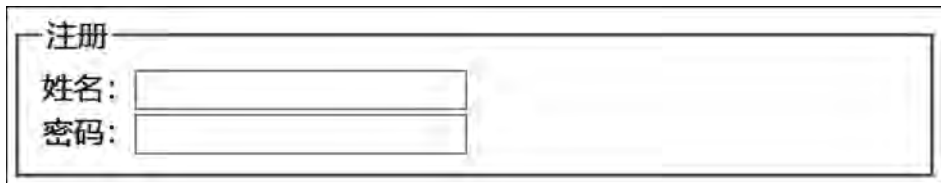


图 5-4 fieldset 标签实例效果

### 5) 表单控件

只有一个表单是无法实现其功能的,需要通过表单的各种控件,用户才可以输入信息、从选项中选择和提交信息等。HTML5 常用的表单控件有 input、select、textarea、label、datalist 等。

所有的表单控件对象都具有一个 name 属性,JavaScript 脚本通过 name 属性的值来引用特定的表单控件元素,同时这也是表单提交到服务器时,每个表单控件元素的值 value 所对应的 key 值;绝大部分对象都具有 value 属性,该属性返回当前表单控件的值;所有的表单控件对象都具有一个 form 属性,该属性返回包含当前控件的 Form 对象。对于一个通用的表单数据检查程序来说,用这个属性来标明哪些控件属于哪个表单;所有的表单元素对象都具有 focus() 和 blur() 方法,同时所有的表单元素对象还具有 onfocus 和 onblur 事件处理器。



2. <input>输入控件

1) <input>标签语法规则

大部分的表单控件元素都是由<input>标签创建的,基本语法如下。

```
<input type = "控件输入类型"/>
```

其中,type 的属性值指定输入类型,即表单控件的类型。

2) <input>标签 type 类型

常见的<input>标签 type 属性值如表 5-3 所示。

表 5-3 <input>标签 type 属性值及描述

| type 属性值 | 描 述           | type 属性值                                                   | 描 述          |
|----------|---------------|------------------------------------------------------------|--------------|
| text     | 单行文本框         | url                                                        | URL 地址的输入域   |
| password | 密码输入框         | number                                                     | 数值的输入域       |
| file     | 文件域           | range                                                      | 一定范围内数字值的输入域 |
| hidden   | 隐藏域           |                                                            |              |
| radio    | 单选按钮          | date pickers(date、month、week、time、datetime、datetime-local) | 日期和时间的输入类型   |
| checkbox | 复选框           |                                                            |              |
| button   | 普通按钮          |                                                            |              |
| submit   | 提交按钮          | search                                                     | 搜索域          |
| reset    | 重置按钮          | color                                                      | 颜色输入类型       |
| image    | 图像按钮          | tel                                                        | 电话号码输入类型     |
| email    | E-mail 地址的输入域 |                                                            |              |

3) <input>标签其他常用属性

<input>标签除 type 属性外,还可以定义很多其他属性,如表 5-4 所示。

表 5-4 <input>标签常用属性及描述

| 属 性          | 描 述                  |
|--------------|----------------------|
| name         | 控件的名称                |
| value        | input 元素设定值          |
| size         | input 控件在页面中的显示宽度    |
| placeholder  | 为 input 类型的输入框提供一种提示 |
| required     | 规定输入框填写的内容不能为空       |
| checked      | 定义选择控件默认被选中的项        |
| maxlength    | 控件允许输入的最多字符数         |
| autocomplete | 设定是否自动完成表单字段的内容      |
| autofocus    | 设定页面加载后是否自动获取焦点      |
| readonly     | 该控件为只读               |
| disabled     | 第一次加载页面时禁用控件         |
| multiple     | 指定输入框是否可以选多个值        |
| min,max,step | 指定输入框允许的最大值、最小值及间隔   |
| pattern      | 验证输入内容是否与定义的正则表达式匹配  |
| form         | 设定字段隶属于哪一个或多个表单      |
| list         | 指定字段的候选数据值列表         |

下面通过一个案例来演示< input >标签的具体使用,代码如下。

```
<!DOCTYPE html >
<html >
<head lang = "en">
<meta charset = "utf - 8">
<title>常用 input 控件</title>
</head>
<body>
<h2>“前端课堂”新用户注册</h2>
<form action = " #" method = "post">
  <!-- text 单行文本输入框 -->
  用户名: <input type = "text" value = "请输入用户名" maxlength = "20">
  <br/>
  <br/>
  <!-- password 密码输入框 -->
  登录密码:<input type = "password" size = "20">
  <br/>
  <br/>
  确认密码:<input type = "password" size = "20">
  <br/>
  <br/>
  <!-- radio 单选按钮 -->
  性别: <input type = "radio" name = "sex" checked = "checked"/>男
  <input type = "radio" name = "sex"/>女<br/>
  <br/>
  <!-- 日期输入域 -->
  出生年月日: <input type = "datetime - local">
  <br/>
  <br/>
  <!-- 颜色选择域 -->
  喜欢的颜色: <input type = "color" value = "# ff0000"/>
  <br/>
  <br/>
  <!-- checkbox 复选框 -->
  喜爱的课程:
  <input type = "checkbox"/>数据库原理
  <input type = "checkbox"/>前端开发
  <input type = "checkbox"/>操作系统
  <br/>
  <br/>
  <!-- file 文件域 -->
  照片: <input type = "file"/>
  <br/>
  <br/>
  <!-- 专门用于搜索关键词的文本框 -->
  座右铭: <input type = "search"/>
  <br/>
  <br/>
  <!-- 一定范围内的数值输入域 -->
  你觉得本课程的难易程度:<input type = "range" min = "1" max = "10"/>
```

```

<br/>
<br/>
<!-- reset 重置按钮 -->
<input type="reset" value="重置"/>
<!-- submit 提交按钮 -->
<input type="submit" value="提交"/>
</form>
</body>
</html>

```

在上面的注册表单中,可输入用户名、登录密码、确认密码、座右铭,还可以选择性别、出生年月日、喜欢的颜色、喜爱的课程、自己的照片、课程的难易程度等,页面效果如图 5-5 所示。

**“前端课堂” 新用户注册**

用户名:

登录密码:

确认密码:

性别: ☒ 男 ☐ 女

出生年月日:

喜欢的颜色:

喜爱的课程: ☐ 数据库原理 ☐ 前端开发 ☐ 操作系统

照片:  未选择任何文件

座右铭:

你觉得本课程的难易程度:

图 5-5 注册表单页面效果

### 课堂小实践

运用已学习的创建表单和创建<input>控件的方法,设计一个在线商城的用户注册表单。

### 5.1.5 知识点检测



### 5.1.6 项目实施

通过项目分析和知识点学习,相信读者已能逐步完成“走进 Web 前端课堂”网站“登录”表单代码的编写。参考代码如下。

HTML 部分:

```
<article class = "loginForm">
  <form action = " #" method = "post">
    <header>
      <h2>登录</h2>
    </header>
    <div class = "loginForm_content">
      <fieldset>
        <!-- <fieldset>标记用来组合表单中的相关元素 -->
        <input type = "text" name = "userName" placeholder = "邮箱/会员账号/手机号"
autofocus required>
        <input type = "password" name = "password" placeholder = "请输入密码" required>
      </fieldset>
      <fieldset>
        <input type = "checkbox" checked = "checked">
          下次自动登录
      </fieldset>
      <fieldset>
        <input type = "submit" value = "登录">
        <a href = " #">忘记密码?</a><a href = " #">注册</a>
      </fieldset>
    </div>
  </form>
</article>
```

对应的 style.CSS 代码:

```
.loginForm {
  width: 96%;
  border-bottom: 1px solid #cccccc;
  margin-bottom: 20px;
}
.loginForm h2 {
  text-align: center;
  color: #666;
  line-height: 3em;
  margin: 16px 0 10px 0;
  letter-spacing: 4px;
  font: normal 24px/1 Microsoft YaHei, sans-serif;
}
.loginForm_content {
  margin-bottom: 20px;
```

```

}
fieldset {
    border: none;
    padding: 10px 10px 0;
    font-size: 12px;
}
/* 文本输入框和密码输入框的样式设计 */
fieldset input[type = text], fieldset input[type = password] {
    line-height: 2em;
    font-size: 12px;
    height: 24px;
    padding: 3px 8px;
    width: 90%;
    border-radius: 3px;
    border: 1px solid #CCC;
    margin: 10px 10px 0;
}
/* 提交按钮的样式设计 */
fieldset input[type = submit] {
    text-align: center;
    padding: 2px 20px;
    line-height: 2em;
    border: 1px solid #cccccc;
    border-radius: 3px;
    background: -webkit-gradient(linear, left top, left 28, from(#EF5356), color-stop
(0%, #FF0000), to(#EF5356));
    background: -moz-linear-gradient(top, #EF5356, #FF0000 0, #EF5356 28px);
    background: -o-linear-gradient(top, #EF5356, #FF0000 0, #EF5356 28px);
    background: linear-gradient(top, #EF5356, #FF0000 0, #EF5356 28px);
    height: 35px;
    cursor: pointer;
    letter-spacing: 6px;
    margin-left: 10px;
    color: #FFF;
    font-weight: bold;
    font-size: 14px;
}
fieldset input[type = submit]:hover {
    background: -webkit-gradient(linear, left top, left 28, from(#FF0000), color-stop
(0%, #EF5356), to(#FF0000));
    background: -moz-linear-gradient(top, #FF0000, #FF6900 0, #FF0000 28px);
    background: -o-linear-gradient(top, #FF0000, #FF6900 0, #FF0000 28px);
    background: linear-gradient(top, #FF0000, #FF6900 0, #FF0000 28px);
}
/* 复选框的样式设计 */
fieldset input[type = checkbox] {
    margin-left: 10px;
    vertical-align: middle;
}
fieldset a {
    font-size: 12px;
    margin: 6px 0 0 28px;
    text-decoration: none;
}

```

### 5.1.7 项目总结

项目实施过程中,主要是练习< form >表单、< fieldset > 标签与< input >标签的使用方法;要特别注意表单的提交方式及复选框的语法规则;搜集、分析、归纳登录表单的设计方法,培养批判性的思维能力、缜密的逻辑推理能力,进一步提升自身的职业素养。

## 项目 5.2 “走进 Web 前端开发”网站 “联系我们”表单设计

### 5.2.1 项目描述

“走进 Web 前端开发”网站“联系页”页面(linkus. html)“联系我们”表单主要是学习者填写姓名、所学专业、电子邮件、手机号码、学习历程简介、感兴趣的前端技术,单击“立即联系”按钮后可以将信息提交到服务器;若信息未填写或填写错误不能提交,出现错误提示。表单效果如图 5-6 所示。

The image shows a web form titled "联系我们" (Contact Us) with a subtitle "\* 表示必填项" (asterisk indicates required field). The form contains the following elements:

- 姓名:** A text input field with a required field asterisk.
- 所学专业:** A text input field containing "软件工程" (Software Engineering) with a required field asterisk.
- 电子邮件:** A text input field containing "chenjx5120@sina.com" with a required field asterisk.
- 手机号码:** A text input field containing "XXX-XXXX-XXXX" with a required field asterisk.
- 感兴趣的前端技术:** A row of four checkboxes labeled "HTML5", "CSS3", "JavaScript", and "jQuery".
- 学习历程简介:** A large text area with the placeholder text "请输入学习历程" (Please enter learning process).
- 立即联系:** A dark button at the bottom of the form.

图 5-6 “联系我们”表单结构效果

### 5.2.2 项目分析

#### 1. 表单结构

“走进 Web 前端开发”网站“联系页”页面“联系我们”表单结构如图 5-7 所示。

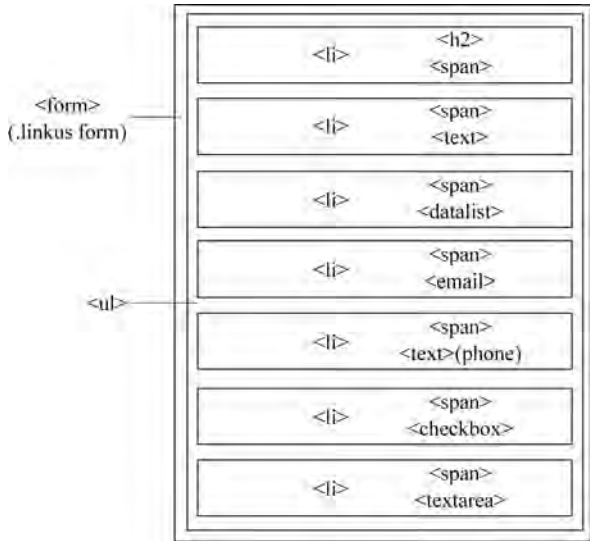


图 5-7 “联系我们”表单结构

#### 2. 具体实现细节

表单简洁、美观、实用，具体细节如下。

(1) `<form>` 表单嵌套 `<ul>`，`<ul>` 中又嵌套了 `<span>`，以便于表单控件能整齐排列。

(2) “感兴趣的前端技术”复选框运用了 `<label>` 标签定义标注，单击文字时选择按钮便被选中。

(3) 表单中所有输入框设置为统一的样式，同时设置了输入框获取焦点、填写内容有效时和填写内容无效时样式的变化。

(4) 表单中的“提交”按钮也进行了美化。

### 5.2.3 项目知识点分解

由上述具体实现细节，该项目涉及的新知识点导图如图 5-8 所示。

### 5.2.4 知识点解析

#### 1. 表单其他控件

除了 `<input>` 标签外，HTML 其他常用的表单控件还有 `<select>`、`<label>`、`<textarea>`、`<datalist>` 和 `<output>` 等。

##### 1) `<select>` 标签

选择列表标签允许访问者从选项列表中选择一项或几项，作用相当于单选框（单选时）或复选框（多选时）。在选项较多的情况下，相对于单选框和多选框来说，选择列表可节约很大的空间。



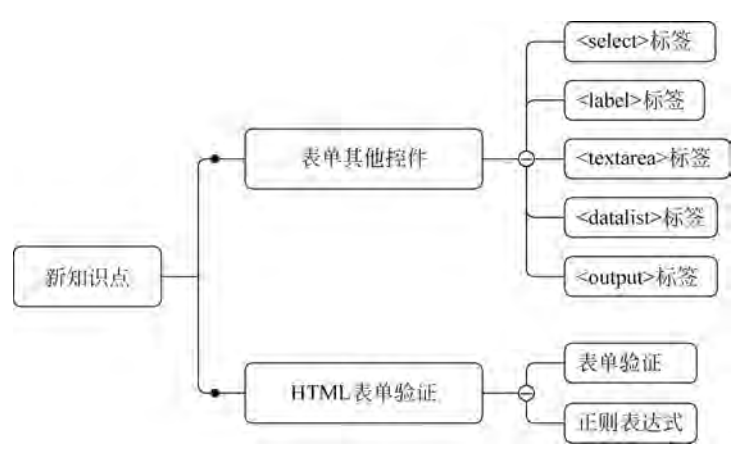


图 5-8 项目 5-2 知识点分解

创建选择列表标签需要使用< select >和< option >标签。< select >标签用于声明选择列表,需由它确定选择列表是否可以多选,以及一次可以显示的选项数;< option >标签用于设置各选项的值以及是否为默认选项。常用属性如表 5-5 所示。

表 5-5 < select >标签常用属性及其描述

| 标 签    | 属 性      | 描 述                            |
|--------|----------|--------------------------------|
| select | name     | 指定列表的名称                        |
|        | size     | 定义能同时显示的列表项个数(默认为 1),取值大于或等于 1 |
|        | multiple | 定义列表中的选项可多选,没有该属性时只能选择一个选项     |
| option | value    | 设置选项值                          |
|        | selected | 设置默认选项,可对一到多个列表选项进行属性的设置       |

按照列表一次可被选择和显示的个数,可分为多项选择列表和下拉列表两种形式。

(1) 多项选择列表

多项选择列表是指一次可以选择多列选项,且一次可以显示一个以上选项的选择列表,下面通过一个案例来演示多项列表的使用方法。

select1.html:

```
<!doctype html >
<html >
<head >
<meta charset = "utf - 8">
<title>多项选择列表</title>
</head>
<body >
<h3>喜欢的城市</h3>
<select name = "city" size = "4" multiple>
  <option value = "yancheng">盐城</option>
  <option value = "suzhou">苏州</option>
  <option value = "wuxi" selected = "selected">无锡</option>
  <option value = "nanjing">南京</option>
```

```

</select>
</body>
</html>

```

页面效果如图 5-9 所示。

## (2) 下拉列表

下拉列表是指一次只能选择一个列表选项,且一次只能显示一个选项的选择列表。下面通过一个案例来演示下拉列表的使用方法。

select2. html:

```

<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>下拉列表</title>
</head>
<body>
<h3>周一第一节课</h3>
<select name="lesson">
  <option value="daishu">高等代数</option>
  <option value="yingyu">大学英语</option>
  <option value="chengxu" selected="selected">程序设计</option>
  <option value="tiyu">体育</option>
</select>
</body>
</html>

```

页面效果如图 5-10 所示。



图 5-9 select1. html 页面效果



图 5-10 select2. html 页面效果

## 2) <label>标签

<label>标签用于为表单 input 元素定义标签,用户单击这个标签后,可以实现将光标聚焦到对应的 input 元素上。label 元素不会向用户呈现任何特殊的样式,但若用户单击 label 元素内的文本,会切换到控件本身,改善了可用性。下面通过一个案例来演示<label>标签的使用方法。

label. html:

```

<!DOCTYPE HTML>

```

```
<html>
<head>
<meta charset = "utf - 8">
<title><label>标签的使用</title>
</head>
<body>
<form>
  <label for = "male">男生</label>
  <input type = "radio" name = "sex" id = "male"/>
  <br/>
  女生
  <input type = "radio" name = "sex" id = "female"/>
</form>
</body>
</html>
```

上述 HTML 代码中,创建了“男生”和“女生”两个单选按钮,其中,“男生”单选按钮中添加了 id 属性,同时添加了<label>标签,给其添加 for 属性,并设置其属性值等于控件的 id 值,这样通过 id 值就建立了<label>和控件的联系。单击进行选择,要选中“男生”选项,除了单击单选框外还可以通过单击文本“男生”来选中;但是未给“女生”设置<label>标签和控件的联系,仅能通过单击单选框选中“女生”选项。

3) <textarea>标签

<textarea>标签用来创建多行文本区域,其常用属性如表 5-6 所示。

表 5-6 <textarea>标签常用属性

| 属 性         | 值                | 描 述                                             |
|-------------|------------------|-------------------------------------------------|
| autofocus   | autofocus        | 规定在页面加载后文本区域自动获得焦点                              |
| disabled    | disabled         | 规定禁用该文本区域                                       |
| form        | form_id          | 规定文本区域所属的一个或多个表单                                |
| maxlength   | number           | 规定文本区域的最大字符数                                    |
| name        | name_of_textarea | 规定文本区域的名称                                       |
| placeholder | text             | 规定描述文本区域预期值的简短提示                                |
| readonly    | readonly         | 规定文本区域为只读                                       |
| required    | required         | 规定文本区域是必填的                                      |
| rows        | number           | 规定文本区域内的可见宽度                                    |
| cols        | number           | 规定文本区域内的可见行数                                    |
| wrap        | soft             | 默认值为 soft,表单提交时,textarea 中的文本不换行                |
|             | hard             | 取值为 hard 时,表单提交时,textarea 中文本换行,同时必须规定 cols 属性值 |

下面通过一个案例来演示<textarea>标签的使用方法。

textarea.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset = "utf - 8">
```

```

<title>多行文本框</title>
</head>
<body>
<textarea rows = "10" cols = "30">
我是一个多行文本框。
</textarea>
</body>
</html>

```

上述代码中,rows 属性用于设置可见行数,当文本内容超出这个值时将显示垂直滚动条;cols 属性用于设置一行可输入多少个字符。当然,rows 和 cols 属性都可以不设置,而通过 CSS 的 width 和 height 属性设置。

#### 4) <datalist>标签

<datalist>标签用于定义选项列表。与 input 元素配合使用,可以制作出输入值的下拉列表。当在 input 内输入时就会有自动完成(autocomplete)的功能,用户将会看见一个下拉列表供其选择。列表通过<datalist>标签嵌套<option>标签进行创建,下面通过一个案例来演示<datalist>标签的使用方法。

datalist.html:

```

<!DOCTYPE html>
<html>
<head>
<meta charset = "utf - 8">
<title>datalist 的使用</title>
</head>
<body>
<h3>浏览器版本: </h3>
<input list = "type"/>
<datalist id = "type">
    <option value = "Internet Explorer">
    <option value = "Firefox">
    <option value = "Chrome">
    <option value = "Safari">
    <option value = "Sogou">
    <option value = "Maxthon">
</datalist>
</body>
</html>

```

在浏览器中打开该页面,当鼠标指针悬停到文本框上时,右侧会出现一个下拉列表的按钮,单击其可显示列表进行选择,此时效果如图 5-11 所示。

#### 5) <output>标签

<output>标签用于定义不同类型的输出。该标签必须从属于某个表单。标签要么写在表单内部,要么对它添加 form 属性。一般用其显示计算的结果或脚本的其他结果。下面通过一个案例来演示<output>标签的使用方法,在页面上通过 number 输入类型输入两个加数,使用 output 标签显示两个数的和。



图 5-11 选项列表“显示列表进行选择时”效果

output.html:

```
<!DOCTYPE html >
<html >
<head >
<meta charset = "utf - 8">
<title> output </title>
</head>
<body>
<form oninput = "sum.value = parseInt(a.value) + parseInt(b.value)">
  <input type = "number" name = "a" value = "50">
  +
  <input type = "number" name = "b" value = "1">
  =
  <output name = "sum" for = "a b">
</form>
</body>
</html>
```

上述代码中,先添加两个输入框,并用< output >标签显示结果,当改变输入值时,在< output >标签上显示计算的结果,这需要添加 oninput 事件属性。当元素获得用户输入时运行此脚本进行计算并设置 output 的 value 值。在文本框中输入 51 和 1 后,结果 52 显示在“=”号之后,如图 5-12 所示。

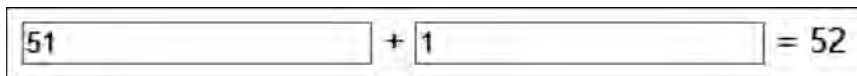


图 5-12 output.html 页面效果

## 2. HTML 表单验证

### 1) 表单验证

在页面表单中输入信息时,都避免不了输入错误的或无效的数据,如果这些错误信息直接发送到服务器端进行处理,就会浪费很多系统资源。可否在客户终端信息提交到服务器之前保证所录入的信息在形式上是正确的? 这样服务器端就只负责处理业务逻辑。



HTML5 中提供的表单验证功能,可以帮助解决这个问题。HTML5 自带表单验证功能有以下两种。

(1) 通过 required 属性校验输入框填写内容不能为空,如果为空将弹出提示框,并阻止表单提交。

(2) 通过 pattern 属性规定用于验证 input 域的模式,它接受一个正则表达式。表单提交时这个正则表达式会被用于验证表单内非空的值,如果控件的值不匹配,就会弹出提示框,并阻止这个正则表达式提交。注意:只有非空的表单才会使用正则验证,如果什么都不输入,pattern 不会被使用,所以还需要 required 协助。

2) 正则表达式

正则表达式,又称规则表达式,是计算机科学中的一个概念,通常被用来检索、替换那些符合某个模式(规则)的文本。形式上用事先定义好的一些特定字符及这些特定字符的组合,组成一个“规则字符串”,这个“规则字符串”用来表达对字符串的一种过滤逻辑。常用的正则表达式如表 5-7 所示。

表 5-7 常用的正则表达式

| 正则表达式                                                                              | 功 能                          |
|------------------------------------------------------------------------------------|------------------------------|
| <code>^[0-9]*\$</code>                                                             | 数字                           |
| <code>^\d{n}\$</code>                                                              | n 位的数字                       |
| <code>^\d{n,}\$</code>                                                             | 至少 n 位的数字                    |
| <code>^[A-Za-z]+\$</code>                                                          | 由 26 个英文字母组成的字符串             |
| <code>^[A-Za-z0-9]+\$</code>                                                       | 由数字和 26 个英文字母组成的字符串          |
| <code>^[a-zA-Z]\w{5,17}\$</code>                                                   | 由字母开头,长度为 6~18,只能包含字母、数字和下划线 |
| <code>^([0-9]){7,18}(x X)?\$</code>                                                | 身份证号码                        |
| <code>^(13[0-9] 14[5 7] 15[0 1 2 3 5 6 7 8 9] 18[0 1 2 3 5 6 7 8 9])\d{8}\$</code> | 手机号码                         |

下面通过一个案例来演示表单验证的方法。

form-validator.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title> Document</title>
</head>
<body>
<form name="register1" id="register1">
  <label for="username">申请人姓名:</label>
  <input id="username" name="username" type="text" placeholder="First and last name"
required autofocus/>
  <br/><br/>
  <label for="phone">手机号码:</label>
  <input id="phone" name="phone" type="text" pattern="\d{3}-\d{4}-\d{4}" placeholder =
"xxx-xxxx-xxxx" required/>
  <br/><br/>
```

```

< label for = "emailaddress">电子邮件:</label>
< input id = " emailaddress" name = " emailaddress" type = " email" placeholder = " For
confirmation only" required/>
< br/>< br/>
< label for = "userurl">个人网站:</label>
< input id = "userurl" name = "userurl" type = "url" required/>
< br/>< br/>
< input type = "submit" name = "register" value = "提交" onclick = " checkForm()" />
</form>
</body>
</html>

```

上述案例中,因为邮箱和 URL 都是 HTML5 内置的正则校验,所以会进行详细的提示。手机号码是自定义的正则表达式,所以只提示“请匹配要求的格式”(火狐浏览器)。

### 课堂小实践

运用已学习的 HTML5 验证表单方法,完善在线商城的用户注册表单的制作。

## 5.2.5 知识点检测



## 5.2.6 项目实施

通过项目分析和表单< label >标签、< textarea >标签,< datalist >标签及表单验证等知识点学习,大家定能逐步理解并完成“走进 Web 前端开发”网站“联系页”页面< linkus. html >“联系我们”表单设计的代码编写。参考代码如下。

HTML 部分:

```

< form class = "linkus_form" action = "# " method = "post" name = "linkus_form">
< ul >
< li >
< h2 >联系我们</h2>
< span class = "required_notification">* 表示必填项</span></li>
< li class = "normal">< span>姓名:</span>
< input type = "text" id = "name" name = "name" required/>
</li>
< li class = "normal">< span>所学专业</span>
< input type = "text" list = "major" placeholder = "软件工程">
< datalist id = "major">
< option value = "软件工程"></option>
< option value = "数字媒体技术"></option>
< option value = "计算机科学与技术"></option>
< option value = "网络工程"></option>
</datalist>
</li>
< li class = "normal">< span>电子邮件:</span>
< input type = "email" name = "email" placeholder = "chenjx5120@sina.com" required/>

```

```

        </li>
        <li class="normal"><span>手机号码:</span>
            <input id="phone" name="phone" type="text" pattern="^[1][3,4,5,7,8][0-9]{9}$" required/>
        </li>
        <li class="special"><span>感兴趣的前端技术:</span>
            <input type="checkbox" id="HTML5" name="interest"/>
            <label for="HTML5">HTML5</label>
            <input type="checkbox" id="CSS3" name="interest"/>
            <label for="CSS3">CSS3</label>
            <input type="checkbox" id="JavaScript" name="interest"/>
            <label for="JavaScript">JavaScript</label>
            <input type="checkbox" id="jQuery" name="interest"/>
            <label for="jQuery">jQuery</label>
        </li>
        <li class="normal"><span>学习历程简介:</span>
            <textarea name="process" placeholder="请输入学习历程"></textarea>
        </li>
        <li>
            <button class="submit" type="submit">立即联系</button>
        </li>
    </ul>
</form>

```

CSS 部分:

```

.linkus_form h2 {
    display: inline;
    margin: 0;
    font-family: "微软雅黑", "华文行楷";
}
.required_notification {
    color: #F30307;
    margin: 5px 0 0 0;
    display: inline;
    float: right;
    font-weight: bold;
}
/* 表单所有控件获得焦点时,不出现虚线框(或高亮框) */
*:focus {
    outline: none;
}
.linkus_form {
    width: 90%;
}
.linkus_form li {
    padding: 16px 12px;
    border-bottom: 1px solid #ddd;
    position: relative;
}
/* 给类名为 linkus_form 的元素的第一个子元素 li 和最后一个子元素 li 加底部边框 */
.linkus_form li:first-child, .linkus_form li:last-child {
    border-bottom: 1px solid #888;
}

```

```

    }
    .linkus_form span {
        width: 150px;
        margin: 3px 0 0 15px;
        display: inline-block; /* 把块元素强制转换为行内块元素 */
        padding: 3px;
        font-family: "微软雅黑", "华文行楷";
        font-size: 1.1em;
    }
    .normal input {
        height: 20px;
        width: 280px;
        padding: 8px 8px;
        background: #fff url(..images/attention.png) no-repeat 93% center;
    }
    .linkus_form textarea {
        padding: 8px;
        width: 320px;
        height: 280px;
    }
    .linkus_form button {
        margin-left: 156px;
    }
    .normal input, .normal textarea {
        border: 1px solid #bbb;
        box-shadow: 0px 0px 3px #ddd, 0 10px 15px #eee inset;
        border-radius: 4px;
        transition: padding .35s;
    }

    /* 当前元素获得焦点时,设置背景颜色和背景图片、边框、外阴影和右内边距 */
    .normal input:focus, .normal textarea:focus {
        background: #fff;
        border: 1px solid #555;
        box-shadow: 0 0 3px #aaa;
        padding-right: 70px;
    }

    /* 当该元素获得焦点填写内容无效时,设置背景颜色、背景图片、盒阴影及边框颜色 */
    .normal input:focus:invalid, .normal textarea:focus:invalid {
        background: #fff url(..images/warn.png) no-repeat 93% center;
        box-shadow: 0 0 8px red;
        border-color: orangered;
    }

    /* 当该元素获取有效的填写内容时,设置背景颜色、背景图片、盒阴影及边框颜色 */
    .normal input:required:valid, .normal textarea:required:valid {
        background: #fff url(..images/right.png) no-repeat 98% center;
        box-shadow: 0 0 8px green;
        border-color: limegreen;
    }
    button.submit {
        padding: 2px 50px;
        line-height: 2em;
        border: 1px solid #cccccc;
        border-radius: 3px;
    }

```

```

background: -webkit-gradient(linear, left top, left 28, from(#EF5356), color-stop
(0%, #FF0000), to(#EF5356));
background: -moz-linear-gradient(top, #EF5356, #FF0000 0, #EF5356 28px);
background: -o-linear-gradient(top, #EF5356, #FF0000 0, #EF5356 28px);
height: 35px;
cursor: pointer;
letter-spacing: 6px;
margin-left: 250px;
color: #FFF;
font-weight: bold;
font-size: 14px;
}
/* 当鼠标指针悬停在“提交”按钮上时,该按钮背景颜色透明度为 85 % */
button.submit:hover {
    opacity: .85;
}

```

### 5.2.7 项目总结

项目实施过程中,主要是练习<form>表单元素的使用及验证方法;要特别注意表单元素获得焦点填写内容无效时样式的变化;搜集、分析、归纳联系页表单的多种设计方法,培养批判性的思维能力、缜密的逻辑推理能力、团结协作能力,进一步提升自身的职业素养。

### 5.2.8 能力拓展

(1) 小组协作搜集、分析、归纳注册、登录、联系页、问卷调查、在线订购、留言本等多种表单的设计方法及验证方式。

(2) 模拟设计如图 5-13 所示的注册表单。

图 5-13 注册表单效果图