

第3章 HTML 5 中的表单应用

表单是客户端和服务端传递数据的桥梁,是实现用户与服务器互动的最主要方式。一般情况下,网站管理者要实现与浏览者之间的沟通,就必须借助表单这个桥梁。表单通常的应用包括用户注册、登录、订单提交和搜索界面等。

HTML 5 中对传统表单的功能进行了改进,添加了许多与应用程序有关的功能,例如新增了表单类型、表单元素,以及输入框的必填属性等。本章将详细介绍 HTML 5 中的表单应用,包括新增的表单元素、表单类型、表单属性和表单验证 4 部分。

本章学习目标:

- 熟悉 HTML 表单的基本语法和常用属性
- 掌握 datalist 元素和 output 元素的使用
- 熟悉 keygen 元素和 optgroup 元素的使用
- 掌握 HTML 5 新增的表单类型
- 掌握 form 属性的简单使用
- 熟悉 HTML 5 中新增的与表单有关的属性
- 掌握 placeholder 属性、required 和 pattern 属性的使用
- 熟悉 autocomplete 和 autofocus 属性
- 了解 HTML 5 中的表单验证

3.1 认识和创建 HTML 表单

表单是网页中最常见的控件(元素)之一,小到网站注册登录,大到一个企业的数据管理系统,都基本上有表单的身影。表单之所以如此重要,主要是因为它担负大量的用户和网页后台数据更新交互的任务。表单通常都是与 input 元素的输入类型一块使用的,完成后将表单的内容进行提交,本章将回顾一下表单的基础知识。

3.1.1 认识 HTML 表单

HTML 中使用 form 元素表示表单,它可以看作是一个容器,用于包含其他的元素,可以设置表单的起始位置,并且指定处理表单的服务器端程序地址。表单拥有非常强大的功能,使用表单可以指定以下内容。

- (1) 表单的布局。
- (2) 指定处理表单的服务器端程序地址,该服务器端程序应该可以处理“名称/值”对。

- (3) 数据传送到服务器的方式。
- (4) 表单数据的字符编码。
- (5) 除了包含表单有关的元素外，form 元素还可以包含其他的文本元素，如段落、列表和层等。

如下代码演示了表单的基本语法形式。

```
<form action="URL" dir="" enctype="" id="" lang="" method="" name=""
onclick="" ondblclick=""
onreset="" onsubmit="" target="" ...>
...
</form>
```

form 元素拥有多个属性，如表 3-1 所示列举了上面语法中比较重要的几个属性。

表 3-1 form 元素的常用属性

属性名称	说明
action	用于设置处理表单数据程序的地址。如 Inex.html
method	用来定义数据传送到服务器的方式。常用的方式有 GET 或 POST
name	定义表单的名称
target	指定输入数据结果显示在哪个窗口，某些情况下需要与 frame 元素配合使用
enctype	设置表单数据的内容类型。例如，在上传文件时设置属性值为 multipart/form-data

通常情况下，提交表单之前需要对表单中所输入的内容进行验证，验证用户输入的内容是否合法。对表单验证常常借助于 JavaScript 或 JQuery 等工具，在操作之前，必须先获取对 form 元素的引用。例如，在 JavaScript 脚本函数中，有以下三种方式完成这一操作。

- (1) 使用 DOM 树中的定位元素进行获取。代码如下：

```
var oForm=document.getElementById("form1");
```

- (2) 用 document 的 form 集合中的位置获取。代码如下：

```
var oForm=document.forms[0];
```

- (3) 用表单中的 name 属性来获取。代码如下：

```
var oForm=document.forms["form1"];
```

3.1.2 input 元素的基本类型

表单是一个强大的控件集，用户与它的交互都是通过控件来进行的，HTML 静态网页中常说的控件就是指元素。form 表单中最常使用的就是 input 元素，设计人员向页面中添加该元素，并且为该元素指定不同的类型，然后在显示页面中向用户提供输入内容。

input 元素最重要的一个属性是 type，该属性决定了 input 元素的类型是什么，如输入框、重置按钮、单选按钮以及图片等。type 的属性值有多个，如表 3-2 所示。


```

        <td><input type="checkbox" checked="" name="expire" id="expire"
        value="7">一周内自动登录</label></td>
    </tr>
</tbody>
</table>

```

(3) 向 form 表单中添加第二个 table 表格，该表格记录了【登录】按钮以及其他信息。其他信息有注册或其他账号登录等，其中，将【登录】按钮的 input 元素的 type 属性的值设置为 submit。代码如下。

```

<table width="100%" border="0" align="center" cellpadding="5" cellspacing="0">
    <tbody>
        <tr>
            <td align="right" width="50">&nbsp;   </td>
            <td height="40" valign="top">
                <input type="submit" name="submitlogin" id="login" class="
                but80" value="登录">
                <a href="#">忘记密码? </a>
            </td>
        </tr>
        <!-- 其他信息 -->
    </tbody>
</table>

```

(4) 为表单中的相关元素添加 CSS 样式，具体样式代码不再列出。

(5) 运行页面查看效果，分别向【账号】输入框和【密码】框中输入内容，如图 3-1 所示。

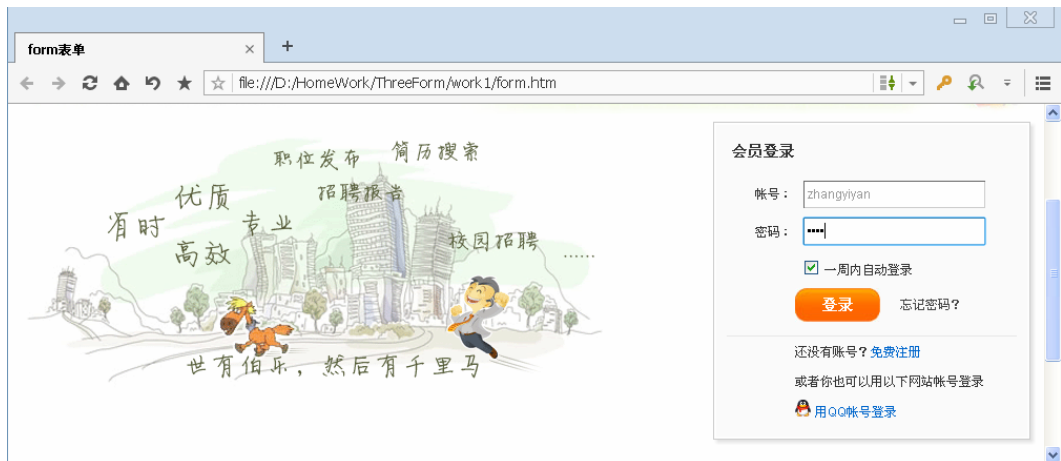


图 3-1 form 表单的使用

在用户输入内容完成后，并且提交数据之前，还需要判断输入内容的合法性。这些输入内容都包含在表单中，可以使用它们的 name 特性或它们在集合中的位置来访问这

些内容，几种常用的方式如下所示，其中，oForm 表示获取的表单的 name 属性的值。

(1) 根据在 form 集合中的位置进行获取。代码如下。

```
var oFirstField=oForm.elements[0];
```

(2) 根据在 form 集合中的 name 属性进行获取。代码如下。

```
var oTextBox1=oForm.elements["textbox1"];
```

(3) 直接通过元素的名称进行访问。代码如下。

```
var oTextBox2=oForm.textbox2;
```

除了与 form 集合绑定外，还有一种方式是根据元素的 id 属性的值进行获取（即使用 DOM 树中的定位元素进行获取）。代码如下。

```
var val = document.getElementById("textbox1");
```

3.1.3 其他输入元素

表单除了常用的 input 元素外，还可能会使用到其他的元素，如段落、层和表格等，除了这些元素外，常用的输入元素还有下拉菜单和列表框等。

1. select 元素

select 元素可以用于创建下拉菜单和列表框，它至少包含一个 option 元素。一般它将包含两个或者两个以上的 option 元素，因为下拉菜单和列表框的每个选项都需要一个 option 元素来呈现。

默认情况下，select 元素显示为下拉列表框，用户可以通过下拉框选择出生地、居住地和当前分类等。该元素有一个 multiple 属性，如果将该属性的值设置为 true 或直接指定为 multiple，那么则会以列表框的形式显示。

2. textarea 元素

通过指定 input 元素的 type 属性的值为 text 时表示单行输入，那只能够在同一行中输入内容。有时候，单行输入并不能满足用户的需要，例如用户在对商品描述时输入的内容比较多，这时可以使用多行文本框，HTML 中使用 textarea 元素，它也叫文本区域。

textarea 元素接收访问者输入多于一行的文本，可以同时呈现多行数据。格式如下：

```
<textarea rows="宽度" cols="长度"></textarea>
```

3.2 新增的表单元素

HTML 5 对表单功能的改进包含多个方面，如对表单元素的增加、对 input 元素类型的增加，以及表单属性的增加等，下面将介绍新增加的几个表单元素。

3.2.1 datalist 元素

`datalist` 元素定义输入框的选项列表，选项列表是通过该元素内的 `option` 元素进行创建的。该元素需要与 `input` 元素配合使用，来定义 `input` 可能的值。如果只在页面中定义 `datalist` 的列表，在浏览页面时选项不会被显示出，它仅仅是合法的输入值列表。

如果要把 `datalist` 元素定义的选项列表绑定到输入框，需要将输入框的 `list` 属性的值设置为 `datalist` 元素的 `id` 属性的值。

【练习 2】

细心的用户可以发现，用户在【账户】输入框中输入内容时，将鼠标定位到该输入框时就会显示一些用户名称列表。本次练习通过 `datalist` 元素实现选项列表，步骤如下。

(1) 在网页中添加 `datalist` 元素，并通过 `option` 向 `datalist` 内添加选项列表。代码如下。

```
<datalist id="usernameList" name="usernameList">
  <option>lovesmyself</option>
  <option>administrator</option>
  <option>mengmeng1990</option>
  <option>huitiaowudetuzi</option>
</datalist>
```

(2) 为【账户】输入框的 `input` 元素添加 `list` 属性，指定该属性的值为 `datalist` 元素的 `id` 的属性值。代码如下。

```
<input name="username" type="text" class="login_input" id="username"
  maxlength="30" value="用户名/邮箱/手机号" style="color:#999999" list=
  "usernameList">
```

(3) 重新运行页面查看效果，如图 3-2 所示。



图 3-2 datalist 元素的使用

3.2.2 output 元素

`output` 元素用于不同类型的输出，例如，计算结果或者脚本的其他结果。`output` 元素必须属于某个表单，或者通过属性指定某个表单。简单来说，`output` 元素必须将它书写在表单内部，或对它添加 `form` 属性。

`output` 元素与 `input` 元素联合使用时，通常还会与 JavaScript 事件进行结合，这样可以十分方便地查到表单中各个元素的输入内容。

【练习 3】

随着社会的不断发展，越来越多的人开始注重个人的修养和保养问题，尤其是女孩子，她们对自己的体重要求特别严格，那么究竟什么样的范围属于合理体重呢？本次练习根据用户输入的身高和体重进行计算，将计算的结果显示到页面，并且向用户反馈信息。

(1) 添加新的页面并且进行设计，在页面中添加表单，向表单中添加用于用户输入的身高和体重输入框，并且添加执行操作的按钮。代码如下。

```
<form method="post" name="resultform">
  <table cellspacing="0" cellpadding="2" border="0" width="250" style="float:left;">
    <tbody>
      <tr>
        <td>
          你的身高: <input size="6" name="sg" class="smin">厘米
          (cm)<br>
          你的体重: <input size="6" name="tz" class="smin">千克
          (kg)
        </td>
      </tr>
      <tr><td><input type="button" name="btn2" class="testBtn">
      </td></tr>
    </tbody>
  </table>
</form>
```

(2) 继续向表单中添加内容，在表单的合适位置添加一个 `output` 元素，该元素向页面输入计算的结果。代码如下。

```
<output name="result"></output>
```

(3) 为表单中的按钮控件添加 `onClick` 事件属性，当用户单击该按钮时触发其 `Click` 事件调用 `JiSuanResult` 脚本函数。页面按钮的代码如下。

```
<input type="button" name="btn2" class="testBtn" onClick="JiSuanResult()">
```

(4) 向页面中添加 `JiSuanResult()` 函数，在该函数中获取用户在页面中输入的内容，

将内容处理完成后,再根据公式进行计算。它的计算公式为:体重(kg)除以身高(m)的平方($BMI = kg/m^2$)。JiSuanResult()函数的代码如下。

```
function JiSuanResult(){
    var forms = document.forms["resultform"]; //获取页面中的表单
    var first = forms['sg'].value; //获取用户输入的身高
    var second = forms['tz'].value; //获取用户输入的体重
    var firstHigh = first/10; //将身高的单位转换为米(m)
    var result = second/(firstHigh*firstHigh); //计算结果
    result = result*100;
    forms['result'].value = result; //为 output 元素赋值
    if(result>=21<=25){ //根据计算的结果进行判断
        alert("恭喜,您的体重在正常范围之内。");
    }else if(result<21){
        alert("您的体重偏轻,要注意营养啊!");
    }else{
        alert("不能再吃了,一定要减肥!!");
    }
}
```

(5) 运行页面查看效果,向网页中输入内容后单击【点击测试】按钮进行测试,效果如图 3-3 所示。



图 3-3 output 元素的使用

3.2.3 keygen 元素

keygen 元素是 HTML 5 中新增的表单元素,它用于表单的密钥生成器字段,作用是提供一种验证用户的可靠方法。keygen 元素在跟随表单提交时会生成两个密钥:私密钥(Private Key)和公密钥(Public Key)。其中,私密钥存储在本地;公密钥发送到服务器

端，可以用于之后验证用户客户端证书。

keygen 元素中提供了多个属性，常用属性如表 3-3 所示。

表 3-3 keygen 元素的常用属性

属性名称	说明
challenge	如果使用该属性，将 keygen 的值设置为在提交时会给出提示
disabled	禁用 keygen 字段
form	定义该 keygen 字段所属的一个或多个表单
keytype	定义生成密钥的算法，默认值为 rsa
name	定义 keygen 元素的唯一名称。用于在提交表单时获取字段的值

【练习 4】

本次练习简单演示 keygen 元素的使用，相关人员在页面中添加 keygen 元素后，运行页面时会自动添加下拉框列表，该列表有两个选项。页面代码如下。

```
<keygen name="secutity" width="300" />
```

运行页面查看 keygen 元素生成的效果，如图 3-4 所示。



图 3-4 keygen 元素的使用

3.2.4 optgroup 元素

optgroup 元素在 HTML 4 中已经存在，HTML 5 中对其进行了扩展。该元素定义选项组，使用该元素允许用户组合选项。当使用一个长的选项列表时，对相关的选项进行组合会使处理更加容易。

使用 optgroup 元素可以将 select 元素中的选项进行逻辑分组，optgroup 元素必须在 select 元素中直接定义，而且不允许嵌套，这就意味着不能进行二级分组。optgroup 元素有两个常用的属性，它们分别是 disabled 属性和 label 属性。其中，前者表示首次加载时是否禁用该选项组，而后者则用来定义分组显示的标签名，但是标签名并不能被当作选

择项。

【练习 5】

读者查看一些网站时会经常看到下拉框列表选项，选择某个选项可以查询不同的内容。但是，有时列表中还会添加子列表，本节就通过 `optgroup` 元素添加组合选项。

(1) 在页面中首先添加 `select` 元素，它用于显示图书的类型。

(2) 向 `select` 元素中加上三个 `optgroup` 元素，分别设置这三个 `optgroup` 元素的 `label` 属性的值，再向该元素中添加 `option` 元素。部分代码如下。

```
<select name="url" id="searchDropDownBox" class="searchSelect" style="
"top: 0px;">
  <optgroup label="按类型">
    <option>音乐</option>
    <option>电器</option>
    <!-- others -->
  </optgroup>
  <optgroup label="按价格">
    <option>0-50</option>
    <option>51-80</option>
    <!-- others -->
  </optgroup>
  <optgroup label="按作者国籍">
    <option>中国</option>
    <option>美国</option>
    <option>英国</option>
  </optgroup>
</select>
```

(3) 运行页面查看效果，单击下拉框查看列表，效果如图 3-5 所示。



图 3-5 `optgroup` 元素的使用

试一试

前面已经提到过，将 select 元素的 multiple 属性的值直接设置为 multiple 或 true，在页面中会显示为列表框。读者可以亲自动手试一试，设置 select 元素的 multiple 值，然后再向列表框中添加组合选项。

3.3 新增的表单类型

HTML 5 兼容 HTML 4 的大部分内容，同时也对 HTML 4 中的内容进行了改进。在 HTML 5 中，HTML 5 的表单定义了一些新的表单类型，这些表单类型在一定程度上方便了程序员的开发。

3.3.1 email 类型

验证用户在文本框中输入的内容是否是邮箱有多种方法，例如，可以通过 JavaScript 脚本进行判断，也可以通过 JQuery 进行判断。HTML 5 中新增加了 email 类型，该类型是 input 元素专门用来输入 E-mail 地址（例如，QQ 邮箱、网易邮箱）的文本框。基本语法如下：

```
<input type="email" name="email"></input>
```

直接将 input 元素的 type 类型指定为 email，该类型会自动要求输入的地址格式正确，否则浏览器不允许提交，并且会有一个错误信息提示。

【练习 6】

用户在一些招聘网站上注册时都会要求填写正确的邮箱地址，这样可以方便企业向邮箱中投递信息，下面将输入邮箱地址的文本框的 type 类型指定为 email。代码如下。

```
<input name="user_mail" type="email" id="user_mail" style="background-color:#EBF8FF; border-color:#ABDAFE; border-width:1px; border-style:solid; height:21px; width:214px;">
```

目前，并不是所有的浏览器全部都兼容 email 类型，不同的浏览器内核，可能导致所显示的效果不太一样。如图 3-6 所示为在 Maxthon 浏览器中的效果，如图 3-7 所示为在 Opera 浏览器中的效果。

注意

email 类型可以检测用户输入的 E-mail 地址是否合法，但是它的有效性并不会判断输入框的内容是否为空，而是针对非空的内容进行格式检测。简单来说，如果要检测输入框中输入的地址是否合法，必须确保输入框的内容不为空。

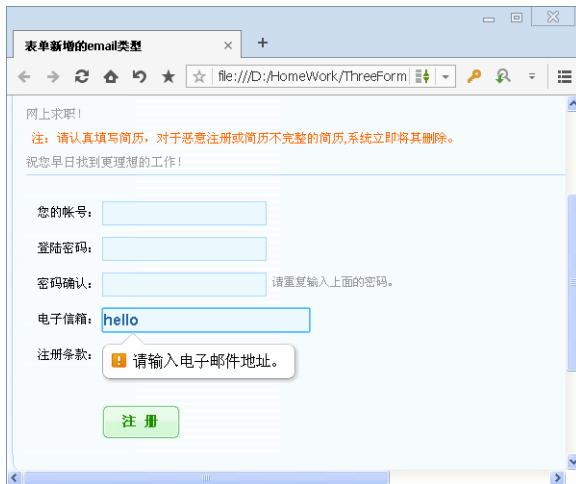


图 3-6 Maxthon 浏览器显示效果

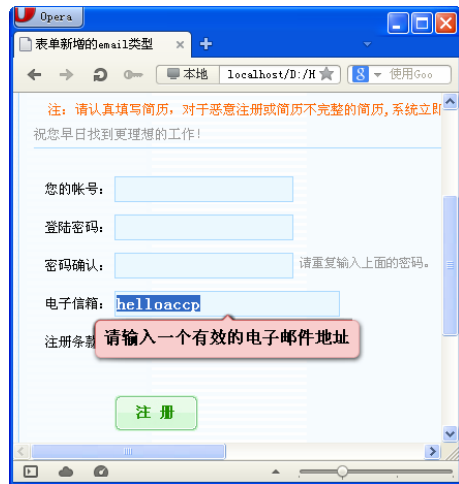


图 3-7 Opera 浏览器显示效果

email 类型的文本框具有一个 `multiple` 属性，该属性允许在文本框中输入使用逗号分隔的有效的 E-mail 地址的一个列表。例如，重新更改练习 6 中的代码，为 email 类型的 `input` 元素添加 `multiple` 属性，将此属性的值指定为 `true`。代码如下。

```
<input name="user_mail" type="email" multiple="true" id="user_mail" style="background-color:#EBF8FF; border-color:#ABDAFE; border-width:1px; border-style:solid;height:21px; width:214px;">
```

重新在浏览器中运行查看效果，在 Chrome 浏览器中的效果如图 3-8 所示，在 Firefox 浏览器中的效果如图 3-9 所示。

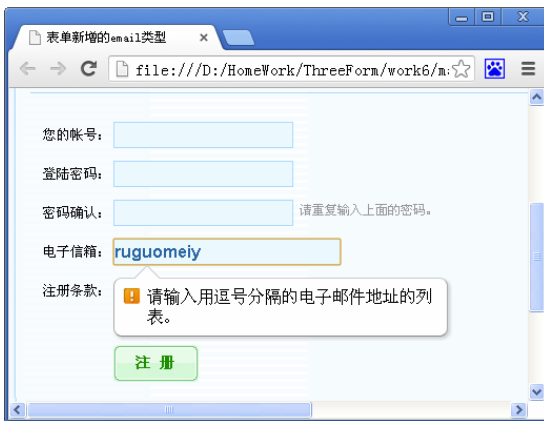


图 3-8 Chrome 浏览器运行效果

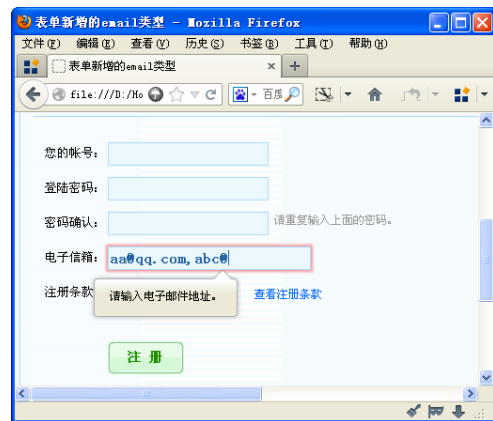


图 3-9 Firefox 浏览器运行效果



提示

用户可以手动使用逗号分隔邮箱地址列表，但是，通常情况下，浏览器可能使用复选框从用户邮件客户端或者手机通讯录中很好地取出用户的联络人列表。

3.3.2 url 类型

url 类型的 input 元素是一种专门用来输入 url 地址的输入框,例如百度地址和谷歌地址等。如果某个浏览器不支持该类型,则浏览器会自动将其识别为普通输入框。与 email 类型一样, url 类型的检测并不会判断输入框是否为空。因此,如果要检测 url 地址是否正确,必须确保输入框中的内容不为空。

【练习 7】

经常用微博的用户可以知道,更改个人资料或发表博客时通常会需要用户的个人博客地址。例如,在腾讯微博的用户个人资料页面输入用户个人主页。相关代码如下。

```
<input id="homepage" name="homepage" type="url" class="inputTxt en">
```

运行页面并在不同的浏览器中查看效果,在 Maxthon 或 Chrome 浏览器中必须输入完整的 URL 地址,它会自动忽略前面的空格。如 Maxthon 浏览器中输入地址不完整时的效果如图 3-10 所示。Opera 浏览器不需要输入完整的路径,它会自动在开始处添加“http://”,但是对空格非常敏感。如果 URL 地址文本框中存在空格,则会提示错误,如图 3-11 所示。

78

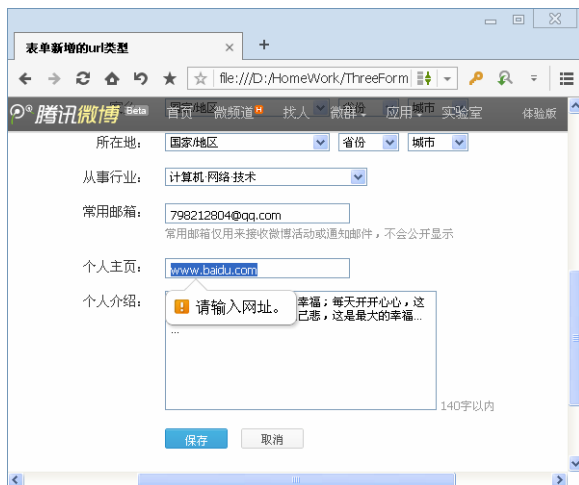


图 3-10 Maxthon 浏览器运行效果

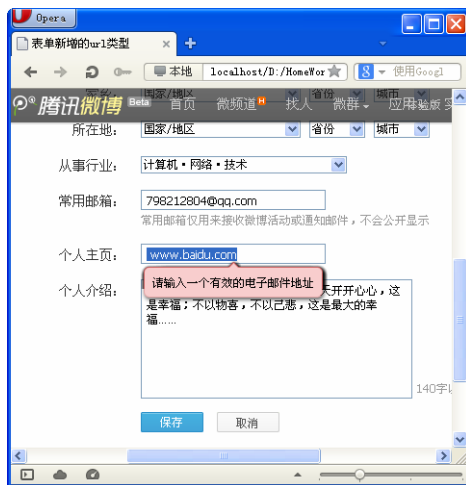


图 3-11 Opera 浏览器效果

3.3.3 number 类型

number 类型的 input 元素是一种专门用来输入数字的文本框,在表单提交时会检查其中的内容是否为数字。如果浏览器不支持该类型时,会自动显示为一个文本框, number 类型和一些常用的属性结合使用,如表 3-4 所示。

表 3-4 number 类型的常用属性

属性名称	说明
max	指定输入框可以接受的最大的输入值
min	指定输入框可以接受的最小的输入值
step	输入域合法的间隔, 如果不设置, 默认值是 1
value	指定默认显示的数值

【练习 8】

本次练习主要演示用户的当前年龄显示, 将用户输入的文本框的类型设置为 number, 然后设置最大值和最小值等属性。代码如下。

```
<input id="age" name="age" type="number" class="inputTxt en" max="120"
min="15" value="23">
```

运行网页查看在不同浏览器中的效果, 如果输入的内容不是数字, 显示错误提交, 在 Chrome 浏览器中的效果如图 3-12 所示。如果输入的内容不在某个范围内也会显示错误, 在 Opera 浏览器中的效果如图 3-13 所示。



图 3-12 输入的内容不是数字时的提示

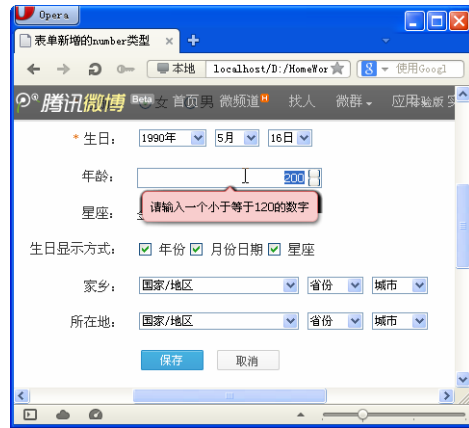


图 3-13 输入的内容超出范围

3.3.4 range 类型

range 类型的 input 元素是一种只允许输入一段范围内数值的文本框, 它与 number 类型类似, 都具有 max、min、step 和 value 属性。但 number 类型和 range 类型的显示方式有所不同: 该类型在页面的输入框为微调格式显示, 而 range 类型以滑动条的形式展示数字, 通过拖动滑块实现数字的改变。

【练习 9】

本次练习向页面显示一个滑动条, 然后通过拖动滑动条更改其值, 达到设置元素的透明度, 实现步骤如下。

(1) 在页面中通过滑动条控制透明度, 且为 input 元素添加 onClick 事件属性。代码

如下。

```
<input type="range" value="1" max="1" min="0" step="0.1" onClick="Change
Opacity(this.value)" />
```

(2) 当用户单击更改滑动条的值时会触发 Click 事件调用 JavaScript 脚本中的 ChangeOpacity()函数, 该函数中的代码用于更改指定元素的透明度。代码如下。

```
function ChangeOpacity(val) {
    document.getElementById("firstContent").style.opacity = val;
}
```

(3) 在浏览器中拖动滑动条查看效果, 如图 3-14 所示。



图 3-14 range 类型的显示效果

3.3.5 tel 类型

tel 类型的 input 元素是一种用来输入电话号码的专用文本框。它没有特殊的校验规则, 甚至不强调只输入数字, 因为很多电话号码常常含有数字以外的字符, 如 410-331245**。

tel 类型的文本框通常用来验证固定电话和手机号码, 如果只将 input 元素的类型设置为 tel 类型, 并不会达到验证电话的效果, 它必须和 pattern 属性 (3.4.5 节详细介绍) 一块使用。

【练习 10】

本次练习将 tel 类型和 pattern 属性结合使用, 验证用户输入的固定电话是否是“010-12345678”或“0371-1234567”的格式。在页面中添加相关代码, 如下所示。

```
<input id="oldtel" name="oldtel" type="tel" class="inputText en" pattern=
"^\d{3}-\d{8}|\d{4}-\d{7}$">
```

运行网页查看效果, 如图 3-15 和图 3-16 所示分别为在 Chrome 浏览器和 Opera 浏览器中的验证效果。



图 3-15 Chrome 浏览器运行效果

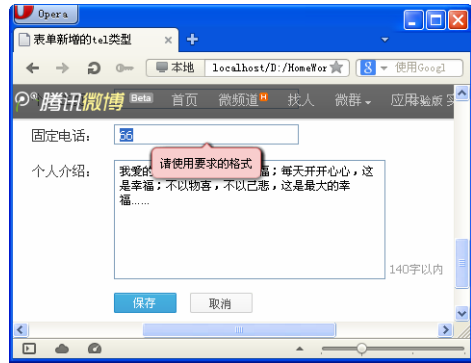


图 3-16 Opera 浏览器运行效果

3.3.6 color 类型

将 `input` 元素设置为 `color` 类型就是用来选取颜色的，它提供了一个颜色选取器用来选取。如果浏览器不支持 `color` 类型，则会自动显示为普通的文本框。

【练习 11】

用户通过 `color` 类型选择的颜色值将保存到它的 `value` 属性中，本次练习就简单演示了该类型的使用。用户通过选中的值，更改指定内容的背景颜色，实现的具体步骤如下所示。

(1) 添加新页面并且进行设计，在页面的合适位置添加用于加载显示颜色的 `color` 类型的 `input` 元素和执行更改颜色的 `button` 类型的 `input` 元素，并且为按钮添加 `onClick` 事件属性。代码如下。

```
<input type="color" name="chgcolor" id="chgcolor" width="200px;" />
<input type="button" onClick="ChangeColor()" value=" Change " />
```

(2) 用户单击页面中的按钮时调用 `ChangeColor()` 函数更改背景颜色，在该函数中首先获取用户设置的颜色的值，然后再进行更改。代码如下。

```
function ChangeColor(){
    var colorval = document.getElementById("chgcolor").value;
    //获取设置的色
    document.getElementById("firstContent").style.backgroundColor = colorval;
    //设置背景色
}
```

(3) 运行页面单击颜色按钮进行测试，如果是 Maxthon 浏览器，直接单击颜色选项框弹出【颜色】对话框。用户可以根据系统提供的颜色进行选择，也可以自定义颜色样式，然后单击对话框中的【添加到自定义颜色】按钮将设置好的颜色添加到自定义颜色列表中，如图 3-17 所示。Opera 浏览器中首先显示一些系统列表，然后单击【其他】按钮弹出【颜色】对话框，如图 3-18 所示。



图 3-17 Maxthon 浏览器效果



图 3-18 Opera 浏览器效果

3.3.7 search 类型

search 类型是一种专门用来输入搜索关键词的文本框，比如站点搜索或 Google 搜索等。search 类型的文本框与普通文本框没有太大的区别，最大的不同在于：在部分浏览器中，会自动把 search 类型的文本框使用圆角边框，当用户开始输入时，输入框的右边会出现一个用于清除内容的图标，单击这个图标可以快速清除内容。

【练习 12】

几乎所有的网站和系统都离不开搜索的功能，本次练习简单使用 search 类型。首先在创建的静态页面中添加搜索框，并且将其类型设置为 search。相关代码如下。

```
<input class="search_text" id="query" name="query" style="margin-top:2px; margin-left: 10px;width: 110px;" type="search">
```

运行页面查看效果，向页面中输入内容进行测试，如图 3-19 所示。

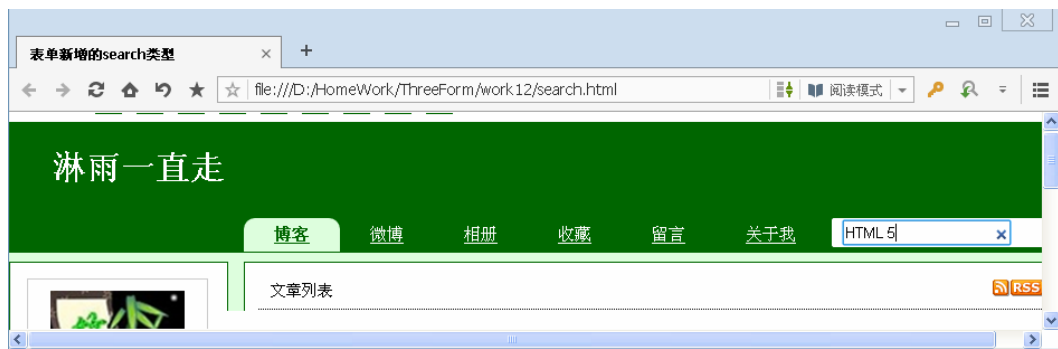


图 3-19 search 类型的使用

3.3.8 日期和时间类型

在传统的网页设计中，相关人员只能通过 JavaScript 或 JQuery 等其他方法实现日期

的选择，它们虽然有封闭好的代码或控件，但是使用起来还是比较麻烦，没有直接显示选择日期的文本框。

HTML 5 中新增加了与日期有关的类型，这些类型不用任何脚本就可以实现日期的选择。HTML 5 新增的日期类型有 6 个，它们分别是 date、time、datetime、datetime-local、month 和 week。

1. date 类型

date 类型是目前比较受欢迎的一种元素，该类型以日历的形式方便用户输入。一般情况下，用户生日、购买日期和订票日期等都可以指定为 date 类型。

【练习 13】

随着社会的发展和水平的提高，越来越多的公司开始向海外发展业务，飞机成为这些地方快速转换的交通工具。在本次练习中，通过 date 类型指定从一个城市到另一个城市所出发的航班信息。代码如下。

```
<input type="date" style="ime-mode: disabled;" class="input06" name="dtGoDate" id="txtAirplaneTime1">
```

不同浏览器所显示的效果可能不同，如图 3-20 和如图 3-21 所示分别为在 Maxthon 浏览器和 Chrome 浏览器中的效果。

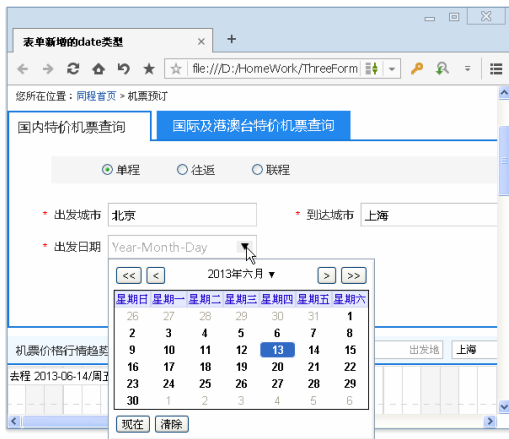


图 3-20 Maxthon 浏览器运行效果



图 3-21 Chrome 浏览器运行效果

2. time 类型

time 类型的 input 元素是一种专门用来输入时间的文本框，并且在提交时会检查输入时间的有效性。

【练习 14】

本次练习在练习 13 的基础上进行扩展，为机票查询的内容添加具体时间。代码如下：

```
<input type="time" id="txtAirplaneTime2" name="dtBackDate" class="input06" holder="小时/分钟">
```

与其他类型一样，支持 time 类型的浏览器不同，其所显示的外观也会有所不同。显示时可能会显示简单的文本框，只是在提交时检查是否输入有效的的时间；也可以以时钟形式出现；还可以显示时区。如图 3-22 和图 3-23 所示分别为 time 类型在 Maxthon 浏览器和 Chrome 浏览器中的显示效果。

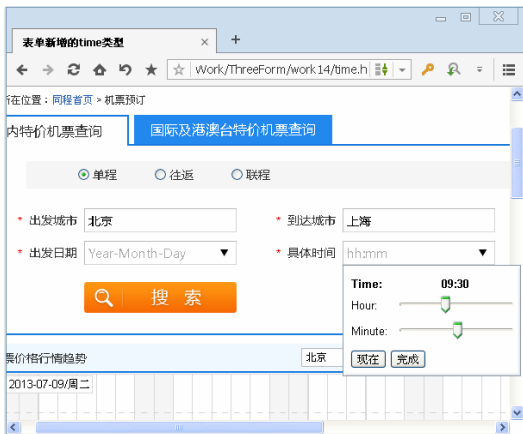


图 3-22 Maxthon 浏览器运行效果



图 3-23 Chrome 浏览器运行效果

3. datetime 类型

datetime 类型的 input 元素是一种专门用来输入 UTC 日期和时间的文本框，它不仅显示年、月、日，还可以显示具体时间。设置 datetime 类型完成后，在提交时会用户对用户输入或选择的日期和时间进行有效性检查。

【练习 15】

用户可以查询在某个时期内所有的车辆售票情况，在页面中添加类型是 datetime 的 input 元素。代码如下。

出发日期和时间：

```
<input type="datetime" name="train_startdate" value="" id="startdatepicker" style="width: 150px;" />
```

到达日期和时间：

```
<input type="datetime" name="train_enddate" id="enddatepicker" style="width: 150px;" />
```

在不同的浏览器中查看页面的运行效果，如图 3-24 和图 3-25 所示分别为在 Maxthon 浏览器和 Opera 浏览器中的效果。

4. datetime-local 类型

datetime-local 类型的 input 元素是一种专门用来输入本地日期和时间的文本框，并且在提交时会输入对输入的日期和时间进行有效性检查。

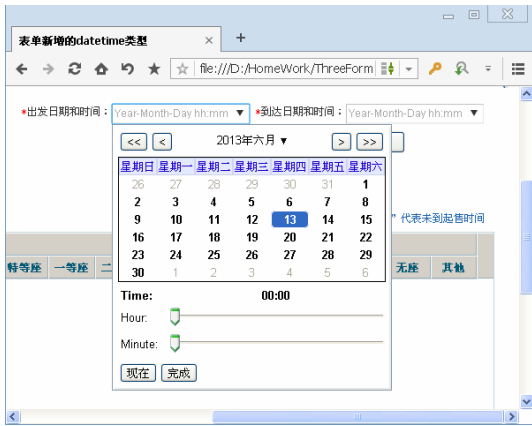


图 3-24 Maxthon 浏览器的效果

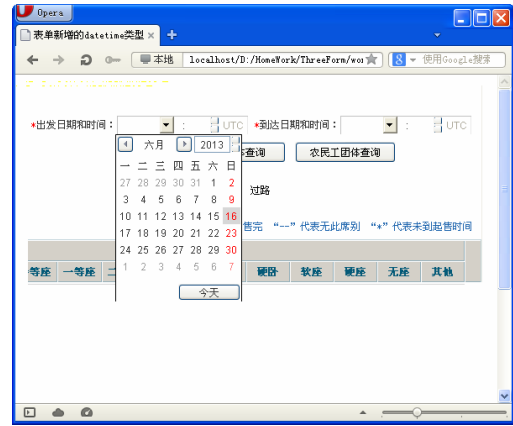


图 3-25 Opera 浏览器的效果

【练习 16】

继续利用前几个练习的页面进行更改，设置 datetime-local 类型的 input 元素效果。代码如下。

```
<input type="datetime-local" name="startdate" id="startdate" style="width: 150px;" class="input_20txt">
<input type="datetime-local" name="enddate" id="enddate" style="width: 150px;" class="input_20txt">
```

在 Maxthon 浏览器中，datetime 和 datetime-local 两种类型的 input 元素的效果基本上没有区别，如图 3-26 和图 3-27 所示分别为 Chrome 和 Opera 浏览器中的显示效果。

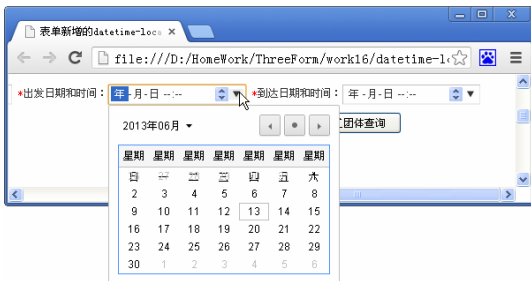


图 3-26 Chrome 浏览器效果

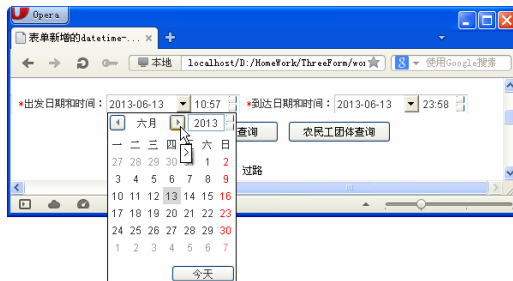


图 3-27 Opera 浏览器效果

5. month 类型

month 类型的 input 元素是一种专门用来输入月份的文本框，并且在提交时会输入月份的有效期进行检查。

month 类型的使用非常简单，代码如下。

```
<input name="monthSel" type="month" value="2013-12" />
```

6. week 类型

week 类型的 input 元素是一种专门用来输入周的文本框，并且在提交时会检查输入周的有效性。它可能是一个简单的输入文本框，允许用户输入一个数字；也可能是更复杂、更精确的内容。

例如，2013-W10 表示 2013 年第 10 周。如下代码直接定义了一个 week 类型的输入框。

```
<input name="weekSel" type="week" value="2013-W12" />
```

3.4 新增的表单属性

HTML 5 增加了许多与表单有关的属性，例如，与 number 和 range 类型的 input 元素结合使用的 max、min 和 step 属性，与 form 有关的 form 属性、formaction 和 formmethod 属性等，与 datalist 元素结合使用的 list 属性，以及与 tel 元素结合使用的 pattern 属性等。本节将对一些比较重要的、常用的属性进行介绍。

3.4.1 form 属性

HTML 4 以及之前的版本中，必须把相关的元素都放在 form 元素表单下才能顺利地提交到服务器。但是，在实际情况中，由于页面设计与实现的特殊性，会存在一些表单之外的元素也需要一并提交的情况，这时就需要使用 HTML 5 中新增的 form 属性。

form 属性规定输入框所属的一个或多个表单，它适用于所有 input 元素的类型。通过使用 form 属性，可以彻底打破必须将元素写入<form></form>标记之间的限制。

【练习 17】

form 属性可以让 HTML 元素孤立于表单之外，但是在提交表单时，不仅可以提交表单内部的元素，也可以将表单外的元素一起提交。本次练习在表单外添加一个普通的输入框，然后通过设置 form 属性获取内容，实现步骤如下。

(1) 添加新的页面并且设计页面，在页面中添加表单，该表单内包含用户名、用于输入的验证码、内容和按钮等。相关代码如下。

```
<form name="Comment" id="CommentSubmit" method="post" action="">
  <table width="90%" border="0" align="center" cellpadding="0"
  cellspacing="0">
    <tbody>
      <tr><td height="35">用户名: <input name="Author" type="text"
      class="inputtext" size="13" maxlength="16">&nbsp; <img src=
      "form_files/GetCode.asp" align="absmiddle"> ←请输入左边数字
      &nbsp; <input name="Code" type="text" class="inputtext" size="6">
      </td></tr>
      <tr><td><textarea name="Content" cols="54" rows="8" style=
      "background:no-repeat center center; background-image:
```

```

url(/Images/Text_Bg_1.gif);"></textarea></td></tr>
<tr><td height="35"><input name="imageField" type="image"
id="imageField"src="./form_files/Icon_Button.gif"align="absmiddle"
onClick="GetSubmitContent()">&nbsp; (200字以内) <input name=
"HideID" type="checkbox" id="HideID2" value="ok">匿名发表<input
name="SaveID" type="checkbox" id="SaveID2" value="ok" checked=
"">保存用户名 </td></tr>
</tbody>
</table>
</form>

```

(2) 在表单外添加与主题有关的 input 元素，代码如下。

```
<div style="margin-left:20px">内容主题: <input type="text" name="user
ContentTitle" /></div>
```

(3) 在第(1)步所示的代码中已经为按钮添加了 onClick 事件属性，用户单击该按钮时会触发其 Click 事件调用 GetSubmitContent()脚本函数。在该函数中首先获取表单对象，然后分别获取表单内的各个元素，同时也会获取表单外 name 属性的值是 userContentTitle 的元素，并且判断其值。代码如下。

```

function GetSubmitContent() {
    var formsval = document.forms["Comment"];           //获取表单对象
    var name = formsval["Author"].value;                 //获取用户名
    var content = formsval["Content"].value;            //获取内容
    var contenttitle = formsval["userContentTitle"];    //获取主题
    if(contenttitle==null || contenttitle==undefined) { //如果主题为空
        alert("没有获取到与发表内容有关的主题信息。");
    }else{
        alert(name+"关于 ["+contenttitle.value+"] 所发表的内容是: "+content);
    }
}

```

(4) 运行页面输入内容后进行测试，效果如图 3-28 所示。

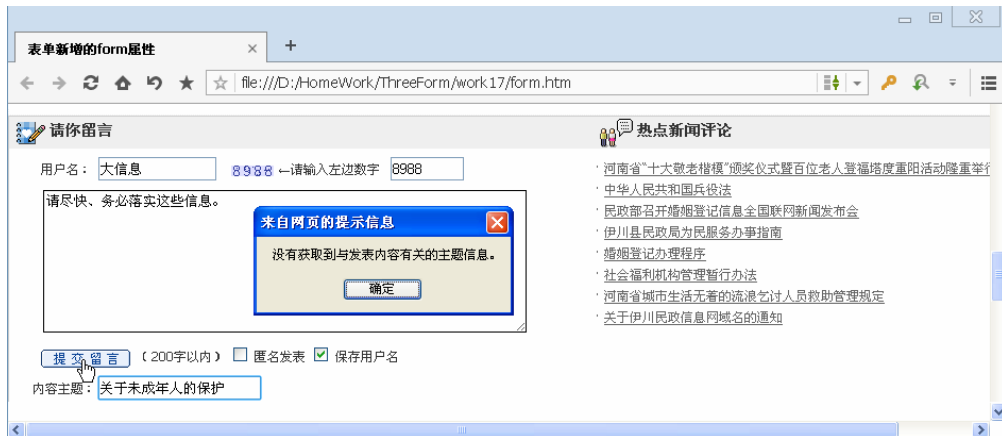


图 3-28 没有使用 form 属性时的效果

分析图 3-28 可以知道,在执行脚本代码时并没有获取到与主题有关的信息,这是因为主题内容在表单的外部。

(5) 重新为表单外的元素添加 form 属性,将 form 属性的值指定为 form 表单的 ID 属性的值。代码如下。

```
<div style="margin-left:20px">
    内容主题: <input type="text" name="userContentTitle" form="Comment-Submit" />
</div>
```

(6) 重新运行页面输入内容进行测试,运行效果如图 3-29 所示。



图 3-29 使用 form 属性后的效果

3.4.2 表单重写属性

HTML 5 中新增了多个与表单有关的属性,除了 form 属性外,还包含其他的属性,如表单重写属性,这些重写属性分别是 formaction、formenctype、formmethod、formnovalidate 和 formtarget,具体说明如下所示。

- (1) formaction 属性: 重写表单的 action 属性。
- (2) enctype 属性: 重写表单的 enctype 属性。
- (3) formmethod 属性: 重写表单的 method 属性。
- (4) formnovalidate 属性: 重写表单的 novalidate 属性。
- (5) formtarget 属性: 重写表单的 target 属性。

【练习 18】

表单中的这些重写属性仅适用于 submit 和 image 类型的 input 元素,如下示例代码简单演示了部分属性。代码如下。

```
<form id="form1" action="Index.html">
    Email: <input type="email" name="user_email" /><Br/>
```

```
<input type="submit" value="Submit" /><br/>
<input type="submit" value="作为管理员提交" formaction="adminIndex.
html" /><br/>
<input type="submit" value="不需要验证就可以提交" formnovalidate="true" />
</form>
```

上述代码在表单中首先添加用于用户输入时的电子邮箱，然后分别添加三个 submit 类型的 input 元素的执行按钮。为第二个按钮添加 formaction 属性，为第三个按钮添加 formnovalidate 属性。

执行上述代码，向输入框中输入内容后单击不同的按钮进行测试。单击名称是“Submit”的按钮时会验证用户输入的 E-mail 是否合法，如果合法提交到 Index.html 页面；单击名称是“作为管理员提交”的按钮时也会自动验证用户输入的 E-mail 是否合法，如果合法则会跳转到 adminIndex.html 页面(formaction 属性重写提交路径)；单击名称是“不需要验证就可以提交”的按钮时由于使用了 formnovalidate 属性，所以无论文本框中输入的 E-mail 是否合法，都会提交到 Index.html 页面。

HTML 4 以及之前的版本中，一个 form 指定了 action 后便只能将<form></form>标记中的元素提交到另一个地址。而在 HTML 5 中可以为所有的按钮添加表单重写属性指定不同的跳转页面。部分代码如下。

```
<form id="testform" action="index.html">
  <input type="submit" name="s1" value="v1" formaction="userIndex.
html" />提交到 s1
  <input type="submit" name="s2" value="v2" formaction="adminIndex.
html" />提交到 s2
</form>
<input form="testform" name="var1" value="val1" />
<input form="testform" name="var2" value="val2" />
```

3.4.3 placeholder 属性

placeholder 属性是 HTML 5 的新增属性，当用户还没有输入值时，可以通过设置 placeholder 属性向用户显示描述性说明或者提示信息。placeholder 属性只需要将说明性文字作为属性值即可，在字段获得焦点时或者输入内容后提示会自动消失；如果没有输入文本内容，那么提示会再次出现。

placeholder 属性不仅适用于普通的文本框，对于其他类型（例如 email、url 和 number 等）的输入框也适用。在支持 placeholder 属性的浏览器中，属性值会以浅灰色的样式显示在输入框中；在不支持 placeholder 属性的浏览器中，该属性会自动被忽略，以输入框的形式默认显示。

【练习 19】

许多网站的页面中都会在输入框内提示一些默认的信息。例如在微博中，用户可以更改个人资料，如果当前某个资料用户没有填写，则通过默认的 placeholder 属性进行提示。在页面中添加如下代码。


```
<input id="homepage" name="homepage" type="url" class="inputTxt en"
value="" placeholder="http://">
<input id="oldtel" name="oldtel" type="tel" class="inputTxt en"
pattern="^\d{3}-\d{8}|\d{4}-\d{7}$" placeholder="格式：410-12345678 或
0371-1234567">
<textarea name="summary" id="summary" class="inputArea" placeholder="
介绍个人资本资料、爱好即可"></textarea><b class="pass" style="display:
none"></b> <span rel="140字以内">140字以内</span>
```

上述代码中的第一个 `input` 元素表示用户所输入的个人主页, `placeholder` 属性指定没有内容时的提示; 第二个 `input` 元素表示固定电话, `placeholder` 属性指定电话的正确格式; `textarea` 表示个人介绍, `placeholder` 属性指定提示。

运行页面查看上述代码的运行效果, 如图 3-30 所示。



图 3-30 placeholder 属性的使用

3.4.4 required 属性

`required` 属性是一个必填项, 即输入框的内容不能为空。一旦将输入框的元素的 `required` 属性设置为 `true`, 那么此项就必须填写, 否则无法提交表单。

`required` 属性可以适用于多个类型的 `input` 元素, 但是 `button` 和 `range` 除外。在单击按钮执行操作之前, 必须对输入的内容进行验证。例如, 验证邮箱格式是否正确, 在指定 `input` 元素的类型是 `email` 的同时, 同时需要设置 `required` 属性的值。

`required` 属性的值是一个布尔类型, 因此, 使用该属性时有三种方法。代码如下。

```
<input type="text" required>
<input type="text" required="true">
<input type="text" required="required">
```

【练习 20】

几乎所有的网站中, 与用户输入内容有关的内容多数都是要求必须填写的, 例如用户名和联系电话。在微博个人资料修改页面也要求用户的某些内容是必须填写的, 例如

修改时的用户名、性别和生日。在页面中,为与用户名有关的 input 元素添加 required 属性,并指定该属性的值为 required。代码如下。

```
<input type="text" name="nick" id="nick" class="inputTxt" value="杜飞"
required="required">
```

重新运行页面进行测试,不向用户名输入框中输入内容直接单击按钮进行提交,Chrome 浏览器和 Opera 浏览器中的效果如图 3-31 和图 3-32 所示。



图 3-31 Chrome 浏览器效果

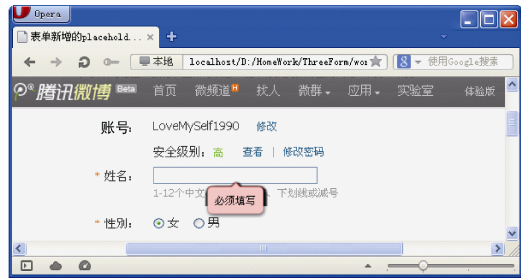


图 3-32 Opera 浏览器效果

3.4.5 pattern 属性

pattern 属性定义用于验证 input 元素的模式,这个模式是正则表达式,只有用户输入的内容与表达式匹配时才会被视为有效。

HTML 5 中 pattern 属性使用的正则表达式与 JavaScript 中所使用的正则表达式的语法是一样的,但是,该属性要求必须与整个值相匹配,而不仅仅是一个子集。

pattern 属性可以匹配许多内容,例如,匹配固定电话、手机号码、电子邮件、身份证号、用户名为中文、密码长度以及日期和时间等。如下代码演示了常用的几种表达式。

```
^\d{3}-\d{8}|\d{4}-\d{7}$ //匹配固定电话
^[u4e00-u9fa5\u2000-\ufc00-\uff00-\uffff]*$ //匹配是否是中文
^(?:\w+\.?)*\w+@(?:\w+\.?)*\w+$ //匹配电子邮箱
^[w-]+(\.[w-]+)*@[w-]+(\.[w-]+)+$ //匹配电子邮箱
^[A-Za-z0-9]+$ //匹配由数字和 26 个英文字母组成的字符串
^(d+)\.(d+)\.(d+)\.(d+)$ //匹配 IP 地址是否合法
^[1][3][0-9]{9}$ //匹配输入的手机号是否以 13 开头
^[ -]{0,1}(\d+)[.](\d+)$ //匹配输入字符串是否是带小数的数字格式,可以是负数
```

如果 pattern 属性所定义的正则表达式内容不是有效的,则该属性不会对输入的内容进行验证。与其他类型一样,pattern 属性不会自动验证内容为空的情况,它可以和 required 属性结合使用。

pattern 属性并不是适用于所有类型的 input 元素,它的适用类型有 text、search、email、url、tel 和 password。

【练习 21】

本次练习继续在练习 20 的基础上进行更改, 使用 `pattern` 属性指定用户所输入的用户名只能是中文, 并且长度在 2~8 位之间。相关代码如下。

```
<input type="text" name="nick" id="nick" class="inputTxt" placeholder="
输入 2-8 位的中文" required="required" pattern="^[\\u4e00-\\u9fa5\\uf900-\\
ufa2d]{2,8}$" title="输入 2-8 位的中文">
```

运行页面后输入内容进行测试, 如图 3-33 和图 3-34 所示分别为在 Chrome 浏览器和 Opera 浏览器中的效果。



图 3-33 Chrome 浏览器效果

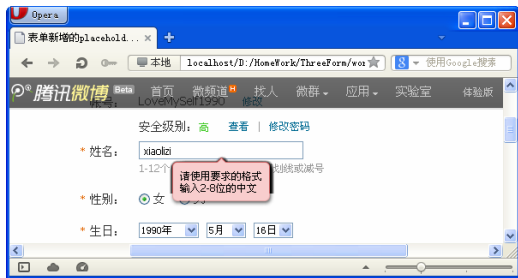


图 3-34 Opera 浏览器效果

92

**试一试**

`pattern` 属性可以指定不同的正则表达式, 感兴趣的读者可以为不同的 `input` 元素类型设置 `pattern` 属性进行测试。

3.4.6 autocomplete 属性

`autocomplete` 属性能够知晓是否应该保存输入值以备将来使用, 简单来说, 使用该属性可以实现自动完成的功能。该属性可以用来保护敏感数据, 避免本地浏览器对它们进行不安全的存储。

`autocomplete` 属性有 `on` 和 `off` 两个属性值, 在对其不指定值时, 使用浏览器的默认值。如果把该属性的值设置为 `on` 时, 可以显示指定候补输入的数据列表。使用 `datalist` 元素与 `list` 属性提供候补输入的数据列表, 自动完成时, 可以将 `datalist` 元素中的数据作为候补输入的数据在文本框中自动显示。

`autocomplete` 属性的使用非常简单, 直接在其指定的元素类型上使用即可。支持 `autocomplete` 属性的 `input` 元素类型可以是 `text`、`search`、`url`、`tel`、`email`、`password`、`range`、`color` 和日期类型。

【练习 22】

`autocomplete` 属性定义的效果要强于 `form` 元素, 本次练习分别为 `form` 表单、用户名输入框和邮箱输入框指定 `autocomplete` 属性。代码如下。

```
<form id="form1" action="newaction.html" method="get" autocomplete="on">
```

```
用户名: <input type="text" name="username" id="username" autocomplete="on"/>
Email: <input type="email" name="email" id="email" autocomplete="off" /><br/>
<input type="submit" value="Submit" />
</form>
```

上述代码将 form 元素的 autocomplete 属性的值指定为 on，然后再指定表单内的子元素的 autocomplete 属性的值。用户名可以自动完成，但是 E-mail 的设置不能自动完成，当用户在自动完成框中开始输入用户名时，浏览器会在该框中显示填写的选项。



试一试

datalist 元素和 list 属性结合使用可以提供数据列表选项，读者可以将它们与 autocomplete 属性结合起来再运行页面查看效果。

3.4.7 autofocus 属性

autofocus 属性规定页面加载后是否自动获取焦点，如果将该属性的值设置为 true，页面打开时，指定该属性的元素会自动获得焦点。该属性适用于所有 input 元素的类型，button、input、keygen、select 和 textarea 等元素都可以定义 autofocus 属性。

例如，网页上有一个搜索框，当网页加载完成后焦点就在搜索框上。代码如下。

```
<form action="" method="get">
  <input type="search" name="txtsearch" id="txtsearch" autofocus="true" />
</form>
```



注意

一个页面上只能有一个元素具有 autofocus 属性。只有当一个页面是以使用某个元素为主要目的时，才会对该元素使用 autofocus 属性，如搜索页面中的搜索文本框。

3.5 表单验证

表单验证是一套系统，它为终端用户检测无效的数据并标记这些错误，使用表单验证有两个好处：既避免了信息无法更新或出现新错误，又减轻了服务器端的压力。表单验证是一种用户体验的优化，让 Web 应用更快地抛出错误，但它仍不能取代服务器端的验证。由于前端验证是可以绕过的，因此，重要数据还要依赖于服务器端的验证。

3.5.1 自动验证

自动验证是 HTML 5 表单默认的验证方式，它会在表单提交时执行自动验证，如果

验证不通过，将无法提交表单。

HTML 5 中通过许多属性都能完成表单的自动验证，如下所示为与自动验证有关的属性说明。

1. required 属性

该属性的主要目的是确保表单元素中的值已填写。在提交时，如果元素中内容为空白则不允许提交，同时在浏览器中显示信息提示文字，提示用户该元素中必须输入内容。

2. pattern 属性

该属性根据设置的正则表达式的格式验证输入的内容是否有效，如果无效则不允许进行提交。

3. max 属性和 min 属性

max 属性和 min 属性主要限制数字类型输入范围的最大值和最小值，不在范围内的不允许进行提交。

4. step 属性

step 属性限制元素的值每次增加或者减少的基数，如果不是基数倍数时不允许提交。例如，当想让用户输入的值在 0~100 之间，而且必须是 5 的倍数时，可以将 step 属性的值指定为 5。

94

3.5.2 显式验证

可以使用自动验证方式对用户输入的内容进行验证，但是，自动验证并不能够完全满足用户的要求，这时，可以通过调用 `checkValidity()` 方法对表单内所有元素或者单个元素进行有效性验证。

当想要覆盖浏览器的默认验证和反馈过程时可以使用 `checkValidity()` 方法，该方法根据验证检查成功与否，返回 `true` 或 `false`，同时也会通知浏览器运行其检查。HTML 5 中的 `form` 元素、`input` 元素、`select` 元素和 `textarea` 元素都具有该方法。

【练习 23】

本次练习完成一个简单的留言验证，在页面中添加用于留言的用户名、邮箱地址和留言内容，然后在单击按钮时判断输入的内容是否合法。

(1) 添加新的页面并且进行设计，在页面中添加表示留言信息的表单，然后向表单中添加元素，再根据需要为每个元素添加属性。代码如下。

```
<form action="" method="post" name="formgb" id="formgb">
  <p>您的大名: <input type="text" size="50" style="width:235px" class=
    "input2" name="name" autofocus="true" required="true"></p>
  <p>您的邮箱:<input type="email" size="50" class="input2" name="email">
</p>
```

```

<p>您的留言内容: <textarea cols="40" rows="3" id="ccontent" name=
"content" class="input2" required="true"></textarea></p>
<p><input type="button" value="立即发表" class="button" onClick=
"CheckInput()"></p>
</form>

```

(2) 用户单击【立即发表】按钮时触发其 Click 事件调用 CheckInput() 脚本函数。该函数的代码如下。

```

function CheckInput(){
    var forms = document.forms["formgb"];           //获取表单元素
    var name = forms.name;                          //获取用户输入的用户名
    var mail = forms.email;                         //获取用户输入的邮箱
    var content = forms.content;                    //获取用户输入的内容
    if(!name.checkValidity()){                      //判断用户名是否合法，如果不合法
        alert("发言时您的大名怎么能为空呢!");
        name.focus();
        return false;
    }
    if(!mail.checkValidity()){                      //判断邮箱是否合法，如果不合法
        alert("为了方便对您的留言进行调查，请输入一个有效的邮箱地址!");
        mail.focus();
        return false;
    }
    if(!content.checkValidity()){                  //判断发言内容是否合法，如果不合法
        alert("您要发言的内容不能为空!");
        content.focus();
        return false;
    }
}

```

(3) 运行网页查看运行效果，向页面的文本框中输入内容后单击【立即发表】按钮测试效果，如图 3-35 所示为验证邮箱时的效果。

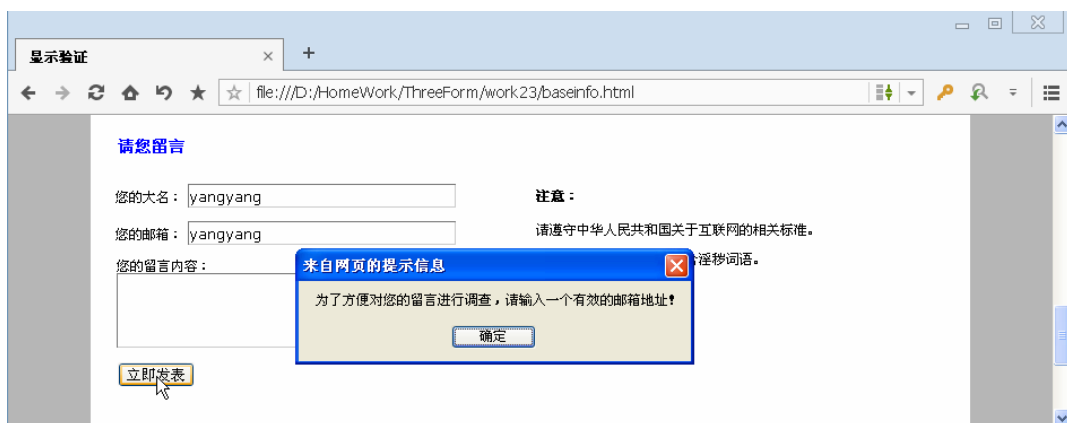


图 3-35 使用 checkValidity() 方法进行验证

除了 `checkValidity()` 方法外, `form` 和 `input` 元素还存在一个 `validity` 属性, 该属性返回一个 `ValidityState` 对象, 该对象的最常用属性是 `valid`, `valid` 表示表单内所有元素的内容是否有效。

【练习 24】

例如, 重新更改练习 23 中的代码, 通过 `input` 元素的 `validity` 属性所返回 `ValidityState` 对象的 `valid` 属性进行判断。JavaScript 中的代码如下。

```
function CheckInput() {
    var forms = document.forms["formgb"];           //获取表单元素
    var name = forms.name;                          //获取用户输入的用户名
    var mail = forms.email;                         //获取用户输入的邮箱
    var content = forms.content;                   //获取用户输入的内容
    if(!name.validity.valid){                       //验证用户名是否合法, 如果不合法
        alert("发言时您的大名怎么能为空呢!");
        name.focus();
        return false;
    }
}
```

3.5.3 自定义验证

HTML 5 中的许多 `input` 元素都带有对输入内容进行有效性检查的功能, 如果有效性检查不通过, 那么浏览器会针对元素提供错误信息。但是, 有时候开发人员并不想使用这些默认的错误信息, 而想自己定义错误信息, 或者想给某个文本框增加一种错误信息等, 这时可以使用 `setCustomValidity()` 方法。

`setCustomValidity()` 方法与 `checkValidity()` 方法一样, 通常都会结合 JavaScript 脚本一起使用, 该方法适用于 HTML 5 中的所有 `input` 元素。

【练习 25】

本次练习继续更改上面内容的代码, 实现的具体步骤如下。

(1) 更改 `form` 表单中的代码, 为输入内容的每个元素都添加 `id` 属性, `name` 属性可以去掉, 也可以保存, 最后更改按钮的类型为 `submit`。部分代码如下。

```
<textarea cols="40" rows="3" id="content" name="content" class="input2">
</textarea>
<input type="submit" value="立即发表" class="submit" onClick=
"CheckInput()">
```

(2) 重新更改 JavaScript 中的 `CheckInput()` 函数, 在该函数代码中判断用户名、邮箱和内容不能为空, 邮箱必须合法, 并且输入的内容不能少于 10 个字。部分代码如下。

```
function CheckInput() {
    var username = document.getElementById("name");
    var email = document.getElementById("email");
    var content = document.getElementById("content");
    if(username.value==" " || username.value==null){
```

```

username.setCustomValidity("请输入人名, 发言人名称不能为空!");
username.focus();
return false;
}
}

```

(3) 运行页面输入内容进行测试, 查看提示效果。由于浏览器不同, 可能提示的样式也会有所不同, 如图 3-36 和图 3-37 所示分别为 Maxthon 浏览器和 Opera 浏览器中的效果。

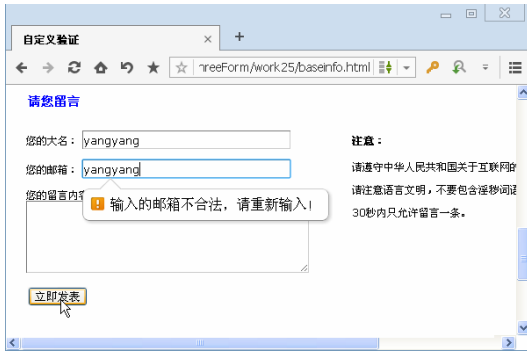


图 3-36 Maxthon 浏览器效果



图 3-37 Opera 浏览器效果

3.5.4 验证事件

除了与验证有关的属性和方法外, HTML 5 还提供了对表单的事件验证。表单元素为通过验证时触发, 无论是提交表单还是直接调用 `checkValidity()` 方法, 只要有表单元素没有通过验证, 就会触发 `invalid` 事件。通过这个事件, 可以处理和检查验证失败有关的信息, 并向用户反馈。

【练习 26】

`invalid` 事件本身不处理任何事情, 可以通过监听该事件, 自定义事件处理, 下面通过一个简单的练习演示 `invalid` 事件的使用方法。

(1) 添加一个新的页面, 在页面的 `form` 表单中添加用于输入的 `email` 类型的 `input` 元素和执行验证操作的 `submit` 类型的 `input` 元素。代码如下。

```

<form name="formgb" id="formgb">
  <input type="email" size="50" id="email" placeholder="电子邮箱"
    required>
  <input type="submit" value="立即发表" class="submit">
</form>

```

(2) 继续向表单中添加一个 `span` 元素, 该元素显示错误提示信息。代码如下。

```

<span id="msginfo" style="color:red; font-size:16px;">f</span>

```

(3) 在 JavaScript 脚本中添加页面初始化的 `load` 事件, 该事件调用 `loadDemo()` 函数。

向 loadDemo()函数中添加注册表单的 invalid 事件, 并且在该事件中调用 invalidHandler()函数进行验证, 然后通过 for 语句遍历表单中的内容, 为每一项内容添加 change 事件, 该事件调用 checkvalue()函数。代码如下。

```
function invalidHandler(evt) {           //定义表单验证方法
    checkvalue(evt);
}
function loadDemo() {
    var myform = document.getElementById("formgb");
    myform.addEventListener("invalid", invalidHandler, true); //注册表
    //单的 oninvalid 事件
    for(var i=0;i< myform.elements.length-1;i++){
        //注册表单元素的 onchange 事件, 优化用户体验
        myform.elements[i].addEventListener("change", checkvalue, false);
    }
}
window.addEventListener("load", loadDemo, false);
//在页面初始化事件时注册的自定义事件
```

(4) checkvalue()函数主要调用当前项的 checkValidity()方法进行验证, 然后进行判断, 如果验证不通过则为获取的 span 元素的 innerHTML 属性进行赋值。该函数代码如下。

```
var checkvalue = function(e) {
    var msg = document.getElementById("msginfo"); //获取 span 元素
    var el = e.target;
    var isvalid = el.checkValidity();           //验证内容
    if(!isvalid) {
        msg.innerHTML = "邮箱编号为空或者格式错误, 请重新输入。";
    }
    e.stopPropagation();
    e.preventDefault();                       //屏蔽浏览器后续错误
}
```

(5) 运行页面, 然后向页面中输入内容后单击按钮进行测试, 具体效果图不再显示。

通过使用 invalid 事件使表单开发更加灵活, 一般情况下, 在 invalid 事件处理完成后, 还是会触发浏览器默认的错误提示。必要时可以屏蔽浏览器后续的错误提示, 可以使用事件的 preventDefault()方法, 阻止浏览器的默认行为, 并自行处理错误提示信息。

3.5.5 取消验证

一个非常大的表单需要分为两部分, 或者更多部分, 在第二部分中有个文本框的内容是一定要填的, 如果填写每一部分内容则会消耗很多的时间。或者填写完第一部分内容后, 还要过一段时间填写第二部分的内容, 这时应该允许用户先提交保存第一部分的内容, 同时临时取消第二部分的内容表单验证。

取消表单验证的常用方法有两种, 第一种是利用 form 表单元素的 novalidate 属性,

它可以关闭整个表单验证，这样可以使 form 内所有表单控件不进行验证。如果表单的第二部分需要验证的内容过多，但是又想要提交表单的第一部分时，可以使用这种方法。

另外一种方法是利用 input 元素或者 submit 元素的 formnovalidate 属性，这样可以控制单个表单控件的验证。利用 input 元素的 formnovalidate 属性可以让表单验证对单个 input 元素失效，例如，当表单的第二部分需要验证的内容数量很少时，可以使用这种方法。为 submit 按钮使用 formnovalidate 属性，用户单击按钮时，相当于利用 form 元素的 novalidate 属性，整个表单都失效了。

3.6 实验指导 3-1：会员资料注册

表单信息提交用户所输入的内容，通常情况下，表单可以接受用户的注册信息、订餐地址信息，以及其他内容的添加或修改（如图书信息添加、订单添加和个人资料修改）等。本节实验指导主要利用表单的新增内容完成会员资料注册页面的设计，该页面所使用的主要技术如下所示。

- (1) 使用 datalist 元素和 list 元素为【姓名】输入框添加选项列表。
- (2) 使用新增的表单类型元素设计表单输入框，如个人主页、电子邮箱。
- (3) 使用新增的表单属性设置或验证表单内容。
- (4) 对表单输入的内容进行验证。

会员资料注册包含用户密码、姓名、密保问题、密保答案、年龄、联系电话、邮箱、个人主页以及所在地区和教育水平等。用户输入内容完成后可以单击页面中的【确认】按钮进行提交，提交时对输入的内容进行验证，验证密码时的效果如图 3-38 所示。

图 3-38 会员资料注册

根据如图 3-38 所示的内容设计页面，将需要填写和选择的内容放到 form 表单元素中，而与执行操作有关的按钮放在表单外，通过属性进行控制。另外，在本节实验指导中，密码、密保和答案通过 setCustomValidity()方法进行验证，其他的必填字段通过为元素指定 title 属性完善自动验证，主要实现步骤如下。

(1) 向页面中添加 form 表单元素，然后向表单中添加多行三列的表格，第一列说明，第二列向用户提供输入，第三列则显示一些说明信息。首先向页面添加【密码】和【确认密码】输入框，并且首次加载时将当前光标定位到【密码】框。部分代码如下。

```
<input id="userpass" size="25" type="password" maxlength="14" required autofocus>
<input id="userconfirmpass" size="25" type="password" maxlength="14">
```

(2) 设置保护问题和答案有助于密码丢失时的找回功能，这两个内容都要求用户必须输入。代码如下。

```
<input type="text" id="useranswer" size="25" maxlength="30" required>
<input type="date" id="userbirth" size="25" value="1990-11-11" maxlength="4" required>
```

(3) 姓名是用户注册时必不可少的一部分，本节实验指导要求姓名必须输入中文，并且在 2~8 位汉字之间。使用 datalist 元素指定选项列表，然后将列表内容绑定到姓名元素中。代码如下。

```
<input type="text" id="fullname" size="25" maxlength="8" pattern="^[\\u4e00-\\u9fa5\\uf900-\\ufa2d]{2,8}$" required title="必须输入 2-8 位的中文名字。" autocomplete="on" list="userlist">
<datalist id="userlist">
  <option>最爱中国文化</option>
  <option>秦时明月</option>
  <option>会说话的猪</option>
  <option>摇摇</option>
</datalist>
```

(4) 添加 number 类型的 input 元素显示年龄，指定年龄在 15~120 岁之间，且默认值是 23 岁。代码如下。

```
<input type="number" id="userage" max="120" min="15" value="23" />
```

(5) 依次添加用户输入时的邮箱和个人主页输入框，其类型分别是 email 和 url 的 input 元素，通过 placeholder 属性指定默认的值，title 属性显示标题。代码如下。

```
<input type="email" id="email" size="25" maxlength="36" placeholder="foverlove@163.com" title="必须是合法邮箱。">
<input type="url" id="personurl" size="25" maxlength="40" placeholder="http://weibo.com/" title="个人主页地址必须合法。">
```

(6) 添加【联系电话】输入框，用户可以输入以 13 或 15 开头的手机号，也可以输入固定电话，并且支持添加 1~4 位的分区号码。代码如下。

```
<input type="tel" id="phone" size="25" maxlength="18" required
placeholder="支持手机号码和固定电话" pattern="([1][3|5][0-9]{9})|^((\d{7,8})|(\d{4}|\d{3})-(\d{7,8})|(\d{4}|\d{3})-(\d{7,8})-(\d{4}|\d{3}|\d{2}|\d{1})|(\d{7,8})-(\d{4}|\d{3}|\d{2}|\d{1})))$" title="联系电话不合法, 请重新输入。">
```

(7) 所在地区、教育水平和职业都通过下拉列表框来指定, 其中为所在地区的 select 元素添加子元素 optgroup 元素, 然后再添加地区。部分代码如下。

```
<select id="province" size="1" class="p1" style="width:160px">
  <optgroup label="北京市">
    <option value="北京市-朝阳区">朝阳区</option>
    <option value="北京市-海淀区">海淀区</option>
  </optgroup>
  <!-- 省略其他 -->
</select>
```

(8) 在表单外部添加【确认】和【重填】按钮, 分别指定其类型是 submit 和 reset, 然后为 submit 类型的按钮添加 onClick 属性, 指定该属性的值。代码如下。

```
<input type="submit" value=" 确 认 " form="registerform" onClick="return
CheckInputContent()">
<input type="reset" value=" 重 填 " name="reset" form="registerform"
style="cursor:hand;">
```

(9) 向名称是 CheckInputContent 的函数中添加脚本代码, 在此函数中根据 ID 获取元素, 然后通过 value 属性获取输入的内容, 再通过 if else 语句进行判断。在判断语句中通过元素对象的 setCustomValidity()方法确定用户输入的内容是否符合要求, 这里以两次输入的密码进行判断。代码如下。

```
function CheckInputContent() {
  var password = document.getElementById("userpass"); //密码
  var passwordconfirm = document.getElementById("userconfirmpass");
  //确认密码
  if(password.value.length<6)
    password.setCustomValidity("请重新输入密码, 密码长度为 6-14 位任何字符!");
  else
    password.setCustomValidity("");
  if( password.value != passwordconfirm.value )
    passwordconfirm.setCustomValidity("两次输入的密码不一致, 请确保一致!");
  else
    passwordconfirm.setCustomValidity("");
  /* 省略其他内容 */
}
```

(10) 补充表单或页面中的其他内容, 例如样式代码和其他前面步骤中没有提到的内

容元素等。

(11) 运行页面并且输入内容进行测试，具体的效果图不再显示。

思考与练习

一、填空题

1. HTML 网页定义表单时，需要使用_____元素。
2. 如果文本框中要显示 `datalist` 元素所定义的选项列表，那么需要将文本框的_____属性的值设置为 `datalist` 元素的 `id` 属性的值。
3. _____属性能够向用户显示描述性说明或者提示信息。
4. _____类型的 `input` 元素是一种专门用来输入数字的文本框，在表单提交时会检查其中的内容是否为数字。
5. 自定义表单验证的错误信息时需要借助_____方法。
6. 页面中添加搜索关键词的文本框时，可以将 `input` 元素的类型设置为_____。

二、选择题

1. `email` 类型的文本框具有一个 `multiple` 属性，该属性的作用是_____。
 - A. 判断该文本框中所输入的值是否为正确的邮箱地址
 - B. 不允许该文本框的内容为空
 - C. 允许在文本框中输入使用逗号分隔的有效的 E-mail 地址的一个列表
 - D. 以上答案都正确
2. 用来输入 UTC 日期和时间的文本框是_____类型的。

- A. `date`
- B. `time`
- C. `datetime-local`
- D. `datetime`

3. HTML 5 改进了 `input` 元素的类型，其中_____是在 HTML 4 中就已经存在的类型。
 - A. `search`
 - B. `email`
 - C. `file`
 - D. `color`
4. 关于 HTML 5 新增的表单属性，下面说法_____是错误的。
 - A. `required` 是一个必填属性，它可以和其他属性结合使用，如 `pattern` 属性
 - B. `autofocus` 规定页面加载后是否自动获取焦点，一个页面中可以有 0 个或多个该属性
 - C. 表单重写属性并不适用于所有的元素，仅适用于 `submit` 和 `image` 类型的 `input` 元素
 - D. `pattern` 用于验证 `input` 元素的正则表达式，只有输入的内容与表达式匹配时才有效，同时要求正则表达式必须正确，否则将不会进行验证

三、简答题

1. JavaScript 脚本中如何获取表单以及表单中的元素？请分别进行说明。
2. HTML 5 中新增加的与日期和时间有关的类型有哪些？分别举例说明。
3. 你对表单验证了解多少？请进行简单的说明。