**第8章 表单**

**学习重点**

常见的表单属性了解

理解表单的作用

掌握文本域、密码字段的创建、单选按钮、提交按钮、复选框的创建

掌握嵌套列表的使用

学会使用列表进行网页开发

**知识结构图**

**本章介绍**

Web页面是一种文档，HTML就是用来编写这些文档的一种标记语言，文档的结构和格式的定义是由HTML元素来完成的，HTML元素是由单一或一对标签定义的包含范围。表单在HTML文档和用户之间，提供了一种交互的方式。利用表单，可以提交信息到Web服务器。利用表单填写好信息以后，点击“提交”按钮，信息就会被浏览器发送到服务器端。当服务器端收到信息后，对这些信息进行一些处理，然后返回信息到浏览器，最终呈现在用户面前。

**8.1表单简介**

HTML表单是一个包含表单元素的区域，用来收集用户输入的内容并提交，使用标签<form>来设置，比如：文本域(textarea)、下拉列表、单选框(radio-button)、复选框(checkbox)等等，常用标签见表8-1。

**表8-1 常用表单标签**

|  |  |
| --- | --- |
| 描述 | 标签符号 |
| 表单 | <form></form> |
| 输入域 | [<input>](http://www.runoob.com/tags/tag-input.html) |
| 文本域 | [<textarea></textarea>](http://www.runoob.com/tags/tag-textarea.html) |
| 输入标题 | [<label></label>](http://www.runoob.com/tags/tag-label.html) |
| 下拉列表 | <select></select> |
| 下拉列表中的选项 | [<option></option>](http://www.runoob.com/tags/tag-option.html) |
| 点击按钮 | [<button></button>](http://www.runoob.com/tags/tag-button.html) |

**8.2 form标签**

**8.2.1 form表单使用简介（标号）**

表单使用表单标签 form 来设置:在定义一个表单时<form>标签开始，以</form>标签结束。而form中往往包含了input、button等子标签。

使用方法如下：

<form action="action\_page.php" method="get">  
 <label>姓名：</label><input type="text" name="firstname"><br>

<input type="submit" value="Submit">  
</form>

**8.2.2 form标签属性**

**action 属性：**

action 属性定义在提交表单时执行的动作。向服务器提交表单的通常做法是使用提交按钮。通常，表单会被提交到 web 服务器上的网页。在上面的例子中，指定了某个服务器脚本来处理被提交表单：

<form action="action\_page.php"></form>

如果省略 action 属性，则 action 会被设置为当前页面。

**method 属性**

method 属性规定在提交表单时所用的 HTTP 方法（GET 或 POST）：

<form action="action\_page.php" method="GET"></form>

或：

<form action="action\_page.php" method="POST"></form>

何时使用 GET？GET是浏览器默认方法，如果表单提交是被动的（比如搜索引擎查询），并且没有敏感信息时就可以使用GET。当您使用 GET 时，表单数据在页面地址栏中是可见的。当发送数据时，GET 方法向 URL 添加数据；URL 的长度是受限制的，URL 的最大长度是 2048 个字节）。

何时使用 POST？如果表单正在更新数据，或者包含敏感信息（例如用户名、密码等）。POST 的安全性更好，因为在页面地址栏中被提交的数据是不可见的。当发送数据时，理论上讲传输大小是无限制的，但是这也要取决于服务器的设置和内存大小。

**name 属性**

name属性在HTML文档以及在后面的JavaScript中的直接作用不是那么明显的。但是它在用于表单中却是必不可少的属性。name属性主要是用于获取提交表单的某表单域信息，作为可与服务器交互数据的HTML元素的服务器端的标识，比如input、select、textarea、框架元素(iframe、frame、window的名字，用于在其他frame或window指定target)和button等，这些元素都与表单提交有关，浏览器会根据name来设定发送到服务器的request，在表单的接收页面只接收有name的元素, 所以赋id的元素通过表单是接收不到值的。 form表单的具体属性见表8-2。

标签的id属性会在css选择器中进行具体讲解。

**表8-2 form的属性全览**

表单还有一个很重要的属性：表单对象。表单对象会在JavaScript章节具体介绍。

|  |  |
| --- | --- |
| 属性 | 说明 |
| accept-charset | 规定在被提交表单中使用的字符集（默认：页面字符集）。 |
| action | 规定向何处提交表单的地址（URL）（提交页面）。 |
| autocomplete | 规定浏览器应该自动完成表单（默认：开启）。 |
| enctype | 规定被提交数据的编码（默认：url-encoded）。 |
| method | 规定在提交表单时所用的 HTTP 方法（默认：GET）。 |
| name | 规定识别表单的名称。 |
| novalidate | 规定浏览器不验证表单。 |
| target | 规定 action 属性中地址的目标（默认：\_self）。 |

何时使用 POST？

您应该使用 POST：

如果表单正在更新数据，或者包含敏感信息（例如密码）。

POST 的安全性更加，因为在页面地址栏中被提交的数据是不可见的。

**8.3 input标签简介**

input 标签规定了用户可以在其中输入字段。input 元素在 form 元素中使用，用来声明允许用户输入数据的 input 控件。输入字段可通过多种方式改变，可以是文本字段、复选框、单选按钮等，取决于 type 属性详细内容见表8-3。

**表8-3 常用表单标签**

|  |  |
| --- | --- |
| type属性 | input控件 |
| text | 单行文本框 |
| password | [密码文本框](http://www.runoob.com/tags/tag-input.html) |
| submit | 提交按钮 |
| reset | 重置按钮 |
| radio | 单选按钮 |
| checkbox | 复选框 |
| button | 普通按钮 |

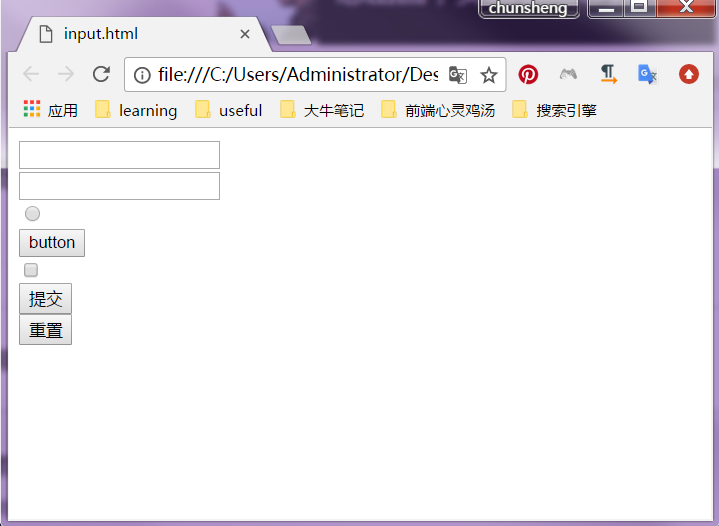
input标签的示例见例8-1，运行结果见图8-1。

**【例8-1】input标签的语法**

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <title>Input语法</title>
6. </head>
7. <body>

input标签主要是通过设置不同的type属性来实现不同的结果。

1. <input type="text" name= "text">
2. <input type="password" name= "password">
3. <input type="radio" name= "radio">
4. <input type="button" value= "button">
5. <input type="checkbox" name= "checkbox">
6. <input type="submit" value= "提交">
7. <input type="reset" value= "重置">
8. </body>
9. </html>



**图8-1 input标签**

**8.4文本框**

**8.4.1文本框text简介**

当用户要在表单中键入字母、数字等内容时，就会用到文本输入框。

注意:表单本身并不可见。同时，在大多数浏览器中，文本框的缺省宽度是20个字符。

文本输入框是服务器交流最常用的表单，使用方法如下：

<form>

<span>姓名：</span><input type="text" name="username">

</form>

文本框的常用属性见表8-3。

**表8-3 text属性**

|  |  |
| --- | --- |
| 属性 | 说明 |
| name  文本框的属性有很多，这里只列举了几个常见的属性，更多属性的请参考W3C用户手册。 | 设置文本框对象名称 |
| width | 设置对象的宽度 |
| alignment | 设置Text文本非对齐方 |
| appearance | 设置是否要用立体效果显示文本 |
| borderStyle | 设置边界模式 |
| dataSource | 设置此对象链接到数据源的名称 |

**8.4.2 密码文本框简介**

密码文本框可以说是一种特殊的文本框，它和普通文本框的属性都相同。密码文本框与普通文本框不同的是，普通文本框输入的字符可见而密码文本框输入的字符不可见，这个设置主要是为了防止周围的人看到自己的密码。密码文本框的语法跟普通的文本框只是把type的值设置为password即可：

密码文本框password仅仅使周围的人看不见输入的文本，但是它并不能真正使得数据安全。为了数据安全，通常采用的方法是前后台进行加密数据传输。

<form>

密码：<input type="password" name="password" >

</form>

密码文本框跟普通文本框属性类型一样。密码文本框的常见属性见表8-4。

**表8-4 密码文本框password属性**

在HTML文档的标签中，对于属性的设置没有先后顺序，可根据自己的习惯进行设置。

|  |  |
| --- | --- |
| 属性 | 说明 |
| value | 定义文本框的默认值，也就是文本框内的文字 |
| size | 定义文本框显示文本的长度，以字符为单位 |
| maxlength | 设置文本框中最多可以输入的字符数 |

1. **实例介绍**

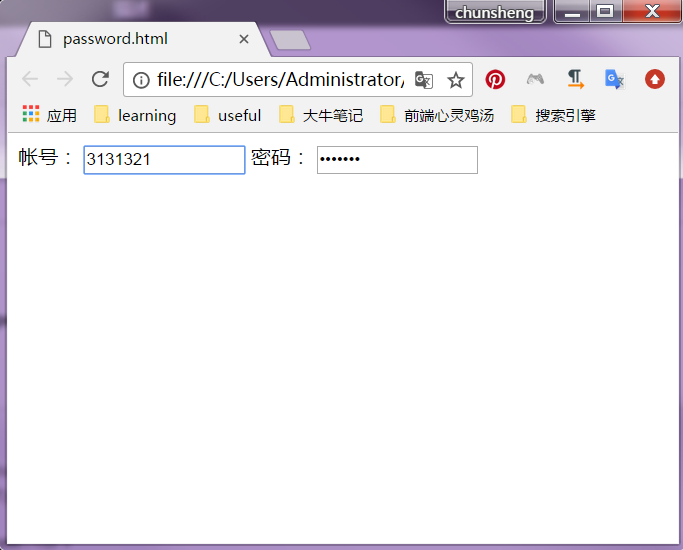
**【例8-2】**密码文本框的使用方法

1. <!DOCTYPE html>
2. <html>
3. <head>
4. <meta charset="UTF-8">

size属性和maxlength的区别在于，maxlength规定了文本框的最大输入长度，也就是超出这个长度之后的字符属于无效字符。而size则是让文本框只显示设置的字符长度，多余的字符有效但不可见。

1. <title>密码文本框</title>
2. </head>
3. <body>
4. <form name="form1" method="post" action="">
5. <span>帐号：</span>
6. <input type="text" />
7. <span>密码：</span>
8. <input type="password" size="10" maxlength="10"/>
9. </form>
10. </body>
11. </html>

在浏览器中，普通的文本框输入的文本会显示正常的文本形式。而在密码文本框中输入的文本就会被“.”代替，避免自己的密码被别人看见。运行结果见图8-2。



**图8-2 密码文本框示例**

**8.5 单选按钮与复选框**

**8.5.1 label简介**

label标签为input元素定义标注。label元素不会在浏览器中呈现特殊的样式效果，但是却可以让鼠标用户更好地进行网页交互。如果在 label 元素内点击文本，浏览器就好自动将焦点转到和标签相关的表单控件上。label元素的for属性应当与相关标签元素的id属性一致。

**8.5.2 单选按钮简介**

在网页中可能会遇到需要选择的时候，比如做选择题，性别选择之类的网页交互事件。在 HTML 表单中 <input type="radio"> 每出现一次，一个 radio 对象就会被创建。单选按钮是表示一组互斥选项按钮中的一个。当一个按钮被选中，之前选中的按钮就变为非选中的。

使用方法为把input标签的type属性设置为radio：

<input type="radio" name="radio" >

**8.5.3 复选框简介**

  多选复选框表单控件在网页中也很常见，故一组复选框可以选择多个答案。checkbox 对象代表一个 HTML 表单中的 一个选择框。在 HTML 文档中 <input type="checkbox"> 每出现一次，checkbox 对象就会被创建。使用方法为把input标签的type属性值设置为checkbox：

<input type="checkbox">

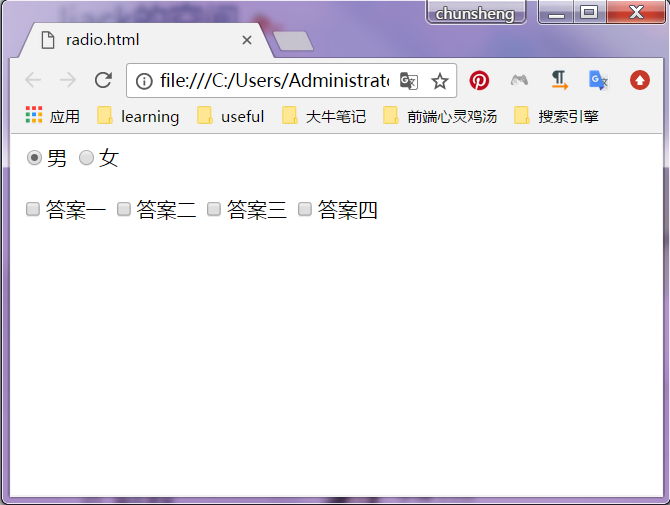
单选按钮一般是一组的形式出现，不会单个出现，每一组的name属性必须保持一致。当设置默认选中的时候，单选按钮和复选框都有checked属性。checked的值为true时，按钮就被选中；checked的值为false时就是没有选中。单选按钮与复选框的示例见例8-3，运行结果见图8-3。

【**例8-3**】单选按钮的使用方法

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. </head>
6. <body>
7. <form>

两个单选按钮name属性都一样，两个单选按钮才能互斥。用label标签把按钮和文本包括在一起的作用是：点击文本单选按钮也会被选中。

1. <label><input type="radio" name="sex" value="male" checked="true">男</label>
2. <label><input type="radio" name="sex" value="female">女</label>
3. <br>
4. <br>
5. <label><input type="checkbox" name= "answer" value="1">答案一</label>
6. <label><input type="checkbox" name= "answer" value="2">答案二</label>
7. <label><input type="checkbox" name= "answer" value="3">答案三</label>
8. <label><input type="checkbox" name= "answer" value="4">答案四</label>
9. </form>
10. </body>
11. </html>



**图8-3 单选按钮和复选框示例**

**8.6表单按钮**

在表单中为了提交数据或清除已经填写好了的数据，我们常常会用到表单按钮。按钮一般分为两类，一类本身就具有特定的功能，叫特别按钮，如Submit（提交按钮）用于传输用户所填写的信息至服务器、Reset（复原按钮）清除所填写的信息以利重新填写；另一类本身不具特别功能的，叫普通按钮。特别按钮只能用于表单(form)中才能发挥特别的功能。而普通按钮除可在表单中应用外，在网页的其他地方使用也非常方便。

**8.6.1普通按钮button**

普通的按钮有两种写法：第一种是直接写<button></button>标签，button标签是非自闭合标签，所以要写完整；第二种是用input标签把type属性设置为button，然后把value属性设置为按钮的名字。使用方法如下：

<input type="button" value="按钮">

或

<button>按钮</button>

**8.6.2提交按钮submit**

提交按钮只能用于表单中，在点击该按钮时把表单中的内容直接提交给服务器端，不需要任何的JavaScript操作。它跟普通的按钮区别是，普通按钮不添加JavaScript事件，按下去不会有任何效果。提交按钮只需把input标签的type属性设置为submit即可，使用方法如下：

<input type="submit" value="提交">

**8.6.3重置按钮reset**

重置按钮同样只能用于表单中，用于填好的表单需要修改时，比如用户输入“用户名”后，发现书写有误，可以使用重置按钮使输入框恢复到初始状态。使用方法为把input标签的type属性的值设置为reset，使用方法如下：

<input type="reset" value="重置">

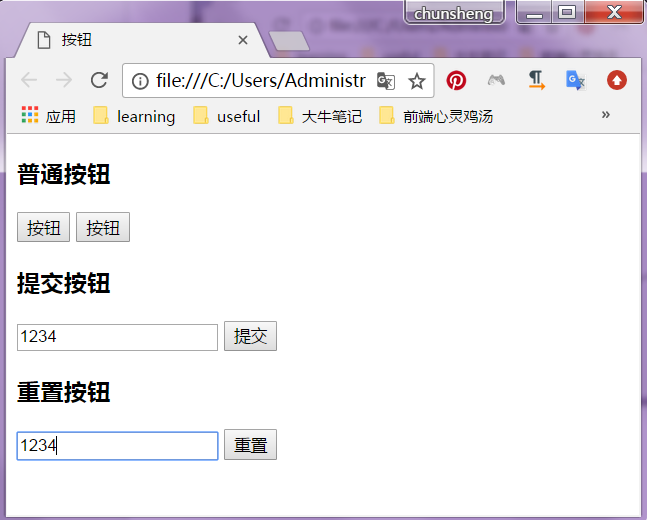
几种按钮的示例见例8-4，运行效果见图8-4。

【**例8-4**】**按钮使用方法**

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

为了效果能够显示在地址栏上，故使用get方法。由于没有服务端程序action就留为空。点击提交之后就会提交文本框的name属性值和文本框的内容。

1. <head>
2. <meta charset="UTF-8">
3. <title>按钮</title>
4. </head>
5. <body>
6. <form method="get" action="">
7. <h3>普通按钮</h3>
8. <button>按钮</button>
9. <input type="button" value="按钮">
10. <h3>提交按钮</h3>
11. <input type="text" name="key">
12. <input type="submit" value="提交">
13. <h3>重置按钮</h3>
14. <input type="text">
15. <input type="submit" value="重置">
16. </form>
17. </body>
18. </html>



点击重置之后该表单中的所有文本框里文本都会被清空。

**图8-4 表单按钮示例**

**8.7图片域image**

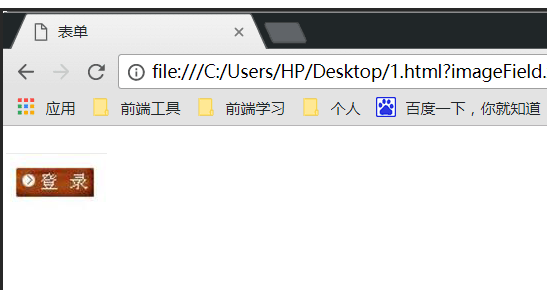
图像域是指用在提交按钮位置的图像，使得这幅图像具有按钮的功能。一般来说，使用默认的按钮形式往往会让人觉得单调，若网页使用了较为丰富的色彩，或者稍微复杂的设计，再使用表单默认的按钮形式可能会破坏整体的美感。这时，可以使用图像域，创建和网页整体效果一致的图像提交按钮。使用方法为把input标签的type属性值设置为image，然后再把src的属性值设置为图片的路径：

<input  name="图像域的名称"  type="image"  src="图像域的地址"/>

下面是一个图片域的使用例子：

<input type="image" name="imageField" src="images/enter．jpg"/>

标记将图像域的名称设置为imageField（不会影响图片的显示），地址设置为images/enter.jpg，在浏览器中浏览效果，如图8-5所示：



**图8-5 图片域image示例**

**8.8隐藏域hidden**

  隐藏域在页面中对于用户是不可见的，在表单中插入隐藏域的目的在于收集或发送信息，以利于被处理表单的程序所使用。浏览者单击发送按钮发送表单的时候，隐藏域的信息也被一起发送到服务器。使用方法为把input标签的type属性设置为hidden：

<input type="hidden" name="field＿name" value="value">

它通常使用的环境是前后台的数据交换，并且不能影响页面内容的时候。比如在处理一个订单号向后台获取数据时，订单号不能直接显示在页面上，但是这个参数又是必不可少的，此时利用隐藏域就是很好的选择。

**8.9文件域file**

最早的http post方法是不支持文件上传的。在1995年ietf出台了rfc1867用以支持文件上传。所以content-type的类型扩充了multipart/form-data用以支持向服务器发生二进制数据。其实form表单在不写enctype是同样存在默认值：application/x-www-form –urlencoded。

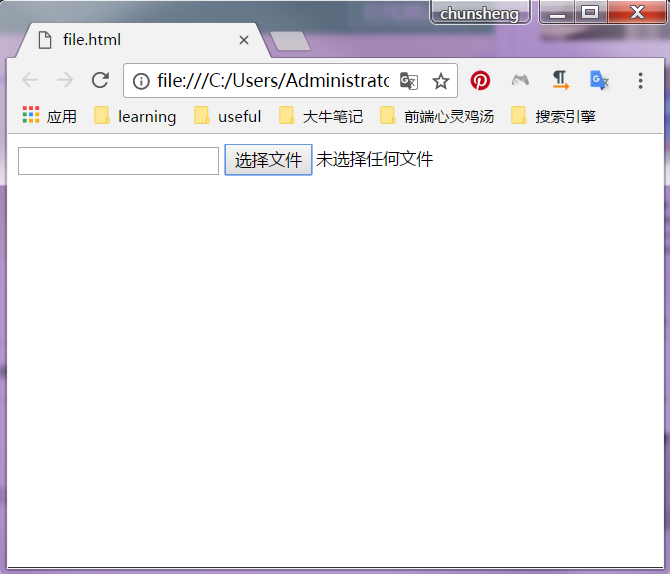
在网页中经常会使用到文件上传功能，在HTML文档中只需要把input标签的type属性值设置为file即可创建一个文件域 。与其他表单元素使用方法不同的是：在使用文件域时，form的enctype属性值必须设置为“multipart/form-data”。文件域可以实现文档、表格、压缩包、图片等文件的上传。文件域使用方法见例8-5，运行结果见图8-6。

**【例8-5】文件域使用方法**

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">

文件域前面增加一个文本框一般用来展示上传文件或是缓存文件的基本信息。

1. <title>文件域</title>
2. </head>
3. <body>
4. <form method="post" action="" enctype= "multipart/form-data">
5. <input type="text" name="desc">
6. <input type="file" name="pic">
7. </form>
8. </body>
9. </html>



**图8-6 文件域示例**

当点击选择文件按钮时就会打开本地的文件夹，用户需要从本地文件中选择一个文件上传。当然这里是不能上传，文件的上传功能需要用到后台服务器的接收及保存。

**8.10 文本域textarea**

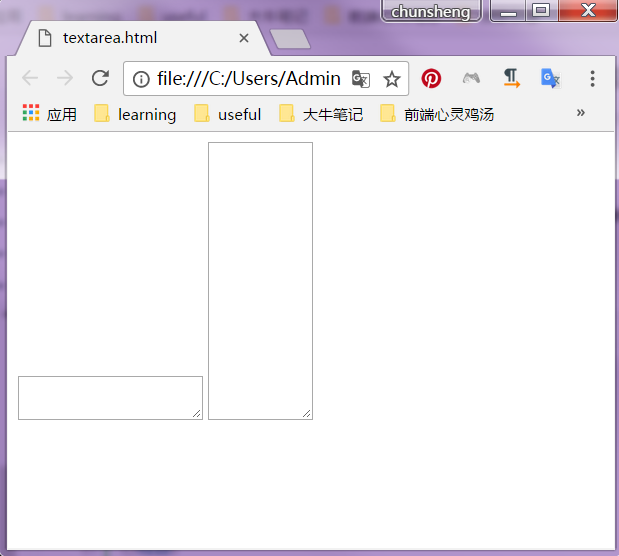
在网页中通常会出现很多的文本内容交互，比如说邮件内容等，这就不是文本框能够完成的功能。textarea标签定义多行的文本输入控件。文本域中可以容纳无限的文本字符。并且可以用cols和rows属性规定文本域的尺寸（最好使用后面的css来写样式），甚至还可以直接在右下角拖动改变大小。使用方法是使用<textarea></textarea>标签来创建文本域，见例8-6，运行结果见图8-7。

**【例8-6】**文本域使用方法

1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="utf-8">
5. </head>

通过cols和rows属性规定文本域的尺寸。

1. <body>
2. <textarea></textarea>
3. <textarea cols="10" rows="15"></textarea>
4. </body>
5. </html>



**图8-7 文本域实例**

**8.11下拉列表select**

在网页中下拉列表也是很常见的，select元素用来创建下拉列表。select元素中的option标签定义了列表中的可用选项。或者收集用逗号分隔的多个选项，将其合成一个单独的参数列表，并且在将 select 表单数据提交给服务器时包括 name 属性。使用方法如下：

<form>

<select name="cars">

<option value="volvo">Volvo</option>

<option value="saab">Saab</option>

<option value="fiat">Fiat</option>

<option value="audi">Audi</option>

</select>

</form>

**8.11.1 select标签属性**

select 元素可创建单选或多选菜单。select 对象代表 HTML 表单中的一个下拉列表。在 HTML 表单中，<select> 标签每出现一次，一个 select 对象就会被创建。常用属性见表8-5。

**表8-5 select标签属性**

|  |  |
| --- | --- |
| 属性 | 说明 |
| [disabled](http://www.w3school.com.cn/jsref/prop_select_disabled.asp) | 设置或返回是否应禁用下拉列表。 |
| [form](http://www.w3school.com.cn/jsref/prop_select_form.asp) | 返回对包含下拉列表的表单的引用。 |
| [id](http://www.w3school.com.cn/jsref/prop_select_id.asp) | 设置或返回下拉列表的 id。 |
| [length](http://www.w3school.com.cn/jsref/prop_select_length.asp) | 返回下拉列表中的选项数目。 |
| [multiple](http://www.w3school.com.cn/jsref/prop_select_multiple.asp) | 设置或返回是否选择多个项目。 |
| [name](http://www.w3school.com.cn/jsref/prop_select_name.asp) | 设置或返回下拉列表的名称。 |
| [selectedIndex](http://www.w3school.com.cn/jsref/prop_select_selectedindex.asp) | 设置或返回下拉列表中被选项目的索引号。 |
| [size](http://www.w3school.com.cn/jsref/prop_select_size.asp) | 设置或返回下拉列表中的可见行数。 |
| [tabIndex](http://www.w3school.com.cn/jsref/prop_select_tabindex.asp) | 设置或返回下拉列表的 tab 键控制次序。 |
| [type](http://www.w3school.com.cn/jsref/prop_select_type.asp) | 返回下拉列表的表单类型。 |

**8.11.2 option标签属性**

option 元素定义下拉列表中的一个选项（一个条目）。浏览器将 option 标签中的内容作为 select 标签的菜单或是滚动列表中的一个元素显示。option 元素位于 select 元素内部。常用属性见表8-6。

**表8-6 option标签属性**

|  |  |
| --- | --- |
| 属性 | 说明 |
| disabled | 规定此选项应在首次加载时被禁用。 |
| text | 定义当使用 <optgroup> 时所使用的标注。 |
| selected | 规定选项（在首次显示在列表中时）表现为选中状态。 |
| text | 定义送往服务器的选项值。 |

**8.11.3 下拉列表实例**

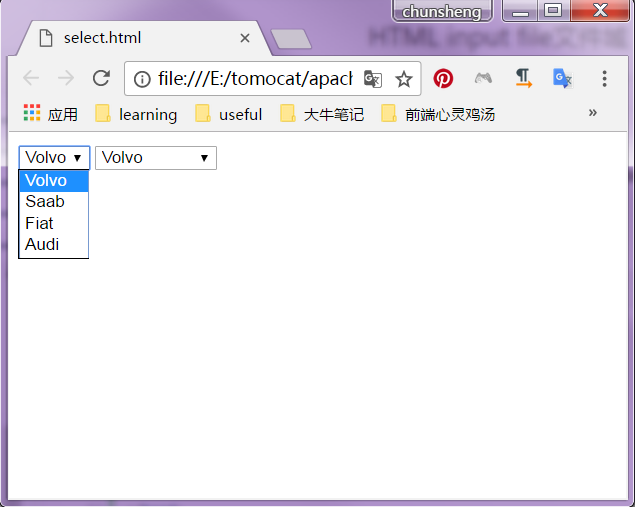
本实例介绍普通的下拉列表和具有optgroup标签的组合选项下拉列表。optgroup 元素用于组合选项。当您使用一个长的选项列表时，对相关的选项进行组合会使处理更加容易，具体使用方法见例8-7，运行结果见图8-8及图8-9。

【**例8-7**】下拉列表使用方法

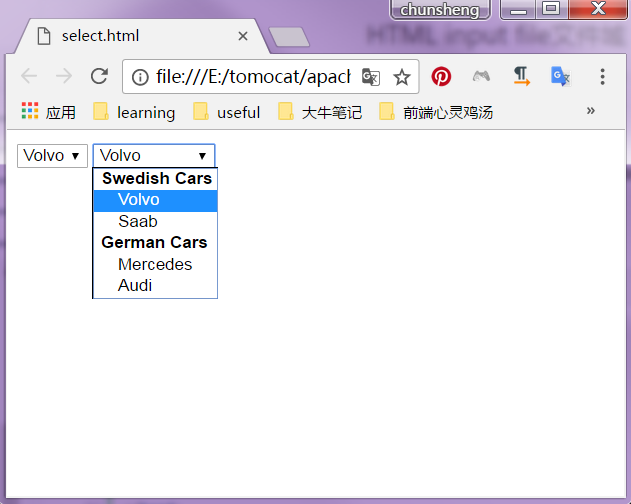
1. <!DOCTYPE html>
2. <html lang="en">
3. <head>
4. <meta charset="UTF-8">
5. <title>下拉列表</title>
6. </head>
7. <body>
8. <form>
9. <select name="cars">
10. <option value="volvo">Volvo</option>
11. <option value="saab">Saab</option>
12. <option value="fiat">Fiat</option>
13. <option value="audi">Audi</option>
14. </select>
15. <select>
16. <optgroup label="Swedish Cars">

optgroup对option元素进行了分组，使下拉表单内容更清楚地呈现。

1. <option value="volvo">Volvo</option>
2. <option value="saab">Saab</option>
3. </optgroup>
4. <optgroup label="German Cars">
5. <option value="mercedes">Mercedes </option>
6. <option value="audi">Audi</option>
7. </optgroup>
8. </select>
9. </form>
10. </body>
11. </html>



**图8-8 普通下拉列表框实例**



**图8-9 组合下拉列表框实例**

**8.15本章总结**

本章学习了form表单的创建。学习了常见表单的各种元素，如input、button、select等。表单是用于与用户交互的重要部分，接收用户信息离不开表单的使用。表单的属性非常多，但我们初学只需要记住常用的几个就可以了。input标签在使用时一定要注意书写它的type属性，不同的type属性值的功能不一样，如text、password。通过本章的学习，我们就可以建立一个简单的获取用户信息的表单啦。

**8.16 最佳实践**