

HTML 表格与表单

本章学习目标

- 掌握 HTML 表格的基本使用;
- 掌握 HTML 表单的基本使用;
- 了解前端与后端如何交互与通信。

表格主要用途是以网格的形式显示二维数据,HTML 早期版本中,常用表格来控制页面的内容布局,表单是 HTML 中获取用户输入的手段。它对于 Web 应用系统极其重要。表单的出现则使网页从单向的信息传递发展到能够与用户进行交互对话。本章将对表格和表单的相关知识进行详细讲解。

3.1 HTML 表格

生活中,经常使用表格来统计数据和信息,这样可以更清晰地显示数据或信息,同理在制作网页时,为了有条理地显示网页中的元素,可以使用表格对网页进行布局和规划,从而可以给浏览者展示大量且清晰的排列数据。表格在网页中应用得极其广泛,下面先来看下网页中表格的展示效果,如图 3.1 和图 3.2 所示。

编号	学号	姓名	学历	模拟面试	是否工作	薪水	入职单位	就业地点
2	BK170813016	张*生	本科	通过	是	15000	数**银行	北京
3	BK171413054	黄*琪	本科	通过	是	8000	长**公司	长沙市
4	BK170813024	肖*伟	专科	通过	是	14000	财**达	北京
6	BK170813010	卢**	专科	通过	是	13500	科**特	北京
7	BK171413034	平*华	本科	通过	是	14000	中**软	杭州市
9	BK171613004	张*维	本科	通过	是	15000	云**佳	杭州市

图 3.1 网页中表格的展示效果

东部排名	排名	球队	胜	负	胜场差	胜率	主场	客场	东部	得分	失分	净胜	连胜/负
01		猛龙	59	23	0.0	72%	34-7	25-16	40-12	111.7	103.7	8.0	1连胜
02		凯尔特人	55	27	4.0	67%	27-14	28-13	33-19	103.9	100.5	3.4	1连胜
03		76人	52	30	7.0	63%	30-11	22-19	34-18	109.6	105.4	4.2	16连胜
04		骑士	50	32	9.0	61%	29-12	21-20	35-17	111.0	109.9	1.1	1连负
05		步行者	48	34	11.0	58%	27-14	21-20	32-20	105.6	104.2	1.4	1连负

图 3.2 网页中表格的展示效果

3.1.1 表格基本结构

每个表格有三个必须的标签，<table>、<tr>和<td>三个标签，用来创建表格，其语法格式如下：

```
<table>
  <tr>
    <td>单元格内容</td>
    .....
  </tr>
</table>
```

上面三个标签是创建表格的基本标签，其中<table></table>标签用于定义一个表格。<tr>标签用于定义表格中的一行，必须嵌套在<table></table>标签中，在<table></table>标签中包含几对<tr></tr>，表示该表格有几行。<td></td>标签用于定义表格中的单元格，必须嵌套在<tr></tr>标签中，一对<tr></tr>包含几对<td></td>，表示该行中有多少个单元格（列）。

接下来通过案例来演示定义表格标签，如例 3-1 所示。

【例 3-1】 定义表格标签的演示案例。

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>HTML 表格</title>
6  </head>
7  <body>
8  <table>
9    <tr>
10     <td>单元格 1</td>
11     <td>单元格 2</td>
12   </tr>
13   <tr>
14     <td>单元格 1</td>
15     <td>单元格 2</td>
16   </tr>
17 </table>
18 </body>
19 </html>
```

运行结果如图 3.3 所示。

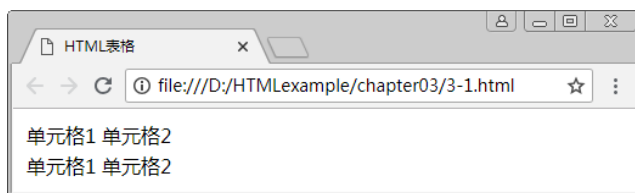


图 3.3 表格标签展示效果

表格内除了可以添加文本外，还可以添加其他标签元素，如：图片、列表、段落等。接下来通过案例来演示，如例 3-2 所示。

【例 3-2】 在表格内添加其他标签元素的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表格</title>
6 </head>
7 <body>
8 <table>
9   <tr>
10    <td></td>
11    <td><p>今天天气晴，温度适宜，适合出行。</p></td>
12  </tr>
13  <tr>
14    <td></td>
15    <td>今天有雨，出门记得带伞。</td>
16  </tr>
17 </table>
18 </body>
19 </html>
```

运行结果如图 3.4 所示。

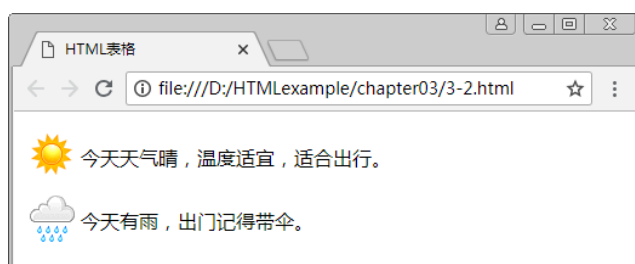


图 3.4 带图片和段落的表格展示效果

例 3-1 和例 3-2 在默认情况下表格是没有边框的。如果想要给表格添加边框，可以

设置表格的 border 属性，数值为边框的宽度。接下来通过案例来演示，如例 3-3 所示。

【例 3-3】 设置表格边框的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表格</title>
6 </head>
7 <body>
8 <table border="1">
9   <tr>
10     <td></td>
11     <td><p>今天天气晴，温度适宜，适合出行。</p></td>
12   </tr>
13   <tr>
14     <td></td>
15     <td>今天有雨，出门记得带伞。</td>
16   </tr>
17 </table>
18 </body>
19 </html>
```

运行结果如图 3.5 所示。

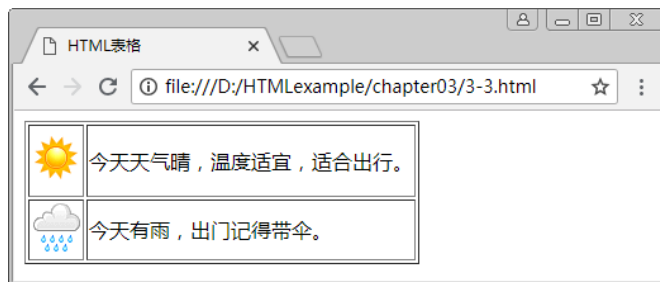


图 3.5 带边框的表格展示效果

3.1.2 表头与标题

为了使表格的格式更清晰方便读者查阅，应用表格时经常需要为表格设置表头，表头<th>是<td>单元格的一种标题，其本质还是一种单元格，一般位于表格的第一行或第一列，用来表明这一行或列的内容类别。浏览器会将表头默认以粗体居中的样式显示在网页中。接下来通过案例来演示表格的表头，如例 3-4 所示。

【例 3-4】 设置表格的表头的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>表头与标题</title>
6 </head>
7 <body>
8 <table border="1">
9   <tr>
10    <th>天气现象</th>
11    <th>出行情况</th>
12  </tr>
13  <tr>
14    <td></td>
15    <td><p>今天天气晴，温度适宜，适合出行。</p></td>
16  </tr>
17  <tr>
18    <td></td>
19    <td>今天有雨，出门记得带伞。</td>
20  </tr>
21 </table>
22 </body>
23 </html>
```

运行结果如图 3.6 所示。

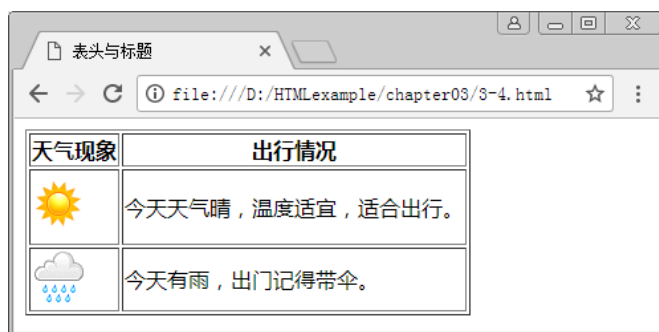


图 3.6 带表头的表格展示效果

`<th>`标签和`<td>`在本质上都是单元格，但这两种不可以互换使用。`th`，即“table header（表头单元格）”。而`td`，即“table data（单元格）”。

表格一般都有一个标题，用来表明表格的内容，一般位于整个表格的第一行，使用`<caption>`标签。一个表格只能含有一个表格标题。接下来通过案例来演示表格标题，如例 3-5 所示。

【例 3-5】 设置表格标题的演示案例。

```

1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>表头与标题</title>
6  </head>
7  <body>
8  <table border="1">
9      <caption>天气预报</caption>
10     <tr>
11         <th>天气现象</th>
12         <th>出行情况</th>
13     </tr>
14     <tr>
15         <td></td>
16         <td><p>今天天气晴，温度适宜，适合出行。</p></td>
17     </tr>
18     <tr>
19         <td></td>
20         <td>今天有雨，出门记得带伞。</td>
21     </tr>
22 </table>
23 </body>
24 </html>

```

运行结果如图 3.7 所示。

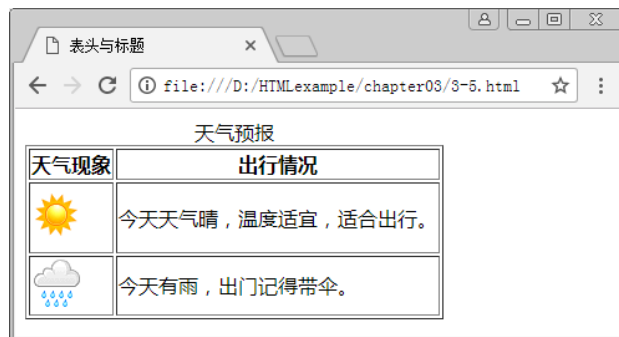


图 3.7 带标题的表格展示效果

3.1.3 表格语义化

为了使网页内容更好地被搜索引擎理解，在使用表格进行布局时，HTML 中引入了 `<thead>`、`<tbody>` 和 `<tfoot>` 这三个语义化标签，用来将表格划分为头部、主体和页脚三

部分。用这三个部分来定义网页中不同的内容，三个标签的详细理解如下。

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

接下来通过案例来演示表格语义化的三个标签，如例 3-6 所示。

【例 3-6】 定义表格语义化的三个标签的演示案例。

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>表格语义化</title>
6  </head>
7  <body>
8  <table border="1">
9      <caption>天气预报</caption>
10     <thead>
11         <tr>
12             <th>天气现象</th>
13             <th>出行情况</th>
14         </tr>
15     </thead>
16     <tbody>
17         <tr>
18             <td></td>
19             <td><p>今天天气晴，温度适宜，适合出行。</p></td>
20         </tr>
21         <tr>
22             <td></td>
23             <td>今天有雨，出门记得带伞。</td>
24         </tr>
25     </tbody>
26     <tfoot>
27     </tfoot>
28 </table>
29 </body>
30 </html>
```

运行结果如图 3.8 所示。

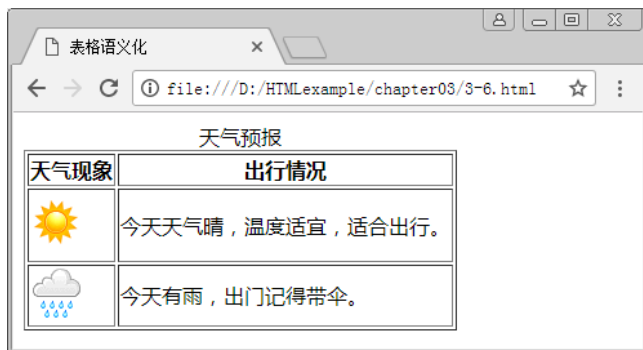


图 3.8 语义化的表格展示效果

由例 3-6 可以发现, 语义化表格效果上并没有什么变化, `<thead>`、`<tbody>`和`<tfoot>`三个标签不带任何效果, 只是更好地提供了语义化的功能, 根据表格的需求尽量添加相应的语义化标签。需要注意`<thead>`、`<tfoot>`这两个标签在一个`<table>`标签中只能出现一次, 而`<tbody>`标签可以出现多次。

3.1.4 合并行与列

在 Word 中设计表格时, 有时需要将两个或多个的相邻单元格组合成一个单元格, 即合并单元格的操作。在 HTML 中, 也需要用到“表格合并行”和“表格合并列”。合并行使用`<td>`标签的`rowspan`属性, 而合并列则用到`<td>`标签的`colspan`属性。接下来通过案例来演示表格合并列, 如例 3-7 所示。

【例 3-7】 设置表格合并列的演示案例。

```

1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>合并行与列</title>
6 </head>
7 <body>
8 <table border="1">
9     <caption>天气预报</caption>
10    <thead>
11    <tr>
12        <th colspan="2">日期</th>
13        <th>天气现象</th>
14        <th>出行情况</th>
15    </tr>
16    </thead>
17    <tbody>
18    <tr>

```

```

19         <td>22 日星期五</td>
20         <td>白天</td>
21         <td></td>
22         <td><p>今天天气晴，温度适宜，适合出行。</p></td>
23     </tr>
24     <tr>
25         <td>22 日星期五</td>
26         <td>夜间</td>
27         <td></td>
28         <td><p>今天有雨，出门记得带伞。</p></td>
29     </tr>
30 </tbody>
31 <tfoot>
32 </tfoot>
33 </table>
34 </body>
35 </html>

```

运行结果如图 3.9 所示。



图 3.9 和并列的表格展示效果

例 3-7 中，第 12 行设置 `colspan` 属性值为 2，用来合并表格中的两列单元格。接下来通过案例来演示合并行，如例 3-8 所示。

【例 3-8】 设置合并行的演示案例。

```

1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>合并行与列</title>
6 </head>
7 <body>
8 <table border="1">

```

```

9      <caption>天气预报</caption>
10     <thead>
11     <tr>
12         <th colspan="2">日期</th>
13         <th>天气现象</th>
14         <th>出行情况</th>
15     </tr>
16     </thead>
17     <tbody>
18     <tr>
19         <td rowspan="2">22 日星期五</td>
20         <td>白天</td>
21         <td></td>
22         <td><p>今天天气晴，温度适宜，适合出行。</p></td>
23     </tr>
24     <tr>
25         <td>夜间</td>
26         <td></td>
27         <td><p>今天有雨，出门记得带伞。</p></td>
28     </tr>
29     </tbody>
30     <tfoot>
31     </tfoot>
32 </table>
33 </body>
34 </html>

```

运行结果如图 3.10 所示。



图 3.10 合并行的表格展示效果

在例 3-8 中，第 19 行设置 `rowspan` 属性值为 2，即合并表格中的两行单元格。

3.1.5 单元格边距与间距

表格还有用于控制单元格边距与间距的属性，即 `cellpadding` 和 `cellspacing` 属性。

`cellpadding` 属性用于设置单元格内容与单元格边框之间的空白间距，默认为 1px；`cellspacing` 属性用于设置单元格与单元格边框之间的空白间距，默认为 2px。接下来通过案例来演示 `cellpadding` 属性，如例 3-9 所示。

【例 3-9】 设置 `cellpadding` 属性的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>单元格边距与间距</title>
6 </head>
7 <body>
8 <table border="1" cellpadding="30">
9   <tr>
10    <td>单元格 1</td>
11    <td>单元格 2</td>
12  </tr>
13  <tr>
14    <td>单元格 1</td>
15    <td>单元格 2</td>
16  </tr>
17 </table>
18 </body>
19 </html>
```

运行结果如图 3.11 所示。



图 3.11 cellpadding 属性展示效果

例 3-9 中，第 8 行将 `cellpadding` 属性设置为 30，即单元格内容和单元格边框之间的空白间距为 30。

3.1.6 表格其他属性

HTML 为表格提供了一系列的属性，用于控制表格的显示样式，除上面 `<table>` 标签

的属性，还有很多控制单元格和单元格内容的属性，具体如表 3.1 所示。

表 3.1 <table>标签常用属性

属性名	含义	属性值
width	设置表格的宽度	像素值
height	设置表格的高度	像素值
align	设置单元格内容的水平对齐方式	left (左对齐)、center (居中对齐)、right (右对齐)
valign	设置单元格内容的垂直对齐方式	baseline (基线对齐)、top (上对齐)、middle (居中对齐)、bottom (下对齐)

1. width 属性

width 属性可以设置单元格的宽度，当一列有多个不同 width 属性值时，取最大值作为这一列的宽度。接下来通过案例来演示 width 属性，如例 3-10 所示。

【例 3-10】 设置 width 属性的演示案例。

```

1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>表格其他属性</title>
6 </head>
7 <body>
8 <table border="1">
9   <tr>
10     <td width="200">单元格 1</td>
11     <td>单元格 2</td>
12   </tr>
13   <tr>
14     <td width="100">单元格 1</td>
15     <td>单元格 2</td>
16   </tr>
17 </table>
18 </body>
19 </html>

```

运行结果如图 3.12 所示。

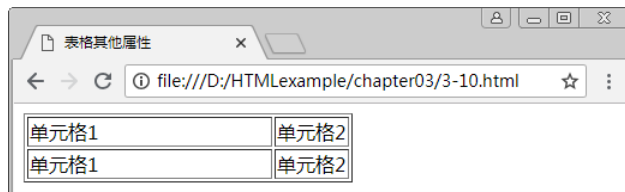


图 3.12 带宽度值的表格展示效果

2. height 属性

height 属性可以设置单元格的高度，当一行有多个不同 height 属性值时，取最大值作为这一行的高度。接下来通过案例来演示 height 属性，如例 3-11 所示。

【例 3-11】 设置 height 属性的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>表格其他属性</title>
6 </head>
7 <body>
8 <table border="1">
9   <tr>
10    <td height="100">单元格 1</td>
11    <td>单元格 2</td>
12  </tr>
13  <tr>
14    <td height="50">单元格 1</td>
15    <td>单元格 2</td>
16  </tr>
17 </table>
18 </body>
19 </html>
```

运行结果如图 3.13 所示。

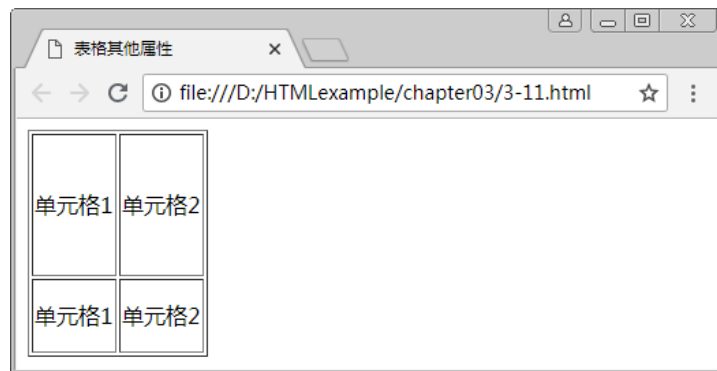


图 3.13 带高度值的表格展示效果

3. align 属性

align 属性可以设置单元格的内容左右对齐方向，<th>标签的 align 属性默认为 center，

<td>标签的 align 属性默认为 left。<table>标签也有 align 属性，用来设置表格在网页中的水平对齐方式。接下来通过案例来演示，如例 3-12 所示。

【例 3-12】 单元格和单元格内容使用 align 属性的显示效果的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>表格其他属性</title>
6 </head>
7 <body>
8 <table border="1">
9     <tr>
10        <td width="200" align="center">单元格 1</td>
11        <td>单元格 2</td>
12    </tr>
13    <tr>
14        <td width="100" align="right">单元格 1</td>
15        <td>单元格 2</td>
16    </tr>
17 </table>
18 </body>
19 </html>
```

运行结果如图 3.14 所示。

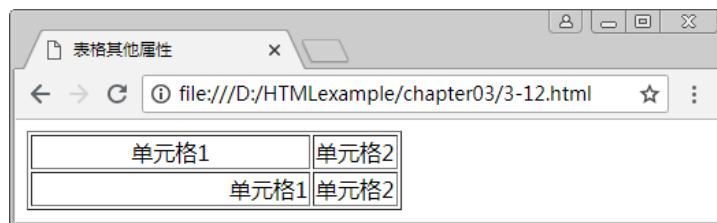


图 3.14 带左右对齐的表格展示效果

4. valign 属性

valign 属性可以设置单元格的内容垂直对齐的方向，默认为 center(居中对齐)。接下来通过案例来演示 valign 属性的使用，如例 3-13 所示。

【例 3-13】 valign 属性使用的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
```

```
5 <title>表格其他属性</title>
6 </head>
7 <body>
8 <table border="1">
9     <tr>
10         <td height="200" valign="top">单元格 1</td>
11         <td height="100" valign="bottom">单元格 2</td>
12     </tr>
13     <tr>
14         <td>单元格 1</td>
15         <td>单元格 2</td>
16     </tr>
17 </table>
18 </body>
19 </html>
```

运行结果如图 3.15 所示。



图 3.15 带上下对齐的表格展示效果

到此已经学习了 HTML 表格的常用标签与属性,如果想要做出一些更加漂亮或是更多需求的表格,还需要配合 CSS 才能完成,这里不作讲解,后续章节中会详细介绍如何实现效果。

3.2 HTML 表单

表单是网页中常用的一种展示效果,如登录页面中的用户名和密码的输入、登录按钮等都是用表单相关的标签定义的。表单是 HTML 中获取用户输入的手段,它的主要功

能是收集用户的信息，并将这些信息传递给后台服务器，实现网页与用户的交流。本节将详细讲解表达使用，先来观察微博登录、注册页面表单的展示效果，如图 3.16 所示。

图 3.16 微博登录、注册展示效果

HTML 中，一个完整的表单通常由表单元素、提示信息和表单域三个部分组成，下面将详细介绍这三个部分。

- 表单元素：包含表单的具体功能项，如文本输入框、下拉列表框、复选框、密码输入框、登录按钮等。
- 提示信息：表单中通常还需包含一些说明性的文字，提示用户要进行的操作。
- 表单域：用来容纳表单控件和提示信息，可以通过它定义处理表单数据所用程序的 URL 地址，以及数据提交到服务器的方法。如果未定义表单域，表单中的数据就无法传送到后台服务器。

表单元素是表单的核心，常用的表单元素如表 3.2 所示。

表 3.2 表单元素

表 单 元 素	含 义
<input>	表单输入框（可定义多种表单项）
<textarea>	定义多行文本框
<select>	定义一个下拉列表（必须包含列表项）
<label>	定义表单辅助项

这里先简单了解一下常用的表单元素，后面的小节将会进行详细讲解。

3.2.1 <form>标签

为了实现网页与用户的交流，需要让表单中的数据传送给后台服务器，这就必须定义表单域。定义表单域用<table>标签定义表格类似，HTML 中<form>标签用于定义表单域，即创建一个表单，用来实现用户信息的收集和传递，<form></form>标签中的所有内

容都会被提交给服务器。其语法格式如下：

```
<form 属性 属性值>
    表单元素和提示信息
</form>
```

接下来通过案例来演示<form>标签的简单使用，如例 3-14 所示。

【例 3-14】 <form>标签的简单使用的演示案例。

```
1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>HTML 表单</title>
6  </head>
7  <body>
8  <form>
9      姓名:<input type="text">
10     性别: <input type="radio"> 男<input type="radio">女
11     <input type="submit" value="提交">
12 </form>
13 </body>
14 </html>
```

运行结果如图 3.17 所示。

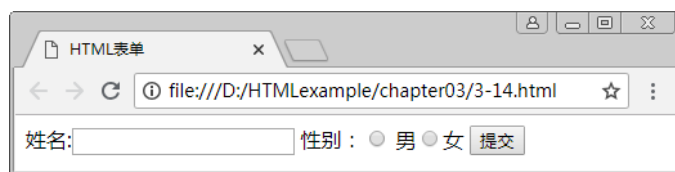


图 3.17 <form>标签展示效果

由图 3.17 可以看出，<form>标签默认情况下并没有什么效果，当输入完姓名和选择完性别后，单击提交按钮，就可以把填写好的数据提交给后台服务器，服务器经过处理后，将数据存储进网站的数据库，这样数据就可以得到保存。

HTML 表单将数据发送给后台服务器，用到<form>的 action、method、enctype 和 target 属性，下面详细了解这四个属性。

1. action 属性

action 属性是用来定义表单数据的提交地址，即一个 URL。HTML 表单要想和后台服务器进行连接，就需要在 action 属性上设置一个 URL。比如两人打电话，双方要通话就必须要知道对方的电话号码，URL 就相当于电话号码。action 属性用于指定接收并处理表单数据的服务器的 URL 地址。具体示例如下。

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

表示提交表单时，表单数据会传送到 `qianfeng_action.asp` 的页面处理。

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

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

表示提交表单时，表单数据以电子邮件的形式传递出去。

2. method 属性

`method` 属性是用来定义表单数据的提交方式，常用的有 `get`（默认）和 `post` 两种方式。提交方式类似于通信方式，可以打电话、发短信或发邮件。一般情况下，获取一些数据用 `get` 方式，这种方式提交的数据将显示在浏览器的地址栏中，保密性差，且有数据量的限制。`post` 方式的保密性好，而且无数据量的限制，使用 `method="post"` 可以大量提交数据。

3. enctype 属性

`enctype` 属性是用来定义表单数据的提交内容形式，常用的有 `application/x-www-form-urlencoded`（默认）和 `multipart/form-data` 两种方式。提交内容可以是网页中的文本，也可以是图片或视频等非文本的内容，因此需要对 `enctype` 属性选择不同的设置。

4. target 属性

`target` 属性是用来定义提交地址的打开方式，常用的有 `_self`（默认）和 `_blank` 两种方式。打开方式可以选择当前页打开，也可以在新页面打开，`<form>` 标签中的 `target` 属性跟 `<a>` 标签中的 `target` 属性一样，这里不再赘述。

接下来通过案例来演示 `<form>` 标签的四个常用属性，如例 3-15 所示。

【例 3-15】 `<form>` 标签的四个常用属性的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form action="demo.html" method="post" enctype="multipart/form-data"
9     target="_blank">
10     姓名:<input type="text">
11     性别: <input type="radio"> 男<input type="radio">女
12     <input type="submit" value="提交">
13 </form>
```

```
14 </body>
15 </html>
```

运行结果如图 3.18 所示。



图 3.18 <form>标签展示效果

例 3-15 中对这些属性并没有做过多的解释与演示,对后端有一定的了解后,才能更好地掌握<form>标签的属性,这里先了解一下这些属性,然后再学习一些后端知识,那么这些内容就非常容易理解了。

3.2.2 <input>标签

网页中经常会包含有单行文本框、单选按钮、复选框、提交按钮等,要想定义这些表单元素需要使用<input>标签,其基本语法格式如下:

```
<input type="元素类型">
```

1. type 属性

<input>标签通过 type 属性取值不同,可以展示出不同的表单类型,其属性取值如表 3.3 所示。

表 3.3 type 属性取值

表 单 元 素	含 义
text	单行文本框
password	密码文本框
radio	单选框
checkbox	复选框
button	按钮
submit	提交按钮
reset	重置按钮
hidden	隐藏域
image	图像形式的按钮
file	文件上传按钮

表 3.3 中列出了 type 属性的取值及含义，下面将分别讲解这些属性。

(1) text、password

text 值用来展示单行文本框，password 值用来展示密码文本框，一般用于登录界面，接下来通过案例来演示这两个属性值，如例 3-16 所示。

【例 3-16】 text 和 password 的演示案例。

```

1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>HTML 表单</title>
6  </head>
7  <body>
8  <form>
9      用户名:<input type="text">
10     密码: <input type="password">
11 </form>
12 </body>
13 </html>

```

运行结果如图 3.19 所示。

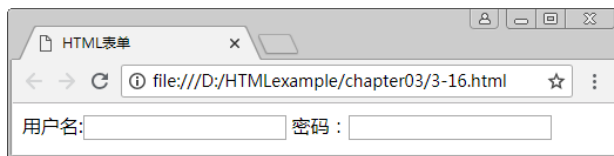


图 3.19 text、password 展示效果

可以往输入框内输入内容，password 值中的字符会被做掩码处理（显示为星号或实心圆），如图 3.20 所示。

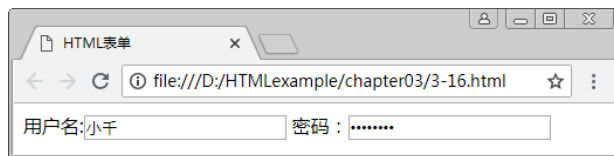


图 3.20 text、password 输入展示效果

(2) radio、checkbox

radio 值用来展示单选框，checkbox 值用来展示复选框，一般在调查问卷中使用，接下来通过案例来演示这两个属性，如例 3-17 所示。

【例 3-17】 radio 和 checkbox 的演示案例。

```

1  <!doctype html>

```

```
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form>
9     性别: <input type="radio" name="gender">男
10         <input type="radio" name="gender">女
11     爱好: <input type="checkbox">音乐
12         <input type="checkbox">体育
13         <input type="checkbox">舞蹈
14 </form>
15 </body>
16 </html>
```

运行结果如图 3.21 所示。



图 3.21 radio、checkbox 展示效果

例 3-17 中,第 9 行和第 10 行单选框加了一个 name 属性,并且两个单选框中的 name 属性值相同,目的是让多个单选框之间建立关系,这样就可以对单选框进行切换操作,因此在属性值为单选框时一定要加 name 属性,否则单选框不能切换状态。可以选择单选框和复选框,其效果如图 3.22 所示。

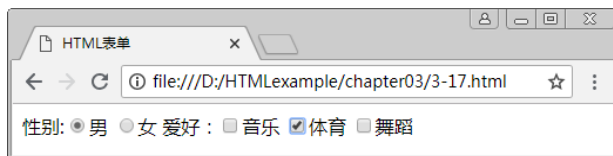


图 3.22 radio、checkbox 展示效果

(3) button、submit、reset

button 值用来设置普通按钮, submit 值用来设置提交按钮, reset 值用来设置重置按钮。接下来通过案例来演示这三个属性值,如例 3-18 所示。

【例 3-18】 button、submit、reset 三个属性值的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
```

```

4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form>
9     <input type="button" value="普通按钮">
10    <input type="submit" value="提交按钮">
11    <input type="reset" value="重置按钮">
12 </form>
13 </body>
14 </html>

```

运行结果如图 3.23 所示。

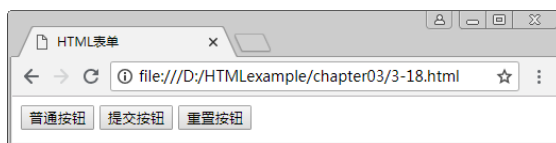


图 3.23 button、submit、reset 展示效果

例 3-18 中，第 9 行、第 10 行和第 11 行中的 value 属性的作用是设置按钮上文本的内容。

普通按钮没有任何行为，常用于在用户单击按钮时启动 JavaScript 程序。提交按钮可以看成一种具有特殊功能的普通按钮，单击提交按钮可以实现将表单内容提交给后台服务器处理。重置按钮也可以看成一种具有特殊功能的普通按钮，单击重置按钮可以清除用户在页面表单中输入的信息。接下来通过案例来演示这三个按钮的展示效果，如例 3-19 所示。

【例 3-19】 button、submit、reset 三个按钮展示效果的演示案例。

```

1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form action="data.php">
9     <input type="text">
10    <input type="button" value="普通按钮" onclick="alert('hello')">
11    <input type="submit" value="提交按钮">
12    <input type="reset" value="重置按钮">
13 </form>
14 </body>
15 </html>

```

运行结果如图 3.24 所示。

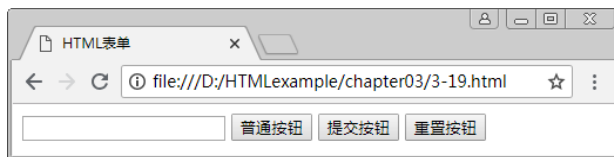


图 3.24 button、submit、reset 展示效果

单击普通按钮时，会弹出一个对话框，里面的内容为“hello”。单击提交按钮，会跳转到 data.php 页面（这需要了解后端技术才可理解）。当往输入框中输入一些内容，然后再单击重置按钮，发现内容被清空。

(4) hidden

hidden 值用于隐藏那些只是往后台服务器发送一些数据，但又不影响页面布局的表单控件。接下来通过案例来演示 hidden 属性值，如例 3-20 所示。

【例 3-20】 hidden 属性值的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form action="data.php">
9     <input type="hidden" name="gender" value="男">
10    <input type="submit" value="提交按钮">
11 </form>
12 </body>
13 </html>
```

运行结果如图 3.25 所示。

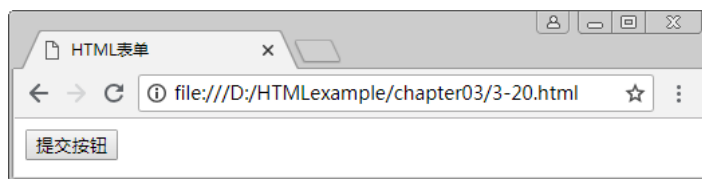


图 3.25 hidden 展示效果

第 9 行 type 属性值设置为 hidden，隐藏域在网页中并没有显示出来，单击提交按钮，就可以把 value=“男”提交给 data.php 这个后台服务器。

(5) image

image 值用来设置图像形式的按钮，src 属性用来引入图像的地址，目的是替换 submit

的默认样式，从而完成更加美观的展示。接下来通过案例来演示 image 属性值，如例 3-21 所示。

【例 3-21】 image 属性值的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form action="data.php">
9     <input type="hidden" name="gender" value="男">
10    <!--<input type="submit" value="提交按钮">-->
11    <input type="image" src="login.jpg">
12 </form>
13 </body>
14 </html>
```

运行结果如图 3.26 所示。

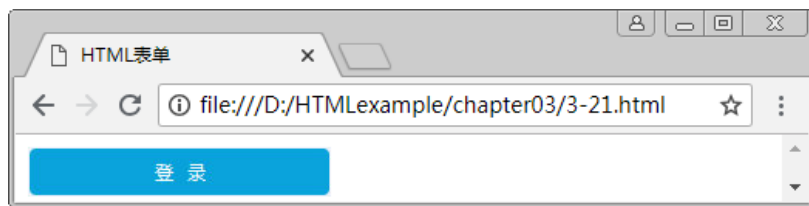


图 3.26 image 展示效果

(6) file

file 值用来设置文件上传的按钮，文件上传是网站中常见的功能，例如网盘文件上传和邮箱文件上传。设置 file 值时，<form>标签的 method 属性必须设置成 post，enctype 属性必须设置成 multipart/form-data。接下来通过案例来演示 file 值，如例 3-22 所示。

【例 3-22】 file 值的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form action="data.php" method="post"
9     enctype="multipart/form-data">
```

```

10     <input type="file">
11 </form>
12 </body>
13 </html>

```

运行结果如图 3.27 所示。

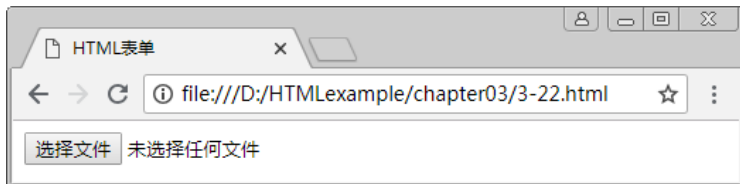


图 3.27 file 展示效果

网页中经常能看到一些漂亮的上传按钮，这些按钮都是通过 CSS 来实现的，后面的章节将会详细讲解。

2. 其他属性

`<input>` 标签除了 `type` 属性，还有一些常用的属性，如表 3.4 所示。

表 3.4 `<input>` 标签其他属性

属 性	属 性 值	含 义
name	自定义	元素的名称
value	自定义	元素的值
maxlength	正整数	元素允许输入的最多字符数
disabled	disabled	第一次加载页面时禁用该元素（显示为灰色）
readonly	readonly	元素内容为只读（不能修改编辑）
checked	checked	定义选择元素默认被选中的项

表 3.4 中列出了 `<input>` 标签中其他常用的属性，下面进行详细讲解。

(1) name、value 属性

`name` 属性用来规定 `input` 元素的名称，`value` 属性用来规定 `input` 元素的值。在前面的案例中，已经接触过 `name` 和 `value` 这两个属性，其中 `name` 值和 `value` 值配合成一对来使用，这样后台服务器就可以通过 `name` 值来找到对应的 `value` 值。接下来通过案例来演示这两个属性，如例 3-23 所示。

【例 3-23】 `name`、`value` 属性的演示案例。

```

1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>

```

```

7 <body>
8 <form action="data.php">
9     姓名: <input type="text" name="myName" value="myValue">
10    性别: <input type="radio" name="gender" value="man">男
11        <input type="radio" name="gender" value="woman">女
12 </form>
13 </body>
14 </html>

```

运行结果如图 3.28 所示。

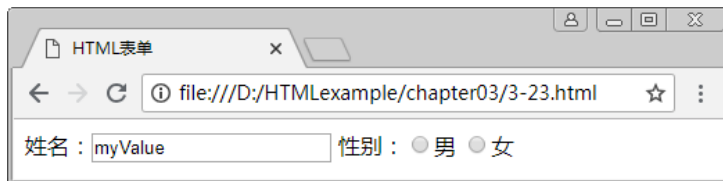


图 3.28 name、value 属性展示效果

(2) maxlength 属性

maxlength 属性规定输入内容允许的最大字符数，如设置 maxlength 属性，则输入控件不会接受超过其所允许字符数。接下来通过案例来演示 maxlength 属性，如例 3-24 所示。

【例 3-24】 maxlength 属性的演示案例。

```

1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form>
9     <input type="text" maxlength="10" minlength="3">最多输入 10 个字符
10 </form>
11 </body>
12 </html>

```

运行结果如图 3.29 所示。



图 3.29 maxlength 属性展示效果

(3) disabled、readonly 属性

disabled 属性规定输入内容是禁用的,被禁用的元素是不可用和不可单击的。readonly 属性规定输入内容为只读(不能修改,但是能获取当前只读的内容)。接下来通过案例来演示这两个属性,如例 3-25 所示。

【例 3-25】 disabled、readonly 属性的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form>
9     <input type="text" value="HTML" disabled>
10    <input type="text" value="HTML" readonly>
11    <input type="checkbox" disabled>
12 </form>
13 </body>
14 </html>
```

运行结果如图 3.30 所示。

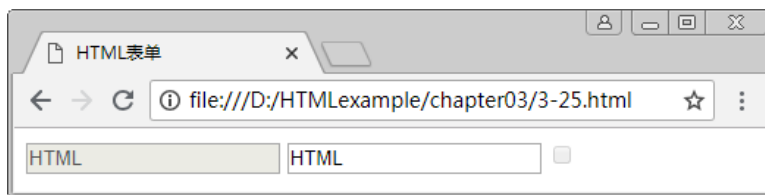


图 3.30 disabled、readonly 属性展示效果

(4) checked 属性

checked 属性规定在页面加载时应该预先选定的 input 元素。checked 属性与<input type="checkbox">或<input type="radio">配合使用。接下来通过案例来演示 checked 属性,如例 3-26 所示。

【例 3-26】 checked 属性的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
```

```

8  <form>
9      性别: <input type="radio" name="gender" checked>男
10     <input type="radio" name="gender" >女
11     爱好: <input type="checkbox" checked>音乐
12           <input type="checkbox" checked>体育
13           <input type="checkbox">舞蹈
14 </form>
15 </body>
16 </html>

```

运行结果如图 3.31 所示。

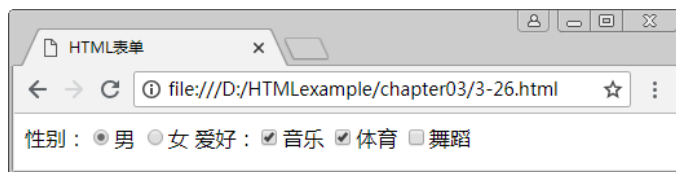


图 3.31 checked 属性展示效果

在后面的章节中，还会学习标签新的 type 属性值和标签其他新的属性，本节只对 input 中常用的元素进行讲解。

3.2.3 <textarea>标签

单行文本框只能输入一行信息，而多行文本框可以输入多行信息。多行文本框使用的是<textarea>标签，而标签只能设置单行文本框。接下来通过案例来演示<textarea>标签，如例 3-27 所示。

【例 3-27】 <textarea>标签的演示案例。

```

1  <!doctype html>
2  <html>
3  <head>
4  <meta charset="utf-8">
5  <title>HTML 表单</title>
6  </head>
7  <body>
8  <form>
9      <textarea rows="10" cols="30">多行文本框内容</textarea>
10 </form>
11 </body>
12 </html>

```

运行结果如图 3.32 所示。

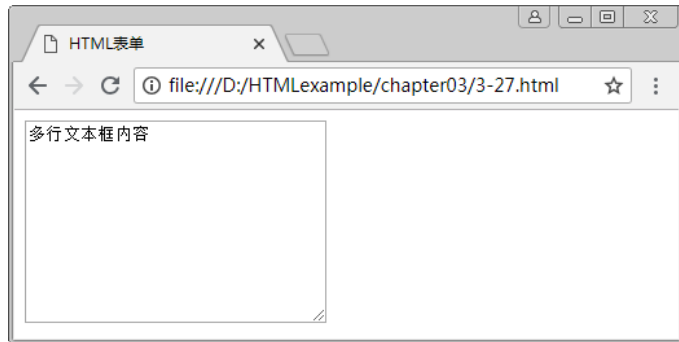


图 3.32 textarea 标签展示效果

例 3-27 中, rows 属性可以设置多行文本框的行数, cols 属性可以设置多行文本框的列数。这样就可以定义多行文本框的尺寸, 更好的办法是使用 CSS 的 height 和 width 属性来定义多行文本输入框的宽高。

3.2.4 <select>标签

网页中经常会看到包含多个选项的下拉菜单, 如选择城市、日期、科目、校区选择等。HTML 中用<select>标签设置下拉列表, 其需要与<option>标签配合使用, 这个特点和列表一样, 如无序列表是由标签和标签配合使用。为了更好地理解, 可以把下拉列表看作一个特殊的无序列表。

下拉列表是一种最节省页面空间的选择方式, 因为在正常状态下只显示一个选项, 单击下拉菜单打开菜单后才会看到全部的选项。接下来通过案例来演示<select>标签, 如例 3-28 所示。

【例 3-28】 <select>标签的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form>
9     <select>
10         <option>HTML</option>
11         <option>CSS</option>
12         <option>JavaScript</option>
13         <option>PHP</option>
14     </select>
15 </form>
```

```

16 </body>
17 </html>

```

运行结果如图 3.33 所示。

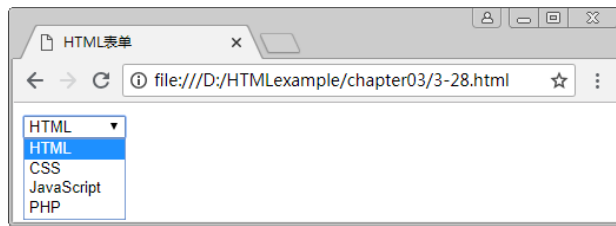


图 3.33 select 标签点开展示效果

和其他标签一样，<select>标签也有其常用的属性，如表 3.5 所示。

表 3.5 type 属性取值

属 性	含 义
multiple	多选操作
size	下拉列表可见选项的数目
selected	选中项

1. multiple 属性

multiple 属性可以设置多选下拉列表，默认下拉列表只能选择一项，而设置 multiple 属性后就可以选择多项了（使用“Ctrl+鼠标左键”进行多选操作）。接下来通过案例来演示 multiple 属性，如例 3-29 所示。

【例 3-29】 multiple 属性的演示案例。

```

1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form>
9   <select multiple>
10     <option>HTML</option>
11     <option>CSS</option>
12     <option>JavaScript</option>
13     <option>PHP</option>
14   </select>
15 </form>
16 </body>
17 </html>

```

运行结果如图 3.34 所示。

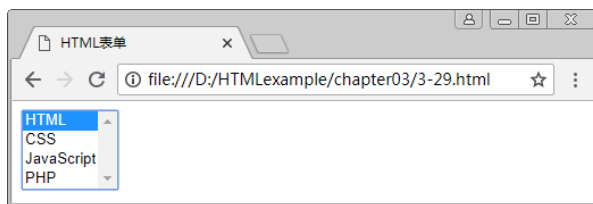


图 3.34 multiple 属性展示效果

2. size 属性

size 属性可以设置下拉列表可见选项的数目，默认情况下单选下拉菜单显示一项。接下来通过案例来演示 size 属性，如例 3-30 所示。

【例 3-30】 size 属性的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form>
9     <select size="2">
10         <option>HTML</option>
11         <option>CSS</option>
12         <option>JavaScript</option>
13         <option>PHP</option>
14     </select>
15 </form>
16 </body>
17 </html>
```

运行结果如图 3.35 所示。

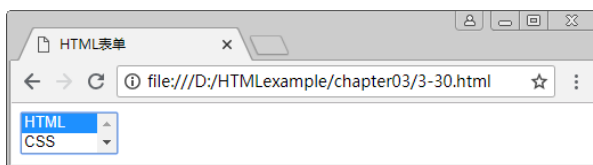


图 3.35 size 属性展示效果

3. selected 属性

selected 属性表示选中项，与单选框的 checked 属性类似，注意 selected 属性是设置

到<option>标签上的。接下来通过案例来演示 selected 属性，如例 3-31 所示。

【例 3-31】 selected 属性的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form>
9     <select>
10         <option>HTML</option>
11         <option selected>CSS</option>
12         <option>JavaScript</option>
13         <option>PHP</option>
14     </select>
15 </form>
16 </body>
17 </html>
```

运行结果如图 3.36 所示。

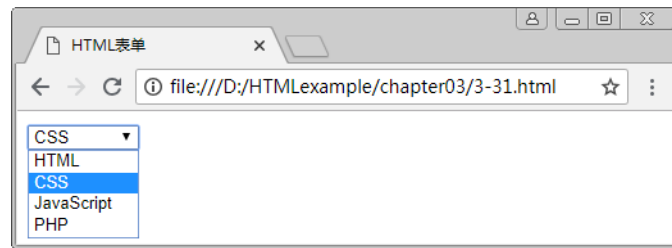


图 3.36 selected 属性展示效果

<select>标签中使用<optgroup>标签进行分组项操作，把相关的选项组合在一起。<optgroup>标签的 label 属性来设置分组项的标题。接下来通过案例来演示<optgroup>标签，如例 3-32 所示。

【例 3-32】 <optgroup>标签的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
8 <form>
```

```
9     <select>
10     <optgroup label="前端技术">
11     <option>HTML</option>
12     <option selected>CSS</option>
13     <option>JavaScript</option>
14     </optgroup>>
15     <optgroup label="后端技术">
16     <option>PHP</option>
17     <option>JAVA</option>
18     <optgroup></optgroup>
19 </select>
20 </form>
21 </body>
22 </html>
```

运行结果如图 3.37 所示。

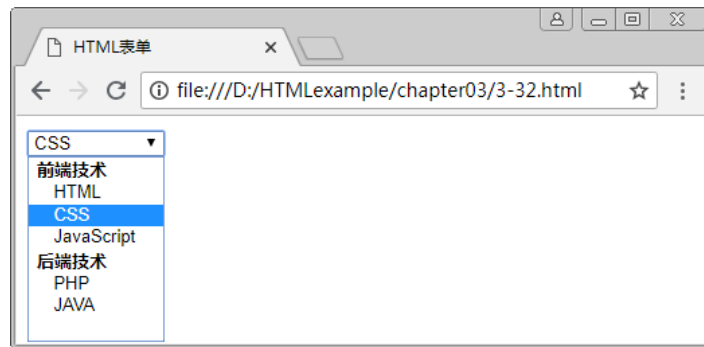


图 3.37 optgroup 标签展示效果

3.2.5 <label>标签

<label>标签用来辅助表单元素，可以更好地提高用户体验。当用户选择<label>标签内文本进行单击时，会自动将焦点转到和标签相关的表单控件上。接下来通过案例来演示<label>标签，如例 3-33 所示。

【例 3-33】 <label>标签的演示案例。

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>HTML 表单</title>
6 </head>
7 <body>
```

```

8 <form>
9   性别: <input type="radio" name="gender" id="man">
10   <label for="man">男</label>
11   <input type="radio" name="gender" id="woman">
12   <label for="woman">女</label>
13 </form>
14 </body>
15 </html>

```

运行结果如 3.38 所示。

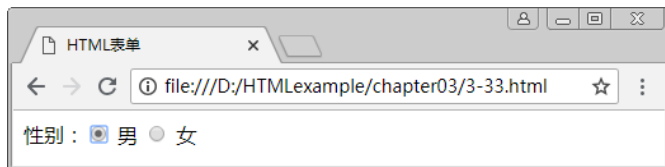


图 3.38 label 标签展示效果

当单击<label>标签中的文本（男、女），也可以对单选框进行切换，从而使用户体验得到提升。其中<label>标签中的 for 属性值一定要和<input>标签中的 id 属性值相同才能找到对应控件。

3.3 本章小结

通过本章的学习，能够掌握 HTML 表格和 HTML 表单的基本使用，了解前端与后台服务器之间如何交互与通信。

3.4 习 题

1. 填空题

- (1) 创建表格必备三个标签为_____、_____、_____。
- (2) 为表格添加边框，需要设置的属性为_____。
- (3) _____用于创建表格的标题。
- (4) valign 属性可以设置单元格的内容_____的方向。
- (5) 表单由_____、_____、_____三个部分组成。

2. 选择题

- (1) 在 HTML 中，下列的哪个可以产生单选框？（ ）

- A. `<input type="radio">` B. `<input type="check">`
C. `<input type="checkbox">` D. `<radio><radio>`
- (2) 下列的 HTML 中哪个可以产生文本域? ()
A. `<textarea></textarea>` B. `<input type="textarea">`
C. `<text></text>` D. `<textarea type="text">`
- (3) 以下元素不是 table 元素子元素的有 ()。
A. `<th>` B. `<tbody>`
C. `<thead>` D. `<td>`
- (4) 以下标签不能体现表格语义化的是 ()。
A. `<tfoot></tfoot>` B. `<thead></thead>`
C. `<table></table>` D. `<tbody></tbody>`
- (5) 以下标签中作用与`<form>`标签类似的是 ()。
A. `<table>` B. `<th>`
C. `<input>` D. `<label>`

3. 思考题

- (1) 请简述`<label>`标签的作用。
(2) 请简述`<select>`标签中 `multiple` 属性的作用。

