

## 第3章 HTML表格与表单



- HTML表格
- HTML表单



3.1

## HTML表格

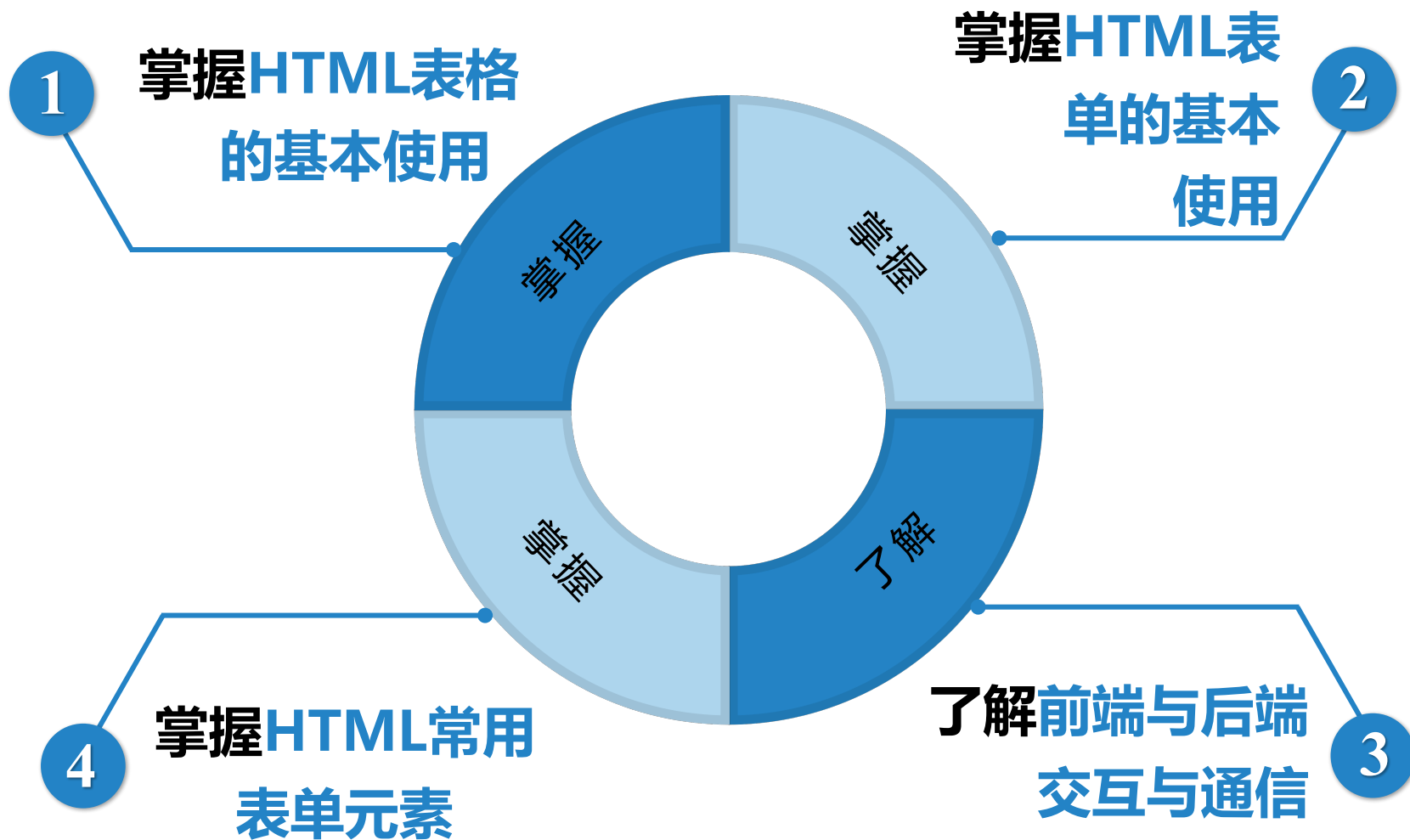
[点击查看本小节知识架构](#)

3.2

## HTML表单

[点击查看本小节知识架构](#)





## 3.1 HTML表格

[返回目录](#)

3.1.1

● 表格基本结构

3.1.2

● 表头与标题

3.1.3

● 表格语义化

3.1.4

● 合并行与列

3.1.5

● 单元格边距与间距

3.1.6

● 表格其他属性





## 3.1 HTML表格

### 3.1.1 表格的基本结构

- 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）。





## 3.1 HTML表格

### 3.1.1 表格的基本结构

- 表格内除了可以添加文本外，还可以添加其它标签元素，如：图片、列表、段落等。
- 默认情况下表格没有边框，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。





## 3.1 HTML表格

### 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。
- <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。





## 3.1 HTML表格

### 3.1.3 表格语义化

- 为了使网页内容更好的被搜索引擎理解，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。
- **<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。







## 3.1 HTML表格

### 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。





## 3.1 HTML表格

### 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。
- 下面通过案例演示cellpadding属性，如例所示。

```
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>
```



## 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，具体如表所示。

width		
height		
align		left(        )   enter(        )   right
valign		baseline                          top middle                          bottom





## 3.1 HTML表格

### 3.1.6 表格其他属性

- 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。
- 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。
- 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。





## 3.1 HTML表格

### 3.1.6 表格其他属性

- 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)。下面通过案例演示valign属性的使用。

```
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>
```





## 3.2 HTML表单



[返回目录](#)

3.2.1

● `<form>` 标签

3.2.2

● `<input>` 标签

3.2.3

● `<textarea>` 标签

3.2.4

● `<select>` 标签

3.2.5

● `<label>` 标签





## 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。
- 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。
- 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。





## 3.2 HTML表单

- 表单元素是表单的核心，常用的表单元素如表所示。

<input>	
<textarea>	
<select>	
<label>	







## 3.2 HTML表单

### 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，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，以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>
```

- 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。





## 3.2 HTML表单

### 3.2.1 <form>标签

- 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。





## 3.2 HTML表单

### 3.2.1 <form>标签

- 2. method属性
- method属性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。





## 3.2 HTML表单

### 3.2.1 <form>标签

- 4. target属性
- target属性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，如例所示。

```
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     《》
11     性别: <input type="radio"> 男<input type="radio">女
12     <input type="submit" value="提交">
13 </form>
14 </body>
15 </html>
```



## 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不同的表单类型，其属性值如表所示。

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





## 3.2 HTML表单

### 3.2.2 <input> 标签

- ( 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，一般使用在登录界面。
- ( 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。
- ( 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。





## 3.2 HTML表单

### 3.2.2 <input> 标签

- ( 4 ) hidden
- hidden值用于隐藏那些知识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。
- ( 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。
- ( 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。



### 3.2.2 <input> 标签

- 2. 其他属性
- <input> 标签除了type属性，还有一些常用的属性，如表所示。

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







## 3.2 HTML表单

### 3.2.2 <input> 标签

- ( 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。这里只需了解name和value这两个属性是为后台服务器提供即可。
- ( 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。





## 3.2 HTML表单

### 3.2.2 <input> 标签

- ( 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）。
- ( 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。





## 3.2 HTML表单

### 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<input>标签只能设置单行文本框。下面通过案例来演示<textarea>标签，如例所示。

```
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>
```



### 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<ul>标签和<li>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。
- 和其他标签一样，<select>标签也有其常用的属性，如表所示。

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





## 3.2 HTML表单

### 3.2.4 <select> 标签

- 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）。
- 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。
- 3.selected属性
- selected属性表示选中项，跟单选框的checked属性类似，注意selected属性是设置到<option>标签上的。





## 3.2 HTML表单

### 3.2.4 <select> 标签

- <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。下面通过案例演示<optgroup> 标签，如例所示。

```
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>
```





## 3.2 HTML表单

### 3.2.5 <label>标签

- <label>标签用来辅助表单元素，可以更好的提高用户体验。当用户选择<label>标签内文本进行点击时，会自动将焦点转到和标签相关的表单控件上。

下面通过案例演示，如例所示

```
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>
```





## 本章小结

- 通过本章的学习，能够掌握HTM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。





# THANK YOU

