

第2章 网页实现技术——HTML5

本章导读

HTML5 是制作网页的超文本标记语言，包含多种标签，标签不区分大小写，大部分标签是成对出现的。用 HTML5 编写的超文本文档被称为 HTML 文档，它能在各种浏览器上独立运行。本章就来介绍网页实现技术——HTML5 的应用，通过本章的学习，读者可以更加全面地了解 HTML5 中标签的使用方法与技巧。

知识导图

网页实现技术——HTML5





2.1 HTML5 中的常用标签

<html>、<head>、<body> 标签是构成 HTML 文档的 3 个不可缺少的标签，但除了这 3 个基本标签之外，还有其他一些常用标签，例如字符标签、超链接标签和列表标签等。

2.1.1 基础标签的应用

利用 HTML5 中的标签可以实现网页的简单设计，其中包括一些基础标签，如表 2-1 所示。

表 2-1 基础标签

标 签	描 述
<!DOCTYPE>	定义文档类型
<html>	定义一个 HTML 文档
<title>	为文档定义一个标题
<body>	定义文档的主体
<h1> to <h6>	定义 HTML 标题
<p>	定义一个段落
 	定义简单的折行
<hr>	定义水平线
<!--...-->	定义一个注释

下面利用 HTML5 基础标签实现一首古诗的混排。这里用到了一个 align 属性，该属性可以设置标题的对齐方式。语法格式如下：

```
<h1 align="对齐方式">文本内容</h1>
```

这里的对齐方式包括 left (文字左对齐)、center (文字居中对齐)、right (文字右对齐)。需要注意的是对齐方式一定要添加双引号。

实例 1：混合排版一首古诗

本实例通过 align="center" 来实现标题的居中效果，通过 align="right" 来实现标题的靠右效果，具体代码如下：

```
<!DOCTYPE html>
<html>
<head>
  <!--指定页头信息-->
  <title>古诗混排</title>
</head>
<body>
```

```
<!--显示古诗名称-->
<h2 align="center">望雪</h2>
<hr>
<!--显示作者信息-->
<h5 align="right">唐代：李世民</h5>
<!--显示古诗内容-->
<p style="font-size:18pt" align =
"center">冻云宵遍岭，素雪晓凝华。<br/>
入牖千重碎，迎风一半斜。<br/>
不妆空散粉，无树独飘花。<br/>
紫空惭夕照，破彩谢晨霞。<br/>
</body>
</html>
```

在浏览器中浏览的效果如图 2-1 所示，实现了古诗的排序，符合阅读的习惯。



图 2-1 混合排版古诗页面效果

2.1.2 文本格式标签

在 HTML5 文档中, 不管其内容如何变化, 文字始终是最小的单位。文本标签通常用来指定文字的显示方式和布局方式。常用的文本格式标签如表 2-2 所示。

表 2-2 常用的文本标签

标 签	描 述	标 签	描 述
<address>	定义文档作者或拥有者的联系信息	<pre>	定义预格式文本
	定义粗体文本	<progress>	定义运行中的任务进度(进程)
<bdo>	定义文本的方向	<s>	定义加删除线的文本
<blockquote>	定义块引用	<small>	定义小号文本
	定义被删除文本		定义语气更为强烈的强调文本
	定义强调文本	<sub>	定义下标文本
<i>	定义斜体文本	<sup>	定义上标文本
<kbd>	定义键盘文本	<time>	定义一个日期/时间
<mark>	定义带有记号的文本	<u>	定义下划线文本

实例 2: 文字的粗体、斜体和下划线效果

下面综合应用 标签、 标签、 标签、<i> 标签和 <u> 标签来设置文字的粗体、斜体和下划线显示效果, 这里还使用了 font-size 属性来设置文字的大小。

```
<!DOCTYPE html>
<html>
<head>
<title>文字的粗体、斜体和下划线</title>
</head>
<body>
<!--显示粗体文字效果-->
<p><b>吴兴自东晋为善地, 号为山水清远。其民足于鱼稻蒲莲之利, 寡求而不争。宾客非特有事于其地者不至焉。</b></p>
<!--显示强调文字效果-->
<p style="font-size:18pt"><em>故凡守郡者, 率以风流啸咏、投壶饮酒为事。</em></p>
<!--显示加强文字效果-->
<p><strong>自莘老之至, 而岁适大水, 上田皆不登, 湖人大饥, 将相率亡去。</strong></p>
<!--显示斜体字效果-->
<p style="font-size:18pt"><i>莘老大
```

```
振廩劝分, 躬自抚循劳来, 出于至诚。富有余者, 皆争出谷以佐官, 所活至不可胜计。</i></p>
<!--显示下划线效果-->
<p><u>当是时, 朝廷方更化立法, 使者旁午, 以为莘老当日夜治文书, 赴期会, 不能复雍容自得如故事。</u></p>
</body>
</html>
```

在浏览器中浏览的效果如图 2-2 所示, 实现了文字的粗体、斜体和下划线效果。



图 2-2 文字的粗体、斜体和下划线的预览效果

2.1.3 超级链接标签

链接是指从一个网页指向一个目标的链接关系。这个目标可以是另一个网页, 也可以是本网页的不同位置, 还可以是一个图片、一个电子邮件地址、一个文件, 甚至是一个应用程序。而在一个网页中作为超链接的对象, 可以是一段文本或者一个图片。如表 2-3 所示为

表 2-4 列表标签

标 签	描 述	标 签	描 述
	定义一个无序列表	<dt>	定义一个定义列表中的项目
	定义一个有序列表	<dd>	定义列表中项目的描述
	定义一个列表项	<menu>	定义菜单列表
<dl>	定义一个定义列表	<command>	定义用户可能调用的命令（比如单选按钮、复选框或按钮）

网页中的列表可以有序地编排一些信息资源，使其结构化和条理化，以便浏览者能更加快捷地获得相应信息。

实例 4：创建嵌套列表

本实例使用 标签和 标签，设计嵌套列表的样式。

```

<!doctype html>
<html>
<head>
<title>无序列表和有序列表嵌套</title>
</head>
<body>
<ul>
  <li><a href="#">课程销售排行
    榜</a>
    <ol>
      <li><a href="#">
        Python爬虫智能训练营</a></li>
      <li><a href="#">
        网站前端开发训练营</a></li>
      <li><a href="#">
        PHP网站开发训练营</a></li>
      <li><a href="#">
        网络安全对抗训练营</a></li>
    </ol>
  </li>
  <li><a href="#">学生区
    域分布</a>
    <ul>
      <li><a href="#">
        北京</a></li>

```

```

      <li><a href="#">
        上海</a></li>
      <li><a href="#">
        广州</a></li>
      <li><a href="#">
        郑州</a></li>
    </ul>
  </li>
</ul>
</body>
</html>

```

在浏览器中浏览的效果如图 2-4 所示。嵌套列表是网页中常用的元素，通过重复使用 标签和 标签，可以实现无序列表和有序列表的嵌套。

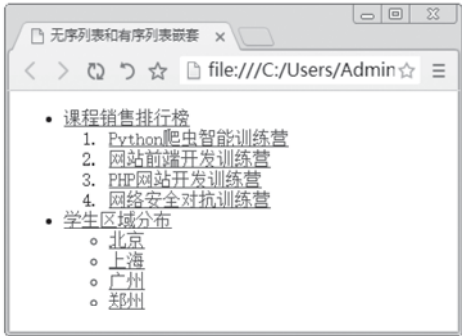


图 2-4 自定义网页列表

2.2 HTML5 中的图像标签

图像是网页中不可或缺的一个元素，在 HTML5 里，专门提供了一些用来处理图像的标签，如表 2-5 所示。



表 2-5 HTML5 中的图像标签

标 签	描 述
	定义图像
<map>	定义图像映射

图 2-5 替换文字和提示文字

注意：随着互联网技术的发展，网速已经不是制约因素，因此一般都能成功下载图像。但在百度、Google等大搜索引擎中，搜索图片没有文字方便，如果给图片添加适当提示，可以方便搜索引擎的检索。

2.2.2 定义图像热点区域

在HTML5中，可以为图像创建3种类型的热点区域：矩形、圆形和多边形。创建热点区域使用标签<map>和<area>。

设置图像热点链接大致可以分为两个步骤。

1. 设置映射图像

要想建立图片热点区域，必须先插入图片。注意，图片必须增加usemap属性，说明该图像是热区映射图像，属性值必须以“#”开头，然后加上名字。具体语法格式如下：

```

```

2. 定义热点区域图像和热点区域链接

定义热点区域图像和热点区域链接的语法格式如下：

```
<map id="#热点图像名称">
  <area shape="热点形状1" coords="热点坐标1" href="链接地址1">
  <area shape="热点形状2" coords="热点坐标2" href="链接地址2">
</map>
```

<map>标签只有一个属性id，其作用是为区域命名，其设置值必须与标签的usemap属性值相同。

<area>标签主要是定义热点区域的形状及超链接，它有3个必需的属性。

(1) shape属性：控制划分区域的形状。其取值有3个，分别是rect（矩形）、circle（圆形）和poly（多边形）。

(2) coords属性：控制区域的划分坐标。如果shape属性取值为rect，那么coords的设置值分别为矩形的左上角x、y坐标点和右下角x、y坐标点，单位为像素。如果shape属性取值为circle，那么coords的设置值分别为圆形圆心x、y坐标点和半径值，单位为像素。如果shape属性取值为poly，那么coords的设置值分别为矩形的各个点x、y坐标，单位为像素。

(3) href属性：为区域设置超链接的目标。设置值为“#”时，表示为空链接。

实例6：添加图像热点链接

```
<!DOCTYPE html>
<html>
<head>
<title>创建热点区域</title>
</head>
<body>

  <map name="Map">
    <area shape="rect" coords=
"30,106,220,363" href="pic/r1.jpg"/>
```

```
    <area shape="rect" coords=
"234,106,416,359" href="pic/r2.jpg"/>
    <area shape="rect" coords=
"439,103,618,365" href="pic/r3.jpg"/>
    <area shape="rect" coords=
"643,107,817,366" href="pic/r4.jpg"/>
    <area shape="rect" coords=
"837,105,1018,363" href="pic/r5.jpg"/>
  </map>
</body>
</html>
```

在浏览器中浏览的效果如图2-6所示。

单击不同的热点区域, 将跳转到不同的页面。例如, 这里单击“超美女装”区域, 跳转页面效果如图 2-7 所示。



图 2-6 创建热点区域



图 2-7 热点区域的链接页面

在创建图像热点区域时, 比较复杂的操作是定义坐标, 初学者往往难以控制。目前比较好的解决方法是使用可视化软件手动绘制热点区域, 例如这里使用 Dreamweaver 软件绘制需要的区域, 如图 2-8 所示。



图 2-8 使用 Dreamweaver 软件绘制热点区域

2.2.3 绘制网页图形图像

canvas 标签是一个矩形区域, 包含两个属性 width 和 height, 分别表示矩形区域的宽度和高度, 这两个属性都是可选的, 并且都可以通过 CSS 来定义, 其默认值是 300px 和 150px。

canvas 在网页中的常用形式如下:

```
<canvas id="myCanvas" width="300" height="200"
  style="border:1px solid #c3c3c3;">
  您的浏览器不支持 canvas!
</canvas>
```

上面的示例代码中, id 表示画布对象名称, width 和 height 分别表示宽度和高度。最初的画布是不可见的, 此处为了观察这个矩形区域, 这里使用 CSS 样式, 即 style 标记。style 表示画布的样式。如果浏览器不支持画布标记, 会显示提示信息。

画布 canvas 本身不具有绘制图形的功能, 它只是一个容器, 用户可以在容器中绘制图形。既然 canvas 画布元素放好了, 就可以使用脚本语言 JavaScript 在网页上绘制图形了。画布 canvas 还有一项功能就是可以引入图像, 可以用于图片合成或者制作背景等。

HTML5 Canvas API 支持图片平铺, 此时需要调用 createPattern 函数, 即调用 createPattern

函数来替代先前的 `drawImage` 函数。函数 `createPattern` 的语法格式如下：

```
createPattern(image, type)
```

其中，`image` 表示要绘制的图像，`type` 表示平铺的类型，其具体含义如表 2-7 所示。

表 2-7 type 的参数

参 数 值	说 明	参 数 值	说 明
no-repeat	不平铺	repeat-y	纵方向平铺
repeat-x	横方向平铺	repeat	全方向平铺

实例 7：在 canvas 画布中引入图像并平铺

```
<!DOCTYPE html>
<html>
<head>
<title>绘制图像平铺</title>
</head>
<body onload="draw('canvas');">
<h1>图形平铺</h1>
<canvas id="canvas" width="800"
height="600"></canvas>
  您的浏览器不支持 canvas!
</canvas>
<script type="text/javascript">
  function draw(id){
    var canvas = document.getElementById
(id);
    if(canvas==null){
```

```
      return false;
    }
    var context = canvas.getContext
('2d');
    context.fillStyle = "#eeeeff";
    context.fillRect(0,0,800,600);
    image = new Image();
    image.src = "pic/02.jpg";
    image.onload = function(){
      var ptrn = context.createPattern
(image,'repeat');
      context.fillStyle = ptrn;
      context.fillRect(0,0,800,600);
    }
  }
</script>
</body>
</html>
```

上面的代码中，用 `fillRect` 创建了一个宽度为 800、高度为 600，左上角坐标位置为 (0,0) 的矩形，然后创建了一个 `Image` 对象，`src` 表示链接一个图像源，然后使用 `createPattern` 绘制一个图像，其方式是完全平铺，并将这个图像作为一个模式填充到矩形中。最后绘制这个矩形，此矩形的大小完全覆盖原来的图形。

在浏览器中浏览的效果如图 2-9 所示，在显示页面上绘制了一个图像，其图像以平铺的方式充满整个矩形。



图 2-9 图像平铺

2.3 HTML5 中的表单标签

在网页中，表单的作用比较重要，主要负责采集浏览者的相关数据。常见的表单有登录表、调查表和留言表等。在 HTML5 中，表单拥有多个新的表单输入类型，这些新特性提供了更好的输入控制和验证。如表 2-8 所示为 HTML5 中常用的一些表单标签。



表 2-8 HTML5 中常用的表单标签

标 签	描 述	标 签	描 述
<form>	定义一个 HTML 表单，用于用户输入	<label>	定义 input 元素的标注
<input>	定义一个输入控件	<fieldset>	定义围绕表单中元素的边框

续表

标 签	描 述	标 签	描 述
<textarea>	定义多行的文本输入控件	<legend>	定义 fieldset 元素的标题
<button>	定义按钮	<datalist>	规定了 input 元素可能的选项列表
<select>	定义选择列表 (下拉列表)	<keygen>	规定用于表单的密钥对生成器字段
<optgroup>	定义选择列表中相关选项的组合	<output>	定义一个计算的结果
<option>	定义选择列表中的选项		

2.3.1 文本类输入框

表单中的文本框有 3 种, 分别是单行文本框、多行文本框和密码输入框。不同的文本框对应的属性值也不同。

1. 单行与多行文本框

(1) 单行文本框 **text**: 单行文本框是一种让访问者自己输入内容的表单对象, 通常被用来填写单个字或者简短的回答, 例如用户姓名和地址等。代码格式如下:

```
<input type="text" name="..." size="..." maxlength="..." value="...">
```

其中, **type="text"** 定义单行文本输入框; **name** 属性定义文本框的名称, 要保证数据的准确采集, 必须定义一个独一无二的名称; **size** 属性定义文本框的宽度, 单位是单个字符宽度; **maxlength** 属性定义最多输入的字符数; **value** 属性定义文本框的初始值。

(2) 多行文本框 **textarea**: 主要用于输入较长的文本信息。代码格式如下:

```
<textarea name="..." cols="..." rows="..." wrap="..."></textarea>
```

其中, **name** 属性定义多行文本框的名称, 要保证数据的准确采集, 必须定义一个独一无二的名称; **cols** 属性定义多行文本框的宽度, 单位是单个字符宽度; **rows** 属性定义多行文本框的高度, 单位是单个字符宽度; **wrap** 属性定义输入内容大于文本域时的显示方式。

实例 8: 创建单行与多行文本框

```
<!DOCTYPE html>
<html>
<head><title>输入用户的姓名</title></
head>
<body>
<form>
请输入您的姓名:
<input type="text" name="yourname"
size="20" maxlength="15">
<br/>
请输入您的地址:
<input type="text" name="youradr"
size="20" maxlength="15">
<br/>
请输入您学习HTML5网页设计时最大的困难是
什么? <br/>
<textarea name="yourworks" cols
```

```
= "50" rows = "5"></textarea>
<br/>
</form>
</body>
</html>
```

在浏览器中浏览的效果如图 2-10 所示, 可以看到两个单行文本输入框和一个多行文本框。

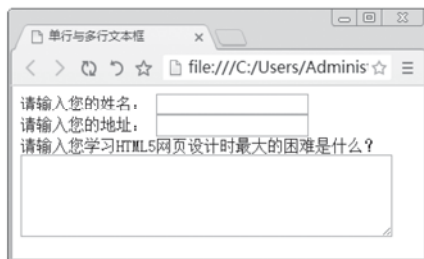


图 2-10 单行文本与多行文本输入框

2. 密码输入框 password

密码输入框是一种特殊的文本域，主要用于输入一些保密信息。当网页浏览者输入文本时，显示的是黑点或者其他符号，这样就增加了输入文本的安全性。代码格式如下：

```
<input type="password" name="..." size="..." maxlength="...">
```

其中 `type="password"` 定义密码框；`name` 属性定义密码框的名称，要保证唯一性；`size` 属性定义密码框的宽度，单位是单个字符宽度；`maxlength` 属性定义最多输入的字符数。

实例 9：创建包含密码域的账号登录页面

```
<!DOCTYPE html>
<html>
<head><title>输入用户姓名和密码
</title></head>
<body>
<form>
<h3>网站会员登录</h3>
账号:
<input type="text" name="yourname">
<br/>
密码:
<input type="password" name=
"yourpw"><br/>
</form>
</body>
</html>
```

在浏览器中浏览的效果如图 2-11 所示。输入用户名和密码时，可以看到密码以黑点的形式显示。



图 2-11 密码输入框

2.3.2 按钮类输入框

在设计调查问卷或商城购物页面时，经常会用到单选按钮和复选框。

(1) 单选按钮 `radio`：主要用于让网页浏览者在一组选项里选择一项。代码格式如下：

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

其中，`type="radio"` 定义单选按钮，`name` 属性定义单选按钮的名称，单选按钮都是以组为单位使用的，在同一组中的单选项都使用同一个名称；`value` 属性定义单选按钮的值，在同一组中，它们的域值不能相同。

(2) 复选框 `checkbox`：可以让网页浏览者在一组选项里同时选择多个选项。每个复选框都是一个独立的元素，都必须有一个唯一的名称。代码格式如下：

```
<input type="checkbox" name="" value="">
```

其中，`type="checkbox"` 定义复选框；`name` 属性定义复选框的名称，同一组中的复选框都必须用同一个名称；`value` 属性定义复选框的值。

实例 10：创建大学生技能需求问卷调查页面

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>问卷调查页面</title>
</head>
<body>
<form>
<h1>大学生技能需求问卷调查</h1>
请选择您感兴趣的技能:
<br/>
```

```
<input type="radio" name="book"
value="Book1">网站开发技能<br/>
<input type="radio" name="book"
value="Book2">美工设计技能<br/>
<input type="radio" name="book"
value="Book3">网络安全技能<br/>
<input type="radio" name="book"
value="Book4">人工智能技能<br/>
<input type="radio" name="book"
value="Book5">编程开发技能<br/>
请选择您需要购买的图书: <br/>
<input type="checkbox" name="book"
value="Book1"> HTML5 Web开发 (全案例微课
版) <br/>
<input type="checkbox" name="book"
value="Book2"> HTML5+CSS3+JavaScript网站
开发 (全案例微课版) <br/>
<input type="checkbox" name="book"
value="Book3"> SQL Server数据库应用 (全案
例微课版) <br/>
<input type="checkbox" name="book"
value="Book4"> PHP动态网站开发 (全案例微课
版) <br/>
<input type="checkbox" name="book"
value="Book5" checked> MySQL数据库应用 (全
案例微课版) <br/>
</form>
```

```
</body>
</html>
```

在浏览器中预览效果如图 2-12 所示, 即可以看到 5 个单选按钮与 5 个复选框, 用户只能选择其中一个单选按钮, 对于复选框, 则可以选择多个。



图 2-12 单选按钮与复选框

2.3.3 网页中的按钮

网页中的按钮按功能通常可以分为普通按钮、提交按钮和重置按钮。

(1) 普通按钮用来控制其他定义了处理脚本的处理工作。代码格式如下:

```
<input type="button" name="..." value="..." onClick="...">
```

其中, `type="button"` 定义为普通按钮; `name` 属性定义普通按钮的名称; `value` 属性定义按钮的显示文字; `onClick` 属性表示单击行为, 也可以是其他的事件, 通过指定脚本函数来定义按钮的行为。

(2) 提交按钮用来将输入的信息提交到服务器。代码格式如下:

```
<input type="submit" name="..." value="...">
```

其中, `type="submit"` 定义为提交按钮; `name` 属性定义提交按钮的名称; `value` 属性定义按钮的显示文字。通过提交按钮, 可以将表单里的信息提交给表单所指向的文件。

(3) 重置按钮又称为复位按钮, 用来重置表单中输入的信息。代码格式如下:

```
<input type="reset" name="..." value="...">
```

其中, `type="reset"` 定义复位按钮; `name` 属性定义复位按钮的名称; `value` 属性定义按钮的显示文字。

实例 11：通过普通按钮实现文本的复制和粘贴效果

```
<!DOCTYPE html>
<html>
<head>
<title>网页中按钮的功能</title>
</head>
<body>
<form>
单击“复制后粘贴”按钮，实现文本的复制和
粘贴：
<br/>
我喜欢的图书： <input type="text"
id="field1" value="HTML5 Web开发">
<br/>
我购买的图书： <input type="text"
id="field2">
<br/>
<input type="button" name="..."
value="复制后粘贴" onClick="document
.getElementById('field2').
value=document
.getElementById('field1').value">
<br/>
单击“提交”按钮，将表单里的信息提交给表
单所指向的文件：
<input type="submit" value="提交">
<br/>
单击“重置”按钮，实现将表单中的数据清空：
<br/>
<input type="reset" value="重置">
</form>
</body>
</html>
```

在浏览器中浏览的效果如图 2-13 所示。单击“复制后粘贴”按钮，即可实现将第一个文本框中的内容复制，然后粘贴到第二个文本框中。单击“提交”按钮，可以将表单信息发送给指定文件；单击“重置”按钮，可以清空表单中的信息，如图 2-14 所示。

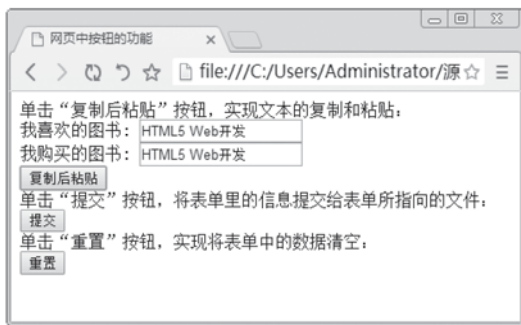


图 2-13 单击按钮后的粘贴效果

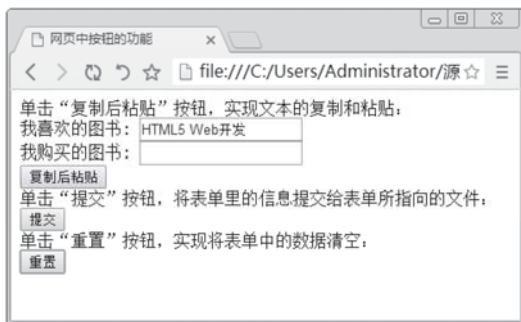


图 2-14 清空表单信息

2.3.4 图像域和文件域

为了丰富表单中的元素，可以使用图像域，从而解决表单中按钮比较单调，与页面内容不协调的问题。如果需要上传文件，往往需要通过文件域来完成。

1. 图像域 image

在设计网页表单时，为了让按钮和表单的整体效果比较一致，有时候需要在“提交”按钮上添加图片，使该图片具有按钮的功能，此时可以通过图像域来完成。语法格式如下：

```
<input type="image" src="图片的地址" name="代表的按键" >
```

其中，src 用于设置图片的地址；name 用于设置代表的按键，比如 submit 或 button 等，默认值为 button。

2. 文件域 file

使用 file 属性实现文件上传框。语法格式如下：

```
<input type="file" accept=" " name=" " size=" " maxlength=" " >。
```

其中, `type="file"` 定义为文件上传框; `accept` 用于设置文件的类别, 可以省略; `name` 属性为文件上传框的名称; `size` 属性定义文件上传框的宽度, 单位是单个字符宽度; `maxlength` 属性定义最多输入的字符数。

实例 12: 创建实名认证页面

```
<!doctype html>
<html>
<head>
<title>文件和图像域</title>
</head>
<body>
<div>
<h2 align="center">银行系统实名认证</h2>
<form>
<h3>请上传您的身份证正面图片: </h3>
<!--两个文件域-->
<input type="file">
<h3>请上传您的身份证背面图片: </h3>
<input type="file"><br/><br/>
<!--图像域-->
<input type="image" src="pic/anniu.
jpg" >
</form>
</div>
```

```
</body>
</html>
```

在浏览器中浏览的效果如图 2-15 所示。单击【选择文件】按钮, 即可选择需要上传的图片文件。



图 2-15 银行系统实名认证页面

2.3.5 设置表单中的列表

列表框主要用于在有限的空间里设置多个选项。列表框既可以用作单选, 也可以用作复选。代码格式如下:

```
<select name="..." size="..." multiple>
<option value="..." selected>
...
</option>
...
</select>
```

其中, `size` 属性定义列表框的行数; `name` 属性定义列表框的名称; `multiple` 属性表示可以多选, 如果不设置本属性, 则只能单选; `value` 属性定义列表项的值; `selected` 属性表示默认已经选中本选项。

实例 13: 创建报名信息调查表页面

```
<!DOCTYPE html>
<html>
<head>
<title>报名信息调查表</title>
</head>
<body>
<form>
<h2 align=" center">报名信息调查表
</h2>
<p>1. 请选择您目前的学历: </p><br/>
```

```
<!-- 下拉菜单实现学历选择-->
<select>
<option>初中</option>
<option>高中</option>
<option>大专</option>
<option>本科</option>
<option>研究生</option>
</select><br/>
<div align=" right">
<p>2. 请选择您感兴趣的技术方向: </
p><br/>
<!-- 下拉菜单中显示3个选项-->
```

```
<select name="book" size = "3"
multiple>
<option value="Book1">网站编程
<option value="Book2">办公软件
<option value="Book3">设计软件
<option value="Book4">网络管理
<option value="Book5">网络安全</
select>
</div>
</form>
</body>
</html>
```

在浏览器中浏览的效果如图 2-16 所示。可以看到列表框中显示了 3 个选项，用户可以按住 Ctrl 键，选择多个选项。



图 2-16 列表框的效果

2.3.6 表单常用属性的应用

除了上述基本表单标签外，HTML5 表单中还有一些高级属性，如表 2-9 所示。下面学习这些属性的使用方法。

表 2-9 表单常用的属性

属性名	说明
url 属性	用于说明网站网址。显示为一个文本字段输入 URL 地址
email 属性	用于让浏览者输入 E-mail 地址
date 和 time 属性	用于输入日期和时间
number 属性	提供了一个输入数字的输入类型
range 属性	显示为一个滑条控件
required 属性	规定必须在提交之前填写输入域（不能为空）

实例 14：使用属性创建一个信息统计表

```
<!DOCTYPE html>
<html>
<head>
<title> 使用表单属性</title>
</head>
<body>
<form>
<br/>
请输入购物网站的网址:
<input type="url" name="userurl"/>
<br/>
请输入您的邮箱地址:
<input type="email" name="user_
email"/>
<br/>
<br/>
请选择购买商品的日期:
<br/>
```

```
<input type="date" name="user_
date"/>
<br/>
<br/>
此网站我曾经来
<input type="number" name="shuzi"/>
次了哦!
<br/>
<br/>
购买次数公布了! 我的购买次数为:
<br/>
<br/>
<input type="range" name="ran"
min="1" max="16"/>
<br/>
下面是输入用户登录信息
<br/>
用户名称
<input type="text" name="user"
required="required">
```

```
<br/>
用户密码
<input type="password" name="
password" required="required">
<br/>
<input type="submit" value="登录">
</form>
</body>
</html>
```

在浏览器中浏览的效果如图 2-17 所示。
根据提示在各个文本框中输入内容。



图 2-17 使用表单属性

2.4 HTML5 中的表格标签

HTML 中的表格不但可以清晰地显示数据,而且可以用于页面布局。HTML 中的表格类似于 Word 软件中的表格,尤其是使用网页制作工具,操作很相似。HTML 是使用相关标签来制作表格的,如表 2-10 所示。



表 2-10 表格标签

标 签	描 述	标 签	标 签
<caption>	定义表格标题	<thead>	定义表格中的表头内容
<table>	定义一个表格	<tbody>	定义表格中的主体内容
<th>	定义表格中的表头单元格	<tfoot>	定义表格中的表注内容(脚注)
<tr>	定义表格中的行	<col>	定义表格中一个或多个列的属性值
<td>	定义表格中的单元	<colgroup>	定义表格中供格式化的列组

2.4.1 表格的基本结构

使用表格显示数据,可以更直观和清晰。在 HTML 文档中,表格主要用于显示数据,表格一般由行、列和单元格组成,如图 2-18 所示。

在 HTML5 中,最基本的表格,必须包含一对 <table></table> 标签、一对或几对 <tr></tr> 标签以及一对或几对 <td></td> 标签。一对 <table></table> 标签定义一个表格,一对 <tr></tr> 标签定义一行,一对 <td></td> 标签定义一个单元格。有时,为了方便表述表格,还需要在表格的上面加上标题。

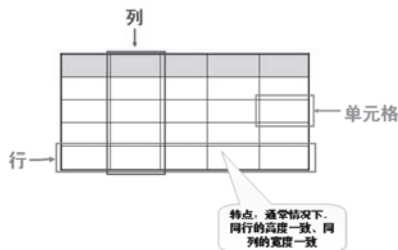


图 2-18 表格的组成

实例 15: 通过表格标签,编写公司销售表

```
<!DOCTYPE html>
<html>
<head>
<title>公司销售表</title>
```

```
</head>
<body>
<!--<table>为表格标签-->
<table border="2">
<caption>产品销售统计表</caption>
<!--<tr>为行标签-->
```

```
<tr>
  <!--<td>为表头标签-->
  <th>姓名</th>
  <th>月份</th>
  <th>销售额</th>
</tr>
<tr>
  <!--<td>为单元格-->
  <td>刘玉</td>
  <td>1月份</td>
  <td>32万</td>
</tr>
<tr>
  <!--<td>为单元格-->
  <td>张平</td>
  <td>1月份</td>
  <td>36万</td>
</tr>
<tr>
  <!--<td>为单元格-->
```

```
<td>胡明</td>
<td>1月份</td>
<td>18万</td>
</tr>
</table>
</body>
</html>
```

运行效果如图 2-19 所示。



图 2-19 公司销售表

2.4.2 使用属性编辑表格

创建好表格之后，还可以编辑表格，包括设置表格的边框类型、设置表格的表头、合并单元格等。用于编辑表格的属性如表 2-11 所示。

表 2-11 编辑表格的属性

属 性	说 明	属 性	说 明
border	边框宽度，边框值必须大于 1 像素才有效	width	表格宽度
bgcolor	表格背景色	height	表格高度
align	设置对齐方式，默认是左对齐	colspan	列合并标记
cellpadding	设置单元格边框和内部内容之间的间隔	rowspan	行合并标记
cellspacing	设置单元格之间的间隔		

1. 合并单元格

在实际应用中，有时需要将某些单元格进行合并，以符合内容安排的需要。在 HTML 中，合并的方向有两种，一种是上下合并，一种是左右合并，这两种合并方式只需要使用 td 标签的 colspan 属性和 rowspan 属性即可。

colspan 属性用于左右单元格的合并，格式如下：

```
<td colspan="数值">单元格内容</td>
```

其中，colspan 属性的取值为数值型整数数据，代表几个单元格进行左右合并。

rowspan 属性用于上下单元格的合并，格式如下：

```
<td rowspan="数值">单元格内容</td>
```

其中，rowspan 属性的取值为数值型整数数据，代表几个单元格进行上下合并。

实例 16: 设计婚礼流程安排表

```

<!DOCTYPE html>
<html>
<head>
<title>婚礼流程安排表</title>
</head>
<body>
<h1 align="center">婚礼流程安排表</h1>
<!--<table>为表格标签-->
<table align="center" border="1px"
cellpadding="12%" >
  <!--婚礼流程安排表日期-->
  <tr bgcolor="#A5AFEDD">
    <th></th>
    <th>时间</th>
    <th>日程</th>
    <th>地点</th>
  </tr>
  <!--婚礼流程安排表内容-->
  <tr align="center">
    <!--使用rowspan属性进行列合并
-->
    <td bgcolor="#FCD1CC" rowspan=
"2">上午</td>
    <td bgcolor="#FCD1CC">7:00--
8:30</td>
    <td>新郎新娘化妆定妆</td>
    <td>婚纱影楼</td>
  </tr>
  <!--婚礼流程安排表内容-->
  <tr align="center">
    <td bgcolor="#FCD1CC">8:30--
10:30</td>
    <td>新郎根据指导接亲</td>
    <td>酒店1楼</td>
  </tr>

```

```

    <!--婚礼流程安排表内容-->
    <tr align="center">
      <!--使用rowspan属性进行列合并
-->
      <td bgcolor="#FCD1CC" rowspan=
"2">下午</td>
      <td bgcolor="#FCD1CC">
12:30--14:00</td>
      <td>婚礼和就餐</td>
      <td>酒店2楼</td>
    </tr>
    <!--婚礼流程安排表内容-->
    <tr align="center">
      <td bgcolor="#FCD1CC">14:00--
16:00</td>
      <td>清点物品后离开酒店</td>
      <td>酒店2楼</td>
    </tr>
  </table>
</body>
</html>

```

运行效果如图 2-20 所示。

	时间	日程	地点
上午	7:00--8:30	新郎新娘化妆定妆	婚纱影楼
	8:30--10:30	新郎根据指导接亲	酒店1楼
下午	12:30--14:00	婚礼和就餐	酒店2楼
	14:00--16:00	清点物品后离开酒店	酒店2楼

图 2-20 婚礼流程安排表

2. 表格的分组

如果需要分组, 可以通过 <colgroup> 标签来完成。该标签的语法格式如下:

```

<colgroup>
  <col style="background-color: 颜色值">
  <col style="background-color: 颜色值">
  <col style="background-color: 颜色值">
</colgroup>

```

<colgroup> 标签可以控制表格列的样式, 其中 <col> 标签控制具体列的样式。

实例 17: 设计企业客户联系表

```

<!DOCTYPE html>
<html>
<head>
<title>企业客户联系表</title>
</head>
<body>
<h1 align="center">企业客户联系表</h1>

```

```

  <!--<table>为表格标签-->
  <table align="center" border="1px"
cellpadding="12%" >
    <!--<table>为表格标签-->
    <table align="center" border="1px"
cellpadding="12%" >
      <!--使用<colgroup>标签进行表格分组
控制-->
      <colgroup>

```

```
<col style="background-color:
#FFD9EC">
<col style="background-color:
#B8B8DC">
<col style="background-color:
#BBFFBB">
<col style="background-color:
#B9B9FF">
</colgroup>
<tr>
<th>区域</th>
<th>加盟商</th>
<th>加盟时间</th>
<th>联系电话</th>
</tr>

<tr align="center">
<td>华北区域</td>
<td>王蒙</td>
<td>2019年9月</td>
<td>123XXXXXXX</td>
</tr>

<tr align="center">
<td>华中区域</td>
<td>王小名</td>
<td>2019年1月</td>
<td>100XXXXXXX</td>
</tr>
```

```
</tr>

<tr align="center">
<td>西北区域</td>
<td>张小明</td>
<td>2012年9月</td>
<td>111XXXXXXX</td>
</tr>

</table>
</body>
</html>
```

运行效果如图 2-21 所示。



图 2-21 企业客户联系表

2.4.3 完整的表格标签

为了让表格的结构更清晰，以及配合后面学习的 CSS 样式，更方便地制作各种样式的表格，表格中还会出现表头、主体、脚注等。按照表格结构，可以把表格的行分组，称为“行组”。不同的行组具有不同的意义。行组分为 3 类——“表头”“主体”和“脚注”。三者相应的 HTML 标签依次为 <thead>、<tbody> 和 <tfoot>。

此外，在表格中还有两个标签：标签 <caption> 表示表格的标题；在一行中，除了用 <td> 标签表示一个单元格以外，还可以使用 <th> 表示该单元格是这一行的“行头”。

实例 18：使用完整的表格标签设计学生成绩单

```
<!DOCTYPE html>
<html>
<head>
<title>完整的表格标签</title>
<style>
tfoot{
background-color:#FF3;
}
</style>
</head>
<body>
<table border="1">
<caption>学生成绩单</caption>
<thead>
```

```
<tr>
<th>姓名</th><th>性别</th><th>
成绩</th>
</tr>
</thead>
<tfoot>
<tr>
<td colspan="2">平均分</td><td>
colspan="2">540</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>张三</td><td>男</td><td>560</
td>
</tr>
</tbody>
```

```
<td>李四</td><td>男</td><td>520</td>
</tr>
</tbody>
</table>
</body>
</html>
```

从上面的代码中可以发现,使用<caption>标签定义了表格标题,<thead>、<tbody>和<tfoot>标签对表格进行了分组。在<thead>部分使用<th>标签代替<td>标签定义单元格,

<th>标签定义的单元格内容默认加粗显示。网页的浏览效果如图 2-22 所示。



图 2-22 完整的表格结构

注意: <caption> 标签必须紧随 <table> 标签之后。

2.5 HTML5 中的多媒体标签

在 HTML5 版本出现之前,要想在网页中展示多媒体,大多数情况下需要用到 Flash。这就需要浏览器安装相应的插件,而且加载多媒体的速度也不快。HTML5 新增了音频和视频的标签,从而解决了上述问题,如表 2-12 所示。



表 2-12 HTML5 中的多媒体标签

标 签	说 明
<audio>	定义声音,比如音乐或其他音频流
<source>	定义 media 元素 (<video> 和 <audio>) 的媒体资源
<track>	为媒体 (<video> 和 <audio>) 元素定义外部文本轨道
<video>	定义一个音频或者视频

2.5.1 audio 标签的应用

audio 标签主要用于定义播放声音文件或者音频流的标准。它支持 3 种音频格式,分别为 Ogg、MP3 和 WAV。

如果需要在 HTML5 网页中播放音频,输入的基本格式如下:

```
<audio src="song.mp3" controls="controls"></audio>
```

提示: src 属性规定要播放的音频的地址,controls 属性用于添加播放、暂停和音量控件。

另外,在 <audio> 和 </audio> 之间插入的内容是供不支持 audio 元素的浏览器显示的。

实例 19: 认识 audio 标签

```
<!DOCTYPE html>
<html>
<head>
<title>audio</title>
</head>
```

```
<body>
<audio src="song.mp3" controls=
"controls">
您的浏览器不支持audio标签!
</audio>
</body>
</html>
```

如果用户的浏览器版本不支持 `audio` 标签, 浏览效果如图 2-23 所示, IE 11.0 以前的浏览器版本不支持 `audio` 标签。

对于支持 `audio` 标签的浏览器, 运行效果如图 2-24 所示, 可以看到加载的音频控制条并听到声音, 此时用户还可以控制音量的大小。



图 2-23 不支持 `audio` 标签的效果

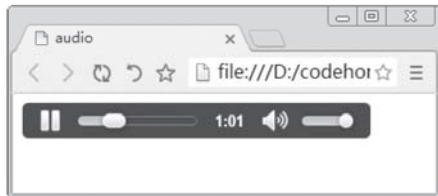


图 2-24 支持 `audio` 标签的效果

`audio` 标签的常见属性和含义如表 2-13 所示。

表 2-13 `audio` 标签的常见属性

属 性	值	含 义
<code>autoplay</code>	<code>autoplay</code> (自动播放)	如果出现该属性, 则音频在就绪后马上播放
<code>controls</code>	<code>controls</code> (控制)	如果出现该属性, 则向用户显示控件, 比如播放按钮
<code>loop</code>	<code>loop</code> (循环)	如果出现该属性, 则每当音频结束时重新开始播放
<code>preload</code>	<code>preload</code> (加载)	如果出现该属性, 则音频在页面加载时进行加载, 并预备播放。如果使用 <code>autoplay</code> , 则忽略该属性
<code>src</code>	<code>url</code> (地址)	要播放的音频的 URL 地址

另外, `audio` 标签可以通过 `source` 属性添加多个音频文件, 具体格式如下:

```
<audio controls="controls">
  <source src="123.ogg" type="audio/ogg">
  <source src="123.mp3" type="audio/mpeg">
</audio>
```

2.5.2 在网页中添加音频文件

当在网页中添加音频文件时, 用户可以根据自己的需要, 添加不同类型的音频文件, 如添加自动播放的音频文件、添加带有控件的音频文件、添加循环播放的音频文件、添加预播放的音频文件等。

1. 添加自动播放的音频文件

`autoplay` 属性规定一旦音频就绪, 马上就开始播放。如果设置了该属性, 音频将自动播放。下面就是在网页中添加自动播放音频文件的相关代码:

```
<audio controls="controls" autoplay="autoplay">
<source src="song.mp3">
```

2. 添加带有控件的音频文件

`controls` 属性规定浏览器应该为音频提供播放控件。其中浏览器控件应该包括播放、暂停、定位、音量、全屏切换等。

添加带有控件的音频文件的代码如下:

```
<audio controls="controls">
<source src="song.mp3">
```

3. 添加循环播放的音频文件

loop 属性规定当音频播放结束后将重新开始播放。如果设置了该属性，则音频将循环播放。添加循环播放的音频文件的代码如下：

```
<audio controls="controls" loop="loop">
<source src="song.mp3">
```

4. 添加预播放的音频文件

preload 属性规定是否在页面加载后载入音频。如果设置了 autoplay 属性，则忽略该属性。preload 属性的值可能有三种，分别如下。

- auto：当页面加载后载入整个音频。
- meta：当页面加载后只载入元数据。
- none：当页面加载后不载入音频。

添加预播放的音频文件的代码如下：

```
<audio controls="controls" preload="auto">
<source src="song.mp3">
```

实例 20：创建一个带有控件、自动播放并循环播放音频的文件

```
<!DOCTYPE html>
<html>
<head>
<title>audio</title>
<head>
<body>
<audio src="song.mp3"controls=
"controls" autoplay="autoplay" loop=
"loop">
```

您的浏览器不支持audio标签!

```
</audio>
</body>
</html>
```

运行效果如图 2-25 所示。音频文件会自动播放，播放完成后会自动循环播放。

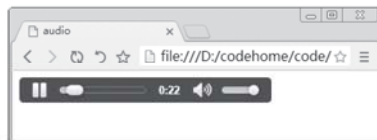


图 2-25 带有控件、自动播放并循环播放的效果

2.5.3 认识 video 标签

video 标签主要用于定义播放视频文件或者视频流的标准。它支持 3 种视频格式，分别为 Ogg、WebM 和 MPEG 4。

如果需要在 HTML5 网页中播放视频，输入的基本格式如下：

```
<video src="123.mp4" controls="controls">...</video>
```

其中，在 <video> 与 </video> 之间插入的内容是供不支持 video 元素的浏览器显示的。

实例 21：认识 video 标签

```
<!DOCTYPE html>
<html>
<head>
<title>video</title>
```

```
<head>
<body>
<video src="fengjing.mp4"
controls="controls">
您的浏览器不支持video标签!
</video>
```

```
</body>
</html>
```

如果用户的浏览器是 IE 11.0 以前的版本, 运行效果如图 2-26 所示, 可见 IE 11.0 以前的浏览器版本不支持 video 标签。



图 2-26 不支持 video 标签的效果

如果浏览器支持 video 标签, 运行效果如图 2-27 所示, 可以看到加载的视频控制条

界面。单击“播放”按钮, 即可查看视频的内容, 同时还可以调整音量的大小。

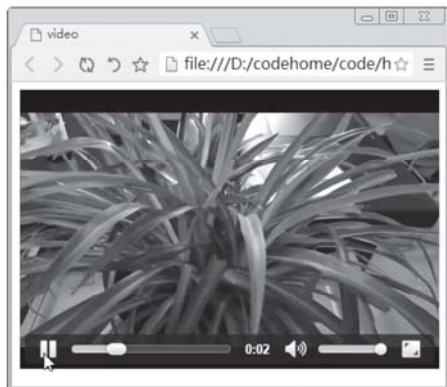


图 2-27 支持 video 标签的效果

video 标签的常见属性和含义如表 2-14 所示。

表 2-14 video 标签的常见属性和含义

属 性	值	含 义
autoplay	autoplay	视频就绪后马上播放
controls	controls	向用户显示控件, 比如播放按钮
loop	loop	每当视频结束时重新开始播放
preload	preload	视频在页面加载时进行加载, 并预备播放。如果使用 autoplay, 则忽略该属性
src	url	要播放的视频的 URL
width	宽度值	设置视频播放器的宽度
height	高度值	设置视频播放器的高度
poster	url	当视频未响应或缓冲不足时, 该属性值链接到一个图像。该图像将以一定比例被显示出来

由表 2-14 可知, 用户可以自定义视频文件显示的大小。例如, 如果想让视频以 320×240 像素大小显示, 可以加入 width 和 height 属性。具体格式如下:

```
<video width="320" height="240" controls src="movie.mp4"></video>
```

另外, video 标签可以通过 source 属性添加多个视频文件, 具体格式如下:

```
<video controls="controls">
  <source src="123.ogg" type="video/ogg">
  <source src="123.mp4" type="video/mp4">
</video>
```

2.5.4 在网页中添加视频文件

当在网页中添加视频文件时, 用户可以根据自己的需要添加不同类型的视频文件, 如添加自动播放的视频文件、添加带有控件的视频文件、添加循环播放的视频文件等, 另外, 还

可以设置视频文件的高度和宽度。

1. 添加自动播放的视频文件

`autoplay` 属性规定一旦视频就绪马上开始播放。如果设置了该属性，视频将自动播放。添加自动播放的视频文件的代码如下：

```
<video controls="controls" autoplay="autoplay">
  <source src="movie.mp4">
</video>
```

2. 添加带有控件的视频文件

`controls` 属性规定浏览器应该为视频提供播放控件。浏览器控件应该包括播放、暂停、定位、音量、全屏切换等。

添加带有控件的视频文件的代码如下：

```
<video controls="controls" controls="controls">
  <source src="movie.mp4">
</video>
```

3. 添加循环播放的视频文件

`loop` 属性规定当视频结束后将重新开始播放。如果设置该属性，则视频将循环播放。添加循环播放的视频文件的代码如下：

```
<video controls="controls" loop="loop">
  <source src="movie.mp4">
</video>
```

4. 添加预播放的视频文件

`preload` 属性规定是否在页面加载后载入视频。如果设置了 `autoplay` 属性，则忽略该属性。`preload` 属性的值有三种，分别说明如下。

- `auto`：当页面加载后载入整个视频。
- `meta`：当页面加载后只载入元数据。
- `none`：当页面加载后不载入视频。

添加预播放的视频文件的代码如下：

```
<video controls="controls" preload="auto">
<source src="movie.mp4">
```

5. 设置视频文件的高度与宽度

使用 `width` 和 `height` 属性可以设置视频文件的显示宽度与高度，单位是像素。

提示：规定视频的高度和宽度是一个好习惯。如果设置这些属性，在页面加载时会为视频预留出空间。如果没有设置这些属性，那么浏览器就无法预先确定视频的尺寸，这样就无法为视频保留合适的空间。结果是，在页面加载的过程中，其布局也会产生变化。



2.8 新手常见疑难问题

疑问 1: HTML 文档页面上边总是留出一段空白, 这是为什么?

答: body 标签默认有上边距, 设置这个值的属性 topmargin=0 就可以了。有时还需要设置 leftmargin、rightmargin 和 bottommargin 属性值。

疑问 2: 使用 <thead>、<tbody> 和 <tfoot> 标签对行进行分组的意义何在?

答: 在 HTML 文档中增加 <thead>、<tbody> 和 <tfoot> 标签虽然从外观上不能看出任何变化, 但是它们却能使文档的结构更加清晰。使用 <thead>、<tbody> 和 <tfoot> 标签除了使文档更加清晰外, 还有一个更重要的意义, 就是方便使用 CSS 样式对表格的各个部分进行修饰, 从而制作出更炫的表格。

疑问 3: 在 HTML5 网页中添加 MP4 格式的视频文件, 为什么在不同的浏览器中视频控件显示的外观不同?

答: 在 HTML5 中规定用 controls 属性来控制视频文件的播放、暂停、停止和调节音量的操作。controls 是一个布尔属性, 一旦添加了此属性, 等于告诉浏览器需要显示播放控件并允许用户进行操作。

因为浏览器负责解释内置视频控件的外观, 所以在不同的浏览器中, 将会显示不同的视频控件外观。

2.9 实战技能训练营

实战 1: 编写一个计算机报价表的页面

利用所学的表格知识, 制作一个笔记本电脑报价表。这里利用 caption 标签制作表格的标题, 用 <th> 代替 <td> 作



为标题行单元格。可以将图片放在单元格内, 即在 <td> 标签内使用 标签, 在浏览器中浏览的效果如图 2-31 所示。

型号	类型	价格	图片
宏碁(Acer) AS4552-P362G32MNC	笔记本	¥2799	
戴尔(Dell) 14VR-188	笔记本	¥3499	
联想(Lenovo) G470AH2310W42G500P7W3(DB)-CN	笔记本	¥4149	
戴尔家用(DELL) I560SR-656	台式	¥3599	

图 2-31 计算机报价表的页面

实战 2: 编写一个多功能的视频播放效果的页面

综合使用视频播放时所用的方法和多媒体的属性, 在播放视频文件时, 包括播放、暂停、停止、加速播放、减速播放和正常速度, 并显示播放的时间。运行结果如图 2-32 所示。



图 2-32 多功能的视频播放效果