

3.1 表格、表单

3.1.1 案例描述

在日常浏览网页中,每个人都使用过很多次登录、注册或者添加商品等功能。本案例中会使用表格和表单来实现如图 3-1 所示的注册表。

姓名：	<input type="text"/>
密码：	<input type="password"/>
确认密码：	<input type="password"/>
邮箱：	<input type="text"/>
性别：	<input type="radio"/> 男 <input type="radio"/> 女
爱好：	<input type="checkbox"/> 吃饭 <input type="checkbox"/> 玩耍 <input type="checkbox"/> 休息
城市：	<input type="text" value="请选择"/>
自我介绍：	<div style="border: 1px solid gray; padding: 5px; min-height: 40px;">属于你个人的介绍,请填写</div>
<input type="button" value="立即注册"/>	

图 3-1 注册表

3.1.2 知识引入

1. 表格

表格是网页制作中使用最多的技术之一。表格可以清晰明了地展现数据之间的关系,使对比分析更容易理解。在很多情况下,也可以使用表格对网页进行排版布局。

1) 表格结构

在 HTML 中使用< table>标签来创建表格,< table>标签内包含了表名和表格本身内容的代码。表格是由特定数目的行和列组成的,其中行用标签< tr>表示,行由若干单元格构成。单元格是表格的基本单元,用标签< td>表示,< td>标签定义了一个列,嵌套于< tr>标签之中。多个单元格结合在一起构成了行,多个行结合在一起就构成了一个表格。

其用法如下所示：

```

<table border = "1">
<tr>
<td>第一行, 第一列</td>
<td>第一行, 第二列</td>
</tr>
<tr>
<td>第二行, 第一列</td>
<td>第二行, 第二列</td>
</tr>
</table>

```

2) 表格标签

HTML 中有 10 个与表格相关的标签, 各标签的含义及作用如下所示。

- <table>标签: 定义一个表格。
- <caption>标签: 定义一个表格标题, 必须紧随在<table>标签之后, 且每个表格只能包含一个标题, 通常这个标题会居中显示于表格上部。
- <th>标签: 定义表格内的表头单元格, 标签内部的文本通常会呈现为粗体。
- <tr>标签: 在表格中定义一行。
- <td>标签: 定义表格中的一个单元格, 包含在<tr>标签中。
- <thead>标签: 定义表格的表头。
- <tbody>标签: 定义一段表格主体(正文), 使用<tbody>标签, 可以将表格中的一行或几行合成一组, 从而将表格分为几个单独的部分, 一个<tbody>标签就是表格中的一个独立的部分, 不能从一个<tbody>跨越到另一个<tbody>中。
- <tfoot>标签: 定义表格的页脚(脚注)。
- <col>标签: 定义表格中针对一个或多个列的属性值。只能在表格或<colgroup>标签中使用此标签。
- <colgroup>标签: 定义表格列的分组。通过此标签可以对列进行组合以便进行格式化, 此标签只能用在<table>标签内部。

注意: 使用<thead>、<tfoot>以及<tbody>标签可以对表格中的行进行分组。如使表格拥有一个标题行、一些带有数据的行, 以及位于底部的一个总计行。这种划分使浏览器有能力支持独立于表格标题和页脚的表格正文滚动。当长的表格被打印时, 表格的表头和页脚可被打印在包含表格数据的每个页面上。

示例代码如下所示:

```

<html>
<head>
  <title>表格标签示例</title>
</head>
<body>
  <table border = "1">
    <caption>Table caption here</caption>
    <colgroup span = "1" style = "background:red;" />
    <colgroup span = "2" style = "background:yellow;" />
    <!-- 表格头 -->
    <thead>
      <tr>

```

```

        <th>头 1 </th>
        <th>头 2 </th>
        <th>头 3 </th>
    </tr>
</thead>
<!-- 表格页脚 -->
<tfoot>
    <tr>
        <td>页脚 1</td>
        <td>页脚 2</td>
        <td>页脚 3</td>
    </tr>
</tfoot>
<!-- 表格主体 -->
<tbody>
    <tr>
        <td>A</td>
        <td>B</td>
        <td>C</td>
    </tr>
    <tr>
        <td>D</td>
        <td>E</td>
        <td>F</td>
    </tr>
</tbody>
</table>
</body>
</html>

```

表格标签代码效果如图 3-2 所示。

3) 表格属性设置

为了使表格的外观更加符合要求,还可以对表格的属性进行设置,比较常用的表格属性包括背景、宽高、对齐方式、单元格间距、文本与边框间距等,如表 3-1 所示。

头 1	头 2	头 3
A	B	C
D	E	F
页脚 1	页脚 2	页脚 3

图 3-2 表格标签示例

表 3-1 表格属性列表

表格属性	说 明
border	此属性定义表格的边框。比如, border=1, 表示表格边框的粗细为 1 个像素(默认值), 为 0 表示没有边框
bgcolor	表格的背景色。取值方法举例: bgcolor=#ff0000 或 bgcolor=red。单元格<td>也可有此属性, 如果设置了表格的背景色, 又设置表格单元格的背景色, 这种情况主要用于多单元格的表格
background	表格的背景图。其值为一个有效的图片地址。<td>也有此属性。同时设置背景色和背景图不冲突
width	表格的宽度。取值从 0 开始, 默认以像素为单位, 与显示器的分辨率的像素是一致的。在 800×600 的显示分辨率下, 如果表格设置成 1000 个像素的宽度, 那么, 得出的效果将导致 IE 的横向滚动条出现, 只有通过滑动它才能看到表格最右边的内容, 所以建议在设置表格的宽度时充分考虑显示分辨率问题。width 的取值还可以使用百分比, 如 width="100%", 这种赋值法的好处是: 表格的宽度将根据可显示的宽度来自我调整

续表

表格属性	说 明
height	表格的高度,取值方法同 width。提示:如果不是特别需要,建议不用设置表格的高度,系统会根据表格的内容自动设置高度。所谓特别需要,是指在一些特殊的情形下,需要表格的高度精确,比如,当通过表格的背景来发一张图片时,如果表格的高度不精确定义,图片就不可能完整或完美地显示
align	表格的对齐方式,值有 left(左对齐,默认)、center(居中)以及 right(右对齐)。align 定义的是表格自身的位置,这是一个很有用的属性,强烈建议使用它来规定表格的对齐方式,尽量不要使用< p align=? >表格</p >、< div align=? >表格</div >和< center >表格</center >标签来规定表格的位置,因为这样做将导致代码冗余。此外,当表格的宽度设置为 100%,或者,表格的宽度设置成了占满它所在的容器的宽度,没有必要定义 align 属性
cellspacing	单元格间距。当一个表格有多个单元格时,各单元格的距离就是 cellspacing 了,如若表格只有一个单元格,那么这个单元格与表格上、下、左、右边框的距离也是 cellspacing
cellpadding	指该单元格里的内容与 cellspacing 区域的距离
colspan	单元格水平合并,值为合并的单元格的数目
rowspan	单元格垂直合并,值为合并的单元格的数目

示例代码如下所示:

```
<html >
<head >
  <title >表格属性示例</title >
</head >
<body >
  <table border = "1" height = "15 %" width = "60 %" cellspacing = "0" style = "font - size:
14px">
    <caption >员工信息表</caption >
    <thead bgcolor = "red">
      <th >部门</th >
      <th >姓名</th >
      <th >联系电话</th >
      <th >E - Mail</th >
    </thead >
    <tbody bgcolor = " #FFFAF0">
      <tr >
        <td >技术部</td >
        <td >张三</td >
        <td >18585426120 </td >
        <td >zhangs@haier.com </td >
      </tr >
      <tr >
        <td >人事部</td >
        <td >李四</td >
        <td >18519529902 </td >
        <td >lis@haier.com </td >
      </tr >
    </tbody >
    <tfoot bgcolor = "yellow">
      <tr >
```

```

        <td colspan = "4" align = "right">Compiled in 2020 by Mr. Zhang</td>
    </tr>
</tfoot>
</table>
</body>
</html>

```

使用过表格标签属性的代码页面效果如图 3-3 所示。

员工信息表			
部门	姓名	联系电话	E-Mail
技术部	张三	18585426120	zhangs@haier.com
人事部	李四	18519529902	lis@haier.com
Compiled in 2020 by Mr. Zhang			

图 3-3 表格属性实例

2. 表单

表单在网页中主要负责数据采集功能。一个表单有 3 个基本组成部分。

(1) 表单标签：这里面包含了处理表单数据所用 CGI 程序的 URL 以及数据提交到服务器的方法。

(2) 表单域：包含了文本框、密码框、隐藏域、多行文本框、复选框、单选按钮、下拉列表框和文件上传框等表单输入控件。

(3) 表单按钮：包括提交按钮、复位按钮和一般按钮；用于将数据传送到服务器上或者取消输入，还可以用表单按钮来控制其他定义了处理脚本的工作。

```

<html>
<head>
    <title>表单示例</title>
</head>
<body>
    <form action = "a. php" method = "POST">
        账号: <input type = "text" name = "name"><br><br>
        密码: <input type = "password" name = "password"><br>
        <input type = "submit" value = "提交">
    </form>
    <p>单击"提交"按钮, 表单数据将被发送到服务器上的"a. php".</p>
</body>
</html>

```

上述代码中,通过<form>和</form>标签表示表单的范围,表单内包含两个文本输入框,分别用于让访问者输入账号和密码;还包含一个提交按钮,用于提交数据。此外,在表单标签中 action 的属性值"a. php"表示表单数据提交的目的地,该表单的提交方式通过 method 属性指定,值为"POST"。

通过 IE 浏览器查看该 HTML,表单效果如图 3-4 所示。

表单标签(<form></form>)用于声明表单,定义采集数据的范围,同时包含了处理表单数据的应用程序以及数据提交到服务器的方法。

其语法如下:

单击“提交”按钮，表单数据将被发送到服务器上的“a.php”。

图 3-4 表单示例

```
<form action = "url" method = "get/post" enctype = "mime" target = "...">
...</form>
```

表单标签属性如表 3-2 所示。

表 3-2 form 标签属性列表

属性	值	描述
accept	MIME_type	HTML5 不支持。规定服务器接收到的文件的类型
accept-charset	character_set	规定服务器可处理的表单数据字符集
action	URL	指定处理表单中用户输入数据的 URL (URL 可为 Servlet、JSP 或 ASP 等服务器端程序), 也可以将输入数据发送到指定的 E-Mail 地址等
enctype	application/x-www-form-urlencoded multipart/form-data text/plain	指定数据发送时的编码类型, 默认值是 application/x-www-form-urlencoded, 用于常规数据的编码。另一种编码类型是 multipart/form-data, 该类型将表单数据编码为一条消息, 每一个表单控件的数据对应消息的一部分, 以二进制的方式发送给服务器端。这种方法比较适合传递复杂的用户数据, 如文件的上传操作
method	get post	指定向服务器传递数据的 HTTP 方法, 主要有 get 和 post 两种方法, 默认值是 get。get 方式是将表单控件的 name/value 信息经过编码之后, 通过 URL 发送, 可以在浏览器的地址栏中看到这些值。而采用 post 方式传输信息则在地址栏看不到表单的提交信息。需要注意的是, 当只为取得和显示少量数据时可以使用 get 方法; 一旦涉及数据的保存和更新, 即大量的数据传输时则应当使用 post 方法
name	text	规定表单的名称
target	_blank _self _parent _top	用于指定在浏览器哪个框架中显示服务器的响应 HTML, 默认值是当前框架。现在大多数专业界面使用框架越来越少, 所以此属性已很少使用

注意：一般情况下, target 属性的取值有如下情况:

_blank——在一个新的浏览器窗口调入指定的文档;

_self——在当前框架中调入文档;

_parent——把文档调入当前框架的直接父框架集中, 这个值在当前框架没有父框架集时等价于 _self;

_top——把文档调入原来最顶部的浏览器窗口中。

HTML5 新增属性如表 3-3 所示。

表 3-3 表单标签的 HTML5 新增属性列表

属性	值	描述
autocomplete	On off	规定是否启用表单的自动完成功能
novalidate	novalidate	如果使用该属性,则提交表单时不进行验证

(1) 表单域。

表单域包含了文本框、密码框、隐藏域、多行文本框、复选框、单选按钮、下拉列表框和文件上传框等,用于采集用户输入或选择的数据。下面分别讲述各表单域。

(2) 文本框。

<input>标签规定用户可以在其中输入数据的输入字段。在表单标签中使用,用来声明允许用户输入数据的是 input 元素,输入字段可通过多种方式改变,取决于 type 属性。

其语法格式如下:

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

文本框属性如表 3-4 所示。

表 3-4 文本框属性列表

属性	值	描述
accept	audio/ * video/ * image/ * MIME_type	规定通过文件上传来提交的文件的类型,只针对 type="file"
align	Left/right/top /middle/bottom	HTML5 已废弃,不建议使用。规定图像输入的对齐方式,只针对 type="image"
alt	text	定义图像输入的替代文本,只针对 type="image"
checked	checked	规定在页面加载时应该被预先选定的 input 元素,只针对 type="checkbox" 或者 type="radio"
disabled	disabled	规定应该禁用的 input 元素
name	text name	规定 input 元素的名称
size	number	规定以字符数计的 input 元素的可见宽度
src	URL	规定显示为提交按钮的图像的 URL,只针对 type="image"
type	button checkbox color date datetime datetime-local email file hidden image month	规定要显示的 input 元素的类型

续表

属性	值	描 述
type	number	
	password	
	radio	
	range	
	reset	
	search	
	submit	
	tel	
	text	
	time	
	url	
week		
value	text	指定 input 元素 value 的值
maxlength	number	规定 input 元素中允许的最大字符数

文本框 HTML5 新增属性,如表 3-5 所示。

表 3-5 文本框 HTML5 新增属性列表

属性	值	描 述
autocomplete	On off	规定 input 元素输入字段是否应该启用自动完成功能
autofocus	autofocus	规定当页面加载时 input 元素应该自动获得焦点
form	form_id	form 属性规定 input 元素所属的一个或多个表单
formaction	URL	规定当表单提交时处理输入控件的文件的 URL,只针对 type="submit" 和 type="image"
formenctype	application/x-www-form-urlencoded/multipart/form-datatext/plain	规定当表单数据提交到服务器时如何编码,只适合 type="submit" 和 type="image"
formmethod	getpost	定义发送表单数据到 action URL 的 HTTP 方法,只适合 type="submit" 和 type="image"
formnovalidate	formnovalidate	覆盖 form 元素的 novalidate 属性
formtarget	_blank_self_parent_topframename	规定表示提交表单后在哪里显示接收到响应的名称或关键词,只适合 type="submit" 和 type="image"
height	pixels	规定 input 元素的高度,只针对 type="image"
list	datalist_id	引用 datalist 元素,其中包含 input 元素的预定义选项
max	numberdate	规定 input 元素的最大值
min	numberdate	规定 input 元素的最小值
multiple	multiple	规定允许用户输入到 input 元素的多个值
pattern	regex	规定用于验证 input 元素的值的正则表达式
placeholder	text	规定可描述输入 input 字段预期值的简短的提示信息
readonly	readonly	规定输入字段是只读的
required	required	规定必需在提交表单之前填写输入字段
stepNew	number	规定 input 元素的合法数字间隔
width	pixels	规定 input 元素的宽度,只针对 type="image"

(3) 多行文本框。

多行文本框(文本域)是一种用来输入较长内容的表单对象。文本区域中可容纳无限数量的文本,多行文本框中的文本的默认字体是等宽字体(通常是 Courier)。

可以通过 cols 和 rows 属性来规定 textarea 的尺寸大小,不过更好的办法是使用 CSS 的 height 和 width 属性。

其语法格式如下:

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

多行文本框属性如表 3-6 所示。

表 3-6 多行文本框属性列表

属性	值	描 述
cols	number	定义多行文本框的宽度,单位是单个字符宽度
name	text	指定文本域的名称
rows	number	规定文本区域内可见的行数
disabled	disabled	规定禁用文本区域
readonly	readonly	规定文本区域为只读

多行文本框 HTML5 新增属性如表 3-7 所示。

表 3-7 多行文本框 HTML5 新增属性列表

属性	值	描 述
autofocus	autofocus	规定当页面加载时,文本区域自动获得焦点
form	text	定义文本区域所属的一个或多个表单
maxlength	number	规定文本区域允许的最大字符数
placeholder	disabled	规定一个简短的提示,描述文本区域期望的输入值
required	readonly	规定文本区域是必需的/必填的
wrap	hardsoft	规定当提交表单时,文本区域中的文本应该怎样换行

(4) 密码框。

密码框是一种用于输入密码的特殊文本域。当访问者输入文字时,文字会被星号或其他符号代替,从而隐藏输入的真实文字。

其语法格式如下:

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

其中,

- type="password": 定义密码框。
- name: 指定密码框的名称。
- size: 定义密码框的宽度,单位是单个字符宽度。
- maxlength: 定义最多输入的字符数。

注意: 密码框并不能保证安全,仅仅是使得周围的人看不见输入的内容,在传输过程中还是以明文传输,为了保证安全可以采用数据加密技术。

（5）隐藏框。

隐藏域是用来收集或发送信息的不可见元素，网页的访问者无法看到隐藏域，但是当表单被提交时，隐藏域的内容同样会被提交。

其语法格式如下：

```
<input type = "hidden" name = "..." value = "..." />
```

其中，

- type="hidden": 定义隐藏域。
- name: 同 text 的 name 属性。
- value: 定义隐藏域的值。

（6）复选框。

复选框允许在待选项中选中一个以上的选项。每个复选框都是一个独立的元素。

其语法格式如下：

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

其中，

- type="checkbox": 定义复选框。
- name: 同 text 的 name 属性。
- value: 定义复选框的值。

注意：通常情况下，对于一组复选框的 name 值推荐使用相同的值，这样提交表单后，在服务器端便于数据的处理。

（7）单选按钮。

单选按钮只允许访问者在待选项中选择唯一的一项。该控件用于一组相互排斥的值，组中每个单选按钮控件的名字相同，用户一次只能选择一个选项。

其语法格式如下：

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

其中，

- type="radio": 定义单选按钮。
- name: 同 text 的 name 属性，name 相同的单选按钮为一组，一组内只能选中一项。
- value: 定义单选按钮的值，在同一组中，单选按钮的值不能相同。

（8）文件上传框。

文件上传框用于让用户上传自己的文件，文件上传框与其他文本域类似，但它还包含了一个浏览按钮。访问者可以通过输入需要上传的文件的路径或者单击“浏览”按钮选择需要上传的文件。

其语法格式如下：

```
<input type = "file" name = "..." size = "15" maxlength = "100" />
```

其中,

- type="file": 定义文件上传框。
- name: 同 text 的 name 属性。
- size: 定义文件上传框的宽度,单位是单个字符宽度。
- maxlength: 定义最多输入的字符数。

注意: 在使用文件域以前,需要确定服务器是否允许匿名上传文件。另外,在表单标签中必须设置 enctype="multipart/form-data"来确保文件被正确编码;表单的传送方式必须设置成 post。

(9) 下拉列表框。

下拉列表框可以让浏览器快速、方便、正确地选择一些选项,同时可以节省页面空间,它通过<select>标签实现,该标签用于显示可供用户选择的下拉列表。每个选项由一个<option>标签表示,<select>标签至少包含一个<option>标签。

其语法格式如下:

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

属性: <select>和<option>的属性如表 3-8 和表 3-9 所示。

表 3-8 <select>属性列表

属性	值	描 述
disabled	disabled	当该属性为 true 时,会禁用下拉列表
multiple	multiple	当该属性为 true 时,可选择多个选项
name	name	定义下拉列表的名称
size	number	规定下拉列表中可见选项的数目

表 3-9 <option>属性列表

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

<select>在 HTML5 新增属性如表 3-10 所示。

表 3-10 <select> HTML5 新增属性列表

属性	值	描 述
autofocus	autofocus	规定在页面加载时下拉列表自动获得焦点
form	form_id	定义<select>所属的一个或多个表单
required	required	规定用户在提交表单前必须选择一个下拉列表中的选项

(10) 表单按钮。

在表单中,按钮的应用非常频繁,表单按钮主要分为3类:提交按钮、复位按钮和普通按钮。

① 提交按钮。

提交按钮用来将输入的表单信息提交到服务器。

其语法格式如下:

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

其中,

- type="submit": 定义提交按钮。
- name: 定义提交按钮的名称。
- value: 定义按钮的显示文字。

② 复位按钮。

复位按钮用来重置表单。

其语法格式如下:

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

其中,

- type="reset": 定义复位按钮。
- name: 定义复位按钮的名称。
- value: 定义按钮的显示文字。

注意: 复位按钮并不是清空表单信息,只是还原成默认值,例如,表单中有文本框 `<input type="text" name="name" value="张三"/>`,在该文本框中填入“李四”,当单击该复位按钮时,清除文本框中的“李四”,还原为“张三”。

③ 普通按钮。

普通按钮通常用来响应 JavaScript 事件(如 onclick),用来调用相应的 JavaScript 函数来实现各种功能。

其语法格式如下:

```
<input type="button" name="..." value="..." onclick="..." />
```

其中,

- type="button": 定义普通按钮。
- name: 定义按钮的名称。
- value: 定义按钮的显示文字。
- onclick: 通过指定脚本函数来定义按钮的行为。

网页中表单的用途很广,下面是一些典型表单的应用:

- 在用户注册某种服务时收集姓名、地址、电话号码、电子邮件和其他信息。
- 收集购买某个商品的订单信息、收集关于调查问卷信息等。

- 通过创建用户注册页面,演示 HTML 表单的综合应用。

在上面的情况下,通常要求用户输入关于个人的基本信息并提交到服务器,这些表单类似于在网站上注册用户时的表单。代码如下:

```
<html >
<head >
<meta http-equiv = "Content-Type" content = "text/html; charset = utf-8" />
<title>表单控件</title>
<style type = "text/CSS">
    input{font-family:Verdana, Arial, Helvetica, sans-serif,"宋体";}
</style>
</head >
<body >
<form method = "post" action = "# ">
<table style = "font-size:12px">
    <tr >
        <td align = "right">用户名:</td>
        <td><input type = "text" id = "username" value = "" size = "20"/></td>
    </tr >
<tr >
        <td align = "right">密码:</td>
        <td><input type = "password" id = "password" value = "" size = "20"/></td>
    </tr >
<tr >
        <td align = "right">性别:</td>
        <td >
            <input type = "radio" id = "sex" value = "male" />男
            <input type = "radio" id = "sex" value = "female" />女
        </td>
    </tr >
<tr >
        <td align = "right">国家:</td>
        <td >
            <select name = "country">
                <option id = "default" selected = "selected">
                    - 请选择您所在的国家 -
                </option >
            <option id = "China">中国</option >
                <option id = "America">美国</option >
                <option id = "Japan">日本</option >
                <option id = "France">法国</option >
                <option id = "England">英国</option >
            </select >
        </td >
    </tr >
<tr >
        <td align = "right">爱好:</td>
        <td >
            <input type = "checkbox" name = "interest" value = "music" />音乐
            <input type = "checkbox" name = "interest" value = "travel" />旅游
            <input type = "checkbox" name = "interest" value = "climbing" />登山
            <input type = "checkbox" name = "interest" value = "reading" />阅读
        </td >
    </tr >
</table >
</form >
</body >
</html >
```


2. 代码实现

```
<!DOCTYPE html >
<html lang = "en">
<head >
  <meta charset = "UTF - 8">
  <title > form test </title >
</head >
<body >
<form action = "# " method = "get" id = "form">
  <table border = "3px" width = "50 % " align = "center" cellpadding = "5px" cellspacing =
"0px">
    <tr align = "center">
      <td width = "20 % ">姓名:</td>
      <td >
        <input type = "text" name = "uname" id = "uname" />
        <span id = "uNameSpan"></span >
      </td>
    </tr>
    <tr align = "center">
      <td>密码:</td>
      <td >
        <input type = "password" name = "pwd" id = "pwd" />
        <span id = "pwdSpan"></span >
      </td>
    </tr>
    <tr align = "center">
      <td>确认密码:</td>
      <td >
        <input type = "password" name = "pwd2" id = "pwd2" />
        <span id = "pwd2Span"></span >
      </td>
    </tr>
    <tr align = "center">
      <td>邮箱:</td>
      <td >
        <input type = "text" name = "email" id = "email" />
        <span id = "emailSpan"></span >
      </td>
    </tr>
    <tr align = "center">
      <td>性别:</td>
      <td >
        <input type = "radio" name = "gender" id = "man" value = "man" />男
        <input type = "radio" name = "gender" id = "girl" value = "girl" />女
      </td>
    </tr>
    <tr align = "center">
      <td>爱好:</td>
      <td >
        <input type = "checkbox" name = "like" id = "eat" value = "eat" />吃饭
        <input type = "checkbox" name = "like" id = "play" value = "play" />玩耍
        <input type = "checkbox" name = "like" id = "sleep" value = "sleep">休息
      </td>
    </tr>
  </table >
</form >
</body >
</html >
```

```

        </td>
      </tr>
      <tr align = "center">
        <td>城市:</td>
        <td>
          <select name = "city" id = "city">
            <option value = "">请选择</option>
            <option value = "bj">北京</option>
            <option value = "sz">深圳</option>
            <option value = "gz">广州</option>
          </select>
        </td>
      </tr>
      <tr align = "center">
        <td>自我介绍:</td>
        <td>
          <textarea id = "myInfo" name = "myInfo" rows = "5" cols = "20">属于你个人的介绍,请填写</textarea>
        </td>
      </tr>
      <tr align = "center">
        <td colspan = "2">
          <input type = "submit" value = "立即注册" />
        </td>
      </tr>
    </table>
  </form>
</body>
</html>

```

3. 运行效果

通过 Chrome 查看该 HTML,上述代码运行结果如图 3-7 所示。

姓名:	<input type="text"/>
密码:	<input type="password"/>
确认密码:	<input type="password"/>
邮箱:	<input type="text"/>
性别:	<input type="radio"/> 男 <input type="radio"/> 女
爱好:	<input type="checkbox"/> 吃饭 <input type="checkbox"/> 玩耍 <input type="checkbox"/> 休息
城市:	<input type="text" value="请选择"/>
自我介绍:	<div style="border: 1px solid gray; padding: 5px; min-height: 50px;">属于你个人的介绍,请填写</div>
<input type="button" value="立即注册"/>	

图 3-7 表格实例

3.2 HTML5 新增标签

HTML5 是一个新的网络标准,目的是取代现有的 HTML 4.01、XHTML 1.0 和 DOM Level 2 HTML 标准。它希望能够减少浏览器对于需要插件的丰富性网络应用服务(plugin)

in-based Rich Internet Application, RIA), 如 Adobe Flash、Microsoft Silverlight 与 Sun JavaFX 的需求。

HTML5 提供了一些新的元素和属性, 其中有些在技术上类似了 <div> 和 标签, 例如 <nav> (网站导航块) 和 <footer>。这种标签将有利于搜索引擎的索引整理、小屏幕装置和视障人士使用。同时为其他浏览要素提供了新的功能。

一些过时的 HTML 4 标签将取消, 其中包括纯粹用于显示效果的标签, 如 和 <center>, 因为它们已经被 CSS 取代。

1. <article> 标签

<article> 标签代表网站制作中的文档、页面或应用程序中独立的、完整的、可以独自被外部引用的内容。它可以是一篇博客或者报刊中的文章、一篇论坛帖子、一段用户评论, 或其他任何独立的内容。除了内容部分, 一个 <article> 标签通常用作它自己的标题, 有时还用作它自己的脚注。

浏览器支持:

IE 9+、Firefox 和 Chrome 都支持 <article> 标签。

注意: IE 8 或更早版本的 IE 浏览器不支持 <article> 标签。

HTML5: <article></article>。

HTML4: <div></div>。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。

其使用语法如下:

```
<article> ... </article>
```

<article> 标签是可以嵌套的, 内层的内容原则上需要与外层的内容相关联。例如, 一篇博客文章中, 针对该文章的评论就可以使用嵌套 <article> 标签的方式, 用来呈现评论的 <article> 标签被包含在表示整体内容的 <article> 标签里面。

示例代码如下所示:

```
<html>
<head>
  <title> article 标签嵌套实例</title>
</head>
<body>
  <article>
    <header>
      <h1>标题</h1>
      <p>发布日期:<time pubdate = "pubdate"> 2020 年 4 月 20 号</time></p>
    </header>
    <section>
      <h2>评论区</h2>
      <article style = "background-color: #ccc; ">
        <header>
          <h4>张三评论了此条微博</h4>
          <p>12 分钟前</p>
        </header>
```

```

        <p>...</p>
    </article>
    <article style="background-color:#ccc;">
        <header>
            <h4>李四评论了这条微博</h4>
            <p>15分钟前</p>
        </header>
        <p>...</p>
    </article>
</section>
</article>
</article>
</body>
</html>

```

<article>标签应用实例效果如图 3-8 所示。

2. <aside>标签

<aside>标签用来表示当前页面或文章的附属信息部分，可以包含与当前页面或主要内容相关的引用、侧边栏、广告、nav 元素组，以及其他类似的有别于主要内容的部分。

根据目前的规范，<aside>标签有两种使用方法：

- 被包含在<article>标签中作为主要内容的附属信息部分，其中的内容可以是与当前文章有关的引用、词汇列表等。
- 在<article>标签之外使用，作为页面或站点全局的附属信息部分；最典型的形式是侧边栏(sidebar)，其中的内容可以是友情链接、附属导航或广告单元等。

浏览器支持：

IE 9+、Firefox 和 Chrome 都支持<aside>标签。

注意：IE 8 或更早版本的 IE 浏览器不支持<aside>标签。

HTML5：<aside></aside>。

HTML4：<div></div>。

示例代码如下所示：

```

<html>
<head>
    <title>aside 标签示例</title>
</head>
<body>
    <p style="color:red;">包含于 article 元素之中:</p>
    <article>
        <h1>世情国情党情是什么?</h1>
        <p>十八大以来,我们党对当前世情国情党情作出怎样的判断,又是如何朝好的方向努力,争取最好的结果的?</p>
        <aside>十八大</aside>
    </article>
    <p style="color:red;">包含于 article 元素之外:</p>

```

标题

发布日期：2020年4月20号

评论区

张三评论了这条微博

12分钟前

李四评论了这条微博

15分钟前

图 3-8 <article>标签应用实例

```

< aside >
  < h2 >友情链接</h2 >
  < ul >
    < li >< a href = " #" >百度网</a ></li >
    < li >< a href = " #" >搜狐网</a ></li >
  </ul >
</aside >
</body >
</html >

```

< aside >标签示例代码效果如图 3-9 所示。

包含于article元素之中:

世情国情党情是什么？

十八大以来，我们党对当前世情国情党情作出怎样的判断，又是如何朝好的方向努力，争取最好的结果的？

十八大

包含于article元素之外:

友情链接

- 百度网
- 搜狐网

图 3-9 aside 标签示例

3. < audio >标签

< audio >标签定义声音，比如音乐或其他音频流。到今天为止，大多数的音频文件播放是通过 Flash 来实现的。而 HTML5 定义了一个新标签< audio >，在播放音频上提供了很多方便的功能。

浏览器支持：

IE 9+、Firefox 和 Chrome 都支持< audio >标签。

注意：IE 8 或更早版本的 IE 浏览器不支持< audio >标签。

HTML5: < audio src="someaudio. wav"> audio 标签。</audio >。

HTML4: < object type="application/ogg" data="someaudio. wav">< param name="src" value="someaudio. wav"></object >。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性，该标签的特有属性如表 3-11 所示。

表 3-11 < audio >标签属性

属性	属性值	描述
autoplay	autoplay	自动播放
controls	controls	显示控件
loop	loop	自动重播
src	url	音频的 URL
preload	preload	预备播放。如果使用“autoplay”，则忽略改属性
muted	muted	静音

到目前为止,有 3 个音频格式是< audio >标签支持的,分别是 MP3、Wav 和 Ogg,其浏览器支持如表 3-12 所示。

表 3-12 支持< audio >标签的浏览器

浏览器	MP3	Wav	Ogg
Internet Explorer 9+	√	×	×
Chrome 6+	√	√	√
Firefox 3.6+	×	√	√

这 3 种音频的 MIME-type 如表 3-13 所示。

表 3-13 支持< audio >标签的 3 种音频的 MIME-type

音频格式	MIME-type
MP3	audio/mpeg
Ogg	audio/ogg
Wav	audio/wav

其使用方法如下所示:

```
< audio src = "音频"> audio 标签.</audio>
```

示例代码如下所示:

```
<html >
<head >
<title > audio 标签示例</title >
</head >
<body >
< audio id = "media" src = "a.mp3" controls >你的浏览器不支持</audio >
</body >
</html >
```

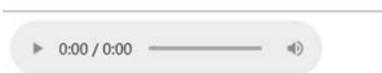


图 3-10 audio 标签示例

< audio >标签示例代码效果如图 3-10 所示。

4. < canvas >标签

< canvas >标签定义图形,使用 JavaScript 在网页上绘制 2D 图像。比如图表和其他图像。这个 HTML 元素是为了客户端矢量图形而设计的。它自己没有行为,却把一个绘图 API 展现给客户端 JavaScript,以使脚本能够将想绘制的内容都绘制到一块画布上。< canvas >标签由 Apple 公司在 Safari 1.3 Web 浏览器中引入。对 HTML 的这一根本扩展的原因在于,HTML 在 Safari 中的绘图能力也为 Mac OS X 桌面的 Dashboard 组件所使用,并且 Apple 公司希望有一种方式在 Dashboard 中支持脚本化的图形。

Firefox 1.5 和 Opera 9 都跟随了 Safari 的引领。这两个浏览器都支持< canvas >标签。我们甚至可以在 IE 中使用< canvas >标签,并在 IE 的 VML 支持的基础上用开源的 JavaScript 代码(由 Google 公司发起)来构建兼容性的画布。< canvas >的标准化的努力由

一个 Web 浏览器厂商的非正式协会在推进,目前< canvas >已经成为 HTML 5 草案中一个正式的标签。

1) canvas 标签和 SVG 以及 VML 之间的差异

< canvas >标签和 SVG 以及 VML 之间的一个重要的不同是,< canvas >有一个基于 JavaScript 的绘图 API,而 SVG 和 VML 使用一个 XML 文档来描述绘图。

这两种方式在功能上是等同的,任何一种都可以用另一种来模拟。从表面上看,它们很不相同,可是,每一种都有强项和弱点。例如,SVG 绘图很容易编辑,只要从其描述中移除元素就行。要从同一图形的一个< canvas >标签中移除元素,往往需要擦掉绘图重新绘制它。< canvas >与 SVG 具体应用比较如表 3-14 所示。

表 3-14 Canvas 与 SVG 具体应用比较

Canvas	SVG
依赖分辨率	不依赖分辨率
不支持事件处理器	支持事件处理器
弱的文本渲染能力	最适合带有大型渲染区域的应用程序(如谷歌地图)
能够以 .png 或 .jpg 格式保存结果图像	复杂度高会减慢渲染速度(任何过度使用 DOM 的应用都不快)
最适合图像密集型的游戏,其中的许多对象会被频繁重绘	不适合游戏应用

2) 如何使用< canvas >标签绘图

大多数 Canvas 绘图 API 都没有定义在< canvas >标签本身上,而是定义在通过画布的 getContext()方法获得的一个“绘图环境”对象上。

Canvas API 也使用了路径的表示法。但是,路径由一系列的方法调用来定义,而不是描述为字母和数字的字符串,比如调用 beginPath()和 arc()方法。

一旦定义了路径,其他的方法,如 fill(),都是对此路径进行操作。绘图环境的各种属性,比如 fillStyle,说明了这些操作如何使用。

注意: Canvas API 非常紧凑的一个原因是它没有对绘制文本提供任何支持。要把文本加入一个 Canvas 图形,必须先自己绘制它再用位图图像合并它,或者在< canvas >上方使用 CSS 定位来覆盖 HTML 文本。

浏览器支持:

IE 9、Firefox、和 Chrome 都支持< canvas >标签。

注意: IE 8 或更早版本的 IE 浏览器不支持< canvas >标签。

HTML5: < canvas id = "myCanvas" width = "200" height = "200"></ canvas >。

HTML4: < object data = "inc/hdr. svg" type = "image/svg + xml" width = "200" height = "200"></ object >。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性,< canvas >标签的特有属性如表 3-15 所示。

表 3-15 < canvas >标签的特有属性

属性	属性值	描 述
height	height	设置 canvas 标签高度
width	width	设置 canvas 标签宽度

其使用方法如下所示：

```
< canvas >...</ canvas >
```

示例代码如下所示：

```
< html >
< head >
  < title >< audio > Canvas 标签示例</ title >
</ head >
< body >
  < canvas id = "myCanvas" width = "200" height = "200" style = "border:solid 1px #000;" >
    您的浏览器不支持 canvas, 建议使用最新版的浏览器</ canvas >
</ body >
</ html >
```

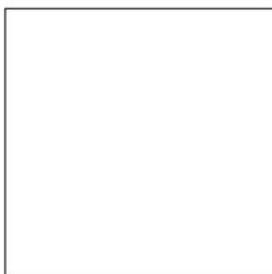


图 3-11 < canvas >标签示例

< canvas >标签示例页面如图 3-11 所示。

5. < command >标签

< command >标签定义一个命令按钮，比如单选按钮、复选框或按钮。只有 Internet Explorer 9(更早或更晚的版本都不支持)支持< command >标签。

HTML5：< command onclick = cut () " label = "cut">。

HTML4：none。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性，< command >标签的特有属性如表 3-16 所示。

表 3-16 < command >标签特有的属性

属性	属性值	描 述
type	checkbox command radio	定义该< command >的类型。默认是 command
radiogroup	groupname	显示控件
label	text	为< command >定义可见的 label
icon	url	定义所显示的图像的 url
disabled	disabled	定义< command >是否可用
checked	checked	定义是否被选中。仅用 radio 或 checkbox 类型

其使用方法如下所示：

```
<menu><command onclick = "事件"> Click here.</command></menu>
```

因为该标签在新版浏览器中没有可以使用的场景，所以不在此举例，对该标签只需了解。

6. <datalist>标签

作为 HTML5 中的新标签，<datalist> 标签实现定义选项列表，这个标签在使用的时候配合<input> 标签来使用，可以实现类似百度搜索框一样的搜索提示功能。<datalist> 与<select> 标签类似。作为选项列表，它们都需要借助<option> 标签来实现选项列表的每一项的内容。

浏览器支持：

IE 10、Firefox 和 Chrome 都支持<datalist> 标签。

注意：IE 9 和更早版本的 IE 浏览器以及 Safari 不支持<datalist> 标签。

HTML5：<datalist></datalist>。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性。

其使用方法如下所示：

```
<input id = "A" list = "B"/><datalist id = "B"><option value = "值"></datalist>
```

示例代码如下所示：

```
<html>
<head>
  <title> datalist 标签示例</title>
</head>
<body>
  <input type = "text" list = "test" />
  <datalist id = "test">
    <option value = "Adidas"></option>
    <option value = "Baidu"></option>
    <option value = "Cctv"></option>
    <option value = "Dreamweaver"></option>
    <option value = "Eclipse"></option>
  </datalist>
</body>
</html>
```

<datalist> 标签示例代码运行效果如图 3-12 所示。

7. <details> 标签

<details> 标签定义元素的细节，用户可进行查看，或通过单击进行隐藏。与<summary> 一起使用可制作<details> 的标题。该标题对用户是可见的，当在其上单击时可打开或关闭<details>。任何形式的内容都能被放在<details> 标签中。



图 3-12 <datalist> 标签示例

浏览器支持：

目前，只有 Chrome 和 Safari 6 支持< details >标签。

HTML5：< details ></ details >。

HTML4：< dl style="display: hidden"></ dl >。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性，< details >标签的特有属性如表 3-17 所示。

表 3-17 < details >标签特有属性

属性	属性值	描 述
open	open	定义 details 是否可见

其使用方法如下所示：

```
< details >...</ details >
```

示例代码如下所示：

```
< html >
< head >
< meta charset = "utf - 8">
< title > details 标签示例</ title >
</ head >
< body >
< details >
< summary > Copyright 1999 - 2011.</ summary >
< p > - by Refsnes Data. All Rights Reserved.</ p >
< p > All content and graphics on this Web site are the property of the company Refsnes Data.</ p >
</ details >
< p >< b >注意:</ b >目前,只有 Chrome 和 Safari 6 支持 &lt; details &gt; 标签.</ p >
</ body >
</ html >
```

< details >标签示例代码运行效果如图 3-13 所示。

▼Copyright 1999-2011.
- by Refsnes Data. All Rights Reserved.
All content and graphics on this Web site are the property of the
company Refsnes Data.
注意：目前，只有 Chrome 和 Safari 6 支持 < details > 标签。

图 3-13 < details >标签示例

8. < embed > 标签

< embed >标签可以用来插入各种多媒体内容，格式可以是 Midi、Wav、AIFF、AU、MP3 等等，Netscape 及新版的 IE 都支持。

浏览器支持：

所有主流浏览器都支持< embed >标签。

HTML5：< embed src="horse. wav" />。

HTML4: `<object data="flash.swf" type="application/x-shockwave-flash"> </object>`。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性, `<embed>` 标签的特有属性如表 3-18 所示。

表 3-18 `<embed>` 标签特有属性

属性	属性值	描述
height	pixels	设置嵌入内容的高度
src	url	嵌入内容的 URL
type	type	定义嵌入内容的类型
width	pixels	设置嵌入内容的高度

其使用方法如下:

```
<embed src = "嵌入内容的 url" />
```

9. `<figcaption>` 标签

`<figcaption>` 标签定义 `figure` 元素的标题。`<figcaption>` 标签应该被置于 `figure` 元素的第一个或最后一个子元素的位置。

浏览器支持:

IE 9、Firefox、和 Chrome 支持 `<figcaption>` 标签。

注意: IE 8 或更早版本的 IE 浏览器不支持 `<figcaption>` 标签。

HTML5: `<figure><figcaption></figcaption></figure>`。

HTML4: 无。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。

其使用方法如下:

```
<figure><figcaption>标题内容</figcaption></figure>
```

示例代码如下所示:

```
<html>
<head>
<title> &lt;figcaption &gt; 标签示例</title>
</head>
<body>
<figure>
<figcaption>一朵花</figcaption>
<img src = "1.jpg" width = "450" height = "234" />
</figure>
</body>
</html>
```

通过 IE 查看该 HTML, 效果如图 3-14 所示。

10. `<figure>` 标签

定义一组媒体内容(图像、图表、照片、代码等)以及它们的标题。如果被删除, 则不应



图 3-14 <figcaption>标签显示效果

文档流产生影响。

浏览器支持：

IE 9、Firefox 和 Chrome 支持<figure>标签。

注意：IE 8 或更早版本的 IE 浏览器不支持<figure>标签。

HTML5：<figure><figcaption>PRC</figcaption><p>The People's Republic of China was born in 1949...</p></figure>。

HTML4：<dl><h1>PRC</h1><p>The People's Republic of China was born in 1949...</p></dl>。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性。

其使用方法如下：

```
<figure><figcaption>标题内容</figcaption>内容</figure>
```

示例代码如下所示：

```
<html>
<head>
<title> &lt; figure &gt; 标签示例</title>
</head>
<body>
<p>The Pulpit Rock is a massive cliff 604 metres (1982 feet) above Lysefjorden, opposite
the Kjerag plateau, in Forsand, Ryfylke, Norway. The top of the cliff is approximately 25 by 25
metres (82 by 82 feet) square and almost flat, and is a famous tourist attraction in Norway.</p>
<figure>
<img src = "2.jpg" alt = "The Pulpit Rock" width = "304" height = "228">
</figure>
</body>
</html>
```

<figure>标签显示效果如图 3-15 所示。

11. <footer>标签

<footer>标签定义文档或文档→部分区域的页脚。典型地，它会包含创作者的姓名、文档的创作日期以及联系信息，在一个文档中，可以定义多个<footer>标签。

浏览器支持：

IE 9、Firefox、和 Chrome 支持<footer>标签。

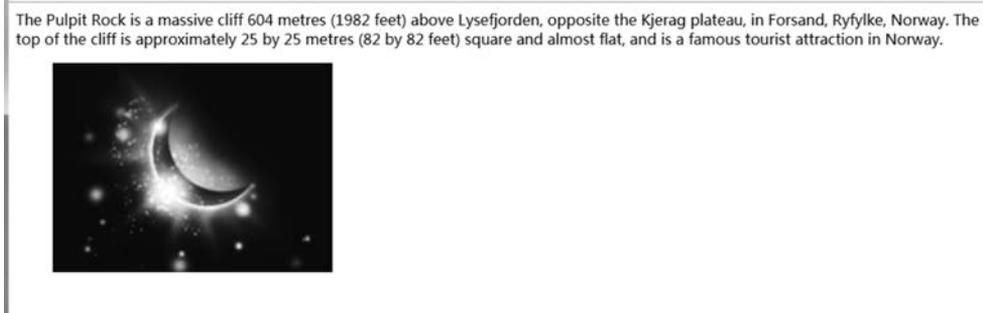


图 3-15 < figure > 标签显示效果

注意：IE 8 或更早版本的 IE 浏览器不支持 < footer > 标签。

HTML5: < footer ></footer >。

HTML4: < div ></div >。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性。

其使用方法如下：

```
< footer >页脚内容</footer >
```

示例代码如下所示：

```
< html >
< head >
  < title > footer 标签示例</title >
</head >
< body >
  < footer >
    < p > Posted by: Hege Refsnes </p >
    < p >< time pubdate datetime = "2020 - 04 - 22"></time ></p >
  </footer >
</body >
</html >
```

< footer > 标签显示效果如图 3-16 所示。

12. < header > 标签

< header > 标签定义文档或者文档的一部分区域的页眉。在一个文档中，可以定义多个 < header > 标签，但是 < header > 标签不能被放在 < footer >、< address > 或者另一个 < header > 标签内部。

浏览器支持：

IE 9、Firefox、Opera、Chrome 和 Safari 支持 < header > 标签。

注意：IE 8 或更早版本的 IE 浏览器不支持 < header > 标签。

HTML5: < header ></header >。

HTML4: < div ></div >。



图 3-16 < footer > 标签显示效果

属性/属性值/描述：支持 HTML5 的全局属性和事件属性。
其使用方法如下：

```
<header>页眉内容</header>
```

示例代码如下所示：

```
<html>
<head>
<title>&lt; header &gt; 标签示例</title>
</head>
<body>
<article>
  <header>
    <h1>Internet Explorer 9</h1>
    <p><time pubdate datetime = "2020 - 04 - 22"></time></p>
  </header>
  <p>Windows Internet Explorer 9(缩写为 IE9 )是在 2011 年 3 月 14 日 21:00 发布的.</p>
</article>
</body>
</html>
```

<header>标签显示效果如图 3-17 所示。



图 3-17 <header>标签显示效果

13. <hgroup> 标签

<hgroup>标签用来指定组合网页或区段的标题,修改 hgroup 样式后,被它包围的 h1、h4 之类的标题元素就会同时继承它设置的样式。

浏览器支持：

IE 9、Firefox、Opera、Chrome 和 Safari 支持 <hgroup> 标签。

HTML5：<hgroup></hgroup>。

HTML4：<div></div>。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性。

其使用方法如下：

```
<hgroup>
<h1>标题一</h1>
</hgroup>
```

示例代码如下所示：

```

<html>
<head>
<title> hgroup 标签示例</title>
</head>
<body>
<hgroup>
<h1> Welcome to my WWF </h1>
<h2> For a living planet </h2>
</hgroup>
<p> The rest of the content...</p>
</body></html>

```

<hgroup>标签显示效果如图 3-18 所示。

14. <keygen> 标签

<keygen>标签规定用于表单的密钥对生成器字段。当提交表单时,私钥存储在本地,公钥发送到服务器。

浏览器支持:

Firefox 和 Chrome 支持<keygen>标签。

HTML5: <keygen>。

HTML4: <div></div>。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。<keygen>标签的特有属性如表 3-19 所示。

Welcome to my WWF

For a living planet

The rest of the content...

图 3-18 <hgroup>标签显示效果

表 3-19 <keygen>标签特有属性

属性	属性值	描述
autofocus	autofocus	使 keygen 字段在页面加载时获得焦点
challenge	challenge	如果使用,则将 keygen 的值设置为在提交时询问
disabled	disabled	禁用 keytag 字段
form	formname	定义该 keygen 字段所属的一个或多个表单
keytype	rsa	定义 keytype。rsa 生成 RSA 密钥
name	fieldname	定义 keygen 元素的唯一名称,用于在提交表单时搜集字段的值。

其使用方法如下:

```
<keygen from = value">
```

示例代码如下所示:

```

<html>
<head>
<title> keygen 标签示例</title>
</head>
<body>
<form action = "/example/html5/demo_form.asp" method = "get">

```

```

用户名:<input type="text" name="usr_name" />
加密:<keygen name="security" />
<input type="submit" />
</form>
</body>
</html>

```

<keygen>标签显示效果如图 3-19 所示。



图 3-19 <keygen>标签显示效果

15. <mark>标签

<mark>标签主要用来在视觉上向用户呈现那些需要突出的文字。<mark>标签的一个比较典型的应用就是在搜索结果中向用户高亮显示搜索关键词。

浏览器支持：

IE 9+、Firefox 和 Chrome 支持<mark>标签。

注意：Internet Explorer 8 及更早版本不支持<mark>标签。

HTML5：<mark></mark>。

HTML4：。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性。

其使用方法如下：

```
<mark>内容</mark>
```

示例代码如下所示：

```

<html>
<head>
<title>mark 标签示例</title>
</head>
<body>
<p>Do not forget to buy <mark>milk</mark> today.</p>
</body>
</html>

```



图 3-20 <mark>标签显示效果

<mark>标签显示效果如图 3-20 所示。

16. <meter>标签

<meter>标签定义度量衡。仅用于已知最大值和最小值的度量。必须定义度量的范围，既可以在元素的文本中，也可以在 min/max 属性中定义。

浏览器支持：

Firefox 和 Chrome 支持<meter>标签。

HTML5: `< meter ></ meter >`。

HTML4: 无。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。`< meter >`标签的特有属性如表 3-20 所示。

表 3-20 `< meter >`标签特有属性

属性	属性值	描 述
high	number	定义度量的值位于哪个点,被界定为高的值
low	number	定义度量的值位于哪个点,被界定为低的值
max	number	定义最大值。默认值是 1
min	number	定义最小值。默认值是 0
optimum	number	定义什么样的度量值是最佳的值。如果该值高于 high 属性的值,则意味着值越大越好。如果该值低于 low 属性的值,则意味着值越小越好
value	number	定义度量的值

其使用方法如下:

```
< meter value = "值">内容</ meter >
```

示例代码如下所示:

```
< html >
< head >
< title >标签示例</ title >
</ head >
< body >
< meter value = "2" min = "0" max = "10"> 2 out of 10 </ meter >< br >
< meter value = "0.6"> 60 % </ meter >
</ body >
</ html >
```

`< meter >`标签显示效果如图 3-21 所示。

17. `< nav >`标签

`< nav >`标签定义导航链接的部分。并不是所有的 HTML 文档都要用到`< nav >`标签。`< nav >`标签只是作为标注一个导航链接的区域。在不同的手机或者 PC 上可以指定导航链接是否显示,以适应不同屏幕的需求。

浏览器支持:

目前大多数浏览器支持`< nav >`标签。

HTML5: `< nav ></ nav >`。

HTML4: `< ul ></ ul >`。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。

其使用方法如下:



图 3-21 `< meter >`标签显示效果

```
<nav> 内容</nav>
```

示例代码如下所示：

```
<html>
<head>
<title> nav 标签示例</title>
</head>
<body>
<header>
</header>
  <article>
    <hgroup>
      <h2>文章的标题</h2>
      <nav>
        <ul>
          <li><a href = "# p1">段一</a></li>
          <li><a href = "# p2">段二</a></li>
          <li><a href = "# p3">段三</a></li>
        </ul>
      </nav>
    </h2></hgroup>
    <p id = "p1">段一</p>
    <p id = "p2">段二</p>
    <p id = "p3">段三</p>
  </article>
  <footer>
</footer>
</body>
</html>
```

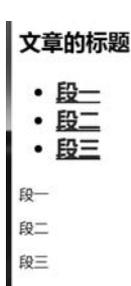


图 3-22 <nav>标签显示效果

<nav>标签显示效果如图 3-22 所示。

18. <output>标签

<output>标签用于计算结果的输出显示（比如执行脚本的输出）。

浏览器支持：

Firefox 和 Opera 浏览器都支持<output>标签。

注意：Internet Explorer 浏览器不支持<output>标签。

HTML5：<output></output>。

HTML4：。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性。

<output>标签的特有属性如表 3-21 所示。

表 3-21 <output>标签属性

属性	属性值	描述
for	id of Another element	定义输出域相关的一个或多个元素

续表

属性	属性值	描 述
form	formname	定义输入字段所属的一个或多个表单
name	unique name	定义对象的唯一名称(提交表单时使用)

其使用方法如下：

```
<output name = "name", from = "from_id", for = "element_id"></output >
```

示例代码如下所示：

```
<html >
<head >
<title>标签示例</title>
</head >
<body >
<form oninput = "x.value = parseInt(a.value) + parseInt(b.value)"> 0
  <input type = "range" id = "a" value = "50"> 100
  + <input type = "number" id = "b" value = "50">
= <output name = "x" for = "a b"></output >
</body >
</html >
```

<output >标签显示效果如图 3-23 所示。

图 3-23 output 标签显示效果

19. <progress > 标签

<progress >标签表示运行中的进程。可以使用

<progress >标签来显示 JavaScript 中耗费时间的函数的进程。

浏览器支持：

IE 10、Firefox 和 Chrome 支持<progress >标签。

注意：IE 9 或者更早版本的 IE 浏览器不支持<progress >标签。

HTML5：<progress ></progress >。

HTML4：无。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性。<progress >标签的特有属性如表 3-22 所示。

表 3-22 <progress >标签属性

属性	属性值	描 述
max	number	定义完成的值
min	number	定义进程的当前值

其使用方法如下：

```
<progress value = "值" max = "最大值"></progress >
```

示例代码如下所示：

```

<html>
<head>
<title> &lt; progress &gt; 标签示例</title>
</head>
<body>
  <p>下载进度:</p>
  <progress value = "33" max = "100"></progress>
  <p>其实一个 &lt; progress &gt; 标签就实现了一个进度条
  <br/>并且我们可以控制进度利用的是 value 这个属性,max 表示最大的长度</p>
</body>
</html>

```

< progress >标签显示效果如图 3-24 所示。

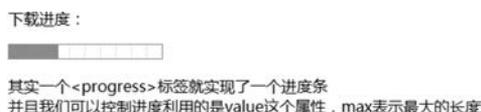


图 3-24 < progress >标签显示效果

20. < rp > 标签

< rp >标签在< ruby >标签中使用,以定义不支持< ruby >标签的浏览器所显示的内容。< ruby >标签中是中文注音或字符。在东亚地区使用,显示的是东亚字符的发音。

浏览器支持:

IE 9+、Firefox 和 Chrome 支持 < rp > 标签。

注意: IE 8 或更早版本的 IE 浏览器不支持 < rp > 标签。

HTML5: < ruby > < rt > < rp > </rp > </rt > </ruby >。

HTML4: 无。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。

其使用方法如下:

```
< ruby > < rt > < rp > </rp > </rt > </ruby >
```

示例代码如下所示:

```

<html>
<head>
<title> &lt; rp &gt; 标签示例</title>
</head>
<body>
<ruby>
  汉 <rp></rp><rt> Kan </rt><rp></rp>
  字 <rp></rp><rt> ji </rt><rp></rp>
</ruby>
</body>
</html>

```

< rp >标签显示效果如图 3-25 所示。

21. <rt> 标签

<rt>标签定义字符(中文注音或字符)的解释或发音,将<rt>标签与<ruby>和<rp>标签一起使用。

浏览器支持:

IE 9+、Firefox 和 Chrome 支持<rt>标签。

注意: IE 8 或更早版本的 IE 浏览器不支持<rt>标签。

HTML5: <ruby>汉 <rt> ㄏ ǎ ㄣˇ </rt></ruby>。

HTML4: 无。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。

其使用方法如下:

```
<ruby><rt><rp></rp></rt></ruby>
```

示例代码如下所示:

```
<html>
<head>
<title>&lt;rt &gt; 标签示例</title>
</head>
<body>
<ruby>
汉 <rt> ㄏ ǎ ㄣˇ </rt>
</ruby>
</body>
</html>
```

Kan ji
汉字

图 3-25 <rp>标签显示效果(1)

<rt>标签显示效果如图 3-26 所示。

22. <ruby> 标签

定义 ruby 标签(中文注音或字符)。在东亚地区使用,显示的是东亚字符的发音。ruby 标签由一个或多个标签字符(需要一个解释/发音)和一个提供该信息的<rt>标签组成,还包括可选的<rp>标签,

图 3-26 <rt>标签显示效果(2)

定义当浏览器不支持<ruby>标签时显示的内容。

浏览器支持:

IE 9+、Firefox 和 Chrome 支持 <ruby> 标签。

注意: IE 8 或更早版本的 IE 浏览器不支持 <ruby> 标签。

HTML5: <ruby><rt><rp></rp></rt></ruby>。

HTML4: 无。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。

其使用方法如下:

```
<ruby><rt><rp></rp></rt></ruby>
```

示例代码如下所示:

ㄏ ǎ ㄣˇ
漢

```

<html>
<head>
<title>&lt ruby &gt 标签示例</title>
</head>
<body>
  <ruby>
    <rb>吉林大学</rb>
    <rp></rp>
    <rt>きつ りん だ い が く</rt>
    <rp></rp>
  </ruby>
<p></p>
  <ruby>
    <rb>吉林大学</rb>
    <rp></rp>
    <rt>j i l i n d a x u e</rt>
    <rp></rp>
  </ruby>
</body>
</html>

```

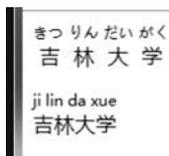


图 3-27 <ruby> 标签显示效果(3)

<ruby> 标签显示效果如图 3-27 所示。

23. <section> 标签

<section> 标签定义文档中的节(区段)。比如章节、页眉、页脚或文档中的其他部分,<section> 标签用来表现普通的文档内容或应用区块。一个<section> 通常由内容及其标题组成,但<section> 标签并非一个普通的容器元素; 当一个容器需要被直接定义样式或通过脚本定义行为时,推荐使用<div> 标签而非

<section>。该标签是成对出现的,以<section> 开始,以</section> 结束,<section> 标签通常带有一个标题和一个内容块。

1) <article> 与 <section> 的异同

<section> 和 <article> 可以互相嵌套,也就是说,它们没有上下级关系,<section> 可以包含<article>,<article> 也可以包含<section>。

两者在使用上都差不多,都可以有 h1、h2、h3,也都有一个主体,那么应该怎么来区分它们呢? 其实很简单,只要从字面上理解就可以了。

(1) <article> 是文章,文章就是一段完整的、独立的内容。

(2) <section> 是块,某种意义上可以理解为<div>,但是比<div> 的意思更加明确。

2) <section> 和 <div> 的异同

(1) <section> 和 <div> 都可以对内容进行分块,但是<section> 是进行有意义的分块,无意义的分块应该由<div> 来做,例如,用作设置样式的页面容器。

(2) <section> 内部必须有标题,标题也代表了 section 的意义所在。

3) 使用 <section> 标签需要注意的地方

(1) 不要将<section> 作为用来设置样式或行为的“钩子”容器,那是<div> 的工作。

(2) 如果<article>、<aside> 或 <nav> 能够满足要求,则不要使用<section>。

(3) 不要对没有标题的内容区块使用< section >。

浏览器支持：

IE 9+、Firefox 和 Chrome 支持< section >标签。

注意：IE 8 或更早版本的 IE 浏览器不支持< section >标签。

HTML5：< section ></section >。

HTML4：< div ></div >。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性。< section >标签的特有属性如表 3-23 所示。

表 3-23 < section >标签特有属性

属性	属性值	描 述
cite	URL	当< section >摘自 Web 时使用

其使用方法如下：

```
< section >内容</section >
```

示例代码如下所示：

```
<html >
<head >
<title > section 标签示例</title >
</head >
<body >
<section >
  <h1 > WWF </h1 >
  <p > The World Wide Fund for Nature (WWF) is an international organization working on issues regarding the conservation, research and restoration of the environment, formerly named the World Wildlife Fund. WWF was founded in 1961.</p >
</section >
<section >
  <h1 > WWF's Panda symbol </h1 >
  <p > The Panda has become the symbol of WWF. The well - known panda logo of WWF originated from a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same year of the establishment of WWF.</p >
</section >
</body >
</html >
```

< section >标签显示效果如图 3-28 所示。

24. < source >标签

< source >标签为媒体元素(比如< video >和< audio >)定义媒体资源,< source >标签规定视频/音频文件根据浏览对媒体类型或者编解码器的支持进行选择媒体资源。

浏览器支持：

IE 9+、Firefox 和 Chrome 都支持< source >标签。

注意：IE 8 或更早版本的 IE 浏览器都不支持< source >标签。

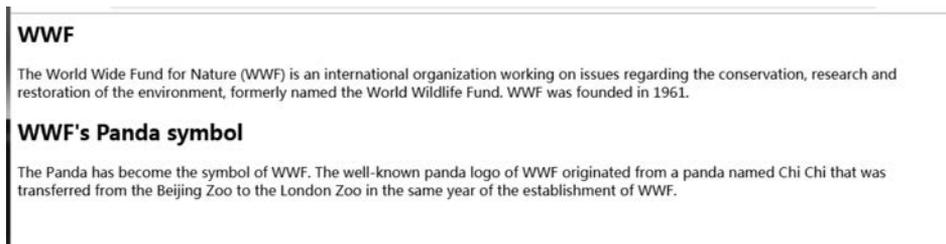


图 3-28 section 标签显示效果

HTML5: < source >。

HTML4: < param >。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。< source >标签的特有属性如表 3-24 所示。

表 3-24 < source >标签特有属性

属性	属性值	描述
media	Media query	定义媒体资源的类型,供浏览器决定是否下载
src	url	媒体的 URL
type	Numeric value	定义播放器在音频流中播放起始位置。默认是从开头播放

其使用方法如下所示:

```
< source src = "url" type = "类型值">
```

示例代码如下所示:

```
<html >
<head >
<title > source 标签示例</title >
</head >
<body >
<audio controls >
< source src = "不顾一切地爱 - 李圣杰.mp3" type = "audio/mpeg">
您的浏览器不支持 audio 元素。
</audio >
</body >
</html >
```



图 3-29 < source >标签显示效果

通过 IE 查看该 HTML, < source >标签显示效果如图 3-29 所示。

25. < summary > 标签

< summary > 标签包含 < details > 元素的标题, < details > 标签用于描述有关文档或文档片段的详细信息。

"summary" 元素应该是 < detail > 标签的第一个子元素。

浏览器支持:

目前只有 Chrome 支持 < summary > 标签。

HTML5: `<details><summary> HTML 5 </summary> This document teaches you everything you have to learn about HTML 5.</details>`。

HTML4: 无。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。

其使用方法如下所示:

```
<details><summary>标题</summary>内容</details>
```

示例代码如下所示:

```
<html>
<head>
<title> &lt;summary &gt; 标签示例</title>
</head>
<body>
<details>
<summary> HTML 5 </summary>
This document teaches you everything you have to learn about HTML 5.
</details>
</body>
</html>
```

通过 Chrome 浏览器查看该 HTML, `<summary>` 标签显示效果如图 3-30 所示。

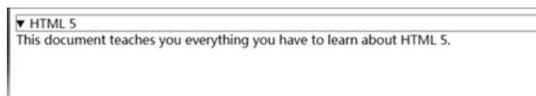


图 3-30 `<summary>` 标签显示效果

26. `<time>` 标签

`<time>` 标签定义日期或时间, 或者两者, `<time>` 标签定义公历的时间(24 小时制)或日期, 时间和时区是可选的。该标签能够以机器可读的方式对日期和时间进行编码。举例来说, 用户代理能够把生日提醒或排定的事件添加到用户日程表中, 搜索引擎也能够生成更智能的搜索结果。

浏览器支持:

IE 9+、Firefox 和 Chrome 都支持 `<source>` 标签。

HTML5: `<time></time>`。

HTML4: ``。

属性/属性值/描述: 支持 HTML5 的全局属性和事件属性。`<time>` 标签的特有属性如表 3-25 所示。

表 3-25 `<time>` 标签特有属性

属性	属性值	描述
datetime	datetime	规定日期或时间。否则, 由元素的内容给定日期或时间
pubdate	pubdate	指示 <code><time></code> 标签中的日期或时间是文档的发布日期

其使用方法如下所示：

```
<time></time>
```

示例代码如下所示：

```
<html>
<head>
<title>&lt time &gt 标签示例</title>
</head>
<body>
<p>我们在每天早上<time>9:00</time>开始营业.</p>
<p>我在<time datetime="2020-02-14">情人节</time>有个约会.</p>
<p><strong>注意:</strong> Internet Explorer 8 及更早版本不支持<time>标签.</p>
</body>
</html>
```

<time>标签显示效果如图 3-31 所示。

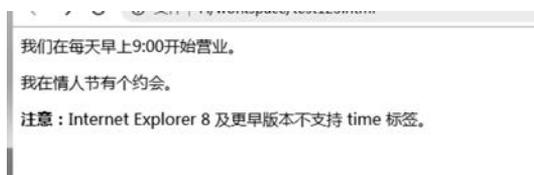


图 3-31 <time>标签显示效果

27. <video> 标签

<video>标签定义视频,比如电影片段或其他视频流,可以在<video>和</video>标签之间放置文本内容,这样不支持<video>标签的浏览器就可以显示出该标签的信息。

浏览器支持：

IE 9+、Firefox 和 Chrome 都支持 <video> 标签。

注意：IE 8 或更早版本的 IE 浏览器不支持 <video> 标签。

HTML5: <video src="movie.ogv" controls="controls">您的浏览器不支持 video 标签.</video>。

HTML4: <object type="video/ogg" data="movie.ogv"><param name="src" value="movie.ogv"></object>。

属性/属性值/描述：支持 HTML5 的全局属性和事件属性。<video>标签的特有属性如表 3-26 所示。

表 3-26 <video>标签特有属性

属性	属性值	描述
autoplay	autoplay	自动播放
controls	controls	显示控件
height	pixels	设置视频播放器的高度
loop	loop	自动重播

续表

属性	属性值	描述
preload	preload	预备播放。如果使用 autoplay,则忽略该属性
src	url	视频的 URL
width	pixels	设置视频播放器的宽度

目前,< video >标签支持 3 种视频格式: MP4、WEBM、Ogg。如表 3-27 所示。

表 3-27 支持格式

浏览器	MP4	WEBM	Ogg
Internet Explorer	YES	NO	NO
Chrome	YES	YES	YES
Firefox	YES(从 Firefox 21 版本开始) (Linux 系统从 Firefox 30 开始)	YES	YES

- MP4 表示 MPEG 4 文件使用 H264 视频编解码器和 AAC 音频编解码器。
 - WEBM 表示 WEBM 文件使用 VP8 视频编解码器和 Vorbis 音频编解码器。
 - Ogg 表示 Ogg 文件使用 Theora 视频编解码器和 Vorbis 音频编解码器。
- 音频格式的 MIME 类型如表 3-28 所示:

表 3-28 MIME 类型

格式	MIME 类型
MP4	video/mp4
WEBM	video/webm
Ogg	video/ogg

其使用方法如下所示:

```
<video src = "URL" controls = "controls">您的浏览器不支持 video 标签.</video >
```

示例代码如下所示:

```
<html >
<head >
<title > &lt video &gt 标签示例</title >
</head >
<body >
<video width = "320" height = "240" controls >
  < source src = "movie.mp4" ttype = "video/mp4">
</video >
</body >
</html >
```

< video >标签显示效果如图 3-32 所示。



图 3-32 < video >标签显示效果

本章小结

本章的主要知识点如下：

- 表格是 HTML 的高级控件之一。表格可以清晰地展现数据之间的关系，便于对比分析。
- HTML 中与表格有关的 10 个标签是：< table >、< caption >、< th >、< tr >、< td >、< thead >、< tbody >、< tfoot >、< col >、< colgroup >。
- 表单由表单标签、表单域、表单按钮组成。
- 创建表单最关键的是掌握 3 个要素，即表单控件、action 属性和 method 属性。
- 向服务器传递数据的 HTTP 方法，主要有 get 和 post 两种方法，默认值是 get。
- 表单域包含了文本框、密码框、隐藏域、多行文本框、复选框、单选按钮、下拉列表框和文件上传框等，用于采集用户输入或选择的数据。
- 表单按钮主要分为 3 类：提交按钮、重置按钮和普通按钮。
- 使用框架可以将浏览器窗口划分成多个相互独立的区域。
- HTML 框架既可以横向分隔，也可以纵向分隔。
- 使用框架技术可以方便地实现页面导航功能。
- < canvas >标签定义图形，使用 JavaScript 在网页上绘制 2D 图像。
- < canvas >标签和 SVG 以及 VML 之间的不同点。