

3

第 3 章 静态网页开发技术

JavaScript

JSP

【学习目标】

- 熟练使用 HTML 编写静态网页。
- 掌握使用 JavaScript 进行客户端验证的方法。
- 熟悉 CSS 的使用。

3.1 HTML

3.1.1 HTML 简介

1. 超文本标记语言

超文本标记语言(hypertext markup language, HTML)不是一种编程语言,而是一种用来描述超本文档的标记语言。用 HTML 编写的超本文档称为 HTML 文档,它独立于操作系统平台。所谓超本文档,是指在其中可以加入图片、声音、动画、视频等内容,并且可以利用超级链接非常方便地从一个文件跳转到网络上其他主机的另一个文件。

HTML 文档在普通文本上加上各种标签,使其达到预期的显示效果。当浏览器打开一个 HTML 文档时,会根据标签的含义显示 HTML 文档中的文本。标签由“<标签名字 属性>”表示,例如:

```
<p align="center">This is my first html file. </p>
```

2. HTML 标签的结构形态

1) <标签>元素</标签>

<标签>是标签的开始,</标签>表示标签的结束,标签的作用范围到</标签>为止。例如:<h2>demo</h2>的作用是将 demo 这段文本按<h2>标签规定的含义来显示,即以二号标题来显示。而<h2>和</h2>之外的文本不受这对标签的影响。





2) <标签 属性名="属性值">元素</标签>

标签的属性表示标签的一些附加信息,一个标签可以包含多个属性,各属性之间无先后次序,但需要用空格分开。例如:在<body background="back.gif" text="red">中,background 属性用来表示 HTML 文档的背景图片,text 属性用来表示文本的颜色。

3) <标签/>

单独出现的标签既是开始标签,又是结束标签,也称为空标签。例如:。

3. HTML 文档结构

HTML 文档分为文档头和文档体两部分。文档头对文档进行了一些必要的定义,文档体中才是要显示的各种文档信息。

```
<html>
  <head>头部信息,例如标题等</head>
  <body>正文部分</body>
</html>
```

其中<html>在最外层,表示这对标签间的内容是 HTML 文档。一些 HTML 文档经常省略<html>标签,因为.html 或.htm 文件被 Web 浏览器默认为是 HTML 文档。<head>之间包括文档的头部信息,如文档的标题等,若不需头部信息则可省略此标签。<body>标签一般不省略,表示正文内容的开始。下面是一个简单的超文本文档的源代码:

```
<html>
  <head>
    <title>一个简单的 HTML 示例</title> } 文档头
  </head>
  <body>
    <center>
      <h3>欢迎光临</h3>
      <br>
      <hr>
      <font size="2">
        这是我的主页,欢迎大家访问!
      </font>
    </center>
  </body> } 文档体
</html>
```

3.1.2 HTML 常用标签

1. <html> 标签

<html> 标签用来通知浏览器该文件是 html 文件。<html></html> 标签任意删去

一个或全部,都不会影响显示效果。在<html>和</html>标签的前后任意加一些字符是错误的。

【例 3-1】 html.html

```
<html>
<head>
  <title>html 标签示例</title>
</head>
<body>这是一个 HTML 文档</body>
</html>
```

2. <body> 标签

1) text 属性(设定文字颜色)

【例 3-2】 body_text.html

```
<html>
<head>
  <title>body 标签 text 属性示例</title>
</head>
<body text="red">文本是红色的</body>
</html>
```

2) bgcolor 属性(设定背景色)

【例 3-3】 body_bgcolor.html

```
<html>
<head>
  <title>body 标签 bgcolor 属性示例</title>
</head>
<body bgcolor="#00AA33"></body>
</html>
```

在 HTML 中,颜色可以用“#RRGGBB”形式的十六进制数表示,也可以用下列单词预定义色彩: Black(黑色)、Olive(橄榄色)、Teal(黑绿色)、Red(红色)、Blue(蓝色)、Maroon(栗色)、Navy(藏蓝色)、Gray(灰色)、Lime(草绿色)、Fuchsia(紫红色)、White(白色)、Green(绿色)、Purple(紫色)、Silver(银灰色)、Yellow(黄色)、Aqua(海蓝色)。

注意: 一般推荐使用样式替代<body>标签的 text 属性和 bgcolor 属性。

3. <h1> 标题

一般文章都有标题、副标题、章和节等结构,HTML 中也提供了相应的标题标签<h_n>,其中 n 为标题的等级,n 越小,标题字号就越大。HTML 总共提供六个等级的标题,以下列出所有等级的标题:

```
<h1>...</h1>          第一级标题
```



<code><h2>...</h2></code>	第二级标题
<code><h3>...</h3></code>	第三级标题
<code><h4>...</h4></code>	第四级标题
<code><h5>...</h5></code>	第五级标题
<code><h6>...</h6></code>	第六级标题

【例 3-4】 hn.html

```
<html>
<head>
  <title>标题标签示例</title>
</head>
<body>
  <p>这是一个演示标题的网页</p>
  <h1>这是一级标题</h1>
  <h2>这是二级标题</h2>
  <h3>这是三级标题</h3>
  <h4>这是四级标题</h4>
  <h5>这是五级标题</h5>
  <h6>这是六级标题</h6>
</body>
</html>
```

例 3-4 的显示结果如图 3-1 所示。



图 3-1 例 3-4 的显示结果

4. `<p>` 标签和 `
` 标签

1) `<p>` 标签

为了使文档排列整齐、清晰,文字段落之间常使用`<p>`标签。`<p>`意味着段落的开

始, </p>意味着段落的结束。即使忘记了使用结束标签</p>, 大多数浏览器也会正确地显示文档内容, 但不要形成这种习惯。

<p>标签有一个属性 align, 它用来设置段落中文本的对齐方式。对齐方式一般有center(居中)、left(居左)、right(居右)三种。

【例 3-5】 p.html

```
<html>
  <head><title>p 标签示例</title></head>
  <body>
    <p align="right">这是第一段, 居右显示。</p>
    <p align="center">这是第二段, 居中显示。</p>
  </body>
</html>
```

2)
标签

标签的作用是换行, 它是一个单标签。

【例 3-6】 br.html

```
<html>
  <head><title>br 标签示例</title></head>
  <body>
    这是第一行<br>
    这是第二行
  </body>
</html>
```

3)
与<p>的区别

强迫跳到下一行。

<p>产生一行空白。

可强迫换行却不多跳一行, <p>则会换行而且多跳一行。

【例 3-7】 br_p.html

```
<html>
  <head>
    <title>br 标签与 p 标签的区别</title>
  </head>
  <body>
    人生无根蒂, 飘如陌上尘。以 p 标签结束<p>
    分散逐风转, 此已非常身。以 br 标签结束<br>
    落地为兄弟, 何必骨肉亲! 以 br 标签结束<br>
    得欢当作乐, 斗酒聚比邻。以 p 标签结束<p>
    盛年不重来, 一日难再晨。多个 p 标签并没有产生多个空行的效果<p><p><p><p><p>
    及时当勉励, 多个 br 标签可以产生多个空行的效果<br><br><br><br><br>
```

```

    岁月不待人。
</body>
</html>

```

例 3-7 的显示结果如图 3-2 所示。

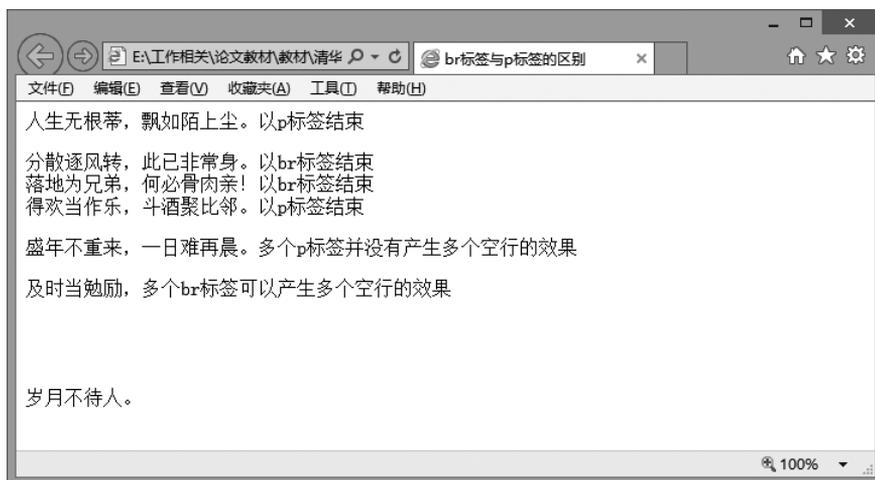


图 3-2 例 3-7 的显示结果

5. 标签

 标签用来设置文本的字体。通过指定它的 size 属性就能设置字号大小，而 size 属性的有效值范围为 1~7，其中默认值为 3。可以在 size 属性值之前加上“+”“-”字符，来指定相对于字号初始值的增量或减量。color 属性用于设置文字颜色，color 属性的值可以是一个十六进制数，也可以是颜色名称的英文单词。face 属性用于设置文本的字体，face 的属性值可以是本机上的任一字体类型。例如：

```
<font size="2" face="华文彩云" color="#008000">这一行用的是绿色、大小是 2 的华文彩云字体</font>
```

注意：一般推荐使用样式替代 标签。

6. <a> 标签

超文本中的链接是其最重要的特性之一，利用它可以从一个页面跳转到其他页面。一个链接的基本格式如下：

```
<a href="URL">文本</a>
```

标签<a>表示一个链接的开始，表示链接的结束；属性 href 定义了这个链接所指的地方；通过单击“文本”可以到达指定的文件。例如：

```
<a href="http://www.taobao.com">淘宝网</a>
```

浏览器读到这个标签时,会用蓝色显示文本“淘宝网”。单击这行文本,页面会跳转到 URL 为 `http://www.taobao.com` 的页面。

在所有浏览器中,链接的默认外观是:未被访问的链接带有下画线而且是蓝色的,已被访问的链接带有下画线而且是紫色的,活动链接带有下画线而且是红色的。

`<a>` 标签有一个 `target` 属性,用于规定在何处打开被链接文档。如果 `target` 属性的值为“`_blank`”,表示在新窗口中打开被链接文档;`target` 属性的默认值为“`_self`”,表示在相同的框架中打开被链接文档。

3.1.3 HTML 中的表格

表格是 HTML 中最重要的标志之一。表格就是由行和列构成的栅格,这些行和列构成一个个单元格,每个单元格可以放置文本或图形等。表格最基本的作用就是让复杂的数据变得更有条理,让人容易看懂。在网页中,还可以用表格来进行图形的拼接和页面元素的定位等巧妙的应用。

一个表格由 `<table>` 标签开始,以 `</table>` 标签结束,表格的内容由 `<tr>`、`<th>` 和 `<td>` 标签定义。`<tr>` 标签定义表格的行,表格有多少行就有多少个 `<tr>` 标签;`<td>` 标签定义表格的单元格,有多少个单元格就有多少个 `<td>` 标签;`<th>` 标签和 `<td>` 标签相似,它主要用来定义表格的表头,用它定义的单元格内容会居中加粗显示。

```
<table>...</table>定义表格
<tr>...</tr>定义表行
<th>...</th>定义表头
<td>...</td>定义单元格(表格的具体数据)
```

【例 3-8】 simpletable.html

```
<table border width="50%">
  <tr><th>第一讲</th><th>第二讲</th><th>第三讲</th></tr>
  <tr><td>HTML 概念</td><td>HTML 元素</td><td>HTML 属性</td></tr>
</table>
```

例 3-8 的显示结果如图 3-3 所示。

第一讲	第二讲	第三讲
HTML概念	HTML元素	HTML属性

图 3-3 例 3-8 的显示结果

1. 设置表格的大小

一般情况下,表格的总长度和总宽度是根据各行和各列的总和自动调整的。如果要直接固定表格的大小,可以使用下列方式:

```
<table width="w" height="h">
```

width 和 height 属性分别指定表格的固定宽度和高度,w 和 h 可以用像素来表示,也可以用百分比(与整个屏幕相比的大小比例)来表示。

这是一个长为 300 像素,高为 100 像素的表格。

```
<table border width="300" height="100">
```

这是一个宽为屏幕 50%,高为屏幕 10%的表格。

```
<table border width="50%" height="10%">
```

2. 设置表格的边框

边框是用 border 属性来体现的,它设置表格的边框线宽度。将 border 设成不同的值,有不同的效果。

【例 3-9】 border.html

```
<table border="5" width="50%">
  <tr><th>第一讲</th><th>第二讲</th><th>第三讲</th></tr>
  <tr><td>HTML 概念</td><td>HTML 元素</td><td>HTML 属性</td></tr>
</table>
```

例 3-9 的显示结果如图 3-4 所示。

【例 3-10】 border0.html

```
<table border="0" width="50%">
  <tr><th>第一讲</th><th>第二讲</th><th>第三讲</th></tr>
  <tr><td>HTML 概念</td><td>HTML 元素</td><td>HTML 属性</td></tr>
</table>
```

例 3-10 的显示结果如图 3-5 所示。

第一讲	第二讲	第三讲
HTML概念	HTML元素	HTML属性

图 3-4 例 3-9 的显示结果

第一讲	第二讲	第三讲
HTML概念	HTML元素	HTML属性

图 3-5 例 3-10 的显示结果

3. 跨多行列的单元格

要创建跨多行、多列的单元格,只需在<th>或<td>中加入 rowspan 或 colspan 属性。这两个属性的值,表明了单元格要跨越的行的个数或列的个数。

1) 跨多列的单元格

```
colspan=# (水平合并单元格)
```

【例 3-11】 colspan.html

```
<table border width="50%">
  <tr><th colspan="3">值班人员 </th>
  <tr><th>星期一</th><th>星期二</th> <th>星期三</th></tr>
  <tr><td>王强</td><td>张伟</td><td>赵平</td></tr>
</table>
```

例 3-11 的显示结果如图 3-6 所示。

2) 跨多行的单元格

```
rowspan=# (竖直合并单元格)
```

【例 3-12】 rowspan.html

```
<table border width="50%">
  <tr><th rowspan="2">值班人员</th><th>星期一</th><th>星期二</th><th>星期三</th></tr>
  <tr><td>李强</td><td>张明</td><td>王平</td></tr>
</table>
```

例 3-12 的显示结果如图 3-7 所示。

值班人员		
星期一	星期二	星期三
王强	张伟	赵平

图 3-6 例 3-11 的显示结果

值班人员	星期一	星期二	星期三
	李强	张明	王平

图 3-7 例 3-12 的显示结果

3.1.4 HTML 表单

HTML 表单用于收集不同类型的用户输入。用户在页面内填写好信息后,可以通过单击“提交”按钮将数据提交到服务器。表单是实现动态交互的基础。

常用的表单元素如表 3-1 所示。

表 3-1 表单元素列表

元素类型	元素描述	元素类型	元素描述
text	单行文本框	textarea	多行文本框
password	密码框	select	下拉列表
radio	单选按钮	submit	提交按钮
checkbox	复选框	reset	重置按钮

1. 单行文本框

【功能】 内容本身比较短,只有一行,需要用户的输入。

【示例】

```
<input type = "text" name = "username" size = "20">
```

网页中的表现形式如图 3-8 所示。

【说明】

- (1) size 属性设置文本框的显示长度,默认值为 20。
- (2) value 属性设置文本框的初始值。
- (3) 如果不希望文本框的内容被编辑,可以使用 readonly 属性。

2. 密码框

【功能】 输入的内容看不到,但是可以被发送到服务器。

【示例】

```
<input type="password" name="pw" size="20">
```

网页中的表现形式如图 3-9 所示。



图 3-8 单行文本框



图 3-9 密码框

3. 单选按钮

【功能】 给出多种选择,可以从中选择一项。

【示例】

```
<p> <input type = "radio" name = "career" value = "工人" checked >工人</p>
<p> <input type = "radio" name = "career " value = "农民">农民</p>
<p> <input type = "radio" name = "career " value = "军人">军人</p>
<p> <input type = "radio" name = "career " value = "学生">学生</p>
```

网页中的表现形式如图 3-10 所示。

【说明】

- (1) checked 属性表明该选项默认被选中。
- (2) 为实现选项之间的单选效果,需要每个选项的 name 属性保持一致。



图 3-10 单选按钮

4. 复选框

【功能】 给出多种选择,可以从中选择多项。

【示例】

```
<p>请选择你的爱好:</p>
<p><input type = "checkbox" name = "hobbies" value = "音乐">音乐
```

```
<input type = "checkbox" name = "hobbies" value = "旅游">旅游
<input type = "checkbox" name = "hobbies" value = "读书">读书</p>
```

网页中的表现形式如图 3-11 所示。

【说明】

为实现在服务器端一次性读取所有选中的选项,每个选项的 name 属性需要保持一致。

5. 多行文本框

【功能】 能够进行多行文本输入。

【示例】

```
<p>请输入你的留言:</p>
<p><textarea name="message" rows="5" cols="30"></textarea></p>
```

网页中的表现形式如图 3-12 所示。

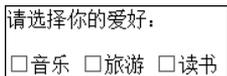


图 3-11 复选框

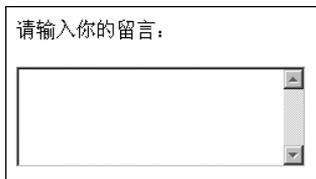


图 3-12 多行文本框

【说明】

- (1) rows 属性表明文本框的行数,cols 属性表明文本框的列数。
- (2) 文本框的初始内容需要写在标签体中,而不是使用 value 属性设定。

6. 下拉列表

【功能】 给出多种选择,可以从中选择一项,也可以支持选择多项。

【示例】

```
<p><select size="1" name="mysite">
<option value="163" selected>网易</option>
<option value="sohu">搜狐</option>
<option value="sina">新浪</option>
</select></p>
```

网页中的表现形式如图 3-13 所示。



图 3-13 下拉列表

【说明】

- (1) 为实现下拉列表的多项选择,可以为 < select > 标签设置 multiple 属性。
- (2) 支持多项选择时,size 属性的值需要设置为大于 1。
- (3) <option> 标签的 selected 属性表明该选项默认被选中。

7. “提交”按钮

【功能】 完成表单的提交。

【示例】

```
<input type="submit" value="提交" name="b1">
```

网页中的表现形式如图 3-14 所示。

【说明】

(1) type="submit"是“提交”按钮的标识;value是“提交”按钮的值,同时也是按钮上面显示的内容;name是“提交”按钮的名字,可以根据这个名字获取“提交”按钮的值。

(2) 每个表单都应该至少有一个“提交”按钮用来完成提交动作,也可以使用多个“提交”按钮来完成不同的提交动作。

(3) 多个“提交”按钮可以使用相同的名字,但是值不一样;也可使用不同的名字,根据名字区分不同的提交按钮。



图 3-14 “提交”按钮

8. “重置”按钮

【功能】 把表单元素的信息恢复到原始状态。

【示例】

```
<input type="reset" value="重置" name="b2">
```

网页中的表现形式如图 3-15 所示。

【说明】

(1) “重置”按钮的作用是恢复页面信息。

(2) 并不是所有页面都需要“重置”按钮。



图 3-15 “重置”按钮

9. 表单

【功能】 一个包含表单元素的区域。

【示例】

```
<form method = "post" action = "deal.jsp">  
</form >
```

【说明】

(1) method="post",表明表单的提交方式是 post。一般而言,表单的提交方式有两种: get 和 post,默认是以 get 方式提交。

(2) action="deal.jsp",表明表单元素的信息是提交给服务器上的 deal.jsp 文件来处理。

(3) 表单所提交的信息是位于<form>开始标志和</form>结束标志之内的表单元素的信息,所有的表单元素都应该在<form>和</form>之内。

(4) 一个页面可以有多个表单,表单之间不可以嵌套或者重叠。

3.1.5 项目 1: 用户注册页面的开发

1. 项目构思

使用表格和表单标签开发一个 HTML 页面,实现用户注册功能。注册信息包括姓名、密码、性别、职业、电话号码、电子邮箱、兴趣爱好和自我简介。

2. 项目设计

注册信息所使用的表单元素如表 3-2 所示。

表 3-2 注册信息及表单元素类型对应表

注册信息	表单元素类型	注册信息	表单元素类型
姓名	单行文本框	电子邮箱	单行文本框
密码	密码框	兴趣爱好	复选框
性别	“单选”按钮	自我简介	多行文本框
职业	下拉列表框	注册按钮	“提交”按钮
电话号码	单行文本框		

3. 项目实施

```

文件名:reg.html
<html>
<head>
<title>用户注册页面</title>
</head>
<body>
<h2 align="center">用户注册</h2>
<form method="post">
<table align="center" border="1" width="35%">
  <tr><th width="30%">姓名</th><td><input type="text" name="name"></td></tr>
  <tr><th>密码</th><td><input type="password" name="pw"></td></tr>
  <tr><th>确认密码</th><td><input type="password" name="pw2"></td></tr>
  <tr><th>性别</th><td><input type="radio" name="gender" value="男" checked>男
    <input type="radio" name="gender" value="女">女</td></tr>
  <tr><th>职业</th>
    <td><select size="1" name="career">
      <option value="教育工作者">教育工作者</option>
      <option value="公司职员" selected>公司职员</option>
      <option value="自由职业者">自由职业者</option>
      <option value="其他">其他</option>
    </td></tr>
  </table>
  <input type="submit" value="提交">
</form>

```



```

        </select></td></tr>
        <tr><th>电话号码</th><td><input type="text" name="telephone"></td></tr>
        <tr><th>电子邮箱</th><td><input type="text" name="email" size="30"></td>
    </tr>
    <tr><th>兴趣爱好</th><td>
        <input type="checkbox" name="fav" value="体育">体育
        <input type="checkbox" name="fav" value="看书">看书
        <input type="checkbox" name="fav" value="旅游">旅游
        <input type="checkbox" name="fav" value="美食">美食
    </td></tr>
    <tr><th>自我介绍</th>
    <td><textarea name="intro" rows="5" cols="30"></textarea></td></tr>
    <tr><td colspan="2" align="center"><input type="submit" value="注册"></td>
</tr>
</table>
</form>
</body>
</html>

```

4. 项目运行

reg.html 的显示结果如图 3-16 所示。

用户注册

姓名	<input type="text"/>
密码	<input type="password"/>
确认密码	<input type="password"/>
性别	<input checked="" type="radio"/> 男 <input type="radio"/> 女
职业	公司职员 ▼
电话号码	<input type="text"/>
电子邮箱	<input type="text"/>
兴趣爱好	<input type="checkbox"/> 体育 <input type="checkbox"/> 看书 <input type="checkbox"/> 旅游 <input type="checkbox"/> 美食
自我介绍	<input type="text"/>
<input type="submit" value="注册"/>	

图 3-16 reg.html 的显示结果

3.1.6 项目 2: 图书管理系统的静态页面

1. 项目构思

开发一个简单的图书管理系统,实现对图书信息的浏览、添加、修改和删除。本项目负责实现图书管理系统的静态页面,其他功能在后续章节依次实现。用户进入 index.html 页

面获取所有图书的信息。在这个页面中有添加、修改和删除图书信息的链接,其中“添加图书信息”链接到 add.html,“修改”链接到 edit.html,删除功能不需要链接到页面。

2. 项目设计

index.html 使用表格列出图书的书名、作者、出版社和价格信息,表格上方有一个添加图书信息的链接。

add.html 显示表单元素,提供书名、作者、出版社和价格信息的输入文本框和“提交”按钮。

edit.html 使用表单元素显示选中图书的书名、作者、出版社和价格信息,同时提供修改按钮。

3. 项目实施

```

文件名:index.html
<html>
<head><title>图书管理系统</title></head>
<body>
<h2 align="center">图书管理系统</h2>
<p align="center"><a href="add.html">添加图书信息</a><p>
<table align="center" width="50%" border="1">
  <tr><th>书名</th><th>作者</th><th>出版社</th><th>价格</th><th>管理</th>
</tr>
  <tr><td>XML 详解</td><td>王红丽</td><td>吉林大学出版社</td><td>34</td>
  <td><a href='edit.html'>修改</a>&nbsp;<a href='#'>删除</a></td></tr>
  <tr><td>JSP 技术大全</td><td>张勇</td><td>清华大学出版社</td><td>45</td>
  <td><a href='edit.html'>修改</a>&nbsp;<a href='#'>删除</a></td></tr>
  <tr><td>Java 编程快速入门</td><td>赵坤</td><td>东软电子出版社</td><td>39
</td>
  <td><a href='edit.html'>修改</a>&nbsp;<a href='#'>删除</a></td></tr>
</table>
</body>
</html>

```

```

文件名:add.html
<html>
<head>
<title>添加图书信息</title>
</head>
<body>
<h2 align="center">添加图书信息</h2>
<form>
<table align="center" width="30%" border="1">
  <tr><th width="30%">书名:</th>
  <td><input type="text" name="bookname"></td></tr>
  <tr><th>作者:</th>

```

```
        <td><input type="text" name="author"></td></tr>
    <tr><th>出版社:</th>
        <td><input type="text" name="press"></td></tr>
    <tr><th>价格:</th>
        <td><input type="text" name="price"></td></tr>
    <tr><th colspan="2">
        <input type="submit" value="添加">
        <input type="reset" value="重置"></th></tr>
</table>
</form>
</body>
</html>
```

文件名:edit.html

```
<html>
<head>
<title>修改图书信息</title>
</head>
<body>
<h2 align="center">修改图书信息</h2>
<form>
<table align="center" width="30%" border="1">
    <tr><th width="30%">书名:</th>
        <td><input type="text" name="bookname" value="XML 详解"></td></tr>
    <tr><th>作者:</th>
        <td><input type="text" name="author" value="王红丽"></td></tr>
    <tr><th>出版社:</th>
        <td><input type="text" name="press" value="吉林大学出版社"></td></tr>
    <tr><th>价格:</th>
        <td><input type="text" name="price" value="34"></td></tr>
    <tr><th colspan="2">
        <input type="submit" value="修改">
        <input type="reset" value="重置"></th></tr>
</table>
</form>
</body>
</html>
```

4. 项目运行

index.html 的显示结果如图 3-17 所示。

add.html 的显示结果如图 3-18 所示。

edit.html 的显示结果如图 3-19 所示。

图书管理系统

添加图书信息

书名	作者	出版社	价格	管理
XML详解	王红丽	吉林大学出版社	34	修改 删除
JSP技术大全	张勇	清华大学出版社	45	修改 删除
Java编程快速入门	赵坤	东软电子出版社	39	修改 删除

图 3-17 index.html 的显示结果

添加图书信息

书名:	<input type="text"/>
作者:	<input type="text"/>
出版社:	<input type="text"/>
价格:	<input type="text"/>
<input type="button" value="添加"/> <input type="button" value="重置"/>	

图 3-18 add.html 的显示结果

修改图书信息

书名:	XML详解
作者:	王红丽
出版社:	吉林大学出版社
价格:	34
<input type="button" value="修改"/> <input type="button" value="重置"/>	

图 3-19 edit.html 的显示结果

3.2 JavaScript

3.2.1 JavaScript 简介

JavaScript 是一种动态的、弱类型的、基于对象和事件驱动脚本语言。它的解释器被称为 JavaScript 引擎,是浏览器的一部分。JavaScript 广泛用于 Web 应用开发,常用来为网页添加各式各样的动态功能,为用户提供更流畅美观的浏览效果。通常情况下,JavaScript 脚本是通过嵌入在 HTML 中来实现自身功能的。

3.2.2 JavaScript 中的事件

事件是 JavaScript 应用的“心脏”,当客户端与浏览器中的 Web 页面进行某些类型的交互时,事件就发生了。事件是指某个对象发出的消息,标志着某个特定行为的发生或某个特定的条件成立。事件可能是由用户在某些内容上的单击、鼠标经过某个特定元素或按下键盘上的某些按键触发的。事件还可能是 Web 浏览器中发生的事情,例如某个 Web 页面加



载完成或者用户改变窗口大小等。JavaScript 的常用事件主要有如下几个。

(1) 单击事件 onClick。当元素被鼠标点击时,产生 onClick 事件。Web 页面元素的 onClick 属性指定事件的处理代码。

(2) 改变事件 onChange。当 text 或 textarea 元素输入的内容改变时引发该事件,当 select 下拉列表选项改变时也会引发该事件。

(3) 选中事件 onSelect。当 text 或 textarea 元素中的内容被加亮后,引发该事件。

(4) 聚焦事件 onFocus。当用户单击 text、textarea 和 select 元素时,引发该事件,此时元素获得焦点。

对事件进行处理的代码称为事件处理程序。JavaScript 中的事件处理程序通常由函数实现,其基本格式如下:

```
function 函数名(参数 1,参数 2,...)
{
    函数执行部分;
    函数返回部分;
}
```

3.2.3 JavaScript 中的对象

JavaScript 语言是基于对象的(object-based),而不是面向对象的(object-oriented)。JavaScript 没有提供抽象、继承、重载等有关面向对象的许多功能,而是把其他语句所创建的复杂对象统一起来,形成一个强大的对象系统。但是,JavaScript 语言仍然具有面向对象的基本特征,可以根据需要创建自己的对象,扩大 JavaScript 的应用范围。

1. JavaScript 支持的对象

(1) 浏览器环境中提供的对象,反映当前加载的 Web 页面及其内容以及浏览器当前会话的信息,如常用的 window、document、history、location 对象等。

(2) JavaScript 的内置对象,是若干与当前窗口或加载的文件无关的对象,如 String、Date、Math 等。

(3) 用户自己定义的对象。

2. String 对象

在 JavaScript 中,一个字符串就是一个 String 对象。String 对象包含描述字符串的属性和各种处理字符串的方法,例如字符串的长度、搜索字符串、提取子串等。

1) String 对象的创建

创建一个字符串对象有两种方式:

```
str1="hello";
str2=new String("hello");
```

2) String 对象的属性

String 对象有一个属性 length,表示字符串的长度。例如:



```
myStr="Hello, JavaScript World";  
length=myStr.length;
```

3) String 对象的常用方法

- (1) `charAt(pos)`: 返回指定位置 `pos` 上的字符。
- (2) `indexOf(subStr)`: 返回某个指定的字符串 `subStr` 在字符串中首次出现的位置。
- (3) `substring(start, end)`: 返回介于下标 `start` 和 `end` 之间的子串。

3. document 对象

`document` 对象反映 HTML 文档的特性,它包含了与文档元素(`elements`)一起工作的对象,将这些元素封装起来供编程人员使用。

1) document 对象的常用属性

- (1) `forms`: `form` 对象数组,每一个元素对应于文档中的一个 HTML 表单标签。
- (2) `links`: `link` 对象数组,每一个元素对应于文档中的一个 HTML 超级链接标签。

2) document 对象的常用方法

- (1) `document.write()`: 用于将文本信息直接输出到浏览器窗口。
- (2) `document.getElementById()`: 通过 HTML 标签的 `id` 属性获得一个 HTML 元素对象。
- (3) `document.getElementsByName()`: 通过 HTML 标签的 `name` 属性获得一些 HTML 元素对象,返回的是具有相同 `name` 属性的 HTML 元素对象数组。

4. window 对象

`window` 对象就是窗口对象,处于文档对象模型的最顶层,代表当前浏览器窗口,是 `document`、`location` 及 `history` 对象的父对象。

1) window 对象的常用属性

- (1) `closed`: 判断窗口是否关闭,返回布尔值。
- (2) `document`: 见上文第 3 点的介绍。
- (3) `history`: 主要用于记录浏览器的访问历史,提供浏览网页的前进与后退功能。
- (4) `location`: 用于获取当前浏览器中 URL 地址栏内的相关数据。

2) window 对象的常用方法

- (1) `alert(message)`: 显示带有警告信息 `message` 的窗口,并有“确定”按钮。
- (2) `confirm(message)`: 显示带有确认信息 `message` 的窗口,有“确定”和“取消”按钮。
- (3) `prompt(message, defaultValue)`: 显示提示对话框,带有提示消息 `message` 和默认值 `defaultValue`。
- (4) `focus()`: 使本窗口获得焦点。
- (5) `open(url, [name], [feature])`: 打开一个新窗口,显示 URL 指定的页面。`name` 指定窗口的名称,`feature` 指定窗口的特性或外观,包括窗口的高度、宽度,是否有菜单条和滚动条,窗口大小是否可改变等。
- (6) `close()`: 关闭当前窗口。



3.2.4 将 JavaScript 代码加入 HTML 文件中

使用<script>…</script>标签将 JavaScript 代码加入到 HTML 中。格式如下：

```
<script language="JavaScript">
Javascript 代码;…
</script>
```

其中,language 属性指定脚本语言的类型。

在实际开发过程中,由于很多页面可能包含相同的 JavaScript 代码,为了提高代码的可重用性,可以将一些常用功能代码写在一个单独的 JavaScript 源文件中(扩展名为.js),在页面中使用<script>…</script>标签引入该文件,格式如下：

```
<script type="text/javascript" src="url"></script>
```

其中,src 属性指定引入的 JavaScript 源文件的 URL 地址。

3.2.5 项目 3: 使用 JavaScript 进行用户注册信息的客户端验证

1. 项目构思

为本章项目 1 的用户注册信息提供客户端验证,验证姓名和密码是否为空,密码长度是是否符合要求,输入的确认密码与密码是否一致,电话号码和电子邮箱是否为空及格式是否正确。

2. 项目设计

- (1) 为 form 标签添加属性 name="form1" onSubmit="return check()"。
- (2) 定义 JavaScript 函数 check()验证表单的信息。

3. 项目实施

```
文件名:reg_js.html(省略的内容同 reg.html)
<html>
<head>
<title>加入验证的用户注册页面</title>
<script language="javascript">
function check() {
    if(form1.name.value=="") {
        alert("姓名不能为空!");
        form1.name.focus();
        return false;
    }
    if(form1.pw.value=="") {
        alert("密码不能为空!");
        form1.pw.focus();
    }
}
```