第 3 课
ASP.NET 的服务器控件

ASP.NET 向页面提供了一系列相关的服务器控件包括标准控件、数据绑定控件和验证控件等，开发人员可以通过这些控件完成对页面的设计操作，本课将主要介绍 ASP.NET 中的标准服务器控件。

标准服务器控件包括 Label 控件、Literal 控件、TextBox 控件、TextBox 控件、DropDownList 控件、Panel 控件、Image 控件以及 Calendar 控件等。通过对本课的学习，读者可以了解常用的服务器控件分类，也可以了解这些控件的常用属性和事件，还可以通过这些控件熟练地构建 ASP.NET 页面。

本课学习目标：

- ❑ 熟悉控件的分类和 HTML 服务器控件
- ❑ 掌握文本控件的基本使用
- ❑ 掌握按钮控件的基本使用
- ❑ 掌握与选择类控件相关的属性和事件
- ❑ 掌握与列表类控件相关的属性和事件
- ❑ 熟悉容器控件 MultiView 和 Panel
- ❑ 掌握如何使用 Image 控件和 Calendar 控件
- ❑ 了解 ImageMap 控件的基本使用
- ❑ 掌握如何使用 Web 服务器控件搭建基本的 Web 窗体页面

3.1 控件概述

ASP.NET 服务器控件是一种服务器端的组件，它封装了用户界面及其相关的功能。ASP.NET 服务器控件直接或间接的从 System.Web.UI.Control 类派生，下面将简单介绍控件的基本知识。

3.1.1 控件分类

在很多情况下，用户无法将可执行代码与 HTML 本身进行分离，使页面难以阅读，也难以维护，ASP.NET 服务器控件很好地解决了这个问题。通常情况下，服务器控件是可以被服务器理解的标签。

常用的服务器控件有三种类型。

（1）HTML 服务器控件

HTML 服务器控件是一种最传统的 HTML 标签，它提供了对标准 HTML 元素类的封装，在 HTML 控件中添加了一个在服务器端运行的属性，即可以由通用的客户端 HTML 控件转变为服务器端的 HTML 控件，使开发人员可以对其进行编程。

（2）Web 服务器控件

Web 服务器控件是一种新型的 ASP.NET 标签控件，比 HTML 服务器控件具有更多的功能。Web 服务器控件不仅包括窗体控件（即标准控件，如按钮和文本框），也包括特殊用途的控件（如日历、菜单和树视图控件）。

（3）Validation 服务器控件

Validation 服务器控件方便用户的输入验证，例如 RequiredFieldValidator 控件和 CompareValidator 控件等。

提示

本书主要介绍 Web 服务器控件和 Validation 控件，在本课将详细介绍 Web 服务器控件，后面将会介绍 Validation 控件。

从某种意义上来讲，Validation 服务器控件也属于 Web 服务器控件。因此，下面从三个方面来说明 HTML 服务器控件与 Web 服务器控件的区别。

（1）是否映射到 HTML 标签

HTML 服务器控件与 HTML 中的元素存在一一对应的映射关系，runat=server 属性把传统的 HTML 标签转换为服务器控件，使得开发人员可以将 ASP 页面移植到 ASP.NET 平台。而 Web 服务器控件不直接映射到 HTML 中的元素，使得开发人员可以使用第三方控件。

（2）对象模型

HTML 服务器控件使用 HTML 中心对象模型，在该模型中，控件包括一个关键字/值对的属性集合。而 Web 服务器控件使用基于组件的对象模型，该模型要求使用一致的对象类型。

（3）是否能自适应输出

HTML 服务器控件不能根据浏览器的不同，调整所输出 HTML 文档的显示效果。而 Web 服务器控件可以自动根据浏览器的不同，调整输出 HTML 文档的显示效果。

3.1.2 HTML 服务器控件

HTML 服务器控件一般被服务器理解为 HTML 标签，在 ASP.NET 中，HTML 元素是作为文本来进行处理的。传统的 HTML 元素不能被 ASP.NET 服务器端直接使用，但是通过将这些 HTML 元

素的功能进行服务器端的封装，开发人员就可以在服务器端使用这些 HTML 元素。

如果要将 HTML 元素标记为 HTML 服务器控件，则需要在 HTML 元素代码中添加 runat=server 属性。runat 属性指示 HTML 元素是一个服务器控件，它需要添加 id 属性来标识该服务器控件，id 属性引用可用于操作运行时的服务器控件。

例如，将普通的用于用户输入的 input 元素设置为服务器控件。

```
<input id="ipName" name="ipName" runat="server" value="陈想芳" />
```

注意

所有的 HTML 服务器控件必须放在带有 runat=server 属性的<form>标签内，将 HTML 元素添加该属性后会指示该表单应该在服务器进行处理，该属性同时指示其包括在内的控件可以被服务器脚本访问。

HTML 服务器控件有多个优点，如下所示。

- HTML 服务器控件映射一对一的，与它们对应的 HTML 标记。
- 当编译 ASP.NET 应用程序时，HTML 服务器控件与 runat=server 属性被编译成程序集。
- 大多数控件包含最常用的控件事件 OnServerEvent。
- 未实现作为特定 HTML 服务器控件的 HTML 元素仍然可以用于服务器端；但是作为 HtmlGenericControl 的程序集添加。
- 当 ASP.NET 页重新发布时，HTML 服务器控件保留它们的值。

ASP.NET 要求所有的 HTML 元素必须正确关闭和嵌套。如表 3-1 所示描述了一些常用的 HTML 服务器控件。

表 3-1 常用的 HTML 服务器控件

HTML 服务器控件	说　明
HtmlAnchor	控制 HTML 中的<a>元素
HtmlForm	控制 HTML 中的<form>元素
HtmlButton	控制 HTML 中的<button>元素
HtmlImage	控制 HTML 中的<image>元素
HtmlSelect	控制 HTML 中的<select>元素
HtmlTable	控制 HTML 中的<table>元素
HtmlTableCell	控制 HTML 中的<td>和<th>元素
HtmlTableRow	控制 HTML 中的<tr>元素
HtmlTextArea	控件 HTML 中的<textarea>元素

HTML 服务器控件具有相同的 HTML 输出和作为其对应的 HTML 标记相同的属性，也提供了自动状态管理和服务器端事件。如 InnerHtml 属性获取或设置控件的开始标记和结束标记之间的内容，但不自动将特殊字符转换为等效的 HTML 实体；InnerText 获取或设置控件的开始标记和结束标记之间的内容，并自动将特殊字符转换为等效的 HTML 实体；onclick 当鼠标单击控件时触发该事件，如按钮的单击；onchange 在内容改变时被触发，如文本框内容发生变化时触发该事件。

3.1.3 ASP.NET 服务器控件

ASP.NET 服务器控件即 Web 服务器控件，由于 ASP.NET 服务器控件的对象模型不一定能够反映 HTML 语法，因此比 HTML 服务器控件更加抽象。

ASP.NET 服务器控件可以自动检测客户端浏览器的类型，产生一个或者多个适当的 HTML 控件，并且自动调整为适合浏览器的输出。该类型的控件支持数据绑定技术，可以和数据源进行连接，用来显示或修改数据源数据，其优点如下所示。

- 使制造商和开发人员能够生成容易的工具或者自动生成用户的应用程序接口。
- 简化创建交互式 Web 窗体的过程。

1．语法形式

每一个 Web 服务器控件都有一个<asp:>的前缀，该前缀表示此控件为 Web 服务器控件，其语法格式如下。

```
<asp:Control id="name" runat="server" />
```

在上述语法中 id 表示控件的惟一标识，而 runat 属性则指示该控件为服务器控件，这两个属性的含义与 HTML 控件中相应的属性含义一致。例如下面代码表示在 ASP.NET 页面中添加 Button 控件。

```
<form id="form1" runat="server">
    <asp:Button ID="Button1" runat="server" Text="Button" /></div>
</form>
```

2．常用的服务器控件

ASP.NET 中提供了许多常用的服务器控件，如 Label 控件、TextBox 控件、Button 控件、DropDownList 控件、Panel 控件、Image 控件以及 Calendar 控件等。

3．服务器控件的常用属性

ASP.NET 中的服务器控件继承自 System.Web.UI.WebControls 类，该类提供了大多数 Web 服务器控件的公共属性、方法和事件。如表 3-2 对服务器控件的最常用的一些属性进行了说明。

表 3-2　服务器控件的常用属性

属 性 名 称	说　　明
AccessKey	获取或设置快速导航到控件的快捷键，可以指定这个属性的内容为数字或者是英文字母
BackColor	设置对象的背景颜色，其属性的值可以是颜色名称，也可以是#RRGGBB 格式
Enabled	获取或设置一个值，该值指示是否启用 Web 服务器控件
Visible	指定控件是否可见
ToolTip	小提示。在设置该属性时，当使用者停留在 Web 控件上时就会出现提示的文字
ID	所有控件的惟一标识列
FailureText	获取或设置当前登录尝试失败时显示的文本

设置控件的属性有两种方法，分别为在设计时通过【属性】窗口设置控件的属性和在运行时以编程方式动态设计控件的属性。

4．服务器控件的事件

事件是一种在满足某种条件（如鼠标单击）后开始运行的一种程序，大部分的 ASP.NET 控件都可以引发服务器端事件完成某些功能，页面事件是在页面加载和撤销时所引发的事件。

页面级别的事件主要有三种：Page_Load、Page_Init 和 Page_Unload。Page_Init 事件和 Page_Load 事件都是在页面加载时引发并用来执行初始化程序的事件，前者只是在页面第一次加载时执行的事件，而 Page_Load 事件在每次加载都执行。Page_Unload 事件执行最后的清理工作，例如关闭打开的文件和数据库连接等。

提示

除了以上的三种页面事件外，服务器控件还有其他许多公共的事件（如 click 事件），有兴趣的读者可以查找相关资料进行总结。

向 ASP.NET 服务器控件添加客户端事件有三种方法，如下所示。

（1）以声明方式向 ASP.NET 服务器控件添加客户端事件处理程序，即在资源视图中直接为控件添加事件属性，如 onmouseover 或 onclick 等。添加事件属性时需要针对不同的属性添加要执行的客户端脚本。

（2）以编程方式向 ASP.NET 控件添加客户端事件处理程序，即在页面的 Init 或 Load 事件中调用控件的 Attributes 集合的 Add()方法来动态添加客户端事件处理程序。

（3）向按钮控件添加客户端 onclick 事件。例如在按钮控件（Button、LinkButton 和 ImageButton）中要添加客户端 onclick 事件，可以在设计视图中将按钮控件的 onclientclick 属性设置为要执行的客户端脚本，也可以在源视图中直接添加该属性。

5．向 Web 窗体中添加服务器控件

将服务器控件添加到 Web 窗体中非常简单，最常用的方法有三种，如下所示。

（1）从【工具箱】中拖动控件到窗体或直接双击控件进行添加。

（2）在资源视图中，直接添加控件的声明代码。

（3）以编程方式动态创建 Web 服务器控件。

注意

所有的 ASP.NET 控件必须定义在.aspx（ASP.NET 页面文件）文件中，如果采用代隐藏技术设计程序，其事件程序一般定义在代码文件（如.cs 或.vb）中。

3.2 文本控件

文本控件是 Web 窗体中经常使用的控件之一，该类型的控件包含多个子控件，如 Label 控件、Literal 控件、HyperLink 控件和 TextBox 控件。

3.2.1 Label 控件

Label 控件是一种最基本的控件，它提供了一种以编程方式设置 Web 窗体页面中文本的方法。Label 控件中的文本是静态的，用户无法在该控件中进行编辑，它经常在页面固定位置显示文本时使用。

Label 控件通常在列表 Web 服务器控件（如 Repeater、DataList 和 GradList 等）中使用，用来显示数据库中的只读信息，还可以将 Label 控件绑定到数据源。声明 Label 控件有两种语法形式，如下所示。

```
<asp:Label ID= "lblName" runat = "server" Text = "文本内容"></asp:Label>
```

或者

```
<asp:Label ID= "lblName" runat = "server">文本内容</asp:Label>
```

从上述语法中可以看出，ID 和 Text 属性是 Label 控件最常用的属性，其中 ID 用来设置控件的惟一标识列；Text 属性向用户显示文本信息，该属性的值可以是 HTML 格式的内容。

Label 控件的 Text 属性可以设置为任何字符串（包括包含标记的字符串），如果字符串包含标记，Label 控件将解释标记，如下代码所示。

```
<form id="form1" runat="server">
    <asp:Label ID="userName" runat="server" Text="<b><font color='red'>名称:
```

```
        </font></b>" />
    </form>
```

运行上述代码时 Label 控件将以红色粗体呈现文本内容的名称。

注意

一般情况下，为了避免安全性问题（例如脚本注入的可能性），开发人员最好不要将 Text 属性的值设置为不受信任源的标记字符串。如果对 Text 属性的字符串不信任，则应该对该字符串进行编码。

【练习 1】

Label 控件可以用于另一个 Web 服务器控件之前的活动标题，虽然该控件无法接收用户输入的焦点，但是可以将该控件与另一个控件相关联，然后用户可以同时按下 Alt 键和 Label 控件定义的访问键导航到相关联的控件。以下通过步骤演示如何将 Label 控件与 Button 控件结合实现将 Label 控件作为标题的功能，主要步骤如下。

（1）添加新的 Web 窗体页，然后从【工具箱】中分别拖动 Label 控件和 Button 控件到窗体页中。

（2）将 Label 控件的 AssociatedControlID 属性设置为要以 Label 控件作为标题控件的 ID。

（3）将 Label 控件的 AccessKey 属性设置为要定义为访问键的单个字母或数字。

（4）设置 Label 控件的 Text 属性，其属性值显示指示访问键的带有下划线的字符。

（5）设置 Button 控件的 ID 属性和 Text 属性，结合前几个步骤，完成以后的具体代码如下。

```
<form id="form1" runat="server">
    <asp:Label AccessKey="T" AssociatedControlID="btnTest" ID="Label1" runat=
    "server"
        Text="<u>T</u>est:">
    </asp:Label> 
    <asp:Button ID="btnTest" runat="server" Text="Names" />
</form>
```

（6）运行页面进行测试，页面呈现时用户可以通过按 Alt+N 键实现单击按钮的提交功能。

3.2.2 Literal 控件

对于静态内容不需要使用容器，可以将标记作为 HTML 直接添加到页面中。如果要动态添加内容，则必须将内容添加到容器中。这时可以使用 Label 控件、Literal 控件、Panel 控件或 PlaceHolder 控件。

Literal 控件表示用于向页面添加内容的几个选项之一，它与 Label 控件最大的区别在于：Literal 控件不向文本中添加任何 HTML 元素（Label 控件呈现一个 span 元素）。因此，Literal 控件不支持包括位置特性在内的任何样式特性。但是该控件允许指定是否对内容进行编码。

通常情况下，如果开发人员希望文本和控件直接呈现在页面中而不使用任何附加标记时可以使用 Literal 控件。除了 ID 和 Text 属性外，Literal 控件最常用的属性是 Mode，该属性用于指定控件对所添加标记的处理方式，其值如下所示。

- **Transform** 将对添加到控件中的任何标记进行转换，以适应请求浏览器的协议。如果向使用 HTML 外的其他协议的移动设备呈现内容，设置该值时非常有用。该值为 Mode 属性的默认值。
- **PassThrough** 添加到控件中的任何标记都将按原样呈现在浏览器中。
- **Encode** 使用 HtmlEncode()方法对添加到控件中的任何标记进行编码，会将 HTML 编码转

换为其文本表示形式。例如<b>将呈现为<b>。当开始人员希望浏览器显示而不解释标记时，该方式将非常有用。

如下代码演示了如何在页面加载时通过动态编码的形式添加 Literal 控件。

```
protected void Page_Load(object sender, EventArgs e)
{
    Literal liter = new Literal();                    //通过 Literal 创建 liter 对象
    liter.ID = "literText";                           //设置 ID 属性
    liter.Text = "我想要添加的内容有很多";              //设置 Text 属性
    form1.Controls.Add(liter);                        //将 lister 添加到表单中
}
```

3.2.3 HyperLink 控件

HyperLink 控件用于创建文本或图片的超链接。声明该控件的语法格式如下。

```
<asp:HyperLink ID="hlName" runat="server"></asp:HyperLink>
```

HyperLink 控件有两个优点，如下所示。

- 可以在服务器代码中设置链接属性，例如开发人员可以根据页面中的条件来动态更改链接文本或目标页。
- 可以使用数据绑定来指定链接的目标 URL（以及必要时与链接一起传递的参数）。

除了 ID 属性和 Text 属性外，HyperLink 控件中包含多个属性。如表 3-3 所示对其他常用的属性进行了说明。

表 3-3 HyperLink 控件的常用属性

属 性 名 称	说　明
ImageUrl	获取或设置该控件显示的图像的路径
NavigateUrl	获取或设置单击控件时链接到的 URL
Target	获取或设置单击控件时显示链接到的网页内容的目标窗口或框架。该属性的值包括 _blank、_self、_top、_parent 和_search

与大多数 Web 服务器控件不同，当用户单击 HyperLink 控件时并不会在服务器代码中引发事件，该控件只执行导航操作。

【练习 2】

本次练习演示了 HyperLink 控件的基本使用，当用户单击页面中的文本内容时打开 Default.aspx 页面，主要步骤如下所示。

（1）添加新的 Web 窗体页，然后在页面中添加 HyperLink 控件，接着设置 Text 属性的值为 click me。

（2）通过【属性】窗口分别设置 NavigateUrl 和 Target 属性的值。

（3）运行窗体，单击呈现的文本内容进行测试，最终效果不再显示。

试一试

如果同时设置了 LinkButton 控件的 ImageUrl 和 Text 属性的值，则 ImageUrl 属性优先，读者可以更改练习 2 中的代码，更改完成后重新查看窗体效果。

3.2.4 TextBox 控件

无论是 Label 控件，还是 Literal 或 HyperLink 控件，它们都通过 Text 属性来显示文本，如果

用户需要根据自身的需要进行输入时，这些控件就不能满足需要，在 ASP.NET 中提供了一种新的控件为 TextBox 控件。

1．TextBox 控件的常用属性

TextBox 控件是一种基本控件，它为用户提供了一种在 Web 窗体中输入信息（包括文本、数字和日期等）的方法。该控件包含多个属性，如 Text 属性、ID 属性和 ReadOnly 属性等，如表 3-4 所示对常用的属性进行了说明。

表 3-4　TextBox 控件的常用属性

属 性 名 称	说　　明
AutoPostBack	获取或设置当 TextBox 控件上的内容发生改变时，是否自动将窗体数据回传到服务器，默认为 false。该属性通常和 TextChanged 事件配合使用
MaxLength	获取或设置文本框中最多允许的字符数。当 TextMode 属性设为 MultiLine 时，此属性不可用
ReadOnly	获取或设置 TextBox 控件是否为只读。默认值为 false
TextMode	获取或设置文本框的行为模式。默认值 SingleLine
Wrap	布尔值，指定文本是否换行。默认值为 true

TextBox 控件可以通过 TextMode 属性设置单行、多行和密码三种形式的文本框。TextMode 属性的值如下所示。

- **SingleLine**　默认值，单行输入模式。用户只能在一行中输入信息，还可以限制控件接受的字符数。
- **Password**　密码框，用户输入的内容将以其他字符代替（如“*”和“●”等），以隐藏真实信息。
- **Multiline**　多行输入模式，用户在显示多行并允许换行的文本框中键入信息。

开发人员可以通过 Text 属性获取或设置 TextBox 控件中的值，例如获取控件 ID 为 txtName 的属性值，代码如下。

```
protected void Page_Load(object sender, EventArgs e)
{
    TextBox2.Text = Server.HtmlEncode(TextBox1.Text);
}
```

网页中的用户可能会输入存在有潜在有害的客户端脚本。在默认情况下，Web 窗体页验证用户是否不包括脚本或 HTML 元素。在上述代码中通过 Server 对象的 HtmlEncode()方法对输入的文本内容进行 HTML 编码。

2．TextBox 控件的事件

TextBox 控件最常用的事件是 TextChanged 事件，当用户离开 TextBox 控件时就会引发该事件。在默认情况下并不会立即引发该事件，而是当下次发送窗体时在服务器代码中引发此事件，这是由于默认情况下 AutoPostBack 属性的值为 false。将该属性的值设置为 true，用户离开 TextBox 控件后可以将页面提交给服务器。

例如为 TextBox 控件添加 TextChanged 事件，当用户离开 TextBox1 文本框后触发该事件将值赋予 Label 控件进行显示，代码如下所示。

```
protected void TextBox1_TextChanged(object sender, EventArgs e)
{
```

```
    Label1.Text = Server.HtmlEncode(TextBox1.Text);
}
```

3．TextBox 控件的自动完成功能

许多浏览器都支持自动完成功能，该功能可以帮助用户根据以前输入的值向文本框中填充信息。自动完成的精确行为取决于浏览器，通常浏览器根据文本框的 name 特性存储值。任何同名的文本框（即使是在不同页上）都将为用户提供相同的值。有些浏览器还支持 vCard 架构，该架构允许用户使用预定义的名、姓、电话号码和电子邮件地址等值在浏览器中创建配置文件。

TextBox 控件支持 AutoCompleteType 属性，该属性为用户提供了用于控制浏览器如何使用自动完成的选项，如下所示。

- 禁用自动完成。如果开发人员不想让浏览器为文本框提供自动完成功能，可以将其禁用。
- 指定 vCard 值以方便用于字段的自动完成后，浏览器必须支持 vCard 架构。

3.3 按钮控件

ASP.NET 中提供了三种向服务器端提交表单的按钮控件：Button、LinkButton 和 ImageButton。这三种控件拥有同样的功能，但是每一种控件的外观截然不同，下面将分别对这三种控件进行介绍。

3.3.1 Button 控件

Button 控件是最常用的一种按钮控件，它通常被称为标准命令按钮。Button 控件显示了一个标准命令按钮，该按钮呈现为一个 HTML 的 input 元素。

Button 控件常用的属性除了 ID、Text 和 Width 属性外，Button 控件还有一些自身的属性。如表 3-5 所示了 Button 控件的特有属性。

表 3-5　Button 控件的特有属性

属 性 名 称	说　　明
CommandArgument	获取或设置可选参数，该参数与关联的 CommandName 一起被传递到 Command 事件
CommandName	获取或设置命令名，该命令名与传递给 Command 事件的 Button 控件相关联
CausesValidation	获取或设置一个值，该值指示在单击控件时是否执行验证
OnClientClick	获取或设置在引发某一个 Button 控件的 Click 事件时所执行的客户端脚本
PostBackUrl	获取或设置单击控件时从当前页发送到网页的 URL
UseSubmitBehavior	获取或设置一个布尔值，该值指示 Button 控件使用客户端浏览器的提交机制还是 ASP.NET 的回发机制。默认值为 true

Button 控件最常用的是 Click 事件和 Command 事件，它们都是在单击 Button 控件时引发。例如为 Button 控件添加常用的 Click 事件，在该事件中将文本框 txtUserName 的值赋予 Button 控件的 Text 值，代码如下所示。

```
protected void Button1_Click(object sender, EventArgs e)
{
    Button1.Text = txtUserName.Text;
}
```

【练习 3】

Button 控件不仅可以引发服务器事件，也可以引发客户端事件。服务器事件在回发后出现，这些事件在服务器端（即后台.CS 页面）进行处理。客户端事件在客户端脚本 JavaScript 中处理并在提交页面前引发。下面通过一个简单的示例步骤演示如何向 Button 控件中添加处理事件的客户端脚本，步骤如下。

（1）添加新的窗体页，在页面中添加 Button 控件并更改相关属性，如 ID 和 Text。

（2）添加 JavaScript 脚本函数，代码如下。

```
<script type="text/javascript">
    function confirm() {
        alert("执行向 Button 控件中添加处理 OnClientClick 事件的客户端脚本");
    }
</script>
```

（3）设置 Button 控件的 OnClientClick 属性，该属性设置要运行的客户端脚本，代码如下。

```
<asp:Button ID="Button1" runat="server" OnClientClick="confirm()" Text=
"Button" />
```

（4）运行页面单击按钮进行测试，效果不再显示。

注意

如果开发人员希望可以取消提交，需要将 OnClientClick 属性设置为字符串 Return 和函数名称。这样客户端脚本可以通过返回 false 取消提交。

3.3.2 LinkButton 控件

LinkButton 控件通常会被称为超级链接式按钮，它呈现为页面中的一个超链接。该控件的外观与 HyperLink 控件相同，但是它实现了与 Button 控件有关的功能，因此 LinkButton 控件的属性、事件以及用法可以参考 Button 控件。

3.3.3 ImageButton 控件

ImageButton 控件通常被称为图形化按钮，该控件对于提供丰富的按钮外观非常有用。另外，ImageButton 控件还提供了有关图形内已单击位置的坐标信息。

ImageButton 控件的使用与 Button 和 LinkButton 控件大同小异，因此该控件的主要属性和常用事件也可以参考 Button 控件。ImageButton 控件还有三个的常用属性，如下所示。

- **ImageUrl** 需要在 ImageButton 控件中显示图像的路径。
- **ImageAlign** 获取或设置 Image 控件相对于网页上其他元素的对齐方式。
- **AlternateText** 图像无法显示时显示的文本；如果图像可以显示则表示提示文本。

例如，在 Web 窗体页中添加 ImageButton 控件，然后在页面加载的 Load 事件中添加代码，ImageUrl 和 AlternateText 分别设置显示的图片路径和提示文本，代码如下所示。

```
protected void Page_Load(object sender, EventArgs e)
{
    ImageButton1.ImageUrl = "~/1.jpg";
    ImageButton1.AlternateText = "新图片";
}
```

3.4 选择控件

选择控件常常为用户提供选择，用户可以根据这些选择项来选择一个或多个内容。本节将详细介绍 ASP.NET 中提供的与选择相关的服务器控件。

3.4.1 RadioButton 控件

开发人员向 ASP.NET 网页中添加单选按钮时，可以使用两种 Web 服务器控件：单个 RadioButton 控件或 RadioButtonList 控件。RadioButton 控件通常被称做单选按钮，开发人员可以向页面中添加单个 RadioButton 控件，并且单独使用这些控件。单个 RadioButton 控件可以使读者更好地控制单选按钮的布局。

开发人员可以通过 RadioButton 控件的相关属性来设置显示的外观，常用的属性如表 3-6 所示。

表 3-6 RadioButton 控件的常用属性

属 性 名 称	说 明
CausesValidation	获取或设置一个值，该值指示选中控件时是否激发验证。默认值是 false
Checked	控件选中的状态，如果选中该值为 true；否则为 false
GroupName	指定单选按钮所属的组名，在一个组内每次只能选中一个单选按钮
TextAlign	获取或设置与控件关联的文本标签的对齐方式。其值有 Left 和 Right
Text	获取或设置与控件关联的文本标签

单个 RadioButton 控件在用户单击该控件时引发 CheckedChanged 事件。默认情况下，该事件并不导致向服务器发送页面，但是通过将 AotoPostBack 属性设置为 true 可以使该控件强制立即发送。

注意

若要在选中 RadioButton 控件时将其发送到服务器，浏览器必须支持 ECMAScript（如 Jscript 和 JavaScript），并且用户的浏览器要启用脚本撰写。

【练习 4】

用户在某些网站注册时常常会看到关于性别的选项（如“男”和“女”）。下面通过一个示例完成 RadioButton 控件的简单使用，步骤如下所示。

（1）创建新的 Web 窗体页并向页面添加内容和样式，在页面的合适位置添加 4 个 TextBox 控件、两个 RadioButton 控件和一个 Button 控件。

（2）将两个 RadioButton 控件设置 GroupName 属性相同的值，然后设置其他控件的相关属性，主要代码如下所示。

```
<form id="form1" runat="server">
    <table align="center" width="500px" border="0" height="150">
        <tr>
            <td align="right">用户名</td>
            <td><asp:TextBox ID="txtUserName" runat="server"></asp:TextBox>
            </td>
        </tr>
        <%--省略其他相关代码--%>
        <tr>
            <td align="right">性别</td>
```

```
            <td>
                <asp:RadioButton ID="rbBoy" GroupName="sex" Checked="true"
                runat="server"
AutoPostBack="True" oncheckedchanged="rbBoy_CheckedChanged" />男  
<asp:RadioButton ID="rbGirl" GroupName="sex" oncheckedchanged="rbBoy_
CheckedChanged" AutoPostBack="True" runat="server" />女
            </td>
        </tr>
        <tr>
            <td colspan="2" style="padding-left:150px;"> <asp:Button ID=
            "Button1" runat="server" Text="立即注册" width="120px" height=
            "36px" style= "background-color: #009933; list-style: none; border:
            0px;" /></td>
        </tr>
    </table>
</form>
```

（3）分别为两个 RadioButton 控件添加 CheckedChanged 事件，根据用户的提示选择不同的内容，代码如下所示。

```
protected void rbBoy_CheckedChanged(object sender, EventArgs e)
{
    if (Page.IsPostBack)            //页面回发
    {
        if (rbBoy.Checked)
            Page.ClientScript.RegisterStartupScript(Page.GetType(), "",
            "<script>alert('您选择了男！')</script>");
        else if (rbGirl.Checked)
            Page.ClientScript.RegisterStartupScript(Page.GetType(), "",
            "<script>alert('您选择了女！')</script>");
    }
}
```

上述代码中首先使用 Page 对象的 IsPostBack 属性判断页面是首次加载还是页面回发，如果是页面回发则根据控件的 Checked 属性值进行判断，然后通过 RegisterStartupScript()方法提示不同的内容。

（4）运行页面单击 RadioButton 控件进行测试，运行效果如图 3-1 所示。



图 3-1　RadioButton 控件示例

技巧

无论 RadioButton 控件是否发送到服务器，通常都没有必要为 CheckedChanged 事件创建事件处理程序，上面的示例仅仅作为练习。相反，更常见的做法是在窗体已被某个控件（如 Button 控件）发送到服务器时测试选定了哪个按钮。

3.4.2 RadioButtonList 控件

RadioButton 控件和 RadioButtonList 控件都允许用户从一组互相排斥的预定义选项中进行选择，并且它们允许开发人员定义任意数目带标签的单选按钮，并将它们水平或垂直排列。

与 RadioButton 控件不同，RadioButtonList 控件是单个控件，它可以作为一组单选按钮列表项的父控件。由于 RadioButtonList 控件派生自 ListControl 基类，因此工作方式与列表控件（如 ListBox、DropDownList 和 BulletedList 等）很相似。

RadioButtonList 控件可以包含 RadioButton 控件的多个属性，同时由于该控件可以动态地绑定数据源，因此与 RadioButton 控件相比，RadioButtonList 控件可以拥有一些特有的属性。如表 3-7 所示对这些属性进行了说明。

表 3-7 RadioButtonList 控件的常用属性

属 性 名 称	说 明
DataSourceID	获取或设置控件的 ID，数据绑定控件从该控件中检索其数据项列表
DataSource	指定该控件绑定的数据源
DataTextField	获取或设置为列表项提供文本内容的数据源字段
DataValueField	获取或设置为各列表项提供值的数据源字段
Item	列表控件项的集合
SelectedIndex	获取或设置列表中选中项的最低序号索引
SelectedItem	获取列表控件中索引最小的选定项
SelectedValue	获取列表控件中选定项的值，或选择列表控件中包含指定值的项
RepeatColumns	获取或设置在该控件上显示的列数
RepeatDirection	获取或设置组中单选按钮的显示方向，它的值有 Vertical（默认值）和 Horizontal
RepeatLayout	获取或设置一个值，该值指定是否使用 table 元素、ul 元素、ol 元素或 span 元素呈现列表。其值分别是 Table、Flow、UnorderedList 和 OrderedList

与单个 RadioButton 控件相反，RadioButtonList 控件在用户更改列表中选定的单选按钮时会引发 SelectedIndexChanged 事件。默认情况下，该事件并不导致向服务器发送页面，但是可以通过将 AutoPostBack 属性设置为 true 来指定此选项。

单选按钮很少单独使用，单选按钮分组时有两种方式，如下所示。

- 首先向页中添加单个 RadioButton 控件，然后将所有这些控件手动分配到一个组中。
- 向页中添加 RadioButtonList 控件，该控件中的列表将自动分组。

上一节已经介绍过 RadioButton 控件的使用，下面通过一个简单的示例演示如何使用 RadioButtonList 控件进行分组。

【练习 5】

本次练习根据用户对图书类型的选择显示相应的选择内容，主要步骤如下。

（1）添加新的 Web 窗体页，在页面的合适位置添加 RadioButtonList 控件，然后设置将控件的 AutoPostBack 属性设置为 true，RepeatLayout 属性设置为 UnorderedList。

（2）在【属性】窗口中找到 Item 属性，然后在弹出的提示框中单击【添加】按钮完成多个图

书类型的添加，完整代码如下所示。

```
<asp:RadioButtonList ID="RadioButtonList1" RepeatLayout="UnorderedList"
AutoPostBack="True" onselectedindexchanged="RadioButtonList1_
SelectedIndexChanged" runat="server" >
    <asp:ListItem Selected="True">言情小说</asp:ListItem>
    <asp:ListItem>军事小说</asp:ListItem>
    <asp:ListItem>穿越小说</asp:ListItem>
    <asp:ListItem>悬疑小说</asp:ListItem>
    <asp:ListItem>武侠小说</asp:ListItem>
    <asp:ListItem>励志小说</asp:ListItem>
    <asp:ListItem>古典小说</asp:ListItem>
    <asp:ListItem>其他小说</asp:ListItem>
</asp:RadioButtonList>
```

（3）接着向页面中添加 Label 控件，然后设置该控件的 ID 属性和 Text 属性等，代码如下所示。

```
<asp:Label ID="lblAnswer" Text="选择结果: " runat="server" style="padding-left:
25px;"></asp:Label>
```

（4）为 RadioButtonList 控件添加 SelectedIndexChanged 事件，在该事件中将用户的选择内容显示到 Label 控件中，代码如下。

```
protected void RadioButtonList1_SelectedIndexChanged(object sender, EventArgs e)
{
    lblAnswer.Text = "选择结果: " + RadioButtonList1.SelectedValue;
}
```

（5）运行本次练习的代码进行测试，最终效果如图 3-2 所示。

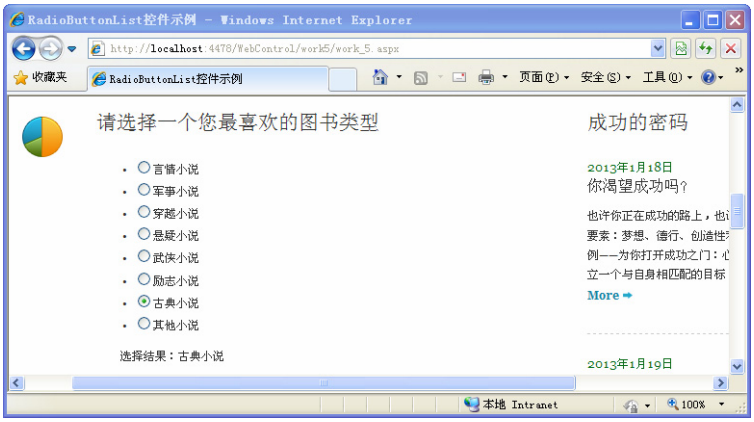


图 3-2　RadioButtonList 控件示例

3.4.3　CheckBox 控件

CheckBox 控件也叫复选框，它在 Web 窗体页面上显示为一个复选框，常用于为用户提供多项选择。使用该控件比使用 CheckBoxList 控件能够更好地控制页面上各个复选框的布局，例如可以在各个复选框之间包含文本（即非复选框的文本）。

CheckBox 控件的一般形式如下所示。

```
<asp:CheckBox ID="控件名称" runat="server" AutoPostBack="true | false" Checked=
"true | false" Text="控件文字" TextAlign="left | right" OnCheckedChanged="事件
程序名称" />
```

从语法形式中可以看出，除了 ID 和 runat 属性外，CheckBox 控件最常用四个属性和一个事件，属性的说明如下所示。

- **AutoPostBack** 默认值为 false，设置当使用者选择不同的项目时，是否自动触发 CheckedChanged 事件。
- **Checked** 该属性传回或设置是否该项目被选取。
- **TextAlign** 该属性设置控件所显示的文字是在按钮的左方还是右方。
- **Text** 该属性设置 CheckBox 控件所显示的文本内容。

单个 CheckBox 控件在用户单击该控件时会引发 CheckedChanged 事件，但是由于 AutoPostBack 属性的值为 false，因此默认情况下，该事件并不导致向服务器发送页面。

【练习 6】

新招聘的员工向人事部门报道的时候，主管通常会要求他们填写自己的住宅电话和工作电话，一般情况下员工的住宅电话和工作电话都是一致的。下面通过简单的练习完成对 CheckBox 控件的训练，当用户选中该控件时自动将工作电话和住宅电话保持一致，主要步骤如下所示。

（1）添加新的页面，然后在页面中添加两个 TextBox 控件和一个 CheckBox 控件，代码如下所示。

```
<form id="form1" runat="server">
    <p>
        家庭电话:
        <asp:TextBox ID="homeTel" runat="server" /><br />
        工作电话:
        <asp:TextBox ID="workTel" runat="server" />
        <asp:CheckBox ID="check1" Text="与家庭电话一致" TextAlign="Right"
        AutoPostBack="True" OnCheckedChanged="Check" runat="server" />
    </p>
</form>
```

（2）用户选中 CheckBox 控件时会引发 CheckedChanged 事件，在事件中通过 Checked 属性进行判断，代码如下所示。

```
protected void check1_CheckedChanged(object sender, EventArgs e)
{
    if (check1.Checked)
        workTel.Text = homeTel.Text;
    else
        workTel.Text = "";
}
```

（3）运行页面输入的内容进行测试，最终效果不再显示。

【练习 7】

在许多企业网站或系统的后台会看到一系列的列表，用户可以选中某个复选框或单击类似于名称“是否全选”的按钮实现所有复选框的全选和全不选效果。实现该功能的主要步骤如下。

（1）添加新的 Web 窗体页，在页面的合适位置添加全选和不选的 CheckBox 控件，并且设置 AutoPostBack 属性的值为 true，代码如下。

```
<asp:CheckBox ID="CheckBox1" runat="server" oncheckedchanged="CheckBox1_
CheckedChanged" AutoPostBack="True" />
```

（2）添加详细列表内容，在该段内容中通过 for 语句循环添加，主要代码如下。

```
<% for (int i = 0; i < 8; i++) { %>
    <tr>
        <td height="20" bgcolor="#FFFFFF">
            <div align="center"><asp:CheckBox ID="CheckBoxNo" runat="server"
            /></div>
        </td>
        <td height="20" bgcolor="#FFFFFF">
            <div align="center" class="STYLE1"><div align="center">0<%=i+1
            %></div></div>
        </td>
        <td height="20" bgcolor="#FFFFFF">
            <div align="center"><span class="STYLE1">13813916585</span></div>
        </td>
        <td height="20" bgcolor="#FFFFFF">
            <div align="center"><span class="STYLE1">2007-11-16 15:00:20
            </span></div>
        </td>
        <td bgcolor="#FFFFFF">
            <div align="center"><span class="STYLE1">tiezhu0902@163.com
            </span></div>
        </td>
        <td height="20" bgcolor="#FFFFFF">
            <div align="center"><span class="STYLE4"><img src="images/edt.
           gif" width="16" height="16" />编辑   <img src="images/del.
            gif" width="16" height="16" />删除</span></div>
        </td>
    </tr>
<% } %>
```

（3）为全选和全不选的 CheckBox 控件（ID 是 CheckBox1）添加 CheckedChanged 事件，根据 Checked 属性完成全选和全不选的功能，代码如下所示。

```
protected void CheckBox1_CheckedChanged(object sender, EventArgs e)
{
    if (CheckBox1.Checked)              //实现全选效果
       CheckBoxNo.Checked = true;
    else                                //实现全不选效果
       CheckBoxNo.Checked = false;
}
```

（4）运行页面代码选中 CheckBox 控件进行测试，全选效果如图 3-3 所示。

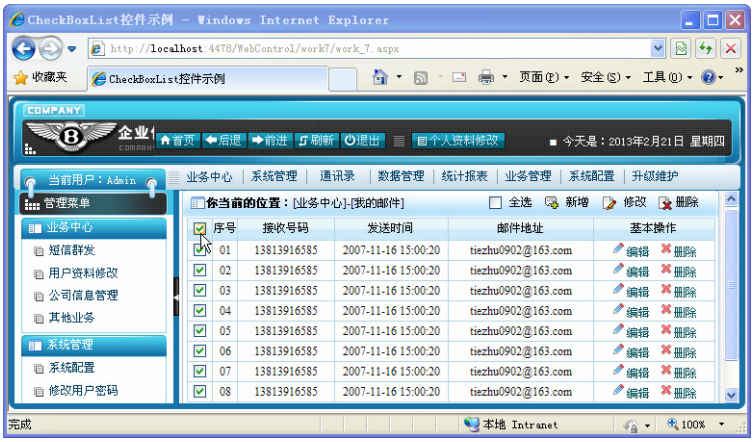


图 3-3　CheckBox 控件的全选效果

3.4.4　CheckBoxList 控件

除了 CheckBox 控件外，CheckBoxList 控件也能够将复选框添加到 Web 窗体页中，而且这两种控件都为用户提供了一种输入布尔型数据（例如真或假、是或否以及开或关等）的方法。

如果想用数据库中的数据创建一组复选框，则 CheckBoxList 控件是较好的选择。另外，由于 CheckBoxList 控件可以动态地绑定数据源，因此该控件比 CheckBox 控件有一些不同的属性。如表 3-8 所示对这些属性进行了说明。

表 3-8　CheckBoxList 控件的常用属性

属 性 名 称	说　　明
DataSourceID	获取或设置控件的 ID，数据绑定控件从该控件中检索其数据项列表
DataSource	指定该控件绑定的数据源
DataMember	用户绑定的表或视图
DataTextField	获取或设置为列表项提供文本内容的数据源字段
DataTextFormatString	获取或设置格式化字符串，该字符串用来控制如何显示绑定到列表控件的数据
DataValueField	获取或设置为各列表项提供值的数据源字段
Items	获取列表项的集合
SelectedIndex	获取或设置列表中选中项的最低序号索引
SelectedItem	获取列表控件中索引最小的选定项
SelectedValue	获取列表控件中选定项的值，或选择列表控件中包含指定值的项
RepeatColumns	获取或设置在该控件上显示的列数
RepeatDirection	获取或设置组中单选按钮的显示方向，它的值有 Vertical（默认值）和 Horizontal
RepeatLayout	获取或设置一个值，该值指定是否使用 table 元素、ul 元素、ol 元素或 span 元素呈现列表

除了上述属性外，CheckBoxList 控件与 CheckBox 控件的事件也不相同，CheckBoxList 控件会引发 SelectedIndexChanged 事件。默认情况下，该事件并不导致向服务器发送窗体，但是可以通过 AutoPostBack 属性的值来控制。

【练习 8】

本次练习演示了 CheckBoxList 控件的简单使用，主要步骤如下所示。

（1）添加新的 Web 窗体页，在页面的合适位置添加 CheckBoxList 控件和其他内容，并向该控件中添加集合项，代码如下所示。

```
<asp:CheckBoxList ID="CheckBoxList1" runat="server" AutoPostBack="True"
epeatLayout="OrderedList" onselectedindexchanged="CheckBoxList1_
SelectedIndexChanged">
    <asp:ListItem Selected="True">中国</asp:ListItem>
    <asp:ListItem>中国台湾</asp:ListItem>
    <asp:ListItem>中国香港</asp:ListItem>
    <asp:ListItem>法国</asp:ListItem>
    <asp:ListItem>美国</asp:ListItem>
    <asp:ListItem>泰国</asp:ListItem>
    <asp:ListItem>韩国</asp:ListItem>
    <asp:ListItem>新加坡</asp:ListItem>
    <asp:ListItem>马来西亚</asp:ListItem>
    <asp:ListItem>其他国家</asp:ListItem>
</asp:CheckBoxList>
<asp:Label ID="lblAnswer" Text="您想要去的地方：" runat="server" Style="padding-
left: 25px;"></asp:Label>
```

（2）为 CheckBoxList 控件添加 SelectedIndexChanged 事件，在事件的代码中遍历全部的集合项，然后再根据 Selected 属性判断某项是否选中，如果选中则显示文本内容，代码如下所示。

```
protected void CheckBoxList1_SelectedIndexChanged(object sender, EventArgs e)
{
    lblAnswer.Text = "您想要去的地方：";
    foreach (ListItem item in CheckBoxList1.Items)      //遍历集合项
    {
        if (item.Selected)                              //某项是否选中
            lblAnswer.Text += item.Value + "、";
    }
    lblAnswer.Text = lblAnswer.Text.Trim('、');
}
```

（3）运行页面选择内容进行测试，运行效果如图 3-4 所示。



图 3-4　CheckBoxList 控件示例

3.5 列表控件

ASP.NET 提供了一系列的列表控件，这些控件可以将多个数据以列表的形式进行呈现。常见的列表控件包括：ListBox 控件、DropDownList 控件和 BulletedList 控件。下面将简单介绍这些控件的相关属性、事件和使用方法。

3.5.1 BulletedList 控件

BulletedList 控件创建一个无序或有序（编号的）的项列表，它们分别呈现为 HTML 的 ul 或 ol 元素。通过 BulletedList 控件可以实现以下效果。

- 可以指定项、项目符号或编号的外观。
- 静态定义列表项或通过将控件绑定到数据来定义列表项。
- 也可以在用户单击项时作出响应。

通过创建静态项或将控件绑定到数据源，可以定义 BulletedList 控件的列表项。而且通过该控件的相关属性可以设置其外观效果，如表 3-9 所示了 BulletedList 控件的常用属性，并对这些属性进行说明。

表 3-9 BulletedList 控件的常用属性

属 性 名 称	说 明
AppendDataBoundItems	获取或设置一个值，该值指示是否在绑定数据之前清除列表项。默认值为 false
BulletImageUrl	获取或设置为控件中的每个项目符号显示的图像路径，把 BulletStyle 的值设置为 CustomImage 时有效
BulletStyle	获取或设置控件的项目符号样式
DataSource	获取或设置对象，数据绑定控件从该对象中检索其数据项列表
DataTextField	获取或设置为列表项提供文本内容的数据源字段
DataValueField	获取或设置为列表项提供值的数据源字段
DisplayMode	获取或设置控件中的列表内容的显示模式。其值包括 Text（默认值）、HyperLink 和 LinkButton
FirstBulletMember	获取或设置排序控件中开始列表项编号的值
Items	获取列表控件项的集合

BulletedList 控件可以通过 BulletStyle 属性自定义列表项外观，如果将控件设置为呈现项目符号，则可以选择与 HTML 标准项目符号样式匹配的预定义项目符号样式字段。BulletStyle 属性的值有 10 个，同一个值在不同的浏览器中呈现项目符号的方式会不同，甚至有些浏览器不支持特定的项目符号样式（如 Disc 字段）。BulletStyle 属性的值如下所示。

- **NotSet** 未设置。
- **Numbered** 数字。
- **LowerAlpha** 小写字母。
- **UpperAlpha** 大写字母。
- **LowerRoman** 小写罗马数字。
- **UpperRoman** 大写罗马数字。
- **Disc** 实心圆。
- **Circle** 圆圈。
- **Square** 实心正方形。

❑ **CustomImage** 自定义图像。

通过 BulletedList 控件添加集合项时有多种方法。例如，在【属性】窗口中找到 Items 属性，然后单击后面的按钮弹出【ListItem 集合编辑器】对话框，如图 3-5 所示。在对话框中分别设置 Text 属性和 Value 属性。Text 属性定义控件在页上显示的内容；Value 属性定义第二个值，该值不会显示，但用户在选择某个项时能返回该值。

所有项添加完成后单击图 3-5 中的【确定】按钮进行添加，添加完成后的主要代码所示。

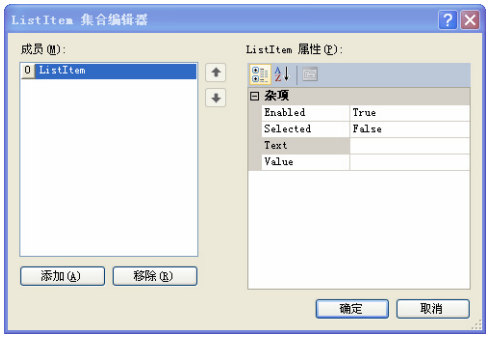


图 3-5 为 BulletedList 控件添加集合项

```
<asp:BulletedList ID="BulletedList1" runat="server" BulletStyle="Circle">
    <asp:ListItem>新建项 1</asp:ListItem>
    <asp:ListItem>新建项 2</asp:ListItem>
    <%--省略其他相关 ListItem 代码--%>
</asp:BulletedList>
```

BulletedList 控件的 Items 属性返回所有集合项的列表对象，该对象包含多个方法，通过这些方法可以添加指定的项、删除项或查找项等，它们的具体说明如表 3-10 所示。

表 3-10 集合列表项的常用方法

方 法 名 称	说 明
Add()	将表示指定字符串的 ListItem 项添加到集合的结尾
AddRange()	将 ListItem 数组中的项添加到集合
Clear()	从集合中移除所有的 ListItem 项
CopyTo()	将集合中的项复制到 Array 中，从指定的索引开始复制
Insert()	将指定的 ListItem 插入到集合中的指定索引位置
Remove()	从集合中移除指定的 ListItem
RemoveAt()	从集合中移除指定索引位置的 ListItem

集合列表项除了上面列举的方法外，还有一个最常用的属性 Count，该属性获取集合中 ListItem 的对象总数。例如开发人员分别通过 Add()、AddRange()和 Insert()方法向集合中添加 3 项，然后分别通过 RemoveAt()进行删除。

```
BulletedList1.Items.Add(new ListItem("Add()方法添加内容", "Add()方法添加内容"));
ListItem[] li = { new ListItem("AddRange()添加 1", "AddRange()添加 1"), new
ListItem("AddRange()添加 2", "AddRange()添加 2") };
BulletedList1.Items.AddRange(li);
BulletedList1.Items.Insert(0, new ListItem("插入到第 1 位", "插入到第一位"));
BulletedList1.Items.RemoveAt(1);
```

【练习 9】

细心的用户可以发现许多网站中的内容都可以使用 BulletedList 控件来实现，例如在某个图书网站中需要对一个月内的图书进行排行，统计本月内销售 1-10 名的图书。首先在 Web 页面中添加 BulletedList 控件，然后将该控件 BulletStyle 属性的值设置为 Numbered，DisplayMode 属性的值设置为 HyperLink，接着向该控件中依次添加前 10 名的销售图书，主要代码如下。

```
<asp:BulletedList ID="BulletedList1" runat="server" BulletStyle="Numbered"
DisplayMode="HyperLink">
    <asp:ListItem>中国古建筑艺术大观 1 门窗艺术</asp:ListItem>
```

```
        <asp:ListItem>中国美术字体图说</asp:ListItem>
        <asp:ListItem>巨匠的艺术</asp:ListItem>
        <asp:ListItem>中国古代艺术论著集注与研究</asp:ListItem>
        <asp:ListItem>拍摄自然图案</asp:ListItem>
        <asp:ListItem>水彩技法百科全书</asp:ListItem>
        <asp:ListItem>现代生活的英雄：论现实主义</asp:ListItem>
        <asp:ListItem>破除迷信十三讲-汉字书法通解...</asp:ListItem>
        <asp:ListItem>世界艺术欣赏--日本绘画艺术</asp:ListItem>
        <asp:ListItem>魔幻与科幻绘画技法百科全书</asp:ListItem>
    </asp:BulletedList>
```

添加完成后运行页面进行测试，最终效果如图 3-6 所示。



图 3-6　BulletedList 控件示例

注意

BulletedList 控件可以动态地绑定数据源，如果需要将这些数据源添加到静态项的后面，可以将该控件 AppendDataBoundItems 属性的值设置为 true，这样就可以将静态列表项与绑定数据的列表项组合起来。

3.5.2　DropDownList 控件

DropDownList 控件使用户可以从单项选择下拉列表框中进行选择，该控件通常被称做下拉列表。开发人员也可以把 DropDownList 控件看做是容器，这些列表项都属于 ListItem 类型，每一个 ListItem 对象都是带有单独属性（如 Text 属性、Selected 属性和 Value 属性）的对象。

开发人员可以通过 DropDownList 控件的 Width 和 Height 控制其外观，部分浏览器不支持以像素为单位设置的高度和宽度时，这些浏览器将使用行计数设置。除了 Width 和 Height 外，表 3-11 显示了其他的常用属性。

表 3-11　DropDownList 控件的常用属性

属 性 名 称	说　　明
AutoPostBack	获取或设置一个值，该值指示用户更改列表中的内容时是否自动向服务器回发
DataSource	获取或设置对象，数据绑定控件从该对象中检索其数据项列表
DataTextField	获取或设置为列表项提供文本内容的数据源字段
DataValueField	获取或设置各列表项提供值的数据源字段
Items	获取列表控件项的集合
SelectedIndex	获取或设置列表中选定项的最低索引
SelectedItem	获取列表控件中索引最小的选定项
SelectedValue	获取列表控件中选定项的值，或选择列表控件中包含指定值的项

除了属性外，DropDownList 控件经常使用 SelectedIndexChanged 事件，当用户使用某一项时将会引发该事件。

【练习 10】

许多网站都会对用户进行了权限设置，因此用户在登录时常常会根据不同的角色或者不同的身份进行登录。下面以简单的登录为例，当用户选择以不同的方式登录时，向 TextBox 控件中添加不同的内容，主要步骤如下。

（1）添加新的 Web 窗体页，在页面的合适位置添加两个 DropDownList 控件、一个 Button 控件、一个 CheckBox 控件和 3 个 TextBox 控件。

（2）分别为添加的控件设置属性，将第一个 DropDownList 控件（ID 为 ddlSelRole）的 AutoPostBack 属性的值设置为 true，主要代码如下所示。

```
<form method="post" name="login" id="loginform" class="gateform" runat=
"server">
    <table cellspacing="0" cellpadding="0" class="formtable">
        <tbody>
            <tr>
                <th>
                    <asp:DropDownList ID="ddlSelRole" runat="server"
                    AutoPostBack="true" OnSelectedIndexChanged="ddlSelRole_
                    SelectedIndexChanged">
                        <asp:ListItem>用户名</asp:ListItem>
                        <asp:ListItem>UID</asp:ListItem>
                        <asp:ListItem>Email</asp:ListItem>
                    </asp:DropDownList>
                </th>
                <td><asp:TextBox ID="txtName" runat="server"></asp:TextBox>
                </td>
            </tr>
            <%--省略其他内容项代码--%>
        </tbody>
    </table>
</form>
```

（3）为 ID 是 ddlSelRole 的 DropDownList 控件添加 SelectedIndexChanged 事件，在该事件中通过 SelectedIndex 属性判断用户选择登录的方式，然后为 TextBox 控件赋值，代码如下所示。

```
protected void ddlSelRole_SelectedIndexChanged(object sender, EventArgs e)
{
    if (ddlSelRole.SelectedIndex == 0)              //通过用户名登录
    {
        txtName.Text = "ForverLove";
    }
    else if (ddlSelRole.SelectedIndex == 1)         //通过 UID 登录
    {
        txtName.Text = "2993414";
    }
    else if (ddlSelRole.SelectedIndex == 2)         //通过 Email 登录
    {
```

```
            txtName.Text = "2982145@163.com";
        }
    }
```

（4）运行页面选择内容进行测试，当用户选择 Email 时的效果如图 3-7 所示。

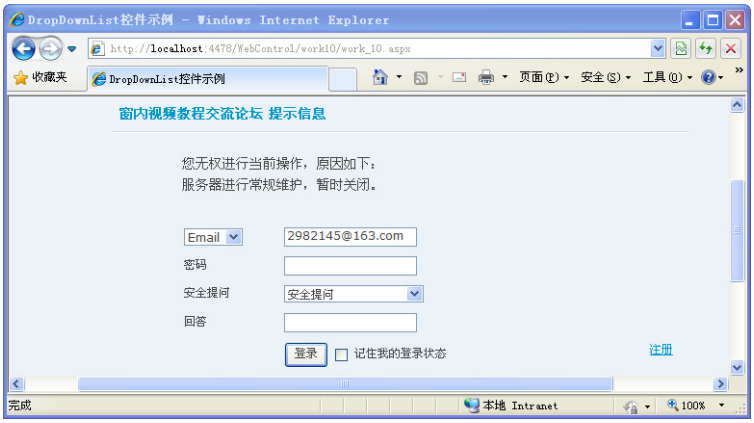


图 3-7　DropDownList 控件示例

3.5.3　ListBox 控件

ListBox 控件允许用户从预定义的列表中选择一项或多项，它类似于 DropDownList 控件，但是它们之间也存在着不同，ListBox 控件可以一次显示多个项，还可以使用户能够选择多个项。

ListBox 控件通常用于一次显示一个以下的项，读者可以通过两个方法控制列表的外观，如下所示。

- **显示的行数**　可以将该控件设置为显示特定的项数，如果该控件中包含比设置的项数更多的项，则显示一个垂直滚动条。
- **宽度和高度**　可以以像素为单位设置控件的大小。在这种情况下，控件将忽略已设置的行数，而是显示足够多的行直至填满控件的高度。

1．ListBox 控件的属性

ListBox 控件中有许多常用的属性，如 DataSource、AutoPostBack、SelectedIndex 和 SelectedValue 等。表 3-12 对常用的属性进行了说明。

表 3-12　ListBox 控件的常用属性

属 性 名 称	说　　明
AutoPostBack	获取或设置一个值，该值指示当用户更改列表中的选定内容时是否自动向服务器回发
DataSource	获取或设置对象，数据绑定控件从该对象中检索其数据项列表
DataTextField	获取或设置为列表项提供文本内容的数据源字段
DataValueField	获取或设置各列表项提供值的数据源字段
Items	获取列表控件项的集合
Rows	获取或设置该控件中显示的行数
SelectedIndex	获取或设置列表中选定项的最低索引
SelectedItem	获取列表控件中索引最小的选定项
SelectedValue	获取列表控件中选定项的值，或选择列表控件中包含指定值的项
SelectionMode	获取或设置控件的选择模式，它的值有两个，分别为 Single 和 Multiple。默认为 Single

通常情况下，用户可以通过单击列表中的单个项来选择。如果将 ListBox 控件 SelectionMode

属性的值设置为 Multiple（即允许进行多重选择），则用户可以在按住 Ctrl 或 Shift 键的同时，单击以选择多个项。

以下通过 SelectedIndex 属性设置了 ListBox 控件的索引项。

```
ListBox1.SelectedIndex = 2;
```

2．ListBox 控件的事件

当用户选择某一项时，ListBox 控件就会引发 SelectedIndexChanged 事件。但是默认情况下，该事件不会导致将页发送到服务器。例如，在 ListBox 的 SelectedIndexChanged 事件中判断某一项是否选中，代码如下所示。

```
protected void ListBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    foreach (ListItem item in ListBox1.Items)
    {
        if (item.Selected)
        {
            //省略代码
        }
    }
}
```

3．向 ListBox 控件添加选项

ASP.NET 可以使用三种方式向 ListBox 控件添加项，如下所示。

- 在设计时添加静态项。
- 使用编程的方式在运行时添加项。
- 使用数据绑定添加项。

如下代码通过 DataSource、DataTextField 和 DataValueField 属性演示了如何通过编程动态绑定数据。

```
ListBoxShow.DataSource = GetPlaceList();    //该方法可以返回 DataTable 对象
ListBoxShow.DataTextField = "placeName";    //对应后台数据库中的字段名
ListBoxShow.DataValueField = "placeId";     //对应后台数据库的字段名
```

4．确定 ListBox 控件的所选内容

使用 ListBox 控件最常见的内容是确定用户已选择了哪一项或哪些项，主要取决于该控件允许单项选择还是多重选择。有两种方式确定单项选择列表控件的选定内容，如下所示。

- 如果获取选择项的索引值，需要使用 SelectedIndex 属性的值。该属性的索引是从 0 开始的，如果没有选择任何项，则该属性的值是-1。
- 如果获取选择项的内容，需要使用 SelectedItem 属性，该属性返回一个 ListItem 类型的对象。通过该对象的 Text 属性或 Value 属性可以获取选择项的内容。

如果 ListBox 控件允许多重选择时，确定 ListBox 控件所选中的内容，需要依次通过控件的 Items 集合，分别测试每一项的 Selected 属性，主要代码如下。

```
Protected void Button1_Click(object sender, System.EventArgs e)
{
   string msg = "" ;
   foreach(ListItem li in ListBox1.Items)          //遍历集合中的内容
```

```
    {
        if(li.Selected == true)                          //判断某一项是否选中
        {
            msg += "<br>" + li.Text + " is selected.";
        }
    }
    Label1.Text = msg;
}
```

5．ListBox 控件简单示例

上面已经了解过与 ListBox 控件相关的属性、事件以及简单操作等内容，下面通过一个简单的示例对 ListBox 控件进行演示。

【练习 11】

下面通过一个简单的练习使用 ListBox 控件，用户可以分别向该控件中添加内容，也可以清空每个 ListBox 控件的内容，还可以向左或向右移动一项或多项，主要步骤如下。

（1）添加新的 Web 窗体页，在页面中的合适位置添加两个 ListBox 控件、6 个 Button 控件和一个 TextBox 控件。然后分别设置这些控件的相关属性，页面的最终设计效果如图 3-8 所示。（注意：将左侧的 ListBox 控件的 SelectedMode 属性设置为 Multiline）



图 3-8　ListBox 控件示例

（2）分别为两个 ListBox 控件的【添加新项】按钮添加 Click 事件，在该事件中判断新添加的项是否存在，如果存在则弹出提示，否则调用 Add()方法进行添加。以左侧的 ListBox 控件为例，代码如下。

```
protected void btn1New_Click(object sender, EventArgs e)
{
    string str = Server.HtmlEncode(txtAddName.Text);    //对文本框中输入的内容进
                                                        行编码
    ListItem lv = new ListItem(str);
    if (ListBox1.Items.Contains(lv))
        Page.ClientScript.RegisterStartupScript(GetType(), "", "<script>alert
        ('已经存在该项，请重新添加！');</script>");
     else
         ListBox1.Items.Add(lv);
 }
```

（3）分别单击两侧的【清空新项】按钮，在按钮的 Click 事件中调用控件的 Clear()方法清空每个 ListBox 控件中的所有内容。以左侧的 ListBox 控件为例，代码如下。

```
protected void btn1Clear_Click(object sender, EventArgs e)
{
    ListBox1.Items.Clear();
}
```

（4）选中左侧的内容一项或多项后单击【>>】按钮实现将左侧内容添加到右侧，代码如下。

```
protected void btnRight_Click(object sender, EventArgs e)
{
    if (ListBox1.Items.Count > 0 && ListBox1.SelectedIndex >=0)//判断 ListBox
                                                              控件中的元素个数
    {
        for (int i = ListBox1.Items.Count - 1; i >= 0; i--)//遍历 ListBox 控件中
                                                           的所有元素
        {
            if (ListBox1.Items[i].Selected == true)      //判断某一项是否选中
            {
                ListBox2.Items.Add(ListBox1.Items[i]);    //向右侧列表添加选中的项
                ListBox1.Items.Remove(ListBox1.Items[i]);//删除左侧列表中选中的项
            }
        }
    }
    ListBox2.ClearSelection();          //清除右侧列表中所有选中的项
    if (ListBox2.Items.Count > 0)
        ListBox2.Items[ListBox2.Items.Count - 1].Selected = true;//将最后一项选中
}
```

上述代码首先判断 ListBox 控件中的元素个数和选中索引，接着遍历该控件中的所有元素，调用 Selected 属性判断是否选中，如果选中向 ListBox2 控件中添加，同时删除 ListBox1 中的选中项。

（5）选中右侧 ListBox 控件中的内容，然后单击【<<】按钮将右侧内容项添加到左侧，同时将 ListBox2 的选中项删除。因为该控件只能选中一项，因此代码比上一步简单得多，如下所示。

```
protected void btnLeft_Click(object sender, EventArgs e)
{
    if (ListBox2.SelectedIndex != -1)
    {
        Page.ClientScript.RegisterStartupScript(GetType(), "", "<script>alert
        ('已经存在该项，请重新添加！');</script>");
        ListBox1.Items.Add(new ListItem(ListBox2.SelectedItem.Text));
        ListBox2.Items.Remove(new ListItem(ListBox2.SelectedItem.Text));
    }
}
```

（6）运行页面添加内容进行测试，添加完成后选中左侧内容的多个内容项，然后单击按钮，最终效果如图 3-9 和图 3-10 所示。

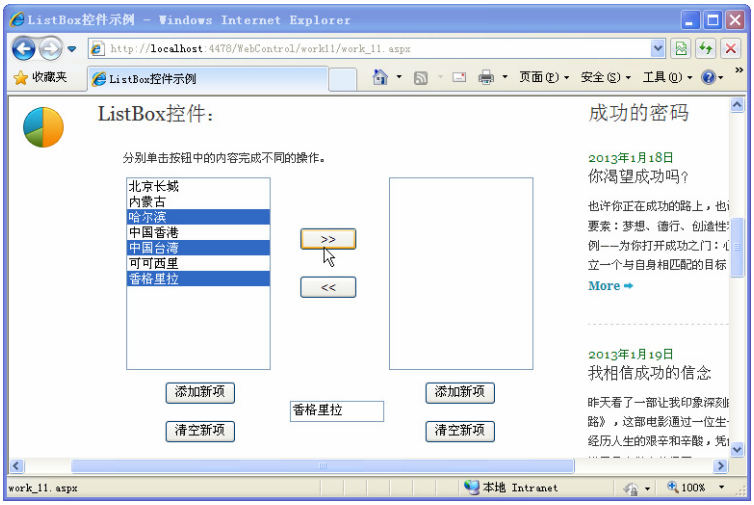


图 3-9　选中左侧内容项的效果

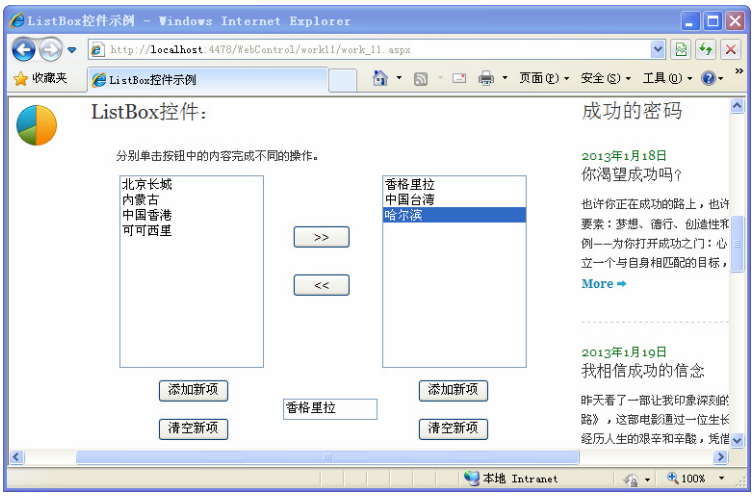


图 3-10　添加选项完成后的效果

3.6 容器控件

顾名思义，容器控件是能够将其他控件放置在该控件中，即该控件中可以放置 0 个、1 个或多个其他控件。容器控件可以作为 Web 服务器控件、HTML 服务器控件和 HTML 元素对象的父控件。ASP.NET 中提供了多个与容器相关的控件，本节将介绍最常用的两个控件：MultiView 控件和 Panel 控件。

3.6.1 MultiView 控件

MultiView 控件表示用做 View 控件组容器的控件。它允许开发人员定义一组 View 控件，其中每个 View 控件均包含子控件，例如在联机调查应用程序中。然后开发人员的应用程序可以根据用户标识、用户喜好或传递到查询字符串参数的信息等条件，向客户端呈现一个特定的 View 控件。

MultiView 控件和 View 控件可以用做其他控件和标记的容器，并提供了一种可以方便地显示信息的替换视图的方式。可以执行以下两个任务，如下所示。

- 根据用户选择或其他条件提供备选控件集。
- 创建多页窗体。

由于 MultiView 控件通常和 View 控件一起使用，因此该控件通常被称做多视图控件。一个 MultiView 控件中可以包含多个 View 控件，但是 MultiView 控件的活动控件只能是这些 View 控件中的一个。无论是 MultiView 控件还是各个 View 控件，除了当前 View 控件的内容外，都不会在页面中显示任何标记。

MultiView 控件最常用的属性有两个：ActiveViewIndex 和 Views。ActiveViewIndex 用来获取或设置 MultiView 控件的活动 View 控件的索引；而 Views 控件用来获取 MultiView 控件的 View 控件的集合。

设置 MultiView 控件的 ActiveViewIndex 属性可以在视图之间移动。另外，MultiView 控件还支持可以添加到每个 View 控件的导航按钮。如果要创建导航按钮，可以向每个 View 控件添加一个按钮控件（Button、LinkButton 或 ImageButton），然后将每个按钮的 CommandName 和 CommandArgument 属性设置为保留值，这些保留值可以使 MultiView 控件移动到另一个视图。如表 3-13 列出了保留的 CommandName 值和相对应的 CommandArgument 值。

表 3-13　CommandName 和 CommandArgument 值

CommandName 值	CommandArgument 值
NextView	（没有值）
PrevView	（没有值）
SwitchViewByID	要切换到的 View 控件的 ID
SwitchViewByIndex	要切换到的 View 控件的索引号

【练习 12】

网站中的个人空间相册，用户单击某张图片后可以直接查看下一张，下面通过 MultiView 控件和导航按钮模拟实现其效果。首先向 Web 窗体页中添加 MultiView 控件，接着向该控件中添加 4 个 View 控件，每个 View 控件都分别包含一个 Image 控件和 Button 控件，主要代码如下所示。

```
<asp:MultiView ID="MultiView1" runat="server">
    <asp:View ID="View1" runat="server">
        <asp:Image ID="img1" runat="server" ImageUrl="images/images_01.jpg">
        </asp:Image><br />
        <asp:Button ID="btnNext1" runat="server" Text="下一张" CommandArgument=
        "View2" CommandName="SwitchViewByID"></asp:Button>
    </asp:View>
    <%--省略中间 3 个项的内容--%>
    <asp:View ID="View4" runat="server">
        <asp:Image ID="img4" runat="server" ImageUrl="images/images_04.jpg">
        </asp:Image><br />
        <asp:Button ID="btnNext4" runat="server" Text="重新开始"
        CommandArgument="View1" CommandName="SwitchViewByID"></asp:Button>
    </asp:View>
</asp:MultiView>
```

在上述代码中为每一个 View 控件中的 Button 控件都添加 CommandName 属性和 CommandArgument 属性，并且对它们进行了赋值。完成后运行页面进行测试，最终运行效果如图 3-11 所示。

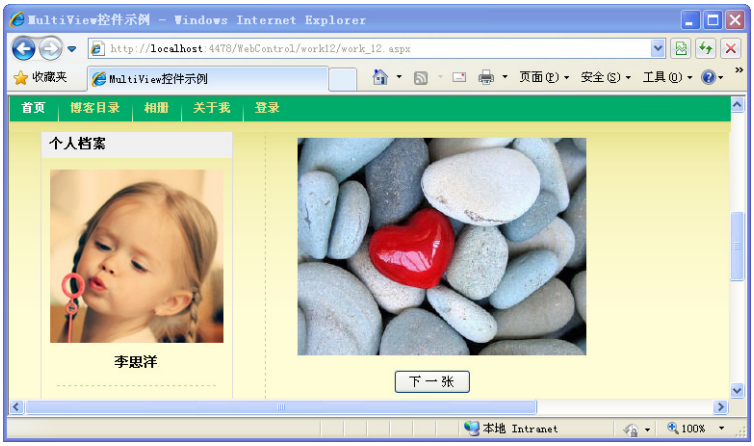


图 3-11　MultiView 控件示例

3.6.2　Panel 控件

Panel 控件也是一种容器控件，可以将该控件作为其他控件的容器，该控件通常会被称做面板。当开发人员需要以编程的方式创建内容并需要一种将内容插入到页中的方法时，Panel 控件最为合适。该控件的三种常用方式如下。

- 对于一组控件和相关的标记可以通过把其放置在 Panel 控件中，然后操作该控件的方式将它们作为一个单元进行管理。
- 可将 TextBox 控件和 Button 控件放置在 Panel 控件中，然后通过将 Panel 控件的 DefaultButton 属性设置为面板中某个按钮的 ID 来定义一个默认的按钮。如果用户在面板内的文本框中进行输入时按 Enter，这与用户单击特定的默认按钮具有相同的效果。
- 部分控件（如 TreeView）没有内置的滚动条，通过在 Panel 控件中放置滚动条控件可以添加滚动行为。可以通过该控件的 Height 和 Width 属性添加滚动条，将 Panel 控件设置为特定的大小，然后再设置 ScrollBars 属性。

与其他控件一样，Panel 控件也包含多个属性，通过这些属性可以设置控件的外观和文本方向等。如表 3-14 所示了 Panel 控件的常用属性。

表 3-14　Panel 控件的常用属性

属 性 名 称	说　　明
BackImageUrl	控件背景图像文件和 URL
DefaultButton	控件中默认按钮的 ID
Direction	控件中内容的显示方向，默认值为 NoSet，其他的值有 LeftToRight 和 RightToLeft
GroupingText	获取或设置控件中包含的控件组的标题，如果指定了滚动条则设置该属性将不显示滚动条
ScrollBars	获取或设置控件中滚动条的可见性和位置
HorizontalAlign	获取或设置面板内容的水平对齐方式
Wrap	获取或设置一个指示面板中内容是否换行的值

注意

不能在 Panel 控件中同时指定滚动条和分组文本，如果设置了分组文本，其优先级高于滚动条。

【练习 13】

例如向 Web 窗体页中添加两个 Panel 控件，接着分别设置 GroupingText 和 ScrollBars 属性，

然后分别向这两个控件中添加不同的内容，添加完成后运行页面观察效果，代码如下。

```
<asp:Panel ID="Panel1" runat="server" GroupingText="必填信息" BackColor=
"AliceBlue" Width="300">
    用户名: <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox><br /><br />
    密   码: <asp:TextBox ID="TextBox2" runat="server"></asp:
    TextBox>
</asp:Panel>
<asp:Panel ID="Panel2" runat="server" ScrollBars="Both" BackColor="AliceBlue"
Width="300">
    联系电话: <asp:TextBox ID="TextBox3" runat="server"></asp:TextBox><br /><br />
    联系地址: <asp:TextBox ID="TextBox4" runat="server" Height="68px"
            Width="148px"></asp:TextBox>
</asp:Panel>
```

3.7 其他常用控件

除了上面介绍的控件之外，ASP.NET 中还提供了其他的一些常用控件，如日历控件 Calendar、广告控件 AdRotator、与图像相关的 Image 控件和 ImageMap 控件等。

3.7.1 Image 控件

Image 控件可以在 ASP.NET 网页上显示图像，它常被称为图像控件。与其他控件不同，Image 控件不支持任何事件。通常情况下，将通过使用 ImageMap 或 ImageButton 控件来创建交互式图像。

Image 控件通常使用四个属性设置相关的样式，如表 3-15 对 ImageButton 控件的常用属性进行了说明。

表 3-15　Image 控件的常用属性

属 性 名 称	说　　明
Width	显示图像的宽度
Height	显示图像的高度
ImageAlign	图像的对齐方式
ImageUrl	要显示图像的 URL

除了显示图像外，Image 控件还可以为图像指定各种类型的文本。与之相关的属性如下所示。

- **ToolTip**　在一些浏览器中作为工具提供显示的文本。
- **AlternateText**　在无法找到图形文件时显示的文本。如果没有指定任何 ToolTip 属性，某些浏览器将使用 AlternateText 值作为工具提示。
- **GenerateEmptyAlternateText**　该属性的值如果为 true，所呈现的图像元素的 alt 特性将设置为空字符串。

ImageUrl 属性可以显示指定的文本，还可以通过该属性绑定数据源，以显示数据库后台的图像。Image 控件的简单使用代码如下。

```
<asp:Image ID="imgShow" ImageUrl="~/sg_icon.png" runat="server" Width="200px"
Height="200px" />
```

3.7.2 ImageMap 控件

ImageMap 控件创建具有用户可以单击单个区域的图像，这些单个的区域点称为作用点。每一个作用点都可以是一个单独的超链接或回发事件。

ImageMap 控件由两个元素组件：一个是图像，它可以是任何标准的 Web 图形格式的图形（如.jpg、.gif 或.png）；另一个是 HotSpot 的集合，每个作用点都是一个类型为 CircleHotSport、RectangleHotSpot 和 PolygonHotSport 的不同项。

ImageMap 控件可以包含 Image 控件的属性，除了那些属性外，该控件还有两个重要属性：HotSpots 和 HotSpotMode。HotSpotMode 属性表示获取或设置单击 HotSpot 对象时该控件对象的默认行为。它的值是 HotSpotMode 枚举的值之一，具体说明如下。

- **NotSet** 未设置。默认情况下控件会执行导航操作，即导航到指定的网页；如果未指定导航的网页，则导航到当前网站的根目录。
- **Navigate** 导航到指定的网页，如果未指定导航的网页，则导航到当前网站的根目录。
- **PostBack** 执行回发操作，用户单击区域时执行预先定义的事件。
- **Inactive** 无任何操作，这时该控件和 Image 控件的效果一样。

【练习 14】

向 Web 窗体页中添加 ImageMap 控件，然后向该控件中分别添加 CircleHotSport、RectangleHotSpot 和 PolygonHotSport，并且分别设置 HotSpotMode 属性、PostBackValue 属性和 PostBackValue 属性，代码如下所示。

```
<asp:ImageMap ID="ImageMap1" ImageUrl="images/images_03.jpg" runat="server"
ImageAlign="Left" style="border:none" onclick="ImageMap1_Click" Width="500">
    <asp:CircleHotSpot Radius="80" X="100" Y="100" AlternateText="圆形区域"
    HotSpotMode="PostBack" PostBackValue="CH" />
    <asp:RectangleHotSpot Bottom="200" Left="300" Right="500" Top="0"
    HotSpotMode="PostBack" AlternateText="方形区域" PostBackValue="RH" />
    <asp:PolygonHotSpot Coordinates="100,100,300,300,200,300" HotSpotMode=
    "PostBack" PostBackValue="PH" AlternateText="多边形区域" />
</asp:ImageMap>
```

为 ImageMap 控件添加 Click 事件，用户单击不同部分的区域时弹出提示，代码如下所示。

```
protected void ImageMap1_Click(object sender, ImageMapEventArgs e)
{
    String region = "";
    switch (e.PostBackValue)
    {
        case "CH":
            region = "圆形区域";
            break;
        case "RH":
            region = "方形区域";
            break;
        case "PH":
            region = "多边形区域";
            break;
    }
```

```
        Page.ClientScript.RegisterStartupScript(GetType(), "", "<script>alert
        ('"+region+"');</script>");
    }
```

添加完成后运行页面查看鼠标悬浮时的效果，然后单击区域查看弹出的提示，其具体效果不再显示。

3.7.3 Calendar 控件

Calendar 控件也叫日历控件，它显示日历中的可选日期，并显示与特定日期关联的数据。使用该控件可以执行以下操作。

- 捕获用户交互（例如在用户选择一个日期或一个日期范围时）。
- 自定义日历的外观。
- 在日历中显示数据库中的信息。

Calendar 控件可以显示某个月的日历，也允许用户选择日期，还可以跳转日期到前一个或下一个月。如表 3-16 对 Calendar 控件的主要属性进行了具体说明。

表 3-16 Calendar 控件的主要属性

属 性 名 称	说 明
Caption	日历的标题
CaptionAlign	日历标题文本的对齐方式。其值包含 NotSet（默认值）、Bottom、Left、Right 和 Top
DayNameFormat	获取或设置一周中各天的名称格式。其值包括 Short（默认值）、Full、FirstTwoLetters 、FirstLetter 和 Shortest
FirstDayOfWeek	获取或设置要在控件的第一天列中显示的一周中的某天。默认值为 Default
NextMonthText	获取或设置为下一个月导航控件显示的文本
NextPrevFormat	获取或设置控件的标题部分中下个月和上个月导航元素的格式。其值包括 CustomText（默认值）、FullMonth 和 ShortMonth
PrevMonthText	获取或设置为前一个月导航控件显示的文本
SelectedDate	获取或设置选定的日期
SelectionMode	获取或设置日期的选定模式。其值包括 Day（默认值）、DayWeek、DayWeekMonth 和 None
SelectMonthText	获取或设置为选择器列中月份选择元素显示的文本
SelectWeekText	获取或设置为选择器列中周选择元素显示的文本
ShowDayHeader	获取或设置一个值，该值指示是否显示一周中每天的标头。默认值为 true
ShowGridLines	获取或设置一个值，该值指示是否用网格线分隔控件上的日期。默认值为 false
ShowNextPrevMonth	获取或设置一个值，该值指示控件是否在标题部分显示下个月和上个月导航元素。默认值为 false
TitleFormat	获取或设置标题部分的格式。默认值为 MonthYear

读者在调用 SelectedDate 属性获取选定的日期时，该属性返回一个 DateTime 对象，通过 DateTime 对象的各个方法可以获取不同的内容。例如调用 Add()方法可以返回一个新的日期对象；调用 ToShortDateString()方法将 DateTime 对象的值转换为等效的短日期字符串表示形式。

提示

除了上述属性外，Calendar 控件还可以使用其他属性指定外观内容。如 ForeColor 属性设置文本的颜色；DayHeaderStyle 属性设置标头行的样式；以及 BackColor 设置背景颜色等。

Calendar 控件主要使用两个事件：DayRender 和 SelectionChanged。当控件创建要发送到浏览器的输出时引发 DayRender 事件，在准备要显示的日期时将为每个日期引发该事件；当用户通

过单击日期选择器控件选择一天、一周或整月时引发该事件。

在 DayRender 事件的方法中有两个参数：包括对引发事件 Calendar 控件的引用和 DayRenderEventArgs 类型的对象。该对象提供对另外两个对象的访问。

- **Cell** 它是一个 TableCell 对象，可以用于设置个别日的外观。
- **Day** 可以用于查询关于呈现日的信息。该对象不仅支持各种可用于了解有关日的信息的属性（如 IsSelected 和 IsToday 等），还支持 Controls 集合，可以操作该集合以将内容添加到日中。

【练习 15】

在本次练习中，分别演示了 DayRender 和 SelectionChanged 事件的使用，用户选定某个日期后重新向页面显示选定的日期，步骤如下所示。

（1）在 Web 窗体页中添加一个 Calendar 控件和两个 Label 控件，这两个 Label 控件分别显示选中日期和当前日期是本年的第几天。

（2）为 Calendar 控件添加 DayRender 事件，在该事件中通过代码自定义个别日的外观，代码如下所示。

```
protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
{
    Style vacationStyle = new Style();                              //设置外观样式
    vacationStyle.BackColor = System.Drawing.Color.Yellow;
    vacationStyle.BorderColor = System.Drawing.Color.Purple;
    vacationStyle.BorderWidth = 3;
    Style weekendStyle = new Style();                               //设置周样式
    weekendStyle.BackColor = System.Drawing.Color.Green;
    if ((e.Day.Date >= new DateTime(2000, 11, 23)) && (e.Day.Date <= new DateTime
    (2000,11, 30)))
    {
        e.Cell.ApplyStyle(vacationStyle);
    }
    else if (e.Day.IsWeekend)
    {
        e.Cell.ApplyStyle(weekendStyle);
    }
}
```

（3）为该控件添加 SelectionChanged 事件，在该事件中分别通过 SelectedDate 属性返回的 DateTime 对象的相关方法显示内容，代码如下所示。

```
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
 {
     Label1.Text = Calendar1.SelectedDate.ToLongDateString();
     Label2.Text = "2013 年的第 " + Calendar1.SelectedDate.DayOfYear.ToString()
     + " 天";
 }
```

（4）运行示例代码选定日期进行测试，最终效果不再显示。

3.8 实例应用：图书作者注册

3.8.1 实例目标

在本节之前，已经通过大量的知识和练习介绍了 Web 服务器控件，这些控件通常在添加或修改内容时使用，如某个论坛网站的用户注册、修改某个会员的基本资料或用户登录等。熟悉腾讯图书的用户可以知道，如果想要发表自己的某篇小说，必须通过图书作者的注册，才能发表原创作品。

本节以用户在腾讯图书注册的作者为例，介绍如何使用常用的 Web 服务器控件，最终的运行效果如图 3-12 所示。



图 3-12 实例应用运行效果

3.8.2 技术分析

读者可以分析效果图 3-12，根据该图可以总结所使用的相关技术，主要技术如下所示。

- 使用 Panel 控件分别显示用户注册时的必填内容和选填信息。
- 使用 TextBox 控件完成用户的输入内容，如作者笔名、真实姓名、身份证号以及出生日期等。
- 使用 RadioButton 控件表示用户注册时的性别“男”和“女”。
- 单击出生日期文本框时显示 Calendar 控件，否则不显示。
- 使用 3 个 DropDownList 控件分别表示省份、城市和县/区。
- 使用 CheckBoxList 控件表示用户的一系列兴趣爱好。

- 使用按钮控件 Button 和 LinkButton 分别表示名称为“提交”和“取消”的按钮。
- 当用户选择 Calendar 控件的日期后，将选中的日期显示到出生日期文本框中。

3.8.3 实现步骤

用户注册成为作者时的主要步骤如下。

（1）添加新的 Web 窗体页，首先添加第一个 Panel 面板控件，接着在该控件中分别添加 6 个 TextBox 控件和两个 RadioButton 控件，设置其相关属性，主要代码如下。

```
<asp:Panel ID="Panel1" runat="server" GroupingText="必填内容" BorderColor="red"
Width="620">
    <li><span class="tit">笔名</span><asp:TextBox ID="txtBiName" class="txt"
    min="1" MaxLength="12" runat="server"></asp:TextBox><em>*</em><span
    class="note">最多 12 个字符，且笔名不能为空</span> </li>
    <li><span class="tit">性别</span>
        <asp:RadioButton ID="rbBoy" GroupName="sex" runat="server" />男 
            <asp:RadioButton ID="rbGirl" GroupName="sex"
        Checked="true" runat="server" />女 <em>*</em></li>
    <%--省略其他相关代码--%>
</asp:Panel>
```

（2）接着在合适的位置添加第二个 Panel 控件，分别向 Panel 控件中添加 4 个 TextBox 控件、3 个 DropDownList 控件、一个 CheckBoxList 控件和 Calendar 控件。其相关主要代码如下。

```
<asp:Panel ID="Panel2" GroupingText="选填信息" runat="server" Width="620">
    <li><span class="tit">出生日期</span>
        <asp:TextBox ID="txtBirth" class="txt" MaxLength="6" runat="server"
        AutoPostBack="true" onfocus="javascript:ShowCalendar();" onblur=
        "javascript:NoneCalendar();"></asp:TextBox><br />
        <div id="showrili" runat="server" style="display: none;position:
        absolute;">
            <asp:Calendar OnSelectionChanged="Calendar1_SelectionChanged"
            ID="Calendar1" runat="server" BackColor="#92FAD2"></asp:Calendar>
        </div>
    </li>
    <li><span class="tit">所在地</span><span id="homeTownSelect">
        <asp:DropDownList ID="ddlCountry" name="addr_c" runat="server">
            <asp:ListItem Value="1" Selected="True">中国</asp:ListItem>
            <asp:ListItem Value="2">北美洲</asp:ListItem>
            <%--省略添加其他内容--%>
        </asp:DropDownList>
            <%--省略添加其他城市和地区--%>
    </span></li>
    <li><span class="tit">兴趣爱好</span>
        <asp:CheckBoxList ID="CheckBoxList1" runat="server" RepeatColumns=
        "10">
            <asp:ListItem Value="写作" Selected="True"> 写作 </asp:ListItem>
            <asp:ListItem Value="爬山"> 爬山 </asp:ListItem>
            <asp:ListItem Value="唱歌"> 唱歌 </asp:ListItem>
```

```
                <asp:ListItem Value="跳舞"> 跳舞 </asp:ListItem>
                <asp:ListItem Value="其他"> 其他 </asp:ListItem>
            </asp:CheckBoxList>
        </li>
        <%--省略邮政编码和详细地区代码--%>
        <li><span class="tit">作者简介</span><asp:TextBox ID="txtIntro" class=
        "txt" MaxLength="100" Columns="100" runat="server" Height="100px"
        TextMode="MultiLine"></asp:TextBox>100 个字符以内</li>
    </asp:Panel>
```

（3）分别为出生日期文本框添加焦点和离开焦点事件，这两个事件分别调用 JavaScript 中不同的函数，在函数中通过 display 属性的值来控制，主要代码如下所示。

```
<script type="text/javascript">
    function ShowCalendar() {
        document.getElementById("showrili").style.display = "block";
    }
    function NoneCalendar() {
        document.getElementById("showrili").style.display = "none";
    }
</script>
```

（4）为 Calendar 控件添加 SelectionChanged 事件，在该事件中将选中的值赋予 txtBirth 控件，并且将隐藏包含 Calendar 控件的 div 元素，代码如下。

```
protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    txtBirth.Text = Calendar1.SelectedDate.ToShortDateString();
    showrili.Style.Add("display", "none");
}
```

（5）在合适的位置分别添加 Button 控件和 LinkButton 控件，然后设置相关属性，主要代码如下。

```
<asp:Button ID="btnJiao" runat="server" Style="cursor: hand" class="btn2" Text=
"提    交" />
<asp:LinkButton ID="lbtnXiao" runat="server" class="gray">取消</asp:
LinkButton>
```

（6）运行页面将鼠标移动至出生日期文本框中进行测试，具体效果不再显示。

3.9 拓展训练

1．BulletedList 控件的显示使用

大部分的网站中都包括一些帮助内容，这些内容提示用户如何进行简单的操作。本次练习主要通过设置 BulletedList 控件的 BulletStyle 属性显示帮助内容，页面的最终效果如图 3-13 所示。



图 3-13　拓展训练 1 运行效果

2．多个 Web 服务器控件的使用

添加新的 Web 窗体页，在页面的合适位置依次添加本节介绍的控件，然后运行页面查看效果。感兴趣的读者也可以更改某个控件某个属性的多个不同值，这些控件的最终运行效果不再显示。

3.10 课后练习

一、填空题

1. _____控件不支持包括位置特性在内的任何样式特性。
2. Button 控件的_____属性表示与控件相关联的命令名。
3. 读者可以将 RadioButton 控件_____属性的值设置为 true 表示向服务器发送页面。
4. CheckBox 控件的_____属性表示某一项是否被选取。
5. 如果 Image 控件设置的图像路径不存在，可以通过_____属性设置无法显示图像时的文本。
6. 将 ListBox 控件的 SelectionMode 属性的值设置为_____时允许用户选择多个项目。
7. MultiView 控件通常和 View 控件一起使用，该控件常用的两个属性是 Views 和_____。

二、选择题

1. BulletedList 控件的 BulletStyle 属性的值_____表示呈现的项目符号是实心圆。
   A. Circle
   B. Disc
   C. Square
   D. Numbered
2. 下面关于 Image 和 ImageMap 控件的说法，选项_____是正确的。
   A. Image 控件比 ImageMap 控件的功能更加强大，所有使用 ImageMap 的地方都可以使用 Image 控件来代替
   B. ImageMap 控件比 Image 控件的功能更加强大，所有使用 Image 控件的地方都可以使用 ImageMap 控件来代替
   C. Image 控件不支持任何事件，通常情况会通过 ImageMap 控件或 ImageButton 控件创建交互式图像
   D. ImageMap 控件不支持任何事件，通常情况会通过 Image 控件或 ImageButton 控件来创建交互式图像
3. 关于服务器控件的说法，_____选项是正确的。
   A. 服务器控件必须分为 HTML 服务器控件和 Web 服务器控件
   B. Validation 服务器控件属于 Web 服务器控件，但是它与 Web 服务器控件最大的不同在于 Validation

控件不包含 runat=server 属性

C. HTML 服务器控件包含了 runat=server 属性，将属性去掉后就是一般的 HTML 元素

D. Web 服务器控件中包含了 runat=server 属性，可以将该属性去掉，去掉后该类控件就是一般的 HTML 元素

4. 如果用户需要获取当前日期（即该天）是这一年中的第几天时，需要通过_____属性获取。

A. FirstDayOfWeek

B. Caption

C. SelectedDate.DayOfMonth

D. SelectedDate.DayOfYear

5. _____选项对于选择控件的说法是错误的。

A. RadioButton 控件和 RadioButtonList 控件属于单选按钮控件，而 CheckBox 控件和 CheckBoxList 控件则属于复选框控件

B. 选择控件都有多个共同的属性，如 RepeatColumns、RepeatDirection 以及 RepeatLayout 属性等

C. 由于选择控件的 AutoPostBack 属性的默认值为 false，因此默认情况下，选择控件的事件并不会向服务器发送页面

D. 多个 RadioButton 控件需要指定其 GroupName 属性，该控件表示在一个组内只能选中一个单选按钮

三、简答题

1. 请从三个方面详细说明 HTML 服务器控件与 Web 服务器控件的区别。

2. 简单说明 Image 控件和 ImageMap 控件的区别。

3. 请简答 Button 控件、LinkButton 控件和 ImageButton 控件的适用情况。

4. 请口述如何向 ASP.NET 服务器控件添加客户端事件。