

项目3 会员注册

3.1 项目引入

会员注册是 Smart On Line 电子商城的必不可少的一项功能,也是进行客户分析数据的来源之一。李明是 Smart On Line 电子商城的一位开发人员,现接到任务,要求协助开发用户注册功能模块。

3.2 项目分析

经过小组讨论和仔细分析,认为 Smart On Line 电子商城用户注册功能可以使用 ASP.NET 常见的服务器控件来设计界面;此外还认为,为使客户拥有良好的体验,对界面输入数据的合法性应进行验证;在验证功能中还须提供验证码功能以防恶意注册。最后当用户单击“注册”按钮时,将用户输入信息保存到数据库中。因此,项目经理将这个项目分为三个任务,一是用户注册界面设计,二是用户输入信息的验证,最后是保存注册用户信息。

3.3 知识准备

3.3.1 ASP.NET 常用服务器控件

每个控件都有一些公共属性,例如字体颜色、边框的颜色、样式等。在 Visual Studio 2012 中,当开发人员用鼠标选择了相应的控件后,属性栏中会简单地显示该属性的作用,如图 3-1 所示。

属性栏用来设置控件的属性,当控件在页面中被初始化时,这些将被应用到控件中。控件的属性也可以通过编程的方法在页面相应代码区域编写,示例代码如下。

```
protected void Page_Load(object sender, EventArgs e)
{
    Label1.Visible=false;
}
```

上述代码编写了一个 Page_Load(页面加载)事件,当页面初次被加载时,会执行 Page_Load 事件中的代码。这里通过编程的方法对控件的属性进行了更改。当页面加载时,控件的属性会被应用并呈现在浏览器中。

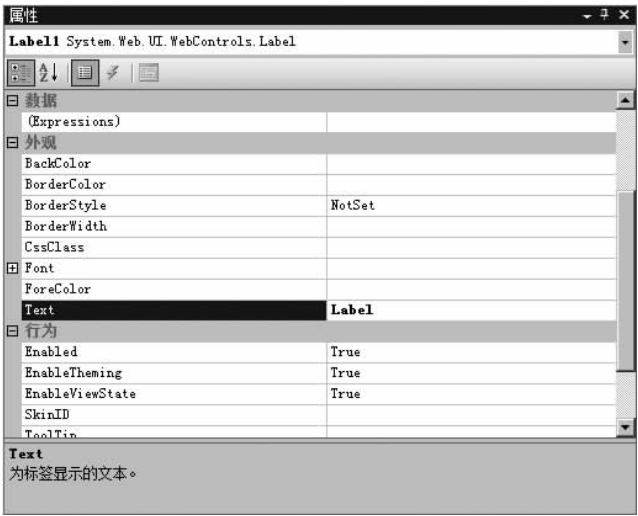


图 3-1 控件的属性

ASP.NET 提供了诸多控件,这些控件包括简单控件、数据库控件、登录控件等强大的控件。在 ASP.NET 中,简单控件是最基础也是经常被使用的控件,简单控件包括标签控件(Label)、超链接控件(HyperLink)以及图像控件(Image)等。

1. 标签控件(Label)

在 Web 应用中,希望显示的文本不能被用户更改;或者当触发事件时,某一段文本能够在运行时更改,则可以使用标签控件(Label)。开发人员可以非常方便地将标签控件拖放到页面中,该页面将自动生成一段标签控件的声明代码,示例代码如下。

```
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
```

上述代码中声明了一个标签控件,并将这个标签控件的 ID 属性设置为默认值 Label1。由于该控件是服务器端控件,所以在控件属性中包含“runat=“server””属性。该代码还将标签控件的文本初始化为 Label,开发人员能够配置该属性并进行不同文本内容的呈现。同样,标签控件的属性能够在相应的.cs 代码中初始化,示例代码如下所示。

```
protected void Page_PreInit(object sender,EventArgs e)
{
    Label1.Text="Hello World";
}
```

上述代码在页面初始化时将 Label1 的文本属性设置为“Hello World”。

2. 超链接控件

超链接控件(HyperLink)相当于实现了 HTML 代码中的“”显示的效果,当然,超链接控件有自己的特点,当拖动一个超链接控件到页面中时,系统会自动生成控件声明代码,示例代码如下所示。

```
<asp:HyperLink ID="HyperLink1" runat="server">HyperLink</asp:HyperLink>
```

上述代码声明了一个超链接控件,相对于 HTML 代码形式,超链接控件可以通过传递

指定的参数来访问不同的页面。当触发了一个事件后,超链接的属性可以被改变。超链接控件通常使用的两个属性如下。

- ImageUrl: 要显示的图像的 URL。
- NavigateUrl: 要跳转的 URL。

(1) ImageUrl 属性

设置 ImageUrl 属性可以设置这个超链接是以文本形式显示还是以图片文件显示,虽然表现形式不同,但是不管是图片形式还是文本形式,全都实现相同的效果。

(2) Navigate 属性

Navigate 属性可以为无论是文本形式还是图片形式的超链接设置超链接属性,即将要跳转的页面,示例代码如下所示。

```
<asp:HyperLink ID="HyperLink1" runat="server"
    ImageUrl="~/Shop/Images/log.gif"
    NavigateUrl="~/Shop/About.aspx">
    HyperLink
</asp:HyperLink>
```

上述代码使用了图片超链接的形式。当单击此超链接控件后,浏览器将跳转到名为 About.aspx 的页面中。

(3) 动态跳转

在前面讲解了超链接控件的优点,超链接控件的优点在于能够对控件进行编程,来按照用户的意愿跳转到自己跳转的页面。以下代码实现了当用户选择 Sina 时,会跳转到 Sina 网站;如果选择 Sohu,则会跳转到 Sohu 页面。示例代码如下所示。

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (DropDownList1.Text == "sina")
    {
        HyperLink1.NavigateUrl="http://www.sina.com";
    }
    else
    {
        HyperLink1.NavigateUrl="http://www.sohu.com";
    }
}
```

上述代码使用了 DropDownList 控件,当用户选择不同的值时,对 HyperLink1 控件进行操作。值得注意的是,如果只是为了单纯地实现超链接,同样不推荐使用 HyperLink 控件,因为过多的使用服务器控件同样有可能造成性能问题。

3. 图像控件

图像控件(Image)用来在 Web 窗体中显示图像,图像控件常用的属性如下。

- AlternateText: 在图像无法显示时,显示备用的文本。
- ImageAlign: 图像的对齐方式。
- ImageUrl: 要显示图像对应的 URL。

当图片无法显示的时候,图片将被替换成 AlternateText 属性中的文字,ImageAlign 属

性用来控制图片的对齐方式,而 ImageUrl 属性用来设置图像连接地址。同样,HTML 中也可以使用来替代图像控件。图像控件具有可控性的优点,就是通过编程可以控制图像控件。除了显示图形以外,Image 控件的其他属性还允许为图像指定各种文本,如 GenerateEmptyAlternateText 属性,如果将此属性设置为 True,则要呈现的图片的 alt 属性将设置为空。

4. 文本框控件

在 Web 开发中,Web 应用程序通常需要和用户进行交互,例如用户注册、登录、发帖等,那么就需要文本框控件(TextBox)来接受用户输入的信息。开发人员还可以使用文本框控件制作高级的文本编辑器并用于 HTML,以及文本的输入/输出。通常情况下,默认的文本框控件(TextBox)是一个单行的文本框,用户只能在文本框中输入一行内容。通过修改该属性,则可以将文本框设置为多行或者以密码的形式显示,文本框控件常用的控件属性如下。

- AutoPostBack: 在文本修改以后,是否自动回传。
- Columns: 文本框的宽度。
- EnableViewState: 控件是否自动保存其状态以用于往返过程。
- MaxLength: 用户输入的最大字符数。
- ReadOnly: 是否为只读。
- Rows: 作为多行文本框时所显示的行数。
- TextMode: 文本框的模式,设置单行、多行、密码和数字等。
- Wrap: 文本框是否换行。

(1) AutoPostBack(自动回传)属性

在网页的交互中,如果用户提交了表单,或者执行了相应的方法,那么该页面将会发送到服务器上,服务器将执行表单的操作或者执行相应方法后,再呈现给用户,例如按钮控件、下拉菜单控件等。如果将某个控件的 AutoPostBack 属性设置为 True 时,则如果该控件的属性被修改,那么同样会使页面自动发回到服务器。

(2) EnableViewState(控件状态)属性

ViewState 是 ASP.NET 中用来保存 Web 控件回传状态的一种机制,它是由 ASP.NET 页面框架管理的一个隐藏字段。在回传发生时,ViewState 数据同样将回传到服务器,ASP.NET 框架解析 ViewState 字符串并为页面中的各个控件填充该属性。而填充后,控件通过使用 ViewState 将数据重新恢复到以前的状态。在使用某些特殊的控件时,如数据库控件,来显示数据库。每次打开页面执行一次数据库往返过程是非常不明智的。开发人员可以绑定数据,在加载页面时仅对页面设置一次,在后续的回传中,控件将自动从 ViewState 中重新填充,减少了数据库的往返次数,从而不使用过多的服务器资源。在默认情况下,EnableViewState 的属性值通常为 True。

(3) 其他属性

上面的两个属性是比较重要的属性,其他的属性也经常使用。

- MaxLength: 在注册时可以限制用户输入的字符串长度。
- ReadOnly: 如果将此属性设置为 True,那么文本框内的值是无法被修改的。
- TextMode: 此属性可以设置文本框的模式,例如单行、多行和密码形式。默认情况下,不设置 TextMode 属性,那么文本框默认为单行。

文本框无论是在 Web 应用程序开发还是 Windows 应用程序开发中都是非常重要的。文本框在用户交互中能够起到非常重要的作用。在文本框的使用中,通常需要获取用户在文本框中输入的值或者检查文本框属性是否被改写。当获取用户的值的时候,必须通过一段代码来控制。例如当用户单击按钮控件时,需要实现标签控件的文本改变。为了实现相应的效果,可以通过编写 cs 文件代码进行逻辑处理,示例代码如下所示:

```
protected void Button1_Click(object sender, EventArgs e) //双击按钮时触发的事件
{
    Label1.Text=TextBox1.Text;
}
```

上述代码中,当双击按钮时,就会触发一个按钮事件,这个事件就是将文本框内的值赋值到标签内。同样,双击文本框控件,会触发 TextChange 事件。而当运行时,当文本框控件中的字符变化后,并没有自动回传,这是因为在默认情况下,文本框的 AutoPostBack 属性被设置为 False。当 AutoPostBack 属性被设置为 True 时,文本框的属性变化,则会发生回传,示例代码如下所示。

```
protected void TextBox1_TextChanged(object sender, EventArgs e)
{
    Label1.Text=TextBox1.Text;
}
```

上述代码中,为 TextBox1 添加了 TextChanged 事件。在 TextChanged 事件中,并不是每一次文本框的内容发生了变化之后,就会重传到服务器,这一点和 WinForm 是不同的,因为这样会大大降低页面的执行效率。而当用户将文本框中的焦点移出导致 TextBox 会失去焦点时,才会发生重传。

5. 按钮控件(Button, LinkButton, ImageButton)

在 Web 应用程序和用户交互时,常常需要提交表单、获取表单信息等操作,因此按钮控件是非常必要的。按钮控件能够触发事件,或者将网页中的信息回传给服务器。在 ASP.NET 中,包含三类按钮控件,分别为 Button、LinkButton、ImageButton。

按钮控件的通用属性说明如下。

按钮控件用于事件的提交,按钮控件包含一些通用属性,按钮控件的常用通用属性包括以下几种。

- Causes Validation: 按钮是否导致激发验证检查。
- CommandArgument: 与此按钮管理的命令参数。
- CommandName: 与此按钮关联的命令。

按钮控件对应的事件通常是 Click(单击)和 Command(命令)事件。在 Click 事件中,通常用于编写用户单击按钮时所需要执行的事件。按钮控件中,Click 事件并不能传递参数,所以处理的事件相对简单。而 Command 事件可以传递参数,负责传递参数的是按钮控件的 CommandArgument 和 CommandName 属性。例如将 CommandArgument 和 CommandName 属性设置为 Show,创建一个 Command 事件并在事件中编写相应代码,示例代码如下所示。

```
protected void Button1_Command(object sender, CommandEventArgs e)
{

```

```
        if (e.CommandName == "Show")
        {
            Labell1.Text=e.CommandArgument.ToString();
        }
    }
```

需要注意的是,当按钮同时包含 Click 和 Command 事件时,通常情况下会执行 Command 事件。Command 有一些 Click 不具备的好处,就是传递参数。可以对按钮的 CommandArgument 和 CommandName 属性分别设置,通过判断 CommandArgument 和 CommandName 属性来执行相应的方法。这样一个按钮控件就能够实现不同的方法,使得多个按钮与一个处理代码关联或者一个按钮根据不同的值进行不同的处理和响应。相比 Click 单击事件而言,Command 命令事件具有更高的可控性。

6. 单选控件和单选组控件(RadioButton 和 RadioButtonList)

在投票等系统中,通常需要使用单选控件和单选组控件。顾名思义,在单选控件和单选组控件的项目中,只能在有限种选择中进行一个项目的选择。在进行投票等应用开发并且只能在选项中选择单项时,单选控件和单选组控件都是最佳的选择。单选控件可以为用户选择某一个选项,单选控件常用属性如下所示。

- Checked: 控件是否被选中。
- GroupName: 单选控件所处的组名。
- TextAlign: 文本标签相对于控件的对齐方式。

单选控件通常需要 Checked 属性来判断某个选项是否被选中,多个单选控件之间可能存在着某些联系,这些联系通过 GroupName 进行约束和联系。与 TextBox 文本框控件相同的是,单选控件不会自动进行页面回传,必须将 AutoPostBack 属性设置为 True 时才能在焦点丢失时触发相应的 CheckedChanged 事件。

与单选控件相同,单选组控件也是只能选择一个项目的控件,而与单选控件不同的是,单选组控件没有 GroupName 属性,但是却能够列出多个单选项目。另外,单选组控件所生成的代码也比单选控件实现的相对较少。

单选组控件还包括一些属性,用于样式和重复的配置。单选组控件的常用属性如下。

- DataMember: 在数据集用作数据源时做数据绑定。
- DataSource: 向列表填入项时所使用的数据源。
- DataTextField: 提供项文本的数据源中的字段。
- DataTextFormat: 应用于文本字段的格式。
- DataValueField: 数据源中提供项值的字段。
- Items: 列表中项的集合。
- RepeatColumn: 用于布局项的列数。
- RepeatDirection: 项的布局方向。
- RepeatLayout: 确定是否在某个表或者流中重复。

与单选控件一样,双击单选组控件时系统会自动生成该事件的声明,同样可以在该事件中确定代码。当选择一项内容时,提示用户所选择的内容,示例代码如下所示。

```
protected void RadioButtonList1_SelectedIndexChanged (object sender,
```

```
EventArgs e)
{
    Label1.Text=RadioButtonList1.Text;
}
```

7. 复选框控件和复选组控件(CheckBox 和 CheckBoxList)

当一个投票系统需要用户能够选择多个选择项时,则单选框控件就不符合要求了。ASP.NET 还提供了复选框控件和复选组控件来满足多选的要求。复选框控件和复选组控件与单选框控件和单选组控件一样,都是通过 Checked 属性来判断是否被选择。而不同的是,复选框控件没有 GroupName 属性。对于复选框而言,用户可以在复选框控件中选择多个选项,所以就没有必要为复选框控件进行分组。在单选框控件中,相同组名的控件只能选择一项用于约束多个单选框中的选项,而复选框就没有约束的必要。

与单选组控件相同,为了方便复选控件的使用,.NET 服务器控件中同样包括了复选组控件,拖动一个复选组控件到页面可以同单选组控件一样添加复选组列表。复选组控件最常用的是 SelectedIndexChanged 事件。当控件中某项的选中状态被改变时,则会触发该事件。通常 CheckBoxList1.Items[i].Selected 是用来判断某项是否被选中,其中 Item 数组是复选组控件中项目的集合,其中 Items[i]是复选组中的第 i 个项目。

注意:复选组控件与单选组控件不同的是,不能够直接获取复选组控件某个选中项目的值,因为复选组控件返回的是第一个选择项的返回值,只能够通过 Item 集合来获取选择某个或多个选中的项目值。

8. 列表控件(DropDownList,ListBox 和 BulletedList)

在 Web 开发中,经常会需要使用列表控件,让用户的输入更加简单。例如在用户注册时,用户的所在地是有限的集合,而且用户不喜欢经常输入,这样就可以使用列表控件。同样,列表控件还能够简化用户输入并且防止用户输入在实际中不存在的数据,如性别的选择等。列表控件能在一个控件中为用户提供多个选项,同时又能够避免用户输入错误的选项。例如,在用户注册时,可以选择性别是“男”或者“女”可以使用 DropDownList 列表控件,同时又避免了用户输入其他的信息。因为性别除了“男”就是“女”,输入其他的信息说明这个信息是错误或者是无效的。下列语句声明了一个 DropDownList 列表控件,示例代码如下所示。

```
<asp:DropDownList ID="DropDownList1" runat="server">
    <asp:ListItem>1</asp:ListItem>
    <asp:ListItem>2</asp:ListItem>
    <asp:ListItem>3</asp:ListItem>
    <asp:ListItem>4</asp:ListItem>
    <asp:ListItem>5</asp:ListItem>
    <asp:ListItem>6</asp:ListItem>
    <asp:ListItem>7</asp:ListItem>
</asp:DropDownList>
```

上述代码创建了一个 DropDownList 列表控件,并手动增加了列表项。同时 DropDownList 列表控件也可以绑定数据源控件。DropDownList 列表控件最常用的事件是 SelectedIndexChanged,当 DropDownList 列表控件选择项发生变化时,则会触发该事件。当用户选择相应的项目时,就会触发 SelectedIndexChanged 事件,开发人员可以通过捕捉相应的用户选中的控件进行编程处理,这里就捕捉了用户选择的数字进行字体大小的更改。

 **示例 1:** 宁波城市学院对学生就业满意度进行调查,需要开发调查问卷页面(如图 3-2 所示)。通过对该图所示效果图分析,发现该界面主要由文本框、下拉列表、单选列表、多选列表等控件设计而成。

宁波城市学院毕业生跟踪调查问卷
(毕业生填写)

姓 名:		性 别:	☒ 女 ☐ 男	学 号:	
所学专业:	计算机信息管理	毕业时间:		联系方式:	电话
工作单位:		所从事的岗位:			

请各位同学为了母校的发展,配合进行这项工作,以下题目没有特殊说明的皆为单选题,请认真填写,非常感谢!

- 您目前的行政职务是什么?
☐ 单位负责人 ☐ 单位中层领导 ☐ 单位科室负责人 ☐ 普通员工
- 您对目前的工作情况感到满意吗?
☐ 很满意 ☐ 较满意 ☐ 有个工作不容易,凑合着 ☐ 不满意,还想调换
- 毕业参加工作后,您在工作中的表现如何?
☐ 得到晋升,受到重用 ☐ 收到表彰和好评 ☐ 表现一般 ☐ 不适合目前的工作
- 您认为自己是能否胜任目前的工作?
☐ 能胜任 ☐ 基本能胜任 ☐ 不能胜任
- 您认为自己能适应目前的工作的最重要原因是?
☐ 专业对口 ☐ 实际工作能力强 ☐ 人际关系好 ☐ 工作态度端正
- 您目前的工作和所学专业是否对口?
☐ 对口 ☐ 基本对口 ☐ 不对口
- 择业时,您考虑过自己专业和岗位是否对口的问题吗?
☐ 考虑过,要与本专业对口 ☐ 考虑过,但不对口也没关系 ☐ 没考虑过
- 毕业后,您调换过几次单位?
☐ 没有调换 ☐ 一次 ☐ 两次 ☐ 三次 ☐ 三次以上
- 您调换工作的原因是?
☐ 专业不对口 ☐ 工作强度太大 ☐ 与同事关系不融洽 ☐ 单位裁员或倒闭 ☐ 领导不重视 ☐ 经济待遇不好
- 您认为在学校所学专业与实际工作中的联系程度紧密吗?
☐ 很紧密 ☐ 较紧密 ☐ 不紧密 ☐ 很不紧密
- 工作中是否遇到因专业知识不精通而不能胜任工作的问题?
☐ 有 ☐ 很少 ☐ 没有
- 您认为自己所学的专业在企业单位中的认可度怎样?
☐ 十分认可 ☐ 比较认可 ☐ 不认可
- 您如何看待自己的工作发展前景?
☐ 十分乐观,有所作为 ☐ 比较看好 ☐ 悲观,没有前景 ☐ 没有考虑过
- 您觉得现在最需要加强哪方面知识的充实?
☐ 人际关系 ☐ 专业知识 ☐ 外语 ☐ 专业技能 ☐ 其他
- 在校期间,哪些教学环节对您现在的工作帮助最大?
☐ 专业理论课 ☐ 公共基础课 ☐ 实验操作学习 ☐ 职业生涯规划 ☐ 课外学术交流活动 ☐ 其他
- 工作后,您对母校学习期间所学课程的体会是?
☐ 学得不够深和不够宽广 ☐ 学得不够活,能力培养不足 ☐ 学得很多知识,对现在工作帮助不大 ☐ 学的知识对现在工作帮助很大
- 您认为自己在在校期间取得的技能证书在工作中起到的作用怎样?
☐ 帮助很大 ☐ 基本没用,不代表个人实际能力 ☐ 不被认可
- 与其他高校毕业的同学比较起来,您认为你从母校学习到的哪方面能力较强?
☐ 思想政治素质 ☐ 专业基础知识 ☐ 实际工作能力 ☐ 学习能力 ☐ 创新能力 ☐ 人际关系处理能力
- 您下一步的打算是什么?
☐ 进一步学习,提高自己学历,为长远打算 ☐ 继续做下去,在实践中提高自己 ☐ 走一步是一步,没明确目标 ☐ 跳槽寻找更好的
- 您如何评价母校对您的培养工作?
☐ 很满意 ☐ 基本满意 ☐ 不太满意 ☐ 不满意
- 结合实际工作经历,您对学校有何建议(教学、管理、就业指导等)?

图 3-2 调查问卷效果图

① 拖动 HTML 标签中的 Table 控件到页面,右击 Table 控件,选择“插入”→“右侧列”命令来添加列。

② 按照图 3-2 所示布局,拖动 TextBox 控件、DropDownList 控件到页面合适位置。

③ 单击“所学专业”下拉列表智能提示标识,单击“编辑项”菜单项,打开“集合编辑器”对话框(如图 3-3 所示)。



图 3-3 “集合编辑器”对话框

④ 单击“添加”按钮,添加项,修改 Text 属性和 Value 属性(见图 3-3)。

⑤ “性别下拉”列表框和“联系方式”下拉列表的设置同步骤④。

⑥ 拖动 RadioButtonList 到页面,单击智能提示按钮,单击“编辑项”菜单项,打开“集合编辑器”对话框,如同步骤④一样设置显示文本和显示值。接着设置 RadioButtonList 按钮的 RepeatDirection 属性值为 Horizontal,让其水平显示。其中的 Q9 多选列表设置也同 RadioButtonList 设置方法一样。

⑦ 拖动 TextBox 控件到 Q21 下面,设置 TextMode 属性为 Multiline,允许多行输入。

⑧ 拖动 Button 控件到页面,设置其 Text 属性为“按钮”。

3.3.2 ASP.NET 输入验证控件

ASP.NET 提供了强大的验证控件,它可以验证服务器控件中用户的输入,并在验证失败的情况下显示一条自定义错误消息。验证控件直接在客户端执行,用户提交后执行相应的验证,无需使用服务器端进行验证操作,从而减少了服务器与客户端之间的往返过程。

1. 必填验证控件(RequiredFieldValidator)

在实际的应用中,如在用户填写表单时,有一些项目是必填项,例如用户名和密码。在传统的 ASP 中,当用户填写表单后,页面需要发送到服务器并判断表单中的某项 HTML 控件的值是否为空,如果为空,则返回错误信息。在 ASP.NET 中,系统提供了 RequiredFieldValidator 验证控件进行验证。使用 RequiredFieldValidator 控件能够指定某个用户在特定的控件中必须提供相应的信息。如果不填写相应的信息,RequiredFieldValidator 控件就会提示错误信息。RequiredFieldValidator 控件示例代码如下。

```
<body>
  <form id="form1" runat="server">
    <div>
      姓名:<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
      <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
        runat="server"
        ControlToValidate="TextBox1"
        ErrorMessage="必填字段不能为空"></asp:
        RequiredFieldValidator>

      <br />
      密码:<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
      <br />
      <asp:Button ID="Button1" runat="server" Text="Button" />
      <br />
    </div>
  </form>
</body>
```

在进行验证时,RequiredFieldValidator 控件必须绑定一个服务器控件,在上述代码中,验证控件 RequiredFieldValidator 的服务器控件绑定为 TextBox1,当 TextBox1 中的值为空时,则会提示自定义错误信息为“必填字段不能为空”。

2. 比较验证控件(CompareValidator)

比较验证控件对照特定的数据类型来验证用户的输入。因为当用户输入信息时,难免会输入错误信息,比如当需要了解用户的生日时,很可能输入了其他的字符串。CompareValidator 比较验证控件能够比较控件中的值是否符合开发人员的需要。CompareValidator 控件的特有属性如下。

- ControlToCompare: 以字符串形式输入的表达式。要与另一控件的值进行比较。
- Operator: 要使用的比较。
- Type: 比较两个值的数据类型。
- ValueToCompare: 以字符串形式输入的表达式。

当使用 CompareValidator 控件时,可以方便地判断用户是否正确输入了相关值,示例代码如下。

```
<body>
  <form id="form1" runat="server">
    <div>
      请输入生日:
      <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
      <br />
      毕业日期:
      <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
      <asp:CompareValidator ID="CompareValidator1" runat="server"
        ControlToCompare="TextBox2" ControlToValidate="TextBox1"
        CultureInvariantValues="True" ErrorMessage="输入格式错误,请改正!"
        Operator="GreaterThan"
        Type="Date">
      </asp:CompareValidator>
```