常用Web服务器控件

1. Label控件
2. TextBox控件
3. HyperLink控件
4. Image控件
5. Button控件
   - LinkButton控件
   - ImageButton控件
6. DropDownList控件
   - ListBox控件
7. CheckBox控件
   - CheckBoxList控件
8. RadioButton控件
   - RadioButtonList控件
9. Panel控件
10. MultiView控件
    - View控件
11. Calender控件

第4章
ASP.NET服务器控件

内容

- 服务器控件概述
- HTML服务器控件
- Web服务器控件
- 验证控件
- 用户控件

4.4验证控件

- 验证控件概述
- 验证控件的使用
- 验证组的使用
- 禁用验证
- 以编程方式测试验证有效性

Part 1

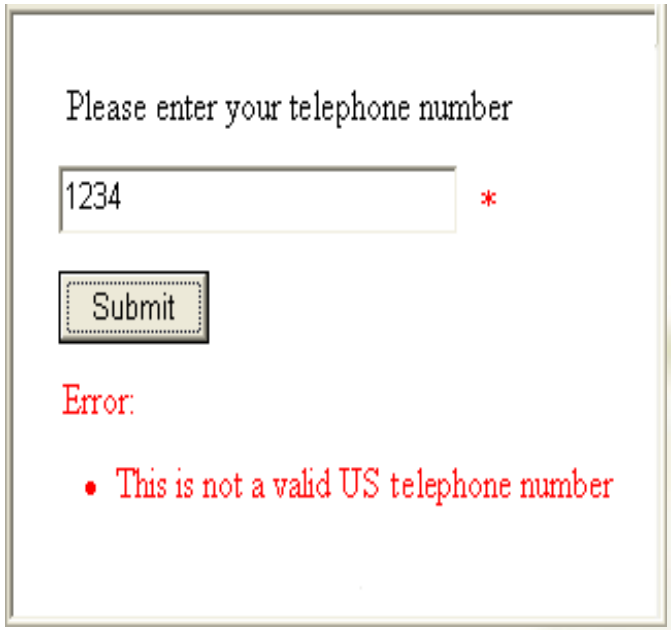
验证控件概述

4.4.1 验证控件概述

- 为什么要验证用户输入
- 验证过程

为什么要验证用户输入

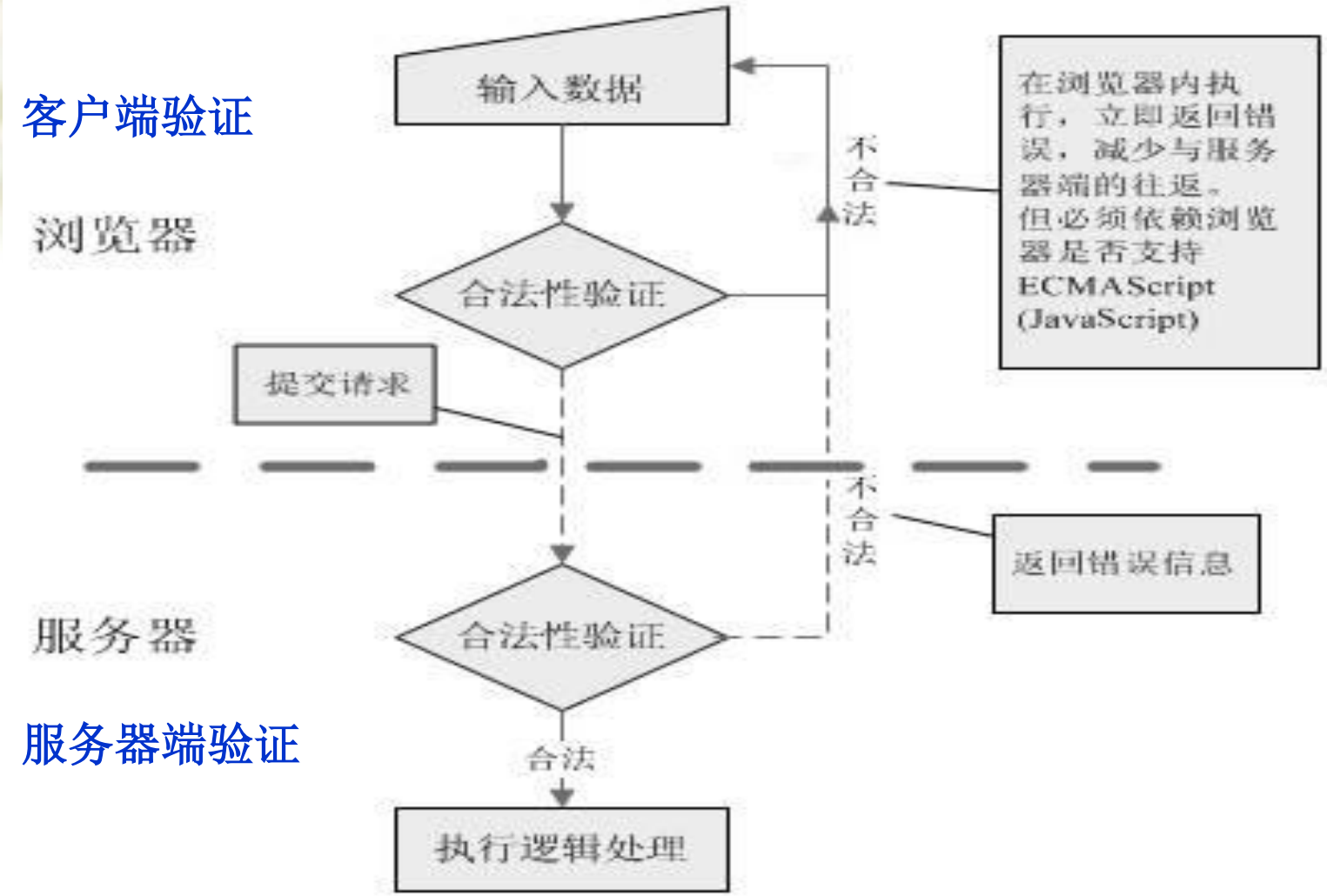
- 输入验证是检验Web窗体中用户的输入是否和期望的数据值、范围或格式相匹配的过程，可以减少等待错误信息的时间降低发生错误的可能性，从而改善用户访问Web站点的体验



- 用户输入验证的目的
  - 减少错误处理的等待时间
  - 避免非法的用户输入导致的错误结果
  - 避免非法的用户输入导致服务器崩溃
  - 避免欺骗或恶意代码
  - 阻止 Web 窗体进行下一步处理，直到所有的用户输入都通过验证

验证过程

客户端验证

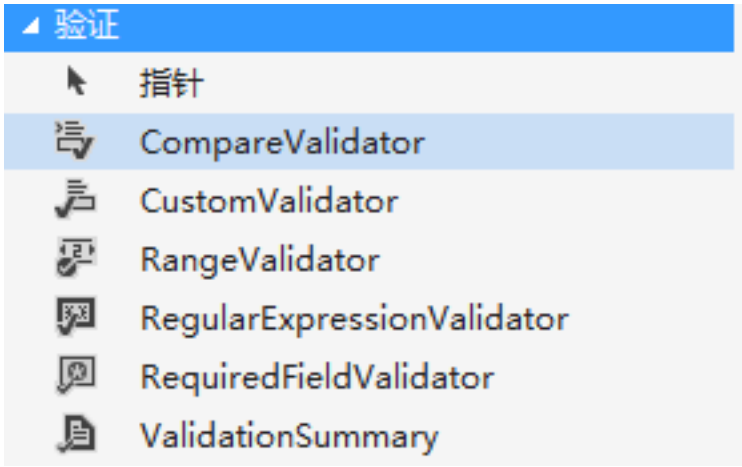


服务器端验证

客户端验证和服务器端验证的区别

客户端验证	服务器端验证
依赖于客户端浏览器版本	与客户端浏览器版本无关
使用Javascript和vbscript实现	使用基于.NET的开发语言实现
即时信息反馈	需要服务器往返以显示错误信息
不能访问服务器资源	可与服务器上存储的数据进行比较验证，如与数据库密码进行比较
不能避免欺骗代码或恶意代码	可以避免欺骗代码和恶意代码
允许禁用客户端验证	必然执行，重复所有客户端验证
安全性较低	安全性较高

ASP.NET验证控件



验证类型	使用的控件	说　明
必需项	**RequiredFieldValidator**	验证一个必填字段，确保用户不会跳过某项输入。
与某值的比较	**CompareValidator**	将用户输入与一个常数值或者另一个控件或特定数据类型的值进行比较（使用小于、等于或大于比较运算符）。
范围检查	**RangeValidator**	用于检查用户的输入是否在指定的上下限内。可以检查数字对、字母对和日期对的限定范围。

模式匹配	**RegularExpressionValidator**	用于检查输入的内容与正则表达式所定义的模式是否匹配。此类验证可用于检查可预测的字符序列，例如电子邮件地址、电话号码、邮政编码等内容中的字符序列。
用户定义	**CustomValidator**	使用自己编写的验证逻辑检查用户输入。此类验证能够检查在运行时派生的值。
验证总汇	**ValidationSummary**	该控件不执行验证，但经常与其它验证控件一起用于显示来自网页上所有验证控件的错误信息。

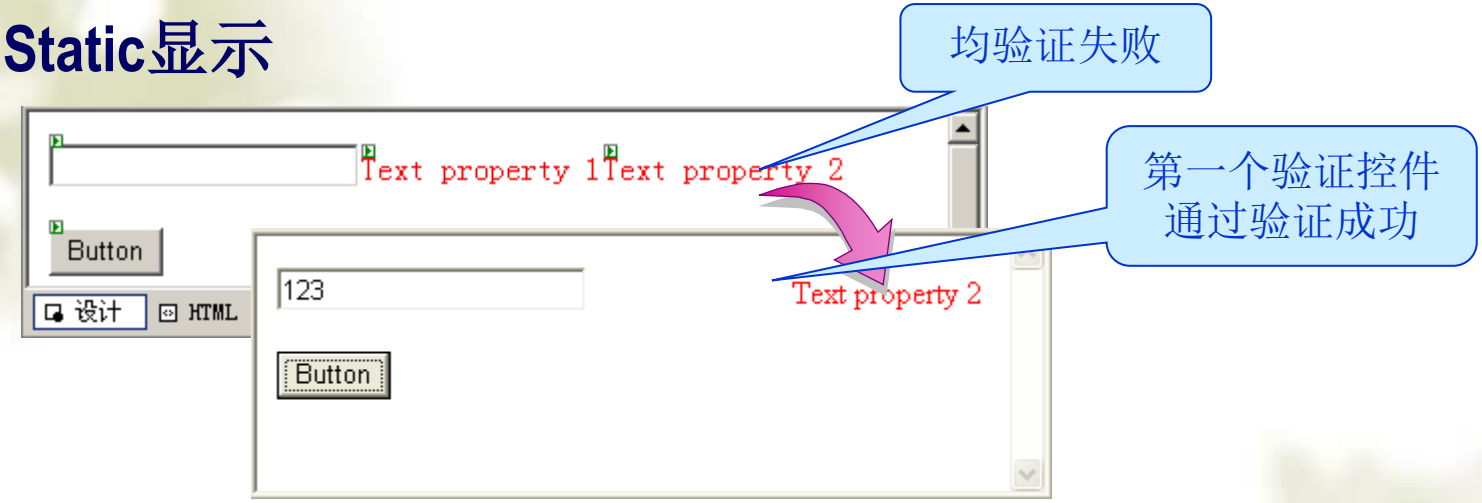
验证控件的共有属性

属性	说 明
Display	获取或设置验证控件中错误信息的显示行为
ErrorMessage	获取或设置验证失败时 ValidationSummary 控件中显示的错误信息的文本
Text	获取或设置验证失败时验证控件中显示的文本
ControlToValidate	获取或设置要验证的输入控件
EnableClientScript	获取或设置一个值，该值指示是否启用客户端验证
SetFocusOnError	获取或设置一个值，该值指示在验证失败时是否将焦点设置到 ControlToValidate 属性指定的控件上
ValidationGroup	获取或设置此验证控件所属的验证组的名称
IsValid	获取或设置一个值，该值指示关联的输入控件是否通过验证

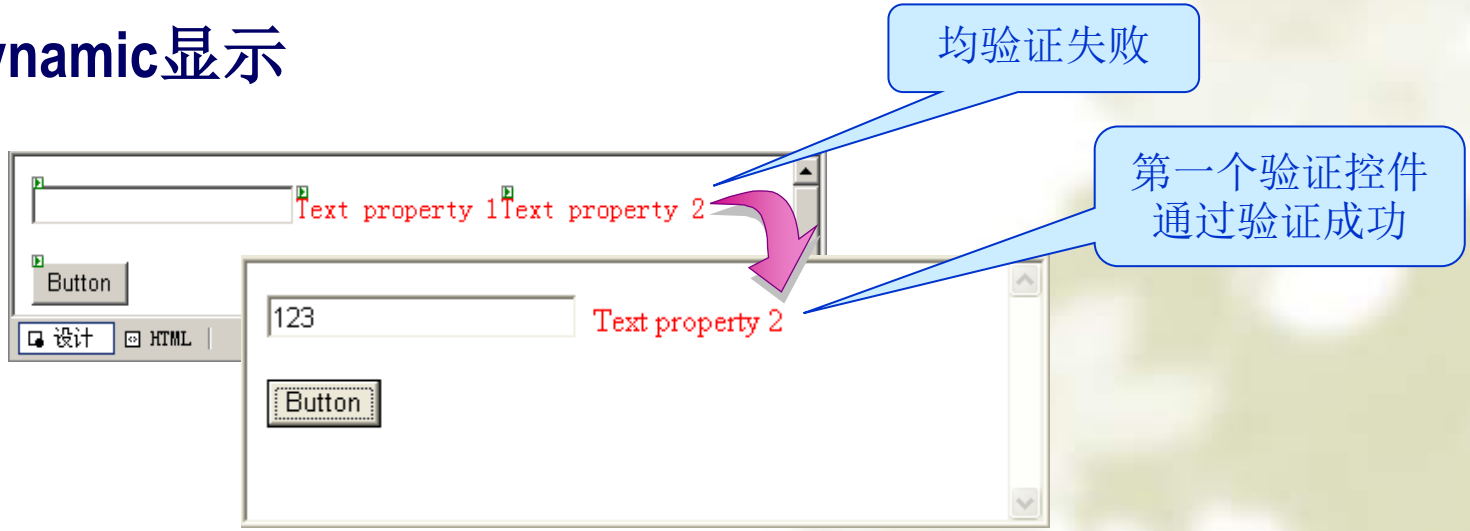
- **Display属性**：流布局模式下，设置多个验证控件的错误信息的空间排列方式。

Display	显示行为
None	在验证控件的位置上不显示错误信息
Static	在页面布局中分配用于显示验证消息的空间
Dynamic	如果验证成功，将不占据页面空间

Static显示



Dynamic显示



ErrorMessage 属性

- 指定验证失败后在验证控件中显示的文本
- 如果设置了 **Text** 属性，不显示 **ErrorMessage**

Text 属性

- 指定将在验证控件中显示的文本

ErrorMessage	Text	验证未通过的结果
已设置	未设置	▪ 显示 ErrorMessage 文本
已设置	已设置	▪ 显示 Text 文本
未设置	已设置	▪ 显示 Text 文本

IsValid属性

- 每个验证控件公开**IsValid**属性，是否通过验证
- 页面公开**IsValid**属性，总结页面上所有验证控件

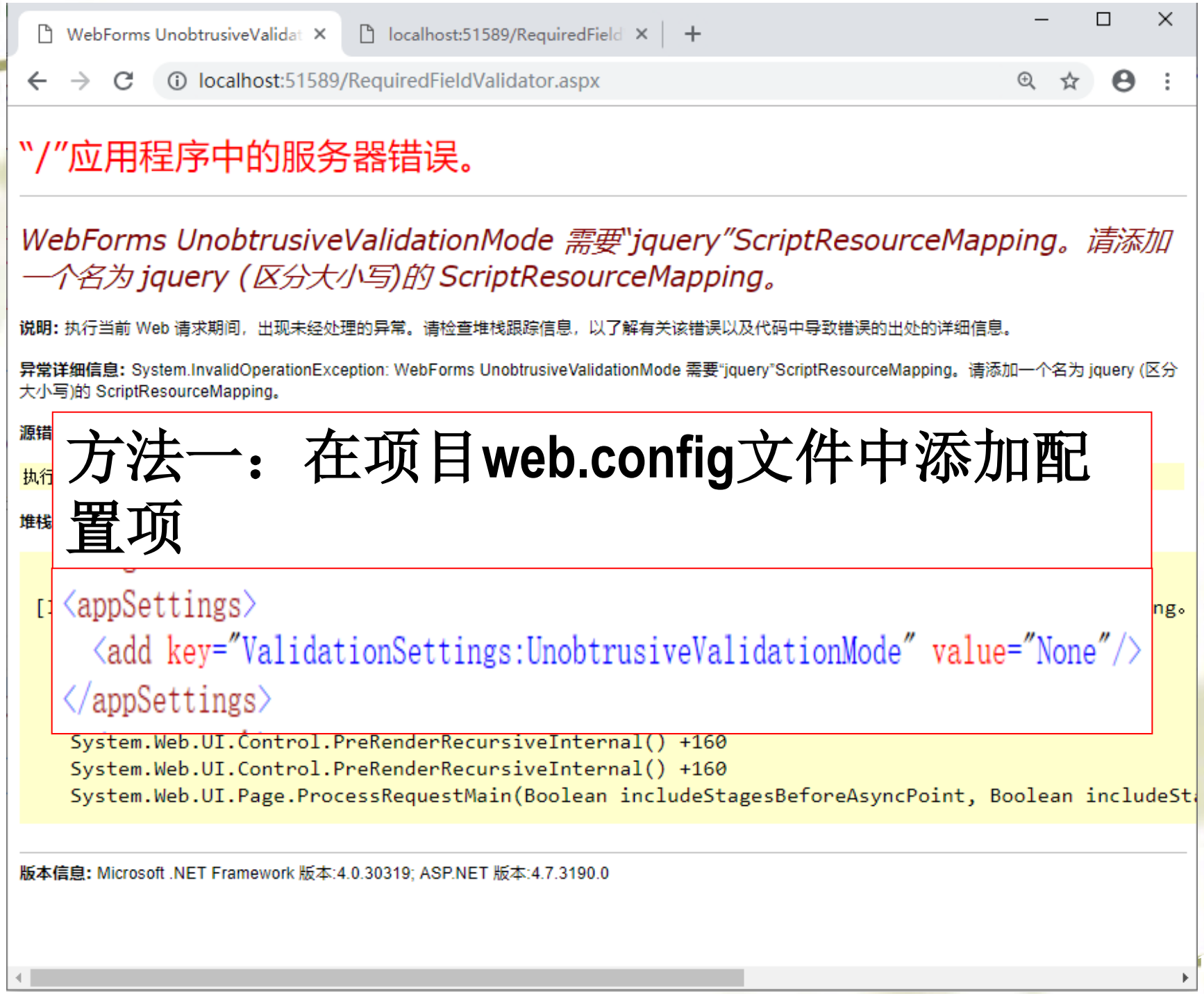
Part 2

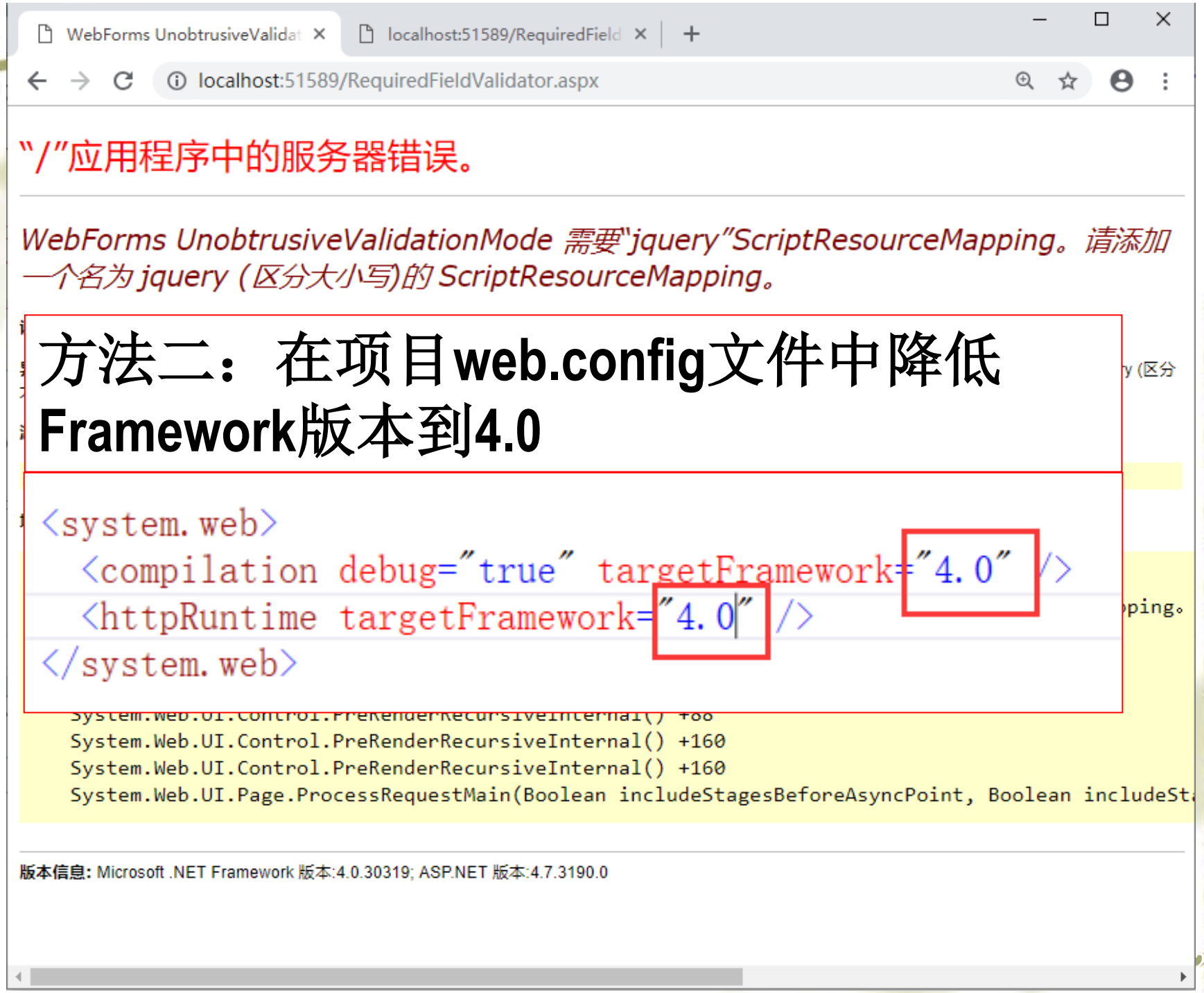
验证控件的使用

1 RequiredFieldValidator控件

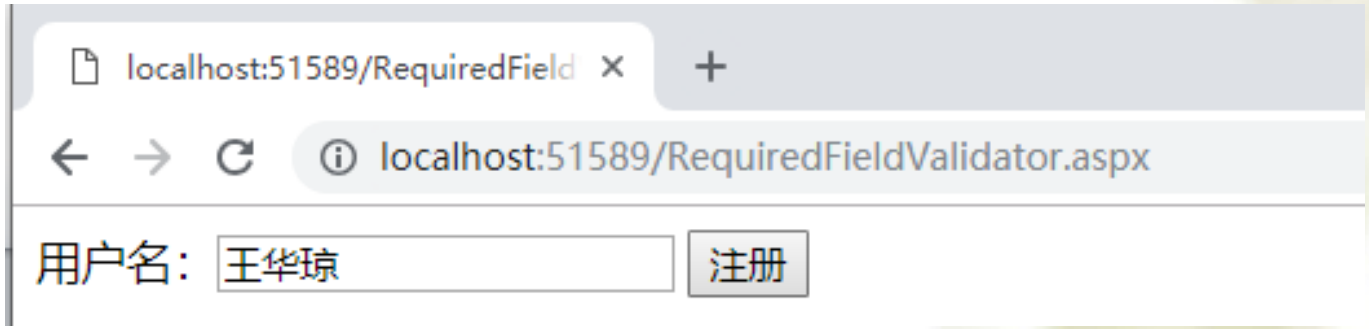
- 在页中添加 **RequiredFieldValidator** 控件并将其链接到必需的控件，可以指定用户在 **ASP.NET** 网页上的特定控件中必须输入信息

```
<div>
用户名：<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
        ControlToValidate="TextBox1" Display="Dynamic" ErrorMessage="不允许为空">
    </asp:RequiredFieldValidator>
    <asp:Button ID="Button1" runat="server" Text="注册" />
</div>
```


WebForms UnobtrusiveValidat
localhost:51589/RequiredField
localhost:51589/RequiredFieldValidator.aspx
"/"应用程序中的服务器错误。
WebForms UnobtrusiveValidationMode 需要"jquery"ScriptResourceMapping。请添加一个名为 jquery (区分大小写)的 ScriptResourceMapping。
说明: 执行当前 Web 请求期间，出现未经处理的异常。请检查堆栈跟踪信息，以了解有关该错误以及代码中导致错误的出处的详细信息。
异常详细信息: System.InvalidOperationException: WebForms UnobtrusiveValidationMode 需要"jquery"ScriptResourceMapping。请添加一个名为 jquery (区分大小写)的 ScriptResourceMapping。
源错
执行
堆栈
方法一：在项目web.config文件中添加配置项
<appSettings>
<add key="ValidationSettings:UnobtrusiveValidationMode" value="None"/>
</appSettings>
System.Web.UI.Control.PreRenderRecursiveInternal() +160
System.Web.UI.Control.PreRenderRecursiveInternal() +160
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeSt
版本信息: Microsoft .NET Framework 版本:4.0.30319; ASP.NET 版本:4.7.3190.0


WebForms UnobtrusiveValidat
localhost:51589/RequiredField
localhost:51589/RequiredFieldValidator.aspx
"/"应用程序中的服务器错误。
WebForms UnobtrusiveValidationMode 需要"jquery"ScriptResourceMapping。请添加一个名为 jquery (区分大小写)的 ScriptResourceMapping。
方法二：在项目web.config文件中降低Framework版本到4.0
<system.web>
<compilation debug="true" targetFramework="4.0" />
<httpRuntime targetFramework="4.0" />
</system.web>
System.Web.UI.Control.PreRenderRecursiveInternal() +160
System.Web.UI.Control.PreRenderRecursiveInternal() +160
System.Web.UI.Page.ProcessRequestMain(Boolean includeStagesBeforeAsyncPoint, Boolean includeSt
版本信息: Microsoft .NET Framework 版本:4.0.30319; ASP.NET 版本:4.7.3190.0


localhost:51589/RequiredField
localhost:51589/RequiredFieldValidator.aspx
用户名：
不允许为空
注册

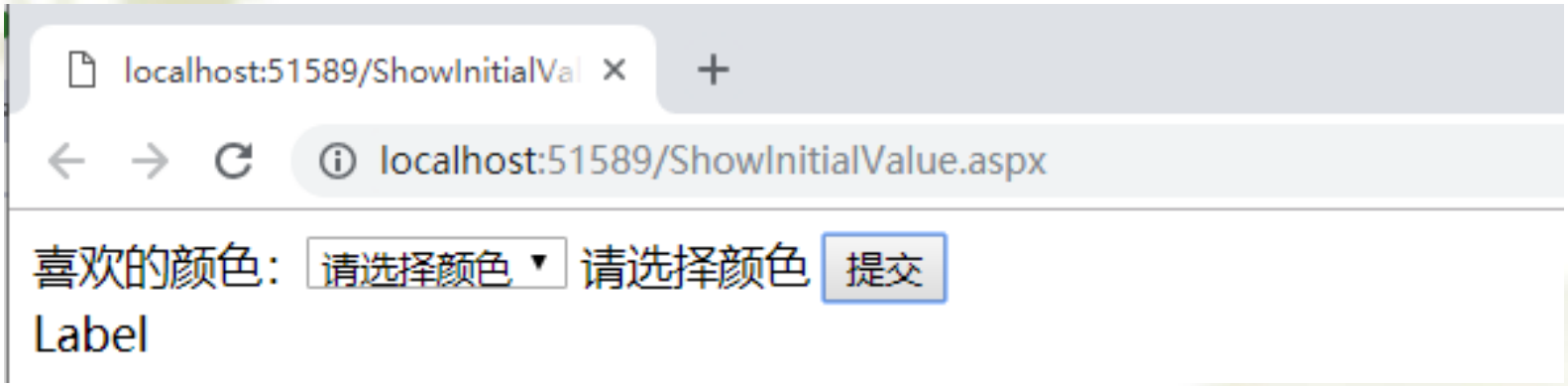

localhost:51589/RequiredField
localhost:51589/RequiredFieldValidator.aspx
用户名：
王华琼
注册

- **InitialValue属性**：当设置该值时，表示控件中输入该值时无效，其他任何情况均有效。

【例4-22】演示如何使用RequiredFieldValidator控件来验证DropDownList控件的输入。


```
<div>
喜欢的颜色：<asp:DropDownList ID="DropDownList1" runat="server">
        <asp:ListItem Value="0">请选择颜色</asp:ListItem>
        <asp:ListItem Value="1">红色</asp:ListItem>
        <asp:ListItem Value="2">蓝色</asp:ListItem>
        <asp:ListItem Value="3">绿色</asp:ListItem>
    </asp:DropDownList>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
        Text="请选择颜色" ControlToValidate="DropDownList1" InitialValue="0">
    </asp:RequiredFieldValidator>
    <asp:Button ID="Button1" runat="server" Text="提交" />
    <br />
    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div>
```

```
protected void Button1_Click(object sender, EventArgs e)
{
    if (Page.IsValid) Label1.Text = DropDownList1.SelectedItem.Text;
}
```



2 CompareValidator控件

测试用户的输入是否符合指定的类型或者符合另一个输入控件的值。空输入作为有效验证。

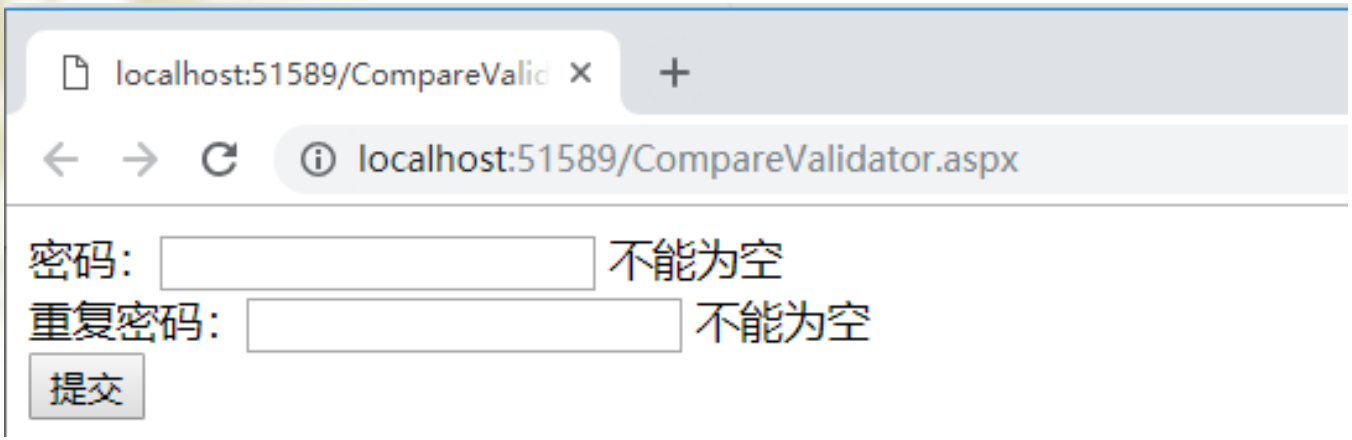
- **ControlToCompare**：确定要比较的另一个控件。
- **Operator**：指定使用的比较运算符。
- **Type**：指定数据类型，希望输入控件中的值与某个数据类型匹配。
- **ValueToCompare**：确定要比较的某个常数值

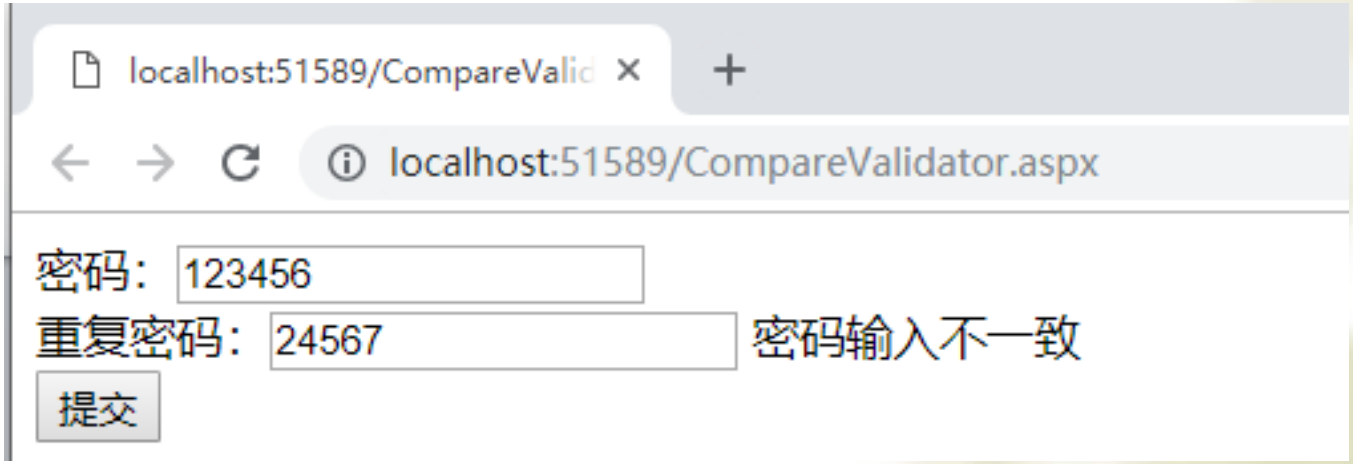
【例4-23】在设计用户注册页面时，希望用户输入两次密码，使用CompareValidator验证控件来判断两次输入的密码是否相等。

```
<div>
密码：<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
        ControlToValidate="TextBox1" ErrorMessage="不能为空" />
    <br />
重复密码：<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server"
        ControlToValidate="TextBox2" Display="Dynamic" ErrorMessage="不能为空" />
    <asp:CompareValidator ID="CompareValidator1" runat="server"
        ControlToValidate="TextBox2" ControlToCompare="TextBox1"
        Display="Dynamic" ErrorMessage="密码输入不一致" />
    <br />
    <asp:Button ID="Button1" runat="server" Text="提交" />
</div>
```

组合使用验证控件

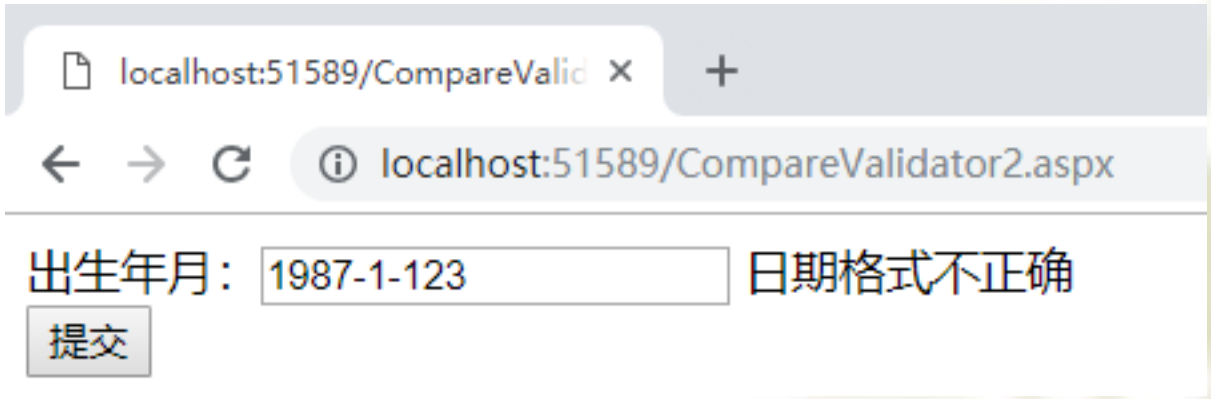
CompareValidator.aspx 25


localhost:51589/CompareValic
localhost:51589/CompareValidator.aspx
密码：
不能为空
重复密码：
不能为空
提交


localhost:51589/CompareValic
localhost:51589/CompareValidator.aspx
密码：
123456
重复密码：
24567
密码输入不一致
提交

【例4-24】使用CompareValidator控件对数据进行类型检查。要求用户输入合法的日期，否则验证失败。

```
<div>
出生年月：<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:CompareValidator ID="CompareValidator1" runat="server"
        ControlToValidate="TextBox1" Display="Dynamic"
        ErrorMessage="日期格式不正确"
        Operator="DataTypeCheck" Type="Date"></asp:CompareValidator>
    <br /><asp:Button ID="Button1" runat="server" Text="提交" />
</div>
```



思考

- 使用CompareValidator控件比较大小，要求输入框中填写大于0的整数，如何设置验证控件的属性？

3 RangeValidator控件

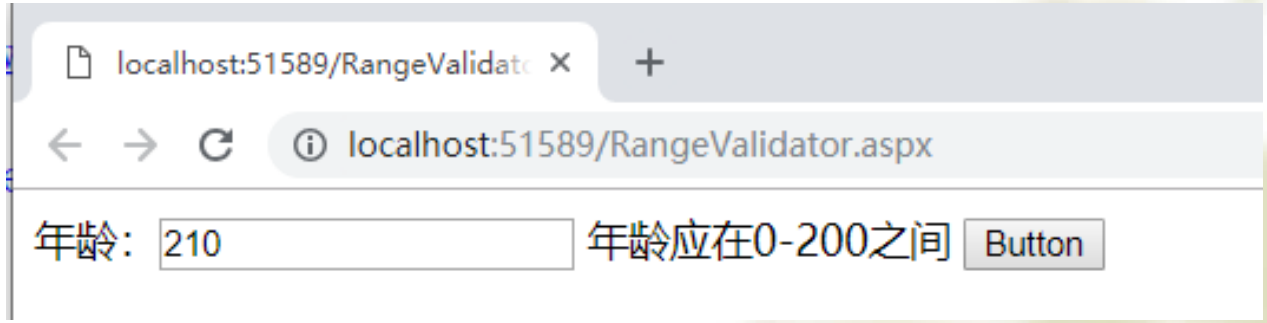
RangeValidator控件：

- 验证输入值是否在给定的范围内。
- 输入值介于最小值和最大值之间是有效的。
- 空输入作为有效验证。

- MinimumValue：指定有效范围的最小值。
- MaxmumValue：指定有效范围的最大值。
- Type：指定要比较的值的数据类型。

【例4-25】演示如何通过RangeValidator控件验证文本框中的年龄输入在0~200之间。

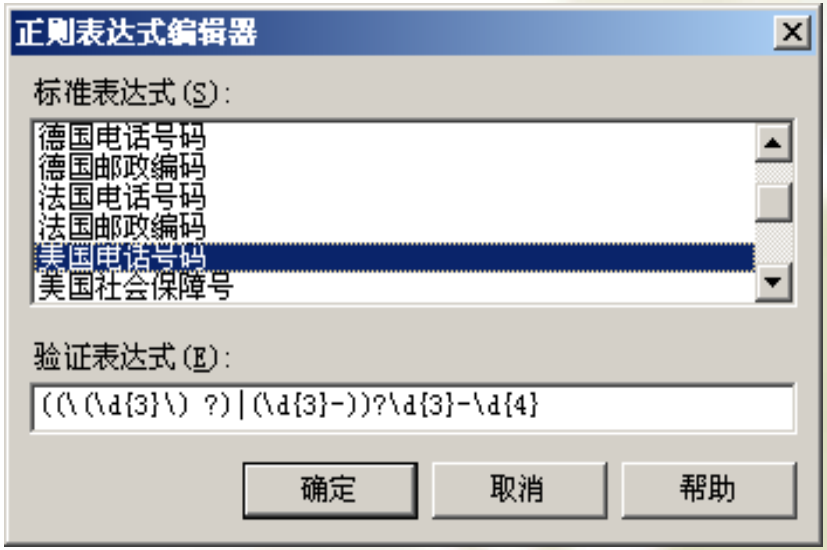
```
<div>
年龄：<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:RangeValidator ID="RangeValidator1" runat="server"
        ControlToValidate="TextBox1" Display="Dynamic"
        ErrorMessage="年龄应在0-200之间"
        MaximumValue="200" MinimumValue="0" Type="Integer">
    </asp:RangeValidator>
    <asp:Button ID="Button1" runat="server" Text="Button" />
</div>
```



RangeValidator.aspx

4. RegularExpressionValidator控件

- 在需要确定该值是否与某个正则表达式所定义的模式相匹配的情况下使用
- **ValidationExpression属性：编辑正则表达式**
- **Visual Studio.NET提供以下预定义模式**
  - 电话号码
  - 邮政编码
  - **E-mail 地址**



注意：提供的这些预定义的模式的正则表达式未必正确。

正则表达式字符含意

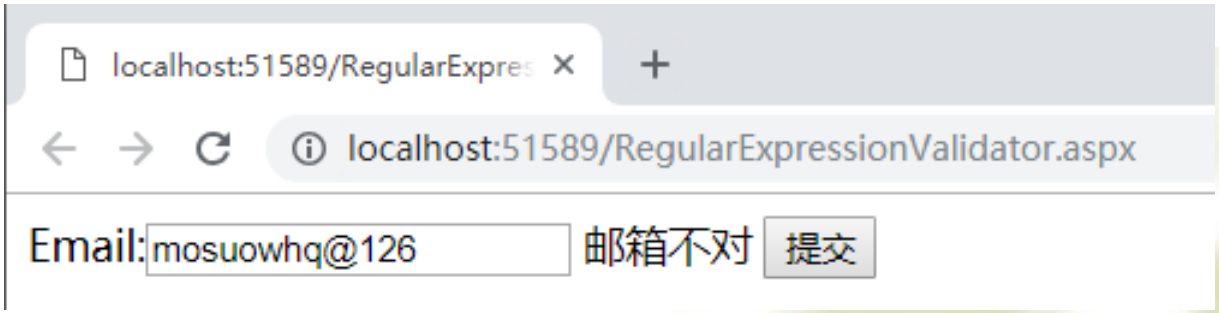
字　符	定　义
?	零次或一次匹配前面的字符或子表达式
*	零次或多次匹配前面的字符或子表达式
+	一次或多次匹配前面的字符或子表达式
.	匹配任意一个字符
\w	匹配任意一个字符
\d	匹配任意一个数字字符
\.	匹配一个点字符
{n,m}	长度是n到m的字符串,必须与\d或\S合用
[0-n]	零到n之间的整数值
|	分隔多个有效的模式

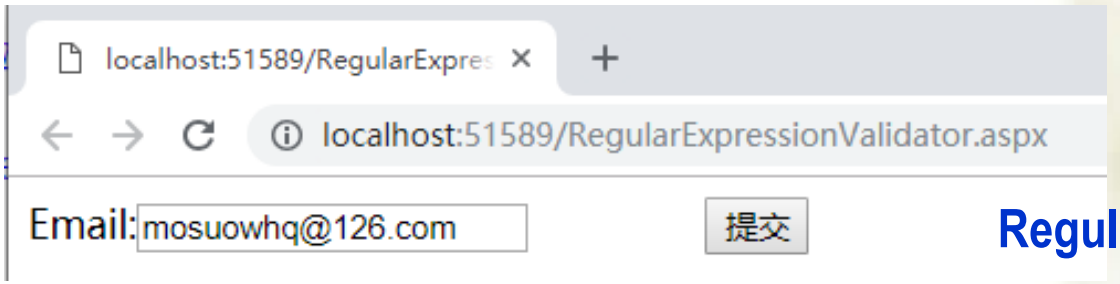
例如：

\S{3,6}	3到6位字符
[A-Za-z]{2,5}	由2-5个字母组成
[A-Z]\d{5}	以一个大写字母开头，加5位数字
\d{5}	5位的整数
.*[@#&].*	至少包含@#&中的一个字符
(\d{11})|((\d{3,4}-)?\d{7,8})	中国电话号码（手机或固定电话）

【例4-26】演示如何使用预定义表达式来验证输入的电子邮件地址。

```
<div>
Email:<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:RegularExpressionValidator ID="RegularExpressionValidator1" runat="server"
        ControlToValidate="TextBox1" ErrorMessage="邮箱不对"
        ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*">
    </asp:RegularExpressionValidator>
    <asp:Button ID="Button1" runat="server" Text="提交" />
</div>
```





RegularExpressionValidator.aspx

【例4-27】演示如何使用自定义表达式验证输入。

如果要求用户输入一个以大写字母开头，再加5位阿拉伯数字的格式化数据。很显然，预定义表达式里没有这样的格式定义只能通过正则表达式进行自定义。按照上面的格式要求，该正则表达式应该为“[A-Z]\d{5}”。

5. CustomValidator 控件

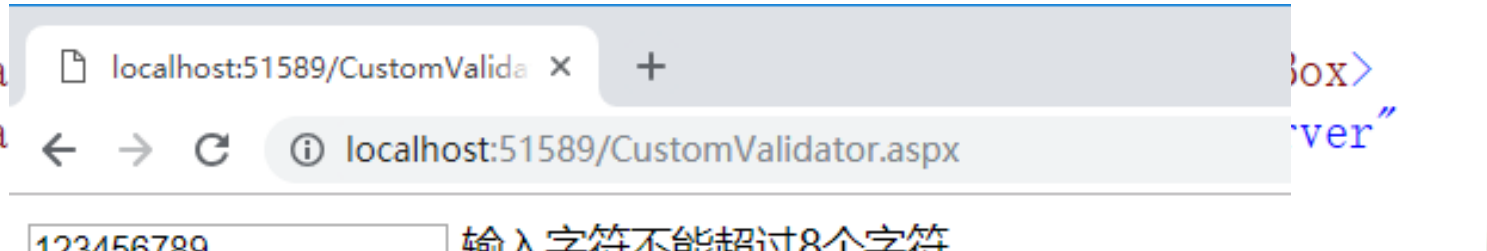
- 自定义验证，必须手动编写客户端和服务器端验证代码

```
<asp: CustomValidator□
  ClientValidationFunction = “客户端验证函数名"/>
```

- **ClientValidationFunction 属性**
  - 用于验证的自定义客户端脚本函数的名称
- **ServerValidate 事件**
  - 服务器端验证事件的名称
- 在验证函数中通过属性**Args.IsValid**来设置**CustomValidator**控件是否有效。

【例4-28】下面将编写一个CustomValidator控件的验证函数用来确定TextBox控件中用户输入是否超过8个字符。

```
<div>
    <a
    <a
        <asp:CustomValidator>
        <br /><asp:Button ID="Button1" runat="server" Text="确定" />
</div>
```




```
protected void CustomValidator1_ServerValidate(object source, ServerValidateEventArgs args)
{
    if (args.Value.Length > 8) args.IsValid = false;
    else args.IsValid = true;
}
```

CostomValidator.aspx

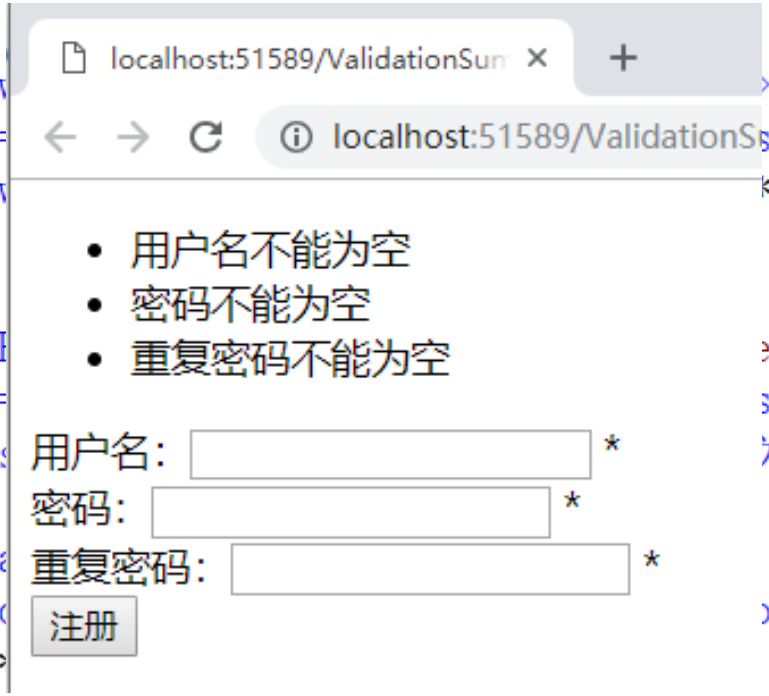
- **CustomValidator 灵活性高，可以**
  - 与给定的公式比较
  - 与存储在服务器中的数据比较，如存储在数据库中的密码
  - 通过调用 COM 对象比较
  - 通过调用 Web 服务比较

6. ValidationSummary 控件

- Page.IsValid 返回 False 时显示错误信息
- 在一个位置上汇总 Web 页上所有验证控件的错误信息
- HeaderText属性：设置标题
- DisplayMode 属性，有3个值可选：
  - BulletList：项目符号列表显示
  - List：无符号列表显示
  - SingleParagraph：单个段落显示
- ShowMessageBox 属性：是否弹出一个错误列表的对话框
- 使用 ValiationSummary 控件时，其他验证控件通常使用 Text=“*” 指示错误出现的位置

【例4-28】演示如何使用ValidationSummary控件显示错误信息摘要。

```
<div>
    <asp:ValidationSummary ID="ValidationSummary1" runat="server" />
用户名：<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"
        ControlToValidate="TextBox1" ErrorMessage="用户名不能为空">*
    </asp:RequiredFieldValidator>
     <br />
    密码：<asp:TextBox ID="txtPassw
    <asp:RequiredFieldValidator ID=
        ControlToValidate="txtPassw
    </asp:RequiredFieldValidator>
    <br />
    重复密码：<asp:TextBox ID="txtR
    <asp:RequiredFieldValidator ID=
        ControlToValidate="txtRePas
    </asp:RequiredFieldValidator>
    <asp:CompareValidator ID="Compa
        ControlToCompare="txtPasswo
        ErrorMessage="密码不一致">*
    <br />
    <asp:Button ID="btnRegister" runat="server" Text="注册" />
</div>
```



Part 3

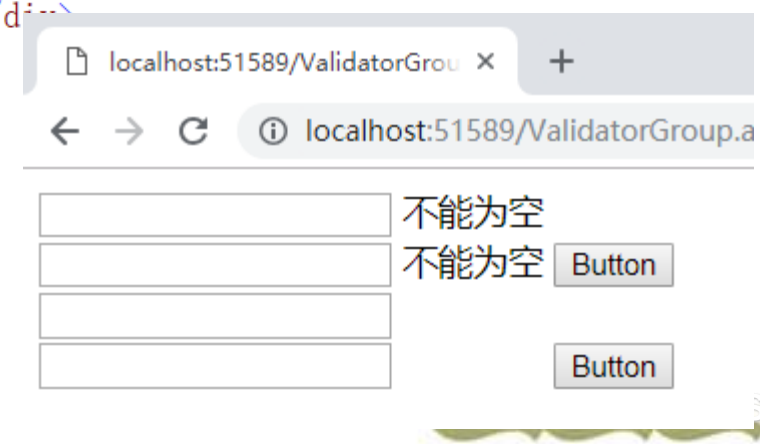
验证组的使用

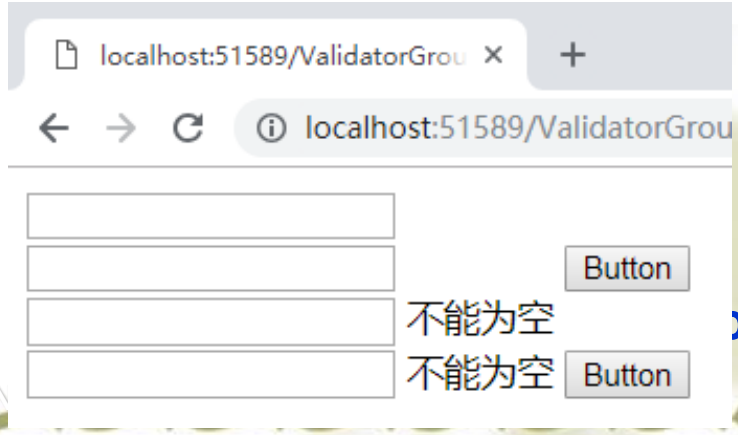
4.4.3 使用验证组

- 在页面上控件比较多时，可以将不同的控件归为一组，**ASP.NET**在对每个验证组进行验证时，与同页的其他组无关
- 通过将要分在同一组的所有控件的**ValidationGroup** 属性设置为同一个名称（字符串）即可创建验证组

【例4-29】演示在Button控件回发到服务器时，如何使用ValidationGroup属性指定要验证的控件。

```
<div>
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server" ControlToValidate="TextBox1"
        ErrorMessage="不能为空" ValidationGroup="Button1Group"></asp:RequiredFieldValidator><br />
    <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator2" runat="server" ControlToValidate="TextBox2"
        ErrorMessage="不能为空" ValidationGroup="Button1Group"></asp:RequiredFieldValidator>
    <asp:Button ID="Button1" runat="server" Text="Button"
        ValidationGroup="Button1Group" onclick="Button1_Click" /><br />
    <asp:TextBox ID="TextBox3" runat="server" ValidationGroup="Button2Group"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator3" runat="server" ControlToValidate="TextBox3"
        ErrorMessage="不能为空" ValidationGroup="Button2Group"></asp:RequiredFieldValidator><br />
    <asp:TextBox ID="TextBox4" runat="server" ValidationGroup="Button2Group"></asp:TextBox>
    <asp:RequiredFieldValidator ID="RequiredFieldValidator4" runat="server" ControlToValidate="TextBox4"
        ErrorMessage="不能为空" ValidationGroup="Button2Group"></asp:RequiredFieldValidator>
    <asp:Button ID="Button2" runat="server" Text="Button" ValidationGroup="Button2Group" /><br />
</div>
```





oup.aspx 43

Part 4

禁用验证

4.4.4 禁用验证

禁用验证的方法：

- 设置 ASP.NET 服务器控件的属性（CausesValidation="false"）来避开客户端和服务器的验证，而不只是客户端验证
- 禁用验证控件，即将控件的属性Enabled设置为false，使它根本不在页面上呈现并且不进行使用该控件的验证
- 如果要执行服务器上的验证，而不执行客户端的验证，则可以将单独验证控件设置为不生成客户端脚本，即将其属性EnableClientScript设为false

4.4.5 以编程的方式测试验证的有效性

- 测试常规错误状态
  - 在代码中测试页的 IsValid 属性，如果为true则执行代码；否则不执行

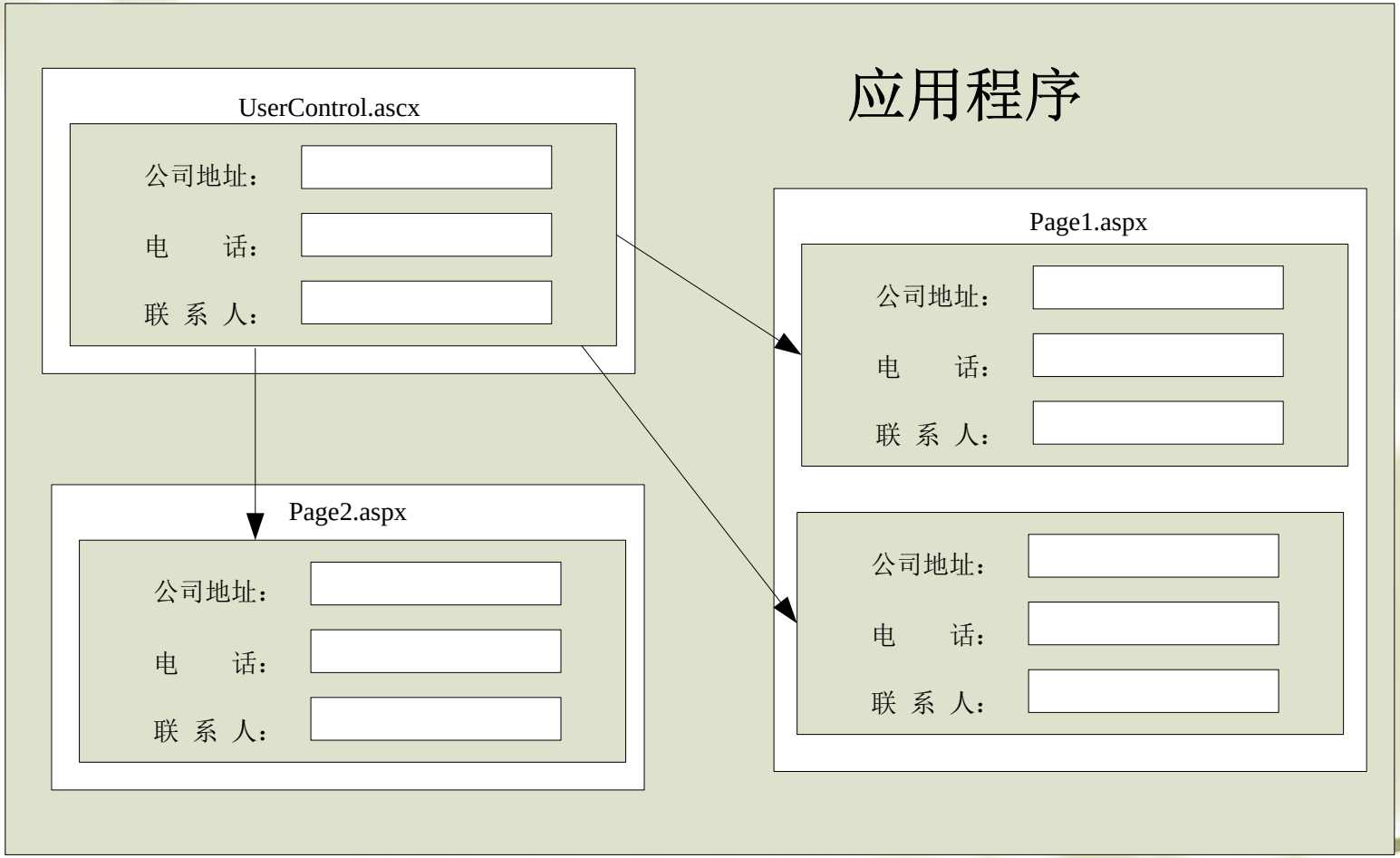
```
void Button1_Click(object sender, System.EventArgs e)
{
        if (IsValid)
        {
                //验证成功后执行的代码
         }
}
```

Part 5

用户控件

4.5 用户控件

- 用户控件是一种复合控件，可以像创建ASP.NET网页一样创建，然后在多个网页上重复使用。



48

用户控件与ASP.NET网页的区别：

- 用户控件的文件扩展名为.ascx。
- 用户控件中没有@Page指令，而是包含@Control指令，该指令对配置及其它属性进行定义。
- 用户控件不能作为独立文件运行。而必须像处理任何控件一样，将它们添加到ASP.NET页中才能运行。
- 用户控件中没有html、body或form元素。这些元素必须位于宿主页中。所谓宿主页即为使用用户控件的页面。
- 可以在用户控件上使用与在ASP.NET网页上所用相同的HTML元素（html、body或form元素除外）和Web控件。例如，如果要创建一个将用作工具栏的用户控件，则可以将一系列Button的Web服务器控件放在该控件上，并创建这些按钮的事件处理程序。

4.5.1用户控件的创建

【4-30】演示如何创建一个用户控件。该控件显示一个文本框和两个up和down按钮，用户可单击两个按钮来增加或减少文本框中的值（值的范围可以设置）。

WebUserControl.ascx 50

用户控件的创建流程

1. 创建用户控件文件：**WebUserControl.ascx**
2. 添加**HTML**控件或服务器控件，设置样式

3. 添加用户控件的私有属性和公开属性
   - **MinValue、MaxValue、CurrentValue**
4. 添加初始化代码
5. 添加用户控件中的事件处理逻辑
   - “-”按钮响应、“+”按钮响应

4.5.2用户控件的使用

- 为了在Web页面上使用用户控件，需要两个步骤：
- 使用@Register指令在页面顶部注册用户控件。（自动）
- 在需要使用用户控件的位置放置用户控件。

【例4-31】演示如何Web窗体中使用用户控件。

UcSample.ascx 52

- **<%@ Register%>指令，该指令需要指定3个属性。**
  - **TagPrefix属性：指定与用户控件关联的命名空间，可以指定任何字符串。**
  - **TagName属性：指定在ASP.NET Web页面中使用的用户控件的名字，可以指定任何字符串。**
  - **Src属性：指定用户控件的虚拟路径。**
- **在用户控件的声明区中，使用<TagPrefix: TagName…>这样的语法来定义用户控件。**

用户控件的使用示例

```
1  <%@ Page Language="C#" AutoEventWireup="true" CodeFile="UcSample.aspx.cs" Inherits="U
2
3  <%@ Register Src="~/WebUserControl.ascx" TagPrefix="uc1" TagName="WebUserControl" %>
4
5  <!DOCTYPE html>
6  <html xmlns="http://www.w3.org/1999/xhtml">
7  <head runat="server">
8  <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
9      <title></title>
10 </head>
11 <body>
12     <form id="form1" runat="server">
13     <div>
14         <uc1:WebUserControl runat="server" ID="WebUserControl" />
15     </div>
16     </form>
17 </body>
18 </html>
```

4.6 小结

- HTML服务器控件
- Web服务器控件
- 验证控件
- 用户控件

本 章 结 束!