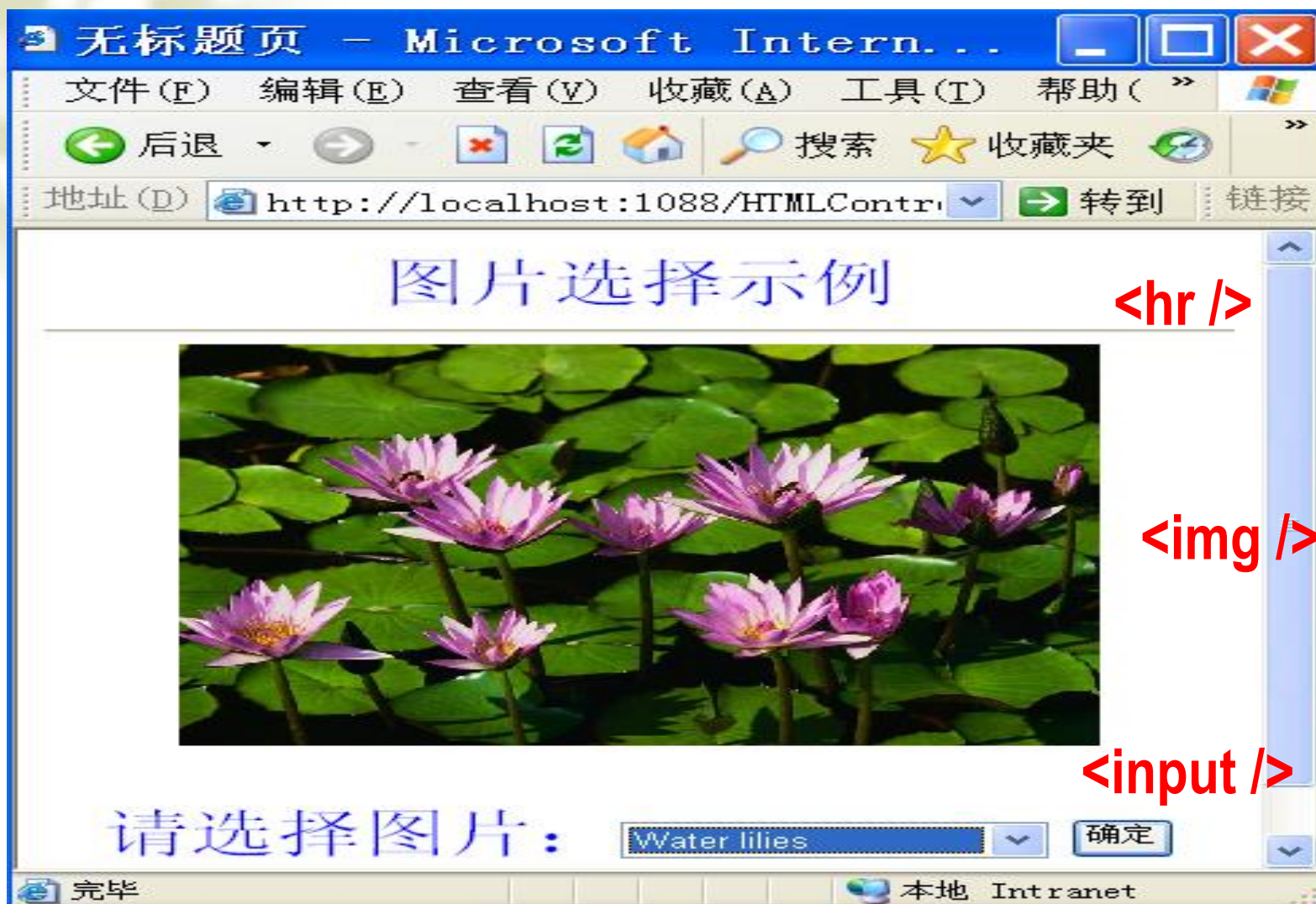


上周回顾



`<hr />`

``

`<input />`

`<select></select>`

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
 - ❧ 4.1 Web服务器控件概述
 - ❧ 4.2 常用Web服务器控件
- ❖ 验证控件
- ❖ 用户控件

Part 1

WEB服务器控件概述



4.3.1 Web服务器控件概述

- ❖ 如何添加Web服务器**控件**？
- ❖ 如何设置Web服务器控件的**属性**？
- ❖ 如何为Web服务器控件添加**客户端事件**？
- ❖ 如何为Web服务器控件添加**服务器端事件**？

Web服务器控件基本语法

从左边工具箱中拖放一个**Web**服务器控件到页面上，其**HTML**源代码标记如下：

<asp:控件名 ID="控件名称" runat="server" 属性1="值"属性2="值".../>

或者：

<asp:控件名 ID ="控件名称" runat="server" 属性1="值"属性2="值"...></asp:控件名>

如**Button**控件，基本语法为：

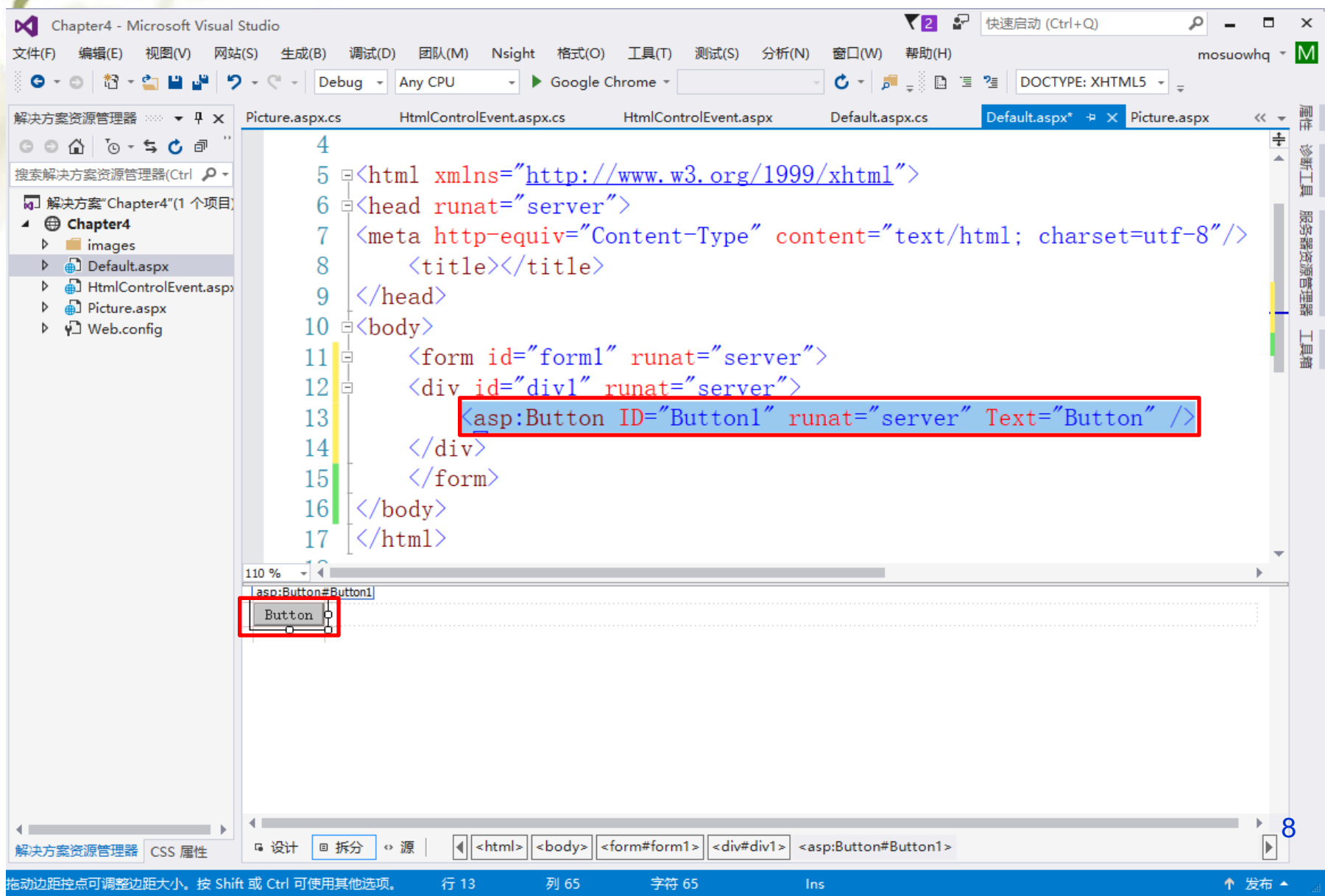
<asp:Button id ="Button2" runat="server" Text="Button" />

1. 将控件添加到Web窗体的方法

将控件添加到Web窗体中，有3种方法。

- ❖ 方法1：从工具箱中添加控件。
- ❖ 方法2：在源视图中，直接添加控件声明代码。
- ❖ 方法3：以编程方式动态创建Web服务器控件。

方法1 从工具箱中添加Web控件



方法2 源视图直接添加控件声明代码

Chapter4 - Microsoft Visual Studio

文件(F) 编辑(E) 视图(V) 网站(S) 生成(B) 调试(D) 团队(M) Nsight 格式(O) 工具(T) 测试(S) 分析(N) 窗口(W) 帮助(H)

Debug Any CPU Google Chrome DOCTYPE: XHTML5

解决方案资源管理器

搜索解决方案资源管理器(Ctrl+Shift+F)

解决方案 "Chapter4" (1 个项目)

- Chapter4
 - images
 - Default.aspx
 - HtmlControlEvent.aspx
 - Picture.aspx
 - Web.config

Picture.aspx.cs

```
4
5 <html xmlns="http://www.w3.org/1999/xhtml">
6 <head runat="server">
7 <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
8 <title></title>
9 </head>
10 <body>
11 <form id="form1" runat="server">
12 <div id="div1" runat="server">
13 <asp:Button ID="Button1" runat="server" Text="Button" />
14 </div>
15 </form>
16 </body>
17 </html>
```

110 %

asp:Button#Button1

Button

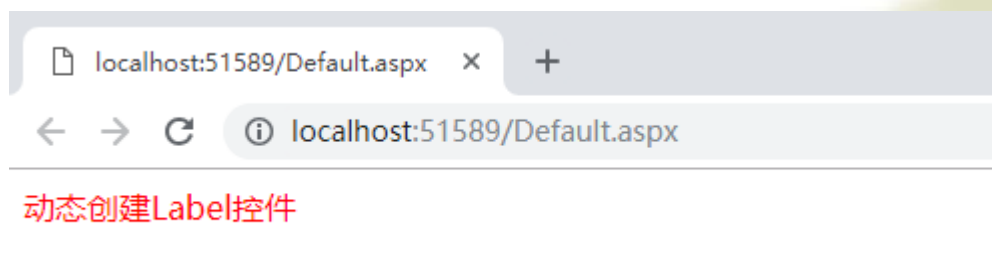
设计 拆分 源

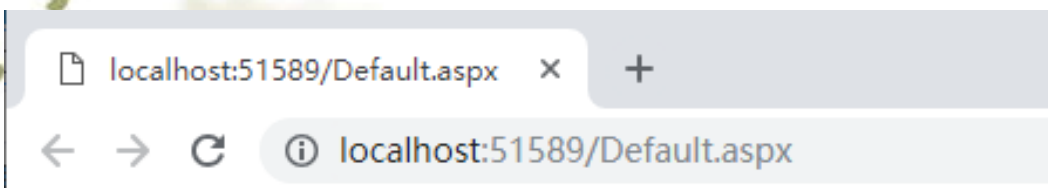
<html> <body> <form#form1> <div#div1> <asp:Button#Button1>

拖动边距控点可调整边距大小。按 Shift 或 Ctrl 可使用其他选项。 行 13 列 65 字符 65 Ins 9

方法3 以编程方式动态创建

```
protected void Page_Load(object sender, EventArgs e)
{
    // 创建控件并设置属性
    Label newLabel = new Label();
    newLabel.Text = "动态创建Label控件";
    newLabel.ForeColor = System.Drawing.Color.Red;
    // 添加控件到Web窗体中
    this.form1.Controls.Add(newLabel);
}
```





测试文本1

测试文本2

动态创建Label控件

如何让动态创建的Label控件放置在“测试文本1”和“测试文本2”之间？

```
<form id="form1" runat="server">
<div id="div1" runat="server">
    <p>测试文本1</p>
    <div id="div2" runat="server">
        </div>
        <p>测试文本2</p>
    </div>
</form>
```

```
// 创建控件并设置属性
Label newLabel = new Label();
newLabel.Text = "动态创建Label控件";
newLabel.ForeColor = System.Drawing.Color.Red;
// 添加控件到Web窗体中
//this.form1.Controls.Add(newLabel);


div2.Controls.Add(newLabel);


```

2. Web服务器控件的公共属性

属 性	说 明
AccessKey	定义控件的快捷键。例如：定义控件的AccessKey属性为A，则表示访问该控件的快捷键为Ctrl+A
TabIndex	设置网页中各控件的Tab键顺序，用户敲击Tab键即可按设置顺序在控件中移动焦点。
Attributes	控件属性集合。该属性只能在编程时指定。
BackColor	控件的背景颜色
Enabled	控件能否被用户访问
Font	控件上的文本字体。如Font.Name表示字体名称，Font.Bold表示是否加粗显示
ForeColor	控件上文本的颜色
Height	控件的高度，以像素点为单位
Width	控件的宽度，以像素点为单位
ToolTip	设置当鼠标指针悬浮停在Web服务器控件上时显示的文本
Visible	控件是否可见

控件属性的设置有两种方法：

- ❖ 方法1：设计时通过属性窗设置控件的属性。
- ❖ 方法2：运行时以编程方式设置控件的属性。

3. Web服务器控件的客户端事件

向**ASP.NET**服务器控件添加客户端事件的3种方法：

❖ **方法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

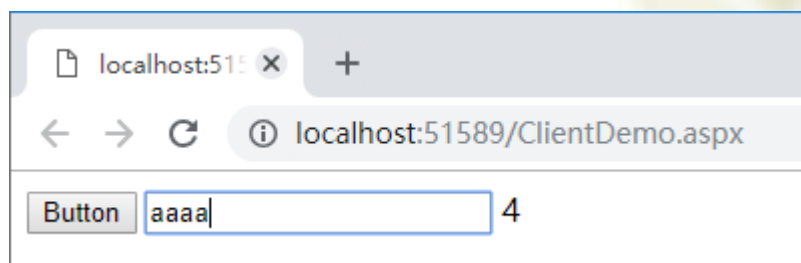
- 在源视图中，为控件直接添加事件属性。例如，**onmouseover**或**onkeyup**。针对不同属性添加要执行的客户端脚本。
- **【例4-3】**演示一个包含客户端脚本的**ASP.NET**网页，当用户将鼠标移到按钮上方时，此脚本就会更改该按钮的文本颜色。


```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title></title>
<script type="text/javascript">
    var previousColor;
    function MakeBlue() {
        previousColor = document.getElementById("Button1").style.color;
        document.getElementById("Button1").style.color = "blue";
    }
    function RestoreColor() {
        document.getElementById("Button1").style.color = previousColor;
    }
</script>
</head>
<body>
<form id="form1" runat="server">
<div>
    <asp:Button ID="Button1" runat="server" Text="Button" onmouseover="MakeBlue()" onmouseout="RestoreColor()" />
</div>
</form>
</body>
</html>
```

❖ 方法2：以编程方式向**ASP.NET**控件添加客户端事件处理程序。

- 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。
- 【例4-4】演示如何动态地向TextBox控件添加客户端脚本。

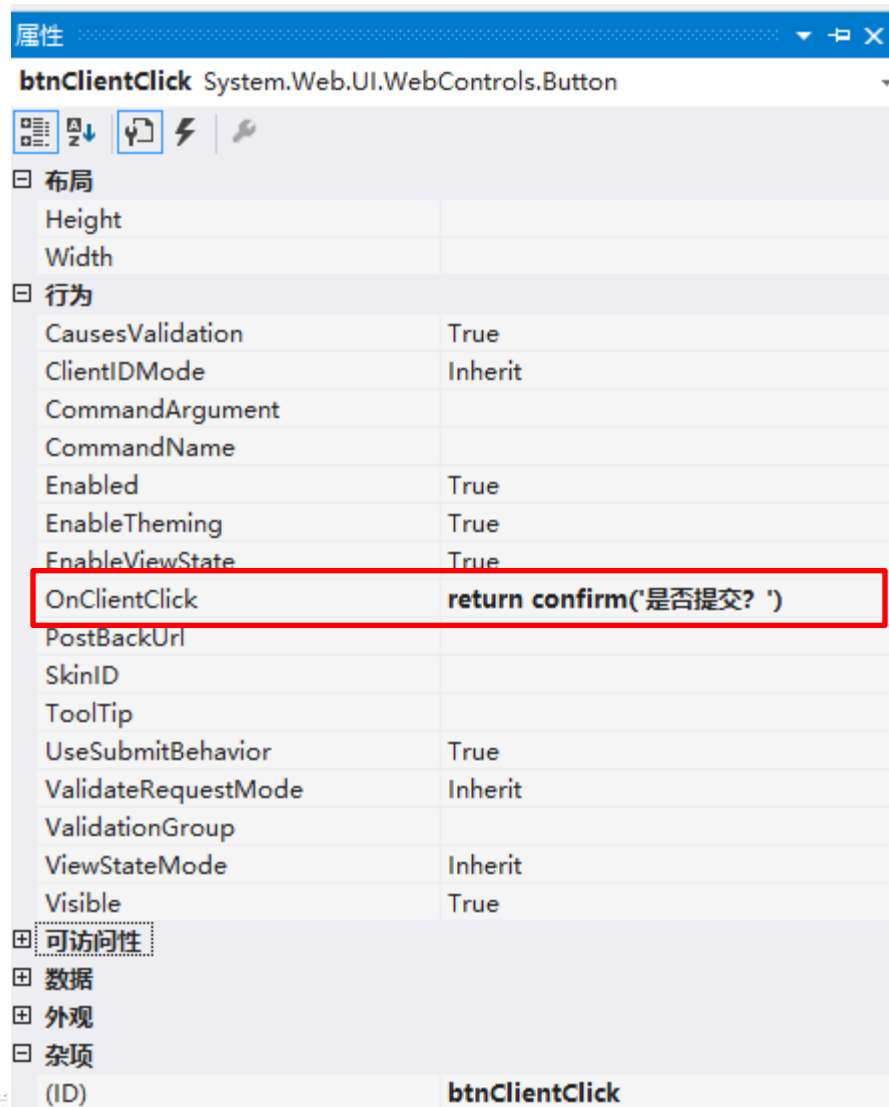
```
protected void Page_Load(object sender, EventArgs e)
{
    TextBox1.Attributes.Add("onkeyup", "spanCounter.innerText = this.value.length");
}
```



方法3：向按钮控件添加客户端单击事件。

- 在按钮控件(例如Button)中, 要添加客户端OnClick事件, 可以在设计视图将按钮控件的**OnClick**属性设置为要执行的客户端脚本。

- 【例4-5】演示如何向Button控件添加客户端Click事件。



HTML按钮和ASP.NET按钮

```
<form id="form1" runat="server">
<div>
    <input id="Submit1" type="submit" value="submit" onclick="return Submit1_onclick()"
        runat="server" onserverclick="Submit1_ServerClick"/>
</div>
</form>
```

客户端事件

服务器端事件

```
<form id="form1" runat="server">
    <div>
        <asp:Button ID="Button1" runat="server" Text="Button" OnClick="Button1_Click"
            OnClientClick="return confirm('是否提交')" />
    </div>
</form>
```

服务器端事件

服务器端事件

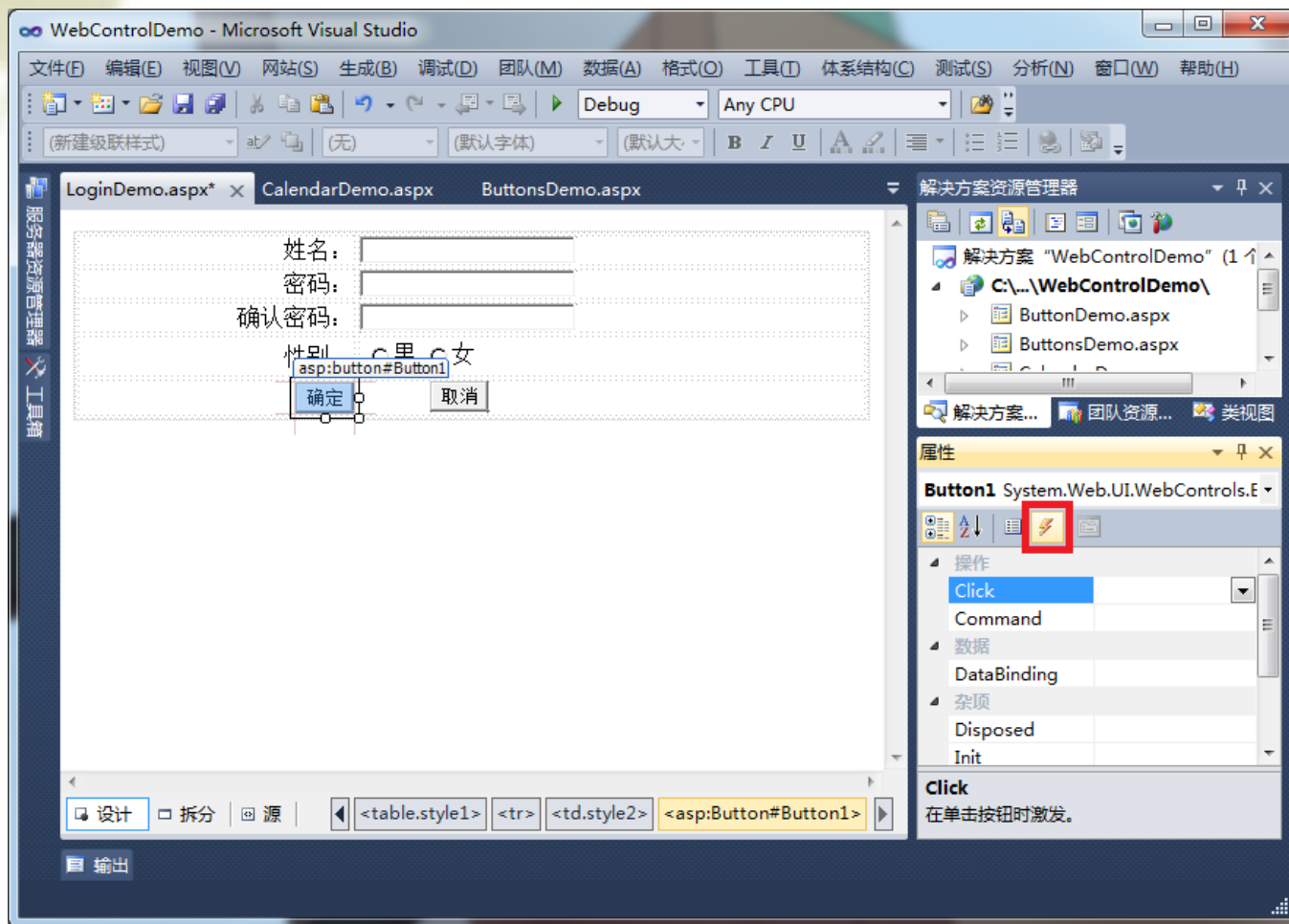
4. Web服务器控件的服务器端事件

事 件	说 明
Click	当Web服务器控件被按下时会触发该事件。 Button 、 ImageButton 控件具有该事件
TextChanged	当Web服务器控件上的文本发生变化时会触发该事件。 TextBox 控件具有该事件
CheckedChanged	当Web服务器控件的选项发生变化时会触发该事件。 CheckBox 、 RadioButton 控件具有该事件
SelectedIndexChanged	当Web服务器控件的列表选项发生变化时会触发该事件。列表类控件 CheckBoxList 、 DropDownList 、 ListBox 、 RadioButtonList 控件具有该事件

19

添加服务器端事件的2种方法

❖ 方法1：设计时在属性窗中添加事件处理



❖ 方法2：运行时以编程方式动态添加事件处理。

```
protected void Page_Load(object sender, EventArgs e)
{
    TextBox1.Attributes.Add("onkeyup", "spanCounter.innerText = this.value.length");
    Button2.Click += new EventHandler(Button2_Click);
}

protected void Button2_Click(object sender, EventArgs e)
{
    Response.Write("按钮2服务器端事件输出文本");
}
```

EventHandler是.net Framework提供的一个事件委托类，为控件的事件和事件处理方法建立关联。

Part 2

常用WEB服务器控件





Image

书目检索

热门推荐

分类浏览

新书通报

期刊导航

读者荐购

学科参考

信息发布

我的图书馆

简单检索

全文检索

多字段检索

馆藏书目简单检索

Label

DropDownList

TextBox

Button

题名

前方一致

检索

更多限制

☒ 所有书刊 ☐ 中文图书 ☐ 西文图书 ☐ 中文期刊 ☐ 西文期刊

☒ 显示电子书刊

热门检索: [文化研究指南](#) | [阴阳师](#) | [中国近代史](#) | [第二性](#) | [强风吹拂](#) | [毛毛](#) | [芋泽](#) | [水妖](#) | [局外人](#) | [政治经济](#) |

[查看30天内的热门检索词](#)

RadioButton

HyperLink

CheckBox

热门检索词 热门借阅

- 想象中国的方法:历史...
- 历史的终结及最后之人...
- 乌合之众:大众心理研...
- 心理学与生活
- 资本论.3版
- 理解媒介:论人的延伸...
- 消费者行为心理学
- 铁血军魂:一八〇师在...
- 法语1(修订本)教学...
- 影像语言

热门图书

- 白夜行
- 平凡的世界:第一部,...
- 解忧杂货店
- 赖声川的创意学
- 娱乐至死
- 娱乐至死:童年的消逝...
- 百年孤独
- 天才在左 疯子在右
- 看见
- 娱乐至死

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

1 Label控件

1.Label控件：用于在网页上显示文本。

❖ 声明Label的语法格式：

```
<asp:Label ID="控件名" runat="server" Text="文本"
"></asp:Label>
```

❖ Label控件只能用来显示文本，它只有**Text属性**，用来获取或设置指定在Label控件上显示的文字。

❖ 后端动态创建Label控件？

2.TextBox控件

- ❖ **TextBox**控件可用于制作单行、多行文本框和密码框。
- ❖ 声明**TextBox**控件的语法格式：

```
<asp:TextBox ID="控件名" runat="server">  
</asp:TextBox>
```


TextBox控件本身特有的属性：

➤ **TextMode属性：** 获取或设置文本框的类型。

TextBox有3种取值：**MultiLine**为多行输入模式；**Password**为密码输入模式；**SingleLine**为单行输入模式。默认值为**SingleLine**。

➤ **MaxLength属性：** 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属性：** 获取或设置**TextBox**控件是否为只读。当该属性设置为**True**时，将禁止用户输入或更改现有值。默认值为**False**。
- **Text属性：** 获取或设置文本框的文本内容。
- **AutoPostBack属性：** 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**，不回传；为**True**时，则要回传。该属性要与**TextChanged**事件配合使用。

❖ **TextBox**主要的服务器端事件是**TextChanged**事件。

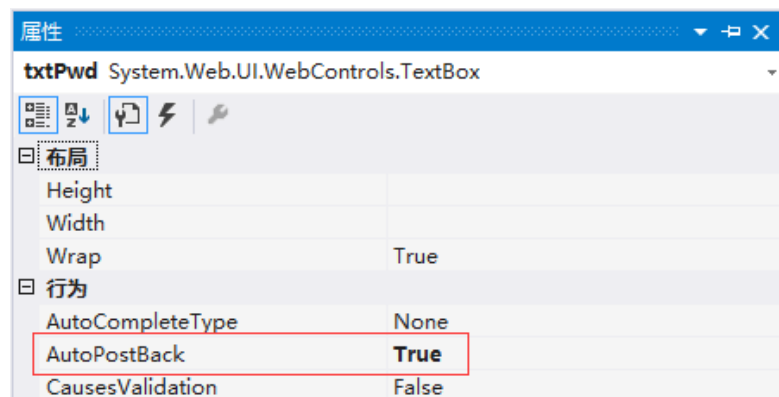
- 当**Text**属性的值改变时，会触发此事件。
- 当**AutoPostBack**属性设置为**True**时，用户更改文本框的内容并将焦点移开文本框时，将自动回传页面并触发**TextChanged**事件。
- 当**AutoPostBack**属性设置为**False**时，即便用户更改文本框的内容并将焦点移开文本框，也不会自动回传页面，需等到回传页面的事件发生时（如按钮控件的**Click**事件），才会触发**TextChanged**事件。

【例4-6】演示如何将TextBox控件的AutoPostBack属性与TextChanged事件配合使用。

请输入用户名和密码：

用户名：

密码：



```
protected void txtPwd_TextChanged(object sender, EventArgs e)
{
    txtResult.Text = txtName.Text + "，您输入的密码是：" + txtPwd.Text;
}
```

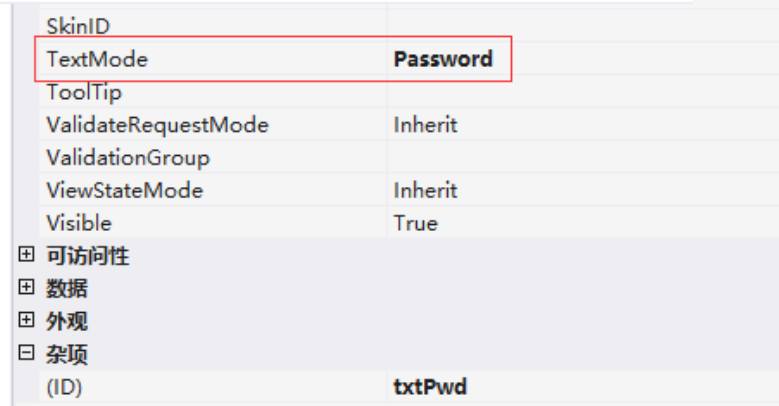
请输入用户名和密码：

用户名：

张三

密码：

张三，您输入的密码是：12345



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

3. HyperLink控件

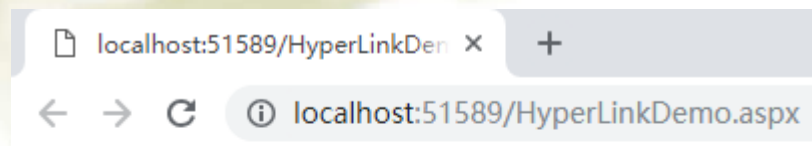
- ❖ 超链接（**HyperLink**）控件用于创建文本或图片超链接。声明**HyperLink**控件的语法格式：

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

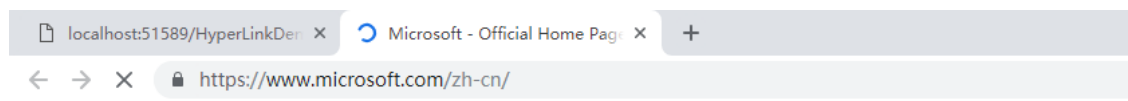
HyperLink控件的主要属性如下：

- **ImageUrl属性**：获取或设置HyperLink控件的图片来源。
- **NavigateURL属性**：获取或设置HyperLink控件所要链接到的URL。
- **Target属性**：获取或设置单击HyperLink控件时显示链接到的网页内容的目标窗口或框架。
- **Text属性**：获取或设置HyperLink控件的超链接文本。

【例4-7】 演示如何设置Target属性来指定窗口或框架，用以显示与HyperLink控件关联的页。



微软网站



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

4. Image控件

- ❖ Image控件只是简单地完成一个**图像显示任务**，与HTML的Image功能相同。
- ❖ 声明Image控件的语法格式为：
`<asp:Image ID="控件名" runat="server"></asp:Image>`

该控件的属性：

- **AlternateText属性**：设置图像无法显示时显示的替换文字。
- **ImageUrl属性**：设置图像所在的位置。
- **ToolTip属性**：将鼠标放置在图片控件上时，显示的工具提示。

注意：Image控件不响应用户事件，但是它可以根据其它控件的输入动态地显示图片。

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

5. Button、LinkButton和ImageButton控件

- ❖ **ASP.NET**包含3类用于向服务器端提交表单的控件：**Button**、**LinkButton**和**ImageButton**。这3类控件拥有同样的功能，但每类控件的**外观**截然不同。

特殊属性：

- **CommandArgument**：用于指定传给**Command**事件的命令参数。
- **CommandName**：指定传给**Command**事件的命令名。
- **OnClickClientClick**：指定点击按钮时执行的客户端脚本。
- **PostBackUrl**：单击按钮时所发送到的URL。
- **Text**：将在按钮上呈现的文本。
- **UseSubmitBehavior**：指示按钮是否呈现为提交按钮。

- ❖ **Button**控件支持**Focus()**方法，用于把焦点设为该**Button**控件。

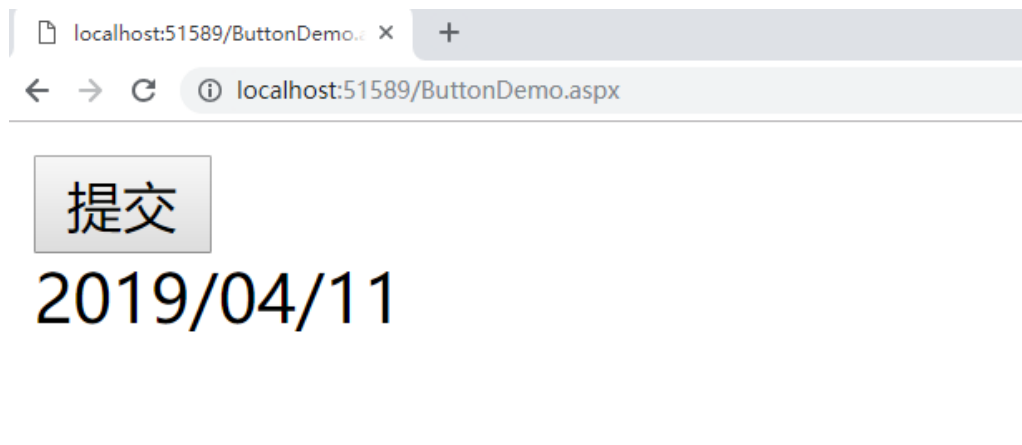
Button控件还支持下面两个事件：

- **Click**: 点击**Button**控件时引发。
- **Command**: 点击**Button**控件时引发。
CommandName和**CommandArgument**的值传给这个事件。

声明**Button**控件的语法格式:

```
<asp: Button ID ="控件名" runat="server"  
Text="按钮上的文字"></asp: Button>
```

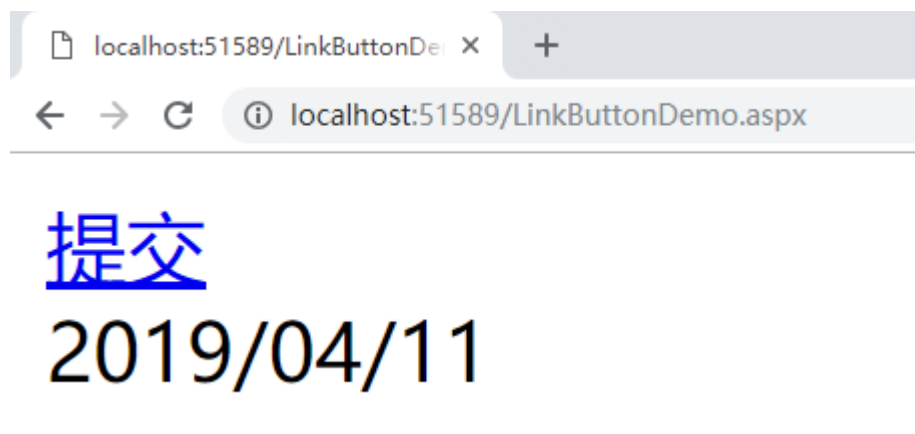
【例4-8】 在页面上包含一个**Button**控件，点击该**Button**控件，将在**Label**控件中显示系统日期。



❖ 声明**LinkButton**控件的语法格式为：

```
<asp: LinkButton ID ="控件名" runat="server" >  
    按钮上的文字</asp:LinkButton>
```

【例4-9】在页面上包含一个**LinkButton**控件，点击该**LinkButton**控件，将在**Label**控件中显示系统日期。



LinkButtonDemo.aspx⁴⁰

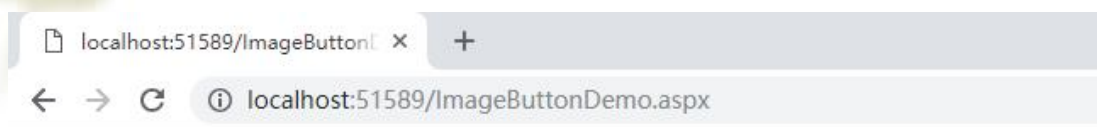
- ❖ **ImageButton**控件是以图片的形式提供的按钮控件，功能与**Button**相同，外观与**Image**控件相同。
- ❖ 声明**ImageButton**控件的语法格式为：

```
<asp: ImageButton ID ="控件名"  
    runat="server"> </asp: ImageButton>
```

示例：

```
<asp:ImageButton ID="ImageButton1" runat="server"  
    ImageUrl="~/images/Sunset.jpg"  
    OnClick="ImageButton1_Click"  
    Width="200px" Height="150px"/>
```

【例4-10】 在页面中包含一个**ImageButton**和一个**Label**控件，点击该**ImageButton**上的图片，将在**Label**控件中显示点击的位置。



你单击的坐标为X:78, Y:71

ImageButtonDemo.aspx 42

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

6. DropDownList和ListBox控件

❖ **DropDownList**控件允许用户从预定义的下拉列表中选择一项。

❖ 声明**DropDownList**控件的语法格式为：

```
<asp:DropDownList ID="控件名" runat="server" >
```

```
<asp:ListItem Value="" >Text</asp:ListItem>
```

...

```
</asp:DropDownList>
```



DropDownList控件的主要属性如下：

- **AutoPostBack属性：** 获取或设置当改变DropDownList控件的选择状态时，是否自动上传窗体数据到服务器。默认为**False**。
- **Items属性：** 包含该控件所有选项的集合。每个列表项都是一个单独的对象，具有自己的属性。
- **SelectedIndex属性：** 获取当前选择项的下标（下标从**0**开始）。
- **SelectedItem属性：** 获取当前选择项对象。

- ❖ **DropDownList**控件有**SelectedIndexChanged**事件，当用户选择一项时，**DropDownList**控件将引发**SelectedIndexChanged**事件。
- ❖ 默认情况下，此事件不会导致向服务器发送页面，但当该控件的**AutoPostBack**属性设置为**True**时，该事件会立即回发页面。

【例4-12】演示DropDownList控件的使用。

```
protected void btnNew_Click(object sender, EventArgs e)
{
    ListItem item = new ListItem(TextBox1.Text, TextBox2.Text);
    DropDownList1.Items.Add(item);
}

protected void btnDelete_Click(object sender, EventArgs e)
{
    DropDownList1.Items.Remove(DropDownList1.SelectedItem);
}

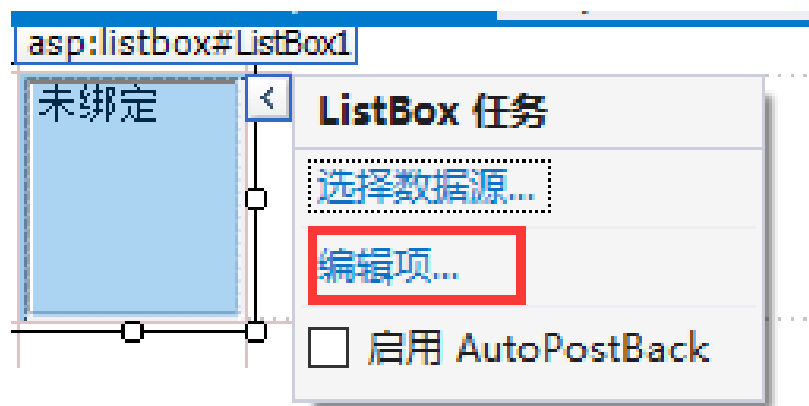
protected void btnShow_Click(object sender, EventArgs e)
{
    Response.Write("选中项的文本为: " + DropDownList1.SelectedItem.Text + "<br/>");
    Response.Write("选中项的值为: " + DropDownList1.SelectedItem.Value + "<br/>");
}
```

❖ **ListBox**控件与**DropDownList**控件的功能基本相似，**ListBox**控件将所有选项都显示出来，提供单选或多选的列表框。

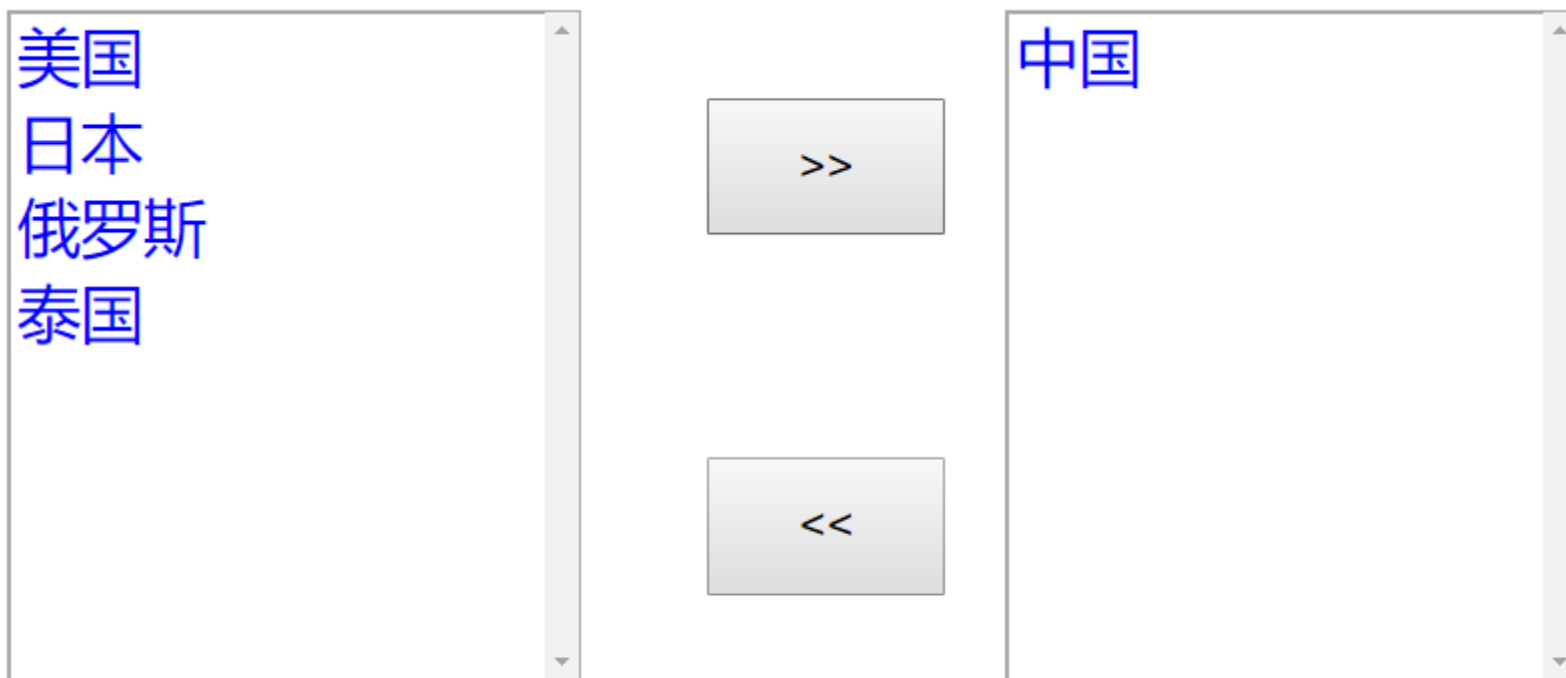
ListBox控件比**DropDownList**控件多两个属性：

➤ **Rows**属性：获取或设置**ListBox**控件显示的选项行数，默认值为4。

➤ **SelectionMode**属性：获取或设置**ListBox**控件的选项模式，**Single**为单选，**Multiple**为多选，默认为**Single**。当允许多选时，只需按住**Ctrl**键或**Shift**键并单击要选取的选项，便可完成多选。



【例4-13】 演示ListBox控件的使用。



```
ListBox2.Items.Add(new ListItem(ListBox1.SelectedItem.Text,  
ListBox1.SelectedItem.Value));
```

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

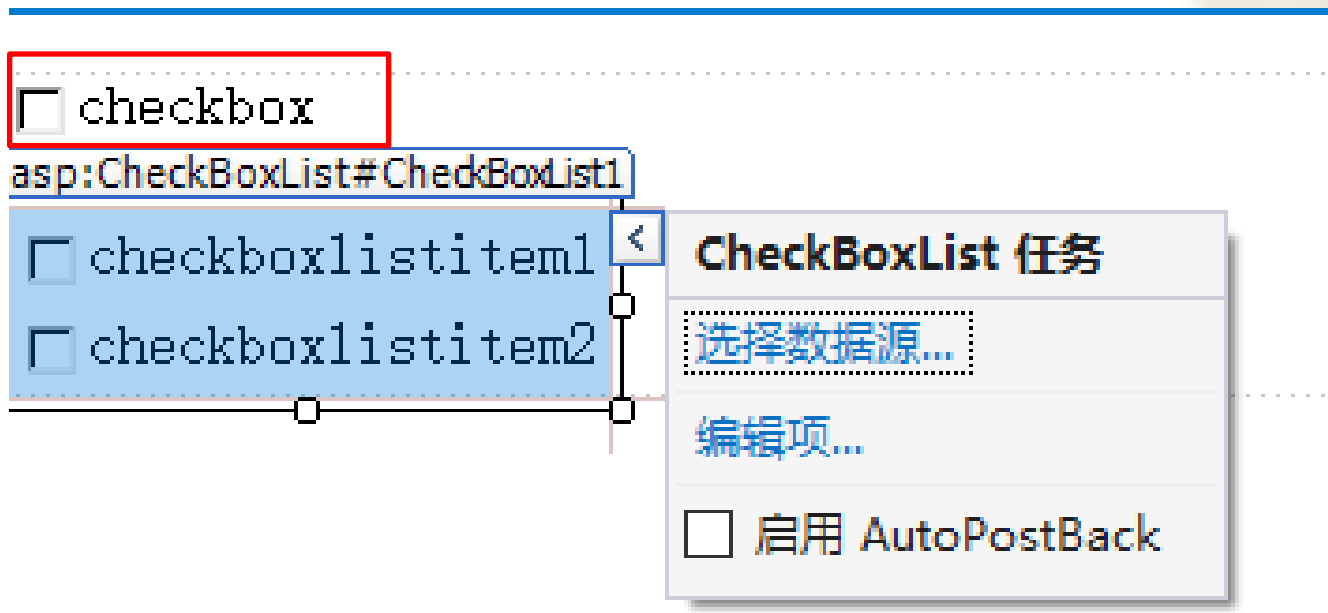
☞ View控件

11. Calender控件

7. CheckBox和CheckBoxList控件

- ❖ **CheckBox**控件用于创建单个复选框，供用户选择。
- ❖ 声明**CheckBox**控件的语法格式：

**<asp: CheckBox ID="控件名" runat="server"
Text="控件的文字" value="" />**



❖ CheckBox控件的常用属性:

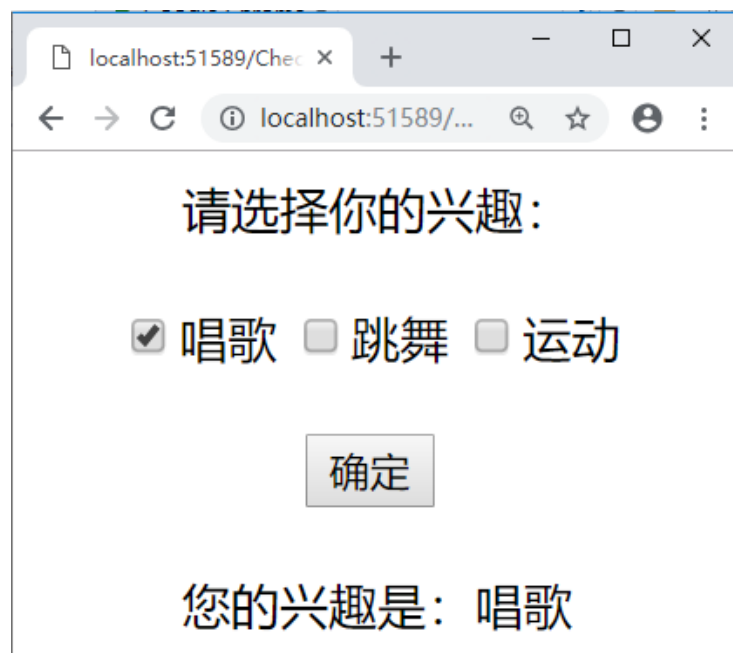
- **Checked**属性: 获取或设置该项是否选中。
- **TextAlign**属性: 控件文字的位置。
- **Text**属性: 获取或设置**CheckBox**控件的文本内容。
- **Value**属性: 获取或设置**CheckBox**控件的值内容。
- **AutoPostBack**属性: 获取或设置当改变**CheckBox**控件的选择状态时, 是否自动回传窗体数据到服务器。值为**True**时, 表示单击**CheckBox**控件, 页面自动回发; 值为**False**时, 不回发。默认值为**False**。

- ❖ **CheckBox**控件具有**CheckedChanged**事件。当**Checked**属性的值改变时，会触发此事件，该事件要与**AutoPostBack**属性配合使用。

【例4-14】CheckBox控件的使用。

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

```
    Label1.Text = "您的兴趣是：";
    if (CheckBox1.Checked)
    {
        Label1.Text += CheckBox1.Text;
    }
    if (CheckBox2.Checked)
    {
        Label1.Text += CheckBox2.Text;
    }
    if (CheckBox3.Checked)
    {
        Label1.Text += CheckBox3.Text;
    }
}
```



❖ 声明**CheckBoxList**控件的语法格式:

```
<asp: CheckBoxList ID="控件名" Runat="server">  
    <asp: ListItem value="" > text </asp: ListItem>  
    .....  
</asp: CheckedBoxList>
```

❖ 该控件的属性、用法及功能与**ListBox**控件基本相同。除此之外，还有自己的特殊属性。

- **RepeatDirection**: 表示是横向还是纵向排列。
- **RepeatColumns**: 一行排几列。
- **TextAlign**属性: 控件文字的位置。
- **Selected**属性: 表示该选项是否选中。

【例4-15】 使用CheckBoxList控件完成【例4-14】的功能。

```
<span>请选择你的兴趣：</span><br /><br />
<asp:CheckBoxList ID="CheckBoxList1" runat="server" RepeatDirection="Horizontal">
    <asp:ListItem>唱歌</asp:ListItem>
    <asp:ListItem>跳舞</asp:ListItem>
    <asp:ListItem>运动</asp:ListItem>
</asp:CheckBoxList><br /><br />
<asp:Button ID="Button1" runat="server" Text="确定" OnClick="Button1_Click" />
<br /><br />
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
```

```
Label1.Text = "你的兴趣是：";
for(int i = 0; i< CheckBoxList1.Items.Count; i++)
{
    if (CheckBoxList1.Items[i].Selected)
    {
        Label1.Text += CheckBoxList1.Items[i].Text;
    }
}
```

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

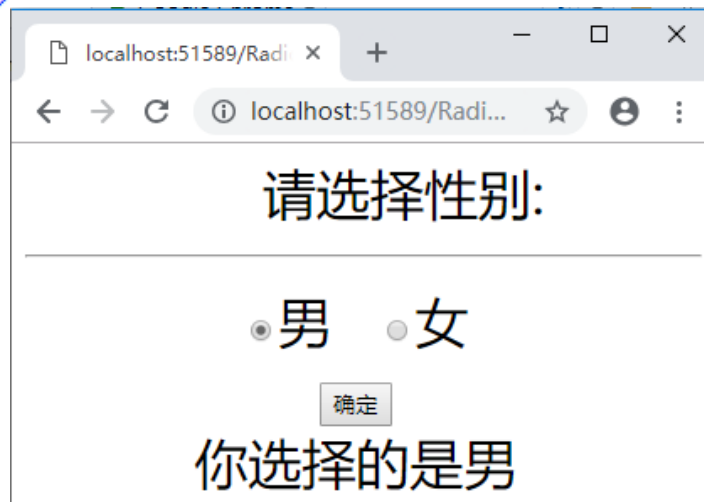


8. RadioButton和RadioButtonList控件

- ❖ **RadioButton**控件用于创建单个单选框，供用户选择。
- ❖ 声明**RadioButton**控件的语法格式：
`<asp: RadioButton ID="控件名" runat="server" text="控件的文字" value=""/>`
- ❖ **RadioButton**控件的常用属性和事件与**CheckBox**控件基本相同。
- ❖ **RadioButton**控件还有一个特殊的属性**GroupName**，用于设置单选按钮所属的组名，通过将多个单选按钮的组名设为相同值，将其分为一组相互排斥的选项。

【例4-16】演示RadioButton控件的使用。

```
<asp:RadioButton ID="RadioButton1" runat="server"
    Text="男" Checked="True" GroupName="Sex" />&nbsp;
<asp:RadioButton ID="RadioButton2" runat="server"
    Text="女" GroupName="Sex" />
```



```
if (RadioButton1.Checked)
    Label1.Text = "你选择的是" + RadioButton1.Text;
else
    Label1.Text = "你选择的是" + RadioButton2.Text;
```

RadioButtonDemo.aspx⁵⁹

- ❖ 与CheckBoxList控件类似，**RadioButtonList**控件是一个RadioButton控件组，当存在多个单选框时，用该控件比RadioButton简单。
- ❖ 声明RadioButtonList控件的语法格式：

```
<asp: RadioButtonList ID="控件名" runat="server">  
    <asp: ListItem Value="" > Text </asp: ListItem>  
    .....  
</asp: RadioButtonList>
```
- ❖ **RadioButtonList**控件和**CheckBoxList**控件的属性和事件相同。

【例4-17】 使用RadioButtonList控件完成【例4-16】的功能。

```
<asp:RadioButtonList ID="RadioButtonList1" runat="server"
    RepeatDirection="Horizontal">
    <asp:ListItem Selected="True">男</asp:ListItem>
    <asp:ListItem>女</asp:ListItem>
</asp:RadioButtonList>
```

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = "你选择的是" + RadioButtonList1.SelectedItem.Text;
}
```

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

9.Panel控件

- ❖ **Panel**控件是一个放置其它控件的容器，可以在其内放置不同控件。利用它的这个特性，可以将不同的控件组成一个群组，并控制它的显示或隐藏。
- ❖ 声明**Panel**控件的语法格式：
<asp: Panel ID="控件名" runat="server" >
</asp:Panel>

localhost:55673/Default.aspx

localhost:55673/Default.aspx

浙江传媒学院图书馆书目检索系统

ILIB Online Public Access Catalogue

馆藏书目简单检索

题名 前方一致

检索 更多限制

☒ 所有书刊 ☐ 中文图书 ☐ 西文图书 ☐ 中文期刊 ☒ 显示西文书刊

显示方式: 20条 ☒ 详细显示 ☐ 表格显示

排序方式: 入藏日期 ☐ 升序排列 ☒ 降序排列

检索范围: 所有校区

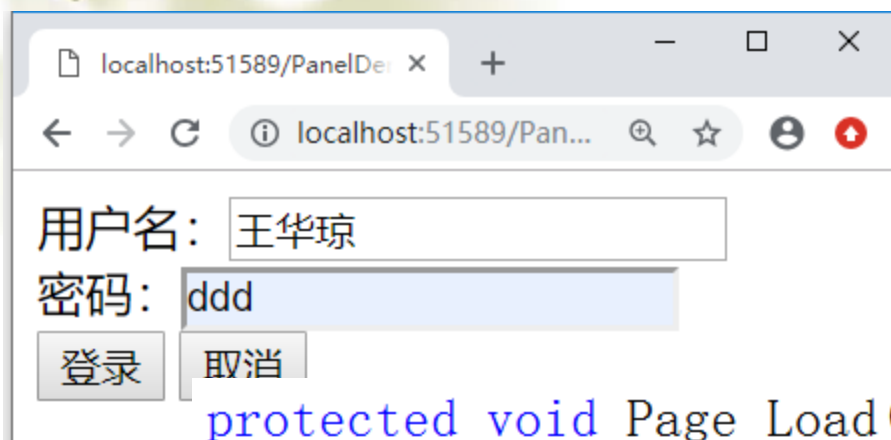
热门检索: 黎锦晖 python 高等数学 文化研究指南 中国近代史 阴阳师 毛毛 中外音乐交流 第二性 辛泽



❖ Panel控件的主要属性:

- **BackColor属性:** 设置Panel背景图形。
- **HorizontalAlign属性:** 设置水平对齐方式。
- **Visible属性:** 是否显示还是隐藏。
- **ScrollBars属性:** 是否设置水平或垂直滚动条。例如, 将该属性设置为Auto时, 当控件长度和宽度超过Panel控件的长或宽时, 将自动显示出滚动条。

演示Panel控件的使用。



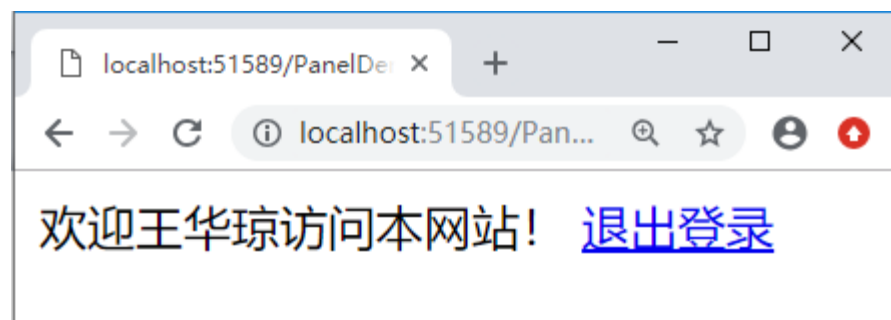
localhost:51589/PanelDe x + - □ ×

← → ↻ ⓘ localhost:51589/Pan... 🔍 ☆ 👤 ⬆

用户名: 王华琼

密码: ddd

登录 取消



localhost:51589/PanelDe x + - □ ×

← → ↻ ⓘ localhost:51589/Pan... 🔍 ☆ 👤 ⬆

欢迎王华琼访问本网站! [退出登录](#)

```
protected void Page_Load(object sender, EventArgs e)
{
    Panel1.Visible = true;
    Panel2.Visible = false;
}
```

```
protected void Button1_Click(object sender, EventArgs e)
{
    Panel1.Visible = false;
    Panel2.Visible = true;
    Label1.Text = TextBox1.Text;
}
```

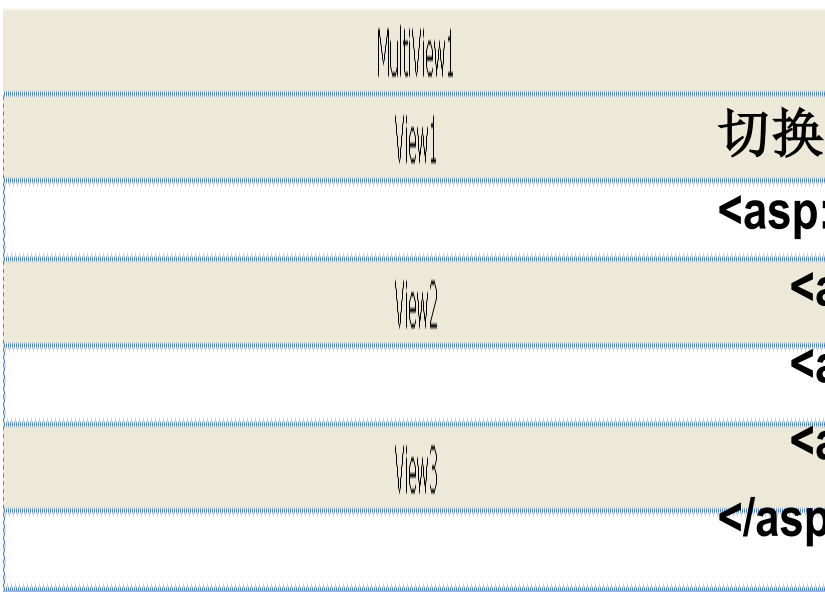
常用Web服务器控件

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

10.MultiView和View控件

- ❖ **MultiView**和**View**控件可以制作选项卡的效果，**MultiView**控件用作一个或多个**View**控件的外部容器。**View**控件又可包含标记和控件的任何组合。



切换到源视图，控件声明代码如下：

```
<asp:MultiView ID="MultiView1" runat="server">  
    <asp:View ID="View1" runat="server"> </asp:View>  
    <asp:View ID="View2" runat="server"> </asp:View>  
    <asp:View ID="View3" runat="server"> </asp:View>  
</asp:MultiView>
```

❖ MultiView控件的常用属性、方法和事件：

- **ActiveViewIndex**属性：用于获取或设置当前被激活显示的**View**控件的索引值。默认值为-1，表示没有**View**控件被激活。
- **SetActiveView**方法：用于激活显示特定的**View**控件。
- **ActiveViewChanged**事件：当视图切换时被激发。

❖在MultiView控件中，**一次只能将一个View控件定义为活动视图**。如果某个View控件定义为活动视图，它所包含的子控件则会呈现到客户端。可以使用MultiView控件的ActiveViewIndex属性或SetActiveView方法定义活动视图。如果ActiveViewIndex为-1，则MultiView控件不向客户端呈现任何内容。

【例4-18】演示View和MultiView控件的使用。

设计视图

按书名、类别或出版社搜索？
☐ 书名 ☐ 类别 ☐ 出版社

MultiView1

View1

输入书名:

View2

输入类别:

View3

输入出版社:

浏览器显示

localhost:51589/MultiViewDen x +

localhost:51589/MultiViewDemo.aspx

按书名、类别或出版社搜索？
☒ 书名 ☐ 类别 ☐ 出版社

输入书名:

```
protected void RadioButtonList1_SelectedIndexChanged(object sender, EventArgs e)
{
    MultiView1.ActiveViewIndex = RadioButtonList1.SelectedIndex;
}
```

MultiViewDemo.aspx 70

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



11.Calendar控件

❖ **Calendar**控件用于创建日历。在该控件上用户可以方便准确地输入日期。

❖ 声明**Calendar**控件的语法格式：

```
<asp:Calendar ID="控件名" runat="server">  
</asp:Calendar>
```

与日期选取及设置的有关属性：

➤ **SelectionMode属性**：设置日期选择模式，可设为Day(只能选择某一天)、None(不能选择日期，只能显示日期)、DayWeek(可以选择整星期或某天)、DayWeekMonth(可以选择某月、某星期或某天)

➤ **SelectedDate属性**：用户选中的日期。

➤ **SelectedDates属性**：用户选择的多个日期，是一个数组。

Calendar控件具有SelectionChanged事件，当用户选择日期时，会驱动该事件

【例4-19】演示Calendar控件的使用。

localhost:51589/CalendarDemo.aspx

localhost:51589/CalendarDemo.aspx

自动套用格式

入住酒店日期

≤	20	
一	二	三
25	26	27
1	2	3
8	9	10
15	16	17
22	23	24
29	30	1

选择架构(S):

- 移除格式设置
- 简明型
- 专业型 1
- 专业型 2
- 传统型
- 彩色型 1
- 彩色型 2

预览:

2019年4月								
<	一	二	三	四	五	六	日	>
	25	26	27	28	29	30	31	
	1	2	3	4	5	6	7	
	8	9	10	11	12	13	14	
	15	16	17	18	19	20	21	
	22	23	24	25	26	27	28	
	29	30	1	2	3	4	5	

五 周六 周日

五	周六	周日
9	30	31
6	6	7
2	13	14
9	20	21
6	27	28
3	4	5

确定 取消 应用(A)