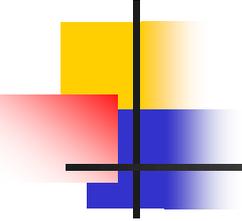


# 第5章

## ASP.NET服务器控件

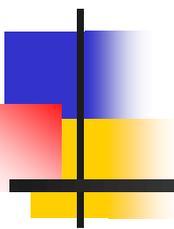
# 5



# Table of contents

---

- 5.1 ASP.NET页面的生命周期
- 5.2 服务器控件概述
- 5.3 标准的Web服务器控件
- 5.4 验证控件
- 5.5 用户控件
- 5.6 实例

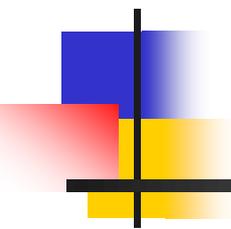


## 5.1 ASP.NET页面的生命周期

---

# Web页面的生命周期





## 5.2 服务器控件概述

---

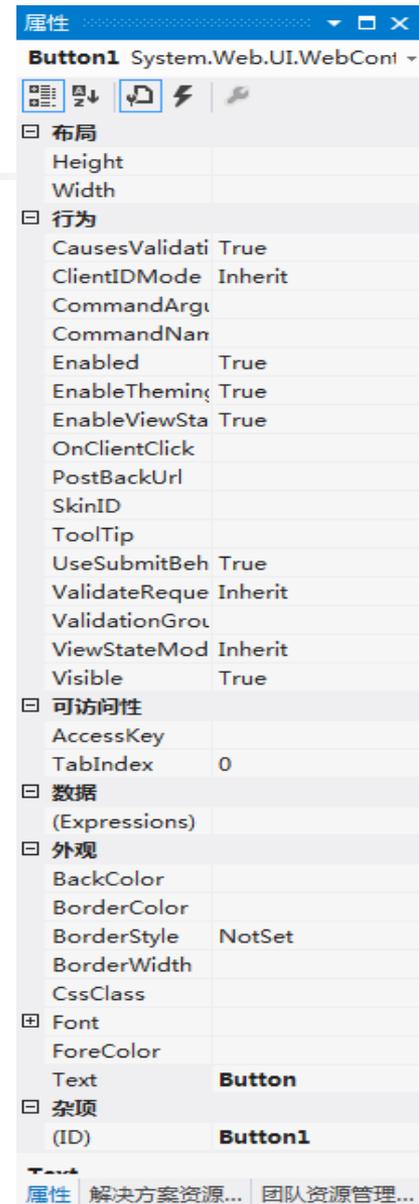
## 5.2.1 服务器端控件的共同属性

- 基本语法:

```
<asp:控件名 id="控件标识" runat="server"  
    属性1=值1, ... , 属性n=值n />
```

- Web服务器控件一般都有规范一致的属性。

- ForeColor、BackColor、BorderColor
- Height、Width、BorderWidth
- Style、CssClass
- Enabled
- TabIndex、ToolTip
- Visible





## 示例

### //判断用户页面是否被提交过

```
<% @ Page Language="C#" %>
<script language="C#" runat="server">
protected void Page_Load(Object sender, EventArgs e){
    if (!IsPostBack) {
        lblMessage.Text = "第一次访问！ ";
    }
    else{
        lblMessage.Text = "页面被提交了！ ";
    }
}
void SubmitBtn_Click(Object sender, EventArgs e) {
}
</script>

<form runat=server>
    <asp:button id="btnSubmit" Text="提交"
        OnClick="SubmitBtn_Click" runat="server" />
    <asp:Label id="lblMessage" runat="server"/>
</form>
```

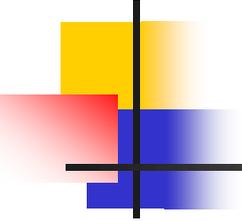


## //案例名称：服务器端动态页面

```
<% @ Page Language="C#" %>
<script language="C#" runat="server">
    protected void Button_Click(object sender, EventArgs e){
        Message.Text = "你的名字是： " + Name.Text;
    }
</script>

<form method="post" runat="server">
    姓名： <asp:textbox id="Name" runat="server" />
    <asp:button text="确定" onclick="Button_Click" runat="server" />
<br><br><br>
<asp:label id="Message" runat="server" />
</form>
```





## 5.2.2 服务器控件的共有事件

---

- 比较常用的页面事件为：
  - **Page\_Load():** 在页面被加载的时候，自动调用该事件。
  - **控件事件:** 由用户在客户端浏览器上触发的各种事件。
  - **Page\_Unload():** 当页面从内存中被卸载的时候，自动调用该事件。

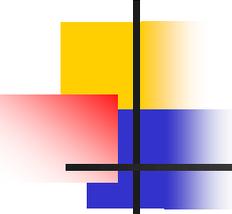
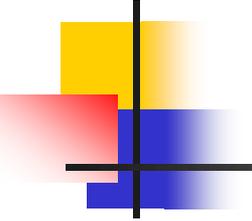


表5-1 服务器控件共有的事件

事件	说明
DataBinding	当控件上的DataBind方法被调用，并且该控件被绑定到一个数据源时被激发
Disposed	从内存中释放一个控件时激发
Init	控件被初始化时激发
Load	把控件装入页面时会激发，该事件在Init后发生
PreRender	控件准备生成它的内容时激发
Unload	从内存中卸载控件时激发



## 5.2.3 服务器控件的分类

---

- **Web 服务器端控件**

Web Server Controls – 新的ASP .NET标签

- **HTML 服务器端控件**

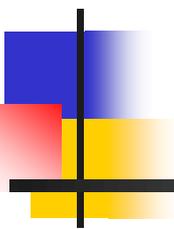
HTML Server Controls - 这是对HTML标记的扩展，每个HTML控件都和原来的HTML标记一一对应。

- **验证控件**

Validation Server Controls - 用于验证用户输入。

- **用户控件**

User Controls - 由用户创建，可以嵌入到Web窗体中的控件。

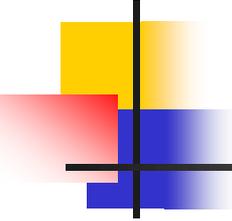


## 5.3 标准的Web服务器控件

---

通常分为4类控件：

- 文本输入与显示控件
- 控制权转移控件
- 选择控件
- 容器控件



## 5.3.1 文本输入和显示的控件

---

- **TextBox (文本) 控件**

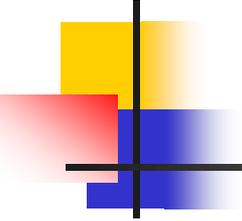
```
<asp:TextBox id="控件名" runat="server"  
    Text="显示文字"  
    TextMode="SingleLine|MultiLine>Password" />
```

属性 TextMode=SingleLine	单行输入框 (默认)
TextMode=MultiLine	多行输入框
TextMode>Password	密码输入框



## 文本控件示例

```
1:<%@page language="C#"%>
2:<script language="C#" runat="server">
3:void WriteInfo(object sender, EventArgs e)
4:{
5:    string strtemp;
6:    strtemp="<font color=red>" + title.Text + "</font><br>";
7:    strtemp+=content.Text.Replace(Convert.ToString(Convert.ToChar(13)),"<br>");
8:    labContent1.Text=strtemp;
9;}
10:</script>
11:<html>
12:<body>
13:<form runat="server">
14:标题<br>
15:<asp:textbox runat=server id=title maxlength=20 />
16:<br>内容<br>
17:<asp:textbox runat=server id=content columns=20 rows=4 textmode=multiline
wrap=true/><br>
18:<button runat=server onserverclick="WriteInfo">提交</button><br>
19:<br><asp:label runat=server id=labContent1/><br>
20:</form>
21:</body>
22:</html>
```



# 标签、图像控件

- Label (标签)

```
<asp:Label id="控件名称" Text="显示文字" runat="server" />
```

表7-3 Label控件的属性和事件

属性/事件	说明
Text属性	获取或设置 Label 控件中的数据
TextChanged事件	用户输入信息后离开 TextBox控件时，引发程序员可以处理的此事件。

例如：

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

# 标签、图像控件

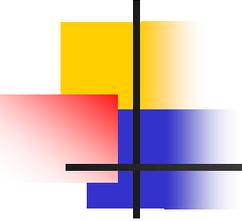
## ■ Image（图像）

```
<asp:Image id="控件名称" ImageUrl="myimage.jpg"  
ImageAlign="left" width=".." height=".." runat="server" />
```

表7-4 Image控件的属性

属性	说明
AlternateText	在图像无法显示时显示的替换文字
DescriptionUrl	包含更详细的图像说明的URL
ImageAlign	图像的对齐方式
ImageUrl	要显示图像的URL

例如： `<asp:Image ID="Image1" runat="server" AlternateText="图片连接失效"  
ImageUrl="http://www.shangducms.com/images/cms.jpg" />`



## 5.3.2 控制权转移控件

---

- Button 控件 - 显示标准 HTML 窗体按钮
- LinkButton 控件 - 在按钮上显示超文本链接
- ImageButton 控件 - 显示图像按钮
- Hyperlink 控件 - 在某些文本上显示超文本链接

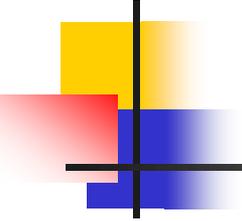
# 1. 按钮控件

## Button, LinkButton, ImageButton

- `<asp:Button ID="Button1" runat="server" Text="click me" />`
- `<asp:LinkButton ID="LinkButton1" runat="server">click me</asp:LinkButton>`
- `<asp:ImageButton ID="ImageButton1" runat="server" ImageURL="a.bmp" />`

表7-6 Button, ImageButton, LinkButton的特殊属性

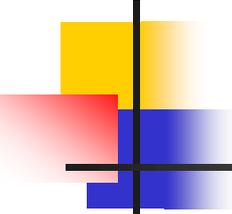
控件名称	属性	说明
Button	UseSubmitBehavior属性	指示按钮是否呈现为提交按钮
	Text属性	在按钮上显示的文本
ImageButton	ImageAlign属性	图像的对齐方式
	PostBackURL属性	单击按钮时所发送到的URL
	AlternateText属性	在图像无法显示时显示的替换文字
	ImageURL属性	要显示的图像的URL
LinkButton	Text属性	要为该链接显示的文本
	PostBackURL属性	单击按钮时所发送到的URL



## 2. 超链接控件（HyperLink）

---

```
<asp:HyperLink id="控件名称" runat="server"  
    Text="显示文字"  
    ImageUrl="要显示的图像的 URL"  
    NavigateUrl="定位到的URL地址"  
    Target="目标框架，默认为本框架，_blank为新窗口"  
>
```



## 5.3.3 选择控件

---

- **单选控件 RadioButton** --- 用于在一个选项列表中选择一个选项，使用时通常会与其它 **RadioButton** 控件组成一组，以提供一组互斥的选项。
- **复选框控件 CheckBox** --- 用于在选中和清除这两种状态间切换。
- **下拉列表控件 DropDownList** --- 允许用户从预定义的列表中选择一项。
- **列表控件 ListBox** --- 允许用户从预定义列表中选择一项或多项。

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

- **单选控件**可以为用户选择某一个选项。
- 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。

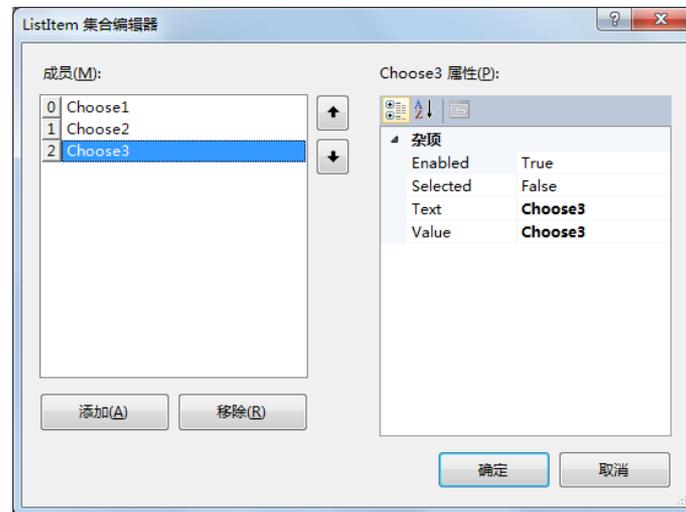
```
<form id="form1" runat="server">  
  <asp:RadioButton ID="RB1" runat="server" GroupName="chos"  
    Text="Choose1" AutoPostBack="true"/>  
  <asp:RadioButton ID="RB2" runat="server" GroupName="chos"  
    Text="Choose2" AutoPostBack="true"/>  
  <asp:RadioButton ID="RB3" runat="server" GroupName="chos"  
    Text="Choose3" AutoPostBack="true"/>  
</form>
```



## RadioButton 示例

```
1:<%@page language="C#"%>
2:<script language="C#" runat="server">
3:void RadClick(object sender, EventArgs e)
4:{
5:    string strtemp;
6:    strtemp="<br>";
7:    if (rb1.Checked) strtemp+="3+2=5<br>";
8:    if (rb2.Checked) strtemp+="3-2=1<br>";
9:    if (rb3.Checked) strtemp+="3*2=6<br>";
10:   if (rb4.Checked) strtemp+="3/2=1.5<br>";
11:   labContent1.Text=strtemp;
12:}
13:</script>
14:<html>
15:<body>
16:<form runat="server" >
17:<asp:radio button id=rb1 runat="server" groupname="rg" text="3+2=?" checked=true</><br>
18:<asp:radio button id=rb2 runat="server" groupname="rg" text="3-2=?" checked=false/><br>
19:<asp:radio button id=rb3 runat="server" groupname="rg" text="3*2=?" checked=false/><br>
20:<asp:radio button id=rb4 runat="server" groupname="rg" text="3/2=?" checked=false/><br>
21:<asp:button id="butsubmit" runat=server text="提交" onclick="RadClick"/>
22:<hr>
23:<asp:label runat=server id=labContent1/><br>
24:</form>
25:</body>
26:</html>
```

- **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。



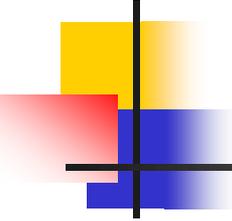
添加项目成员后，系统自动在.aspx页面声明服务器控件代码，代码如下所示。

```
<asp:RadioButtonList ID="RadioButtonList1" runat="server"
    SelectedIndexChanged="RadioButtonList1_SelectedIndexChanged">
    <asp:ListItem>Choose1</asp:ListItem>
    <asp:ListItem>Choose2</asp:ListItem>
    <asp:ListItem>Choose3</asp:ListItem>
</asp:RadioButtonList>
```



## RadioButtonList 实例

```
1:<%@page language="C#"%>
2:<script language="C#" runat="server">
3:void RadClick(object sender, EventArgs e)
4:{
5:    string strtemp;
6:    int inti;
7:    strtemp="<br>";
8:    for(inti=0;inti<radbutlst.Items.Count;inti++)
9:    {
10:        if (radbutlst.Items[inti].Selected)
11:        {
12:            switch(radbutlst.Items[inti].Value)
13:            {
14:                case "+":
15:                    strtemp+="3+2=5<br>";break;
16:                case "-":
17:                    strtemp+="3-2=1<br>";break;
18:                case "*":
19:                    strtemp+="3*2=6<br>";break;
20:                case "/":
21:                    strtemp+="3/2=1.5<br>";break;
22:            }
23:        }
24:    }
25:    labContent1.Text=strtemp;
26:}
27:</script>
28:<html>
29:<body>
30:<form runat="server" >
31:<asp:radiobuttonlist id="radbutlst" runat=server autopostback=true onselectedindexchanged="RadClick"
32:<asp:listitem text="3+2=?" value="+" selected="false"/>
33:<asp:listitem text="3-2=?" value="-" selected="false"/>
34:<asp:listitem text="3*2=?" value="*" selected="false"/>
35:<asp:listitem text="3/2=?" value="/" selected="false"/>
36:</asp:radiobuttonlist>
37:<hr>
38:<asp:label runat=server id=labContent1/><br>
39:</form>
40:</body>
41:</html>
```



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

---

- **复选框控件 (CheckBox)**：同单选控件一样，复选框也是通过Check属性判断是否被选择。不同的是，复选框控件没有GroupName属性。
- **复选组控件 (CheckBoxList)**：同单选组控件相同，.NET服务器控件中同样包括了复选组控件 (CheckBoxList)，拖动一个复选组控件到页面可以添加复选组列表。



以下代码声明了两个复选框控件。

```
<form id="form2" runat="server">  
    <asp:CheckBox ID="CheckBox1" runat="server" Text="Check1"  
        AutoPostBack="true" />  
    <asp:CheckBox ID="CheckBox2" runat="server" Text="Check2"  
        AutoPostBack="true" />  
</form>
```

当双击复选框控件时，系统会自动生成CheckedChanged事件的声明。当复选框控件的选中状态被改变后，会激发该事件。

```
protected void CheckBox1_CheckedChanged(object sender, EventArgs e) {  
    Label1.Text = "选框1被选中";           //当选框1被选中时  
}  
protected void CheckBox2_CheckedChanged(object sender, EventArgs e) {  
    Label1.Text = "选框2被选中";           //当选框2被选中时  
    Label1.Font.Size = FontUnit.XXLarge;  
}
```



## CheckBoxList 示例

```
1:<%@page language="C#" %>
2:<script language="C#" runat="server">
3:void ChkClick(object sender, EventArgs e)
4:{
5:    string strtemp;
6:    int inti;
7:    strtemp="<br>";
8:    for(inti=0;inti<chkboxlst.Items.Count;inti++)
9:    {
10:        if (chkboxlst.Items[inti].Selected)
11:        {
12:            switch(chkboxlst.Items[inti].Value)
13:            {
14:                case "+":
15:                    strtemp+="3+2=5<br>";break;
16:                case "-":
17:                    strtemp+="3-2=1<br>";break;
18:                case "*":
19:                    strtemp+="3*2=6<br>";break;
20:                case "/":
21:                    strtemp+="3/2=1.5<br>";break;
22:            }
23:        }
24:    }
25:    labContent1.Text=strtemp;
26:}
27:</script>
28:<html>
29:<body>
30:<form runat="server" >
31:<asp:checkboxlist id="chkboxlst" runat=server autopostback=true onselectedindexchanged="ChkClick">
32:<asp:listitem text="3+2=?" value="+" selected="false"/>
33:<asp:listitem text="3-2=?" value="-" selected="false"/>
34:<asp:listitem text="3*2=?" value="*" selected="false"/>
35:<asp:listitem text="3/2=?" value="/" selected="false"/>
36:</asp:checkboxlist>
37:<hr>
38:<asp:label runat=server id=labContent1/><br>
39:</form>
40:</body>
41:</html>
```

## 3. 下拉列表控件（DropDownList）

- 使用DropDownList下拉列表控件，可以有效地避免用户输入无效或错误的信息。 DropDownList控件也可以绑定数据源控件。

```
<asp:DropDownList ID="DropDownList1" runat="server" AutoPostBack="True"
    SelectedIndexChanged="List_Changed" >
    <asp:ListItem>1</asp:ListItem>
    <asp:ListItem>2</asp:ListItem>
    <asp:ListItem>3</asp:ListItem>
</asp:DropDownList>
```

当用户选择相应的项目，使得DropDownList控件选择项发生变化时，则会触发SelectedIndexChanged事件。

```
protected void DropDownList1_SelectedIndexChanged1(object sender, EventArgs e)
{
    Label1.Text = "你选择了第" + DropDownList1.Text + "项";
}
```



## DropDownList 实例

```
<% @ Page Language="C#" %>
<script language="C#" runat="server">
void Image_Changed(Object Sender, EventArgs e) {
    lblUid.Text = lstAddr.SelectedItem.Value;
}
</script>
<form runat="server">
    选择你要去的城市:
    <asp:DropDownList id="lstAddr" AutoPostBack="True"
        OnSelectedIndexChanged="Image_Changed"
        runat="server">
        <asp:ListItem value="北京" selected="true">北京</asp:ListItem>
        <asp:ListItem value="上海">上海</asp:ListItem>
    </asp:DropDownList>
    <asp:Label id="lblUid" runat="server"/>
</form>
```



## 4. ListBox列表控件

- ListBox控件与DropDownList控件的属性基本相同，只增加了2个属性：**Rows** 和 **SelectionMode**。ListBox控件可以通过SelectionMode属性指定用户是否允许多项选择。

```
<asp:ListBox ID="ListBox1" runat="server" AutoPostBack="True"
  onselectedindexchanged="ListBox1_SelectedIndexChanged">
  <asp:ListItem>第1项</asp:ListItem>
  <asp:ListItem>第2项</asp:ListItem>
  <asp:ListItem>第3项</asp:ListItem>
</asp:ListBox>
<asp:Label ID="Label1" runat="server" Text="你所选的项目为: "> </asp:Label>
```

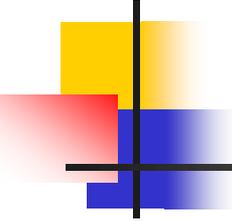
同样，SelectedIndexChanged也是ListBox控件中最常用的事件。

```
protected void ListBox1_SelectedIndexChanged(object sender, EventArgs e) {
    Label1.Text = "你选择了" + ListBox1.Text ;
}
```



## Listbox 实例

```
1:<%@page language="C#"%>
2:<script language="C#" runat="server">
3:void LstClick(object sender, EventArgs e)
4:{
5:    string strtemp;
6:    int inti;
7:    strtemp="<br>";
8:    for(inti=0;inti<listbox.Items.Count;inti++)
9:    {
10:        if (listbox.Items[inti].Selected)
11:        {
12:            switch(listbox.Items[inti].Value)
13:            {
14:                case "+":
15:                    strtemp+="3+2=5<br>";break;
16:                case "-":
17:                    strtemp+="3-2=1<br>";break;
18:                case "*":
19:                    strtemp+="3*2=6<br>";break;
20:                case "/":
21:                    strtemp+="3/2=1.5<br>";break;
22:            }
23:        }
24:    }
25:    labContent1.Text=strtemp;
26:}
27:</script>
28:<html>
29:<body>
30:<form runat="server" >
31:<asp:listbox id="listbox" runat=server autopostback=true selectionmode="single" rows=3 onselectedindexchanged="LstClick">
32:<asp:listitem text="3+2=?" value="+" selected="false"/>
33:<asp:listitem text="3-2=?" value="-" selected="false"/>
34:<asp:listitem text="3*2=?" value="*" selected="false"/>
35:<asp:listitem text="3/2=?" value="/" selected="false"/>
36:</asp:listbox>
37:<hr>
38:<asp:label runat=server id=labContent1/><br>
39:</form>
40:</body>
41:</html>
```



## 5.3.4 容器控件

---

有两种类型的容器控件：

- 面板控件（**Panel**） – 面板控件可以作为一组控件的容器，其常用功能就是显示或隐藏一组控件。
- 占位控件（**PlaceHolder**） – 存储动态添加到网页上的服务器控件的容器。



设置Panel控件的HTML代码如下。

```
<form id="form1" runat="server">
  <asp:Button ID="Button1" runat="server" Text="Show" />
  <asp:Panel ID="Panel1" runat="server" Visible="False">
    <br /> The controls in a Panel are in follows.  <br />
    <asp:Label ID="Label1" runat="server" Text="Hello"></asp:Label>
    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
  </asp:Panel>
</form>
```

上述代码创建了一个Panel控件，初始状态为不可见。在Panel控件外有一个Button控件。当用户单击Button控件时，将显示Panel控件。

```
protected void Button1_Click(object sender, EventArgs e) {
    Panel1.Visible = true;
    //Panel控件显示可见
}
```

当页面初次被载入时，Panel控件以及Panel控件内的全部控件都为隐藏。当用户单击Button时，则Panel控件及其内部的控件都为可见。



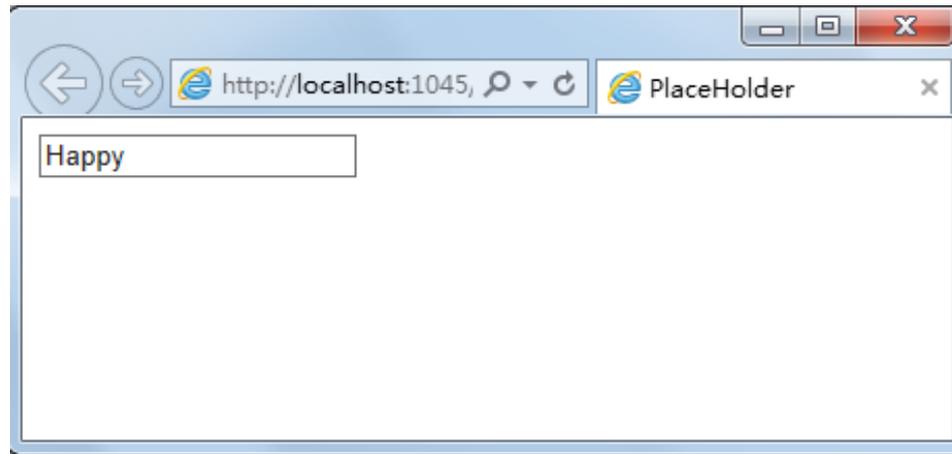
示例

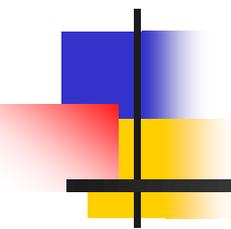
与面板控件Panel控件相同的是，占位控件Placeholder也是控件的容器，但是在HTML页面呈现中本身并不产生HTML

```
<asp:Placeholder ID="Placeholder1" runat="server">
</asp:Placeholder>
```

在CS页面中，允许用户动态地在Placeholder上创建控件。

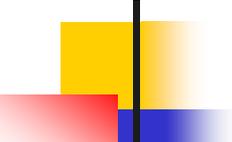
```
protected void Page_Load(object sender, EventArgs e) {
    TextBox text = new TextBox(); //创建一个TextBox对象
    text.Text = "happy";
    this.PlaceHolder1.Controls.Add(text); //为占位控件动态增加一个控件
}
```





## 5.4 验证控件

---



# 验证控件

控件名称	说明
RequiredFieldValidator	确保用户不跳过输入。
CompareValidator	使用比较运算符（小于、等于、大于等）将用户的输入与另一控件的常数值或属性值进行比较。
RangeValidator	检查用户的输入是否在指定的上下边界之间。可以检查数字、字母或日期对内的范围。可以将边界表示为常数。
RegularExpressionValidator	检查输入是否与正则表达式定义的模式匹配。该验证类型允许检查可预知的字符序列，如社会保障号、电子邮件地址、电话号码、邮政编码等中的字符序列。
CustomValidator	使用您自己编写的验证逻辑检查用户的输入。该验证类型允许检查运行时导出的值。
ValidationSummary	以摘要的形式显示页上所有验证程序的验证错误。

## 5.4.1 RequiredFieldValidator（必填验证）

```
<asp:RequiredFieldValidator id="控件名称"  
    runat="server"  
    ControlToValidate="要检查的控件名称"  
    ErrorMessage="出错信息"  
    Display="Dynamic | Static | None" />
```



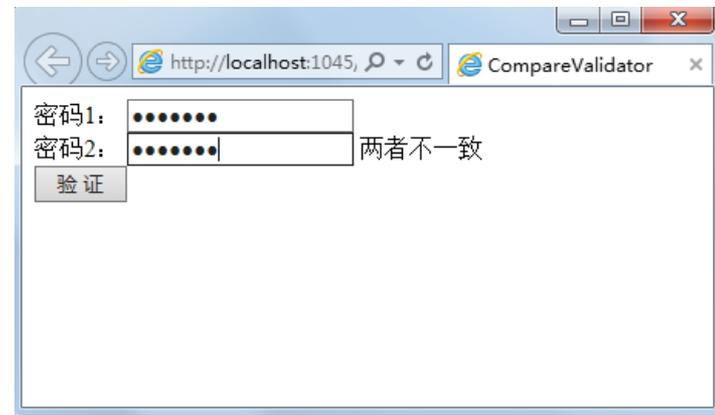
```
<ASP:RequiredFieldValidator  
    id="Validator1" Runat="Server"  
    ControlToValidate="txtName"  
    ErrorMessage="姓名必须输入"  
    Display="Static"> *姓名必须输入  
</ASP:RequiredFieldValidator>
```

## 5.4.2 CompareValidator（比较验证）

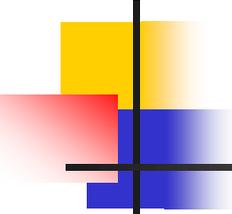
```
<asp:CompareValidator id="控件名称"  
  ControlToValidate="要验证的控件ID"  
  ControlToCompare="要比较的控件ID"  
  Type="String | Integer | Date |..., 比较类型"  
  Operator="Equal | NotEqual | GreaterThan  
  | GreaterThanEqual | LessThan | LessThanEqual, 分别表示=、  
  <>、>、>=、<、<=""  
  ErrorMessage="出错提示信息"  
  Display="Dynamic | Static | None"  
  runat="server" />
```



## 比较验证控件示例



```
<%@ Page clienttarget=downlevel Language="C#" %>
<form runat="server">
<b>密码1:</b>
<asp:TextBox id="passwd1" TextMode="Password" runat="server" />
<br><b>密码2:</b>
<asp:TextBox id="passwd2" TextMode="Password" runat="server" />
<asp:CompareValidator id="CV"
  ControlToValidate="passwd1"
  ControlToCompare="passwd2"
  Type="String"
  Operator="Equal"
  Display="static" runat="server"> 两者不一致!
</asp:CompareValidator>
<br><br><asp:Button id="Validate" text="验证" runat="server" />
</form>
```



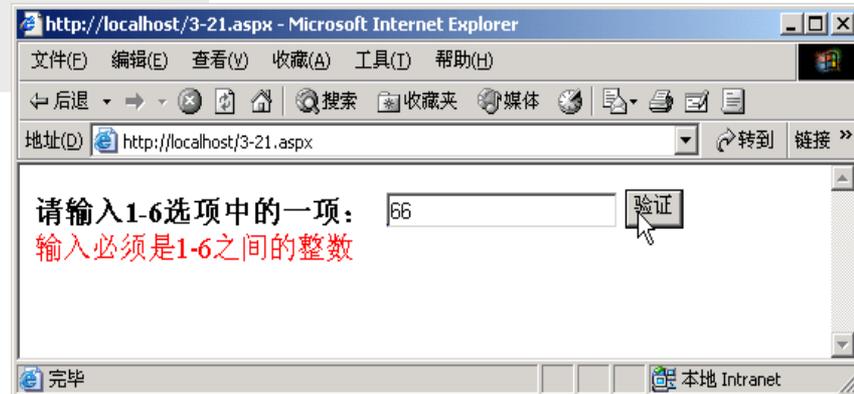
## 比较验证控件的属性

属性	说明
ControlToValidate	要验证的控件ID
ControlToCompare	用于进行比较的控件ID
Type	表示要比较的两个值的数据类型，取值有5种： String, Integer, Date, Double, Currency
Operator	表示要使用的比较运算符
ErrorMessage	出错提示信息
Text	当验证的控件无效时显示的验证程序文本
Display	验证程序的显示方式。取值包括3种。 Dynamic：不出错的时候该控件不占用页面位置。 Static：不出错的时候该控件占用页面位置。 None：不显示出错信息
SetFocusOnError	控件无效时，验证程序是否在控件上设置焦点。默认为False
ValueToCompare	用于进行比较的值

## 5.4.3 RangeValidator (范围验证)

```
<form runat="server">  
<b>请输入1-6选项中的一项: </b>  
<asp:TextBox id="Choice" runat="server" />  
<asp:Button text="验证" runat="server" /><br>  
<asp:RangeValidator id="rv"  
    ControlToValidate="Choice"  
    MinimumValue="1" MaximumValue="6"  
    ErrorMessage="输入必须是1-6之间的整数"  
    Type="Integer"  
    Display="dynamic" runat="server">  
</asp:RangeValidator>  
</form>
```

**Type** 是要比较值的数据类型，取值有5种：  
String, Integer, Date, Double, Currency



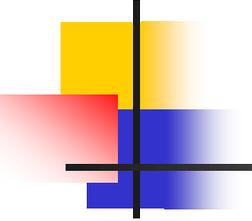
# 5.4.4 RegularExpressionValidator

## (正则表达式验证)

示例

```
<form runat="server">  
<b>请数字开头后接一个大写字母: </b>  
<asp:TextBox id="Email" runat="server" />  
<asp:Button text="验证" runat="server" /><br>  
<asp:RegularExpressionValidator id="rev"  
    ControlToValidate="Email"  
    ValidationExpression="\d[A-Z]*"  
    Display="Dynamic"  
    runat="server"> 输入错误  
</asp:RegularExpressionValidator>  
</form>
```





# 关于正则表达式

---

- 所谓正则表达式，就是比通常用的\*和?通配符更复杂的一种字符串定义规则。 例如：
  - `[a-zA-Z]{3,6}` 可以输入3到6个字母。  
[]表示其中是可输入字符，a-z表示小写字母，A-Z表示大写字母。  
{ }表示字符数目，{3,6}表示3到6个。
  - `[0-9]{6}` 可以输入6个数字。0-9表示所有的数字。
  - `.{0,}` 可以输入任意个字符。
  - `{n}`表示出现n次。 `{n,}`表示n个以上， `{}`表示1个。

## 限定符举例:

- \*: 匹配所限定的表达式0次或多次
- +: 匹配所限定的表达式1次或多次
- ?: 匹配所限定的表达式0次或1次
- {n, m}: 限定表达式出现n到m次
- ^ \$: 以限定的表达式开头和结尾
- |: 或匹配
- \: 匹配限定符本身

例如: `\w*`

例如: `[a-zA-Z]+`

例如: `a?`

例如: `ABC {5, 8}`

例如: `^J\w {4}$`

例如: `(abc) | (ABC)`

例如: `\*`

### ■ 实际例子:

- **电话验证**: `[0-9]{3,4}-[0-9]{7,8}` 如0371-92345678或010-12345678。
- **18位身份证验证**: `[0-9]{6}[12][0-9]{3}[01][0-9][0123][0-9][0-9]{3}[12]` 主要考虑生日和最后一位。
- **E-mail验证**: `.{1,}@.{1,}\.[a-zA-Z]{2,3}`

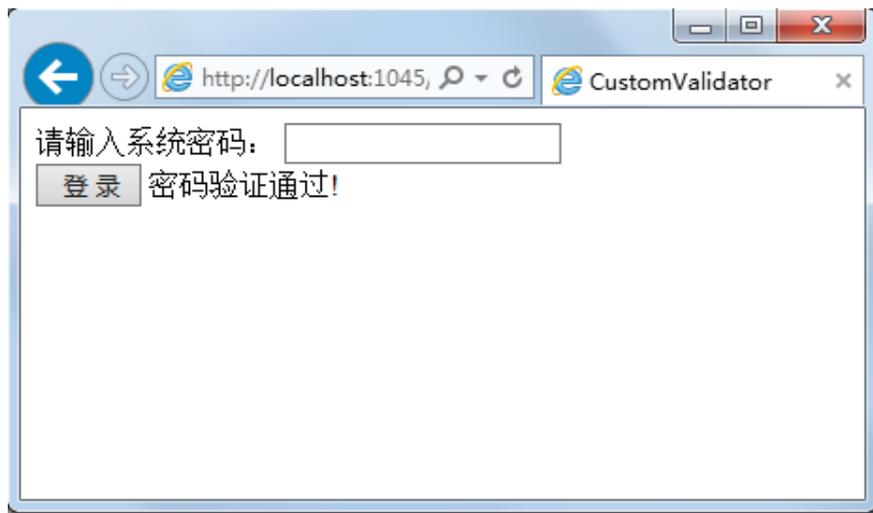
## 5.4.5 CustomValidator（自定义验证控件）

CustomValidator控件与其他验证控件的使用方法基本一致。其特殊之处在于它提供了一个ServerValidate事件，可以在此编写数据验证的代码。

```
<form id="form4" runat="server">
<div>请输入系统密码:
<asp:TextBox id="passwd4" TextMode="Password" runat="server" />
<asp:CustomValidator id="CustomValidator1" runat="server"
    ControlToValidate="passwd4" ErrorMessage="输入的密码不正确“
    onservvalidate="CustomValidator1_ServerValidate" >
</asp:CustomValidator>
<br />
<asp:Button id="btnLogin" runat="server" text=" 登 录" onclick="btnLogin_Click" />
<asp:Label ID="lblmessage" runat="server" Text=""></asp:Label>
</form>
```

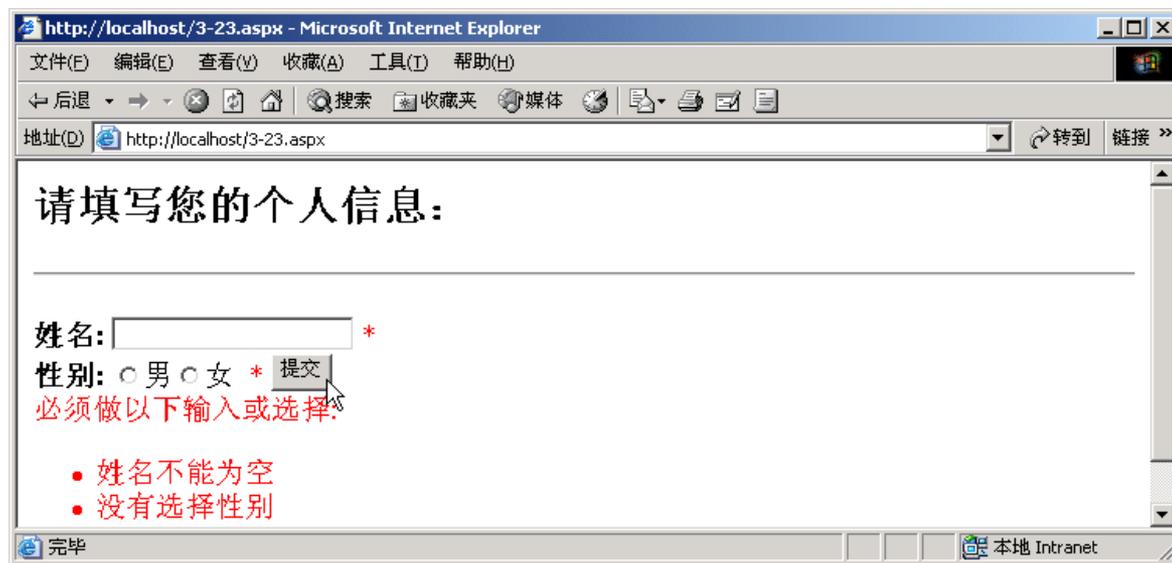
ServerValidate事件的处理程序如下。

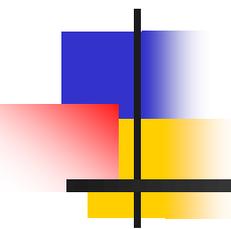
```
protected void CustomValidator1_ServerValidate(object source, ServerValidateEventArgs args) {  
    string strVal = args.Value.ToUpper();  
    if ( strVal.Equals("ADMINISTRATOR") )  
        args.IsValid = true;  
}  
else {  
    args.IsValid = false;  
}  
}  
protected void btnLogin_Click(object sender, System.EventArgs e) {  
    if (CustomValidator1.IsValid ) {  
        lblmessage.Text = "密码验证通过!";  
    }  
}
```



## 5.4.6 ValidationSummary (验证总结)

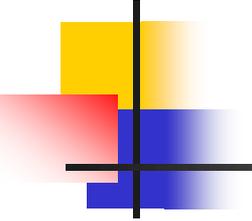
- 验证总结控件本身并不提供任何验证，但是它可以和前面讲到过的其他控件一起使用，并集中给出验证结果。验证总结控件所显示的错误列表中的错误信息都是由其他验证控件的 ErrorMessage 属性提供的。





## 5.5 用户控件

---



## 5.5.1 用户控件概述

---

- ◆ 用户控件是一种自定义的、可复用的组合控件。
- ◆ 可以像设计**Web**窗体一样设计用户控件，并设置属性和方法。
- ◆ 用户控件可以单独编译，但不能单独运行，必须嵌入到**Web**页面中才能运行。
- ◆ 用户控件在第一次请求时被编译并存储在服务器内存中，从而缩短以后请求的响应时间。
- ◆ 用户控件的文件扩展名为`.ascx`和`.ascx.cs`。
- ◆ 用户控件不包含`<html>`、`<body>`和`<form>`标记。
- ◆ 用户控件不含`@Page`指令，而是`@Control`指令。

## 5.5.2 创建用户控件

要创建一个用户控件，可以有两种方式：

(1) 在**Visual Studio 2012**环境中直接创建用户控件；

- 在“添加新项”对话框，选择“**Web**用户控件”，添加一个新的用户控件。命名为**myControl.ascx**。
- 向设计页面**myControl.ascx**中，添加各种**Web**控件并修改其属性。
- 在代码文件**myControl.ascx.cs**窗口中，给编写各种子控件的事件响应代码。

(2) 将已经设计完成的**Web**窗体页面改为用户控件。

- 删除**ASPX**文件中的**<HTML>**、**<BODY>**和**<FORM>**等标记，因为它们可能与包含页面有冲突。
- 将**@page**指令改为**@control**指令。
- 在代码文件**.cs**中定义的类的基类由**Page**类改为**UserControl**类。
- 将文件的扩展名改为**.ascx**

## 5.5.3 用户控件的使用



直接在Web窗体页面添加用户控件

程序名称: **test.ascx**

`<p>用户自定义控件</p>`



程序名称: **test.aspx**

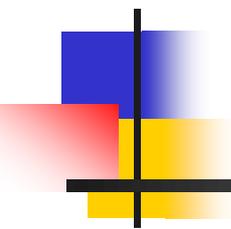
```
<%@ Page Language="C#" %>
```

```
<%@ Register TagPrefix="myCo" TagName="myTxt" Src="test.ascx" %>
```

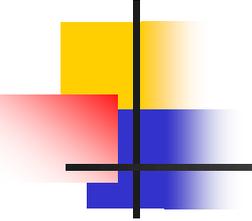
.....

```
<myCo:myTxt runat="server" />
```

## 5.6 实例：商品信息录入及其有效性验证



---



## 5.6.1 设计说明

---

本实例是一个添加书籍信息的页面，要求输入书名、价格、作者、出版社、出版时间、ISBN、所属分类、入库数量等。注意：书籍名称、价格、ISBN、类别、入库数量等都是必须输入的，要使用必须输入验证控件RequiredFieldValidator。ISBN号、价格均使用正则表达式控件RegularExpressionValidator，出版时间使用范围验证控件RangeValidator

## 5.6.2 程序实现



The screenshot shows a web browser window with the address bar displaying 'http://localhost:2303'. The page title is '商品信息录入及其有...'. The main content is a form titled '图书添加' (Add Book). The form contains several input fields and a submit button, with some fields having red error messages.

图书名称:	<input type="text"/>	此项不能为空
ISBN:	<input type="text" value="978756009056"/>	请输入13位ISBN号
价格:	<input type="text" value="13q"/>	价格输入错误
作者:	<input type="text" value="Kevin"/>	
出版社:	<input type="text" value="机械工业出版社"/>	
出版时间:	<input type="text" value="2015-3-32"/>	请输入有效的日期格式
所属类别:	<input type="checkbox"/> 教育 <input type="checkbox"/> 金融 <input type="checkbox"/> 历史 <input checked="" type="checkbox"/> 科技 <input type="checkbox"/> 文学	
新增册数:	<input type="text" value="300"/>	
图书封皮:	<input type="text" value="C:\Users\Public\Pictures\"/> <input type="button" value="浏览..."/>	