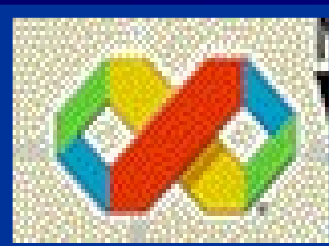


# Welcome to **ASP.net**



Microsoft®  
Visual Studio® .NET

# Validating User Input

## ■ Required Field Validator

- `<asp:TextBox`
- `id="txtUserName"`
- `Runat="Server" />`
- `<asp: RequiredFieldValidator`
- `ControlToValidate="txtUserName"`
- `Text="You must enter a username."`
- `Runat="Server" />`

# *Using the RangeValidator Control*

- `<asp:TextBox id="tbox1" runat="server" />`
- `<br /><br />`
- `<asp:Button Text="Submit" runat="server" />`
- `<br />`
- `<asp:RangeValidator`
- `ControlToValidate="tbox1"`
- `MinimumValue="1"`
- `MaximumValue="100"`
- `Type="Integer"`
- `EnableClientScript="false"`
- `Text="The value must be from 1 to 100!"`
- `runat="server" />`

# Regular Expression Validator

- **<asp:TextBox**
- **id="txtProductID"**
- **Runat="Server" />**
  
- **<asp:RegularExpressionValidator**
- **ControlToValidate="txtProductID"**
- **Text="Invalid Product ID."**
- **ValidationExpression="[A-Z]{4}"**
- **Display="Dynamic"**
- **Runat="Server" />**

# ***Compare Validator Control***

- **<asp:TextBox**
- **id="txtSDate"**
- **Runat="Server" />**
- **<p>**
- **<b>Ending Date:</b>**
- **<br>**
- **<asp:TextBox**
- **id="txtEDate"**
- **Runat="Server" />**
- **</p>**
- **<asp:CompareValidator**
- **ControlToValidate="txtEDate"**
- **ControlToCompare="txtSDate"**
- **Text="Ending Date must be greater than starting date."**
- **Display="Dynamic"**
- **Operator="GreaterThan"**
- **Type="Date"**
- **Runat="Server" />**
  
- **<asp:Button**
- **Text="Submit"**
- **OnClick="Button\_Click"**
- **Runat="Server" />**

# ***CompareValidator Control property***

Property	Definition
Id	This property gets or sets the identifier for the control. This identifier is used via programming to access the control's properties
ControlToValidate	Gets or sets the name of the control to validate.
ControlToCompare or ValueToCompare	Gets or sets the identifier of the control on the Web page to compare with. Gets or sets a specific value that is

# CompareValidator Control property

display	Gets or sets the display appearance of the validator control on the Web page.
ErrorMessage	Gets or sets the error message displayed for this control.
Runat	Specifies that this control runs on the server.

# Using the RangeValidator Control

Property	Definition
Id	This property gets or sets the identifier for the control. This identifier is used via programming to access the control's properties.
ControlToValidate	Gets or sets the name of the control to validate.
MaximumValue	Gets or sets the maximum value of the validation range.
MinimumValue	Gets or sets the minimum value of the validation range.



# Using the RangeValidator Control

Property	Definition
MaximumControl	Gets or sets the ID of the control that specifies the maximum value of the validation range.
MinimumControl	Gets or sets the ID of the control that specifies the minimum value of the validation range.
display	Gets or sets the display appearance of the validator control on the Web page.
ErrorMessage	Gets or sets the error message displayed for this control.

# Using the RangeValidator Control

type	Gets or sets the data type to determine how the values should be compared. For instance, string to string or int to int.
Runat	Specifies that this control runs on the server.

# ASP.NET – CheckBox

```
<asp:CheckBox  
    id="chkMailingList"  
    Text="Do you want to hear from  
us?"  
    Runat="Server" />
```

# ASP.NET – CheckBoxList

```
<asp:CheckBoxList  
    id="chkFavouriteMusic"  
    Runat="Server">  
    <asp:ListItem Text="Blues" />  
    <asp:ListItem Text="Rock & Roll" />  
    <asp:ListItem Text="Pops" />  
    <asp:ListItem Text="Altanatives" />  
</asp:CheckBoxList>
```

# ASP.NET – CheckBoxList

```
<script language="vb" runat="server">  
Sub Button_Click(a as Object, e as EventArgs)  
Dim itmMusic As ListItem  
Dim strMusic As String  
For each itmMusic in chkFavouriteMusic.Items  
If itmMusic.Selected then  
    strMusic &= "<li>" & itmMusic.Text  
End If  
Next  
    lblFavouriteMusic.Text = strMusic  
End Sub  
</script>
```

# ASP.NET – DropDownList

Open and Close tag

```
<asp:DropDownList  
    id="DropdownName"  
    Runat="Server">  
    <asp:ListItem Text="Singapor" />  
    <asp:ListItem Text="Malaysia" />  
    <asp:ListItem Text="Vietname" />  
    <asp:ListItem Text="Laos"  
    selected="true" />  
</asp:DropDownList>
```

# ASP.NET – DropDownList

```
■ <script language="vb"
  runat="server">
  Sub Button_Click(a as Object,e as
    EventArgs)
  `you must have lblMessage in your form
  lblMessage.Text =
    myDropDownList.SelectedIndex & "- "
    & myDropDownList.SelectedItem.Text
  End Sub
</script>
```

# ASP.NET –ListBox

```
<asp:ListBox  
    id="ListName"  
    Runat="Server">  
    <asp:ListItem Text="Windows XP" />  
    <asp:ListItem Text="Windows 2k" />  
    <asp:ListItem Text="Windows Vista"  
    />  
    <asp:ListItem Text="Linux"  
    selected="true" />  
</asp:ListBox>
```



# ASP.NET –ListBox

```
<script language="vb"  
  runat="server">
```

```
Sub Button_Click(a As Object, e As  
  EventArgs)
```

```
lblMessage.Text =  
  myListBox.SelectedItem.Text
```

```
End Sub
```

```
■ </script>>
```

# ASP.NET –ListBox

```
<asp:ListBox  
    id="ListName"  
    SelectionMode="Multiple"  
    Runat="Server">  
    <asp:ListItem Text="Windows XP" />  
    <asp:ListItem Text="Windows 2k" />  
    <asp:ListItem Text="Windows Vista"  
    />  
    <asp:ListItem Text="Linux"  
    selected="true" />  
</asp:DropDownList>
```

# ASP.NET –ListBox

```
Sub Button_Click(a As Object, e As EventArgs)  
  Dim myList as ListItem  
  Dim strMessage as String  
  For each myList in myListBox.Items  
    If myList.Selected then  
      strMessage &= "<li>" &  
        myListBox.SelectedItem.Text  
    End If  
  Next  
  lblMessage.Text = strMessage  
End Sub
```