



CHƯƠNG 2: ASP SERVER CONTROL

Lập trình aSp.net



I. HTML Server Control

- Các HTML control thông thường như `<h1>`, `<a>`, `<input>` sẽ không được xử lý bởi server mà được gửi trực tiếp cho browser để hiển thị
- Các *HTML control* có thể được xử lý ngay tại phía server bằng cách chuyển chúng thành các *HTML server control*.

I. HTML Server Control

- Chuyển một HTML control thành một HTML server control bằng cách thêm thuộc tính **runat="server"** vào trong các tag HTML
- Cú pháp:

```
<TagName runat="server"></TagName>
```

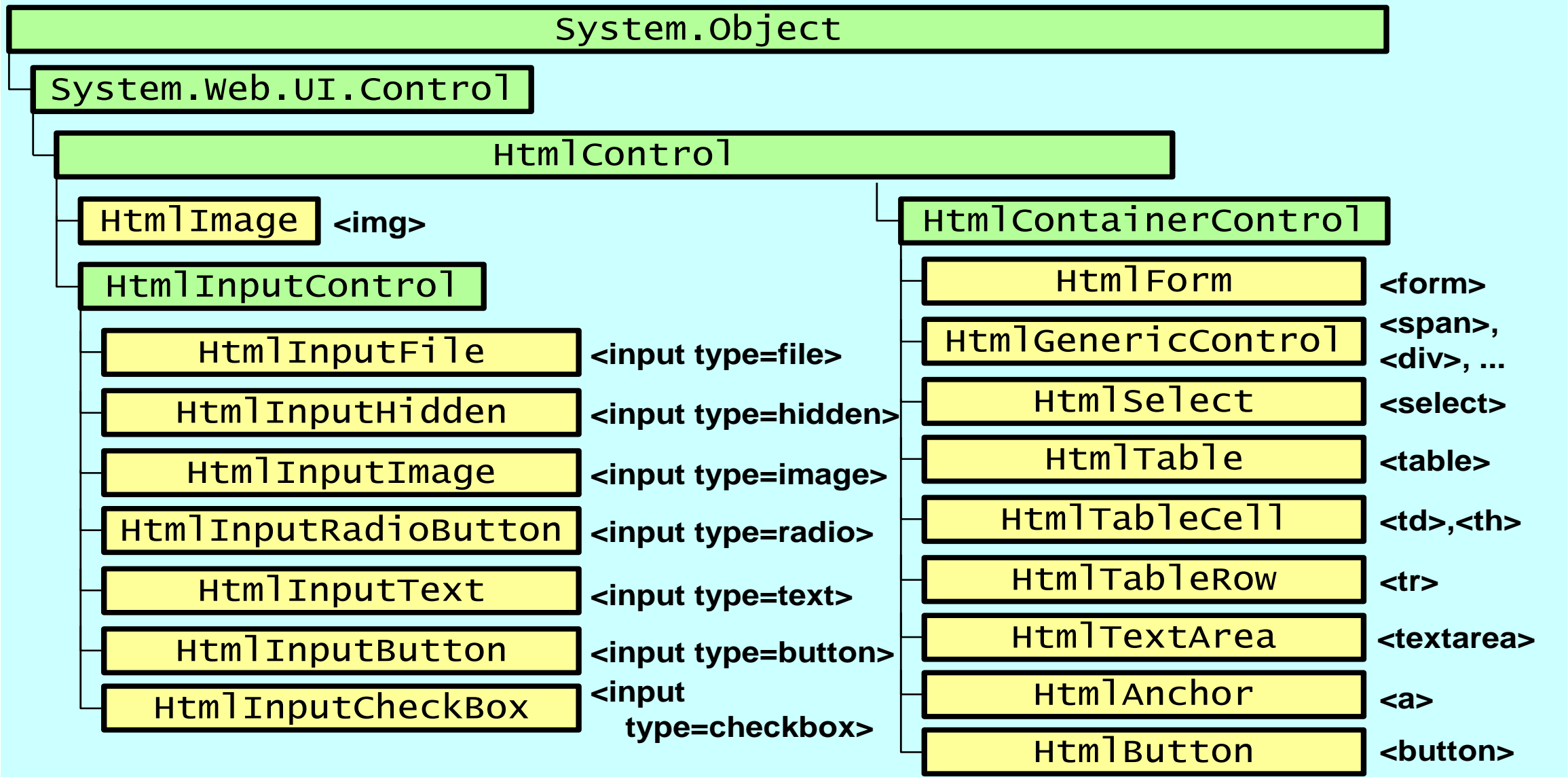
Ví dụ:

```
< type="text" name="txtName" runat="server"/>
```

I. HTML Server Control

- Tất cả HTML Server Control phải được đặt trong tag <form> với thuộc tính runat = “server”
- HTML Server Control tồn tại bên trong không gian tên **System.Web.UI.HtmlControls**

■ Hệ thống thứ bậc của HTML Server Controls



I. HTML Server Control

- **Các sự kiện (event) của HTML server Control:**

- onServerClick
- onServerChange
- onStartSelect,...

Cú pháp:

```
<tagName event="Function_Process"></tagName>
```

- Lưu ý: *Function không có tham số.*

I. HTML Server Control

Ví dụ:

```
<input type="Submit" value="Gui" id="btnSend"  
runat="server" onServerClick="btnSend_ServerClick"/>
```

I. HTML Server Control

- Cách lấy dữ liệu từ các HTML server

Cú pháp:

controlfield_id.Value

- Đối với tag DIV, SPAN dùng thuộc tính **.innerHTML**

I. HTML Server Control

Ví dụ:

```
public void btnSend_ServerClick(object  
    sender, System.EventArgs e)  
{  
    String strName;  
    strName=txtName.Value;  
    Response.Write("Hello :"+strName)  
}
```

Ví dụ

```
void Button_ServerClick(object sender, EventArgs e)
```

```
{
```

```
    MySpan.InnerHtml = "Chao ban : " + myText.Value + ".";
```

```
}
```

```
<body>
```

```
<form id="myForm" runat="server">
```

```
    <input type="text" id="myText" runat="server"><br>
```

```
        <input id="Submit1" type="submit" Value="Click Here!"
```

```
            OnServerClick="Button_ServerClick" runat="server">
```

```
    <span id="MySpan" runat="server"/><b>
```

```
</form>
```

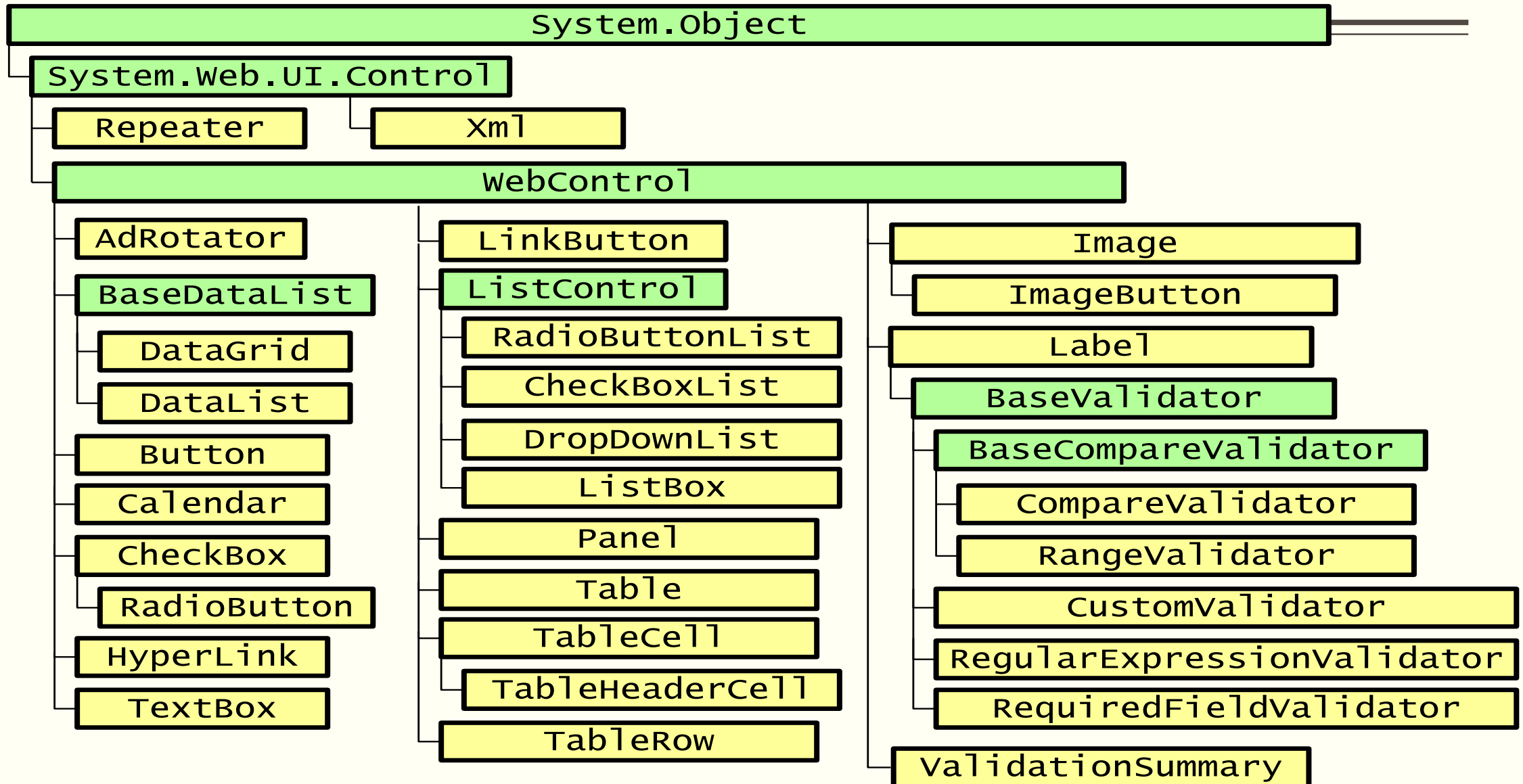
```
</body>
```

II. Web Server Control

- **Web server control** là những tag đặc biệt của ASP.NET. Các control này được xử lý trên server và đòi hỏi phải có thuộc tính **runat= "server"**
- **Web server control** tồn tại bên trong không gian tên **System.Web.UI.WebControls**
- Cú pháp:

```
<asp:controlname id="some_id" runat="server"/>
```

Hệ thống thứ bậc của Web Server Control



II. Web Server Control

Nhóm control cơ bản

- Cú pháp chung:

```
<asp:ControlName runat="sever" id= 'ControlID' />
```

- Các control cơ bản gồm:

II. **Web Server Control**

- Label
- Textbox
- Button
- CheckBox and Radio
- Image, Hyperlink, Panel
- List Controls group
- Table
- AddRotator
- FileUpload
- Validation

II. Web Server Control

- **Label server control:** dùng hiển thị văn bản trên trình duyệt.
Thuộc tính:

Text: sử dụng để nhận hoặc gán text

ví dụ:

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

```
String strName=lblMsg.Text;
```

Or

```
lblMsg.Text="Value";
```

II. Web Server Control

- **TextBox**: dùng để nhập liệu từ người sử dụng và hiển thị văn bản chỉ đọc
- **Thuộc tính :**
 - **AutoPostBack**: có 2 giá trị True và False khi một hành động trên trang web xảy ra một sự kiện.

Ví dụ:

```
<asp:TextBox ID="txtName" runat="server" AutoPostBack="true"/>
```


II. Web Server Control

- **TextMode:** Loại textbox: singleLine, MultiLine, Password
- **Text:** trả về giá trị hoặc gán giá trị
- **ReadOnly:** Dữ liệu không thay đổi

Ví dụ:

```
<asp:TextBox ID="txtName" runat="server">
```

```
</asp:TextBox>
```

```
String strName=txtName.Text;
```

```
Hoặc txtName.Text="Value";
```

II. Web Server Control

- **Sự kiện:**
 - **Text_Changed()**
 - **Focus():** cho phép đưa trở về phần tử được chỉ định trên form.

ví dụ:

```
txtName.Focus();
```

II. Web Server Control

- **Button**: thường sử dụng để submit form
 - *Phân loại*:
 - Button
 - LinkButton
 - ImageButton
 - *Sự kiện*
 - Onclick()
 - OnserverClick()

II. Web Server Control

- **Checkbox**
 - *Các thuộc tính:*
 - AutoPostBack
 - Checked
 - Text
 - *Sự kiện:*
 - CheckedChange()

II. Web Server Control

❑ RadioButton

- *Thuộc tính:*
 - **GroupName:** tên của nhóm
 - **Text:** nội dung văn bản của radioButton
 - **Checked:** radioButton được chọn
- *Sự kiện*
 - **onClick()**
 - **onCheckedChanged()**

II. Web Server Control

- **Image:**
- *Thuộc tính:*
 - **ImageUrl** : Địa chỉ của hình cần hiển thị
 - **AlternateText**: Dòng văn bản hiển thị khi hình không có sẵn
 - **ImageAlign**: canh vị trí tương đối của hình so với văn bản trên trang

II. Web Server Control

- **Hyperlink**
- *Thuộc tính:*
 - **ImageUrl**: đường dẫn đến hình cần hiển thị (nếu dùng thuộc tính này thì hyperlink có tác dụng giống như Imagebutton)
 - **NavigateUrl**: Địa chỉ URL cần link đến
 - **Text**: chuỗi văn bản chỉ mục liên kết hiển thị trên trình duyệt
 - **Target**: chỉ cửa sổ hiển thị trang đích

II. Web Server Control

- **Panel**

- Ý nghĩa: được dùng như ContainerControl đối với các control khác, nó thi hành nhiều chức năng:Kiểm soát các control chứa trong Panel Control
- Được dẫn xuất từ lớp Webcontrol

II. Web Server Control

▪ Cú pháp

```
<asp:Panel id="IdName" BackColor="DeepPink"></asp:Panel
```

▪ Thuộc tính:

- *Visible*: thuộc tính nhìn thấy được của control bên trong Panel control
- *BackImageUrl*: URL của hình ảnh hiển thị phía sau table.
- *BackColor*: màu nền của control

II. Web Server Control

Nhóm ListControl:

- DropDownList
- ListBox
- CheckBoxList
- RadioButtonList

```
<asp:ListControlName Id="IdName" runat="Server">  
    <asp:ListItem Value="value" Text="text"></asp:ListItem>  
</asp:ListControlName>
```

II. Web Server Control

- **ListItem**: Hoạt động giống nhau với 2 thuộc tính **Value** và **Text**
- **Item** có thể được thêm vào theo cách tĩnh (Design) hoặc Lập trình (Coding), **Add**
- Các thuộc tính:
 - SelectedIndex**: trả về chỉ số của phần tử được chọn.
 - SelectedItem**: trả về phần tử được chọn.
 - SelectedValue**: trả về giá trị được chọn.
- Sự kiện: **SelectedIndexChanged**

II. Web Server Control

AutoPostBack: **true** hoặc false

Items.Count: trả về số phần tử trong listControls

Items.Add(ListItem): thêm phần tử vào listControls

Items.Remove(ListItem): xóa phần tử khỏi ListControl

Items.Clear(): Xóa tất cả các phần tử.

Items[i].Selected: trả về true hoặc false. True nếu phần tử được chọn

Với $i=0, n-1$ ($n=Items.Count$).

II. ~~Web Server Control~~

- **DataSource**: thiết lập giá trị từ DataSource như DataTable, DataSet, Array, Collection, DataView
- **DataTextField**: thiết lập phần tử text từ DataSource
- **DataValueField**: thiết lập giá trị phần tử từ DataSource
- **DataBind()**: binding data vào ListControl

*(Là các thông tin **Data** lấy được từ CSDL tại*

Chương 7)

II. Web Server Control

Các đặc tính	Checkbox List	RadioButton List	DropDown List	ListBox
Chọn 1 Item duy nhất		X	X	
Chọn hơn 1 Item	X			X
Hiển thị toàn bộ danh sách			X	

II. Web Server Control

- Ví dụ: DropDownList

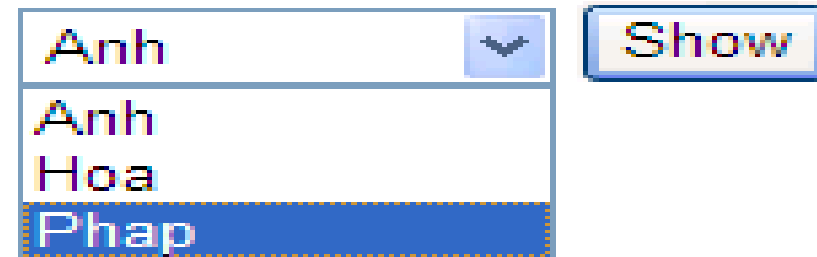
```
<asp:DropDownList Id="drpLan" runat="Server">
```

```
<asp:ListItem Value="English" Selected="true" >  
    Anh</asp:ListItem>
```

```
<asp:ListItem Value="French"> Pháp</asp:ListItem>
```

```
<asp:ListItem Value="Chinese"> Hoa</asp:ListItem>
```

```
</asp:ListControlName>
```



The screenshot displays a web control consisting of a dropdown menu and a button. The dropdown menu is currently open, showing three options: 'Anh', 'Hoa', and 'Phap'. The 'Anh' option is highlighted, indicating it is the selected item. To the right of the dropdown menu is a button labeled 'Show'.

II. Web Server Control

```
protected void btnThem_Click(object sender, EventArgs e)
{
    drpLan.Items.Add(txtThem.Text);
}
```

```
protected void Page_Load(object sender, EventArgs e)
{
    if(drpLan.SelectedItem!=null)
        lblKq.Text="The selected item is: " + drpLan.SelectedItem.Text;
    else
        lblKq.Text="No item is selected";
}
```


II. Web Server Control

Ví dụ: **CheckBoxList**

```
<asp:CheckBoxList ID="chkMonhoc" runat="server" Width="173px"
AutoPostBack="True">
    <asp:ListItem Value="Phan cung">Cau hinh Phan
cung</asp:ListItem>
    <asp:ListItem Value="CSDL">Co so du lieu</asp:ListItem>
</asp:CheckBoxList><br />
```

- ☐ Cau hinh Phan cung
- ☐ Co so du lieu

II. Web Server Control

```
protected void Page_Load(object sender, EventArgs e)
{
    string[] maMH = { "001", "002", "003", "004" };
    string[] tenMH = { "LT Windows", "LT Internet", "AVCN", "Cau hinh Mang" };
    for (int i = 0; i < maMH.Length; i++)
    {
        chkMonhoc.Items.Add(new ListItem(tenMH[i], maMH[i]));
    }
}
```

- ☐ Cau hinh Phan cung
- ☐ Co so du lieu
- ☐ LT Windows
- ☐ LT Internet
- ☐ AVCN
- ☐ Cau hinh Mang

II. Web Server Control

Ví dụ: **RadioButtonList**

```
<asp:RadioButtonList ID="radFont" runat="server" Width="126px"
AutoPostBack="True" OnSelectedIndexChanged="radFont_SelectedIndexChanged">
    <asp:ListItem>10</asp:ListItem>
    <asp:ListItem>12</asp:ListItem>
    <asp:ListItem>14</asp:ListItem>
    <asp:ListItem>16</asp:ListItem>
</asp:RadioButtonList><br />
<asp:Label ID="lblKq" runat="server" Height="32px" Text="To be or not to be"
Width="368px"></asp:Label>
```

II. Web Server Control

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        int[] sizefont = { 18, 20, 22, 24 };
        for (int i = 0; i < sizefont.Length; i++)
            radFont.Items.Add(new ListItem(sizefont[i].ToString()));
    }
}
```

II. Web Server Control

```
protected void radFont_SelectedIndexChanged(object sender, EventArgs e)
{
    int sizesselected = Convert.ToInt32(radFont.SelectedValue);
    lblKq.Font.Size = sizesselected;
}
```

- ☐ 10
- ☐ 12
- ☐ 14
- ☐ 16
- ☐ 18
- ☒ 20
- ☐ 22
- ☐ 24

To be or not to be

II. Web Server Control

- **Table:** Hiển thị thông tin dưới dạng dòng và cột.
- Table control cho phép xây dựng các bảng động bằng mã lệnh sử dụng các thuộc tính tập hợp Table Rows và Table Cells

II. Web Server Control

Cú pháp

```
<asp:Table id="Table1" runat="server" Width="100px"  
Height="70px"></asp:Table>
```

ASP ControlHTML	HTMLControl
Table	<table>
TableRow	<tr>
TableCell	<td>
TableHeaderCell	<th>

II. Web Server Control

- **AdRotator Server Control** :quảng cáo trên trang web
- **Thuộc tính:**
 - **ImageUrl**: URL của hình ảnh cần được hiển thị
 - **NavigateUrl**: URL của trang web phải chuyển đến control khi có sự kiện click.
 - **AlternateText**: Dòng văn bản hiển thị khi hình không có sẵn
 - **Keyword**: loại quảng cáo

II. Web Server Control

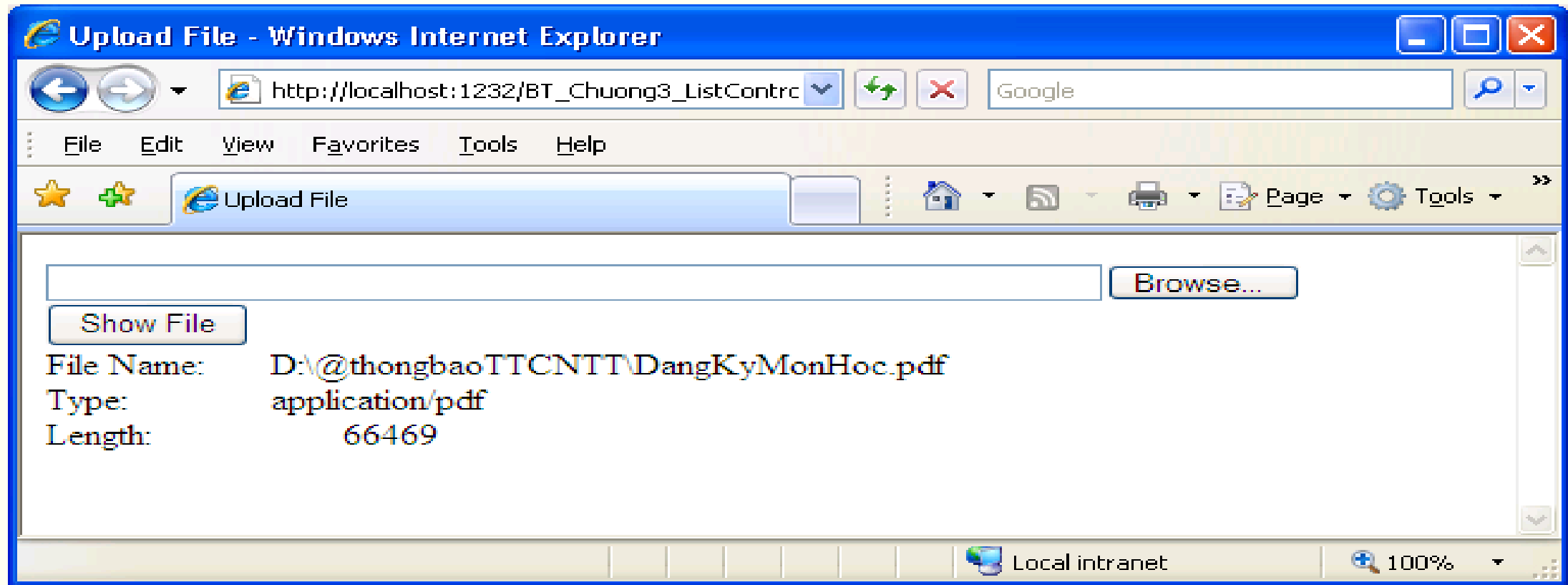
- **FileUpload Server Control**: dùng thực hiện upload file lên server
- Các thuộc tính:
 - **FileName**: Tên file được upload lên
 - **FileBytes**: Mảng bytes chứa nội dung file upload
 - **PostedFile**: Hiển thị đầy đủ như 1 đối tượng **HttpPostedFile**. **HttpPostedFile** có các thuộc tính
 - **FileName**: Ten File
 - **ContentType**: Loại File (.doc, .mdb, ...)
 - **ContentLength**: Kích thước của File.
 - **SaveAs**: Lưu file upload vào 1 thư mục bất kỳ

II. Web Server Control

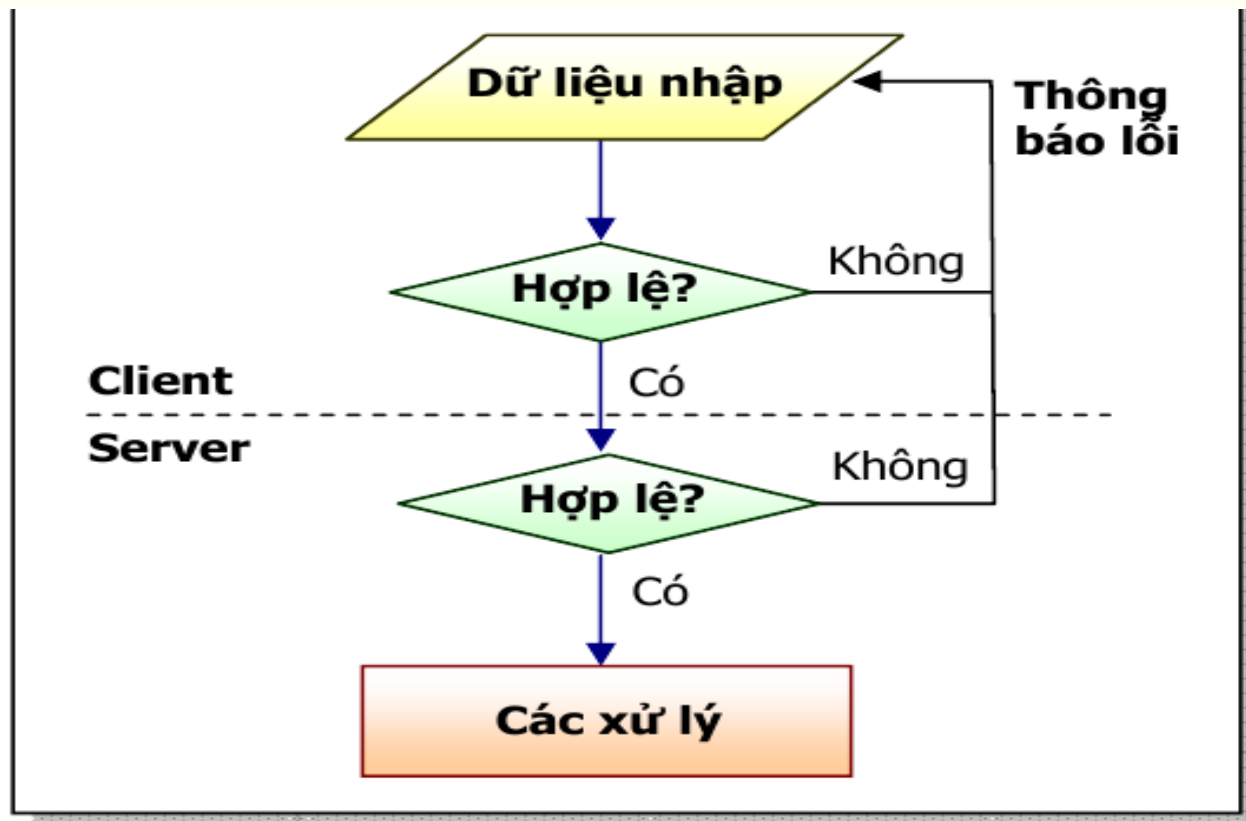
Ví dụ:

```
protected void btnShow_Click(object sender, EventArgs e)
{
    lblFilename.Text = FileUpload1.PostedFile.FileName.ToString();
    lblType.Text = FileUpload1.PostedFile.ContentType.ToString();
    lblLength.Text = FileUpload1.PostedFile.ContentLength.ToString();
    //Luu file
    string filename = FileUpload1.FileName.ToString();
    FileUpload1.PostedFile.SaveAs(Server.MapPath("") +
"\BT_Chuong3_ListControl" + filename);
}
```

II. Web Server Control



Validation Server Control



Sơ đồ xử lý kiểm tra dữ liệu nhập tại Client và Server

V. Validation Server Control

1. **RequiredFieldValidator Server Control**: yêu cầu người dùng bắt buộc phải nhập liệu
 - Cú pháp:

```
<asp:RequiredFieldValidator ID="RequiredFieldValidator1" runat="server"  
ControlToValidate="input_Control" ErrorMessage="Content_Message"  
</asp:RequiredFieldValidator>
```

V. Validation Server Control

▪ *Thuộc tính*

- ***ControlToValidate***: nhận một ID của một control khác ở trên form để kiểm tra nhập liệu.
- ***Display***: có ba giá trị : none, **static**, dynamic. thể hiện thông báo động hay tĩnh.
- ***EnableClientScript***: nhận true thì hiểu các script ở phía client, false thì không.
- ***ErrorMessage***: dòng thông báo khi dữ liệu không hợp lệ.
- ***Initialvalue***: giá trị khởi tạo

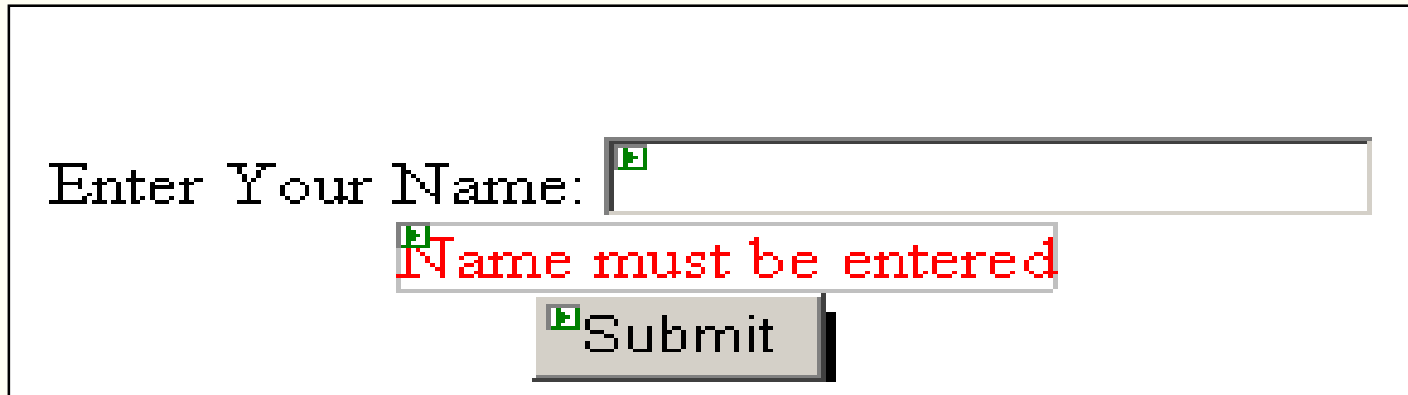
V. Validation Server Control

- ***Các sự kiện***

- `DataBinding()`
- `Disposed()`
- `Int()`
- `Load()`: Lập trình sự kiện này là để đặt giá trị ban đầu cho control.
- `PreRender()`
- `Unload()`

V. Validation Server Control

- Ví dụ: thiết kế form có dạng như sau, khi click nút submit, nếu chưa nhập tên thì xuất hiện thông báo lỗi



Enter Your Name:

Name must be entered

Mã chương trình

```
<body>
```

```
<form runat="server">
```

Enter Your Name:

```
<asp:TextBox id="txtName" rows="1 " width="50" runat="server"/>
```

```
<asp:RequiredFieldValidator id="validTxtName" runat="server"  
    controlToValidate="txtName"
```

```
    errorMessage="Name must be entered" display="static">
```

```
</asp:RequiredFieldValidator> </br>
```

```
<asp:Button id="btnSubmit" runat="server" text="Submit" />
```

```
</form></body>
```

V. Validation Server Control

2. **Range Validator Server Control:** Kiểm tra giới hạn nhập liệu, giá trị nhập phải nằm trong khoảng giới hạn cho trước, giới hạn này có thể là các hằng được đưa vào lúc thiết kế hoặc so sánh với các control khác trên trang web

V. Validation Server Control

- **Cú pháp:**

```
<asp:RangeValidator id="RangeValidator1" runat="server" display="static"
controlToValidate=" input_Control "
errorMessage=" Content_Message " type = " Integer"
minimumValue=minValue maximumValue=maxValue>
</asp:RangeValidator></br>
```

V. Validation Server Control

- Thuộc tính:
 - ***minimumValue, maximumValue***: khoảng giới hạn giá trị nhập liệu
 - ***ControlToValidate***: chứa ID của một control khác ở trên form để kiểm tra giới hạn nhập liệu.
 - ***Display***: có giá trị none, static, dynamic. Chọn kiểu thể hiện thông báo.

V. Validation Server Control

3. **RegularExpressionValidator Server Control:** kiểm tra dữ liệu nhập với khuôn biểu thức mẫu (RegularExpression) đã được định nghĩa trước. Visual Studio .NET cung cấp các khuôn biểu thức mẫu:
- Telephone numbers
 - Postal codes
 - E-mail addresses

V. Validation Server Control

❑ Thuộc tính:

ValidationExpression: Khung của biểu thức mẫu để so sánh kiểm tra

❑ Cú pháp:

```
<asp:RegularExpressionValidator ...
```

```
ControlToValidate="Input_control"...
```

```
ValidationExpression ="RegularExpression "...>*
```

```
</asp:RegularExpressionValidator >
```

V. Validation Server Control

4. **CompareValidator Server Control:** So sánh dữ liệu nhập với một trị trong một control khác hoặc một hằng được cho trước khi thiết kế hoặc một giá trị trong dữ liệu. Các phép toán so sánh >, >=, <, <=

Cú pháp:

```
<asp:CompareValidator id="comID" runat="server" display="static"  
controlToValidate="txtR"      errorMessage=" Content_Message "  
      type="Double" operator="DataTypeCheck">  
</asp:CompareValidator>
```

V. Validation Server Control

▪ Thuộc tính

- ***ControlToCompare***: chứa ID của một control mà giá trị của control ID này sẽ so sánh với dữ liệu của một control khác
- ***ControlToValidate***: chứa ID của một control mà dữ liệu của control này được so sánh với dữ liệu của control ở thuộc tính trên.
- ***Operator***: Toán tử so sánh
- ***Type*** : Kiểu dữ liệu để so sánh gồm (String, Integer, Date, Double, Currency)
- ***ValueToCompare***: chứa giá trị so sánh hằng

V. Validation Server Control

▪ Các sự kiện

- ☐ `DataBinding()`

- ☐ `Disposed()`

- ☐ `Int()`

- ☐ `Load()`: Lập trình sự kiện này là để đặt giá trị ban đầu cho control.

- ☐ `PreRender()`

- ☐ `Unload()`

V. Validation Server Control

5. CustomValidator Server Control

- Kiểm tra tính hợp lệ dữ liệu của một control theo một yêu cầu, một ràng buộc nào đó, hay một kiểu dữ liệu được người sử dụng định nghĩa trước đó.
- CustomValidator Server Control có thể kiểm tra hợp lệ cả phía client và server

V. Validation Server Control

- **Thuộc tính**

- ***ClientValidationFunction***: thuộc tính này nó chứa một tên hàm, mà hàm này được lập trình ở client (bằng javascript).
- ***ControlToValidate***: Nhận ID của một control trên form để kiểm tra dữ liệu.

V. Validation Server Control

▪ Các sự kiện

- `DataBinding()`
- `Disposed()`
- `Init()`
- `Load()`: Lập trình sự kiện này là để đặt giá trị ban đầu cho control.
- `PreRender()`
- **`ServerValidate()`**: sự kiện này được lập trình trên server để kiểm tra tính hợp lệ của dữ liệu.
- `Unload()`

V. Validation Server Control

Ví dụ: thiết kế form như sau, yêu cầu kiểm tra số Pin

User ID:	<input type="text" value="myusername"/>
PIN	<input type="text" value="a"/>
<input type="button" value="Save"/>	
Invalid PIN number!	

//Mã giao diện

```
User ID:<asp:textbox id=txtUserID runat=server />
```

```
PIN:<asp:textbox id=txtPIN runat=server />
```

```
<asp:Button id=Button1 text="Save" OnClick="SaveBtn_Click"
runat=server /><BR>
```

```
<asp:CustomValidator id="CustomValidator1"
ControlToValidate="txtPIN" OnServerValidate="ServerVerify"
runat="server"> Invalid PIN number!
```

```
</asp:CustomValidator>
```

```
<asp:label id='lblOutput' runat='server' />
```

//hàm ServerVerify kiểm tra trên server

```
void ServerVerify(Object Sender, ServerValidateEventArgs Value)
{
    if (txtPIN.Text == "A999")
        Value.IsValid = true;
    else
        Value.IsValid = false;
}
```

V. Validation Server Control

6. **ValidationSummary Server Control**: tập hợp các thông báo lỗi từ tất cả các điều khiển trên một trang

- Cú pháp:

```
<asp:ValidationSummary id="valSummary" runat="server"  
HeaderText="These errors were found" ShowSummary="True"  
DisplayMode="List"/>
```


V. Validation Server Control

- **Thuộc tính**
 - ***DisplayMode***: cung cấp 3 định dạng hiển thị MessageBox: List, BulletList, SingleParagraph ***HeaderText***: Dòng tiêu đề cho thông báo của các control.
 - ***ShowMessageBox*** : True thì hiện thông báo động, False thì hiện thông báo tĩnh.
 - ***ShowSummary***: True là hiện thị control này khi chạy ứng dụng , False thì không (thường dùng nhất.)

V. Validation Server Control

- **Các sự kiện**
 - `DataBinding()`
 - `Disposed()`
 - `Int()`
 - `Load()`: Lập trình sự kiện này là để đặt giá trị ban đầu cho control.
 - `PreRender()`
 - `Unload()`