ASP.NET 第 3 - 4章

*第三章 : 使用網頁控制項 P 83*

**基本類型**

* Web Server Control

ASP.NET內建，如Button, TextBox, 月曆, 功能表

<asp: xxxx> </asp: xxx>

可觸發事件，並針對事件寫Code

伺服器端自動保存控制項的狀態

根據前端的環境，自動轉化做不同的HTML 樣子。

* Html Control

使用Html 或它所支持 ( 如XHTML ) 的控制項,

<input id = “Text”type=”text”/>

所有網頁開發程式和瀏覽器都可以支援

有固定的標準，所有使用者都看到一樣的結果

但是預設無法和ASP.NET的伺服器溝通，要特殊手法開啟

* Moblie Web Control

ASP.NET內建，只能在 Moblie Web From 中使用。

**控制項事件** P88

* 宣告

每個控制項點兩下, 馬上跳去建立常式。

在設計頁面左鍵控制項一下，出屬性視窗

~~文字編輯視窗選擇~~

屬性視窗按閃電圖形，可以看到它綁定的EVENT和方法

* 動態

Protected void Page\_Load(object sender, EventArgs e)

{Button1.Click += new EventHandler(Button1\_Click);}

// EventHandler 在WIN FORM比較常用，WEB FORM 少用，略。

Protected void Button1\_Click(object sender, EventArgs e)

{Response.Write(“XXX”)}

* 事件觸發的相關屬性

AutoPostBack="true"

AutoPostBack這個是大部份控制項的共用屬性，預設為”false”

假設將 <asp:TextBox ID="TextBox1" runat="server" AutoPostBack="true"></asp:TextBox>

當滑鼠在網頁上隨意點擊，也會觸發這個TextBox的PostBack

(原本至少是要按ENTER的 , 不過這個功能不常開。)

*各種常見的控制項*

**TextBox**

可單行，多行，密碼輸入用

* 常用屬性︰

Text = 方塊中的文字

TextMode = SingleLine / MutiLine / Password ,

預設為SingleLine, 其他要自己改。

Password 會隱藏輸入為\*

MaxLength = 最多可以輸入字元，預設為0 = 不限。

AccessKey = 在網頁中，可以用快速鍵跳到的該位置的設定。

AutoCompeleteType = 讓使用者選擇以前輸入過的值

* 常用事件︰

TextChanged = 當使用者修改文字後觸發的事。

**Image** (沒有實作)

* 常用屬性︰

ImageUrl = 設定影像的來源地址

AlternateText = 找不到影像時，會出現的文字

**HiddenFied**

透過操作 <input type = “hidden”> 的標籤來儲存資料。

可以隱藏而不顯示在瀏覽器上。

例︰

<asp:HiddenField ID = “XXX” runat=”server” Value = “YYY”>

產生︰

<input type = “hidden” name = “XXX”

id = “HiddenField1” value = “YYY”>

**Button, ImageButton, LinkButton**

ImageButton = 可以用ImageUrl自訂按鈕圖片。(沒有實作)

LinkButton = 外觀似超連結，卻是按鈕。?? (沒有實作)

* 三者共有的屬性和事件︰

Commad = 按下後，取CommandName , CommandArgument 執行

CommandName = 可供多個Button**共用** 的命令

CommandArgument = CommandName的 參數。

OnClick = 按下會發生的事件

OnClientClick = 按下後發生的前端事件

額外︰要Button 不會觸發POST BACK 的方法︰

<http://codingnoteofccs.blogspot.com/2014/08/aspnetbuttonpostback.html>

*P 97 使用Button, HiddenField, TextBox 控制項 實作︰ ASP.NET(EX3.1)*

*CommandName屬性自創實作︰ ASP.NET(EX3.6)*

*CommandName是button 中可以自己加入的String 參數。*

*BUTTONS的方法不同，各傳入一個CommandName的Str參數到另一個方法。*

*該整合方法，用Switch Case 判斷要做什麼事情。*

**CheckBox 與 RadioButton**

CheckBox = ture / false 二選一，以打勾表示

一群CheckBox = 多選題

RadioButton = 有一個GroupName的屬性, 同組內，多選一

* 兩者共有的屬性和事件︰

Text = 選項的文字

Checked = 是否有被勾,只能 ture / false

CheckChange = 選取狀態改變的時候會觸發，做出處理

**清單系列控制項**

* ButtonList

建立一個帶有符號或編號的清單

符號 : BulletStyle屬性 (可挑選)

數字 : FirstBulletNumber屬性 (設定第一個數字從什麼開始)

* ListBox

項目清單方塊

多選或複選? : SelectoinMode 屬性

* DropdownList

下拉清單, 多選一

* RadioButtonList

一次建多個RadioButton, 並均屬於一個群組, 多選一

* CheckBoxList

一次建多個CheckBox

* 清單系列共有的屬性和事件︰

Items = 所有成員的 Text 和 Value 集合

(Value是藏起來的程式運作的關鍵字)

SelectedIndexChanged = 任何一個成員的狀態改變會觸發的事。

(除了ButtonList, 他沒有選項。)

SelectedItem = 取得被選取的成員的內容

* 清單系列設定成員方法︰
* 各控制項的屬性視窗
* 設定(Array/Collection)作成員資料來源(見實作)
* 程式︰用 Item 屬件的 Add 方法來動態加入成員。
* 程式︰用系列的 DataSource屬性

來讀取Collection或Array作資料來源

* 取得被選取的成員的內容 :

多選一 : ListBox, DropdownList,RadioButtonList

可以直接用 SelectedItem 屬性

多選多 : ListBox, CheckBoxList

需要用迴圈一一檢視成員內容︰

protected void Button1\_Click(object sender, EventArgs e)

{ for(int i = 0; i < CheckBoxList1.Items.Count; i++ )

{ if(CheckBoxList1.Item[i].Selected)

{ Response.Write(CheckBoxList1.Item(i).Text + “<BR>”); }

}}

*P105 使用ListBox 實作 ASP.NET(EX3.2)*

*# 項目要按 Shift才可以多選。*

**Panel**

可以用來組合其他控制項的控制項

有視覺化的群組效果 ( 例如: 要不要一起隱藏起來)

* Panel常用屬性 :

GroupingText = 標題文字

BackImgUrl = 指定背景圖片的網址

ScrollBar = 預設為None, 可指定 Horizonal / Vertical / Both / Auto

**Calender**

月曆

* Calender常用屬性和事件 :

SelectedDate = 選擇的日期

VisibleDate = 所顯示的月份 (如只有SelectedDate, 將不顯示月曆)

SelectionChanged = 選日期所觸發的事件

DayRander = 客製化月份中的某些日字

* Auto Format

在 Calender 控制項上右鍵 > Auto Format > 選一個預定樣式

*P 111 實作 Calender ASP.NET(EX3.3)*

補充指引︰

在原始檔的控制項宣告，加入︰ OnDayRender="Calendar1\_DayRender"

<asp:Calendar ID="Calendar1"

OnDayRender="Calendar1\_DayRender"

runat="server" …….. </asp:Calendar>

這個會指向一個 .aspx.cs 的方法 (不在Page\_Load之內), 要自己開︰

protected void Calendar1\_DayRender(object sender, DayRenderEventArgs e)

{ 裡面才是填上的書中程式碼的地方 }

# 我不知道是不是有什麼可以自動開出來的方法，但是我找不到。

*用戶端指令碼*

**呼叫用戶端指令碼** P114

一種在用戶端瀏覽器執行的程式碼，常見如javascript 等。

在 .aspx 的程式碼頁，在form 之後 進行宣告︰

*<script type="text/javascript">*

*function check()*

*{ alert("What?");*

*return true; }*

*</script>*

在 .aspx 的程式碼頁，在html 之前，進行宣告︰

*void Button1\_Click (object sender, EventArgs e)*

*{*

*Label1.Text += " Thank you for visiting our site.";*

*}*

用特定標籤來繫結？ (這個只關 javascript 的事，略。)

<script type = “text/javascript” src=”JScript”></script>

OnClientClick 和 Onclick 的分別︰

<https://docs.microsoft.com/zh-tw/dotnet/api/system.web.ui.webcontrols.button.onclientclick?view=netframework-4.8>

<https://blog.csdn.net/xiaouncle/article/details/66973909>

*依參考自創實作 OnClientClick 和 Onclick ASP.NET(EX3.4)*

**動態註冊用戶端指令碼**

* 註冊 用戶端指令碼 至 html <form>標籤的 開端

ClientScript.RegisterClientScriptBlock(

this.GetType(),

"1",

"function function1() {alert('XXX'); return false;}",

true);

* 註冊 用戶端指令碼 至 htm <form>標籤的 結尾

ClientScript.RegisterStartupScript(

this.GetType(),

"2",

"function function2() {alert('YYY');}",

true);

* 註冊 JavaScript 的檔案

ClientScript.RegisterClientScriptInclude(

this.GetType(),

"3",

"JavaScript1.js");

*P 118 動態註冊 實作 ASP.NET(EX3.5)*

*第四章 : 驗證使用者輸入 P 123*

**驗證的類型**

* 必填欄位 RequiredFieldValidator

在資料送出前檢查有沒有填好

* 數值範圍限制 RangeValidator

可以是數字﹑英文字母﹑日期的範圍

* 輸入內容比較 CompareValidator

會和另一個控制項的內容對比，是否等於，大於或小於

如再次確認密碼。

* 檢查格式 RegularExpressionValidator

如是否電子郵件

* 自行定義 Custom Validator

如和資料庫比對

**驗證執行的地方**

* 用戶端驗證

不需要POSTBACK ，改善效率和減輕伺服器的負擔

* 伺服器端驗證

當用戶端不支援驗證指令碼等，失效時的第二層保險

**驗證的特性**

* 無法單獨存在

如果沒有輸入的控制項，卻有驗證的存在，執行會產生例外

* 單一個輸入的控制項，可以組合多個驗證類型

**驗證類型 共有的屬性**

* ControlToValidate

輸入要檢查控制項的ID

* ErrorMessage

當驗證失敗時，給使用者看的訊息

* Display

所檢查的控制項顯示與否︰

None = 不顯示

Static = 通不通過也保留空間

Dynamic = 通過了則不保留空間

* EnableClientScript

是否啟動用戶端驗證

* SetFocusOnError

當驗證失敗時，游標是否停回輸入控制項上

*各驗證類型範例*

**必填欄位 RequiredFieldValidator**

<asp:RequiredFieldValidator

ID="RequiredFieldValidator1"

runat="server"

ControIToValidate="TextBox1 "

ErrorMessage="帳號必須輸入" >

<asp:RequiredFieldValidator>

可以加入︰ Initial Value =""

可以不加入，預設為空白，即是不可為空白的意思。

如果寫了內容，即預設為該內容，而驗查時不可以是那個內容。

**數值範圍限制 RangeValidator**

<asp:RangeValidator

ControIToValidate="TextBox2"

ErrorMessage="必須介於 10-80歲"

MaximumValue="80"

MinimumValue="10"

Type="lnteger" >

</asp:RangeValidator>

如果Type和Max , Min 他們不合的話，會有判斷的錯誤。

**輸入內容比較 CompareValidator**

<asp:CompareValidator

ErrorMessage="密碼與確認密碼不一致"

ControIToCompare="TextBox3"

ControlToValidate="TextBox4" >

</asp:CompareValidator>

可加入的特殊屬性︰

ValueToCompare = 要比的指定值

ControIToCompare = 要比的控制項

Type = 設定對比雙方的值是以什麼類型來比

Operator = 這個值是 X 於對方？

X = Equal,NoEqual,GreaterThan,GreaterThanEqual....

要輸入英文，不是符號。

**檢查格式 RegularExpressionValidator**

<asp:RegularExpressionValidator

ID="RegularExpressionValidator1"

runat="server"

ControIToValidate="TextBox5"

ErrorMessage="格式不正確"

ValidationExpression="'a|b'">

</asp:RegularExpressionValidator>

RegularExpression的參考網頁︰

<http://kaiching.org/pydoing/py/python-library-re.html>

https://kknews.cc/zh-tw/code/4p2zym2.html

屬性欄中，有基本參考格式的選單。

*P 138 使用驗證項 實作 ASP.NET(EX4.1)*

在PageLoad 要加入一句話，

UnobtrusiveValidationMode = UnobtrusiveValidationMode.None;

詳見︰

https://dotblogs.com.tw/rainmaker/2013/05/03/102768

在瀏覽器要輸入後再按Enter ,

然後如果該 格 應輸入的格式不對，Message 就會出現。

直到輸入對了再按 Enter 才消失。

**自行定義 CustomValidator**

必須自行定義用戶端和伺服器端的處理。

特殊屬性:

ClientValidationFunction 指定客戶端Script 函式

例子︰

<script type="text/javascript" >

function Client\_Validate(sender,e)

{

if (e.Value % 2 ==0)

e.IsValid=true;

else

e.IsValid=false;

}

</script>

特殊事件:

ServerValidata 撰寫伺服器端驗證程式

例子︰

protected void CustomValidatorl\_ServerValidate

(object sender, EventArgs args)

{

if (int.Parse(args.Value) % 2 == 0)

args.IsValid = true;

else

args.IsValid = false;

}

args.Value = 控制項傳入的值

args.IsValid = true / false 通過和不通過。

*P 144 自行定義屬性 兼 關閉在用戶端驗查 實作 ASP.NET(EX4.2)*

*錯誤訊息的顯示方法*

* Inline

預設 就是顯示在它原本放置的原位,

* Summary

在某個指定位置顯示

用法︰

使用ValidationSummary控制項,

確定它的 ShowSummary 屬性為 true

執行時，會自動取得網頁上所有ErrorMessage。

它所收集的其他驗查控制項，

最好設定 Text 和 ErrorMessage 屬性

因為該驗查控制項不通過時，

Text 會在 原地 留下顯示符號，通常為 \*。

ErrorMessage 則給 ValidationSummary 去用。

* MessageBox

跳出小窗戶

用法︰

使用ValidationSummary控制項,

確定它的 ShowMessageBox 屬為 true

* Custom

自行決定外觀和取得訊息。

用法︰

foreach (IValidator vtor in this.Validators)

{ if (!vtor.IsValid)

{Response. Write(vtor.ErrorMessage); }

}

// 這個貼上過PAGE LOAD 實作，

// 明明看上去是可行的，但就是不行，原因不明。

this.Validators = 這個頁面所有的檢查控制項

Response. Write = 寫在網頁的最後面

*設計ASP.NET網頁驗證的注意事項 P150*

* Button

CausesValidation 屬性 = true / false

封住一個按鈕原本會去觸發檢查的功能，如︰

<asp:Button ID="Button1" runat="server" Text="檢查上面的" ValidationGroup="group1" OnClick="Button1\_Click" CausesValidation="false" />

* Validation Group

按下一個按鈕只觸發一些檢查項，另一個按鈕又觸發另一些

用法︰

將驗證項和按鈕的 Validation Group 輸入為相同的自取名稱。

* Page

<https://docs.microsoft.com/zh-tw/dotnet/api/system.web.ui.page.validate?view=netframework-4.8>

Validate() 方法 – 有前後次序地驗證所有控制項。詳見實作。

用在CausesValidation = false 的 Button

用在AutoPostBack 的 輸入項。

*自創實作 ︰ ASP.NET(EX4.4)*

*關於︰ ENABLE CLIENT SCRIPT 和 Validate() 的次序關係*

*SET FOUCE ON ERROR ,*

*CausesValidation,*

*Validate("指定群組")*

*if(this.IsPostBack)*

Page.IsValid 屬性 – 可以知道是否整頁都通過了

如︰

protected void Button1\_Click(object sender, EventArgs e)

{

Response.Write(Page.IsValid);

}

那麼只要按下Button1 它就會做整頁的檢查，

如果都通過，那麼在網頁的最底就會顯示 “true”

*P152 驗證項 與 其訊息 分組 ASP.NET(EX4.3)*

記得把按鈕也加入Validation Group 要不然 ValidationSummary 不會觸發。