



# ASP.NET 程式設計

## ASP.NET 網頁簡介

講師：林賢達

Peter.lin@imestech.com



# 課程大綱

- Web特性
- ASP.NET網頁
- 事件驅動模式
- 頁面生命週期
- Global.asax

# Web特性

- 使用者介面不夠豐富，而且使用性較差
- 網頁是無狀態的
- 使用<Form></Form>或者QueryString來傳遞資訊
- 分成用戶端程式與伺服器端程式
  - Client Side Script：即時互動和資料驗證
  - Server Side Script：存取後端伺服器的資源
- 使用PostBack機制來進行瀏覽器和伺服器之間的互動

# ASP.NET網頁

- ASP.NET是由微軟所提供開發Web應用程式的基礎結構
  - 使用ASP.NET網頁作為Web應用程式的使用者介面，支援任何的瀏覽器或移動設備
  - 使用伺服器端的程式來實現Web應用程式的邏輯
  - 在Web伺服器上執行，用於動態建立和管理網頁輸出，可包含任何支援HTTP的程式語言，例如HTML, XML, WML和Client Script等

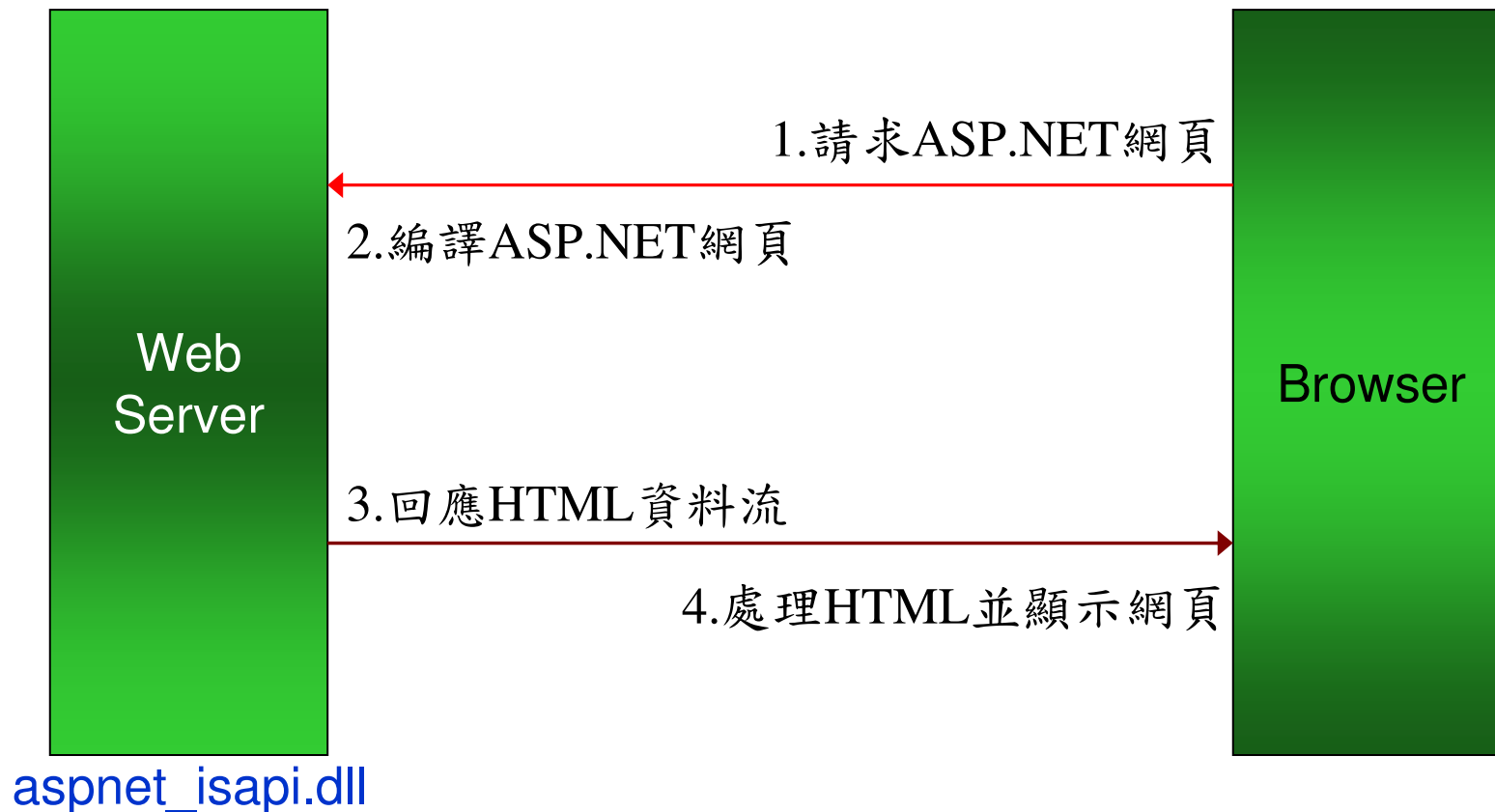
# ASP.NET網頁

## ■ ASP.NET的特性

- 使用Microsoft .NET Framework和ASP.NET技術
- 支援任何的瀏覽器或移動設備。根據瀏覽器廠商和版本，自動產生正確的網頁樣式和佈局
- 支援C#和VB.NET等CLS語言
- 提供Visual Studio 2005 整合開發環境，內建約80個伺服器端控制項
- 採用開放式架構，允許開發人員和協力廠商開發自訂控制項

# ASP.NET工作原理

## ■ Request / Response運行機制



# 建立ASP.NET網站

- 將Web專案範本獨立成"網站"
- 存放位置可以在檔案系統、HTTP和FTP
- 內建一個用於開發的輕量級Web服務器
- 檔案管理，加入參考和Web Page、HTML網頁和樣式表...等
- ASP.NET Configuration網站管理工具
- Web.Config組態設定檔
- 複製和發行網站

# ASP.NET網頁

- 隱藏複雜的PostBack機制，讓開發人員專注於商務邏輯
- 將使用者介面與程式邏輯分開
- 內建大量豐富的伺服器端控制項
- 使用事件驅動模式來進程式開發
  - 使用PostBack模擬使用者事件
- 使用隱藏欄位來儲存狀態，對同一個網頁的請求，其狀態均會自動被保存下來
- 使用ViewState儲存頁面的狀態



# ASP.NET網頁

- 加入Web 表單(類似Windows表單)作為使用者與網站之間的互動介面
- 在Web 表單加入所需的控制項，分成HTML控制項與ASP.NET控制項(runat="server")兩大類
- 設定控制項的屬性
- 撰寫控制項的事件處理程序

# 事件驅動模式

## ■ Client Side Event

```
<input id="Button1" type="button" value="Client Side Event" onclick="Button1_Click()" />
```

```
<script language="javascript" type="text/javascript">  
    function Button1_Click()  
    {  
        alert("hello from client")  
    }  
</script>
```

## ■ Server Side Event

### □ 使用 OnServerClick

```
<input id="Button2" type="button" value="Server Side Event" onserverclick="Button2_Click" runat="server" />
```

```
<script runat="server">  
    Protected Sub Button2_Click(ByVal sender As Object, ByVal e As EventArgs)  
        Response.Write("hello form server")  
    End Sub  
</script>
```

# 事件驅動模式

## ■ Server Side Event(續)

### □ 使用 Handles

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load  
    Response.Write("使用Handles關鍵字")  
End Sub
```

### □ 使用 AddHandler

```
Protected Sub Page_PreLoad(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.PreLoad  
    AddHandler Me.Load, AddressOf Me.PageLoad  
End Sub
```

---

```
Protected Sub PageLoad(ByVal sender As Object, ByVal e As EventArgs)  
    Response.Write("使用AddHandler關鍵字")  
End Sub
```

# 事件驅動模式

## ■ Server Side Event(續)

### □ 使用 Overrides

```
Protected Overrides Sub OnLoad(ByVal e As System.EventArgs)
    Response.Write("Override OnLoad Method")
    MyBase.OnLoad(e)
End Sub
```

### □ 使用 AutoEventWireUp

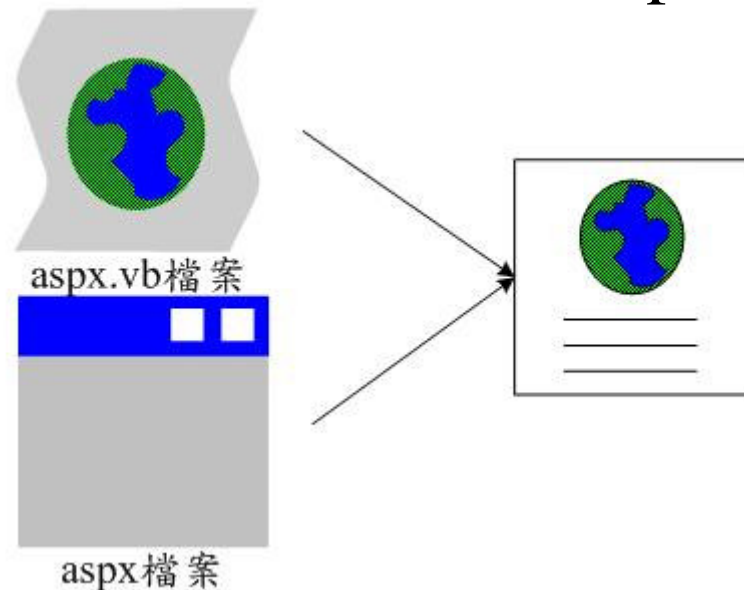
```
<%@ Page Language="VB" AutoEventWireup="true" CodeFile="DemoEvent2.aspx.vb" Inherits="DemoEvent2" %>

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
    Response.Write("使用AutoEventWireup")
End Sub
```

# ASP.NET網頁

## ■ ASP.NET網頁的組成

- aspx檔案：HTML和Server Side控制項等UI元素
- aspx.vb檔案：定義System.Web.UI.Page的衍生類別和Server Side Script，編譯成dll



# ASP.NET網頁

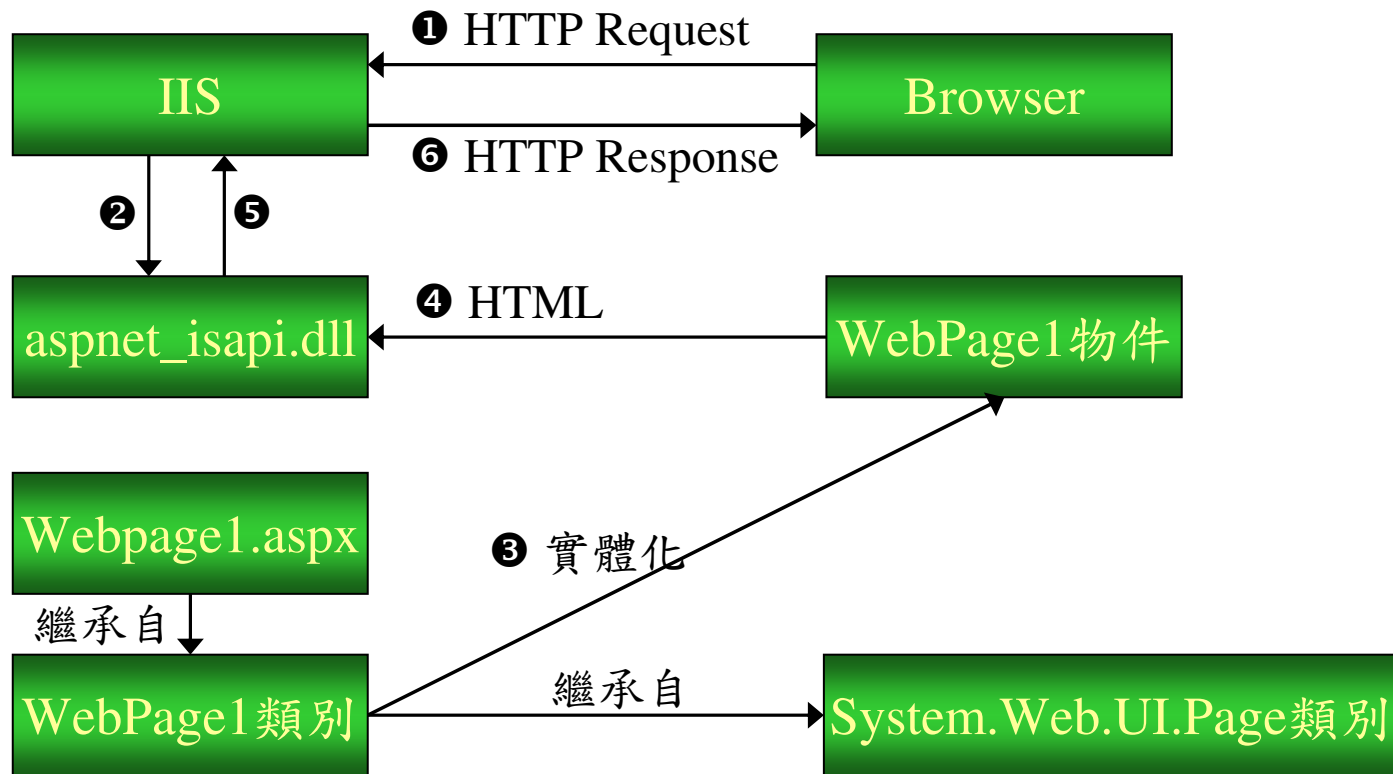
- Code Inline開發模式：Server Side Script內嵌在aspx檔案，即UI元素與Server Side Script在同一個檔案
  - 使用<script runat=server>將網頁的程式邏輯內嵌在aspx檔案
  - 在第一次請求時，動態編譯程式
  - 需自行將事件關聯到事件處理函式
  - 不支援部分偵錯功能

# ASP.NET網頁

- Code Behind開發模式：UI元素置於在aspx檔案，而Server Side Script放在aspx.vb檔案(使用VB.NET)，此為預設模式
  - aspx.vb檔案，定義Page的衍生類別和撰寫網頁的程式邏輯
  - aspx.vb檔案會事先被編譯成dll，能夠保護程式碼，提供安全性和執行效能。
  - 因為UI元素與程式邏輯分開，因此美工設計和系統功能可專業分工，開發和維護上更為容易

# ASP.NET網頁的生命週期

## ■ 執行過程





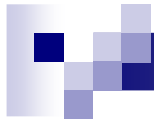
# ASP.NET網頁的生命週期

## ■ 往返處理(Round Trip)

- 使用者操作伺服器端控制項，若AutoPostBack屬性為True
- 提交(Submit)網頁到Web伺服器，執行相關的事件程序，重建網頁，然後回應瀏覽器的請求

## ■ 重建網頁

- 第一次請求網頁時或當事件被提交到Web伺服器
- 執行程式和重建整個網頁
- 將結果(HTML資料流)輸出至瀏覽器
- 清除網頁的變數、控制項狀態和屬性等訊息



# ASP.NET 網頁事件

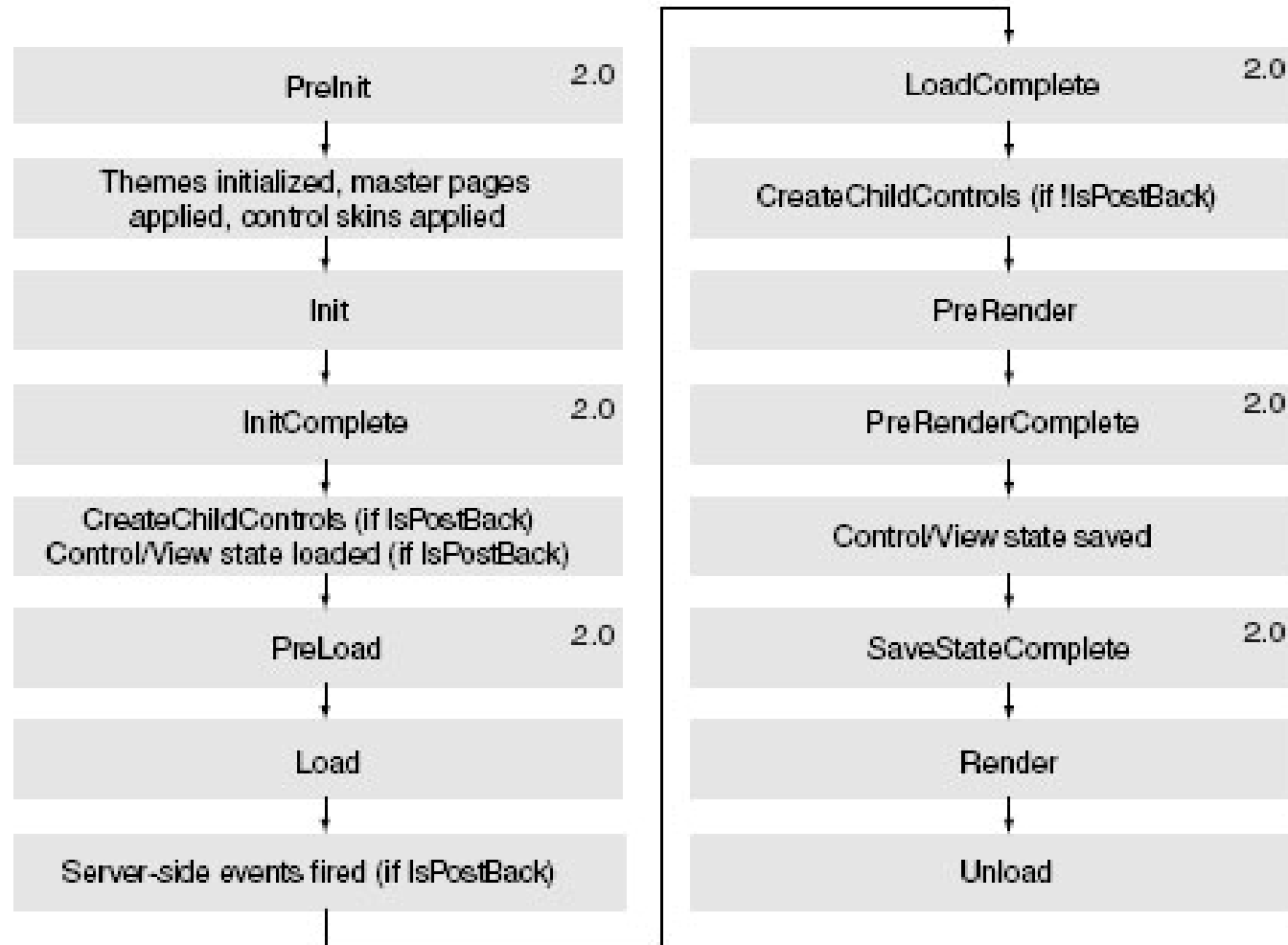


FIGURE 1-2: Events in the page lifecycle

# ASP.NET網頁事件

## ■ 重要事件

- Load：參考Web表單的控制項。可使用Page物件 IsPostBack屬性來判斷是否為第一次請求？
- Unload：清除暫存的資料
- Error：如果在網頁處理時發生未處理的例外，便會觸發此事件

# Page物件

- 每一個ASP.NET網頁都是一個Page物件
- System.UI.Web.Page類別
- IsPostBack屬性判斷是否為第一次執行
  - 若為True表示第一次被執行
  - 若為False表示由使用者按下按鈕後重新執行

# Global.asax

## ■ ASP.NET全域應用程式檔案

- 只有一個Global.asax，通常是放在根目錄下
- 不會產生使用者介面，也不回應單一網頁的請求
- 主要負責處理應用程式和使用者連線(Session)級別的事件

## ■ Application和Session事件

- Application\_Start：當Web應用程式接受第一次請求時，開始Application物件的生命週期，便會觸發此事件而且只會觸發一次

# Global.asax

## ■ Application和Session事件(續)

- Application\_End：當停止WWW服務時，結束Application物件的生命週期，便會觸發此事件，而且只會觸發一次
- Session\_Start：每當建立新的使用者連線時，便會觸發此事件
- Session\_End：當離開Web應用程式、呼叫Session物件的Abandon方法或者Session Timeout時，均會觸發此事件

# 頁面指令

- 指示編譯器如何來處理頁面
  - 網頁指令是放在aspx檔案的上方
  - 語法，字母大小寫無關
    - <%@ **direction** {attribute=value} %>
- 使用Page 指令如何編譯此網頁
  - Language參數，此網頁使用的程式語言
  - Debug參數，是否使用偵錯字元編譯此頁
  - Buffer參數：是否啟用網頁緩衝區

# 頁面指令

- AutoEventWireup 參數，指定網頁事件是否自動關聯事件
- ErrorPage 參數，當發生未處理例外時，重新導向的網頁
- Trace 參數，是否啟動追蹤功能
- 使用 Import 指令匯入命名空間，用於 Code Inline 模式，例如  
`<% @Import Namespace="System.IO"%>`



# App\_Code 資料夾

- App\_Code 資料夾是ASP.NET網站的特殊目錄
  - 可以放置任何的程式碼檔案
  - 常用於定義類別、業務邏輯層或資料存取層
  - 執行時會自動編譯App\_Code的程式