Web Service with ASP.NET

林光龍

大綱

- .NET 基本概念
- ASP.NET 工作平台
- HelloWorld.aspx
- VB.NET 語法簡介
- HTML 控制項
- Web 控制項
- My Amazon 製作

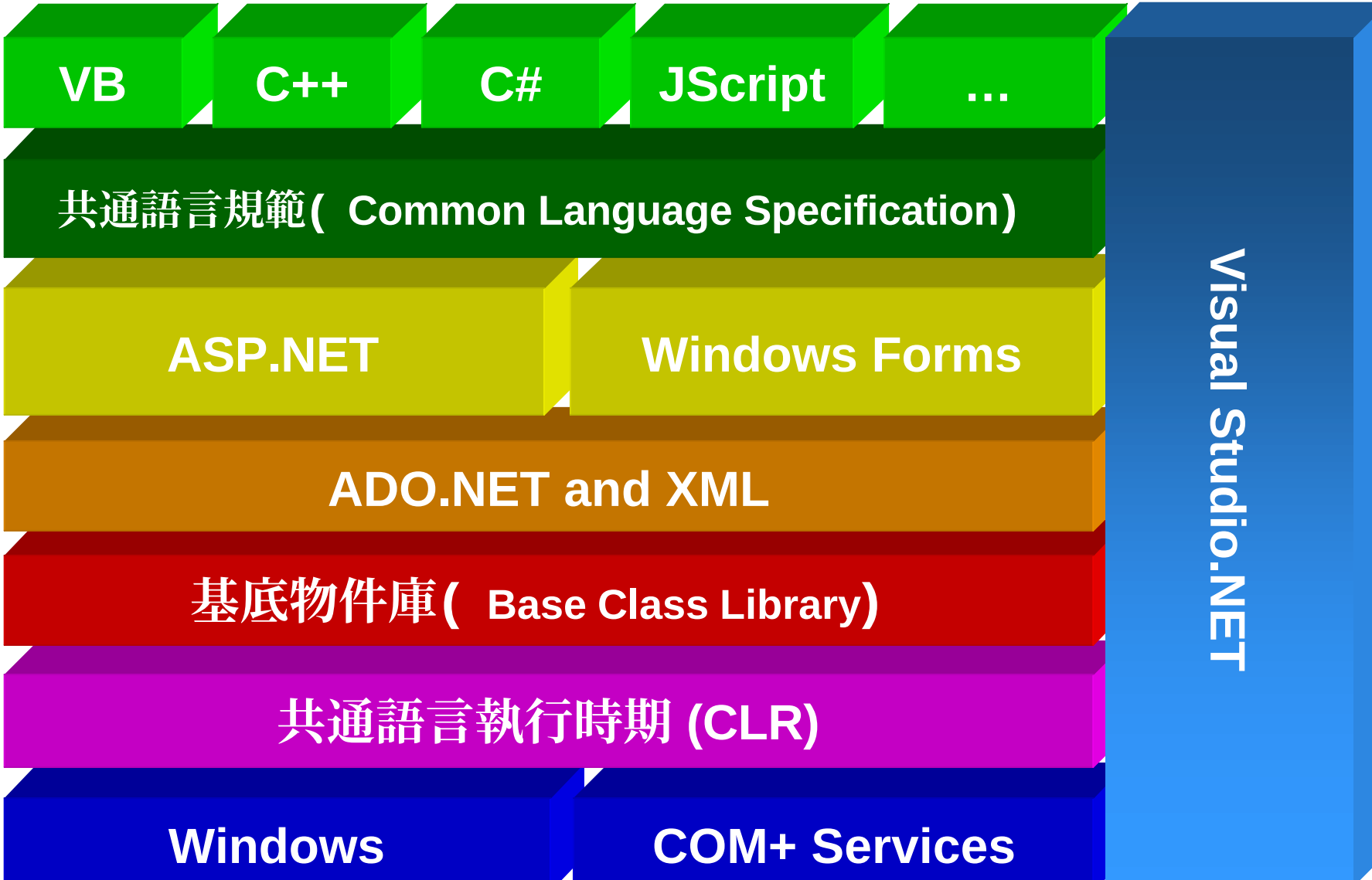
基本概念

- 「.NET」（唸作dot NET）是微軟下一世代的平台
- 「.NET平台」（.NET Platform）就是**以網路為執行平台**，只要任何在網路上的資源皆可存取，其特色為**跨語言**、**跨硬體**及**跨平台**，是專門為Internet 解決方案所量身打造的平台，所以只要符合上述定義的平台即可稱為.NET平台
- 程式開發的複雜性降低
- 讓開發人員輕鬆的產生各種解決方案

.NET Framework

- .NET Framework 是微軟的幾個開發團隊一起努力發展的成果， 最主要用來產生一個可以用來快速開發、部署網站服務及應用程式的開發平台。
- .NET Framework 特色：
  - 透過網際網路的標準做整合
  - 鬆散的整合元件
  - 支援多種程式語言
  - 提高程式設計師的生產力
  - 嚴謹的安全機制
  - 使用作業系統服務

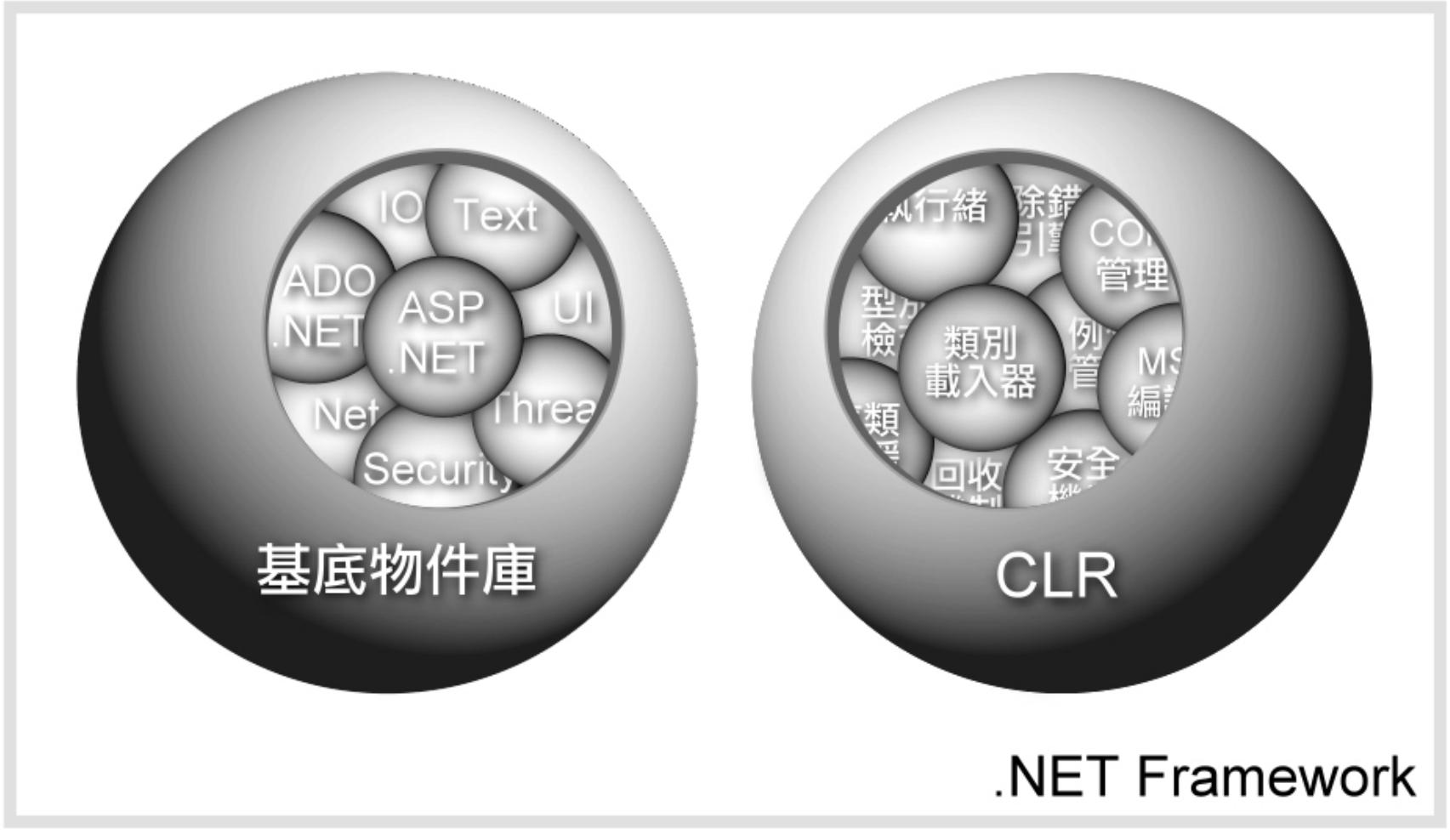
Microsoft.NET 平台架構圖



.NET Framewrok

- .NET Framework 是微軟最新推出的程式開發與動態網頁製作平台，它提供了.NET Framework 物件類別庫檔案，以便讓 .NET 相關系統軟體，例如 ASP.NET、VB.NET、C# … 都能去使用這個物件類別庫設計各類程式，或是編輯各種網頁
- .NET Framework 由兩大部分組成
  - 共通語言執行時期元件
  - 基底物件庫

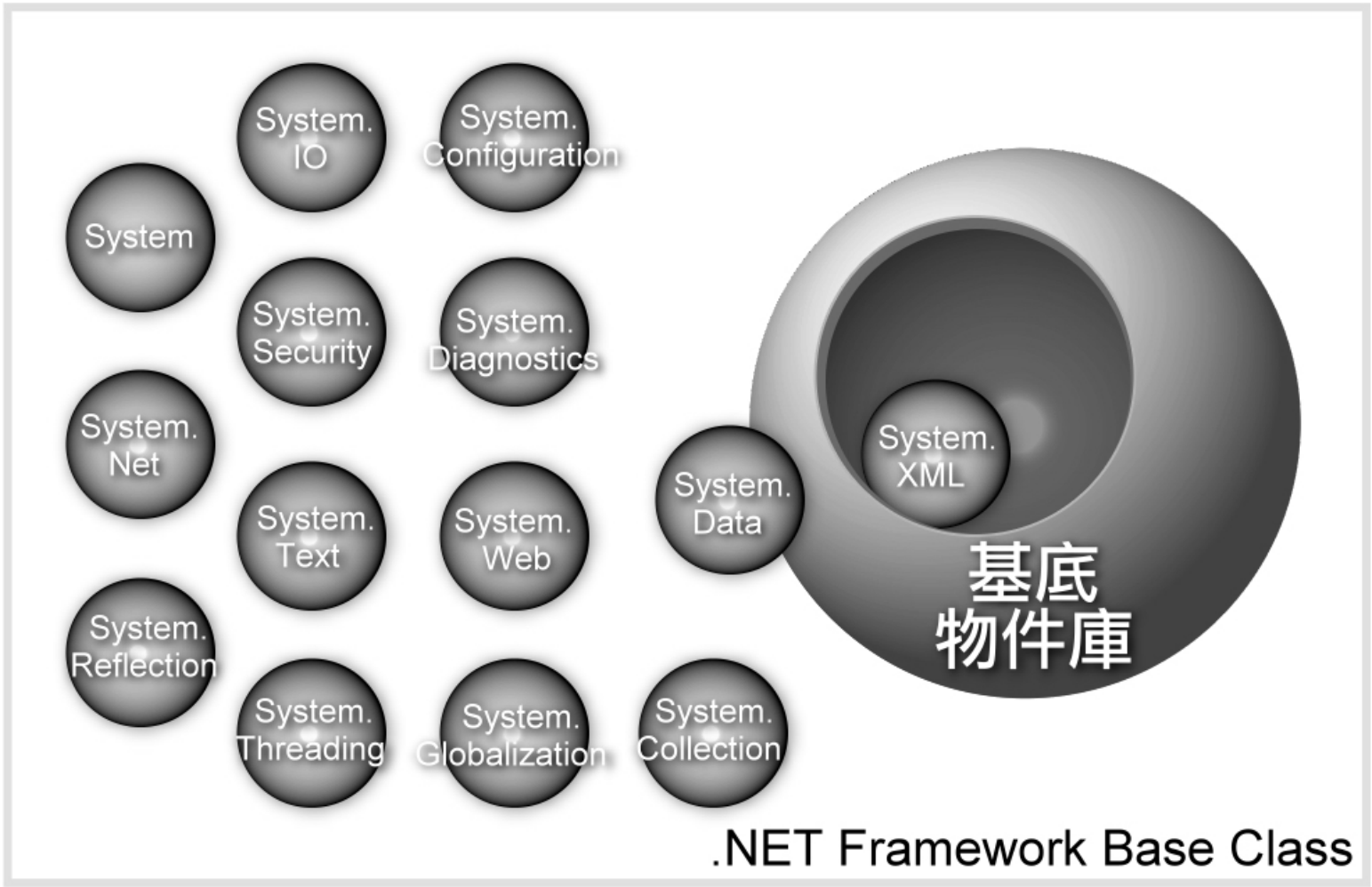
.NET Framewrok



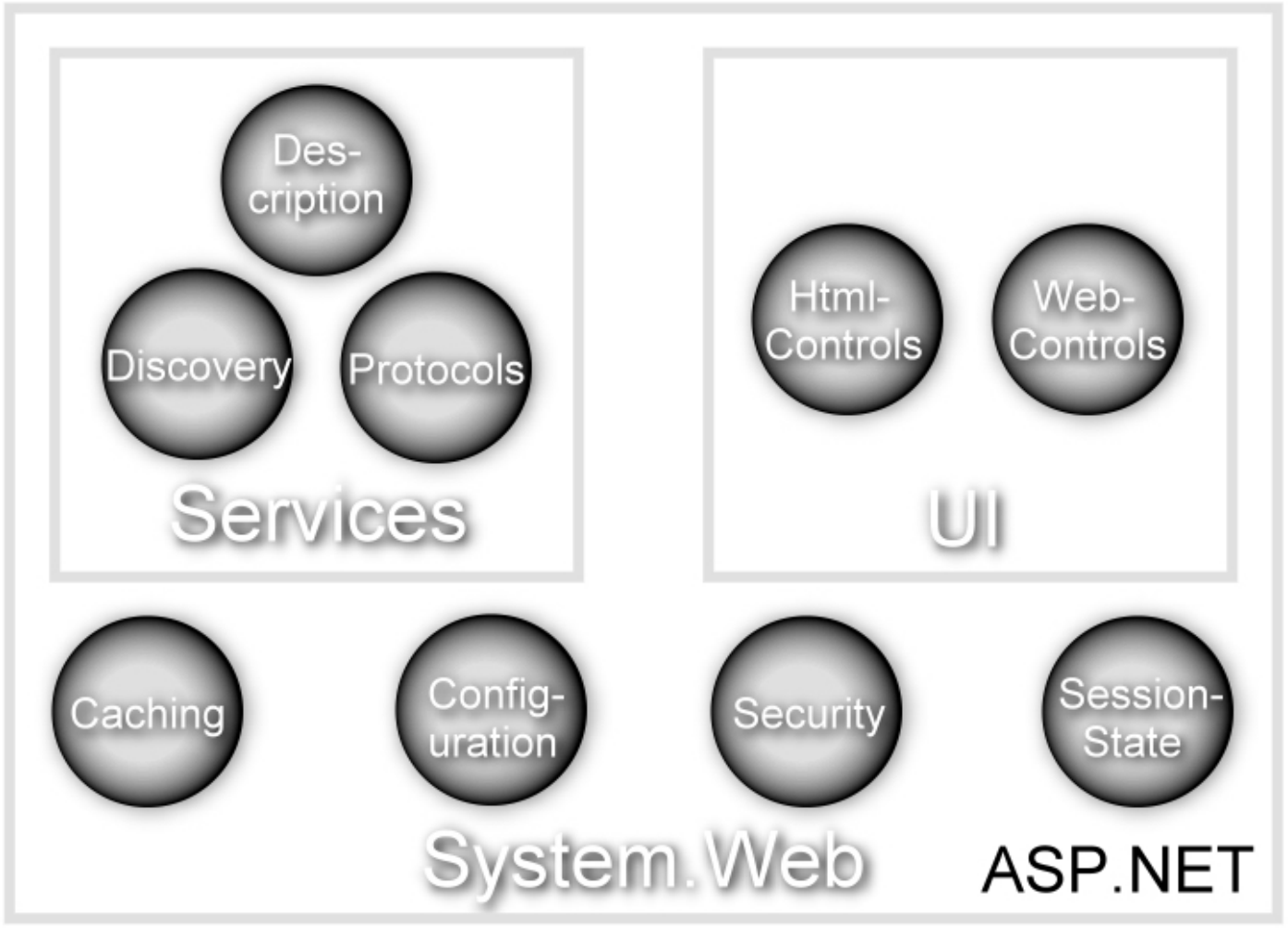
基底物件庫

- .NET Framework 提供了一個讓所有程式語言使用的基底物件庫，這個物件類別庫提供了統一、物件導向、結構化並可擴充的物件類別庫，來協助程式設計師快速的開發軟體。這些物件類別庫包括集合、IO、資料型別等。
- 程式設計師可以直接使用由.NET Framework基底物件庫所提供的物件，或者藉由繼承某個物件來擴充該物件的能力。

基底物件庫


System.
IO
System.
Configuration
System
System.
Security
System.
Diagnostics
System.
Net
System.
Text
System.
Web
System.
Reflection
System.
Threading
System.
Globalization
System.
Collection
System.
Data
System.
XML
基底
物件庫
.NET Framework Base Class

ASP.NET 所包含的基底物件庫



共通語言執行時期元件

- CLR 元件全名叫做「共通語言執行時期元件」（Common Language Runtime，簡稱為 CLR）。CLR 元件是一個高效率的執行引擎，程式碼執行時是由 CLR 元件所管理，CLR 元件負責的工作有產生物件、方法的呼叫、記憶體管理以及資源收集等。凡是支援.NET Framework的應用程式都是受管程式碼，其執行時一律都被CLR元件所管理。

共通語言執行的特性

- 跨程式語言整合（Cross-language Integration）
- 自我描述的元件（Self-describing Component）
- 簡化的部署與版本管理（Simple Deployment and Versioning）
- 整合的安全服務（Integrated Security Service）

Cross-language Integration

- **共通型別系統（Common Type System；CTS）是達成跨程式語言整合的關鍵。CTS定義了一組標準的型別和建立新型別的規則（CLR知道如何去產生和執行這些型別），編輯器（Compiler）或轉譯器（Interperter）則利用CLR提供這項服務去定義型別、管理元件，以及執行元件方法的呼叫，而不是利用開發工具或程式語言本身既有的機制。舊有的元件技術僅提供跨程式語言的互通（Interoperability），亦即採用不同程式語言撰寫的元件（Component），可以彼此呼叫；跨程式語言的整合，則是將程式碼再用的可能性更進一步提昇：不同語言撰寫的類別（Class）可以彼此繼承（Inheritance）、不同語言撰寫的程式間，例外（Exception）可以彼此傳遞。相對的開發工具（Visual Studio.NET）則提供了多重語言的專案發展環境，程式的除錯也可以在跨程式語言的執行中進行。**

Self-describing Component

- Metadata是元件能自我描述的關鍵。Metadata是編輯器在產生執行碼時，伴隨產生的定義性資訊，舉凡元件所使用的型別、屬性、方法、事件甚至輔助與備註等資訊都可包含在內，而且保證與執行碼的一致性，完全取代並且超越了傳統分離式的IDL（Interface Definition Language）檔案與型別庫（Type Library）所扮演的功能，同時元件服務要求與執行所需的資訊皆動態來自Metadata，傳統Proxy/Stub的機制也沒有存在的價值。

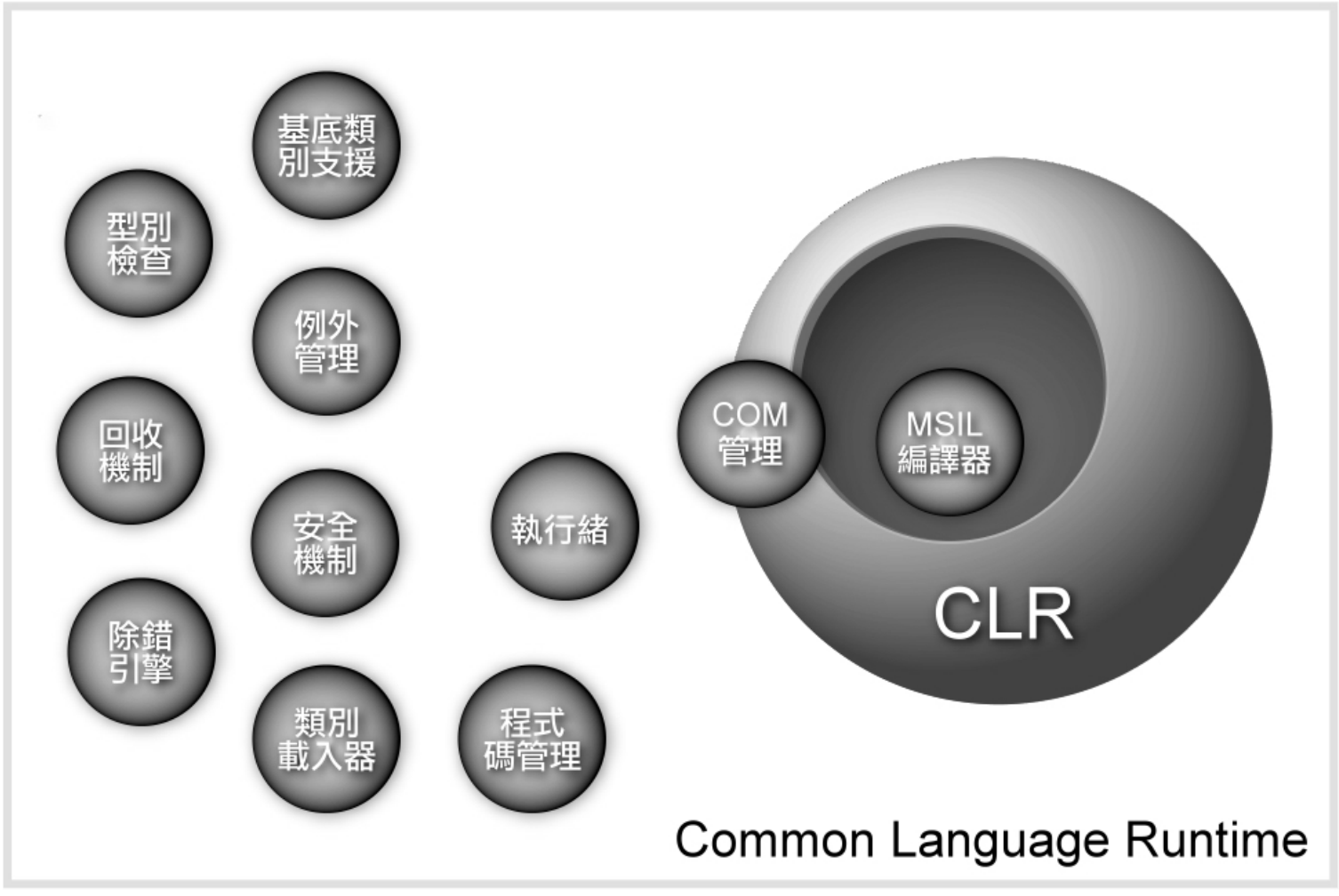
Simple Deployment and Versioning

- Assembly（組件）是簡化部署與版本管理的關鍵。Assembly是部署與版本管理的基本單元，其中包含一群資源（Resource）與型別（Type），以及它們內含的 Metadata，同時也包括此組合在建造時所依賴的其他組合的版本資訊。有了獨立完整的組態資訊，同一組件的不同版本也可安裝在同一部機器上，搭配共通語言執行環境具備根據各組件的組態資訊，載入正確版本的依賴組合（Dependent Assemblies）的能力，安裝與解除安裝的過程，就如同複製檔案與刪除檔案一樣單純，以往因先後安裝彼此覆蓋而產生的所謂“DLL Hell”的版本失控的現象不復存在。

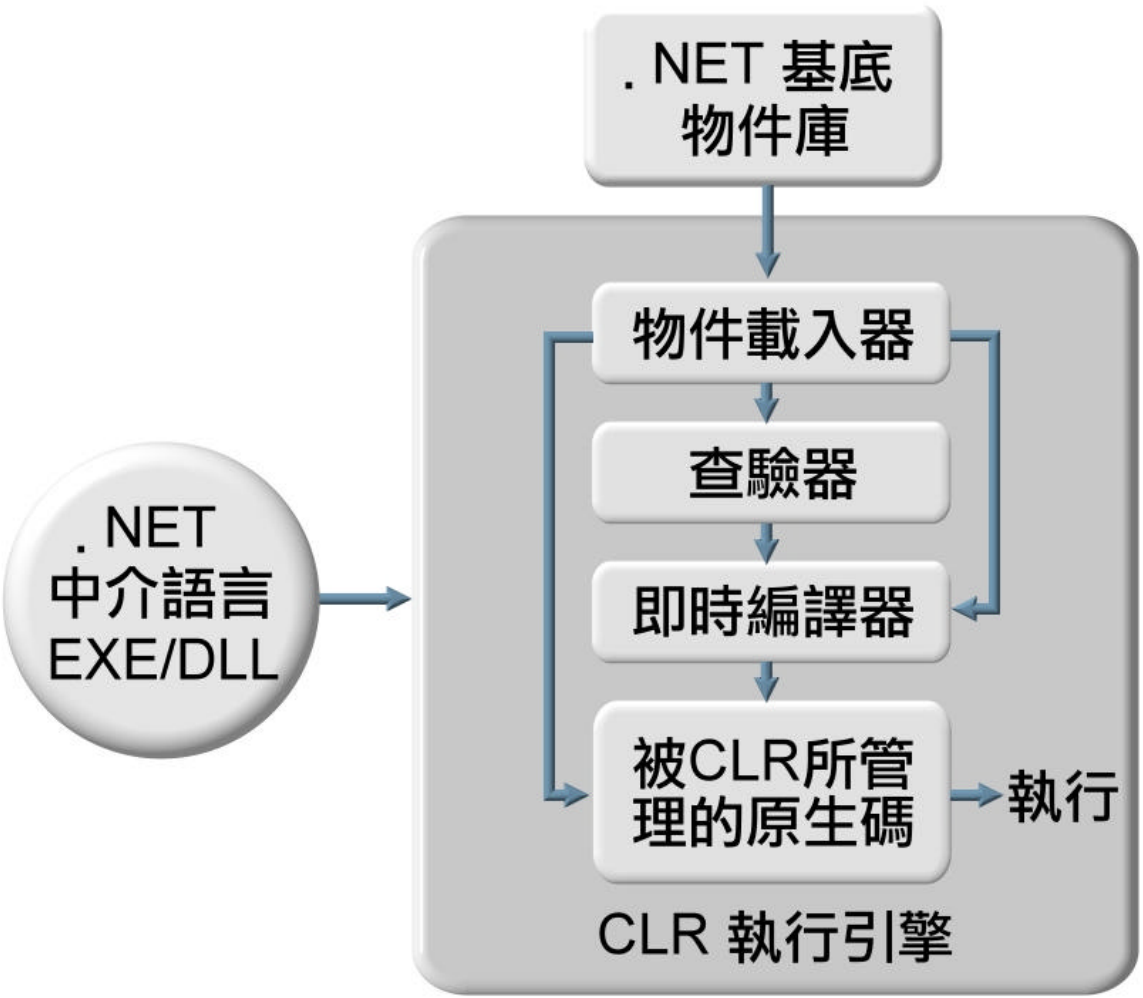
Integrated Security Service

- 安全服務可分為兩方面
  - 角色安全（Role-based Secuirty）
    - 角色安全的權限管理以使用者識別（User Identity）為基礎，資源存取的管制來自判定使用者所歸屬的角色是否被授權
  - 程式安全（Code Secuirty）
    - 程式安全的權限管理則是以程式碼識別（Code Identity）為基礎，搭配組件（Assembly）所描述的資源存取需求，以避免程式碼被他人作出超出授權範圍的運用

CLR 元件所負責的工作



CLR 執行引擎執行的流程

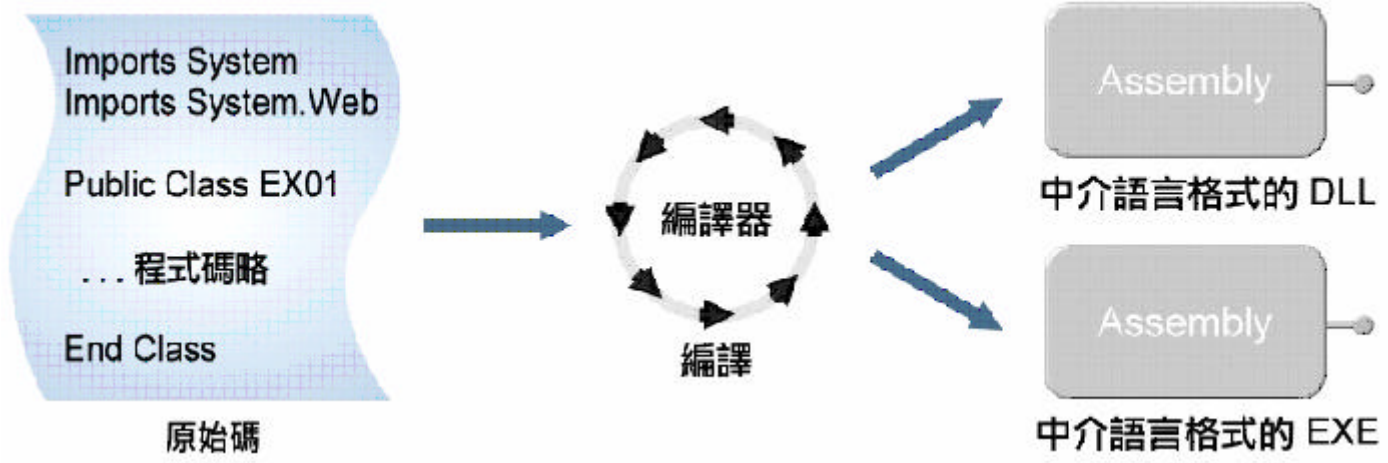


CLR 元件執行.NET 應用程式的步驟

1. Microsoft.NET 應用程式在執行時，會動態的連結到 CLR。
2. 此時若使用到 .NET Framework 基底物件庫的物件時，會由載入器將物件載入到記憶體中，並查驗該物件的型別、是否為原始物件，以及是否遭到竄改。
3. 查驗正確無誤後，即呼叫即時編譯器將程式立即編譯成原生碼（Native Code）。
4. 編譯成原生碼的程式，其執行時所需的記憶體管理、資源收集等工作，還是由 CLR 執行引擎所管理。

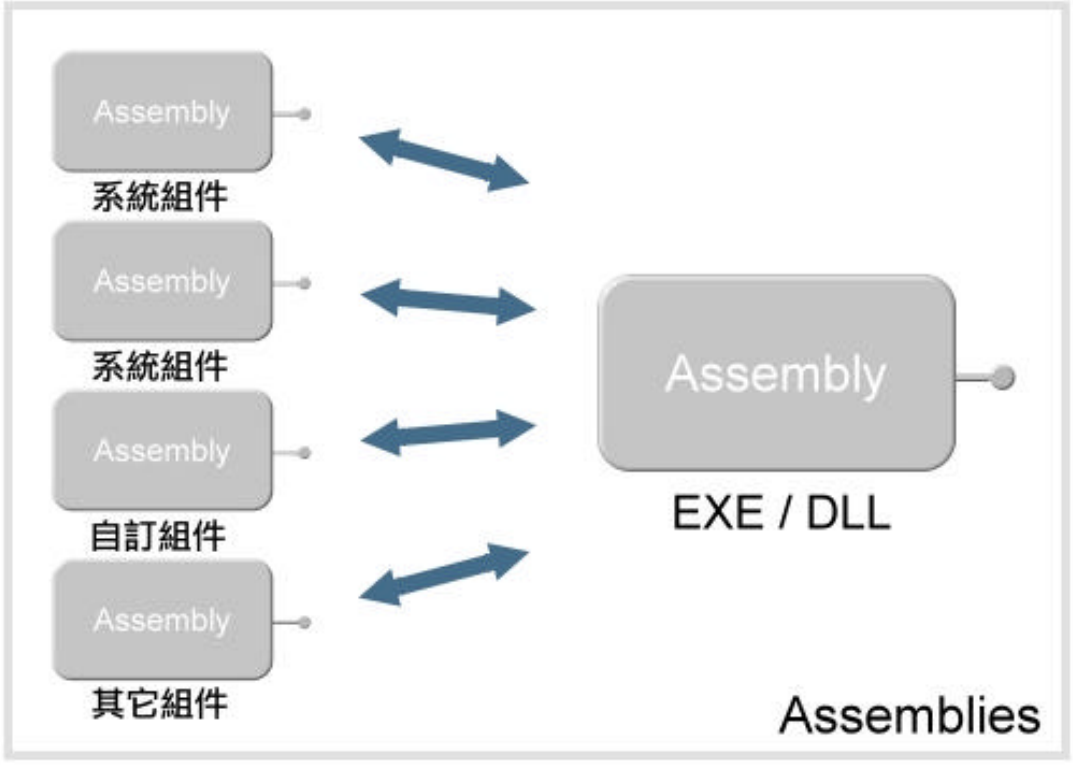
中介語言與即時編譯器

- 任何支援 .NET Framework 的程式撰寫完畢後，首先要經過「即時編譯器」（JITC，Just In Time Compiler）將原始碼編譯成「微軟中介語言（Microsoft Intermediate Language，MSIL）」



組件(Assembly)概念

- 一個 .NET Framework 應用程式是由許多「組件（Assembly）」所組裝成

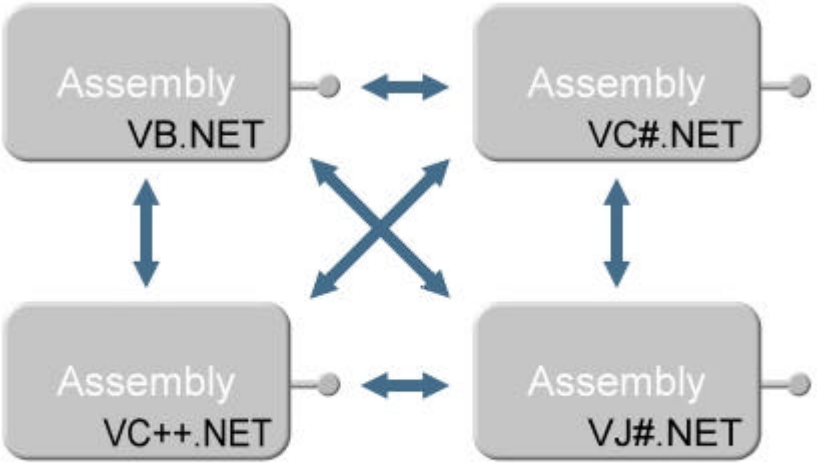


中介語言

- 編譯成中介語言有兩個主要的優點
  - 跨語言
    - 共通語言規範（Common Language Specification；CLS）
  - 跨平台

共同語言規範的特色

- 跨語言呼叫及存取
- 跨語言繼承



跨語言呼叫及存取

跨語言繼承

跨平台

- 中介語言可以跨越不同作業系統以及硬體平台
- 中介語言不是針對特定作業系統或硬體所設計
- 即時編譯器將中介語言即時編譯成該種硬體環境的原生碼。
- 可以解決程式開發人員長久以來在開發解決方案時所會面對的平台特性問題，大幅提高程式的可攜性以及程式開發人員的生產力。
- 中介語言透過即時編譯器編譯成原生碼後再執行，但是執行的效能並不會比較差。
- 中介語言的格式非常接近原生碼，直接由這種格式透過即時編譯成原生碼的速度非常的快。

Programming Model

- .NET Framework提供兩種低階的應用程式模型
  - 視窗式應用程式模型
    - 視窗式應用程式模型提供Win Form的程式模型，協助開發者建立傳統豐富化使用者界面（Rich User Interface）的視窗應用程式
  - 網頁式（Web）應用程式模型
    - 網頁式（Web）應用程式模型－ASP.NET，則是充分運用共通語言執行環境（CLR）與服務框架（Service Framework）的服務，除了透過高效能的HTTP Runtime與進階的狀態管理（State Management）等服務，讓傳統ASP的執行環境變得更加可靠（Reliable）和更具延展（Scalable）之外，進一步提供了兩項高階的應用程式模型：Web Form 與 Web Service

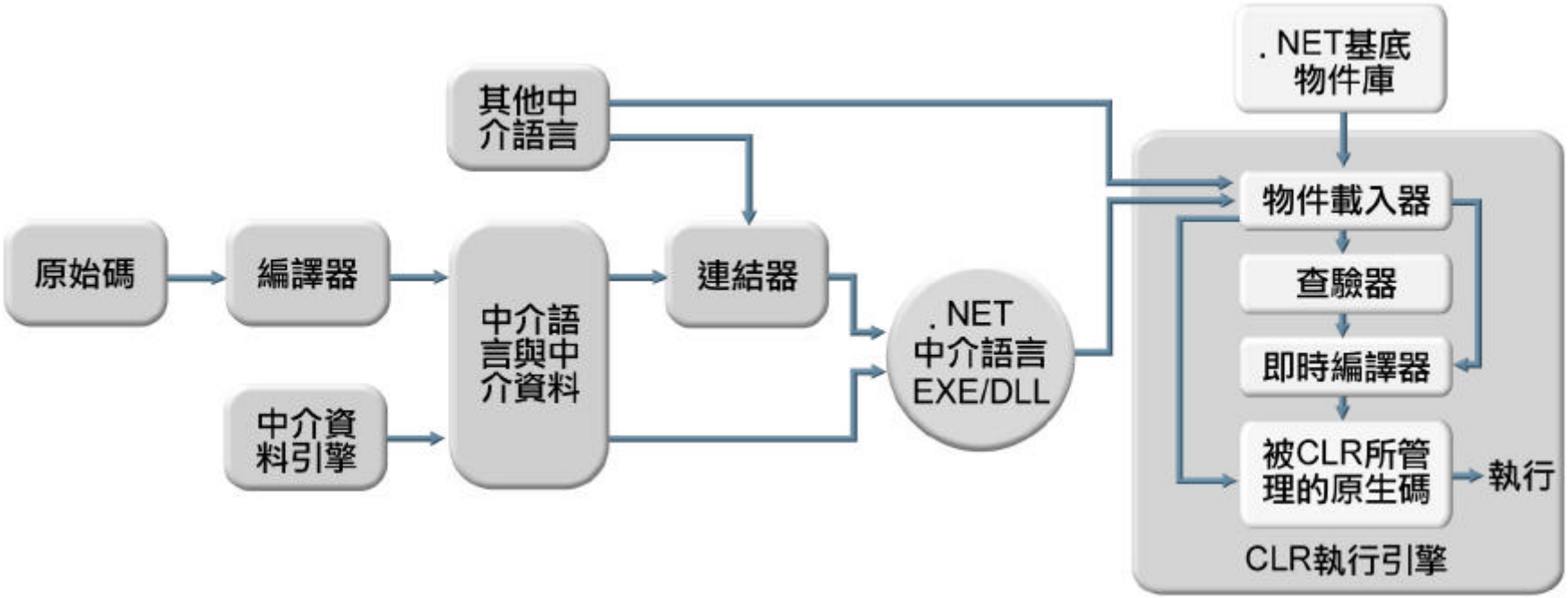
Web Form

- Web Form 所提供的元件化事件驅動式（Event-driven）的程式模型，讓網頁程式的開發也能達到 WYSIWYG 的高生產工作模式，同時產生的 HTML 版本更可通行於各種瀏覽器

Web Service

- 採用簡單、彈性與依據標準的模式來整合網際網路上的應用程式，因為傳統緊密耦合式（Tightly Coupled）的元件通訊技術/協定，如 DCOM, RMI, IIOP 等，主從端必須存在於同質的基礎設施（Infrastructure）之上，這樣的限制在網際網路多元的環境下是不符實際的要求，所以 Web Service 運用鬆散耦合式（Loosely Coupled）的元件通訊技術/協定，結合現有網際網路通行的標準 HTTP 與 XML，發展出 SOAP 作實現微軟網際程式網（Programmable Web）願景的基石，應用軟體將以黑箱的模式，透過 WSDL 來描述服務界面，同時運用 UDDI 的機制來公佈自己的網路服務或搜尋他人提供的網路服務

.NET 應用程式的執行流程

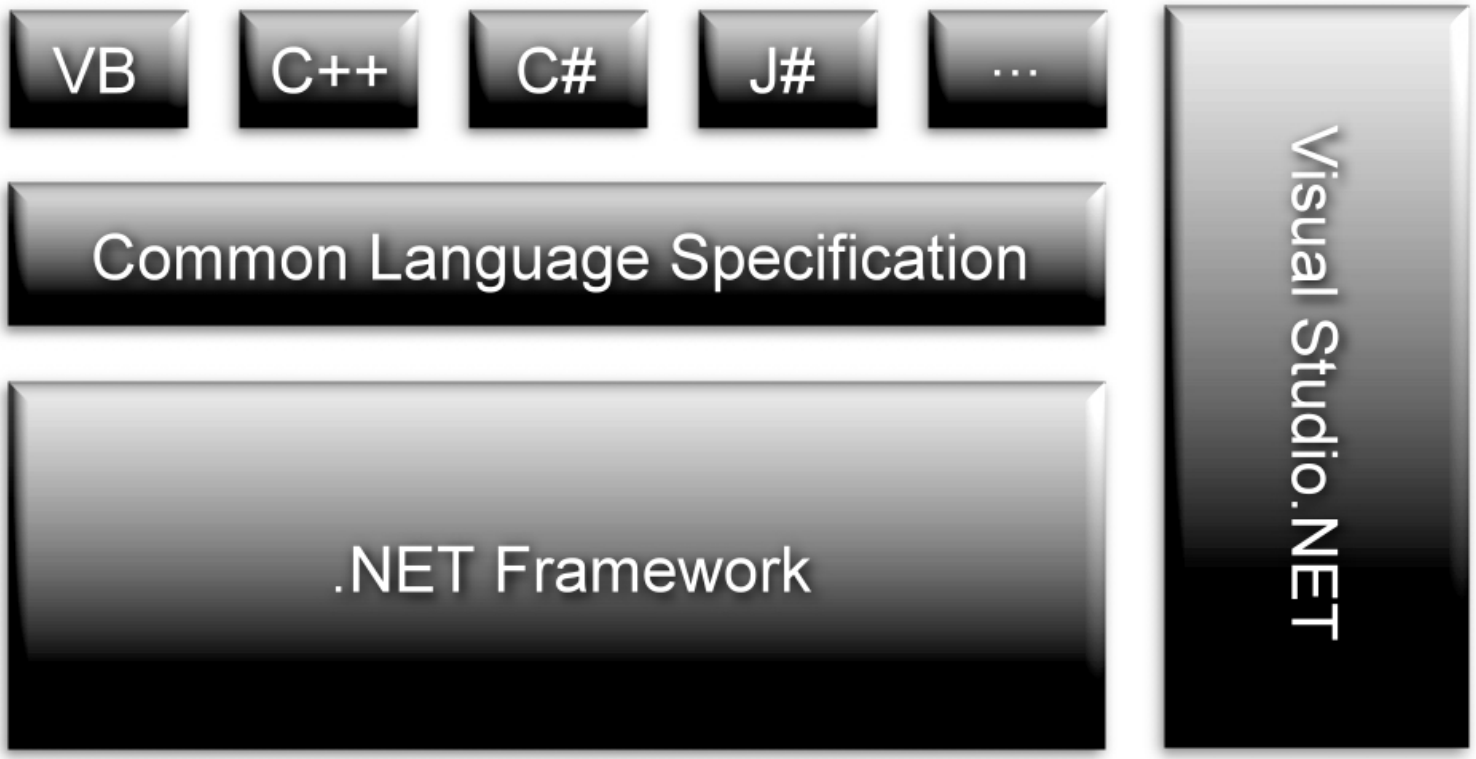


ASP.NET 開發工具

- Visual Studio.NET

  Visual Studio.NET是 Visual Studio 6 的下一個版本，也可以說是 Visual Studio 7 或 Visual Studio 2002 版，不過由於是特別針對 Microsoft.NET 平台所量身打造的開發平台，所以微軟將其取為更貼切的 Visual Studio.NET，也可以簡稱 VS.NET。

Visual Studio.NET Stack



ASP.NET 開發工具

- NET Framework Redistributable/SDK
  要開發 Microsoft.NET 應用程式除了使用 Visual Studio.NET 整合開發環境外，還可以只使用如小作家等一般文字編輯器，只要系統安裝了 .NET Framework Redistributable 版或 .NET Framework SDK 即可。.

ASP.NET 開發工具

- Microsoft Web Matrix
  由於許多開發人員只需要開發 ASP.NET 解決方案，故微軟特別針對這個需求開發了 Web Matrix軟體，提供 ASP.NET 開發人員除了 VS.NET 之外的選擇。
- Macromedia Dreamweaver MX
  習慣使用 Macromedia 解決方案的讀者，也可以選擇使用 Dreamweaver MX 作為開發 ASP.NET 網頁的工具。

建立 ASP.NET 開發平台

- 要建立 ASP.NET 平台需要的軟體如下
  - Windows 2K 以上 且安裝 Service Pack 2
  - IIS 5.0
  - .NET Framework SDK
  - Internet Explorer 6.0
  - MDAC 2.7
  - MSDE
    - MSDE（Microsoft SQL Server Desktop Engine）為存取 SQL Server 所需要的工具，倘若您機器已經安裝了 SQL Server，則 MSDE 可以省略。

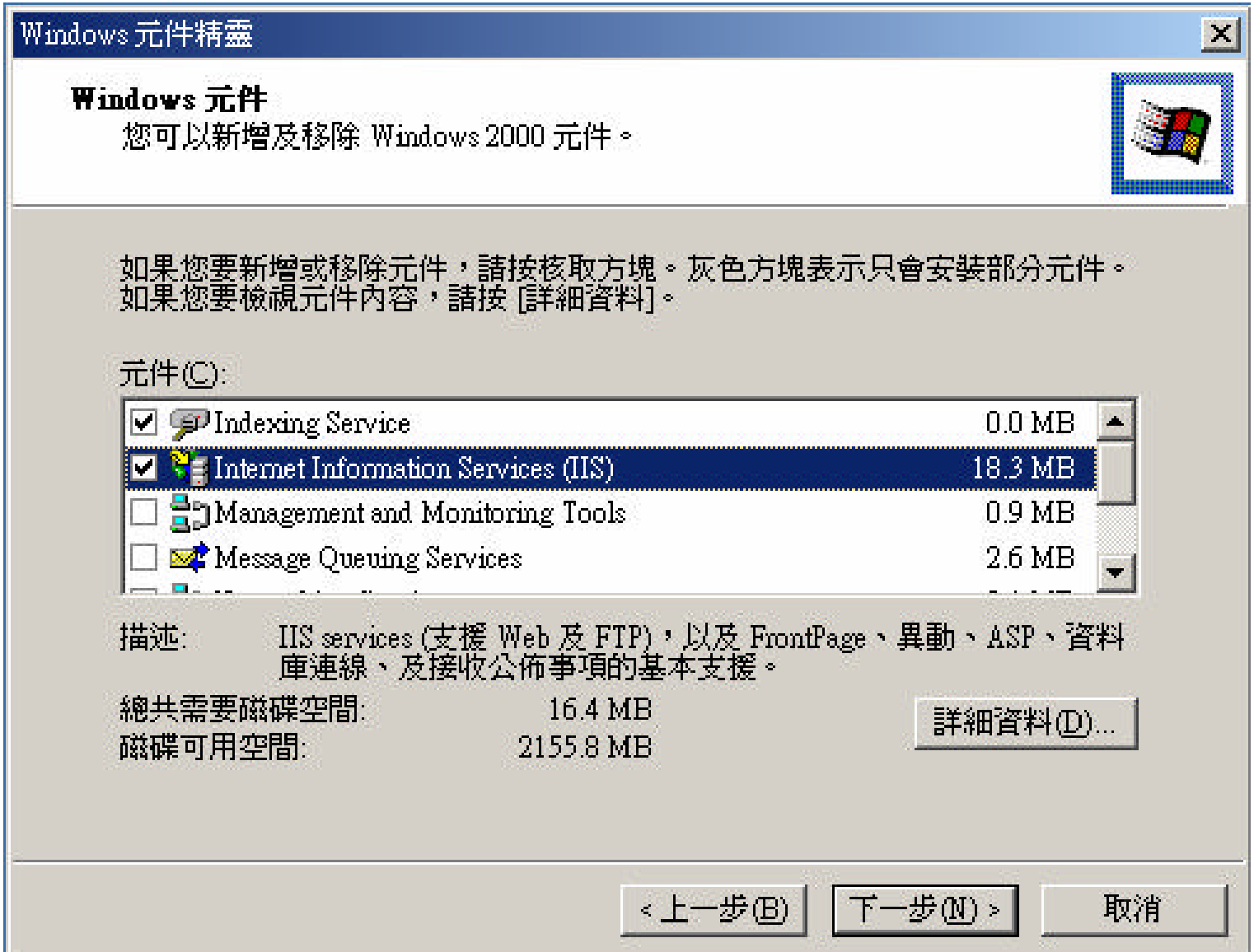
IIS

- IIS 最主要的功能大略為：
  - 回應用戶端的要求，將所要瀏覽的網頁內容傳輸給他們。
  - 管理及維護 Web 站台。
  - 管理及維護 FTP 站台。
  - SMTP（Simple Mail Transfer Protocol）虛擬伺服器。
  - 執行 ASP 的程式（要執行 ASP.NET 程式，需要安裝 .NET Framework SDK）。

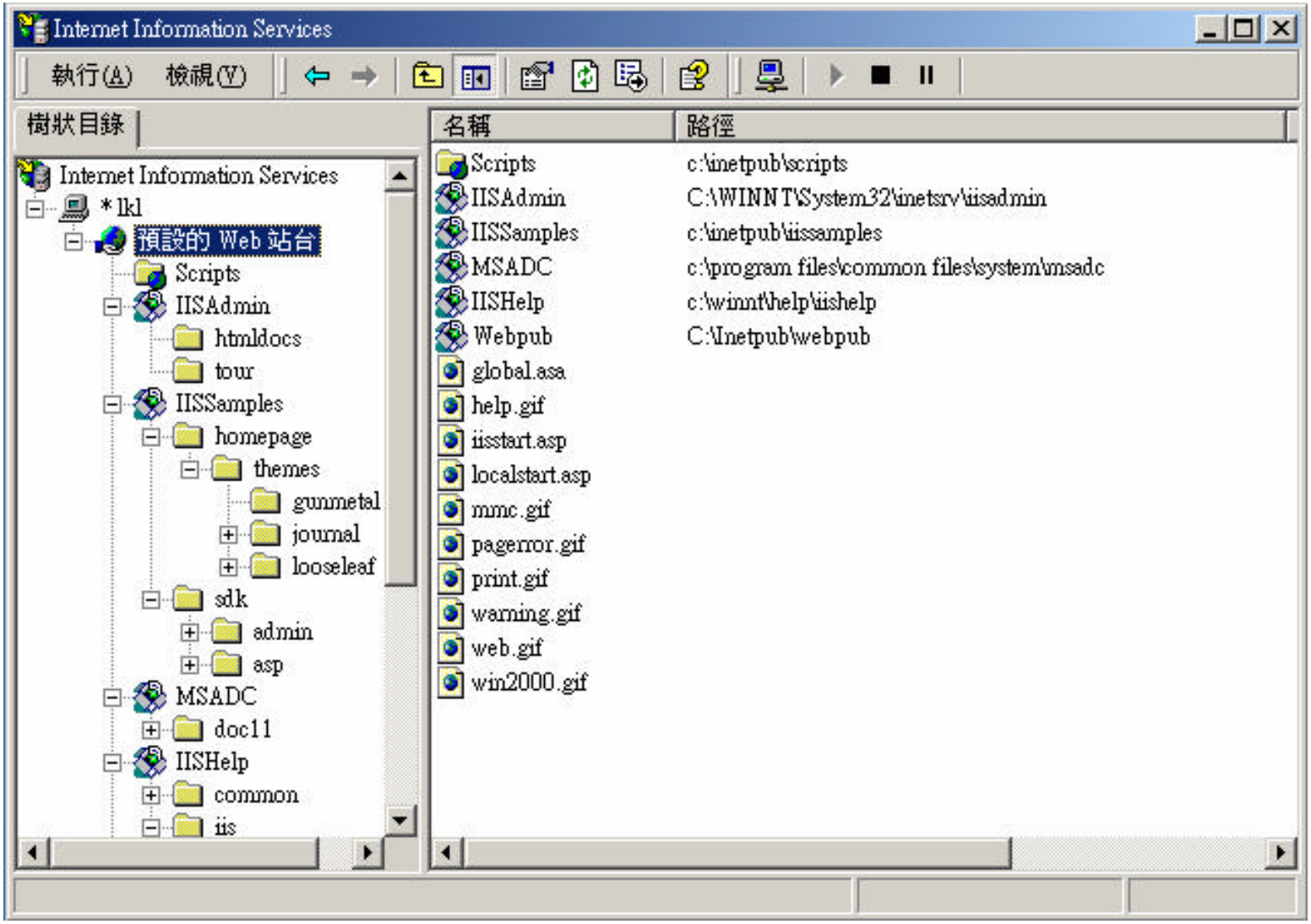
安裝 IIS


Microsoft Windows 2000 CD
Microsoft
Windows 2000
Professional
安裝 Windows 2000(I)
安裝附加元件(A)
瀏覽此 CD(B)
結束(X)
安裝附加元件
使用不同的選用元件，自訂您的 Windows 2000 操作方式。

安裝 IIS


Windows 元件精靈
Windows 元件
您可以新增及移除 Windows 2000 元件。
如果您要新增或移除元件，請按核取方塊。灰色方塊表示只會安裝部分元件。
如果您要檢視元件內容，請按 [詳細資料]。
元件(C):
Indexing Service 0.0 MB
Internet Information Services (IIS) 18.3 MB
Management and Monitoring Tools 0.9 MB
Message Queuing Services 2.6 MB
描述: IIS services (支援 Web 及 FTP)，以及 FrontPage、異動、ASP、資料庫連線、及接收公佈事項的基本支援。
總共需要磁碟空間: 16.4 MB
磁碟可用空間: 2155.8 MB
詳細資料(D)...
< 上一步(B)
下一步(N) >
取消

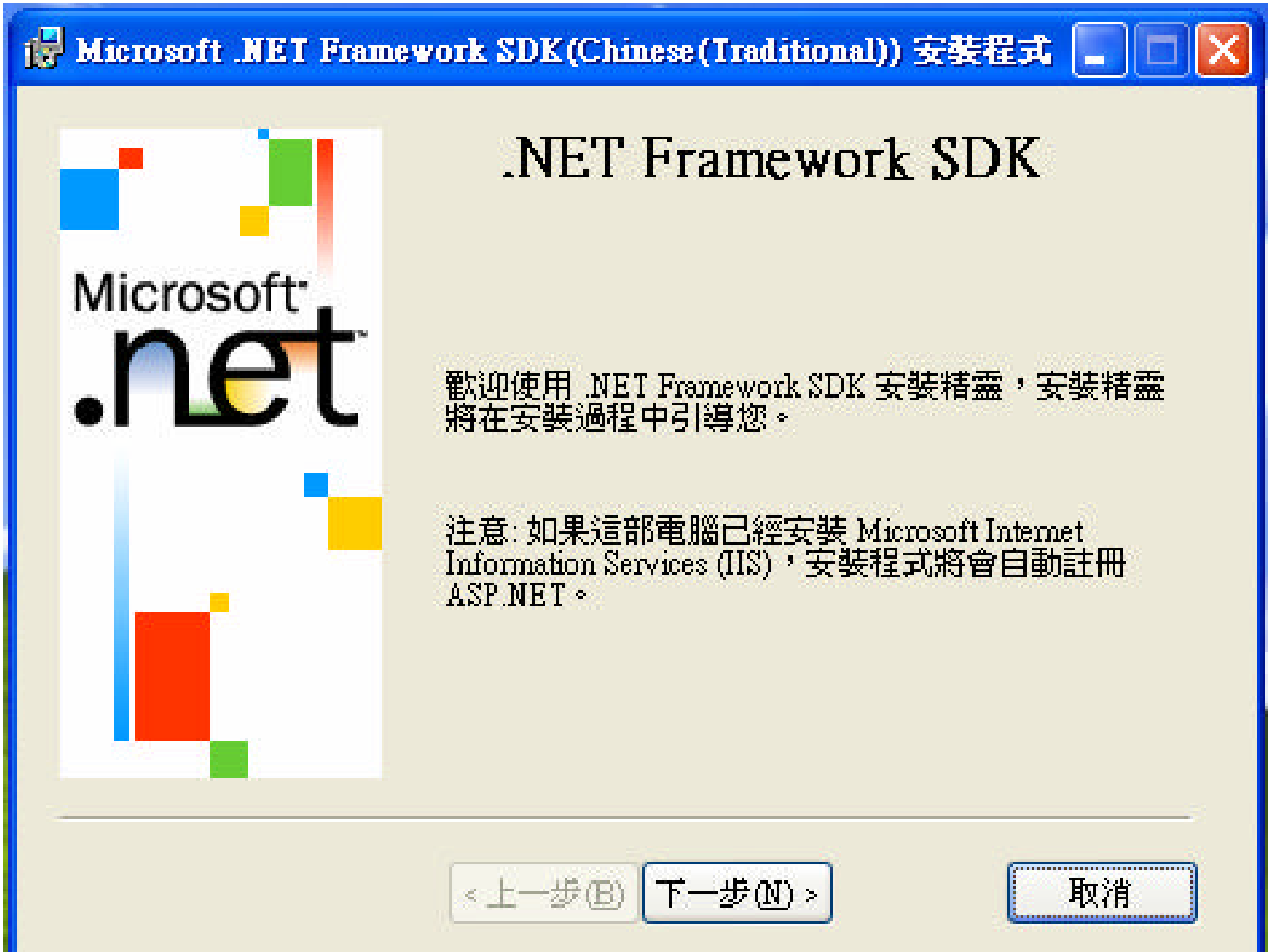
IIS 管理介面


Internet Information Services
執行(A)
檢視(V)
樹狀目錄
名稱
路徑
Internet Information Services
*lkl
預設的 Web 站台
Scripts
IISAdmin
htmldocs
tour
IISSamples
homepage
themes
gunmetal
journal
looseleaf
sdk
admin
asp
MSADC
doc11
IISHelp
common
iis
Scripts c:\inetpub\scripts
IISAdmin C:\WINNT\System32\inetsrv\iisadmin
IISSamples c:\inetpub\iissamples
MSADC c:\program files\common files\system\msadc
IISHelp c:\winnt\help\iishelp
Webpub C:\Inetpub\webpub
global.asa
help.gif
iisstart.asp
localstart.asp
mmc.gif
pagerror.gif
print.gif
warning.gif
web.gif
win2000.gif

IIS 的設定

- 主目錄
- 虛擬目錄
- Port number
- 目錄安全設定

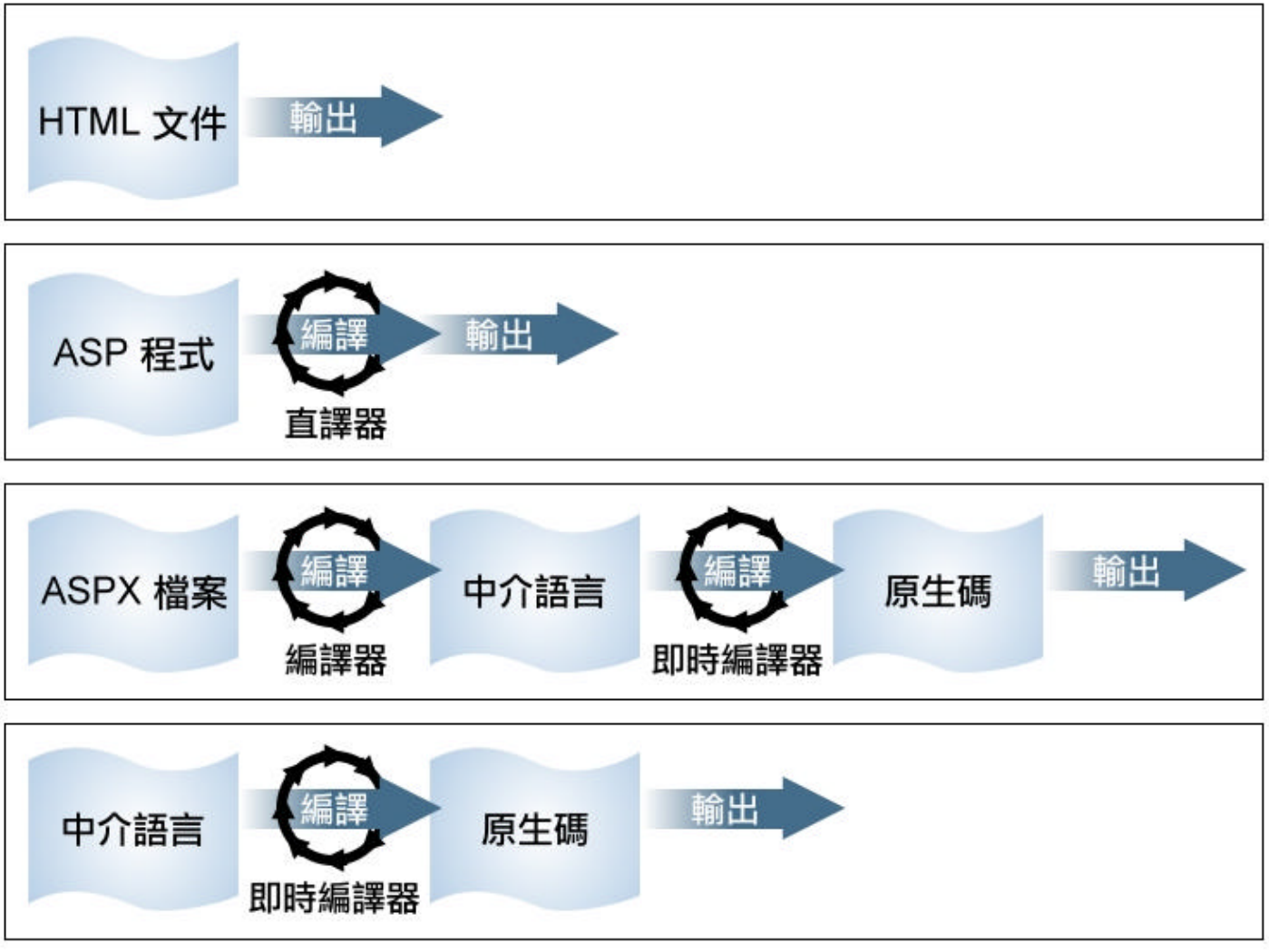
安裝 .NET Framework SDK



HelloWorld.aspx

```
<%@Page Language="C#"%>
<html>
<title>HelloWorld.aspx</title>
<%
Response.Write("Hello world in C#<br>");
Response.Write("The Time is: " + DateTime.Now);
%>
</html>
```

IIS 動作原理


HTML 文件
輸出
ASP 程式
編譯
輸出
直譯器
ASPX 檔案
編譯
中介語言
編譯
原生碼
輸出
編譯器
即時編譯器
中介語言
編譯
原生碼
輸出
即時編譯器

VB.NET 語法簡介

- Variable
- Sub and Function
- Flow Control

變數的資料型態

資料型別	儲存大小	說明	可儲存的資料範圍
Char	2Bytes	字元	0到65535
String	10Bytes加上（2乘以字串長度）	字串	0到大約2百萬個雙位元字元（Unicode）
Short	2Bytes	精簡整數	-32,768到32,767
Integer	4Bytes	整數	-2,147,483,648 到 2,147,483,647
Long	8Bytes	長整數	-,223,372,036,854,775,808 到 9,223,372,036,854,775,807
Single	4Bytes	單精浮點數	負數部分為-3.402823E38 到-1.401298E-45 正數部分為1.401298E-45 到3.402823E38

資料型別	儲存大小	說　　明	可　儲　存　的　資　料　範　圍
Double	8Bytes	雙精浮點數	負數部分為-1.7976931348623 1E308到-4.940656458412 47E-324，正數部分為4.940656458412 47E-323到1.7976931348623 2E308
Boolean	4Bytes	布林	True或False
Object	4Bytes	物件	任何型別的物件其記憶體位址都可以被物件型別的變數儲存
Decimal	16Bytes	數值	+/-79,228,162,514,264,337,593,543,950,335 以上未帶小數，若帶小數可存28位，為+/-7.9228162514264337593543950335 最小為+/-0.0000000000000000000000000001
Date	8Bytes	日期	西元1年1月1號至9999年12月31日
Byte	1Byte	位元	0到255

宣告變數

- 宣告變數
  VB.NET 是以「Dim」 關鍵字來宣告變數，並且在 AS 關鍵字後面指定所要使用的變數型別。倘若不指定變數的型別，VB.NET 則預設為 Object 型別。
- 宣告變數的語法
  - Dim 變數名稱 [As 資料型別]
  - Dim strUserName As String
  - Dim shtPrice As Short
  - Dim objPen As Object
  - Dim objPen

陣列

- **陣列的宣告語法**
  Dim 陣列名稱(元素上界限) [AS 資料型別]
  Dim shtAge(5) As Short
- **動態陣列**
  當無法精確的確定陣列所要使用的大小時，在宣告陣列的時候空出元素的上界限即可

  之後再使用 ReDim 敘述來重新配置陣列的大小。注意，ReDim是敘述不是宣告，故必須要先宣告任意一陣列後，才可以使用ReDim敘述。

動態陣列

```
Dim shtStudent() As Short
ReDim shtStudent(5)
```

陣列常用的屬性

屬　　性	說　　明
IsFixedSize	（唯讀）傳回陣列是否為固定大小
IsReadOnly	（唯讀）傳回陣列是否為唯讀
IsSynchronized	（唯讀）傳回是否同步存取陣列
Length	（唯讀）傳回陣列所有維度的元素總計
Rank	（唯讀）傳回陣列維度數
SyncRoot	（唯讀）傳回對陣列進行同步存取的物件

陣列常用的方法

方　　法	說　　明
BinarySearch	利用二進位搜尋法搜尋一維陣列內第一個符合指定資料的元素
Clear	設定陣列內指定範圍的元素，並依陣列型別設定指定範圍的元素其資料為0、False或Nothing
Clone	複製整個陣列
Copy	複製陣列內指定範圍的元素至另一個陣列
CopyTo	複製一維陣列所有的元素到指定的一維陣列
CreateInstance	產生一個新的陣列實體
GetLength	傳回陣列中指定維度的元素數目

陣列常用的方法

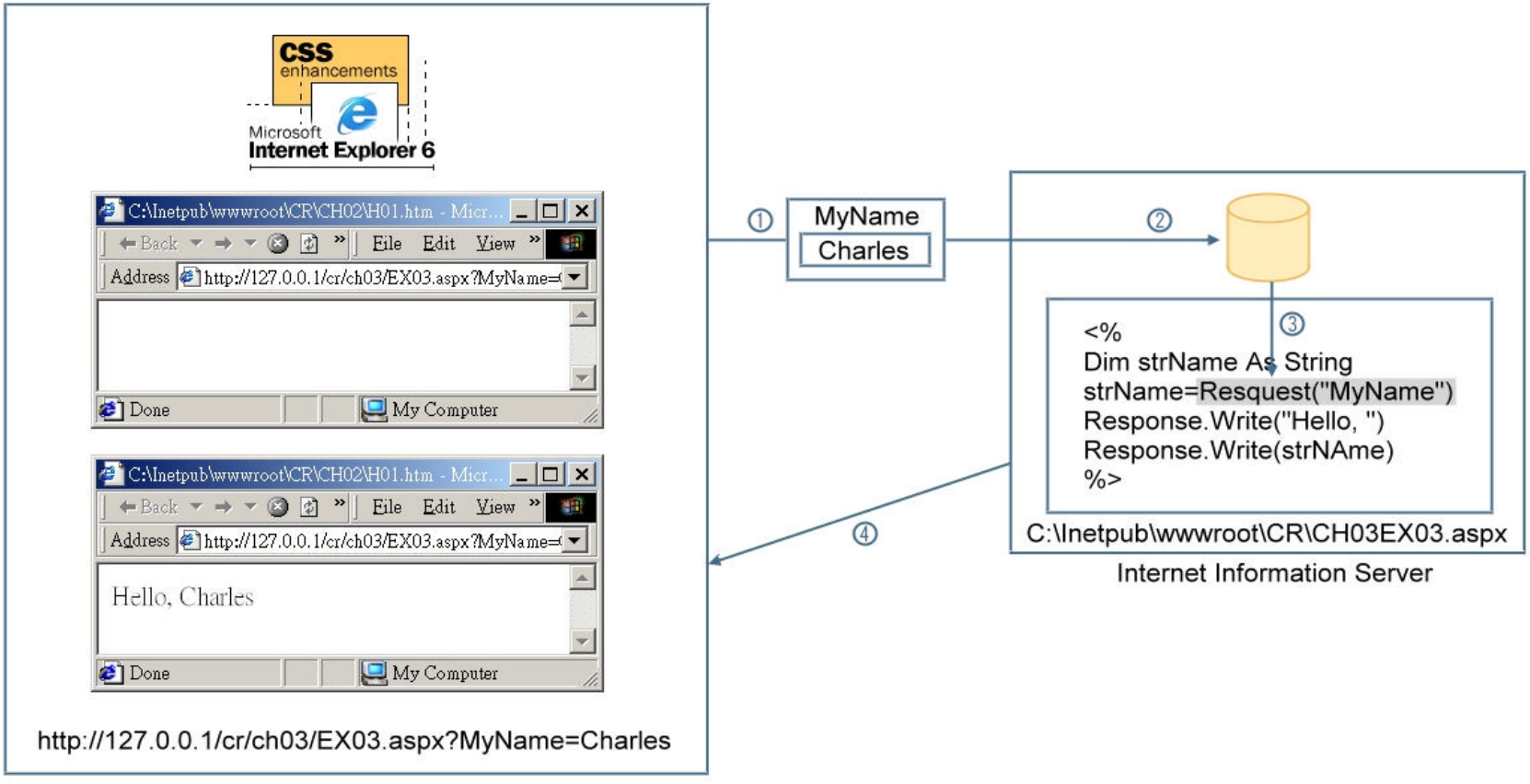
方　　法	說　　明
GetLowerBound	傳回陣列的下界限
GetUpperBound	傳回陣列的上界限
GetValue	傳回陣列中指定元素的值
IndexOf	搜尋一維陣列或陣列內指定範圍的元素內容，並傳回第一個符合條件的元素索引
LastIndexOf	搜尋一維陣列或陣列內指定範圍的元素內容，並傳回最後一個符合條件的元素索引
Reverse	反轉一維陣列全部的元素或某部分範圍元素的順序
SetValue	設定指定元素的內容為指定的值
Sort	排序一維陣列內的元素

參數上傳

- 客戶端要傳送資料給網頁伺服器，只要在網址後面加上**問號**，並將資料名稱以及指定的值填入即可，這個輸入的資料稱為「參數」，如下敘述所示：

```
http://127.0.0.1/cr/ch03/EX07.aspx?MyName=Charles
```

參數上傳的流程



① 在要求瀏覽aspx網頁，時一併傳遞輸入資料。

② 將資料傳給伺服端後，伺服端會將資料儲存於暫存區中。

③ 程式使用Resquest物件指定一個資料時，再將暫存區中使用者輸入之資料取出。

④ 使用Response物件的Write方法所輸出的內容下載至瀏覽器

設定IIS的編碼及解碼語系

- 要設定IIS伺服器對於接收及傳送資料的編碼語系，才能正確的顯示中文，這個設定檔案名稱為「web. config」。

```
<!--web.config Configuration File-->
<configuration>
<system.web>
<globalization requestEncoding="big5" responseEncoding="big5"/>
</system.web>
</configuration>
```

程序

- 程序的種類
  - 「一般程序」（General Procedure）
  - 「事件程序」（Event Procedure）
  - 「屬性程序」（Property Procedure）
- 一般程序
  一般程序有兩種，分別為Sub（副程式）以及Function（函式）。

Sub

- 以Sub方式寫成的程序稱為副程式，其執行完畢後沒有傳回值，也就是不會傳回執行的結果。
- 宣告為Public表示所有的程式都可以使用，而宣告為Private則表示只有這個網頁中的程式可以使用該程序，若沒有宣告則預設為Public。

Sub 的語法

```
[Public | Private] Sub 程序名稱(參數1 As 型別, …參數N)
    程式碼…
End Sub
```

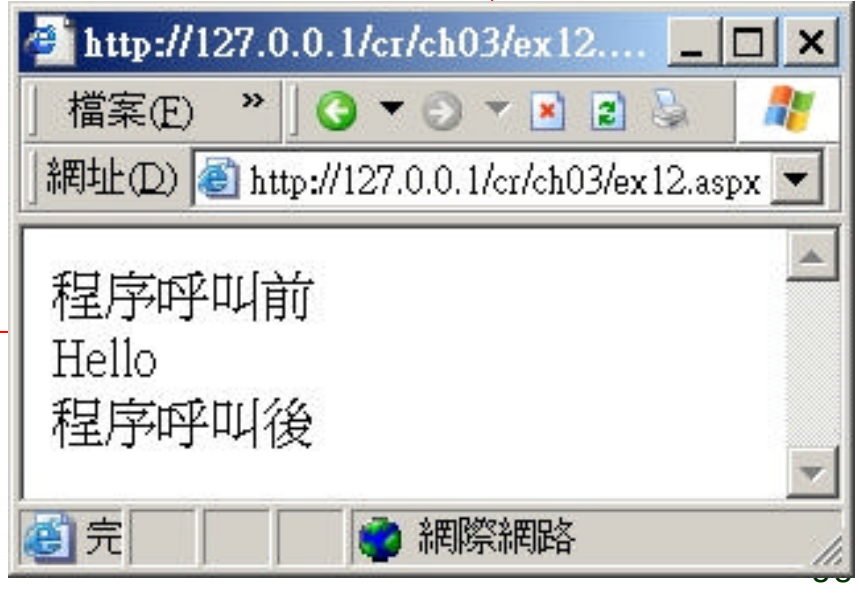
```
<script language="VB" runat="server">
Private Sub 程序一()
      程式碼…
  End Sub
Private Sub 程序二()
      程式碼…
  End Sub
</script>
```

<script> 標籤

- 在一.aspx檔中被 <script> 標籤所圍起來的部分表示為程序，不管是一般程序或是事件程序都必須被 <script> 標籤所圍起來，程序只有被呼叫或是發生一件事件的時候才會被執行。
- 執行程序的時候，只要直接輸入程序的名稱並且要加上小括號即可，這個使用程序的動作為「呼叫」（Call）。

程式遇到 Sub 時的執行流程

```
<html>
<%
Response.Write("程序呼叫前<br>")
SayHello()
Response.Write("程序呼叫後")
%>
<script language="VB" runat="server">
Private Sub SayHello()
  Response.Write("Hello<br>")
End Sub
</script>
</html>
```



Function

- Function 有「傳回值」（Return Value），也就是執行完畢後有執行結果可以接收，所以在使用 Function 時可以使用變數、物件的屬性來接收，甚至可以直接將 Function 直接拿來作為其他方法或其他 Function 的參數值。在 VB.NET 中不接收傳回值是被允許的。

Function 的語法

[Public | Private] Function 程序名稱(參數1 As 型別, …) [As 傳回值資料型別]
    程式碼…
    Return 傳回值 | 程序名稱 = 傳回值
End Function

Function

- Function也是一樣必須被Script標籤所括起來。
- Function一樣可以接收參數，Function程序執行後的結果可以利用Return關鍵字傳回（Return 傳回值）
- 使用Return以及將傳回值指定給程序名稱這兩種傳回資料方式的差異，在於使用Return會立即跳出Function程序；而使用指定傳回值給程序名稱的方式則會繼續將Function程序執行完畢。
- Function在宣告時的敘述「[As 傳回值資料型別]」即為明確宣告其傳回值的資料型別。

程式遇到 Function 時的執行流程

```
<html>
<%
Response.Write(Test())
%>
<script language="VB" runat="Server">
Private Function Test() As String
  Test = "Step 1<br>"
  Response.Write("Step 2<br>")
  Return "Step 4<br>"
  Response.Write("Step5<br>")
End Function
</script>
</html>
```

事件程序

- 事件（Event）是物件所認識的動作。對動態網頁來說，一頁動態網頁就是一個Page物件。
- 網頁每次在載入時會先發生Page_Init事件，然後再發生Page_Load事件，最後網頁內容全部下載完畢時則發生Page_Unload事件。
- Page_Init事件通常在Page物件被重新執行時，被ASP.NET網頁用來取得網頁的狀態。
- Page_Load事件由開發人員使用，用來初值網頁上的物件，或是使用者重新執行網頁時用來回復物件的狀態用。

事件程序的語法

```
[Public | Private] Sub 物件名稱_事件名(sender As Object, e As EventArgs)
    程式碼…
End Sub
```

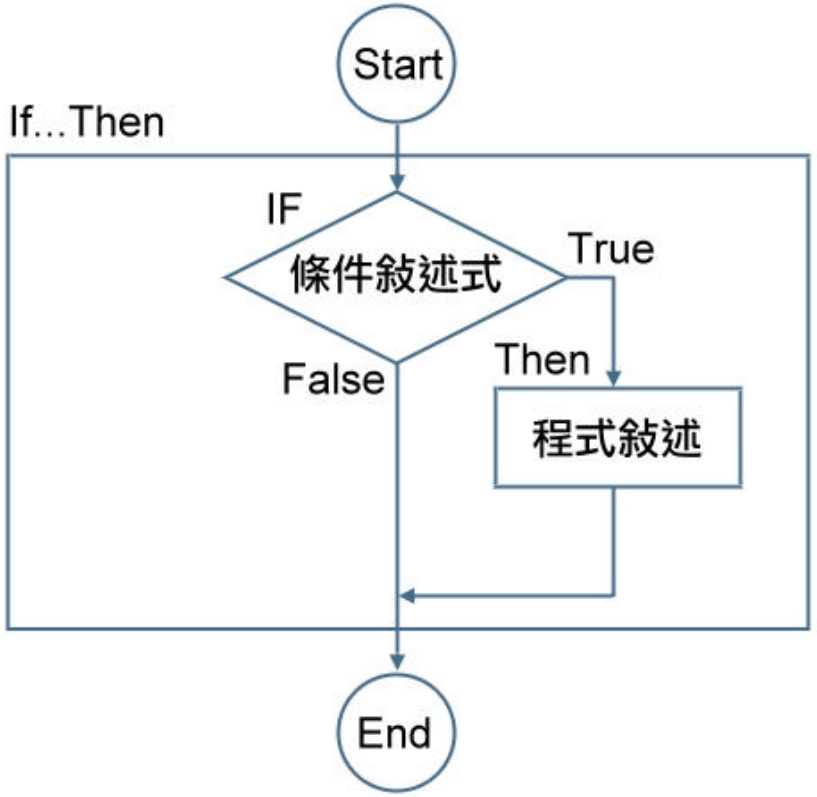
Page_Load 事件程序

```
<html>
<%
Response.Write("事件發生後")
%>
<script language="VB" runat="Server">
Private Sub Page_Load(sender As Object, e As EventArgs)
  Response.Write("網頁在載入時發生Load事件, 即自動執行本程序<br>")
End Sub
</script>
</html>
```

流程控制

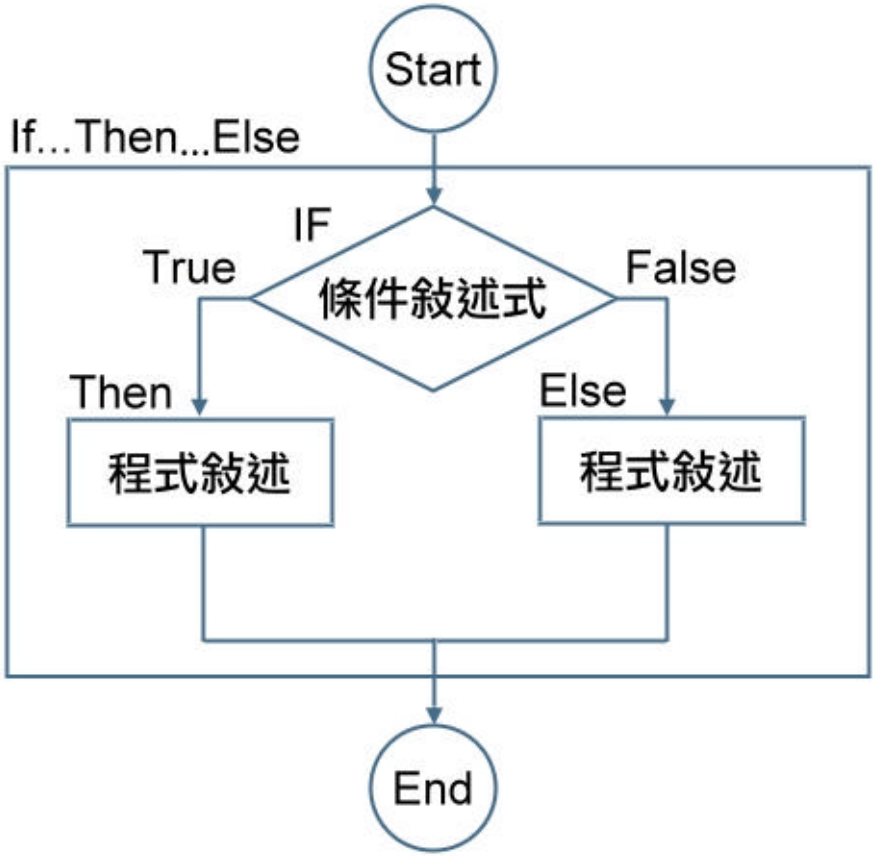
If 條件判斷 Then 程式敘述

```
If 條件判斷 Then
 程式敘述一
 程式敘述二
 程式敘述N…
End If
```



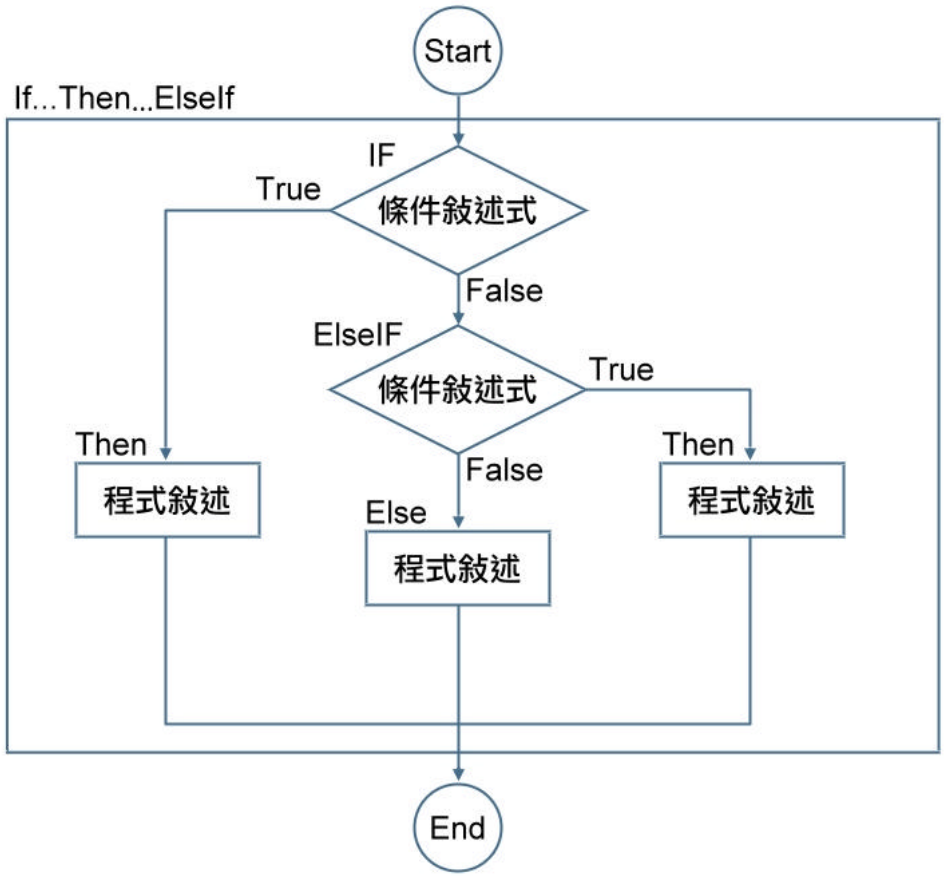
流程控制

```
If 條件判斷 Then
   程式區塊一
Else
  程式區塊二
End If
```



流程控制

```
If 條件判斷一 Then
   程式區塊一
ElseIf 條件判斷二
  程式區塊二
[Else
  程式區塊三]
End If
```



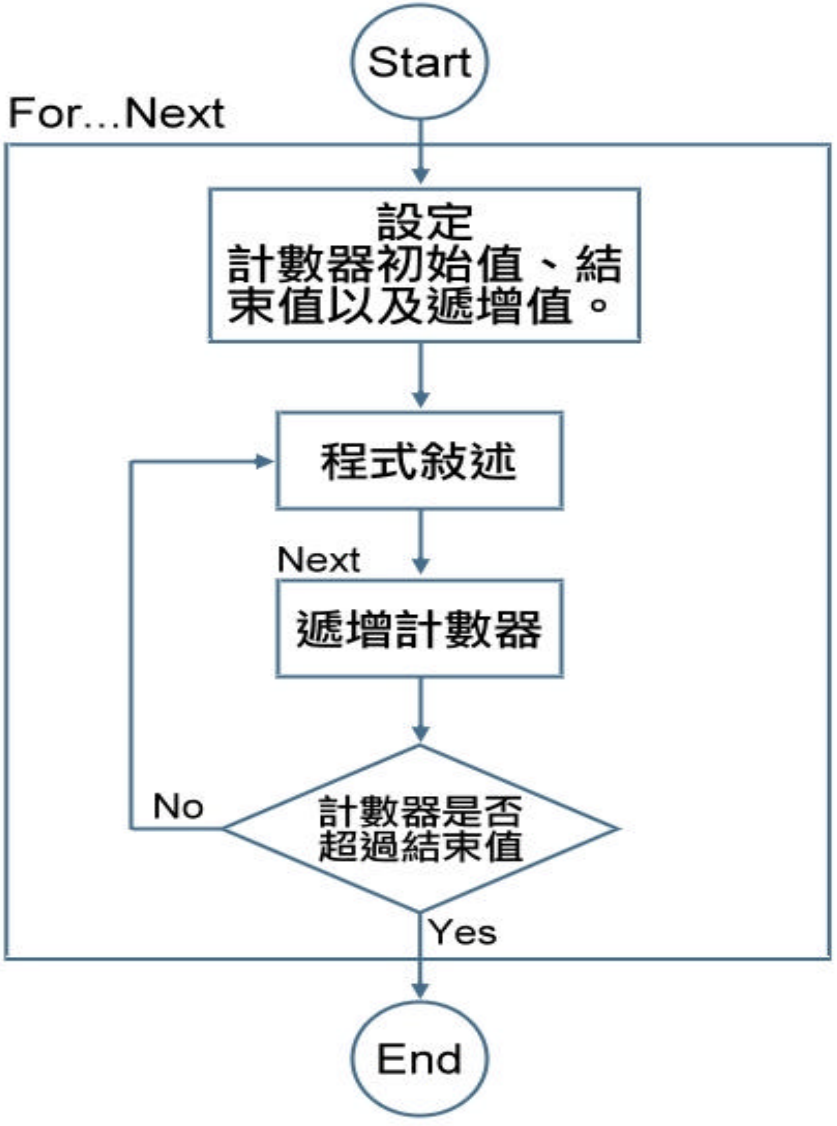
巢狀結構

```
If 條件判斷 Then
 【If 條件判斷 Then
     程式區塊
  Else
     程式區塊
  End If】
Else
  程式區塊
End If
```

迴圈

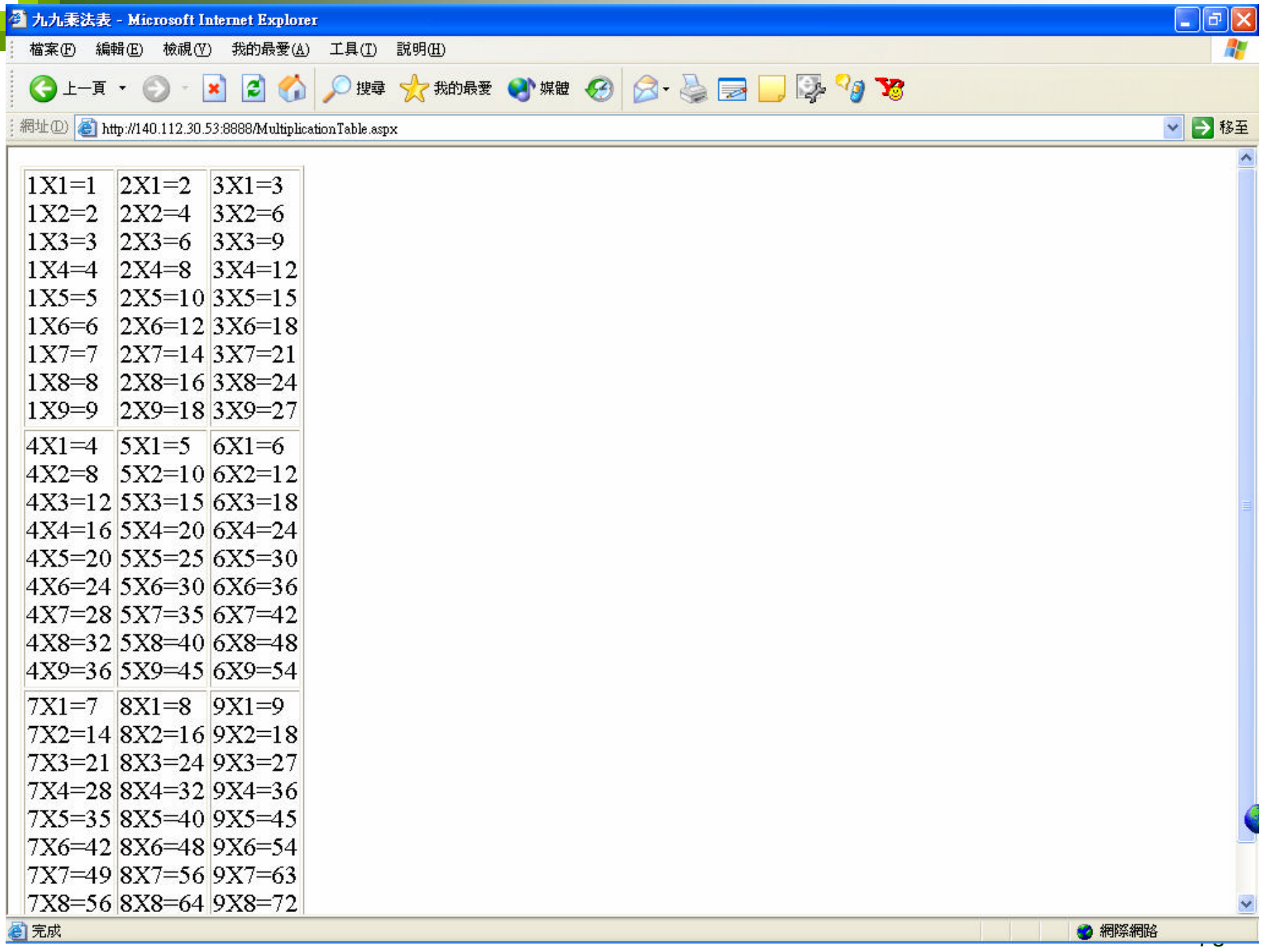
- For…Next 迴圈
  當知道一段程式碼所要執行的條件以及次數時，就可以使用For…Next迴圈。For…Next迴圈比Do…Loop迴圈容易使用及維護。

```
For 計數器=起始值 to 結束值 [Step 遞增值]
    程式敘述
Next [計數器]
```


Start
For...Next
設定
計數器初始值、結束值以及遞增值。
程式敘述
Next
遞增計數器
No
計數器是否超過結束值
Yes
End

九九乘法表

```
<html>
<title>九九乘法表</title>
<table border="1">
<%
Dim shtX,shtY As Short
For shtX=1 To 9 Step 1
  If shtX=1 OR shtX=4 OR shtX=7 Then Response.Write("<tr>")
  Response.Write("<td>")
  For shtY =1 To 9 Step 1
    Response.Write(CStr(shtX) & "X" & CStr(shtY) & "=" & _
                   CStr(shtX * shtY) & "<br>")
  Next shtY
  Response.Write("</td>")
  If shtX=3 OR shtX=6 OR shtX=9 Then Response.Write("</tr>")
Next shtX
%>
</table>
</html>
```


九九乘法表 - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://140.112.30.53:8888/MultiplicationTable.aspx
移至
1X1=1 2X1=2 3X1=3
1X2=2 2X2=4 3X2=6
1X3=3 2X3=6 3X3=9
1X4=4 2X4=8 3X4=12
1X5=5 2X5=10 3X5=15
1X6=6 2X6=12 3X6=18
1X7=7 2X7=14 3X7=21
1X8=8 2X8=16 3X8=24
1X9=9 2X9=18 3X9=27
4X1=4 5X1=5 6X1=6
4X2=8 5X2=10 6X2=12
4X3=12 5X3=15 6X3=18
4X4=16 5X4=20 6X4=24
4X5=20 5X5=25 6X5=30
4X6=24 5X6=30 6X6=36
4X7=28 5X7=35 6X7=42
4X8=32 5X8=40 6X8=48
4X9=36 5X9=45 6X9=54
7X1=7 8X1=8 9X1=9
7X2=14 8X2=16 9X2=18
7X3=21 8X3=24 9X3=27
7X4=28 8X4=32 9X4=36
7X5=35 8X5=40 9X5=45
7X6=42 8X6=48 9X6=54
7X7=49 8X7=56 9X7=63
7X8=56 8X8=64 9X8=72
完成
網際網路

HTML 控制項

- ASP.NET帶來了許多動態網頁的新技術，這些技術其中之一就是將所有的HTML標籤物件化，讓程式可以直接控制。物件化之後的HTML標籤稱為「HTML控制項」（HTML Control），HTML控制項可以使用如VB.NET或VC#.NET等語言來撰寫控制的程式。ASP.NET把HTML標籤物件化，可以讓網頁物件的互動、程式的寫作及維護變的更輕鬆容易，也增進網頁程式的執行的效能。

HTML 標籤物件化

```
<!--傳統的HTML靜態網頁寫作方式-->

<html>
<a href="http://www.microsoft.com">請按這裡</a>
</html>
```

```
<!--為了動態的設定標籤的屬性，必須在標籤中插入許多程式-->

<html>
<%
 strAddress="http://www.microsoft.com"
%>
<a href=<%=strAddress%>>請按這裡</a>
</html>
```

HTML 標籤物件化

- ASP.NET為了要解決這種雜亂無章的程式寫作風格，便將HTML標籤物件化而產生出HTML控制項。HTML控制項可以讓程式直接控制並設定其屬性，如下範例所示：

```
<html>
<a id="Anchor1" runat="server">請按這裡</a>
</html>
<script language="VB" runat="server">
Private Sub Page_Load(Sender As Object, e As EventArgs)
  Anchor1.Href="http://www.microsoft.com"
End Sub
</script>
```

HTML控制項基本觀念

- HTML控制項比HTML標籤多了「id」以及「runat」這兩種屬性。
  - 「Id」屬性表示程式是以本屬性的內容來控制物件的，所以**任何物件的名稱不可重複，不管它們是否為同一種類**。
  - 「runat="server"」**表示這個物件是在server端執行**，所有的HTML控制項都必須設定這個屬性值為「server」，若物件在程式執行時不需要被程式控制，則可以忽略id屬性的設定。

HTML控制項基本觀念

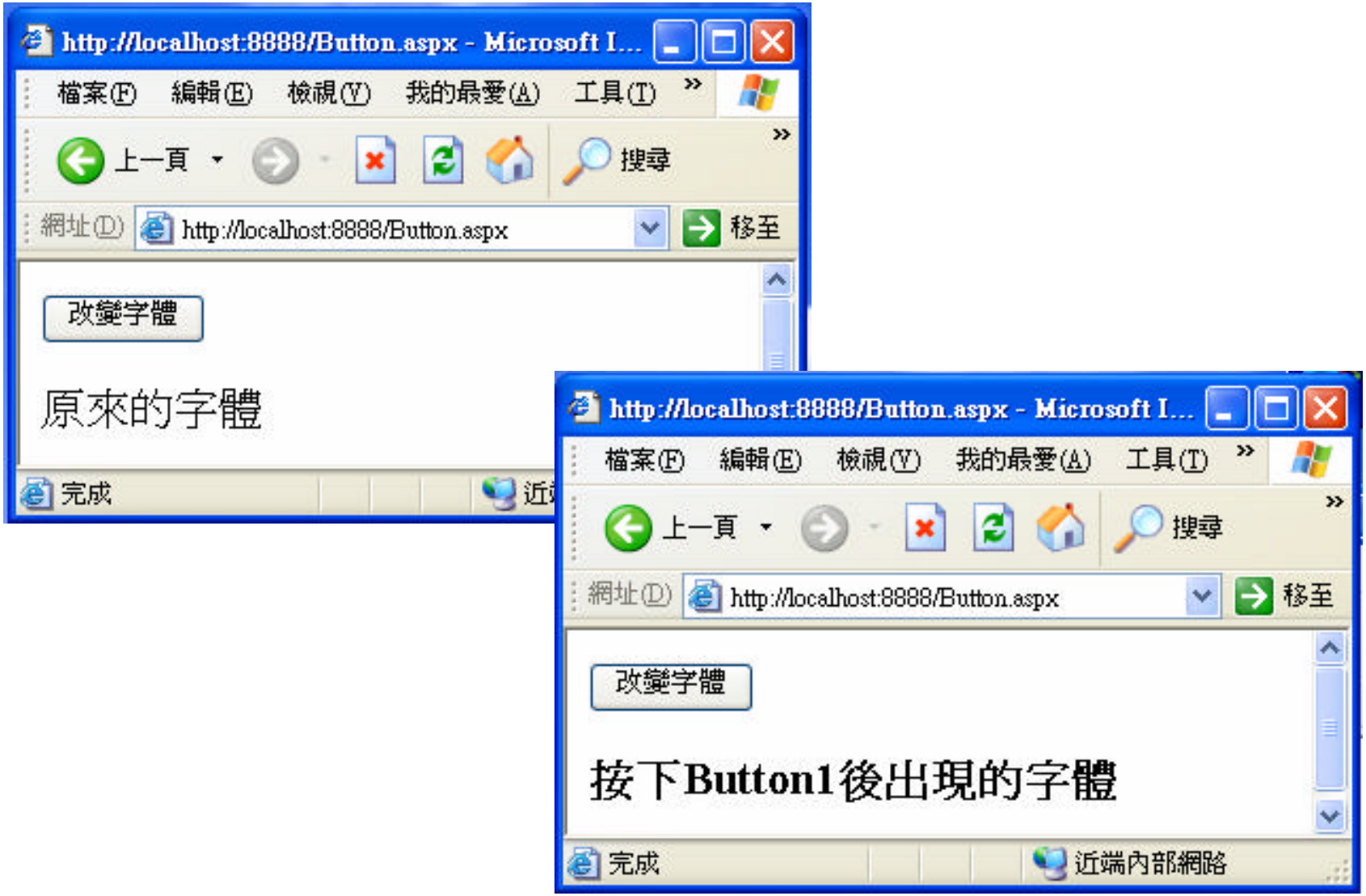
<標籤 id=控制項名稱 runat="server" 屬性1="值" 屬性2…>所要顯示的文字</標籤>

<標籤 id=控制項名稱 runat="server" 屬性1="值" 屬性2…/>

HTML控制項基本觀念

- 在ASP.NET中每個事件程序中都要加入「sender As Object, e As EventArgs」這兩個參數宣告

```
<html>
<form runat="server">
<button id="Button1" runat="server"
      OnServerClick="Button1_Click">改變字體</button><br>
</form>
<span id="Sp1" runat="server">原來的字體</span>
<script language="VB" runat="server">
Private Sub Button1_Click(Sender As Object,E As EventArgs)
  Sp1.InnerHtml="<b>按下Button1後出現的字體</b>"
End Sub
</script>
</html>
```


http://localhost:8888/Button.aspx - Microsoft I...
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T)
上一頁
搜尋
網址(D) http://localhost:8888/Button.aspx
移至
改變字體
原來的字體
完成
http://localhost:8888/Button.aspx - Microsoft I...
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T)
上一頁
搜尋
網址(D) http://localhost:8888/Button.aspx
移至
改變字體
按下Button1後出現的字體
完成
近端內部網路

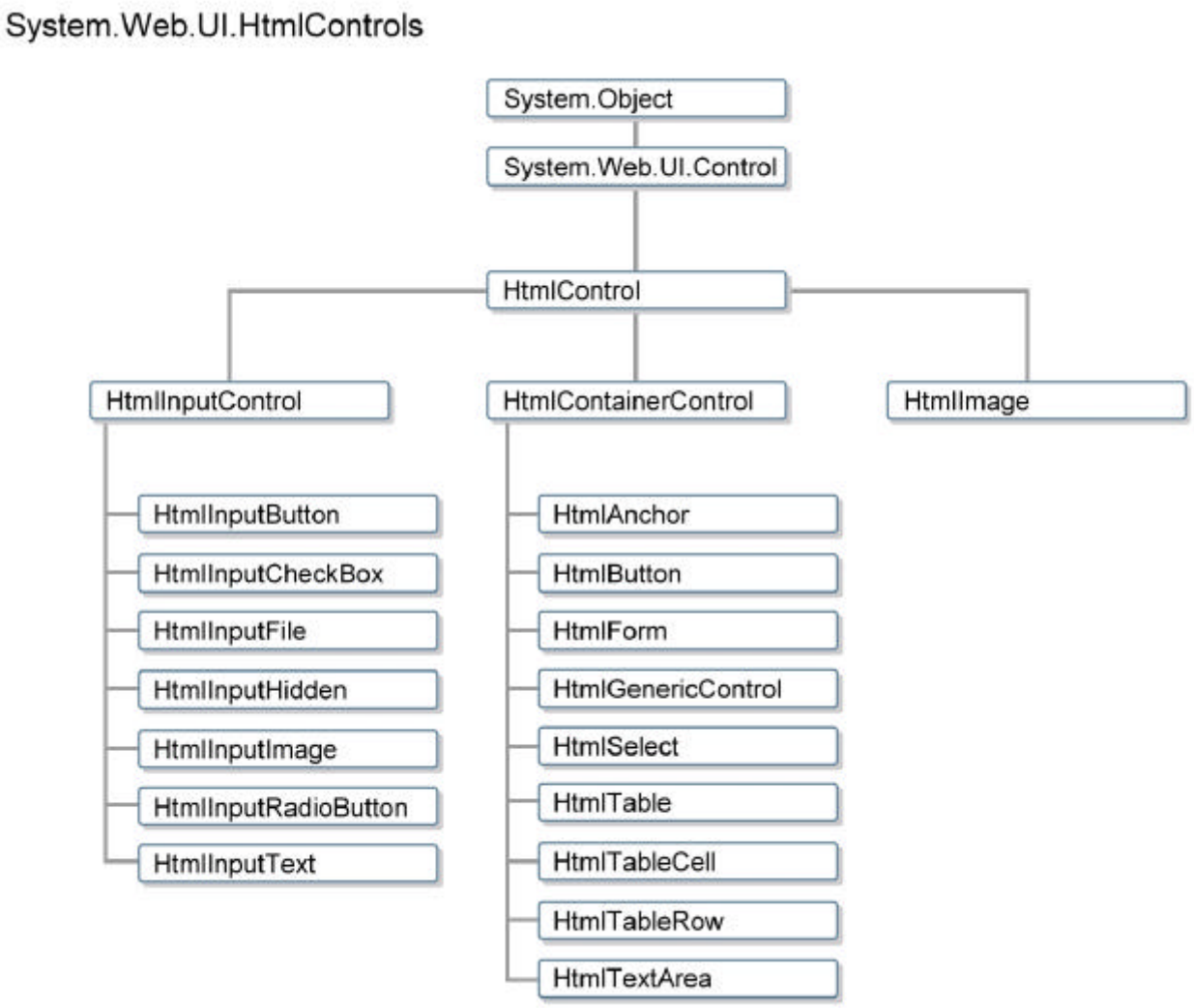
```
<html>
<form name="_ctl0" method="post" action="Button.aspx" id="_ctl0">
<input type="hidden" name="__EVENTTARGET" value="" />
<input type="hidden" name="__EVENTARGUMENT" value="" />
<input type="hidden" name="__VIEWSTATE"
value="dDwtMTU5NTE3Nzc1Mzt0PDtsPGk8Mz47PjtsPHQ8cDxsPGlubmVyaHRb
Ds+O2w8XDxiXD7mjInkuItCdXR0b24x5b6M5Ye654++55qE5a2X6auUXDwvYlw+O
z4+Ozs+Oz4+Oz7pCp3CAZcNU3We6xPvBdjI0zHj5w==" />
<script language="javascript">
<!--
        function __doPostBack(eventTarget, eventArgument) {
                var theform = document._ctl0;
                theform.__EVENTTARGET.value = eventTarget;
                theform.__EVENTARGUMENT.value = eventArgument;
                theform.submit();
        }
// -->
</script>

<button language="javascript" onclick="__doPostBack('Button1','')" id="Button1">改
變字體</button><br>
</form>
<span id="Sp1"><b>按下Button1後出現的字體</b></span>
</html>
```

HTML 控制項執行流程

- 當ASP.NET網頁執行時，會先由編譯器檢查標籤有無runat="server"屬性設定
  - 如果標籤沒有設定這個屬性，那麼該標籤就會被視為HTML文件內容。
  - 如果標籤有設定這個屬性，則會依照該標籤所對應的類別產生物件。也就是說，會**將該類別從.NET Framework基底類別庫實作到記憶體中並等待程式的執行**。等到程式執行完畢後再將HTML控制項的執行結果轉換成HTML標籤，然後送到字串流等待傳送至客戶端瀏覽器進行解譯。

HTML 控制項執行流程

HTML控制項的架構圖



Web控制項簡介

- **Web控制項基本概念**
  - Web控制項（又可稱為Server控制項）和HTML控制項不一樣。
  - Web控制項的功能比較強，它會依 Client 端的狀況產生一個或多個適當的 HTML 標籤，並可以依 Client 端瀏覽器的種類自動調整成適合瀏覽器的輸出。
  - 可以和資料源進行資料繫結（Data Binding）的工作。

Web控制項簡介

- Web控制項的基本架構
  - 好處在於發展分散式，並混合標準視窗應用程式以及網際網路應用程式的解決方案時，可以因為這些控制項的行為及架構非常相似，可以讓我們轉移舊有的知識到新環境上。
  - 這些控制項的行為及架構非常相似，所以不用再浪費時間學習那些功能一樣但架構不一樣的物件上。

Web控制項分為四種類型

名稱	內容
內建控制項（Intrinsic Control）	對應一般的HTML元素
清單控制項（List Control）	提供選項給使用者選擇
豐富控制項（Rich Control）	提供許多以前沒有提供的控制項
驗證控制項（Validation Control）	提供驗證資料的控制項

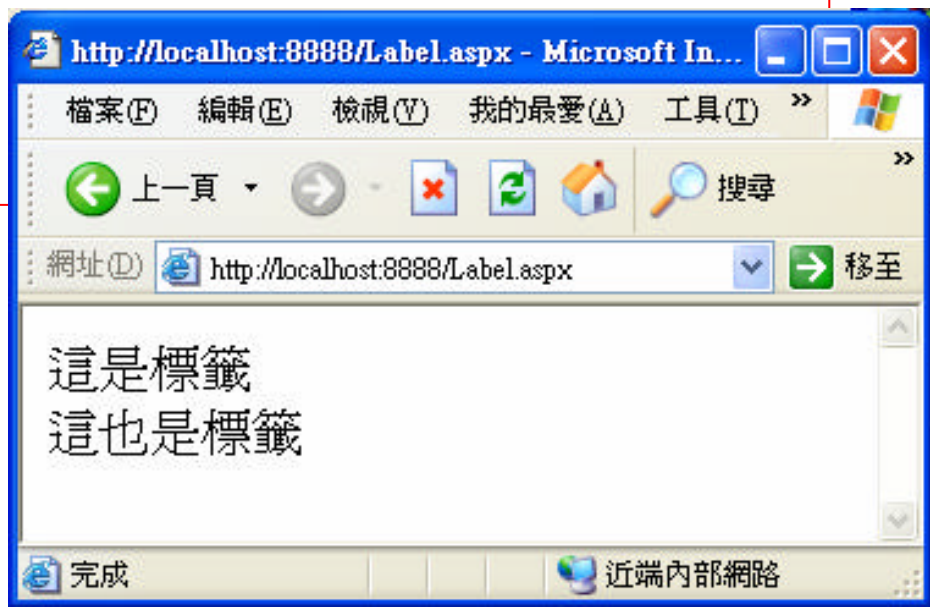
Web控制項的使用

- 要使用 Web 控制項，只要在標籤中先加上“ASP:”，並指定 Web 控制項的類別名稱即可。下表列出所有的Web控制項：

AdRotator	Button	Calendar
CheckBox	CheckBoxList	DataGrid
DataList	DropDownList	Hyperlink
Image	ImageButton	Label
ListBox	Literal	LinkButton
Panel	PlaceHolder	RadioButton
RadioButtonList	Repeater	Table
TableCell	TableRow	TextBox

Web控制項的使用

```
<html>
<Asp:Label id="Label1" text="這是標籤" runat="Server"/><br>
<Asp:Label id="Label2" runat="Server"/>
<script language="VB" runat="Server">
Private Sub Page_Load(sender As Object, e As EventArgs)
  Label2.Text="這也是標籤"
End Sub
</script>
</html>
```



Web控制項的基礎屬性

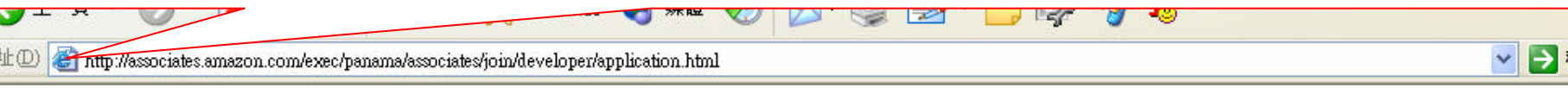
- 所謂基礎屬性就是所有的Web控制項共同都有的屬性。

AccessKey	Attributes	BackColor
BorderWidth	BorderStyle	CSSClass
CSSStyle	Enabled	Font-Bold
Font-Italic	Font-Name	Font-Names
Font-Overline	Font-Size	Font-Strikeout
Font-Underline	ForeColor	Height
TabIndex	ToolTip	Width

MyAmazone


My Amazon - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
網址(D) http://localhost:8888/wsdemo/booksearch.aspx
移至
My Amazon搜尋: SOAP
搜尋
上一頁 | 下一頁
封面 書名 作者 書價 出版日
Understanding Web Services: XML, WSDL, SOAP, and UDDI Eric Newcomer $39.99 13 May, 2002
The Soapmaker's Companion: A Comprehensive Guide With Recipes, Techniques & Know-How Susan Miller Cavitch $18.95 July, 1997
Making Natural Liquid Soaps: Herbal Shower Gels / Conditioning Shampoos / Moisturizing Hand Soaps Catherine Failor $14.95 15 April, 2000
The Natural Soap Book: Making Herbal and Vegetable-Based Soaps Susan Miller Cavitch $14.95 August, 1995
Soap: Making It, Enjoying It Ann Bramson $7.95 January, 1981
Writing Soap Notes Ginge Kettenbach $26.95 15 January, 1995
Milk-Based Soaps: Making Natural, Skin-Nourishing Soap Casey Makela $12.95 September, 1997
完成
近端內部網路

http://associates.amazon.com/exec/panama/associates/join/developer/application.html



amazon.com. amazon.co.uk

AMAZON WEB SERVICES

DOWNLOAD KIT | APPLY FOR TOKEN | HOW-TO | DISCUSSION BOARDS | ANNOUNCEMENTS | UPDATE ACCOUNT

Three Easy Steps

1) Download the Developer's Kit.

2) Apply for a Token.

3) Write your Application!

Explore Web Services

Discussion Board

Announcements

FAQ

Web Services Books

Apply for a Developer's Token

Applying for an Amazon Web Services developer's token is a snap--all we need is your e-mail address and a password. Please note: This token can be used for both Amazon.com and Amazon.co.uk Web Services. (If you already have an Amazon.com or Amazon.co.uk login, and you want to use it for your Web services account as well, click here).

Your e-mail:

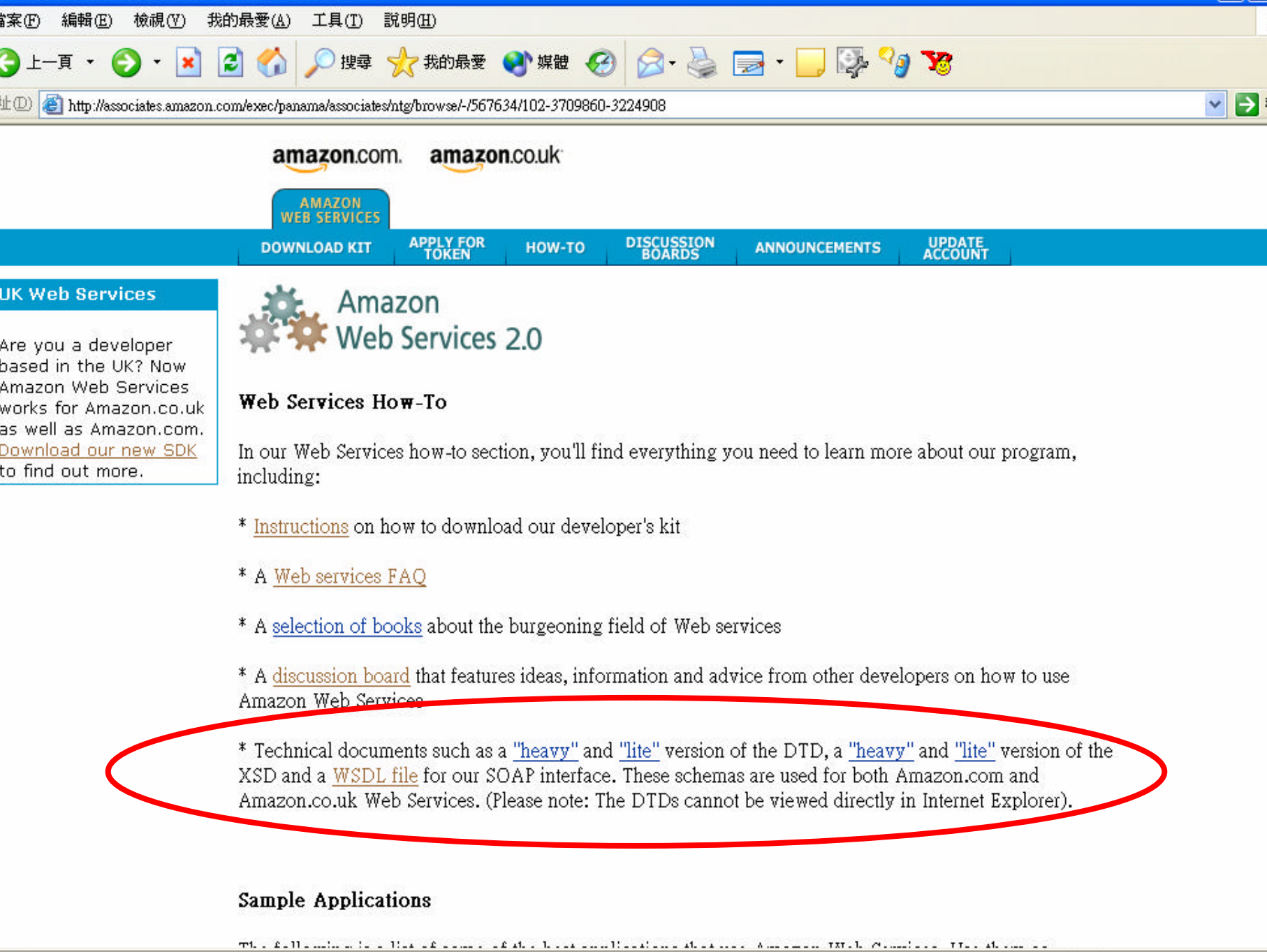
Enter password:

Re-enter password:

Please read the Amazon Web Services terms and conditions:

Amazon.com (as defined below) provides Amazon.com Web Services (as defined below) to you subject to the following conditions. This is a legal agreement between you and Amazon.com. In consideration of Amazon.com allowing you to use Amazon.com Web Services and by receiving Amazon.com Properties (as defined below) or accessing Amazon.com Web Services you agree to be bound by the terms of this agreement ("Agreement"). If you do not agree, click on your browser's back button to decline this Agreement.

Amazon.com Web Services is a platform to publish and invoke applications


檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁 搜尋 我的最愛 媒體
http://associates.amazon.com/exec/panama/associates/ntg/browse/-/567634/102-3709860-3224908
amazon.com. amazon.co.uk
AMAZON WEB SERVICES
DOWNLOAD KIT APPLY FOR TOKEN HOW-TO DISCUSSION BOARDS ANNOUNCEMENTS UPDATE ACCOUNT
UK Web Services
Are you a developer based in the UK? Now Amazon Web Services works for Amazon.co.uk as well as Amazon.com. Download our new SDK to find out more.
Amazon Web Services 2.0
Web Services How-To
In our Web Services how-to section, you'll find everything you need to learn more about our program, including:
* Instructions on how to download our developer's kit
* A Web services FAQ
* A selection of books about the burgeoning field of Web services
* A discussion board that features ideas, information and advice from other developers on how to use Amazon Web Services
* Technical documents such as a "heavy" and "lite" version of the DTD, a "heavy" and "lite" version of the XSD and a WSDL file for our SOAP interface. These schemas are used for both Amazon.com and Amazon.co.uk Web Services. (Please note: The DTDs cannot be viewed directly in Internet Explorer).
Sample Applications

編譯WSDL檔案

D:\>wsdl AmazonWebServices.wsdl [Enter]

- 在按下Enter鍵後，稍等一下，可以看到成功輸出AmazonSearchService.cs的C#類別檔案

使用Amazon搜尋服務的Web Service-建立物件

- ASP.NET程式只需將C#類別檔案視為外部程式檔案，如下所示：

```
<%@ Page language="vb" Src="AmazonSearchService.cs" %>
```

- 只需知道此類別檔案定義的類別屬性和方法，就可以在ASP.NET程式以VB.NET語法使用Amazon的Web Service，如下所示：

```
Dim objAmazon As AmazonSearchService = New AmazonSearchService()
```

- 上述程式碼使用AmazonSearchService建構子建立AmzoneSearchService物件。

KeywordRequest 類別的屬性

屬　　性	說　　明
keyword	Search pattern
devTag	Token
mode	Product type: book, dvd, classical, …
type	Lite or heave
page	Page number

mode 屬性的屬性值

- mode=baby (Baby)
- mode=books (Books)
- mode=classical (Classical Music)
- mode=dvd (DVD)
- mode=electronics (Electronics)
- mode=garden (Outdoor Living)
- mode=kitchen (Kitchen & Housewares)
- mode=magazines (Magazines)
- mode=music (Popular Music)
- mode=pc-hardware (Computers)
- mode=photo (Camera & Photo)
- mode=software (Software)
- mode=toys (Toys & Games)
- mode=universal (Tools & Hardware)
- mode=vhs (Video)
- mode=videogames (Computer & Video Games)

Details 類別的屬性

屬性	說明
Asin, ISBN or UPC	ASIN, ISBN, or UPC. ASIN (Amazon Standard Item Number) is a unique identification number for all of Amazon's products. For books, ASIN is similar to ISBN (International Standard Book Number). For CDs and cassettes, Amazon uses the UPC (Universal Product Code)
ProductName	產品名稱
Authors	作者
ListPrice	定價、OurPrice屬性是 Amazon 網站的價格
ImageURLSmall ImageURLMedium ImageURLLage	圖片的網址
URL	產品資料的網址
Publisher	出版商
ReleaseDate	出版日

SOAP Request 範例

```
<?xml version="1.0" encoding="UTF-8" ?>
<SOAP-ENV:Envelope xmlns:SOAP
    ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
    <namesp1:KeywordSearchRequest
       xmlns:namesp1="urn:PI/DevCentral/SoapService">
      <KeywordSearchRequest xsi:type="namesp1:KeywordRequest">
          <keyword xsi:type="xsd:string">[subject keyword goes
           here]</keyword>
          <page xsi:type="xsd:string">[page # goes here]</page>
          <mode xsi:type="xsd:string">[product line goes here]</mode>
          <tag xsi:type="xsd:string">webservices-20</tag>
          <type xsi:type="xsd:string">[lite or heavy]</type>
          <dev-tag xsi:type="xsd:string">[developer's token goes
           here]</dev-tag>
          <format xsi:type="xsd:string">xml</format>
          <version xsi:type="xsd:string">1.0</version>
      </KeywordSearchRequest>
    </namesp1:KeywordSearchRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```