

# ADO.NET建立網頁資料庫

資料系  
林偉川

## 資料庫系統

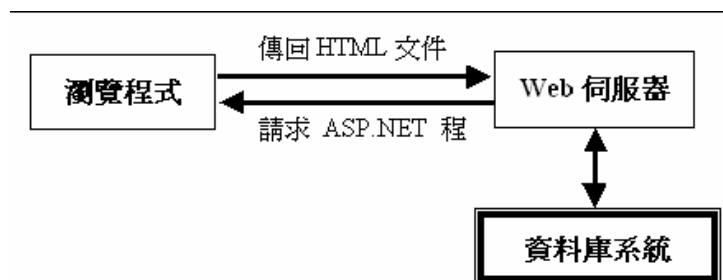
- 在電腦計算機科學的應用領域，資料庫才是公司行號真正電腦化的推手，眾多的出勤管理系統、倉庫管理系統、進銷存系統或小至錄影帶店管理系統，這些都屬於不同應用的資料庫系統。
- 資料庫系統本身就是一套應用程式，使用者在無需撰寫任何程式碼的情況下，就可以使用資料庫儲存和管理所需的資料，事實上，只有比較複雜的應用系統，程式開發者才會撰寫預設的資料庫語言設計客戶所需的應用系統。

## ASP.NET與資料庫

- Web舞台的資料庫本質並沒有轉變，仍然是用來儲存和查詢資料，只是資料庫的使用介面變成HTML文件或Web表單，網頁資料庫是一種結合前端HTML文件或Web表單的使用介面，配合後端Web伺服器 and 資料庫系統的一種應用程式架構

3

## ASP.NET與資料庫



4

## ADO.NET簡介

- ADO.NET是微軟新一代的技術，它是「ADO」(ActiveX Data Object)元件的後繼者，其主要的目的是在.NET Framework平台存取資料，簡單的說，ASP.NET技術就是使用ADO.NET進行資料庫的存取。
- ADO.NET的目的是提供一致的物件模型，用來存取和編輯資料來源的資料，至於資料來源並不限資料庫，以ASP.NET的應用來說，ADO.NET的主要功能是存取資料庫。

5

## 如何建立ASP.NET網頁資料庫

- ASP.NET建立網頁資料庫的步驟，如下所示：
  - 建立資料庫，使用Access建立所需的資料庫。
  - 使用OLE DB提供者建立資料庫連結。
  - ASP.NET程式是使用ADO.NET存取資料庫的內容，以便建立ASP.NET網頁資料庫。
  - 啟動瀏覽程式執行ASP.NET程式測試網頁資料庫。

6

## Access建立資料庫

- Access是Office家族的資料庫系統，提供個人和中小企業使用的資料庫系統，只要電腦安裝Office XP專業版本，就擁有能夠配合**ASP.NET技術**建立網頁資料庫的Access資料庫系統。

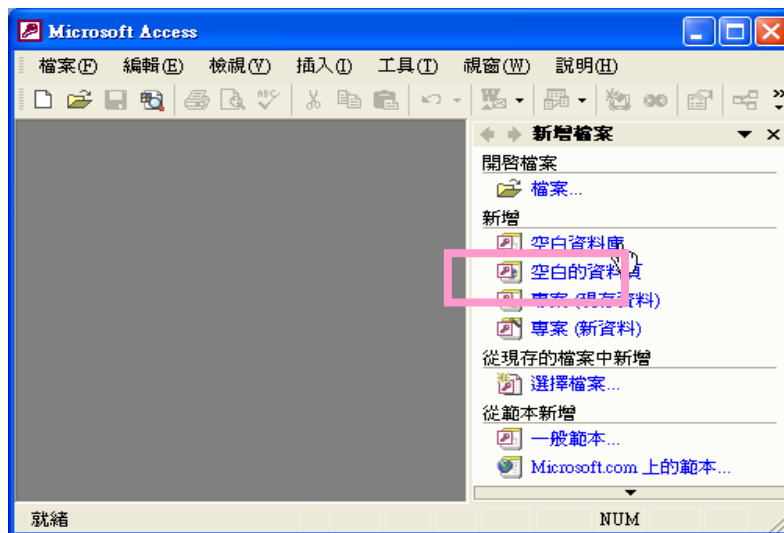
7

## Access欄位的資料型態

資料類型	說明
是/否	一個位元，用來作為 True/False 的開關欄位
數字	數字資料，使用 1、2、4 和 8 位元組儲存的數字
自動編號	欄位數值每新增記錄時自動加一，其大小為 4 位元組
日期/時間	日期時間資料
貨幣	貨幣資料，使用 8 位元組，精確到小數點左邊 15 位，右邊第 4 位
文字	字串，最大的字串長度為 255 個字元
備忘	較長的字串，最大長度為 65536 個字元
OLE 物件	二進位資料，可以用來儲存圖片資料，最多 1GB
超連結	儲存 URL 網址，最多 20456 個字元

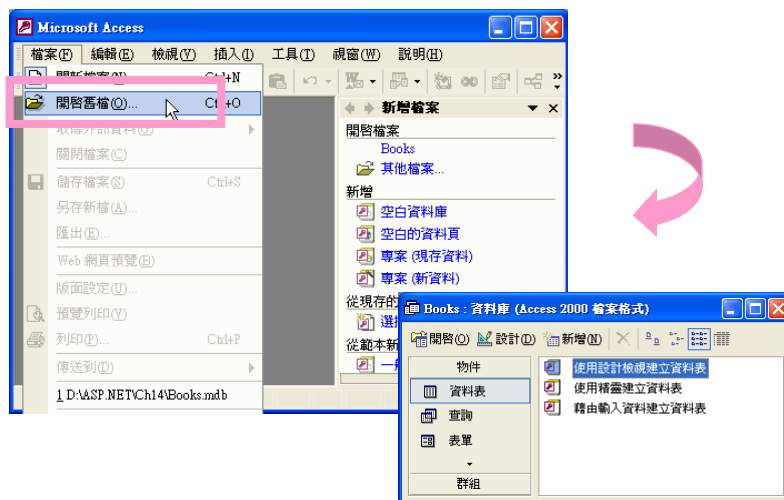
8

## 建立空白資料庫



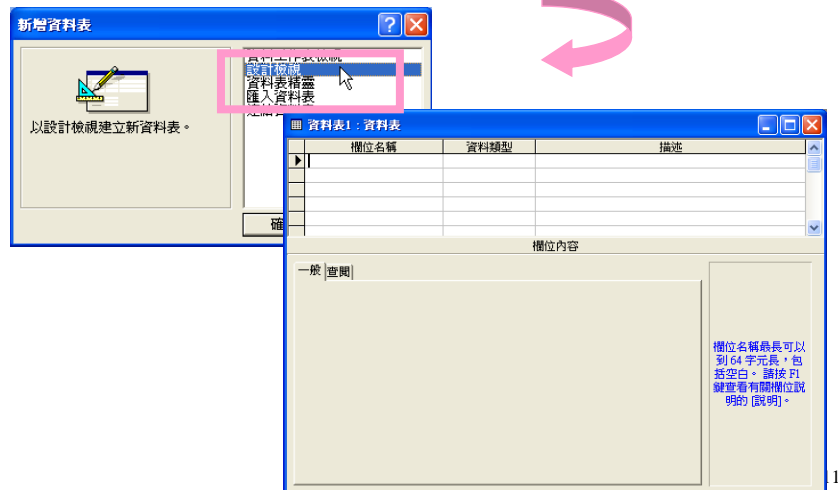
9

## 開啟存在的資料庫



0

## 新增資料表



## 新增資料表欄位

欄位名稱	資料類型	長度
BookID	文字	6
BookTitle	文字	30
BookAuthor	文字	10
BookPrice	貨幣	
BookPubDate	日期/時間	

12

## 刪除資料表欄位



13

## 新增資料表的記錄資料

The screenshot shows the 'Book: 資料表' dialog box with a table of record data. The table has columns for BookID, BookTitle, BookAuthor, BookPrice, and BookPubDate. The data is as follows:

BookID	BookTitle	BookAuthor	BookPrice	BookPubDate
P611	ASP 3.0網頁設計範例教本	陳會安	NT\$550.00	2001/4/1
S708	資料庫理論與實務	陳會安, 陳峰棋	NT\$590.00	2002/5/1
P631	Java 2程式設計範例教本	陳會安	NT\$630.00	2002/3/1
F8920	ASP.NET網頁製作徹底研究	陳會安	NT\$650.00	2002/10/1
F8534A	XML網頁製作徹底研究	陳會安	NT\$650.00	2002/7/1
*			NT\$0.00	

The status bar at the bottom shows '記錄: 5 之 5' (Records: 5 of 5).

14

## 建立資料庫連結

- 在建立好資料庫後，**ASP.NET**程式需要使用**ADO.NET**的**Connection**物件建立**ASP.NET**程式與資料庫間的連結，如此才能夠存取資料庫的記錄資料。

15

## ADO.NET的名稱空間

名稱空間	說明
System.Data	提供欄位類型、常數和 DataSet 物件的相關類別，DataSet 物件可以將資料庫直接儲存在記憶體，詳見第 16 章的說明
System.Data.OleDb	OLE DB 的 .NET 提供者，提供 OleDbCommand、OleDbConnection、OleDbDataReader 類別處理 OLE DB 資料來源的資料庫
System.Data.SqlClient	SQL 的 .NET 提供者，提供 SqlCommand、SqlConnection、SqlDataReader 類別處理微軟 Microsoft SQL Server 7.0 以上版本的資料庫

16



## 建立資料庫連結的Connection物件

- Connection物件是用來建立ASP.NET程式與存取資料來源的連結，以資料庫來說**Connection物件還負責初始資料庫**。Connection物件一共有兩種：**OLE DB**是使用**OleDbConnection物件**，**SQL Server**使用**SqlConnection物件**。
- 在ASP.NET程式需要先匯入所需的**名稱空間 Namespace**，才能使用**ADO.NET**的類別和物件，如下所示：(Access or Oracle)

```
<%@ Import Namespace="System.Data" %>
```

```
<%@ Import NameSpace="System.Data.OleDb" %>
```

17

## 建立資料庫連結的Connection物件

- 若是使用SQL server，在ASP.NET程式要先匯入所需的**名稱空間**如下所示：

```
<%@ Import Namespace="System.Data" %>
```

```
<%@ Import NameSpace="System.Data.SqlClient" %>
```

- 匯入所需的**名稱空間**，就可使用**Connection物件**建立資料庫連結，其步驟如下：

18

## 建立資料庫連結的Connection物件

- 第一步：建立Connection物件
- 在ASP.NET程式建立Connection物件，首先宣告OleDbConnection物件變數obj，如下所示：  

```
Dim obj As OleDbConnection (SqlConnection)  
obj = New OleDbConnection(str)
```
- 參數str是OLE DB提供者建立的資料來源字串。以Access資料庫系統為例，資料庫Books.mdb的str字串，如下所示：  

```
str = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &  
Server.MapPath("Books.mdb")
```

19

## 連結字串說明

- 參數str以Access為例，只要Provider及DataSource設定即可  

```
str = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" &  
Server.MapPath("Books.mdb")
```
- 參數str以Oracle為例，除要Provider及DataSource設定以外，還要給定帳號密碼  

```
str =  
"Provider=MSDAORA;DataSource=Oracle8i7;UserID=wayne;Password=123"
```
- 參數str以SQL Server為例，要利用SqlConnection物件建立連結，其字串如下：  

```
str = "server=WW;database=user; uid=wayne;pwd=1234"
```

20

## 建立資料庫連結的Connection物件

- 第二步：開啟資料庫連結
- 在建立好**Connection物件**後，接著使用**Open()**方法開啟資料庫連結，如下所示；  
**obj.Open()**
- 程式碼使用名為**obj**的**Connection物件**開啟資料庫連結。

21

## 建立資料庫連結的Connection物件

- 第三步：資料庫操作與查詢
- 在建立好資料庫連結**Connection物件**後，可以使用**State**屬性檢查目前資料庫連結狀態，如下：  
**If obj.State = ConnectionState.Open Then**  
    **Response.Write("<b>目前為開啟狀態</b><br>")**  
**Else**  
    **Response.Write("<b>目前為關閉狀態</b><br>")**  
**End If**
- 屬性**State**值為**ConnectionState.Open**表示開啟，**ConnectionState.Closed**表示關閉，在確定資料庫連結為開啟狀態時，就可以建立其它ADO.NET物件。

22

## 建立資料庫連結的Connection物件

- 第四步：關閉資料庫連結
- 最後在完成資料庫操作後，需要關閉資料庫連結，使用的是Connection物件的Close()方法，以Connection物件obj為例，其關閉的指令如下：**obj.Close()**

23

## ASP.NET建立網頁資料庫

- ADO.NET網頁資料庫可以使用**DataReader**或**DataSet**物件取得資料表的記錄資料，在本章主要是說明**Command**物件執行**SQL**指令來取得**DataReader**物件的資料表記錄。
- ASP.NET程式範例使用的**SQL**查詢指令都是同一個，如下所示：  
**SELECT \* FROM Book**
- **SQL**查詢指令**SELECT**並沒有加上任何條件，查詢結果是取資料表**Book**所有的記錄和欄位。

24

## Command與DataReader物件

- 在開啟Connection物件的資料庫連結後，我們就可以使用**Command物件**和**DataReader物件**建立**ASP.NET網頁資料庫**。

25

## Command物件

- Command物件可以針對**資料來源執行指令**，以資料庫的資料來源而言，就是執行**SQL指令**，我們可以使用**Command物件**送出SQL指令來**新增、刪除、更新和查詢**資料表的記錄。
- ASP.NET程式需要先使用**Connection物件**建立和**開啟資料庫連結**，之後才能使用Command物件執行SQL指令，Command物件也分為兩種：**OLE DB**是使用**OleDbCommand物件**，**SQL Server**是使用**SqlCommand物件**。

26

## DataReader物件

- DataReader物件可以從資料來源使用Command物件執行SQL查詢指令，以取得「唯讀、只能向前」的串流資料，ASP.NET程式可以使用類似讀取文字檔案串流的方式取得資料表記錄，因為每次只會從資料來源讀取一行資料儲存到記憶體，所以執行效率非常的高。
- DataReader物件分為兩種：OLE DB是使用OleDbDataReader物件，SQL Server是使用SqlDataReader物件。

27

## 開啟DataReader物件

- 第一步：開啟Connection物件的資料庫連結
- ASP.NET程式建立DataReader物件需要使用Connection物件開啟資料庫連結，如下：  
`obj = New OleDbConnection(str) 【New SqlConnection(str)】`  
`obj.Open()`
- 程式碼建立obj的資料連結物件，然後使用Open()方法開啟資料庫連結。

28

## 開啟DataReader物件

- 第二步：建立Command物件
- 在建立好Connection物件後，使用**SQL指令**和**Connection物件**作為參數建立Command物件，如下所示：  

```
cmd = New OleDbCommand("SELECT * FROM Book", obj)  
cmd = New SqlCommand("SELECT * FROM Book", obj)
```
- 程式碼使用**New運算子**建立Command物件，第2個參數是**開啟的資料庫連結物件**，表示是向此資料來源執行**第1個參數的SQL指令**。

29

## 開啟DataReader物件

- 第三步：執行SQL指令查詢資料表
- 接著使用Command物件的**ExecuteReader()**方法執行**SQL查詢指令**，如下所示：  

```
dr = cmd.ExecuteReader()
```
- 程式碼取得**DataReader物件**dr，讀者可以想像是**開啟資料庫的檔案串流**。

30

## 開啟DataReader物件

- 第四步：檢查狀態或讀取記錄資料
- DataReader物件是一種資料串流，我們可以使用IsClose屬性檢查DataReader串流是否開啟，如下所示：

**If dr.IsClosed = False Then**

**Response.Write("<b>DataReader物件為開啟狀態</b><br>")**

**Else**

**Response.Write("<b>DataReader物件為關閉狀態</b><br>")**

**End If**

31

## 開啟DataReader物件屬性

屬性	說明
FieldCount	取得欄位數
IsClosed	檢查 DataReader 物件串流是否關閉，True 是關閉，False 為開啓

32



## 開啟DataReader物件

- 第五步：關閉DataReader和資料庫連結
- 最後需要關閉**DataReader串流物件**和**資料庫連結**，如下所示：

`dr.Close()`

`obj.Close()`

- 程式碼使用Close()方法先關閉**DataReader**，  
，先再關閉**Connection物件**。

33

## 取得記錄的欄位名稱與值

- 當Command物件使用ExecuteReader()方法取得**DataReader物件dr**後，**DataReader物件**可以視為一種檔案串流，檔案中的**每一列是一筆記錄**，一次可以讀取**一筆記錄**，如下圖所示：

記錄指標 →

P611 ASP 3.0 網頁設計範例教本 陳會安 550
S708 資料庫理論與實務 陳會安、陳峰棋 590
P631 Java 2 程式設計範例教本 陳會安 620
.....

34

## 取得記錄的欄位名稱與值

- 在開啟DataReader物件時，記錄指標是指向第1筆記錄之前，我們需要使用**Read()方法**讀取下一筆記錄，如下所示：

`dr.Read()`

- 程式碼執行**Read()方法**，目前記錄指標移到下一筆，指向**第1筆記錄**，即P611。

35

## 取得記錄的欄位名稱與方法

方法	說明
<code>IsDBNull(index)</code>	取得參數原始順序的欄位值是否為空值 DBNull，即沒有欄位值，如果是，傳回 True，否則為 False
<code>GetName(index)</code>	取得參數原始順序的欄位名稱
<code>GetValue(index)</code>	取得參數原始順序的欄位值
<code>GetFieldType(index)</code>	取得欄位的資料類型，使用 <code>ToString()</code> 方法轉換成字串，這是.NET Framework 的資料類型

36

## 使用表格顯示資料表

- 如果想顯示資料表的記錄，我們可以使用**表格方式**，**一行代表一筆記錄**，**每一欄為一個欄位顯示資料表的記錄**，可使用HTML表格標籤來顯示資料表。
- 使用**Data Binding技術**，使用ASP.NET的Web控制項顯示資料表的記錄資料。

37

## 使用表格顯示資料表

- 取得DataReader物件如同開啟**DataReader串流物件**，我們可以使用**While迴圈讀取整個資料表的記錄**，然後**顯示記錄的欄位值**，如下所示：

```
While dr.Read()  
    Response.Write("<tr>")  
    Response.Write("<td>" & dr.Item("BookID") & "</td>")  
    Response.Write("<td>" & dr.Item("BookTitle") & "</td>")  
    Response.Write("<td>" & dr.Item("BookAuthor") & "</td>")  
    ...  
    Response.Write("</tr>")  
End While
```

38

## 使用表格顯示資料表

- 在DataReader物件提供Item屬性取得各欄位的集合物件，可以使用欄位名稱或欄位順序取得欄位值，如下所示：

```
Response.Write("<td>" & dr.Item("BookTitle") & "</td>")
```

```
Response.Write("<td>" & dr.Item(1) & "</td>")
```

- 程式碼的字串BookTitle是欄位名稱，數值1是欄位BookTitle的原始順序，取得的資料型態就是資料庫欄位的資料類型，以此例是String字串，如果是BookPrice欄位就是Decimal資料型態。

39

## ASP.NET控制項的Data Binding

- ASP.NET的Web控制項支援Data Binding技術，可以將不同資料來源的Collections物件、陣列、DataReader或DataView物件整合到ASP.NET的Web控制項。
- 並非每一個Web控制項都支援Data Binding技術，只有擁有DataSource屬性的控制項才支援Data Binding，例如：Repeater、DataList和DataGrid三個控制項。

40

## ASP.NET控制項的Data Binding

- 在ASP.NET程式的**Web控制項**使用Data Binding的步驟，如下所示：
  - **Step 1**：定義資料來源和取得資料來源的資料物件。
  - **Step 2**：指定控制項的**DataSource**屬性為此資料來源的資料物件，以資料庫來說是**DataReader**或**DataView**物件。
  - **Step 3**：執行控制項的**DataBind()**方法建立資料連結。

41

## Repeater控制項表格顯示資料表

- **Repeater**控制項是使用清單方式顯示資料，能夠讓使用者自行定義**Template**範本標籤（內含**HTML**標籤），**Repeater**控制項自動以範本標籤的項目如同迴圈一般重複編排資料來源的資料。

42

## Repeater控制項表格顯示資料表

- Repeater控制項基本語法，如下所示：

```
<asp:Repeater id="odr" runat="Server">  
  <HeaderTemplate> </HeaderTemplate>  
  <ItemTemplate> </ItemTemplate>  
  <FooterTemplate> </FooterTemplate>  
</asp:Repeater>
```
- Repeater控制項使用**Template範本標籤**（標籤內容可以使用HTML標籤）來編排資料。

43

## Template標籤

Template 範本標籤	說明
HeaderTemplate	定義清單的標題，以資料表的表格來說，就是開頭標籤<table>和記錄的標題列，如果沒有定義就不顯示
ItemTemplate	定義清單的項目，也就是重複顯示的部分，以資料表來說就是每一筆記錄，這是 Repeater 控制項的必需標籤
FooterTemplate	定義清單的註腳，以資料表的表格來說，就是結尾標籤</table>，如果沒有定義就不顯示

44

## Data Binding

- 首先需要建立Repeater控制項的Data Binding，其資料來源是Book資料表的DataReader物件，如下所示：

```
odr.DataSource =cmd.ExecuteReader()  
odr.DataBind()
```

- 程式碼的odr是Repeater控制項，在執行DataBind()方法建立Data Binding後，就可以將記錄資料填入Repeater控制項。

45

## Template標籤內容

Template 範本標籤	表格標籤
HeaderTemplate	<pre>&lt;table border=1 cellspacing=0 cellpadding=5&gt; &lt;tr bgcolor="FFCC99"&gt; &lt;td&gt;書號&lt;/td&gt;&lt;td&gt;書名&lt;/td&gt;&lt;td&gt;書價&lt;/td&gt; &lt;/tr&gt;</pre>
ItemTemplate	<pre>&lt;tr&gt; &lt;td&gt;&lt;%=# Container.DataItem("BookID")%&gt;&lt;/td&gt; &lt;td&gt;&lt;%=# Container.DataItem("BookName")%&gt;&lt;/td&gt; &lt;td&gt;&lt;%=# Container.DataItem("BookPrice")%&gt;&lt;/td&gt; &lt;/tr&gt;</pre>
FooterTemplate	<pre>&lt;/table&gt;</pre>

46

## 顯示資料表欄位值

- 在<td>標籤的每一個儲存格是使用「<%#」符號和「%>」符號標示的**Data Binding**運算式，以此例是顯示**資料表欄位的值**，如下所示：

```
<%# Container.DataItem("BookID")%>
```

- **Container**屬性可以取得資料來源控制項的**RepeaterItem**物件，然後使用**RepeaterItem**物件的屬性**DataItem**取得欄位值，參數字串是資料表的欄位名稱。

47

## Repeater控制項更多的範本標籤

- Repeater控制項提供更多的範本標籤可以讓**每一列交叉以不同格式顯示或插入分隔線**，如下：

```
<asp:Repeater id="odr" runat="Server">  
  <HeaderTemplate> </HeaderTemplate>  
  <ItemTemplate> </ItemTemplate>  
  <AlternatingItemTemplate> </AlternatingItemTemplate>  
  <SeparatorTemplate> </SeparatorTemplate>  
  <FooterTemplate> </FooterTemplate>  
</asp:Repeater>
```

- AlternatingItemTemplate標籤對應ItemTemplate標籤，可以指定不同的顯示樣式。

48



## Template範本標籤

Template 範本標籤	說明
AlternatingItemTemplate	如果想每個項目交叉使用不同的樣式，例如：資料表的記錄輪流使用不同色彩顯示，就可以定義此標籤，在奇數項目（以 0 開始）使用此範本顯示，偶數是使用 ItemTemplate 範本
SeparatorTemplate	如果在各項目間需要分隔，可以在此標籤定義，通常是使用 HTML 標籤的 或<hr>，如果沒有定義並不會顯示

49

## DataList控制項顯示資料表

- DataList控制項預設使用單欄表格顯示資料，其使用方式和Repeater控制項相似，也是使用範本標籤，如下所示：

```
<asp:DataList id="dl"
  GridLines="Both" runat="Server">
  <HeaderTemplate>圖書目錄清單</HeaderTemplate>
  <ItemTemplate>
    <%#Container.DataItem("BookTitle")%>
    (<%#Container.DataItem("BookAuthor")%>)
    - $<%#Container.DataItem("BookPrice")%>
  </ItemTemplate>
  <FooterTemplate>DataList控制項顯示資料表
  </FooterTemplate>
</asp:DataList>
```

50

## Data Binding

- DataList控制項以DataReader物件為資料來源，其Data Binding的程式碼，如下所示：

```
dl.DataSource = cmd.ExecuteReader()  
dl.DataBind()
```

51

## DataList控制項的Repeat屬性

- DataList控制項預設使用表格方式顯示，不過還是可以使用Repeat的相關屬性設定表格擁有幾欄、顯示方向和編排的版面配置，如下：

屬性	說明
RepeatColumns	設定 DataList 控制項的表格是分成幾欄顯示
RepeatDirection	設定 DataList 控制項顯示的方向是 Vertical（垂直）或 Horizontal（水平）
RepeatLayout	設定 DataList 控制項的顯示方式版面配置為 Table（表格）或 Flow（水流，即一直線）

## 標籤的樣式屬性

- DataList控制項的樣式屬性是用來設定控制項的顯示外觀，各樣式屬性對應各**Template 範本標籤**，常用的樣式屬性，如下表所示：

標籤的樣式屬性	說明
HeadStyle	設定控制項 HeadTemplate 標籤的樣式
AlternatingItemStyle	設定控制項 AlternatingItemTemplate 標籤的樣式
FooterStyle	設定控制項 FooterTemplate 標籤的樣式
ItemStyle	設定控制項 ItemTemplate 標籤的樣式
SeparatorStyle	設定控制項 SeparatorTemplate 標籤的樣式

## DataList控制項的樣式屬性

樣式名稱	說明
Font-Size	指定顯示的字型尺寸
Font-Italic	指定字體是否是斜體字，其值 True 表示是，不指定或 False 為否
Font-Bold	指定字體是否是粗體字，其值 True 表示是，不指定或 False 為否
Horizontalalign	指定文字內容水平的對齊方式，left 靠左、center 置中、right 為靠右
BackColor	指定背景色彩
ForeColor	指定文字色彩

54

## DataList控制項的樣式屬性

- DataList控制項樣式屬性的使用一共有兩種方式：一種是在開頭標籤設定，如下所示：

```
<asp:DataList id="dl" runat="Server"
  Headerstyle-Font-Size="20pt"
  Headerstyle-Horizontalalign="center"
  ...
  Footerstyle-Font-Size="12pt"
  Footerstyle-Font-Italic="True">
```

- 另一種方式是使用獨立的標籤，屬於DataList控制項標籤的子標籤，如下所示：

```
<HeaderStyle BackColor="#778899"> </HeaderStyle>
<ItemStyle BackColor="Gainsboro"> </ItemStyle>
```

55

## DataGrid控制項的基本使用

- DataGrid控制項只需建立Data Binding，並不需要設定屬性，就可以使用表格方式顯示資料表記錄，其基本語法如下所示：

```
<asp:DataGrid id="dg" HeaderStyle-BackColor="#CC99FF"
  runat="Server"/>
```

- DataGrid控制項只設定樣式屬性HeaderStyle，這些屬性和DataList控制項相同。

56

## Data Binding

- DataGrid控制項以DataReader物件為資料來源，其Data Binding的程式碼，如下所示：

```
dg.DataSource =cmd.ExecuteReader()  
dg.DataBind()
```