

新觀念的 Visual Basic.NET 教本

第 4 章 表單的程式設計

4-1 表單與程式模組

上機：用按鈕啓動程式

- 前言：在日常生活中，當我們按下電器用品的開關按鈕，電器用品裡面設計好的電路就會開始運作，完成預定的工作；在表單的程式設計中，我們也經常把設計好的程式放在按鈕裡面，而藉著按下按鈕來達到執行程式的目的。
 1. 建立一Windows應用程式專案，專案名稱定為ch0401。
 2. 在表單上佈置一個Button，並將其Text屬性設定為「計算標準體重」，如圖-1。
 3. 以滑鼠雙按「計算標準體重」鈕，接著會出現Form1.vb程式模組視窗，如圖-2。

上機：用按鈕啓動程式

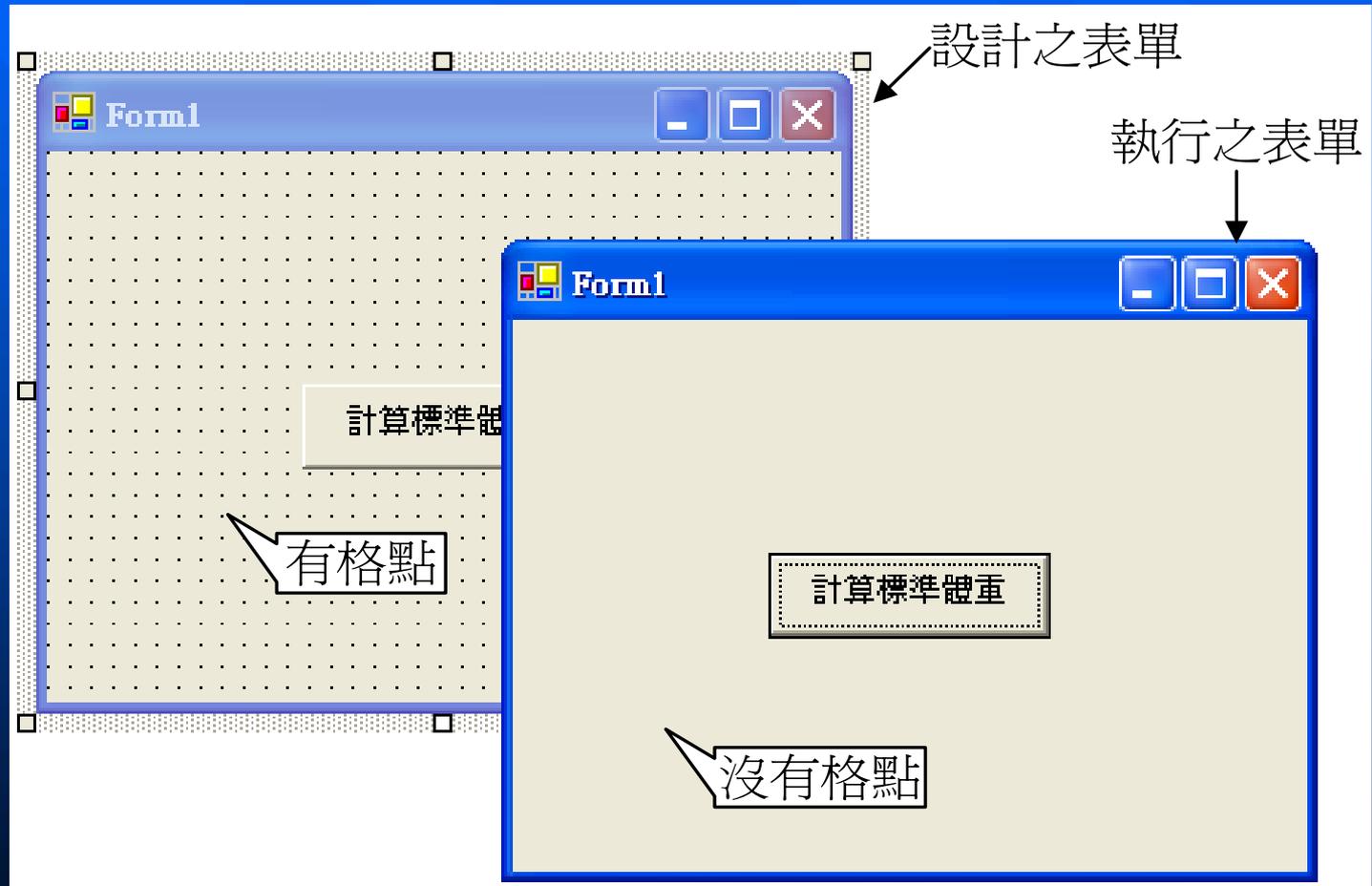
4. 在Private Sub Button1_Click(...) 與End Sub之間輸入程式，如以下粗體字部分：

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    Dim H, W
    H = 175
    W = H - 105
    MsgBox("標準體重 =" & W)
End Sub
```

上機：用按鈕啓動程式

5. 按下F5啓動程式，啓動之後還不會馬上執行到剛才輸入的程式，待出現表單視窗時，按下表單上的「計算標準體重」鈕才會執行剛才輸入的程式，結果會出現以下訊息窗，欲關閉此一訊息窗請按下「確定」鈕。接著試按幾次「計算標準體重」鈕，結果總會重新執行程式，並顯示以上訊息窗，而程式的執行過程如圖-4。
6. 結束程式：(方法一)先按下訊息窗的「確定」鈕，再按下執行之表單的關閉鈕。(方法二)選取VB.NET功能表的「偵錯>>停止偵錯」。

執行之表單 vs. 設計之表單

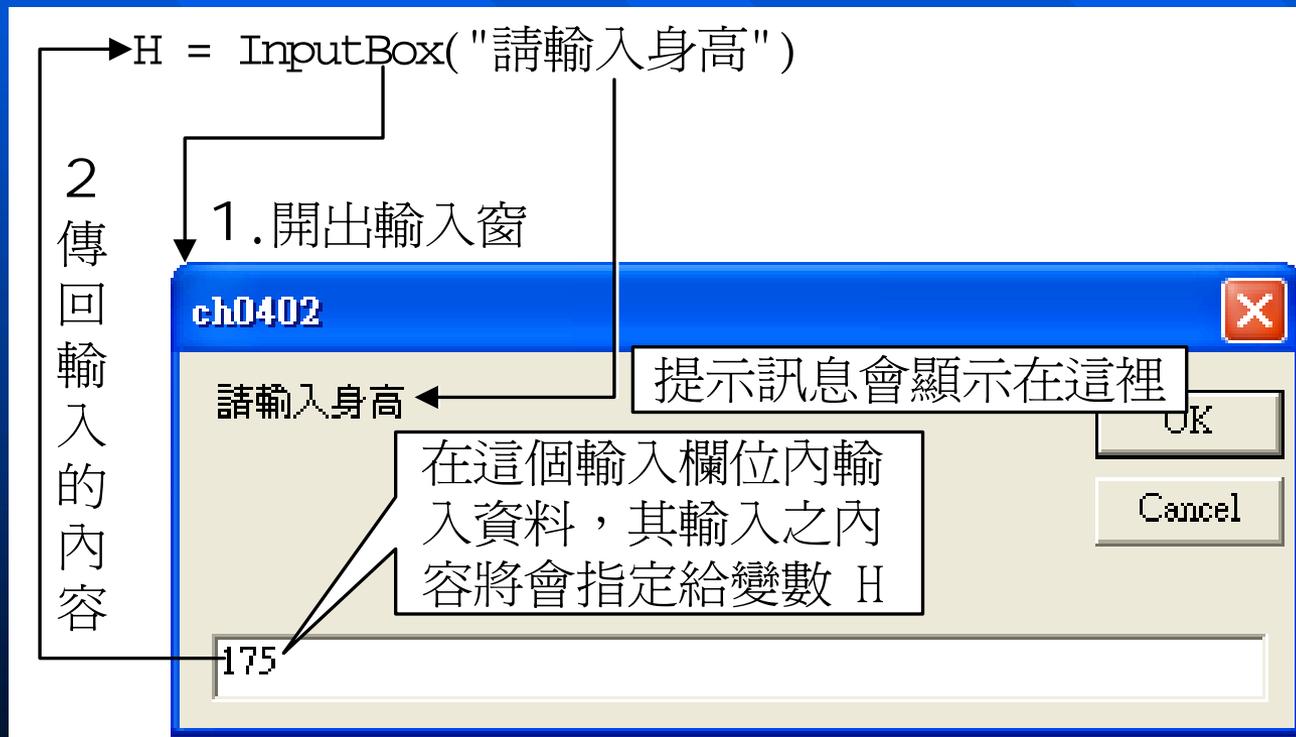


表單視窗與程式模組的切換

1. 選取Form1.vb[設計] 頁籤切換到表單視窗，或選取Form1.vb頁籤切換到程式模組。 如圖-6。
2. 選取功能表的「視窗>>Form1.vb[設計]」切換到表單視窗，或選取功能表的「視窗>>Form1.vb」切換到程式模組。
3. 在方案總管視窗中(註：方案總管位於螢幕右上方)，先選取Form1.vb，然後按下圖-7之按鈕。

可輸入資料的 InputBox

- 格式：變數 = InputBox(提示訊息)
- 實例及其運作過程：



上機：修改計算標準體重程式

1. 開啓剛才所完成的 ch0401 專案。
2. 利用方案總管開啓Form1.vb程式模組，然後將H = 175敘述改成H = InputBox("請輸入身高")，使成爲：

```
Private Sub Button1_Click(...) Handles Button1.Clic
    Dim H, W
    H = InputBox("請輸入身高")
    W = H - 105
    MsgBox("標準體重 =" & W)
End Sub
```

上機：修改計算標準體重程式

3. 執行程式，然後按下「計算標準體重」鈕，程式的執行過程如圖-9。當我們要計算不同身高的標準體重，只要重新按下「計算標準體重」鈕，並且在輸入窗中輸入新的身高即可，不必再修改程式。

4-2 設定物件屬性的敘述

上機：設定物件屬性的敘述

- 目的：在表單上佈置兩個標籤元件，其中一個使用屬性視窗設定屬性，另一個使用程式設定屬性。
 1. 建立一Windows應用程式專案，專案名稱定為ch0403。
 2. 在表單上佈置兩個Label及三個Button控制元件，如圖-10。其中Label2應設定以下屬性：

屬性	屬性值
Text	Visual Basic.NET
BorderStyle	Fixed3D
AutoSize	TRUE

上機：設定物件屬性的敘述

3. 雙按第一個按鈕(Button1)，調出程式模組，然後在Private Sub Button1_Click與End Sub之間輸入以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Click  
    Label1.Text = "Visual Basic.NET"  
End Sub
```

以上程式輸入到Label1.時，會出現一快顯功能表，列出所有Label1物件的所有屬性，接著輸入到Label1.Te時，Text屬性會變成被選取的，此時只要按下 Tab 鍵就可以完成Label1.Text的輸入，如圖-11，接著再輸入「 = "Visual Basic.NET" 」即完成輸入。

上機：設定物件屬性的敘述

4. 切換到表單視窗，然後仿照步驟3方法，在Button2按鈕之中輸入以下程式：

```
Private Sub Button2_Click(...) Handles  
Button2.Click  
    Label1.BorderStyle = BorderStyle.Fixed3D  
End Sub
```

以上程式輸入到「Label1.BorderStyle =」時，會出現快顯功能表供我們選取BorderStyle的屬性值，此時只要選取BorderStyle.Fixed3D即可完成輸入，[如圖-12](#)。

上機：設定物件屬性的敘述

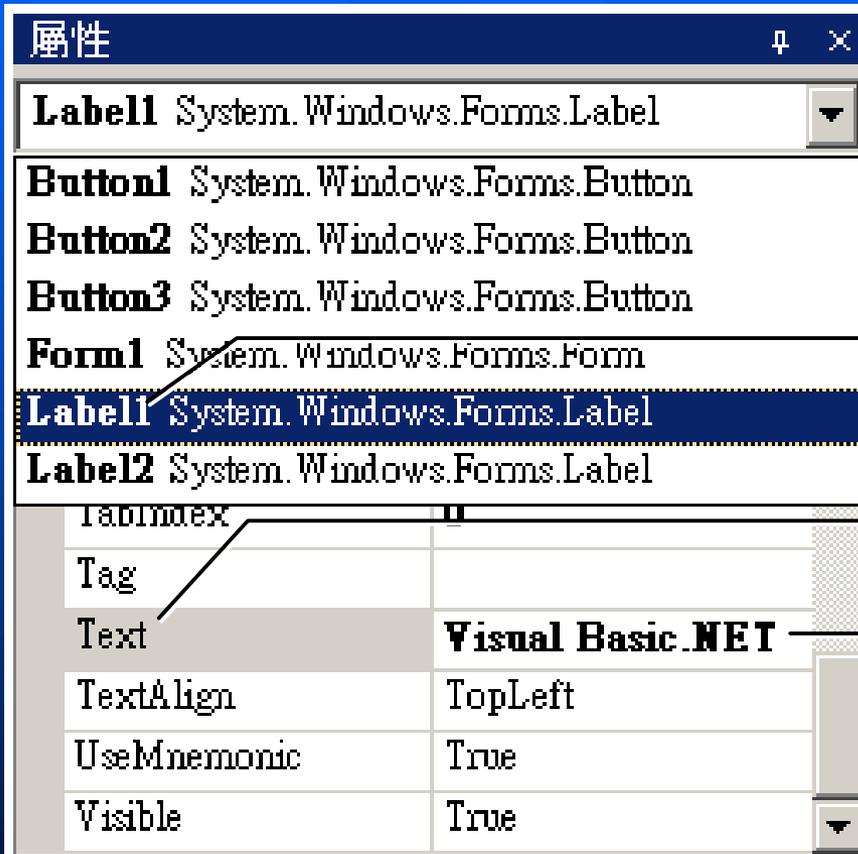
5. 切換到表單視窗，然後仿照步驟3的方法，在Button3按鈕之中輸入以下程式：

```
Private Sub Button3_Click(...) Handles Button3.Click  
    Label1.AutoSize = True
```

```
End Sub
```

6. 執行程式，然後依序按下Button1、Button2、Button3按鈕，結果變化如圖-13。最後Label1的屬性與我們利用屬性視窗所設定出來的Label2完全相同。

利用屬性視窗設定屬性的過程



1. 選取欲設定的物件

2. 選取欲設定的屬性

3. 設定屬性值

設定物件屬性的三個關鍵

屬性

Label1	System.Windows.Forms.Label
Button1	System.Windows.Forms.Button
Button2	System.Windows.Forms.Button
Button3	System.Windows.Forms.Button
Form1	System.Windows.Forms.Form
Label1	System.Windows.Forms.Label
Label2	System.Windows.Forms.Label

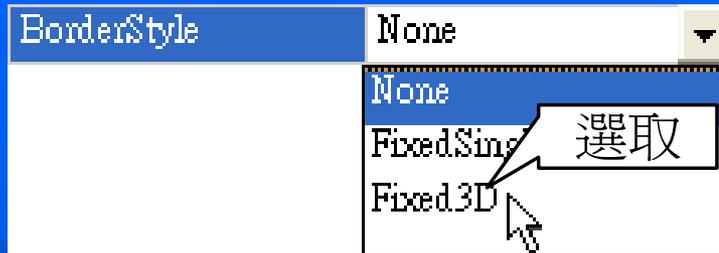
TabIndex	0
Tag	
Text	Visual Basic.NET
TextAlign	TopLeft
UseMnemonic	True
Visible	True

1.物件名 2.屬性名 3.屬性值

Label1 · Text = "Visual Basic.NET"

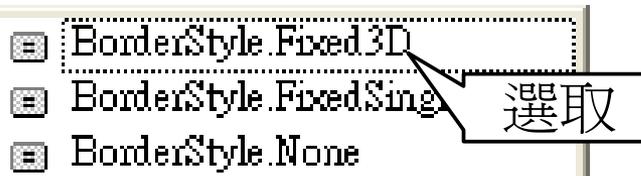
- 敘述中的「.」在功能上就跟國語的"的"一樣，例如前面這個敘述可以讀成：「將Label1的Text屬性設定成"Visual Basic.NET"」。

屬性設定 - 直接選取型



- 屬性視窗的設定方法：按下下拉按鈕，即可選取該屬性的屬性值，如上圖。
- 撰寫程式時，我們只要輸入了「物件.屬性 = 」，VB.NET也會自動顯示快顯功能表供我們選取該屬性的屬性值，如下圖：

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click  
    Label1.BorderStyle =  
End Sub
```



Location 及 Size 屬性

- Location及Size屬性可用來設定控制元件的位置(X座標及Y座標)及大小(寬度及高度)，設定的方法如下：

```
物件名.Location = New Point(x座標, y座標)
```

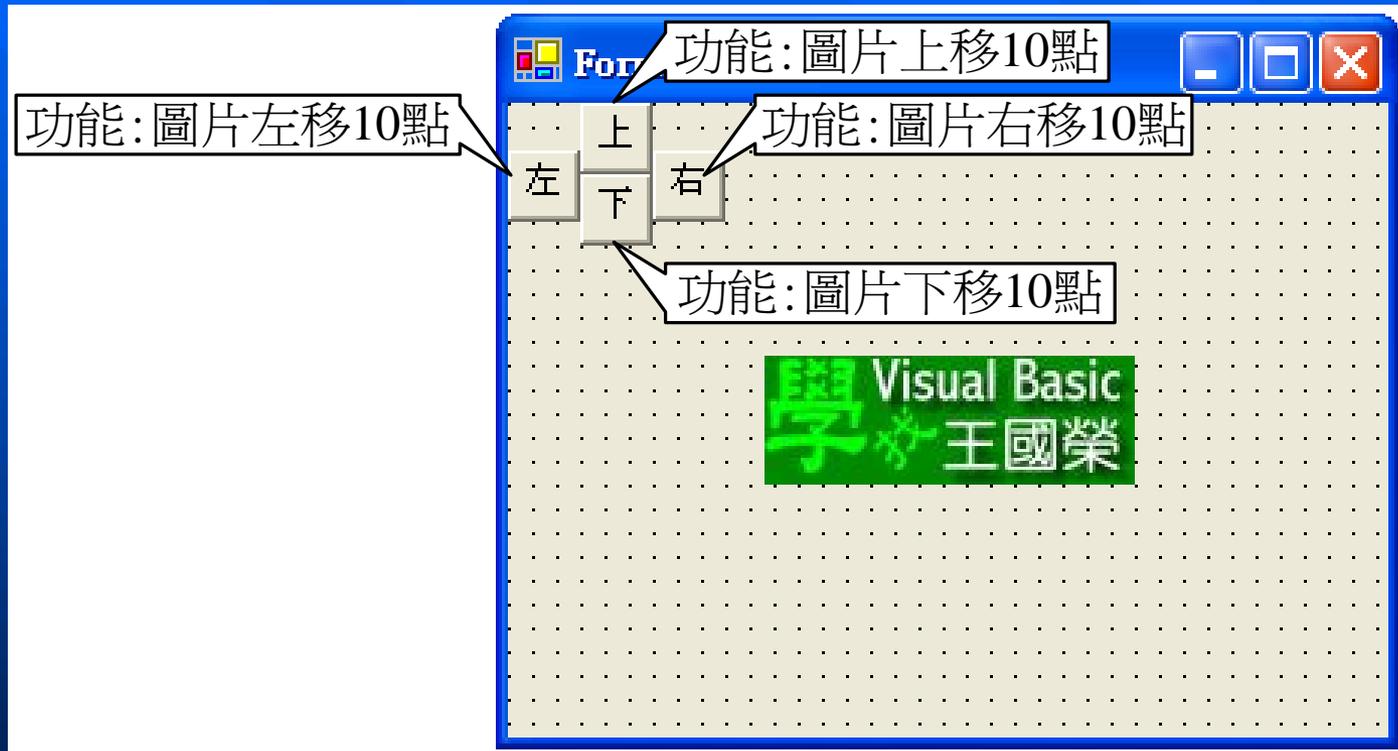
```
物件名.Size = New Size(寬度, 高度)
```

- 例如將PictureBox1的位置移到(10, 50)、大小改成(200, 80)的敘述如下：

```
PictureBox1.Location = New Point(10, 50)
```

```
PictureBox1.Size = New Size(200, 80)
```

上機：可移動圖片位置的程式



上機：可移動圖片位置的程式

1. 建立一Windows應用程式專案，專案名稱定為ch0404。
2. 在表單上佈置四個Button及一個PictureBox，如圖-19。並且設定以下屬性：

物件	屬性	屬性值
Button1	Text	左
Button2	Text	右
Button3	Text	上
Button4	Text	下
PictureBox1	Image	C:\VB教本\picture\Logo.bmp

上機：可移動圖片位置的程式

3. 在表單視窗上面，以滑鼠雙按Button1(「左」鈕)開啓Button1_Click程式架構，然後輸入以下程式：

'左移 10 點

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim X As Integer = PictureBox1.Location.X
    Dim Y As Integer = PictureBox1.Location.Y
    PictureBox1.Location = New Point(X - 10, Y)
End Sub
```

上機：可移動圖片位置的程式

4. 仿照步驟3，分別在Button2_Click～Button4_Click之中撰寫以下程式：(參閱次頁)
5. 執行程式，試著按下上、下、左、右鈕，結果圖片可以依所按之按鈕上移、下移、左移或右移10點。

上機：可移動圖片位置的程式

'右移 10 點

```
Private Sub Button2_Click(...) Handles Button2.Click
    Dim X As Integer = PictureBox1.Location.X
    Dim Y As Integer = PictureBox1.Location.Y
    PictureBox1.Location = New Point(X + 10, Y)
End Sub
```

'上移 10 點

```
Private Sub Button3_Click(...) Handles Button3.Click
    Dim X As Integer = PictureBox1.Location.X
    Dim Y As Integer = PictureBox1.Location.Y
    PictureBox1.Location = New Point(X, Y - 10)
End Sub
```

'下移 10 點

```
Private Sub Button4_Click(...) Handles Button4.Click
    Dim X As Integer = PictureBox1.Location.X
    Dim Y As Integer = PictureBox1.Location.Y
    PictureBox1.Location = New Point(X, Y + 10)
End Sub
```

Image 屬性與圖片設定

- Image屬性可用來設定控制元件所顯示的圖片，設定之方法如下：

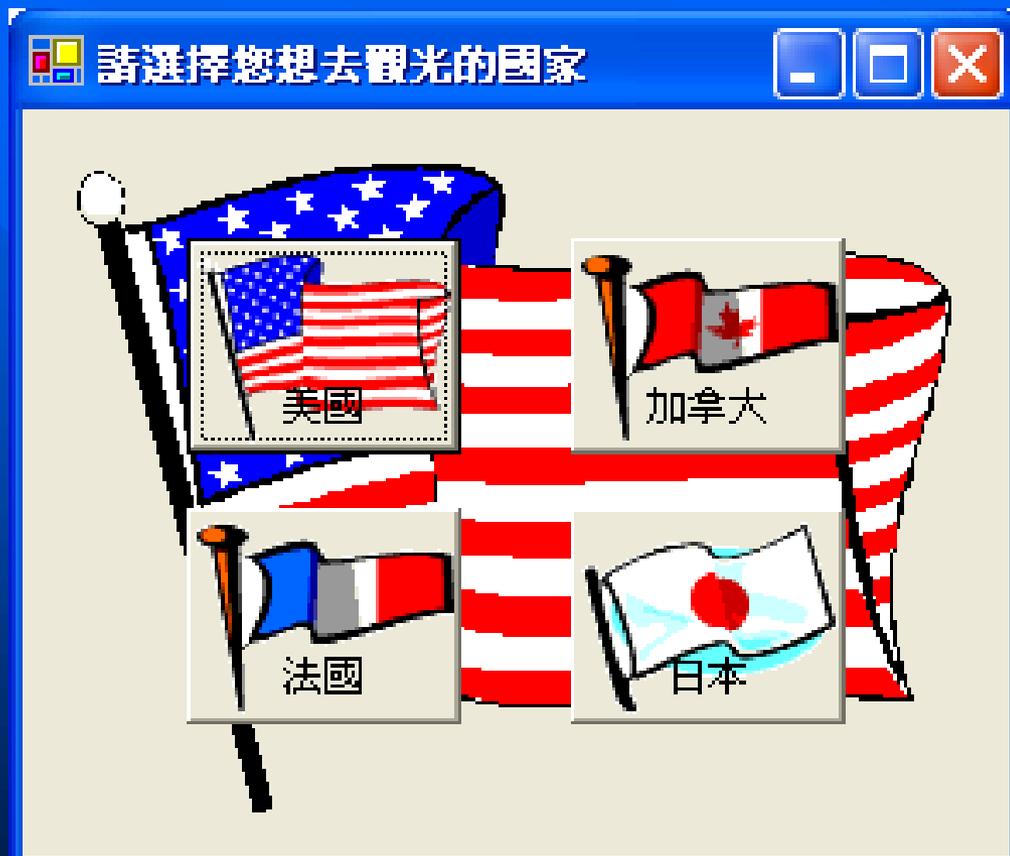
物件名.Image = New Bitmap(“圖檔”)

- 假設我們想把“C:\VB教本\picture”資料夾的“Logo.bmp”圖檔設定給 PictureBox1，那麼所撰寫的敘述如下：

PictureBox1.Image = New Bitmap("C:\VB教本\picture\Logo.bmp")

- New Bitmap() 可以讀取的圖檔格式包含 .bmp、.gif、.jpg、.jpeg、.png、.ico、.wmf、.emf 等

上機：「選擇國家」的程式



上機：「選擇國家」的程式

1. 建立一Windows應用程式專案，專案名稱定為ch0405。
2. 在表單上佈置4個Button及一個PictureBox，如圖-22，並且設定好以下屬性。

物件	屬性	屬性值
Form1	Text	請選擇您想去觀光的國家
PictureBox1	SizeMode	CenterImage (圖片置中)
Button1	Text	美國
	TextAlign	BottomCenter (中間下方)
	Picture	C:\VB教本\picture\USA.gif
Button2	Text	加拿大
	TextAlign	BottomCenter (中間下方)
	Picture	C:\VB教本\picture\Canada.gif
Button3	Text	法國
	TextAlign	BottomCenter (中間下方)
	Picture	C:\VB教本\picture\France.gif
Button4	Text	日本
	TextAlign	BottomCenter (中間下方)
	Picture	C:\VB教本\picture\Japan.gif

上機：「選擇國家」的程式

3. 分別在Button1_Click～Button4_Click之中撰寫以下程式：(參閱次頁)。
4. 啓動程式，試著按下「美國、加拿大、法國、日本」按鈕，結果PictureBox1的圖片就會跟著改變。

上機：「選擇國家」的程式

```
Private Sub Button1_Click(...) Handles Button1.Click  
    PictureBox1.Image = New Bitmap("C:\VB教本\picture\USA2.  
End Sub
```

```
Private Sub Button2_Click(...) Handles Button2.Click  
    PictureBox1.Image = New Bitmap("C:\VB教本\picture\Canad  
End Sub
```

```
Private Sub Button3_Click(...) Handles Button3.Click  
    PictureBox1.Image = New Bitmap("C:\VB教本\picture\Franc  
End Sub
```

```
Private Sub Button4_Click(...) Handles Button4.Click  
    PictureBox1.Image = New Bitmap("C:\VB教本\picture\Japan  
End Sub
```

BackColor及 ForeColor屬性

■ 設定敘述：

物件名.BackColor = 顏色設定值

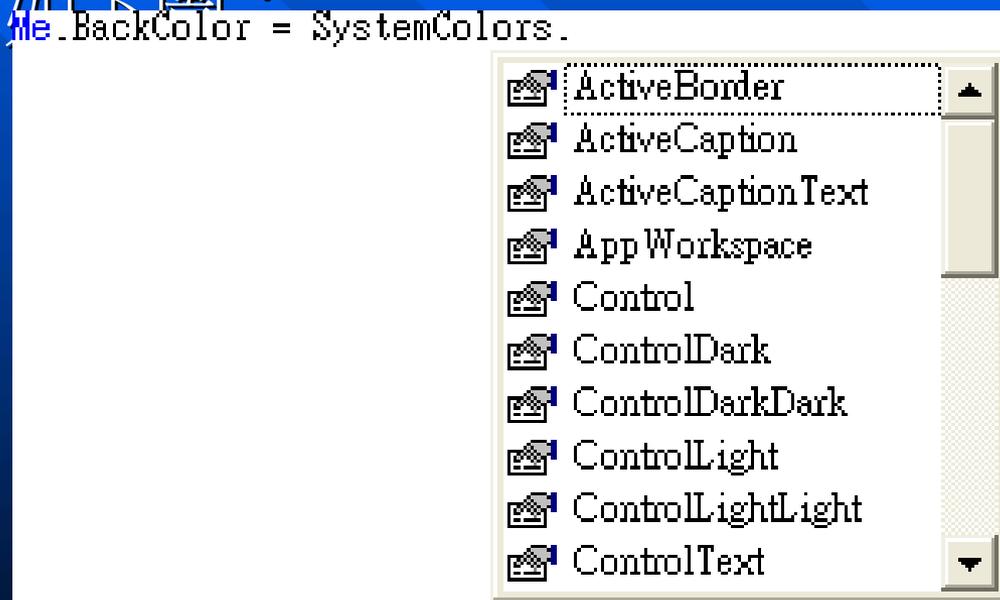
物件名.ForeColor = 顏色設定值

■ 顏色設定值的取得方法：

- SystemColors 預設的顏色。
- Color 預設的顏色。
- 利用 Color.FromArgb方法所取得的顏色設定值。

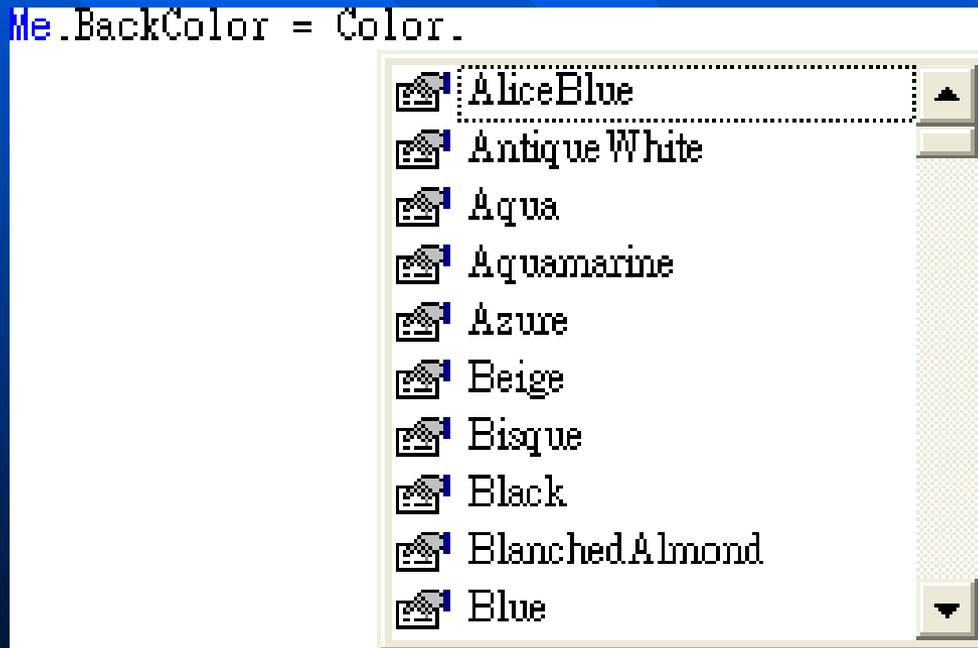
SystemColors預設的顏色

- 利用System.Colors來取得顏色設定值十分簡便，當程式輸入到「SystemColors.」時，VB.NET就會自動顯示所有的顏色設定值供我們選取，如下圖：



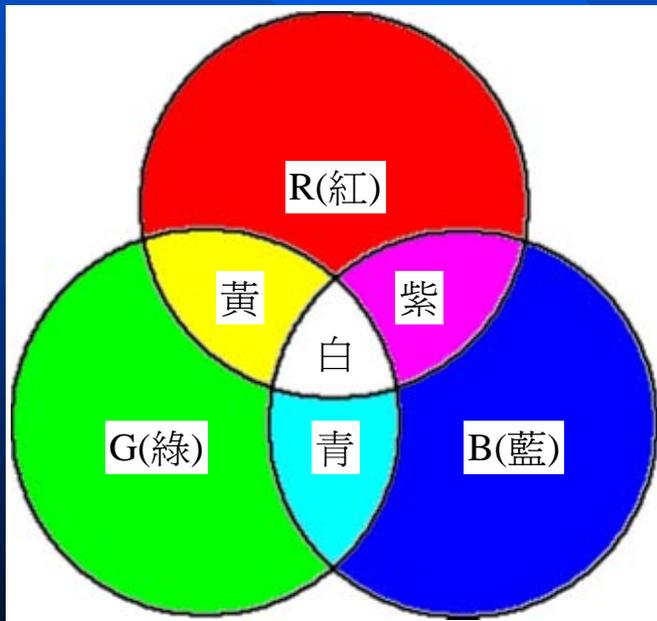
Color預設的顏色

- 利用Color來取得顏色設定值也很簡便，當程式輸入到「Color.」時，VB.NET就會自動顯示所有的顏色設定值供我們選取，如下圖：



Color.FromArgb 方法

- 格式：顏色設定值 = Color.FromArgb(R, G, B)
 - 其中 R、G、B 三個英文字母代表的是組成各種顏色變化的三原色



Color.FromArgb 方法

- 傳入Color.FromArgb方法的R、G、B三個參數，即是紅、綠、藍三原色的強度，而Color.FromArgb方法的傳回值則是組合出來的顏色設定值。參數R、G、B的強度值是0~255，超過255將會產生錯誤，使用例如下：
 - ′ 由紅色及綠色組合的顏色，即為黃色，將表單的背景設定成黃色
`Me.BackColor = Color.FromArgb(255, 255, 0)`
 - ′ 綠色強度只有一半，會變成橘色，將表單的背景設定成橘色
`Me.BackColor = Color.FromArgb(255, 128, 0)`

Font(字型)屬性

- Font 屬性包含文字的字型名稱、大小、樣式等副屬性，假設有一控制元件其Font屬性被設定成「標楷體, 12pt, 粗斜體」，則其副屬性的值將如下：

Font的副屬性	意義	屬性值
Name	字型名稱	標楷體
Size	大小	12
Bold	粗體	TRUE
Italic	斜體	TRUE
Underline	加底線	FALSE
Strikeout	加刪除線	FALSE

Font 屬性的副屬性

Font 的副屬性

Label1. <u>Font</u> .Name	等於 "標楷體"
Label1. <u>Font</u> .Size	等於 12
Label1. <u>Font</u> .Bold	等於 True
Label1. <u>Font</u> .Italic	等於 True
Label1. <u>Font</u> .Underline	等於 False
Label1. <u>Font</u> .Strikeout	等於 False

- 上述的副屬性(包含Name、Size、Style、Bold...等)都是唯讀的(ReadOnly)。

′ 想要將Label1的字型改成「新細明體」

Label1.Font.Name = "新細明體" ′ 錯誤, Name副屬性是唯讀的

′ 想要將Label1的字型大小改成16

Label1.Font.Size = 16 ′ 錯誤, Size副屬性是唯讀的

改變字型的敘述

■ 敘述格式：

物件名.Font = New Font(字型名稱, 大小, 樣式)

- 「樣式」參數可以等於FontStyle.Regular(正常字體)或是以下樣式的組合：

樣式	意義
FontStyle.Bold	粗體
FontStyle.Italic	斜體
FontStyle.Underline	加底線
FontStyle.Strikeout	加刪除線

改變字型的敘述

- 實例一：(利用 Or 運算符號組合出粗體加底線的樣式)

```
Dim Style As FontStyle  
Style = FontStyle.Bold Or  
FontStyle.Underline
```

- 實例二：(將Label1的字型設定成「細明體, 14pt, 粗體加底線」)

```
Dim Style As FontStyle  
Style = FontStyle.Bold Or  
FontStyle.Underline  
Label1.Font = New Font( "細明體", 14,  
Style )
```

上機：設定控制元件的字型

1. 建立一Windows應用程式專案，專案名稱定為ch0406。
2. 在表單上佈置一個Label及七個Button，如圖-24。
3. 分別在Button1_Click～Button7_Click之中撰寫程式(參閱次頁)。
4. 執行程式，然後參考以下步驟進行測試：

動作	結果
按下Button1(新細明體, 12pt)	Label1字型變成新細明體, 12pt
按下Button2(標楷體, 16pt)	Label1字型變成標楷體, 16pt
按下Button3(粗體)	Label1變成粗體
按下Button4(斜體)	Label1變成粗斜體
按下Button7(正常字體)	Label1變成正常字體, 粗斜體消失
按下Button5(加底線)	Label1變成加底線字體
按下Button6(加刪除線)	Label1變成加底線、加刪除線字體

上機：設定控制元件的字型

```
#01 '新細明體, 12pt
#02 Private Sub Button1_Click(...) Handles Button1.Click
#03     Label1.Font = New Font("新細明體", 12, Label1.Font.Style)
#04 End Sub
#05
#06 '標楷體, 16pt
#07 Private Sub Button2_Click(...) Handles Button2.Click
#08     Label1.Font = New Font("標楷體", 16, Label1.Font.Style)
#09 End Sub
#10
#11 '粗體
#12 Private Sub Button3_Click(...) Handles Button3.Click
#13     Label1.Font = New Font(Label1.Font.Name, Label1.Font.Size, _
#14         Label1.Font.Style Or FontStyle.Bold)
#15 End Sub
#16
#17 '斜體
#18 Private Sub Button4_Click(...) Handles Button4.Click
#19     Label1.Font = New Font(Label1.Font.Name, Label1.Font.Size, _
#20         Label1.Font.Style Or FontStyle.Italic)
#21 End Sub
#22
```

上機：設定控制元件的字型

```
#23     ' 加底線
#24     Private Sub Button5_Click(...) Handles Button5.Click
#25         Label1.Font = New Font(Label1.Font.Name, Label1.Font.Size, _
#26             Label1.Font.Style Or FontStyle.Underline)
#27     End Sub
#28
#29     ' 加刪除線
#30     Private Sub Button6_Click(...) Handles Button6.Click
#31         Label1.Font = New Font(Label1.Font.Name, Label1.Font.Size, _
#32             Label1.Font.Style Or FontStyle.Strikeout)
#33     End Sub
#34
#35     ' 正常字體
#36     Private Sub Button7_Click(...) Handles Button7.Click
#37         Label1.Font = New Font(Label1.Font.Name, Label1.Font.Size, _
#38             FontStyle.Regular)
#39     End Sub
```

4-3 輸入與輸出

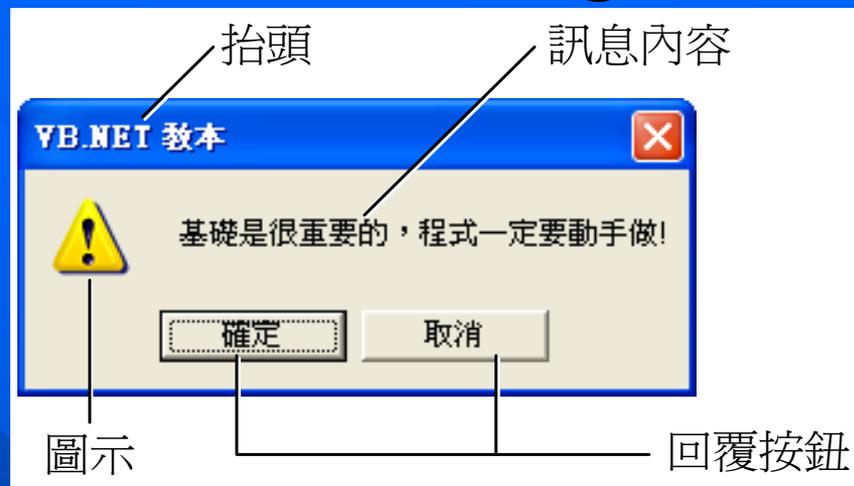
上機：使用 ReadLine 讀取資料

- 功能：建立一主控台應用程式，讓使用者輸入姓名，然後顯示「姓名, 歡迎使用 VB.NET!」。
 1. 建立一主控台應用程式專案，專案名稱定為 ch0407。
 2. 在Module1.vb程式模組中輸入程式(參閱次頁)
 3. 執行程式，首先會出現一主控台視窗並顯示「請輸入您的姓名:」，此時輸入任意姓名(例如:王國榮)，接著會顯示「王國榮, 歡迎使用 VB.NET!」及「請按 Enter 鍵結束程式.」，最後按下Enter鍵，程式即告結束，[如圖-26](#)。

上機：使用 ReadLine 讀取資料

```
Sub Main()  
    Dim 姓名  
    Console.Write("請輸入您的姓名: ")  
    姓名 = Console.ReadLine()  
    Console.Write(姓名)  
    Console.WriteLine(", 歡迎使用 VB.NET!")  
    Console.Write("請按 Enter 鍵結束程式.")  
    Console.ReadLine()  
End Sub
```

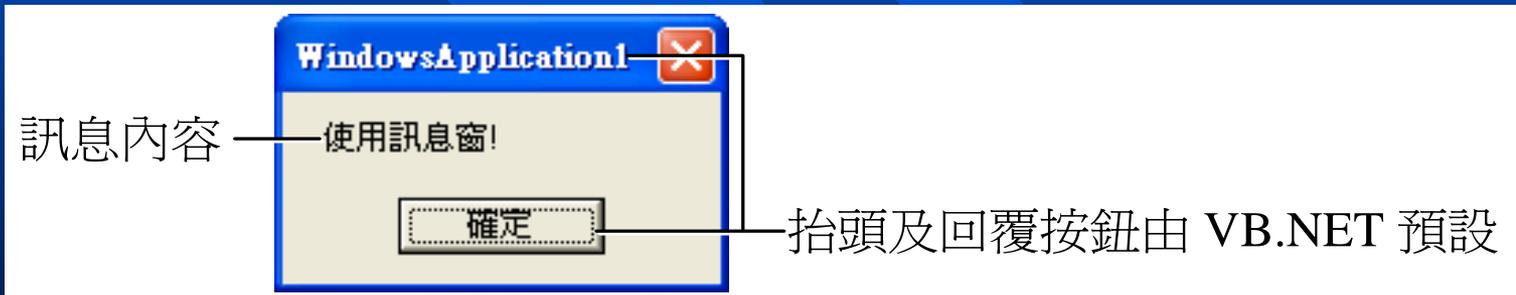
訊息窗(MsgBox)



- 抬 頭：通常用來顯示程式名稱，也就是告訴使用者這個訊息窗是哪一個程式顯示出來的。
- 訊息內容：程式想告訴使用者的訊息。
- 圖 示：提醒使用者注意的示意圖。
- 回覆按鈕：使用者回覆這個訊息的按鈕。

MsgBox, 最簡單的格式

- 格式：
 - MsgBox(訊息內容)
- 實例：
 - MsgBox(“使用訊息窗!”)



MsgBox, 設定回覆按鈕

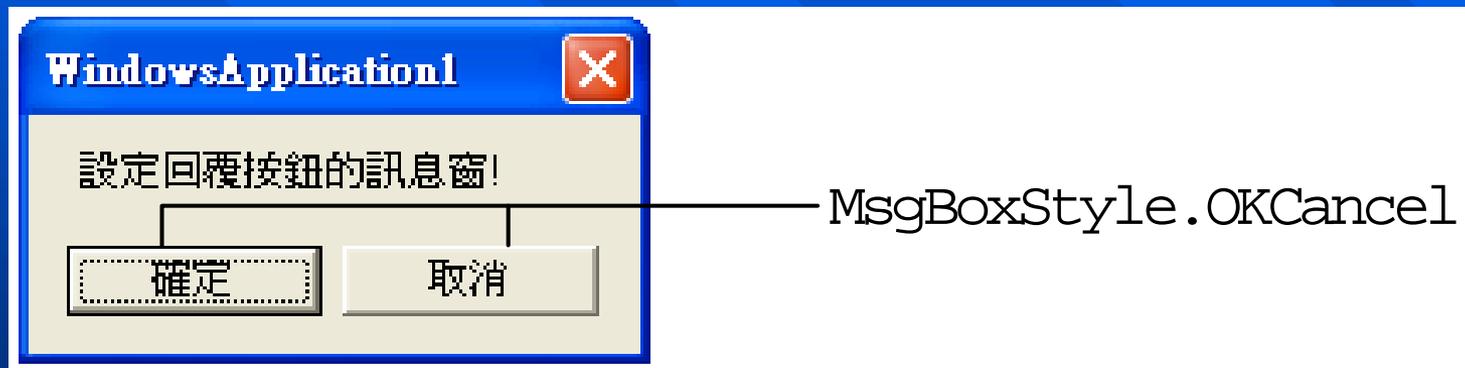
- 格式：
 - MsgBox(訊息內容, 回覆按鈕設定值)
- 回覆按鈕設定值及回覆按鈕的對照如下：

回覆按鈕設定值	回覆按鈕
MsgBoxStyle.OKOnly	<input type="button" value="確定"/>
MsgBoxStyle.OKCancel	<input type="button" value="確定"/> <input type="button" value="取消"/>
MsgBoxStyle.AbortRetryIgnore	<input type="button" value="中止(A)"/> <input type="button" value="重試(R)"/> <input type="button" value="略過(I)"/>
MsgBoxStyle.YesNoCancel	<input type="button" value="是(Y)"/> <input type="button" value="否(N)"/> <input type="button" value="取消"/>
MsgBoxStyle.YesNo	<input type="button" value="是(Y)"/> <input type="button" value="否(N)"/>
MsgBoxStyle.RetryCancel	<input type="button" value="重試(R)"/> <input type="button" value="取消"/>

MsgBox, 設定回覆按鈕

■ 實例：

- `MsgBox("設定回覆按鈕的訊息窗!", MsgBoxStyle.OKCancel)`



MsgBox, 設定圖示

- 格式：
 - MsgBox(訊息內容, 圖示設定值)
- 圖示設定值及圖示的對照如下：

圖示設定值	圖示
MsgBoxStyle.Critical	
MsgBoxStyle.Question	
MsgBoxStyle.Exclamation	
MsgBoxStyle.Information	

MsgBox, 設定圖示

■ 實例：

```
MsgBox("設定圖示的訊息窗!", MsgBoxStyle.Information)
```



MsgBoxStyle.Information

MsgBox, 同時設定回覆按鈕及圖示

■ 格式：

– `MsgBox(訊息內容, 回覆按鈕設定值+圖示設定值)`

■ 實例：

```
MsgBox("喜歡訊息窗嗎?", MsgBoxStyle.YesNo + MsgBoxStyle.Question)
```



MsgBoxStyle.Question

MsgBoxStyle.YesNo

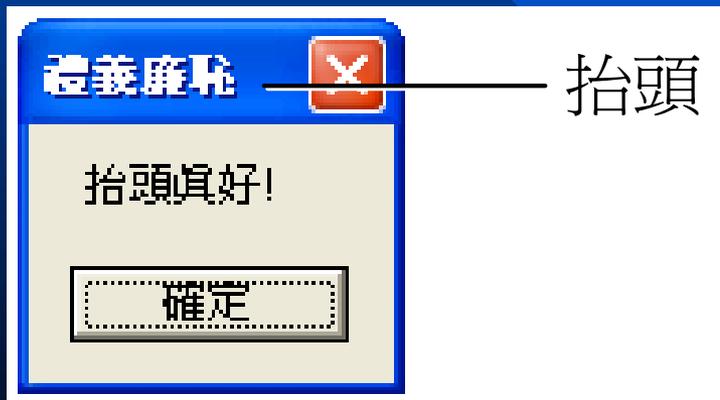
MsgBox, 設定抬頭

■ 格式：

- MsgBox(訊息內容, 回覆按鈕設定值+圖示設定值, 抬頭)
或
MsgBox(訊息內容,, 抬頭)

■ 實例：

- MsgBox("抬頭真好!", , "禮義廉恥")



MsgBox, 通用格式

■ 格式與實例：

```
MsgBox( 訊息內容 [, 回覆按鈕設定值 + 圖示設定值] [, 抬頭] )
```

↑
參數 1

↑
參數 2

↑
參數 3

```
MsgBox( "訊息內容" ) ' 省略參數二、三
```

```
MsgBox( "訊息內容", MsgBoxStyle.OKCancel ) ' 省略參數三
```

```
MsgBox( "訊息內容", , "這是抬頭" ) ' 省略參數二
```

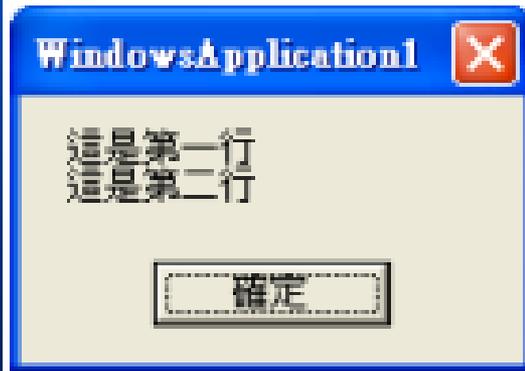
↑ 如果是中間的參數被省略，
要多加一個分隔符號『,』

MsgBox, 顯示多行訊息

- 有時候我們想顯示多行的訊息內容，則可在行與行之間插入vbCrLf的跳行字元，例如：

```
MsgBox("這是第一行" + vbCrLf + "這是第二行")
```

跳行字元



MsgBox 與傳回值

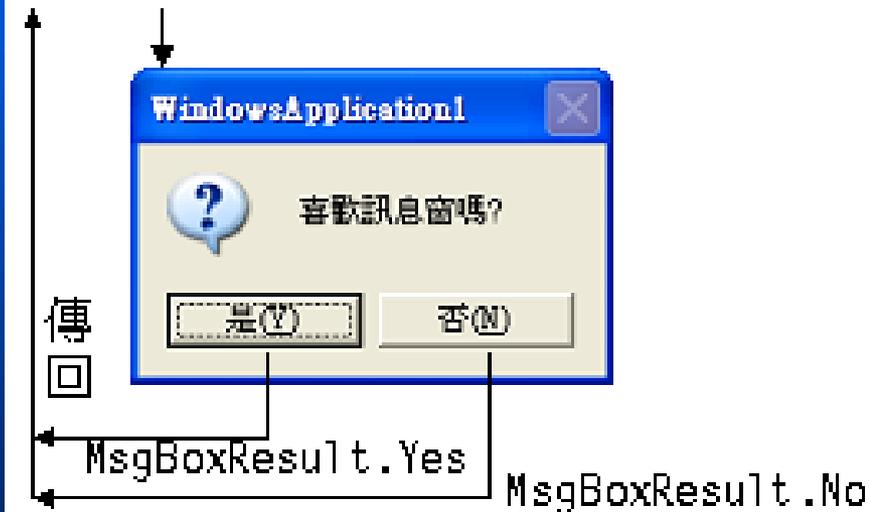
- 格式：
 - `X = MsgBox(訊息內容, 回覆按鈕設定值+圖示設定值, 抬頭)`
- 傳回值X是由使用者按下哪一個回覆按鈕來決定，以下是按鈕與傳回值的對照表：

按鈕	傳回值
	MsgBoxResult.OK
	MsgBoxResult.Cancel
	MsgBoxResult.Abort
	MsgBoxResult.Retry
	MsgBoxResult.Ignore
	MsgBoxResult.Yes
	MsgBoxResult.No

MsgBox 與傳回值

■ 實例：

```
X = MsgBox("喜歡訊息窗嗎?", MsgBoxStyle.YesNo+MsgBoxStyle.Question)
```



上機：完成以下訊息窗



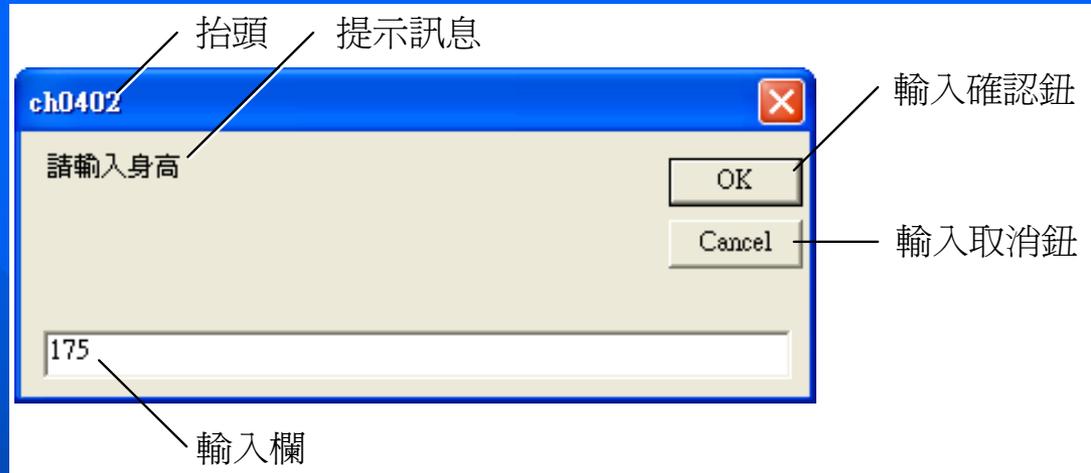
上機：完成以下訊息窗

1. 建立一主控台應用程式專案，專案名稱定為 ch0408。
2. 在Module1.vb程式模組視窗中輸入以下程

式：

```
#01 Sub Main()  
#02     Dim X  
#03     X = MsgBox( "星期六一起看場電影好嗎？", _  
#04                 MsgBoxStyle.YesNo + MsgBoxStyle.Question, _  
#05                 "星期六一起看場電影好嗎？")  
#06     If X = MsgBoxResult.Yes Then  
#07         MsgBox( "太好了！" )  
#08     Else  
#09         MsgBox( "啊！..." )  
#10     End If  
#11 End Sub
```

輸入窗(InputBox)



- 抬 頭：作用與MsgBox的抬頭相同。
- 提示訊息：提示使用者該輸入什麼資料。
- 輸 入 欄：輸入資料的欄位。
- 輸入確認鈕：使用者在輸入欄輸入資料後，若按下確認鈕即可將輸入欄裡面的資料傳回來。
- 輸入取消鈕：如果使用者按下取消鈕，則會傳回空字串(等於 "")。

InputBox 的使用

■ 格式：

– $X = \text{InputBox}(\text{提示訊息} [, \text{抬頭}] [, \text{輸入欄預設值}])$

■ 呼叫實例：

$X = \text{InputBox}(\text{"請輸入身高"})$ ' 省略後面兩個參數

$X = \text{InputBox}(\text{"請輸入身高"}, , \text{"175"})$ ' 省略抬頭

$X = \text{InputBox}(\text{"請輸入身高"}, \text{"計算標準體重"})$ ' 省略輸入欄預設值

■ 呼叫敘述與 InputBox 各組成的關係，[參閱圖-31](#)。

InputBox 的注意事項

- InputBox的傳回值是字串，當我們利用它來輸入數值資料時。

- 問題程式：

```
Dim A, B, C
A = InputBox("請輸入 A 值")
B = InputBox("請輸入 B 值")
C = A + B
```

- 修正後的程式：

```
Dim A, B, C
A = InputBox("請輸入 A 值")
B = InputBox("請輸入 B 值")
A = Val(A)           ' 將A轉換成數值資料，再指定給A
B = Val(B)           ' 將B轉換成數值資料，再指定給B
C = A + B
```

上機：計算平均分數

1. 建立一Windows應用程式專案，專案名稱定為ch0409。
2. 在表單上佈置一Button，並且將這個Button的Text屬性設定成“計算平均分數”。
3. 在Button1_Click事件程序中撰寫以下程式(參閱次頁)
4. 最後執行程式，在按下“計算平均分數”鈕之後，依序在InputBox之中輸入 90、80、76，結果應得到 82。

上機：計算平均分數

```
Private Sub Button1_Click(...) Handles Button2.  
    Dim A, B, C  
    A = Val(TextBox("請輸入[數學]成績"))  
    B = Val(TextBox("請輸入[英文]成績"))  
    C = Val(TextBox("請輸入[國文]成績"))  
    MsgBox( (A + B + C) / 3, , "平均分數")  
End Sub
```

上機：Label 與資料輸出

- 功能：將計算平均分數的程式，改用Label顯示平均分數。
 1. 開啓剛才完成的ch0409專案。
 2. 在表單上多佈置二個Label，[如圖-33](#)。並設定以下屬性：

控制元件	屬性	屬性值
Label1	Text	平均分數:
Label2	Text	(清成空白)

上機：Label 與資料輸出

3. 將Button1_Click之中的程式修改成：

```
Private Sub Button1_Click(...) Handles  
Button2.Click
```

```
    Dim A, B, C
```

```
    A = Val(TextBox("請輸入[數學]成績"))
```

```
    B = Val(TextBox("請輸入[英文]成績"))
```

```
    C = Val(TextBox("請輸入[國文]成績"))
```

```
    MsgBox("平均=" & (A + B + C) / 3, , "計算平  
均分數") '刪除
```

```
    Label2.Text = (A + B + C) / 3
```

```
End Sub
```

4. 最後執行程式，在按下 "計算平均分數" 鈕之後，依序在InputBox之中輸入90、80、76，結果平均分數會顯示在表單上。

上機：TextBox 與資料輸入

- 功能：將計算平均分數的程式，改用TextBox輸入成績，用Label顯示平均分數。
 1. 建立一Windows應用程式專案，專案名稱定為ch0410。
 2. 在表單上佈置五個Label、三個TextBox及一個Button，如圖-36。並且設定以下屬性：

控制元件	屬性	屬性值	控制元件	屬性	屬性值
Label1	Text	數學：	TextBox1	Text	(清成空白)
Label2	Text	英文：	TextBox2	Text	(清成空白)
Label3	Text	國文：	TextBox3	Text	(清成空白)
Label4	Text	平均：	Button1	Text	計算平均分數
Label5	Text	(清成空白)			

上機：TextBox 與資料輸入

3. 在Button1_Click之中輸入以下程式：

```
Private Sub Button1_Click(...) Handles Button1.Click
    Dim A, B, C
    A = Val(TextBox1.Text)
    B = Val(TextBox2.Text)
    C = Val(TextBox3.Text)
    Label5.Text = (A + B + C) / 3
End Sub
```

4. 執行程式，然後分別在TextBox1~TextBox3之中輸入90、80、76，然後按下「計算平均分數」鈕，結果如圖-37。