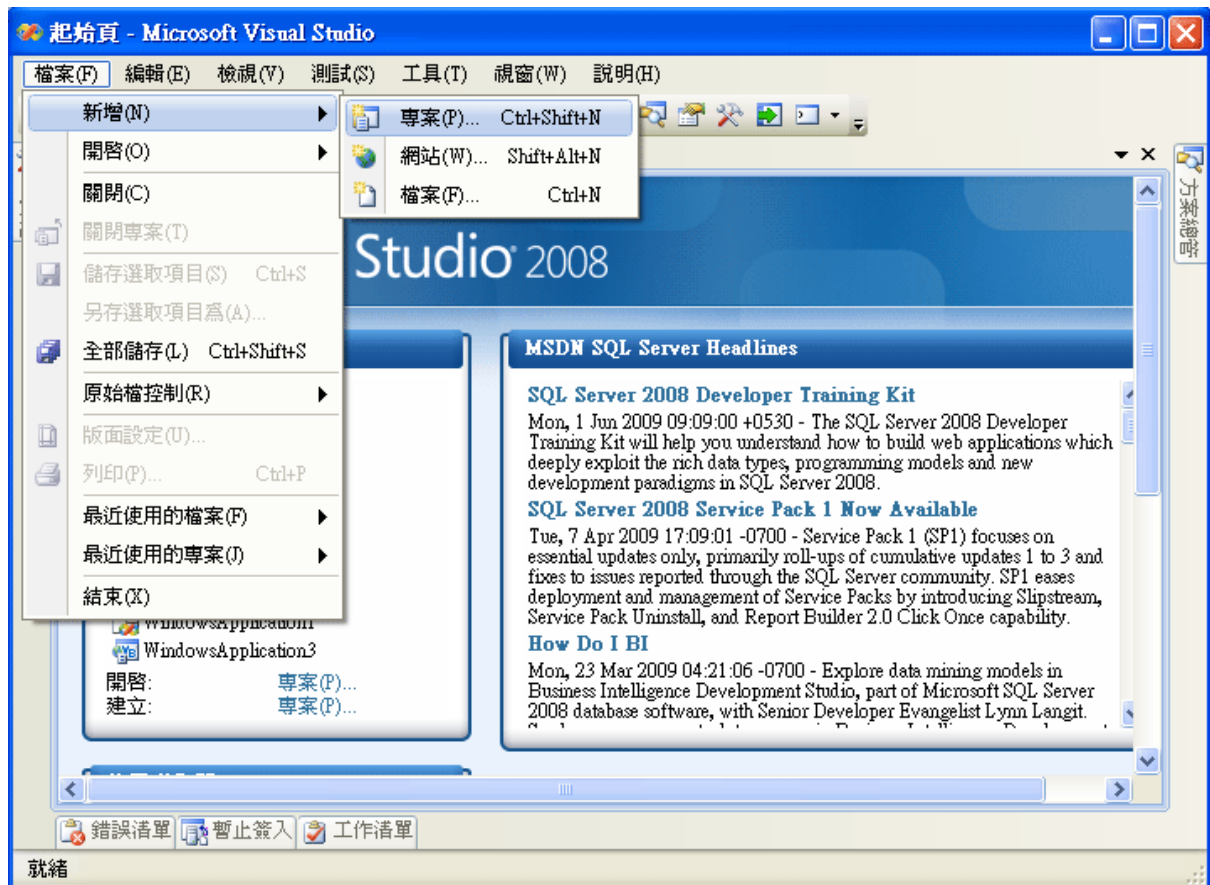


第 1 章數學計算程式

簡介：

本章是我們學習 VB 的第一章，在此使用的是標準版的 Visual Studio 2008 軟體，可以支援微軟公司的多種程式語言寫作，應該所有大專院校都有購置。如果你想在家練習又不想付費購買，可以到微軟公司的官方網站「合法」免費的下載 Express 版本，差別只是 Express 版本將各種語言分開為個別軟體，要學 VB 當然就應該選擇下載 Visual Basic 2008 Express 了！如果你下載更新的版本應該也無妨，但是最好能配合學校版本以免上下課移動程式時發生版本衝突。現在的電腦甚麼事都會做，但是它的英文名稱是 Computer，本意就是「計算機」，所以我們就從使用 VB 程式作簡單的數學計算開始這趟學習之旅吧！

1-1 認識 Visual Studio 2008



上面是我們打開 Visual Studio 2008 軟體的畫面，隨著應用程式的內容功能越來越豐富，「寫程式」的意義已經不只是『寫』程式碼而已了！還包括許多圖形化物件介面的設計，以及如填表格一般的屬性設定等等，所以乍看之下會覺得設計軟體很複雜。但是這些操作環境其實是為了減少設計者的負擔，每個圖形化物件或者屬性設定的背後都隱藏了非常龐大的原始程式碼，程式設計軟體的目的就是讓設計者可以更簡單的使用既

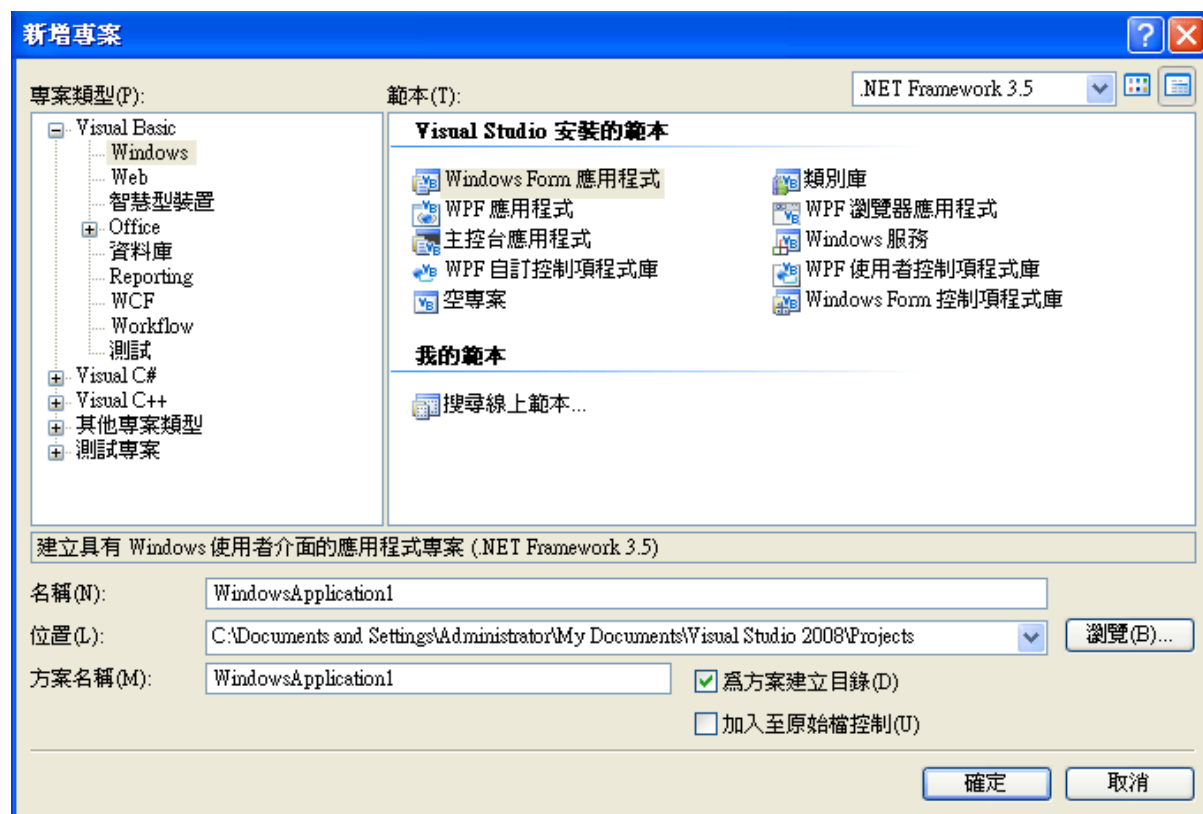
有的資源，迅速有效的組織出新的應用程式。

所以開始學習程式設計的第一個心理建設就是不要害怕複雜的程式設計環境！反而要相信這一切都是為了「幫助」你能夠又快又好的寫出任何程式。不必要求自己在第一天就熟悉所有的操作環境與功能，而是讓「寫程式的目的」來主導！依據需求一步一步地找到你需要的協助。要相信：「只要你想得到的東西，軟體都有支援！」你只需要去找到它。找不到？就問老師或軟體提供的線上說明，真正的學習障礙其實只有一個！就是你自己不動啦！

1-2 建立專案

要寫程式的第一步是建立一個「專案」，聽起來很龐大，事實上也是如此。一個現代化的「程式」不只是「一個」而是「一群」檔案。不過這些東西都是用來幫你以最少的操作來完成最多樣化的功能，通常你不必擔心如何處理它們，它們比較像一群資深的助理，可以幫啥都不懂的新老闆(就是你啦!)自動完成絕大多數例行的工作。

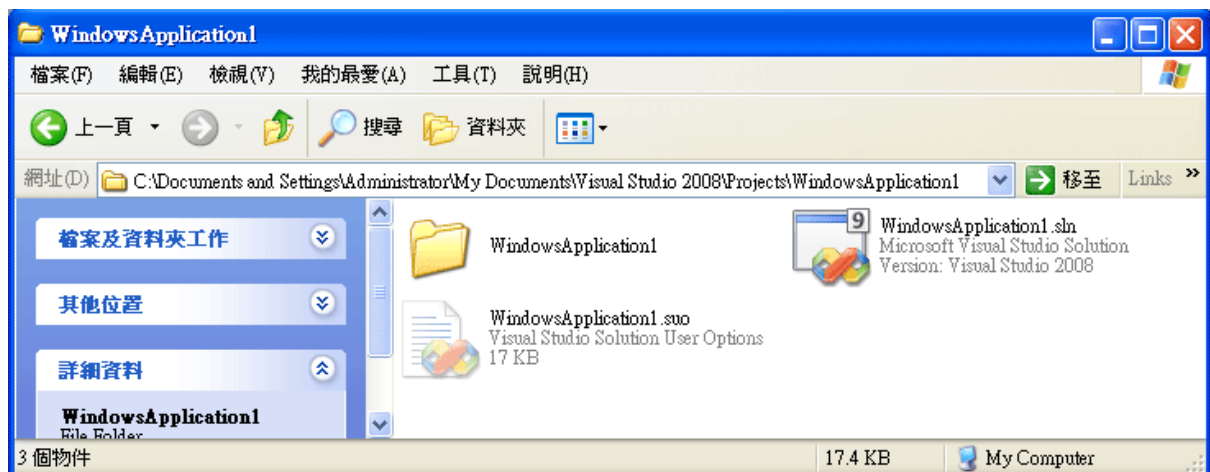
如上節的軟體開啟畫面，你可以在功能表的「新增」項目下選擇「專案」，或者在主頁面的「起始頁」選擇「建立專案」。之後會出現下面的視窗：



接下來是要選擇專案的種類，最先得選擇使用的語言，通常會預設為 Visual Basic，如果不是！就必須選 Visual Basic 或在「其他語言」的選項中找到 VB 語言，再到子項目選擇「Windows」，就是視窗程式的意思。接下來請看視窗下方的幾個可以編輯的文字框：「名稱」是你準備給程式專案取的名稱；「方案名稱」通常與程式名稱相同即可；

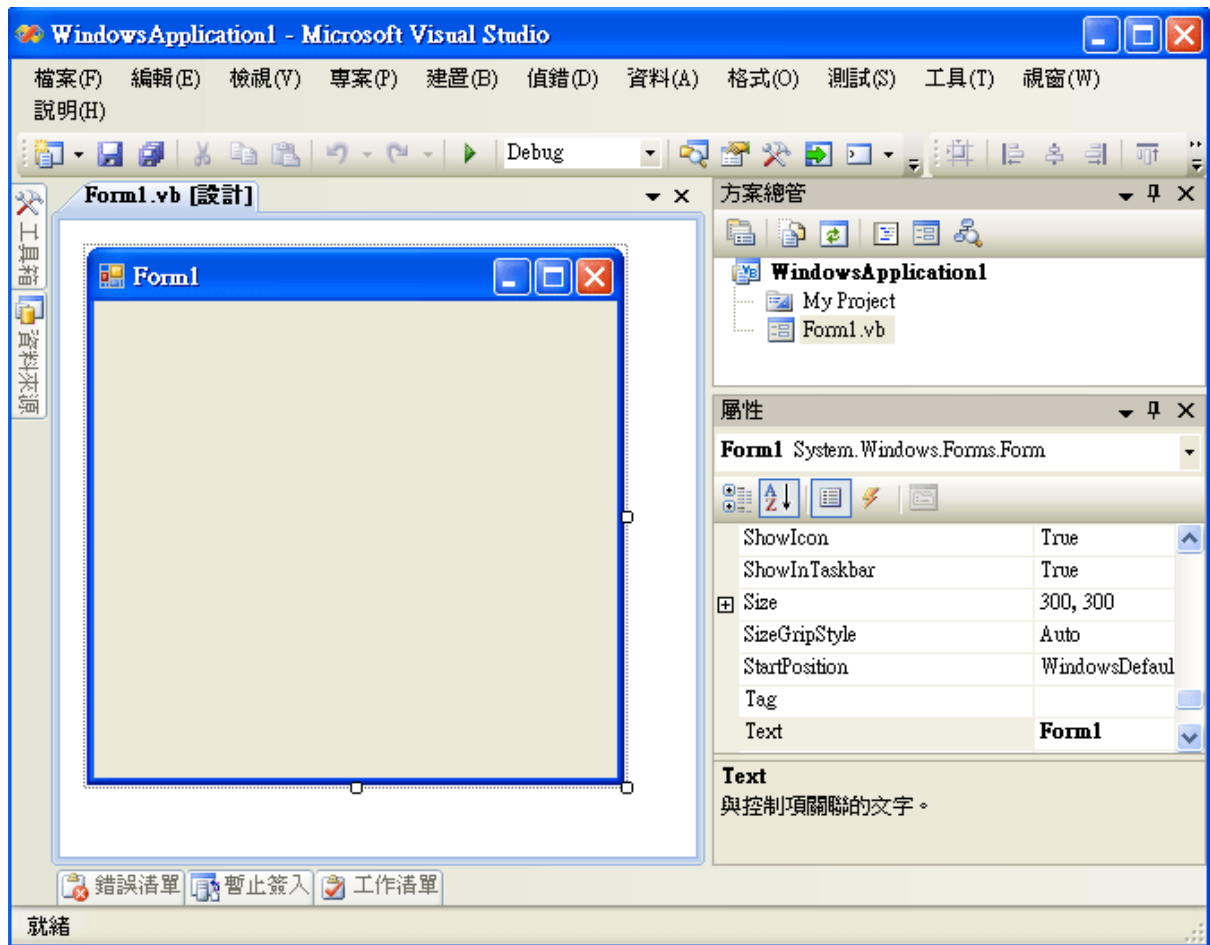
「位置」是本專案會儲存在電腦磁碟的目錄位置，預設是在「我的文件\Visual Studio 2008\Projects 目錄」，當然你也可以自行更改到其他方便的地方，譬如桌面。這部分可以完全使用預設值，但是必須記住儲存的位置，太多的初學者會寫完程式後不知道自己的檔案在哪裡？交作業時可就麻煩了！按下確定鍵等個幾秒鐘，專案就建立好了！

隨著版本與預設喜好的不同，有些時候儲存專案的選擇項目要在你決定存檔時才會出現，未存檔前你的專案是處於暫存狀態！為了預防萬一(當機)請盡可能先存檔，儲存時也應該使用功能表的「檔案」→「全部儲存」，以確保專案之內的所有檔案都有存到！並到檔案總管中找到你的專案目錄，內容應該是這樣的：



其中副檔名為 **sln** 的檔案是專案的總管檔，要重新開啟這個專案時請直接用滑鼠雙擊這個圖示就可以了！副檔名為 **su0** 的模糊圖示是一個隱藏檔，紀錄一些專案使用者的個人設定狀態，刪除它也不會影響專案的執行。真正程式的主要內容藏在以專案名稱為名的目錄之下(在此是 **WindowsApplication1**)，裡面眾多檔案與目錄的意義就暫時不必管它了！只要記得交作業時不能只交一個 **sln** 檔案，而是必須將此目錄包含 **sln** 檔案一起，如此才能夠完整的重新開啟專案。

下面是開啟專案之後的畫面：



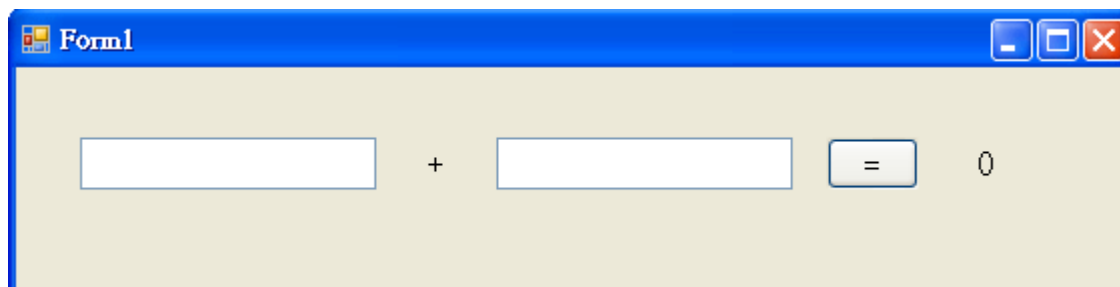
因為是「視窗程式」(Windows Application)，專案預設給一個已經作好的標準視窗，當然隨著使用者的操作設定，此畫面中的每一個附屬視窗都可以開啟、關閉或自動隱藏到上下左右各個邊界(如上圖左側的工具箱與資料來源)，這些視窗如果需要時找不到就到「檢視」功能表去找，在此不一一詳述。但是通常一定會要用到的是「方案總管」、「屬性視窗」與「工具箱」，如果現在沒看到，就把它們先叫出來吧！

「方案總管」記錄著專案內主要的檔案，目前你只會看到 My Project 與 Form1.vb 兩個檔，Form1 代表你看到的預設視窗，在程式內稱呼它為「表單」。事實上它包含著圖形設計介面及程式碼兩個部分，想看表單目前的程式碼時可以在方案總管上面的圖示中找到一個提示為「檢視程式碼」的圖示，點下去就會到程式碼頁面了！同樣的，提示為「設計工具檢視」的圖示代表回到上圖的物件設計頁面，寫程式過程中這一定是常用的功能。至於 My Project 代表的是與整個專案有關的設定，以後用到時再陸續介紹。

「屬性視窗」顯示的是目前被選取物件的屬性，在此預設的物件只有一個，就是 Form1 表單，所以屬性視窗內就是表單的屬性，如果你的英文還好，就可以輕易地學會很多修改表單外觀與特性的功能，譬如改變 BackColor 就會改變背景色，修改 Text(文字)屬性就會改變表單的標題等等。「工具箱」顯示的則是你設計表單內容時可以使用的預設物件，相當於程式零件的倉庫，這也是設計程式時一定需要的。

1-3 四則運算

終於要開始寫程式了！我們準備寫一個可以作加法運算的程式，不過一般視窗程式其實不是從「寫」程式開始的，而是先設計好使用者介面！就是先想好使用者應該如何操作這個程式？要在哪裡填進要計算的數字？要按甚麼鍵可以執行計算？答案要顯示在哪裡？等等…下圖就是我們要設計的畫面：



上圖的物件(或稱控制項 **Control**)都來自於「工具箱」視窗的「通用控制項」分類，空白框框是可以填寫文字的 **TextBox**，在連續叫用時會被自動命名為 **TextBox1** 以及 **TextBox2**，呼叫多個同型控制項時命名規則都是如此，你可以在物件的屬性視窗修改這些名稱，但是目前沒有太大必要；加號與預計顯示答案處的 **0** 是 **Label** 物件，它們在工具箱的圖示是一個大寫的 **A**，顯示的字是在屬性欄的 **Text**(文字)屬性修改的，它們可以顯示文字，但是執行時使用者不能修改內容。最後是寫著等號的 **Button** 物件，通常它圓滿突出的外觀就暗示著使用者按它一下時一定會有事情發生！我們要程式作的計算動作(程式碼)也就是寫在按它一下的「事件」之中！現在就用滑鼠點它兩下，畫面會自動切換到程式碼頁面，還會自動產生一個事件副程式的框架，如下：

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _  
    Handles Button1.Click  
  
End Sub
```

在實際操作的畫面中，上面程式碼的前兩行其實是同一行，但是為了編排版面之方便，我們將它以底線""標記斷行，這是 **VB** 程式本來就允許的語法，完全不會影響程式的執行。因為通常副程式的標題行都很長，本書之後的程式碼都會這樣處理，適時給予斷行，各位寫程式時則不必完全模仿。這個程式框架稱為「事件副程式」，標題最後 **Handles** 之後的 **Button1.Click** 就是指出這段程式必須執行的時機！或者說發生 **Button1.Click**「事件」時就應該執行此段程式碼。**Click** 是滑鼠點一下的意思，**Button1.Click** 就是按鍵 **Button1** 被滑鼠點一下。接著加入計算用的程式碼如下：

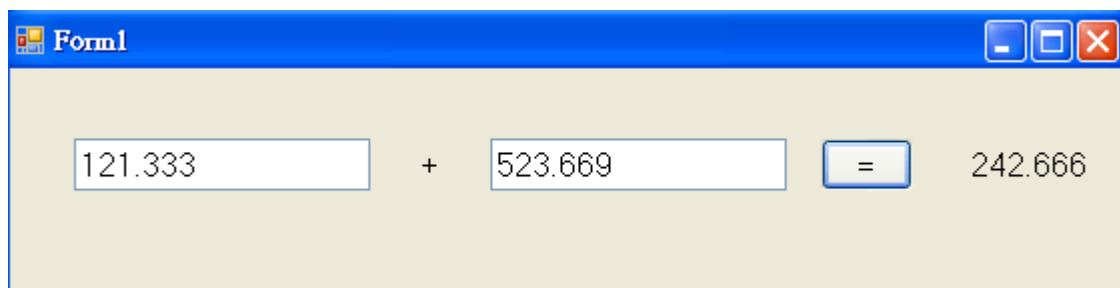
```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _  
    Handles Button1.Click  
    Dim A As Double = Val(TextBox1.Text)  
    Dim B As Double = Val(TextBox2.Text)  
    Dim C As Double = A + B  
    Label2.Text = C.ToString  
End Sub
```

Dim 是宣告的意思，任何變數在使用之前都應該宣告，主要目的是告訴電腦這個變數的資料型態，譬如它是一個「數字」或「文字」？如果是數字的話，是一個整數？還是有小數點的實數？在此我們宣告的是數字，也要計算小數，所以就使用 **Double** 資料型態，中文稱為「倍準數」，它可以有 15 位有效數字！如果不必如此精確，也可以使用 **Single** 資料型態，只有 7 位有效位數，或者使用整數 **Integer**。當然，越簡單的資料型態電腦算得越快。其他的的資料型態還有很多，需要時再陸續介紹。

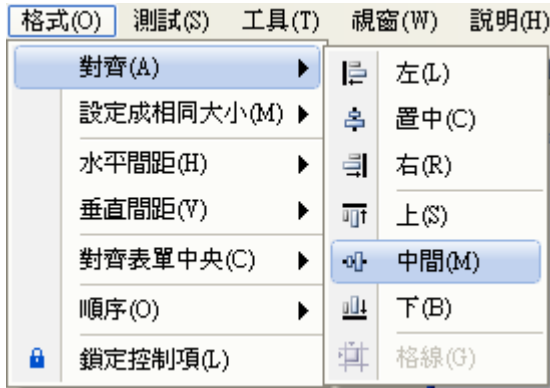
上述程式先宣告兩個變數 **A** 與 **B**，分別記下 **TextBox1** 與 **TextBox2** 內填入的數字。函數 **Val** 的作用是将文字型態轉為數值，對於電腦來說，「文字方塊」內輸入的都算「文字」，即使你打的是阿拉伯數字也一樣！使用 **Val** 是明確告知電腦某資料必須當數字使用。通常在簡單的狀況下 **VB** 可以自動轉換文數字，不過你最好隨時記住自己處理中的資料是文字或數字？這樣如果文數字交錯，程式無法處理而當掉時就不會霧煞煞了！

接下來使用變數 **C** 作為 **A+B** 的答案，最後將 **C** 轉成文字(**ToString**)顯示於 **Label2** 標籤的文字中。如果你是第一次寫程式，在此要先習慣於程式碼中的「**=**」符號其實常常是「設定」或「指定」的意思，譬如 **C=A+B** 意思是將 **A+B** 的答案「設定」給 **C**！就是輸入值在右方，而目標在左邊；這與一般數學計算時左邊是輸入資料，右邊為答案 (**A+B=C**) 不一樣！開始習慣它吧！

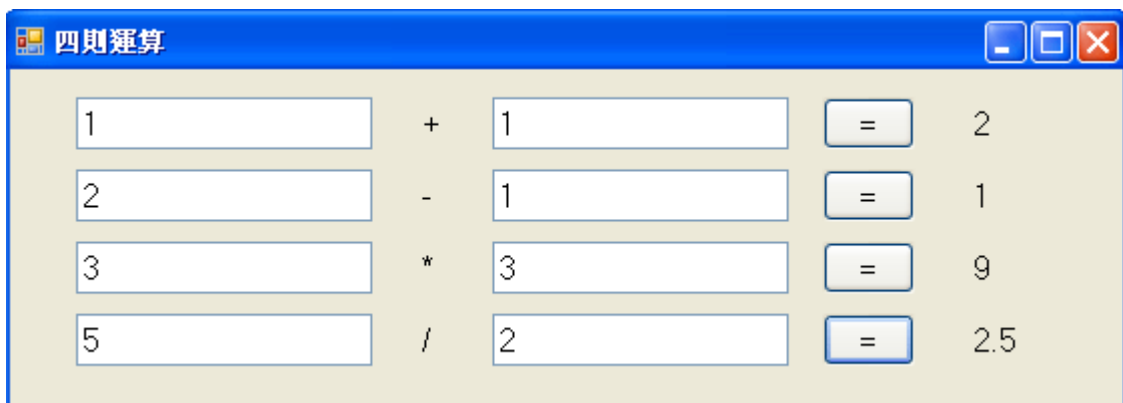
試試看！你的第一個 **VB** 程式應該可以執行了！啊？不知道怎麼執行？方法多了！最簡單的是按下「**F5**」按鍵，或者點擊工具列的一個綠色三角形圖示，滑鼠移上去時它會顯示「開始偵錯」，或者用功能表的「偵錯」→「開始偵錯」。執行畫面如下：



你在設計上述畫面，用滑鼠拖曳物件時一定會感受到軟體有提供自動對齊的功能！一些引導線會不時出現，輔助你將物件對齊。如果還是不滿意，可以將需要對齊的物件用滑鼠一起同時框選起來，或加上 **Ctrl** 鍵可以連續點選多個物件，再到功能表的「格式」→「對齊」選項中，選擇適當的選項處理(如下圖)，相信一定可以讓你對得非常整齊，甚至物件的間距也可以調整。



試試看！模仿上述過程製作減法與乘除法介面及程式，結果如下：



```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Button2.Click
    Dim A As Double = Val(TextBox4.Text)
    Dim B As Double = Val(TextBox3.Text)
    Dim C As Double = A - B
    Label3.Text = C.ToString
End Sub

Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Button4.Click
    Dim A As Double = Val(TextBox8.Text)
    Dim B As Double = Val(TextBox7.Text)
    Dim C As Double = A * B
    Label7.Text = C.ToString
End Sub

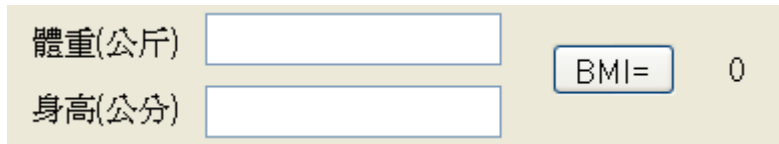
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
Handles Button3.Click
    Dim A As Double = Val(TextBox6.Text)
    Dim B As Double = Val(TextBox5.Text)
    Dim C As Double = A / B
    Label5.Text = C.ToString
End Sub

```

1-4 使用數學函式庫

加減乘除只是小學生的程度，如果碰到複雜一點的數學計算怎麼辦？譬如大家關心

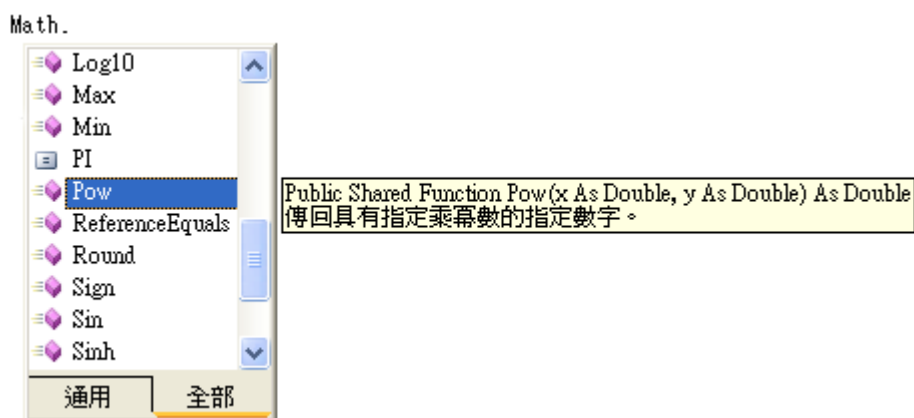
體重太重常常會計算的 BMI 值公式是： 體重(公斤) / 身高(公尺)² 程式怎麼寫？可以先設計以下介面讓使用者輸入身高體重，其中兩個文字方塊分別為 TextBox9 與 TextBox10，按鍵為 Button5 答案為 Label11，如果你的物件名稱因為複製順序稍有不同，稍後請修改一下對應的程式碼即可。操作介面與程式碼如下：



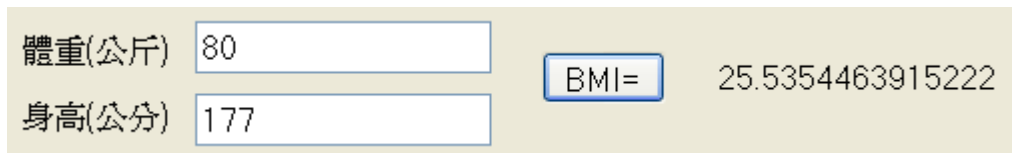
```
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _  
    Handles Button5.Click  
    Dim W As Double = Val(TextBox9.Text) '體重公斤  
    Dim H As Double = Val(TextBox10.Text) '身高公分  
    H = H / 100 '身高變成公尺單位  
    Dim BMI As Double = W / Math.Pow(H, 2)  
    Label11.Text = BMI.ToString  
End Sub
```

上面程式碼中居然有中文！請注意看它們的前面有一個小小的英文單引號『』這是 VB 中寫程式「註解」的方法，在編寫時它們會變成綠色，與一般程式碼預設的藍或黑色不同，程式執行時不會處理這些註解，但是可以幫設計者記憶程式的意義也讓其他讀程式的人容易理解，建議同學養成勤於加註解的習慣。在實務上，註解還有一個用途，就是當某些程式碼暫時不用，但是以後可能還要參考不想刪除時，那就在前面點一下變成註解就好了！

上述程式先取得身高體重的輸入值，再將身高(公分單位)除以 100 變成以公尺為單位，計算 BMI 值的時候使用 Math.Pow 函數計算身高的平方值。在同學打字到 Math 時應該會看到如下的自動提示選項，Math 在微軟的軟體工具架構中可以視為一個與數學相關的函式庫，裡面內建很多數學函數，包括下面畫面看到的最大(Max)與最小值(Min)、三角函數與對數(Log)，甚至數學常數如圓周率 PI(π)等等。最可貴的是當指標移到某一函數名稱時旁邊還會進一步出現中文化的提示！提示中會有函數的簡要功能說明，含有語法格式，你真的很難辯稱不會使用了！（除非數學課沒學啦！）

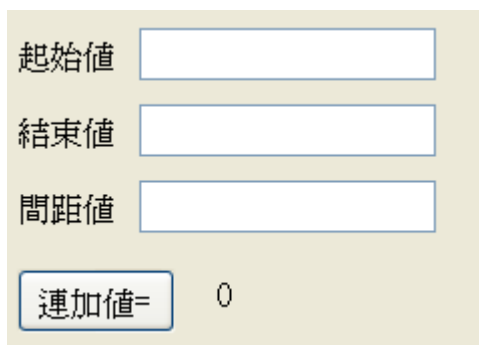


上例中使用次方函數 `Pow(H, 2)` 計算 `H` 的平方值，事實上 `VB` 也可以用 `H^2` 的語法做出一樣的次方計算，此範例的主要目的在介紹 `Math` 函式庫的存在，有了它就不必擔心所有的數學問題了！曾聽到有人說 `VB` 不「善於」作數學運算，所以必須用 `C++` 或 `MatLab` 等語言或軟體工作，其實是沒有必要的！個人就常常使用 `VB` 程式進行需要高度計算功能的影像處理與辨識工作，沒問題的！下面是 `BMI` 程式執行的畫面：



1-5 用迴圈作連加法

電腦常常用來處理重複性高的工作，尤其是大量的數值運算，這時就必須用到迴圈，在此我們以連加法的計算來介紹這個程式設計的重量級語法。首先請設計介面如下：



上述三個文字框物件在此範例中分別為 `TextBox11`、`TextBox12` 與 `TextBox13`，按鍵為 `Button6`，答案標籤為 `Label15`，程式碼設計如下，如果你的物件名稱因為複製過程稍有不同，請因應物件的位置修改一下。

```
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _  
    Handles Button6.Click  
    Dim A As Double = Val(TextBox11.Text)  
    Dim B As Double = Val(TextBox12.Text)  
    Dim C As Double = Val(TextBox13.Text)  
    Dim Ans As Double = 0  
    For i As Double = A To B Step C  
        Ans += i  
    Next  
    Label15.Text = Ans.ToString  
End Sub
```

上述程式碼先用變數 `A`、`B` 與 `C` 取得連加法的起始、結束與間距值。譬如問題是：1 到 100 之間的奇數總和，在此就是 `A=1`，`B=100`，`C=2`，接著宣告答案變數 `Ans` (初值為 0)。迴圈運算的基本語法是起於 `For` 結束於 `Next` 指令，中間的程式碼就是要重複運算的動作，在此就是依序加入連加的數字，如前面的例子就是依序加入 1, 3, 5, 7... 到 99

的奇數值。Ans += i 是程式語言特殊的簡寫，意義與 Ans = Ans + i 完全相同。

重要的語法關鍵在 For 的敘述，它起於宣告一個指標變數「i」為一個倍準數(與需要計算的起始及結束值同類型)，等號後面交代此變數 i 的起終點及間隔，語法如：[起點] To [終點] Step [間距]。在連加法的例子中，我們直接使用指標變數 i 為需要被加入答案的連續數字，事實上多數時候迴圈的指標變數只作為計次之用，所以資料類型多半只用 Integer，且 Step 都是 1，所以如果你忽略了 Step，程式會自動認定間距是 1！如下是連加法的執行畫面之一：

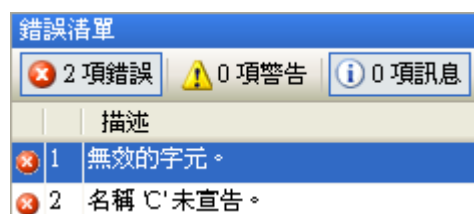
起始值	<input type="text" value="1"/>
結束值	<input type="text" value="100"/>
間距值	<input type="text" value="2"/>
連加值=	2500

1-6 除錯基礎

有人說寫程式時間為 1 的話，除錯的時間至少是 3！即使是資深程式師寫程式也很少一寫就全對！差別是高手可以很快知道錯誤的位置與原因，迅速更正，菜鳥就不知所措了！所以程式設計老師第一節課總會為了幫學生除錯忙到爆！事實上，現在的設計軟體可以幫非常多的忙！首先在你打字時電腦已經在檢查程式碼了！如果確定是錯誤的狀況，文字底下會有紅色底線標示，如果只是警告，提示你執行時可能會出錯，就是綠或藍色波浪線！如下圖是因為將代表乘法的「*」號打成全形的「·」號，電腦無法正確決定 C 值，最終就說 C 是錯的！

```
Dim C As Double = A · B
Label7.Text = C.ToString
```

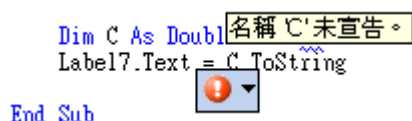
此時你打開「錯誤清單」(通常自動出現於主視窗下方)會看到如下畫面：



告訴你錯誤的情況，「無效的字元」當然就是說電腦看不懂那個字囉！因為看不懂，因此也算不出 C 值，所以認為 C 沒有宣告完成(未宣告)。看懂電腦說話的邏輯了嗎？此時你點擊錯誤清單的任何一個項目也會跳回到程式碼出錯的地方。事實上不用到此視

窗，僅用滑鼠擺在出錯的地方也會出現工具提示，像這樣：

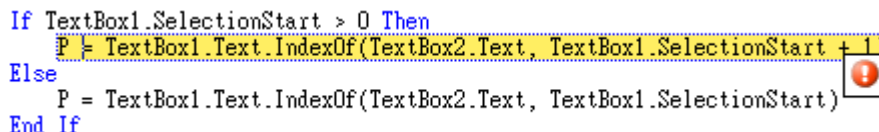
```
Dim C As Double
Label7.Text = C.ToString
End Sub
```



按下紅色驚嘆號圖示還會有更詳細的說明與修改建議。

另一種情況是寫的時候好好的，執行中卻出了問題無法繼續而當掉，畫面就會變成錯誤行被黃色背景醒目標示，你一樣可以發現很多錯誤說明。如下圖：

```
If TextBox1.SelectionStart > 0 Then
    P = TextBox1.Text.IndexOf(TextBox2.Text, TextBox1.SelectionStart + 1)
Else
    P = TextBox1.Text.IndexOf(TextBox2.Text, TextBox1.SelectionStart)
End If
```



寫程式的除錯機制非常多，多數初學者因為有這些機制不難找到錯誤的地方，但是多半看不太懂電腦說話的方式，只能建議多看資料多問老師，慢慢習慣就好！最常見的錯誤說明提示通常也是最詳細完整的，耐心閱讀必有回報。

1-7、進階挑戰

一、如何建立一個輸入角度計算各種三角函數值的程式？

提示：使用 **Math** 函式庫，請注意三角函數運算時弧度與一般習用度數之轉換。

二、如何設計一個攝氏與華氏互相轉換的程式？

提示：攝氏=(華氏-32)*5/9。

三、如何設計一個英制身高(英尺英吋)與公制身高(公分)互相轉換的視窗程式？

提示：1 英尺=12 英吋；1 英吋=2.54 公分。

課後閱讀

希望 VB 程式設計的第一堂課是讓大家充滿興奮與成就感的經驗，本書是為程式設計入門課程而寫，希望有別於多數以嚴謹理論及詳實技術為架構的書籍，我們不期待一門課或一本書就能讓大家成為無所不知的高手，但是很希望短時間內讓多數用功與「不太用功」的同學們都能充分體驗到程式設計的樂趣與無限的可能性。

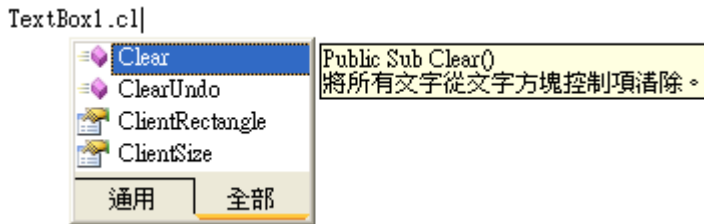
理論與技術細節是比較枯燥的，本書一開始就不以此為編寫目標，而是希望以兼具趣味性與實用性的程式範例為核心，讓同學們從作中學！如何引發與持續同學們的學習動機是本書撰寫的最高守則。老實說，只要你想學，網路世界裡有甚麼資訊找不到呢？但是當你玩得高興之餘，如果能適時地多學一點理論概念一定會學得更快，也走得更遠！因此在每個單元辛苦完成程式操作之後，就以輕鬆的心情，在此課後閱讀篇多學一點相關的理論與技術知識吧！

物件導向程式設計概念—屬性、方法與事件

程式設計的目的就是在電腦上製作出一些有用(或好玩)的「物件」！我們知道任何一部電腦的細節都複雜到沒有一個人可以完全了解，我們周遭的多數科技產品也是如此，但是只要學會一點基礎概念，多數人都可以自己買零件組裝一台電腦！關鍵就是多數複雜或者龐大的「物件」其實都可以拆解成很多的小物件(零件)，如果你想要做個大物件，手邊又可以找到足夠的零件，那麼只要組裝一下，你也可以輕鬆的「作」出一台電腦！這就是物件導向程式設計的基本精神：盡量使用現有物件以組裝的方式完成你的程式。

在本單元中你應該已經從使用工具箱體驗到組裝物件的快感，事實上工具箱裡的所有東西都是前人寫好的程式，你不必重寫，直接抓來用就可以了！但是當然零件未必百分百的合用，譬如螺絲釘可能剛剛好不夠長，拴不住兩塊木板。這時「程式零件」就比實體物件好用多了，這些「物件」的靜態外觀或一些性質我們稱之為「屬性」(Attributes)，譬如文字方塊的長、寬、顏色或字型大小等等。它們多數都是可以修改的，我們可以在屬性視窗修改，可以在設計頁面直接用滑鼠拖曳修改，也可以直接用程式碼改寫！另一方面，我們在學數學操作時常常會使用到 XYZ 等等變數，它們在我們寫程式的過程中也被視為「物件」哦！要用它們時宣告變數就等於製造出需要的物件了。

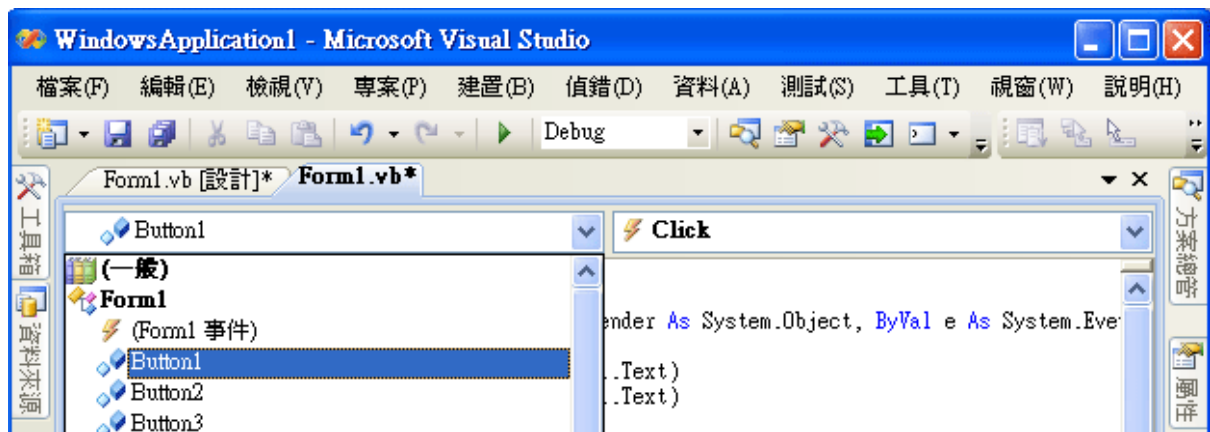
當然程式物件不只有靜態的屬性，多數也有功能與動作，就像馬達物件就會旋轉一樣，程式設計的術語稱之為「方法」(Method)，譬如本單元有用到 C.ToString 的語法就是請 C 變數「將自己轉換(To)成文字(String)」，這就是一個功能或動作。當你打程式碼時，寫完物件名稱繼續打入小數點(".")會發現類似下圖的提示：



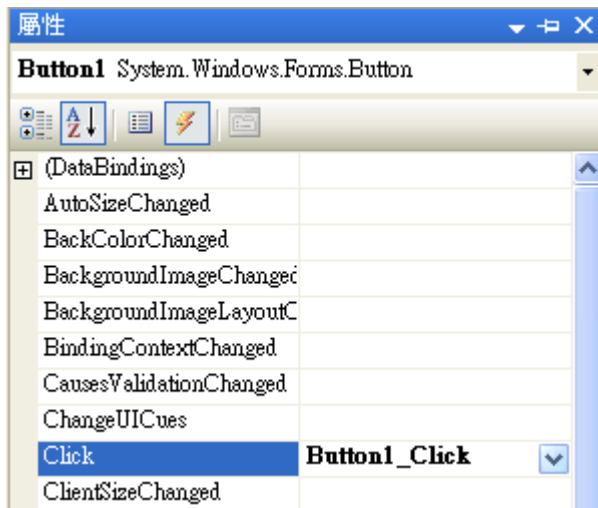
這些選項之前如果是桃紅色像在飛行的塊狀圖示就是動態的「方法」，文件上方有隻手的圖示就是靜態的「屬性」。你會發現多數物件的方法與屬性都有好幾十個之多！

除此之外，多數物件還可以對一些使用者的操作或系統引發的「事件」(Event)有反應！最典型常見的例子就是本單元用到的按鍵(Button)被滑鼠點到時可以產生 `Button.Click` 事件！現在多數的程式執行方式都是由「事件驅動」的！意思是程式一開始將必要的準備動作做好之後就會停下來，接下來如何運作要看使用者如何操作？就是製造出不同的「事件」讓程式作出不同的回應啦！譬如操作提款機時，你按下「轉帳」或「提款」按鍵就會有不同的回應程序。當然，每個事件該如何反應就要靠你寫程式去定義了！就像本單元我們寫的四則運算程式，我們必須將腦袋裡想的數學公式變成電腦可以理解的程式碼。

還要記得「事件」是與「物件」相關的，初學者常常搞不清楚，將程式碼寫在不同物件的相同事件裡面！然後問老師我的程式怎麼沒反應？所以當我們寫事件反應程式時請務必先選對物件！在 VB 程式碼頁面可以看到下面兩個下拉式選單，左邊是物件(目前選的是 `Button1`)，右邊是事件(目前選 `Click`)，你會發現任何一個物件都可以回應很多種事件，開始寫事件反應程式碼的標準作法應該是在這裡選擇的！



但是為什麼前面操作時都說要以「滑鼠雙擊」物件來產生程式碼框架呢？這是 VB 為了減少你的操作步驟貼心的設計。當我們想寫某物件相關的事件副程式時，直接點它 VB 會自動選擇「最常用的事件」來開啟程式碼框架，稱為「預設事件」！當然如果你只會這樣玩，那麼要寫「非」預設事件的程式時就糟了！事實上，在屬性視窗上面還有個閃電標誌，也是可以選擇產生不同事件副程式的地方，如下圖參考一下囉！



總之，記得現代化的程式基本上是由很多現成的程式「物件」組裝而成的，物件通常包含：「屬性」、「方法」與「事件」三類成員，能夠了解這些概念，你的第一節課學習成果已經超好了！繼續加油哦！