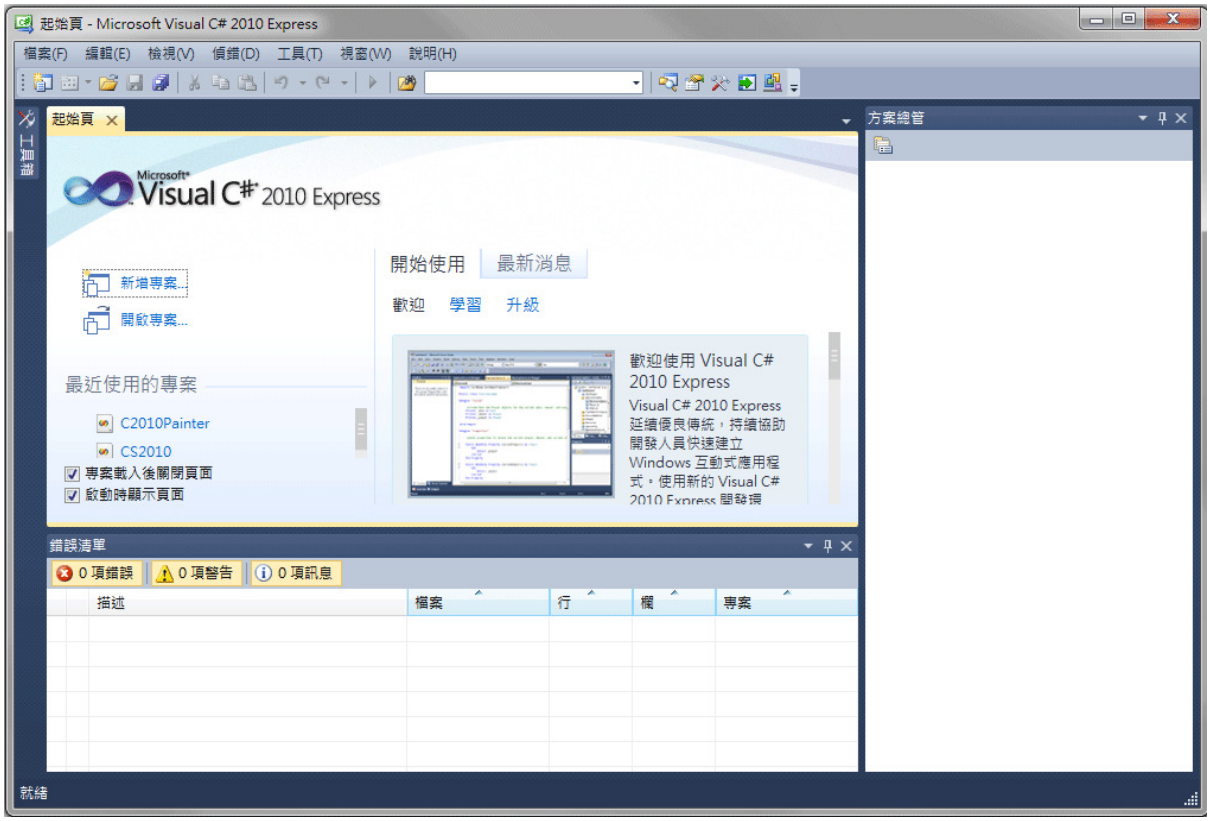
第 1 章　用 C#作數學

簡介：

本章是我們學習 C#程式語言的第一章，在此使用的是可以自微軟官方網站合法免費下載的 Microsoft Visual C# 2010 Express 軟體。如果您使用較完整的 Visual Studio 付費版本，主要差別只是他們涵蓋多種語言，建立專案時必須選擇 C#語言而已。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 C#程式作簡單的數學計算開始我們的學習之旅吧！

1-1 認識 Visual C# 2010



上面是我們打開 Visual C# 2010 Express 軟體時出現的畫面，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都是為了減少設計者的工作量，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最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看起來很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你想要的程式。不必要求自己在第一天就孰悉所有的操作環境與功能，而是讓「寫程式的目的」來主導！

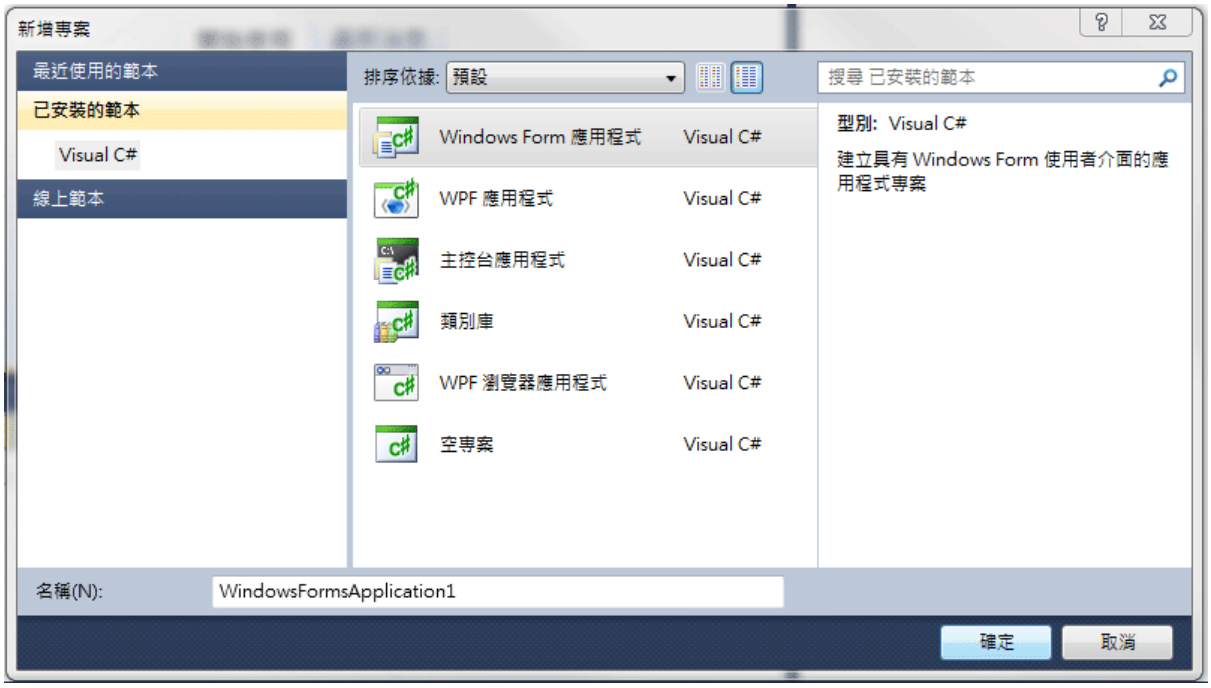
依據需求一步一步地找到你需要的功能，也逐步學會使用複雜龐大的程式設計軟體。

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(使用按鍵 F1)，真正的學習障礙應該只有一個！就是你自己不動啦！

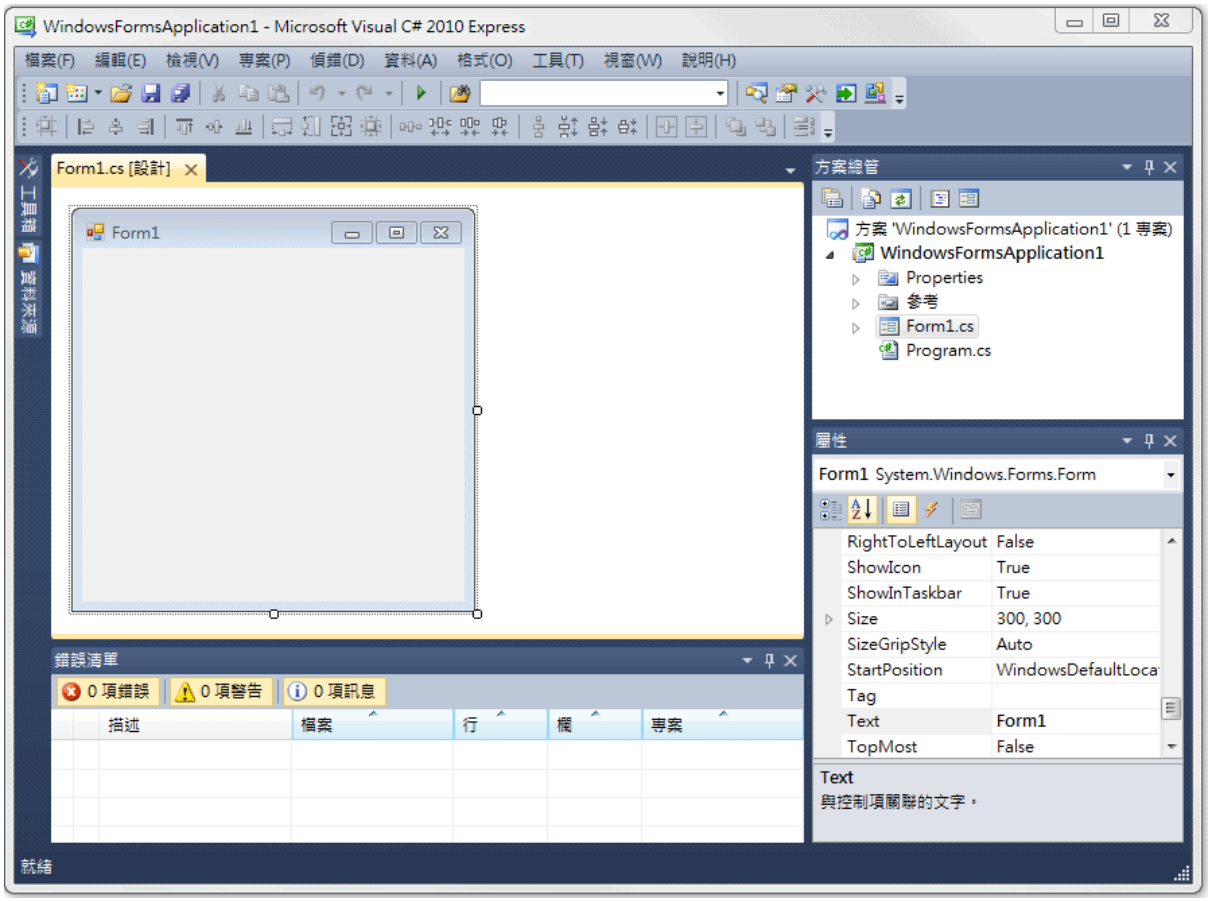
1-2 建立專案

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，可以幫甚麼都不懂的新老闆(就是你啦！)自動完成大多數的工作。

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：

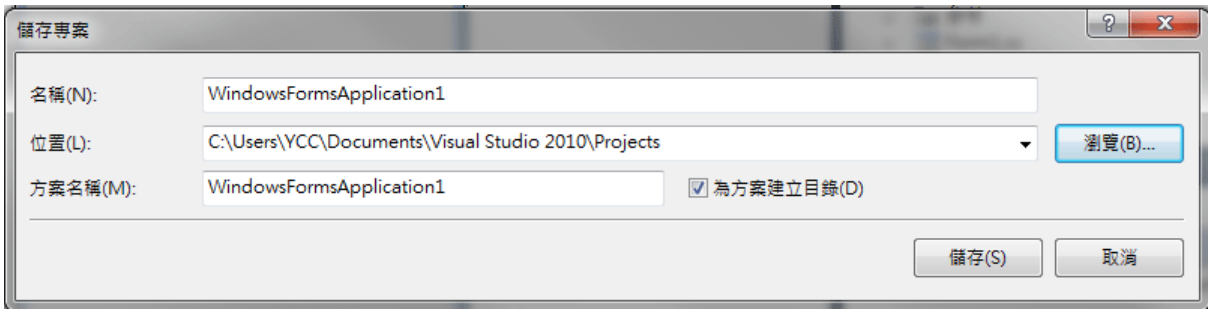


請選擇專案的種類，本書所有範例都是使用 Windows Form 應用程式，就是製作標準視窗程式即可。不同的專案選項只是依據最後要完成的程式種類，盡量預先準備可能需要的資源而已。接下來請看視窗下方可以編輯的文字框：「名稱(N)」這裡請鍵入你準備給程式專案取的名稱，使用預設名稱也無妨。按下確定鍵等個幾秒鐘，專案就建立好了！新增的專案畫面如下：



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、關閉或自動隱藏到左右邊界(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！

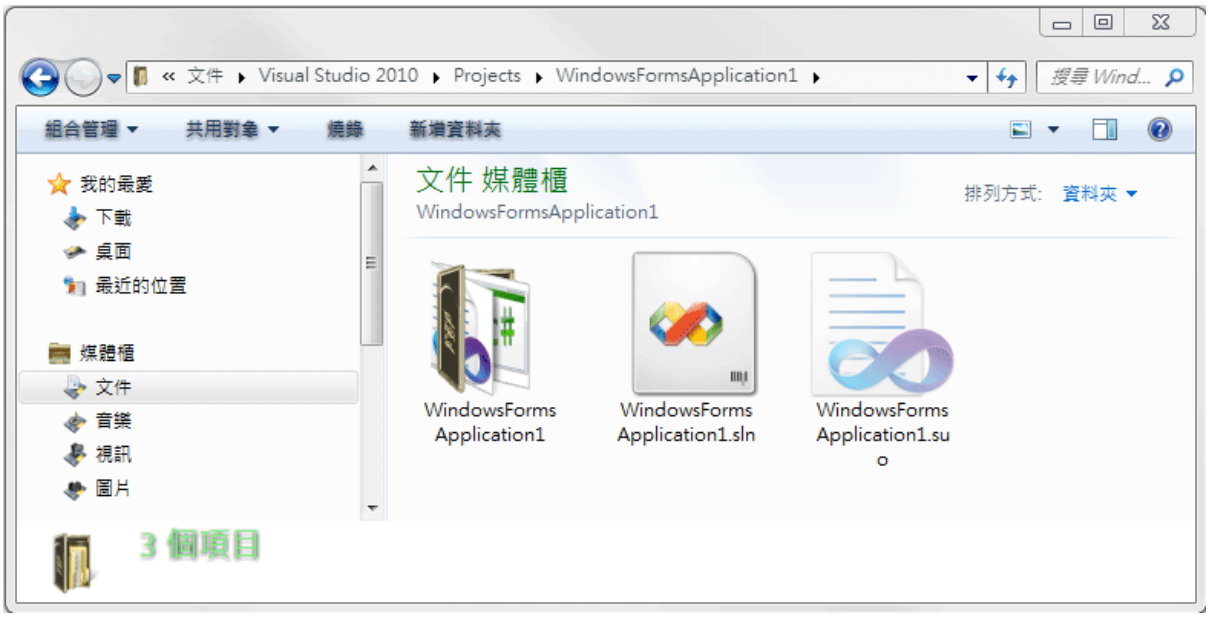
此時為了預防萬一(當機)請盡可能先存檔，儲存時應該使用功能表的「檔案」→「全部儲存」，以確保專案之內的所有檔案都有存到！工具列上多個磁碟片的圖示也有一樣的功能。接著會出現如下小視窗：



當然你可以更改名稱與儲存位置，預設位置會在我的文件(My Documents)裡面的 Visual Studio 2010\Projects 裡面，Project 就是程式專案的意思，每個程式專案會有一個

獨立的目錄。這裡的「方案名稱」或許會讓你感覺迷惑，它代表的是比專案(Project)更高一級的程式單位，一個方案可以包含多個專案，在此我們都當作一方案內只有一專案即可。

按下儲存之後接著請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 suo 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 WindowsFormApplication1)，裡面眾多檔案與目錄的意義就暫時不必管它了！只要記得要移動專案或交作業時不能只交一個 sln 檔案，而是必須將此目錄包含 sln 檔案一起移動，如此才能夠完整的重新開啟專案。

回到開啟的專案中的「方案總管」視窗，此處記錄著專案內主要的檔案，目前你會看到 Properties、參考、Form1.cs 以及 Program.cs 四個項目。其中 Form1.cs 代表你看到的左邊主要窗格內的預設視窗，cs 意謂 C Sharp(C#的唸法)。在程式內稱呼它為「表單」，也是我們寫程式時工作的主要對象。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先點選.Form1.cs，再到上面的幾個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！

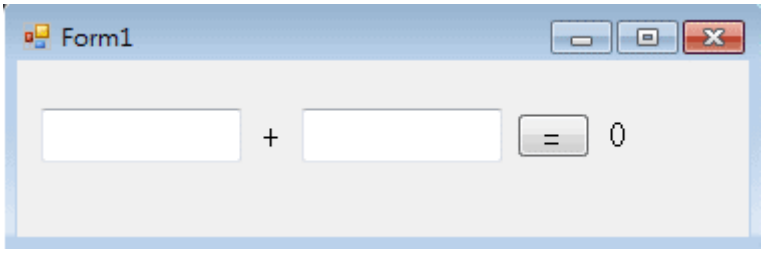
同樣的，提示為「設計工具檢視」的圖示代表回到上圖的物件設計頁面，寫程式過程中這一定是常用到的功能。至於 Properties 代表的是與整個專案有關的屬性設定，「參考」表示目前專案引用的函式庫項目，Program 則是整個專案的主程式，這些檔案目前都不必處理，以後用到時再陸續介紹。

「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(或稱組件、元件)的倉庫，這也是設計程式時一定需要的。

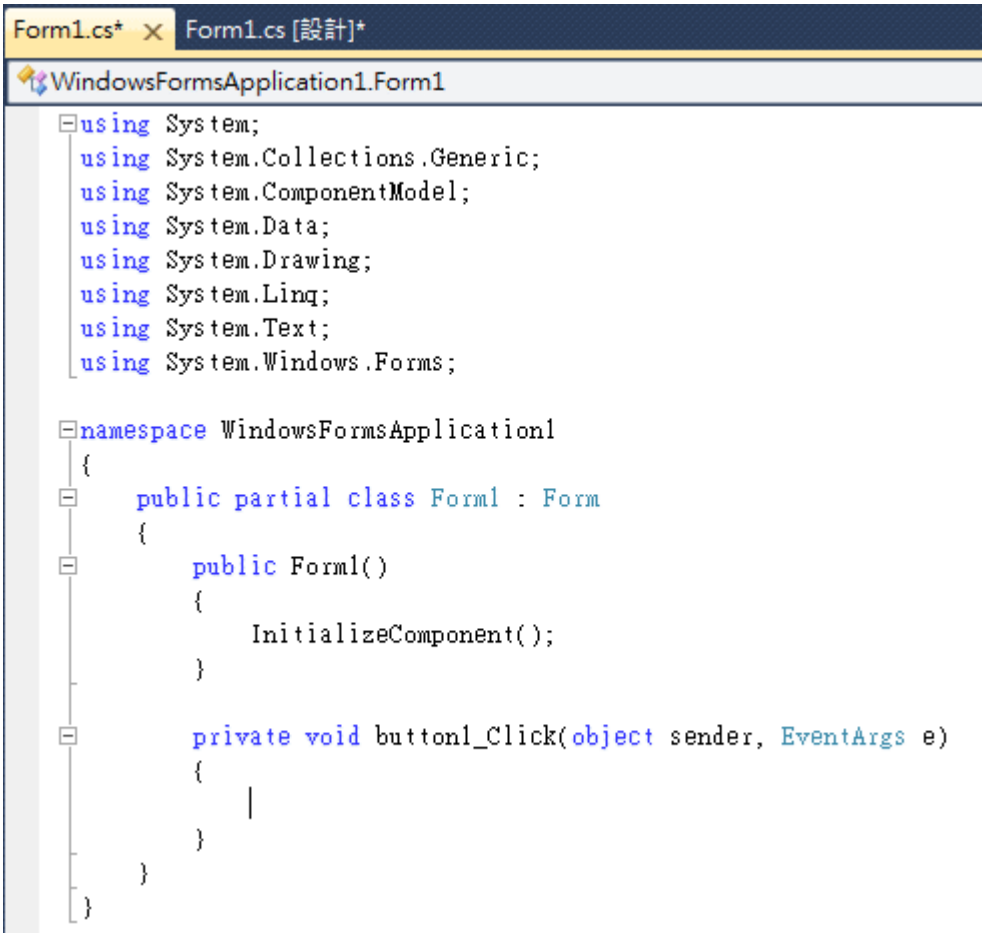
1-3 四則運算

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 textBox**1** 以及 textBox**2** 依此類推，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，但是目前還沒有必要。

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，畫面如下：



最上方顯示的是圖形化設計畫面(表單外觀)與程式碼編輯視窗的頁籤選擇，你可以隨時在此切換物件設計與程式碼撰寫工作。程式碼的最上面會有很多以「using」開頭的程式碼代表本專案預設引用的函式庫，我們寫程式其實是使用很多的內建函式組裝的，整個函式庫的正式名稱是「.NET Framework」(在此為 4.0 版)，它有如圖書館一般，有很龐大的階層式分類，如 System.Text 就表示與文字處理相關的函式集合，也稱之為 Namespace(命名空間)，因為整個函式庫太大了，我們只會因應需要載入部分，初學者只要用預設的設定就夠了。

接下來會看到 namespace WindowsFormApplication1，這是本專案的命名空間，換言之，我們接下來寫的程式也被視為一個函式庫或命名空間，然後是 public partial class Form1:Form，這代表表單一(Form1)的程式單元，class 翻譯成類別，在物件導向程式語言中就是一個程式物件的模型。後面多寫的一個 Form 是表示此類別繼承自稱為 Form 的預設類別，也就是說在我們還沒設計任何東西時表單已經有個預設樣板(程式)作為基礎的意思。

至於 partial 原意是「部分」的意思，也代表此處並沒完整包含所有 Form1 的程式碼，那其他的程式碼在哪裡？你可以在方案總管的 Form1 選項展開看到如 Form1.Designer.cs 這樣的檔案，裡面其實是你設計頁面物件時軟體自動幫你產生的程式碼，這兩個檔案的程式碼其實都會一併執行，因此寫在一起或將某些程式碼交換位置其

實都可以的。

接下來是一段程式碼：

```
public Form1()
{
    InitializeComponent();
}
```

這是初始化物件的程式，就是執行上述 Form1.Designer.cs 中的 InitializeComponent 副程式的意思，那裡面定義了目前所有物件的屬性，如位置、大小、文字或顏色等等。你可以想像就是將我們設計的頁面，以及上面的物件一一在電腦上畫出來的意思。通常我們也不需要管它，知道就好！(但不能刪掉哦！)

在此要開始注意到 C#語言的幾個基本語法結構：每一個程式區塊或段落都必須以大括號為邊界，小括號通常代表寫參數的地方，只要是代表執行某些動作(方法)的程式碼後就必須附加小括號，即使沒有參數需要寫，空括號也是必須保留的！如果是靜態的屬性像是物件的長寬顏色等等，就不必小括號。最後每個指令完成還必須加上一個分號(;)！這些括號與分號可以說是 C#語言初學者的噩夢！只要一個括號分號漏寫或多寫都會造成程式錯誤無法執行的！

終於來到必須自己寫程式的地方了！剛剛我們雙擊 button1 物件時產生了標題為 private void button1_Click…的一個副程式框架，其中 private 表示此程式僅供表單內部使用，void(空的)表示程式執行完不會回傳任何訊息資料，Click 就是滑鼠點一下的動作了！在此應該寫的程式碼如下：

```
private void button1_Click(object sender, EventArgs e)
{
    double a = double.Parse(textBox1.Text);
    double b = double.Parse(textBox2.Text);
    double c = a + b;
    label2.Text = c.ToString();
}
```

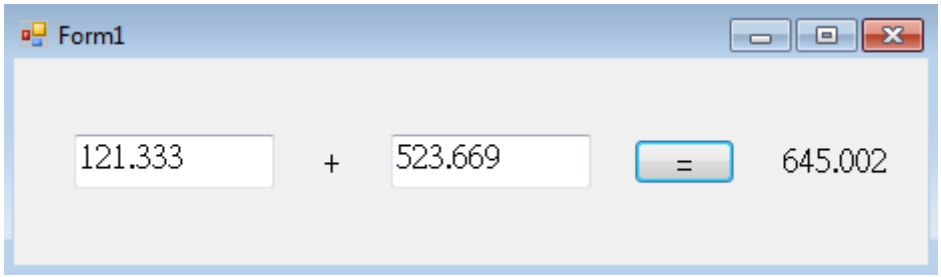
其中 double **a** 的語法是「宣告」a 是一個倍準數，任何變數在使用之前都應該宣告，主要目的是告訴電腦這個變數的資料型態，譬如它是一個「數字」或「文字」？如果是數字的話，是一個整數？還是有小數點的實數？在此我們宣告的是數字，也可能要計算小數，所以就使用倍準數 double 資料型態(佔 8 個 byte)，它可以有 15 位(十進位)有效數字！如果不必如此精確，也可以使用浮點數 float 資料型態(佔 4 個 byte)，只有 7 位有效位數，或者使用整數 int(佔 4 個 byte)。當然，越簡單的資料型態電腦算得越快，也越節省資源。其他的資料型態還有很多，用到時再陸續介紹。

上述程式先宣告兩個變數 a 與 b，分別來自 textBox1 與 textBox2 內填入的數字。函數 double.Parse 的作用是將文字資料轉為倍準數的意思，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！當然如果你宣告的資料型態是 float 就應該用 float.Parse；如果是 int 就必須用 int.Parse 了！

C#語言重要的標準語法之一就是每行指令碼都必須以一個分號(；)表示結束。事實上 C#語言編輯器只會辨認分號，文字換行與否或連續空白則是完全忽略的！也就是說你可以將好幾行程式碼以分號相隔串成一行，程式照樣執行不會錯！相對的，即使你將程式碼分行了，但只要沒打分號作結束就是錯的！很多時候為了閱讀段落清晰起見，程式碼會被加入很多空行，這對程式也是毫無影響的。

接下來程式使用變數 double 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，而答案在左邊；這與一般數學計算時左邊是輸入資料，右邊為答案(a+b=c)相反！開始習慣它吧！

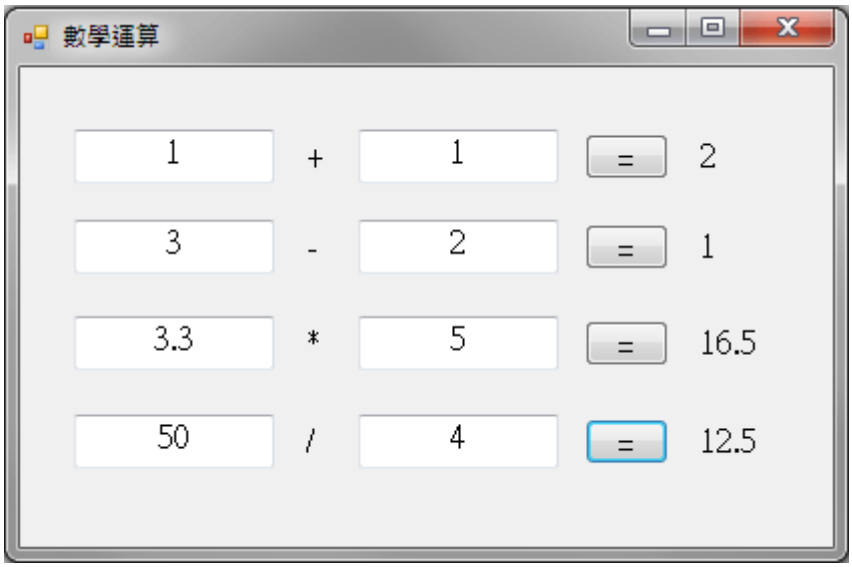
試試看！你的第一個 C#程式應該可以執行了！啊？不知道怎麼執行？方法多了！最簡單的是按下「F5」按鍵，或者點擊工具列的一個綠色三角形圖示，滑鼠移上去時它會顯示「開始偵錯」，或者用功能表的「偵錯」→「開始偵錯」。執行畫面如下：



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，提醒您！物件與程式碼都可以大量選取、複製及貼上，善用複製可以快速產生類似的介面與程式。還有在所有程式語言中乘法都是以星號(*)代表，除法以斜線(/)代表，請**勿**使用 × 或 ÷ 的記號。設計

結果如下：



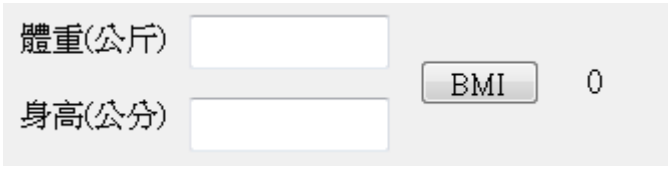

```
private void button2_Click(object sender, EventArgs e)
{
    double a = double.Parse(textBox4.Text);
    double b = double.Parse(textBox3.Text);
    double c = a - b;
    label3.Text = c.ToString();
}

private void button3_Click(object sender, EventArgs e)
{
    double a = double.Parse(textBox6.Text);
    double b = double.Parse(textBox5.Text);
    double c = a * b;
    label5.Text = c.ToString();
}

private void button4_Click(object sender, EventArgs e)
{
    double a = double.Parse(textBox8.Text);
    double b = double.Parse(textBox7.Text);
    double c = a / b;
    label7.Text = c.ToString();
}
```

在此必須注意的是多個物件一起複製時物件名稱次序未必完全如你所預期，譬如上面的 textBox1 與 textBox2 一起複製後，新物件名稱次序變成 textBox4 與 textBox3！總之，寫程式時必須自己一一確認物件位置與名稱是否相符。

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？譬如平方或開根號等等。我們試試看寫出大家關心體重常常會計算的 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

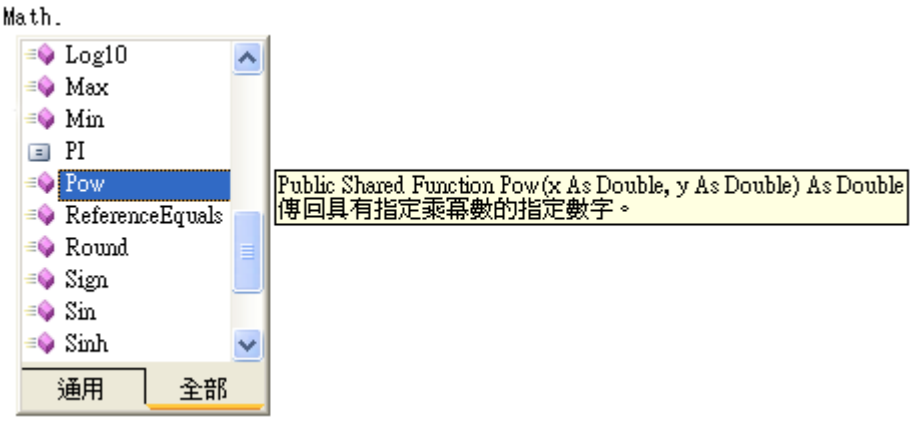
因為複製順序稍有不同，稍後請修改一下對應的程式碼即可。操作介面與程式碼如下：




```
private void button5_Click(object sender, EventArgs e)
{
    double w = double.Parse(textBox9.Text);
    double h = double.Parse(textBox10.Text);
    h /= 100; //公分轉為公尺單位
    double bmi = w / Math.Pow(h,2);
    label11.Text = bmi.ToString();
}
```

上面程式碼中居然有中文！請注意看它們的前面有兩個斜線字元『//』這是 C#語言中寫程式「註解」的方法，在軟體中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，那就在前面加兩條斜線變成註解就好了！

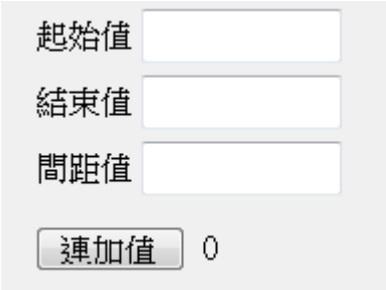
上述程式先取得身高體重的輸入值，再將身高(公分單位)除以 100 變成以公尺為單位，計算 BMI 值的時候使用 Math.Pow 函數計算身高的平方值。在同學打字到 Math 時應該會看到如下的自動提示選項，Math 在微軟的軟體工具架構中可以視為一個與數學相關的函式庫，裡面內建很多數學函數，包括下面畫面看到的最大(Max)與最小值(Min)、三角函數與對數(Log)，甚至數學常數如圓周率 PI(π)等等。最可貴的是當指標移到某一函數名稱時旁邊還會進一步出現中文化的提示！提示中會有函數的簡要功能說明，還有語法格式，你真的很難辯稱不會使用了！(除非數學課沒學過啦！)



上例中使用次方函數 Pow(h, 2)計算 h 的平方值，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！

1-5 用迴圈作連加法

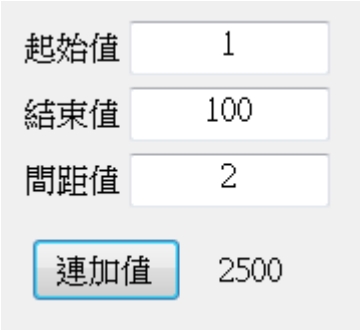
電腦常常用來處理重複性高的工作，尤其是大量的數值運算，這時就必須用到迴圈，在此我們以連加法的計算來介紹這個程式設計的重量級語法。首先請設計介面如下：



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 label16，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 void button6_Click(object sender, EventArgs e)
{
    double a = double.Parse(textBox11.Text);
    double b = double.Parse(textBox12.Text);
    double c = double.Parse(textBox13.Text);
    double d = 0;
    for (double i = a; i <= b; i += c)
    {
        d += i;
    }
    label16.Text = d.ToString();
}
```

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 d(初值為 0)。迴圈運算的基本語法就是 for 之後跟著一對小括號與大括號，小括號內以分號相隔，等於有三個小指令，第一個是迴圈起始值，的二個是終點值，第三個是間距值；大括號內則是迴圈必須重複的動作，在此就是以 i 為變數，持續將變數 i 加入 d。最終將答案 d 輸出顯示於 label16。下面是連加法的執行畫面之一：



在此例中，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 int，且間距值都是 1，所以常寫成類似這樣 (int i=0; i<n; i++)，表示 i 的結束值是 n 間距則是 1！

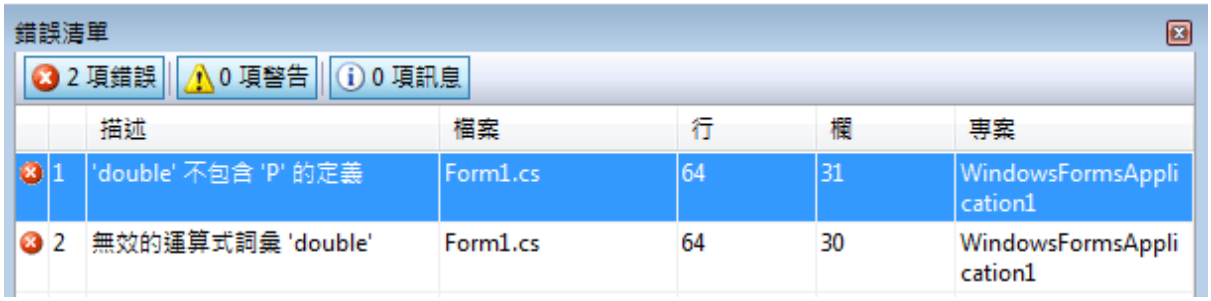
1-6 除錯基礎

有人說過寫程式所花的時間如果是 1 的話，除錯的時間至少是 3(倍)！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！所以程式設計老師的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！如下圖是使用者忘了打 Parse，出現波浪紋後將滑鼠移到錯誤地點還會有錯誤說明("無效的……")。

```
double a = double.Parse(textBox11.Text);
double b = double.Parse(textBox12.Text);

double c = double(textBox13.Text);
```

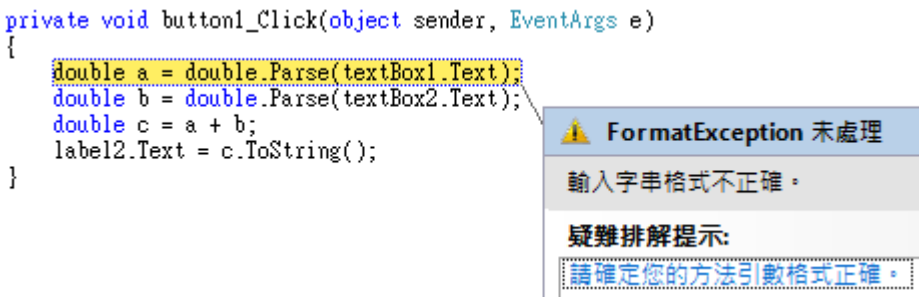
無效的運算式詞彙 'double'

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：它會告訴你錯誤的詳細情況，包括錯誤的檔案與行號。



但是請注意到 C#對於更正錯誤的反應不是非常自動，常常已經更正了錯誤，波浪線還是停在那邊！此時可以點選功能表的「建置」→「建置方案」功能(Ctrl+Shift+B)就是請軟體立即檢查目前程式碼的意思。

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，稱為「執行階段錯誤」，畫面就會變成錯誤行被黃色背景醒目底色所標示，你一樣可以發現很多錯誤說明。如下圖是故意在應該輸入數字的 textBox 中輸入文字的結果：



寫程式的除錯機制其實非常多，在此只介紹一小部分，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的提示說明，建議多看資料多問老師，慢慢習慣就好！最常見的錯誤說明與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 360 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 C#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 C 語言書籍，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！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！

最嚴謹的語法，最廣用的語言

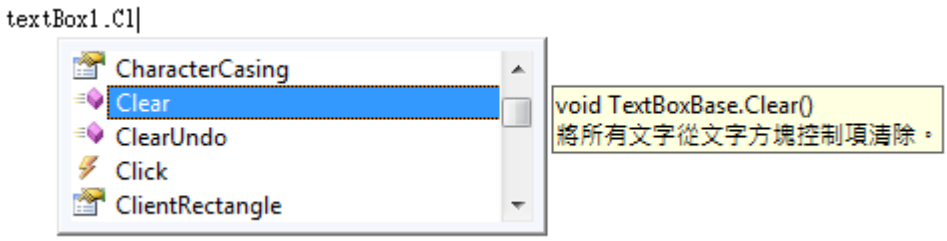
如果你是初次寫 C 語言程式應該會覺得它非常龜毛，一點點錯都不行！確實如此，傳統上 C 是專家級的語言，差不多所有複雜龐大的程式，像 Windows 作業系統，都是用 C 寫的。不像 VB 一開始就是讓初學玩家『玩』的語言！面對 VB 好像是國小老師在教你，面對 C 就像是嚴肅的大學教授在上課了！不過，如果你想在資訊領域生存，不會 C 語言可以說寸步難行！因為多數目前熱門使用的語言，如設計手機程式常用的 Java，設計網頁程式常用的 JavaScript、PHP，Flash 內部的 Action Script 以及很多遊戲引擎內的 Script 語言，其實都是 C 語言的簡化或進化版，如果你看不懂任何的 C 語言就很難找到工作了！相對的，好好學會任何一種 C 系列語言，其他的語言應該還沒學就看得懂七八成了！C#是目前最新與功能最完整的 C 語言版本，最新其實也表示輔助功能最好，是開始學 C 系列語言的最佳選擇！

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」！舉例來說，任何一部電腦都非常複雜，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桌上型電腦！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：盡量使用既有物件組裝完成你的程式。

在本單元中你應該已經從使用工具箱體驗到組裝物件的成就感，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！但是當然零件未必完全合用，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)成字串(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(".")以及字元會發現類似下圖的提示：

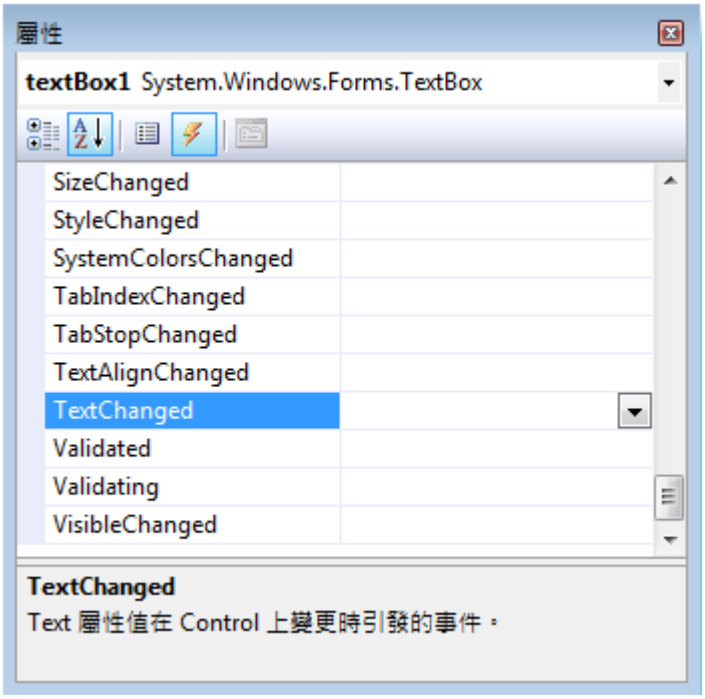


這些選項之前如果是桃紅色，像在飛行的塊狀圖示就是動態的「方法」，文件上方有隻手的圖示就是靜態的「屬性」。你會發現多數物件的方法與屬性都有幾十個之多！除此之外，多數物件還可以對一些使用者的操作或系統引發的「事件」(Event)有反應！圖示是一道金黃色閃電。最常見的例子就是本單元用到的按鍵(button)被滑鼠點到時可以產生 button.Click 事件！

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而定？就是使用者會製造出不同的「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畫面。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，常常將程式碼寫在不同物件的相同事件裡面！譬如應該寫在 button1_Click 事件的程式碼，因為按錯物件寫到 label1_Click 去了！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**！

另一方面，雖然雙擊指定的物件(如 button1)可以直接產生該物件的事件副程式的框架，但是這只能產生該物件「最常用」的**預設**副程式，而實際上一個物件通常可以回應很多種事件。建立事件副程式框架的標準程序應該是在設計頁面上先選擇物件，再到屬性視窗中選擇黃色閃電標記，如下圖，textBox1.的事件列表就出現了，你點選某一事件之後就可以產生事件副程式框架開始寫程式了！



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，標準物件通常都包含：「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！