

1. EASY VB

1.1 Visual Basic 簡介

1.2 Visual Basic 的安裝

1.3 進入 Visual Basic

1.4 使用 VB 計算圓的面積

1.5 與滑鼠有關的術語

1.1 Visual Basic 簡介

近年來許多軟體喜歡在名稱前面冠上 Visual 的字眼，而 Visual Basic 可說是其中的先驅，Visual 在字義上是『看的見的，視覺的...』的意思。在 Visual Basic 中增加了許多視覺化的工具以供程式設計師撰寫視窗程式，而且這些設計出來的視窗就跟最後執行時的外觀是相同的，彷彿『未來執行的結果在設計階段就看的見』一樣。

本書所使用 Visual Basic 版本為 6.0，而 Visual Basic 的新一代版本是『VB.NET』，但 VB.NET 與 VB 6.0 有相當大的差異，在此就不詳細列舉。

Visual Basic 6.0 目前有三個版本，皆分別有中、英文版本

1. **普及版**：取名叫普及版，希望大家都能夠買的起，此一版本含有三片光碟，售價大約在 2000 元左右，雖然它是 VB 裡面最低階的版本，但是此版本之程式設計相關功能已經相當齊全，足夠初學者做基礎的運用。
2. **專業版**：提供豐富的控制元件，可開發 ActiveX 元件、資料庫程式、ODBC 與主從式架構應用程式、Internet 網頁製作等，若有 VB 的程式設計經驗，可採購此一版本。
3. **企業版**：此為最高階版本，定價大約是專業版的數倍，相對的增加了遠端存取物件、分散式應用程式開發、視覺化模組工具、應用程式效能瀏覽器等功能，對於專業的軟體工作者來說，不妨考慮採購此一版本。

雖然 VB 有以上三種版本，但其操作方式皆相同，無論您使用那一個版本，都不會影響你對學習 VB 的熱誠。

1.2 Visual Basic 的安裝

安裝 VB6.0 之前，請先確定您的電腦符合以下的軟硬體需求。

硬體需求

CPU：80486 以上(建議使用 Pentium II 以上)。

螢幕：彩色螢幕。

顯示卡：支援彩色之顯示卡

硬碟空間：安裝完整的 Visual Basic 普及版至少 60MB，完整版及企業版則需要 250MB 以上。若同時安裝 MSDN Library 則需要額外再多出 1.1G 硬碟空間。

主記憶體：16MB 以上(建議使用 64MB 以上)

其他：滑鼠、CD-ROM。

軟體需求

Microsoft Windows 95、98、ME 中文版；Windows NT 4.0、5.0 中文版；Windows 2000、XP、2003 中文版、Vista 中文版。

1.將 VB 光碟放入光碟機中，稍候片刻即會出現如下視窗，請點選『下一步』



2.出現版權宣告，點選『我接受此合約』，並點選『下一步』



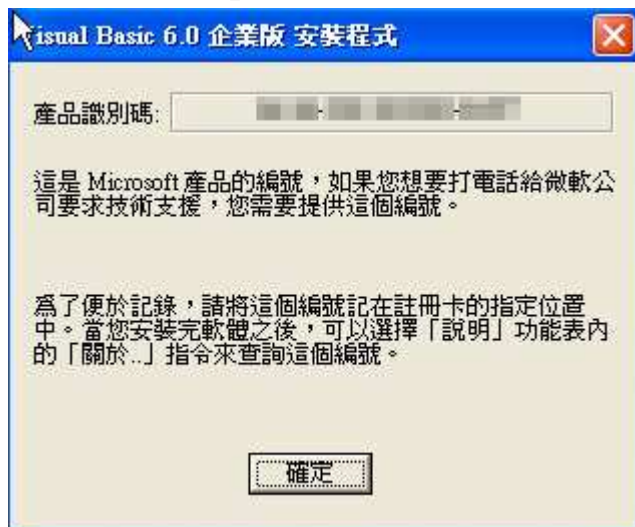
3.輸入產品識別碼，通常在光碟外盒可以找到此一識別碼，並點選『下一步』



4. 點選『繼續』，以進行安裝



5. 請點選『確定』，以繼續安裝



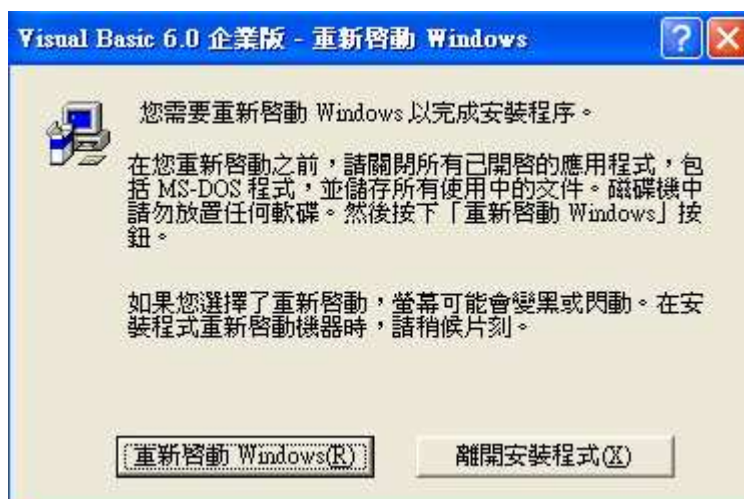
6.在此可以設定安裝元件以及自訂安裝目錄，初學者建議點選『一般安裝』



7. 等待程式安裝至 100%



8. 點選『重新啓動 windows』，此時將會立刻重開機



9. 重開機後會出現如下畫面，在此先不安裝 MSDN，請將『安裝 MSDN』取消，並點選『下一步』。



10. 若不安裝 MSDN 則會出現如下的警告視窗，點選『是』以繼續安裝



11. 請點選『下一步』，繼續安裝

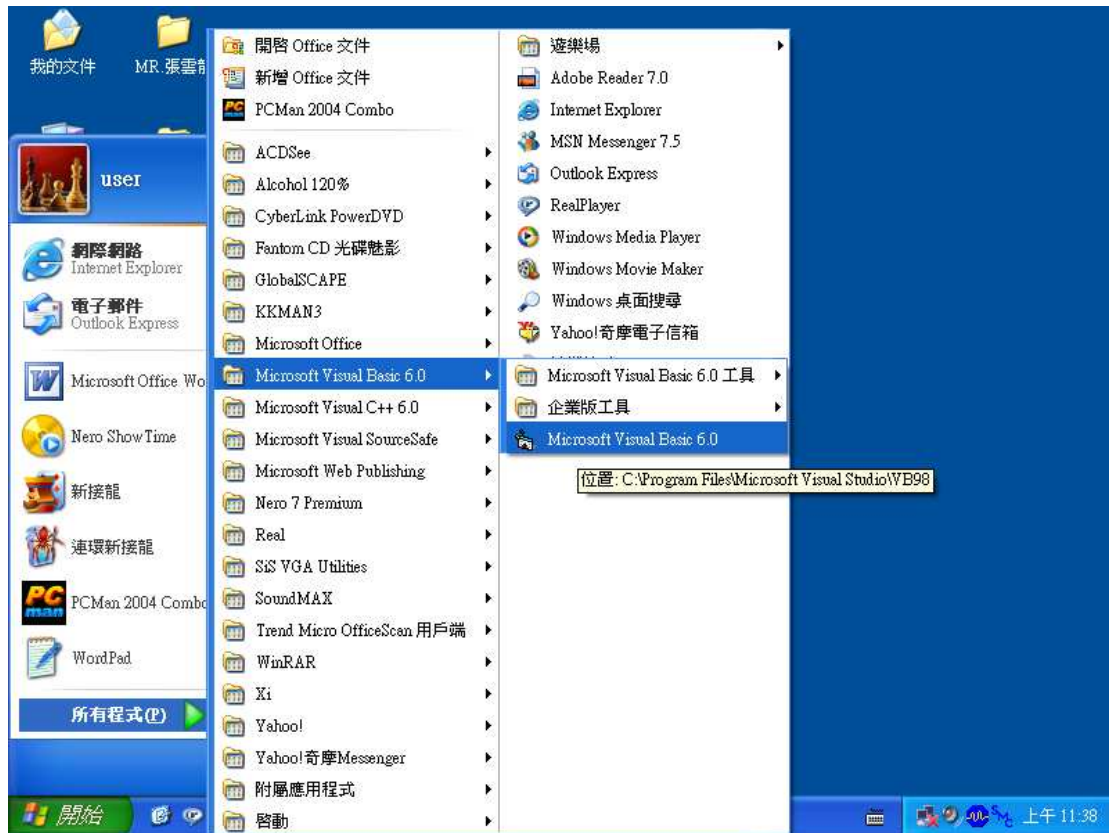


12.請將『立即註冊』取消，稍候再自行註冊，並點選『完成』，恭喜您，VB 已安裝完成。



1.3 進入 Visual Basic

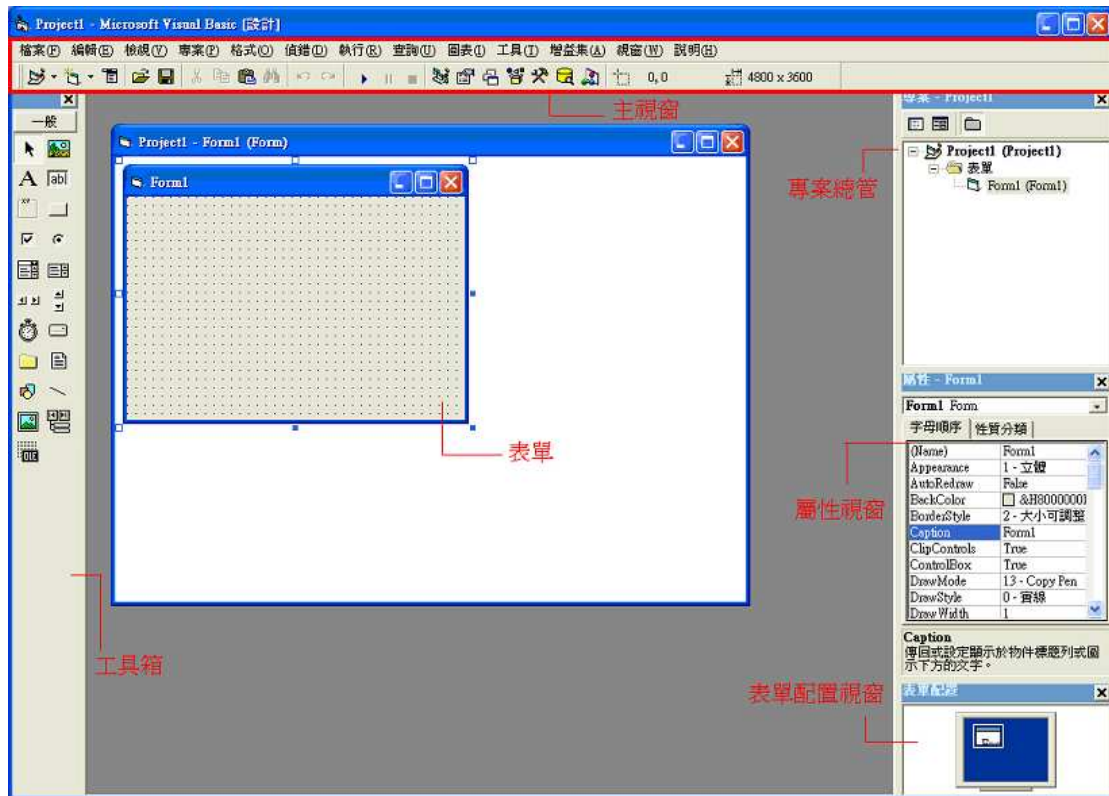
1. 程式集 → Microsoft Visual Basic 6.0 → Microsoft Visual Basic 6.0



2. 建立 → 標準執行檔 → 開啓



3. 接下來我們可以看見 VB 的整合開發環境

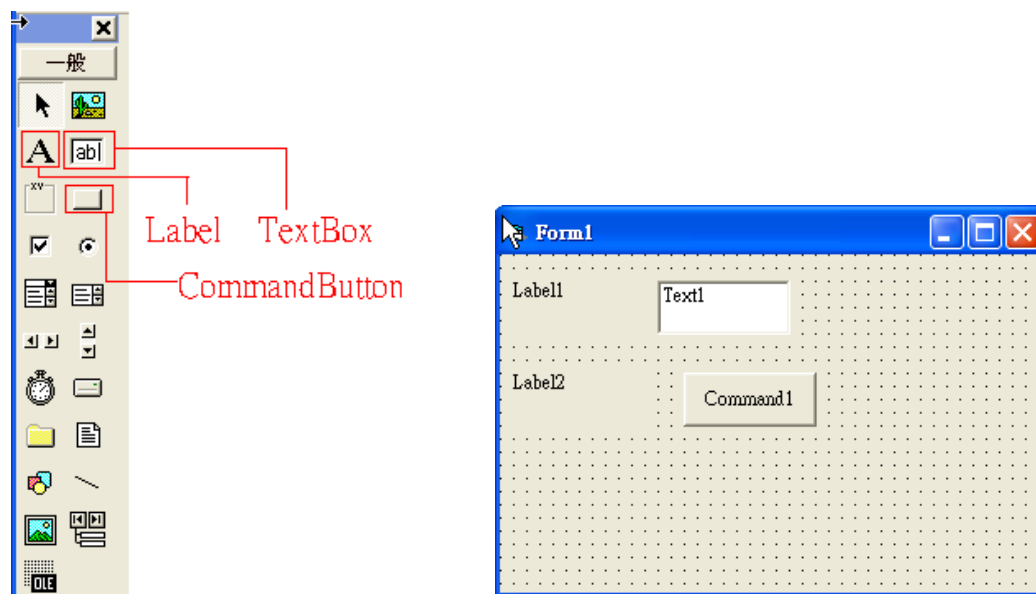


視窗	用途說明
主視窗	提供完整的功能表列
工具箱	提供各種控制項供程式設計師使用
表單	資料的輸出入以及控制項物件都放在這裡
屬性視窗	設定控制項物件的屬性值
專案總管	管理專案、表單、模組等相關檔案
表單配置視窗	決定表單執行時在桌面上的位置

1.4 使用 VB 計算圓的面積

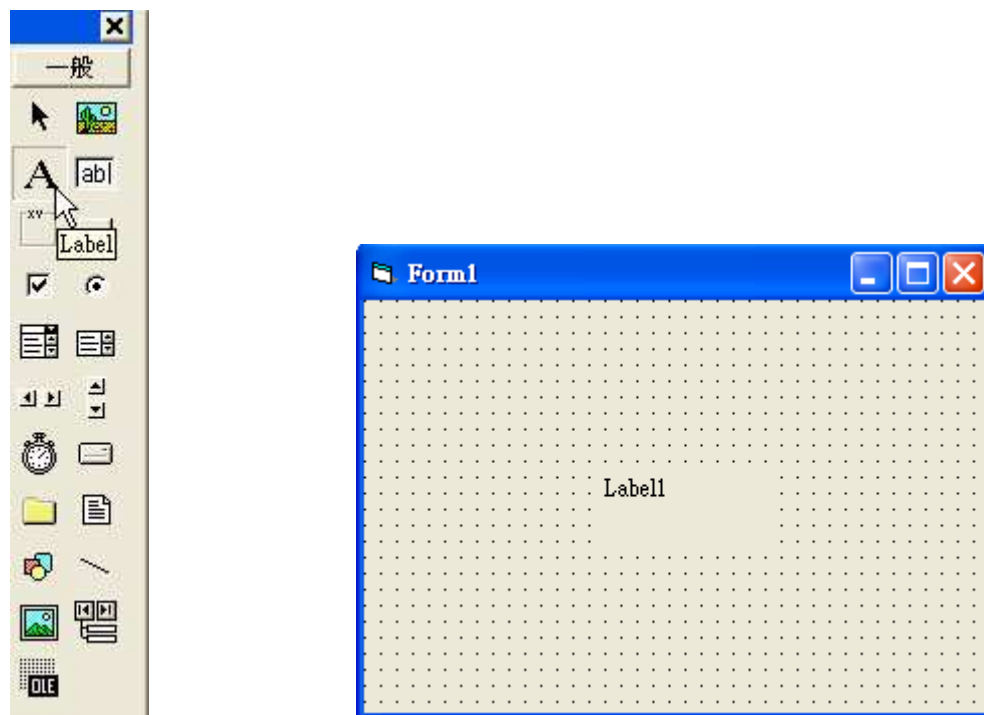
1.4.1 表單及控制配置

首先我們需要 2 個 Label(標籤)、1 個 textBox (文字方塊)以及 1 個 commandButton(命令按鈕)

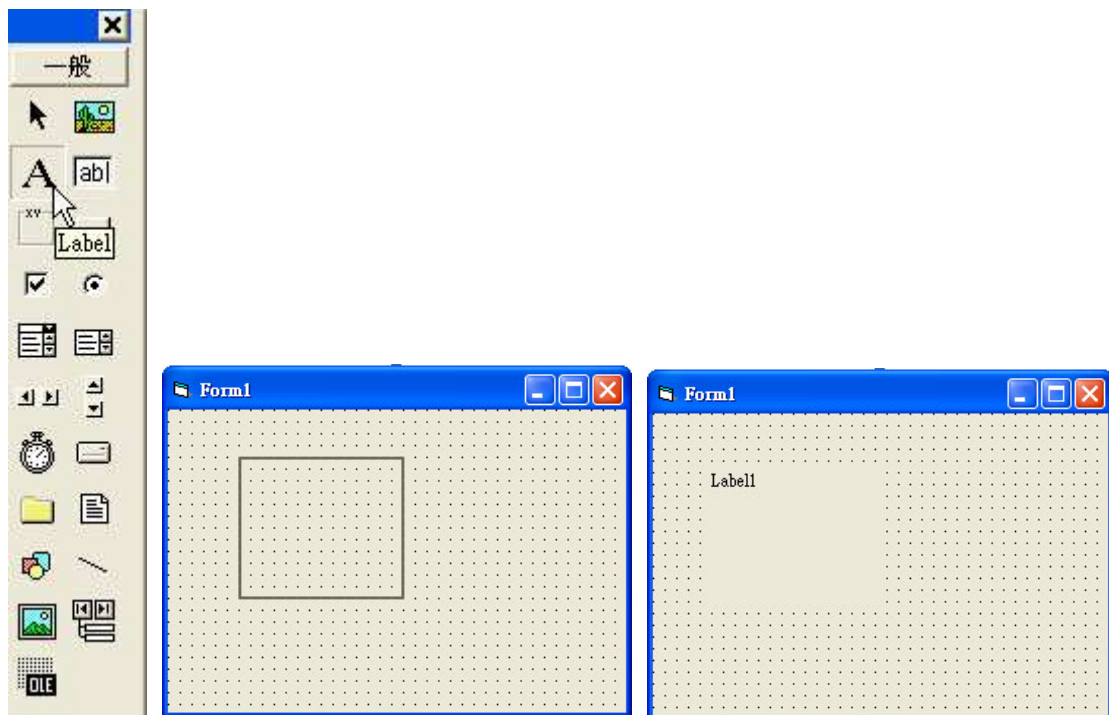


點選控制項的方法有二種，如下所示

1. 用滑鼠左鍵在工具箱中所需之控制項連續點二下，表單上即會出現一個所點選之控制項。

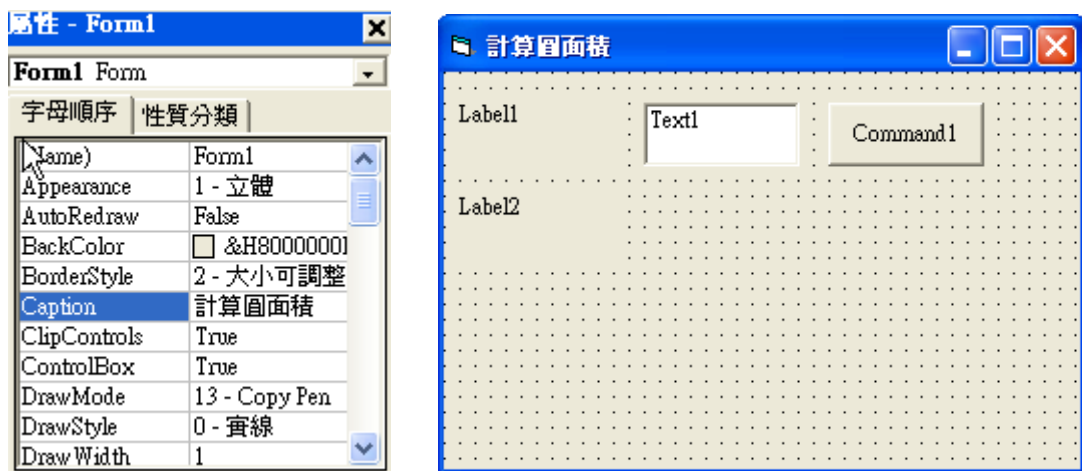


2. 用滑鼠左鍵點一下工具箱中所需之控制項，接下來在表單上拉出所需之大小，然後鬆開滑鼠左鍵，控制項就跑出來了。

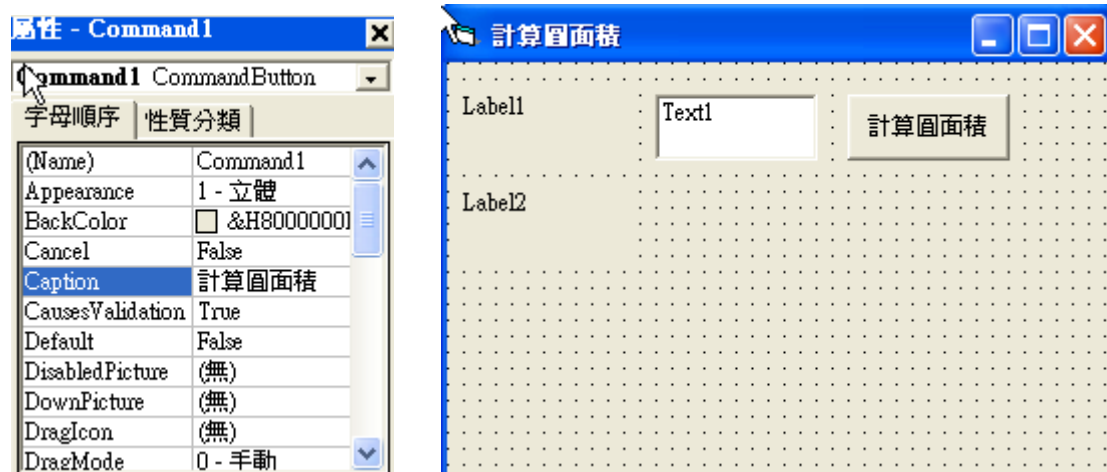


接下來，我們必須在屬性視窗中變更一些設定以符合我們的需求，如下步驟

- 1.修改 Form1 的 Caption 屬性為『計算圓面積』



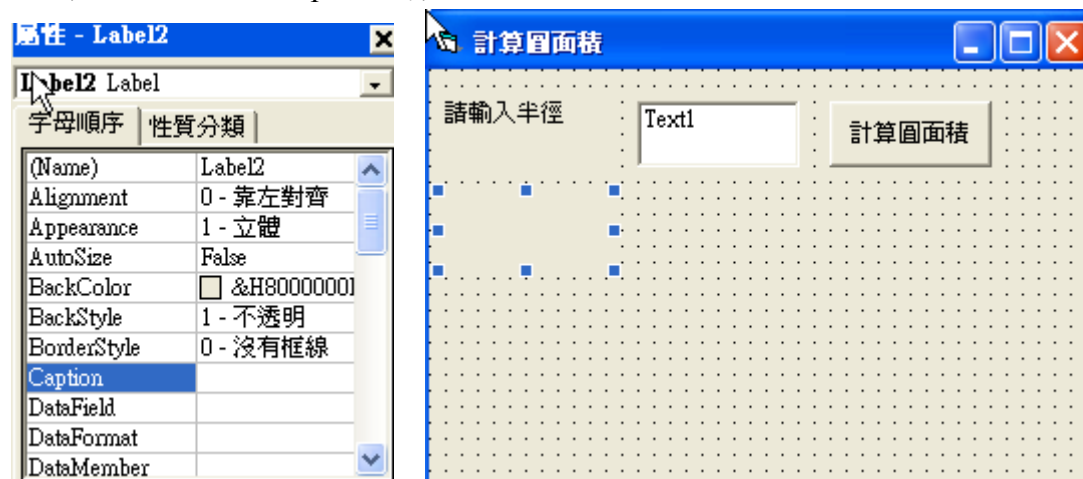
2.修改 Command1 的 Caption 屬性為『計算圓面積』



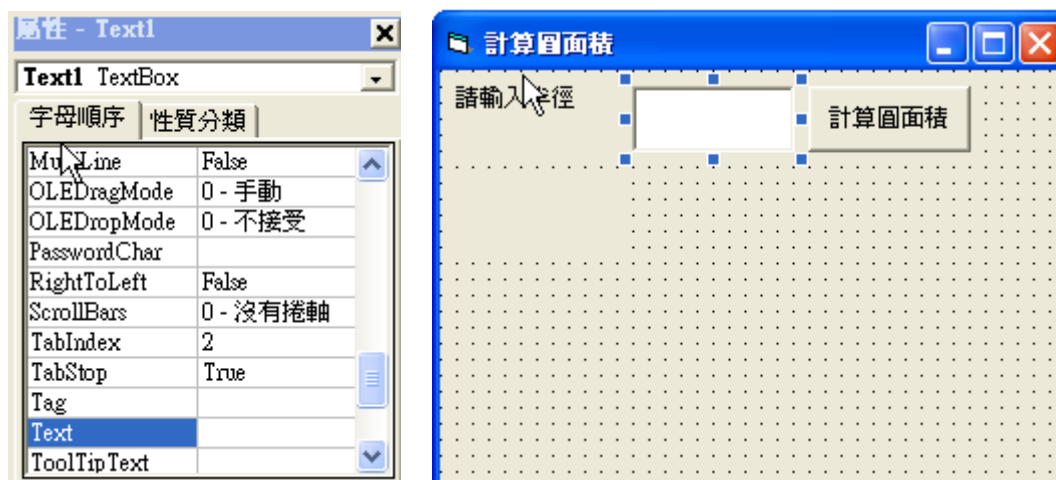
3.修改 Label1 的 Caption 屬性為『請輸入半徑』



4.修改 Label2 的 Caption 屬性為空白



5. 修改 Text1 之 Text 屬性為空白



1.4.2 程式碼之撰寫

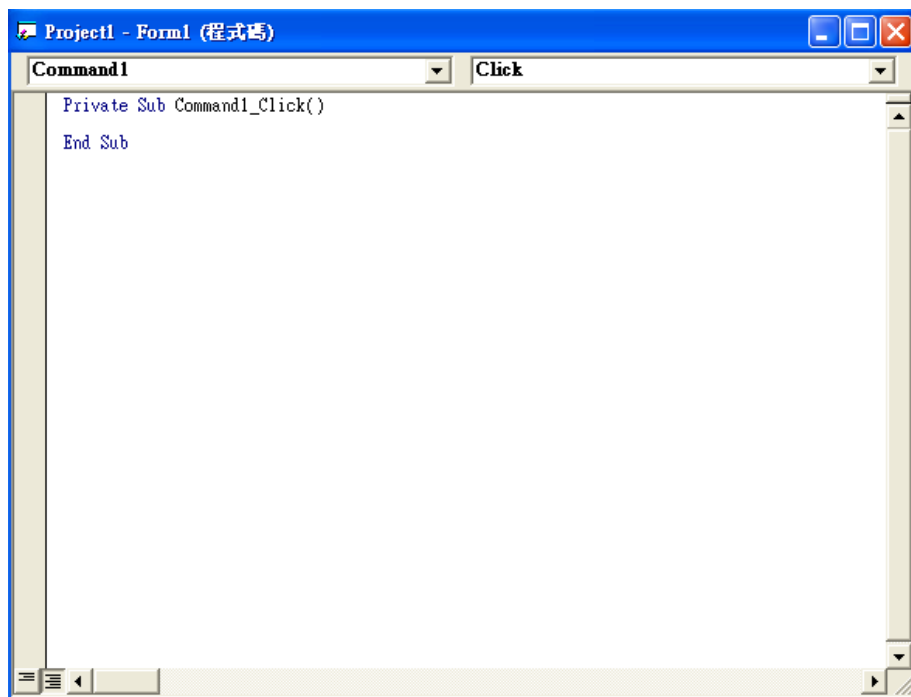
事件驅動觀念的精髓就在於『你要用什麼控制項物件啟動程式執行，就要把程式碼寫在該控制項中』，本範例中，我們想要在點選 command1 後便自動計算 Text1 中所輸入的半徑所圍成之圓面積，並且輸出在 Label2 上，程式的啟動點就在於 command1，所以我們必須將相關之程式碼寫在 command1 中，如下方法

1. 在 command1 上用滑鼠左鍵連續點二下

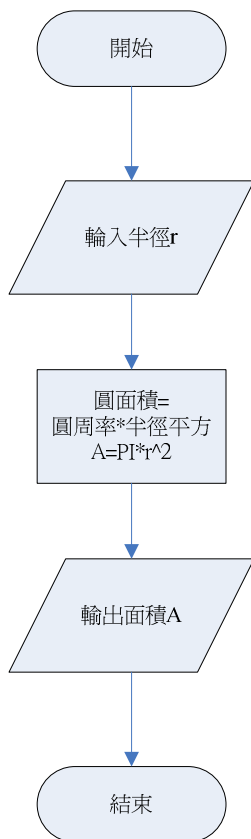


3. 即出現程式碼編輯視窗，程式碼即寫在 Private Sub Command1_Click() 與

End sub 之間的空白處。



4.分析程式流程



5..撰寫如下程式碼，在『 ‘ 』後方為註解

```
Private Sub Command1_Click()
```

```
Const PI = 3.14 '常數宣告
```

```
Dim r As Integer '變數宣告, r 代表半徑
```

```
Dim A As Single '變數宣告, A 代表面積
```

```
r = Val(Text1.Text) '取得半徑
```

'vb 預設輸入的格式為"字串"，val() 函數會將字串轉為數值

```
A = PI * r ^ 2 '圓面積= 圓周率 * 半徑的平方
```

```
Label2.Caption = A 'Label2 顯示變數 A(面積)的值
```

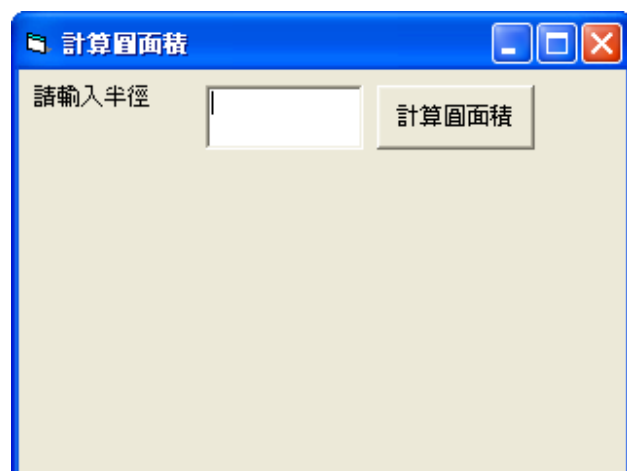
```
End Sub
```

1.4.3 執行結果

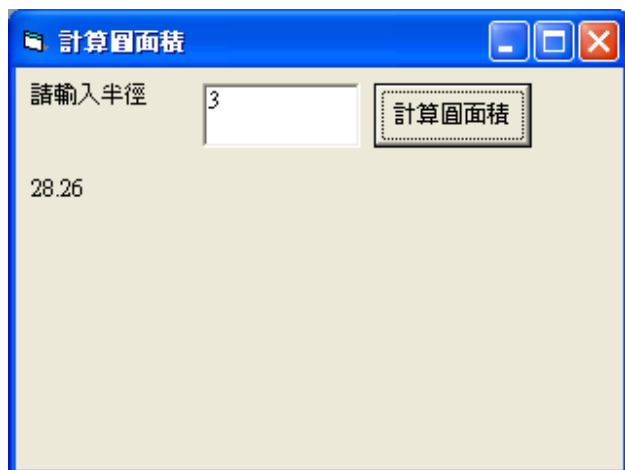
1.按下功能列上的  以執行我們所編輯的程式




2.程式執行階段，果然和表單設計時一模一樣，並且等待使用者輸入半徑



3. 輸入任一個數值，並按下『計算圓面積』



4. 按下右上角的 ，可以關掉程式，回到設計階段。

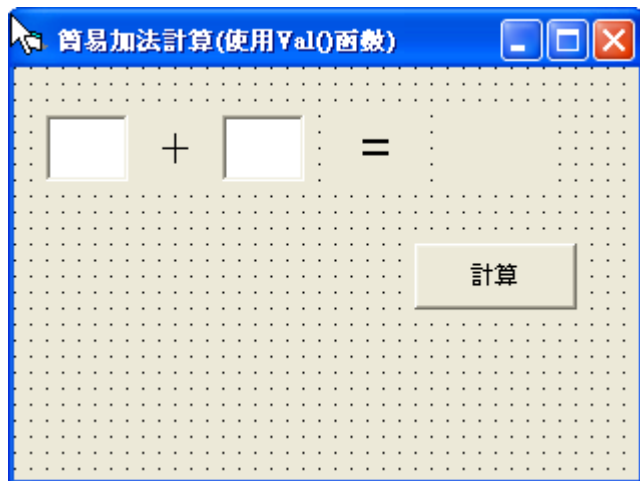


小幫手特區：Val() 一定要加嗎？

以下將舉 2 個數字相加的程式範例來說明 Val() 函數的重要性

1. 使用 Val() 函數

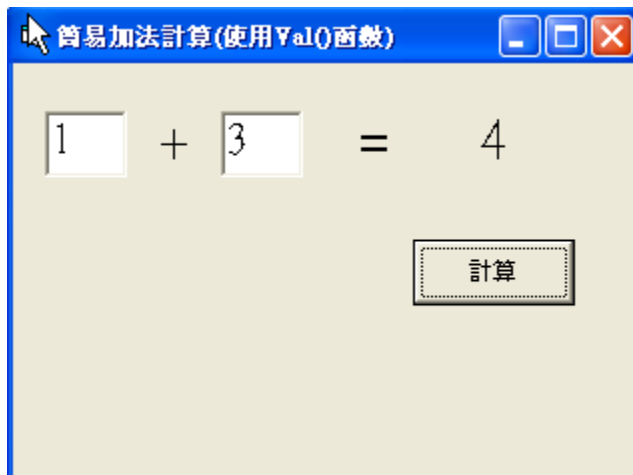
(1) 將表單編排如下



(2) 在 command1(計算)中編輯如下程式碼

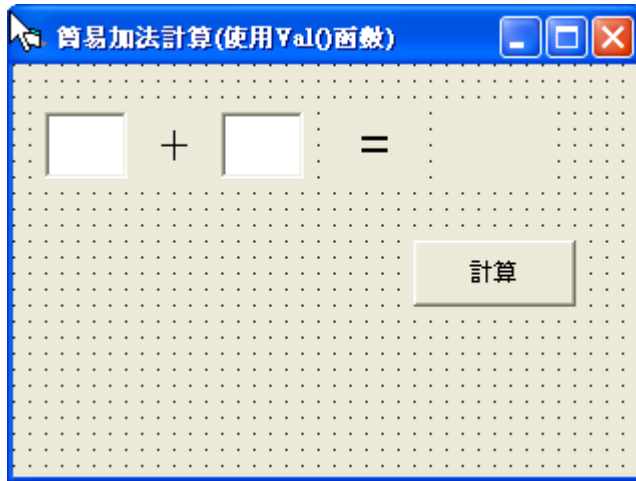
```
Private Sub Command1_Click()  
  
Dim a, b, c As Integer '宣告所需變數  
  
a = Val(Text1.Text) 'a 代表被加數  
b = Val(Text2.Text) 'b 代表加數  
c = a + b 'c 代表 a、b 之和  
  
Label3.Caption = c '將 c 顯示在 Label3 上  
  
End Sub
```

(3) 執行結果如下圖



2. 不使用 Val() 函數

(1) 表單編排如下



(2)編輯以下程式碼

```
Private Sub Command1_Click()
```

```
Dim a, b, Sum As Integer
```

```
a = Text1.Text
```

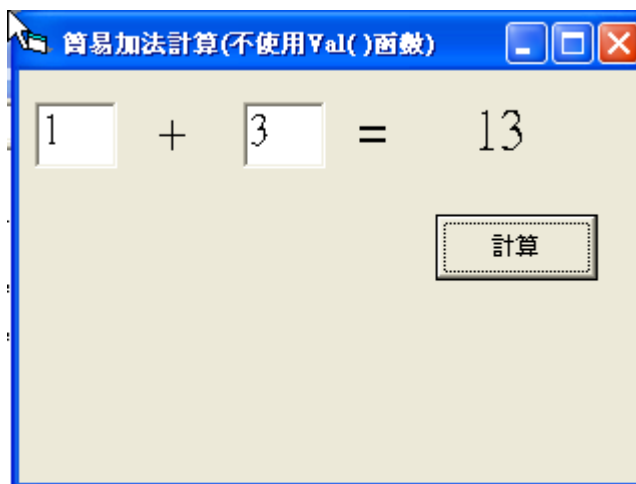
```
b = Text2.Text
```

```
Sum = a + b
```

```
Label3.Caption = Sum
```

```
End Sub
```

(3)執行結果



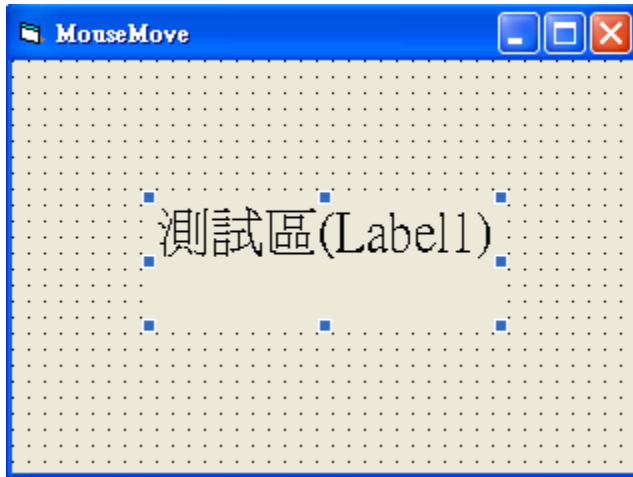
※爲什麼會造成這麼大的不同，原因在於 VB 無論是字串相加或數值相加，皆使用

『+』所致，字串『1』與字串『3』相加，就是『13』，但若為數值之『1』與『3』相加則應為『4』，所以在 VB 中若遇到相加的二數時，要注意其型別是否為『數值』，否則將會發生無可預知的錯誤。

1.5 與滑鼠有關的術語

1.move(移到)：例如將滑鼠移到 Label1 上，即會觸發 Label1_MouseMove 中的程式碼，如下範例中，我們讓滑鼠移到 Label 上，Label1 的 Caption 屬性將變為『MouseMove 事件已觸發』

(1) 表單配置如下



(2) 撰寫程式碼於 Label1_MouseMove 中

```
Private Sub Label1_MouseMove(Button As Integer, Shift As Integer, X As Single,  
Y As Single)
```

```
Label1.Caption = "MouseMove 事件已被觸發"
```

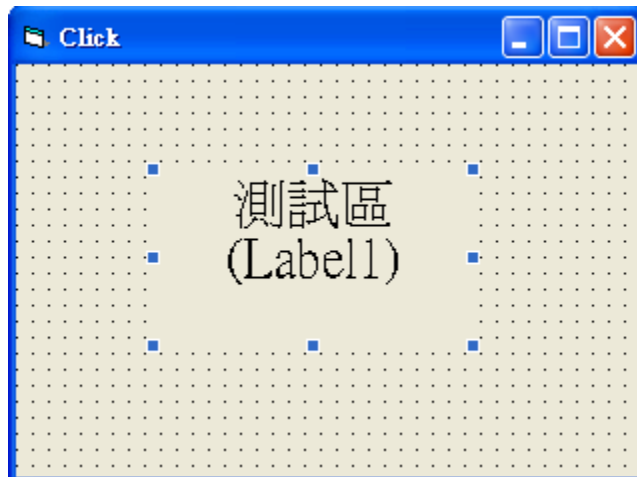
```
End Sub
```

(3) 執行結果



2.Click(按一下): 例如用滑鼠左鍵點一下 Label1, 即會觸動 Label1_Click()中的程式碼, 如下範例我們在 Label1 上點一下滑鼠, Label1 的 Caption 屬性將變為『Click 事件已觸發』

(1) 表單配置如下



(2) 撰寫以下程式碼於 Label1_Click 中

```
Private Sub Label1_Click()  
  
Label1.Caption = "Click 事件已被觸發"  
  
End Sub
```

(3) 執行結果



2. **Double Click(連續點二下左鍵)**: 例如在 Label1 上連續點二下才會產生動作，如下範例所示

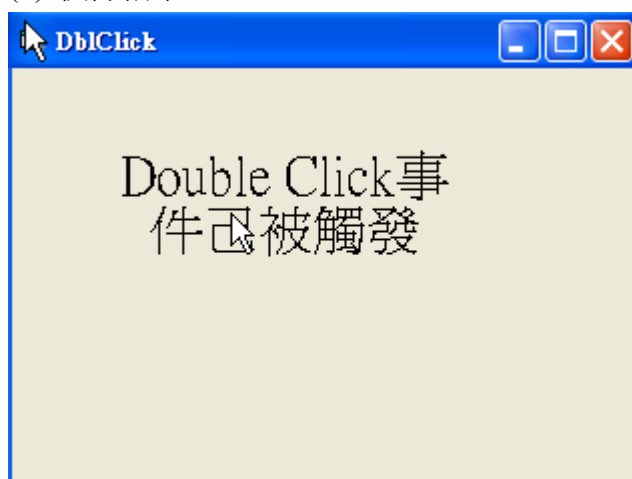
(1) 表單配置



(2) 撰下如下程式碼於 Label1_DblClick 中

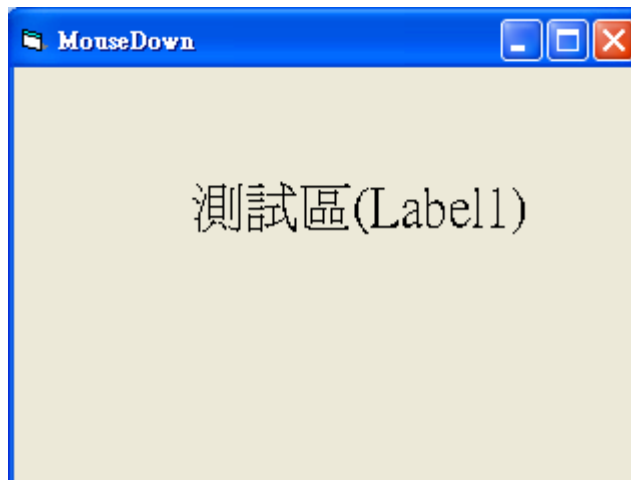
```
Private Sub Label1_DblClick()  
  
Label1.Caption = "Double Click 事件已被觸發"  
  
End Sub
```

(3) 執行結果



3. **MouseDown(按下滑鼠)**:無論是按下滑鼠右鍵或左鍵,均會觸發 MouseDown。

(1) 表單配置



(2) 撰寫以下的程式碼在 Label1_MouseDown 中

```
Private Sub Label1_MouseDown(Button As Integer, Shift As Integer, X As  
Single, Y As Single)
```

```
Label1.Caption = "MouseDown 已觸發"
```

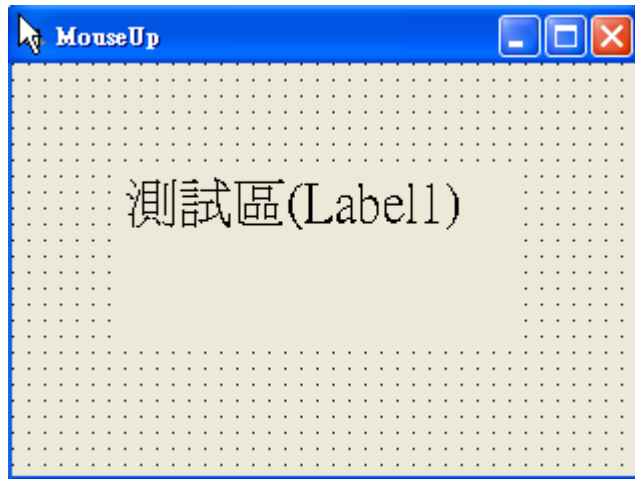
```
End Sub
```

(3) 執行結果



4. **MouseUp(放開滑鼠鍵)**：無論是左鍵或右鍵按下之後再放開，即使產生 MouseUp 事件，例如在本範例中配合 MouseDown 來示範 MouseUp 的功能。

(1) 表單配置



(2) 撰寫以下程式碼

```
=====
Private Sub Label1_MouseDown(Button As Integer, Shift As Integer, X As Single,
Y As Single)

Label1.Caption = "滑鼠鍵已按下"

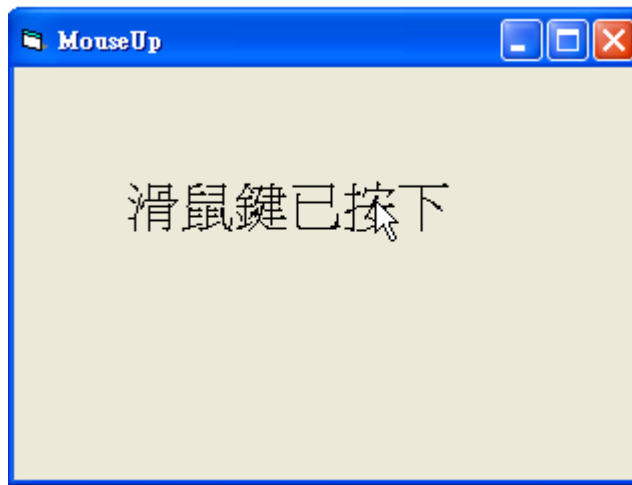
End Sub
=====
=====
Private Sub Label1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y
As Single)

Label1.Caption = "滑鼠鍵已放開"

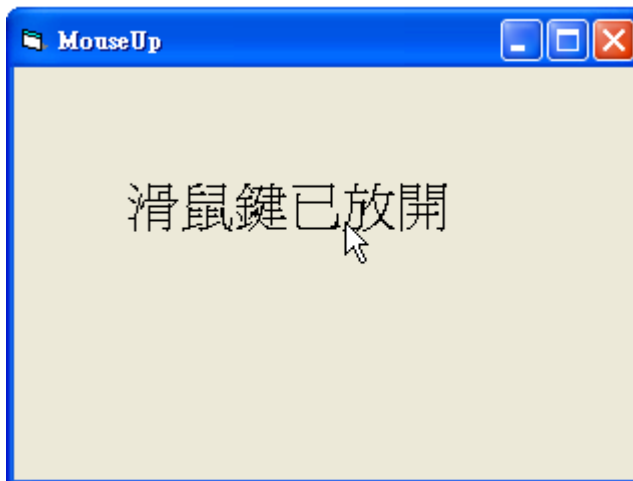
End Sub
=====
```

(3) 執行結果

1. 按下任一滑鼠鍵時



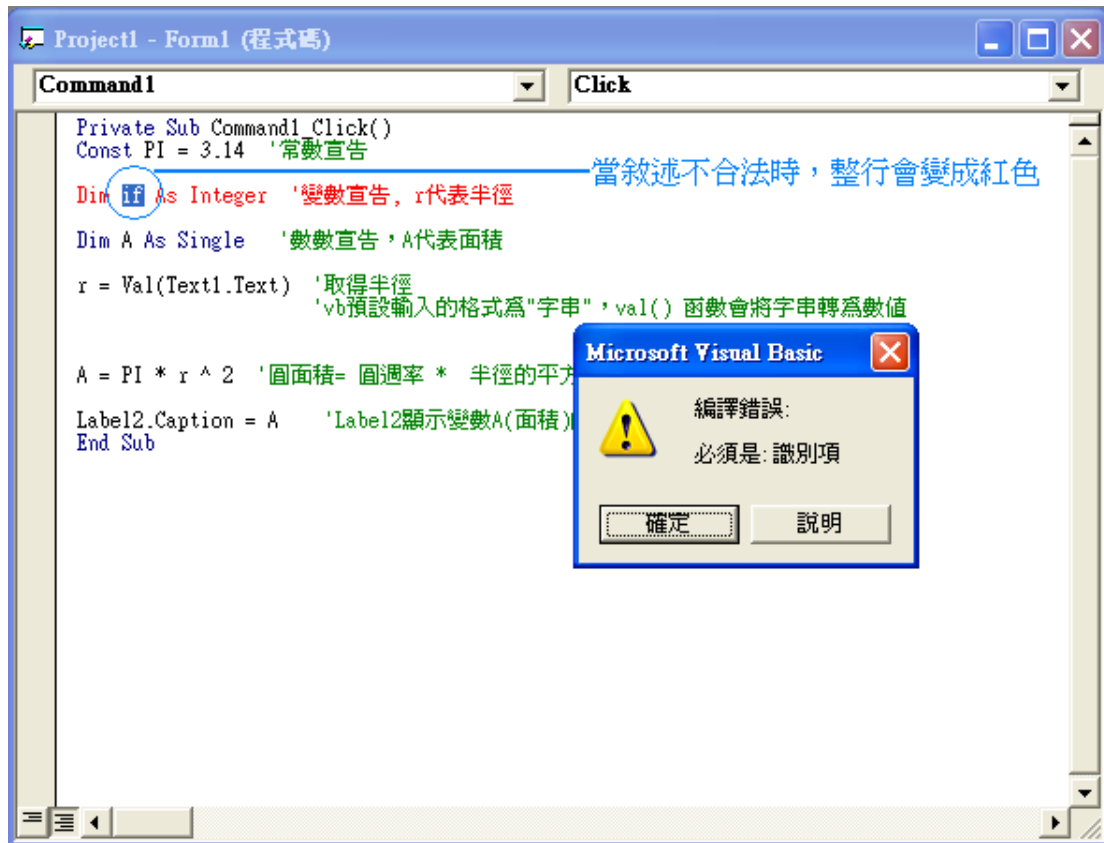
2. 放開滑鼠鍵時





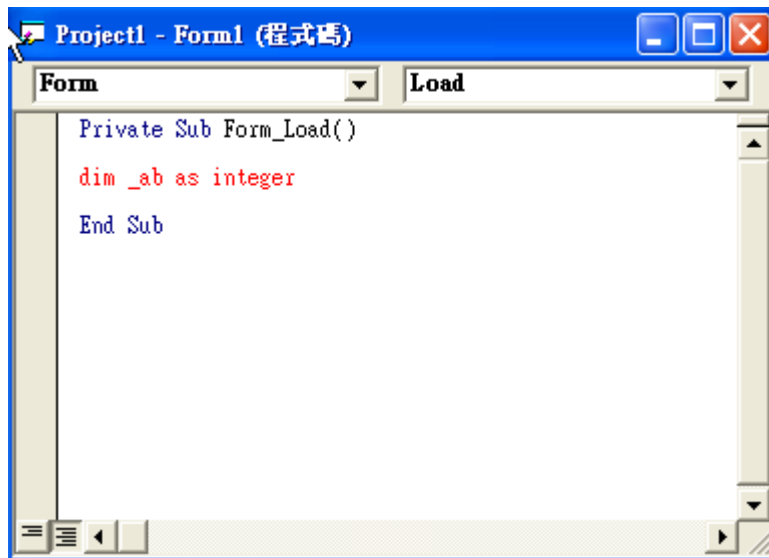
小幫手特區：Dim 是什麼、變數的命名有什麼限制？

1. Dim 是一個保留字，在 vb 中有許多的保留字，常見的例如 If、For 等等，是用來做為命令，不能當作宣告變數時所用的名稱。以下我們將舉圓面積的例子，將其變數名稱改為『保留字』時，會有什麼樣的情況發生

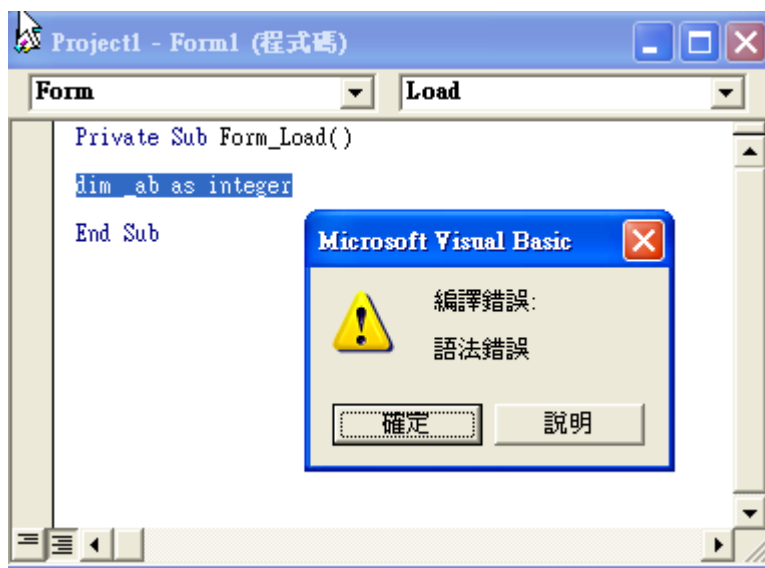


※讀者不妨將本警告訊息記在腦中，當下次出現本訊息時，即可輕易發現錯誤

2. 變數的命名只能使用大小寫的英文及底線『_』組成，唯一較嚴格的地方就是變數名稱的第一個字不能使用底線『_』，當然更不能使用上述所提到的保留字。如下圖所示，如果變數命名不合法，該行敘述也會變紅，執行時也會出現警告訊息。



```
Private Sub Form_Load()  
    dim _ab as integer  
End Sub
```



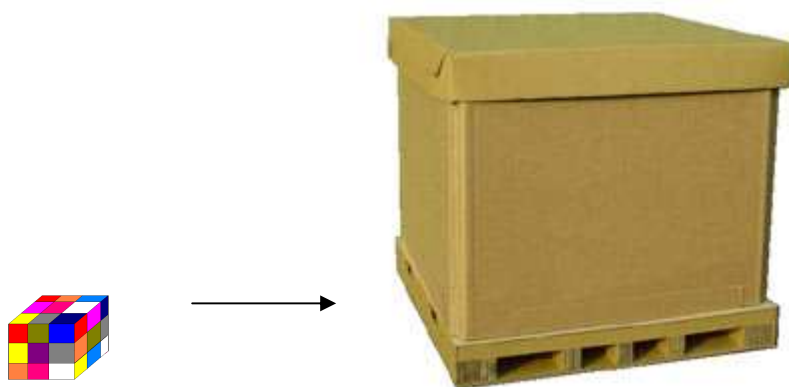
※這樣各位讀者就知道宣告的好處果然是『利多』，不但可以及早發現錯誤，也可免去在未來修改程式時的困擾。

3.變數的資料型態

變數的資料型態代表著變數所佔用的記憶體空間，就好像用可以容納冰箱的紙箱來裝一個魔術方塊，是不是很浪費呢~!在電腦中，一個變數只能同時儲存一組數據，就好比大紙箱放了魔術方塊之後，就不被允許放其他東西。

若是用只能裝的下魔術方塊的盒子來電冰箱，想必盒子一定會壞掉，而這種情形應用在變數上稱為『溢位』，在第三章會有詳細的介紹，以下是在 VB 中，各種型別所佔用的記憶體大小以及容量。

資料型態	佔用記憶體空間	有效範圍
Byte(位元)	1Byte	0~255
Integer(整數)	2Bytes	-32768~32767
Long(長整數)	4Bytes	-2147483648~2147483647
Single(單精度浮點數)	4Bytes	-3.402823E+38~-1.401298E-45 +1.401298E-45~+3.402823E+38
Double(雙精度浮點數)	8Bytes	-1.79769313486231E+308~ -4.94065645841247E-324 +4.94065645841247E-324~ ++1.7969313486231E+308
String(固定長度字串)	視字串長度而定	視字串長度而定
Boolean(布林)	2Bytes	True(成立)或 False(不成立)
Date(日期)	8Bytes	January 1, 100 ~ December 31, 9999
Object	4Bytes	可引用任何一個 Object



※ 如上圖，若使用不當的變數型態，可能造成額外的浪費，也會使電腦中可用的記憶體大大減少。