

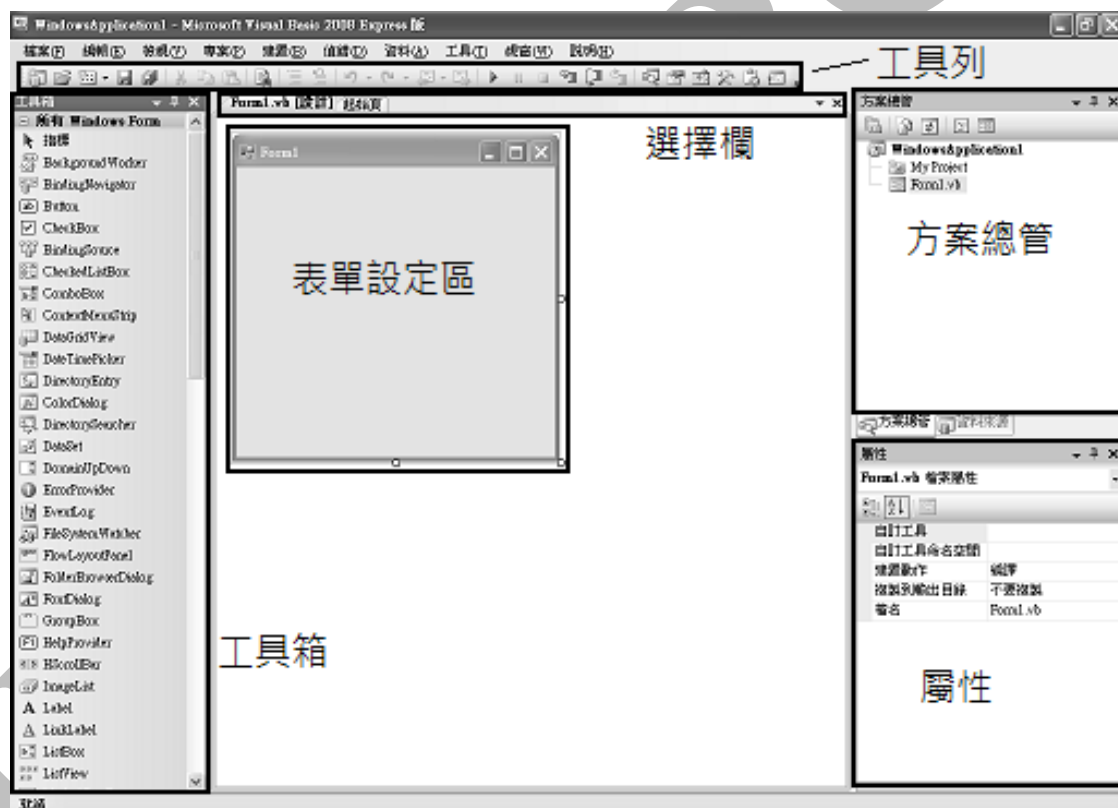
Visual Basic 2008

前言

一般人對於程式語言的印象，不外乎是在一個黑黑的 DOS 介面下，打著一些晦澀難懂的東西。其實，這些都是早期的程式設計才會有的情況，經過程式設計師們的改良與更新，現在的程式設計擁有人性化的介面、簡明易懂的語法。像現在要介紹的 Visual Basic（簡稱 VB），就是一個非常適合做為入門的程式語言，所以，讓我們一起來好好探討程式設計的奧妙吧！

準備工作

開啟 VB 後，點選工具列最左邊的新增專案圖示，選擇第一項「Windows Form 應用程式」，就會看到如以下的介面。



表單：呈現在使用者面前的視窗，可供使用者操作

選擇欄：切換不同的表單以及程式碼

工具箱：裡面有許多物件可拖曳到表單上，如按鈕、文字方塊…等

屬性：每個物件都有許多屬性，如字型、名稱…等

方案總管：一個程式可能包含一個以上的表單，可利用此來協助管理

四大天王

VB 中有所謂四大天王：物件、屬性、事件、方法。我們接下來要利用這四大天王，以及各式各樣的程式碼，寫出只屬於我們自己獨特的程式！

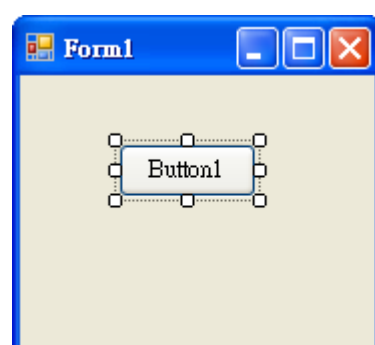
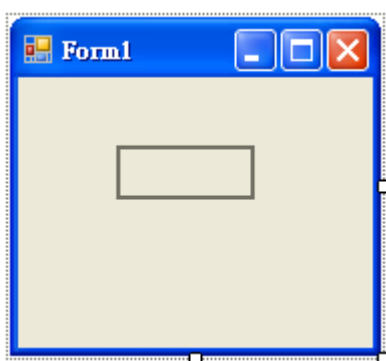
物件（由程式碼構成）

在日常生活中，任何實體的物品皆可視為物件，例如：車子、桌子、手表…等，在 VB 中也有類似的概念，程式執行時所見之按鈕、文字方塊…等皆可視為物件。早期的程式設計中並沒有物件這個選項，但隨著程設的演進，程式設計師創造了物件，使得程式設計更為方便，常用的物件如下：

圖示	物件名稱	功用	常用屬性	常用事件
	表單 (Form)	在程式執行時所看到的視窗，可以放入按鈕、標籤等物件	BackColor Text	Load
	按鈕	按下去後可以執行按鈕裡面的程式碼	BackColor Text	Click
	標籤	顯示文字	Font Text	Click
	文字方塊	讓使用者輸入及輸出文字	Font Text	TextChanged
	計時元件	定時間內做一次處理	Enabled Interval	Tick
	選項按鈕	單一選擇	Text Checked	Click
	核取方塊	多重選擇	Text Checked	Click
	群組容器	對控制項元件進行分類	Text	
	圖片控制項	顯示圖片	ImageLocation	

新增一個物件

想要新增一個物件，只要點選工具箱中的物件，在表單上拉出你想要的大小即可。



屬性（描述物件的特性）

在生活中也有所謂屬性的概念，例如：車子的顏色、手錶的大小…等，在 VB 中，格式的設定為「物件名稱.物件屬性=屬性值」。其中，「.」可以解釋成「的」，「=」則代表「指定成」，也就是把左邊的屬性變成右邊的屬性值，也可以解釋成把右邊的屬性值「指定給」左邊。例如 `Label1.Text="1252"` 就是說把一個標籤（Label1）的文字（Text）指定成「1252」，常見的屬性如下：

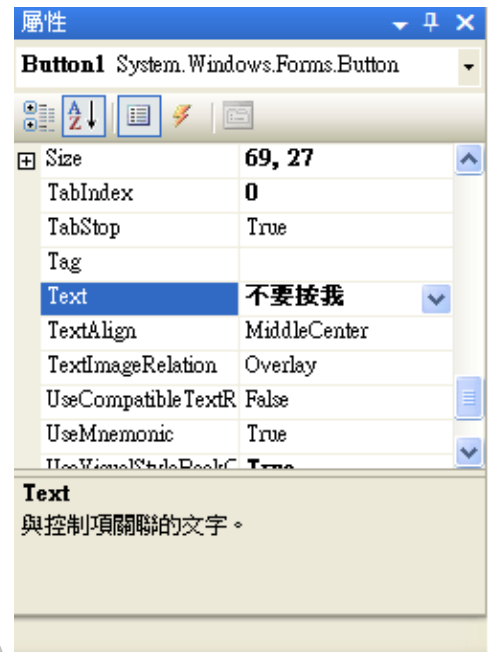
屬性	屬性說明	適用物件
Name	物件在程式碼中的名稱	所有物件
Text	物件的標題或是顯示文字	Form、Label、Button、TextBox
BackColor	物件的背景顏色	Form、Label、Button、TextBox
ForeColor	物件的前景(文字)顏色	Label、Button、TextBox
Enabled	物件是否可以被使用	Form、Label、Button、TextBox、Timer、PictureBox
Font	物件上文字的相關設定	Label、Button、TextBox
Size	物件的大小，以像素為單位	Form、Label、Button、TextBox、PictureBox
Location	物件的座標，以表單左上角為原點，以像素為單位	Label、Button、TextBox、PictureBox
Visible	物件是否可被使用者看見	Form、Label、Button、TextBox、PictureBox
Image	物件的圖片	PictureBox
Interval	物件 Timer 每幾毫秒 Tick 一次	Timer
Tag	物件的註解、說明	所有物件

修改屬性

先點選要修改屬性的物件，然後在右下角的屬性欄中選擇要修改的屬性。

若要在程式執行中修改屬性，可以直接從程式碼中修改。

```
Button1.Text = "不要按我"
```



事件（對物件所做的動作→被動性）

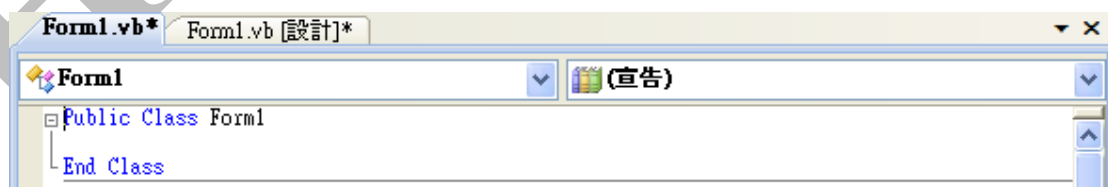
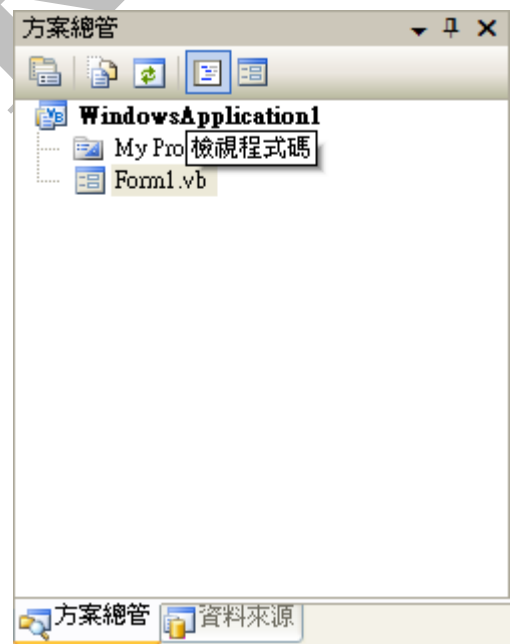
不同於其他命令列程式語言如 C，視窗化的程式是以事件方式驅動，如按滑鼠一下，鍵盤輸入、表單載入…等與應用程式進行互動，造成事件的觸發；在 VB 中格式的設定為「控制項名稱.控制項的事件」；例如 Button1.Click 則表示物件 Button1 偵測到滑鼠按下左鍵 Click 時會觸發事件，進而處理程序中的動作，常見的事件如下：

事件名稱	事件發生時機	事件名稱	事件發生時機
Load	程式載入時	KeyPress	鍵盤傳送字元給系統時
Click	物件被滑鼠按一下時	KeyUp	鍵盤的鍵放開時
DoubleClick	物件被滑鼠按兩下時	KeyDown	鍵盤的鍵按下時
MouseMove	滑鼠移動時	Deactivate	（表單）失去焦點時
MouseUp	滑鼠放開時	TextChanged	物件的 Text 屬性改變時
MouseDown	滑鼠按下時	Move	物件移動時

新增事件

要新增事件，必須先進入程式碼的視窗。點擊方案總管裡的「檢視程式碼」圖示，就可以看到程式碼了。

我們要編輯的程式碼必須擺在「Public Class Form1」跟「End Class」中間，也就是擺在 Form1 的程式碼裡面，否則程式可能會出錯。



接下來的工作就是在程式碼中新增一個事件，選擇左上的「類別名稱」找到要新增事件的物件，再從右邊的「方法名稱」找到事件，就會看到如下的程式碼：

```
Private Sub Button1_Click(...) Handles Button1.Click
```

```
End Sub
```

接著，就可以在事件中編寫程式碼，若一個事件中含有多行程式碼，則執行順序為由上至下。同時若在類別「FormX」中的程式碼出現「FormX」的話，需以「Me」代替。

方法（物件本身進行一項動作→主動性）

生活中的例子像是：「命令我的狗開始跑」，和事件相比，最大的不同處在於方法屬於主動性的去執行，而事件則是被動性的等待某件事情發生才去執行；在 VB 中格式的設定為「物件.方法」，在方法之後的小括弧「()」是用來放入一些參數，物件可以依據參數的內容來決定該如何去執行動作，例如 Form2.Close()就是讓 Form2 把自己關閉（Close），常見的方法有：

Close	關閉	Show	顯示(開啟)	Hide	隱藏
-------	----	------	--------	------	----

關於 Close 跟 Hide 的差別：

Hide 只能把表單隱藏起來讓表單變成看不見的狀態，不能真正關閉表單，須使用 Close 才能將表單真正關閉。

關閉表單本身

上文提過，若在類別「Form1」中的程式碼出現「Form1」，需以「Me」代替。

```
Me.Close()
```

補充：兩個文字方塊之內容互換

```
Dim a As Integer
a=TextBox1.Text
TextBox1.Text= TextBox2.Text
TextBox2.Text= a
```

運算子

指定運算子

指定運算子的目的是將左邊的結果指定給右邊，可以想像成左邊為一個大箱子，右邊亦是，把右邊所有東西丟給左邊，也可以想像成左指定成右，下面例子假設 $A=1242$ ，則最後結果 A 為 1252（此類式子可將算術運算子移至指定運算子前，簡寫為 $A+=10$ ）

```
A=A+10
```

算術運算子

算術運算是將兩運算元依運算子類型，進行運算並回傳運算結果，例如： $1+2$ ， 1 、 2 為運算元， $+$ 為運算子，其執行結果為 3，以下為詳細介紹：

運算子	說明	例子	執行結果
+	加	$8+4$	12
-	減	$6-1$	5
*	乘	$120*2$	240
/	除	$80/2$	40
\	相除求商之整數	$18 \setminus 11$	1(18 除以 11 為 1 餘 7)
Mod	相除後求餘數	$17 \text{ Mod } 7$	3(17 除以 7 為 2 餘 3)
^	指數	5^3 $16^{0.5}(16^{(1/2)})$	125(5 的三次方) 4(16 開根號)

串連運算子



	說明	數字	字串	綜合
+	將數字相加，字串串連	$1+2 \rightarrow 3$	"12"+"52" \rightarrow "1252"	"7" & "5"+21 \rightarrow "726"
&	將兩相同型態 或不同型態資料連結	$3 \& 8 \rightarrow 38$	"HS" & "NU" \rightarrow "HSNU"	



比較運算子

運算子	說明	範例	執行結果
>	大於	$1 > 0$	True(1)
<	小於	$2 < 1$	False(0)
=	等於(相等)	$12 = 52$	False(0)
>=	大於等於	$74 >= 12$	True(1)
<=	小於等於	$2 <= 8$	True(1)
<>	不等於	$56 <> 56$	False(0)

邏輯運算子

用來運算布林值 (True 或 False) 的運算子，其中「True」亦可用「1」表示，「False」亦可用「0」表示。

x And y 		y		x Or y 		y	
		True	False			True	False
x	True	True	False	x	True	True	True
	False	False	False		False	True	False

Not 		x		x Xor y 		y	
		True	False			True	False
Not X		False	True	x	True	False	True
					False	True	False

運算子之運算優先順序

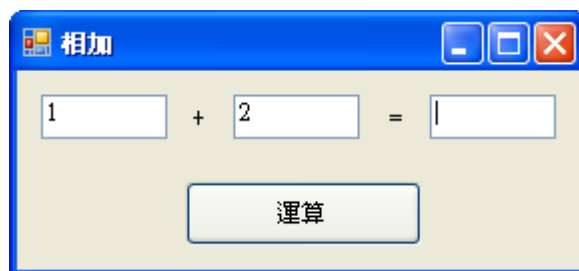
類別	優先權	運算子	類別	優先權	運算子
括號	01	()	比較運算子	09	=
算數運算子	02	^		10	<>
	03	-(負號)		11	<、>
	04	*、/		12	<=、>=
	05	\	邏輯運算子	13	Not
	06	Mod		14	And
	07	+、-		15	Or
串連運算子	08	+、&		16	Xor

結論：括號萬能。

函式

函式也叫做函數，VB 中的函式，就跟數學的函式一樣，只要輸入一個值，函式就會根據一定流程跑出我們想要的結果，函式的表示方式為「函式名稱(欲運算之值)」，這邊以介紹系統內建函式為主。

Val 字串轉數字



以程式碼來解釋：`TextBox3.Text = TextBox1.Text + TextBox2.Text`，此時得解為 `TextBox3.Text` 為 12，很明顯電腦是把兩個文字方塊的內容「連」在一起而非相加，因為電腦預設文字方塊的內容型態為文字，文字無法相加得值，所以電腦將他處理成字串相加，此時需要利用 `Val()` 轉換型態；`Val()` 表示文字轉成數字。但若是直接在程式碼中使數字相加（如：`TextBox3.Text = 1 + 2`），則不需使用 `Val`。

程式碼	TextBox3.Text
<code>TextBox3.Text = TextBox1.Text + TextBox2.Text</code>	12
<code>TextBox3.Text = Val(TextBox1.Text) + Val(TextBox2.Text)</code>	3

Asc 字元轉 ASCII 碼

ASCII 的中文是美國資訊交換標準代碼，主要用於顯示現代英語和其他西歐語言，為現今最通用的單位元組編碼系統之一，常用的 ASCII 碼為下列幾個：

字元	ASCII 碼	字元	ASCII 碼
(空格)	32	0~9	48~57
A~Z	65~90	a~z	97~122

藉由上表我們可以得知，如果現在希望使用者輸入任一個大寫英文字母，我們就能夠使用 `Asc` 函數判斷是否介於 65~90：

```
If Asc(TextBox1.Text) < 65 Or Asc(TextBox1.Text) > 90 Then
    MsgBox("您輸入的不是大寫英文字母")
End If
```

MsgBox 訊息視窗

想將文字輸出，可以使用訊息視窗，也就是 **MsgBox**。當輸出的東西為一個字串(固定不變的資料)如「CRSC」時，記得使用「""」框住，若資料是數字或變數（變動性的資料）如「1200+52」，則無需加註，直接輸出結果「1252」。

程式碼	訊息視窗顯示結果
<code>MsgBox("CRSC")</code>	CRSC
<code>MsgBox(1200+52)</code>	1252

另外，也可以在括號裡修改一些參數，以逗號分隔。如不想改變前面的項目只想改變後面的項目，可在逗號後方空白直接加上下一個逗號。

```
MsgBox(內容, 按鈕類別, "標題")
```

此時可以使用一個變數 **a** 來接 **MsgBox**，這個值的結果將會是使用者選擇按鈕後所要執行的事，至於變數，後方會有詳細說明：

```
Dim a As String
a = MsgBox("CRC is the best!", MsgBoxStyle.YesNo, "我是標題")
If a = vbYes Then
    MsgBox("You are right!")
End If
```

InputBox 輸入視窗

在一個對話方塊中顯示訊息，並等待使用者輸入或按下按鈕時返回其輸入文字：

```
Dim a As Integer
a = InputBox("How old are you?", "年齡調查")
MsgBox("一年後你" & a + 1 & "歲")
```

格式設定：**InputBox("訊息內容", "標題", 預設值, X 座標, Y 座標)**；預設值可有可無，可為數字或字串。

Len 字元數目

Len 函數會傳回一個 **Long**(長整數)的值，其值表示字串內的字元數目，或是儲存一個變數所需的位元組數，下方程式結果會是 4，因為 1252 字串中有 4 個字：

```
Len("1252")
```

Int/Fix 取整數

Int 函數和 Fix 函數都是用來取整數，但兩者的最大差異在於 Int 在負數時是傳回小於等於，Fix 則是小於等於：

	相當於	正數	例子	負數	例子
Int	高斯符號	小於等於	3.14→3	小於等於	-3.14→-4
Fix	無條件捨去	小於等於	3.14→3	大於等於	-3.14→-3

這可以拿來應用於判斷輸入的數是否為小數：

```
Dim a As String
a = InputBox("請輸入數字","Test1")
If Val(a) = Int(Val(a)) Then
    MsgBox("整數")
Else
    MsgBox("小數")
End If
```

Mid 字串中指定數量字元

Mid 函數是從字串中，設定始點及數量，從中抓取一部分出來：

```
Mid("6034125214306",5,4)
```

Mid 函數的格式：**Mid(字串,第幾個開始,抓取幾個)**；所以上述結果為 1252，如果省略後方「抓取幾個」，則抓到字串最尾端。

Rnd 亂數

在寫程式的過程中，常常會需要一些隨機的狀況，而非寫死的固定程式碼，此時就可以用到亂數「Rnd()」來解決這個問題，簡單介紹一下，Rnd()本身有範圍，是 $0 \leq \text{Rnd}() < 1$ ，當範圍設定完成時，電腦即會產生範圍內的亂數，可是由上述條件可以判斷其為小數，若假設我要 0~9 的正整數呢？很簡單，我們只要將 Rnd() 乘 10，此時 $0 \leq 10 * \text{Rnd}() < 10$ ，不過你會發現一個特點：雖然第一個亂數會和下一個不一樣，但是重新跑一次亂數跳出的順序還是一樣(7.055475→5.33424→5.795186→2.895625→3.01948)，為什麼會這樣？由於電腦有內建的亂數表，所以此時電腦只是依序從中抓取，所以嚴格來說也不算是個亂數，此時只要在最上頭加上「Randomize()」，意思是亂數化，這個問題就解決了。

```
Randomize()
MsgBox(10 * Rnd())
```

小秘訣：要取 m~n 之間的整數：「Int((n - m + 1) * Rnd() + m)」

Math 類別之數學函數

函數	功用	例子	結果
Math.Abs	取絕對值	Math.Abs(-1252)	1252
Math.Pow	計算次方	Math.Pow(2,10)	1024
Math.Sqrt	取平方正根	Math.Sqrt(16)	4
Math.Sign	判斷正負（正數傳回 1，負數傳回-1，0 傳回 0）	Math.Sign(29)	1
		Math.Sign(0)	0
Math.Ceiling	取得大於等於該數的最小整數	Math.ceiling(12.52)	13
		Math.ceiling(-12.52)	-12
Math.Floor	取得小於等於該數的最小整數	Math.Floor(12.52)	12
		Math.Floor(-12.52)	-13
Math.Max	兩數取大	Math.Max(1252,1242)	1252
Math.Min	兩數取小	Math.Min(1252,1242)	1242
Math.E	自然對數	Math.E	2.718281828...
Math.PI	圓周率	Math.PI	3.1415926535...
Math.Log10	取對數值（底數為 10）	Math.Log10(2)	0.3010299...
Math.Sin	取得該數的三角函數值，輸入的單位為徑	Math.Sin(Math.PI / 2)	1
Math.Cos		Math.Cos(Math.PI)	-1
Math.Tan		Math.Tan(Math.PI / 4)	1

InStr 字串搜尋、比較位置

當我們需要從 A 字串中搜尋 B 字串在其中最先出現的位置，則可使用 InStr 函數，格式為 **InStr(從哪個數字開始搜尋,字串 A,字串 B)**，以下為範例，可得結果為 7；若未找到則傳回 0：

```
Dim a As String = "KerKer1252Yo!!!"
Dim b As String = "1252"
MsgBox(InStr(1, a, b))
```

Split 字串切割

假設我們在需要處理五位學生的成績，而我們的輸入資料是以逗號做分隔，例如「98,67,91,83,88」，我們該如何將各個成績挑出呢？這時候可以用到 Split 函數，他可以將某個字串 A 依據字串 B 做切割，並以陣列方式儲存，從第 0 號開始往後放，如果省略字串 B 則以空白字元做分割：

```
Dim c()
c = Split("98,67,91,83,88", ",")
```

StrComp 字串比對

字串比對的方法為：先從字串第一個字元開始比較，若字串 A 的 Asc 碼較大則輸出 1，較小則輸出 -1，一樣的話則繼續比較下一個字元，若是其中一個字串已無下一個字元則判定另一字串較大，若兩個字串完全一樣則輸出 0。

字串 A	字串 B	StrComp(A,B)
ra123456	retop	-1
1252	1242	1
CRC	CRC	0

其它字串函數

函數	功用	例子	結果
StrReverse	字串反轉	StrReverse("!社算電")	電算社!
Replace	字串取代	Replace("1242", "42", "52")	1252
UCase	英文小寫轉大寫	UCase("1252Best!!!")	1252BEST!!!
LCase	英文大寫轉小寫	LCase("1252Best!!!")	1252best!!!
Space	產生空白字元	Space(10)	

Shell 執行

簡單來說，Shell 的功能就等同於 開始功能表->執行 的功能。

關機指令：

```
Shell("Shutdown -s -t 30 -c CRCBest!!")
```

超連結功能：

```
Shell("cmd.exe /c start " & "http://dwy6626.github.io")
```

變數與陣列

變數的宣告使用 `Dim`，為 `Dimension` 的縮寫，意思是在 VB 程式中跟電腦要一個空間來儲存一筆資料，格式的設定為「`Dim 變數名稱 As 型態 (= 內容)`」，其中的 `As` 表示「是」，變數名稱的命名最好使用有意義的文字，例如代表成績的變數名稱可用 `score` 表示，以利程式的閱讀、撰寫及維護，而命名變數有幾個規則：

不能使用數字作開頭，如： <code>1A</code>
變數名稱不能有 VB 保留字(屬性、事件、方法、運算子、函式)，如： <code>Integer</code>
變數中無分大小寫，即 <code>abC</code> 和 <code>ABC</code> 是相同的
長度不可超過 1023 個字
變數名稱在同一個有效作用範圍內必須是唯一的，不可重複

下表為常見的變數型態：

變數型態	32 位元整數 (-32768~32767)	64 位元整數 (-2147483648~2147483647)	字串	布林值 (True/False)
	Integer	Long	String	Boolean
預設值	0	0	(空)	False

字串有一點需要注意，假設須讀取字串中的特定字元（第 `n` 個字元），可使用「字串(`n-1`)」，若是在陣列中的字串則為「字串(`x, y, z, ...`)(`n-1`)」。

浮點數

浮點數其實就是小數，又可分為單精度浮點數及雙精度浮點數。單精度浮點數稱為 `Single`，用了四個位元組存放，而雙倍精度浮點數是 `Double`，用了八個位元組存放。

表示數的方式

浮點數的表達方式是以科學計號來表示，日常生活中的科學記號我們會表示為 1.252×10^{31} 、 4.4×10^{-7} （電腦中表示為 1.252×10^{31} 、 4.4×10^{-7} ）... 等等，不過其實在電腦或計算機中，常用的則是第三種方式，它一樣是科學計號，不過就上面的例子，它會這麼表達： $1.252\text{E}+31$ 、 $4.4\text{E}-7$ 。

	單精度浮點數	雙精度浮點數
型態名稱	Single	Double
正數值域	1.401298E-45~3.402823E38	4.940656458127E-324~1.79769313486232E308
負數值域	-3.402823E38~-1.401298E-45	-1.79769313486232E308~-4.940656458127E-324

陣列

如果多個變數是屬於同一個類型的，我們可以使用陣列來存放資料，你可以把陣列想像成一個置物櫃，第一層是 0 號，依序遞增，宣告方式如下：

```
Dim 名稱(最大註標值,最大註標值...) As 資料型態
```

由此可知，當陣列的最大註標值設為 10，因為有包含第 0 號，所以會有 11 格，因此 Dim 名稱(n) As 資料型態，會有 n+1 個；如果程式跑到一半才能知道陣列大小，則可以先略過陣列大小，之後再使用 Redim 重新宣告，格式設定為：Redim 名稱(陣列大小)：

```
Dim strXDD( , ) As String
ReDim strXDD(5, 0)
```

依據最大註標值的個數可以判斷其為一維、二維亦或 n 維陣列；但最重要的，千萬要留意有 0 號，所以說，二維陣列 Dim 名稱(m,n) As 資料型態會有(m+1)×(n+1)個，依此類推；下例有(1+1)×(4+1)=10 個：

A(0,0)	A(0,1)	A(0,2)	A(0,3)	A(0,4)
A(1,0)	A(1,1)	A(1,2)	A(1,3)	A(1,4)

補充：換行方式

也許有人會問，在一個字串中，如何做出換行的效果，在 VB 中有個很特別的換行方式：

```
MsgBox("第一行" & vbCrLf & "第二行" & vbCrLf & vbCrLf & "第四行")
```

上面的訊息視窗會顯示出：

```
第一行
第二行

第四行
```

陳述式

選擇性敘述

If 判斷式

日常生活常常使用到判斷結構，例如，若今天放學是晴天，我就去打球，這就是一個判斷結構，依據今天的天氣狀況，來決定處理的動作，當今天放晴這件事為真時，就去打球，在程式碼中會用「If...Then」來表示，以下為判斷奇偶數的程式：

```
If Val(TextBox1.Text) Mod 2 = 0 Then
    MsgBox("偶數")
Else
    MsgBox("奇數")
End If
```

如果要處理的動作只有一行程式碼的話，可以將其縮寫到 Then 後面：

```
If TextBox1.Text = "1252" Then MsgBox("Best")
```

如果要判斷的情況有 3 種以上，可在 Else 的前面加入 ElseIf：

```
If Val(TextBox1.Text) > 0 Then
    MsgBox("正數")
ElseIf Val(TextBox1.Text) < 0 Then
    MsgBox("負數")
Else
    MsgBox("0")
End If
```

Select 選擇

Select 的功用和 If 相似，只是 Select 主要是用在一個特定變數的各種情況下要做出什麼處理：

```
Dim intScore As Integer
Select Case intScore
    Case 90 To 100
        MsgBox("甲等")
```



```
Case 80 To 89
    MsgBox("乙等")
Case 70 To 79
    MsgBox("丙等")
Case 60 To 69
    MsgBox("丁等")
Case Else
    MsgBox("戊等")
End Select
```

迴圈

迴圈就是一直重複執行某件事情，常見的有 For...Next 及 Do...Loop。

For 迴圈

For 迴圈主要用於需執行固定次數的迴圈，格式為：For 變數 = 起始值 To 終值 (Step 變量)，變量意指每次變數增減的量，可正可負，預設為 1，For 和 Next 之間則撰寫欲重複執行的事情。以下程式為計算 1 加到 10 的值：

```
Dim ans As Integer
For i = 1 To 10 Step 1
    ans += i
Next
```

Do 迴圈

Do 迴圈主要用於在某種條件下重複執行的迴圈，分兩種，一種是在執行後判斷是否重複執行，也表示至少會跑一次，如上述程式碼，另一種則是先判斷；再者，是否重複執行的條件也分成兩種，一是 While，另一個是 Until，以下程式也是計算 1 加到 10 的值：

```
Dim i, ans As Integer
i = 1
Do
    ans += i
    i += 1
Loop Until i = 10
```

將兩種迴圈做個比較：

迴圈別			次數
For...Next			固定
Do Loop	While	While 緊接 Do	不固定，條件為真則重複
		While 緊接 Loop	不固定，但至少一次，條件為真則重複
	Until	Until 緊接 Do	不固定，條件為假則重複
		Until 緊接 Loop	不固定，但至少一次，條件為假則重複
	無條件		無窮迴圈

再者，兩者還有一個差別，就是 For 迴圈帶有變數可以利用，而 Do 沒有；有時候，我們會在迴圈執行的過程中，做一些判斷，再藉由判斷結果告訴電腦要繼續跑還是跳出：

Continue	回到開頭繼續執行迴圈	Exit	跳出迴圈
----------	------------	------	------

比較看看：

以下的 k 值為 3+6+9=18：

```
Dim k As Integer
For i = 1 to 10
    If i Mod 3 <> 0 Then Continue For
    k += i
Next
```

以下的迴圈會不斷跳出輸入視窗，直到使用者輸入的數為 1252 為止，則跳出訊息視窗「恭喜猜中！」：

```
Dim ans As Integer
Do
    ans=InputBox("請輸入一數", "終極密碼")
    If ans = 1252 Then Exit Do
Loop
MsgBox("恭喜猜中!")
```

常用工具與技巧

計時器 Timer

在程式設計時，常常會用到計時的功能。Timer 是一個每隔一段時間重複執行一段程式碼的元件，修改間隔時間的屬性名稱為「Interval」，要注意的是其計算單位為毫秒（千分之一秒），如果要啟動，則切記要將屬性「Enabled」設成「True」，下方為倒數計時器的範例：

```
Private Sub Timer1_Tick(...) Handles Timer1.Tick
    Label1.Text = Val(Label1.Text) - 1
    If Val(Label1.Text) = 0 Then
        Timer1.Enabled = False
        MsgBox("時間到")
    End If
End Sub
```

即時運算 Debug.Print

你可以把即時運算想成一張計算紙，有時候我們會希望以簡單的方式來計算或呈現某些運算的結果，此時可使用即時運算功能，例如說我想知道 $(2 > 3) \text{ Xor } (5 > 3)$ 的結果為何，則可將即時運算視窗打開 (Ctrl+G)，輸入 $?(2 > 3) \text{ Xor } (5 > 3)$ ，可得結果為 True。

監看式

當我們撰寫一個比較複雜的程式，常常出了錯卻不知道問題在哪，此時可以利用偵錯配合監看式的功能，來找出哪兒有問題，例如程式片段中 **For i = 2 to a - 1**，我想知道 i 跑到多少時出錯了，或是隨時監看 i 的值，則可先按 F8 偵錯，讓程式逐步執行，執行過程中對於某些變數想做監看，可按右鍵選擇「加入監看式」，找出問題所在。

選項按鈕及核取方塊 **RadioButton**、**CheckBox**

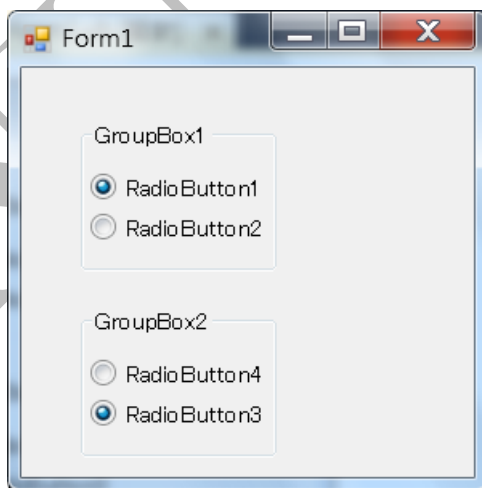
假設現在需要利用 VB 做一份問卷，其中有性別欄，興趣欄，星座…等等，為避免使用者以文字方塊輸入的格式與設計此程式的人有所出入，所以我們可以利用選項按鈕(**RadioButton**)和核取方塊(**CheckBox**)來獲得所需的資訊，兩者最大的差別在於 **RadioButton** 適用於單選，如性別、星座等等，**CheckBox** 可複選，如興趣；而這兩者最常使用的屬性為 **Checked**，用以表示此選項是否被選取，也可搭配 **If** 判斷，當使用者勾選何者會出現何種狀況，如下：

```
If RadioButton1.Checked = True Then MsgBox("您是男性")
```

```
If CheckBox1.Checked = True And Not CheckBox2.Checked = True Then  
MsgBox("您的興趣是唱歌")  
End If
```

然而這種寫法會稍嫌麻煩，以第一個例子來說，`If RadioButton1.Checked = True`，用中文來解釋，會變成「如果 **RadioButton1** 被勾選是真的」，聽起來很繞舌，所以我們習慣把後方的比較拿掉，變成 `If RadioButton1.Checked`，直接解釋成「如果 **RadioButton1** 被勾選」，這樣就清楚多了。

通常在表單內，使用者一次只能選取一個 **RadioButton**，若要做像考卷中選擇題一樣把 **RadioButton** 分組的效果，可以使用群組方塊 **GroupBox**：



圖片控制項 **PictureBox**

在 VB 中可以利用圖片控制項來放圖片，首先先拉出一個 **PictureBox**，然後可以從右上角的選擇改變調整大小的模式，一般來說選擇 **AutoSize**，接著從屬性中選擇 **Image** 匯入圖片即可。

繪圖

在 VB 中我們也可以透過函式的繪圖物件 **Graphics** 來繪圖，首先我們必須先宣告一個變數，型態是 **Graphics**，接著產生一個 **Graphics** 物件，指定給我們宣告的變數，接著就可以利用此變數繪圖，最後再將其釋放，假設現在要畫一條直線，則格式為：變數名稱.**DrawLine**(**Pens**.顏色, 始點 x, 始點 y, 終點 x, 終點 y)；現在我們試著畫出一條藍色的線段，從(0,0)到(120,310)：

```
Dim g As Graphics
g = Me.CreateGraphics
g.DrawLine(Pens.Blue, 0, 0, 120, 310)
g.Dispose()
```

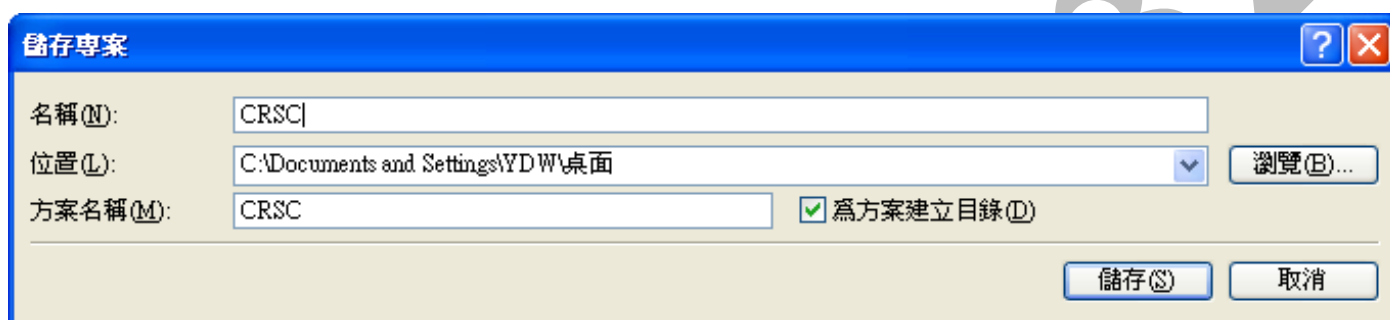
我們也可以做一個簡易小畫家：

```
Dim down As Boolean
Private Sub Form1_MouseDown(...) Handles Me.MouseDown
    down = True
End Sub
Private Sub Form1_MouseMove(...) Handles Me.MouseMove
    If down Then
        Dim color
        If RadioButton1.Checked Then
            color = Pens.Red
        ElseIf RadioButton2.Checked Then
            color = Pens.Blue
        ElseIf RadioButton3.Checked Then
            color = Pens.Green
        End If
        Dim g As Graphics
        g = Me.CreateGraphics
        g.DrawLine(color, e.X, e.Y, e.X + 1, e.Y + 1)
        g.Dispose()
    End If
End Sub
Private Sub Form1_MouseUp(...) Handles Me.MouseUp
    down = False
End Sub
```

專案處理

儲存專案

當做出了一個精彩的作品之後，要做的事情當然就是把它儲存起來。在工具列中找到「全部儲存」按鈕，按下去之後你會看到以下視窗：



「為方案建立目錄」的意思是說會建立一個資料夾把要儲存的專案都丟到裡面，不過基本上差別不大，打勾與否皆可。

建置專案

若要把做好的程式分享出去，就在工具列上方找到「建置」的第一項「建置 專案名稱」，按下之後即可。不過要注意就是 VB 不會讓你選擇建置的地點，所以建議先儲存專案之後再建置，建置產生的檔案會被放在資料夾「專案名稱」裡的「bin」裡的「Release」。

細部修改

有時候我們會想要修改像例如專案名稱、版本，或商標之類的東西，可以點選方案總管中樹狀圖最左上方的(專案名稱)，再點選左上方的「屬性」鈕，即可調整。其中有一項「啟動表單」代表程式預設會開啟哪個表單。而版本之類的東西則要從「組件資訊」中修改。

附錄

參考資料

1. 李啟龍老師的「Visual Basic 2008 程式設計 16 堂特訓」
2. Dels Lu 學長的 Visual Basic 講義
3. 大量網路資料以及自己的經驗

後記

這是筆者 2010 年於師大附中電算社的社課所開設的一學期 Visual Basic 小組社課課程講義。並於 2020 年（也隔太久了吧）潤飾公開於網路上。

因為是入門課程，因此講義中沒有提到一些比較進階的重要概念，例如「自訂函式」、「類別」等。另外，如今 Visual Studio 已經出到了 2019 版，不過基本的概念是相同的，程式碼部分也可以通用，因此仍然將這套講義公開，希望能夠對社群有所幫助。