

# 目錄

- ◆ 第二頁      假設運算式
- ◆ 第三頁      迴圈函數
- ◆ 第四頁      副程式
- ◆ 第五頁      陣列（一維、二維、**ArrayList**）
- ◆ 第六頁      陣列（清單方塊）、訊息視窗
- ◆ 第七頁      數學類別內建函數 (**Math class**)
- ◆ 第八頁
- ◆ 第九頁
- ◆ 第十頁

## 假設運算式：

```
If 條件 then 執行程式
Elseif 條件
    執行程式
Else： 執行程式 (以上條件皆沒成立時才會執行)
End if
```

---

```
IIf ( 條件 , 成立傳回此值 , 不成立執行 )
```

---

```
Select case 要比對ㄉ值
Case Is 比對ㄉ條件
    執行程式
End Select
```

p.s：此函數中的 Is 代表你要比對的值

---

Choose ( 要比對ㄉ值 , “要執行ㄉ東西” , “同旁” , “同旁” , ..... )

例： a = Choose ( i , "1" , "1123" )

如果 i 為 2 則 a 結果為 1123

如果 i 為 3 則 a 無結果\

---

Microsoft.VisualBasic.Switch ( 條件 1 , 成立執行此 , 條件 2 , 成立執行此 ..... )

p.s: 在此函數中條件無法使用省略運算式

例： b = Microsoft.VisualBasic.Switch ( a = "男" , "先生" , a = "女" , "小姐" ) \

如果 a 為男，則 b 為先生

如果 a 為女，則 b 為小姐

---

## 迴圈函數：

For **i** = 數 to 數 Step 每多少走一步

p.s: **i** 可用符合還未使用的英文字母來代替,吾...有些英文字母好像不行...

執行程式

Next

p.s : 可用 **Exit for** 中途離開迴圈

---

For Each 變數 in 陣列名稱

執行程式

Next

p.s : (1)執行次數為陣列中有多少元素

(2)執行第 n 次，變數為陣列中的第 n 位

(3)變數行別必須跟陣列中元素的型別一樣或宣告為 **Object** 也 ok

---

Do

執行程式

Loop

p.s: 當發生無窮迴圈時能按 **Ctrl + Backspace**

p.s : 可用 **Exit Do** 中途離開迴圈

---

Do While 條件 (成立此條件執行下面)

執行程式

Loop

---

Do (不管數是多少先丟進來判斷)

執行程式

Loop While 條件 (不成立此條件才可出來)

---

Do Until 條件 (不成立此條件執行下面程式)

執行程式

Loop

---

Do (不管數是多少先丟進來判斷)

執行程式

Loop Until 條件 (成立此條件時才可出來)

## 呼叫副程式：

範圍 => Private 私有 / Public 共用

傳址方法 => ByVal 兩變數不重疊 / ByRef 變數被覆蓋重疊

## 建造副程式：

```
[ 範圍 ] Sub 程序名稱 [ ( 傳址方法 虛引數 As 變數樣式 , ... ) ]  
    [ Exit Sub ]  
End sub
```

p.s：若無傳遞資料則引數可省略

---

```
[ 範圍 ] Function 程序名稱 [ ( 傳址方法 虛引數 As 變數樣式 , ... ) ]  
    [ Exit Fncion ]  
End Function
```

(此函數會自動回傳值)

呼叫方法可直接打： 某變數 = 程序名稱( 實引數 )

## 呼叫副程式：

語法 1：Call 程序名稱( 實引數 , ... )

語法 2：程序名稱 ( 實引數 , ... )

p.s：實引數可為常數、變數、運算式、陣列、紀錄、物件，但虛引數不可為常數和運算式

## 鍵盤事件：

## (重要)陣列：

通用：

算出指定維度最多的註標是多少 => UBound ( 陣列名稱 , 第幾維 )

算出共有多少陣列元素 => 陣列名稱.Length

算出指定維度共有多少註標 => 陣列名稱.GetLength ( 第幾維 )

清除整個陣列 => Erase 陣列名稱

重新配置陣列元素不留下原有資料 => ReDim 陣列名稱 ( 維度 )

重新配置陣列元素並且留下原有資料 => ReDim Preserve 陣列名稱 ( 維度 )

**p.s：** (1)若在同一段程序中還要再使用被 Erase 的陣列 則需再用 ReDim 來宣告

限定一維：

將陣列中的數由小到大排序 => Array.Sort ( 陣列名稱 )

將陣列反轉排列 => Array.Reverse ( 陣列名稱 )

找出一個值在陣列中的第一個位置 => Array.IndexOf ( 陣列名稱 , 搜尋物件 )

**ArrayList(為 Object 的型別)：** Dim 陣列 As New ArrayList ( )

將陣列反轉排列 => Array.Reverse ( 陣列名稱 )

將陣列中的數由小到大排序 => Array.Sort ( 陣列名稱 )

算出陣列中元素的數目 => 陣列名稱.Count

得到指定位址的內容 => 陣列名稱.Item ( 位址 )

新增元素到陣列的最後一位=> 陣列名稱.Add ( 新元素 )

清除陣列中所有的元素 => 陣列名稱.Clear ( )

搜尋指定元素是否存在 => 陣列名稱.Contains ( 會傳回 False or True )

搜尋指定元素是否在陣列中=> 陣列名稱.IndexOf ( if 存在傳回 )

插入元素到陣列中任意位置=> 陣列名稱.Insert ( 位置 ) ( 超出範圍會出錯 )

移除陣列中指定註標的元素=> 陣列名稱.RemoveAt ( 位置 ) p.s:後面的元素皆會倒退一個

## 清單方塊 **ListBox**

屬性：

Items 所有選項之集合

SelectedIndex 選項的索引註標

Sorted 選項是否使用字母順序排列

MultiColumns 是否多欄顯示

SelectionMode 選項選取方式

依序為：無選取、選一、能配合 **Ctrl**、能配合 **Ctrl** or **Shift**

函數：

增加新的單項目 => 清單名稱.Items.Add ( 字串變數 or 字串資料 )

增加新的多項目 => 清單名稱.Items.AddRange ( 陣列名稱 )

移除指定的選項 => 清單名稱.Items.Remove( 字串名稱 or 字串變數 )

移除全部的選項 => 清單名稱.Items.Clear

取得項目總數 => 清單名稱.Items.Count

## 下拉式清單方塊 **ComboBox**

屬性：

DropDownStyle 下拉式選單的型式

(其餘同清單方塊的屬性、函數)

## 訊息視窗：

另設視窗：Inputbox => ( "內容", [ "標題", 預設值 , x 座標 , y 座標 ] )

- (1) 內容不可省略，若要沒有的話必須使用 “ ” 表示
- (2) 標題可省略，但逗號必須保留，省略會以專案名稱當作標題
- (3) x y 座標省略會顯示在螢幕中央偏上方 1/3 處
- (4) 按取消，會傳回空字串

• 訊息視窗：Msgbox => ( "內容", 格式碼 , "標題" )

## **TextBox** 常用語法

清空文字裡的内容 => Clear( ) or “ ”

停駐焦點 => Focus ( )

資料的格式 => Format ( 運算式 , “格式” )

換行 => vbNewLine or vbCrLf

定位點 => vbTab **p.s :** 只限用在指標能停在那上面的物件

## 數學類別內建函數 (Math class)

判斷正負 => Math.Sign ( num )

例：Math.Sign(12) 結果為 1

Math.Sign( 0 ) 結果為 0

Math.Sign(-15.1) 結果為 -1

傳回最大值 => Math.Max ( num1 , num2 )

傳回最小值 => Math.Min ( num1 , num2 )

絕對值 => Math.Abs ( num )

取大於或等於最小整數 => Math.Ceiling ( num )

取小於或等於最大整數 => Int ( num ) / Math.Floor ( num )

例：Int(99.8) = 99

Int(-99.7) = -100

四捨六入 => Cint ( num ) / Math.Round ( num )

注意！當小數第一位是 5 時

Cint(64.5) = 64 因 64.5 的個位數為偶數，故不進位

Cint(65.5) = 66 因 65.5 的個位數為奇數，故進位

Cint(42.53) = 43 因小數第一位為 5 且小數有兩位會進位

無條件捨去小數 => Fix ( num )

平方根 => Math.Sqrt ( num )

次方 => Math.Pow ( num , 次方 )

## 字串函數：

刪除最左邊全部空白 => LTrim ( str )

刪除最右邊全部空白 => RTrim ( str )

刪除頭尾兩邊全部空白=> Trim ( str )

計算有幾位數 => Len ( str )

從右邊開始取 => Microsoft.VisualBasic.Right( str, 取幾位 )

從左邊開始取 => Microsoft.VisualBasic.Left( str, 取幾位 )

取中間 => Mid( str, 從哪裡開始取 [,取幾位數 ] )

空格：Space ( num )

## 轉換函數：

轉換成數值 => Val ( num )

轉換成字串 => Str ( num ) or CStr ( num )

轉換成日期型態 => CDate ( str )

大寫英文轉換成小寫 => LCase ( str ) p.s:若有非英文的字，則該字將不會改變

小寫英文轉換成大寫 => UCase ( str )

轉換編碼 => ASC ( str ) p.s:只會傳回第一個字的編碼

將編碼轉換成 ASC 所對應字元 => Chr ( 數字 )

## 日期與時間函數：

傳回系統日期與時間 => Now p.s:時間日期型態資料的兩邊皆會有 # 號

設定或傳回系統日期 => Today

設定或傳回系統時間 => TimeOfDay

傳回西元年 => Year ( datetime ) ( datetime = 日期型別資料 )

傳回月 => Month ( datetime )

傳回日期 => Microsoft.VisualBasic.DateAndTime.Day ( datetime )

傳回星期 => WeekDay ( datetime ) p.s: 傳回值範圍(1~7) = (星期日~星期六)

傳回小時 => Hour ( datetime ) 24hr

傳回分 => Minute ( datetime )

傳回秒 => Second ( datetime )

得到從凌晨 0 時 0 分 0 秒到目前累積的總秒數 => Microsoft.VisualBasic.Timer

## 設定圖片：

設定圖檔 => Name.Image = Image.FromFile ( “路徑” )

移動圖片 => Name.Location = New Point ( x , y )

圖片方塊的寬高 => Name.Size = New Size ( 寬 , 高 )



## 其他：

設製字之樣式： `New Font(“字型”，大小，樣式)`

`Fontstyle .bold`       => 粗體  
      `.Italic`       => 斜體  
      `.Regular`       => 標準  
      `.Strikeout`   => 刪除線  
      `.Underline`   => 底線

設定數值的樣式： `Format ( num, “樣式” )`

p.s : 0=>若該位數無值則補 0

#=>若該位數無值則不補 0

(會自動 4 捨 5 入，更詳細請翻課本 4-11)

使用三原色選色方法： `Color.FromArgb ( R ,G ,B )`

超連結： `Process.Start ( “位址” )`

關閉程式： `End` / `Application.Exit ( )`

`Me` 為視窗本身的名子

關閉表單   =>   表單名稱.

顯示表單   =>   表單名稱.Show( )

隱藏表單   =>   表單名稱.Hide( )

• 亂數： `Rnd ( )`

p.s : 若不希望每次使用皆得到相同順序的亂數，則可在亂數的前一行使用

`Randomize( )` 來以系統時間當作亂數產生的種子

資料型別		記憶體	有效範圍
位元組	Byte	1byte	0~255 的整數
位元組	SByte	1byte	-128~127 的整數
短整數	Short	2byte	-32,768~32,767 的整數
整數	Integer	4byte	-2,147,483,648~2,147,483,647 的整數 (約 10 位數的正負整數)
長整數	Long	8byte	-9,223,372,036,854,775,808~9,223,372,036,854,775,807 (約 19 位數的正負整數)
單精確度	Single	4byte	(可含小數，有效位數為 7 位)

倍精確度 Double	8byte	(可含小數，有效位數為 15 位)
貨幣型別 Decimal	16byte	(可含小數，有效位數為 28 位)