

ch02

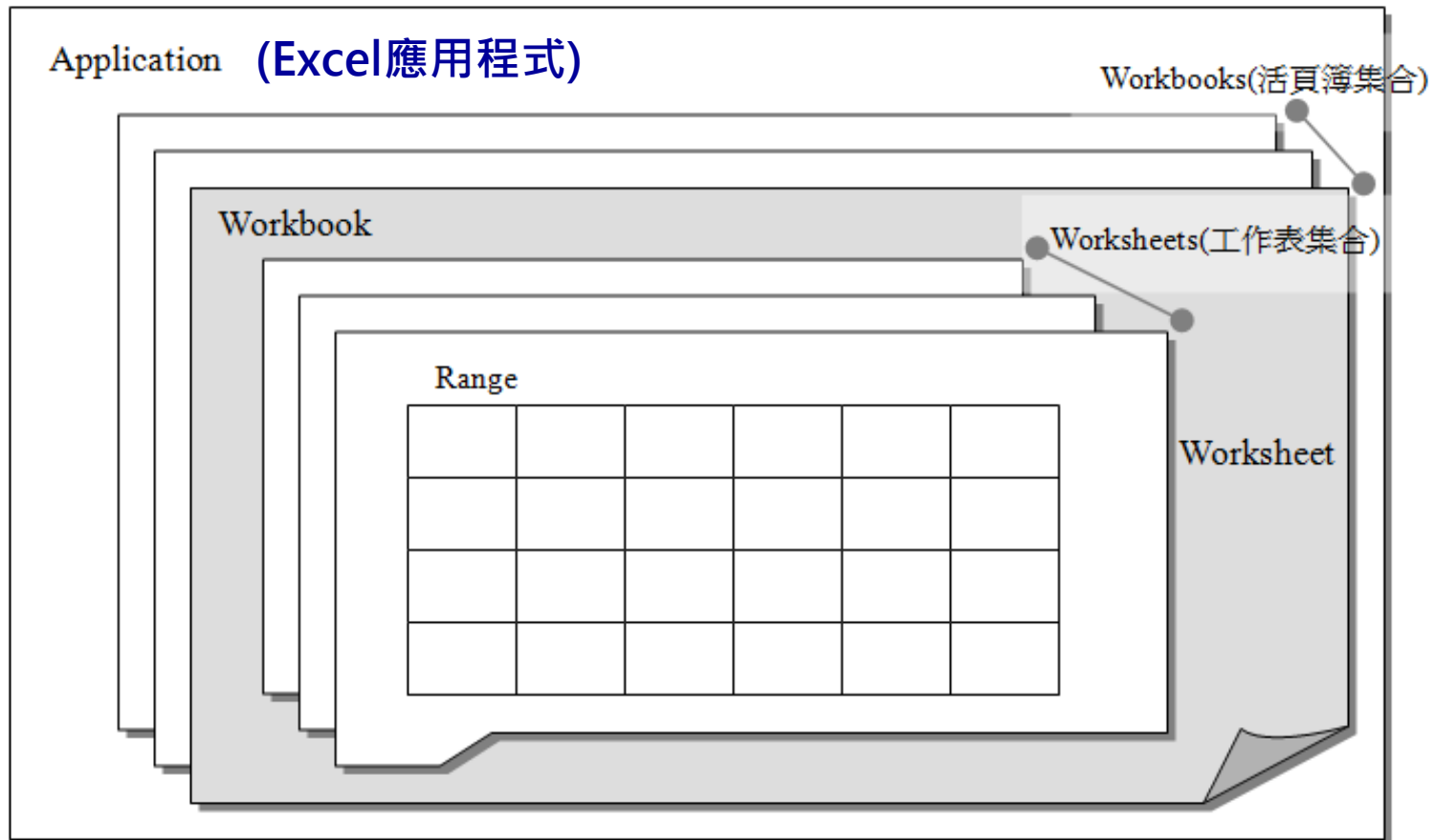
第一個Excel VBA程式

Ch02 內容

- 2.1 Excel VBA 物件架構
- 2.2 自行撰寫巨集程序
- 2.3 Excel VBA 程式編輯器
- 2.4 VBA程式的編輯和除錯
- 2.5 設定 Excel VBA 專案的保護

2.1 Excel 物件架構

Excel基本物件模型(階層結構)



Excel基本物件

1. **Application** : Excel應用程式就是一個Application物件，其中可以包含多個活頁簿。
2. **Workbooks** : 每一個活頁簿檔案就是一個Workbook物件，多個Workbook物件就存在Workbooks活頁簿集合(Collection)中。
3. **Worksheets** : 每一個工作表就是一個Worksheet物件，多個Worksheet物件就存在Worksheets工作表集合中。
4. **Range** : Range就是儲存格的範圍。
5. 其他 : **Cells**儲存格集合和**Charts**圖表工作表集合、**Windows**集合...等多種物件。

屬性與方法

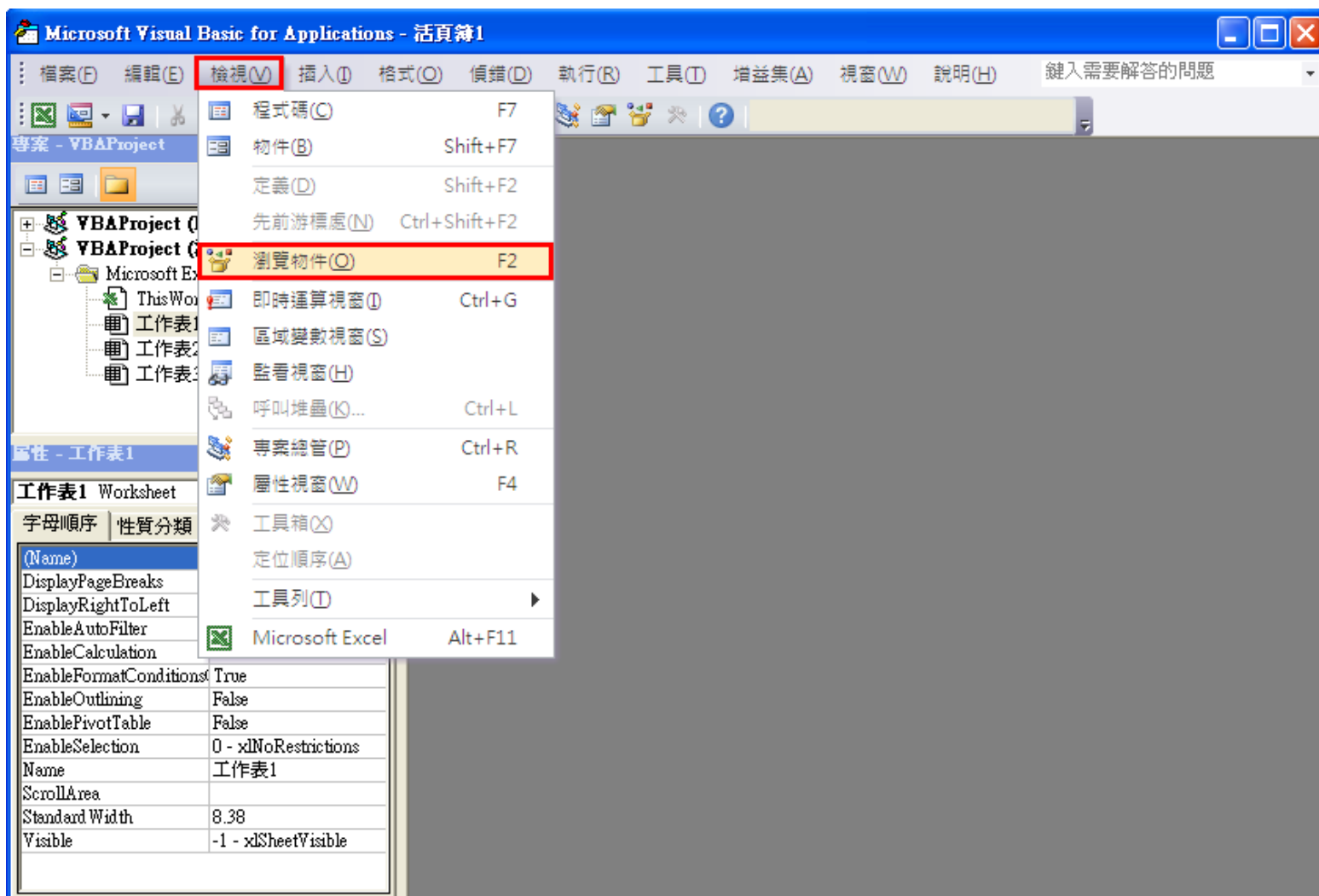
- 每個物件都有其所屬的**屬性**(Attribute)和**方法**(Method)。
- 物件**不同的特質**稱為物件的「**屬性**」。
- 物件具有的**特定功能**稱為物件的「**方法**」。
- 每個人屬性值和動作方式不同，能夠區分出不同的物件。
- 物件的屬性、物件的方法，用「**.**」連接。
- Value是儲存格(Cell)的屬性，記錄儲存格的內容。
 - 如指定A1儲存格的內容為「Excel VBA」，其寫法為：
`Range("A1").Value="Excel VBA"`
- Select是儲存格(Cell)的方法，設定為作用儲存格。
 - 如指定A1儲存格的為作用儲存格，其寫法為：
`Range("A1").Select`

事件(Event)

- 物件彼此間也會有**互動關係**，這些互動會**觸動**(引起)「**事件**」(Event)。
- 事件的發起者**觸發事件**→
事件的接受者**執行觸發事件**的程式碼來作「**回應**」。
- VBA針對各種物件**定義一些事件**，這些事件會在該物件**發生特定狀況時被觸動**，針對該狀況要**處理的程式碼**寫在該事件程序中。
 - － 例如：在開啟活頁簿物件時**會觸動Open事件**，可以將開啟活頁簿要處理的程式碼寫在Open事件程序中。

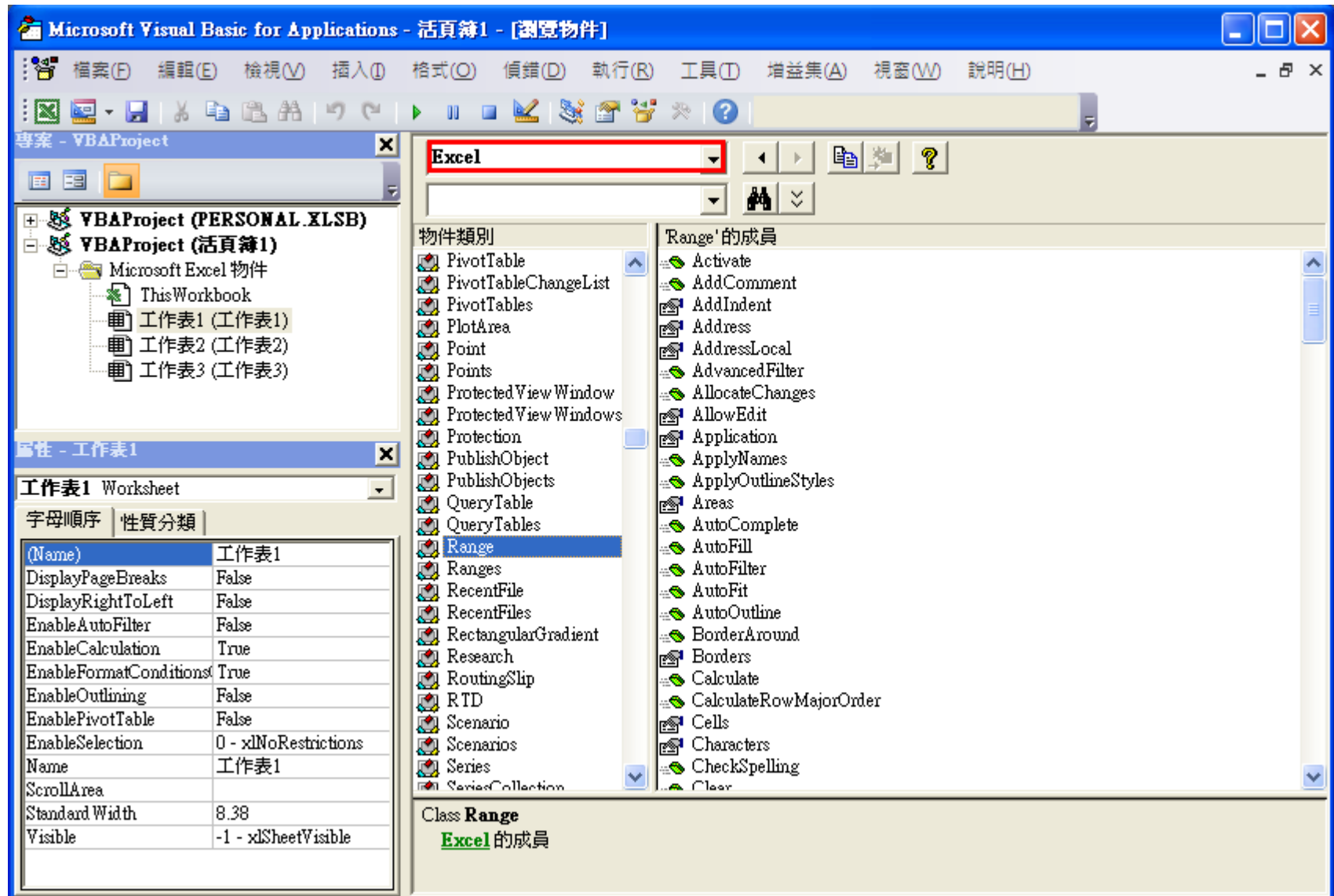
瀏覽物件

開啟「瀏覽物件」視窗來查看VBA所屬的物件：





Excel所屬的物件



指定儲存格--Range

指定Excel應用程式之Book1.xlsm活頁簿檔案中，名稱為「工作表1」的工作表中A1儲存格範圍，其完整的寫法為：



指定目前工作表中A1到B3儲存格範圍，其簡略的寫法為：

```
Range("A1:B3")
```

指定儲存格--Cells

語法：

Cells (水平列編號， 垂直欄編號)

列和欄都是以數字標示，適合用**變數**指定儲存格時。

例如：要指定B4儲存格時，

列編號為4、

欄編號則為2(A是1、B是2...)，

所以其寫法為：**Cells(4,2)** → 等同Range("B4")

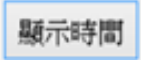
2.2 自行撰寫巨集程序

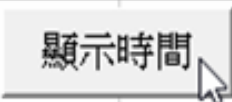
- 錄製巨集可自動產生程式碼，但是功能有限。
- 了解Excel VBA 物件的基本架構，以及指定儲存格的語法後，可自行撰寫巨集程序的程式碼。



實作

FileName : First.xlsm

設計一個在  按鈕控制項上按一下，會將目前時間顯示在 B1 儲存格的巨集程序。

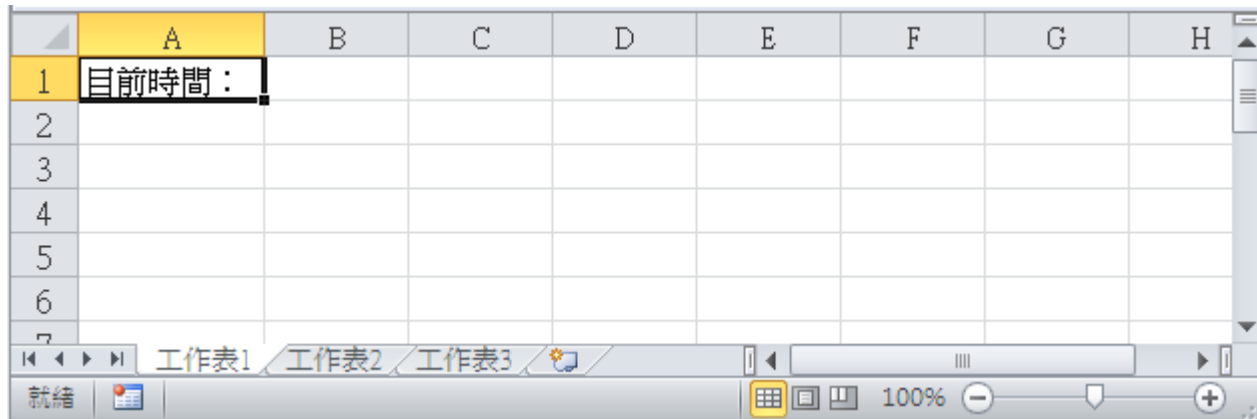
	A	B	C	D
1	目前時間：	2015/11/6 16:37		
2				

First.xlsm 操作步驟

1. 建立工作表內容
2. 新增巨集程序
3. 撰寫程式碼
4. 建立按鈕控制項並指定巨集
5. 測試巨集

1. 建立工作表內容

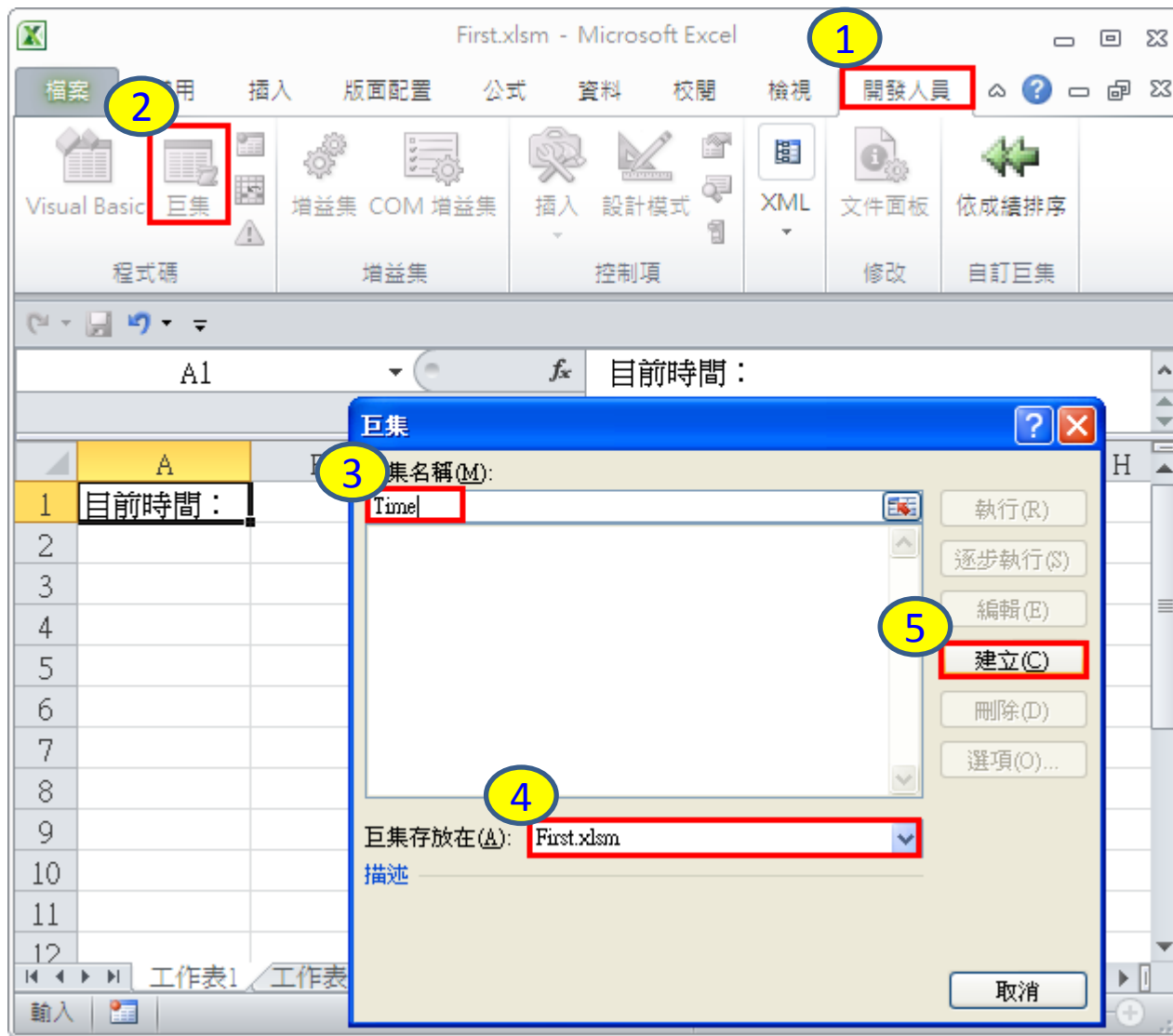
- 新增空白活頁簿，並在「工作表1」工作表建立內容：



- 以First為檔名，存檔類型選取「Excel 啟用巨集的活頁簿」存檔。

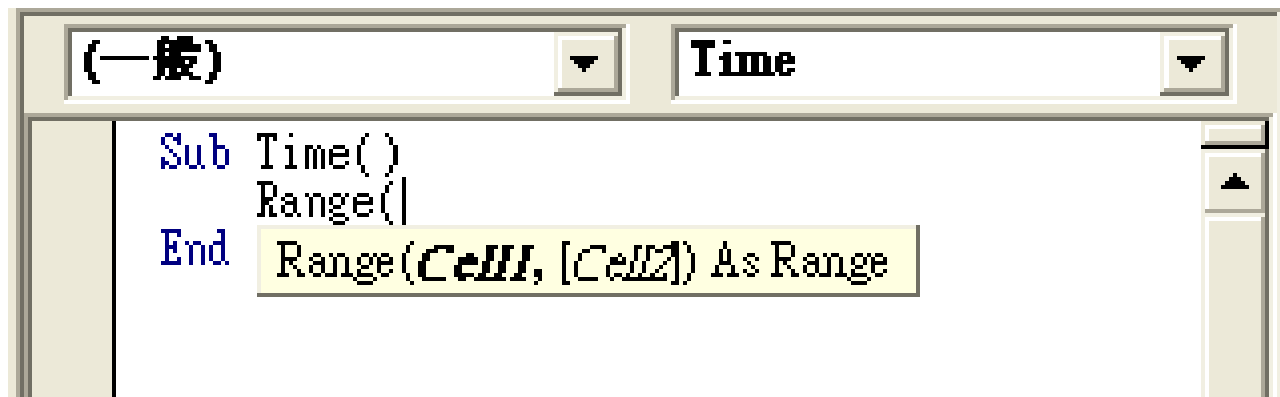
2.新增巨集程序

- 開啟「巨集」對話方塊設定「巨集」對話方塊：



3. 撰寫程式碼

- 按 <Tab> 鍵向右縮排方式編寫程式碼以方便閱讀。
由鍵盤鍵入「range(」

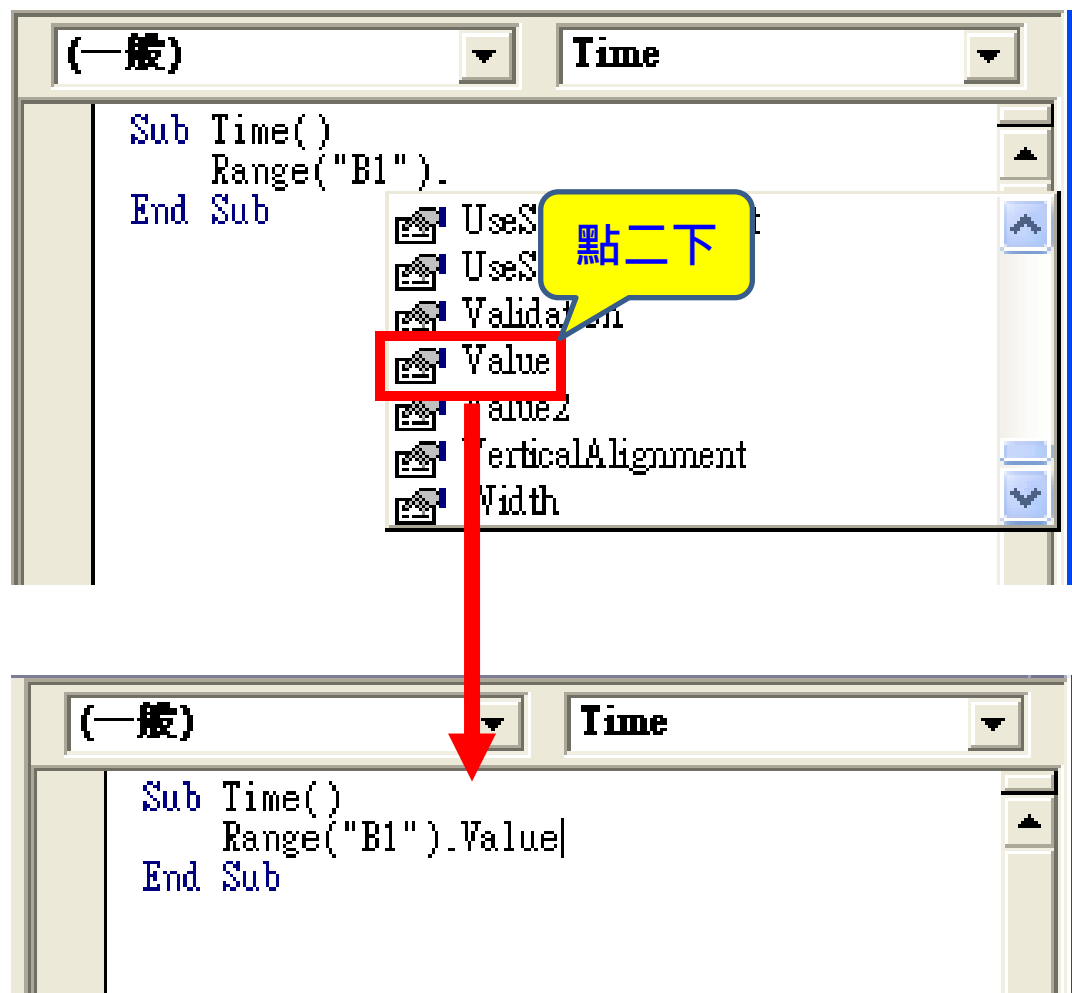


The screenshot shows a VBA editor window with a title bar containing two dropdown menus: "(一般)" and "Time". The code editor area displays the following VBA code:

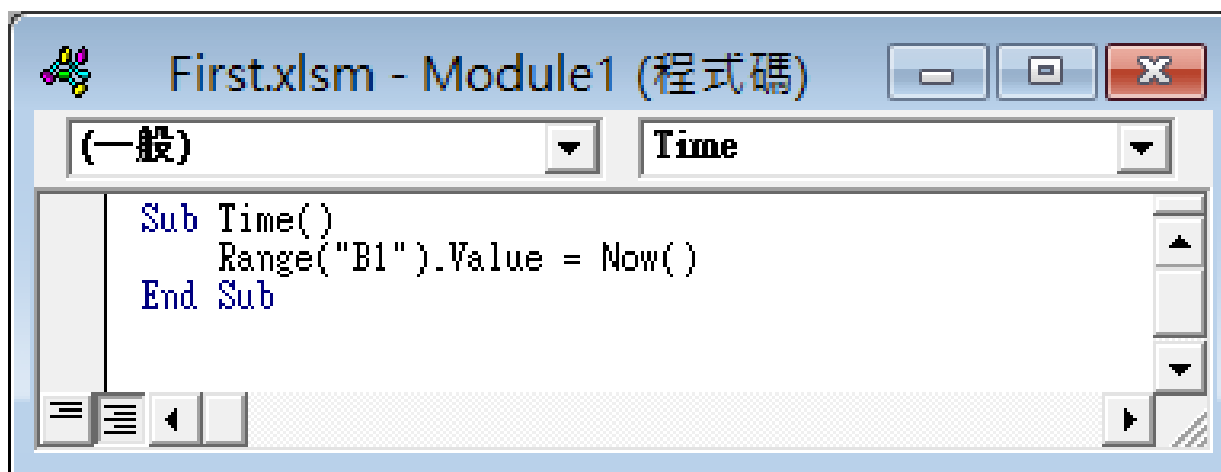
```
Sub Time()  
    Range(  
End Range(Cell, [Cell2]) As Range
```

The text "Range(*Cell*, [*Cell2*]) As Range" is highlighted in yellow, indicating it is the current line being edited. The code is indented to the right, demonstrating the use of the Tab key for readability.

- 編輯器的智慧感知(IntelliSense)功能：

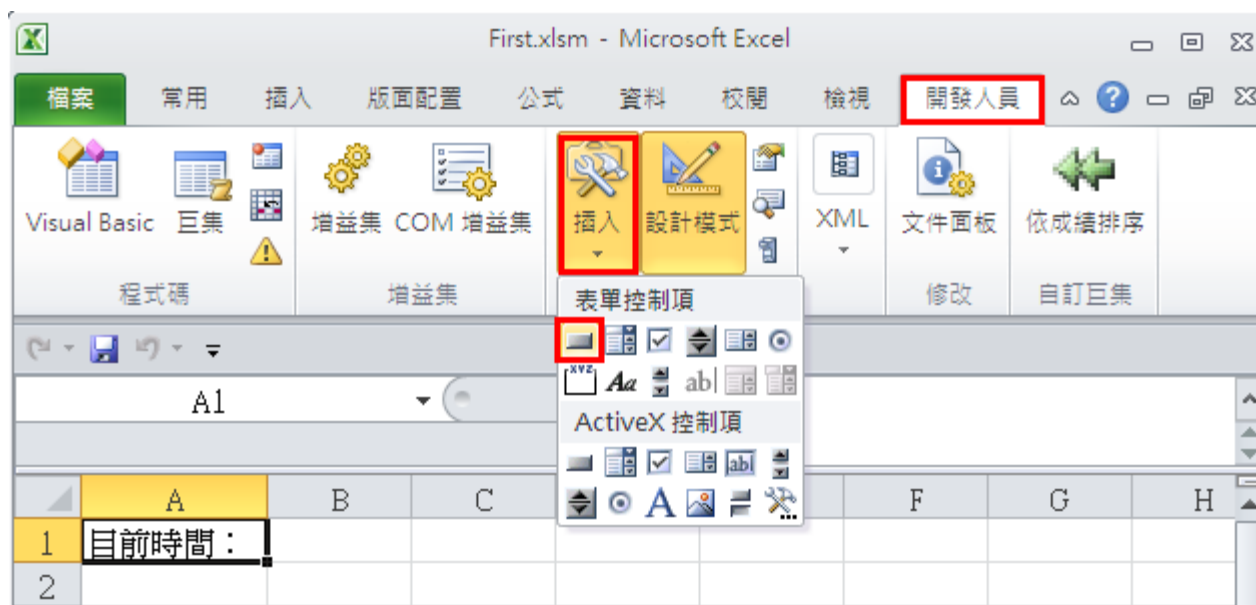


- 輸入「range("B1").Value=now()」後，輸入游標移到下一行時，系統會自動調整程式碼，該變大寫、加空白...等都會自動完成。

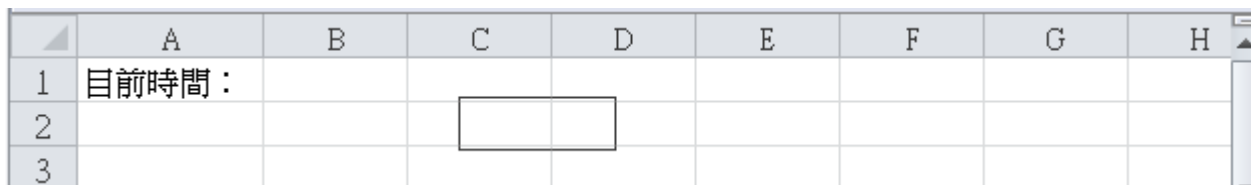


4. 建立按鈕控制項並指定巨集

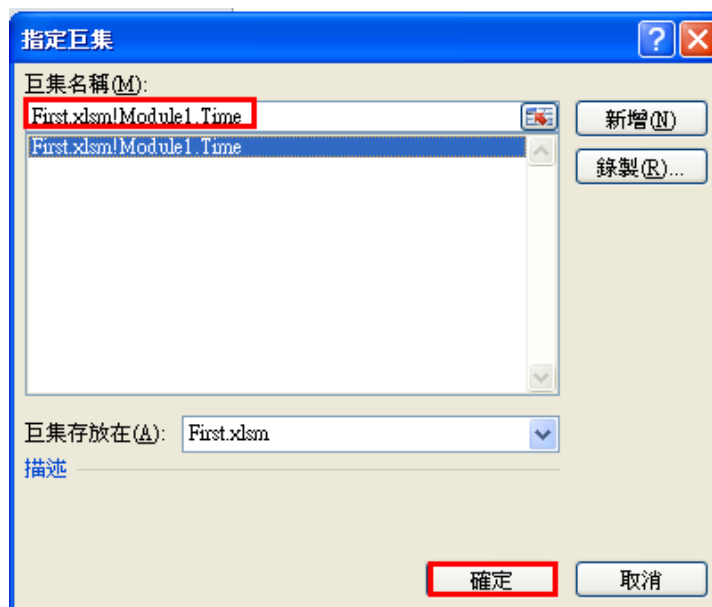
- 開發人員 → 插入 → 按鈕控制項



- 拉出按鈕大小



- 「指定巨集」對話方塊，點選「Time」巨集名稱。



- 將按鈕上面的文字修改為「目前時間」，並調整按鈕控制項的大小。

	A	B	C	D	E	F	G	H
1	目前時間：							
2				目前時間				
3								

5.測試巨集

- 按  鈕測試執行結果是否正確？

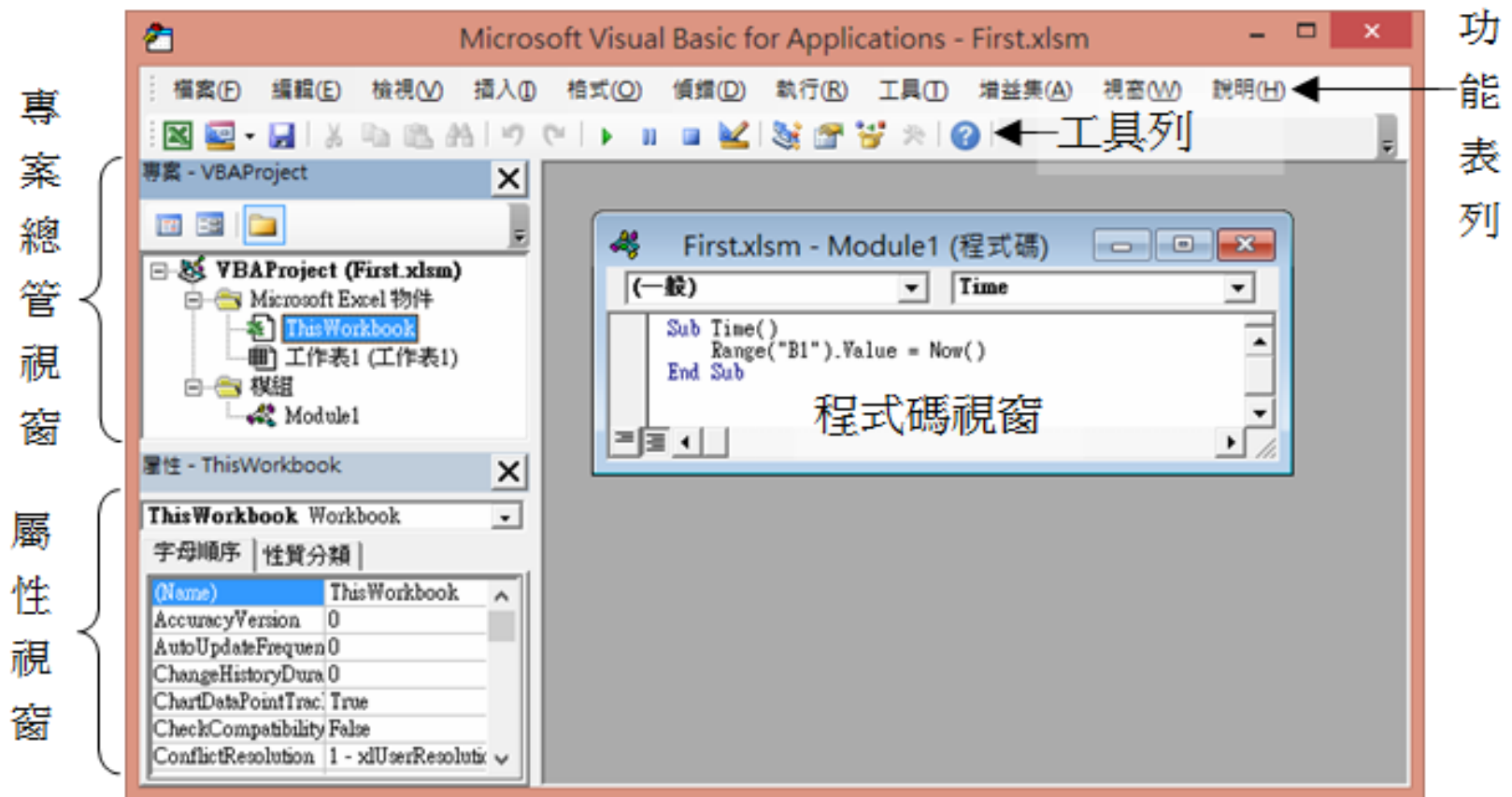
隨堂測驗

增加一個Clear巨集，使用Cells格式將B1儲存格的內容清空，並將該巨集指定給 **清空時間** 按鈕。

	A	B	C	D	E	F
1	目前時間：		顯示時間		清空時間	
2						

2.3 Excel VBA 程式編輯器

開啟Excel VBA程式編輯器



功能表與工具列

- 功能表列：
 - VBA 編輯視窗的功能都分門別類，集中在功能表列內供使用者選用。
- 工具列
 - VBA 編輯視窗的**主要功能以圖示鈕**的形式，集中在工具列內供使用者快速選用。



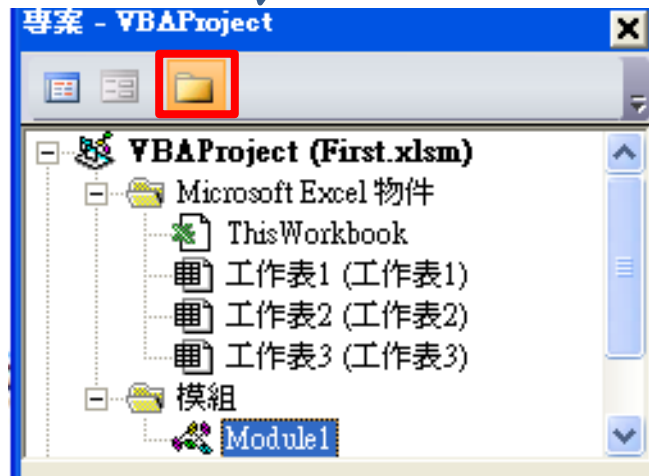
專案總管視窗

- 每個Excel活頁簿檔案視為一個專案(Project)
- 視窗內用階層方式來顯示專案中的所有項目

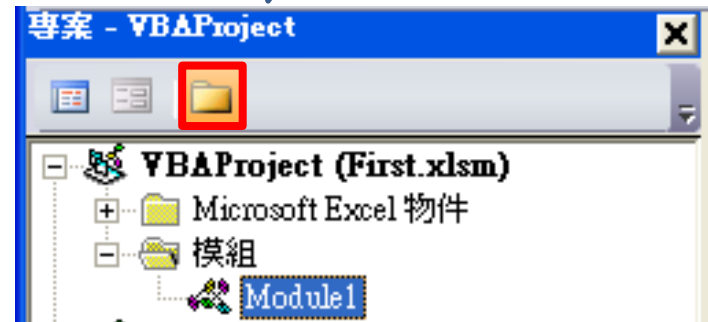


顯示或隱藏物件資料夾

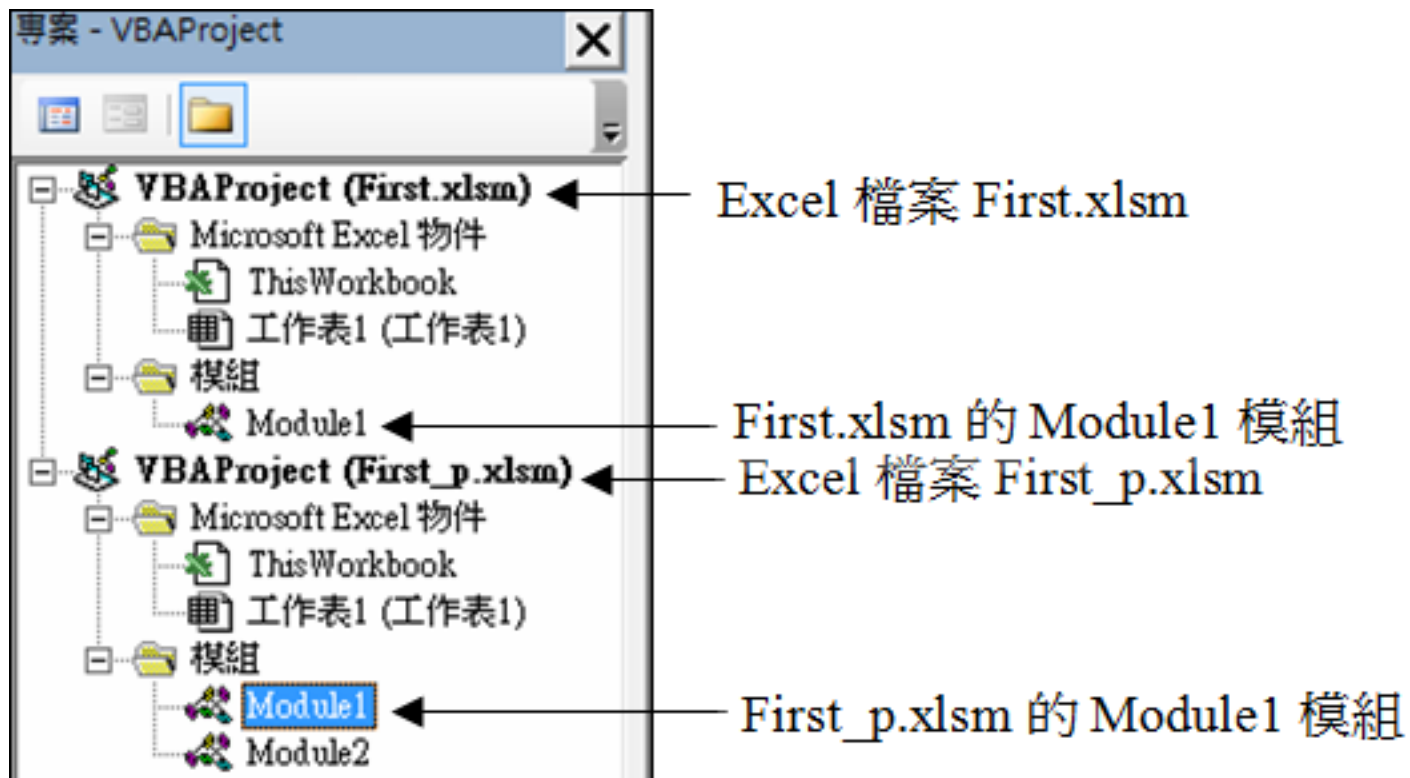
顯示物件資料夾



隱藏物件資料夾



開啟多個Excel活頁簿檔案時，專案總管視窗中會以樹狀圖清楚呈現階層關係。



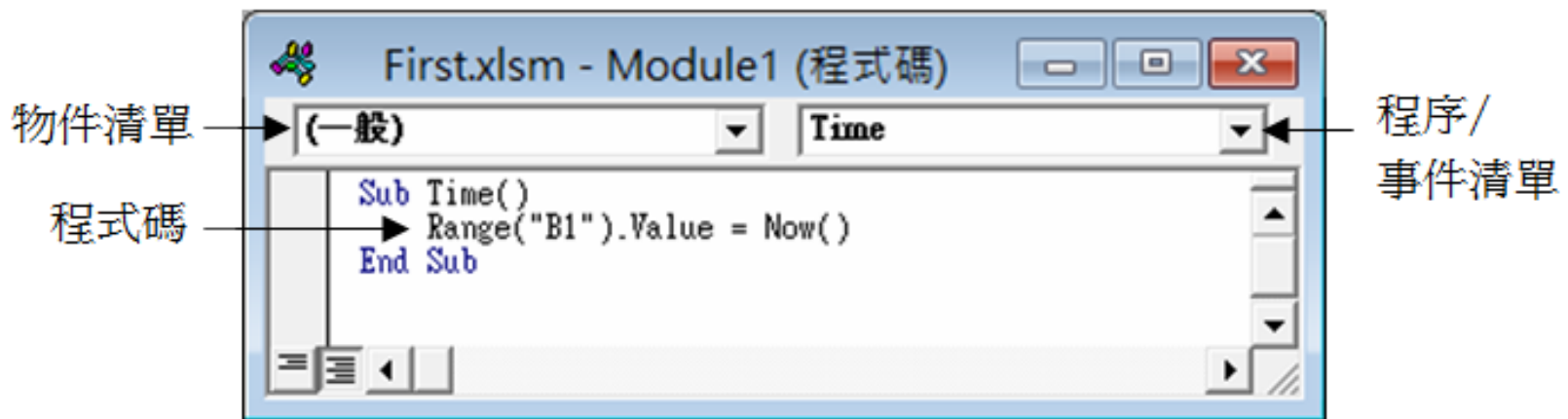
屬性視窗

- 屬性視窗可以設定物件的各種屬性值
- 屬性視窗會隨點選不同的物件，而顯示不同的屬性清單。

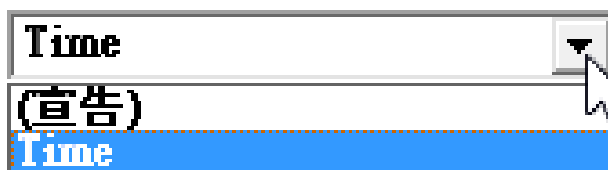


程式碼視窗

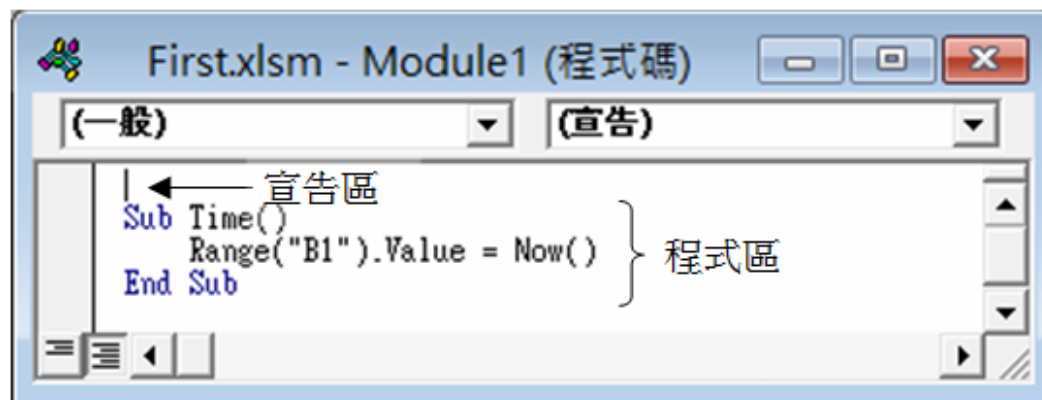
在專案總管視窗內的物件上快按兩下，
就會開啟該物件的程式碼視窗。



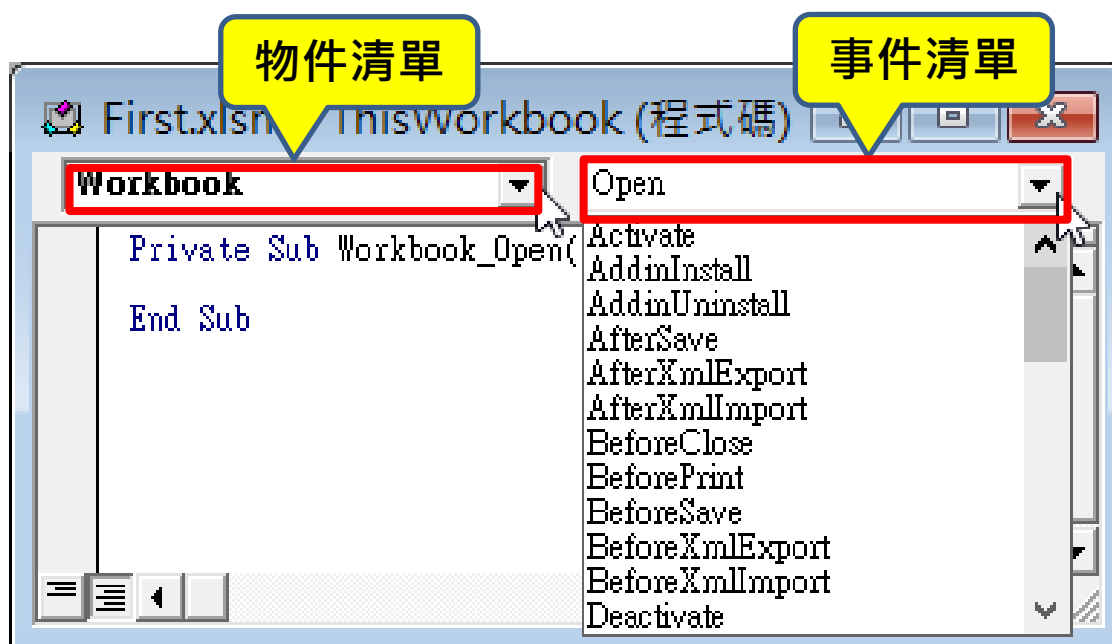
1. 按物件清單的下拉鈕，清單中有(一般)和所屬物件等項目。
2. 如果物件清單是選(一般)項目就是要編寫一般的程序，此時右邊的清單會列出程序的項目。



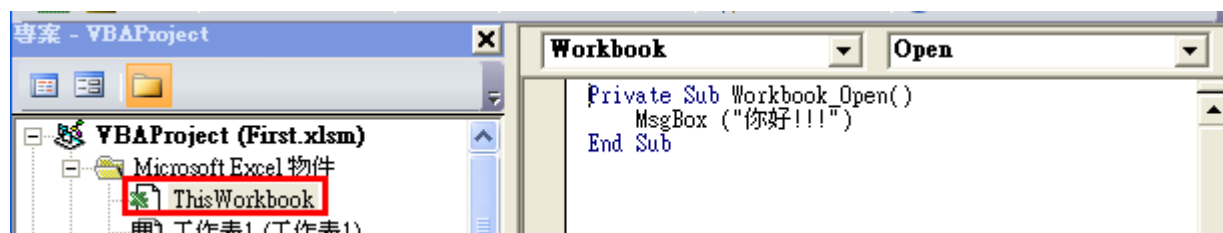
3. 程序清單中有(宣告)和所屬程序等項目，點選(宣告)項目會進入宣告區，可以宣告共用的變數等程式碼。



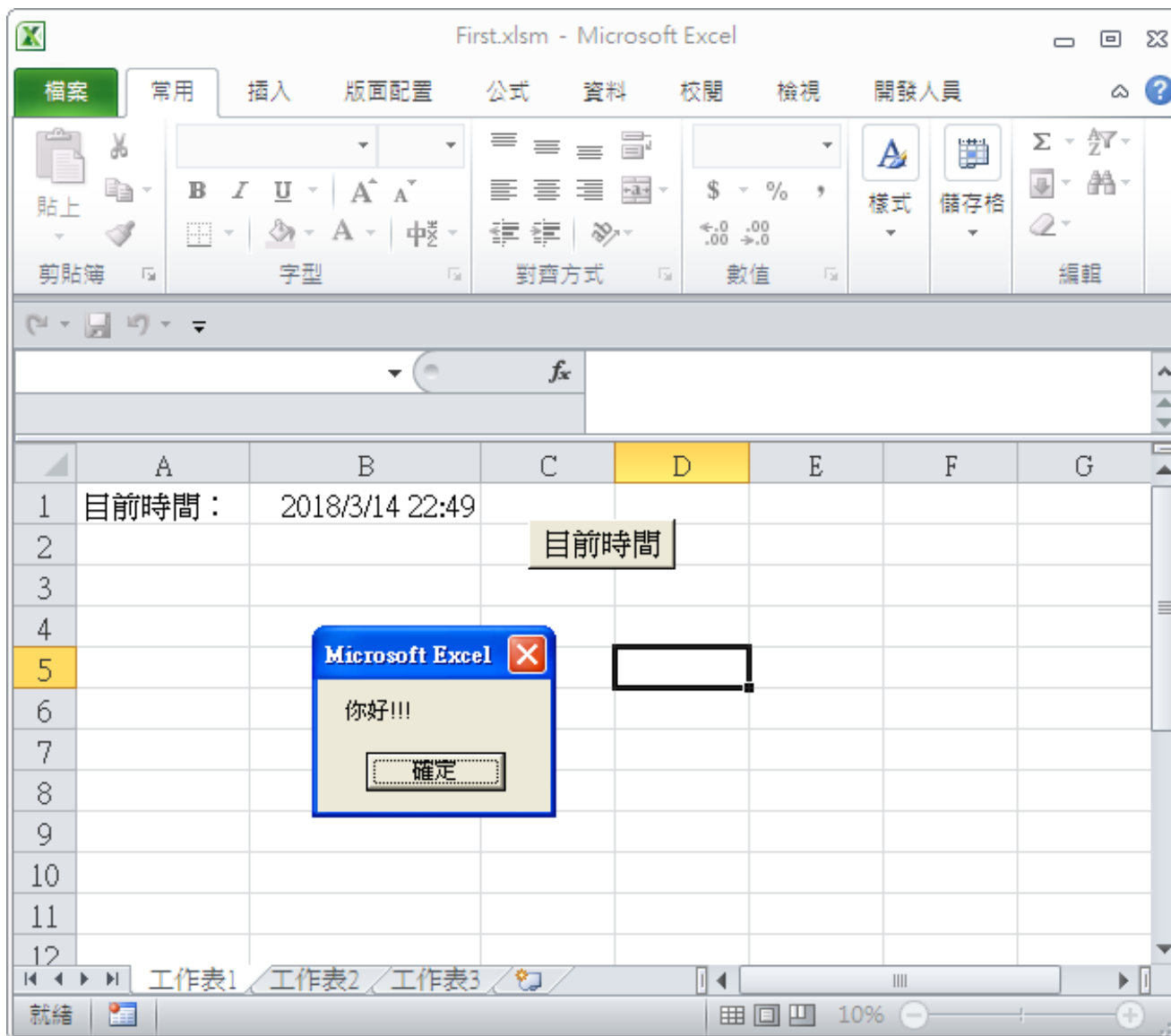
4. 如果物件清單是選物件項目時，此時右邊的清單會列出該物件的事件項目，此時可以編輯物件的事件程序。下圖為Workbook的事件程序：



MsgBox("你好!!!")



重新開啟 First.xlsm → 觸發Workbook_Open()事件



2.4 VBA 程式的編輯和除錯



實作 FileName : Second. xlsx

設計一個兩個儲存格數值相加的巨集程序，將 B1 和 B2 儲存格內的數值相加，將相加結果顯示在 B3 儲存格。

	A	B
1		5
2	+	10
3		15

操作步驟

- 建立工作表內容
- 新增程序
- 撰寫程式
- 測試程序
- 程序除錯

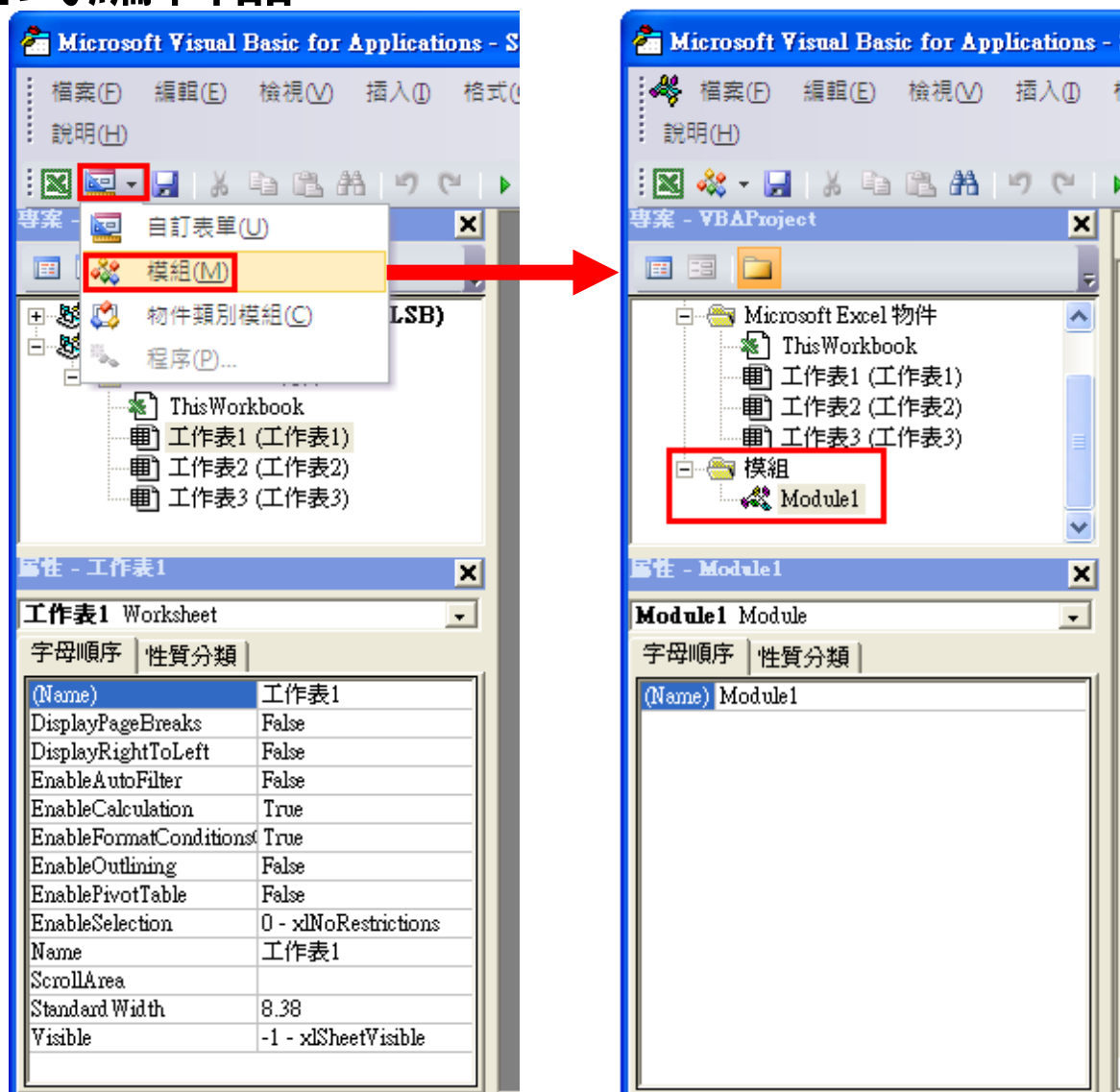
1.建立工作表內容

- 新增一個空白活頁簿，並在名稱為「工作表1」工作表上建立如下的內容：
- 以Second檔名，「Excel啟用巨集的活頁簿」的存檔類型來存檔，儲存後檔名為Second.xlsm。

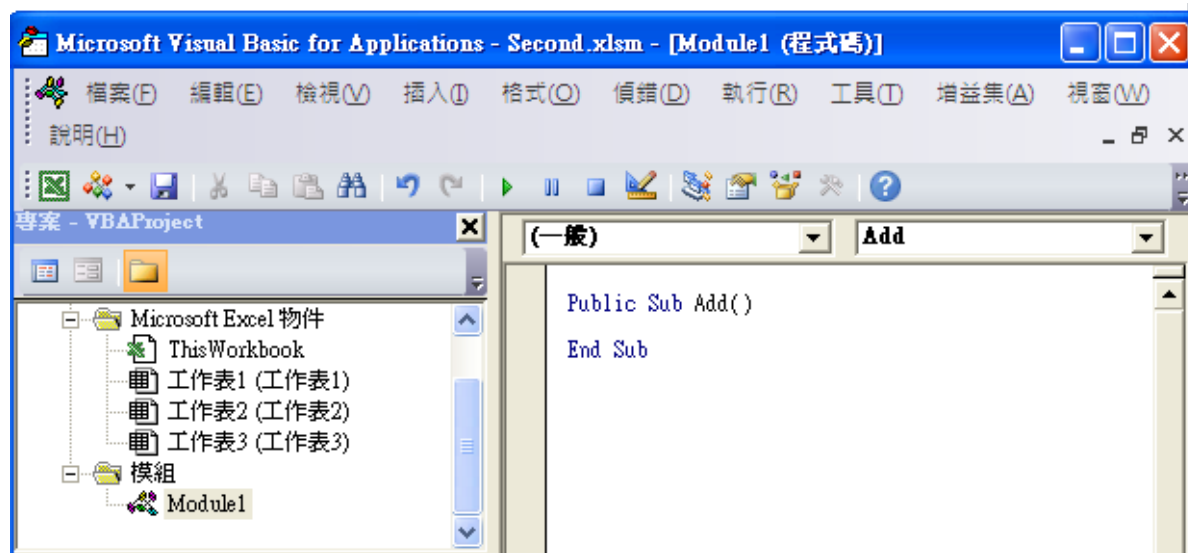
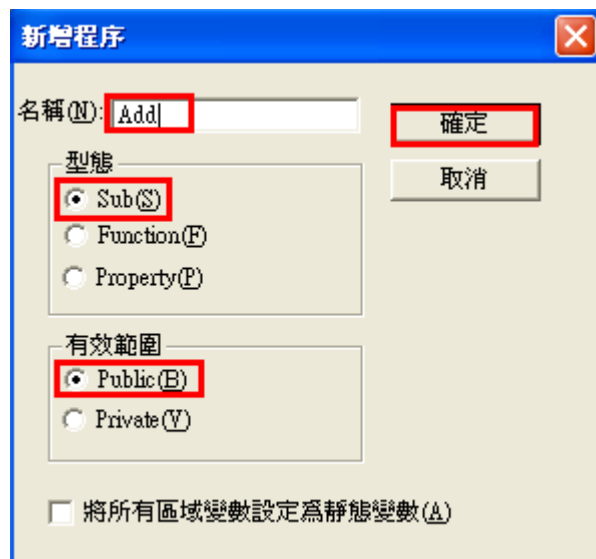
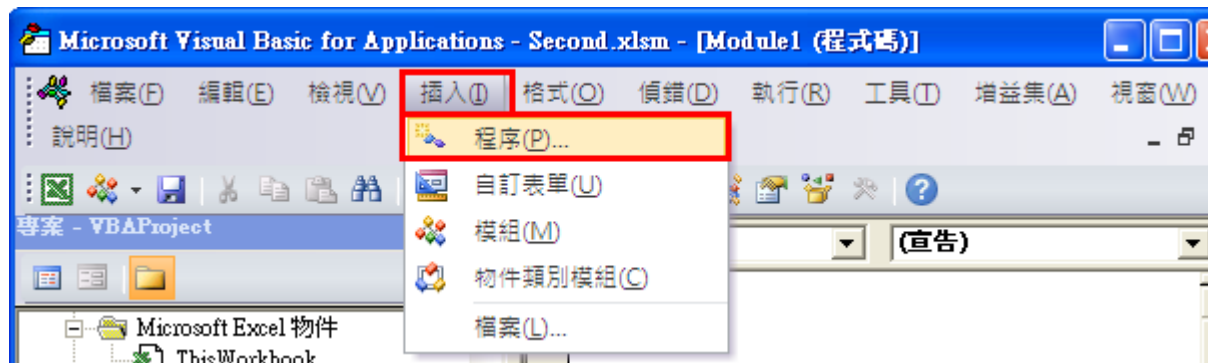
	A	B
1		5
2	+	10
3		

2.新增程序

- 開啟 VBA 程式編輯器。
- 新增模組。



新增程序



3. 撰寫程式

- 先在「Public Sub Add()」的下一行輸入「a = rang("B1).value」後，然後按下 <Enter> 鍵完成第一行程式。

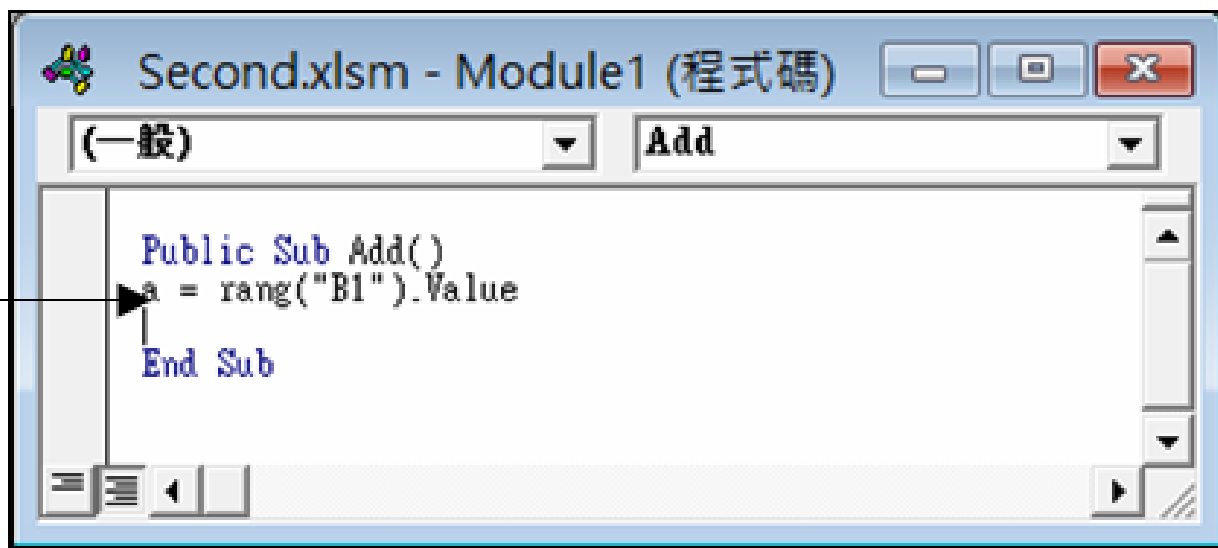
程式碼輸入錯誤，所以系統會將整行程式改成紅色，並出現對話方塊提醒錯誤。



3. 撰寫程式

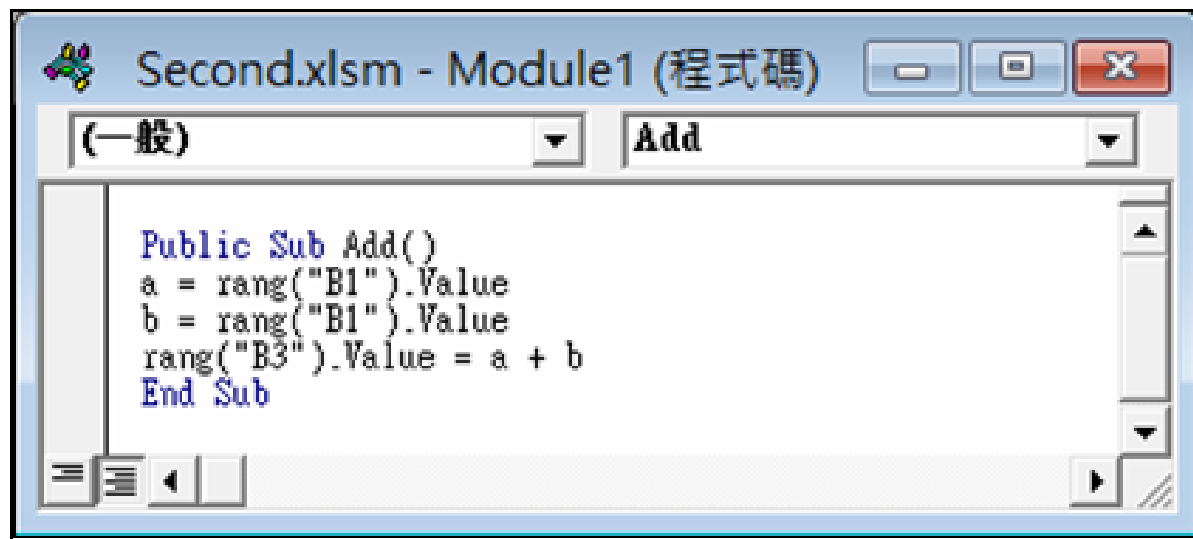
- 我們在「("B1" 後面補上「" 」，程式改為「a = rang("B1").value」後，字體變成黑色就完成第一行程式。
為了示範程式除錯，程式碼故意將range誤植為rang。

黑色字



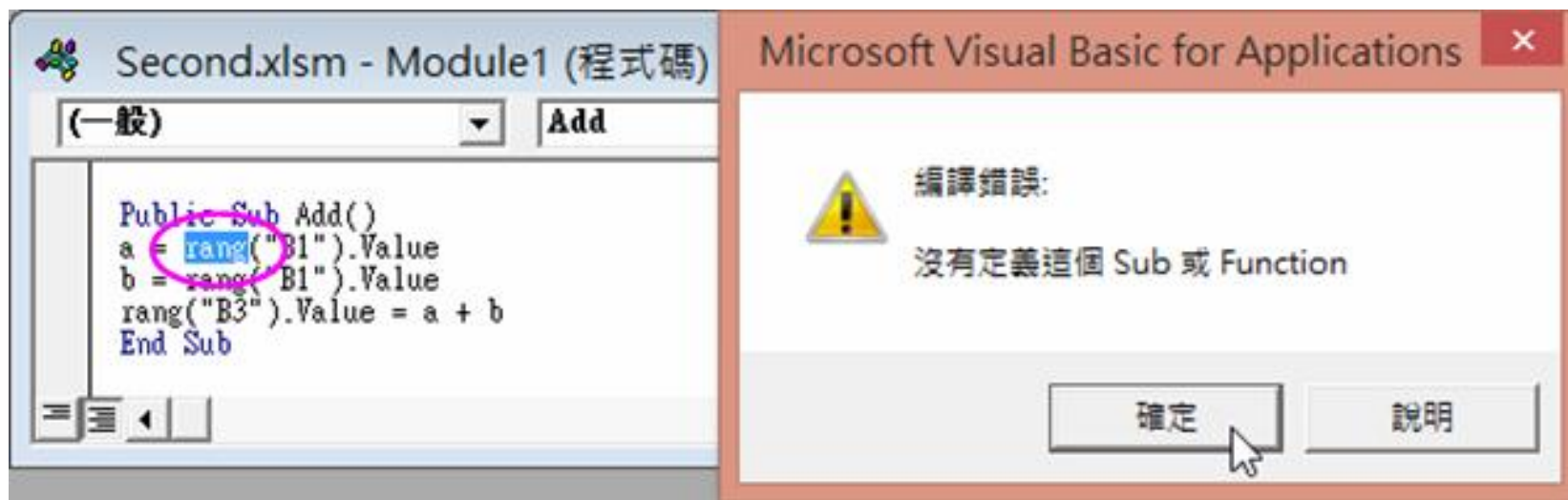
3. 撰寫程式

- 繼續輸入下列兩行的程式碼，就完成Add程序的輸入。
 - 第1行：將B1儲存格的值指定給變數a。
 - 第2行：將B2儲存格的值指定給變數b。此處為了示範程式除錯，程式碼故意誤植為"B1"。
 - 第3行：將B3儲存格的值設為變數a + 變數b。



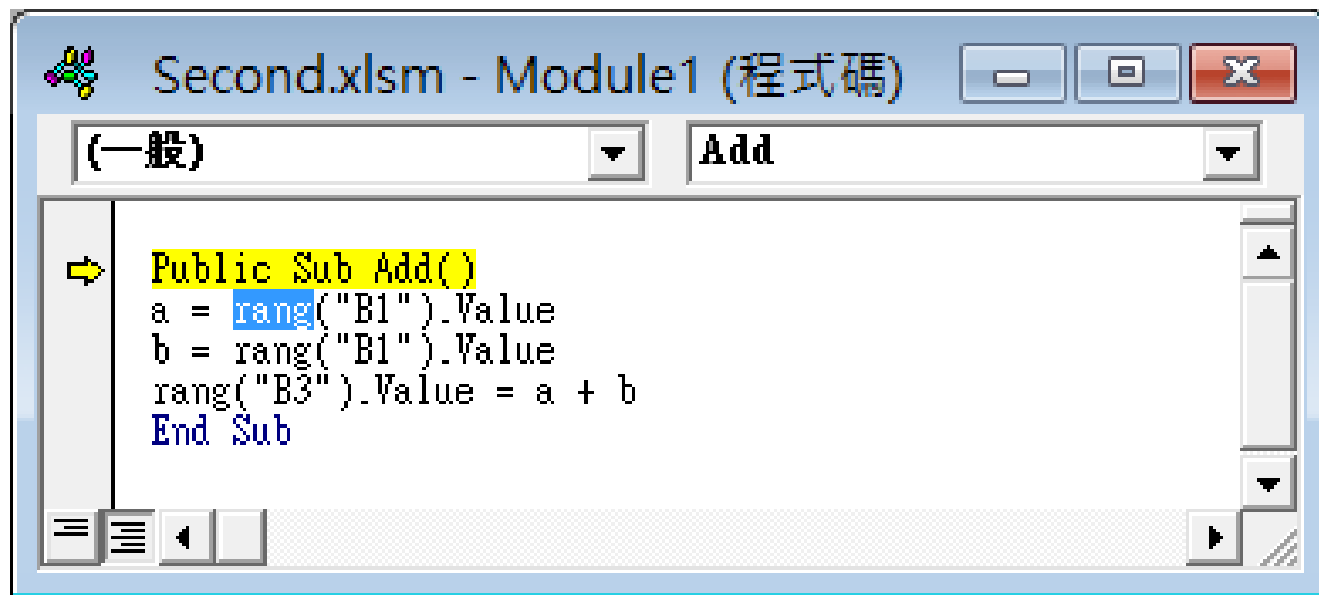
4.測試程序

- 先點按Add程序使插入點在程序內。然後點按工具列中的或直接按 **<F5> 鍵**，就可以執行Add程序。
- 因為**程式碼有錯**所以會反白「rang」中斷程式執行，並提醒錯誤，此時我們按 確定 鈕，重回程式編輯環境。




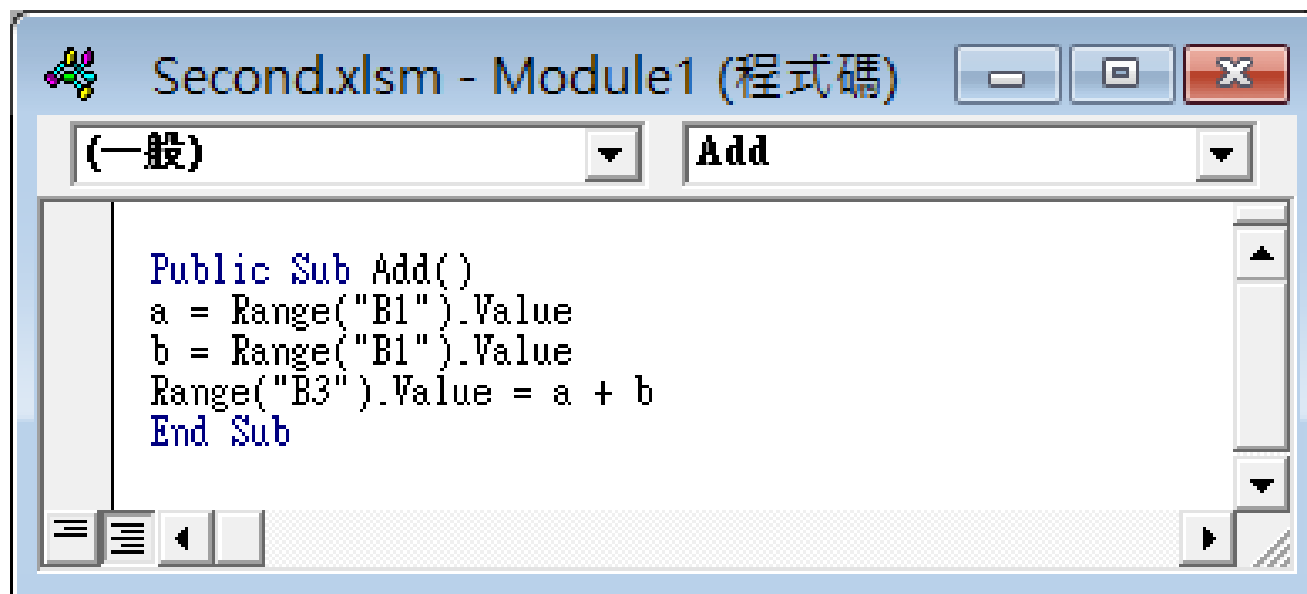
4.測試程序

- 此時Add程序會加上黃色網底，並在前面出現黃色箭頭，表示該程序在執行階段發生錯誤，無法繼續執行目前進入**中斷模式**。



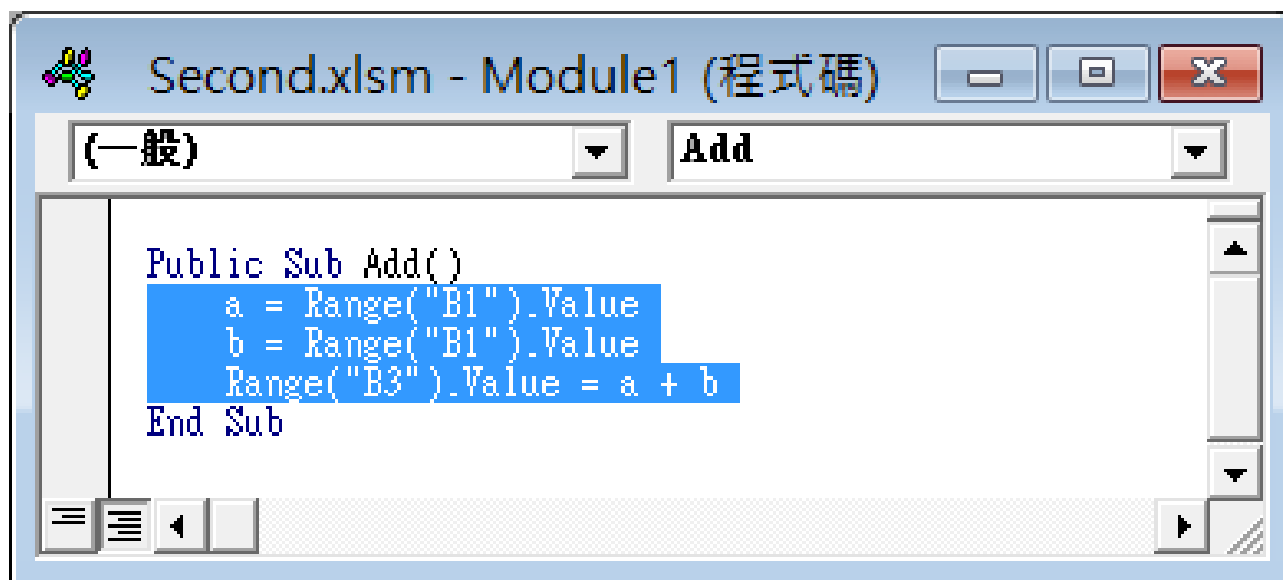
四.測試程序

- 點按工具列的  停止圖示鈕，回編輯模式。
檢查程式碼發現「rang」沒變成大寫，因為輸入錯誤。
將三行程式修正如下：



4.測試程序

- 選取程序內三行程式，然後執行執行功能表【編輯/縮排】指令或按 **<Tab>** 鍵設定縮排。將三行程式修正如下：



The screenshot shows a VBA editor window titled "Second.xlsm - Module1 (程式碼)". The window has a menu bar with "(一般)" and "Add". The code editor contains the following VBA code:

```
Public Sub Add()  
    a = Range("B1").Value  
    b = Range("B1").Value  
    Range("B3").Value = a + b  
End Sub
```

The three lines of code are highlighted in blue, indicating they are selected. The window also features a scroll bar on the right and a status bar at the bottom.

4.測試程序

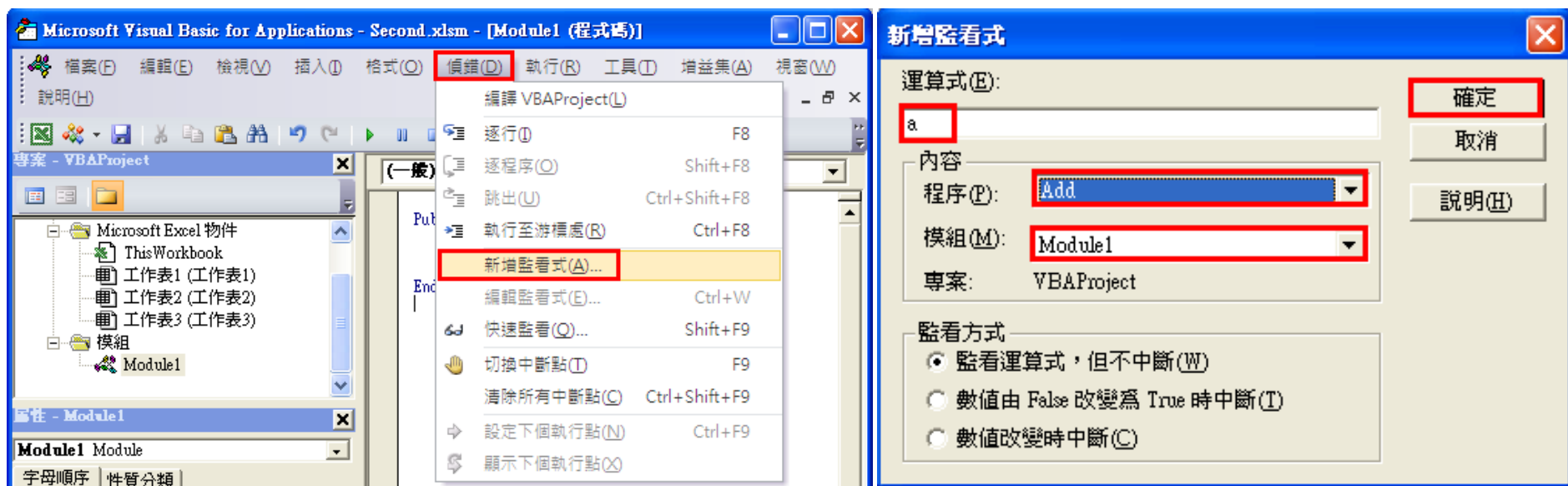
- 程式碼修正完畢後，在重新執行Add程序，此時可以執行完畢沒有發生錯誤。
在「工作表1」可以看到執行結果：

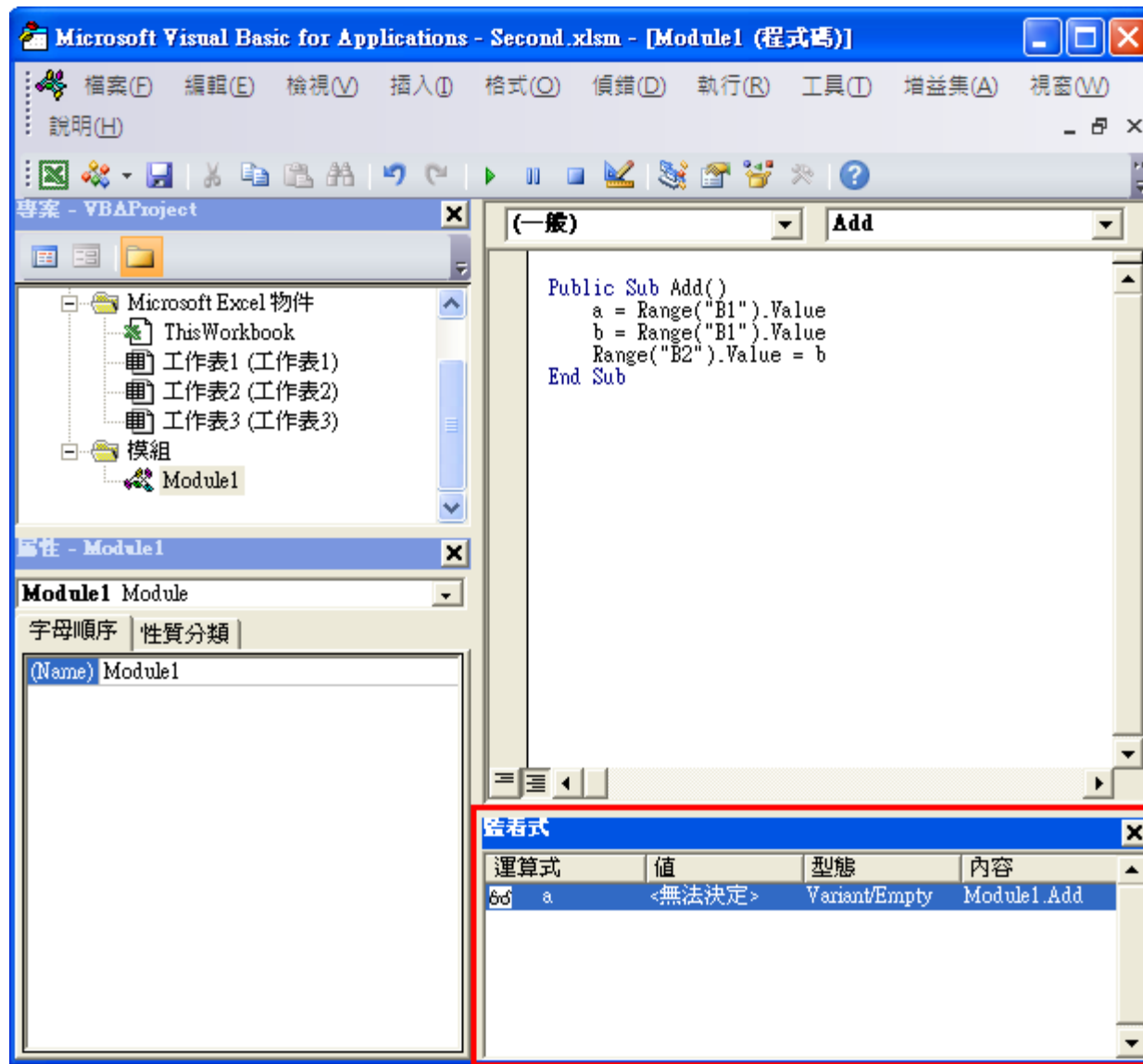
	A	B	C
1		5	
2	+	10	
3		10	

5. 程序除錯

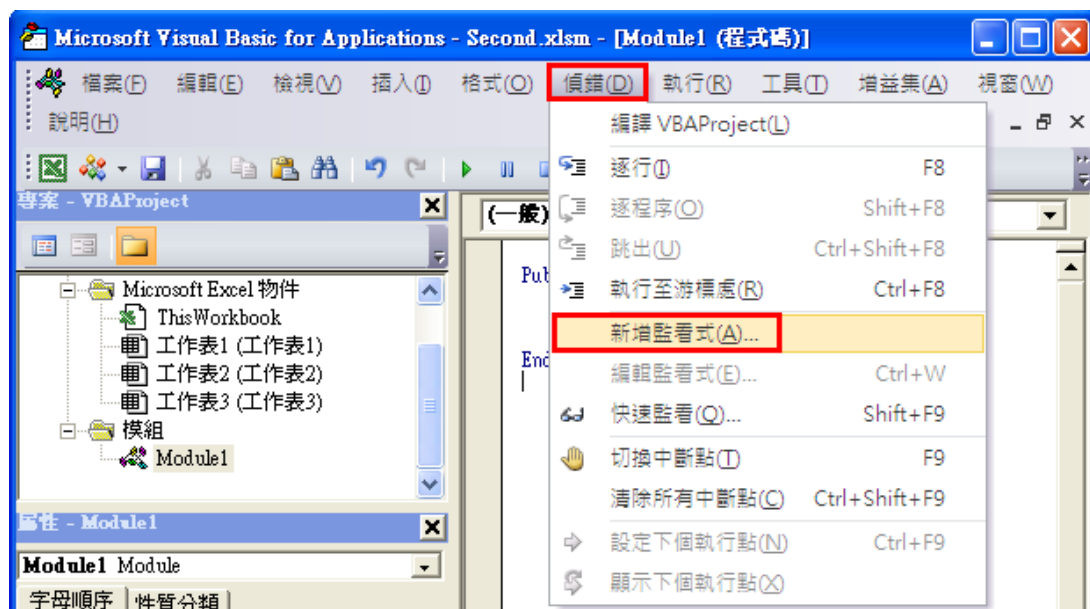
雖然上面的Add程序可以執行，但是執行結果是錯誤
(5 + 10 = 10?)。此時需要使用一些除錯(Debug)方法。

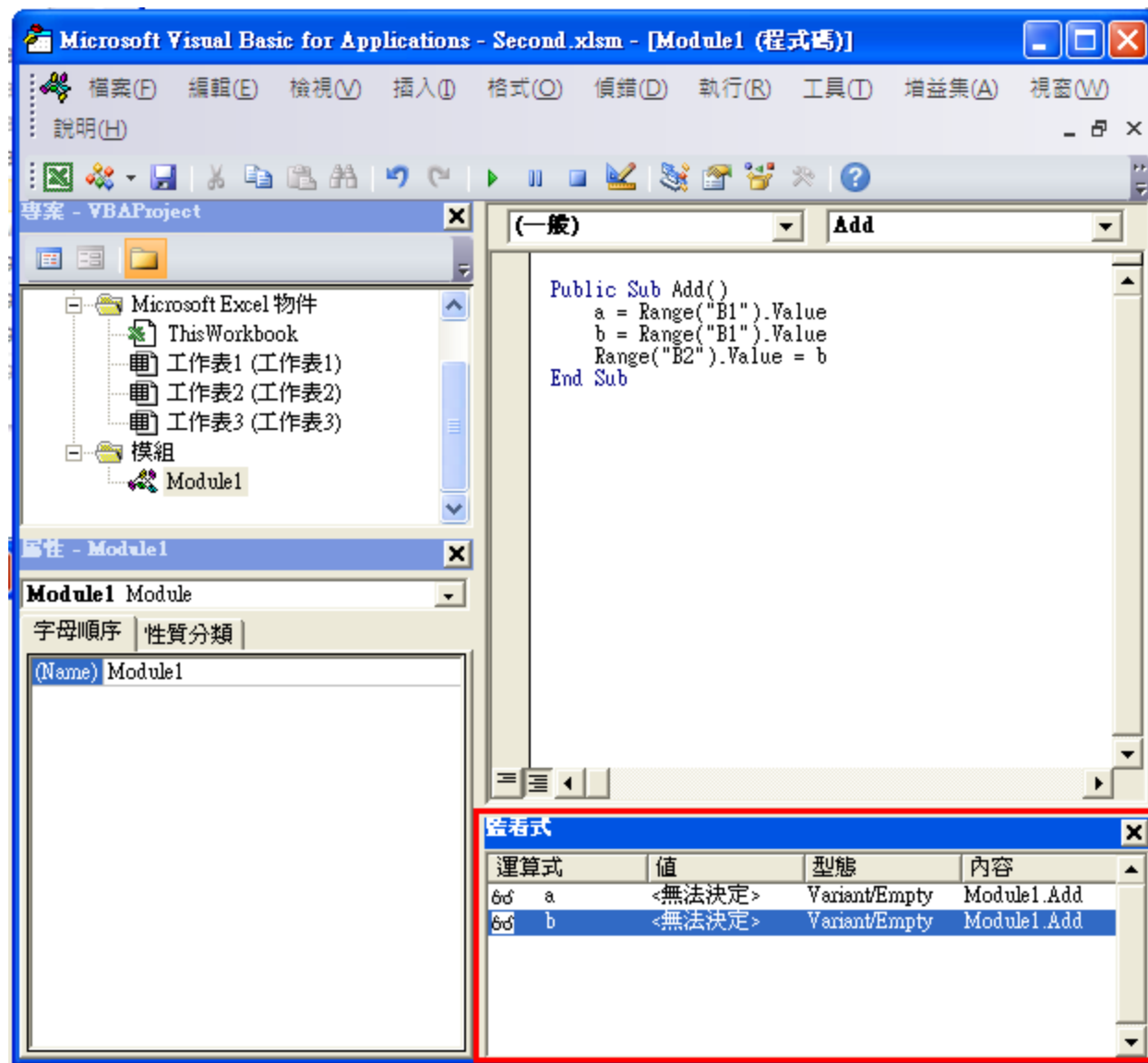
- 新增監看式a：





● 新增監看式b：



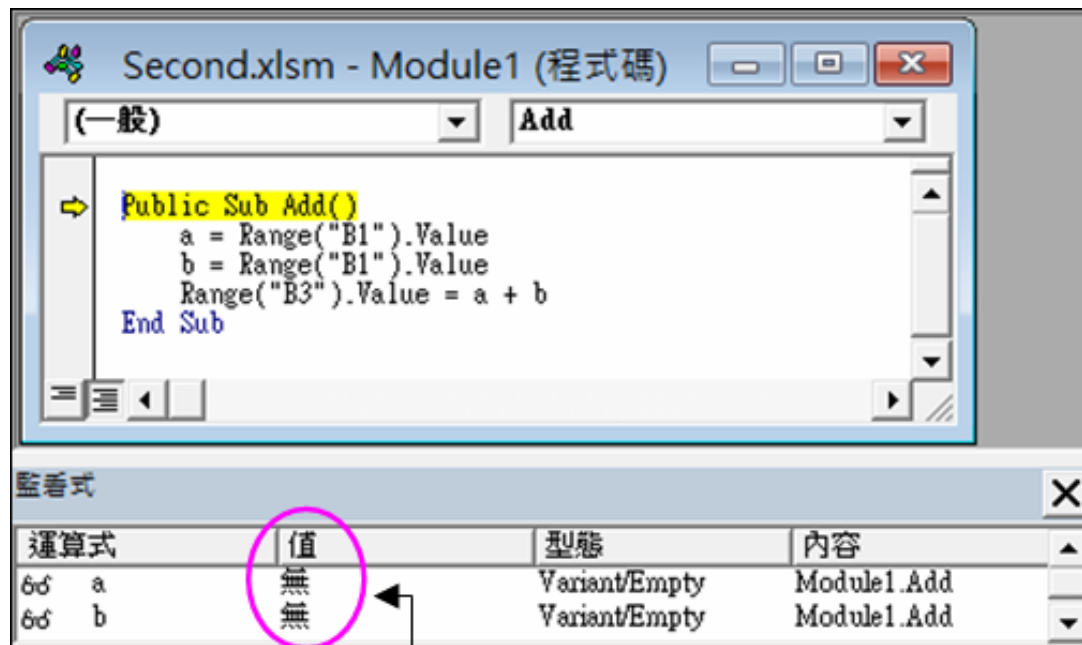


5. 程序除錯

● 逐行執行程式：

游標置入 `Sub Add()` 中，執行功能表【偵錯/逐行】指令，或直接按 **<F8>** 快速鍵。

此時程式會逐行執行，由監看式視窗中觀察監看式的變化。



變數值尚未指定，為預設值

觀察到變數b值不等於B2儲存格的值。

Second.xlsm - Module1 (程式碼)

(一般) Add

```
Public Sub Add()  
    a = Range("B1").Value  
    b = Range("B1").Value  
    Range("B3").Value = a + b  
End Sub
```

監看式

變數 b 值不等於 B2 儲存格的值 X

運算式	值	型態	內容
6d a	5	Variant/Double	Module1.Add
6d b	5	Variant/Double	Module1.Add

程式修正如下：

FileName: Second.xlsm (Module1 程式碼)

01 Public Sub Add()

02 a = Range("B1").Value

03 b = Range("B2").Value

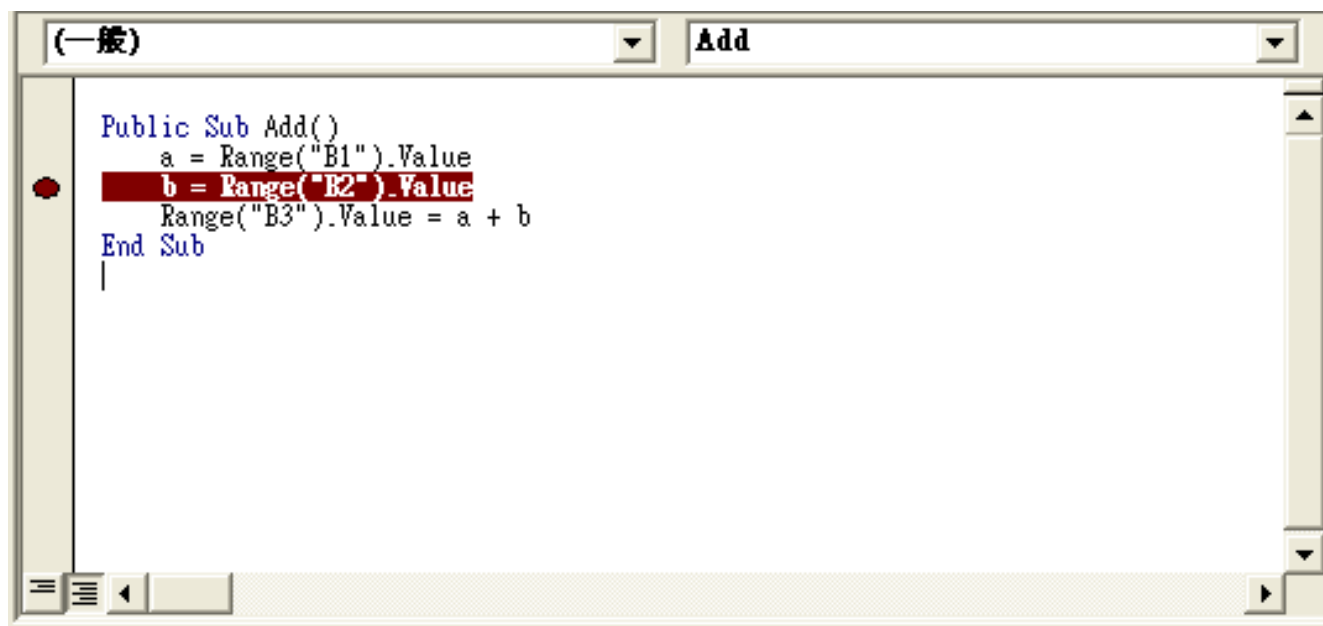
04 Range("B3").Value = a + b

05 End Sub

5.程序除錯

● 設定中斷點：

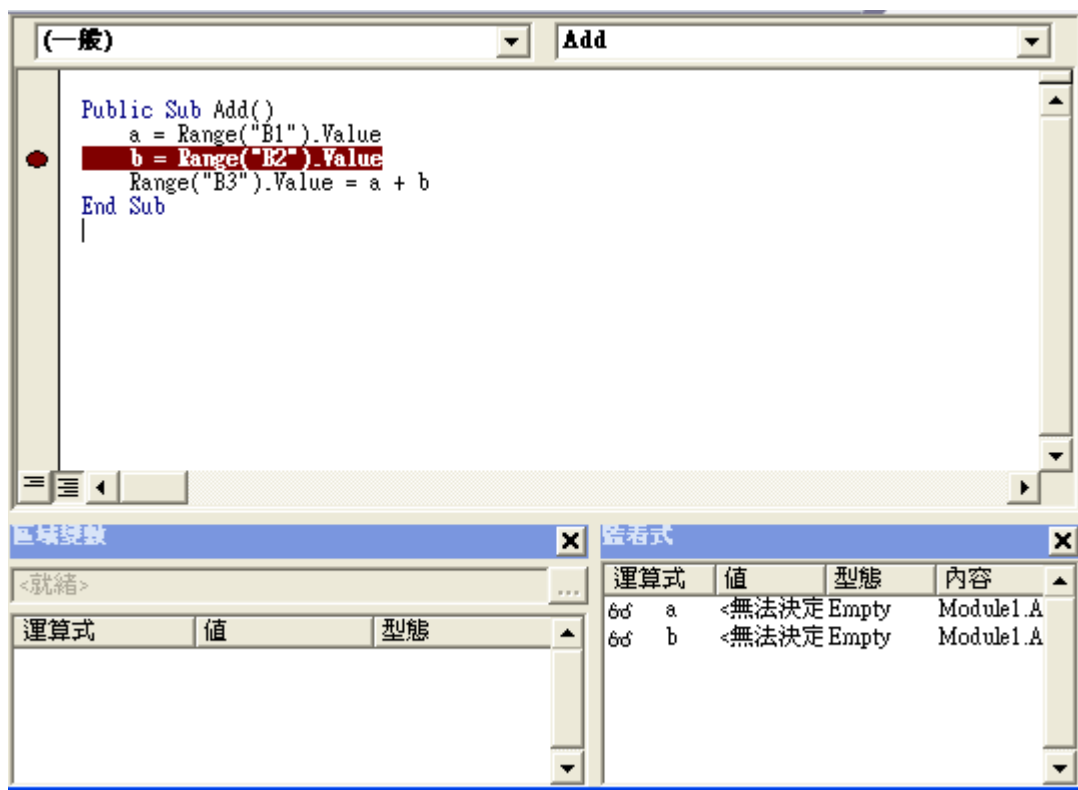
執行【**偵錯/切換中斷點**】指令，或按 **<F9>** 快速鍵，在插入點所在程式設立中斷點。



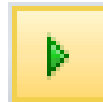
5. 程序除錯

- 區域變數視窗：

可執行【**檢視/區域變數視窗**】指令，開啟區域變數視窗。



游標置入 Sub Add() 中→



Microsoft Visual Basic for Applications - Second.xlsm [中斷] - [Module1 (程式碼)]

檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 偵錯(D) 執行(R) 工具(T) 增益集(A) 視窗(W) 說明(H)

專案 - VBAProject

- VBAProject (PERSONAL.XLSB)
- VBAProject (Second.xlsm)
 - Microsoft Excel 物件
 - ThisWorkbook
 - 工作表1 (工作表1)
 - 工作表2 (工作表2)
 - 工作表3 (工作表3)

屬性 - Module1

Module1 Module

字母順序 性質分類

(Name) Module1

執行 Sub 或 UserForm (F5) Add

```
Public Sub Add()  
    a = Range("B1").Value  
    b = Range("B2").Value  
    Range("B3").Value = a + b  
End Sub
```

區域變數

運算式	值	型態
Module1		Module1/Module1
a	5	Variant/Double
b	無	Variant/Empty

監視式

運算式	值	型態	內容
a	5	Variant/Double	Module1.A
b	無	Variant/Empty	Module1.A

5.程序除錯

- **MsgBox**函數：

在程式適當地方用MsgBox函數顯示要觀察的變數值。例如程式碼修改如下：

```
a = Range("B1").Value  
MsgBox(a)
```

由執行結果可以知道執行第一行程式後，變數a的值為5。



MsgBox函數

語法：

`MsgBox Prompt [, Buttons] [, Title]`

⇨ 沒有傳回值

`Result = MsgBox(Prompt [, Buttons] [, Title])`

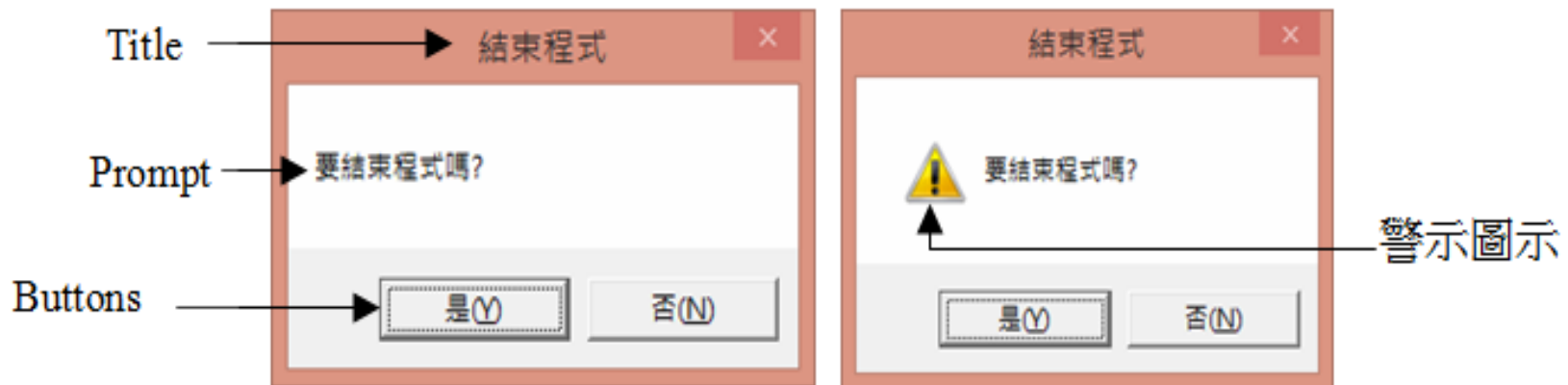
⇨ 有傳回值

1. **Prompt**引數是指定對話方塊內顯示的文字
2. **Buttons**引數是指定對話方塊的格式
3. **Title**引數是指定標題文字。

MsgBox函數

[例] 詢問是否要結束，程式寫法為：

```
MsgBox "要結束程式嗎?", 4, "結束程式"      '使用代碼  
MsgBox "要結束程式嗎?", vbYesNo, "結束程式"  '使用常數  
MsgBox "要結束程式嗎?", vbYesNo + vbExclamation, "結束程式" '顯示警示圖示
```



MsgBox函數

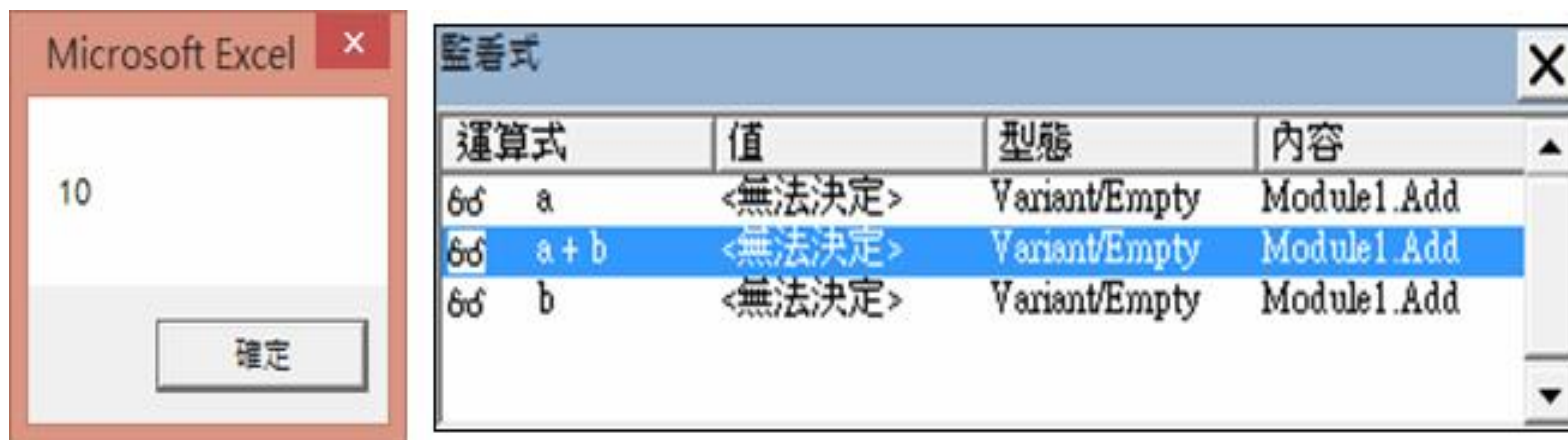
如果希望得知使用者**按下哪一個按鈕**，以便程式做不同的反應時，應該使用第二個語法。

[例] 按 是 鈕時，用Quit方法關閉Excel應用程式，程式寫法為：

```
Dim rst As Integer  
rst = MsgBox("要結束程式嗎?", vbYesNo, "結束程式")  
If rst = vbYes Then Application.Quit
```

► 隨堂測驗

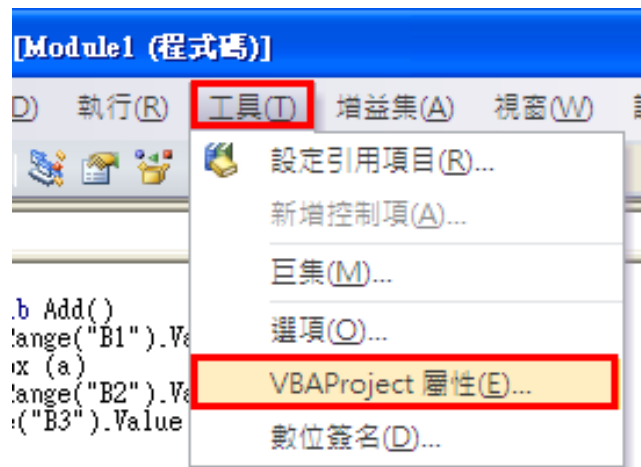
在上面實作中增加使用MsgBox函數，來顯示變數b的值。在編輯器中再新增一個a+b的監看式，用逐行方式觀察變數值的變化情形。



2.5 設定Excel VBA 專案的保護

要保護專案不被更動和程式碼保密，就要設定專案的密碼。設定步驟如下：

1. 在VBA 編輯視窗的功能表列中，執行【工具/VBA Project屬性】指令，此時會開啟「VBAProject - 專案屬性」對話方塊：



2. 設定專案保護密碼時應該謹慎，萬一忘記密碼將會很麻煩。



1

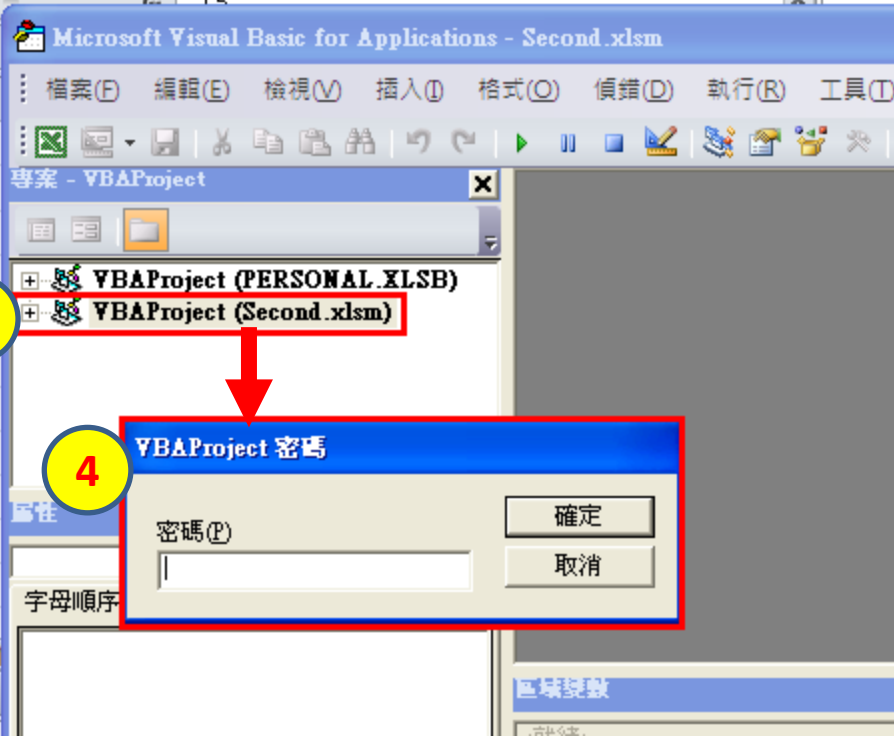


Second.xlsm
Microsoft Excel 啓用巨集的工作表
15 KB

2

Visual Basic

3



4