

第二章

現金流量與折現

第一節 現金流量與現值

第二節 Excel VBA 的計算

第三節 變數與資料型態

第四節 折現率與複利頻率

第五節 Excel VBA 的敘述與運算

第六節 Excel VBA 的程式編輯環境

投資一項金融資產，代表未來將擁有一連串的現金流量。例如，股票投資人將可收到未來的股利，與賣出股票的資本利得。債券投資人則可收到未來的債息，與到期時的本金返還。然而，如果我們想要去比較不同的投資標的時，則需要一個一致的基礎，否則這些不同時點的現金流量之間，便無法進行比較。這個一致的基礎，便是將所有的現金流量，轉換成目前的現金價值，此過程稱之為折現(Discounting)。事實上，一個金融工具的價值，就是它所有未來現金流量的目前現金價值的總和。

第一節 現金流量與現值

假設目前銀行一年期的存款利率為 6%，如果你投資（購入）此定期存單 100 元，則一年後你可以收到本利和 106 元，

$$100 \times (1 + 6\%) = 106 \dots\dots\dots(2.1.1)$$

換一個角度看，此項投資一年後帶給你 106 元的現金流量，但是目前的價值是 100 元，這是因為你可以在市場上用 100 元購入此定期存單。因此，隱含的折現率為 y 可由下式求得，

$$\frac{106}{(1 + y)} = 100 \dots\dots\dots(2.1.2)$$

求得 $y = 6\%$ 。在此案例中， $y = 6\%$ 為折現率，100 元為現值，106 元為一年後的現金流量。

(2.1.2)式可以改寫為

$$100 = \frac{1}{(1 + 6\%)} \times 106 \dots\dots\dots(2.1.3)$$

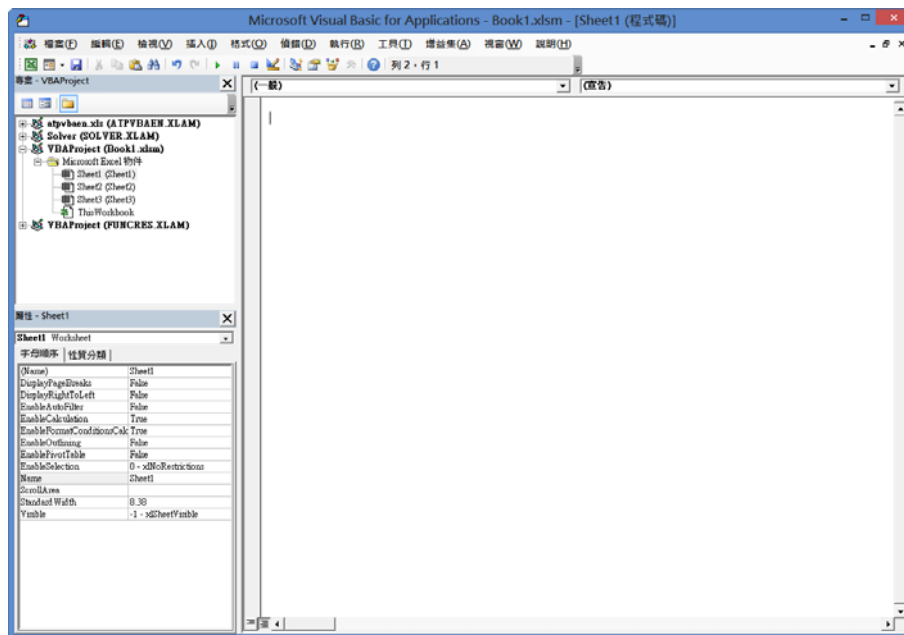
其中，等式右邊的第一項，可以看成是一年後的一元，目前的現值，我們將之稱為折現因子(Discounting Factor)。由於市場上的計息慣例，折現因子的計算因投資期間而有不同。我們將在第三節在進一步說明。

(2.1.3)式的一般化的表示式，可寫為

$$PV = CF_t \times DF_t \dots\dots\dots(2.1.4)$$

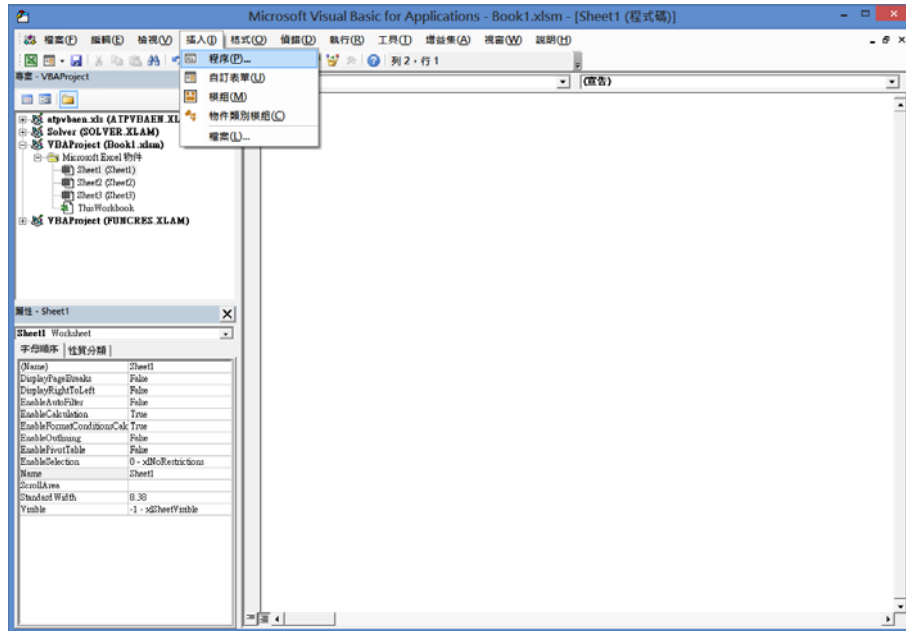
第二節 Excel VBA 的計算

本節中，我們將初步說明 Excel 撰寫程式的步驟，並以前一節(2.1.3)式的數值做為實例來示範。打開 Excel 程式，按 **Alt** + **F11** 進入 VBA 的編輯環境，如下圖所示，

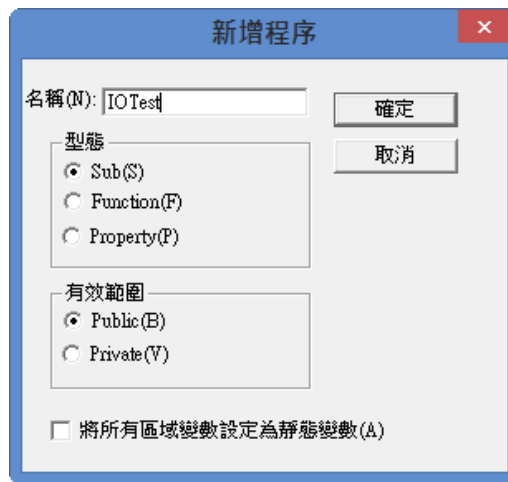


點選左上方《專案總管》中的《Sheet1(Sheet1)》，右方出現《Sheet1(程式碼)》編輯區。如果沒有看到《專案總管》視窗，可由功能選單中勾選【檢視／專案總管】打開視窗，左下方《屬性視窗》也可由功能選單中勾選【檢視／屬性視窗】打開視窗。

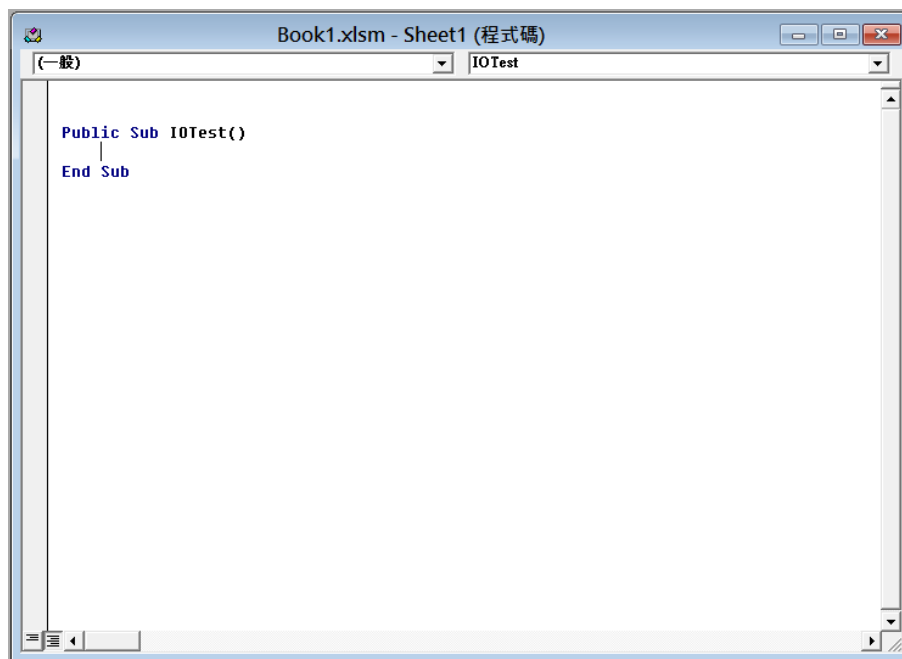
如下圖，由功能選單中勾選【插入／程序】，



出現如下的對話視窗，在名稱中打入“IOTest”，中間不要有空格，直接按【確定】，



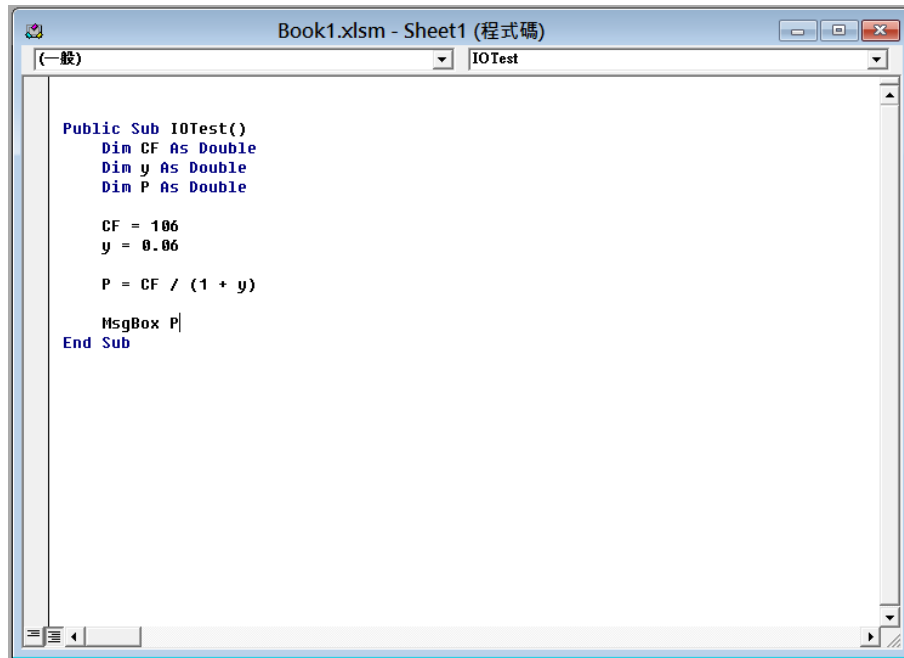
VBE 自動產生下面的程式碼，



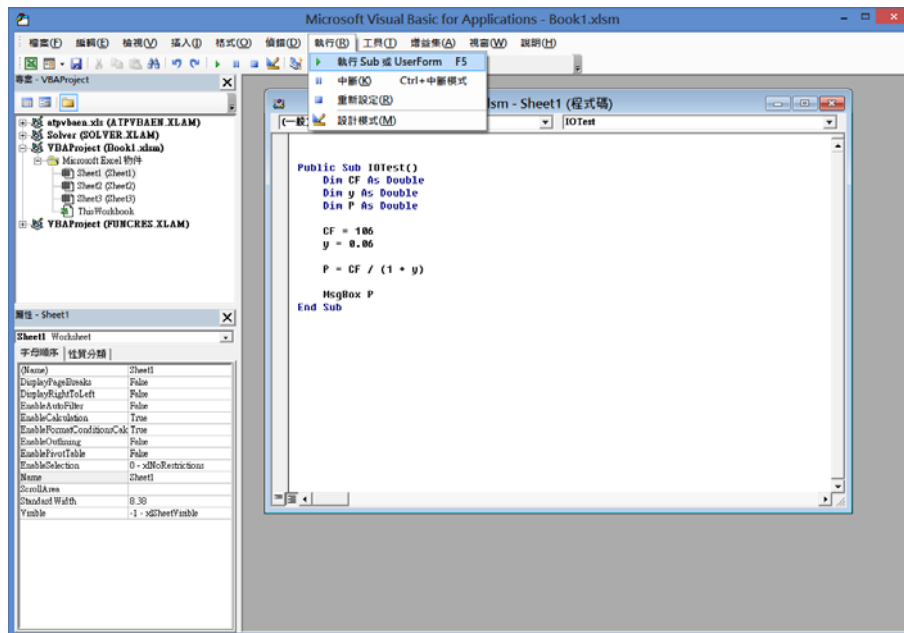
由 Public Sub IOtest() 到 End Sub 間，便是我們撰寫程式的地區。視窗左上方的文字“一般”，表示此程序是由使用者產生的一般目的程序，視窗右上方的文字“IOtest”，表示此程序的名稱。由視窗的正上方標題，顯示這是在 Book1 活頁簿中，Sheet1 試算表中的程式碼。

如下圖，輸入下面的程式碼，

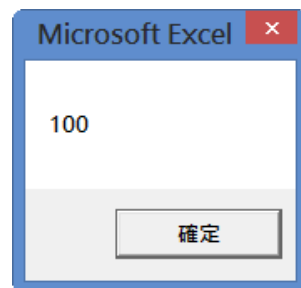
```
#01 Public Sub IOtest()  
#02     CF = 106  
#03     y = 0.06  
#04  
#05     P = CF / (1 + y)  
#06  
#07     MsgBox P  
#08 End Sub
```



將游標放在 Public Sub IOtest() 到 End Sub 間，如下圖勾選功能選單中【執行／執行 Sub 或 UserForm】，或直接按 **F5** 快捷鍵。



執行後，出現下面的對話盒，顯示出現值 100 元。



這正是(2.1.3)式的結果。

第三節 變數與資料型態

在第二節的 IOTest 程序中，我們一共輸入了六行程式，其中兩行空白，因此有效的程式只有四行，重列程式碼如下，

```
#01 Public Sub IOTest()  
#02     CF = 106  
#03     y = 0.06  
#04  
#05     P = CF / (1 + y)  
#06  
#07     MsgBox P  
#08 End Sub  
程式 2.3.1
```

#02 行中我們以 CF 代表一年後的現金流量，並將之設定為 106。#03 行中我們以 y 代表一年期的折現率，並將之設定為 6%。#04 行中我們以 P 代表此一年後的現金流量，目前的現值，並使用(2.1.3)式計算之，其間使用 CF 與 y 執行計算。#07 行中我們利用 VBA 的內建函數 MsgBox，將計算的現值以對話盒顯示出來。

本節中，我們將以此範例程式說明 VBA 程式中，關於變數與資料型態的規範，並再以較為優良的撰寫方式改寫此程式，希望讀者不但會寫計算正確的程式，還要會寫好的程式。以 IOTest 為例，適度的空行與行首的縮排，在程式的可讀性與未來的維護上，都有很大的助益。

一、VBA 的變數

在程序的處理或計算的過程中，可能需要將某些內容暫時儲存起來，作為其它處理程序的資料來源。例如，程式 2.3.1 中的現金流量、折現率與現值。而這些暫存的資料內容，可以用一個名稱來代表，如 CF、y、P，這讓我們更加容易使用那些資料內容。這些暫存的資料內容可以是數值、文字、或日期等。

如果以程式的觀點來說，當程序中需要存放一些暫時的資料時，必須先在

記憶體中找到一塊沒有被使用的區域，來儲存這些暫時的資料內容，而這一塊區域可以用一個名稱來代表，這個名稱就是所謂的變數名稱(Variable Name)，或稱為“變數”(Variable)。變數的內容可能隨著程序的執行而改變，也就是說變數的內容是變動的，並非固定不變。

二、變數命名

變數名稱的命名與檔案名稱的命名都有一定的規則，如果不依照此一規則來命名，可能會發生錯誤。變數名稱的命名規則如下：

- 變數名稱可以是中文字、英文字、或中英文混合，它的長度不可超過 255 個字元(一個中文字佔二個字元)。
- 變數名稱的字首不可以是 0~9 + - * / @ ! \$ % . 等特殊字元。
- 變數名稱可以包含底線“_”，但不可包含空白。
- 變數名稱不可以使用關鍵字。所謂的“關鍵字(Keyword)”、“保留字(Reserved Word)”就是在程式語言或作業系統內已經指定用途之字句，不可用來做為變數名稱，否則將造成錯誤。
- 變數名稱最好以能夠代表該資料意義文字來命名，如此可提高變數的可讀性。例如：您可以使用 TraderNum 來代表交易員代號的變數名稱，比使用 Num 來表示更清楚明瞭。
- 如果變數名稱內包含英文字，字母的大寫與小寫都是代表同一個變數。例如：使用 CF、Cf、或 cf 都表示同一個變數。

三、變數的宣告

下列所示內容是正確的命名方式：

```
Dim CallPutFlag As String
Dim CashFlowAmt as Single
Dim Maturity As Date
Dim RecordNum As Long, CahFlowNum as Long
```

下列所示內容是不正確的命名方式：

```
Dim #TraderName    變數名稱的第一個字為特殊字元
Dim @TraderName    變數名稱的第一個字為特殊字元
Dim TraderName*    變數名稱的最後一個字為特殊字元
```

宣告變數的語法：您可以在程式中使用下列的方式來宣告變數：

■ Dim 變數名稱〔As 資料型態〕

Dim 變數名稱〔資料型態宣告字元〕

■ Public 變數名稱〔As 資料型態〕

Public 變數名稱〔資料型態宣告字元〕

■ Static 變數名稱〔As 資料型態〕

Static 變數名稱〔資料型態宣告字元〕

■ Private 變數名稱〔As 資料型態〕

Private 變數名稱〔資料型態宣告字元〕

在〔〕內的部份是可以省略的，而當省略〔〕中的內容時，這個變數會被宣告成變動型態(Variant)的變數，它的內容可以是數值、文字、日期、物件等。變數名稱可以與資料型態的宣告字元搭配使用，在表 2.3.1 中將第二種宣告變數的語法列出來。

到目前為止，我們的宣告都以”Dim 變數名稱”的方式執行。至於這些方式的差異與始用時機，因涉及到變數的有效範圍(Effective Scope)與生命週期(Life Period)，我們將在後面再加以說明。

表 2.3.1 變數宣告與資料型態

資料型態名稱	資料型態 宣告字元	說 明
Boolean		Dim ExpFg As Boolean
Integer	%	Dim Connter As Integer Dim Counter%
Long	&	Dim Total As Long Dim Total&
Single	!	Dim Rate As Single Dim Rate!
Double	#	Dim DF As Double Dim DF#
Currency	@	Dim NPA As Currency Dim NPA@
String	\$	Dim CallPutFg As String Dim CallPutFg\$
Date		Dim Maturity As Date
Variant		Dim Temp As Variant
Object		Dim DataRange As Object

在宣告變數時，使用資料型態（例如：Integer、String）或資料型態的宣告字元（例如：%、\$）時，都可以明確指定變數的資料型態。

四、VBA 的常數

常數 (Constant) 不同於變數，因為變數的內容可能因為程序的執行而改變，但是常數的內容並不會因為程序的執行而變動。在 VBA 中，常數可以分為內建常數 (Build In Constant) 與使用者自訂常數 (User-defined Constant) 二種。

(一)內建常數

內建常數是 Excel 基於自身設計的需要，並便於程式設計人員使用上的便利所設定的。依照內建常數所屬的程式庫來區分，大概可以分為下列三種：

- Excel 內建常數：以“xl”開頭，例如：xlAverage、xlOn。
- VBA 內建常數：以“vb”開頭，例如：vbYes、vbYesNo。
- MsForm 內建常數：以“fm”開頭，例如：fmIMEModeOn、fmIMEModeOff。

(二)使用者自訂常數

在程序的設計過程中，有些資料的內容可能會固定不變。例如：我們可能由外部讀入 1,000 筆現金流量來計算現值。因此，在程式中這 1,000 筆的數目便固定不變。但也可能是一些數學數值，有一定的大小，是不可能改變的，如圓周率 π 就是固定不變。在程序內，您可以使用有意義的名稱，來表示這些固定的內容，而在處理過程中，您只需要使用這些事先指定的名稱，來參照這些固定的內容就可以了。

常數命名規則與變數的命名規則相同，只是它的宣告語法不同於變數宣告語法，常數宣告的語法是：

Const 常數名稱 [As 資料型態] = 常數值

Const 常數名稱 [資料型態宣告字元] = 常數值

Public Const 常數名稱 [As 資料型態] = 常數值

Public Const 常數名稱 [資料型態宣告字元] = 常數值

例如：

```
Const PI As Double = 3.1415926
Const MaxCFNum As Long = 1000000
```

如果沒有明確宣告常數的資料型態，這個常數的資料型態將會被視為與常數值最相近的資料型態。例如：

```
Const PI = 3.1415926           'PI 屬於 Double 資料型態
Const MaxCFNum = 1000000       'MaxCFNum 屬於 Long 資料型態
```

五、VBA 的資料型態

表 2.3.2 中列出一些 Excel VBA 中所提供的資料型態 (Data Type)、以及資料內容的有效範圍。由於不同的資料型態將佔用不同的記憶體大小，因此選用不同的資料型態，可能會影響到記憶體的分配及執行運算處理的速度，所以在資料型態的選用上，必須明確而且審慎。

- 當數值型態的資料被轉換成 Boolean 資料型態時，只有 0 會被轉換成 False，其他非 0 的值將被轉換成 True
- 當 Boolean 型態的資料被轉換成數值型態時，False 會被轉換成 0，True 則被轉換成-1
- Currency 資料型態是一小數點之前最多可以顯示 15 位數字，而小數點後面則只能顯示 4 位數字

- 固定長度的字串變數所佔記憶體的大小，是根據該字串變數所宣告的長度為準(1 Byte／字元)。可變動長度的字串變數所佔記憶體的大小，是根據該字串變數內容的長度，再加上 10 Bytes。

表 2.3.2 資料型態與有效範圍

資料型態名稱	資料型態宣告字元	佔記憶體大小(Bytes)	資料內容有效範圍
Boolean		2	True / False
Integer	%	2	-32,768 – 32,767
Long	&	4	-2,147,483,648 – 2,147,483,647
Single	!	4	負值：-3.402823E38 ~ -1.401298E-45 正值：1.401298E-45 ~ 3.402823E38
Double	#	8	負值：-1.79769313486232E308 ~ -4.94065645841247E-324 正值：4.94065645841247E-324 ~ 1.79769313486232E308
Currency	@	8	-922,337,203,685,477.5808 ~ 922,337,203,685,477.5807
String	\$	固定長度 可變動長度	0 ~ 65,535 (2 ¹⁶) Bytes 0 ~ 2 ³¹ Bytes
Date		8	日期 Jan 1,100 ~ Dec 31,9999 時間 0:00:00 ~ 23:59:59
Variant		視資料內容而定	其內容可是數值、字串、日期、物件

六、程式註解

在程序或函數裡加入適當的說明文字，可以提高程序或函數內容的可讀性。如果在程序或函數內容都沒有加入任何註解文字，當我們在修改程序內容時，可能需要再下一番功夫，才能了解程序的作用以及處理流程。事實上，只要我們在程序或函數內的適當位置加入註解文字，就可免除這種困擾了。

Visual Basic 語言提供了兩種在程序內加入註解文字的方式：

- **Rem 註解文字內容**

- **' 註解文字內容**

以上兩種加入註解文字的方式，都可以在程序內使用，但是它們二者之間還是有些差別的：

- 以 **Rem** 做為註解文字內容的開端時，這些註解文字與 **Rem** 之間必須空一格；但是 **'** 與註解文字之間就不需空一格。
- 如果註解文字需要加在指令敘述後面時，**Rem** 與指令敘述之間必須要用冒號(:)將它們分隔開。

Rem 的註解方式，是傳統 **Basic** 語言的方式，目前的使用情況則以 **"'**" 註解的方式，較為一般程式人員所使用。

下面的 **CommentTest** 程序，將前述的宣告以及如何在程序內使用註解文字，運用在之前的程式碼上，一併說明之。

```
#01 Public Sub CommentTest()  
#02     Dim CF As Double      'Cash Flow Amount  
#03     Dim y As Double       'Discount Rate  
#04     Dim P As Double       'Present Value  
#05  
#06     CF = 106  
#07     y = 0.06  
#08  
#09     P = CF / (1 + y)      'Discount One Year CF  
#10  
#11     MsgBox P              'Show the Result  
#12 End Sub
```

第四節 折現率與複利頻率

讓我們再回到第二節定存單投資的例子，假設在同一時點你可以選擇投資一年後到期的短期政府債券（國庫券，Treasury Bill），你覺得國庫券的報酬率會高於 6%嗎？事實上，國庫券的報酬率會低於 6%。因為即使是銀行，仍舊有經營的風險，一年後定存到期時，有可能會因經營不善而倒閉，造成本金與利息償還的問題。而投資國庫券則無此問題，因此投資人會要求定存單的報酬率，高於國庫券的報酬率。

假定一年後到期的國庫券報酬率為 5%，則這 1%的差額便是反應銀行可能倒閉的信用風險，亦即投資人需要額外的誘因，以承受可能的風險。我們可以下式表示一般投資的報酬率 r_i ，

$$r_i = r_f + RP \dots\dots\dots(2.4.1)$$

其中 r_f 表相同投資期限的政府債券報酬率，通常稱之為無風險報酬率(Risk Free Rate)； RP 稱為風險溢酬(Risk Premium)，表風險性投資報酬中高於 r_f 的差額。

一般金融工具風險溢酬的高低，反應著此工具信用風險的大小，通常信用評等機構(Rating Agency)會根據金融工具發行人或金融工具本身的信用情況，給予一個信用等級(Credit Grade)。市場上的投資人，會參酌信用等級，要求合理的風險溢酬。根據 Elton & al. (2001)所作的公司債研究，從 1987 到 1996 年 AA、A、BBB 的信用價差平均值，以五年期為例，政府債券持有到期的報酬率(Yield To Maturity, YTM)為 7.108%，AA 級信用等級的信用價差平均值為 0.493%，因此 AA 級債券持有到期的平均報酬率為 7.601%。由表 2.4.1 中可以看到 BBB 級的信用價差平均值，高於 A 級的信用價差平均值，A 級的信用價差平均值高於 AA 級的信用價差平均值。

表 2.4.1 信用價差與信用評等

Maturity	Treasuries	AA	A	BBB
2	6.414	0.414	0.621	1.167
3	6.689	0.419	0.680	1.205
4	6.925	0.455	0.715	1.210
5	7.108	0.493	0.738	1.205
6	7.246	0.526	0.753	1.199
7	7.351	0.552	0.764	1.193
8	7.432	0.573	0.773	1.188
9	7.496	0.589	0.779	1.184
10	7.548	0.603	0.785	1.180

Elton & al. (2001)

圖 2.4.1 則是 A 級公司債平均持有到期報酬率對到期日的作圖，從圖中可以看到持有到期報酬率與到期日有關，此關係稱之為利率的期限結構(Term Structure of Interest Rate)。應用到前述定存單的例子，六個月期的存款利率可能是 4%。因此，當我們在計算半年後到期的存款，現金流量的現值時，便要以 4%當作其折現率。

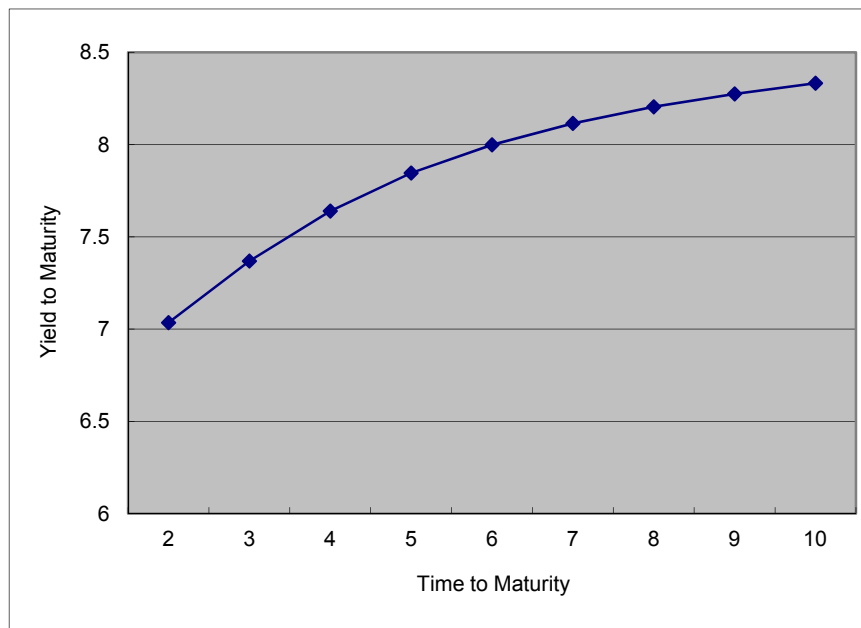


圖 2.4.1 A 級公司債平均持有到期報酬率

由於本書主要在討論選擇權契約，此類金融工具大都在一年內到期，而且六個月內到期的契約流動性也比較高，因此關於利率期限結構的討論，也集中在一年期以內的即期利率(Spot Rate)。所謂即期利率便是如(2.1.4)式中，用來計算未來單一現金流量現值的折現率。在金融市場上，每天都可以看到以銀行信用為擔保的一個月(30 天)、兩個月(60 天)、三個月(90 天)、六個月(180 天)的即期利率，至於九個月(270 天)以及一年(360 天)的利率，因流動性較差比較不具代表性，通常是由資本市場中的利率交換(Interest Rate Swap)報價利率，利用拔靴法(Bootstrapping)的方法，配合半年內的即期利率求得。

這裡所謂 30 天是代表一個月的意思，未必真的是 30 天後，需根據實際的情況判斷。例如，假定今天為 2004/10/15，則 30 天的即期利率代表折現 2004/11/15(31 天)現金流量的利率。假定今天為 2004/2/15，則 30 天的即期利率代表折現 2004/3/15 (29 天)現金流量的利率。表(2.4.2)為 2005/1/6 日，各天期的銀行信用即期利率。

表 2.4.2 短期票券次級市場利率

10 天	20 天	30 天	60 天	90 天	180 天
1.182	1.182	1.186	1.215	1.236	1.293

另外一個需要注意的問題是，金融市場上計算利息的頻率，會因到期日的長短，以及適用的工具而有所不同。通常到期日為一年以內的市場，稱之為貨幣市場(Money Market)，我們以單利(Simple Compounding)的方式折現。因此，(2.1.4)的折現因子計算如下，

$$DF_t = \frac{1}{\left(1 + y_t \times \frac{Act}{365}\right)} \dots\dots\dots(2.4.2)$$

其中 Act 表實際天數(Actual Days)。到期日為一年以上的市場，稱之為資本市場(Capital Market)，我們以複利(Compounded Compounding)的方式折現。

$$DF_t = \frac{1}{(1 + y_t)^{Act/365}} \dots\dots\dots(2.4.3)$$

需注意的是式(2.4.2)與(2.4.3)乃針對新台幣的折現方式，並不適用於美元。這是因為美元市場的慣例，是以一年 360 天來計息。

市場上針對一些特定的商品，主要是衍生性商品，其折現的概念乃是採取連續複利(Continuous Compounding)。所謂連續複利乃是將複利的週期由一般複利的一年，不斷的減少到一小時、一分鐘至一秒鐘，以此概念可將折現因子表

示為

$$DF_t = \lim_{n \rightarrow \infty} \frac{1}{\left(1 + \frac{y_t}{n}\right)^{n \times \frac{Act}{365}}} \dots\dots\dots(2.4.5)$$

根據數學上的定義，自然指數 e 的定義如下

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n \dots\dots\dots(2.4.6)$$

因此，(2.4.5)可以改寫成

$$DF_t = \lim_{\frac{n}{y_t} \rightarrow \infty} \frac{1}{\left(1 + \frac{1}{n/y_t}\right)^{n/y_t \times y_t \times \frac{Act}{365}}}$$

代入(2.4.6)自然指數 e 的定義，可以表示為

$$DF_t = e^{-y_t \times \frac{Act}{365}} \dots\dots\dots(2.4.7)$$

第五節 Excel VBA 的敘述與運算

一、VBA 的敘述

電腦程式是由一連串的敘述(Statements)所構成，再回到 IOTest 程序為例，#02、#03、#04、#06、#07、#09 與#11 行都是敘述，電腦由上而下，一行一行的執行這些敘述。其中#02、#03、#04 為宣告敘述，只涉及記憶體的配置，與資料的計算處理無關。其他行的敘述則由指定運算、算數運算或函數呼叫所構成。#06、#07 行為指定運算，#05 行為算數運算，#07 行則為函數呼叫。本節中，我們將針對 VBA 的敘述與運算，做進一步的說明，並撰寫第四節中各種折現的計算程式。

```
#01 Public Sub IOTest()  
#02     Dim CF As Double  
#03     Dim y As Double  
#04     Dim P As Double  
#05  
#06     CF = 106  
#07     y = 0.06  
#08  
#09     P = CF / (1 + y)  
#10  
#11     MsgBox P  
#12 End Sub
```

程式 2.5.1(Book1.xls)

VBA 的程式敘述，基本上就是在進行資料的處理。這些處理涉及對資料作特定的運算，所謂的運算子(Operator)就是指資料做某些特定的運算時，所需使用的符號或字組。至於運算元(Operand)則是被處理的資料。以#09 行為例，此行的敘述涉及加法與除法兩個運算。現金流量與折現率則是被處理的資料。因此，我們有“/”、“+”兩個運算元。加法運算的對象為“1”與“y”，除法運算的對象為“CF”與“(1+y)”。

如果依使用運算子的對象來區分時，可分為下列二種：

■ 數值運算子

■ 連結運算子

數值運算子，顧名思義是針對數值作指定運算的符號或字組。依它們的處理方式可細分為下列三類：

■ 算術運算子(Arithmetic Operators)

■ 比較運算子(Comparsion Operators)

■ 邏輯運算子(Logic Operators)

二、算術運算子

算術運算子可用來執行下列七種算術運算的工作（如下表）：

表 2.5.1 VBA 之算術運算子

運算子	說 明	表示方式	X=15,Y=2	結 果
^	指數	$X \wedge Y$	$15 \wedge 2$	225
*	乘法	$X * Y$	$15 * 2$	30
/	除法	X / Y	$15 / 2$	7.5
\	整數除法	$X \backslash Y$	$15 \backslash 2$	7
mod	餘數除法	$X \bmod Y$	$15 \bmod 2$	1
+	加法	$X + Y$	$15 + 2$	17
-	減法	$X - Y$	$15 - 2$	13

- 整數除法運算子(\)：這運算子在執行整數除法之前，會先將被除數及除數四捨五入成 Integer 或 Long 的資料型態，而運算結果的小數部份將被無條件捨去，所以傳回的結果是一個屬於 Integer 或 Long 資料型態的數值。
- 取餘數的算術運算子(Mod)：它運算的結果是傳回整數除法所得到的餘數。

三、比較運算子

比較運算子可用來比較數值資料或字串資料，它的比較結果有下列二種情況：比較結果成立時，傳回真（True 或 -1）；比較結果不成立時，傳回偽（False 或 0）。

比較運算子可分為下列八種：

表 2.5.2 VBA 之比較運算子

運算子	說 明	表 示 方 式		結 果
=	等於	X = Y	10 = 10	True 或 -1
<>	不等於	X <> Y	100 <> 10	True 或 -1
<	小於	X < Y	"Ab" < "AB"	False 或 0
>	大於	X > Y	"Ab" > "AB"	True 或 -1
<=	小於或等於	X <= Y	100 <= 10	False 或 0
>=	大於或等於	X >= Y	100 >= 10	True 或 -1

四、邏輯運算子

邏輯運算子可用來做邏輯值的運算，常用的邏輯運算子有 NOT、AND、OR 三種，而表 2.5.3 是實際使用的結果。

表 2.5.3 VBA 之邏輯運算子

		非	且	或
A	B	NOT A	A AND B	A OR B
F	F	T	F	F
F	T	T	F	T
T	F	F	F	T
T	T	F	T	T

- F (False)：表示變數內容是偽(False, 0)、或邏輯運算的結果不成立。
- T (True)：表示變數的內容是真(True, -1)、或邏輯運算的結果成立。
- 非(Not)：傳回與運算前相反的結果。
- 且(AND)：在邏輯運算中各元素都是真(True)的情況下，傳回的邏輯運算結果是真；只要其中有一個元素是偽(False)時，傳回的運算結果將是偽(False)。

如果 A 是 False，B 是 False 時，則 A AND B 是 False

如果 A 是 True，B 是 False 時，則 A AND B 是 False

如果 A 是 True，B 是 True 時，則 A AND B 是 True

- **或 (OR)：**在邏輯運算的元素中只要有一個是真(True)時，傳回邏輯運算的結果就是真(True)。

如果 A 是 False，B 是 True 時，則 A OR B 是 True

如果 A 是 False，B 是 False 時，則 A OR B 是 False

五、連結運算子

字串運算子又稱為連結運算子，Visual Basic 語言提供了下列二種連結字串的運算子：“+”與“&”。分別說明如下：

■ + 運算子

“+”運算子與數值運算子中的加法運算子一樣，因此在使用上可能會造成困擾，什麼時候應將它視為數值運算子中的加法運算子？什麼時候又該將它視為字串運算子？萬一運算的資料中具有屬於變異數型態的數值資料時，則又可能產生令人無法理解的結果。所以在選用字串結合運算子時，取好避免使用“+”運算子，而改用“&”字串運算子。

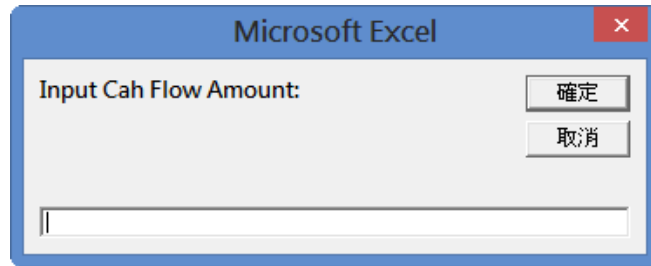
例如宣告 tmp1, tmp2, tmp3, tmp4 為變異數型態的變數

```
Dim tmp1, tmp2, tmp3, tmp4
tmp1 = "50"
tmp2 = 50
tmp3 = tmp1 + tmp2
tmp4 = tmp1 & tmp2
```

執行結果 tmp3 的內容是 100，它的資料型態是“Double”；tmp4 的內容是“5050”，它的資料型態是“String”。

六、折現計算程式碼

下列程式碼 2.5.2 與程式碼 2.5.3 為折現計算程式碼進一步的設計，程式碼 2.5.2 為單利計算，程式碼 2.5.3 為複利計算。程式碼 2.5.2 中，#02 到#05 為變數的宣告，不過這次變數的值並不是寫死在程式碼中，我們使用輸入的對話盒由程式外部取得資料。InputBox 函數的引數會成為輸入對話盒的提示文字，如 #07 的執行結果會產生下面輸入對話盒。



輸入金額 100，在依次輸入折現率 0.04，到期日數 182，便可得到現值 98.0445。

```
#01 Public Sub IOTest_1()  
#02     Dim CF As Double  
#03     Dim y As Double  
#04     Dim T As Double  
#05     Dim P As Double  
#06  
#07     CF = InputBox("Input Cah Flow Amount: ")  
#08     y = InputBox("Input Discount Rate: ")  
#09     T = InputBox("Input Time to Maturity(in Days): ")  
#10  
#11     P = CF / (1 + y * T / 365)  
#12  
#13     MsgBox P  
#14 End Sub
```

程式 2.5.2 (Book1.xls)

```

#01 Public Sub IOTest_2()
#02     Dim CF As Double
#03     Dim y As Double
#04     Dim T As Double
#05     Dim P As Double
#06
#07     CF = InputBox("Input Cah Flow Amount :")
#08     y = InputBox("Input Discount Rate :")
#09     T = InputBox("Input Time to Maturity(in Years): ")
#10
#11     P = CF / (1 + y) ^ T
#12
#13     MsgBox P
#14 End Sub
程式 2.5.3 (Book1.xls)

```

程式 2.5.4 是一個簡單的範例，使用電腦的計算能力，求得連續複利中使用的常數－自然指數 e 。根據數學上的定義，

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$$

利用下面程式碼，輸入不同的 n 值，可以看到逼近的現象。

```

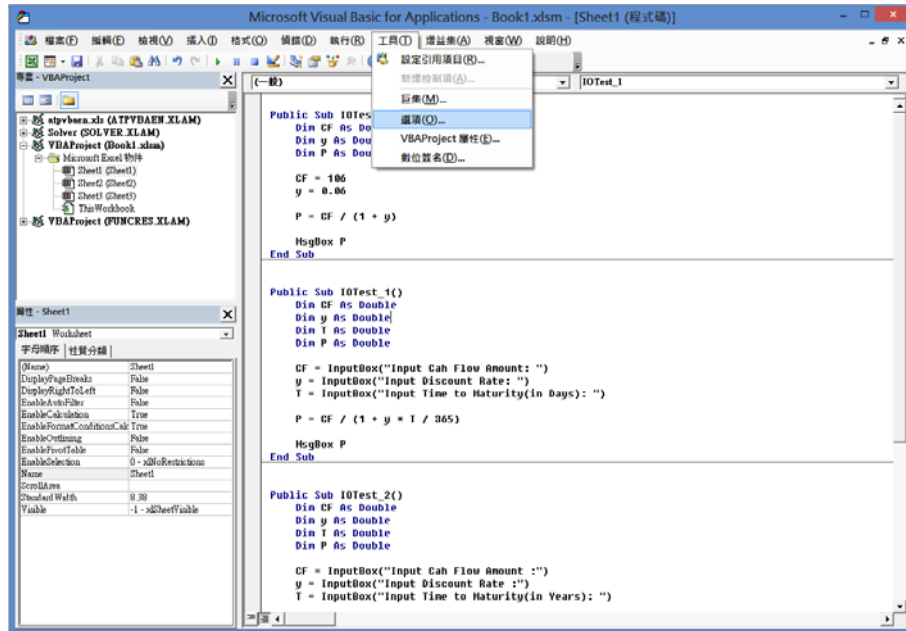
#01 Public Sub ExpApproximate()
#02     Dim N As Double
#03     Dim ans As Double
#04
#05     N = InputBox("Input N :")
#06
#07     ans = (1 + 1 / N) ^ N
#08
#09     MsgBox ans
#10 End Sub
程式 2.5.4 (Book1.xls)

```

N	10	100	1000	10 ⁴	10 ⁵	10 ⁶	10 ⁹
	2.593742	2.704814	2.716924	2.718146	2.718268	2.718280	2.718282

第六節 Excel VBA 的程式編輯環境

要設定 Excel VBA 的程式編輯環境，可以如下勾選功能表指令【工具／選項】。



便會出現下面的對話視窗，提供使用者設定編輯環境的選項，一共有四大類。善用這些選項可以便利我們程式的撰寫與測試，提昇我們的生產力。

一、編輯器選項



在編輯器選項中，我們可以設定是否要自動進行語法檢查，以及一些自動提示的功能。編輯器也提供自動縮排的功能，使用者可以自行決定縮排的字數多寡，預設是四個字母的空間，也是 Tab 鍵的移動空間。

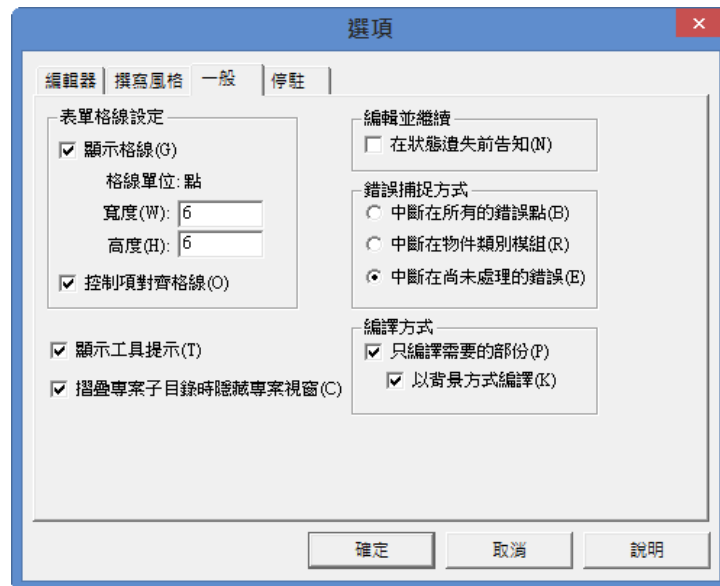
二、撰寫風格選項



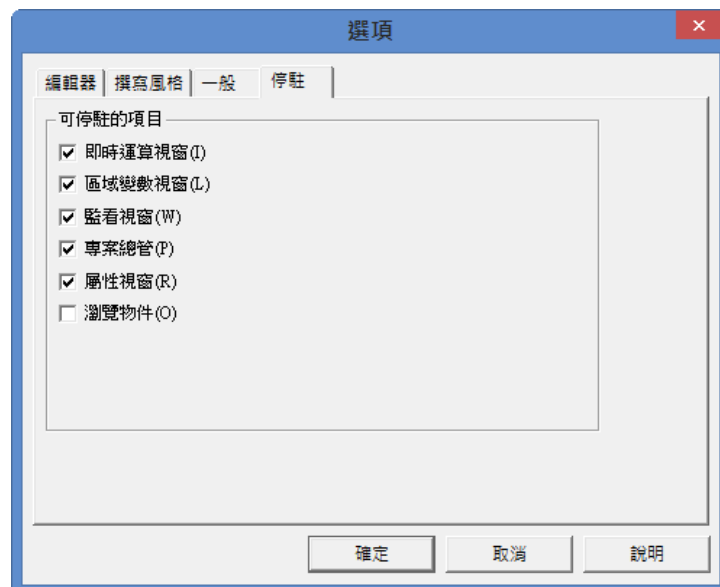
在撰寫風格選項中，我們可以設定程式碼的色彩與一些字型的功能。包括程式碼前景、後景的色彩以及字型種類與大小。作者在字型上，偏好使用 Fixedsys 字形，因為看起來比較清楚。

三、一般與停駐選項

在一般選項中，我們可以設定表單與編譯的選項。

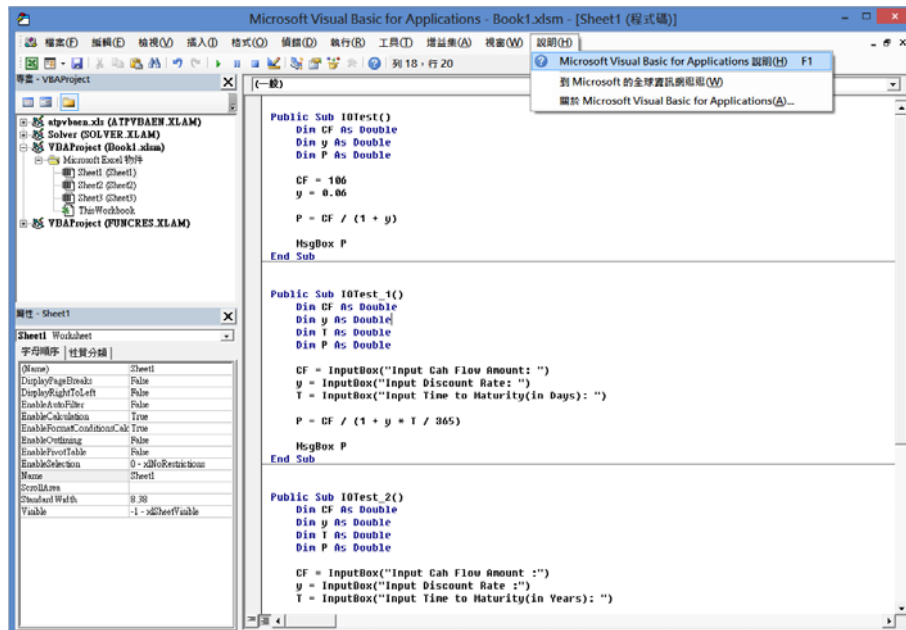


在停駐選項中，我們可以設定一些輔助視窗的停駐與否。



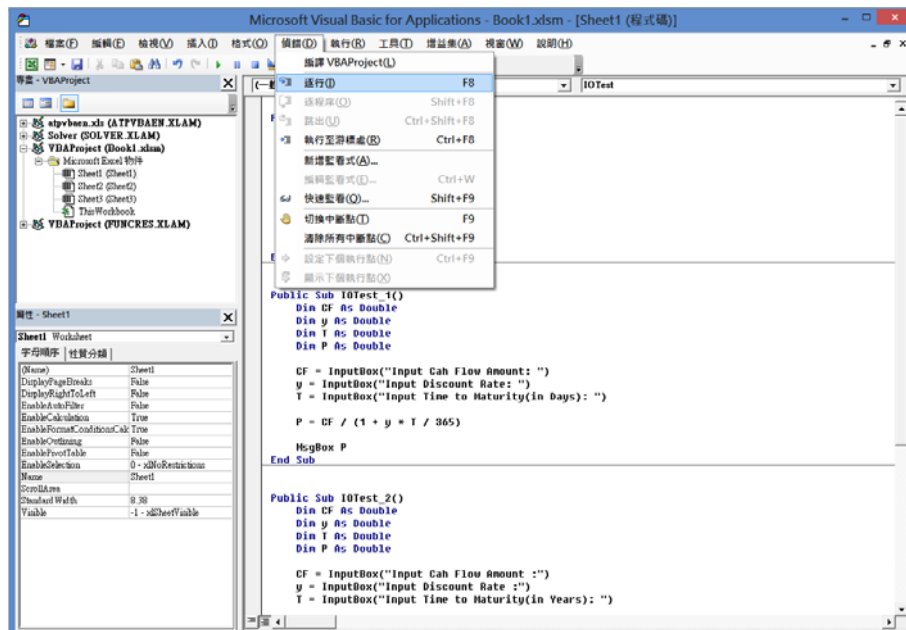
四、求助功能

功能表指令【說明／Microsoft Visual Basic 說明】與快速鍵 **F1** 提供完善的說明，讀者可善加利用。

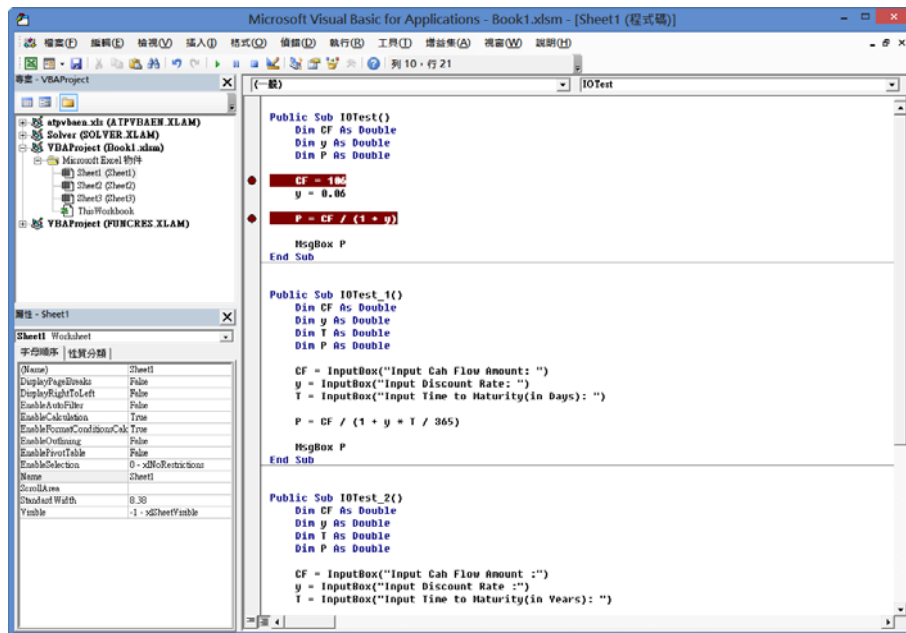


五、偵錯功能

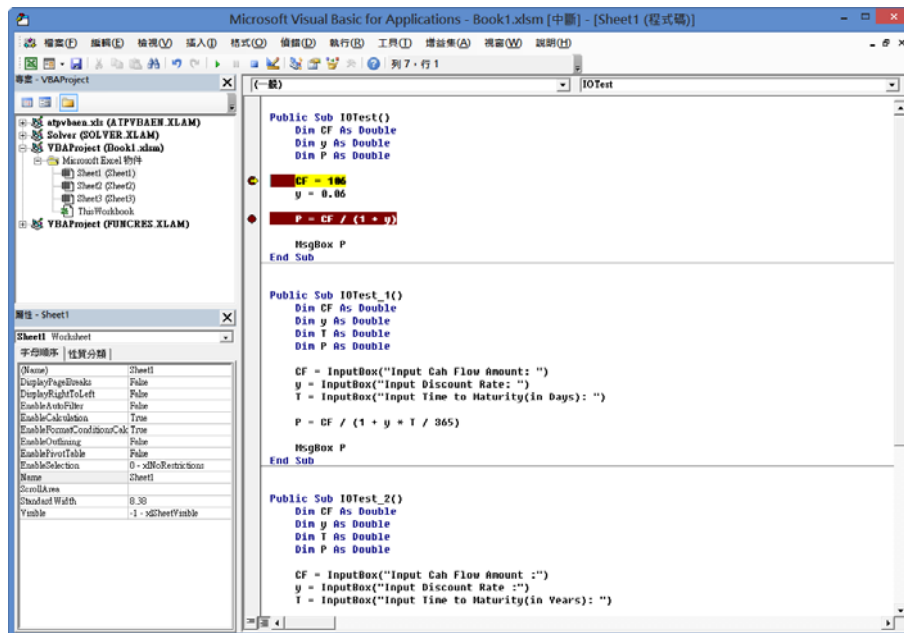
功能表指令【偵錯】有許多便於程式除錯的功能，在此我們只提供一種簡易的方法，便於讀者使用，其他的功能請讀者參考 VBA 的專書。



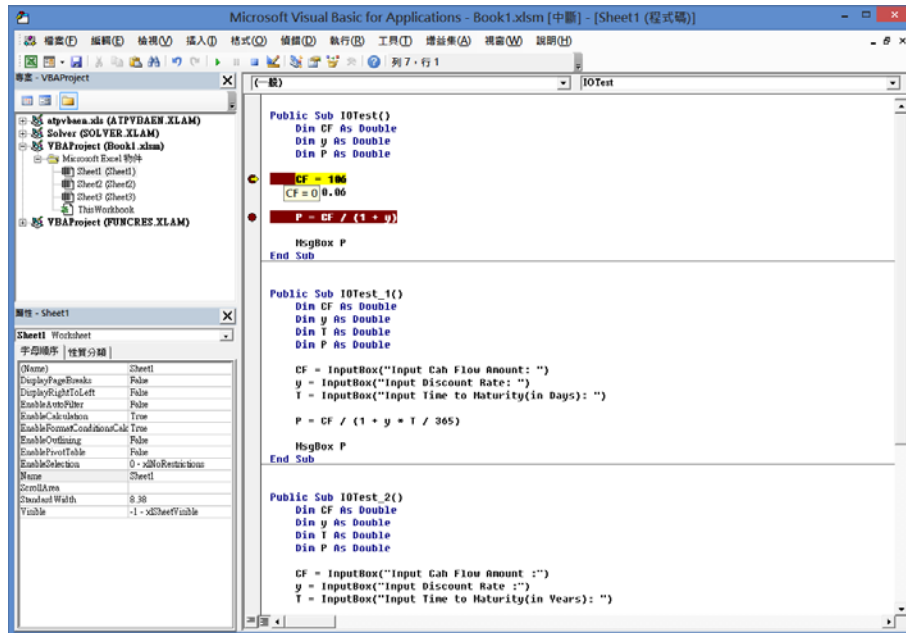
以 IOTest()程式為例，我們可以點選程式碼左方邊框，此時便會在邊框上產生紅點，同時整行程式碼反白。



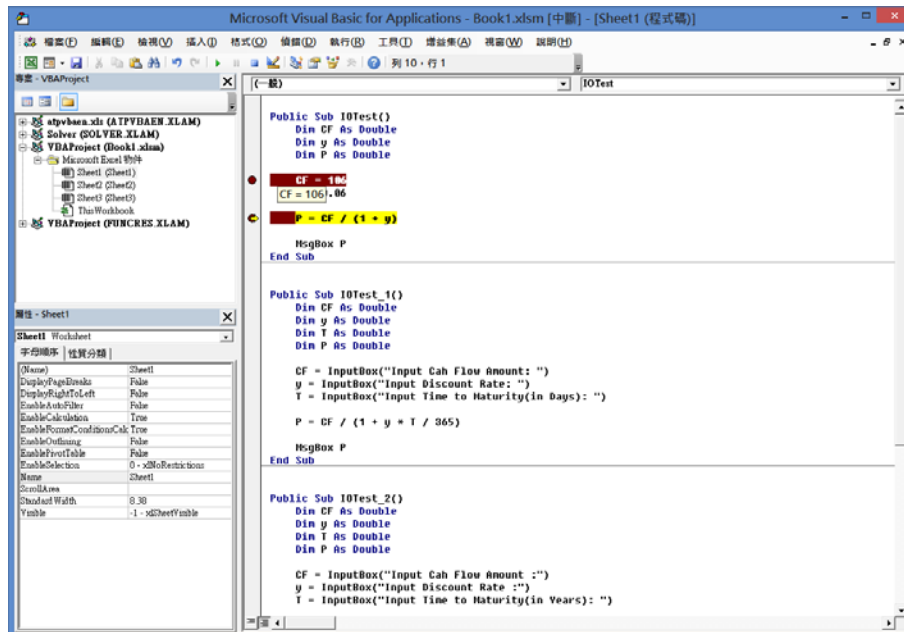
此時，將游標放在 IOTest()程式區塊中，按 **F5** 執行程式，程式開始執行到第一個紅點的中斷點停下執行。



將游標指到變數 CF 之上，便會出現一個標籤，**CF=0**。這表示程式執行到這行，但尚未完成，此時變數尚未設定數值，因此顯示預設值為 0。



再按 **F5** 往下執行，程式開始執行到第二個紅點的中斷點停下執行。將游標指到變數 **CF** 之上，便會出現一個標籤，**CF=100**。這表示程式因為已經執行到 $P = CF / (1+y)$ 這行，此時 **CF** 已經完成設值，因此顯示其值為 100。



另外，此時 y 也已經完成設值，游標指到 y ，會顯示其值為 0.06。這個中斷設定非常簡單、好用，對於我們撰寫程式有很大的助益，讀者務必善加使用。如果要取消中斷點，只需在紅點上再按一次，即可取消。

