

Excel 常用函數

許明宗

公式與函數

公式


- Excel的公式是以『儲存格參照位址』、『名稱』、『函數』或是『常數』等為**運算元**（要計算的內容），透過**運算符號**（**運算子**，『+』、『-』、『*』、『/』等）加以連結而成之計算式
 - B30+銷售+SUMIF(A30:A38,"女",C30:C38)
- 函數為因應某一特殊功能或較複雜計算所寫成之內建子程式，用以簡化輸入之公式
 - 例如，要算A1~A5的平均值， $(A1+A2+A3+A4+A5)/5$
 - 或是使用函數 **AVERAGE(A1:A5)**
 - 利用函數可以簡化公式的內容，縮短公式的建立時間，提高其執行效率

運算符號

| 符 號 | 作 用 | 優先順序 |
|-----|-----|------|
| () | 括號 | 1 |
| + - | 正負號 | 2 |
| % | 百分比 | 3 |
| ^ | 指數 | 4 |
| * / | 乘、除 | 5 |

| 符 號 | 作 用 | 優先順序 |
|------|------------|------|
| + - | 加、減 | 6 |
| & | 連結文字 | 7 |
| = <> | 等於、不 等於 | 8 |
| < > | 小於、大 於 | 8 |
| >= | 大於等於 | 8 |
| <= | 小於等於 | 8 |

公式的輸入

- 資料編輯列的開始輸入『=』
- 直接在資料編輯列輸入要計算的內容（儲存格的參照位址、名稱、函數或是常數等），或是使用滑鼠選取所要計算的儲存格；再配合適當的運算符號
- 最後，再以『**Enter**』鍵或『』鈕，完成整個運算公式之輸入
- 公式內容的最大長度為8192個字元



函數

- 函數為因應某一特殊功能或較複雜運算所寫成之內建子程式，用以簡化輸入之公式
- 基本格式：**函數名稱([引數1], ([引數2],...))**
 - 部份函數不需要有引數，這些函數名稱之後仍然必需要加上一組括號讓 Excel 能夠辨認其為函數
 - TODAY()、NOW()

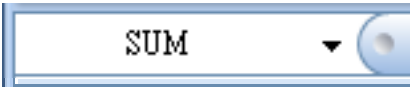
函數的輸入 (1/3)

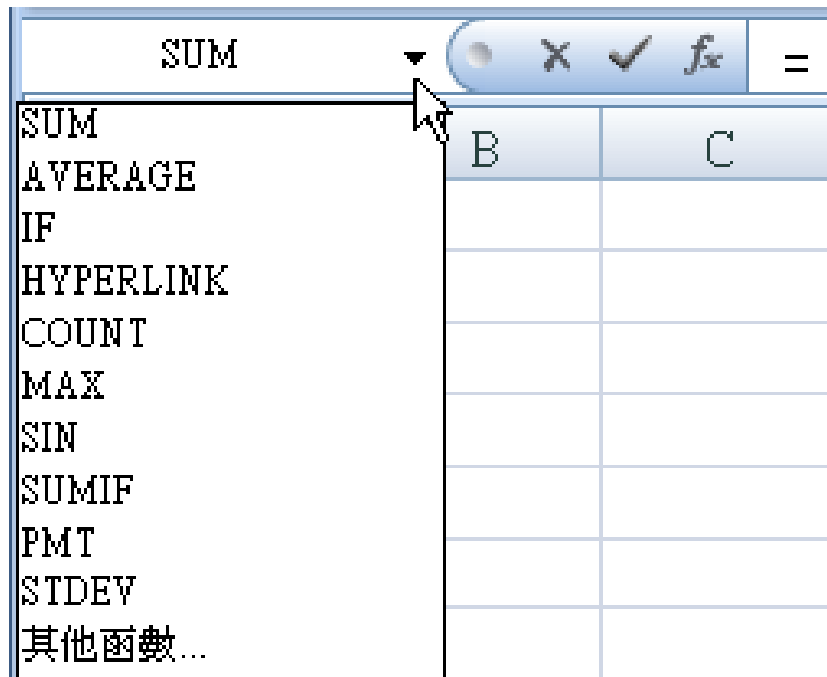
- 在『=』之後輸入函數的第一個字母，可取得以該字母為首的所有函數名稱，可以使用上下鍵或滑鼠來查詢函數的用途
- 雙按滑鼠左鍵選取後，可在儲存格內先輸入函數及左括號，並在其下方顯示函數之語法
- 按下『插入函數』按鈕，可以顯示『函數引數』對話框，來輸入函數的引數

函數的輸入 (2/3)

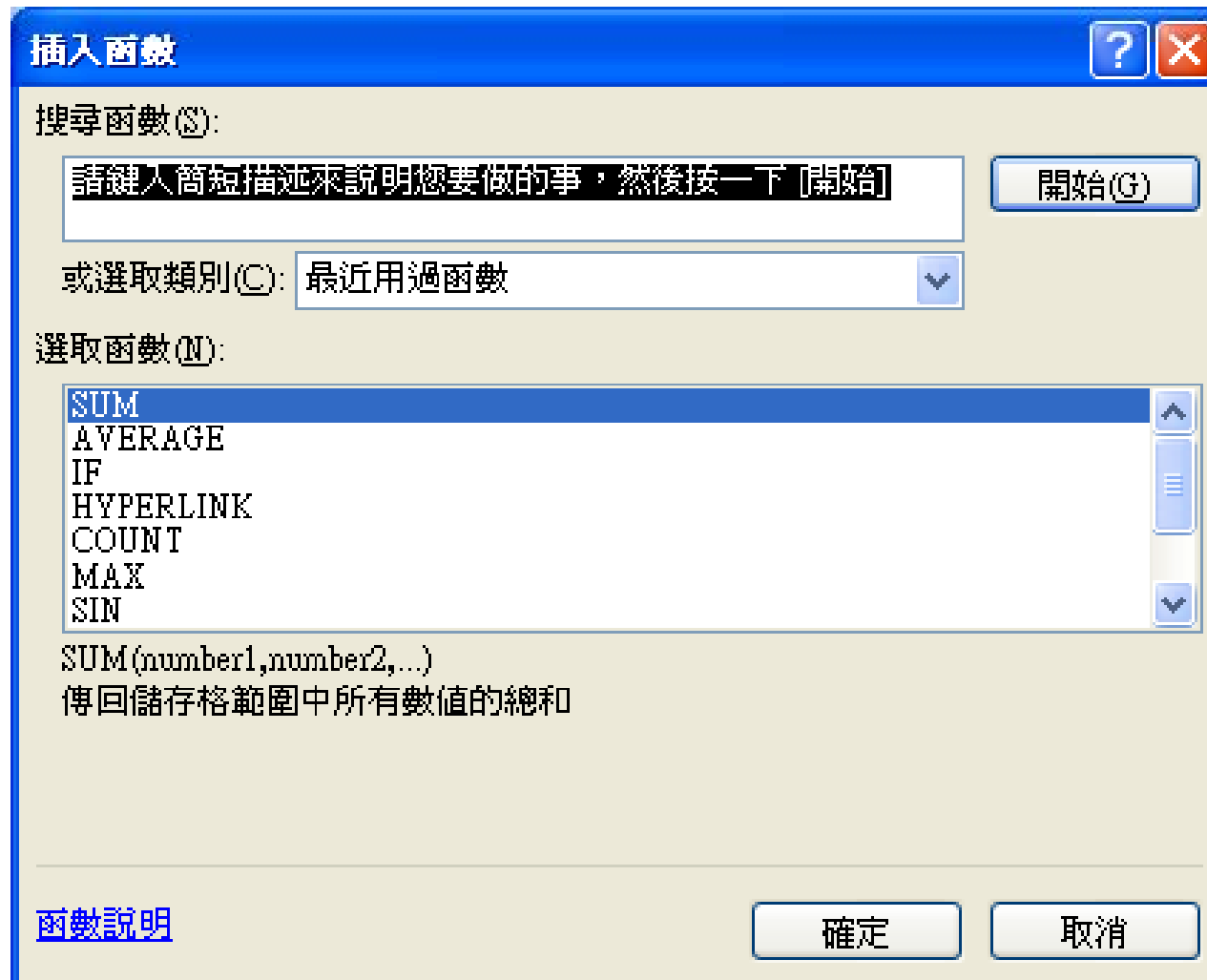
- 選按『公式』索引標籤『函數程式庫』群組『插入函數』鈕 
- 按Shift+F3快速鍵
- 按『資料編輯』列上之  鈕

函數的輸入 (3/3)

- 先輸入等號 (=)，於『資料編輯』列上，按  後之向下箭頭鈕（會出現最近使用的函數），在下拉式選單選擇「其它函數...」



插入函數 (1/2)



插入函數 (2/2)

或選取類別(C): 最近用過函數

選取函數(N):

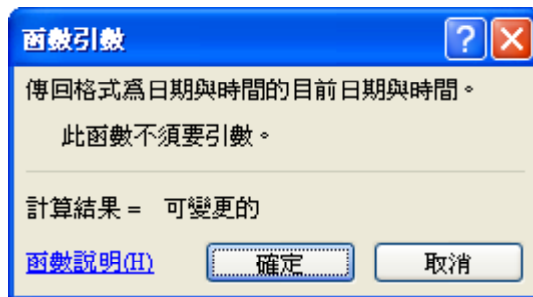
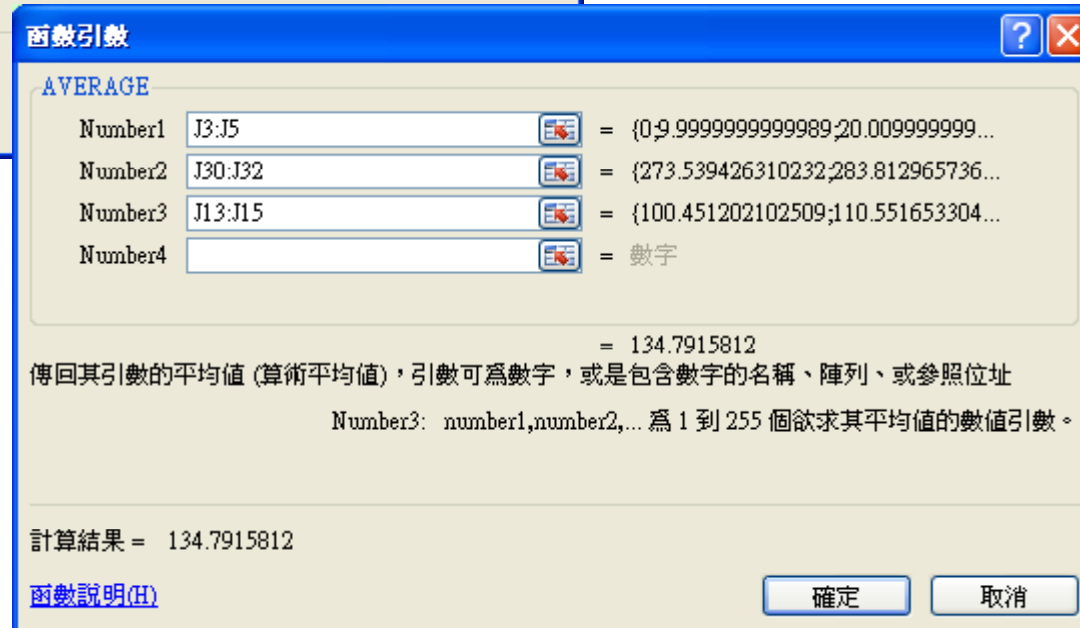
- SUM
- AVERAGE
- IF
- HYPERLINK
- COUNT
- MAX
- SIN

SUM(number1,number2,...)
傳回儲存格範圍

最近用過函數

- 全部
- 財務
- 日期及時間
- 數學與三角函數
- 統計
- 檢視與彙照
- 資料庫
- 文字
- 邏輯
- 資訊
- 工程

函數的引數



公式的各錯誤值之意義

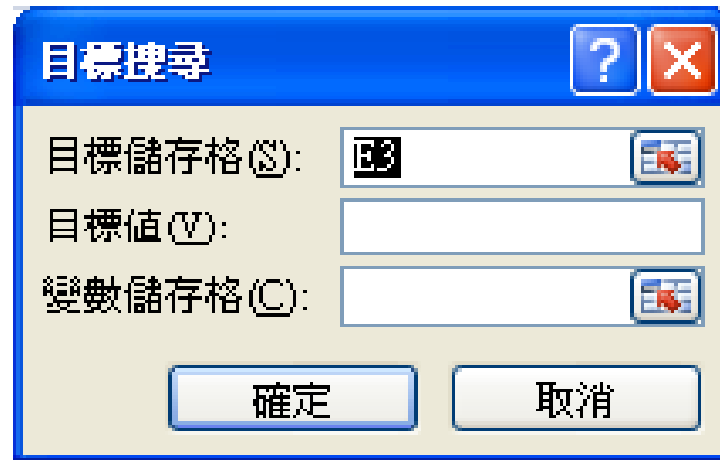
| 錯誤值 | 發生原因 |
|---------|----------------------|
| #DIV/0! | 除數為零 |
| #N/A | 參照到沒有可用數值之儲存格 |
| #NAME? | 公式裡有Excel無法辨識之名稱或函數 |
| #NULL! | 所指定的兩個區域沒有交集 |
| #NUM! | 所輸入之數字有問題 |
| #REF! | 參照到無效之儲存格，如：該儲存格已被刪除 |
| #VALUE! | 使用錯誤之引數或運算元 |
| ##### | 欄寬不夠大 |

目標搜尋 (1/2)

- 由最後的函數值回推某一個函數的引數值為何
- 例如，使用PMT函數來計算每期的還款金額
 - 當貸款總額改變時，每期應該還多少金額？
 - 此時，僅需於改變函數的引數值即可
 - 若是已知每期能夠還多少金額，請問可以借多少錢？

目標搜尋 (2/2)

1. 選取要進行求解之公式格
2. 執行『資料』索引標籤『資料工具』群組『假設狀況分析』鈕之「目標搜尋(G)...」



邏輯函數

比較計算運算子與邏輯運算式

- 比較計算運算子
 - = 等於
 - <> 不等於
 - < 小於
 - > 大於
 - >= 大於等於
 - <= 小於等於
- 邏輯運算式
 - A10 = 100
 - B100 = “女生”

條件判斷

IF(邏輯測試,條件為真的值,條件為假的值)

- 如果指定的邏輯測試的結果為 **TRUE**，會傳回某一個值；如果結果為 **FALSE**，會傳回另一個值

交集

AND(邏輯測試1, [邏輯測試2], ...)

- 如果所有邏輯測試都評估為 **TRUE**，會傳回 **TRUE**；如果有一個或多個引數評估為 **FALSE**，則傳回 **FALSE**

聯集

OR(邏輯測試,[邏輯測試2],...)

- 如果有任何一個邏輯測試的評估為 **TRUE**，即傳回 **TRUE**；唯有所有引數的邏輯值均為 **FALSE** 時，才會傳回 **FALSE**

數學函數

取整數

INT(數值)

- 取得一個數字或數值運算結果的整數部份，將其小數部份無條件捨去
- 例如，要求取某一個數值（Num）的小數部份
 - $\text{Num} - \text{INT}(\text{Num})$

餘數

MOD(被除數,除數)

- 求被除數除以除數後之餘數
 - 例如，5/3 餘 2
- 應用實例：計算通話的單位（多少個6秒）
 - $\text{INT}(125/6)+\text{IF}(\text{MOD}(125,6)=0,0,1)$

進位與捨位

ROUND(數值,小數位數)

- 四捨五入
 - 依小數位數指定由第幾位小數以下四捨五入。若為0，表整數以下四捨五入

ROUNDUP(數值,小數位數)

- 無條件進位
 - 由指定之小數位，無條件進位

ROUNDDOWN(數值,小數位數)

- 無條件捨位
 - 由指定之小數位，無條件捨位

乘積與乘冪

PRODUCT(數值1,[數值2],...)

- 可求算最多可達255個數值引數之乘積
- $B2 * C2 * D2 \equiv \text{PRODUCT}(B2:D2)$

POWER(底數,指數)

- 計算底數的指數次方之結果
- $\text{POWER}(x,y) \equiv x^y$

平方根與絕對值

SQRT(數值)

- 本函數是用來求某數值的平方根，若數值為負值，本函數將回應#NUM!之錯誤
- $\text{SQRT}(10) \equiv 10^{(0.5)}$

ABS(數值)

- 求某數值或運算式之結果的絕對值

圓周率與自然對數的底數

PI()

- 求傳回圓周率 (π) 3.14159265358979，其精確度可達**15**位數

Exp(數值)

- 傳回自然對數的底數**e**的乘冪
- 常數 **e** 等於 **2.71828182845904**

亂數

RAND()

- 隨機產生一介於0~1之亂數
- 每當遇有輸入運算式或按F9鍵要求重新計算，亂數結果將會重算

進位到最接近之倍數

CEILING(數值,基底)

- 將數值進位到最接近之基底的倍數
- 例如，某產品價格是93，因不想使用一或五元的零錢，可使用**CEILING(93,10)**，將產品價格進位到最接近之10的倍數100

捨位到最接近之倍數

FLOOR(數值,基底)

- 將數值捨位到最接近之基底的倍數
- 某產品價格是93，因不想使用到一或五元的零錢，可使用FLOOR(93,10)，將產品價格捨位到最接近之10的倍數90。

最接近之偶數、奇數

EVEN(數值)

- 將數值進位到最接近的偶數整數

ODD(數值)

- 將數值進位到最接近的奇數整數

最大公因數與最小公倍數

GCD(數值1,[數值2], ...)

- 求算最多255組數字之最大公因數（greatest common divisor）

LCM(數值1,[數值2], ...)

- 求算最多255組數字之最小公倍數（lowest common multiple）

階乘

FACT(n)

- 傳回數字n的階乘（ $n! = n \times (n-1) \times (n-2) \times \dots \times 1$ ）。
- 如果數字不是整數，小數部份會被自動捨去
- 而 $0! = 1$

排列

PERMUT(n,r)

- 自n件完全相異之物品，任取r件排成一行，其排列方式有幾種方法之公式為：

$$P_r^n = \frac{n!}{(n-r)!}$$

組合

COMBIN(n,r)

- 傳回自n個相異之項目中，任取r個進行不重複組合之方式總數。兩引數均必須為數值，若含小數將被自動捨去小數。如果n<0、r<0或n<r，將傳回#NUM!之錯誤值。其公式為：

$$C_r^n = \frac{n!}{r!(n-r)!}$$

羅馬字

ROMAN(數值)

- 將阿拉伯字表示之數值，改爲以羅馬字顯示。其數值不得爲負值，且必須不超過3999，否則將獲致#VALUE!之錯誤

對數、自然對數、基底為10的對數

LOG(數值,[基底])

- 計算數值為基底的幾次方（對數值），省略[基底]，預設其值為10

LN(數值)

- 傳回以 e 為基底的對數值（數值應為 e 的幾次方）
- 本函數為 **EXP()** 的反函數

LOG10(數值)

- 計算數值為10的幾次方

三角函數

SIN(x)

COS(x)

TAN(x)

- 分別傳回x角度的正弦、餘弦與正切值，x為以弧度為單位的角度。如果角度的單位是度，得將其乘上 $\text{PI}()/180$ 轉換為弧度

文字函數

字串、串接字串

- 字串（或稱為文字串）：一段的文字（可能只有一個字）
 - 在儲存格中，格式為文字或是以一個單引數『』開始的數值內容
 - 在函數引數中，包含在一對雙引號『“”』的任何內容
- 使用『&』可以串接兩個字串
 - `DATEVALUE("2009"&"/05"&"/02")` ≡ `DATEVALUE("2009/05/02")`

字串長度

LEN(字串)

LENB(字串)

- 可算出文字串之字數或位元數（包括空白）

由左至右取出部份字串內容

LEFT(字串,[字元數])

LEFTB(字串,[位元數])

- 從字串的左邊開始取出指定[字元數]（或[位元數]）之子字串，若是省略，預設值為1
 - 以字數為單位時，每一個中文與全/半型的英文/數字，均為一個字元
 - 以位元數為單位時，一個中文/全型字佔兩個位元，半型英文/數字佔一個位元

由右至左取出部份字串內容

RIGHT(字串,[字數])

RIGHT(字串,[位元數])

- 自某字串之右邊取出指定字數（或位元數）之子字串，若省略，其值自動補為1。
- 以字數為單位時，全型中文與半型英文/數字均無差異，均為一個字。
- 而以位元數為單位時，半型英文/數字仍佔一個位元；而一個中文字或全型字則佔兩個位元

取出中間的子字串 (1/2)

MID(字串,第幾個字開始,字數)

MIDB(字串,第幾個位元開始,位元數)

- 自某字串之第幾個字開始，取出指定字數（或位元數）之子字串

取出中間的子字串 (2/2)

- 以MID()中間子字串即可完全取代LEFT()與RIGHT()。如：

MID(文字串,1,長度) ≡ LEFT(字串,長度)

MID(字串,總長度-長度+1,長度) ≡ RIGHT(字串,長度)

複製文字

REPT(字串,複製次數)

- 依指定次數複製字串之內容
- 複製結果的長度不可超過32,767個字元
- 若複製次數含有小數，則僅取其整數

取代字串

REPLACE(舊字串,起始位置,移出字元長度,新字串)

- **REPLACEB**(舊字串,起始位置,移出位元長度,新字串)
- 將舊字串由起始位置開始，消去移出字元長度所指定之字元數（或位元數），代之以新字串內容

取代文字串

SUBSTITUTE(字串,要取代之舊字串,要換成之新字串,[第幾組])

- 可將字串中的指定的某一組要取代之舊字串（原內容可能有多組要更換之舊字串），更換為要換成之新字串
- 本函數是用於知道要處理之舊字串時，其控制內容是字串；而若是以位置進行處理，則必須使用**REPLACE()**函數，其控制內容是數字。
- 若省略控制要處理之[第幾組]引數，則字串中的每一組舊字串均會被取代為要換成之新字串

以ASCII碼產生字元

CHAR(數字)

- 用來產生數字值所代表的ASCII字元，數字之值應介於1到255

查字元的ASCII碼

CODE(字串)

- 用來傳回字串之『第一個』字所代表的字元集的數字代碼

數值轉文字

TEXT(數值,格式字串)

- 將數值結果（只要是可轉成數值之數字或字串均可），依格式字串指定的格式轉成字串

貨幣數值轉文字

DOLLAR(數值,[小數位])

- 將數值依貨幣格式，以指定之小數位將其轉換為文字。若省略小數位，系統將預設為2

固定小數位數值轉文字

FIXED(數值,小數位,不要逗號)

- 將數值依句點和逗點格式，於指定之小數位將其轉換為字串。若省略小數位，其預設值為2
- 不要逗號為一邏輯值，省略時，其預設值為**FALSE**，處理結果將含千分位之逗號。其與**DOLLAR()**之差別僅在最前面無金錢符號而已

移除多餘空格

TRIM(文字串)

- 本函數會消除文字串前、後及中間之多餘空白字元。但無論是中文或英文字串，字間仍均保有一格空白。
- 這對英文是合理的，但對中文卻是一種錯誤
- 針對清除中文間之多餘空白，最佳途徑還是執行『取代』

標題大寫

PROPER(字串)

- 可將字串之每個英文全字（**word**）的第一個字母轉成大寫；其餘字母轉為小寫

轉小寫

LOWER(字串)

- 將字串中之英文字母均轉換成小寫字體
(lower case letter)

轉大寫

UPPER文字串)

- 文字串中之英文字母均轉換成大寫字體
(upper case letter)

文字轉數值

VALUE(文字串)

- 可將數字組成之文字串轉換成數值；若無法轉換，其結果為#VALUE!

財務函數

固定利率下的期末總金額

FV(利率,期數,[每期的金額],[現值],[期初或期末])

- 在已知利率、期數、每期的金額（或是期初的現值）下，計算未來值
- 可以省略現值或是每期的金額其中之一（使用0為預設值）
- 『期初或期末』為『每期的金額』的發生時間點，省略（或0）為期末發生，1為期初發生
- 例如，可以用來計算複利（沒有每期的金額）或是零存整付（沒有現值、會在期初存入）的期末所得
- 計算結果為負數，因為Excel使用現金流出的觀念。若不習慣，可以在每期的金額或是現值加上負號

固定利率下的現值

PV(利率,期數,每期的金額,[期末值],[期初或期末])

- 在已知利率、期數、每期的金額（或是期末值）下，計算現值
- 可以省略期末值或是每期的金額其中之一（使用0為預設值）
- 『期初或期末』為『每期的金額』的發生時間點，省略（或0）為期末發生，1為期初發生
- 例如，可以用來計算投資後（沒有每期的金額）或是每期支付（沒有期末值、會在期初存入）的現值

每期的金額

PMT(利率,期數,現值,[未來值],[期初或期末])

- 在已知利率、期數、現值（或是期末值）下，計算每期的金額
- 現值：為未來各期年金現值的總和
- 未來值：期末所能獲得的金額。
- 現值或未來值兩者之一可以省略（使用0為預設值）
- 期初或期末：用以界定各期金額的給付時間點。省略或0為期末給付，1為期初給付
 - 銀行貸款，通常是期末償付；零存整付則又得期初給付。金額的給付發生在期末者為普通年金；發生在期初者則為期初年金。
- 年金指的是每一期支付固定的金額
 - 例如，貸款的每期償還金額、保險年金等

付款中的利息

IPMT(利率,第幾期,總期數,本金,[未來值],[期初或期末])

- 傳回付款方式為定期、定額及固定利率之投資，某一期付款中的利息金額

期數

NPER(利率,每期給付金額,本金,[未來值],[期初或期末])

- 傳回每期付款金額及利率固定之情況下，償還全部貸款或達成某項投資的期數

利率

RATE(期數,每期給付金額,本金,[未來值],[期初或期末],[猜測利率])

- 本函數可傳回年金的利率。猜測利率若省略，其預設值為**10%**
- 計算利率時，是以反覆運算進行，可能是無解或有多組解。如果在**20**次反覆運算後，仍無法收斂到**0.0000001**以內，將傳回**#NUM!**錯誤值。此時，可改另一個猜測利率，再進行重算
- 通常，只要猜測利率在**0~1**之間，本函數通常都會收斂。注意，公式內每期給付金額應安排為負值

非固定年金之現值

NPV(利率,現金流量1,[現金流量2], ...)

- 傳回利率固定、現金流量非固定時的現值，現金流量最多可達255個
- PV()函數，適用於年金為單一固定值；每期之年金都不相同時，就得使用NPV()

查詢及參考函數

水平查表

HLOOKUP(查表值,表,資料所在的列,範圍尋找)

- 在一個陣列或表格的**第一列**中尋找含有某特定值的欄位，再傳回同一欄中某一指定儲存格中的值
- 當查表值是在一定的範圍內（例如，介於**50到60**之間），要使用範圍尋找（**True**）
- 當查表值是一個確定的值（例如，某一個同學的學號）時，不可使用範圍尋找（**False**）

垂直查表

VLOOKUP(查表值,表,資料所在的欄,範圍尋找)

- 在表格陣列的第一欄中搜尋某個數值，並傳回該表格陣列中同一列之其他欄中的數值
- 當查表值是在一定的範圍內（例如，介於50到60之間），要使用範圍尋找（**True**）
- 當查表值是一個確定的值（例如，某一個同學的學號）時，不可使用範圍尋找（**False**）

查表

LOOKUP(查表值, 範圍向量, 結果向量)

- 在單列或單欄範圍 (亦稱為向量) 中搜尋值，並從第二個單列或單欄範圍內的相同位置中傳回
值

LOOKUP(查表值, 陣列)

- 在陣列的第一列或第一欄中搜尋指定的值，然後從陣列最後一列或最後一欄的相同位置中傳
回
值