

前所未見的 EXCEL 寶典

{點、陣列、VBA 運算}

egg

January 9, 2018

目錄

I	一維	3
1	基礎訊息	5
1.1	儲存格	5
1.2	運算子	6
1.2.1	算術運算子	6
1.2.2	比較運算子	7
1.2.3	聯結運算子	7
1.2.4	參考運算子	7
2	函數	9
2.1	邏輯	10
2.1.1	=AND()	11
2.1.2	=OR()	13
2.1.3	=IF()	17
2.2	文字	25
2.2.1	=LEFT()	25
2.2.2	=RIGHT()	27
2.2.3	=MID()	28
2.2.4	=REPLACE()	30
2.2.5	=SUBSTITUTE()	31
2.3	查閱參照	33
2.3.1	=VLOOKUP()	33
2.3.2	=HLOOKUP()	35
2.3.3	=ROW()	37

2.3.4	=COLUMN()	41
2.3.5	=ROWS()	45
2.3.6	=COLUMNS()	46
2.3.7	=OFFSET()	47
2.4	數學	51
2.4.1	=INT()	51
2.4.2	=ROUND()	53
2.4.3	=ROUNDUP()	55
2.4.4	=ROUNDDOWN()	56
2.4.5	=COUNT()	58
2.4.6	=COUNTA()	60
2.4.7	=COUNTIF()	61
2.4.8	=SUMIF()	63
2.4.9	=RAND()	65
2.5	統計	67
2.5.1	=MAX()	67
2.5.2	=MIN()	68
2.5.3	=MEDIAN()	69
2.5.4	=LARGE()	70
2.5.5	=SMALL()	72
2.6	房貸	73
2.6.1	=PMT()	74
2.6.2	=CUMIPMT()	76
2.6.3	=CUMPRINC()	78
2.7	自訂函數	80
2.8	=SUM()	81
2.9	其它函數	82
2.9.1	=TRANSPOSE()	82
3	綜合函數練習	83
3.1	參照函數	83

4	虛擬資料	87
4.1	ID	87
4.2	年齡	88
4.3	婚姻	88
4.4	消費日	88
4.5	消費金額	88

前言

一切的故事起源於以下這個題目：

A 欄是貸款編號、B 欄為貸款利率、C 欄是餘額，共有 657 筆貸款資料，用 Excel 計算這 657 筆貸款的平均利率，限在 D1 儲存格求出 (只能用 D1 格計算)？

* * * * *

自大學時代接觸 Excel 開始，今年正好滿 20 年，一開始覺得 Excel 的複製拖曳實在很好用，內建許多函數很方便，直到遇上了以上這一個考題，限用一個儲存格求解。若沒有限一個儲存格的限制，我不需經過思考，

1. 在 D2 格計算 $B1 * C1$,
2. 點 2 下「填滿控點」以複製公式填滿 D 欄，
3. 接著在 E2 格計算 D 欄的加總除以 C 欄的加總。

這三個簡單步驟即為答案。大概會 Excel 的人就都有辦法計算出來，但這樣的程序並不符合題目的設定「限用一格」。就這樣一個際會，開啟了我對 Excel 陣列運算涉獵。

一般人都懂用輸入 **Enter** 來得到結果，可是不知道可以鍵入 **Ctrl** + **Shift** + **Enter** 來處理「陣列」運算。這十餘年來我的瞎子摸象，也算小有所成，所以跟大家分享這麼一個好用的程序。

本書共分三個部分，以公式函數能處理的維度來分一維、二維、多維。第一部分一維，係指函數能處理的「點」，儲存格單格運算過程或結果，這部分也是大家比較熟悉的部分。

第一部分二維，係指「陣列」功能的運算，只要用一點皮毛就可以解決上述那個平均利率題目。功能強大，卻是大家很不熟的一塊，即使坊間的書也很少介紹或碰觸運用，以我工作十餘年的經驗，它實在太好用、太實用了，不

會 Excel 的這部分真是太可以,所以我一定要推廣它,於是我著手試著寫寫看,這是這本書最初的起源。

很少有人會把 Excel 這樣用,大部分的人都僅認為 Excel 只能運算一維,所以乖乖的 A1 格與 A2 格運算;輸入也是,一格一格輸入,不知道可以一列、一欄或一整個 $n \times n$ 的陣列輸入與運算。

查詢 Microsoft 的 Excel 公式概觀,它對「陣列公式」說明如下:

陣列公式可以同時執行多個計算,然後傳回單一結果或多個結果。陣列公式可以在兩組以上的數值上(也就是陣列引數)進行運算。每個陣列引數必須具有相同的列數和欄數。除了是用按 CTRL+SHIFT+ENTER 的方法來輸入公式外,建立陣列公式的方法與建立其他公式的方法相同。部分的內建函數是陣列公式,而且必須以陣列方式輸入才能得到正確的結果。

可能有人會有涉獵到 SUMIF()、COUNTIF()、SUMPRODUCT() 等準陣列函數,可是這還離我所說的陣列運算還有一小段距離,對於我所謂真正的「陣列」運算,好像還沒有遇到有人真正使用。不過,你可以找到 Microsoft Office 的技術支援網站介紹陣列公式為 $\widehat{\text{CSE}}$ 公式(按 Ctrl + Shift + Enter 鍵陣列運算)。

陣列的運算可以處理很複雜的多維度問題,而且它比樞紐分析表好用的一點是可以自動更新,不像樞紐一旦產出,當資料變更時,需要再重跑才會更新結果。

我總是建議,學會了陣列運算,就可以把樞紐分析表丟了;所以爲了要把樞紐分析表丟掉,應該要學好陣列運算。

第三部分是多維,要把樞紐丟掉前也該了解它有多不濟,所以介紹一下;接著介紹 VBA,它更是一個強大的程序功能,別說可以解決前述平均利率的問題,只要有解的問題就能使用 VBA 解決,尤其是大量重複運算的程序問題,使用 VBA 迴圈可以節省大量的時間。

Part I

一維

第 1 章

基礎訊息

 +  + 

1.1 儲存格

儲存格是組成試算表最小單位, 直的儲存格稱作「欄」, 以英文字母編號, 橫的儲存格稱作「列」, 以數字編號, 形成每個儲存格有一個位址編號為英文字+數字, 如 A1 格, 或 AAA1000 格 等。數欄及數列組成工作表, Excel 預設為 3 個工作表, 可增加、刪除、隱藏, 工作表名稱可以依自己偏好命名, 由數個工作表組成 *.xlsx 檔。

以 Excel 預設的儲存格來說, 它的格式為「通用格式」, 如果不經調整, 把數字鍵入儲存格, 該數字會靠右, 若鍵入數字以外的字串, 它會靠左。

格式種類

類別	說明	範例
通用格式	沒特定格式	
數值	減法或負	$3 - 1$ 或 -1
貨幣	乘法	$3 * 3$
會計專用	除法	$3/3$
日期	百分比	20%
時間		
百分比		
分數		
科學計號		
文字		
特殊		
自訂		

1.2 運算子

1.2.1 算術運算子

算術運算子	意義	範例
+ (加號)	加法	$3 + 3$
- (減號)	減法或負	$3 - 1$ 或 -1
* (星號)	乘法	$3 * 3$
/ (斜線)	除法	$3/3$
% (百分比符號)	百分比	20%
^ (插入號)	乘冪	3^2

1.2.2 比較運算子

比較運算子	意義	範例
= (等號)	等於	A1=B1
> (大於符號)	大於	A1>B1
< (小於符號)	小於	A1<B1
>= (大於等於符號)	大於或等於	A1>=B1
<= (小於等於符號)	小於或等於	A1<=B1
<> (不等於符號)	不等於	A1<>B1

1.2.3 聯結運算子

聯結運算子	意義	範例
& (&符號)	聯結二個值以產生一個連續的文字值	A1&B1

1.2.4 參考運算子

參考運算子	意義	範例
:(冒號)	範圍運算子	B5:B15
, (逗號)	聯集運算子	SUM(B5:B15, D5:D15)
(單行間距)	交集運算子	SUM(B7:D7 C6:C8)

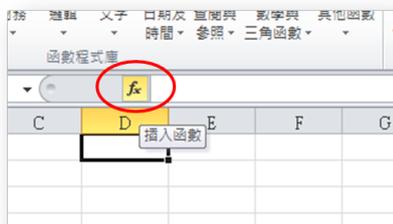
第 2 章

函數

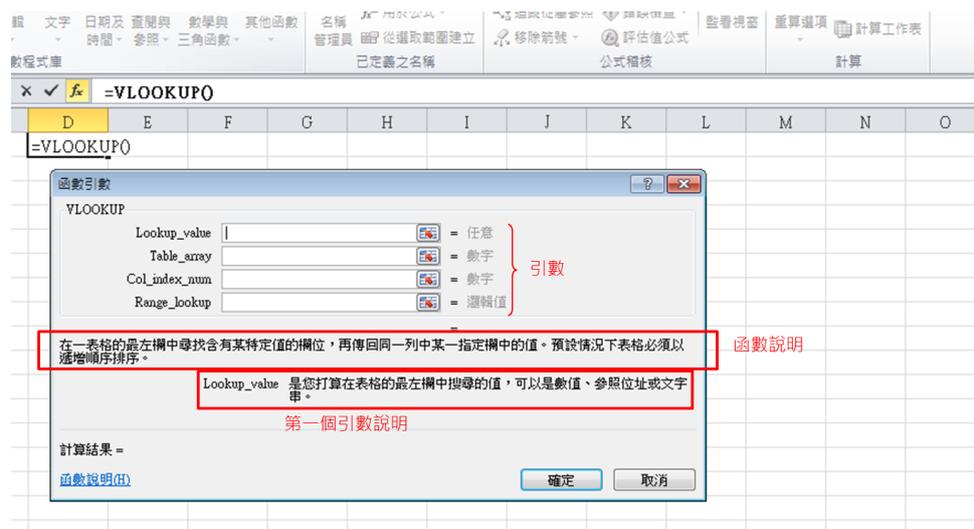
函數組成有 3 個部分, 第 1 部分為等號 =, 第 2 部分函數名稱, 如 SUM, 第 3 部分為括號 (), 及連同括號其中的參數。這 3 個部分缺一不可, 就算其中不需要參數的函數如 =TODAY()、=NOW()、=RAND()、=TRUE() 等, 也要有括號。

Excel 內建函數很多, 這裡無法一一介紹, 我依據 Excel 工具區的分類, 挑一些我認為一般企劃工作會用到的重要函數介紹。

進入函數之前, 我先教大家該怎麼熟悉一個新函數, 那就是善用工具區的 fx 按鈕。



按了該按鈕就有選單可以挑選你想要使用了函數, 選了函數之後, 電腦會跳出「函數引數」視窗, 告訴你該函數有幾個引數, 每個引數各有一個計算框, 也會告知你該函數是做甚麼計算, 回傳什麼。當你將游標放到其中一個引數時, 電腦會對該引數做註解及說明, 若你把引數輸入資料, 他也會針對該引數先回傳資料。



你也可以先在儲存格鍵入你要使用的函數,再按 *fx* 鈕,依然會跳出函數引數視窗來供操作。

多善用 *fx* 鈕去學習各函數用法,若是它內建的說明仍不知所云,該視窗左下角有一藍色「[函數說明\(H\)](#)」超連結,點下去就會得到更詳細的解釋,也會有案例供參考。

我們依據 Excel 歸類,將要介紹的函數分邏輯、文字、查閱參照、數學、統計、房貸等 6 大類介紹,最後再說明如何自訂函數。

2.1 邏輯

這裡介紹 3 個函數, =AND()、=OR()、=IF(), 前 2 個函數為邏輯判斷, 傳回 TRUE 或 FALSE, 第 3 個函數為邏輯判斷後, 依我們指定成立與否傳回我們各指定的敘述。

這 3 個函數在往後陣列運算時很重要, 所以要先搞清楚, 以利之後的陣列運算理解。

2.1.1 =AND()

=AND()

fx

判斷邏輯, 如果為「真」, 回傳 TRUE; 為「否」則回傳 FALSE。

= AND(logical1, [logical2], ...)

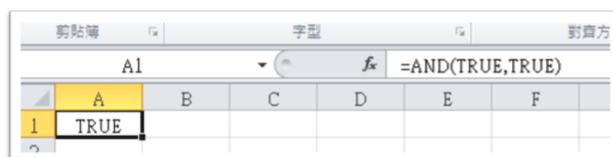
其中,

- logical1: 必要, 邏輯運算式, 運算結果評估它是 TRUE 或 FALSE 的第一條件。
- logical2: 選用, 邏輯運算式為 TRUE 或 FALSE 的其他條件, 最多 255 個條件。

=AND() 至少評估 1 個參數為 TRUE 或 FALSE, 如果其中的參數全部皆成立, 回傳 TRUE, 若有任一參數不成立, 則回傳 FALSE。例如:

= AND(TRUE, TRUE)

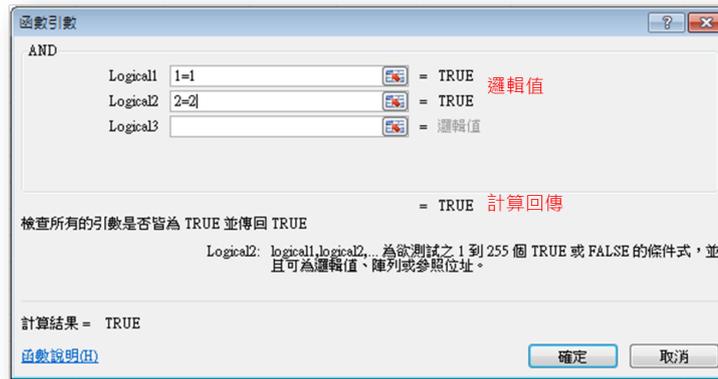
因為全為真, 所以會傳回 “TRUE”。



一樣的意思, 若

= AND(1 = 1, 2 = 2)

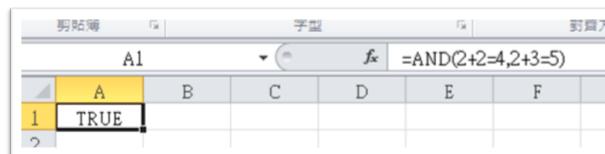
=AND() 的第一個參數鍵入 1=1, Excel 邏輯判斷為 TRUE, 就跟該參數鍵入 TRUE 一樣; 同理, 第二個參數 2=2 也是 TRUE。



所以回傳答案必為“TRUE”。

$$= \text{AND}(2 + 2 = 4, 2 + 3 = 5)$$

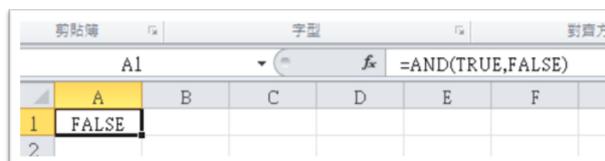
二個引數都為真, 所以成立, 傳回“TRUE”, 如圖。



接著, 把第 2 個參數改成 FALSE,

$$= \text{AND}(\text{TRUE}, \text{FALSE})$$

因為有一個 FALSE, 所以不成立, 故會傳回“FALSE”, 如圖。



同前,

$$= \text{AND}(1 = 1, 2 = 3)$$

其中第 2 個參數 $2 = 3$ 為 FALSE, 所以回傳就為“FALSE”。參數當中為算式也相同 $=\text{AND}(2 + 2 = 4, 2 + 3 = 6) \Rightarrow \text{FALSE}$, 因為 $2 + 3 = 5$ 不為 6, 所以不成立。

2.1.2 =OR()

=OR()

fx

判斷邏輯，當函數內參數有任一條件成立時，回傳 TRUE，反之則回傳 FALSE。

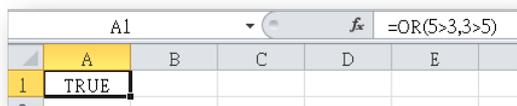
= OR(logical1, [logical2], ...)

其中，

- logical1: 必要，邏輯運算式，運算結果評估它是 TRUE 或 FALSE 的第一條件。
- logical2: 選用，邏輯運算式為 TRUE 或 FALSE 的其他條件，最多 255 個條件。

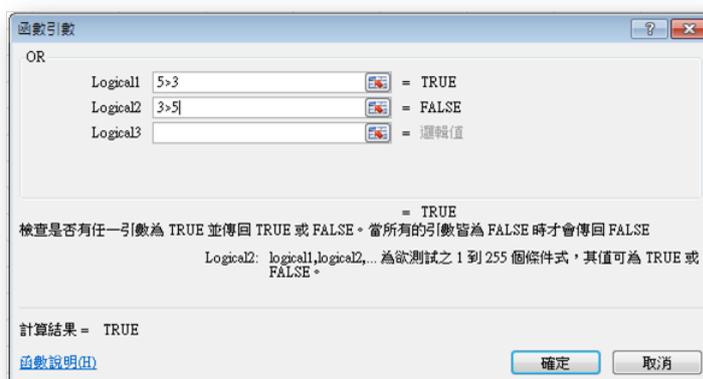
=OR() 來判斷測試中是否有任何條件為 TRUE，若有則回傳 TRUE，若全數參數皆不成立，則回傳 FALSE。例如：

= OR(5>3, 3>5) = TRUE

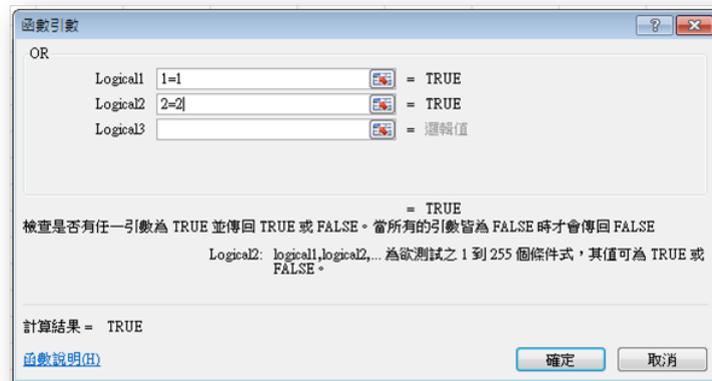


	A	B	C	D	E
1	TRUE				

因為第一個參數成立 TRUE，第二個參數不成立 FALSE，所以回傳成立 TRUE。

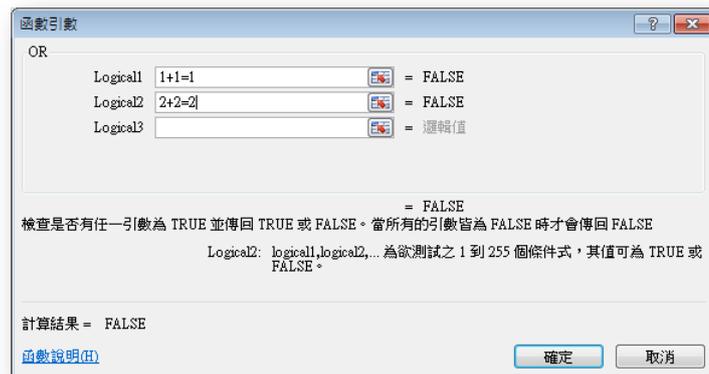


上面是 TRUE 碰上 FALSE, 來看一個 TRUE 碰上 TRUE,



還是回傳 TRUE。

接著, 試看看 FALSE 碰上 FALSE,



回傳 FALSE, 因為全部的參數皆不成立, 所以 FALSE。

=OR() 函數下就只要有一個成立, 最後結果就會是 TRUE。

附帶一說, 其實 =AND() 與 =OR() 函數回傳 TRUE 或 FALSE, 除了給 =IF() 函數當第一個參數以外 (下節介紹), 似乎對使用者沒多大用處。如果我們把 TRUE 及 FALSE 各改成 1 及 0, 這樣就很有用了。因為我們知道任何數乘上 1 都會等於自己本身, 任何數乘上 0 會等於 0, 利用這 2 個特性, =AND() 與 =OR() 函數就會變得很好運用。

怎麼變? 只要在 TRUE 的前面加上 2 個負號, 負負得正, 強迫 Excel 把文字變數字, TRUE 就會變成 1,

A1		fx			
A	B	C	D	E	
1	1				
2					

在 FALSE 前面加 2 個負號, FALSE 會變成 0,

A2		fx			
A	B	C	D	E	
1	1				
2	0				
3					

不管你有沒有鍵入 = 號, “= --TRUE”、或者只是 “--TRUE”, 都會得到 1, 而且 Excel 編輯列都會自動調整為 “= --TRUE”。

試試看,

$$= -- AND(1 + 1 = 2, 2 + 2 = 4)$$

A1		fx			
A	B	C	D	E	F
1	1				
2					

$$= -- OR(1 + 1 = 2, 2 + 2 = 2, 3 + 3 = 2)$$

A2		fx				
A	B	C	D	E	F	G
1	1					
2	1					

另一個方法, 文字變數字就是 “*1”, 利用任何數乘上 1 都還是自身的特性, 所以

$$= AND(1 + 1 = 2, 2 + 2 = 4) * 1$$

函數樣式庫						
A1		fx =AND(1+1=2,2+2=4)*1				
	A	B	C	D	E	F
1	1					
2						

同理, = OR(1 + 1 = 2, 2 + 2 = 2, 3 + 3 = 2) * 1 會等於 1。

函數樣式庫						
A2		fx =OR(1+1=2,2+2=2,3+3=2)*1				
	A	B	C	D	E	F
1	1					
2	1					
3						

實際用一個例子來演算, 假設我們有某公司消費資料, 資料中有性別、年齡、消費金額, 我們想要找 20 歲男生消費金額, 如圖, 我在 D 欄利用邏輯函數 =AND(測試性別, 測試年齡) 回傳 TRUE 或 FALSE, 再乘以消費金額, 強迫 TRUE 或 FALSE 轉成 1 或 0, 只要條件符合者, 就變成 1 * 消費金額, 不符者, 因會傳 0 最終答案為 0。

D2		fx =AND(A2="男",B2>=20)*C2			計算方式	
	A	B	C	D	E	F
1	性別	年齡	消費金額	20歲男生消費金額		
2	男	12	888	0		
3	女	15	450	0		
4	男	22	2480	2480		
5						

2.1.3 =IF()

=IF()

fx

如果指定之條件的計算結果為 TRUE, =IF() 函數會傳回一個值, 如果計算結果為 FALSE, 則會傳回另一個值。

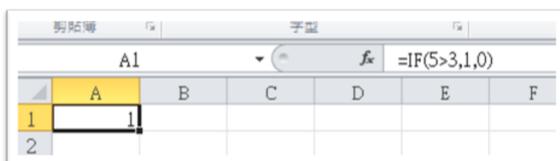
= IF(logical_test, value_true, value_false)

其中,

- logical_test: 邏輯運算式, 運算結果為 TRUE, 或 FALSE。
- value_true: 邏輯運算式為 TRUE 時, 傳回的值。
- value_false: 邏輯運算式為 FALSE 時, 傳回的值。

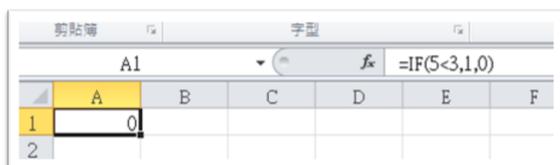
=IF() 有三個參數, 第一個參數是邏輯運算式, 運算的結果不是“TRUE”, 就是“FALSE”, 依此判斷回傳要是第二個參數或第三個參數。

舉個例子,



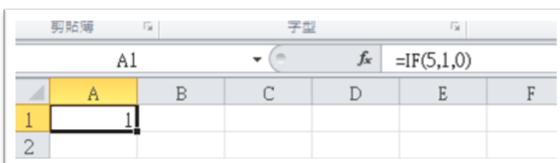
剪貼簿		字型				
A1		fx =IF(5>3,1,0)				
	A	B	C	D	E	F
1	1					
2						

5 的確是大於 3, 所以回傳 1。然後把它反過來,



剪貼簿		字型				
A1		fx =IF(5<3,1,0)				
	A	B	C	D	E	F
1	0					
2						

3 大於 5 當然是錯的, 所以回傳 0。若第一個參數是非 0 的數字, 如 5,



剪貼簿		字型				
A1		fx =IF(5,1,0)				
	A	B	C	D	E	F
1	1					
2						

第一個參數檢驗結果為 TRUE, 所以回傳 1; 若第一個參數是 0,

剪貼簿		字型				
A1		fx =IF(0,1,0)				
A	B	C	D	E	F	
1	0					
2						

則為 FALSE, 回傳 0。

就某一方面而言, 0 被 Excel 保留為 FALSE, 所以當 0 放在 =IF() 的第一個參數時, Excel 自動轉換為 FALSE, 把剩下的非 0 數字全部當 TRUE。就另一方面, Excel 認定 FALSE 為 0, 認定 TRUE 為 1。

若第一個參數隨便輸入一個字串, 如 egg,

剪貼簿		字型				
A1		fx =IF(egg,1,0)				
A	B	C	D	E	F	
1	#NAME?					
2						

會出現錯誤, 因為第一個參數無法定義, 但有二個例外, 第一個是 TRUE,

剪貼簿		字型				
A1		fx =IF(TRUE,1,0)				
A	B	C	D	E	F	
1	1					
2						

給定第一個參數是 TRUE, 當然 =IF() 函數回傳為 TRUE 的 1, 另一個是 FALSE,

剪貼簿		字型				
A1		fx =IF(FALSE,1,0)				
A	B	C	D	E	F	
1	0					
2						

第一個參數是 FALSE, 回傳自然會是設定的 0。

=IF() 函數有三個參數, 有些時候第三個參數可以省略不鍵入, 如此一來, 當第一個參數檢驗為 TRUE 時, 它還是回傳第二個參數沒問題, 但若檢驗為 FALSE 時, 它是直接回傳 “FALSE”, 如

剪貼簿		字型				
A1		fx =IF(3>5,1)				
A	B	C	D	E	F	
1	FALSE					
2						

另外,

A1		fx				
A	B	C	D	E	F	
1	3					
2						

第一個 =IF() 為 “FALSE”, 其實它是 “0”, 加上第二個 =IF() 為 1, 所以 $0+1+2 = 3$, 由此可知 =IF() 函數產出的結果 0 或 1 是可以拿來計算的。因為這個特性, 當邏輯條件為 FALSE 且無足計算輕重時, 我們就可以只輸入前二個參數, 讓第三個參數依 Excel 內建為 FALSE 或你需要的 0。

如果 =IF() 函數中有任何參數是陣列,¹ 則執行 =IF() 陳述式時, 會計算陣列的每個元素。

以上是單一條件在 =IF() 函數中, 若要檢驗 2 個以上條件, 就要配合 =AND()、=OR() 函數了。

把 =AND() 代入 =IF() 函數的第一個參數, 假設資料為 $A2=65$, $A3=35$, 檢驗 $A2$ 儲存格是否大於 40, 同時小於 80, 且為奇數等三個條件,

$$= IF(AND(A2>40, A2<80, MOD(A2, 2)=1), 1, 0) = 1$$

A1		fx								
A	B	C	D	E	F	G	H	I		
1	1									
2	65									
3	35									

另一個表現方式為:

$$= IF((A2>40)*(A2<80)*(MOD(A2, 2)=1), 1, 0) = 1$$

A1		fx								
A	B	C	D	E	F	G	H	I		
1	1									
2	65									
3	35									

¹陣列: 用來建立產生多個結果或運算一組以列及欄排列之參數的單一公式。陣列範圍共用一個公式; 一個陣列常數是用作一個參數的一組常數。

=MOD()

fx

函數傳回相除後之餘數。

= MOD(number, divisor)

其中,

- number: 要計算餘數的數字。
- divisor: number 的除數。

舉個例子, = MOD(65, 3) = 2, 意思是 65/3 後, 商是 21, 餘數是 2, =MOD() 函數就是指餘數。

其中 =MOD() 函數可以用 =ISODD() 邏輯函數取代, 如

= IF((A2>40)*(A2<80)*ISODD(A2), 1, 0) = 1

	A1	B	C	D	E	F	G	H	I
1	1								
2	65								
3	35								

=ISODD()

fx

檢查數字是否為奇數。

= ISODD(number)

其中,

- number: 要檢查的數字。

若是奇數則傳回“TRUE”, 若為偶數則傳回“FALSE”; 舉個例子, = ISODD(11) = TRUE, 檢查 11 是否為奇數, 傳回 TRUE, 因此 11 為奇數。另外, 若要檢查數字是否為偶數則可以用 =ISEVEN() 函數,

=ISEVEN()

fx

檢查數字是否為偶數。

= ISEVEN(number)

其中,

- number: 要檢查的數字。

若是偶數則傳回“TRUE”,若為奇數則傳回“FALSE”;舉例, = ISEVEN(11) = FALSE, 得知 11 非偶數。

=AND() 之所以可以用在 =IF() 函數的第一個參數, 是因為它的回傳值不是 TRUE 就是 FALSE, 剛好符合 IF() 第一個參數的特性。

除了 =AND() 函數是檢驗 2 個以上條件是否成立外, 可以利用另一個方式 “*” 運算子來測試邏輯,

= (條件一)*(條件二)*(條件三)

條件一、二、三皆成立狀況下, 回傳 1, 反之, 只要其中一條件不成立, 則回傳 0。舉成立例, $1 * 1 = 1$

	A	B	C	D	E	F
1	1					
2	0					

不成立例, $1 * 1 * 0 = 0$

	A	B	C	D	E	F	G
1	1						
2	0						
3							

它有點是利用“0”為 FALSE 及“1”為 TRUE 的特性來檢驗運算出最後答案。

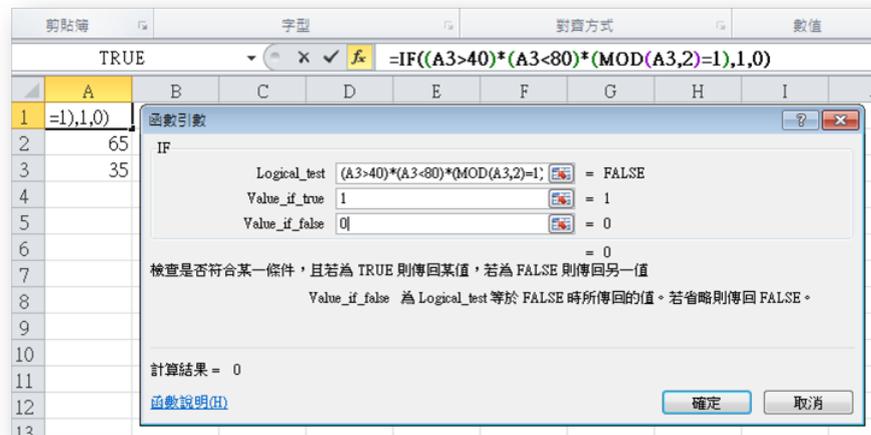
最後,我們直接運用進 =IF() 函數的第 1 個參數,

$$= IF((A3>40)*(A3<80)*(MOD(A3, 2)=1), 1, 0) = 0$$

A3 格並沒有大於 40, 所以得到的回傳值當然是 FALSE 的 0。

	A1	B	C	D	E	F	G	H	I
1	0								
2	65								
3	35								

把游標指到編輯列上的 IF 中, 按一下 *fx* 鈕, 我們看一下 =IF() 函數的第 1 個參數註解,



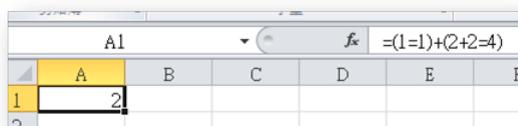
如上圖, 第 1 個參數 Logical_test=FALSE, 不再是 1。

接著, 把 =OR() 代入 =IF() 函數的第一個參數, 承前面資料 A2=65, A3=35, 套在 =IF() 函數中,

$$= IF(OR(A2>60, A3<30), 1, 0) = 1$$

	A1	B	C	D	E	F	G
1	1						
2	65						
3	35						

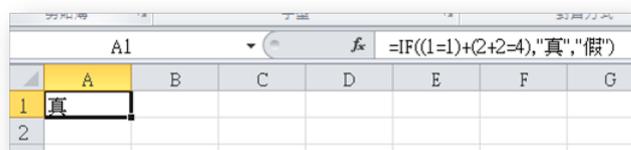
相對 =AND() 函數有運算子 “*” 的使用, =OR() 函數也有取代的運算子 “+”, 但是運用上有些不一樣, 因為 TRUE 為 1, 所以 2 個條件成立時會有 “1 + 1 = 2” 的狀況出現。



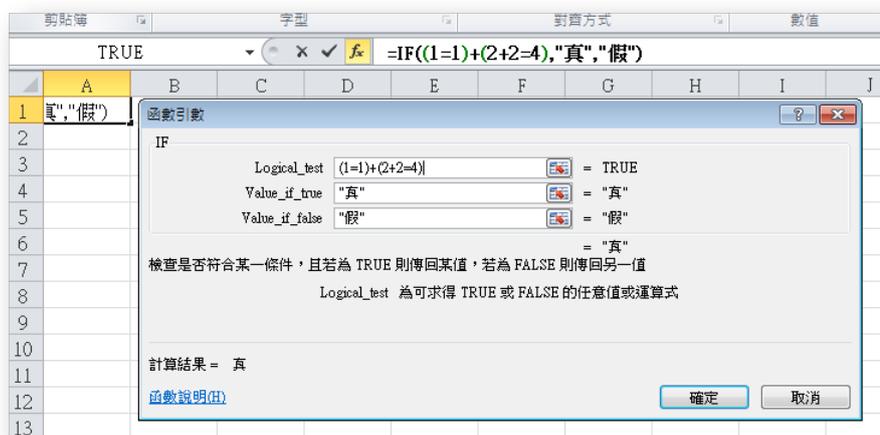
運用進 =IF() 函數的第 1 個參數,

$= IF((1 = 1) + (2 + 2 = 4), "真", "假")$

回傳的答案為「真」,



用 fx 鈕來檢查第 1 個參數運算結果,



第 1 個參數 Logical_test=TRUE, 不再是 2, 正好呼應了回想前頭所講過的, 當 0 放在 =IF() 的第一個參數時, Excel 自動轉換為 FALSE, 其它剩下的非 0 數字全部當 TRUE, 所以這裡不再是四則運算後的 2。

所以邏輯判斷使用運算子“*”及“+”時，每個條件各自獨立成 TRUE (1)、FALSE (0) 來四則運算，當它進入 =IF() 函數的第 1 個參數時，0 為 FALSE，非 0 的運算結果都為 TRUE，剛好運用成 =AND() 與 =OR() 函數一樣。

$$= IF((A2>60)+(A3<30), 1, 0) = 1$$

A1		fx =IF((A2>60)+(A3<30),1,0)					
	A	B	C	D	E	F	G
1	1						
2	65						
3	35						

至此，我們知道 =AND() 可用 * 運算子來檢測交集，=OR() 要用 + 運算子來測試聯集，接著我們就要開始交、聯集混用了。

根據民法 973 條，男未滿十七歲，女未滿十五歲者，不得訂定婚約。所以可訂婚的條件就是一個交、聯集的應用，男且年齡大於等於 17 歲，或女且大於等於 15 歲，轉換為算式，

$$= IF(OR(AND((D2="男"), (E2>=17)), AND((D2="女"), (E2>=15))), "可訂婚", "不可訂婚")$$

fx =IF(OR(AND((D2="男"),(E2>=17)),AND((D2="女"),(E2>=15))),"可訂婚","不可訂婚")								
D	E	F	G	H	I	J	K	L
性別	年齡							
男	16.2	不可訂婚						
男	17.8							
女	21.3							

這個式子也可以改成

$$= IF(((D1="男") * (E1>=17)) + ((D1="女") * (E1>=15)), "可", "不可")$$

fx =IF(((D2="男")*(E2>=17))+((D2="女")*(E2>=15)),"可","不可")						
D	E	F	G	H	I	J
性別	年齡					
男	16.2	不可				
男	17.8					
女	21.3					

分解來看 =IF() 函數的第一個參數:

- 男的條件: $((D1="男") * (E1>=17))$
- 女的條件: $((D1="女") * (E1>=15))$
- 再把男女條件聯集: $((D1="男") * (E1>=17)) + ((D1="女") * (E1>=15))$
當作 =IF() 的第一個參數。

第一個參數會回傳 TRUE 或 FALSE, 依此來判斷最後答案是 TRUE 的“可”或是 FALSE 的“不可”。

最後在 F 欄往下複製公式就可以檢查全部的人符不符合訂婚的條件了。

fx =IF(((D2="男")*(E2>=17))+((D2="女")*(E2>=15)),"可","不可")						
D	E	F	G	H	I	J
性別	年齡					
男	16.2	不可				
男	17.8	可				
女	21.3	可				
女	15.4	可				

2.2 文字

文字類函數在處理一段字串的組成、擷(截)取、置換、取代、測長度等, 這裡我們挑選較常用的擷取函數, =LEFT()、=RIGHT()、=MID(), 另外介紹 2 個常用的取代函數, =REPLACE()、=SUBSTITUTE()。

2.2.1 =LEFT()

=LEFT()

fx

從文字串的第一(最左邊)個字元傳回特定長度之間的所有字元。

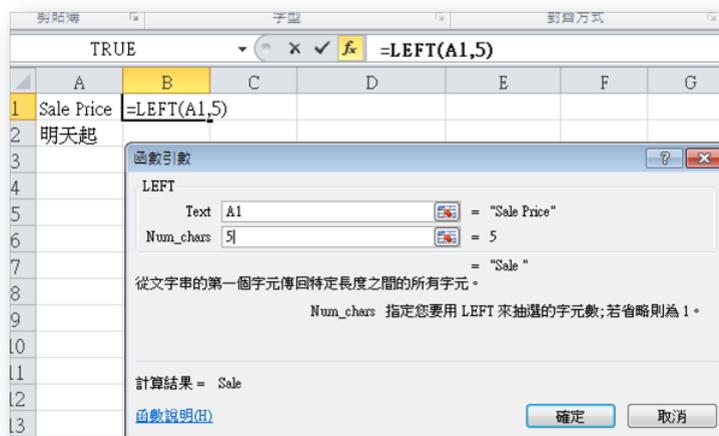
= LEFT(text, num_chars)

其中,

- text: 我們所要抽選之字元的文字串。
- num_chars: 指定我們要用此函數來抽選的字元數, 若省略則為 1。

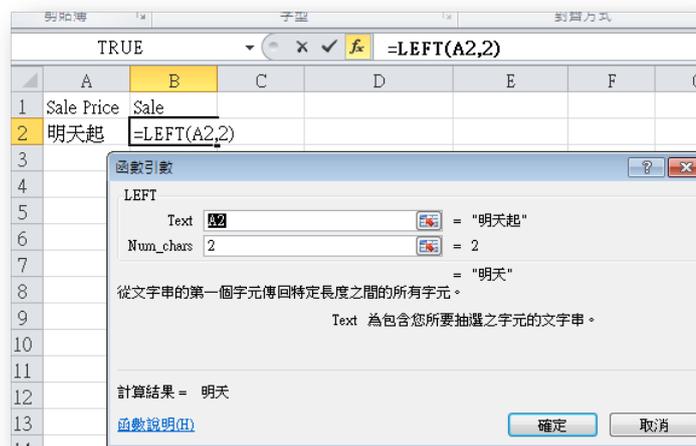
=LEFT() 函數有 2 個參數, 第一個參數是打算被抽取其中部分字元的原始字串, 第二個參數則是想抽取第一個參數自最左端起的字元數。

舉個例子,



“Sale Price” 字左端取 5 個字元, 自然就是 “Sale”, 當中應包含 Sale 及 Price 之間的空白字元。如果第二個參數省略, =LEFT(A1) 則會傳回 “S”, 答案會同於 =LEFT(A1, 1)。

中文例,



“明天起” 自左端取 2 個字元, 就會是 “明天” 2 個字。

2.2.2 =RIGHT()

=RIGHT()

fx

從文字串的最後一 (最右邊) 個字元傳回特定長度之間的所有字元。

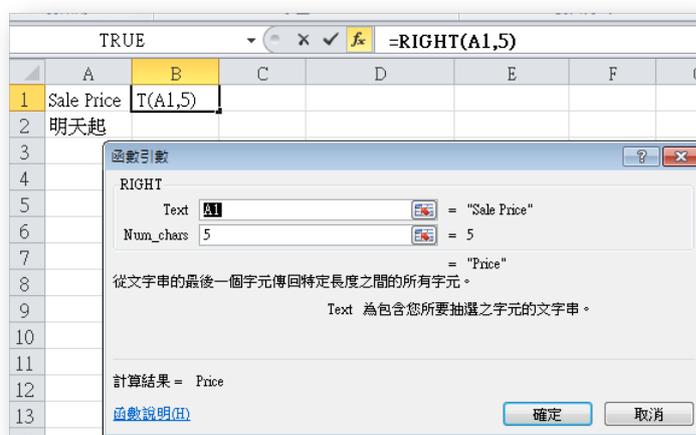
= RIGHT(text, num_chars)

其中,

- text: 我們所要抽選之字元的文字串。
- num_chars: 指定我們要用此函數來抽選的字元數, 若省略則為 1。

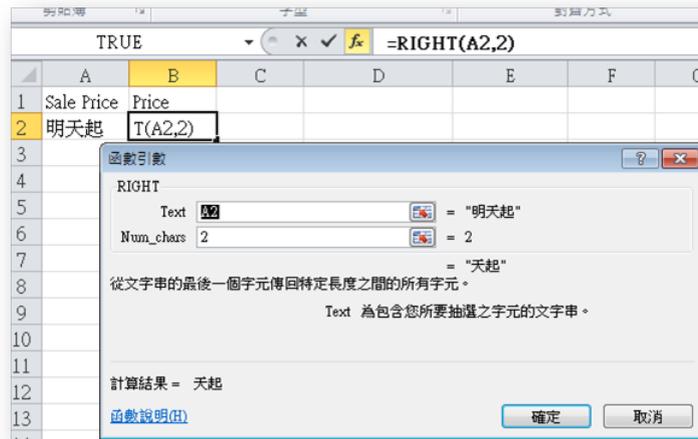
=RIGHT() 函數與 =LEFT() 函數一樣有 2 個參數, 第一個參數是打算被抽取其中部分字元的原始字串, 第二個參數則是想抽取第一個參數自最右端起的字元數。=RIGHT() 與 =LEFT() 的差別在於 =LEFT() 擷取的字元是自左端算起, =RIGHT() 則是自右端算起。

舉個例子,



“Sale Price” 自右端取 5 個字元, 自然就是 “Price” 5 個字元。如果第二個參數省略, =RIGHT(A1) 則會傳回 “e”, 答案會同於 =RIGHT(A1, 1)。

中文例,



“明天起”自右端取 2 個字元，就會是“天起” 2 個字。

2.2.3 =MID()

fx

=MID()

從文字串的某個起始位置到指定長度之間的所有字元。

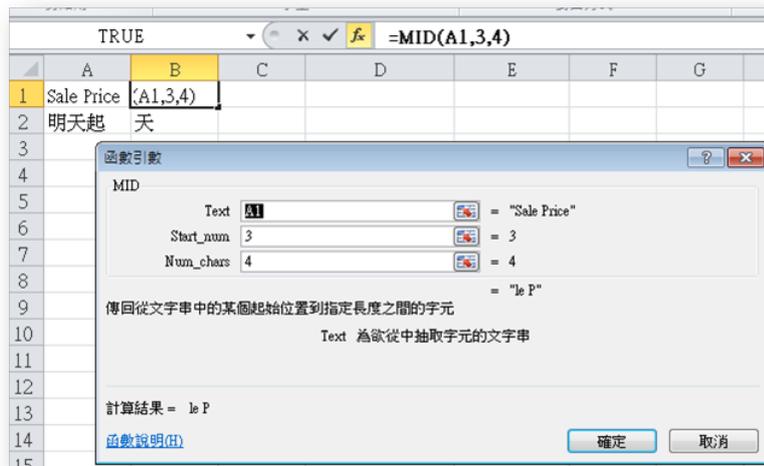
= RIGHT(text, start_num, num_chars)

其中，

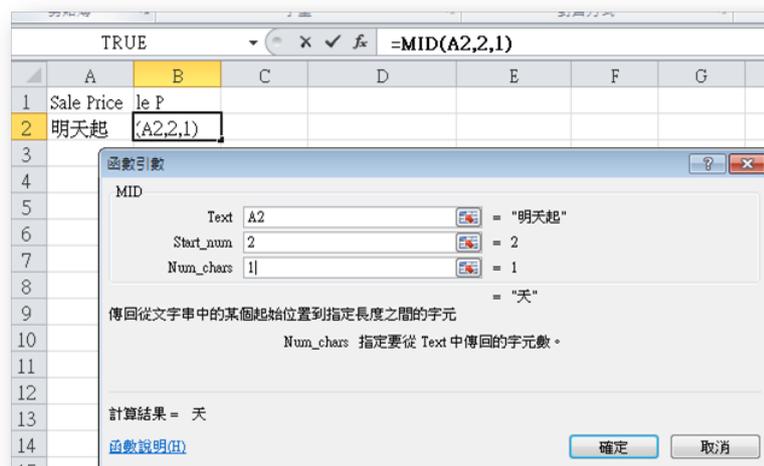
- text: 我們所要抽選之字元的文字串。
- start_num: 我們所要抽取的第一個字元的位置。
- num_chars: 指定我們要用此函數來抽選的字元數。

=MID() 函數有 3 個參數，第一個參數是打算被抽取其中部分字元的原始字串，第二個參數是想抽取的字串起始位置，第三個參數則是自起始位置起要截取多少字元。

舉個例子，



“Sale Price” 自第 3 個字元起擷取 4 個字元，自然就是 “le P” 共 4 個字元。
中文例，



“明天起” 自第 2 個字元起取擷取 1 個字元，就回傳 “天” 1 個字。另外注意，
=MID() 函數就算擷取 1 個字元，第三個參數也不能省略。

=MID() 函數其實可以取代 =LEFT() 及 =RIGHT() 函數，

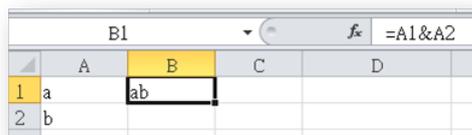
$$= \text{MID}(A1, 1, 3) = \text{LEFT}(A1, 3)$$

A1 儲存格自第一個字元取 3 個字元，就會等於自左邊取 3 個字元。另外，

$$= \text{MID}(A1, \text{LEN}(A1)-3, 3) = \text{RIGHT}(A1, 3)$$

其中 =LEN() 函數會傳回文字字串之字元個數, 用來計算字串長度。=LEN(A1)-3 就為字串長度個數減去 3 個字元數, 所以 =MID(A1, LEN(A1)-3, 3) 為取 A1 儲存格最後 3 個字元, 當然就會等於 =Right(A1, 3)。

以上介紹擷取字元, 若是要合併則要用 “&” 符號。假設 A1 儲存格為 a, A2 儲存格為 b, 把 A1 及 A2 格在 B1 儲存格合併成 “ab”, 則 B1 儲存格格應鍵入 =A1&A2, 別忘記輸入等號 “=”。



	A	B	C	D
1	a	ab		
2	b			

=LEFT()、=MID()、=RIGHT() 及 & 的組合就可以任意拆解排列字串。

2.2.4 =REPLACE()

=REPLACE()

fx

將字串中的一部分以其他字串取代。

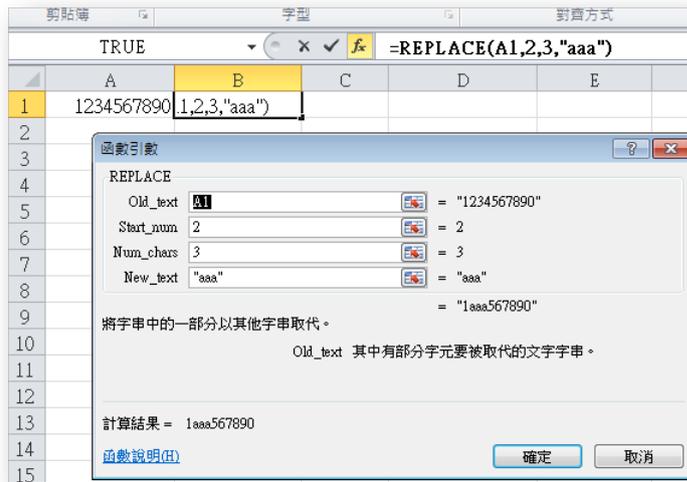
= REPLACE(old_text, start_num, num_chars, new_text)

其中,

- old_text: 其中有部分字元要被取代的文字字串。
- start_num: 在 old_text 中要以 old_text 取代的開始位置。
- num_chars: 被取代的字串長度。
- new_text: 所要換入的新字串。

=REPLACE() 函數有 4 個參數, 第一個參數有部分字元要被取代的原始字串, 第二個參數是想被取代的字串起始位置, 第三個參數是想被取代的字串字元數, 第四個參數則是將被取代的字串換入新的字串。

舉個例子,



儲存格 A1 為 1234567890，今將“234”取代為“aaa”，所以從第 2 個字元取 3 個字元，換入“aaa”，最後結果就為 1aaa56789。

2.2.5 =SUBSTITUTE()

=SUBSTITUTE()

fx

將字串中的一部分字串以新字串取代。

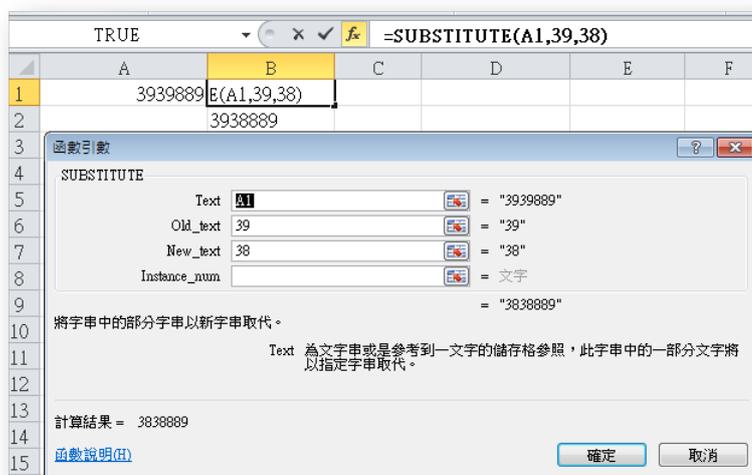
= REPLACE(text, old_text, new_text, instance_num)

其中，

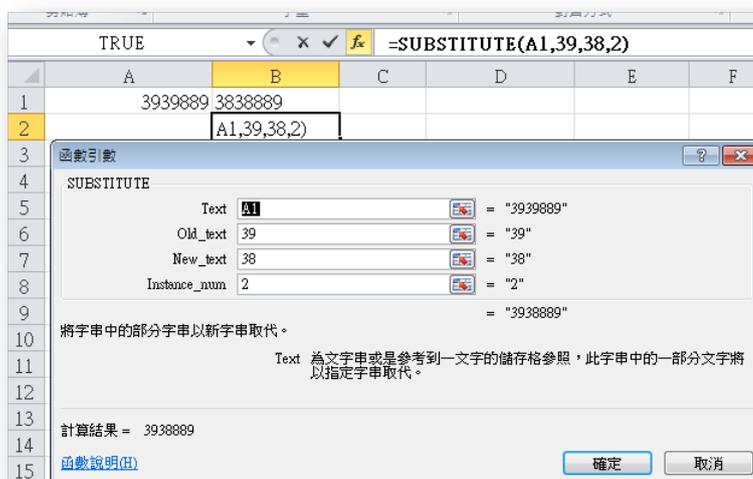
- text: 我們要取代字元的文字，或含有該文字之儲存格的參照。
- old_text: 為要被取代的文字。如果找不到此文字，則將不會執行取代作業。
- new_text: 為將取代 old_text 的新字串。
- instance_num: 當文字串含有數組 old_text 時，此引數用以指定所要被取代的字串是文字串中的第幾組。如果省略此值，每一組皆會被取代。

=SUBSTITUTE() 函數有 4 個參數，第一個參數有部分字元要被取代的原始字串，第二個參數是想被取代掉的字串，第三個參數是想被置入的字串，

第四個參數可省略, 惟被取代掉的字串若有數組時, 可指定為哪一種被取代。
舉個例子,



儲存格 A1 為 “3939889”, 若是把 “39” 取代換成 “38”, 則 A2 儲存格就會變成 “3838889”, 其中二組 39 會置換為 38。若只要指定第二組換成 “38”, 則為 “3938889”。



=REPLACE()、與 =SUBSTITUTE() 都是置換其中字元的函數, 但使用上仍有一點差異, =REPLACE() 是指定字串其中的某幾個字元位置, 如第幾位到第幾位, =SUBSTITUTE() 是指定其中特定字元, 需要指定特定具體字。

2.3 查閱參照

2.3.1 =VLOOKUP()

=VLOOKUP()

fx

在一表格的最左欄中尋找含有某特定值的欄位, 傳回同一列中某一指定欄中的值。

```
= VLOOKUP(lookup_value, table_array,  
           col_index_num, range_lookup)
```

其中,

- lookup_value: 想要搜尋的值, 可以是數值、參照位址或文字串。
- table_array: 要在其中搜尋資料的文字、數字或邏輯值的表格。
- col_index_num: 代表所要傳回的值位於 table_array 中的第幾欄, 1 代表表格中的首欄。
- range_lookup: 為邏輯值。1/TRUE 或省略表示找出首欄中最接近的值 (以遞增順序排序), 0/FALSE 表示僅尋找完全符合的數值。

=VLOOKUP() 函數有 4 個參數, 以最簡單的方式來呈現,

```
= VLOOKUP(想要查的值, 想要查值的範圍, 範圍中包含傳回值的欄號,  
          完全符合或大約符合)
```

很常用在查詢一張表當中的資訊, 如一張全班每個同學段考各科成績, 我們想查其中一位同學某科成績, 便可使用 =VLOOKUP() 函數; 又或者我們想將人轉換成編號等。

假設我們已有台北市 12 個行政區的郵遞區號如 A1 至 B13, 然後我們想要查「士林區」的郵遞區號, 我們可以在 E1 欄輸入

```
= VLOOKUP(“士林區”, A1:B13, 2, FALSE)
```

或者 FALSE 改成 0,

```
= VLOOKUP(“士林區”, A1:B13, 2, 0)
```

	A	B	C	D	E	F	G	H
1	台北市行政區	郵遞區號			111			
2	中正區	100						
3	大同區	103						
4	中山區	104						
5	松山區	105						
6	大安區	106						
7	萬華區	108						
8	信義區	110						
9	士林區	111						
10	北投區	112						
11	內湖區	114						
12	南港區	115						
13	文山區	116						

我們想查士林區，所以以 A1:B13 這台北市行政區郵遞區號的表來查，因為郵遞區號是第 2 欄，所以第 3 個參數輸入 2，我們要完全符合所以第 4 個參數為 FALSE。另外，其中 A1:B13 可以省略成 A:B，

= VLOOKUP(“士林區”，A:B, 2, 0)

第 4 個參數 range_lookup=TRUE 時，要特別注意被查詢的表首欄一定要經過「遞增排列」，否則會出現錯誤。數字要遞增排列比較容易，如過是文字可能就比較困擾，所以建議除非有把握，否則盡量不要用 TRUE 查詢，可另外用 =INDEX() 函數加上 =MATCH() 函數。

舉一個 range_lookup=TRUE 的例子，假設我們要對數學段考成績打上優、甲、乙、... 等第，我們有 A1:B6 的對照表，數學段考分數如 D2:D5，我們可以在 E 欄輸入

= VLOOKUP(D2, \$A\$2:\$B\$6, 2, TRUE)

D2 儲存格分數為 59 分，對照等級表為不及格，所以為「丁」等第，=VLOOKUP() 函數查詢 59 分在 A1:B6 對照表上第 2 欄，找到遞增排序後最接近且不大於等於條件下，傳回「丁」；同理，D3 格 65 分會回傳「丙」。

	A	B	C	D	E	F	G	H
1	成績	等級		數學分數	等級			
2		0	丁	59	丁			
3		60	丙	65	丙			
4		70	乙	77	乙			
5		80	甲	98	優			
6		90	優					
7								

2.3.2 =HLOOKUP()

=HLOOKUP()

fx

在一陣列或表格的第一列中尋找指定值，然後傳回指定值所在那一列 (記錄) 中所要的欄位值。

= HLOOKUP(lookup_value, table_array,
col_index_num, range_lookup)

其中,

- lookup_value: 想要搜尋的值, 可以是數值、參照位址或文字串。
- table_array: 要在其中搜尋資料的文字、數字或邏輯值的表格。
- row_index_num: 代表所要傳回的值位於 table_array 中的第幾列, 1 代表表格中的首列。
- range_lookup: 為邏輯值。1/TRUE 或省略表示找出首欄中最接近的值 (以遞增順序排序), 0/FALSE 表示僅尋找完全符合的數值。

=HLOOKUP() 函數有 4 個參數, 相較於 =VLOOKUP() 只有欄列差別, 其餘都一樣。以最簡單的方式來呈現,

= HLOOKUP(想要查的值, 想要查值的範圍, 範圍中包含傳回值的列號,
完全符合或大約符合)

被查詢的表格中, 想要查的值在第一列就用 =HLOOKUP() 函數, 在第一欄

就用 =VLOOKUP() 函數, 如此一來就看擁有的對找表示欄形式或列形式了。

假設我們已有台北市 12 個行政區的郵遞區號如 A1 至 M2, 然後我們想要查「士林區」的郵遞區號, 我們可以在 A5 欄輸入

= VLOOKUP(“士林區”,1 : 2, 2, FALSE)

或者 FALSE 改成 0,

= VLOOKUP(“士林區”,1 : 2, 2, 0)

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	行政區	中正區	大同區	中山區	松山區	大安區	萬華區	信義區	士林區	北投區	內湖區	南港區	文山區
2	郵遞區號	100	103	104	105	106	108	110	111	112	114	115	116
3													
4													
5		111											
6													

我們想查士林區, 所以以 A1:M2 這台北市行政區郵遞區號的表來查, 我們可以省略到只鍵入列號: 1:2, 第一列到第二列, 又因為郵遞區號是第 2 列, 第 3 個參數所要傳回的資料為 2, 所以第 4 個參數 完全符合為 FALSE。

第 4 個參數 range_lookup=TRUE 時, 與 =VLOOKUP() 函數一樣首列要經過「遞增排列」, 否則會出現錯誤。

舉一個 range_lookup=TRUE 的例子, 假設我們要對數學段考成績打上優、甲、乙、... 等第, 我們有 A1:F2 的對照表, 數學段考分數如 B2:E2, 我們可以在第 5 列輸入

= HLOOKUP(B5,1 : 2, 2, TRUE)

B4 儲存格分數為 59 分, 對照等級表為不及格, 所以為「丁」等第, =HLOOKUP() 函數查詢 59 分在 A1:F2 對照表上第 2 列, 找到遞增排序後最接近且不大於等於條件下, 傳回「丁」; 同理, C4 格 65 分會回傳「丙」。

	A	B	C	D	E	F	G	H
1	成績	0	60	70	80	90		
2	等級	丁	丙	乙	甲	優		
3								
4	數學分數	59	65	77	98			
5	等級	丁	丙	乙	優			
6								

2.3.3 =ROW()

=ROW()

fx

傳回指定儲存格之列號。

= ROW(reference)

其中,

- reference: 為所要求算列號的單一儲存格或儲存格範圍。如果省略, 傳回本函數所在的儲存格列號。

=ROW() 函數有 1 個參數, 該參數為我們所想知道的列號, 如果不填該參數, 則傳回所在位址的列號。

舉例,

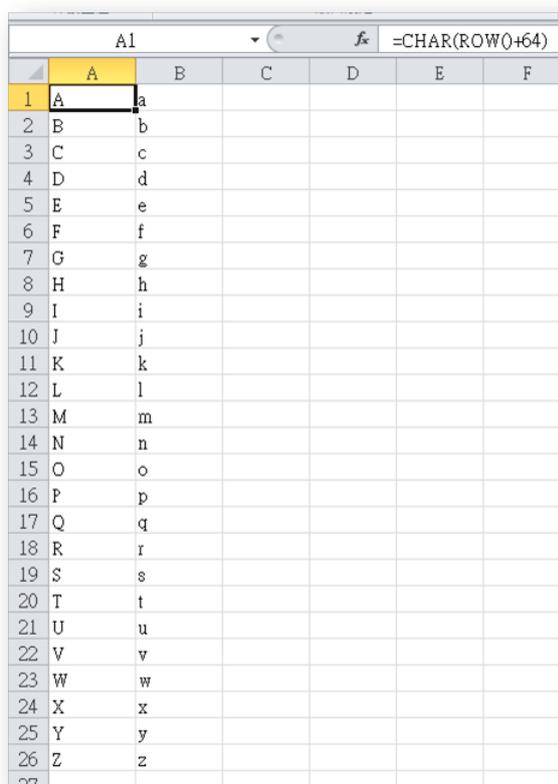
- =ROW(A8) 傳回⇒ 8,
- =ROW(C24) 傳回⇒ 24,
- =ROW(X66) 傳回⇒ 66,

依輸入的指定儲存格回傳所在列號, A8 格的列號為 8, C24 格的列號為 24, X66 格的列號自然為 66。若是不指定參數, 在 A1 儲存格輸入 =ROW(), 則該函數會回傳 1 至 A1 儲存格。

	A	B	C	D	E
1	1				

=ROW() 因為會回傳自己所在列號的特性，當它往下填滿時，會根據所在位置因應傳回不同的值，如此一來特別好運用。

假設我們要在 A1~A26 儲存格產生 A~Z 的英文字母，可以利用 =CHAR() 函數。² 已知 =CHAR(65) 為 A，又儲存格 A1 的 =ROW() 為 1，所以只要我們將儲存格 A1 鍵入 =CHAR(64+ROW())，該格就會回傳“A”；向下複製 A2 格的 =ROW() 為 2，A2 格的 =CHAR(64+ROW())，就為 =CHAR(66)，依此類推，複製填滿至 A26 格，就為 A~Z。

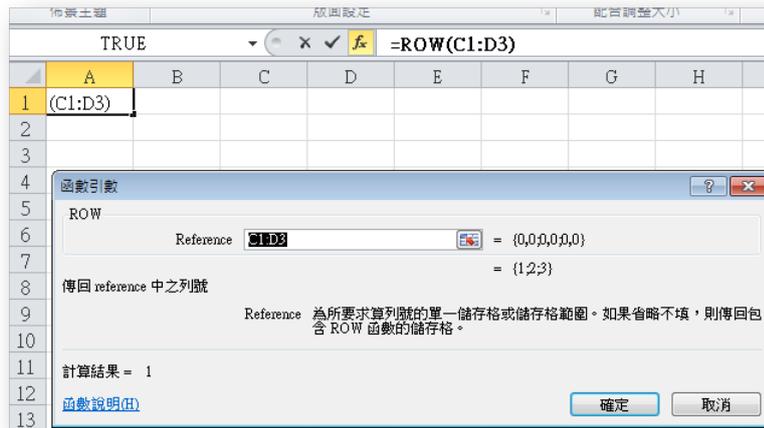


	A1					
	A	B	C	D	E	F
1	A	a				
2	B	b				
3	C	c				
4	D	d				
5	E	e				
6	F	f				
7	G	g				
8	H	h				
9	I	i				
10	J	j				
11	K	k				
12	L	l				
13	M	m				
14	N	n				
15	O	o				
16	P	p				
17	Q	q				
18	R	r				
19	S	s				
20	T	t				
21	U	u				
22	V	v				
23	W	w				
24	X	x				
25	Y	y				
26	Z	z				
27						

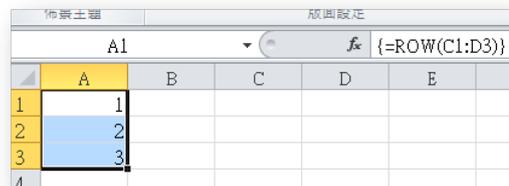
小寫英文字母 a~z 的 =CHAR() 代碼為 97~122。

在 A1 儲存格輸入 =ROW(C1:D3)，按 enter 鍵後，A1 格回傳 1 的答案。我們按 *fx* 鈕來觀看函數計算過程，

²=CHAR() 根據電腦的字元集，傳會代碼所對應的字元。而字母 A 所對應的代碼為 65，字母 B 所對應的代碼為 66，依此類推。

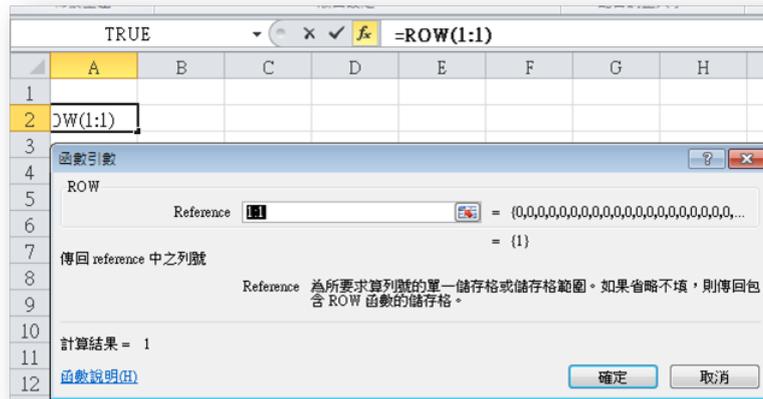


發現 reference 參照窗格是 C1:D3, 參照結果是 $=\{0, 0; 0, 0; 0, 0\}$, 函數運算結果是 $=\{1; 2; 3\}$ 。這二個結果正是前二章介紹的「陣列」, reference 結果是 3×2 的陣列, 函數運算結果是 3×1 的陣列。我們來試試, 選擇 A1:A3, 鍵入 $=\text{ROW}(C1:D3)$, 接著按 $\overbrace{\text{CSE}}$, 它會在 A1:A3 分別輸出 $=\{1; 2; 3\}$ 。



做 $=\text{ROW}()$ 計算時, 若點選單格, 只會回傳陣列的第 1 個元素, 其它元素被忽(省)略, 所以 $=\text{ROW}()$ 函數很自然可以做陣列運算, 而當它做陣列運算時, 因為測度列號, 所以計算結果就會是 $n \times 1$, 因為同一列的列號皆相同, 不同行的資訊就沒有用, 所以會退化為 $n \times 1$, 至於 n 的大小, 就試我們輸入的範圍了。

既然欄的訊息皆被省略了, $=\text{ROW}()$ 的欄號就不重要, 所以不帶欄號的函數 $=\text{ROW}(1:1)$ 會回傳 1, 可是它是包含了一整個第一列, 回傳卻僅只是單一值。



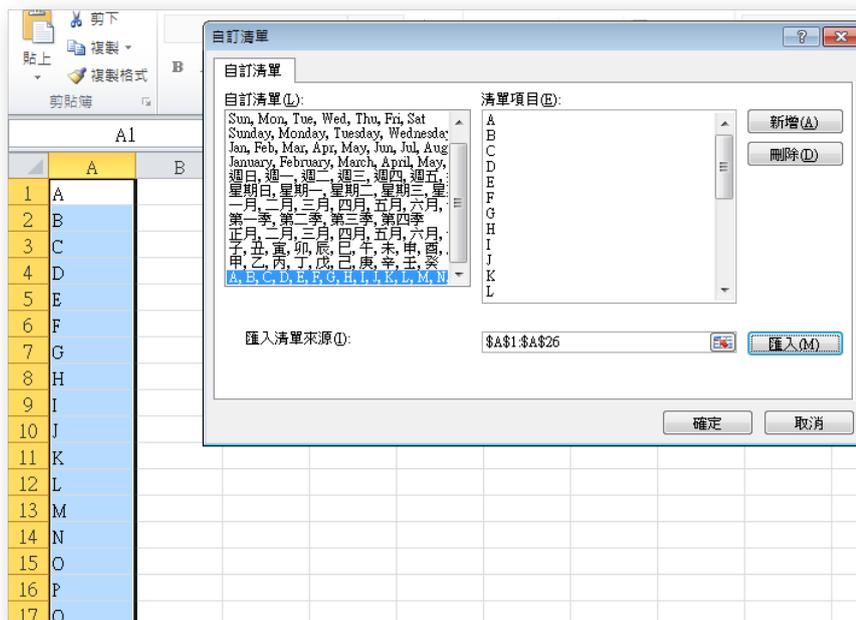
利用 Excel 相對儲存格特性往下複製填滿, =ROW(2:2)、=ROW(3:3)... 都是單一值。回到英文字母的產出, 小寫 a 也可以是 =CHAR(96 + ROW(1:1)), 同理, b 則為 =CHAR(96 + ROW(2:2)), 向下填滿, 就可以得出 a~z。

建立清單以用於排序和填滿序列

附帶一提, 當我們在 A1 儲存格輸入 1, A2 儲存格輸入 2, 點選 A1、A2 儲存格, 向下拖曳填滿控點, Excel 會順序在 A3、A4、A5... 產生連續數字 3, 4, 5...。

試試看, 在 A1 格輸入 A, A2 格輸入 B, 向下拖曳填滿控點是不是沒有產生 C, D, E..., 連續英文字母。沒有產生自動順序填滿的原因主要是「自訂清單」中沒有英文字母, 我們只要在 **檔案 | 選項 | 進階 | 編輯自訂清單** 匯入清單項目就行了。

依上小節得出的 *AsimZ*, 複製後選擇性貼上, 以貼上「值」的方式移除公式, 接著打開 **檔案 | 選項 | 進階 | 編輯自訂清單**, 選取 A~Z, 並按匯入,



左端自訂清單區出現 A, B, C, D,...時, 再按「確定」就完成了。回到 Excel 試算表, 在新的儲存格輸入 A, 向下拖義填滿控點, 就會產生 B 之後的順序字母。

小寫字母依照此方法匯入, 以後就都可依序填滿了。

2.3.4 =COLUMN()

=COLUMN()

fx

傳回指定儲存格之列號。

= COLUMN(reference)

其中,

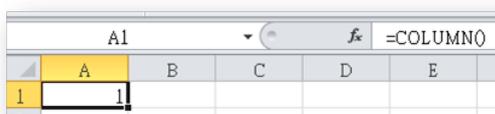
- reference: 為欲取其欄號的儲存格或連續儲存格範圍。如果省略, 傳回本函數所在的儲存格欄號。

=COLUMN() 函數有 1 個參數, 該參數為我們所想知道的欄號, A 轉為 1, B 轉為 2, 依此類推。如果不填該參數, 則傳回所在位址的欄號。

舉例,

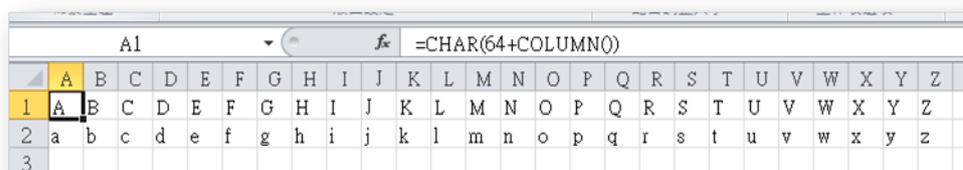
- =COLUMN(A8) 傳回⇒ 1,
- =COLUMN(C24) 傳回⇒ 3,
- =COLUMN(X66) 傳回⇒ 24,

依輸入的指定儲存格回傳所在列號, A8 格為 A 欄, 傳回欄號 1, C24 格為 C 欄, 傳回欄號 3, X66 格為 X 欄, 自然傳回欄號 24。若是不指定參數, 在 A1 儲存格輸入 =COLUMN(), 則該函數會回傳 1 至 A1 儲存格。



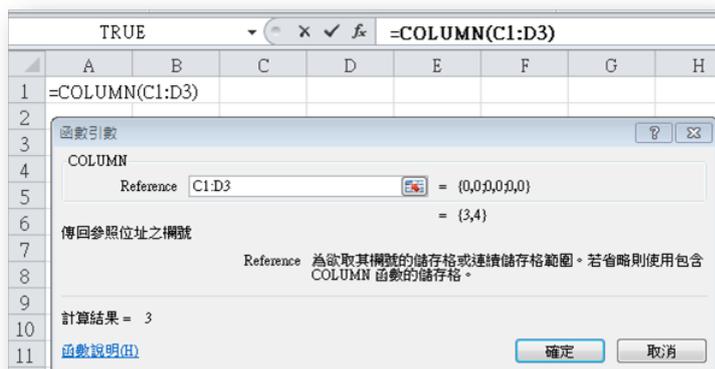
=COLUMN() 因為會回傳自己所在欄號的特性, 當它往右填滿時, 會根據所在位置因應傳回不同的值, 如此一來特別好運用。

假設我們要在 A1~A26 儲存格產生 A~Z 的英文字母, 可以利用 =CHAR() 函數。已知 =CHAR(65) 為 A, 又儲存格 A1 的 =COLUMN() 為 1, 所以只要我們將儲存格 A1 鍵入 =CHAR(64+COLUMN()), 該格就會回傳“A”; 向右複製 B1 格的 =COLUMN() 為 2, B1 格的 =CHAR(64+COLUMN()), 就為 =CHAR(66), 依此類推, 複製填滿至 Z26 格, 就為 A~Z。

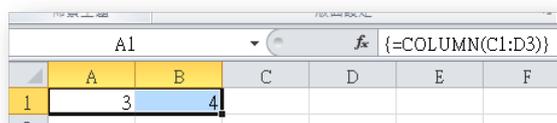


小寫英文字母 a~z 的 =CHAR() 代碼為 97~122。

在 A1 儲存格輸入 =COLUMN(C1:D3), 按 enter 鍵後, A1 格回傳 C1 的欄號 3 的答案。我們按 fx 鈕來觀看函數計算過程,

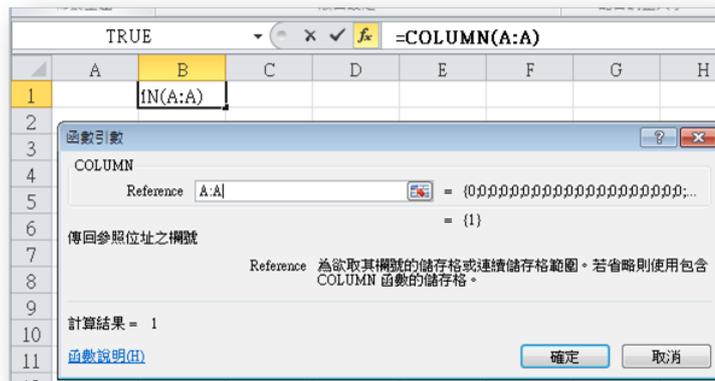


發現 reference 參照窗格是 C1:D3, 參照結果是 $=\{0, 0; 0, 0; 0, 0\}$, 函數運算結果是 $=\{3, 4\}$ 。這二個結果正是前二章介紹的「陣列」, reference 結果是 3×2 的陣列, 函數運算結果是 1×2 的陣列。我們來試試, 選擇 A1:B1, 鍵入 $=\text{COLUMN}(C1:D3)$, 接著按 $\overline{\text{CSE}}$, 它會在 A1:B1 分別輸出 $=\{3, 4\}$ 。



做 $=\text{COLUMN}()$ 計算時, 若點選單格, 只會回傳陣列的第 1 個元素, 其它元素被忽 (省) 略, 所以 $=\text{COLUMN}()$ 函數很自然可以做陣列運算, 而當它做陣列運算時, 因為測度欄號, 所以計算結果就會是 $1 \times m$, 因為同一欄的欄號皆相同, 不同行的資訊就沒有用, 所以會退化為 $1 \times m$, 至於 m 的大小, 就試我們輸入的範圍了。

既然列的訊息被省略了, $=\text{COLUMN}()$ 的列號就不重要, 所以不帶列號的函數 $=\text{COLUMN}(A:A)$ 會回傳 1, 可是它是包含了一整個第一欄, 回傳卻僅只是單一值。



利用 Excel 相對儲存格特性往右複製填滿, =COLUMN(2:2)、=COLUMN(3:3)... 都是單一值。回到英文字母的產出, 小寫 a 也可以是 =CHAR(96+COLUMN(1:1)), 同理, b 則為 =CHAR(96 + COLUMN(2 : 2)), 向右填滿, 就可以得出 a~z。

=ROW(1:1) 與 =COLUMN(A:A) 的差別僅在於前者的 reference 是 {0, 0, 0, 0, ...}, 用逗點隔開, 顯示是一整列陣列; 而後者的 reference 是 {0; 0; 0; 0; ...}, 用分號隔開, 顯示是一整欄陣列。

Exercise 2.1:

給一個對角線, 使 A1 儲存格等於 1, B2 儲存格等於 2, C1 儲存格等於 3, D1 儲存格等於 4, 依此類推到 CV100 儲存格。

C1 儲存格等於 3,

Solution 2.1:

1. A1 格輸入 = IF(COLUMN() = ROW(), ROW(), "")
2. 複製到 CV100 格。

2.3.5 =ROWS()

=ROWS()

fx

傳回陣列或參照位址所含的列數。

= ROWS(array)

其中,

- array: 為陣列、陣列公式或儲存格範圍的參照位址。

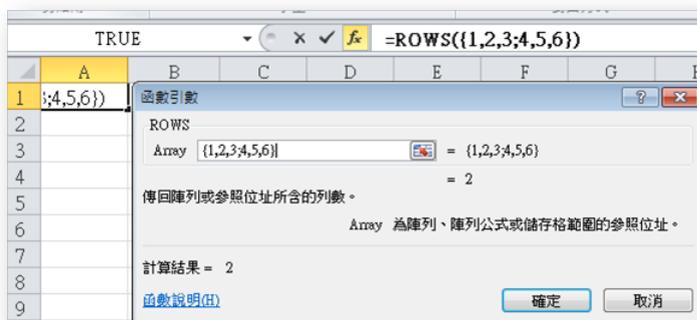
=ROWS() 函數有 1 個參數, 該參數是一個儲存格範圍, 而此函數就是測度該範圍的列數。如果 array 輸入的為常數陣列, 則此函數回傳該常數陣列的列數。

舉例,

- =ROWS(A8) 傳回⇒ 1,
- =ROWS(A8:C24) 傳回⇒ 17,

依輸入的指定儲存格回傳該範圍包含列的個數, A8 格的列號為 8, 而 A8 格的範圍僅占 1 列, 所以 =ROWS() 函數回傳 1; 至於 A8 格到 C24 格的範圍共占 17 列, 所以 =ROWS() 函數回傳 17。

如果參數為一常數陣列, 如 =ROWS({1, 2, 3; 4, 5, 6}), 根據之前章節介紹, 列輸入時用「逗號」隔開, 而欄輸入時用「分號」隔開, 所以 {1, 2, 3; 4, 5, 6} 為 2×3 的陣列, 中文稱作 3 行 (欄) 2 列, 故 =ROWS({1, 2, 3; 4, 5, 6}) 會回傳 2。



特別回顧一下,這裡輸入陣列時需要手動輸入大括號,告訴 Excel 你輸入的是一陣列。

結合 =ROW() 函數,我們得知,沒有“S”的函數 =ROW() 是測度第幾列,有“S”的函數 =ROWS() 是計算共包含幾列。

2.3.6 =COLUMNS()

=COLUMNS()	<i>fx</i>
傳回陣列或參照位址所含的欄數。	
= COLUMNS(array)	
其中,	
<ul style="list-style-type: none">• array: 為陣列、陣列公式或儲存格範圍的參照位址。	

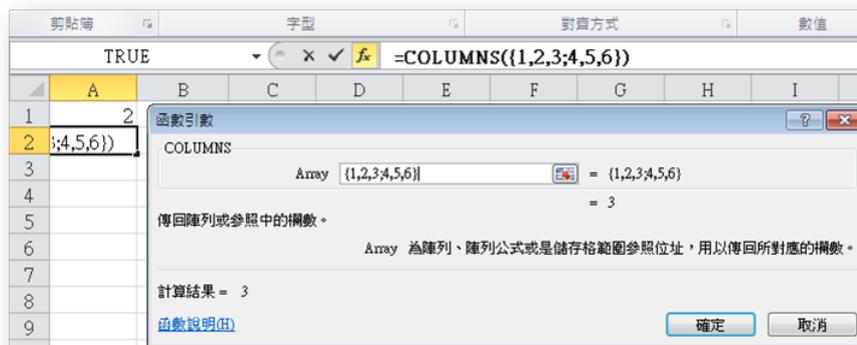
=COLUMNS() 函數有 1 個參數,該參數是一個儲存格範圍,而此函數就是測度該範圍的欄數。如果 array 輸入的為常數陣列,則此函數回傳該常數陣列的欄數。

舉例,

- =COLUMNS(A8) 傳回⇒ 1,
- =COLUMNS(A8:C24) 傳回⇒ 3,

依輸入的指定儲存格回傳該範圍包含欄的個數, A8 格的列號為 8,而 A8 格的範圍僅占 1 欄,所以 =COLUMNS() 函數回傳 1;至於 A8 格到 C24 格的範圍共占 A, B, C 共 3 欄,所以 =COLUMNS() 函數回傳 3。

如果參數為一常數陣列,如 =COLUMNS({1, 2, 3; 4, 5, 6}), 根據之前章節介紹,列輸入時用「逗號」隔開,而欄輸入時用「分號」隔開,所以 {1, 2, 3; 4, 5, 6} 為 2×3 的陣列,中文稱作 3 行(欄) 2 列,故 =COLUMNS({1, 2, 3; 4, 5, 6}) 會回傳 3。



特別回顧一下，這裡輸入陣列時需要手動輸入大括號，告訴 Excel 你輸入的是一陣列。

結合 =COLUMN() 函數，我們得知，沒有“S”的函數 =COLUMN() 是測度第幾欄，有“S”的函數 =COLUMNS() 是計算共包含幾欄。

2.3.7 =OFFSET()

=OFFSET()

fx

傳回根據所指定的儲存格位址、列距及欄距而算出的參照位址。傳回的參照位址可以是單一個儲存格或一個儲存格範圍。可以指定要傳回來的列數和欄數。

= OFFSET(reference, rows, cols, [height], [width])

其中，

- reference: 參照位址，用以計算位移結果的起始位置。
- rows: 用以指示左上角儲存格要垂直（往上或往下）移動的列數。
- cols: 用以指示左上角儲存格要水平（往左或往右）移動的欄數。
- height: 可省略，設定傳回的參照位址應包括的儲存格高度（儲存格範圍的列數）的數值，省略時高度同 reference。
- width: 可省略，設定傳回的參照位址應包括的儲存格寬度（儲存格範圍的欄數）的數值，省略時寬度同 reference。

=OFFSET() 函數有 5 個參數, 前 3 個參數為必要參數, 依著 reference 的位址, 位移 rows 參數的列數及 cols 的欄數, 找到一個新的位址, 傳回該位址的值。

舉例, A1:G2 為一週的中文名及英文名,

	A	B	C	D	E	F	G
1	星期日	星期一	星期二	星期三	星期四	星期五	星期六
2	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
3							
4							
5	Wednesday						
6							

我們在 A5 格輸入,

= OFFSET(A1, 1, 3)

用 A1 格當參照, 往下 (rows) 1 格, 往右 (cols) 3 格, 於是 A1 參照找到 D2, 自然回傳 “Wednesday”。

rows 為正時, 往下走, 為負時, 往上走; cols 為正時, 往右走, 為負時, 往左走。試試看, 若自 G2 格當參照, 要找到「星期二」, 就該為

= OFFSET(G2, -1, -4)

Exercise 2.2:

假設 A1:G2 為中、英文日期名稱,

	A	B	C	D	E	F	G
1	星期日	星期一	星期二	星期三	星期四	星期五	星期六
2	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
3							

我們現在要把 A1:G2 的內容轉置成 A4:B10,

Solution 2.2:

1. 在 A4 格輸入

= OFFSET(\$A\$1,COLUMN()-1,ROW()-4)

2. 複製到 [B4, A5:B10]

[A4:A10] 則是要 [A1:G1], 以 [A1] 當參照位址,

新位址	舊位址	(rows,cols)	轉成公式
[A4]	[A1]	原點	(COLUMN(A4)-1, ROW(A4)-4)
[A5]	[B1]	(0,1)	(COLUMN(A5)-1, ROW(A5)-4)
[A6]	[C1]	(0,2)	(COLUMN(A6)-1, ROW(A6)-4)
[A7]	[D1]	(0,3)	(COLUMN(A7)-1, ROW(A7)-4)
[A8]	[E1]	(0,4)	(COLUMN(A8)-1, ROW(A8)-4)
[A9]	[F1]	(0,5)	(COLUMN(A9)-1, ROW(A9)-4)
[A10]	[G1]	(0,6)	(COLUMN(A10)-1, ROW(A10)-4)
[B4]	[A2]	(1,0)	(COLUMN(B4)-1, ROW(B4)-4)
[B5]	[B2]	(2,0)	(COLUMN(B5)-1, ROW(B5)-4)
[B6]	[C2]	(3,0)	(COLUMN(B6)-1, ROW(B6)-4)
[B7]	[D2]	(4,0)	(COLUMN(B7)-1, ROW(B7)-4)
[B8]	[E2]	(5,0)	(COLUMN(B8)-1, ROW(B8)-4)
[B9]	[F2]	(6,0)	(COLUMN(B9)-1, ROW(B9)-4)
[B10]	[G2]	(7,0)	(COLUMN(B10)-1, ROW(B10)-4)

=COLUMN() 及 =ROW() 函數的參數省略時即為所在儲存格, 所以以上 =COLUMN() 與 =ROW() 函數的參數皆可省略, 最後答案的每個儲存格都為

= OFFSET(\$A\$1,COLUMN()-1,ROW()-4)

請注意參照位址 A1 要固定住成 \$A\$1。

以上皆是省略第 4, 5 個參數例子, 省略 height、width 傳回的範圍皆比照原始參照 (reference) 的大小, 以上都是單格 1×1, 所以傳回的也是單格。請注意, 這點很重要, 如果參照位址是 1×1 則傳回 1×1, 若是參照位址不是 1×1 呢? 我們試著把參照放大為 2×2, 在 E1 儲存格輸入

= SUM(OFFSET(A1:B2, 1, 1))

reference 為 A1:B2 共 4 儲存格, 往下 1 格, 再往右 1 格, 會變成 B2:C3。因為傳回的位址超過一個儲存格的範圍, 所以我們用 =SUM() 函數將之加總起來, 看下圖。

E1		fx =SUM(OFFSET(A1:B2,1,1))				
	A	B	C	D	E	F
1	1	6	11		40	
2	2	7	12			
3	3	8	13			
4	4	9	14			
5	5	10	15			
6						

因為第 1 個參數 reference 為一 2×1 的範圍, 又第 4, 5 參數被省略, 所以參照之後的範圍比照 reference 的 2×1 , 是為 B2:C3, 加總為 $7 + 8 + 12 + 13 = 40$, 為 E1 的數字。

接著我們不要省略第 4, 5 個參數, 同上例, 我們在 E2 格輸入

= SUM(OFFSET(A1:B2, 1, 1, 3, 2))

自 A1:B2 開始, 往右、往下各 1 格, 變成 B2:C3, 又取 3 列 2 欄的高寬, 最後變成 B2:C4。

E2		fx =SUM(OFFSET(A1:B2,1,1,3,2))				
	A	B	C	D	E	F
1	1	6	11		40	
2	2	7	12		63	
3	3	8	13			
4	4	9	14			
5	5	10	15			
6						

所以 E2 回傳的值為 $7 + 8 + 9 + 12 + 13 + 14 = 63$ 。

現在設定為一維, 所以還無法檢視傳回超過一個儲存格的位址, 等進入二維以上的陣列後, 將可以分別得到傳回的範圍。

Exercise 2.3:

假設 A1:B1230 為某基金近 5 年淨值, A 欄為日期, B 欄為淨值, 資料係自 2017/10/20 往前推至 2012/10/19 每日淨值。

重新排列, 倒轉資訊 C1:D1229 為 2012/10/19 至 2017/10/20 資料。

Solution 2.3:

解 1:

C2 格輸入

= OFFSET(\$A\$2, 1229-ROW(), 0)

D2 格輸入

= OFFSET(\$A\$2, 1229-ROW(), 1)

解 2:

E2 格輸入

= INDEX(\$A\$2:\$B\$1229, 1230-ROW(), 1)

F2 格輸入

= INDEX(\$A\$2:\$B\$1229, 1230-ROW(), 2)

解 1 使用 =OFFSET() 函數, 以 [A2] 參照位址

2.4 數學

2.4.1 =INT()

=INT()

fx

傳回無條件捨去後的整數值。

= INT(number)

其中,

- number: 為陣列、陣列公式或儲存格範圍的參照位址。

=INT() 函數有 1 個參數, 該參數為任一數字, 包含有理數、無理數, 而此函數就傳回最接近且不大於 number 數字的整數。

舉例,

- =INT(8.9) 會傳回最接近且不大於 8.9 的整數 8;
- =INT(0.11111) 會傳回最接近且不大於 0.11111 的 0;
- 至於 =INT(-8.9) 會傳回最接近且不大於 -8.9 的 -9, 特別注意, 把小數點以下的數字去掉, 只在於正數, 當參數為負數時, 它應該是以背離於 0 的方式進行捨位, 也就是在數線上, 一律向左取整數;



- =INT(10/3) 會回傳 3, 因為 $10/3 = 3.333\dots$
- =INT(2^0.5) 會回傳 1, 因為 $2^{0.5} = \sqrt{2} = 1.4142\dots$



這個函數是取整數, 意思是到個位數字, 若是要指定位數, 則要靠 =ROUND()、=ROUNDUP()、=ROUNDDOWN() 函數。

2.4.2 =ROUND()

=ROUND()

fx

傳回無條件捨去後的整數值。

= ROUND(number, num_digits)

其中,

- **number**: 為所要執行四捨五入的數字。
- **num_digits**: 為要執行四捨五入計算時所採用的位數。當為負值時, 則表示四捨五入到小數點前的指定位數。當為正值時, 表示到小數點後的指定位數。

=ROUND() 函數有 2 個參數, 第一個參數是要執行四捨五入的數字, 若要四捨五入到小數點右邊的指定位數, 則第二個參數就該指定為正數; 若要四捨五入到小數左邊的指定位數, num_digits 就為負數。

舉例,

- =ROUND(0.9, 0) 回傳 1, 因為四捨五入至個位數, 0.9 進位成 1。
- =ROUND(0.4, 0) 回傳 0, 因為四捨五入至個位數, 0.4 捨去成 0。
- =ROUND(3.345, 1) 回傳 3.3, 因為四捨五入至小數點以下第一位, 小數點以下第二位為 4, 故捨去, 最後變成 3.3。
- =ROUND(3.345, 3) 回傳 3.345, 因為四捨五入至小數點以下第三位, 小數點以下第四位為 0, 捨去後還是變回 3.345。
- =ROUND(3.3, 3) 四捨五入至小數點以下第三位, 因為小數點以下第二位後皆為 0, 並不影響原始輸入的 3.3, 故回傳仍是 3.3, 若要顯示為 3.300, 則要修正該格的「儲存格格式」。



- =ROUND(2^0.5, 3) 回傳 1.414, 之後的 213562... 皆被捨去。
- =ROUND(9.9, -1) 四捨五入至十位數, 個位數為 9, 故進位成 10。



- =ROUND(-0.9, -1) 四捨五入至十位數, 個位數為 0, 故捨去成 0。
- =ROUND(4, -1) 四捨五入至十位數, 個位數為 4, 故捨去成 0。
- =ROUND(5, -2) 回傳 0, 四捨五入至百位數, 十位數為 0, 故捨去成 0。
- =ROUND(494, -2) 回傳 500, 四捨五入至百位數, 十位數為 9, 故進位成 500。

2.4.3 =ROUNDUP()

=ROUNDUP()

fx

將一數字以背離於零之方式無條件進位。

= ROUNDUP(number, num_digits)

其中,

- number: 為所要捨位的數字。
- num_digits: 為所要捨位之小數位數。當為負值時, 則表示捨位至小數點左方的位數。當為正值時, 表示到小數點後的指定位數。

=ROUNDUP() 函數有 2 個參數, 第一個參數是要執行進位的數字, 若要進位到小數點右邊的指定位數, 則第二個參數就該指定為正數; 若要進位到小數左邊的指定位數, num_digits 就為負數。本函數是以背離 0 的方式進位, 意思是正數進位後會大於或等於原始數字, 負數進位後會小於或等於原始數字。

舉例,

- =ROUNDUP(0.9, 0) 回傳 1, 因為 0.9 進位成 1。
- =ROUNDUP(0.4, 0) 回傳 1, 因為 0.4 進位成 1。
- =ROUNDUP(3.345, 1) 回傳 3.4, 因為小數點以下第二位以下為 45, 進位成 3.4。
- =ROUNDUP(3.345, 3) 回傳 3.345, 因為小數點以下第四位為 0, 沒得進位, 還是 3.345。
- =ROUNDUP(3.3, 3) 同理, 沒得進位, 還是 3.3。若要顯示為 3.300, 則要修正該格的「儲存格格式」。
- =ROUNDUP(2^0.5, 3) 回傳 1.415, 因為小數點第四位後大於 0。
- =ROUNDUP(9.9, -1) 進位成 10。
- =ROUNDUP(-0.9, -1) 背於零的方式進位成 -10。
- =ROUNDUP(4, -1) 進位為 10。

- =ROUNDUP(5, -2) 進位為 100。
- =ROUNDUP(494, -2) 進位成 500。

=ROUNDUP() 函數是無條件進位, 例如 =ROUNDUP(1.4012, 1), 進位到小數點第 1 位, 若因為小數點第 2 位是 0 而不進位成 1.4, 那就錯了。這裡要特別注意, =ROUNDUP() 函數是要檢視數線上的點, 除非剛好不偏不倚落在小數點第 1 位上, 否則只要還帶有一點偏差, 就一定要進位, 即使小數點第 2 位是 0, 小數點第 3 位卻有值, 所以 =ROUNDUP(1.4012, 1)= 1.5。



2.4.4 =ROUNDDOWN()

=ROUNDDOWN()

fx

將一數字以趨近於零之方式無條件捨位。

= ROUNDDOWN(number, num_digits)

其中,

- number: 為所要捨位的數字。
- num_digits: 為所要捨位之小數位數。當為負值時, 則表示捨位至小數點左方的位數。當為正值時, 表示到小數點後的指定位數。

=ROUNDDOWN() 函數有 2 個參數, 第一個參數是要執行捨位的數字, 若要捨位到小數點右邊的指定位數, 則第二個參數就該指定為正數; 若要進位到小數左邊的指定位數, num_digits 就為負數。本函數是以趨近 0 的方

式捨位,意思是正數捨位後會小於或等於原始數字,負數進位後會大於或等於原始數字。

舉例,

- =ROUNDDOWN(0.9, 0) 回傳 0, 因為 0.9 捨位成 0。
- =ROUNDDOWN(0.4, 0) 回傳 0, 因為 0.4 捨位成 0。
- =ROUNDDOWN(3.345, 1) 回傳 3.3, 因為小數點以下第二位以下為 45, 全捨去成 3.3。
- =ROUNDDOWN(3.345, 3) 回傳 3.345, 因為小數點以下第四位為 0, 捨去還是 3.345。
- =ROUNDDOWN(3.3, 3) 同理, 還是 3.3。若要顯示為 3.300, 則要修正該格的「儲存格格式」。
- =ROUNDDOWN(2^0.5, 3) 回傳 1.414, 因為小數點第四位後皆捨去。
- =ROUNDDOWN(9.9, -1) 捨去十位數, 個位數以下捨去, 回傳 0。
- =ROUNDDOWN(-0.9, -1) 趨於零的方式捨位成 0。
- =ROUNDDOWN(4, -1) 捨位為 0。
- =ROUNDDOWN(5, -2) 捨位為 0。
- =ROUNDDOWN(494, -2) 個位數及時位數捨去, 成 400。

這邊整理一下, =ROUND()、=ROUNDUP()、=ROUNDDOWN() 三個函數各自輸入參數時的差別。

	=ROUND() 四捨五入	=ROUNDUP() 進位	=ROUNDDOWN() 捨去
$f(0.9, 0)$	1	1	0
$f(0.4, 0)$	0	1	0
$f(3.345, 1)$	3.3	3.4	3.3
$f(3.345, 3)$	3.345	3.345	3.345
$f(3.3, 3)$	3.3	3.3	3.3
$f(2^{0.5}, 3)$	1.414	1.415	1.414
$f(9.9, -1)$	10	10	0
$f(-0.9, -1)$	0	-10	0
$f(4, -1)$	0	10	0
$f(5, -2)$	0	100	0
$f(494, -2)$	500	500	400

2.4.5 =COUNT()

=COUNT()

fx

以取得數值範圍或陣列中, 數值的項目數。

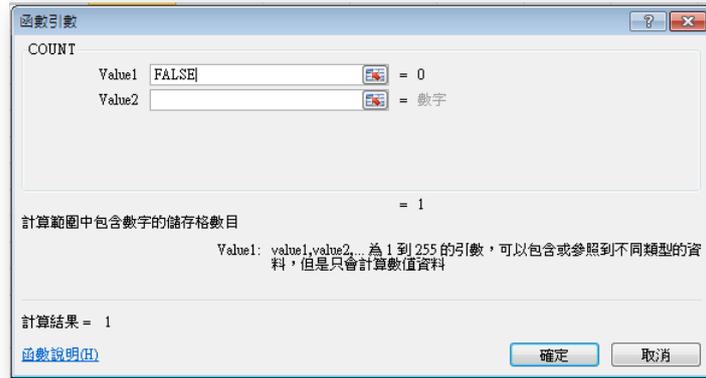
= COUNT(value)

其中,

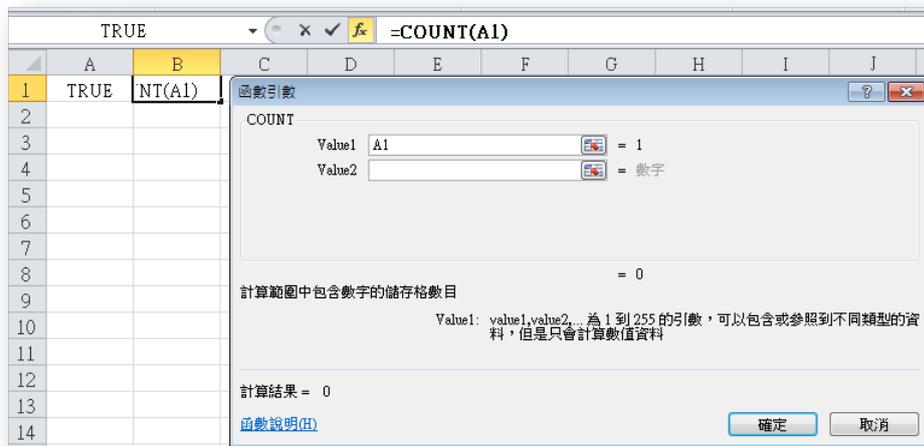
- value: 為所計算其中數字的項目、儲存格參照或範圍。

舉例,

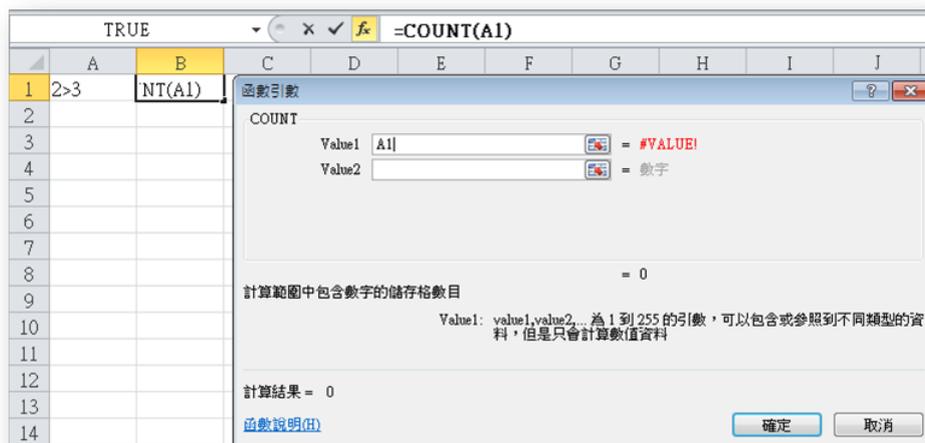
- =COUNT(2) 回傳 1, 1 個參數為數字。
- =COUNT(2, 2) 回傳 2, 2 個項目為數字。
- =COUNT(today()) 會傳回 1, 因為日期也是數字格式之一。
- =COUNT(FALSE) 會計算直接輸入引數清單之邏輯值及數字的文字格式, 所以回傳 1。



- =COUNT(TRUE()) 同上。
- =COUNT(A1) 給定 A1 儲存格為 TRUE, 因為引用邏輯值, 把它當文字不會直接計算為數字, 故回傳。



- =COUNT(2>3) 會回傳 0。若給定 A1 儲存格為 2>3, 則 =COUNT(A1)=0。



依據 Excel 的函數說明：

1. 數字、日期或數字的文字格式 (例如, 以引號括住的數字“1”) 等引數都會進行計算。
2. 會計算直接輸入引數清單之邏輯值及數字的文字格式。
3. 若引數為錯誤值或無法轉換成數字的文字, 則不進行計算。
4. 如果引數為陣列或參照, 則只會計算該陣列或參照中的數字。陣列或參照中的空白儲存格、邏輯值、文字或錯誤值將不會計算。
5. 如果要計算邏輯值、文字或錯誤值, 請使用 =COUNTA() 函數; 若要計算符合特定準則的數字, 請使用 =COUNTIF() 函數或 =COUNTIFS() 函數。

2.4.6 =COUNTA()

=COUNTA()

fx

計算範圍中非空白儲存格的數目。

= COUNTA(value)

其中,

- value: 欲計算的值和儲存格。值可為任何類型的資訊。

依據 Excel 的函數說明:

1. =COUNTA() 函數會計算包含任何類型資訊的儲存格, 包括錯誤值和空白文字 (“”). 例如, 如果範圍包含會傳回空字串的公式, =COUNTA() 函數會計算該值。但 =COUNTA() 函數不會計算空白儲存格。
2. 如果不需要計算邏輯值、文字或錯誤值 (換句話說, 如果您只要計算包含數字的儲存格), 請使用 COUNT 函數。
3. 如果只要計算符合特定準則的儲存格, 請使用 =COUNTIF() 函數或 =COUNTIFS() 函數。

2.4.7 =COUNTIF()

=COUNTIF()

fx

計算範圍內符合指定條件儲存格的數目。

= COUNTIF(range, criteria)

其中,

- range: 欲計算符合給定條件儲存格數目的範圍。
- criteria: 為比較的條件, 條件可以是數字、表示式或文字, 用以指定哪些儲存格會被計算。

舉例,

- =COUNTIF(A1:A20, B1) 計算儲存格 A1~A20 有幾個儲存格等於 B1 格的值。
- =COUNTIF(A1:A20, “=aaa”) 計算儲存格 A1~A20 有幾個儲存格為 aaa 格的值。請注意, 這邊輸入條件時需要雙引號 “” 括起來, 且要加等號。
- =COUNTIF(A1:A20, “>55”) 計算儲存格 A1~A20 有幾個儲存格數字大於 55。
- =COUNTIF(A1:A20, “???s”) 計算 A1~A20 有幾個儲存格為 4 個字母且以 s 結尾。³

³ 問號 (?) 做為萬用字元使用時, 可用來比對「個別」字元。萬用字元 * 用來比對任何字

- =COUNTIF(A1:A20, "<>"&B2) 計算儲存格 A1~A20 有幾個儲存格「不」等於 B2 格的值。⁴

=COUNTIFS()

=COUNTIFS()

fx

當有多個條件同時成立時, 就需要用 =COUNTIFS()。

= COUNTIFS(range1, criteria1, [range2], [criteria2],...)

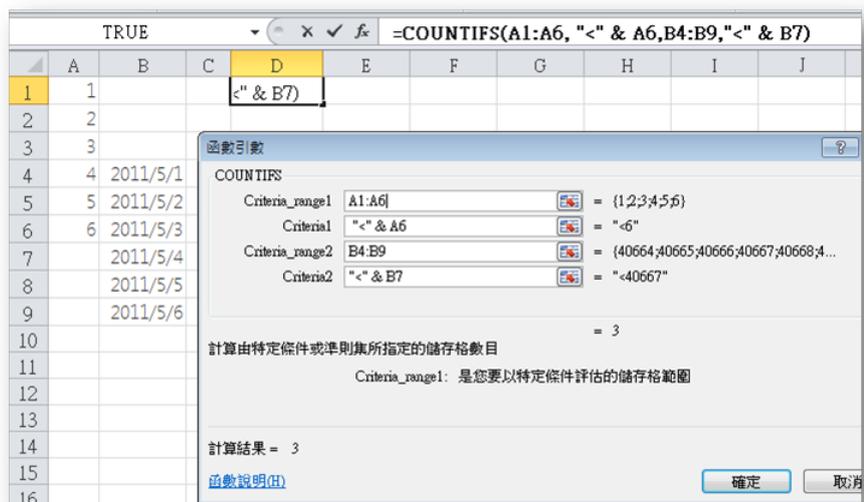
其中,

- range1: 欲計算符合給定條件儲存格數目的範圍。
- criteria1: 為比較的條件, 條件可以是數字、表示式或文字, 用以指定哪些儲存格會被計算。
- range2: 欲計算符合給定的第 2 個條件儲存格數目的範圍, 可省略, 若不省略, 維度應與 range1 相同。
- criteria2: 為比較的第 2 條件。

當 =COUNTIFS() 只有一個條件時, 意思是只有 range1 及 criteria1 而沒有 range2 以上時, 就跟 =COUNTIF() 一樣。

元, 且不限個數字元。

⁴ <> 為不等於符號。



2.4.8 =SUMIF()

=SUMIF()

fx

計算所有符合指定條件的儲存格的總和。

= SUMIF(range, criteria, [sum_range])

其中,

- range: 依據準則進行評估的儲存格範圍。
- criteria: 這是以數字、運算式、儲存格參照、文字或函數的形式來定義要加總哪些儲存格的準則。
- sum_range: 如果您要加總的不是 range 引數中指定的儲存格, 這便是要加總的實際儲存格。如果省略 sum_range 引數, Excel 會加總 range 引數中所指定的儲存格 (即是套用準則的相同儲存格)。

=SUMIF() 函數有 3 個參數, 第一個參數是進行評估條件的範圍, 第二個參數則為條件準則, 第三個參數為符合條件時的加總, 可省略, 若省略狀況下, 加總的範圍變成第一個參數符合條件時的總和。例如, 假設在包含數字的欄中, 只想要加總大於 5 的數值。可以使用下列公式:=SUMIF(B2:B25, ">5")。

另可以視需要將準則套用到一個範圍，然後加總不同範圍中對應的值。例如，公式 =SUMIF(B2:B5, "John", C2:C5) 只會將範圍 C2:C5 中在範圍 B2:B5 中的對應儲存格等於 "John" 的值加總。

=SUMIFS()

=SUMIFS()

fx

同 =COUNTIF() 與 =COUNTIFS() 一樣，當有多個條件時，就需要用 =SUMIFS()。

= SUMIFS(sum_range, range1, criteria1, [range2], [criteria2],...)

其中，

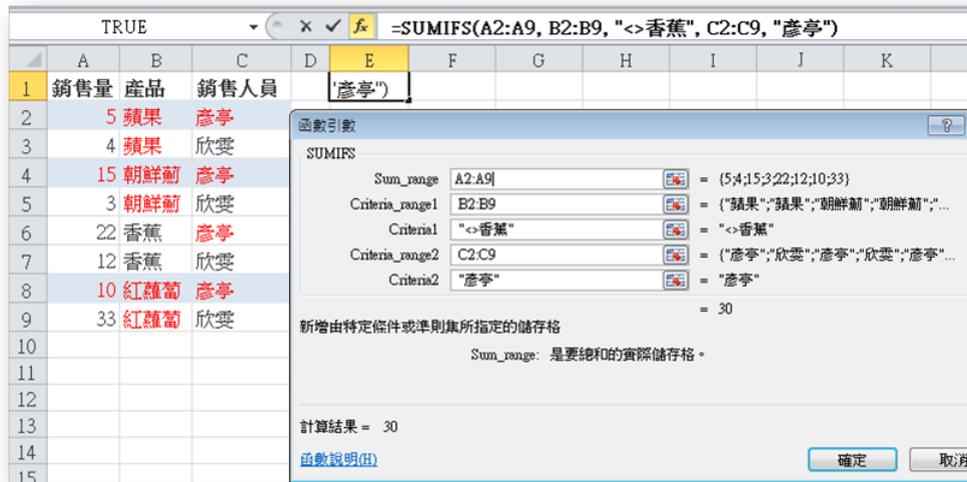
- sum_range: 要加總的實際儲存格。相對 =SUMIF() 單條件的函數 sum_range 由第三個參數變成第一個參數。
- range1: 以特定條件評估的儲存格範圍。
- criteria1: 為比較的條件，條件可以是數字、表示式或文字，用以指定哪些儲存格會被計算。
- range2: 欲計算符合給定的第 2 個條件儲存格範圍，可省略，若不省略，維度應與 range1 相同。
- criteria2: 為比較的第 2 條件。

當 =SUMIFS() 只有一個條件時，意思是只有 range1 及 criteria1 而沒有 range2 以上時，就跟 =SUMIF() 一樣，只是第 1 及 3 個參數互換。

舉一個例子，我們有彥亭、欣雯銷售蘋果、朝鮮薊、香蕉、紅蘿蔔的銷售數字，現我們要計算「彥亭銷售香蕉以外的銷售量」，

= SUMIFS(A2:A9, B2:B9, "<>香蕉", C2:C9, "彥亭")

如圖，



答案為 A2 的 5, A4 的 15, 及 A8 的 10, 合計 $5 + 15 + 10 = 30$ 個。

2.4.9 =RAND()

=RAND()

fx

傳回大於或等於 0 且小於等於 1 的均等分配隨機實數。每次計算工作表時, 都會傳回新的隨機實數。

= RAND()

其中並不需要引數、參數。

傳回的數字是介於 0 跟 1 之間的隨機數字, 等於是抽取一個 $[0, 1]$ 均等分配之間的點, 如果把它想像成連續的線段, $[0, 1]$ 之間就會有無數個點, 每個點被抽中的機率相等且幾乎等於零, 也就是就不必特地考慮 0 跟 1 這兩個點, 就會變成 $(0, 1)$, 因為理論上不會抽中 0 和 1 這二點。

=RAND() 這個函數最大的用處在於選取特定的數字, 如 1~10 之間隨機抽取一個整數, 我們可以

= INT(RAND()*10) + 1

A1		fx				
		=INT(RAND()*10)+1				
	A	B	C	D	E	F
1	6					
2						

$=RAND()*10$ 為 $(0, 1)*10$, 變成 $(0, 10)$, 取了整數 $=INT()$ 函數後, 會變成 $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$, 跟我們想要的 $1\sim 10$ 整數多了 0 , 少了 10 , 所以我們再 $+1$ 往右位移 1 , 就變成 $=INT(RAND()*10) + 1$, 如此就符合了 $1\sim 10$ 的 10 個整數。當然

$$= \text{ROUNDDOWN}(\text{RAND}() * 10 + 1, 0)$$

B1		fx					
		=ROUNDDOWN(RAND()*10+1,0)					
	A	B	C	D	E	F	G
1	1	2	10				
2							

或

$$= \text{ROUNDDOWN}(\text{RAND}() * 10, 0) + 1$$

C1		fx					
		=ROUNDDOWN(RAND()*10,0)+1					
	A	B	C	D	E	F	G
1	1	2	10				

捨棄到個位數, 也得到一樣效果。

也可以是用進位的方式,

$$= \text{ROUNDUP}(\text{RAND}()*10, 0)$$

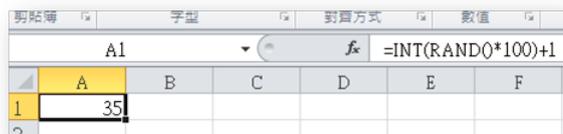
A1		fx					
		=ROUNDUP(RAND()*10,0)					
	A	B	C	D	E	F	G
1	5						

用進位的方式就不必再 +1 了。

特別注意, 若用 =ROUND() 函數四捨五入, 會產生 (0, 0.5) 捨去成爲 0; [0.5,1.5) 成爲 1, 但這 2 段距離不一樣大, 顯示機率不一樣, 若是要在機率一樣的狀況下抽取一個數字, 就不應用 =ROUND()。

所以 1~100 之間隨機抽取一個整數, 就可以是

$$= \text{INT}(\text{RAND()} * 100) + 1$$



若要抽二個整數, 當然可以複製後在下個儲存格貼上, 產生 2 個抽取後的數字, 但這 2 個有可能重複, 因爲每次抽取都是獨立的, 這就是所謂的抽出放回, 若是要抽出不放回, 就要到後面介紹陣列運算的運用了。

2.5 統計

2.5.1 =MAX()

=MAX()

fx

傳回參數中最大值, 邏輯值及文字將被略過而不計。若參數中沒有數字, 則傳回 0。

$$= \text{MAX}(\text{number}_1, [\text{number}_2], \dots)$$

其中,

- number1: 內容可爲數值、陣列、空白儲存格、邏輯值、文字字串。
- number2: 可省略, 最大可至 255 個參數。

傳回的數字是參數中最大值。例如, 在 B1 格計算 =MAX(A1:A5) 會傳回 27, 因爲 {10; 7; 9; 27; 2} 中最大值爲 27。

	A	B	C	D	E	F
1	10	27				
2	7					
3	9					
4	27					
5	2					

2.5.2 =MIN()

=MIN()

fx

傳回參數中最小值，邏輯值及文字將被略過而不計。若參數中沒有數字，則傳回 0。

= MIN(number1, [number2],...)

其中，

- number1: 內容可為數值、陣列、空白儲存格、邏輯值、文字字串。
- number2: 可省略，最大可至 255 個參數。

傳回的數字是參數中最小值。例如，在 B1 格計算 =MIN(A1:A5) 會傳回 2，因為 {10; 7; 9; 27; 2} 中最小值為 2。

	A	B	C	D	E	F
1	10	2				
2	7					
3	9					
4	27					
5	2					

2.5.3 =MEDIAN()

=MEDIAN()

fx

傳回參數中的中位數，邏輯值及文字將被略過而不計。若參數中沒有數字，則產生錯誤。

= MEDIAN(number1, [number2],...)

其中，

- number1: 內容可為數值、陣列、空白儲存格、邏輯值、文字字串。
- number2: 可省略，最大可至 255 個參數。

傳回的數字是參數中的中位數，若參數中的數字為偶數個數，則 =MEDIAN() 會計算中間兩個數字的平均值。例如，在 B1 格計算 =MEDIAN(A1:A5) 會傳回 9，因為 {10; 7; 9; 27; 2} 中經由大至小排序值為 {2; 7; 9; 10; 27}，中間的數字為 9。

	A	B	C	D	E	F
1	10	9				
2	7					
3	9					
4	27					
5	2					

在 B1 格輸入 =MEDIAN(A1:A6)，則 A6="" 空值將被忽略，9 的答案將不會被影響。

	A	B	C	D	E	F
1	10	9				
2	7					
3	9					
4	27					
5	2					
6						

偶數個數的例, {10; 7; 9; 27; 2; 18} 的中位數為經排序後第 3 及第 4 個數字的平均, $1/2(9 + 10) = 9.5$ 。

	A	B	C	D	E	F
1	10	9.5				
2	7					
3	9					
4	27					
5	2					
6	18					

2.5.4 =LARGE()

=LARGE()

fx

傳回第一個參數資料集中的第 k 大的值, 邏輯值及文字將被略過而不計。若參數中沒有數字, 則產生錯誤。

= LARGE(array, k)

其中,

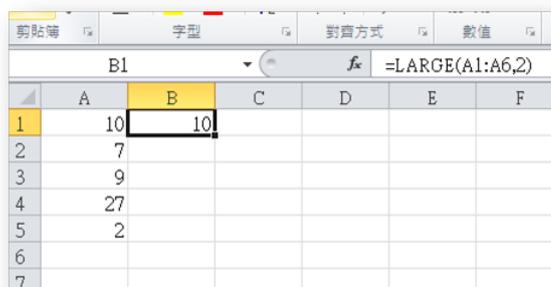
- array: 要判斷第 k 個最大值得陣列或資料範圍。
- k: 要傳回之資料陣列或儲存格範圍中的位置 (由最大起算)。

傳回的數字是參數中的第 k 大值, 若 $k \leq 0$ 或 k 大於資料個數, 資料 =LARGE() 會產生錯誤。例如, 在 {10; 7; 9; 27; 2} 中計算第二大的數字, 則在 B1 格輸入 =LARGE(A1:A5, 2), =LARGE() 函數會回傳 10, 經由排序後 {2; 7; 9; 10; 27}, 第二大的數字為 10。

	A	B	C	D	E	F
1	10	10				
2	7					
3	9					
4	27					
5	2					

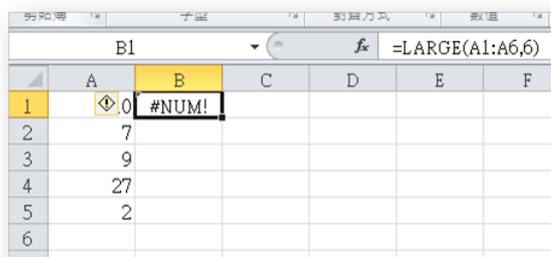
若第二個參數 k 改成 3, =LARGE() 會回傳 9; 若 k 改成 5, 因為個數只有 5 個, 則 =LARGE() 會傳回第五大數字, 就是 2 了, 而跟 =MIN(A1:A5) 一樣取最小值了。同理, =LARGE(A1:A5, 1) 就等於 =MAX(A1:A5) 最大值。

若在 5 個資料數字中輸入 =LARGE(A1:A6, 2), 則因 A6 格為空值, 予以忽略, 答案還是回傳為 10;



	A	B	C	D	E	F
1	10	10				
2	7					
3	9					
4	27					
5	2					
6						
7						

但若是輸入 =LARGE(A1:A6, 6), 在 5 個資料中要求第 6 大, 當然會回傳 #NUM!。



	A	B	C	D	E	F
1	10	#NUM!				
2	7					
3	9					
4	27					
5	2					
6						
7						

2.5.5 =SMALL()

=SMALL()

fx

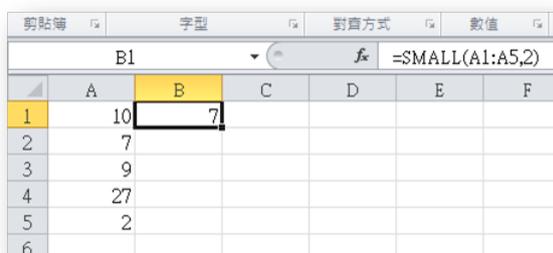
傳回第一個參數資料集中的第 k 小的值, 邏輯值及文字將被略過而不計。若參數中沒有數字, 則產生錯誤。

= SMALL(array, k)

其中,

- array: 要判斷第 k 個最大值得陣列或資料範圍。
- k: 要傳回之資料陣列或儲存格範圍中的位置 (由最小起算)。

傳回的數字是參數中的第 k 小值, 若 $k \leq 0$ 或 k 大於資料個數, 資料 =SMALL() 會產生錯誤。例如, 在 {10; 7; 9; 27; 2} 中計算第二小的數字, 則在 B1 格輸入 =SMALL(A1:A5, 2), =SMALL() 函數會回傳 7, 經由排序後 {2; 7; 9; 10; 27}, 第二小的數字為 7。



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F
1	10	7				
2	7					
3	9					
4	27					
5	2					
6						

The formula bar shows the formula =SMALL(A1:A5,2) and the result 7 is displayed in cell B1.

若第二個參數 k 改成 3, =SMALL() 會回傳 9; 若 k 改成 5, 因為個數只有 5 個, 則 =SMALL() 會傳回第五小數字, 就是 27 了, 而跟 =MAX(A1:A5) 一樣取最大值了。同理, =SMALL(A1:A5, 1) 就等於 =MIN(A1:A5) 最大值。

若在 5 個資料數字中輸入 =SMALL(A1:A6, 2), 則因 A6 格為空值, 予以忽略, 答案還是回傳為 7;

		B1		fx		=SMALL(A1:A6,2)	
	A	B	C	D	E	F	
1	10	7					
2	7						
3	9						
4	27						
5	2						
6							
7							

但若是輸入 =SMALL(A1:A6, 6), 在 5 個資料中要求第 6 小, 當然會回傳 #NUM!。

		B1		fx		=SMALL(A1:A6,6)	
	A	B	C	D	E	F	
1	10	#NUM!					
2	7						
3	9						
4	27						
5	2						
6							
7							

2.6 房貸

一般房貸的本息攤還方式分二種, 一是「本息平均攤還法」, 一是「本金平均攤還法」。

本息平均攤還法顧名思義, 在利率不變下, 每期所應繳金額應一致, 所以又稱年金法, 即將總貸款金額現值換算成年金方式平均分攤在未來還款期間; 因每期所繳金額相同, 而期初本金餘額較大, 導致利息較多, 故所攤本金較少, 所以各期所攤還的本金應為每期應繳金額扣除當期利息, 且會隨著期數增加而增加。本息平均攤還是房貸的主流, 所以我就介紹這一方法。

2.6.1 =PMT()

=PMT()

fx

傳回房貸每期應付款的金額, 其金額包括本金和利息。

= PMT(rate, nper, pv, [fv], [type])

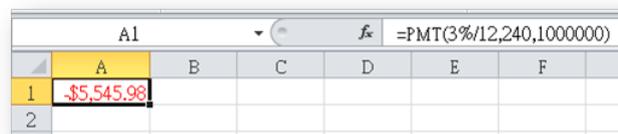
其中,

- rate: 貸款的利率。若繳款週期為「月」, 應將年利率 $\div 12$ 。
- nper: 總付款期數。
- pv: 現值, 即未來各期付款現值總額; 也稱為本金。
- fv: 這是最後一次付款完成後, 所能獲得的未來值或現金餘額。如果省略, 則假設其值為 0, 也就是說, 貸款的未來值是 0。
- type: 選用, 為給付時點。0 或省略時為期末付款; 1 為期初付款。運用在房貸上通常期款是期末繳款, 所以直接省略。

傳回一負數金額, 數字部分為房貸稱之期款, 是為每期所應該付的金額, 負號則指「付出」方向。Excel 的財務函數正負號有資金流動方向得指示, 正號為流入, 負號為流出。在這裡因為 pv 為本金, 是銀行給貸款者, 以貸款者的角度為現金流入, 是為「正號」; 期款是貸款者每期應繳給銀行, 為現金流出, 所以是「負號」。以上是以貸款者的角度看, 若以銀行的角度, pv 應為負數, 得到的期款就為正數。

舉個實際案例, 假設跟銀行貸款 100 萬, 20 年, 年利率 3%, 則每月應繳期款為

$$= \text{PMT}\left(\frac{3\%}{12}, 240, 1000000\right)$$



The screenshot shows an Excel spreadsheet with the formula bar at the top displaying `=PMT(3%/12,240,1000000)`. The spreadsheet grid shows column A and row 1 selected, with the cell A1 containing the result `-$5,545.98`.

	A	B	C	D	E	F	G
1	-\$5,545.98						
2							

房貸為每月繳款, 所以第 1 個參數為貸款利率, 這裡以月為單位, 係年利率除以 12 以為月利率, 第 2 個參數為總付款期數, 是為 20 年乘以每年 12 個月,

共 240 期。最後答案回傳 -\$5,545.98 表示每月期款要付 5545.98 元，本例平均攤還法有個特性是每期期款都一樣，所以 240 期每期都是 5545.98 元。如此一來，20 年總繳金額為

$$240 \times \text{PMT}\left(\frac{3\%}{12}, 240, 1000000\right) = 1331034.2$$

代表借 100 萬，在利率 3% 的狀況下，20 年要繳 133 萬元，扣掉本金 100 萬，所付出的利息就為 331034.2 元。

回到每期期款 5545.98 元，利息的部分應為

$$1000000 \times \frac{3\%}{12} = 2500$$

表示 100 萬一個月利息為 2500 元，5545.98 中扣除利息外，餘下的就為攤還本金，攤本金部分就為 3045.98 元。特別注意，這裡是以 100 萬本金出發計算，期款 5545.98 元，利息 2500 元，攤還本金 3045.98 元這僅是指第 1 期的狀況，因為過了第 1 期攤了一點本金，貸款本金餘額會下降變為 $1000000 - 3045.98 = 996954.02$ 元。所以第 2 期起，期款

$$= \text{PMT}\left(\frac{3\%}{12}, 239, 996954.02\right)$$

	A	B	C	D	E	F	G
1	-\$5,545.98						
2	-\$5,545.98						
3							

這裡注意，總付款期數 nper 會減少 1 期，本金也比 100 萬少了，但最後期款還會是 5545.98 元。雖然期款維持不變，這期的利息與攤還本金不再跟第 1 期一樣，因為本金減少，利息自然就減少，攤還本金的部分會變多，這就是本利平均攤還法的特性。

2.6.2 =CUMIPMT()

=CUMIPMT()

fx

傳回兩個週期之間的貸款上所支付的累計利息。

= CUMIPMT(rate, nper, pv, start_period, end_period, type)

其中,

- rate: 貸款的利率。若繳款週期為「月」, 應將年利率 $\div 12$ 。
- nper: 總付款期數。
- pv: 現值, 即未來各期付款現值總額; 也稱為本金。
- start_period: 要計算一段累計利息的開始期數。
- end_period: 要計算一段累計利息的結束期數。
- type: 為給付時點, 0 為期末付款, 1 為期初付款。

=CUMIPMT() 函數共有 6 個參數, 皆為必要。計算依指定期數之一段期間所支付利息, 回傳答案一樣具有資金流動方向指示的正負號。

延續上一個例子, 假設跟銀行貸款 100 萬, 20 年, 年利率 3%, 則第 1 期利息應為

$$= \text{CUMIPMT}\left(\frac{3\%}{12}, 240, 1000000, 1, 1, 0\right)$$

	A	B	C	D	E	F	G	H
1	-\$5,545.98	-2500						
2	-\$5,545.98							

只要算第 1 期, 所以是第 1 期開始到第 1 期結束, 得到答案 2500 元, 前小節已算過。

如果只要算第 1 期利息, 很容易: $1000000 \times \frac{3\%}{12} = 2500$, 因為我們知道本金 100 萬元, 可是若要算第 2 期, 則要先知道第 1 期攤還多少本金, 剩下的本金要在剩下的期數攤還才能計算。利用本函數可以很容易計算利息, 不必先

去求算每期餘下之本金。第 2 期利息

$$= \text{CUMIPMT}\left(\frac{3\%}{12}, 240, 1000000, 2, 2, 0\right)$$

B2		fx =CUMIPMT(3%/12,240,1000000,2,2,0)							
	A	B	C	D	E	F	G	H	
1	-\$5,545.98	-2500							
2	-\$5,545.98	-2492.39							
3									

為 2492.39 元，符合了前述利息每期減少中。

第 1 期到第 2 期利息

$$= \text{CUMIPMT}\left(\frac{3\%}{12}, 240, 1000000, 1, 2, 0\right)$$

B3		fx =CUMIPMT(3%/12,240,1000000,1,2,0)							
	A	B	C	D	E	F	G	H	
1	-\$5,545.98	-2500							
2	-\$5,545.98	-2492.39							
3		-4992.39							
4									

共 4992.39 元，正是 $2500 + 2492.39 = 4992.39$ 元。

任意算一段區間，如第 8 年利息支出，

$$= \text{CUMIPMT}\left(\frac{3\%}{12}, 240, 1000000, 7 * 12 + 1, 8 * 12, 0\right)$$

B4		fx =CUMIPMT(3%/12,240,1000000,7*12+1,8*12,0)							
	A	B	C	D	E	F	G	H	
1	-\$5,545.98	-2500							
2	-\$5,545.98	-2492.39							
3		-4992.39							
4		-20845.4							
5									

於是第 85 期到第 96 期共要支出利息 20845.4 元。

整個 20 年的利息支出，

$$= \text{CUMIPMT}\left(\frac{3\%}{12}, 240, 1000000, 1, 240, 0\right)$$

	A	B	C	D	E	F	G	H
1	-\$5,545.98	-2500						
2	-\$5,545.98	-2492.39						
3		-4992.39						
4		-20845.4						
5		-331034						
6								

共 331034 元, 跟上一小節所計算總支出扣除本金後答案一樣。

2.6.3 =CUMPRINC()

=CUMPRINC()

fx

傳回兩個週期之間的貸款上所支付的累計本金。

= CUMPRINC(rate, nper, pv, start_period, end_period, type)

其中,

- rate: 貸款的利率。若繳款週期為「月」, 應將年利率 ÷ 12。
- nper: 總付款期數。
- pv: 現值, 即未來各期付款現值總額; 也稱為本金。
- start_period: 要計算一段累計利息的開始期數。
- end_period: 要計算一段累計利息的結束期數。
- type: 為給付時點, 0 為期末付款, 1 為期初付款。

=CUMPRINC() 函數共有 6 個參數, 皆為必要。計算依指定期數之一段期間所攤還本金, 回傳答案一樣具有資金流動方向指示的正負號。

同樣的例子, 假設跟銀行貸款 100 萬, 20 年, 年利率 3%, 則第 1 期應攤還本金為

$$= \text{CUMPRINC}\left(\frac{3\%}{12}, 240, 1000000, 1, 1, 0\right)$$

C1		fx =CUMPRINC(3%/12,240,1000000,1,1,0)						
	A	B	C	D	E	F	G	H
1	-\$5,545.98	-2500	-3045.98					
2	-\$5,545.98	-2492.39						
3		-4992.39						
4		-20845.4						
5		-331034						

只要算第 1 期, 所以是第 1 期開始到第 1 期結束, 得到答案 3045.98 元, 前前小節已算過。

如果只要算第 1 期攤還本金, 只要知道期款, 扣除第 1 期利息即可得到。可是若要算第 2 期攤還本金, 可能要先知道第 1 期攤還多少, 第 2 期期款、本金、利息等, 再求得第 2 期攤還本金。利用 =CUMPRINC() 函數皆可省去以上程序, 第 2 期攤還本金

$$= \text{CUMPRINC}\left(\frac{3\%}{12}, 240, 1000000, 2, 2, 0\right)$$

C2		fx =CUMPRINC(3%/12,240,1000000,2,2,0)						
	A	B	C	D	E	F	G	H
1	-\$5,545.98	-2500	-3045.98					
2	-\$5,545.98	-2492.39	-3053.59					
3		-4992.39						
4		-20845.4						
5		-331034						
6								

為 3053.59 元, 較第 1 期多, 符合了前述攤還本金每期增加中。

第 1 期到第 2 期的攤還本金

$$= \text{CUMPRINC}\left(\frac{3\%}{12}, 240, 1000000, 1, 2, 0\right)$$

C3		fx =CUMPRINC(3%/12,240,1000000,1,2,0)						
	A	B	C	D	E	F	G	H
1	-\$5,545.98	-2500	-3045.98					
2	-\$5,545.98	-2492.39	-3053.59					
3		-4992.39	-6099.57					
4		-20845.4						
5		-331034						

共 6099.57 元, 正是 3045.98 + 3053.59 = 6099.57 元。

2.8 =SUM()

=SUM()

fx

會加總您指定為參數^a的所有數字。每一個參數可以是範圍、^b 儲存格參照、^c 常數、公式、^d 或另一個函數的結果。

= SUM(#1, #2, ..., #255)

其中參數第一個為必要, 其他則可以省略, 最多可以有 255 個。

^a參數: 將資訊提供給動作、事件、方法、屬性、函數或程序的值。

^b範圍: 工作表上的兩個或多個儲存格。範圍中的儲存格可以相鄰或不相鄰。

^c儲存格參照: 儲存格在工作表上佔據的一組座標。例如, 出現在欄 B 與列 3 交叉處儲存格的參照是 B3。

^d公式: 儲存格中共同產生新值的一系列值、儲存格參照、名稱、函數或運算子。公式必定以等號 (=) 開頭。

如果參數為陣列或參照, 則只會計算該陣列或參照中的數字。陣列或參照中的空白儲存格、邏輯值或文字會被忽略。如果有任何參數是錯誤值, 或有任何參數是無法轉換成數字的文字, Excel 會顯示錯誤。

延續前小節的概念, 列舉出 SUM() 函數一些結果:

1. =SUM(3, 2)=5
2. =SUM(TRUE)=1
3. =SUM(TRUE, TRUE)=2
4. =SUM(TRUE, FALSE)=1
5. =SUM("5", 20, TRUE)=26
6. =SUM("TRUE", 1)=#VALUE!
7. Given A1=TRUE, Then =SUM(A1, 1)=1

TRUE 可以當 1 來運算, 但加了引號的 "TRUE" 則會變成一般字串, 失去了 TRUE 的特性。如上述的第 6. 點會成為錯誤。

因為 TRUE 可當成 1 來運算, 結合 SUM() 函數加總參數的特性, 便可以把邏輯函數、IF(++) 放進 SUM() 函數內當參數, 就可以為計數, 加進陣列運算的處理, 就可以一整欄的計算, 如此一來就比樞紐分析表更好用了。

2.9 其它函數

2.9.1 =TRANSPOSE()

陣列函數

第 3 章

綜合函數練習

3.1 參照函數

Exercise 3.1:

給一個對角線, 使 A1 儲存格等於 1, B2 儲存格等於 2, C1 儲存格等於 3, D1 儲存格等於 4, 依此類推到 CV100 儲存格。

C1 儲存格等於 3,

Solution 3.1:

1. A1 格輸入 = IF(COLUMN() = ROW(), ROW(), "")
2. 複製到 CV100 格。

=IF()

=IF(ISERROR((B3-C3)/C3),"-",(B3-C3)/C3) =IFERROR(5/0,"-") =RAND()

常用 =COLUMN()、=ROW() 產生欄號、列號供使用

Exercise 3.2:

假設 A1:G2 為中、英文日期名稱,

	A	B	C	D	E	F	G
1	星期日	星期一	星期二	星期三	星期四	星期五	星期六
2	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
3							

我們現在要把 A1:G2 的內容轉置成 A4:B10,

Solution 3.2:

= OFFSET(\$A\$1,COLUMN()-1,ROW()-4)

Exercise 3.3:

假設 A1:B1230 為某基金近 5 年淨值, A 欄為日期, B 欄為淨值, 資料係自 2017/10/20 往前推至 2012/10/19 每日淨值。

重新排列, 倒轉資訊 C1:D1229 為 2012/10/19 至 2017/10/20 資料。

Solution 3.3:

解 1:

C2 格輸入

= INDEX(\$A\$2:\$B\$1229, 1230-ROW(), 1)

D2 格輸入

= INDEX(\$A\$2:\$B\$1229, 1230-ROW(), 2)

解 2:

E2 格輸入

= OFFSET(\$A\$2, 1229-ROW(), 0)

F2 格輸入

= OFFSET(\$A\$2, 1229-ROW(), 1)

第 4 章

虛擬資料

利用前面所闡述內容來虛擬出一些資料供第 ?? 章運算練習, 所以我們要產出 99 筆資料, 欄位有身分字號 (id)、年齡 (age)、婚姻 (married)、消費日 (date)、消費金額 (amount) 等 5 個變數。

4.1 ID

中華民國的身分證字號有其特定的編碼原則。第一個字是大寫的英文字母, 其餘 9 個字必須為數字, 但在套用編碼原則時, 第一個英文字母將會先依下表被轉換為數字:

字母	A	B	C	D	E	F	G	H	J	K	L	M	N
數字	10	11	12	13	14	15	16	17	18	19	20	21	22
字母	P	Q	R	S	T	U	V	X	Y	W	Z	I	O
數字	23	24	25	26	27	28	29	30	31	32	33	34	35

轉換後的身分證字號 (共 11 位數字) 每一位數均有固定的權重 (weight), 由左往右依序為 {1, 9, 8, 7, 6, 5, 4, 3, 2, 1, 1}。判斷身分證字號是否正確的方法為: 各位數字與其相對應的權重相乘後再加總, 加總後的結果若為 10 的倍數則身分證字號即屬正確。

我們要虛擬初的 ID 不包含以上規則, 只有 ID 的組成共 10 碼, 第 1 碼英文字母, 第 2 碼僅 {1, 2} 來揭露男女, 後面 8 碼皆為數字, 在這裡先不考慮 ID 的規則是否證確, 存粹產生 10 碼外觀看起來像 ID 的 ID, 在 ?? 「自訂函數」篇幅會寫一個產生 ID 的函數, 包含正確規則。

英文字母我用 =char() 性別用 =int(rand()*2+1) 剩下 8 碼用 Rand()+1 放大 8 位數, 使之每位數產生值, 再取 8 位數字 3 個步驟合起來

避免重複

4.2 年齡

4.3 婚姻

4.4 消費日

4.5 消費金額