

## 6.5 統計值與資料大小排序 影片 [Link \(上\)](#) / [Link \(下\)](#)

本節會介紹多個與統計資料相關的函式，將先介紹各函式的用法，再使用範例說明：

- 基本統計資料：

語法	<a href="#">COUNT</a> / <a href="#">SUM</a> / <a href="#">MIN</a> / <a href="#">MAX</a> / <a href="#">AVERAGE</a> / <a href="#">MEDIAN</a> / <a href="#">MODE</a> / <a href="#">STDEV</a> / <a href="#">STDEVP</a> / <a href="#">VAR</a> / <a href="#">VARP</a> ( <b>值_1, [值_2,...]</b> )
輸入	<b>值</b> ：要計算的範圍，可以是多個儲存格，或是單一值都可以，例如 =MIN(A1:A4) 或 =MIN(1, 2, 3, 4, 5)。
輸出	<ul style="list-style-type: none"> <li>COUNT / SUM : <b>值</b> 中的數字數量 / 數字加總。</li> <li>MIN / MAX : <b>值</b> 中的最小值 / 最大值。</li> <li>AVERAGE / MEDIAN / MODE : <b>值</b> 的平均數 / 中位數 / 眾數。</li> <li>STDEV / STDEVP : <b>值</b> 的樣本標準差 / 母體標準差。</li> <li>VAR / VARP : <b>值</b> 的樣本變異數 / 母體變異數。</li> </ul> <p>若上述的 <b>值</b> 包含非數字資料時，將會自動忽略。</p>

Google Sheets 提供另一系列的函式，若遇到 **值** 包含非數字資料時會視為 0，只要將函式名稱加上 A 即可，即 [COUNTA](#) / [MINA](#) / [MAXA](#) / [AVERAGEA](#) / [STDEVA](#) / [STDEVPA](#) / [VARA](#) / [VARPA](#)，其函式輸入與輸出完全相同。

- 四分位數與百分位數的值：

語法	<a href="#">QUARTILE</a> / <a href="#">PERCENTILE</a> ( <b>資料, 四分位數 / 百分位數</b> )
輸入	<ul style="list-style-type: none"> <li><b>資料</b>：要計算的範圍，計算時會將非數字資料會忽略。</li> <li><b>四分位數</b>：QUARTILE 要輸出的四分位數，為 0~4 的整數。</li> <li><b>百分位數</b>：PERCENTILE 要輸出的百分位數，為 0~1 的小數。</li> </ul>
輸出	<b>資料</b> 中的最接第 <b>四分位數 / 百分位數</b> 個四分位數 / 百分位數的值，若介於兩筆資料之間會自動計算與兩者的間距。

- 取得第 n 小 / 大的值：

語法	<a href="#">SMALL</a> / <a href="#">LARGE</a> ( <b>資料, n</b> )
輸入	<ul style="list-style-type: none"> <li><b>資料</b>：要計算的範圍，計算時會將非數字資料會忽略。</li> <li><b>n</b>：要輸出第 n 小 / 大的數字。</li> </ul>
輸出	<b>資料</b> 中第 <b>n</b> 小 / 大的值。

- 取得某個值的排序是第幾大 / 小：

語法	<b>RANK (值, 資料, [遞增])</b>
輸入	<ul style="list-style-type: none"> <li>● <b>值</b>：要計算排名的值，注意一定要出現在 <b>資料</b> 中。</li> <li>● <b>資料</b>：要比較的範圍，比較時會將非數字資料會忽略。</li> <li>● <b>遞增</b>：排序的方式，TRUE 為遞增、FALSE 為遞減，預設為遞減。</li> </ul>
輸出	<b>值</b> 在 <b>資料</b> 中排第幾大 / 小，非數字資料會自動忽略。

- 取得某個值的排序是第幾百分位：

語法	<b>PERCENTRANK (資料, 值, [有效位數])</b>
輸入	<ul style="list-style-type: none"> <li>● <b>資料</b>：要比較的範圍。</li> <li>● <b>值</b>：要計算排名百分比的值，不一定要出現在 <b>資料</b> 中。</li> <li>● <b>有效位數</b>：輸出的百分比要取到小數點後幾位，預設為 3。</li> </ul>
輸出	<b>值</b> 在 <b>資料</b> 中位於多少百分位，在 0~1 之間，若在兩筆資料之間會計算與兩者的間距。若小於等於 MIN( <b>資料</b> ) 或大於 MAX( <b>資料</b> ) 則輸出 #N/A。

以下使用四個基本的範例說明各函式的用法，完成結果如下：

	A	B	C	D	E
1	學生	國文	數學	英文	總分
2	甲	80	75	90	245
3	乙	75	60	75	210
4	丙	70	70	70	210
5	丁	65	80	65	210
6	戊	60	55	80	195
7	己	55	65	55	175
8	庚	50	50	50	150
9	辛	90	95	80	265

#### A. 在 E2:E9 分別計算每一位學生的總分

公式	E2 的公式為 =SUM(B2:D2)。
解說	直接將 B:D 欄的值加總，再複製到第 3~9 列即可。

#### B. 數學分數的各項指標，包括 (1) 平均、(2) 標準差、(3) 最低分、(4) 最高分、(5) 第二高分、(6) Q1~Q3

公式	(1) =AVERAGE(C2:C9) (2) =STDEV(C2:C9) (3) =MIN(C2:C9)	(4) =MAX(C2:C9) (5) =LARGE(C2:C9,2) (6) =QUARTILE(C2:C9,1/2/3)
解說	(6) 的 Q2 即中位數，可使用 =MEDIAN(C2:C9) 代替。	

**C. 計算數學 60 分 (1) 是所有學生的第幾高分、(2) PR 值 (0~99 的整數)**

公式	(1). =RANK(60,C2:C9,0) (2). =PERCENTRANK(C2:C9,60,2)*100
解說	(2). 因為 PR 值為 0~99 的整數，因此要將百分位數取到小數點後兩位再乘上 100。

上述三個範例都相當簡單直觀，接下來將再示範兩個之前介紹過的範例，但使用不同的方法完成：

**D. 將三科平均 > 80 分的學生整列 (A:E 欄) 填滿顏色改為「淺綠色 3」**

結果	A	B	C	D	E
	1	學生	國文	數學	英文
2	甲	80	75	90	245
3	乙	75	60	75	210
4	丙	70	70	70	210
5	丁	65	80	65	210
6	戊	60	55	80	195
7	己	55	65	55	175
8	庚	50	50	50	150
9	辛	90	95	80	265

公式	套用範圍「A1:E9」，自訂公式「=AVERAGE(\$B1:\$D1)>80」。
解說	在第 6.1 節範例 A. 使用三個科目加起來計算，但在這節介紹 AVERAGE 後，可以用 AVERAGE 取代之，公式更直觀簡單。

**E. 將「有兩科以上 (含) 未達 60 分的學生」整列 (A:E 欄) 填滿顏色設定為「淺紅色 3」**

結果	如範例 D. 所示。
公式	套用範圍「A1:E9」，自訂公式「=SMALL(\$B1:\$D1,2)<60」。
解說	在第 6.3 節範例 F. 計算有幾個科目小於 60 分，然後再判斷是否大於等於 2 科，但在此可以直接判斷第二低分的科目是否小於 60 分，如果小於 60 分代表至少有兩個科目未達 60 分。 了解上述內容後，可以嘗試使用 =LARGE(...) 替換上述公式，確保自己是否理解。

## 6.6 包含條件的統計值 影片 [Link \(上\)](#) / [Link \(下\)](#)

本節會介紹包含條件的統計資料相關的函式，又包括單一條件、多條件的函式，將先介紹各函式的用法，再使用範例說明：

- 計算空格數量 / 不重複的值的數量：

語法	<a href="#">COUNTBLANK / COUNTUNIQUE (值_1, [值_2,...])</a>
輸入	<b>值</b> ：要計算的範圍，可以是多個儲存格，或是單一值都可以，例如 =COUNTBLANK(A1:A4) 或 =COUNTUNIQUE(A1:A4, 0, 1)。
輸出	<b>值</b> 中有多少個空白 / 不重複的值。

- 單一條件 — 計算符合條件的數量 / 比例：

語法	<a href="#">COUNTIF / PERCENTIF (條件範圍, 條件)</a>
輸入	<ul style="list-style-type: none"> <li>• <b>條件範圍</b>：要計算是否符合 <b>條件</b> 的範圍。</li> <li>• <b>條件</b>：<b>條件範圍</b> 是否符合某條件，可判斷數字或文字，規則如下：           <ul style="list-style-type: none"> <li>○ 檢查數字：條件可以是單一數字或包含運算子的字串，例如 100、&gt;=100"、&lt;=2023/1/1" 等。</li> <li>○ 檢查文字：條件必須是字串，英文不會區分大小寫，另外可使用萬用字元，包括以下兩個：               <ul style="list-style-type: none"> <li>■ ?：用於比對任何單一字元。</li> <li>■ *：用於比對零個以上的連續字元。</li> </ul>           若要實際比對 ? 或 *，可在其前面加上 ~，例如 ~?。         </li> </ul>           另外，若要在 <b>條件</b> 中使用儲存格的資料，可使用 &amp; 連接儲存格，例如 "&lt;="&amp;A1、"**"&amp;A1&amp;"**" 等。         </li> </ul>
輸出	<b>範圍</b> 符合 <b>條件</b> 的資料筆數 / 比例。

- 單一條件 — 計算符合條件的項目加總 / 平均：

語法	<a href="#">SUMIF / AVERAGEIF (條件範圍, 條件, [加總 / 平均範圍])</a>
輸入	<ul style="list-style-type: none"> <li>• <b>條件範圍</b>：要計算是否符合 <b>條件</b> 的範圍。</li> <li>• <b>條件</b>：<b>條件範圍</b> 是否符合某條件，規則同 COUNTIF。</li> <li>• <b>加總 / 平均範圍</b>：要計算加總 / 平均的範圍，其範圍大小需要與 <b>條件範圍</b> 相同，預設為 <b>條件範圍</b>。</li> </ul>
輸出	<b>條件範圍</b> 符合 <b>條件</b> 的資料中， <b>加總 / 平均範圍</b> 的加總 / 平均。

- 多條件 — 計算符合所有條件的數量：

語法	<b>COUNTIFS</b> ( <b>條件範圍_1, 條件_1, [條件範圍_2, 條件_2,...]</b> )
輸入	<ul style="list-style-type: none"> <li><b>條件範圍_n</b>：要計算是否符合 <b>條件_n</b> 的範圍。</li> <li><b>條件_n</b>：<b>條件範圍_n</b> 是否符合某條件，規則同 COUNTIF。</li> </ul>
輸出	所有 <b>條件範圍_n</b> 皆符合對應的 <b>條件_n</b> 的資料筆數。

- 多條件 — 計算符合所有條件的不重複值數量 / 最小值 / 最大值 / 加總 / 平均：

語法	<b>COUNTUNIQUEIFS / MINIFS / MAXIFS / SUMIFS / AVERAGEIFS</b> ( <b>範圍, 條件範圍_1, 條件_1, [條件範圍_2, 條件_2,...]</b> )
輸入	<ul style="list-style-type: none"> <li><b>範圍</b>：要計算不重複值數量 / 最小值 / 最大值 / 加總 / 平均的範圍。</li> <li><b>條件範圍_n</b>：要計算是否符合 <b>條件_n</b> 的範圍。</li> <li><b>條件_n</b>：<b>條件範圍_n</b> 是否符合某條件，規則同 COUNTIF。</li> </ul>
輸出	所有 <b>條件範圍_n</b> 皆符合對應的 <b>條件_n</b> 者，其 <b>範圍</b> 的不重複值數量 / 最小值 / 最大值 / 加總 / 平均。

以下使用第 1 章的交易記錄檔並搭配九個範例說明各函式的用法，為了方便說明在此已將所有交易記錄檔的各欄位使用已命名範圍命名成表頭名稱：

#### A. 計算交易記錄檔有多少筆交易

公式	=COUNTUNIQUE( <b>交易編號</b> )-1
解說	因為一筆交易若購買多個貨品時會超過一列，因此交易筆數為計算不重複的交易編號，另外因為表頭也會算進去，因此要減掉 1。

#### B. 計算有多少比例的資料店號是 105

公式	=PERCENTIF( <b>店號</b> , 105)
解說	條件也可輸入 "=105"。

#### C. 計算有多少筆的資料貨名是以「台灣」開頭

公式	=COUNTIF( <b>貨名</b> , "台灣*")
解說	在此使用萬用字元 *，即可將條件設為開頭為「台灣」。

#### D. 計算所有 2023 年的訂單總金額

公式	=SUMIF( <b>交易日期</b> , ">=2023/1/1", <b>貨品總金額</b> )
解說	<ul style="list-style-type: none"> <li>訂單總金額為貨品總金額而非交易總金額的加總。</li> <li>交易日期的條件可使用各種 Google Sheets 可辨識的日期寫法，例如 "&gt;=2023-01-01"、"&gt;=2023/01/01"、"&gt;=23/1/1" 等皆可。</li> </ul>

### E. 計算「MATSUSEI背心袋 10KG」在 2023 年有幾筆資料

公式	=COUNTIFS(交易日期,">=2023/1/1",貨名,"MATSUSEI背心袋 10KG")
解說	有兩個條件所以使用 COUNTIFS，條件的順序不限。

### F. 計算「會員卡號 8159」在 2022 年下半年消費總額

公式	=SUMIFS(貨品總金額,交易日期,">=2022/7/1",交易日期,"<=2022/12/31",會員卡號,8159)
解說	有三個條件所以使用 SUMIFS，前兩個為日期區間、第三個為會員卡號。

### G. 計算 2022 年下半年有多少筆訂單金額超過 500 元的訂單

公式	=COUNTUNIQUEIFS(交易編號,交易日期,">=2022/7/1",交易日期,"<=2022/12/31",交易總金額,">500")
解說	要計算有條件的不重複項目使用 COUNTUNIQUEIFS，有三個條件，前兩個為日期區間設定、第三個為訂單金額(交易總金額)超過 500 元。

### H. 將貨號的銷售記錄超過 10 筆的資料整列 (A:L 欄) 填滿顏色改為「淺藍色 3」

公式	套用範圍「A:L」，自訂公式「=COUNTIF(\$G:\$G,\$G1)>10」。
解說	<ul style="list-style-type: none"> <li>在條件式格式設定中不適用已命名範圍，所以使用儲存格範圍。</li> <li>若要判斷貨號是否超過 10 次，可以用 COUNTIF 計算每一列的貨號共出現幾次，再判斷是否 &gt;10 即可。</li> </ul>

### I. 將當日單店銷售金額超過 1,000 元的資料整列 (A:L 欄) 填滿顏色改為「淺綠色 3」

公式	套用範圍「A:L」，自訂公式 「=SUMIFS(\$K:\$K,\$C:\$C,\$C1,\$D:\$D,\$D1)>1000」。
解說	若要判斷當日單店銷售金額是否超過 1,000 元，可使用 SUMIFS 計算該列的店號、交易日期的貨品總金額加總，再判斷是否 >1000 即可。

本節的公式在工作上非常實用，本章最後一節也會進一步介紹如何使用這些公式來建立儀表板 (Dashboard)，絕對是文書處理時最重要的技能之一。

## 6.7 計算加權平均與加總

影片 [Link](#)

本節將介紹 SUMPRODUCT / AVERAGE.WEIGHTED 兩個函式：

- SUMPRODUCT：計算多個對應陣列的乘積加總

語法	<b>SUMPRODUCT (陣列或範圍_1, [陣列或範圍_2,...])</b>
輸入	陣列或範圍_1：可以是陣列或多個儲存格的範圍，例如 {1, 2, 3, 4, 5} 或 A2:A5，每一個陣列大小需相同，不然會跳出 #VALUE!。
輸出	所有 陣列或範圍_n 相對應的乘積，例如 =SUMPRODUCT({1,2,3},{4,5,6}) 為 $(1 \times 4) + (2 \times 5) + (3 \times 6) = 32$ 。

- AVERAGE.WEIGHTED：計算加權平均

語法	<b>AVERAGE.WEIGHTED (值_1, 權重_1, [值_2, 權重_2,...])</b>
輸入	<ul style="list-style-type: none"> <li>• 值_n：要計算平均值的數值，可以是陣列或多個儲存格的範圍。</li> <li>• 權重_n：要套用的權重對應清單，可以是陣列或多個儲存格的範圍，大小需要與 值_n 相同，且不得為負數、至少有一個大於 0 的權重，不然會跳出 #VALUE!。</li> </ul>
輸出	所有 值_n 根據 權重_n 權重的加權平均，例如 =AVERAGE.WEIGHTED({90,70},{3,1}) 為 $[(90 \times 3) + (70 \times 1)] \div (3+1) = 85$ 。

在此使用一個基本範例說明兩函式的用法，並比較異同之處，完成後結果如下。

	A	B	C	D	E
1	權重	30%	40%	30%	100%
2	學生	國文	數學	英文	加權平均
3	甲	80	75	90	81
4	乙	75	60	75	69
5	丙	70	70	70	70
6	丁	65	80	65	71
7	戊	60	55	80	64
8	己	55	65	55	59
9	庚	50	50	50	50
10	辛	90	95	80	89

### A. 計算每位學生的加權平均，其中權重為 B1:D1

公式	E3 的公式為下列任一個都可以： <ul style="list-style-type: none"> <li>• =SUMPRODUCT(B3:D3,B\$1:D\$1)</li> <li>• =AVERAGE.WEIGHTED(B3:D3,B\$1:D\$1)</li> <li>• =AVERAGE.WEIGHTED(B3,B\$1,C3,C\$1,D3,D\$1)</li> </ul>
解說	<ul style="list-style-type: none"> <li>• 權重的列設為絕對參照，以便將公式套用到第 4~10 列。</li> <li>• 在此使用 SUMPRODUCT 與 AVERAGE.WEIGHTED 結果相同，因為 B\$1:D\$1 的加總為 1 (100%)。</li> </ul>

上述範例會覺得兩者的用法相同，但在以下兩種狀況結果可能會不同：

- 權重的顯示方式加總並非 1 (100%)：如下圖，例如要讓加總是 110%，此情況下 AVERAGE.WEIGHTED 會自動換為 100%。

	A	B	C	D	E	F
1					SUMPRODUCT	AVERAGE.WEIGHTED
2	權重	30%	40%	40%	110%	100%
3	學生	國文	數學	英文	加權平均	加權平均
4	甲	80	75	90	90	81.81818182
5	乙	75	60	75	76.5	69.54545455
6	丙	70	70	70	77	70
7	丁	65	80	65	77.5	70.45454545
8	戊	60	55	80	72	65.45454545
9	己	55	65	55	64.5	58.63636364
10	庚	50	50	50	55	50
11	辛	90	95	80	97	88.18181818

- 權重並非以百分比而是用整數顯示：如下圖，此時 SUMPRODUCT 會直接乘上 100，若要避免此問題可以使用 =SUMPRODUCT(B3:D3,B\$1:D\$1)/SUM(B\$1:D\$1) 即可將權重調整為正確的加權平均。

	A	B	C	D	E	F
1					SUMPRODUCT	AVERAGE.WEIGHTED
2	權重	30	40	30	100	100%
3	學生	國文	數學	英文	加權平均	加權平均
4	甲	80	75	90	8100	81
5	乙	75	60	75	6900	69
6	丙	70	70	70	7000	70
7	丁	65	80	65	7100	71
8	戊	60	55	80	6400	64
9	己	55	65	55	5900	59
10	庚	50	50	50	5000	50
11	辛	90	95	80	8900	89

了解兩個函式各自適合的使用時機後，會發現 SUMPRODUCT 除了可以達成所有 AVERAGE.WEIGHTED 計算加權平均之功能，還多出了計算多個陣列相乘的功能，因此建議可以優先記熟 SUMPRODUCT 的用法，而 AVERAGE.WEIGHTED 只需要知道有這個函式即可。

藉由這個範例也可以知道，在 Google Sheets 中有很多函式，功能與輸入方法都沒有很困難，最困難的是如何在有需要時馬上知道可以使用哪些函式完成計算，唯一的方式就是勤加練習讓自己熟悉各種試算表操作時可能遇到的問題，之後面對到一樣或類似的情況時可以快速的解決。

## 6.8 進階應用 影片 [Link](#)

本節將會介紹如何在試算表使用交易記錄檔建立一個儀表板 (Dashboard)，是工作中非常實用的技能。在此張儀表板中，使用者可以在 E2 儲存格選取店號，並在下方顯示出該店號的銷售額、訂單數、平均訂單價值，以及各期的 QoQ (與上一季相比)、YoY (與去年同期相比) 的成長率，完成結果如下圖。

但因為整個過程較為複雜，在此不一一說明各步驟的完成方式，而是在教學影片中示範，在影片中將會從零開始，示範設計儀表板的邏輯、公式的思考脈絡與調整，以及如何快速調整格式等內容，非常推薦各位同學有空的時候跟著影片一起思考並操作，可以從中學到非常多的東西。

本章完成檔案：[+ Chapter 6. 函式（一）：邏輯與運算 完成版](#)

## Appendix 6. Excel 函式 (一)：邏輯與運算

### 6.1 運算符號、布林值與參照

Excel 中沒有四則運算 (ADD / MINUS / MULTIPLY / DIVIDE) 的函式，但影響不大，因為直接使用符號運算即可。

另外，在 Google Sheets 中允許函式使用 A2:A (從第二列開始的整列)、C1:1 的範圍設定，但在 Excel 中不支援此用法，只能選擇整列或是部分範圍，例如 A:A、A2:A10000。

### 6.2 資料錯誤的型態

在 Excel 中有 #NULL! 錯誤，出現在公式中使用不正確的範圍運算子，或是在多個參照的範圍之間使用空格連接時。在 Google Sheets 中則會顯示 #ERROR!。

### 6.6 包含條件的統計值

Excel 中沒有 COUNTUNIQUE、PERCENTIF、COUNTUNIQUEIFS。

### 6.7 計算加權平均與加總

Excel 中沒有 AVERAGE.WEIGHTED。

### 6.8 進階應用

因為此範例會使用到 COUNTUNIQUEIFS，因此使用 Excel 的同學可以先跳過此節，帶後面學習到更多函式再尋找替代做法。

## Chapter 7. 函式 (二)：查詢與篩選 影片 [Link](#)

本章示範檔案： [Chapter 7. 函式 \(二\)：查詢與篩選](#)

本章前五節將介紹查詢相關的函式，包括最常聽到的 VLOOKUP、INDEX + MATCH 等，後兩節介紹功能類似篩選器的函數，最後一節再使用兩個進階應用來總結本章所有的公式，本章的函式都非常實用且重要，熟悉後可以有效率的解決工作中大部分的問題。

### 7.1 查詢函式：儲存格位置 影片 [Link \(上\)](#) / [Link \(下\)](#)

本節將介紹四種不同功能的函式，主要用於儲存格位置相關資訊之轉換，將先介紹各函式的用法，再使用範例說明之。各函式說明如下：

- 儲存格所在位置：

語法	<a href="#">ROW</a> / <a href="#">COLUMN</a> ([ <b>儲存格參照</b> ])
輸入	<b>儲存格參照</b> ：要輸出哪個儲存格所在的列 / 欄，預設為公式所在的儲存格。若儲存格參照包含多個儲存格，會輸出最左上角的儲存格。
輸出	<b>儲存格參照</b> 所在的列 / 欄，其中欄的 A=1、B=2，以此類推。

- 儲存格範圍的列 / 欄數量：

語法	<a href="#">ROWS</a> / <a href="#">COLUMNS</a> ( <b>範圍</b> )
輸入	<b>範圍</b> ：要計算哪個範圍的列 / 欄數量。
輸出	<b>範圍</b> 有幾列 / 欄，若範圍超出該工作表的列 / 欄數時，只會計算存在的列數，例如某工作表有 1000 列，若使用 =ROWS(A1:A10000) 會輸出 1000。

- 將儲存格位置以字串顯示：

語法	<b>ADDRESS (列, 欄, [絕對相對模式], [使用 A1 表示法], [工作表])</b>															
輸入	<ul style="list-style-type: none"> <li><b>列 / 欄</b>：第幾列 / 欄。</li> <li><b>絕對相對模式</b>：1~4，預設為 1，其意義分別如下：</li> </ul> <table border="1"> <thead> <tr> <th>絕對相對模式</th><th>1</th><th>2</th><th>3</th><th>4</th></tr> </thead> <tbody> <tr> <td>列</td><td>絕對參照</td><td>絕對參照</td><td>相對參照</td><td>相對參照</td></tr> <tr> <td>欄</td><td>絕對參照</td><td>相對參照</td><td>絕對參照</td><td>相對參照</td></tr> </tbody> </table> <ul style="list-style-type: none"> <li><b>使用 A1 表示法</b>：TRUE / FALSE，預設為 TRUE，即使用 A1 Notation 輸出，若選 FALSE 則會以 R1C1 Notation 的方式輸出。兩個 Notation 的差異如下：           <ul style="list-style-type: none"> <li>A1 Notation：一般公式中表示儲存格的方式，字母與數字分別表示欄、列，若要使用絕對參照則在欄 / 列中加上金錢符號「\$」即可。</li> <li>R1C1 Notation：R / C 分別表示列 / 欄，表示方式是在 R 和 C 後面加上數字表示儲存格的位置，若為相對參照則在數字外包上 []。在 R1C1 Notation 中絕對參照與相對參照的意義不太相同，如下：               <ul style="list-style-type: none"> <li>絕對參照：第幾列 / 欄的資料，例如 R3C2 代表 B3 儲存格，此時 <b>列 / 欄</b> 必須是正數。</li> <li>相對參照：目前的儲存格往上 / 下 / 左 / 右移動幾格，正數代表往右 / 下移動、負數代表往左 / 上移動，例如對 C4 儲存格而言 R[3]C[2] 代表 E7、R[-3]C[-2] 代表 A1。</li> </ul> </li> </ul> </li> <li><b>工作表</b>：儲存格字串的工作表名稱，需為字串格式，預設為空值。</li> </ul>	絕對相對模式	1	2	3	4	列	絕對參照	絕對參照	相對參照	相對參照	欄	絕對參照	相對參照	絕對參照	相對參照
絕對相對模式	1	2	3	4												
列	絕對參照	絕對參照	相對參照	相對參照												
欄	絕對參照	相對參照	絕對參照	相對參照												
輸出	符合各項輸入設定的儲存格字串，例如 ' <b>工作表!</b> \$A1、R[-3]C[-1]'。															

- 輸出儲存格字串的內容：

語法	<b>INDIRECT (儲存格參照字串, [使用 A1 表示法])</b>
輸入	<ul style="list-style-type: none"> <li><b>儲存格參照字串</b>：以字串寫成的儲存格參照。</li> <li><b>使用 A1 表示法</b>：TRUE / FALSE，預設為 TRUE，即使用 A1 Notation 輸出，若選 FALSE 則會以 R1C1 Notation 的方式輸出。</li> </ul>
輸出	<b>儲存格參照字串</b> 的內容。

接下來將使用八個範例說明上述函式的用法：

#### A. 當前儲存格所在列數

公式	=ROW()
解說	因為 ROW 預設為當前所在儲存格，所以可以不用輸入參數。

## B. B3 儲存格所在欄數

公式	=COLUMN(B3)
解說	因為欄的 A=1、B=2，因此在此輸出 2。

## C. B3:C30 的 (1) 列數與 (2) 欄數

公式	(1) =ROWS(B3:C30)	(2) =COLUMNS(B3:C30)
解說	(1) 因為在工作表中只有 14 列，因此會輸出 $14-3+1=12$ 。	

## D. 已命名範圍「會員卡號」的列數

公式	=ROWS(會員卡號)
解說	已命名範圍 會員卡號 共有 31 列，因此輸出 31。

## E. 用字串表示 (1) \$B3、(2) '會員資料檔'!\$B3

公式	(1) =ADDRESS(3,2,3)	(2) =ADDRESS(3,2,3,, "會員資料檔")
解說	(2) 在此要使用 A1 表示法，但 ADDRESS 已經預設使用 A1 表示法，因此可直接留空。	

## F. 儲存格字串 (1) C7、(2) "C7" 的內容

公式	(1) =INDIRECT(C7)	(2) =INDIRECT("C7")
解說	(1) 會將 C7 的內容轉為字串，為 \$B3，因此輸出 B3 儲存格的內容。 (2) 會輸出 C7 儲存格的內容。	

## G. 使用 R1C1 Notation 表示 (1) R3C1、(2) R[-3]C[-1]

公式	(1) =ADDRESS(3,1,1,TRUE)	(2) =ADDRESS(-3,-1,4,TRUE)
解說	在此 TRUE / FALSE 也可以使用 1 / 0 代替。	

## H. 取得 (1) R3C1、(2) R[-3]C[-1] 的內容

公式	(1) =INDIRECT("R3C1",FALSE)	(2) =INDIRECT("R[-3]C[-1]",FALSE)
解說	(1) 在此為絕對參照，因此會輸出 A3 儲存格的內容。 (2) 在此為相對參照，公式所在位置為 C14，因此會輸出第 $14-3=11$ 列、第 $3-1=2$ 欄，也就是 B11 儲存格的內容。	

## 7.2 查詢函式：INDEX / MATCH 影片 [Link](#)

本節將介紹兩個非常重要的函式 — INDEX / MATCH

- INDEX：輸出第 m 列第 n 欄的值：

語法	<b>INDEX</b> ( <b>參照</b> , [ <b>列</b> ], [ <b>欄</b> ])
輸入	<ul style="list-style-type: none"> <li>• <b>參照</b>：一個多欄多列的陣列或儲存格範圍。</li> <li>• <b>列 / 欄</b>：要輸出 <b>參照</b> 的第幾列 / 欄，預設為 0 (輸出全部欄 / 列)。</li> </ul>
輸出	<b>參照</b> 的第 <b>列</b> 列第 <b>欄</b> 欄，若超出範圍將輸出 #NUM!。

- MATCH：尋找某個值出現在第幾列 / 欄：

語法	<b>MATCH</b> ( <b>搜尋準則</b> , <b>範圍</b> , [ <b>搜尋類型</b> ])
輸入	<ul style="list-style-type: none"> <li>• <b>搜尋準則</b>：要找的項目。</li> <li>• <b>範圍</b>：要在哪裡找，單欄或單列的陣列或範圍。</li> <li>• <b>搜尋類型</b>：要怎麼找，有 1 / 0 / -1 三種，預設為 1，使用時機如下：           <ul style="list-style-type: none"> <li>○ 0：完全比對，即 <b>搜尋準則</b> 在 <b>範圍</b> 中有完全一樣的值才會輸出。</li> <li>○ 1：<b>範圍</b> 遞增排列、輸出小於等於 <b>搜尋準則</b> 的最大值所在位置。</li> <li>○ -1：<b>範圍</b> 遞減排列、輸出大於等於 <b>搜尋準則</b> 的最小值所在位置。</li> <li>○ 上述的 1 / -1 僅適用在 search_key 是數字的情形下。</li> </ul> </li> </ul>
輸出	<b>搜尋準則</b> 在 <b>範圍</b> 中的位置，若找不到則會輸出 #N/A。

以下使用第 1 章的會員資料檔並搭配四個範例說明各函式的用法，為了方便說明在此進行以下設定：

- 將所有交易記錄檔的各欄位使用已命名範圍命名成表頭名稱。
  - 將 A:I 命名為「會員資料」、第 1 列命名為「會員資料表頭」。
  - 將會員卡號的格式設定為自動 (數字)。
- A. (1) A 欄的第 5 個元素、(2) 第 1 列的第 3 個元素、(3) 會員資料的第 3 列、第 2 欄的元素

公式	(1) =INDEX('會員資料檔'!A:A,5) (2) =INDEX('會員資料檔'!1:1,0,3) (3) =INDEX(會員資料,3,2)
解說	(2) 因為參照只有一列，在此也可以代換為 =INDEX('會員資料檔'!1:1,3)。

## B. 卡號 (1) 1777、(2) 1775 在會員卡號的位置

公式	(1) =MATCH(1777,會員卡號,0)	(2) =MATCH(1775,會員卡號,1)
解說	<p>(1) 因為會員卡號是數字格式，所以輸入 =MATCH("1777",會員卡號,0) 會輸出 #N/A!。</p> <p>(2) 1775 沒有出現在會員卡號，且在會員資料檔中會員卡號按遞增排序，若在此將 <b>搜尋類型</b> 設定為 1，會輸出會員卡號中 <math>\leq</math> 1775 的最大值，也就是 1771 的所在位置。</p>	

## C. 「家庭月收入」出現在會員資料表頭的位置

公式	=MATCH("家庭月收入",會員資料表頭,0)
解說	MATCH 不只可以查詢 <b>搜尋準則</b> 出現在某一欄的第幾列，也可以查詢出現在某一列的第幾欄。

## D. 會員卡號 1777 的家庭月收入

公式	<p><b>方法一：已知輸出哪一欄，並使用 MATCH 找出要輸出的列數</b>  <math>=INDEX(\text{家庭月收入}, MATCH(1777, \text{會員卡號}, 0))</math></p> <p><b>方法二：使用兩個 MATCH 分別找出要輸出的列 / 欄</b>  <math>=INDEX(\text{會員資料}, MATCH(1777, \text{會員卡號}, 0), MATCH("家庭月收入", \text{會員資料表頭}, 0))</math></p>
解說	<p>將 INDEX、MATCH 合併使用能完成更進階的搜尋，在此介紹兩種方法：</p> <p><b>方法一：已知輸出哪一欄，並使用 MATCH 找出要輸出的列數</b>          已知要輸出 <b>家庭月收入</b> 的其中一列，使用 MATCH 找出 1777 出現在會員卡號的第幾列，輸出該列的 <b>家庭月收入</b>。</p> <p><b>方法二：使用兩個 MATCH 分別找出要輸出的列 / 欄</b>          要輸出會員資料的某一列、某一欄，但取決於要輸出的會員卡號及欄位，在此用兩個 MATCH 分別找 1777 出現在會員卡號的第幾列、"家庭月收入" 出現在 <b>會員資料表頭</b> 的第幾欄。</p>

由以上範例可以發現，INDEX、MATCH 特別適用於和儲存格互動，例如在某一個儲存格輸入要查詢的欄位及會員卡號，就能使用 INDEX+MATCH 輸出該會員想要取得的資訊，因此在第 7.8 節會介紹 INDEX+MATCH 在實務中最常運用到的方式。

### 7.3 查詢函式：-LOOKUP 影片 [Link \(上\)](#) / [Link \(下\)](#)

INDEX + MATCH 是實務上十分常見的功能，熟悉兩個可以完成大部分的查詢功能，而此節要介紹另一組常用的查詢函式 — LOOKUP，包括 VLOOKUP / HLOOKUP / LOOKUP 三個函式，學會這兩節所介紹的函式的用法後，可以自由視習慣選擇其中一種作法完成各種查詢：

- VLOOKUP / HLOOKUP：垂直 (Vertical) / 水平 (Horizontal) 查詢，找欲搜尋的資料出現在範圍第一欄 / 列的哪一列 / 欄，並輸出該列的某一欄 / 列：

語法	<a href="#">VLOOKUP / HLOOKUP (搜尋準則, 範圍, 索引, [已排序])</a>
輸入	<ul style="list-style-type: none"> <li>● <b>搜尋準則</b>：欲搜尋的資料，必須是單一值。</li> <li>● <b>範圍</b>：要在哪裡找，為多欄多列的陣列或範圍。</li> <li>● <b>索引</b>：要輸出 <b>範圍</b> 的第幾欄 / 列。</li> <li>● <b>已排序</b>：TRUE / FALSE，預設為 TRUE，意義如下：           <ul style="list-style-type: none"> <li>○ TRUE：用於 <b>搜尋準則</b> 為數字資料且 <b>範圍</b> 第一欄遞增排序時，搜尋的方式會同 MATCH 的搜尋類型 = 1。</li> <li>○ FALSE：完全比對，同 MATCH 的搜尋類型 = 0。</li> </ul> </li> </ul>
輸出	<ul style="list-style-type: none"> <li>● VLOOKUP：在 <b>範圍</b> 第一欄找 <b>搜尋準則</b> 所在列，輸出第 <b>索引</b> 欄。</li> <li>● HLOOKUP：在 <b>範圍</b> 第一列找 <b>搜尋準則</b> 所在欄，輸出第 <b>索引</b> 列。</li> </ul>

- LOOKUP：結合 VLOOKUP 與 HLOOKUP 「已排序」的搜尋用法

語法	<ul style="list-style-type: none"> <li>● 用法一：<a href="#">LOOKUP (搜尋準則, 搜尋範圍, 結果範圍)</a></li> <li>● 用法二：<a href="#">LOOKUP (搜尋準則, 搜尋與結果範圍)</a></li> </ul>
輸入	<ul style="list-style-type: none"> <li>● <b>搜尋準則</b>：欲搜尋的資料，必須是單一值。</li> <li>● <b>搜尋範圍</b>：要在哪裡找，為單欄或單列的陣列或範圍。</li> <li>● <b>結果範圍</b>：要輸出的範圍，大小與 <b>搜尋範圍</b> 相同。</li> <li>● <b>搜尋與結果範圍</b>：要在哪裡找、輸出，為多欄多列的陣列或範圍。</li> </ul>
輸出	<p>兩種用法的查詢方式同 MATCH 的搜尋類型 = 1，輸出內容分別如下：</p> <ul style="list-style-type: none"> <li>● 用法一：在 <b>搜尋範圍</b> 中查詢 <b>搜尋準則</b> 所在列 / 欄，輸出 <b>結果範圍</b> 的同一列 / 欄。</li> <li>● 用法二：在 <b>搜尋與結果範圍</b> 第一欄 / 列查詢 <b>搜尋準則</b> 所在列 / 欄，輸出 <b>搜尋與結果範圍</b> 最後一列 / 欄的同一欄 / 列。其中要搜尋第一列或第一欄取決於 <b>搜尋與結果範圍</b> 的列數與欄數，若欄數 &gt; 列數則搜尋第一列，反之搜尋第一欄。</li> </ul>

上述函式說明較為複雜，因此以下使用兩個範例說明之：

A. 下圖：其中已命名範圍「成績表 / 學生 / 國文成績 / 數學成績 / 英文成績」

學生	國文	數學	英文	範例	結果	公式
甲	80	75	90	(1)	80	=VLOOKUP("丁",成績表,3, FALSE)
乙	75	60	75	(2)	75	=HLOOKUP("英文",成績表,3, FALSE)
丙	70	70	70	(3)	80	=LOOKUP("丁",學生,數學成績)
丁	65	80	65	(4)	65	=LOOKUP("丁",成績表)
戊	60	55	80			
己	55	65	55			
庚	50	50	50			
辛	90	95	80			

公式	(1) =VLOOKUP("丁",成績表,3, FALSE) (2) =HLOOKUP("英文",成績表,3, FALSE) (3) =LOOKUP("丁",學生,數學成績) (4) =LOOKUP("丁",成績表)
解說	(1) 在 成績表 第一欄找到 "丁" 所在的列 (第 5 列)，輸出第 3 欄的第 5 列，即丁的數學成績 80。 (2) 在 成績表 第一欄找到 "英文" 所在的欄 (第 4 欄)，輸出第 4 欄的第 3 列，即乙的英文成績 75。 (3) 在 學生 找到 "丁" 所在的列 (第 5 列)，輸出 數學成績 第 5 列的值。 (4) 因為 成績表 的列數 $\geq$ 欄數，因此在 成績表 第一欄找到 "丁" 所在的列 (第 5 列)，輸出 成績表 最後一欄第 5 列的值，即丁的英文成績 65。

B. 下圖：其中已命名範圍「成績換算表 / 最小值 / 最大值 / GPA / 等第」

最小值	最大值	GPA	等第	範例	結果	公式
0	59	0	F	(1)	#N/A	=VLOOKUP(75,成績換算表,4, FALSE)
60	62	1.7	C-	(2)	B	=VLOOKUP(75,成績換算表,4, TRUE)
63	66	2	C	(3)	B	=LOOKUP(75,最小值,等第)
67	69	2.3	C+	(4)	B	=LOOKUP(75,成績換算表)
70	72	2.7	B-			
73	76	3	B			
77	79	3.3	B+			
80	84	3.7	A-			
85	89	4	A			
90	100	4.3	A+			

公式	(1) =VLOOKUP(75,成績換算表,4, FALSE) (2) =VLOOKUP(75,成績換算表,4, TRUE) (3) =LOOKUP(75,最小值,等第) (4) =LOOKUP(75,成績換算表)
----	--

解說	<p>(1) 在 成績換算表 第一欄找到 75 所在的列，但因為 已排序=FALSE 且最小值中找不到 75 所以輸出 #N/A!。</p> <p>(2) 因為 已排序=TRUE 且第一欄為遞增排序，所以會找到小於 75 的最大值所在的列 (即 73、第 7 列)，輸出第 7 列的第 4 欄，即 B。</p> <p>(3) 在 最小值 找到小於 75 的最大值所在的列 (第 7 列)，輸出 等第 第 7 列的值。</p> <p>(4) 因為 成績換算表 的列數 <math>\geq</math> 欄數，因此在 成績換算表 第一欄找到小於 75 的最大值所在的列 (第 7 列)，輸出 成績換算表 最後一欄第 7 列的值。</p>
----	---

了解這些函式後也會發現這三個函式都有一些限制，例如 VLOOKUP / HLOOKUP 的 **搜尋準則** 一定要出現在 **範圍** 的首欄 / 列、而 LOOKUP 在搜尋數字時只能針對遞增的資料進行精確的搜尋。

然而上述 VLOOKUP / HLOOKUP 的限制可以使用「陣列」解決，簡單來說陣列包括以下四個要素：

- 大括號 {}：創立陣列，之後會在大括號內填入陣列的內容。
- 半形逗號 ,：陣列中的分欄符號。
- 半形分號 ;：陣列中的分列符號。
- 欲組合的內容：可以是單一值、儲存格範圍或某公式的輸出等，其中內容的欄或列必須一致才能組合，例如可以使用半形逗號連接 A1:B3 和 D1:D3，因為其列數相同，但若使用分號則會跳出 #REF!，因為前者有兩欄、但後者只有一欄。

詳細陣列的用法及其相關公式將會在第 11 章說明，在此將延續範例 B. 示範如何在 VLOOKUP 中使用陣列，以解決上述欲搜尋項目並非在資料的非第一欄 / 列之限制：

### C. 延續範例 B.，輸出等第 A- 的 GPA

公式	<code>=VLOOKUP("A-", {等第, GPA}, 2, FALSE)</code>
解說	<ul style="list-style-type: none"> <li>• 使用大括號將 <b>等第</b> 和 <b>GPA</b> 合成一個兩欄的陣列，就能使用 VLOOKUP 找出 "A-" 在陣列首欄 (即 <b>等第</b>) 的第幾列 (第 9 列)，輸出第 2 欄 (即 <b>GPA</b>) 的第 9 列，即 3.7。</li> <li>• 當然在此可使用 INDEX+MATCH 完成，即 =INDEX(GPA, MATCH("A-", 等第, 0))。</li> </ul>

總而言之，VLOOKUP / HLOOKUP、LOOKUP、INDEX + MATCH 有各自適合的使用時機，根據作者的使用習慣歸納如下：

- VLOOKUP / HLOOKUP：要參照的欄位是「完全相符」的情況下。
- LOOKUP：要參照的欄位是「遞增的數字且不用完全相符」的情況下。
- INDEX + MATCH：要參照的欄位是「遞減的數字且不用完全相符」的情況下，因為 VLOOKUP / HLOOKUP / LOOKUP 皆無法達成。

然而上述函式都還是存在一些限制，例如無法決定搜尋順序 (都是預設左至右、上至下)、搜尋文字時無法使用萬用字元搜尋等，而下一節介紹的兩個函式 — XMATCH、XLOOKUP 將能完美解決這兩個限制。

## 7.4 查詢函式：XMATCH / XLOOKUP 影片 [Link](#)

XMATCH、XLOOKUP 基本上就是允許使用萬用字元、選擇搜尋順序的 MATCH、-LOOKUP，相較前兩節的函式更加靈活。

語法	<ul style="list-style-type: none"> <li><b>XMATCH</b> (<b>搜尋準則</b>, <b>搜尋範圍</b>, [<b>比對方式</b>], [<b>搜尋方式</b>])</li> <li><b>XLOOKUP</b> (<b>搜尋準則</b>, <b>搜尋範圍</b>, <b>結果範圍</b>, [<b>缺值</b>], [<b>比對方式</b>], [<b>搜尋方式</b>])</li> </ul>
輸入	<ul style="list-style-type: none"> <li><b>搜尋準則</b>：欲搜尋的資料，必須是單一值。</li> <li><b>搜尋範圍</b>：要在哪裡找，為單欄或單列的陣列或範圍。</li> <li><b>結果範圍</b>：要輸出的範圍，大小與 <b>搜尋範圍</b> 的列數 / 欄數相同。</li> <li><b>缺值</b>：找不到相符項目時要傳回的值，預設為 #N/A。</li> <li><b>比對方式</b>：2 / 1 / 0 / -1，尋找的方式，預設為 0，意義分別如下： <ul style="list-style-type: none"> <li>0：完全比對，同 MATCH 的「搜尋類型=0」</li> <li>1：完全比對，或尋找大於 <b>搜尋準則</b> 的最接近值。</li> <li>-1：完全比對，或尋找小於 <b>搜尋準則</b> 的最接近值。</li> <li>2：使用萬用字元的尋找模式，即第 6.6 節說明的萬用字元。</li> </ul> </li> <li><b>搜尋方式</b>：1 / -1 / 2 / -2，尋找的順序，預設為 1，意義分別如下： <ul style="list-style-type: none"> <li>1 / -1：從上往下找 / 從下往上找。</li> <li>2 / -2：用二分法來搜尋，<b>搜尋範圍</b> 分別需要遞增 / 遲減排序，用來節省搜尋時所需的運算效能，通常用於資料量較大時。</li> </ul> </li> </ul>
輸出	<ul style="list-style-type: none"> <li>XMATCH：<b>搜尋準則</b> 在 <b>搜尋範圍</b> 中的第幾列 / 欄。</li> <li>XLOOKUP：<b>搜尋準則</b> 在 <b>搜尋範圍</b> 中的第幾列 / 欄，輸出 <b>結果範圍</b> 的該列 / 欄，若 <b>結果範圍</b> 有多欄 / 列，則會輸出整欄 / 列。</li> </ul>

以下說明這兩個函式最主要的使用時機為何：

- 需要在未遞增 / 遲減的資料中比對**：XMATCH / XLOOKUP 的比對方式不需要像 MATCH / -LOOKUP 一樣遞增或遞減排序，因此可以使用於數字但未排序的比對。
- 需要使用萬用字元搜尋**：例如無視商品規格的銷售記錄，例如商品名稱為「iPhone 15 - (顏色)」想要搜尋第一筆 iPhone 15 的銷售日期，可以使用 =XLOOKUP("iPhone 15 - (\*)",商品名稱,銷售日期,,2) 完成。
- 需要由下往上搜尋**：延續上一點的例子，若要搜尋最近一筆的日期可將 **搜尋方式** 設為 -1 即可。

以上的說明相信應該相當清楚，在此就不用範例說明 XMATCH / XLOOKUP 的用法，若有興趣練習的同學不仿嘗試將第 7.2~7.3 節的練習改成使用 XMATCH / XLOOKUP 完成，確認是否已真的理解函式的用法。

## 7.5 查詢函式：OFFSET 影片 [Link](#)

OFFSET 的查詢與前幾節不同，其輸出會是某個儲存格或某個範圍，其查詢的方式是以某個儲存格為基準點，向上下左右移動得到要查詢的範圍，說明如下：

語法	<b>OFFSET (參照, 偏移列數, 偏移欄數, [高度], [寬度])</b>
輸入	<ul style="list-style-type: none"> <li>● <b>參照</b>：以哪個儲存格作為基準點。</li> <li>● <b>偏移列數 / 偏移欄數</b>：往下 / 右移動幾列 / 欄，負數則為往上 / 左移動。</li> <li>● <b>高度 / 寬度</b>：輸出的範圍共有幾列 / 欄，預設為 1。</li> </ul>
輸出	<p>指定範圍的輸出，如下圖的黑框範圍所示。若最終的範圍包括負數的欄或列會輸出 #REF!。</p>

以下先使用一些簡單的範例說明之：

### A. 判斷以下公式會輸出的儲存格範圍

公式	(1) =OFFSET(A1,2,1) (2) =OFFSET(A1,-1,0) (3) =OFFSET(H8,-4,-3) (4) =OFFSET(A1,2,1,3,2) (5) =OFFSET(H8,-4,-3,3,2)
解說	(1) A1 往下移 2 列、往右移 1 欄，會輸出 B3 的值。 (2) A1 往上移 1 列，但已超出工作表範圍，會輸出 #REF!。 (3) H8 往上移 4 列、往左移 3 欄，會輸出 E4 的值。 (4) A1 往下移 2 列、往右移 1 欄後，輸出 3 列及 2 欄的內容，會輸出 B3:C5 的值。 (5) H8 往上移 4 列、往左移 3 欄後，輸出 3 列及 2 欄的內容，會輸出 E4:F6 的值。

了解 OFFSET 的運作方式後，接下來說明 OFFSET 經常使用的時機：

- 搭配 SUM / COUNTIF / SUMIF 等函式，加總 / 計算特定範圍
- 搭配 MATCH，用於欄 / 列轉置的範圍查詢

以下使用一個進階的範例說明上述兩者的用法：

### B. 計算丁公司二月到四月的金額加總

	A	B	C	D	E	F	G
1	公司	一月	二月	三月	四月	五月	六月
2	甲	0	10,143	20,224	0	0	31,255
3	乙	744	10,640	19,782	17,911	0	14,269
4	丙	0	0	18,542	25,064	26,765	351
5	丁	0	0	22,462	50,638	10,160	1,440
6	戊	31,764	0	0	12,417	13,061	4,784
7	己	36,827	11,655	1,876	54,331	53,849	9,815

公式	=SUM(OFFSET(A1,MATCH("丁",A:A,0)-1,MATCH("二月",A1:G1,0)-1,1,MATCH("四月",A1:G1,0)-MATCH("二月",A1:G1,0)+1))
解說	<p>在此可以想像成要製作一個搜尋器，讓使用者輸入公司 ("丁")、起始月份 ("二月")、結束月份 ("四月")，因此接下來會使用 SUM + OFFSET + MATCH 完成這個需求，因為範例較為複雜，以下分成幾個步驟說明：</p> <ul style="list-style-type: none"> <li>丁公司二月到四月的總金額：<math>=\text{SUM}(\text{OFFSET}(\text{A1},4,2,1,3))</math> 可以使用 OFFSET 決定要加總的範圍，在此以 A1 為參照，已知丁在第五列所以下移 4 列，二月到四月可以用 OFFSET 以 A1 為起始點右移 2 欄、選取 3 欄得到。</li> <li>丁公司：<math>4 \rightarrow \text{MATCH}("丁",A:A,0)-1</math> 偏移列數可以使用 MATCH 取代，尋找 "丁" 出現在第幾列，但偏移列數要 -1 因為參照的 A1 已經是第一列，也可以直接用 <math>\text{MATCH}("丁",A2:A,0)</math> 代替之。</li> <li>二月：<math>2 \rightarrow \text{MATCH}("二月",A1:G1,0)-1</math> 偏移欄數也可以使用 MATCH 取代，尋找 "二月" 出現在第幾列，但偏移欄數要 -1 因為參照的 A1 已經是第一列，也可以用 <math>\text{MATCH}("二月",B1:G1,0)</math> 代替之。</li> <li>二月到四月：<math>3 \rightarrow \text{MATCH}("四月",A1:G1,0)-\text{MATCH}("二月",A1:G1,0)+1</math> 決定寬度可使用兩個 MATCH 分別計算 "四月"、"二月" 所在的欄數，兩者相減後要 +1 因為二月、四月都要算入，和 <math>3/10-3/21=12</math> 天一樣的道理。</li> </ul>

依據上述步驟完成後，能進一步使用資料驗證搭配此公式使用，製作成搜尋器，讓公式更靈活的運用，將會在第 7.8 節示範到。

## 7.6 篩選函式：UNIQUE / SORT / SORTN 影片 [Link \(上\)](#) / [Link \(下\)](#)

若想要對資料進行移除重複資料或排序資料時，可以使用第 3.4 節之方式完成，但如果不想動到原始資料，而是另一個地方進行移除重複資料或排序時，可以使用 UNIQUE / SORT / SORTN 這三個函式完成，函式說明如下：

- UNIQUE：取得不重複的列 / 欄：

語法	<b>UNIQUE</b> ( <b>範圍</b> , [ <b>根據欄</b> ], [ <b>只出現一次</b> ])
輸入	<ul style="list-style-type: none"> <li>● <b>範圍</b>：要篩選的資料範圍。</li> <li>● <b>根據欄</b>：TRUE / FALSE，是否以欄做為篩選依據，預設為 FALSE (即用列篩選)。</li> <li>● <b>只出現一次</b>：TRUE / FALSE，是否只傳回沒有重複項目的值，預設為 FALSE。例如有一筆資料出現兩次以上，則輸出時不會輸出此筆資料。</li> </ul>
輸出	<b>範圍</b> 中的不重複資料列 / 欄，會根據第一次出現的順序排序。

- SORT / SORTN：排序所選範圍：

語法	<ul style="list-style-type: none"> <li>● <b>SORT</b> (<b>範圍</b>, [<b>排序欄_1</b>, <b>遞增_1</b>], [<b>排序欄_2</b>, <b>遞增_2</b>,...])</li> <li>● <b>SORTN</b> (<b>範圍</b>, [<b>n</b>], [<b>顯示所有和局</b>], [<b>排序欄_1</b>, <b>遞增_1</b>], [<b>排序欄_2</b>, <b>遞增_2</b>,...])</li> </ul>
輸入	<ul style="list-style-type: none"> <li>● <b>範圍</b>：要排序的資料範圍。</li> <li>● <b>n</b>：只輸出前 <b>n</b> 個值，必須大於 0，預設為 1。</li> <li>● <b>顯示所有和局</b>：遇到一樣順位時的輸出方式，有 0 / 1 / 2 / 3 四種，預設為 0，意義分別如下： <ul style="list-style-type: none"> <li>○ 0：顯示前 <b>n</b> 列。</li> <li>○ 1：如果有與第 <b>n</b> 列重複的值，則一起顯示。</li> <li>○ 2：先將 <b>範圍</b> 有重複的值刪除，再顯示前 <b>n</b> 列。</li> <li>○ 3：篩選前 <b>n</b> 個不重複的 <b>範圍</b>，再顯示其所有重複的值。</li> </ul> </li> <li>● <b>排序欄_n</b>：要根據哪個欄位進行排序，最左邊是 1，以此類推。若有多個排序條件，須與 <b>遞增_n</b> 成對</li> <li>● <b>遞增_n</b>：TRUE / FALSE，分別為遞增、遞減排序。</li> </ul>
輸出	<b>範圍</b> 根據 <b>排序欄_n</b> 進行 <b>遞增_n</b> 排序的結果， <b>SORTN</b> 則顯示前 <b>n</b> 個值。

以下使用第 1 章的會員資料檔並搭配五個範例說明各函式的用法：

### A. 取得不重複的婚姻狀態

公式	=UNIQUE( <b>婚姻狀態</b> )
解說	會輸出婚姻狀態、已婚、未婚三種，若將 <b>只出現一次</b> 設為 TRUE，即 =UNIQUE( <b>婚姻狀態</b> , <b>TRUE</b> )，則會刪除所有有重複出現過的資料，只會出現表頭 (婚姻狀態) 一列。

## B. 取得不重複的婚姻狀態與性別

公式	=UNIQUE({ <b>婚姻狀態,性別</b> })
解說	<ul style="list-style-type: none"> <li>在此使用陣列自由連接想要輸出的欄位，前提是各欄位列數必須相同。</li> <li>若 UNIQUE 要篩選的資料範圍有多欄，會篩選所有資料都不重複的列，因此前者會輸出表頭、2 種婚姻狀態 × 2 種性別五列。若將 <b>只出現一次</b> 設為 TRUE，即 =UNIQUE({<b>婚姻狀態,性別</b>},,TRUE) 則只會輸出表頭。</li> </ul>

## C. 取得所有不重複的家庭月收入選項，並將其遞減排序

公式	<ul style="list-style-type: none"> <li>=UNIQUE(SORT(家庭月收入,家庭月收入,FALSE))</li> <li>=SORT(UNIQUE(家庭月收入),UNIQUE(家庭月收入),FALSE)</li> </ul>
解說	<p>篩選函式之間可以一起使用，在此提供兩種做法：</p> <ul style="list-style-type: none"> <li>先將所有 <b>家庭月收入</b> 排序，再使用 UNIQUE 移除重複項目。</li> <li>將先篩選出不重複的 <b>家庭月收入</b>，再根據字串進行排序，但此方式需要使用到兩次 UNIQUE(<b>家庭月收入</b>)。</li> </ul>

## D. 將資料以生日遞減排序，並顯示前 5 筆資料的所有欄位

公式	=SORTN(會員資料,5,,2,FALSE)
解說	<ul style="list-style-type: none"> <li>在此已知生日在第 2 欄，可使用 MATCH 尋找生日在表頭的位置替換。</li> <li>TRUE / FALSE 可代換成 1 / 0。</li> </ul>

## E. 顯示前三筆會員編號、婚姻、性別，其排序依據為婚姻、性別，皆為遞增排序

公式	=SORTN({ <b>會員卡號,婚姻狀態,性別</b> },3,X,2,0,3,0)
解說	<p>其中 X (<b>顯示所有和局</b>) 選擇 0~3 會有不一樣的結果，如完成檔案所示，說明分別如下：</p> <ul style="list-style-type: none"> <li>X = 0：直接選擇前三筆資料。</li> <li>X = 1：原本的第三筆資料為未婚 + 男，因此會輸出所有未婚 + 男的資料。</li> <li>X = 2：會篩選不重複的婚姻狀態 + 性別，會篩選未婚 + 男、未婚 + 女、已婚 + 男的資料，並輸出這三種類別的第一筆資料。</li> <li>X = 3：會輸出未婚 + 男、未婚 + 女、已婚 + 男的所有資料。</li> </ul>

本節的函式非常適合用於儲存格有多列且有可能重複的輸出時，而下一節將介紹篩選器函式 FILTER 的用法，其可以在不動到原始資料的情況下完成想要的篩選與排序，更能完成一些使用 SUMIFS 等函式無法達成的計算。

## 7.7 篩選函式：FILTER 影片 [Link](#)

FILTER 用於想要在另一個工作表或範圍篩選資料時，其使用方法如下：

語法	<b>FILTER(範圍, 條件_1, [條件_2,...])</b>
輸入	<ul style="list-style-type: none"> <li>● <b>範圍</b>：最後要輸出的範圍，可以是一欄或多欄。</li> <li>● <b>條件_n</b>：要篩選的條件，列數需與 <b>範圍</b> 相符，不一定要跟 <b>範圍</b> 的資料有關，但每個項目都要是邏輯運算式 (即輸出為 TRUE / FALSE)。</li> </ul>
輸出	符合所有 <b>條件_n</b> 的 <b>範圍</b> ，若都不符合則會輸出 #N/A。

以下使用第 1 章的會員資料檔並搭配三個範例由淺入深說明 FILTER 的用法：

### A. 輸出所有子女人數 2 位以上的會員卡號

公式	=FILTER(會員卡號, 子女人數>=2)
解說	<p>在此會發現輸出中包含表頭「會員卡號」，是因為在試算表中將字串與數字比大小時，會認為字串比較大。</p> <p>補充：若要排除表頭，以下提供兩種方式，大家也可以參考完成檔案中有其他三種方式，並自行思考有哪些函式 / 方式可以做到，增加自己對 Google Sheets 的熟悉程度：</p> <ul style="list-style-type: none"> <li>● 直接使用儲存格範圍：即 =FILTER('會員資料檔'!A2:A,'會員資料檔'!H2:H&gt;=2)。</li> <li>● 增加子女人數的上限值：在此可增加一個條件 子女人數&lt;=MAX(子女人數)，在篩選時就不會包含字串。</li> </ul>

### B. 輸出所有家庭月收入 9 萬以上且子女人數 2 位以上的會員卡號

公式	=FILTER(會員卡號, 家庭月收入="9萬以上", 子女人數>=2)
解說	若有兩個以上的條件，各條件之間使用逗號分隔即可。

### C. 輸出所有家庭月收入 9 萬以上或子女人數 2 位以上的會員卡號

公式	=FILTER(會員卡號, (家庭月收入="9萬以上")+(子女人數>=2)>=1)
解說	若要表示多個條件之間是「或」的關係，可以使用第 6.1 節介紹到的將 TRUE / FALSE 轉換成 1 / 0 的概念。在此將兩個條件使用括號及加號連接，並判斷是否 >=1，即至少有一個是 TRUE。

FILTER 在 Google Sheets 的使用時機也非常廣，除了可以搭配上一節的其他篩選函式一起使用外，也能搭配第 6.6、7.2~7.4 節的值或範圍使用。

學習完第 6~7 章的內容後，已經足以解決 80% 以上工作上的問題，但仍有其他重要的函式如日期、文字相關的處理，以及如何將公式寫得更簡潔易懂，將會是後續幾章強調的重點。

## 7.8 進階應用 影片 [Link \(上\)](#) / [Link \(下\)](#)

本節將會使用兩個範例示範第 7 章各個函式的用法，第一個範例將建立一個「會員資訊查詢」表，只要輸入會員編號及輸出欄位，就能輸出該會員該項欄位的資訊，將示範完成此搜尋的七種做法，各位可以先自行練習，確認自己是否已經確實了解本章各項函式的運用，完成結果如下圖。

會員資訊查詢	
會員卡號	1852 ▾
輸出欄位	家庭人口數 ▾
搜尋結果	7人以上
	方法一：INDEX+MATCH
	7人以上
	方法二：VLOOKUP+MATCH
	7人以上
	方法三：HLOOKUP+MATCH
	7人以上
	方法四：VLOOKUP+INDIRECT
符合條件會員數	7人以上
	方法五：XLOOKUP+INDIRECT
符合條件會員數	7人以上
	方法六：OFFSET+MATCH
符合條件會員數	7人以上
	方法七：FILTER+INDIRECT

第二個範例則是建立「活動適用會員清單」，只要輸入活動適用的條件 (家庭月收入 / 性別 / 生日區間)，即可篩選出所有符合上述條件的會員，並計算其數量，此外這個例子也會介紹到如何在公式中考慮到「非必填」的條件。

活動適用條件 (非必填)								
家庭月收入	9萬以上 ▾							
性別	女 ▾							
生日晚於...								
生日早於...								
符合條件會員數	7							
	方法一：COUNTIFS							
	方法二：COUNTA+FILTER							
符合條件會員數	7							
	方法三：ROWS+FILTER							
活動適用會員清單								
會員卡號	生日	性別	家庭人口數	職業	學歷	婚姻狀態	子女人數	家庭月收入
9762	1962/1/1	女	3~4人	家庭主婦	專科	已婚	0	9萬以上
3706	1965/2/28	女	3~4人	學生	大學	已婚	2	9萬以上
3753	1973/7/23	女	3~4人	商	大學	已婚	2	9萬以上
7724	1975/5/1	女	3~4人	軍公教	專科	已婚	2	9萬以上
8159	1980/12/24	女	7人以上	其它	高中職	已婚	2	9萬以上
2181	1981/7/5	女	3~4人	商	大學	已婚	1	9萬以上
1777	1988/7/23	女	3~4人	家庭主婦	大學	已婚	2	9萬以上

但因為整個過程較為複雜，在此不一一說明各步驟的完成方式，而是在教學影片中示範資料驗證的設定及各公式的思考脈絡，非常推薦各位同學有空的時候跟著影片一起思考並操作，可以從中學到非常多的東西。

---

本章完成檔案：[+](#) Chapter 7. 函式 (二)：查詢與篩選 完成版

## Appendix 7. Excel 函式 (二)：查詢與篩選

### 7.3 查詢函式：-LOOKUP

Excel 中若將陣列作為函式的輸入時，公式輸入完畢不能直接按 Enter，必須按 Option / Ctrl + Shift + Enter 才能正確輸出。

### 7.6 篩選函式：UNIQUE / SORT / SORTBY

- SORT：語法與 Google Sheets 不同，如下：

語法	<b>SORT</b> ( <b>範圍</b> , <b>排序</b> , <b>遞增遞減</b> , <b>根據欄</b> )
輸入	<ul style="list-style-type: none"> <li>● <b>範圍</b>：要排序的資料範圍。</li> <li>● <b>排序</b>：要根據哪個欄位進行排序，最左邊是 1，以此類推。只能有一個排序依據。</li> <li>● <b>遞增遞減</b>：1 / -1，分別代表遞增 / 遞減。</li> <li>● <b>根據欄</b>：TRUE / FALSE，是否以欄做為排序依據，預設為 FALSE (即用列篩選)。</li> </ul>
輸出	<b>範圍</b> 以 <b>根據欄</b> 作為排序依據根據 <b>排序</b> 進行 <b>遞增遞減</b> 排序的結果。

- SORTN：無此函式。
- SORTBY：在 Google Sheets 無此函式，其功能與 SORT 類似，如下：

語法	<b>SORTBY</b> ( <b>範圍</b> , [ <b>排序範圍_1</b> , <b>遞增遞減_1</b> ], [ <b>排序範圍_2</b> , <b>遞增遞減_2,...</b> ])
輸入	<ul style="list-style-type: none"> <li>● <b>範圍</b>：要排序的資料範圍。</li> <li>● <b>排序範圍_n</b>：要根據哪個範圍的值進行排序，要輸入範圍而非數值。</li> <li>● <b>遞增遞減_n</b>：1 / -1，分別代表遞增 / 遞減，須與 <b>排序範圍_n</b> 成對</li> </ul>
輸出	<b>範圍</b> 根據 <b>排序範圍_n</b> 進行 <b>遞增遞減_n</b> 依據排序的結果。

### 7.7 篩選函式：FILTER

FILTER 的語法與 Google Sheets 不同，如下：

語法	<b>FILTER</b> ( <b>範圍</b> , <b>條件</b> , [ <b>缺值</b> ])
輸入	<ul style="list-style-type: none"> <li>● <b>範圍</b>：最後要輸出的範圍，可以是一欄或多欄。</li> <li>● <b>條件</b>：要篩選的條件，列數需與 <b>範圍</b> 相符，不一定要跟 <b>範圍</b> 的資料有關，但每個項目都要是邏輯運算式 (即輸出為 TRUE / FALSE)。</li> <li>● <b>缺值</b>：如果篩選後沒有資料符合 <b>條件</b> 時的輸出。</li> </ul>
輸出	符合 <b>條件</b> 的 <b>範圍</b> ，若都不符合則會輸出 <b>缺值</b> 。

若要在 Excel 的 FILTER 中使用多個條件篩選，可在 **條件** 中使用 (條件\_1)\*(條件\_2) 表示，例如第 7.7 節的範例 B 可以更改成 =FILTER(會員卡號,(家庭月收入="9萬以上")\*(子女人數>=2))。

## Chapter 8. 函式 (三)：日期與時間 影片 [Link](#)

本章示範檔案： [Chapter 8. 函式 \(三\)：日期與時間](#)

本章將介紹與日期、時間相關的函式，相較於前面兩章有較多複雜的函式，本章的函式相對直觀不少。學完此章節的內容後，就可以靈活在公式中運算日期，在處理會計分錄、每日銷售等資料都非常實用。

### 8.1 建立日期與時間 影片 [Link](#)

在 Google Sheets 中，若想要建立日期與時間，直接在儲存格中輸入格式直觀的日期與時間，Google Sheets 就會自動將其轉為日期。此外，還有多個函式可以取得或建立日期與時間，如下：

- TODAY / NOW：建立現在日期與時間：

語法	<a href="#">TODAY / NOW ()</a>
輸出	現在的日期 / 日期與時間，但輸出會隨時間變動，進而影響計算效能。

- DATE：使用數字建立日期：

語法	<a href="#">DATE (年, 月, 日)</a>
輸入	<b>年 / 月 / 日</b> ：日期的各項組成部分，相關規則如下： <ul style="list-style-type: none"><li>• 若輸入為小數，將會無條件捨去，若無輸入內容會輸出 #VALUE!。</li><li>• <b>年</b> 必須在 0~10,000 之間，否則會輸出 #NUM!；若輸入 0~1899 會自動加上 1900，例如輸入 125 會自動轉為 2025 年。</li><li>• <b>月 / 日</b> 會自動進位，例如 2025/1/32 會進位為 2025/2/1、2024/12/32 會進位為 2025/1/1。</li></ul>
輸出	根據輸入項目輸出日期。

- TIME：使用數字建立時間：

語法	<a href="#">TIME (小時, 分鐘, 秒)</a>
輸入	<b>小時 / 分鐘 / 秒</b> ：時間的各項組成部分，相關規則如下： <ul style="list-style-type: none"><li>• 若輸入為小數，將會無條件捨去，若無輸入內容會輸出 #VALUE!。</li><li>• <b>分鐘 / 秒</b> 會自動進位，例如 3:00:60 會進位為 3:01:00、3:70:30 會進位為 4:10:30。</li></ul>
輸出	根據輸入項目輸出時間。

- DATEVALUE / TIMEVALUE：使用文字建立日期與時間：

語法	<ul style="list-style-type: none"> <li>• <a href="#">DATEVALUE (日期字串)</a></li> <li>• <a href="#">TIMEVALUE (時間字串)</a></li> </ul>
輸入	<b>日期字串 / 時間字串</b> ：以 Google Sheets 可辨認的字串形式表達的日期與時間，必須輸入文字，否則會輸出 #VALUE!。可辨認的形式例如 1/23/2012、2012-01-23、1/23/2012 8:10:30、2:15 PM、14:15 等，其他形式詳見官方文件說明。
輸出	根據輸入項目輸出日期 / 日期與時間。

上述的函式用法都非常直觀，範例如下圖。在第 8.4 節將進一步介紹上述函式如何搭配前兩章的函式使用。

範例	結果	公式
A	2024/4/19	=TODAY()
	2024/4/19 上午 12:40:42	=NOW()
B	2025/1/2	=DATE(2025,1,2)
	2025/1/2	=DATE(125,1,2)
	2025/2/1	=DATE(2025,1,32)
	2025/1/2	=DATE(2024,13,2)
	2025/1/2	=DATE(2024,12,33)
C	上午 1:02:03	=TIME(1,2,3)
	上午 1:02:03	=TIME(1,1,63)
	上午 1:02:03	=TIME(0,62,3)
	上午 1:02:03	=TIME(25,2,3)
D	2025/01/02	=DATEVALUE("2025/1/2")
	2025/01/02	=DATEVALUE("2025-01-02")
	2025/01/02	=DATEVALUE("2/1/2025")
	上午 1:02:00	=TIMEVALUE("1:02")
	下午 1:02:00	=TIMEVALUE("1:02PM")
	上午 1:02:03	=TIMEVALUE("1:2:3")

## 8.2 取得日期與時間

影片 [Link](#)

除了建立日期與時間外，也可以從日期與時間中取得日期與時間的某個部分，如下：

- 取得年 / 月 / 日 / 小時 / 分鐘 / 秒：

語法	<ul style="list-style-type: none"><li>• <u>YEAR</u> / <u>MONTH</u> / <u>DAY</u> (<b>日期</b>)</li><li>• <u>HOUR</u> / <u>MINUTE</u> / <u>SECOND</u> (<b>時間</b>)</li></ul>
輸入	<b>日期</b> / <b>時間</b> ：要取得元素的日期 / 時間，可以是字串形式。
輸出	<b>日期</b> / <b>時間</b> 中的年 / 月 / 日 / 小時 / 分鐘 / 秒。

- 取得星期幾：

語法	<b><u>WEEKDAY</u> (<b>日期</b>, [類型])</b>
輸入	<ul style="list-style-type: none"><li>• <b>日期</b>：要取得星期幾的日期，可以是字串形式。</li><li>• <b>類型</b>：1 / 2 / 3，預設為 1，意義分別如下：<ul style="list-style-type: none"><li>◦ 1：星期日~星期六分別輸出 1~7。</li><li>◦ 2：星期一~星期日分別輸出 1~7。</li><li>◦ 3：星期一~星期日分別輸出 0~6。</li></ul></li></ul>
輸出	<b>日期</b> 的星期在 <b>類型</b> 的規則下的輸出。

- 取得第幾週：

語法	<b><u>WEEKNUM</u> (<b>日期</b>, [類型])</b>
輸入	<ul style="list-style-type: none"><li>• <b>日期</b>：要取得第幾週的日期，可以是字串形式。</li><li>• <b>類型</b>：一週的起始 / 結束日為何，預設為 1，其定義分別如下：<ul style="list-style-type: none"><li>◦ 1 / 2：一週的起始為星期日 / 星期一。</li><li>◦ 11~17：一週的起始為星期一 ~ 星期日。</li></ul></li></ul>
輸出	<b>日期</b> 是該年的第幾週，其中將 1/1 所在的週定義為該年第一週。

上述的函式用法都非常直觀，範例如下圖。在第 8.4 節將進一步介紹上述函式如何搭配前兩章的函式使用。

範例	結果	公式	說明
A	2025	=YEAR("2025/1/25")	
	1	=MONTH("2025/1/25")	
	25	=DAY("2025/1/25")	
	13	=HOUR("13:14:15")	
	14	=MINUTE("13:14:15")	
	15	=SECOND("13:14:15")	
B	7	=WEEKDAY("2025/1/25")	2025/1/25 是星期六
	6	=WEEKDAY("2025/1/25",2)	
	5	=WEEKDAY("2025/1/25",3)	
C	4	=WEEKNUM("2025/1/25")	- 2025/1/1 是星期三
	4	=WEEKNUM("2025/1/25",2)	- 2025/1/25 是 2025 年第四個星期六
	4	=WEEKNUM("2025/1/25",13)	→ 當星期的起始日為星期四 / 五 / 六時 ，會輸出 5，其他會輸出 4
	5	=WEEKNUM("2025/1/25",15)	

## 8.3 日期與時間運算

影片 [Link \(上\)](#) / [Link \(下\)](#)

在第 2.5 節有提到過，日期與時間是「數字」的一種，是以 1899/12/30 為 0，之後每一天多加 1，既然日期與時間可以轉為數字，那肯定也可以進行加減，例如在日期運算中，`DATE(2024,1,1)+1` 就會輸出 2024/1/2，以此類推。如下圖所示：

結果	結果 (數字格式)	公式
2024/1/2	45,293.00	=DATE(2024,1,1)+1
下午 8:00:00	0.83	=TIME(8,0,0)+0.5
2024/1/1 上午 8:00:00	45,292.33	=DATE(2024,1,1)+TIME(8,0,0)

瞭解日期運算的基本概念後，接下來將介紹多個計算日期時常用的函式：

- EDATE / EOMONTH：輸出特定日期前後 n 個月的日期 / 最後一天：

語法	<a href="#">EDATE / EOMONTH (開始日期, [個月])</a>
輸入	<ul style="list-style-type: none"> <li>● <b>開始日期</b>：特定日期，必須是日期或可轉換為日期的文字格式，若有時間部分將會被無條件捨去。</li> <li>● <b>個月</b>：要前 / 後幾個月，正數代表往後算、負數代表往前算，若並非整數會以無條件捨去計算。</li> </ul>
輸出	<ul style="list-style-type: none"> <li>● EDATE：<b>開始日期</b> 前 / 後 <b>個月</b> 個月的日期。</li> <li>● EOMONTH：<b>開始日期</b> 前 / 後 <b>個月</b> 個月的最後一天。</li> </ul>

- DATEDIF / DAYS：計算兩個日期的間隔單位數 / 天數：

語法	<ul style="list-style-type: none"> <li>● <a href="#">DATEDIF (開始日期, 結束日期, 單位)</a></li> <li>● <a href="#">DAYS (結束日期, 開始日期)</a></li> </ul>												
輸入	<ul style="list-style-type: none"> <li>● <b>開始日期 / 結束日期</b>：開始 / 結束日期，必須是日期或可轉換為日期的文字格式，若有時間部分將會被無條件捨去。</li> <li>● <b>單位</b>：計算間隔的單位，共有以下六種：</li> </ul> <table border="1"> <tbody> <tr> <td>"Y"</td> <td>"M"</td> <td>"D"</td> <td>"YM"</td> <td>"MD"</td> <td>"YD"</td> </tr> <tr> <td>年</td> <td>月</td> <td>日</td> <td>去掉年之後的月</td> <td>去掉月之後的日</td> <td>去掉年之後的日</td> </tr> </tbody> </table>	"Y"	"M"	"D"	"YM"	"MD"	"YD"	年	月	日	去掉年之後的月	去掉月之後的日	去掉年之後的日
"Y"	"M"	"D"	"YM"	"MD"	"YD"								
年	月	日	去掉年之後的月	去掉月之後的日	去掉年之後的日								
輸出	<ul style="list-style-type: none"> <li>● DATEDIF：<b>開始日期</b> 與 <b>結束日期</b> 的間隔 <b>單位</b> 數。</li> <li>● DAYS：<b>開始日期</b> 與 <b>結束日期</b> 的間隔天數。</li> </ul> <p>由上述可知，DAYS 可用 DATEDIF (<b>開始日期</b>, <b>結束日期</b>, "D") 取代。</p>												

- WORKDAY / WORKDAY.INTL：計算某日後的幾個工作天是幾月幾號：

語法	<ul style="list-style-type: none"> <li>• <b><u>WORKDAY</u></b> (<b>開始日期</b>, <b>天數</b>, [<b>假日</b>])</li> <li>• <b><u>WORKDAY.INTL</u></b> (<b>開始日期</b>, <b>天數</b>, [<b>週末</b>], [<b>假日</b>])</li> </ul>
輸入	<ul style="list-style-type: none"> <li>• <b>開始日期</b>：特定日期，必須是日期或可轉換為日期的文字格式，若有時間部分將會被無條件捨去。</li> <li>• <b>天數</b>：幾個工作天後，負數則為幾個工作天前，無條件捨去至整數位。</li> <li>• <b>假日</b>：特殊假日清單，可以是陣列或範圍，必須是日期格式，不能是可轉換為日期的文字格式，預設沒有特殊假日。</li> <li>• <b>週末</b>：哪幾天是非工作日，預設為 1，有兩種表達形式分別如下： <ul style="list-style-type: none"> <li>◦ 字串形式：會是由七個 0 / 1 組成的字串，0 代表工作日、1 代表非工作日，從禮拜一開始算，例如 "1100001" 代表非工作日為週一、週二、週日。</li> <li>◦ 數字形式：1~7 或 11~17。 <ul style="list-style-type: none"> <li>▪ 1~7：1 代表週六 + 週日、2 代表週日 + 週一，以此類推。</li> <li>▪ 11~17：11 代表週日、12 代表週一，以此類推。</li> </ul> </li> </ul> </li> </ul>
輸出	<p><b>開始日期</b> 後的 <b>天數</b> 個工作天的日期，在 WORKDAY 中假設 <b>週末</b> 為週六 + 週日，而 WORKDAY.INTL 可視為能自訂 <b>週末</b> 的 WORKDAY。</p>

- NETWORKDAYS / NETWORKDAYS.INTL：計算某段日期區間有幾個工作天：

語法	<ul style="list-style-type: none"> <li>• <b><u>NETWORKDAYS</u></b> (<b>開始日期</b>, <b>結束日期</b>, [<b>假日</b>])</li> <li>• <b><u>NETWORKDAYS.INTL</u></b> (<b>開始日期</b>, <b>結束日期</b>, [<b>週末</b>], [<b>假日</b>])</li> </ul>
輸入	<ul style="list-style-type: none"> <li>• <b>開始日期 / 結束日期</b>：開始 / 結束日期，必須是日期或可轉換為日期的文字格式，若有時間部分將會被無條件捨去。</li> <li>• <b>假日</b>：特殊假日清單，預設沒有特殊假日，表達形式同 WORKDAY。</li> <li>• <b>週末</b>：哪幾天是非工作日，預設為 1，表達形式同 WORKDAY。</li> </ul>
輸出	<p><b>開始日期</b> 後的 <b>天數</b> 個工作天的日期，在 WORKDAY 中假設 <b>週末</b> 為週六 + 週日，而 WORKDAY.INTL 可視為能自訂 <b>週末</b> 的 WORKDAY。</p> <p><b>開始日期</b> 與 <b>結束日期</b> 之間有幾個工作天 (含頭尾)，在 NETWORKDAYS 中假設 <b>週末</b> 為週六 + 週日，而 NETWORKDAYS.INTL 可視為能自訂 <b>週末</b> 的 NETWORKDAYS。</p>

了解以下公式的用法後，接下來使用七個範例說明：

#### A. 取得今天往回推 3 個月的日期

公式	=EDATE(TODAY(), -3)
解說	<b>開始日期</b> 可使用其他公式的輸出，在此使用 TODAY() 取得今天日期。

## B. 取得下個月底的日期

公式	=EOMONTH(TODAY(),1)
解說	開始日期 可使用其他公式的輸出，在此使用 TODAY() 取得今天日期。

## C. 取得今年年初日期

公式	<ul style="list-style-type: none"> <li>=DATE(YEAR(TODAY()),1,1)</li> <li>=EOMONTH(TODAY(),-MONTH(TODAY()))+1</li> </ul>
解說	<ul style="list-style-type: none"> <li>可直接使用第 7.1~7.2 節介紹的函式，先取得今年年份，再使用 DATE 輸出今年的 1/1 即可。</li> <li>可使用 EOMONTH 取得去年底的日期，再加上 1 天得到今年年初的日期。其中去年底的日期即 EOMONTH(今天, -今天所在月份)。</li> </ul>

## D. 計算今年迄今已過天數 (不含今日)

公式	<ul style="list-style-type: none"> <li>=DAYS(TODAY(),DATE(YEAR(TODAY()),1,1))</li> <li>=DATEDIF(DATE(YEAR(TODAY()),1,1),TODAY(),"D")</li> <li>=TODAY()-DATE(YEAR(TODAY()),1,1)</li> </ul>
解說	上述三種方式都是計算今天、今年 1/1 的差距，其中今年 1/1 使用 C. 的第一種方式。使用時可依據自己平常的偏好使用。

## E. 計算 2022/8/17 與 2024/6/24 距離幾年、幾個月又幾天

公式	=DATEDIF("2022/8/17","2024/6/24","Y")，其中 "Y" 換成 "YM"、"MD"。
解說	使用三個 DATEDIF 分別取得 幾年 / 幾個月 / 幾天。分別輸出 1 / 10 / 7，即 間隔 1 年 10 個月又 7 天。

## F. 計算 2025/10/5 往後算 5 個上班日是哪一天

公式	<ul style="list-style-type: none"> <li>不考慮國定假日 :=WORKDAY("2025/10/5",5)</li> <li>考慮國定假日 :=WORKDAY("2025/10/5",5,國定假日)</li> </ul>
解說	其中 國定假日 是將 10 月的國定假日放在儲存格中，在此有 2025/10/6、2025/10/10，也可以使用陣列代替，在此為 =WORKDAY("2025/10/5",5,{DATEVALUE("2025/10/6"),DATEVALUE("2025/10/10")})。

## G. 計算 2025/10 整個月的上班日數

公式	<ul style="list-style-type: none"> <li>=NETWORKDAYS("2025/10/1","2025/10/31")</li> <li>=NETWORKDAYS("2025/10/1","2025/10/31",國定假日)</li> </ul>
解說	國定假日 是將 10 月的國定假日放在儲存格中，也可使用陣列代替。

## 8.4 進階應用 影片 [Link \(上\)](#) / [Link \(下\)](#)

學習完前三節的內容後，大家已經了解如何進一步處理日期資料，本節將使用第 4 章的 Sales Data 介紹這些日期與時間相關的函式如何搭配其他 Google Sheets 的功能使用。本節將會介紹兩個範例，第一個是建立兩個統計表，分別計算每個月平日與週末的平均 Rating 與日均營業額，兩者的計算方式不同，前者是直接將所有交易平均，但後者必須算出每天的營業額再取平均，因此計算上比較困難。各位可以先自行練習，確認自己是否已經確實了解本章各項函式的運用，完成結果如下圖。

平均 Rating	平日	週末	日均營業額	平日	週末
2019/1	6.91	7.28	2019/1	3,645	4,056
2019/2	7.19	6.78	2019/2	3,417	3,611
2019/3	6.86	6.79	2019/3	3,344	3,924

第二個範例是建立「訂單清單搜尋器」，只要輸入訂單的起始與結束時間，便能依訂單時間順序輸出 Datetime (日期 + 時間)、Invoice ID、Branch、Total、Rating，此外為了避免輸出過於龐大，在此搜尋器限制最多輸出 50 筆資料，完成結果如下圖。

訂單起始時間	2019/2/17 18:00			
訂單結束時間	2019/2/17 21:00			
<b>訂單清單 (最多顯示 50 筆)</b>				
Datetime	Invoice ID	Branch	Total	Rating
2019/2/17 18:06	389-25-3394	C	62.0025	9.4
2019/2/17 19:11	123-35-4896	C	440.937	5.3
2019/2/17 19:41	827-77-7633	A	926.9505	9.3
2019/2/17 19:57	420-11-4919	C	374.7975	5.5

為了完成以上兩個範例，在此我們將在 Sales Data 增加 Datetime、Weekday 兩個元素。但因為整個過程較為複雜，在此不一一說明各步驟的完成方式，而是在教學影片中示範資料驗證的設定及各公式的思考脈絡，非常推薦各位同學有空的時候跟著影片一起思考並操作，可以從中學到非常多的東西。

本章完成檔案：[+ Chapter 8. 函式 \(三\)：日期與時間 完成版](#)

## Chapter 9. 函式 (四)：字串處理 影片 [Link](#)

本章示範檔案：[!\[\]\(61f857d186383e36e862eb041761959b\_img.jpg\) Chapter 9. 函式 \(四\)：字串處理](#)

本章將介紹 Google Sheets 中有關文字相關之函式，也會使用 [ptt 熱門看板](#) 的網頁資料示範各函式的用法。此外，在各節範例及最後一節的進階應用中，也會說明在處理字串的思路，絕對是學會函式使用之外更重要的技能。

### 9.1 字串基本轉換與取得相關資訊 影片 [Link](#)

本節會介紹各種字串轉換、取得字串中某些部分內容的函式，將先說明各函式的用法，再使用範例說明之：

- UPPER / LOWER / PROPER：將字串轉為大 / 小寫

語法	<a href="#"><u>UPPER / LOWER / PROPER (文字)</u></a>
輸入	文字：要轉換大小寫的字串。
輸出	將 文字 轉換成全部大寫 / 全部小寫 / 每個字的第一個字母大寫。

- LEN：取得字串的長度

語法	<a href="#"><u>LEN (文字)</u></a>
輸入	文字：要計算長度的字串。
輸出	文字 的字串長度。

- LEFT / RIGHT：取得字串的左邊 / 右邊幾個字

語法	<a href="#"><u>LEFT / RIGHT (文字, [字元數])</u></a>
輸入	<ul style="list-style-type: none"><li>• 文字：要取得某個部分的字串。</li><li>• 字元數：要取得幾個字，預設為 1。</li></ul>
輸出	文字 的左邊 / 右邊 字元數 個字。

- MID：取得字串的中間幾個字

語法	<a href="#"><u>MID (文字, 開始搜尋, 摶取長度)</u></a>
輸入	<ul style="list-style-type: none"><li>• 文字：要取得某個部分的字串。</li><li>• 開始搜尋：要從第幾個字開始計算。</li><li>• 摶取長度：要取得幾個字。</li></ul>
輸出	文字 從第 開始搜尋 個字開始，取得 摶取長度 個字。

Google Sheets 提供另一系列的函式，其長度使用 Byte 計算，用這種計算方式主要的差異在於全形的中文字是 2 個 Bytes，因此長度會算成 2，只要將函式名稱加上 B 即可，即 [LENB](#) / [LEFTB](#) / [RIGHTB](#) / [MIDB](#)，其函式輸入與輸出完全相同。

上述函式的用法都非常直觀，範例如下圖。在第 9.5 節將進一步介紹上述函式如何搭配前兩章的函式使用。

	A	B	C	D
1	Baseball 2607 棒球 ◎[棒球] Baseball is life			
2	範例	結果	公式	說明
3	A	BASEBALL 2607 棒球 ◎[棒球] BASEBALL IS LIFE	=UPPER(A1)	
4	A	baseball 2607 棒球 ◎[棒球] baseball is life	=LOWER(A1)	
5	A	Baseball 2607 棒球 ◎[棒球] Baseball Is Life	=PROPER(A1)	
6	B	39	=LEN(A1)	
7	B	B	=LEFT(A1)	
8	C	Baseball	=LEFT(A1,8)	
9	C	e	=RIGHT(A1)	
10	C	is life	=RIGHT(A1,8)	
11	D	棒球 ◎[棒球]	=MID(A1,15,9)	
12	E	44	=LENB(A1)	A1 共 5 個中文字，因此長度比 =LEN(A1) 多 5
13	E	FALSE	=LENB(A1)=LEN(A1)	判斷 A1 儲存格是否有中文字
14	E	棒球 ◎[棒	=MIDB(A1,15,9)	第二個「棒」是第 9~10 個 Bytes，但因為沒辦法一只輸出半個字所以會輸出第 9 個 Bytes

## 9.2 分割與合併字串

影片 [Link \(上\)](#) / [Link \(下\)](#)

在 Google Sheets 中與字串分割、合併相關之函式如下：

- TRIM：移除多餘的空格

語法	<b>TRIM (文字)</b>
輸入	文字：要移除空格的字串。
輸出	移除 文字 中所有的空格，只在字詞之間留下一個半形空格，即開頭、結尾、字串中重複的空格都會移除。

- SPLIT：分割字串

語法	<b>SPLIT (文字, 分隔符號, [逐個分割], [移除空白文字])</b>
輸入	<ul style="list-style-type: none"> <li>• 文字：要切割的字串，可以是單一內容或多個儲存格。</li> <li>• 分隔符號：要用什麼符號 / 文字切割，可以放多個字。</li> <li>• 逐個分割：TRUE / FALSE，分隔符號 中的字是否獨立，預設為 TRUE。 舉例來說若 分隔符號 為 "the"、逐個分割 為 TRUE 則只要出現 t / h / e 都會切割，FALSE 則是將 the 視為切割符號。</li> <li>• 移除空白文字：TRUE / FALSE，是否要在分割後移除空白文字訊息。 預設為 TRUE，舉例來說若 分隔符號 為空格，若 文字 包含多個連續空格，若 移除空白文字 為 TRUE 則只會切割一次，FALSE 則會逐一切割成多欄。</li> </ul>
輸出	將 文字 根據 分隔符號 切割，輸出會有多個欄位。

- CONCAT / &：將兩個值連接在一起

語法	<ul style="list-style-type: none"> <li>• <b>CONCAT (值_1, 值_2)</b></li> <li>• <b>值_1 &amp; 值_2</b></li> </ul>
輸入	值_1 / 值_2：要連接在一起的字串，可以是數字或文字。
輸出	將兩個值合併成字串形式，即 值_1值_2。在第 6.6 節在 COUNTIF 等函式的條件中 & 連接儲存格內容即是此道理。

- JOIN：使用指定分隔符號合併多個字串

語法	<b>JOIN ([分隔符號], 值或陣列_1, [值或陣列_2,...])</b>
輸入	<ul style="list-style-type: none"> <li>• 分隔符號：要用什麼符號 / 文字連接所有字串，預設為空 ("")。</li> <li>• 值或陣列_n：要連接的字串或陣列，每個陣列必須是單欄或單列，不然會出現 #VALUE。</li> </ul>
輸出	所有 值或陣列_n 依序使用 分隔符號 連接後的結果。

- CONCATENATE / TEXTJOIN：合併字串

語法	<ul style="list-style-type: none"> <li><b>CONCATENATE</b> (<b>文字_1</b>, [<b>文字_2</b>,...])</li> <li><b>TEXTJOIN</b> (<b>分隔符號</b>, [<b>忽略空白</b>], <b>文字_1</b>, [<b>文字_2</b>,...])</li> </ul>
輸入	<ul style="list-style-type: none"> <li><b>分隔符號</b>：要用什麼符號 / 文字連接所有輸入，預設為空 ("")。</li> <li><b>忽略空白</b>：TRUE / FALSE，若 <b>文字_n</b> 中有空值是否要忽略，預設為 FALSE，若選取 TRUE 合併時將會把空值轉為一格空格。</li> <li><b>文字_n</b>：要連接在一起的字串，可以是數字、文字或儲存格範圍。</li> </ul>
輸出	<ul style="list-style-type: none"> <li>CONCATENATE：將所有輸入連接在一起。</li> <li>TEXTJOIN：將所有輸入連接在一起，每個輸入間用 <b>分隔符號</b> 連接。</li> </ul> <p>若 CONCATENATE / TEXTJOIN 的輸入是儲存格範圍，且寬度和高度均大於 1 時，會先連接列再連接欄，例如 =CONCATENATE(A1:B2) 的合併順序為 A1 → B1 → A2 → B2。</p>

以下使用七個範例說明上述函式的用法，以及搭配其他函式使用的常見方式，其中已命名範圍「ptt看板範例」的內容為 "Baseball 2607 棒球 ◎[棒球] Baseball is life"：

#### A. 移除 ptt看板範例 多餘的空格

公式	=TRIM(ptt看板範例)
解說	使用 TRIM 即可，輸出會將 Baseball 和 is 中間的空格移除至剩下一個。

#### B. 將 ptt看板範例 的內容使用空格分割成多欄

公式	=SPLIT(ptt看板範例, " ")																								
解說	目前輸出共有七欄。若在此將 <b>移除空白文字</b> 設為 FALSE，即 =SPLIT(ptt看板範例, " ", , FALSE)，則輸出會多一欄空字串。兩者的輸出分別如下： <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>公式</th> <th colspan="7">結果</th> </tr> </thead> <tbody> <tr> <td>=SPLIT(ptt看板範例, " ")</td> <td>Baseball</td> <td>2607</td> <td>棒球</td> <td>◎[棒球]</td> <td>Baseball</td> <td>is</td> <td>life</td> </tr> <tr> <td>=SPLIT(ptt看板範例, " ", , FALSE)</td> <td>Baseball</td> <td>2607</td> <td>棒球</td> <td>◎[棒球]</td> <td>Baseball</td> <td>is</td> <td>life</td> </tr> </tbody> </table>	公式	結果							=SPLIT(ptt看板範例, " ")	Baseball	2607	棒球	◎[棒球]	Baseball	is	life	=SPLIT(ptt看板範例, " ", , FALSE)	Baseball	2607	棒球	◎[棒球]	Baseball	is	life
公式	結果																								
=SPLIT(ptt看板範例, " ")	Baseball	2607	棒球	◎[棒球]	Baseball	is	life																		
=SPLIT(ptt看板範例, " ", , FALSE)	Baseball	2607	棒球	◎[棒球]	Baseball	is	life																		

#### C. 將 ptt看板範例 的內容使用「Baseball」分割成多欄

公式	=SPLIT(ptt看板範例, "Baseball")															
解說	因為 <b>逐個分割</b> 預設為 TRUE，所以也會將 is life 的 s / i / e 當作分隔符號使用，最後輸出共有四欄，若將 <b>逐個分割</b> 設為 FALSE，即 =SPLIT(ptt看板範例, "Baseball", , FALSE)，就只會把 Baseball 當作分隔符號，此時輸出就只會有兩欄？兩者的輸出分別如下： <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>公式</th> <th colspan="4">結果</th> </tr> </thead> <tbody> <tr> <td>=SPLIT(ptt看板範例, "Baseball")</td> <td>2607</td> <td>棒球</td> <td>◎[棒球]</td> <td>i</td> </tr> <tr> <td>=SPLIT(ptt看板範例, "Baseball", , FALSE)</td> <td>2607</td> <td>棒球</td> <td>◎[棒球]</td> <td>is life</td> </tr> </tbody> </table>	公式	結果				=SPLIT(ptt看板範例, "Baseball")	2607	棒球	◎[棒球]	i	=SPLIT(ptt看板範例, "Baseball", , FALSE)	2607	棒球	◎[棒球]	is life
公式	結果															
=SPLIT(ptt看板範例, "Baseball")	2607	棒球	◎[棒球]	i												
=SPLIT(ptt看板範例, "Baseball", , FALSE)	2607	棒球	◎[棒球]	is life												

**D. 取得 B. 分割後的以下資訊：(1) 共有幾個字串、(2) 第二個字串、(3) 共出現幾次 Baseball、(4) is 出現在第幾個元素**

公式	(1) =COLUMNS(SPLIT(ptt看板範例, " ")) (2) =INDEX(SPLIT(ptt看板範例, " "),2) (3) =COUNTIF(SPLIT(ptt看板範例, " "), "Baseball") (4) =MATCH("is",SPLIT(ptt看板範例, " "),0)
解說	SPLIT 後的結果有多欄，即為陣列，可以搭配其他輸入為陣列的函式使用。

**E. 將 ptt看板範例 前面加上「看板名稱：」**

公式	<ul style="list-style-type: none"> <li>● =CONCAT("看板名稱：" , ptt看板範例)</li> <li>● ="看板名稱：" &amp; ptt看板範例</li> </ul>
解說	可使用 CONCAT / & 將兩字串連接。

**F. 將 ptt熱門看板!A2:A4 的內容合併，其合併的分隔符號為 CHAR(10)**

公式	<ul style="list-style-type: none"> <li>● =CONCATENATE('ptt熱門看板'!A2,CHAR(10),'ptt熱門看板'!A3,CHAR(10),'ptt熱門看板'!A4)</li> <li>● ='ptt熱門看板'!A2&amp;CHAR(10)&amp;'ptt熱門看板'!A3&amp;CHAR(10)&amp;'ptt熱門看板'!A4</li> <li>● =JOIN(CHAR(10), 'ptt熱門看板'!A2:A4)</li> <li>● =TEXTJOIN(CHAR(10),, 'ptt熱門看板'!A2:A4)</li> </ul>
解說	<ul style="list-style-type: none"> <li>● CHAR(10) 是在儲存格中換行的符號，CHAR 是根據 Unicode 編碼表將數字轉換成字元，在此不詳細介紹。</li> <li>● 可使用 CONCATENATE / &amp; 完成，但要重複輸入兩次 CHAR(10)。</li> <li>● 可使用 JOIN / TEXTJOIN，兩函式最主要的差別是 TEXTJOIN (1) 可以忽略空白、(2) 支援多欄多列的輸出。但在此 'ptt熱門看板'!A2:A4 並無空白、也不是多欄多列，因此兩者的用法相同。</li> </ul>

### 9.3 尋找與取代字串 影片 [Link](#)

在 Google Sheets 中與字串尋找與取代相關之函式如下：

- FIND / SEARCH：尋找特定的字出現在字串的哪個位置

語法	<a href="#"><b>FIND / SEARCH</b></a> ( <b>搜尋</b> , <b>搜尋文字</b> , [ <b>開始搜尋</b> ])
輸入	<ul style="list-style-type: none"><li>● <b>搜尋</b>：要找的字串。</li><li>● <b>搜尋文字</b>：要在哪裡找。</li><li>● <b>開始搜尋</b>：從 <b>搜尋文字</b> 的哪裡開始找，預設為 1。</li></ul>
輸出	<b>搜尋文字</b> 從第 <b>開始搜尋</b> 個字元起算，第一次出現在 <b>搜尋</b> 的位置，若找不到會輸出 #VALUE!。其中 FIND 會區分大小寫、SEARCH 則不區分。

- REPLACE：取代特定位置的字串

語法	<a href="#"><b>REPLACE</b></a> ( <b>文字</b> , <b>位置</b> , <b>長度</b> , <b>新文字</b> )
輸入	<ul style="list-style-type: none"><li>● <b>文字</b>：要在哪裡找要被取代的字串。</li><li>● <b>位置</b>：第幾個位置開始要被取代。</li><li>● <b>長度</b>：要取代的內容的長度。</li><li>● <b>新文字</b>：取代的字串。</li></ul>
輸出	<b>文字</b> 的第 <b>位置</b> 起算 <b>長度</b> 個字被取代成 <b>新文字</b> 的結果。

FIND / SEARCH / REPLACE 和字串的長度有關，因此也有使用 Bytes 計算的函式，即 [\*\*FINDB\*\*](#) / [\*\*SEARCHB\*\*](#) / [\*\*REPLACEB\*\*](#)，其函式輸入與輸出完全相同。

- SUBSTITUTE：將某字串取代為某字串

語法	<a href="#"><b>SUBSTITUTE</b></a> ( <b>搜尋文字</b> , <b>搜尋</b> , <b>取代目標</b> , [ <b>出現次數</b> ])
輸入	<ul style="list-style-type: none"><li>● <b>搜尋文字</b>：要在哪裡找要被取代的字串。</li><li>● <b>搜尋</b>：要被取代的字串。</li><li>● <b>取代目標</b>：取代的字串。</li><li>● <b>出現次數</b>：要取代第幾次出現的 <b>搜尋</b>，預設為全部都取代。</li></ul>
輸出	<b>搜尋文字</b> 的 <b>搜尋</b> 被取代為 <b>取代目標</b> 的結果，會區分大小寫。

接下來將繼續使用已命名範圍「ptt看板範例」，搭配三個基本的範例及一個進階範例說明上述函式的用法：

### A. 取得 baseball 出現在 ptt看板範例 的位置

公式	=SEARCH("baseball",ptt看板範例)
解說	<ul style="list-style-type: none"> <li>若使用 FIND 會輸出 #VALUE!，因為 FIND 會區分大小寫。</li> <li>此公式會輸出 1，若指定 開始搜尋 為任一個大於 1 的數，則會輸出第二次出現的位置 (即 24)，但若 開始搜尋 大於 24 則會輸出 #VALUE!。</li> </ul>

### B. 將 ptt看板範例 的第 10~13 個字取代為 XXX

公式	=REPLACE(ptt看板範例,10,4,"XXX")
解說	在 REPLACE 中選擇第 10~13 個字的方式與 MID 相同，輸出起始位置及字串長度，字串長度則為 13-10+1=4。

### C. 將 ptt看板範例 中棒球皆更改成籃球，中英文都修改

公式	=SUBSTITUTE(SUBSTITUTE(ptt看板範例,"Baseball","Basketball"),"棒球","籃球")
解說	在此使用兩次 SUBSTITUTE 完成，分別取代英文、中文字串。

### D. 將 ptt看板範例 的人氣 (即 2607) 取代為 XXX

公式	<ul style="list-style-type: none"> <li>=REPLACE(ptt看板範例,FIND(" ",ptt看板範例)+1,FIND(" ",ptt看板範例,FIND(" ",ptt看板範例)+1)-FIND(" ",ptt看板範例)-1,"XXX")</li> <li>=SUBSTITUTE(ptt看板範例,MID(ptt看板範例,FIND(" ",ptt看板範例)+1,FIND(" ",ptt看板範例,FIND(" ",ptt看板範例)+1)-FIND(" ",ptt看板範例)-1),"XXX")</li> </ul>
解說	<p>上述公式看起來很複雜，以下分成幾個步驟說明：</p> <ul style="list-style-type: none"> <li>將人氣取代成 XXX：可以使用以下兩種方式完成：           <ul style="list-style-type: none"> <li>=REPLACE(ptt看板範例,人氣起始位置,人氣的長度,"XXX")</li> <li>=SUBSTITUTE(ptt看板範例,人氣,"XXX")，其中 人氣 可以使用 MID(ptt看板範例,人氣起始位置,人氣的長度) 完成。</li> </ul> </li> <li><b>人氣起始位置</b>：已知 ptt看板範例 的結構為「英文名稱 人氣 其他內容」，因此人氣的起始位置為「第一個空格的位置 + 1」，因此可使用 FIND(" ",ptt看板範例)+1。</li> <li><b>人氣的長度</b>：長度為「第二個空格的位置 - 第一個空格的位置 - 1」，其中第二個空格所在的位置可使用 FIND 並將 開始搜尋 設「第一個空格的位置 + 1」即可。因此可使用 FIND(" ",ptt看板範例),FIND(" ",ptt看板範例)+1)-FIND(" ",ptt看板範例)-1。</li> <li>將人氣的起始位置、人氣的長度回填到公式中即可。</li> </ul>

上述公式看起來相當複雜，但掌握這些函式的精髓後，便能將字串內容中進行一番整理並靈活運用，第 9.5 節也會使用這些函式進一步拆解其他 ptt 的看板。

## 9.4 正規表示式相關函式 影片 [Link](#)

正規表示式是電腦科學的一個概念。使用單個字串來描述符合一系列符合某個句法規則的字串，通常被用來檢索、替換那些符合某個模式的文字。許多程式設計語言都支援利用正規表示式進行字串操作。而 Google Sheets 也有正規表示式相關的函式，提供找尋、取代、擷取內容之功能。常見正規表示式的規則如下 (改編自[官方文件](#))：

表示式	說明	例子	符合項目	不符合項目
.	任何一個字元。	d.g	d g、dog	dg、doog
*	前面字元重複 0 次以上。	do*g	dg、doog	dug
+	前面字元顯示 1 次以上。	do+g	dog、doog	dg
?	前面字元顯示 0~1 次。	do?g	dg、dog	doog
[abc]	是否為括號中的內容，括弧內的字元通常都有效，限一個字元。	d[iou]g	dog、dug	dag、deg
[a-z]	是否為括號內範圍的內容，一般範圍包括 a-z、A-Z 和 0-9，也可以將多個範圍合併，例如 [a-zA-Z0-9]、[a-zA-Z, &*]。	d[i-u]g	dog、dug	dag、deg
[^abc]	是否「不是」括號中的內容。	d[^iou]g	dag、deg	dog、dug
(aa bb)	是否為括號中的內容，括弧內的字元通常都有效，用於多個字元時。	d(a oo)g	dag、doog	dog
^	位於正規表示式的開始。	^[dh]og	dog toy	hot dog
\$	位於正規表示式的結尾。	[dh]og\$	hot dog	dog toy
{A, B}	前一個運算式重複 A 至 B 次，且 A 和 B 都是數字。	d(o{1, 3})g	dog、dooog	dg、doooog
\s	空格字元。	d\sg	d g	dg、d g
\	跳脫字元，用於字元本身有表示式意義，但欲搜尋其字元形式者，例如 .+?()[] 等。	\(dog\)	(dog)	dog
(...)	用於要取得某段字串時，用括號表示要取得的部分。	d(.*)g	doog → oo	略

各表示式符合與不符合之項目如下圖所示：

REGEX	d.g	do*g	do+g	do?g	d[iou]g	d[i-ug]	d[^iou]g	d(a oo)g	^[dh]og	[dh]og\$	d(o{1,3})g	d\sg	\(dog)
d g	V						V						V
dog	V	V	V	V	V				V	V	V		
dug	V				V	V							
dogs	V	V	V	V	V	V			V		V		
A dog	V	V	V	V	V	V			V	V	V		
d\$g	V						V						
dag	V						V	V					
deg	V						V						
dig	V				V	V							
dg		V		V									
doog		V	V				V			V			
dooog		V	V							V			
doooog		V	V										
doug									V	V			
hog									V	V			
hogs									V	V			
dog toy	V	V	V	V	V	V			V		V		
hot dog	V	V	V	V	V	V			V	V			
(dog)	V	V	V	V	V	V			V	V	V		V

了解正規表示式的規則後，接下來介紹三個使用正規表示式搜尋與取代的函式，如下：

語法	<ul style="list-style-type: none"> <li><b>REGEXMATCH</b> (文字, 規則運算式)</li> <li><b>REGEXEXTRACT</b> (文字, 規則運算式)</li> <li><b>REGEXREPLACE</b> (文字, 規則運算式, 取代)</li> </ul>
輸入	<ul style="list-style-type: none"> <li><b>文字</b>：要搜尋的文字。</li> <li><b>規則運算式</b>：正規表示式，要以文字形式呈現。</li> <li><b>取代</b>：要取代的字串。</li> </ul>
輸出	<ul style="list-style-type: none"> <li>REGEXMATCH：TRUE / FALSE，<b>文字</b> 是否有符合 <b>規則運算式</b> 的內容。</li> <li>REGEXEXTRACT：<b>文字</b> 中符合 <b>規則運算式</b> 的內容，若沒有符合的內容會輸出 #N/A!。</li> <li>REGEXREPLACE：將 <b>文字</b> 中全部符合 <b>規則運算式</b> 的內容取代成 <b>取代</b>，若沒有符合的內容則會輸出原本 <b>文字</b> 的內容。</li> </ul>

以下將繼續使用已命名範圍「ptt看板範例」搭配兩個基本範例及一個進階範例說明如何列式正規表示式並運用在公式中：

#### A. 取得 ptt看板範例 ◎ 之後的內容 (包含 ◎)

公式	=REGEXEXTRACT(ptt看板範例,"◎.*")
解說	通常要取得所有內容會使用 ".*"，在此因為只需要 ◎ 之後的內容，所以 <b>規則運算式</b> 為 "◎.*"。

#### B. 取得 ptt看板範例 [] 中的內容 (不包含 [])

公式	=REGEXEXTRACT(ptt看板範例,"\[([.*])\]")
解說	<ul style="list-style-type: none"> <li>因為 [...] 在正規表示式的語法中有意義，所以前面要加上跳脫字元 \，即 "\[.*\]"，會輸出「[棒球]」。</li> <li>因為輸出不要包含 []，所以要用小括號包住要取得的部分，即 "\[([.*])\]"。</li> </ul>

### C. 將 ptt看板範例 的人氣 (即 2607) 取代為 XXX

公式	<ul style="list-style-type: none"><li>=REGEXREPLACE(ptt看板範例, "[0-9]+", "XXX")</li><li>=REGEXREPLACE(ptt看板範例, "^(.*)\s[0-9]+\s", "\$1 XXX ")</li></ul>
解說	<ul style="list-style-type: none"><li>第一種做法較簡單，直接將在此可使用 "[0-9]+" 找出所有的數字部分，並取代成 "XXX"。</li><li>第一種做法不能應用到其他看板上，例如「PC_Shopping 363 硬體 ◎[電蝦] intel 13 14代災情回報」，會將所有的數字都變成 XXX，因此要先找出所有要取代的文字的規律，可發現人氣一定會出現在第一個空格及第二個空格之間，因為前面的看板名稱絕對不會有空格。</li><li>發現規律後，可將正規表示式調整成 "^.*?\s[0-9]+\s"，說明如下：<ul style="list-style-type: none"><li>^.*?：開頭是任意字元。</li><li>\s：空格字元，人氣的前後都有空格。</li></ul></li><li>若使用 =REGEXREPLACE(ptt看板範例, "^.*?\s[0-9]+\s", "XXX")，會發現前面的看板名稱會消失，在此可使用 (...) 包住要保留的字串，並在取代中使用 \$X (X 為數字，代表第幾個括號) 沿用原本的內容，因此規則運算式 改為 "^(.*)\s[0-9]+\s"、取代 改為 "\$1 XXX"。</li></ul>

範例 C. 有些複雜，是為了在 Google Sheets 中針對字串做出更精確、細緻的判斷。而精通正規表示式的難度不低，在此僅希望透過範例讓大家瞭解此函式的用法，因此不再針對正規表示式進行更多介紹，有興趣的同學歡迎自行研究官方文件，或是詢問 ChatGPT 相關的做法。

## 9.5 進階應用 影片 [Link \(上\)](#) / [Link \(下\)](#)

本節將使用前四節介紹的各種字串處理函式整理 ptt熱門看板的所有資料，可發現每個熱門看板的組成都是「一個空格 + 英文名稱 + 一個空格 + 人氣 + 一個空格 + 中文名稱 + 一個空格 + 看板公告」。本節將會介紹兩個範例，第一個是將 Raw Data 整理成「人氣概況 | 英文名稱 | 中文名稱 | 看板公告」的形式，步驟如下：

1. 裁減多餘的空格。
2. 分割英文名稱、人氣、中文名稱、看板公告。
3. 整理英文名稱及人氣概況：英文名稱改成首字大寫、人氣若小於 100 則輸出「<100」，否則將無條件捨去至百位數並加上 + 符號，例如 1234 變成 1200+。
4. 將字串合併成「人氣概況 | 英文名稱 | 中文名稱 | 看板公告」。

本範例將根據以上步驟逐一完成，最後再貼到 ptt熱門看板的 B 欄。

第二個範例是在使用「條件式格式設定」依序完成以下設置：

1. 將人氣 2,000 以上的看板資料設為粗體、填滿顏色設為淺紅色 2。
2. 將看板英文名稱首字是小寫的資料整列填滿顏色設為淺黃色 2。
3. 將看板中文名稱不是中文 (例如 NBA.、CPBL) 的資料整列填滿顏色設為淺藍色 2。
4. 將看板公告的開頭不是「◎[...]」的資料 (例如 ◎ 哼哼哼哼) 填滿顏色設為淺綠色 2。

因為上述兩個範例較為複雜，在此不一一說明各步驟的完成方式，而是在教學影片中示範字串處理與函式選擇時的思考脈絡，非常推薦各位同學有空的時候跟著影片一起思考並操作，可以從中學到非常多的東西。

	A	B
1	Raw Data	人氣概況   英文名稱   中文名稱   看板公告
2	Gossiping 11957 綜合 ◎[八卦] 不肥 不要自稱阿肥 =.=	11900+   Gossiping   綜合   ◎[八卦] 不肥 不要自稱阿肥 =.=
3	Stock 7273 學術 ◎[股票] 地震頻傳 請注意安全	7200+   Stock   學術   ◎[股票] 地震頻傳 請注意安全
4	C_Chat 4413 閒談 ◎[希洽] Seed Freedom 上映中	4400+   C_Chat   閒談   ◎[希洽] Seed Freedom 上映中
5	KoreaStar 2854 韓國 ◎[韓星] 韓星商業大戰？	2800+   Koreastar   韓國   ◎[韓星] 韓星商業大戰？
6	Baseball 2607 棒球 ◎[棒球] Baseball is life	2600+   Baseball   棒球   ◎[棒球] Baseball is life
7	NBA 1932 NBA. ◎[NBA] 季後賽首輪火熱進行中	1900+   NBA   NBA.   ◎[NBA] 季後賽首輪火熱進行中
8	Lifeismoney 1446 省錢 ◎推薦碼板規已修改，推文前請注意	1400+   Lifeismoney   省錢   ◎推薦碼板規已修改，推文前請注意
9	home-sale 836 房屋 ◎[房版] 房屋買賣版	800+   Home-Sale   房屋   ◎[房版] 房屋買賣版
10	HatePolitics 768 Hate ◎[政黑] 拒絕划手機、度過沒電危機	700+   HatePolitics   Hate   ◎[政黑] 拒絕划手機、度過沒電危機
11	car 748 車車 ◎[汽車] 恭喜Pellaeon成為新版主	700+   Car   車車   ◎[汽車] 恭喜Pellaeon成為新版主
12	LoL 688 遊戲 ◎[LoL] 隨時留意拉姆斯Gank	600+   LoL   遊戲   ◎[LoL] 隨時留意拉姆斯Gank
13	sex 686 男女 ◎[西斯] 我是畜生，我跟毛怪作愛	600+   Sex   男女   ◎[西斯] 我是畜生，我跟毛怪作愛
14	Tech_Job 623 工作 ◎[科技] 這裡是科技板	600+   Tech_Job   工作   ◎[科技] 這裡是科技板
15	movie 552 綜合 ◎[電影] 禁政治版規延續 注意置底	500+   Movie   綜合   ◎[電影] 禁政治版規延續 注意置底
16	Japan_Travel 495 旅遊 ◎黃金周人擠人 多估時間不可少	400+   Japan_Travel   旅遊   ◎黃金周人擠人 多估時間不可少
17	Beauty 489 聊天 ◎[表特] 貼AI圖一律水桶+退文	400+   Beauty   聊天   ◎[表特] 貼AI圖一律水桶+退文
18	MobileComm 473 資訊 ◎[通訊]	400+   Mobilecomm   資訊   ◎[通訊]
19	SportLottery 466 博弈 ◎[運彩] 板規12.0宣導至4月底	400+   Sportlottery   博弈   ◎[運彩] 板規12.0宣導至4月底

---

本章完成檔案： Chapter 9. 函式 (四)：字串處理 完成版

## Appendix 9. 函式 (四)：字串處理

### 9.2 分割與合併字串

- SPLIT：Excel 中無此函式，但是有功能類似的 TEXTSPLIT，說明如下：

語法	<b>TEXTSPLIT</b> ( <b>文字</b> , <b>欄分隔符號</b> , [ <b>列分隔符號</b> ], [ <b>移除空白文字</b> ], [ <b>區分大小寫</b> ], [ <b>缺值</b> ])
輸入	<ul style="list-style-type: none"> <li><b>文字</b>：要切割的字串，可以是單一內容或多個儲存格。</li> <li><b>欄分隔符號</b>：要用什麼符號 / 文字切割成多欄，只能放一個項目。</li> <li><b>列分隔符號</b>：要用什麼符號 / 文字切割成多欄，只能放一個項目，預設為空值 (不分隔成多列)。</li> <li><b>移除空白文字</b>：TRUE / FALSE，是否要在分割後移除空白文字訊息。預設為 FALSE (與 Google Sheets 不同)。</li> <li><b>區分大小寫</b>：0 / 1，分別代表考慮大小寫 / 忽略大小寫，預設值為 0。</li> <li><b>缺值</b>：遇到缺值的輸出，預設為 #N/A。用於分割後的欄 / 列數不同時，剩餘的儲存格即為缺值。</li> </ul>
輸出	將 <b>文字</b> 根據 <b>欄分隔符號</b> / <b>列分隔符號</b> 切割，輸出會有多欄 / 列。

- JOIN：Excel 中無此函式
- TEXTSPLIT 與一些函式不相容，例如 COUNTIF，因此無法完成範例 D(3)。

### 9.3 尋找與取代字串

各函式都適用，此外 Excel 另外有 TEXTBEFORE / TEXTAFTER 兩個函式，說明如下：

語法	<b>TEXTBEFORE</b> / <b>TEXTAFTER</b> ( <b>文字</b> , <b>分隔符號</b> , [ <b>第幾次</b> ], [ <b>區分大小寫</b> ], [ <b>文字結尾</b> ], [ <b>NA的輸出</b> ])
輸入	<ul style="list-style-type: none"> <li><b>文字</b>：要輸出的文字。</li> <li><b>分隔符號</b>：要尋找的文字。</li> <li><b>第幾次</b>：要輸出第幾個 <b>分隔符號</b> 之前 / 之後的資料，預設為 1，若輸入負數則是倒數第幾次，不得為 0。</li> <li><b>區分大小寫</b>：0 / 1，分別為考慮大小寫 / 忽略大小寫，預設值為 0。</li> <li><b>文字結尾</b>：0 / 1，將 <b>文字</b> 結尾不視為 / 視為一次 <b>分隔符號</b>，預設為 0。</li> <li><b>NA的輸出</b>：找不到第 <b>第幾次</b> 個 <b>分隔符號</b> 的話的輸出，預設為 #N/A。</li> </ul>
輸出	輸出 <b>文字</b> 中第 <b>第幾次</b> 個 <b>分隔符號</b> 之前 / 之後的文字。

### 9.4 正規表示式相關函式

Excel 中無正規表示式相關的函式。

## Chapter 10. 函式 (五)：特殊功能函式 影片 [Link](#)

本章示範檔案： [Chapter 10. 函式 \(五\)：特殊功能函式](#)

本節將會介紹一些功能較為特殊的函式，前五節包括會計、資料生成、資料型態與格式相關的函式，最後一節會介紹資料庫函式，提供一種不同的方式完成第 6.5~6.6 節學到的資料統計值，大家可以根據自己的習慣選擇使用的函式。另外因為各節的函式用途不太相同，因此本章沒有整合各節內容的進階應用。

### 10.1 計算資產折舊 影片 [Link](#)

Google Sheets 提供四種計算資產折舊費用、一種計算累計折舊的函式，分別如下：

- SLN / DDB / DB / SYD：使用直線法 / 倍數餘額遞減法 / 定率遞減法 / 年數合計法計算資產折舊費用

語法	<ul style="list-style-type: none"><li>• <b>SLN</b> (<b>費用</b>, <b>殘值</b>, <b>期限</b>)</li><li>• <b>DDB</b> (<b>費用</b>, <b>殘值</b>, <b>期限</b>, <b>周期</b>, [<b>係數</b>])</li><li>• <b>DB</b> (<b>費用</b>, <b>殘值</b>, <b>期限</b>, <b>周期</b>, [<b>月</b>])</li><li>• <b>SYD</b> (<b>費用</b>, <b>殘值</b>, <b>期限</b>, <b>周期</b>)</li></ul>
輸入	<ul style="list-style-type: none"><li>• <b>費用</b>：資產的原始成本。</li><li>• <b>殘值</b>：折舊結束時的資產價值。</li><li>• <b>期限</b>：資產折舊的總期數。</li><li>• <b>周期</b>：要計算第幾期的。</li><li>• <b>係數</b>：倍數餘額遞減法的倍數，預設為 2。</li><li>• <b>月</b>：定率遞減法折舊第一年的月數，預設為 12。</li></ul>
輸出	根據給定的輸入，計算直線法 / 倍數餘額遞減法 / 定率遞減法 / 年數合計法計算資產折舊費用。

- VDB：計算倍數餘額遞減法的累計折舊

語法	<b>VDB</b> ( <b>費用</b> , <b>殘值</b> , <b>期限</b> , <b>開始期間</b> , <b>結束期間</b> , [ <b>係數</b> ], [ <b>不切換</b> ])
輸入	<ul style="list-style-type: none"><li>• <b>費用</b> / <b>殘值</b> / <b>期限</b> / <b>係數</b>：與 DDB 相同。</li><li>• <b>開始期間</b> / <b>結束期間</b>：計算第幾期末到第幾期末的累計折舊，例如想計算前三年的累計折舊，且一期為 1 年，就使用 0 / 3。</li><li>• <b>不切換</b>：TRUE / FALSE，預設為 FALSE，當折舊大於遞減餘額計算時，是否切換到直線折舊法。</li></ul>
輸出	使用倍數餘額遞減法計算指定期間內的資產折舊值。

雖然這些折舊方式的計算過程也可以透過四則運算完成，但使用上述函式可以節省相當多時間，且用法也非常直觀且類似，範例如下圖：

費用	50,000	週期	折舊法	公式	1	2	3	4	5
殘值	5,000	折舊費用	直線法	=SLN(費用,殘值,期數)	9,000	9,000	9,000	9,000	9,000
期數	5		倍數餘額遞減法 (2倍)	=DDB(費用,殘值,期數,周期)	20,000	12,000	7,200	4,320	1,480
			倍數餘額遞減法 (1.5 倍)	=DDB(費用,殘值,期數,周期,1.5)	15,000	10,500	7,350	5,145	3,602
			定率遞減法	=DB(費用,殘值,期數,周期)	18,450	11,642	7,346	4,635	2,925
			年數合計法	=SYD(費用,殘值,期數,周期)	15,000	12,000	9,000	6,000	3,000
累計折舊		累計折舊	倍數餘額遞減法 (2 倍)	=VDB(費用,殘值,期數,0,周期)	20,000	32,000	39,200	43,520	45,000
			倍數餘額遞減法 (1.5 倍)	=VDB(費用,殘值,期數,0,周期,1.5,FALSE)	15,000	25,500	32,850	38,925	45,000

## 10.2 計算年金數值

影片 [Link](#)

因為會計中貨幣有時間價值，導致今年的 100 元會大於明年的 100 元，而有了年金終值與現值的概念。舉例來說，連續十年年底在銀行存入 100 元，於 10 年後的總價值即為終值、於今日的價格即為現值。在計算終值與現值時有五個要素，分別為：(1) 利率 (時間價值)、(2) 投入期數、(3) 每一期投入金額、(4) 現值、(5) 終值。而在 Google Sheets 中，只需要之後五個要素中的其中四項，就可以使用對應的函式求得最後一個要素的值。

各函式分別如下：

語法	<ul style="list-style-type: none"> <li><b>FV</b> (利率, 期數, 支付金額, 現值, [期末或期初])</li> <li><b>PV</b> (利率, 期數, 支付金額, [未來值], [期末或期初])</li> <li><b>PMT</b> (利率, 期數, 現值, [未來值], [期末或期初])</li> <li><b>NPER</b> (利率, 支付金額, 現值, [未來值], [期末或期初])</li> <li><b>RATE</b> (期數, 支付金額, 現值, [未來值], [期末或期初], [猜測利率])</li> </ul>
輸入	<ul style="list-style-type: none"> <li><b>利率</b>：利率。</li> <li><b>期數</b>：期數。</li> <li><b>支付金額</b>：每期需要投入的金額。</li> <li><b>現值</b>：年金的現值。</li> <li><b>未來值</b>：完成最後一次支付後的剩餘金額，預設為 0。</li> <li><b>期末或期初</b>：0 / 1，付款的時間點，分別代表期末 (普通年金) / 期初 (到期年金)，預設為 0。</li> <li><b>猜測利率</b>：利率估計值，預設為 0.1，用來使利率計算更快速，通常省略。</li> </ul>
輸出	給定各項條件下的終值 / 現值 / 每期需支付的金額 / 需支付的期數 / 隱含利率。

上述五個函式之間的關係如下方程式所示，因此部分項目也可以直接透過四則運算得到相關的值：

- 普通年金 (**期末或期初** = 0)： $PV \times (1 + RATE)^{NPER} + PMT \times \left( \frac{(1+RATE)^{NPER} - 1}{RATE} \right) + FV = 0$
- 到期年金 (**期末或期初** = 1)：

$$PV \times (1 + RATE)^{NPER} + PMT \times (1 + RATE) \times \left( \frac{(1+RATE)^{NPER} - 1}{RATE} \right) + FV = 0$$

若將上述函式直接使用於公式中，結果如下圖所示，其中需注意在函式中若金額為負代表現金流出、金額正代表現金流入：

	A	B	C	D	E	F	G	H
1	函式	利率	期數	支付金額	現值	未來值	期末或期初	公式
2	<b>FV</b>	10%	3	-100.0	-1,000.0	1,662.0	0	=FV(B2,C2,D2,E2,G2)
3	<b>PV</b>	10%	3	-100.0	-1,000.0	1,662.0	0	=PV(B3,C3,D3,F3,G3)
4	<b>PMT</b>	10%	3	-100.0	-1,000.0	1,662.0	0	=PMT(B4,C4,E4,F4,G4)
5	<b>NPER</b>	10%	3	-100.0	-1,000.0	1,662.0	0	=NPER(B5,D5,E5,F5,G5)
6	<b>RATE</b>	10%	3	-100.0	-1,000.0	1,662.0	0	=RATE(C6,D6,E6,F6,G6)

以下使用五個範例說明各函式的使用時機：

- A. 目前我的銀行帳戶有 20 萬元，之後每個月初領到薪水存 2 萬元進去，若利率為每年 3%，每月複利一次，那 20 年後會有多少錢？

公式	=FV(0.25%, 20*12, -20000, -200000, 1)
解說	要計算年金終值可使用 FV，其中要注意是每月複利一次，因此利率期數都要調整，利率變 $3\% \div 12 = 0.25\%$ 、期數變 $20 * 12$ ，此外為期初支付，因此 <b>期末或期初</b> = 1，會輸出 \$6,946,606。

- B. 我最近向銀行借了一筆貸款，該貸款在 12 年後到期，這 12 年間每年年底必須繳交貸款本息共 1 萬元，若目前年利率為 6%。請問我向銀行貸了多少錢？

公式	=PV(6%, 12, -10000, , 0)
解說	計算借款的現值可使用 PV，會輸出 \$83,838。

- C. 我向銀行貸款 800 萬元，期限 20 年，每年年底本息攤還 50 萬元，請問此貸款年利率是多少？

公式	=RATE(20, -500000, 8000000, , 0)
解說	計算利率應使用 RATE，其中每年流出 50 萬元為負值，會輸出 2.23%。

- D. 若我向朋友借了 50 萬元，然後每年年底還他 5 萬元，若目前年利率為 5%，我需要還幾年才能還清這筆錢？

公式	=NPER(5%, -50000, 500000, , 0)
解說	計算期數應使用 NPER，會輸出 14.21 年。

- E. 我預計 3 年後買一輛新車 80 萬元，但我目前只有 20 萬元，若目前年利率 3%，每月複利一次，從現在起我需要每月月初存多少錢，才能在 3 年後買到車？

公式	=PMT(0.25%, 3*12, -200000, 800000, 1)
解說	計算每期投入金額可使用 PMT，其中每月複利一次，因此利率期數都要調整，利率變 $3\% \div 12 = 0.25\%$ 、期數變 $3 * 12$ ，會輸出 -15,410。

本節介紹的函式在會計中非常重要，但不用記住每個函式輸入的順序，更重要的是在需要計算時間價值時能辨別清楚各項數字應該輸入什麼內容。

## 10.3 資料生成 [影片 Link](#)

本節將介紹與生成資料相關的函式，包括重複資料、等差級數資料與隨機資料，如下：

- REPT：將文字重複多次

語法	<b>REPT (重複文字, 重複次數)</b>
輸入	<ul style="list-style-type: none"><li>• <b>重複文字</b>：要重複的字元或字串。</li><li>• <b>重複次數</b>：要重複的次數，最多 100 次。</li></ul>
輸出	將 <b>重複文字</b> 重複 <b>重複次數</b> 次，輸出在單一儲存格。

- SEQUENCE：生成多列多欄的等差數列

語法	<b>SEQUENCE (列數, [欄數], [起始值], [間距])</b>
輸入	<ul style="list-style-type: none"><li>• <b>列數 / 欄數</b>：輸出列數 / 欄數，列數必填、欄數預設為 1。</li><li>• <b>起始值</b>：首項，預設為 1。</li><li>• <b>間距</b>：公差，預設為 1。</li></ul>
輸出	輸出 <b>列數</b> 列、 <b>欄數</b> 欄、首項為 <b>起始值</b> 、公差為 <b>間距</b> 的等差數列，若 <b>欄數</b> 大於 1 會先左至右填滿整列，填滿後再換下一列。

- RAND / RANDARRAY：隨機生成 0~1 之間的小數

語法	<ul style="list-style-type: none"><li>• <b>RAND ()</b></li><li>• <b>RANDARRAY ([列數], [欄數])</b></li></ul>
輸入	<b>列數 / 欄數</b> ：輸出列數 / 欄數，皆預設為 1。
輸出	0~1 之間的隨機小數，每編輯一次試算表就會重新生成。其中 RAND 只會在使用函式的儲存格輸出，RANDARRAY 則輸出 <b>列數</b> 列、 <b>欄數</b> 欄。

- RANDBETWEEN：隨機生成 m~n 之間的整數

語法	<b>RANDBETWEEN (最小值, 最大值)</b>
輸入	<b>最小值 / 最大值</b> ：隨機整數的最小、最大值。
輸出	隨機生成一個 <b>最小值</b> 與 <b>最大值</b> 之間的整數。

- NORMINV：輸出指定值、平均值和標準值的反常態分佈函式值

語法	<b>NORMINV (x, 平均值, 標準差)</b>
輸入	<ul style="list-style-type: none"> <li>• <b>x</b>：指定的值，必須介於 0~1 之間。</li> <li>• <b>平均值 / 標準差</b>：常態分布的平均數、標準差。</li> </ul>
輸出	<p>傳回 <b>x</b>、<b>平均值</b> 和 <b>標準差</b> 的反常態分佈函式值。          例如 =NORMINV(0.025, 100, 15) 會輸出 129.4，在平均 = 100、標準差 = 15 的情況下，數值大於 129.4 的機率只有 <math>1 - 0.975 = 0.025</math>。</p>

以下使用六個範例說明上述函式的用法：

#### A. 重複 5 次 "Hi"，每次中間用半形逗號間隔

公式	<ul style="list-style-type: none"> <li>=LEFT(REPT("Hi,",5),LEN("Hi,")*5-1)</li> <li>=JOIN(", ",SPLIT(REPT("Hi,",5),","))</li> </ul>
解說	<ul style="list-style-type: none"> <li>兩者的做法相似，都是先使用 REPT 重複五次 "Hi,"，再想辦法將最後的逗點去掉。</li> <li>第一種方法使用 LEFT 取得除了最後一個逗點之外的內容，長度則使用 ("Hi," 的長度 × 5) - 1 即可。</li> <li>第二種方法先用 SPLIT 分割成多欄，再使用 JOIN 合併，是因為 SPLIT 的 移除空白文字 預設 TRUE，為會自動去掉最後一個逗號的內容，因此合併後不會有最後的逗點。</li> </ul>

#### B. 生成一個 2 列、3 欄的等差級數，其首項為 1、公差為 5

公式	=SEQUENCE(2,3,1,5)
解說	依序輸入列數、欄數、首項、公差 (間距) 即可。

#### C. 隨機生成 1 列、3 欄在 0~1 之間的小數

公式	=RANDARRAY(1,3)
解說	依序輸入列數、欄數即可。

#### D. 隨機生成一個 20~21 之間的小數

公式	=20+RAND()
解說	RAND 只會生成 0~1，即非整數的小數部分，因此只要加上整數 20 即可。

## E. 隨機生成一個 30~40 之間的小數

公式	<ul style="list-style-type: none"><li>=RANDBETWEEN(30,39)+RAND()</li><li>=30+10*RAND()</li></ul>
解說	<ul style="list-style-type: none"><li>第一種方式是與 D. 相同，只是將整數部分改成使用 RANDBETWEEN，但因為 RANDBETWEEN 包含 最大值，因此要將 最大值 設為 39。</li><li>第二種方式是將 RAND 倍數放大，例如 10*RAND() 會輸出 0~10 之間的小數，再加上 30 就會變成 30~40 之間的小數。</li></ul>

## F. 隨機生成一個數字，其平均值為 0、標準差為 1

公式	=NORMINV(RAND(),0,1)
解說	可以將這個例子當作是如何生成一個常態分配的隨機數字，而因為 NORMINV 的 $x$ 要在 0~1 之間，所以 RAND() 剛好可以作為 $x$ 。

了解本節的函式後，可以用於各種需要抽獎、實驗等需要隨機數字的時機 (例如：[蒙地卡羅模擬法](#))，更可以在學習統計學之後搭配 Google Sheets / Excel 中各種統計相關的函式進行計算，但各項統計函式礙於篇幅的關係，在此僅介紹較常用的 NORMINV，請有興趣的同學再自行參考各式官方文件。

## 10.4 資料型態判斷 影片 [Link](#)

在第 2.5 節曾經介紹 TYPE 函式的用法，其輸出是代表不同資料型態的數字，但在部分資料中其實包括更多其他種類，例如數字可能包含日期、奇偶數，文字可能包含 Email、網址等，本節介紹各種 IS- 開頭的函式即是用來判斷儲存格是否是某種資料型態，如下：

語法	<a href="#">ISTEXT</a> / <a href="#">ISNONTEXT</a> / <a href="#">ISNUMBER</a> / <a href="#">ISLOGICAL</a> / <a href="#">ISEMAIL</a> / <a href="#">ISURL</a> / <a href="#">ISODD</a> / <a href="#">ISEVEN</a> / <a href="#">ISDATE</a> / <a href="#">ISERROR</a> / <a href="#">ISNA</a> / <a href="#">ISERR</a> / <a href="#">ISREF</a> / <a href="#">ISBLANK</a> / <a href="#">ISFORMULA</a> (值)
輸入	值：要判斷其類型的資料。
輸出	TRUE / FALSE，分別判斷： <ul style="list-style-type: none"> <li>• ISTEXT / ISNONTEXT / ISNUMBER / ISLOGICAL：值 是否是文字 / 非文字 / 數字 / 邏輯運算子。</li> <li>• ISEMAIL / ISURL：值 是否為有效的 Email / 有效的網址。</li> <li>• ISODD / ISEVEN / ISDATE：(可轉換為) 奇數 / 偶數 / 日期。</li> <li>• ISERROR / ISNA / ISERR：任何錯誤型態 / #N/A! / 非 #N/A! 的錯誤型態。</li> <li>• ISREF：值 是否為有效的儲存格參照。</li> <li>• ISBLANK / ISFORMULA：值 的儲存格是否是空值 / 含有公式。</li> </ul>

上述的函式用法都非常直觀，範例如下圖，這些函式適用於限定儲存格內容的資料型態、條件式格式設定中，也可以搭配其他函式使用，例如 ISNUMBER + MATCH 可以判斷想要查找的項目是否有出現在 MATCH 指定的範圍中。

	A	B	C	D	E	F
1	範例	結果	公式	範例	結果	公式
2	A	TRUE	=ISTEXT("Hi")	D	TRUE	=ISERROR(1+A")
3		FALSE	=ISNONTEXT("Hi")		FALSE	=ISNA(1+A")
4		FALSE	=ISNUMBER("Hi")		TRUE	=ISERR(1+A")
5		TRUE	=ISNUMBER(123)	E	FALSE	=ISREF(#REF!)
6		TRUE	=ISLOGICAL(1+1<>2)		TRUE	=ISREF(D1:D)
7	B	TRUE	=ISEMAIL("123@gmail.com")		FALSE	=ISREF(#REF!+C1)
8		TRUE	=ISURL("google.com")	F	#N/A	=ISBLANK()
9	C	TRUE	=ISEVEN(32)		FALSE	=ISBLANK("")
10		TRUE	=ISEVEN("32")		FALSE	=ISBLANK(C1)
11		FALSE	=ISDATE(2024/12/31)		TRUE	=ISFORMULA(B2)
12		FALSE	=ISDATE(DATEVALUE("2024/12/31"))		#N/A	=ISFORMULA(B2+B3)
13		TRUE	=ISDATE("2024/12/31")		#N/A	=ISFORMULA(1+2)

## 10.5 數值調整與格式轉換 影片 [Link](#)

在 Google Sheets 中有許多數值調整與格式轉換相關的函式，但使用率普遍較低，以下僅說明幾個使用頻率較高的函式，主要包括四捨五入 / 無條件捨去 / 無條件進位、數值格式調整，以及數字與文字之間的格式轉換

- ROUND / ROUNDDOWN / ROUNDUP：四捨五入 / 無條件捨去 / 無條件進位

語法	<a href="#">ROUND / ROUNDDOWN / ROUNDUP (值, [位數])</a>
輸入	<ul style="list-style-type: none"> <li><b>值</b>：要四捨五入 / 無條件捨去 / 無條件進位的值。</li> <li><b>位數</b>：要四捨五入 / 無條件捨去 / 無條件進位到小數點後幾位數，預設為 0，若 -1 則為十位數、-2 為百位數，以此類推。</li> </ul>
輸出	<b>值</b> 四捨五入 / 無條件捨去 / 無條件進位至小數點後第 <b>位數</b> 位。若 <b>值</b> 是負數時會取絕對值再捨去或進位，例如 =ROUNDUP(-12.34,0) 會輸出 -13。

- MROUND / FLOOR / CEILING：四捨五入 / 無條件捨去 / 無條件進位到指定倍數

語法	<ul style="list-style-type: none"> <li><a href="#">MROUND (值, 係數)</a></li> <li><a href="#">FLOOR / CEILING (值, [係數])</a></li> </ul>
輸入	<ul style="list-style-type: none"> <li><b>值</b>：要四捨五入 / 無條件捨去 / 無條件進位的值。</li> <li><b>係數</b>：要四捨五入 / 無條件捨去 / 無條件進位的基數，不一定要是 10 的倍數。</li> </ul>
輸出	<b>值</b> 四捨五入 / 無條件捨去 / 無條件進位至最接近 <b>係數</b> 的倍數，例如 =FLOOR(22,3) 會輸出 21、=FLOOR(-22,3) 會輸出 -24。

- TO\_DATE / TO\_DOLLARS / TO\_PERCENT / TO\_PURE\_NUMBER：將數字轉成指定格式的數字

語法	<a href="#">TO_DATE / TO_DOLLARS / TO_PERCENT / TO_PURE_NUMBER (值)</a>
輸入	<b>值</b> ：要調整格式的數字，必須是數字格式。
輸出	將 <b>值</b> 調整調整成日期 / 金額 / 百分比 / 純數字格式，其中 TO_DATE 會以 1899/12/30 為 0、1899/12/31 為 1，以此類推。若 <b>值</b> 無法本身不是數字或數字格式的字串，會輸出原本的 <b>值</b> 。

使用 TO\_DATE / TO\_DOLLARS / TO\_PERCENT / TO\_PURE\_NUMBER 時若儲存格本身設定有格式時，會以儲存格設定的格式為主，例如將 =TO\_PERCENT(45000) 所在的儲存格格式設為日期，輸出會是日期格式 (2023/3/15)，而非 4500000%。

- VALUE：將日期、時間或數字格式字串轉換成數字

語法	<u>VALUE</u> ( <b>值</b> )
輸入	值：要調整格式的數字，可以是數字或數字格式的字串。
輸出	將 值 的日期、時間或數字格式字串轉換成沒有格式的純數字。若 值 無法本身不是數字或數字格式的字串，會輸出 #VALUE!。

- TO\_TEXT / FIXED / DOLLAR：將數字轉成文字

語法	<ul style="list-style-type: none"> <li>• <u>TO_TEXT</u> (<b>值</b>)</li> <li>• <u>FIXED</u> (<b>值</b>, [位數], [隱藏千分位符號])</li> <li>• <u>DOLLAR</u> (<b>值</b>, [位數])</li> </ul>
輸入	<ul style="list-style-type: none"> <li>• 值：要調整格式的數字，可以是數字或數字格式的字串。</li> <li>• 位數：要四捨五入到小數點後第幾位，若 -1 則為十位數、-2 為百位數，以此類推。在 FIXED 和 DOLLAR 中預設為 2。</li> <li>• 隱藏千分位符號：TRUE / FALSE 或 1 / 0，是否要顯示千分位符號，預設為 FALSE (有千分位符號)。</li> </ul>
輸出	<ul style="list-style-type: none"> <li>• TO_TEXT：將 值 轉換為純文字。</li> <li>• FIXED：將 值 轉換為至小數點後第 位數 位的文字。</li> <li>• DOLLAR：將 值 轉換為指定語言版本的貨幣格式，台灣為 NT\$。</li> </ul>

- TEXT：將數字轉成文字，其文字格式可設定更詳細

語法	<u>TEXT</u> ( <b>值</b> , <b>格式</b> )
輸入	<ul style="list-style-type: none"> <li>• 值：要調整格式的數字，可以是數字或數字格式的字串。</li> <li>• 格式：要輸出的格式，為字串，重要格式如下 (詳見官方文件說明)：           <ul style="list-style-type: none"> <li>◦ 0：若有數值就輸出數值，沒有數字就輸出 0。</li> <li>◦ #：若有數值就輸出數值，沒有數字就省略。</li> <li>◦ yy / yyyy：日期的「年」要兩位數 / 四位數。</li> <li>◦ m / mm / mmm：日期的「月」要一位數 / 強制兩位數 / 全名。</li> <li>◦ d / dd：日期的「日」要一位數 / 強制兩位數。</li> <li>◦ ddd / dddd：日期的星期要縮寫 (週幾) / 全名 (星期幾)。</li> </ul> </li> </ul>
輸出	將 值 轉換為 格式 的純文字。

使用 TO\_TEXT / FIXED / DOLLAR / TEXT 將數值轉換為文字後，便不會隨著儲存格設定的數值格式而改變，因為儲存格的資料已經被轉為文字格式。然而，若要將使用過這些函式的資料進行加減還是會正常運作，因為 Google Sheets 在進行四則運算時，會將嘗試將內容轉為數字計算，例如 =1+"2" 會輸出 3。

上述的函式用法都非常直觀，範例如下圖：

範例	結果	公式	範例	結果	公式
A	12.3	=ROUND(12.34,1)	E	45,000	=TO_TEXT("45,000")
	-12.4	=ROUNDUP(-12.34,1)		45,000.00	=FIXED(45000)
	-12.3	=ROUNDDOWN(-12.34,1)		45000	=FIXED(45000,0,TRUE)
B	12.3	=MROUND(12.34,0.1)	F	NT\$45,000.00	=DOLLAR(45000)
	12.5	=CEILING(12.34,0.5)		NT\$45,000	=DOLLAR(45000,0)
	-12.5	=FLOOR(-12.34,0.5)		\$1,235	=TEXT(1234.567,"\$#,##")
C	2023/3/15	=TO_DATE(45000)	F	\$1,234.57	=TEXT(1234.567,"\$#,#.00")
	NT\$45,000.00	=TO_DOLLARS(45000)		001234.6	=TEXT(1234.567,"000000.##")
	63%	=TO_PERCENT(0.63)		23/3/15	=TEXT(45000,"yy/m/d")
D	0.63	=TO_PURE_NUMBER(63%)	F	2023/03/15 週三	=TEXT(45000,"yyyy/mm/dd ddd")
	45000	=VALUE("2023/3/15")		2023年3月15日 星期三	=TEXT(45000,"yyyy年m月d日 dddd")
D	45000	=VALUE("NT\$45,000")	F	2023年3月15日 星期三	=TEXT("2023/3/15","yyyy年m月d日 dddd")

## 10.6 資料庫函式 影片 [Link \(上\)](#) / [Link \(下\)](#)

[資料庫函式 \(Database Function\)](#) 主要用於結構化的數據，可以使用資料庫函式快速的計算符合特定條件的基本統計資料。共包括 12 項，雖然每個資料庫函式計算的項目不同，但其輸入皆相同。

語法	<a href="#">DCOUNT</a> / <a href="#">DCOUNTA</a> / <a href="#">DMIN</a> / <a href="#">DMAX</a> / <a href="#">DSUM</a> / <a href="#">DPRODUCT</a> / <a href="#">DAVERAGE</a> / <a href="#">DVAR</a> / <a href="#">DVARP</a> / <a href="#">DSTDEV</a> / <a href="#">DSTDEVP</a> / <a href="#">DGET</a> ( <a href="#">資料庫</a> , <a href="#">欄位</a> , <a href="#">條件</a> )
輸入	<ul style="list-style-type: none"> <li>● <b>資料庫</b>：資料庫的範圍 / 陣列，會自動將第一列視為欄位名稱。</li> <li>● <b>欄位</b>：要計算統計資料或輸出的欄位，可以有以下兩種形式：             <ul style="list-style-type: none"> <li>○ 字串：欄位名稱。</li> <li>○ 數字：左邊數來第幾列。</li> </ul> </li> <li>● <b>條件</b>：要篩選的條件，可以是陣列或儲存格範圍。第一列是有條件的欄位名稱，第二列開始是各欄位的條件，以陣列形式表達即為：{欄位1, [欄位2...]; 欄位1條件1, [欄位2條件1...]; [欄位1條件2, 欄位2條件2...]}。另外條件的輸入方式與 SUMIF / COUNTIF 等函式相似，如下：             <ul style="list-style-type: none"> <li>○ 檢查數字：條件可以是單一數字或包含運算子的字串，例如 100、"&gt;=100"、"&lt;=2023/1/1" 等。</li> <li>○ 檢查文字：可使用第 6.6 節介紹到的萬用字元 (?/*)。</li> <li>○ 無限制：可直接輸入空值 ("") 即可。</li> </ul> </li> </ul>
輸出	<p>資料庫中「符合任一列所有條件」的資料的統計結果。例如 =DCOUNTA(<a href="#">資料</a>, "學號", {"國文分數", "數學分數"; "&gt;=80", "", "&gt;=60", "&gt;=60"}) 會計算「國文分數 80 分以上」或「國文與數學分數都 60 分以上」的學號數。</p> <p>輸出的格式都會預設為自動，因此輸出是字串或日期則要自行更改 (DGET 才會有此類輸出)，各函式分別統計以下結果：</p> <ul style="list-style-type: none"> <li>● DCOUNT / DCOUNTA：計算資料筆數，同 COUNT / COUNTA。</li> <li>● DMIN / DMAX：計算最大值 / 最小值，同 MIN / MAX。</li> <li>● DSUM / DPRODUCT / DAVERAGE：計算加總 / 相乘總和 / 平均，同 SUM / <a href="#">PRODUCT</a> / AVERAGE。</li> <li>● DVAR / DVARP / DSTDEV / DSTDEVP：計算變異數 / 標準差，同 VAR / VARP / STDEV / STDEVP。</li> <li>● DGET：輸出符合條件的值，類似支援多條件的 VLOOKUP，但輸出僅限一筆，若無符合條件或有超過一筆符合條件的項目，會輸出 #NUM!。</li> </ul>

以下使用第 1 章的交易記錄檔並搭配七個範例說明函式中的條件設定，為了方便已將交易記錄檔整張工作表 (A:K) 命名為「交易記錄」：

### A. 店號 105 的總金額

公式	=DSUM(交易記錄,"貨品總金額",{"店號";105})
解說	<ul style="list-style-type: none"> <li>計算 "貨品總金額" 的加總，可使用 DSUM。</li> <li>已知貨品總金額在 交易記錄 的第 11 欄，因此也可使用 =DSUM(交易記錄,11,{"店號";105})。</li> </ul>

### B. 店號 105 及 108 的總金額

公式	=DSUM(交易記錄,"貨品總金額", {"店號";105;108})
解說	使用「或」的條件要分成兩列，因此使用 {"店號";105;108}。

### C. 店號 105 大分類名稱為 410 的商品的總金額

公式	=DSUM(交易記錄,"貨品總金額", {"店號","大分類名稱";105,410})
解說	使用「和」的條件要分成兩欄，因此使用 {"店號","大分類名稱";105,410}。

### D. 貨名有「台灣」的商品的交易總筆數

公式	=DCOUNTA(交易記錄,"貨名", {"貨名";"*台灣"})
解說	<ul style="list-style-type: none"> <li>計算交易總筆數可使用 COUNTA，欄位 放入隨便一欄都可以。</li> <li>"*台灣*" 在條件中使用萬用字元可撈出所有包含台灣的貨名。</li> </ul>

### E. 2022/11/1~2022/11/30 的總金額

公式	=DSUM(交易記錄,"貨品總金額", {"交易日期","交易日期";">=2022/11/1", "<=2022/11/30"})
解說	和 SUMIFS 一樣要將起始與結束日期視為兩個條件。

### F. 大分類名稱為 410，或是大分類名稱為 411 且貨名提及「奶粉」的消費總額

公式	=DSUM(交易記錄,"貨品總金額", {"大分類名稱","貨名";410,"";411,"*奶粉*"})
解說	此範例有兩個條件，但第一個條件並不用進一步篩選貨名，因此在貨名的條件直接輸入空值 ("")。

### G. (1) 清境礦泉水-1410cc (2) 花蓮煙台大白菜的交易日期

公式	(1) =DGET(交易記錄,"交易日期", {"貨名";"清境礦泉水-1410cc"}) (2) =DGET(交易記錄,"交易日期", {"貨名";"花蓮煙台大白菜"})
解說	(1) 會輸出 44821 (可將格式轉成日期 2022/9/17)，但 (2) 會輸出 #NUM!，因為有多筆 "花蓮煙台大白菜" 項目。

除了使用陣列的方式在公式中輸入條件外，也可以使用儲存格作為 **條件**，如下圖所示。  
使用儲存格作為條件後，可以搭配資料驗證製作下拉式選單，便能讓檔案的使用者用更簡單的方式查詢想要的資料。

	A	B	C	D	E	F	G
1	範例	內容	結果	公式	欄位	條件	
2	A	店號 105 的總金額	46,397	=DSUM(交易記錄,E2,{F2:F3})	貨品總金額	店號	
3						105	
4	B	店號 105 及 108 的總金額	58,249	=DSUM(交易記錄,E4,F4:F6)	貨品總金額	店號	
5						105	
6						108	
7	C	店號 105 大分類名稱為 410 的商品的總金額	6,578	=DSUM(交易記錄,E7,F7:G8)	貨品總金額	店號	大分類名稱
8						105	410
9	D	貨名有「台灣」的商品的交易總筆數	142	=DCOUNTA(交易記錄,E9,F9:F10)	貨名	貨名	
10						*台灣*	
11	E	2022/11/1~2022/11/30 的總金額	37,546	=DSUM(交易記錄,E12,F12:G13)	貨品總金額	交易日期	交易日期
12						>=2022/1/1	<=2022/11/30
13	F	大分類名稱為 410，或是大分類名稱為 411 且貨名提及「奶粉」的消費總額	10,855	=DSUM(交易記錄,E15,F15:G17)	貨品總金額	大分類名稱	貨名
14						410	
15						411	*奶粉*
16	G(1)	「清境礦泉水-1410cc」的交易日期	2022/9/17	=DGET(交易記錄,E18,F18:F19)	交易日期	貨名	
17							清境礦泉水-1410cc
18	G(2)	「花蓮煙台大白菜」的交易日期	#NUM!	=DGET(交易記錄,E20,F20:F21)	交易日期	貨名	
19							花蓮煙台大白菜

本章完成檔案： Chapter 10. 函式 (五)：特殊功能函式 完成版

## Appendix 10. 函式 (五)：特殊功能函式

### **10.4 資料型態判斷**

Excel 中沒有 ISEMAIL、ISURL、ISDATE。

### **10.5 數值調整與格式轉換**

Excel 中沒有 TO\_DATE、TO\_DOLLARS、TO\_PERCENT、TO\_PURE\_NUMBER、  
TO\_TEXT。

## Chapter 11. 函式 (六)：陣列與陣列函式 影片 [Link](#)

本章示範檔案： [Chapter 11. 函式 \(六\)：陣列與陣列函式](#) 影片 [Link](#)

本章介紹了 Google Sheets 中非常重要的陣列概念，再介紹如何使用 ARRAYFORMULA 將原本僅用於單一資料的函式延伸至整個函式，後兩節介紹一些陣列相關的函式，讓大家能用更靈活的方式操作 Google Sheets。另外本章也不會有進階應用，而是將較困難的範例直接放在各節中。

### 11.1 陣列 (Array) 基本概念 影片 [Link](#)

在第 7.3 節有稍微提及陣列的概念，其實就是包含多個欄列組成的表格，其表示方式可以參考[官方文件](#)之說明，簡單來說陣列包括以下四個要素：

- 大括號 {}：創立陣列，之後會在大括號內填入陣列的內容。
- 半形逗號 ,：陣列中的分欄符號。
- 半形分號 ;：陣列中的分列符號。
- 欲組合的內容：可以是單一值、儲存格範圍或某公式的輸出等，其中內容的欄或列必須一致才能組合，例如可以使用半形逗號連接 A1:B3 和 D1:D3，因為其列數相同，但若使用分號則會跳出 #REF!，因為前者有兩欄、但後者只有一欄。各種類型的內容範例如下：
  - 單一值：例如 ={1,2,3,4}、={"A","B";"C","D"}。
  - 儲存格範圍：例如 ={B1:B5,D1:D5}、={A3:A6,C3:D6}，但要避免輸出範圍超過儲存格範圍，例如在第二列輸入={B:B,D:D}，會使試算表自動插入無限多列造成當機，這是因為輸出的列數會超過原有的儲存格範圍，因此會自動插入更多列，但同時使 B:B 有更多列，輸出也因此更多列，形成無限循環。
  - 某公式的輸出：例如使用 IMPORTRANGE、OFFSET、UNIQUE、FILTER 等函式輸出的值。

其實在前面多個章節的函式都有使用陣列作為輸入，只要函式的輸入會需要多欄 / 列時，就可以使用陣列作為輸入，以下舉幾個常用的範例說明：

- SPARKLINE 的 選項 中使用陣列調整圖表細節。
- VLOOKUP 將要搜尋的項目移至 範圍 首欄，例如 =VLOOKUP("A",{B:B,A:A},2,0)。
- UNIQUE / FILTER 要連接不相鄰的多個欄位，例如 =UNIQUE({A:A,D:D})。
- UNIQUE 去除單一欄位的重複項，但來自不同範圍，例如 =UNIQUE({**最近半年註冊會員**;最近半年消費會員}) 可以輸出所有最近半年有註冊或消費的會員。

其實除了輸入是範圍的函式外，輸入如果是單一的值也可以使用陣列，只是這時候會需要另一個輔助的函式：ARRAYFORMULA，而下一節將進一步說明這個函式的用法。

## 11.2 ARRAYFORMULA 影片 [Link \(上\)](#) / [Link \(下\)](#)

ARRAYFORMULA 的用途是讓公式中非陣列的輸入轉為支援陣列的形式，說明如下：

語法	<b>ARRAYFORMULA (陣列公式)</b>
輸入	<b>陣列公式</b> ：可以是陣列或數學運算式，包含一或多個大小相同的儲存格範圍，也可以是傳回多個儲存格結果的函式。
輸出	<b>陣列公式</b> 使用 ARRAYFORMULA 後的輸出，為多欄多列的陣列。

簡單來說，ARRAYFORMULA 讓原本只會有單一儲存格輸出的函式變成可以使用陣列輸入 + ARRAYFORMULA 變成多欄多列的輸出，以下以四則運算搭配四個範例說明 ARRAYFORMULA 搭配函式的計算邏輯，如下圖：

	A	B	C	D	E	F	G
1	範例	結果				公式	
2	A		1	2	3	4	
3		+	5	6	7	8	
4		A(1)	6				=C2:F2+C3:F3
5		A(2)	6	8	10	12	=ARRAYFORMULA(C2:F2+C3:F3)
6		A(3)	6				=ARRAYFORMULA(C2:F2)+ARRAYFORMULA(C3:F3)
7	B		5	6			
8		*	7	8			
9		1	2	5	12		=ARRAYFORMULA(B9:C10*D7:E8)
10		3	4	21	32		
11	C	*	1	2	3	4	
12		5	5	10	15	20	
13		6	6	12	18	24	=ARRAYFORMULA(B12:B15*C11:F11)
14		7	7	14	21	28	
15		8	8	16	24	32	
16	D		1	2	3	4	
17			5	6	7	8	
18		5	5	10	15	20	
19		6	30	36	42	48	
20		7	#N/A	#N/A	#N/A	#N/A	
21		8	#N/A	#N/A	#N/A	#N/A	

- A. 因為 + 的前後只能輸入單一一個值，因此若直接使用 =C2:F2+C3:F3 只會輸出第一個值的相加，在此必須在外面包上一層 ARRAYFORMULA，如 A(2) 所示。但如果使用 A(3) 則無法輸出陣列相加，因為 + 並沒有用到 ARRAYFORMULA。
- B. 兩個相同大小的陣列運算時，其原理與相加相同，會輸出對應位置兩兩相乘的值。
- C. 前面有提及陣列在組合與運算時需要有相同的欄、列數，但**若陣列分別為「單欄多列」和「單列多欄」運算時將不受限制，會將其自動轉為多欄、多列重複的值**，如上圖會第一列會輸出  $5 \times 1 \sim 5 \times 4$  的值，以此類推。
- D. 上述 C. 的例外僅限於「單欄多列」和「單列多欄」運算時，因為在此有兩列多欄，所以後面兩列的值會跳出 #N/A。

了解陣列的基本運算方式後，接下來將使用 ARRAYFORMULA 完成第 1.3 節的五個範例，不同的地方是第 1.2 節需要在整欄輸入相同的公式，但使用 ARRAYFORMULA 後可以直接在第二列輸出整欄的內容：

#### E. 交易記錄檔 K 欄：計算每列貨品交易的金額

公式	=ARRAYFORMULA(I2:I*J2:J)
解說	原公式為 =I2*J2，只要使用 ARRAYFORMULA 後，會自動計算每一列的值，即 I2*J2、I3:J3，以此類推至最後一列。

#### F. 交易記錄檔 L 欄：計算每筆訂單所有貨品交易金額的總和

公式	=ARRAYFORMULA(SUMIF(A:A,A2:A,K:K))
解說	原公式為 =SUMIF(A:A,A2,K:K)，在此可將 <b>條件</b> 從單一值改成使用陣列表示，但須注意超過一個條件時，SUMIFS 並無法使用 ARRAYFORMULA 完成，需要使用其他替代方法完成，將於下一章介紹。

#### G. 會員資料檔 J 欄：計算每位會員的年齡

公式	=ARRAYFORMULA(DATEDIF(B2:B,TODAY(),"Y"))
解說	原公式為 =DATEDIF(B2,TODAY(),"Y")，在此將 <b>開始日期</b> 設為陣列，但 <b>結束日期</b> 還是單一值，公式會自動將 <b>結束日期</b> 也轉成列數與 <b>開始日期</b> 相同的陣列。

#### H. 會員資料檔 K 欄：將會員所屬的年齡區間，包括 30 歲以下、31~40 歲、41~50 歲、51~60 歲、61 歲以上

公式	=ARRAYFORMULA(IF(J2:J<=30,"30 歲以下",J2:J<=40,"31~40 歲",J2:J<=50,"41~50 歲",J2:J<=60,"51~60 歲",J2:J>60,"61 歲以上"))
解說	原公式為 =IFS(J2<=30,"30 歲以下",J2<=40,"31~40 歲",J2<=50,"41~50 歲",J2<=60,"51~60 歲",J2>60,"61 歲以上")，將所有 J2 都換成 J2:J 即可。

#### I. 會員資料檔 L 欄：使用交易記錄檔計算該會員在 2022~2023 年之交易總金額。

公式	=ARRAYFORMULA(SUMIF('交易記錄檔'!B:B,A2:A,'交易記錄檔'!K:K))
解說	原公式為 =SUMIF('交易記錄檔'!B:B,A2,'交易記錄檔'!K:K)，將 A2 換成 A2:A 即可。

由上面幾個例子可知，通常只要在原本只有單一輸入的儲存格改為多個輸入，再搭配 ARRAYFORMULA 即可完成整列的運算，避免公式被人工更改或刪除，而若要再更進一步避免被更改到第 2 列的資料，可以直接在表頭使用 ARRAYFORMULA 一次完成整欄的資料。以 A. 為例，可以直接在 K1 使用 =ARRAYFORMULA({"金額";I2:I\*J2:J})，善用陣列內容可以是單一值、儲存格範圍及公式輸出的特性。此外，ARRAYFORMULA 有快捷鍵 **⌘ / Ctrl + Shift + Enter**，可以快速在公式最外面加上一層 ARRAYFORMULA。

雖然 ARRAYFORMULA 很好用，但仍有部分函式不能使用 ARRAYFORMULA，主要原因有以下幾種：

- 函式輸入可以是多欄多列：例如 SUM、AND、OR 等。
- 函式輸出可以是多欄多列：例如 IMPORTRANGE、IMPORTXML、INDEX、OFFSET、UNIQUE、SORT、SORTN、FILTER 等。
- 其他：例如 SPARKLINE、GETPIVOTDATA、SUMIFS、AVERAGEIF 等。

雖然上述函式無法使用 ARRAYFORMULA 完成，但可以使用另一個方式 — LAMBDA + LAMBDA 輔助函式，將在下一章詳細說明。然而，要純熟的使用這兩種用法之前，必須先對其他的函式足夠熟悉，這兩個函式才有價值。

### 11.3 陣列欄列轉換 影片 [Link \(上\)](#) / [Link \(下\)](#)

本節將介紹與轉換陣列欄列相關的函式，包括轉置、轉換成特定欄 / 列數與合併等：

- TRANSPOSE：將資料轉置

語法	<b>TRANSPOSE (陣列或範圍)</b>
輸入	陣列或範圍：要轉置的資料。
輸出	將 陣列或範圍 轉置，例如 m 欄 n 列會變成 n 欄 m 列。

- FLATTEN：將二維資料轉為一維資料

語法	<b>FLATTEN (範圍_1, [範圍_2,...])</b>
輸入	範圍_n：要轉為一維的資料。
輸出	將每個 範圍_n 按列轉為單欄的資料，轉置順序為先左到右、再上到下。

- TOCOL / TOROW：將資料轉為單欄 / 列

語法	<b>TOCOL / TOROW (陣列或範圍, [忽略], [依欄掃描])</b>
輸入	<ul style="list-style-type: none"> <li>• 陣列或範圍：要轉置的資料。</li> <li>• 忽略：0~3，是否要忽略空白值及錯誤值，預設為 0，意義如下：           <ul style="list-style-type: none"> <li>○ 0：保留空白值及錯誤值。</li> <li>○ 1：只保留錯誤值，刪去空白值。</li> <li>○ 2：只保留空白值，刪去錯誤值</li> <li>○ 3：刪除空白值及錯誤值。</li> </ul> </li> <li>• 依欄掃描：TRUE / FALSE，TRUE 是先上到下、再左到右，FALSE 則是先左到右、再上到下，預設為 FALSE。</li> </ul>
輸出	將 陣列或範圍 轉為單欄 / 列。

- WRAPCOLS / WRAPROWS：將資料轉置為特定數量的列 / 欄

語法	<b>WRAPCOLS / WRAPROWS (範圍, 截斷數目, [填充值])</b>
輸入	<ul style="list-style-type: none"> <li>• 範圍：要轉置的資料，必須是單欄或單列。</li> <li>• 截斷數目：轉置時每列 / 欄要有幾筆資料。</li> <li>• 填充值：缺少的值要輸出的值，預設為 #N/A!。</li> </ul>
輸出	將資料轉置為每 截斷數目 筆資料一欄 / 列，兩者轉置的方向不一樣。

- HSTACK / VSTACK：將多個範圍水平 / 垂直合併

語法	<b>HSTACK / VSTACK (範圍_1, [範圍_2,...])</b>
輸入	範圍_n：要水平 / 垂直合併的範圍。
輸出	將所有 範圍_n 水平 / 垂直合併的結果。若水平合併但每個 範圍_n 列數不同，或是垂直合併但每個 範圍_n 欄數不同時，缺少的部分會出現 #N/A!。相當於支援欄、列數不同的陣列合併 (使用 {} 直接合併會出現 #REF!)。

接下來使用四個基本範例說明以上函式的用法，使用的資料如下，共有三個已命名範圍—性別、年齡區間、性別與年齡區間，三個範圍都不包含表頭。

性別	年齡區間
女	30 歲以下
男	31~40 歲
	41~50 歲
	51~60 歲
	61 歲以上

#### A. 將 性別與年齡區間 轉置

公式	=TRANSPOSE(性別與年齡區間)
解說	直接使用 TRANSPOSE 即可，如果在此使用整欄的話，轉置時會自動將(有內容的) 最後一列之後的空白列都刪除。

#### B. 將 性別與年齡區間 轉成一列

公式	<p><b>方法一：分開使用 性別 / 年齡區間 (僅適用於欄位數較少時)</b></p> <ul style="list-style-type: none"> <li>• =TRANSPOSE({性別;年齡區間})</li> <li>• =TRANSPOSE(FLATTEN(性別,年齡區間))</li> <li>• =TRANSPOSE(VSTACK(性別,年齡區間))</li> </ul> <p><b>方法二：使用 性別與年齡區間</b></p> <ul style="list-style-type: none"> <li>• =TRANSPOSE(FLATTEN(性別與年齡區間))</li> <li>• =TRANSPOSE(TOCOL(性別與年齡區間,TRUE,TRUE))</li> <li>• =TOROW(性別與年齡區間,TRUE,TRUE)</li> </ul>
解說	<p><b>方法一：分開使用 性別 / 年齡區間 (僅適用於欄位數較少時)</b></p> <p>可先將資料合併成一欄，再使用 TRANSPOSE 轉置成一列，合併成一欄的方式很多，在此分別使用陣列、FLATTEN、VSTACK 完成。</p> <p><b>方法二：使用 性別與年齡區間</b></p> <ul style="list-style-type: none"> <li>• 使用 FLATTEN(性別與年齡區間) 轉為一欄時空白的資料也轉置，因此可改為使用 TOCOL(性別與年齡區間,TRUE)，此外可將 依欄掃描 調整成 TRUE，就會先顯示所有性別再顯示所有年齡區間。</li> <li>• TOROW 等同於使用 TRANSPOSE + TOCOL。</li> </ul>

### C. 將年齡區間轉換為 3 欄 2 列的資料

公式	<ul style="list-style-type: none"> <li>=WRAPCOLS(年齡區間,2,"")</li> <li>=WRAPROWS(年齡區間,3,"")</li> </ul>																
解說	<p>兩者轉置的方向不同，輸出如下圖，可以想成 WRAPCOLS 是轉成幾欄，所以會先填滿一欄再換下一欄，WRAPROWS 則相反。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>=WRAPCOLS(年齡區間,2,"")</td> <td>30 歲以下</td> <td>41~50 歲</td> <td>61 歲以上</td> </tr> <tr> <td></td> <td>31~40 歲</td> <td>51~60 歲</td> <td></td> </tr> <tr> <td>=WRAPROWS(年齡區間,3,"")</td> <td>30 歲以下</td> <td>31~40 歲</td> <td>41~50 歲</td> </tr> <tr> <td></td> <td>51~60 歲</td> <td>61 歲以上</td> <td></td> </tr> </table>	=WRAPCOLS(年齡區間,2,"")	30 歲以下	41~50 歲	61 歲以上		31~40 歲	51~60 歲		=WRAPROWS(年齡區間,3,"")	30 歲以下	31~40 歲	41~50 歲		51~60 歲	61 歲以上	
=WRAPCOLS(年齡區間,2,"")	30 歲以下	41~50 歲	61 歲以上														
	31~40 歲	51~60 歲															
=WRAPROWS(年齡區間,3,"")	30 歲以下	31~40 歲	41~50 歲														
	51~60 歲	61 歲以上															

### D. 列出所有性別 × 年齡區間的組合，共 $2 \times 5$ 種

公式	=ARRAYFORMULA(SPLIT(TOCOL(性別&" "&TRANSPOSE(年齡區間)), " "))
解說	<p>本範例較複雜，以下分成多個步驟說明：</p> <ul style="list-style-type: none"> <li>在上一節有提到，若陣列分別為「單欄多列」和「單列多欄」運算時將不受限制，會將其自動轉為多欄、多列重複的值，而在此 性別 和 TRANSPOSE(年齡區間) 分別為「單欄多列」和「單列多欄」的陣列。</li> <li>使用 ARRAYFORMULA 及 &amp; 將兩個陣列合併之後，會成為多欄、多列的陣列，為了讓組合可以進一步分割成兩欄，可以在兩個陣列合併時插入一個無關的字串，作為之後的分割符號，在此為 "   "。</li> <li>使用 ARRAYFORMULA + SPLIT 將所有合併完成的陣列分割。&amp; 和 SPLIT 都需要使用 ARRAYFORMULA，可以統一包在最外層即可。</li> </ul>

## 11.4 陣列統計與篩選 [影片 Link](#)

在第 6.5~6.7、7.7 節有介紹到統計與篩選的各項相關函式，而在本節將進一步介紹四個相關的函式，用途與先前介紹的函式不同。

- FREQUENCY：計算資料的各級距出現的次數

語法	<b>FREQUENCY (資料, 級距)</b>
輸入	<ul style="list-style-type: none"> <li>● <b>資料</b>：要計算分布頻率的資料。</li> <li>● <b>級距</b>：一組數值型態的級距，會自動轉為一欄多列的遞增排序。</li> </ul>
輸出	<b>資料</b> 值在各 <b>級距</b> 的次數，共會有 <b>級距</b> + 1 列輸出。各 <b>級距</b> 為該級距的上限值，例如 =FREQUENCY( <b>資料</b> ,{30,40}) 輸出的三列分別代表 $x \leq 30$ 、 $30 < x \leq 40$ 、 $x > 40$ 。

- ARRAY\_CONSTRAIN：輸出陣列或範圍的前幾列、前幾欄

語法	<b>ARRAY_CONSTRAIN (輸入範圍, 列數, 欄數)</b>
輸入	<ul style="list-style-type: none"> <li>● <b>輸入範圍</b>：陣列或範圍。</li> <li>● <b>列數 / 欄數</b>：要輸出前幾列、前幾欄。</li> </ul>
輸出	<b>輸入範圍</b> 的前 <b>列數</b> 列、前 <b>欄數</b> 欄。

- CHOOSECOLS / CHOOSEROWS：在資料中選取特定的欄 / 列

語法	<b>CHOOSECOLS / CHOOSEROWS (陣列, [資料欄/列_1, 資料欄/列_2,...])</b>
輸入	<ul style="list-style-type: none"> <li>● <b>陣列</b>：要選取特定欄 / 列的原始資料。</li> <li>● <b>資料欄/列_n</b>：要選取第幾欄 / 列的資料。</li> </ul>
輸出	<b>陣列</b> 的第 <b>資料欄/列_n</b> 欄 / 列的資料，會依照 <b>資料欄/列_n</b> 的順序輸出。

以下使用第 1 章的會員資料檔並搭配兩個基本範例、兩個進階範例說明各函式的用法，而為了方便說明在此進行以下設定：

- 將所有交易記錄檔的各欄位使用已命名範圍命名成表頭名稱。
- 將 A:I 命名為「會員資料」、第 1 列命名為「會員資料表頭」。
- 將會員卡號的格式設定為自動 (數字)。

### A. 統計會員年齡分布情形，以 30、40、50、60 歲分成五個級距

公式	=FREQUENCY( <b>年齡</b> ,{30,40,50,60})
解說	{30,40,50,60} 可以隨意調動順序，FRQUENCY 會自動調整為遞增排序。

## B. 取得前 3 列會員資料

公式	<ul style="list-style-type: none"> <li>=ARRAY_CONSTRAIN(會員資料,3,COLUMNS(會員資料))</li> <li>=CHOSEROWS(會員資料,1,2,3)</li> </ul>
解說	<ul style="list-style-type: none"> <li>因為要輸出前 n 列，可使用 ARRAY_CONSTRAIN 完成，但需要同時指定列數與欄數，在此可使用 COLUMNS 便能輸出所有欄。</li> <li>若不是前 n 列的話，一般來說使用 CHOSEROWS 比較方便易懂。</li> </ul>

## C. 取得會員卡號 1902、2010、2181 的會員資料並加上標題列

公式	=CHOSEROWS(會員資料,1,ARRAYFORMULA(MATCH({1902,2010,2181},會員卡號,0)))
解說	<p>本範例較複雜，以下分成多個步驟說明：</p> <ul style="list-style-type: none"> <li>要取得特定列，可使用 CHOSEROWS，特定列包括標題列 (第一列)、三位會員所在的位置列，即 =CHOSEROWS(會員資料,1,...)。</li> <li>要取得三位會員所在的位置，可以使用 MATCH(...,會員卡號,0) 完成，可使用三次 MATCH 完成，或使用陣列加上 ARRAYFORMULA，在此使用後者，即 ARRAYFORMULA(MATCH({"1902","2010","2181"},會員卡號,0))。</li> <li>將 ARRAYFORMULA(...) 放至 CHOSEROWS 中即可。</li> </ul>

## D. 取得會員卡號 1902、2010、2181 的生日、年齡、近兩年消費總金額並加上標題列

公式	<ul style="list-style-type: none"> <li>=CHOSEROWS({生日,年齡,近兩年消費總金額},1,ARRAYFORMULA(MATCH({1902,2010,2181},會員卡號,0)))</li> <li>=CHOOSECOLS(CHOSEROWS(會員資料,1,ARRAYFORMULA(MATCH({1902,2010,2181},會員卡號,0))),ARRAYFORMULA(MATCH({"生日","年齡","近兩年消費總金額"},會員資料表頭,0)))</li> </ul>
解說	<p>本範例延續上例，變成只輸入特定欄位，有以下兩種做法：</p> <ul style="list-style-type: none"> <li>將 CHOSEROWS 的 陣列 改成指定的欄位即可，即 {生日,年齡,近兩年消費總金額}。</li> <li>在 CHOSEROWS 外面包一層 CHOOSECOLS，並一樣使用 MATCH 搜尋要輸出的欄位內容，即 ARRAYFORMULA(MATCH({"生日","年齡","近兩年消費總金額"},會員資料表頭,0))。</li> </ul> <p>兩種方式各有優劣，第一種方式適用於輸出固定的欄位或有已命名範圍時，第二種方式可以將陣列轉換成儲存格，便能輕鬆的轉換要輸出的欄位。</p>

經過本章的介紹後，感覺 Google Sheets 的函式越來越複雜是非常正常的，實務上的狀況往往更加複雜，而下一章我們將進一步介紹自定義函數，將常用的公式進一步包裝。

## Appendix 11. 函式 (六)：陣列與陣列函式

### 11.1 陣列 (Array) 基本概念

Excel 中可以使用陣列 {}，但若將陣列作為函式的輸入時，公式輸入完畢不能直接按 Enter，必須按 Option / Ctrl + Shift + Enter 才能正確輸出。

### 11.2 ARRAYFORMULA

在新版本的 Excel 中沒有 ARRAYFORMULA 此函式，本節所有的範例都不用使用 ARRAYFORMULA 即可完成，例如 A(1) 直接輸入 =C2:F2+C3:F3 便能輸出 C:F 欄各自的加總結果。

### 11.3 陣列欄列轉換

- TRANSPOSE：空值的儲存格轉置後會變成 0。
- FLATTEN：Excel 中無此函式

### 11.4 陣列統計與篩選

Excel 中沒有 ARRAY\_CONSTRAIN。

## Chapter 12. 函式 (七)：自定義函式 影片 [Link](#)

本章示範檔案：[Chapter 12. 函式 \(七\)：自定義函式](#)

本節將介紹 LET、已命名函式、LAMBDA 三個功能，主要是用來命名重複出現的變數與公式，增加變數與函式的可讀性。本章會用到許多前面介紹過的函式，因此若對前面幾章內容還不熟的同學可以在用到相關函式時回去複習其用法。

### 12.1 LET 影片 [Link](#)

LET 適用於需要將某一個項目在同一個公式中反覆使用時，函式說明如下：

語法	<b>LET(變數_1, 值運算式_1, [變數_2, 值運算式_2,...], 公式運算式)</b>
輸入	<ul style="list-style-type: none"><li><b>變數_n</b>：要用於之後 <b>值運算式_n</b> 或 <b>公式運算式</b> 的名稱，其名稱相關規定如下：<ul style="list-style-type: none"><li>不能是 A1 Notation，例如 A1、A2。</li><li>不得包含空格或特殊字元，點號、底線除外。</li><li>不得以數字開頭，例如 3times。</li><li>可以使用中文。</li></ul></li><li><b>值運算式_n</b>：要代表 <b>變數_n</b> 的公式，可以用先前已命名過的 <b>變數_1 ~ 變數_n-1</b>。</li><li><b>公式運算式</b>：要計算的公式，可使用前面所有的 <b>變數_n</b>。</li></ul>
輸出	<b>公式運算式</b> 搭配所有 <b>變數_n</b> 的計算結果。

上述的函式說明可能有點抽象，以下使用兩個先前章節的範例作為說明：

A. 同第 1.3 節的範例 C.& D.，根據會員的生日輸出會員的年齡區間包括 30 歲以下、31~40 歲、41~50 歲、51~60 歲、61 歲以上，如下圖

公式	C2 的公式為 =ARRAYFORMULA(LET(年齡,DATEDIF(B2:B,TODAY(),"Y"),IFS(年齡<=30,"30 歲以下",年齡<=40,"31~40 歲",年齡<=50,"41~50 歲",年齡<=60,"51~60 歲",年齡>60,"61 歲以上")))
解說	<ul style="list-style-type: none"><li>在第 1.3 節中，先使用 =DATEDIF(B2,TODAY(),"Y") 在 J2 計算出每一個會員的年齡，再使用 =IFS(J2&lt;=30,"30 歲以下",J2&lt;=40,"31~40 歲",J2&lt;=50,"41~50 歲",J2&lt;=60,"51~60 歲",J2&gt;60,"61 歲以上") 輸出年齡區間。</li><li>在此範例中，要直接輸出年齡區間，可以將前面 IFS 的 J2 全部 DATEDIF 的函式，但這樣公式會變得十分冗長，因此在此就可以使用 LET 將 DATEDIF(...) 設為 <b>變數</b> 年齡，並用於 <b>公式運算式</b> 中。</li><li>使用 ARRAYFORMULA 將 B2:B 一次使用 DATEDIF。</li></ul>

## B. 同第 9.3 節的範例 D. ptt看板範例 的人氣 (即 2607) 取代為 XXX

公式	=LET(人氣起始位置,FIND(" ",ptt看板範例)+1,人氣的長度,FIND(" ",ptt看板範例,人氣起始位置)-人氣起始位置,REPLACE(ptt看板範例,人氣起始位置,人氣的長度,"XXX"))
解說	<ul style="list-style-type: none"><li>在第 9.3 節中有說明到可以使用 REPLACE(ptt看板範例,人氣起始位置,人氣的長度,"XXX") 完成，在此可使用 LET 將其作為 公式運算式。</li><li>人氣起始位置：如第 9.3 節所述，可使用 FIND(" ",ptt看板範例)+1。</li><li>人氣的長度：如第 9.3 節所述，可使用 FIND(" ",ptt看板範例,FIND(" ",ptt看板範例)+1)-FIND(" ",ptt看板範例)-1，但在此可以使用 人氣起始位置 取代 FIND(" ",ptt看板範例)+1，因此可寫成 FIND(" ",ptt看板範例,人氣起始位置)-人氣起始位置。</li><li>後面命名的變數可以用到前面命名的變數，所以在此不能將兩個變數對調，因為 人氣的長度 會用到 人氣起始位置。</li></ul>

由以上範例可知，是否使用 LET 對於公式本身的輸出沒有影響，但若公式需要重複用到某個計算項目時，使用 LET 可以讓公式看起來更直觀易懂！

## 12.2 已命名函式 影片 [Link](#)

以往自定義函式需要使用 Google Apps Script 才能完成，而現在直接點選「資料 → 已命名函式」，可以點選「瞭解詳情」查看[官方說明](#)，或是點選「查看範例」直接看例子學習，了解用法後可以點選「新增函式」進入「新增已命名函式」視窗建立自己的函式，或是點選「匯入函式」匯入其他 Google Sheets 中已命名的函式，如下圖：



在「新增已命名函式」視窗中，有以下幾個元素：

- 已命名函式詳細資料：包括函式名稱、函式說明，函式說明會在輸入函式時出現。
- 引數預留位置：已命名函式有哪些 Input，可以使用中文。
- 公式定義：此函式的計算方式，可以使用其他函式及引數計算。

輸入上述內容後，點選「繼續」會出現函式預覽及其他資訊，其他資訊是用來輸入各項引數說明與引數範例。

瞭解介面的各項內容後，接下來將舉四個例子說明已命名函式的用法：

### A. AGE：輸入生日，輸出其年齡

結果

=AGE()

AGE(生日)

已命名函式

範例

AGE(1966/01/01)

說明

根據輸入的日期，計算其年齡。

公式定義

=DATEDIF(生日,TODAY(),"Y")

生日

出生日期。

詳細資料	<ul style="list-style-type: none"> <li>函式說明：根據輸入的日期，計算其年齡。</li> <li>引數預留位置：<b>生日</b></li> <li>公式定義：<code>=DATEDIF(生日, TODAY(), "Y")</code></li> </ul>
函式預覽	引數說明 / 範例： <b>生日</b> ：出生日期 / 1966/01/01

## B. CONTAIN：檢查某個值是否有落在某範圍間，輸出 TRUE / FALSE

結果	
詳細資料	<ul style="list-style-type: none"> <li>函式說明：特定值是否落在某範圍內。</li> <li>引數預留位置：<b>搜尋準則</b>、<b>範圍</b></li> <li>公式定義：<code>=ISNUMBER(MATCH(搜尋準則,範圍,0))</code></li> </ul>
函式預覽	<ul style="list-style-type: none"> <li>引數說明 / 範例：<b>搜尋準則</b>：要找的項目 / A1</li> <li>引數說明 / 範例：<b>範圍</b>：要在哪裡找 / B1:B20</li> </ul>

## C. CONTAIN2：優化 B. 的功能，使其能支援多欄多列的搜尋

在前一個範例中，因為 MATCH 的 **範圍** 必須是單欄 / 單列的陣列或範圍，因此 CONTAIN 的 **範圍** 也會受到此限制，若要讓 CONTAIN 的 **範圍** 可以使用多欄 / 列，可以將公式改成以下任一種方式：

- `=ISNUMBER(MATCH(搜尋準則,TOCOL(範圍),0))`：將 **範圍** 使用 TOCOL (TOROW / TRANSPOSE 皆可) 轉為單欄 / 列即可。
- `=COUNTIF(範圍,搜尋準則)>0`：COUNTIF 的 **條件範圍** 不受單欄 / 單列的限制。

使用此函式後，若輸入多欄多列的 **範圍** 便會得到正確的結果，由 B.、C. 兩個例子可知，在設計已命名函式時，要考慮各種可能發生的情況，並對 Google Sheets 本身的函式概念夠熟悉，才能應付各種可能遇到的情況。

## D. MY\_FV：使用四則運算計算出結果年金終值

結果	<p><b>結果</b></p> <p>=MY_FV(利率, 期數, 每期投入金額, 現值, 到期年金)</p> <p>範例 MY_FV(10%, 3, -100, -1000, 0)</p> <p>說明 根據定期定額支付金額和固定利率，計算年金投資的未來值。</p> <p>公式定義 <math>=-(\text{現值} * (1 + \text{利率})^{\text{期數}} + \text{每期投入金額} * \text{IF}(\text{到期年金} = 0, 1, (1 + \text{利率})) * (1 - (1 + \text{利率})^{\text{期數}}) / -\text{利率})</math></p> <p><b>利率</b> 年金的有效利率</p> <p><b>期數</b> 付款期數</p> <p><b>每期投入金額</b> 每期需要投入的金額</p> <p><b>現值</b> 年金的現值</p> <p><b>到期年金</b> 是否為到期年金，0 = 到期年金、1 = 普通年金</p>
詳細資料	<ul style="list-style-type: none"> <li>函式說明：根據定期定額支付金額和固定利率，計算年金的未來值。</li> <li>引數預留位置：<b>利率</b>、<b>期數</b>、<b>每期投入金額</b>、<b>現值</b>、<b>到期年金</b></li> <li>公式定義：用等比級數和公式計算，在此為 <math>=-(\text{現值} * (1 + \text{利率})^{\text{期數}} + \text{每期投入金額} * \text{IF}(\text{到期年金} = 0, 1, (1 + \text{利率})) * (1 - (1 + \text{利率})^{\text{期數}}) / -\text{利率})</math></li> </ul>
函式預覽	<ul style="list-style-type: none"> <li>引數說明 / 範例：<b>利率</b>：年金的有效利率 / 10%</li> <li>引數說明 / 範例：<b>期數</b>：付款期數 / 3</li> <li>引數說明 / 範例：<b>每期投入金額</b>：每期需要投入的金額 / -100</li> <li>引數說明 / 範例：<b>現值</b>：年金的現值 / -1000</li> <li>引數說明 / 範例：<b>到期年金</b>：是否為到期年金，0 = 到期年金、1 = 普通年金 / 0</li> </ul>

使用此函式後，其用途和 FV 完全相同，而第 10.2 節的 PV、PMT 也使用四則運算計算出與使用函式相同的結果，各位同學可以自行嘗試完成！

以上函式完成後便可在 Google Sheets 中進一步使用，也可以使用 ARRAYFORMULA (如果自定義函式內的公式可使用 ARRAYFORMULA 完成) 如下圖：

範例	公式	結果
A	=AGE(DATE(2003,12,4))	20
B	=CONTAIN(2944,會員卡號)	TRUE
	=CONTAIN(2944,會員資料)	FALSE
C	=CONTAIN_2(2944,會員資料)	TRUE
D	=FV(10%,12,-1000,-20000,0)	84,152.9
	=MY_FV(10%,12,-1000,-20000,0)	84,152.9

了解已命名函式的用法後，若之後有需要頻繁使用的函式，可以直接建立屬於自己的函式並廣泛運用，絕對會是非常實用的技能，而引數的命名方式與其他相關規定在此省略，但[官方說明](#)中皆有詳細的說明。

## 12.3 LAMBDA 影片 [Link](#)

自定義函式除了使用上一節所述的已命名函式之外，Google Sheets 亦提供另一種方式—LAMBDA 函式，使用方式如下：

語法	<b>LAMBDA</b> ([變數_1, 變數_2,...], 公式)
輸入	<ul style="list-style-type: none"> <li>變數_n：要用於 公式 中的變數名稱，其名稱規定與 LET 的 變數 相同。</li> <li>公式：要計算的公式，可以使用 變數_n 及任何前面介紹過的函式完成。</li> </ul>
輸出	本身輸出會是 #N/A，是因為 LAMBDA 的後面需有實際值的呼叫，就會有該實際值使用 LAMBDA 計算後的輸出。

由上說明可知，LAMBDA 本身不會有任何輸出，而是會形成一個可以使用的公式，後面要有實際的數字套入公式中，才會有相應的輸出。若要在 LAMBDA 使用實際值，只要使用括號即可，若有多個 變數，則使用逗號將各變數分隔，如下圖：

範例	公式	結果	說明
A	=LAMBDA(x,2*x)(5)	10	$2 \times 5 = 10$
B	=LAMBDA(x,y,x*y)(5,6)	30	$5 \times 6 = 30$
C	=LAMBDA(消費金額,IFS(消費金額<5000,"一般會員",消費金額<50000,"VIP",消費金額>=50000,"VVIP"))(1000)	一般會員	輸入消費金額，若達 5,000 就成為 VIP、達 50,000 就成為 VVIP
	=LAMBDA(消費金額,IFS(消費金額<5000,"一般會員",消費金額<50000,"VIP",消費金額>=50000,"VVIP"))(10000)	VIP	
	=LAMBDA(消費金額,IFS(消費金額<5000,"一般會員",消費金額<50000,"VIP",消費金額>=50000,"VVIP"))(100000)	VVIP	
D	=LAMBDA(年,月,日,DATE(IF(年<=115,年+1911,年),月,日))(2003,8,10)	2003/8/10	若輸入的年份小於 115 年，將其視為民國年，並轉成西元年
	=LAMBDA(年,月,日,DATE(IF(年<=115,年+1911,年),月,日))(92,8,10)	2003/8/10	

基本上 LAMBDA 的用法和上一節的已命名函式相同，差別在於 LAMBDA 創造的函式並沒有一個真正的名稱，而已命名函式可以對創造的函式有更詳細的說明。目前 LAMBDA 看起來有點不實用，是因為 LAMBDA 本身只能使用在單一儲存格上，所以看起來有一點多餘。而下一節我們將介紹六個 LAMBDA 輔助函式，配合 LAMBDA 使用可以執行更多複雜的操作。

## 12.4 LAMBDA 輔助函式

影片 [Link \(上\)](#) / [Link \(下\)](#)

LAMBDA 輔助函式共有六個，專門與 LAMBDA 搭配使用，分別如下：

- MAP：同時對多個範圍使用 LAMBDA 函式

語法	<a href="#"><b>MAP</b> (陣列_1, [陣列_2,...], <b>LAMBDA</b>)</a>
輸入	<ul style="list-style-type: none"> <li>• <b>陣列_n</b>：要使用 <b>LAMBDA</b> 的範圍，要和 <b>LAMBDA</b> 的輸入數量相同，且各 <b>陣列_n</b> 的範圍大小要相同，例如都是 2 欄 × 3 列等。</li> <li>• <b>LAMBDA</b>：LAMBDA 函式。</li> </ul>
輸出	各個 <b>陣列_n</b> 使用 <b>LAMBDA</b> 的結果，輸出範圍同 <b>陣列_n</b> 的範圍大小。

- BYROW / BYCOL：以列 / 欄為單位的使用 LAMBDA 函式

語法	<a href="#"><b>BYROW</b> / <b>BYCOL</b> (陣列或範圍, <b>LAMBDA</b>)</a>
輸入	<ul style="list-style-type: none"> <li>• <b>陣列或範圍</b>：要使用 <b>LAMBDA</b> 的陣列或範圍。</li> <li>• <b>LAMBDA</b>：有一個 <b>變數</b> 的 LAMBDA 函式。</li> </ul>
輸出	<ul style="list-style-type: none"> <li>• BYROW：逐列使用 <b>LAMBDA</b>，輸出範圍同 <b>陣列或範圍</b> 的列數。</li> <li>• BYCOL：逐欄使用 <b>LAMBDA</b>，輸出範圍同 <b>陣列或範圍</b> 的欄數。</li> </ul>

- SCAN / REDUCE：以迴圈的方式執行 LAMBDA 函式

語法	<a href="#"><b>SCAN</b> / <b>REDUCE</b> (初始值, 陣列或範圍, <b>LAMBDA</b>)</a>
輸入	<ul style="list-style-type: none"> <li>• <b>初始值</b>：第一次執行 <b>LAMBDA</b> 的初始值。</li> <li>• <b>陣列或範圍</b>：要使用 <b>LAMBDA</b> 的陣列或範圍。</li> <li>• <b>LAMBDA</b>：有兩個 <b>變數</b> 的 LAMBDA 函式。</li> </ul>
輸出	<p>將每次 LAMBDA 的輸出作為下一次的 <b>變數_1</b>，假設 <b>陣列或範圍</b> 共有 n 個值，分別為 <b>輸入值_n</b> 其執行步驟如下：</p> <ul style="list-style-type: none"> <li>• Step 1：LAMBDA(<b>初始值</b>, <b>輸入值_1</b>,...), 輸出 <b>中繼值_1</b>。</li> <li>• Step 2~n-1：計算 LAMBDA(<b>中繼值_x</b>, <b>輸入值_x+1</b>,...), 輸出 <b>中繼值_x+1</b>。</li> <li>• Step n：計算 LAMBDA(<b>中繼值_n-1</b>, <b>輸入值_n</b>,...), 輸出 <b>中繼值_n</b>。</li> </ul> <p>SCAN / REDUCE 的輸出分別如下：</p> <ul style="list-style-type: none"> <li>• SCAN：<b>中繼值_1</b> ~ <b>中繼值_n</b>，輸出範圍同 <b>陣列或範圍</b>。</li> <li>• REDUCE：只輸出 <b>中繼值_n</b>。</li> </ul>

- MAKEARRAY：讓 LAMBDA 輸出多列、多欄的陣列

語法	<b>MAKEARRAY (列數, 欄數, LAMBDA)</b>						
輸入	<ul style="list-style-type: none"> <li>• <b>列數 / 欄數</b>：要輸出的列 / 欄數。</li> <li>• <b>LAMBDA</b>：有兩個 <b>變數</b> 的 LAMBDA 函式。</li> </ul>						
輸出	<p>使用 <b>LAMBDA</b>(當前列,當前欄,公式) 的結果，輸出範圍共有 <b>列數</b> 列、<b>欄數</b> 欄，如下圖，左圖直接輸出當前是第幾列、右圖直接輸出當前是第幾欄。</p> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>=MAKEARRAY(2,3,LAMBDA(x,y,x))</td> </tr> <tr> <td>1      1      1</td> </tr> <tr> <td>2      2      2</td> </tr> </table> <table border="1" style="display: inline-table; vertical-align: middle;"> <tr> <td>=MAKEARRAY(2,3,LAMBDA(x,y,y))</td> </tr> <tr> <td>1      2      3</td> </tr> <tr> <td>1      2      3</td> </tr> </table>	=MAKEARRAY(2,3,LAMBDA(x,y,x))	1      1      1	2      2      2	=MAKEARRAY(2,3,LAMBDA(x,y,y))	1      2      3	1      2      3
=MAKEARRAY(2,3,LAMBDA(x,y,x))							
1      1      1							
2      2      2							
=MAKEARRAY(2,3,LAMBDA(x,y,y))							
1      2      3							
1      2      3							

以上函式缺乏例子有點難理解，接下來分別使用四個範例說明各函式的適用時機：

#### C. 根據會員的生日輸出會員的年齡，如下圖：

	A	B	C
1	會員卡號	生日	年齡
2	1771	1949/12/26	74
3	1777	1988/7/23	35
4	1852	1978/2/4	46
5	1902	1988/7/24	35
6	2010	1976/10/29	47
7	2181	1981/7/5	42
8	2193	1990/5/7	34
9	2944	1983/9/25	40

公式	C2 的公式為 =MAP(B2:B,LAMBDA(生日,AGE(生日)))
解說	<ul style="list-style-type: none"> <li>• 要將整欄使用相同的公式，因此適合使用 MAP 完成。</li> <li>• 在此使用 =ARRAYFORMULA(AGE(B2:B)) 其實比 MAP 直觀。</li> </ul>

#### D. 計算每位客戶一到六月的金額總計，及每個月所有客戶的金額總計，如下圖：

	A	B	C	D	E	F	G	H
1	客戶	一月	二月	三月	四月	五月	六月	總計
2	A Company	0	10,143	20,224	0	0	31,255	61,622
3	B Company	744	10,640	19,782	17,911	0	14,269	63,346
4	C Company	0	0	18,542	25,064	26,765	351	70,722
5	D Company	0	0	22,462	50,638	10,160	1,440	84,700
6	E Company	31,764	0	0	12,417	13,061	4,784	62,026
7	F Company	36,827	11,655	1,876	54,331	53,849	9,815	168,353
8	G Company	2,428	55,489	34,057	24,808	777	8,710	126,269
9	H Company	24,920	0	28,945	0	2	19,298	73,165
10	I Company	30,784	17,178	10,993	4,472	1,372	2,888	67,687
11	J Company	26,754	0	0	13,392	21,270	0	61,416
12	總計	154,221	105,105	156,881	203,033	127,256	92,810	839,306

公式	<ul style="list-style-type: none"> <li>H2 的公式為 =BYROW(B2:G11,LAMBDA(值,SUM(值)))</li> <li>B12 的公式為 =BYCOL(B2:H11,LAMBDA(值,SUM(值)))</li> </ul>
解說	<p>以下以 H2 為例，B12 的概念相同只是換成逐欄加總</p> <ul style="list-style-type: none"> <li>在 H2:H11 中，要將每一列 B:G 欄的值加總，但 ARRAYFORMULA+SUM 無法針對每一列個別加總，因此需要用到 BYROW + LAMBDA(SUM) 完成。</li> <li>在 LAMBDA 中，使用 SUM 將所有值加總，再套上一層 BYROW 之後，就會將 B2:G2、B3:G3、...、B11:G11 依序套入 BYROW 中，輸出每一列加總的結果。</li> </ul>

E. 根據每天的安打、打數，計算出截至當天的打擊率 (累積安打 / 累積打數)，如下圖：

	A	B	C	D
1	日期	安打	打數	打擊率
2	2024/04/19	1	3	0.333
3	2024/04/21	1	5	0.250
4	2024/04/22	0	3	0.182
5	2024/04/25	0	1	0.167
6	2024/04/26	2	3	0.267
7	2024/04/27	1	4	0.263
8	2024/04/28	0	4	0.217
9	2024/05/01	0	0	0.217
10	2024/05/03	3	3	0.308

公式	=MAP(SCAN(0,B2:B,LAMBDA(累積安打,今日安打,累積安打+今日安打)), SCAN(0,C2:C,LAMBDA(累積打數,今日打數,累積打數+今日打數)), LAMBDA(累積安打,累積打數,累積安打/累積打數))
解說	<p>本範例較複雜，以下分成多個步驟說明：</p> <ul style="list-style-type: none"> <li>SCAN 可以計算累積安打及打數，但 SCAN 只能有一個陣列或範圍的輸入，因此只能分別算出累積安打、累積打數，並沒有辦法一次完成。在此會需要使用 MAP 計算兩者相除的結果。即 =MAP([累積安打], [累積打數], LAMBDA(累積安打,累積打數,累積安打/累積打數))。</li> <li>累積安打的計算方式為 SCAN(0,B2:B,LAMBDA(累積安打,今日安打,累積安打+今日安打))，初始值為 0。</li> <li>累積打數與累積安打的算法相同，在此省略。</li> </ul>

#### F. 製作一張乘法表，計算 11~19 的兩兩相乘值，如下圖：

	A	B	C	D	E	F	G	H	I	J
1	乘法表	11	12	13	14	15	16	17	18	19
2	11	121	132	143	154	165	176	187	198	209
3	12	132	144	156	168	180	192	204	216	228
4	13	143	156	169	182	195	208	221	234	247
5	14	154	168	182	196	210	224	238	252	266
6	15	165	180	195	210	225	240	255	270	285
7	16	176	192	208	224	240	256	272	288	304
8	17	187	204	221	238	255	272	289	306	323
9	18	198	216	234	252	270	288	306	324	342
10	19	209	228	247	266	285	304	323	342	361

公式	=MAKEARRAY(9,9,LAMBDA(列,欄,(列+10)*(欄+10)))
解說	<ul style="list-style-type: none"> <li>在此要相乘的數字是遞增的，因此可以使用 MAKEARRAY 完成，11~19 共九個數字，因此 列數、欄數 都設為 9。</li> <li>在 LAMBDA 中列、欄會分別是 1~9，因此要計算 11~19 相乘需要分別加上 10，即 (列+10)*(欄+10)。</li> <li>在此也可以使用 =ARRAYFORMULA(A2:A10*B1:J1) 完成，因為 A2:A10、B1:J1 分別是單欄、單列的值，比 MAKEARRAY 更方便且直觀。若不要使用到儲存格的數字，則可以使用 =ARRAYFORMULA(SEQUENCE(9,1,11,1)*SEQUENCE(1,9,11,1)) 完成。</li> </ul>

在上述的範例中，我將所有可以使用 ARRAYFORMULA 代替的範例都補上 ARRAYFORMULA 的做法，是因為作者對 LAMBDA 與 ARRAYFORMULA 兩者的取捨標準為如果 ARRAYFORMULA 無法完成，就使用 LAMBDA (例如 SUMIFS)，雖然 LAMBDA 函式的用途與功能更多元，但 ARRAYFORMULA 的做法更為直觀，實務上大家可以依據自己的使用習慣決定使用哪一個函式。

本章完成檔案：[Chapter 12. 函式\(七\)：自定義函式完成版](#)

## Appendix 12. 函式 (七)：自定義函式

### **12.2 已命名函式**

Excel 中無此用法。

### **12.3 LAMBDA**

過舊的版本無此功能。

### **12.4 LAMBDA 輔助函式**

過舊的版本無此功能。

## Chapter 13. 函式(八)：QUERY 影片 [Link](#)

本章示範檔案：[!\[\]\(e164acaa35fd9aaecde1e4c613f06f35\_img.jpg\) Chapter 13. 函式\(八\)：QUERY](#)

QUERY 是用 [Google 視覺化 API 搜尋語言](#) 完成的，可以輕鬆的針對結構化資料進行各種分析。本章將由淺入深的介紹 QUERY 的各項語法，並在最後一節的進階應用示範如何處理複雜的資料，讓大家之後使用 Google Sheets 進行資料分析，或是之後學習使用 SQL 處理更大量的資料時更得心應手！另外 QUERY 是 Google Sheets 獨有的函式，Excel 目前沒有此功能，將不特別使用 Appendix 說明之。

### 13.1 QUERY 基本用法 影片 [Link](#)

Google 視覺化 API 搜尋語言是一套由 Google 開發的語言，其語法和 SQL 非常類似，因此若之前有使用過 SQL 的話應該可以非常得心應手學習此章的內容，而實務上 QUERY 也相較於其他函式直覺不少，而且可以只使用一個函式就完成複雜的操作與計算。在了解 Google 視覺化 API 搜尋語言之前，先來看看 QUERY 的用法：

語法	<a href="#">QUERY (資料, 查詢, [標題])</a>
輸入	<ul style="list-style-type: none"><li><b>資料</b>：資料庫的範圍，每一欄的資料型態要統一，若單欄包含多種資料型態，則會視該欄位大多數資料的類型決定，其他少數的資料會被視為空值。</li><li><b>查詢</b>：要執行的搜尋，以字串表示。</li><li><b>標題</b>：<b>資料</b> 的前幾列是標題，預設為 -1，將視資料的內容自動判斷。</li></ul>
輸出	<b>資料</b> 依據 <b>查詢</b> 整理後的輸出，會占用多格儲存格 (所以是陣列公式)。

接下來各節將會介紹各項 **查詢** 的語法，會使用第 1 章的會員資料檔和交易記錄檔進行各項示範，兩檔案已分別被命名為會員資料 (A:I)、交易記錄 (A:J)。本節將先介紹 select、匯總函式及四則運算三種語法，使用方法如下：

語法	說明
select	<b>select 欄位 [, 欄位...]</b> 放在最開頭，後面接要選取的欄位，選取方式有以下兩種： <ul style="list-style-type: none"><li>選取所有欄位：使用 *，例如 <code>select *</code>。</li><li>使用欄位編號：例如 A、B、C 等，若要選擇多個欄位則使用半形逗號分隔，例如：<code>select A,B,C</code>。</li><li>使用 Coln：用於沒有欄位編號時，即 QUERY 的 <b>資料</b> 不是直接來自儲存格範圍，例如：<code>select Col1,Col2</code>。</li></ul>
匯總函式	<b>sum / avg / count / min / max (欄位)</b> 可以在 select 選擇的欄位中使用上述的匯總函式，但因為匯總後會變成單一值，因此不能將特定欄位與匯總欄位同時選取，例如 <code>sum(A)</code> 、 <code>avg(C)</code> 。
四則運算	執行跨欄位內容計算，僅適用於數值欄位，例如 <code>select A+B</code> 。

以下使用四個範例說明上述語法的使用方式：

**A. 選取所有會員資料**

公式	=QUERY(會員資料,"select *")
解說	直接使用 "select *" 即可。

**B. 計算交易記錄 J 欄的總和**

公式	=QUERY(交易記錄,"select sum(J)")
解說	輸出會出現 sum 售價，後續章節會說明如何更改或移除輸出的表頭名稱。

**C. 選取交易記錄的 H、I、J、I × J 四欄**

公式	=QUERY(交易記錄,"select H,I,J,I*J")
解說	四則運算可以與一般的欄位一起輸出。

**D. 計算交易記錄的 I × J 欄的總和**

公式	<ul style="list-style-type: none"><li>方法一 : =SUM(QUERY(交易記錄,"select I*J"))</li><li>方法二 : =QUERY(QUERY(交易記錄, "select I*J"), "select sum(Col1)")</li></ul>
解說	因為匯總函式不能和四則運算一起使用，在此可用上述兩種方式解決： <ul style="list-style-type: none"><li>方法一：使用 QUERY 計算各列 I × J 欄的值，再使用 SUM 將 QUERY 輸出的陣列加總，會輸出 83,694。</li><li>方法二：使用兩層 QUERY 完成，內層的 QUERY 先各列 I × J 欄的值，外層 QUERY 再將內層計算的結果加總，但因為內層輸出的結果並沒有欄位名稱，因此使用 Col1 代替，會輸出第一列表頭為 sum product( 數量售價 )，第二列為 83,694。</li></ul>

在這一節介紹了如何選取欄位，並對欄位進行簡單的計算後，下一節將介紹如何對資料進行篩選，再執行下一步的計算與欄位選取。

## 13.2 資料條件篩選 影片 [Link](#)

本節將介紹如何使用 where 進行資料篩選，及使用 and / or 使用多個篩選條件，並說明在公式中如何搭配儲存格做使用。

語法	說明
where	<p><b>where 條件</b>          上述的條件通常包括以下幾種內容：</p> <ul style="list-style-type: none"> <li>比較運算子：包括 <code>&lt;=</code>、<code>&lt;</code>、<code>&gt;</code>、<code>&gt;=</code>、<code>=</code>、<code>!=</code> (不等於)、<code>&lt;&gt;</code> (不等於)，例如 <code>where A &gt;= 500</code>。</li> <li>選取空白 / 非空白：<code>is null</code>、<code>is not null</code>，例如 <code>where D is not null</code>。          此外，若條件中有字串的話，前後要使用單引號 '' 包住字串的內容，例如 <code>where B &lt;&gt; 'ABC'</code>。</li> </ul>
and / or	<p><b>where 條件1 and / or 條件2</b>          有多個條件，取多個條件的交集或聯集，可使用括號定義優先順序。例如 <code>where (A &gt;= 500) and ((B &lt;&gt; 'ABC') or (C = ""))</code>。</p>

以下使用兩個範例說明上述語法的使用方式：

- A. (1) 選取子女人數 1 名以上的會員資料，輸出卡號、職業、子女人數  
 (2) 選取職業是「商」的會員資料，輸出卡號、職業、子女人數  
 (3) 選取子女人數 1 名以上且職業是「商」的會員資料，輸出卡號、職業、子女人數

公式	(1) =QUERY(會員資料,"select A,E,H where H >= 1") (2) =QUERY(會員資料,"select A,E,H where E = '商'") (3) =QUERY(會員資料,"select A,E,H where (H >= 1) and (E = '商')")
解說	如果有兩個以上的條件，建議使用括號增加公式的可讀性。

- B. 計算交易資料檔中單筆交易總額 ( $I \times J$ ) 在 300~500 元的筆數

公式	=QUERY(交易記錄, "select count(A) where (I*J >= 300) and (I*J <= 500)")
解說	300~500 元需要使用兩個條件，與 SUMIFS 相同。

### 13.3 日期與時間處理 [影片 Link](#)

在選取欄位時，可能會只想取得日期欄位的特定部分，也有可能只想選取特定日期或時間以後的資料，而本節將介紹與上述操作相關的語法，在官方文件中分成純量函式、文字表示兩部分，如下：

語法	說明
取得時間 特定部分	<ul style="list-style-type: none"> <li><b>year / quarter / month / day / dayOfWeek</b> (日期欄位)</li> <li><b>hour / minute / second / millisecond</b> (時間欄位)</li> <li><b>now()</b></li> </ul> 會輸出日期與時間的特定部分，可以放在選擇的欄位或條件中，例如 <code>select hour(A)</code> 、 <code>select B where dayOfWeek(B) &lt;&gt; 2</code> 特殊的輸出如下： <ul style="list-style-type: none"> <li><b>dayOfWeek</b>：禮拜日是 1、禮拜一是 2，以此類推。</li> <li><b>month</b>：一月是 0、二月是 1，以此類推。</li> </ul>
<code>dateDiff</code>	<b><code>dateDiff(結束日, 起始日)</code></b> 會輸出結束日與起始日之間差距的天數，可以用在 <code>select</code> 或是 <code>where</code> 中，例如 <code>select dateDiff(A,B)</code> 、 <code>where dateDiff(A,B) &gt; 3</code>
條件中 包含時間	<ul style="list-style-type: none"> <li><b>date 'yyyy-MM-dd'</b></li> <li><b>timeofday 'HH:mm:ss[.sss]'</b></li> <li><b>datetime 'yyyy-MM-dd HH:mm:ss[.sss]'</b></li> </ul> 上述語法用於欲在條件中加入日期、時間、日期與期間時，此外必須在前面加入 <b>date</b> / <b>timeofday</b> / <b>datetime</b> ，才能將其轉為日期或時間判讀，例如 <code>where A &gt;= date '2022-01-01'</code> 、 <code>where A &gt;= timeofday '08:00:00'</code> 。
<code>toDate</code>	<b><code>toDate(日期 / 日期時間 / 數字)</code></b> 將資料轉成日期格式，輸入有以下形式 <ul style="list-style-type: none"> <li><b>日期</b>：輸出和輸入相同的值，例如 <code>select toDate(date '2021-12-12')</code>。</li> <li><b>日期時間</b>：會輸出日期部分，例如 <code>select toDate(datetime '2021-12-12 12:00:00')</code>。</li> <li><b>數字</b>：會輸出與 1970-01-01 00:00:00 相差的毫秒數，例如 <code>toDate(1650000000000)</code> 會輸出 2022-04-15。</li> </ul>

以下使用兩個範例說明上述語法的使用方式：

#### A. 取得所有 1970/1/1 以後出生的會員的卡號、生日、性別

公式	<ul style="list-style-type: none"> <li><code>=QUERY(會員資料, "select A,B,C where year(B) &gt;= 1970")</code></li> <li><code>=QUERY(會員資料, "select A,B,C where B &gt;= date '1970-01-01'")</code></li> </ul>
解說	可以使用 <code>year(B) &gt;= 1970</code> 篩選 B 欄的年份大於等於 1970 的資料，或是直接使用 <code>B &gt;= date '1970-01-01'</code> 篩選大於 1970/1/1 的會員。

## B. 輸出會員卡號 1777 最後一次消費的日期及距今天數

公式	=QUERY(交易記錄,"select max(D),dateDiff(now(),max(D)) where (B = '1777')")
解說	可以使用 <code>max(D)</code> 得到最後一筆消費的日期，並使用 <code>dateDiff(now(),max(D))</code> 計算最後一筆消費日期與今天的差距。

了解 QUERY 中與日期相關的用法後，下一節將介紹與字串處理相關的用法！

## 13.4 字串處理 [影片 Link](#)

在 QUERY 中有一些與字串處理相關之語法，可以將字串進行大小寫轉換及取得字串符合某個條件的資料，如下：

語法	說明
大小寫	<b>upper / lower (字串)</b> 將字串轉為全大寫 / 小寫，例如 <code>select upper(A)</code> 、 <code>where lower(B)='x'</code> 。
包含 / 開頭 / 結尾	<b>contains / starts with / ends with '字串'</b> 用在 where 後面作為篩選依據，可以篩選出包含 / 開頭 / 結尾為指定字串的資料，例如 <code>where A contains 'Hi'</code> 、 <code>where B starts with 's'</code> 。
matches	<b>matches '正規表示式字串'</b> 用在 where 後面作為篩選依據，可以篩選出符合正規表示式的資料，例如 <code>where A matches '.*ing'</code> (結尾是 ing 的資料)。
like	<b>like '條件字串'</b> 用在 where 後面作為篩選依據，可以篩選出符合條件字串的資料，其中條件字串包括以下兩項： <ul style="list-style-type: none"> <li>• <code>%</code>：0 個或多個字元，例如 <code>where A like '%ing'</code> (結尾是 ing)。</li> <li>• <code>_</code>：1 個字元，例如 <code>where A like 'h_t'</code> (內容是 h + 任意字元 + t)。</li> </ul>

以下使用兩個範例說明上述語法的使用方式：

### A. 貨名是「台灣」開頭的總銷售額

公式	<ul style="list-style-type: none"> <li>• <code>=QUERY(交易記錄,"select sum(K) where H starts with '台灣'")</code></li> <li>• <code>=QUERY(交易記錄,"select sum(K) where H matches '台灣.*'")</code></li> <li>• <code>=QUERY(交易記錄,"select sum(K) where H like '台灣%'")</code></li> </ul>
解說	以上分別使用三種不同的語法表示「開頭是台灣的」的條件，可以根據自身習慣使用其中一種。

### B. 篩選所有貨名中有標註「重量」或「容量」的貨名，並移除重複內容

公式	<code>=UNIQUE(QUERY(交易記錄,"select H where lower(H) matches '.*[0-9]+[mlkgcoz].*'"))</code>
解說	先觀察原始資料了解貨名的特性，大致上可以發現有標註者的語法都是「數字 + ml / l / cc / oz / g / kg」，而重量或容量的單位又有可能是大寫或小寫。綜合以上，將條件設定為 <code>where lower(H) matches '.*[0-9]+[mlkgcoz].*'</code> ，將字串轉換成小寫後再使用正規表示式判斷是否包含符合條件的字串。

在這四節了解欄位選取與篩選的基本操作後，接下來將介紹 QUERY 中最重要的功能 — 根據欄位的內容聚合以及製作資料透視表。

## 13.5 group by / pivot 影片 [Link](#)

在 QUERY 中最重要的功能是使用公式來完成資料透視表，可以使用 group by 、pivot 兩個語法來完成，如下：

語法	說明
group by	<p><b>select 欄位, 匯總函式 group by 欄位 [, 欄位...]</b></p> <p>例如 <code>select A,sum(B) group by A</code> 、<code>select A,B,count(C),max(D) group by A,B</code> ，其注意事項如下：</p> <ul style="list-style-type: none"><li>• group by 可根據某欄位相同的值匯總運算，若要根據多個欄位的值匯總可以使用逗點分格。</li><li>• 假設欄位中有 M 個獨特的值、要輸出 N 個匯總函式，則最後的輸出會有 M 列 N 欄。</li><li>• <b>group by 輸出不會包含匯總的欄位名稱，因此在 select 的時候也要附上這些欄位。</b></li></ul>
pivot	<p><b>select 匯總函式 pivot 欄位 [, 欄位...]</b></p> <p>例如 <code>select sum(B),min(B),max(B) pivot A</code> ，注意事項如下：</p> <ul style="list-style-type: none"><li>• 假設欄位中有 M 個獨特的值、要輸出 N 個匯總函式，則最後的輸出會有 2 列 (標題 &amp; 值) M 欄。</li><li>• pivot 輸出時會包含匯總的欄位名稱，因此在 select 的時候不用附上這些欄位。</li></ul>

換句話說，若要將 group by 和 pivot 分別對應到資料透視表的列和欄，而 select 和 where 則是值和篩選器。接下來使用一個範例介紹輸出的格式：

## A. 顯示各性別、家庭人口數的會員數量

公式	<ul style="list-style-type: none"> <li>=QUERY(會員資料,"select D,count(A) group by D pivot C")</li> <li>=QUERY(會員資料,"select C,count(A) group by C pivot D")</li> <li>=QUERY(會員資料,"select C,D,count(A) group by C,D")</li> <li>=QUERY(會員資料,"select count(A) pivot C,D")</li> </ul>																																																									
解說	<p>四種方式的輸出格式不同，如下圖：</p> <ul style="list-style-type: none"> <li>=QUERY(會員資料,"select D,count(A) group by D pivot C")</li> </ul> <table border="1"> <thead> <tr> <th>家庭人口數</th> <th>女</th> <th>男</th> </tr> </thead> <tbody> <tr> <td>1~2人</td> <td>1</td> <td>5</td> </tr> <tr> <td>3~4人</td> <td>17</td> <td>1</td> </tr> <tr> <td>7人以上</td> <td>4</td> <td>2</td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li>=QUERY(會員資料,"select C,count(A) group by C pivot D")</li> </ul> <table border="1"> <thead> <tr> <th>性別</th> <th>1~2人</th> <th>3~4人</th> <th>7人以上</th> </tr> </thead> <tbody> <tr> <td>女</td> <td>1</td> <td>17</td> <td>4</td> </tr> <tr> <td>男</td> <td>5</td> <td>1</td> <td>2</td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li>=QUERY(會員資料,"select C,D,count(A) group by C,D")</li> </ul> <table border="1"> <thead> <tr> <th>性別</th> <th>家庭人口數</th> <th>count 會員卡號</th> </tr> </thead> <tbody> <tr> <td>女</td> <td>1~2人</td> <td>1</td> </tr> <tr> <td>女</td> <td>3~4人</td> <td>17</td> </tr> <tr> <td>女</td> <td>7人以上</td> <td>4</td> </tr> <tr> <td>男</td> <td>1~2人</td> <td>5</td> </tr> <tr> <td>男</td> <td>3~4人</td> <td>1</td> </tr> <tr> <td>男</td> <td>7人以上</td> <td>2</td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li>=QUERY(會員資料,"select count(A) pivot C,D")</li> </ul> <table border="1"> <thead> <tr> <th>女,1~2人</th> <th>女,3~4人</th> <th>女,7人以上</th> <th>男,1~2人</th> <th>男,3~4人</th> <th>男,7人以上</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>17</td> <td>4</td> <td>5</td> <td>1</td> <td>2</td> </tr> </tbody> </table>	家庭人口數	女	男	1~2人	1	5	3~4人	17	1	7人以上	4	2	性別	1~2人	3~4人	7人以上	女	1	17	4	男	5	1	2	性別	家庭人口數	count 會員卡號	女	1~2人	1	女	3~4人	17	女	7人以上	4	男	1~2人	5	男	3~4人	1	男	7人以上	2	女,1~2人	女,3~4人	女,7人以上	男,1~2人	男,3~4人	男,7人以上	1	17	4	5	1	2
家庭人口數	女	男																																																								
1~2人	1	5																																																								
3~4人	17	1																																																								
7人以上	4	2																																																								
性別	1~2人	3~4人	7人以上																																																							
女	1	17	4																																																							
男	5	1	2																																																							
性別	家庭人口數	count 會員卡號																																																								
女	1~2人	1																																																								
女	3~4人	17																																																								
女	7人以上	4																																																								
男	1~2人	5																																																								
男	3~4人	1																																																								
男	7人以上	2																																																								
女,1~2人	女,3~4人	女,7人以上	男,1~2人	男,3~4人	男,7人以上																																																					
1	17	4	5	1	2																																																					

有了 group by 及 pivot 兩個語法後，基本上可以完成各種資料透視表的功能，但是在輸出的格式還需要進行一些調整，而 QUERY 語法中也有相對應的功能，例如排序輸出、限制輸出列數、修改標題欄位名稱等，將在下一節有更深入的介紹！

## 13.6 其他資料篩選與設定

影片 [Link](#)

本節將介紹其他與資料篩選及設定相關的語法，將先簡單介紹各個語法的功能、範例，並說明在所有 QUERY 的語法中使用的順序，相關語法共有以下七個：

語法	說明
limit	<b>limit 數字</b> 用來限制輸出的列數，例如 <code>select * limit 30</code> 。
offset	<b>offset 數字</b> 跳過資料中指定數量的列數，例如 <code>select * offset 30</code> (從第 31 列開始)。
skipping	<b>skipping 數字</b> 每隔幾列抽取一筆資料，例如 <code>select * skipping 30</code> (第 1 / 31 / 61 /... 列)。
order by	<b>order by 欄位 [desc / asc] [, 欄位...]</b> 依據哪個欄位的值排序，後面可加上 <code>desc / asc</code> 表示遞減 / 遞增排序，預設為 <code>asc</code> ，例如 <code>select * order by A asc, B asc, C desc</code> 。
label	<b>label 欄位 '名稱' [, 欄位 '名稱'...]</b> 將某欄位的表頭名稱改成指定的名稱，例如 <code>select A, sum(B) group by A label A '項目', sum(B) '加總'</code> 。
format	<b>format 欄位 '內容格式' [, 欄位 '內容格式']</b> 將欄位的輸出改成指定格式，例如 <code>select * format A '#,###.00', B 'yyyy-MM-dd', C 'Yes:No'</code> ，有以下三種格式： <ul style="list-style-type: none"><li>• 數字：例如 <code>'#,##0.00'</code>。</li><li>• 日期與時間：例如 <code>'yyyy-MM-dd'</code>。</li><li>• 布林值：使用的格式為 <code>'value-if-true:value-if-false'</code>，例如 <code>'Yes:No'</code>。</li></ul> 雖然以上三種方式可以設定輸出的格式，但若輸出的儲存格本身有格式設定時，會以儲存格的格式為主。
options	<b>options no_format / no_values</b> <ul style="list-style-type: none"><li>• <code>no_format</code>：只輸出值，例如 <code>select * options no_format</code>。</li><li>• <code>no_values</code>：只輸出格式，例如 <code>select * options no_values</code>。</li></ul>

到了這裡，我們已經介紹了所有 QUERY 相關的語法，而在使用這些語法時實際上需要有固定的呈現順序，如下：

`select ... where ... group by ... pivot ... order by ... skipping ... limit ... offset ...  
label ... format ... options ...`

以下將舉幾個例子說明上述語法的用法：

**A. 顯示第 4~8 位會員的卡號**

公式	=QUERY(會員資料,"select A limit 5 offset 3")
解說	使用 offset 3 跳過 3 筆資料，並 limit 5 使用輸出五筆資料。

**B. 顯示第 3 / 5 / 7 / 9 / 11 位會員的卡號**

公式	=QUERY(會員資料,"select A skipping 2 limit 5 offset 1")
解說	skipping 與 offset 一起使用時，會先使用 skipping 2 輸出第 1 / 3 / 5 / 7... 筆資料，再使用 offset 1 跳過上述的第一筆資料，因此會輸出 3 / 5 / 7 ... 筆資料。

**C. 顯示金額最大的 10 筆交易編號及其金額，將金額欄位取名為「交易金額」，格式使用 \$X,XXX**

公式	=QUERY(交易記錄,"select A,sum(K) group by A order by sum(K) desc limit 10 label sum(K) '交易金額' format sum(K) '\$#,##'")
解說	<ul style="list-style-type: none"><li>• 使用 select A,sum(K) group by A 輸出每位會員卡號的貨品金額加總。</li><li>• 使用 order by sum(K) desc limit 10 讓資料以總金額遞減排序，並輸出金額前十多的會員。</li><li>• 使用 label sum(K) '交易金額' format sum(K) '\$#,##' 重新命名表頭與設定資料的格式。</li></ul>

到這裡已經介紹完所有 QUERY 相關的語法了，也許你會認為 QUERY 本身沒有很實用，但其實若將 QUERY 與儲存格的輸入並用，也就是在 **查詢** 中使用儲存格的值，會大幅增加此函式的實用性，而做法也相當簡單，只要在字串中使用 & 連接儲存格的內容即可，例如 "select A where A > "&A1&"。而在下一節的進階應用中，將會示範多層 QUERY 的用法，及如何將 QUERY 與其他函式搭配使用。

## 13.7 進階應用 影片 [Link](#)

本節將會介紹如何在試算表使用交易記錄檔建立一個搜尋器，使用者可以選取特定店號、要搜尋的日期區間，並在下方顯示出該店號的訂單總額、訂單數、訂單 AOV (平均訂單價值)，此外也可以選擇不填寫店號、日期區間，若不填寫會使用預設值輸出，完成結果如下圖。

但因為整個過程較為複雜，在此不一一說明各步驟的完成方式，而是在教學影片中示範，在影片中將會從零開始，示範公式的思考脈絡與調整，從最基本的公式 (都有輸入內容) 開始到如何在公式中考慮預設值等流程，非常推薦各位同學有空的時候跟著影片一起思考並操作，可以從中學到非常多的東西。

店號	<input type="button" value="▼"/>	預設輸出所有店號
日期區間 (起始)		預設為 2022-01-01
日期區間 (結束)		預設為 2023-12-31

各分店訂單資料			
店號	訂單總額	訂單數	訂單 AOV
105	46,397	69	672
101	19,903	30	663
108	11,852	19	624
122	3,951	8	494
210	1,591	2	796

---

本章完成檔案：[Chapter 13. 函式 \(八\) : QUERY 完成版](#)

## Chapter 14. Google Sheets 擴充功能簡介 影片 [Link](#)

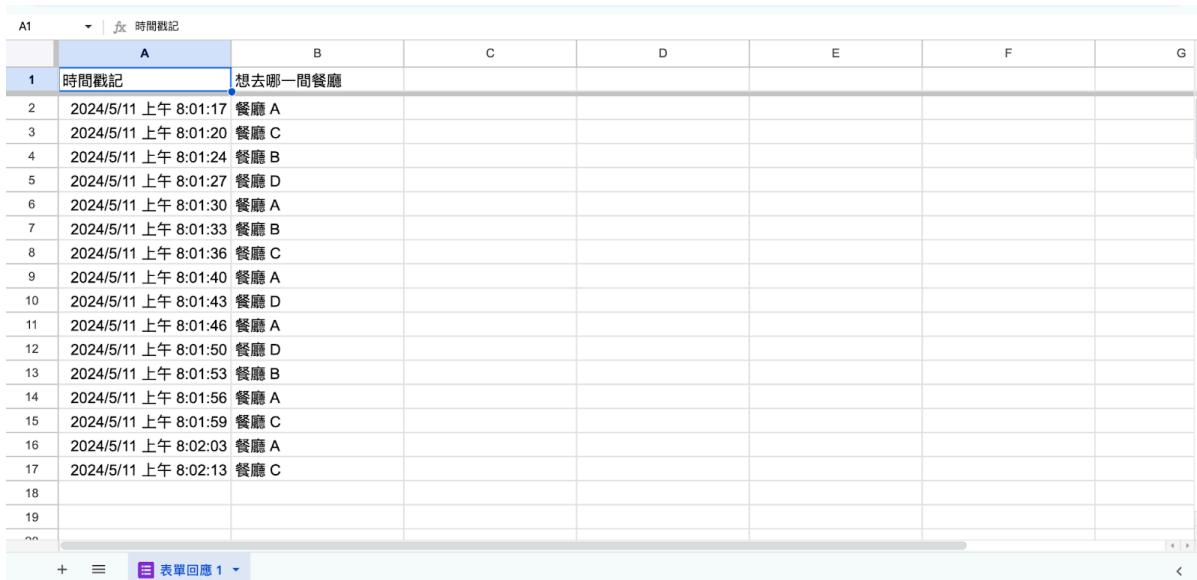
本章示範檔案：[!\[\]\(78e98417dae2881281a898bfb4b588fc\_img.jpg\) Chapter 14. Google Sheets 擴充功能簡介](#)

Google Sheets 除了自身強大的功能外，還能與其他 Google 的工具搭配使用，此外還可以透過 Google Workplace Marketplace 取得其他以 Google 工具為基礎的強大工具，增加 Google Sheets 的實用性。本章第 1 節將會介紹如何與其他工具如 Google 文件、簡報、表單等工具並用；第 2 節將介紹如何取得與使用外掛程式，並以 GoalSeek 作為範例說明其用法；第 3 節將簡單介紹另一個強大的功能 — Google Apps Script，可以用於 Google 的各項服務。

### 14.1 與其他 Google 工具的串接 影片 [Link](#)

本節將介紹 Google Sheets 與 Google 表單 / 文件 / 簡報相關的用法。

首先是 Google 表單，若要將現有的表單內容輸出到 Google Sheets 中，可以點選在編輯模式的「回覆 → 連結至試算表」選取要輸出的位置，可以是新的試算表或現有的試算表，選取後便能在 Google Sheets 中查看即時的表單回應，以進行更進一步的分析，如下圖：



The screenshot shows a Google Sheets spreadsheet with a single row of data. Row 1 contains the header "A1 時間戳記 想去哪一間餐廳". Rows 2 through 17 contain survey responses from a Google Form. Column A shows the timestamp and date (e.g., 2024/5/11 上午 8:01:17), and column B shows the restaurant choice (e.g., 餐廳 A, 餐廳 C, 餐廳 B, 餐廳 D). The bottom of the sheet shows a "表單回應 1" tab.

A1	時間戳記	想去哪一間餐廳
1	時間戳記	想去哪一間餐廳
2	2024/5/11 上午 8:01:17	餐廳 A
3	2024/5/11 上午 8:01:20	餐廳 C
4	2024/5/11 上午 8:01:24	餐廳 B
5	2024/5/11 上午 8:01:27	餐廳 D
6	2024/5/11 上午 8:01:30	餐廳 A
7	2024/5/11 上午 8:01:33	餐廳 B
8	2024/5/11 上午 8:01:36	餐廳 C
9	2024/5/11 上午 8:01:40	餐廳 A
10	2024/5/11 上午 8:01:43	餐廳 D
11	2024/5/11 上午 8:01:46	餐廳 A
12	2024/5/11 上午 8:01:50	餐廳 D
13	2024/5/11 上午 8:01:53	餐廳 B
14	2024/5/11 上午 8:01:56	餐廳 A
15	2024/5/11 上午 8:01:59	餐廳 C
16	2024/5/11 上午 8:02:03	餐廳 A
17	2024/5/11 上午 8:02:13	餐廳 C
18		
19		
...		

若要對現有的表單進行任何設定，可以點選「工具 → 管理表單」進行設定，包括以下幾種設定項目：

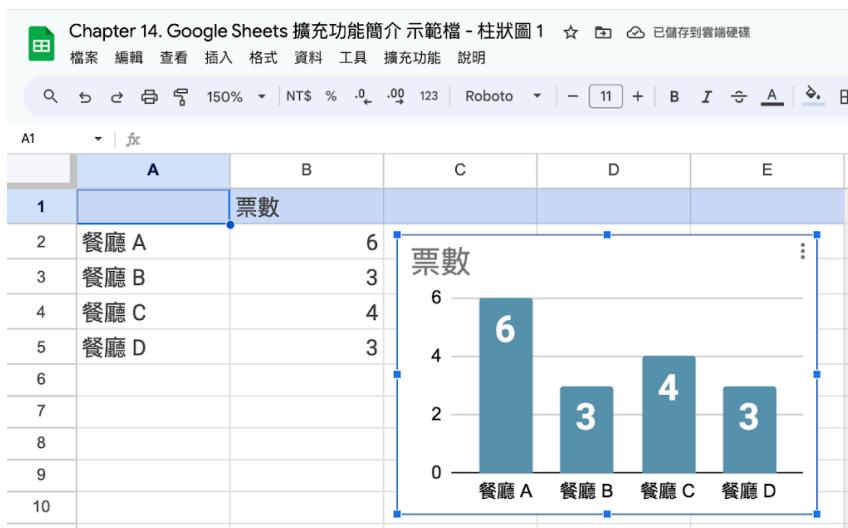
- 編輯表單 / 查看即時表單：會跳轉至表單編輯模式的問題頁 / 填寫模式。
- 傳送表單 / 在網頁中嵌入表單：會跳轉至表單編輯模式，並跳出傳送表單 / 嵌入網頁相關的視窗，在視窗中進行設定即可。
- 顯示回覆內容摘要：會跳轉至表單編輯模式的回覆頁。
- 取消表單連結：取消後將不會自動更新表單填答的內容，也可以點選左下角的工作表名稱取消表單連結。

接下來介紹在 Google 文件 / 簡報中使用 Google Sheets，在文件 / 簡報中主要會用到 Google Sheets 的圖表功能，若要插入圖表，主要有以下兩種方式：

- 直接在文件 / 簡報中整理資料 / 繪圖：在文件 / 簡報點選「插入 → 圖表」選取想要的圖表，下圖以柱狀圖為例，插入後點選右上角的「... → 開啟來源文件」後進入該圖表的 Google Sheets 編輯。



開啟來源文件後就能設定各項圖表內容，基本上就是一張專屬於此份文件 / 簡報的試算表，如下圖：



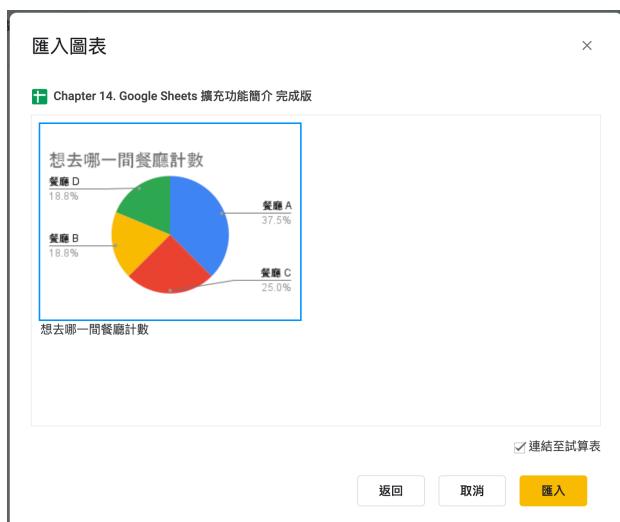
更新完畢後回到簡報 / 文件點選「更新」即可將圖表更新成最新的狀態，如下圖。



- 連結現有試算表資料：需要先在試算表中繪製好圖表，再將圖表轉到文件 / 簡報中，有以下兩種做法：
  - 直接複製試算表的圖表，然後在文件 / 簡報中貼上：此時文件 / 簡報會詢問是否要連結至試算表，如下圖。兩者的差異在於之後試算表的值 / 圖表變動時，簡報是否會出現「更新」讓使用者選擇。



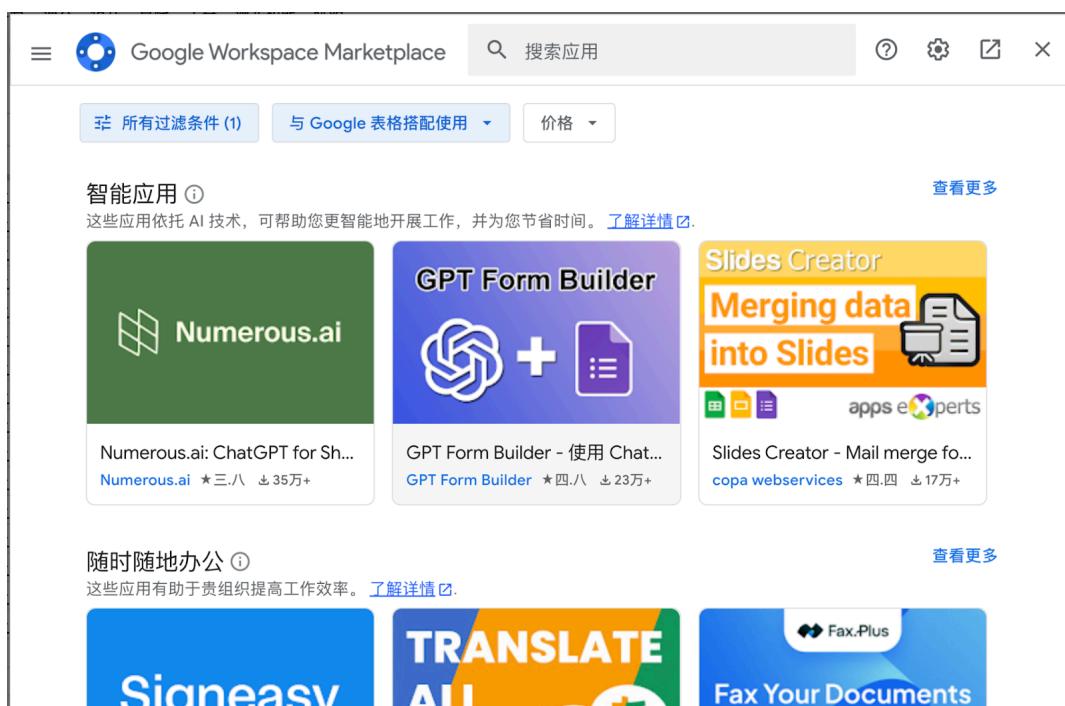
- 點選「插入 → 圖表 → 使用試算表資料」選取想要的圖表，右下點有「連結至試算表」的勾選符號決定是否會出現「更新」。



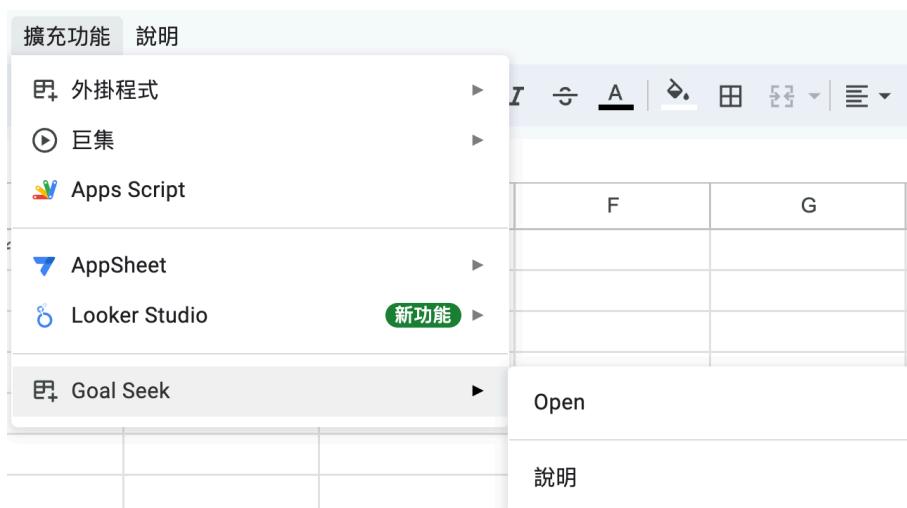
了解如何讓 Google Sheets 與其他 Google 的產品連動後，可以將所有需要分析的資料都透過 Google 的工具完成，節省做報告時在不同系統之間轉換的難度與時間。

## 14.2 Google Workspace 外掛程式簡介 影片 [Link](#)

Google 工具另一個強大之處在於有很多官方或民間開發的外掛程式，可以節省日常工作中的時間，這些外掛程式會安裝在你的 Google 雲端硬碟上，因此不限於 Google Sheets，文件 / 簡報 / Gmail 等都有實用的外掛程式可供下載，只要在 Google Sheets 中點選「擴充功能 → 外掛程式 → 取得外掛程式」，就會進入到 Google Workspace Marketplace，如下圖：



在此將以 Google 自己開發的 Goal Seek 為例，說明如何下載、應用外掛程式，在上方的搜尋中輸入 Goal Seek，找到之後點選「安裝」，安裝後就能在 Google Sheets 的「擴充功能」中找到外掛程式，如下圖：



Goal Seek 的操作方式可以見[官方文件](#)說明，點選「Open」後，右方會跳出視窗，設定的項目包括以下幾項：

- Set Cell：要為哪個儲存格的值設定目標。
- To Value：希望 Set Cell 最終的值是多少。
- By Changing Cell：透過改變哪個儲存格的值來使 Set Cell 達到 To Value，有以下三項限制：
  - 儲存格中不能有公式。
  - 只能有一個儲存格，不然會無法在超過一個以上的值之間決定結果。
  - 儲存格的值必須會影響 Set Cell 的值，也就是說 Set Cell 的儲存格會有公式，且會與 By Changing Cell 的值有關聯。

要輸入 Set Cell、By Changing Cell 的方式為「點選目標儲存格 / 可變變動的儲存格 → 點選 Set Cell / By Changing Cell 右方的儲存格符號」，而 To Value 只要直接輸入數字即可，輸入完所有項目後點選「Solve」就能開始進行目標查找，以下用兩個範例說明 Goal Seek 的用法：

#### A. 計算期末考需要幾分才會及格，各項成績與占比如下圖

	A	B	C	D	E
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					

項目	占比	分數
期中考	30%	45
第一次小考	5%	50
第二次小考	5%	40
作業一	5%	80
作業二	5%	85
作業三	5%	80
平時成績	10%	80
期末考	30%	
總成績	100%	

結果	期末考需要 72.5 分。
解說	<ul style="list-style-type: none"> <li>• 儲存格 D11 公式：總成績公式為 <math>=SUMPRODUCT(C3:C10,D3:D10)</math>，因此期末考 (D10) 的值會影響到總成績是否能超過 60 分。</li> <li>• Set Cell：在此要設定總成績的目標，因此為 D11。</li> <li>• To Value：及格是 60 分，輸入 60。</li> <li>• By Changing Cell：要知道「期末考」要考幾分，因此為 D10。</li> </ul>

**B. 計算 2025 年銷售量需要較前一年上漲多少百分比 ( $x = ?$ )，才能讓總利潤成長 50%，各項收入與成本如下圖：**

	A	B	C	D	E	F	G	H	I	J
1										
2		項目		2024 年			2025 目標			
3		收入	銷售量	每單位售價	銷售額		銷售量 (目標上漲 $x\%$ )	每單位售價 (比去年上漲 10%)	銷售額	
4		產品收入								
5		產品 A	800	250	200,000			275	0	
6		產品 B	500	200	100,000			220	0	
7		產品 C	1,200	40	48,000			44	0	
8		其他收入			5,000				4,000	
9		總收入			353,000				4,000	
10										
11		成本	銷售量	每單位生產 成本	總成本		銷售量 (目標上漲 $x\%$ )	每單位生產成本 (比去年上漲 7%)	總成本	
12		產品生產成本								
13		產品 A	800	225	180,000			241	0	
14		產品 B	500	175	87,500			187	0	
15		產品 C	1,200	38	45,600			41	0	
16		其他固定成本			7,900				8,300	
17		總成本			321,000				8,300	
18										
19		總利潤			32,000		希望成長 50% (48,000) >>			
20										
21		x%								
22										

結果	$x\% = 9.45315286517143\%$
解說	<ul style="list-style-type: none"> <li>儲存格 G5:G7、G13:G15 公式：都是比去年成長 <math>x\%</math>，因此輸入 <math>=D_*(1+\$C\$21)</math> 即可，其中 <math>D_</math> 為 D5:D7、D13:D15。</li> <li>Set Cell：在此要設定總利潤的目標，因此為 I19，並在 I19 輸入公式 <math>=I9-I17</math>。</li> <li>To Value：比 2024 年的總利潤 32,000 成長 50%，所以是 48,000。</li> <li>By Changing Cell：要求出銷售量，但銷售量都是由比去年上漲 <math>x\%</math> 影響，因此影響利潤最終的儲存格為 C21。</li> </ul>

Goal Seek 只是其中一項外掛程式，在此如果沒有辦法理解此功能的用法也沒有關係，更重要的是在遇到問題或有想要的功能時，知道可以在 Google Workspace Marketplace 尋找看看有沒有適用的工具，並閱讀相關的官方文件，讓工作事半功倍。

## 14.3 Apps Script 簡介 影片 [Link](#)

Apps Script 是一種 JavaScript 雲端腳本語言用來擴展 Google Apps 以及建立 Web 應用程式。可以做到幾乎所有與 Google Sheets 或其他 Google 服務有關的項目，而且都是免費的。在本節將使用兩個範例介紹如何在 Google Sheets 中使用 Apps Script，至於程式碼的細節並不會做過多的描述，有興趣的同學可以在[官方網站](#)進行學習，作者自己也是透過官方文件、ChatGPT、網路論壇 Stack Overflow 進行學習。

若要在 Google Sheets 中建立一個 GAS 的專案，可以點選「擴充功能 → Apps Script」，點選後會進入到 Apps Script 的開發環境，如下圖：



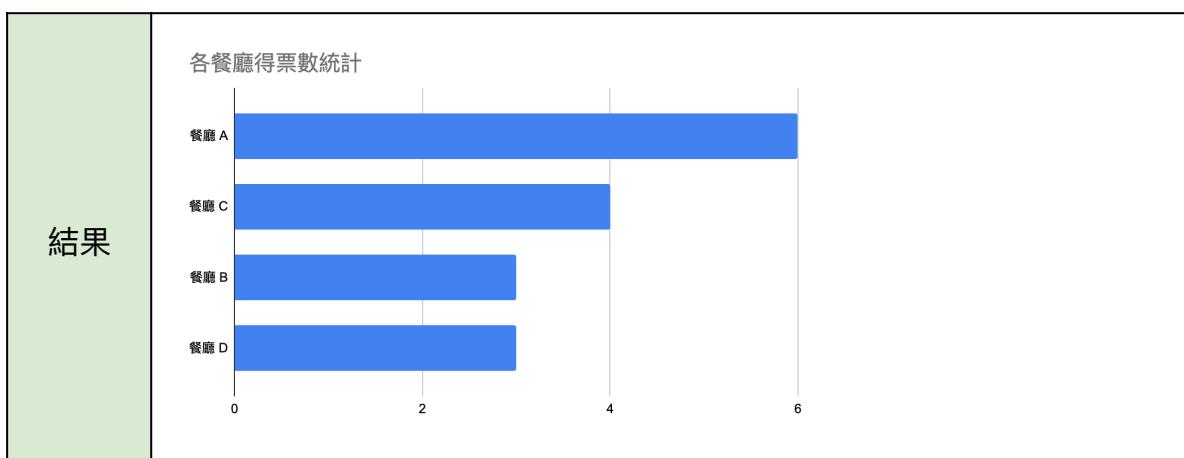
若將程式碼寫完並存檔後，便能點選「執行」開始執行程式，Apps Script 就可以根據程式的要求對資料進行處理，以下將介紹兩個範例說明 Apps Script 是如何運作與執行，其中範例的程式碼都是透過 ChatGPT 完成。

### A. 統計工作表「14.1 與其他 Google 工具的串接」的票數

結果	<table border="1"> <thead> <tr> <th></th><th>A</th><th>B</th></tr> </thead> <tbody> <tr> <td>1</td><td>餐廳</td><td>票數</td></tr> <tr> <td>2</td><td>餐廳 A</td><td>6</td></tr> <tr> <td>3</td><td>餐廳 C</td><td>4</td></tr> <tr> <td>4</td><td>餐廳 B</td><td>3</td></tr> <tr> <td>5</td><td>餐廳 D</td><td>3</td></tr> </tbody> </table>			A	B	1	餐廳	票數	2	餐廳 A	6	3	餐廳 C	4	4	餐廳 B	3	5	餐廳 D	3
	A	B																		
1	餐廳	票數																		
2	餐廳 A	6																		
3	餐廳 C	4																		
4	餐廳 B	3																		
5	餐廳 D	3																		
幫我用 Apps Script 寫一隻程式，判斷目前 Google Sheets 試算表的「14.1 與其他 Google 工具的串接」工作表的 B2:B 每一個儲存格的值分別出現幾次，幫我將結果輸出至「14.3 Apps Script 簡介」工作表中，其中輸出有兩欄，分別把表頭命為「餐廳」和「票數」。																				

程式碼	<pre>function countValuesAndOutput() {   var ss = SpreadsheetApp.getActiveSpreadsheet();   var sourceSheet = ss.getSheetByName('14.1 與其他 Google 工具的串接');   var targetSheet = ss.getSheetByName('14.3 Apps Script 簡介');   // 清空目標工作表   targetSheet.clear();   // 取得原始資料範圍   var dataRange = sourceSheet.getRange('B2:B');   var values = dataRange.getValues();   // 初始化一個物件來存放值與出現次數的對應關係   var countObject = {};   // 計算每個值的出現次數   values.forEach(function(row) {     var value = row[0];     if (value !== "") {       if (countObject[value]) {         countObject[value]++;       } else {         countObject[value] = 1;       }     }   });   // 將結果輸出至目標工作表   var outputData = [];   outputData.push(['餐廳', '票數']);   for (var key in countObject) {     outputData.push([key, countObject[key]]);   }   targetSheet.getRange(1, 1, outputData.length, outputData[0].length).setValues(outputData); }</pre>
-----	--

B. 將 A. 產出的資料繪製成長條圖，並將圖表標題設為「各餐廳得票數統計」，貼在 C1 儲存格中



Prompt	幫我用 Apps Script 寫一隻程式，將 Google Sheets 試算表的「14.3 Apps Script 簡介」工作表的 A2:B 的資料繪製成長條圖，其中 A 欄是餐廳、B 欄是票數，幫我將圖表的標題設為「各餐廳得票數統計」，完成後把圖表貼在「14.3 Apps Script 簡介」工作表的 C1 儲存格中。
程式碼	<pre>function createBarChart() {   var ss = SpreadsheetApp.getActiveSpreadsheet();   var sheet = ss.getSheetByName('14.3 Apps Script 簡介');   // 取得 A2:B 範圍的資料   var dataRange = sheet.getRange('A2:B');   var data = dataRange.getValues();   // 創建圖表   var chart = sheet.newChart()     .setChartType(Charts.ChartType.BAR)     .addRange(dataRange)     .setPosition(1, 3, 0, 0)     .setOption('title', '各餐廳得票數統計')     .build();   // 將圖表插入工作表的 C1 儲存格中   sheet.insertChart(chart); }</pre>

在此不會介紹 Apps Script 更深入的用法，因為這是一門需要很多時間與精力學習的語法主要是讓各位同學知道有這樣一套工具可以完成工作中較為複雜的需求。此外，現在也有 ChatGPT 可以能快速撰寫程式碼並了解背後的語法邏輯，所以在此不用詳盡介紹，若對於使用 Google Sheets / Apps Script 進行自動化與數據分析有興趣的同學，歡迎大家上網找更多資源學習。

---

本章完成檔案：[+ Chapter 14. Google Sheets 擴充功能簡介 完成版](#)

## 結語

影片 [Link](#)

在這份教材中介紹了 99% 工作中 Google Sheets 相關的功能與函式，回顧一下這套教材至今帶給各位什麼內容：

- 匯入資料並使用內建功能快速了解資料分佈。(Chapter 2~3)
- 使用內建的資料透視表整理資料。(Chapter 4)
- 決定資料要使用哪種圖表呈現，呈現時要怎麼調整圖表的細節。(Chapter 5)
- 使用函式解決各式運算、查詢與篩選相關的問題。(Chapter 6~13)
- 處理特殊格式的資料，包括日期、字串等。(Chapter 8~9)
- 使用 Google Sheets 進行會計折舊、年金現值相關的計算。(Chapter 10)
- 使用陣列公式或自定義函式完成更進階的資料處理自動化。(Chapter 11~12)
- 使用類似 SQL 的語法撈取與統計資料。(Chapter 13)
- 將 Google Sheets 串接其他 Google 工具或外掛程式，增加應用範圍。(Chapter 14)

然而，要精通所有 Google Sheets 的操作並不是看完影片、跟著檔案照做一次就能熟練的，更重要的是平常生活與工作中的經驗累積。資料庫該如何設計、什麼情境適合使用什麼公式、某個公式出錯的原因與排除錯誤的方式等，都沒有任何的正確答案，而是需要透過長期的摸索與試錯後找到自己習慣的方法。這本手冊能做的是給大家一個指引，減少大家試錯的時間成本。

最後，Google Sheets 的變動速度非常快，隨時都可能推出新功能與介面，本手冊的各個功能說明、範例都是作者一人完成的，過程十分耗時，作者另外還有正職的工作，因此無法隨時與時俱進，望各位學生諒解。

若大家有任何 Google Sheets 上的問題，歡迎透過 Email ([hsintu0809@gmail.com](mailto:hsintu0809@gmail.com)) 或是 [Facebook \(Messenger\)](#) 與作者交流！

杜昕

2024/7

## 附錄：函式清單與頁數對照表 (按字母排序)

<b>A</b>		<a href="#">DCOUNTA</a>	163	<a href="#">ISBLANK</a>	159
<a href="#">ABS</a>	96	<a href="#">DDB</a>	152	<a href="#">ISDATE</a>	159
<a href="#">ADD</a>	96	<a href="#">DGET</a>	163	<a href="#">ISEMAIL</a>	159
<a href="#">ADDRESS</a>	117	<a href="#">DIVIDE</a>	96	<a href="#">ISERR</a>	159
<a href="#">AND</a>	101	<a href="#">DMAX</a>	163	<a href="#">ISERROR</a>	159
<a href="#">ARRAY_CONSTRAIN</a>	174	<a href="#">DMIN</a>	163	<a href="#">ISEVEN</a>	159
<a href="#">ARRAYFORMULA</a>	168	<a href="#">DOLLAR</a>	161	<a href="#">ISFORMULA</a>	159
<a href="#">AVERAGE</a>	106	<a href="#">DPRODUCT</a>	163	<a href="#">ISLOGICAL</a>	159
<a href="#">AVERAGE.WEIGHTED</a>	112	<a href="#">DSTDEV</a>	163	<a href="#">ISNA</a>	159
<a href="#">AVERAGEA</a>	106	<a href="#">DSTDEVP</a>	163	<a href="#">ISNONTEXT</a>	159
<a href="#">AVERAGEIF</a>	109	<a href="#">DSUM</a>	163	<a href="#">ISNUMBER</a>	159
<a href="#">AVERAGEIFS</a>	110	<a href="#">DVAR</a>	163	<a href="#">ISODD</a>	159
<b>B</b>		<a href="#">DVARP</a>	163	<a href="#">ISREF</a>	159
<a href="#">BYCOL</a>	183	<b>E</b>		<a href="#">ISTEXT</a>	159
<a href="#">BYROW</a>	183	<a href="#">EDATE</a>	136	<a href="#">ISURL</a>	159
<b>C</b>		<a href="#">EOMONTH</a>	136	<b>J</b>	
<a href="#">CEILING</a>	160	<a href="#">ERROR.TYPE</a>	99	<a href="#">JOIN</a>	142
<a href="#">CHOOSECOLS</a>	174	<b>F</b>		<b>L</b>	
<a href="#">CHOSEROWS</a>	174	<a href="#">FILTER</a>	129	<a href="#">LAMBDA</a>	182
<a href="#">COLUMN</a>	116	<a href="#">FIND</a>	145	<a href="#">LARGE</a>	106
<a href="#">COLUMNS</a>	116	<a href="#">FINDB</a>	145	<a href="#">LEFT</a>	140
<a href="#">CONCAT</a>	142	<a href="#">FIXED</a>	161	<a href="#">LEFTB</a>	141
<a href="#">CONCATENATE</a>	143	<a href="#">FLATTEN</a>	171	<a href="#">LEN</a>	140
<a href="#">COUNT</a>	106	<a href="#">FLOOR</a>	160	<a href="#">LENB</a>	141
<a href="#">COUNTA</a>	106	<a href="#">FREQUENCY</a>	174	<a href="#">LET</a>	177
<a href="#">COUNTBLANK</a>	109	<a href="#">FV</a>	154	<a href="#">LOOKUP</a>	121
<a href="#">COUNTIF</a>	109	<b>G</b>		<a href="#">LOWER</a>	140
<a href="#">COUNTIFS</a>	110	<a href="#">GETPIVOTDATA</a>	75	<b>M</b>	
<a href="#">COUNTUNIQUE</a>	109	<b>H</b>		<a href="#">MAKEARRAY</a>	184
<a href="#">COUNTUNIQUEIFS</a>	110	<a href="#">HLOOKUP</a>	121	<a href="#">MAP</a>	183
<b>D</b>		<a href="#">HOUR</a>	134	<a href="#">MATCH</a>	119
<a href="#">DATE</a>	132	<a href="#">HSTACK</a>	172	<a href="#">MAX</a>	106
<a href="#">DATEDIF</a>	136	<b>I</b>		<a href="#">MAXA</a>	106
<a href="#">DATEVALUE</a>	133	<a href="#">IF</a>	103	<a href="#">MAXIFS</a>	110
<a href="#">DAVERAGE</a>	163	<a href="#">IFERROR</a>	104	<a href="#">MEDIAN</a>	106
<a href="#">DAY</a>	134	<a href="#">IFNA</a>	104	<a href="#">MID</a>	140
<a href="#">DAYS</a>	136	<a href="#">IFS</a>	103	<a href="#">MIDB</a>	141
<a href="#">DB</a>	152	<a href="#">INDEX</a>	119	<a href="#">MIN</a>	106
<a href="#">DCOUNT</a>	163	<a href="#">INDIRECT</a>	117	<a href="#">MINA</a>	106

<a href="#">MINIFS</a>	110	<a href="#">REPLACEB</a>	145	<a href="#">TO_TEXT</a>	161
<a href="#">MINUS</a>	96	<a href="#">REPT</a>	156	<a href="#">TOCOL</a>	171
<a href="#">MINUTE</a>	134	<a href="#">RIGHT</a>	140	<a href="#">TODAY</a>	132
<a href="#">MOD</a>	96	<a href="#">RIGHTB</a>	141	<a href="#">TOROW</a>	171
<a href="#">MODE</a>	106	<a href="#">ROUND</a>	160	<a href="#">TRANSPOSE</a>	171
<a href="#">MONTH</a>	134	<a href="#">ROUNDDOWN</a>	160	<a href="#">TRIM</a>	142
<a href="#">MROUND</a>	160	<a href="#">ROUNDUP</a>	160	<a href="#">TYPE</a>	31
<a href="#">MULTIPLY</a>	96	<a href="#">ROW</a>	116	<b>U</b>	
<b>N</b>		<a href="#">ROWS</a>	116	<a href="#">UNIQUE</a>	127
<a href="#">NETWORKDAYS</a>	137	<b>S</b>		<a href="#">UPPER</a>	140
<a href="#">NETWORKDAYS.INTL</a>	137	<a href="#">SCAN</a>	183	<b>V</b>	
<a href="#">NORMINV</a>	157	<a href="#">SEARCH</a>	145	<a href="#">VALUE</a>	161
<a href="#">NOT</a>	101	<a href="#">SEARCHB</a>	145	<a href="#">VAR</a>	106
<a href="#">NOW</a>	132	<a href="#">SECOND</a>	134	<a href="#">VARA</a>	106
<a href="#">NPER</a>	154	<a href="#">SEQUENCE</a>	156	<a href="#">VARP</a>	106
<b>O</b>		<a href="#">SLN</a>	152	<a href="#">VARPA</a>	106
<a href="#">OFFSET</a>	125	<a href="#">SMALL</a>	106	<a href="#">VDB</a>	152
<a href="#">OR</a>	101	<a href="#">SORT</a>	127	<a href="#">VLOOKUP</a>	121
<b>P</b>		<a href="#">SORTN</a>	127	<a href="#">VSTACK</a>	172
<a href="#">PERCENTIF</a>	109	<a href="#">SPARKLINE</a>	92	<b>W</b>	
<a href="#">PERCENTILE</a>	106	<a href="#">SPLIT</a>	142	<a href="#">WEEKDAY</a>	134
<a href="#">PERCENTRANK</a>	107	<a href="#">SQRT</a>	96	<a href="#">WEEKNUM</a>	134
<a href="#">PMT</a>	154	<a href="#">STDEV</a>	106	<a href="#">WORKDAY</a>	137
<a href="#">POWER</a>	96	<a href="#">STDEVA</a>	106	<a href="#">WORKDAY.INTL</a>	137
<a href="#">PROPER</a>	140	<a href="#">STDEVP</a>	106	<a href="#">WRAPCOLS</a>	171
<a href="#">PV</a>	154	<a href="#">STDEVPA</a>	106	<a href="#">WRAPROWS</a>	171
<b>Q</b>		<a href="#">SUBSTITUTE</a>	145	<b>X</b>	
<a href="#">QUARTILE</a>	106	<a href="#">SUM</a>	106	<a href="#">XLOOKUP</a>	124
<a href="#">QUERY</a>	188	<a href="#">SUMIF</a>	109	<a href="#">XMATCH</a>	124
<a href="#">QUOTIENT</a>	96	<a href="#">SUMIFS</a>	110	<b>Y</b>	
<b>R</b>		<a href="#">SUMPRODUCT</a>	112	<a href="#">YEAR</a>	134
<a href="#">RAND</a>	156	<a href="#">SWITCH</a>	103		
<a href="#">RANDARRAY</a>	156	<a href="#">SYD</a>	152		
<a href="#">RANDBETWEEN</a>	156	<b>T</b>			
<a href="#">RANK</a>	107	<a href="#">TEXT</a>	161		
<a href="#">RATE</a>	154	<a href="#">TEXTJOIN</a>	143		
<a href="#">REDUCE</a>	183	<a href="#">TIME</a>	132		
<a href="#">REGEXEXTRACT</a>	148	<a href="#">TIMEVALUE</a>	133		
<a href="#">REGEXMATCH</a>	148	<a href="#">TO_DATE</a>	160		
<a href="#">REGEXREPLACE</a>	148	<a href="#">TO_DOLLARS</a>	160		
<a href="#">REPLACE</a>	145	<a href="#">TO_PERCENT</a>	160		
		<a href="#">TO_PURE_NUMBER</a>	160		