



數值控制的藝術：打破「算數」迷思的 Excel 函數圖鑑

從位準捨入到倍數進階，重塑商業規則的高階控制指南

專為財務、人資與營運數據分析打造的精準儀表板。

Excel 只是用來「算數學」？

✗ 迷思：我們總是在尋求最「精確」的數學答案。



不，Excel 是用來「控制商業結果」。

✓ 規則：你想對公司有利？



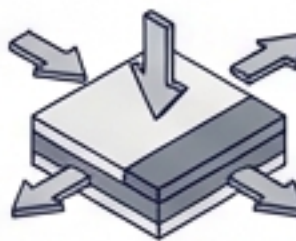
SALES: 1,250,000
SALES: 1,290,000
SALES: 900,000
SALES: 370,000

✓ 風險：你想確保絕不超支？



COST: 750,000
COST: 200,000
COST: 100,000
COST: 10,000

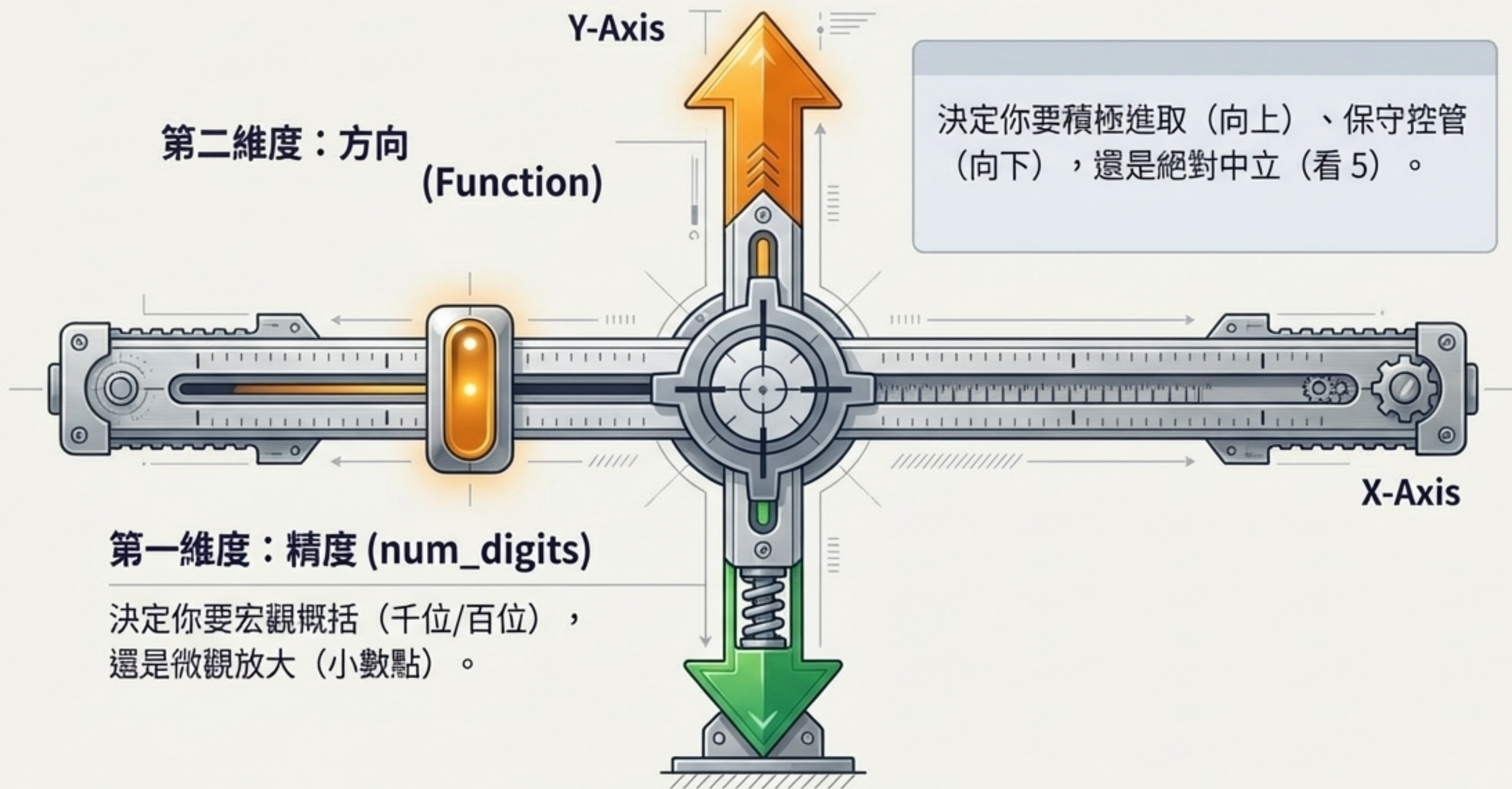
✓ 公平：你想維持常規標準？



PERFORMANCE: 95%

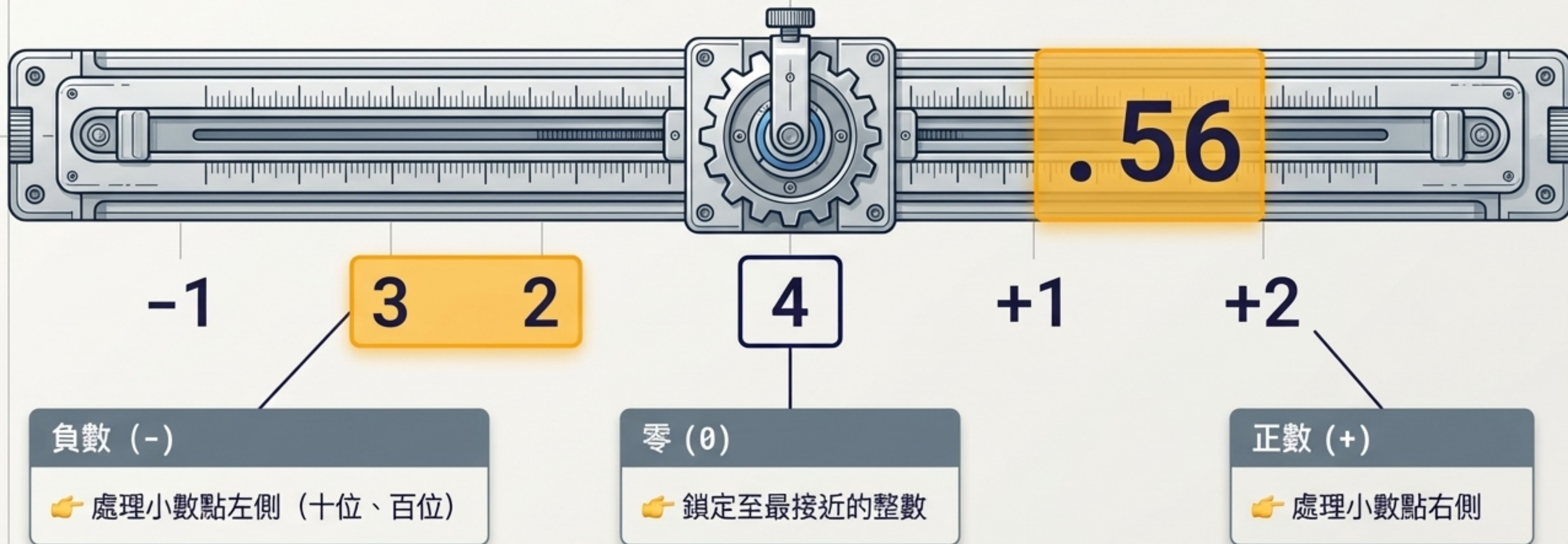
所有的進位與捨去，本質上都是一種商業策略。

建立您的「數值控制系統」



精度滑桿 (num_digits)：決定處理的絕對位置

1234.567





正數：保留微觀細節

`=ROUND(3.14159, 2) → 3.14`

取到小數點後第 2 位。



零：抹除所有小數

`=ROUND(3.6, 0) → 4`

四捨五入至個位（整數）。

負數：大額數字的宏觀簡化

用於報表的高階視角，僅需顯示整百或整千的數值。



`=ROUND(123, -1) → 120`

`num_digits = -1` (處理至十位)

`=ROUND(1234, -2) → 1200`

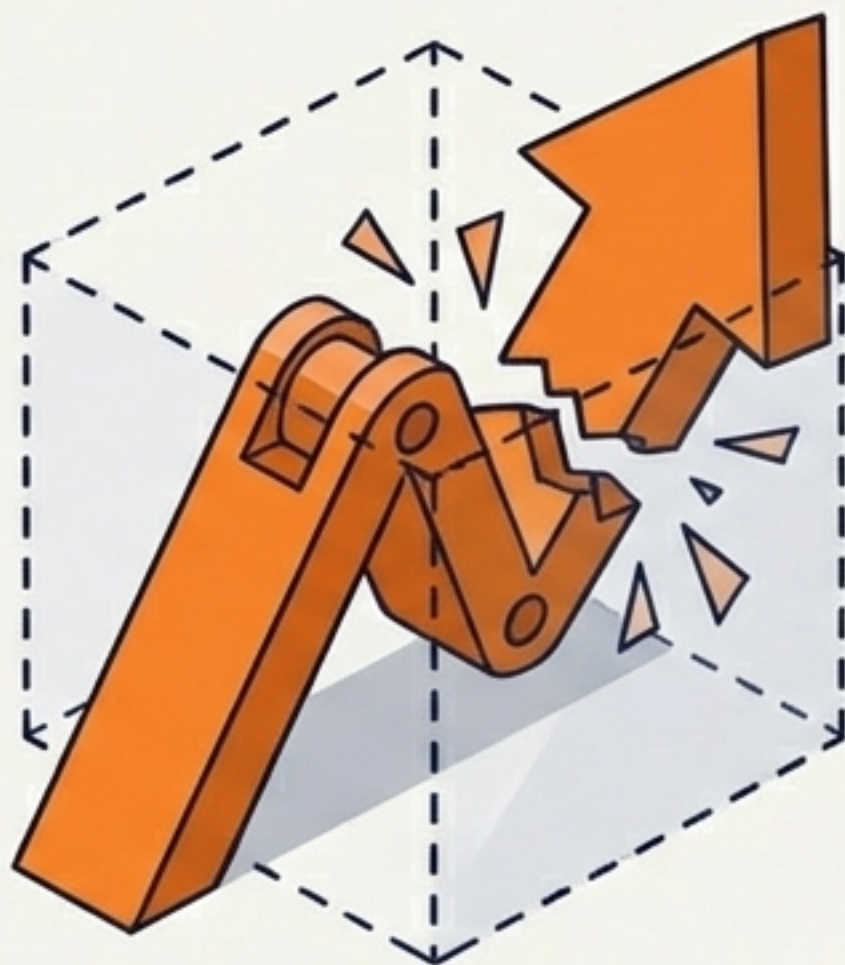
`num_digits = -2` (處理至百位)

方向控制：選擇你的商業動機

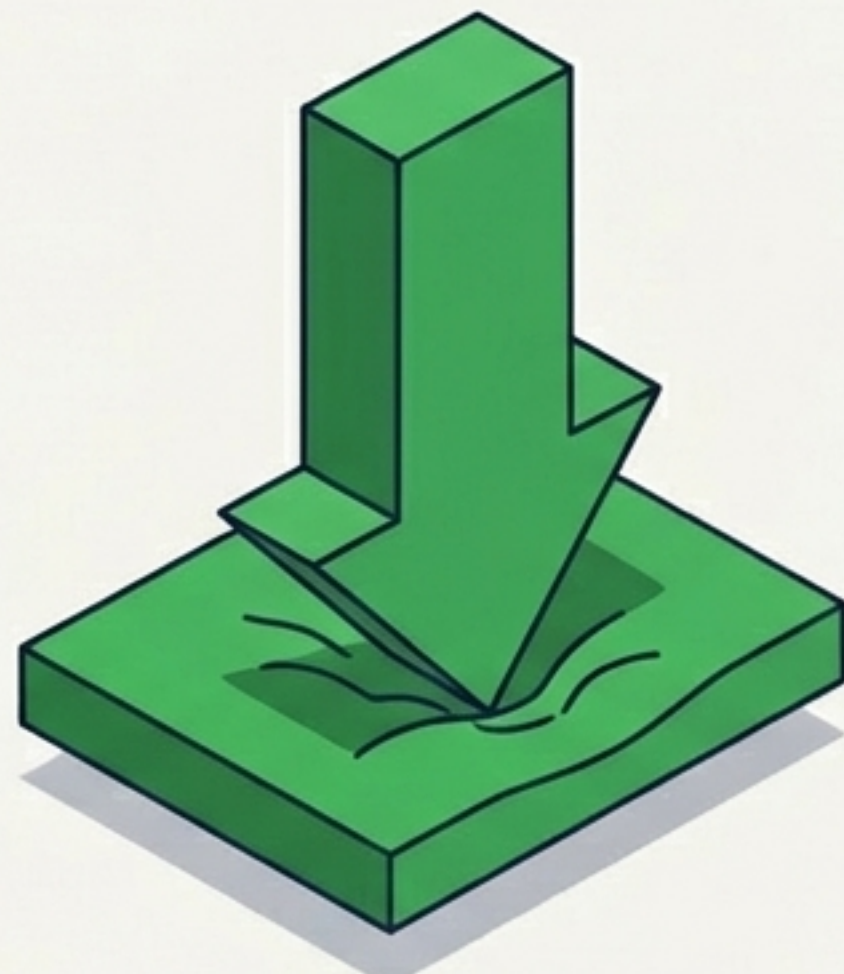
語法的差異，反映的是背後的決策思維。



ROUND
(中立 / 公平)



ROUNDUP
(進取 / 利己)



ROUNDDOWN
(保守 / 控管)

ROUND：四捨五入的公平仲裁者



邏輯：只看 5，決定進退。追求常規標準的公平性。

`=ROUND(3.14159, 2) → 3.14`（小數處理）

`=ROUND(123, -1) → 120`（取到十位）



實務應用：薪資與獎金計算

發薪只發整數：`=ROUND(薪資, 0)`

報表呈現整百：`=ROUND(總額, -2)`

ROUNDUP：確保公司利益的無條件進位



Core 概念

邏輯：只要有小數，就一定往上加（變大）。不看 5。

$\text{ROUND}(3.1, 0) \rightarrow 3$

$\text{ROUNDUP}(3.1, 0) \rightarrow 4$

$=\text{ROUNDUP}(1234, -2) \rightarrow 1300$ （進位到百位）



實務應用：運費與定價計算

每 1kg 算一單位，不足也算一單位。

2.1kg $\rightarrow$ 3kg ($=\text{ROUNDUP}(2.1, 0)$)。完全符合現實世界的收費邏輯。

ROUNDDOWN：極小化風險的無條件捨去



邏輯：永遠往「小」的方向，不管小數是多少，直接砍掉。

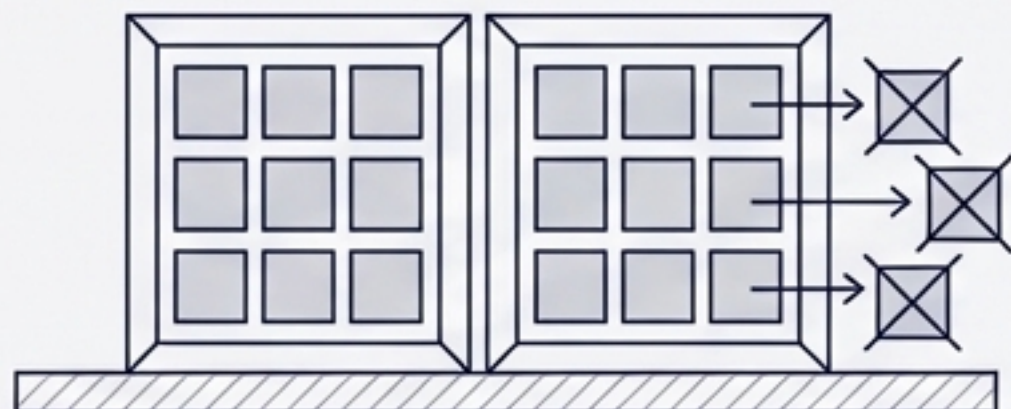
$\text{ROUND}(3.9, 0) \rightarrow 4$

$\text{ROUNDDOWN}(3.9, 0) \rightarrow 3$

$=\text{ROUNDDOWN}(1299, -2) \rightarrow 1200$ (捨去到百位)



實務應用：
預算控管與
庫存計算



一箱只能放 10 件，25 件只能裝滿 2 箱（剩下的不算）。

$=\text{ROUNDDOWN}(25/10, 0) \rightarrow 2$ 。
確保資源絕不超發。

三大核心函數：實體化比較矩陣

函數與行為	商業思維	絕對情境
ROUND: 看 5 才決定進捨	追求公平⚖️	獎金發放、常規財報
ROUNDUP: 無條件往上變大	公司利益導向📈	運費計算、定價策略
ROUNDDOWN: 無條件往下變小	風險極小化控管🛡️	預算不超支、實體庫存

知識降維打擊：從「位階」到「倍數」



如果你需要將數字進位到「5 的倍數」或「每 500 元一個基數」？

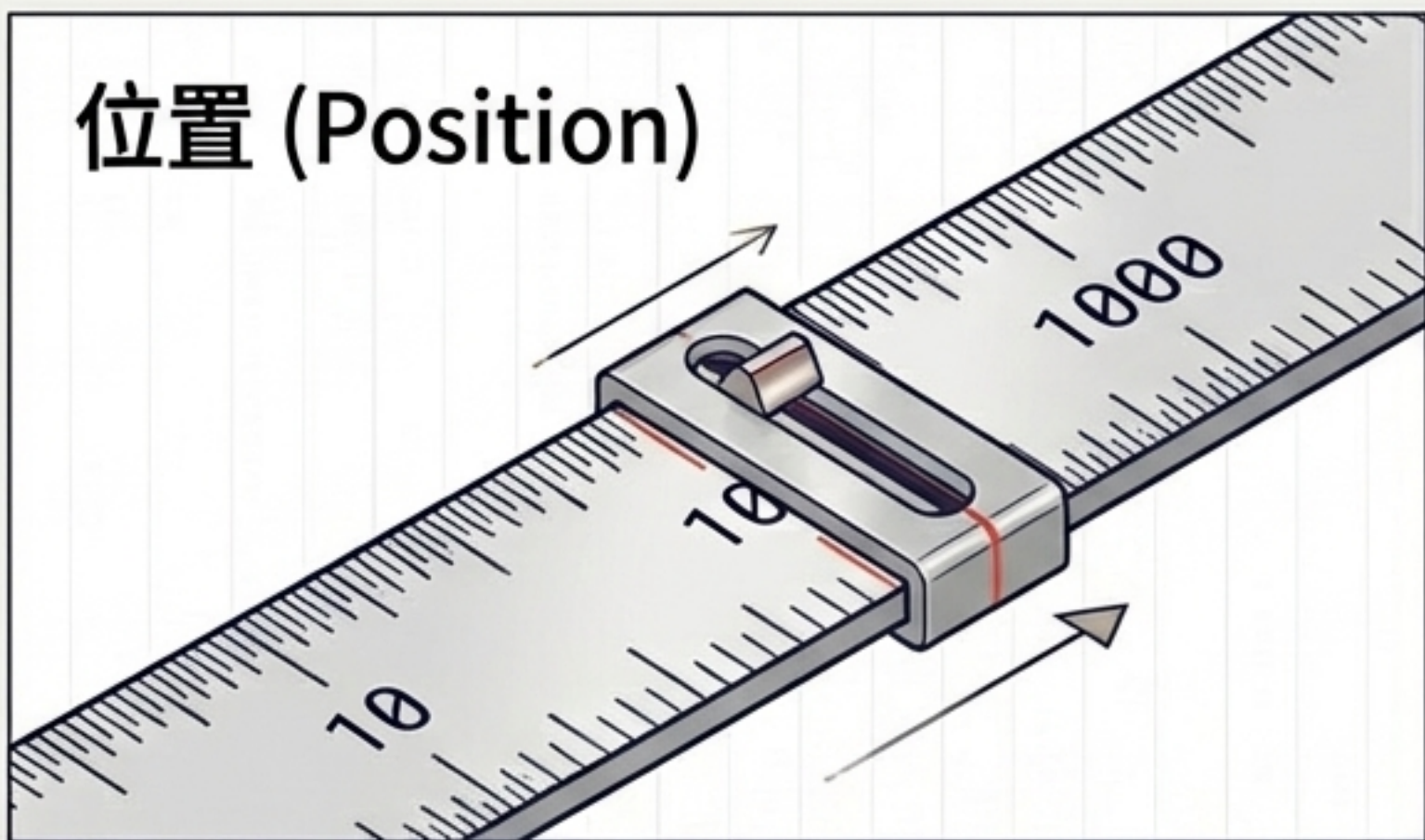
ROUNDUP 只能依據「十位、百位」等絕對位置 (Digit) 進行操作。無法直接達成自訂間距。

導入 CEILING.MATH

⇒ 以特定倍數 (Multiple) 為基準的高階進位。

底層邏輯對比：位置 vs. 倍數

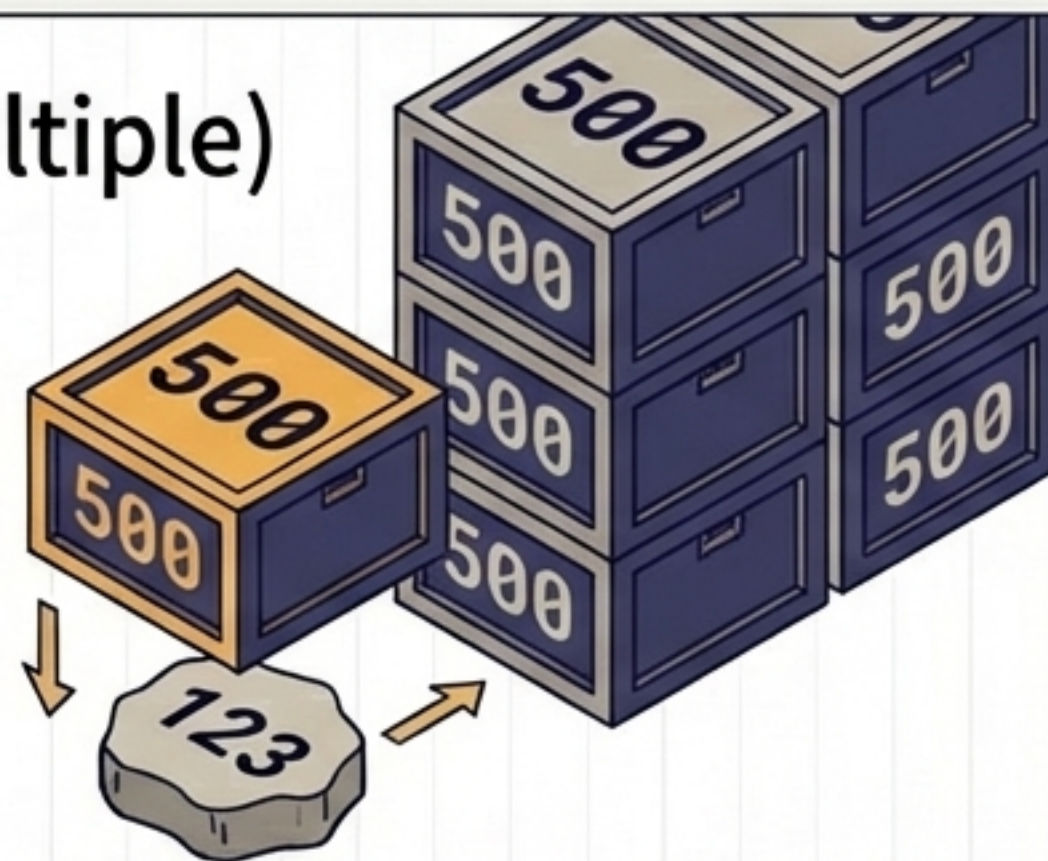
位置 (Position)



ROUNDUP (數值, 位數)

⇒ 對齊尺規刻度

倍數 (Multiple)



CEILING.MATH (數值, 倍數)

⇒ 疊加自訂積木

CEILING.MATH：自訂倍數的極致彈性

`=CEILING.MATH(數值, [倍數], [Mode])`

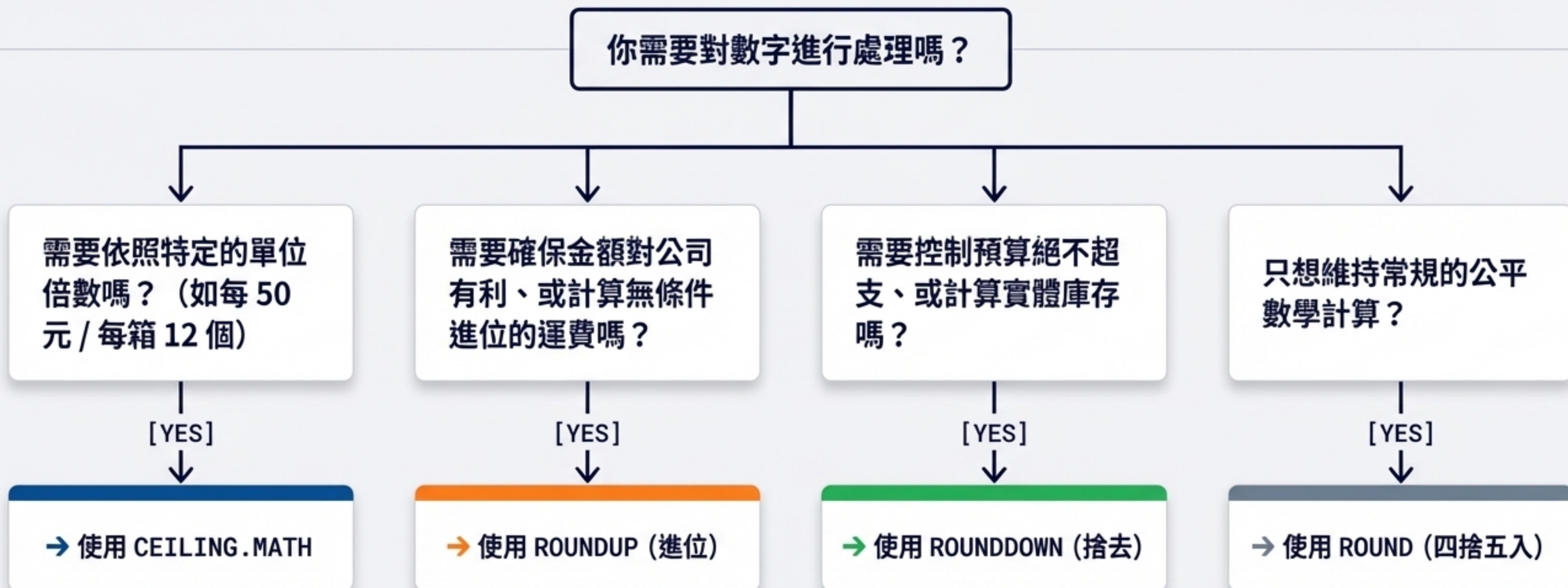
`=CEILING.MATH(123, 10) → 130` (最接近的 10 的倍數)

🎯 為何它更強大？

- 1. 發放獎金以 500 為單位：`=CEILING.MATH(獎金, 500)`
- 2. 預設行為：若不輸入倍數，預設向上捨入至最接近整數（倍數=1）。
- 3. 負數控制 (Mode)：可精細決定要「朝向零」或「遠離零」進位，滿足複雜財務合規。



終極決策樹：一秒選對數值控制武器



記住，Excel 的重點不是算數，而是「你想要什麼結果」。