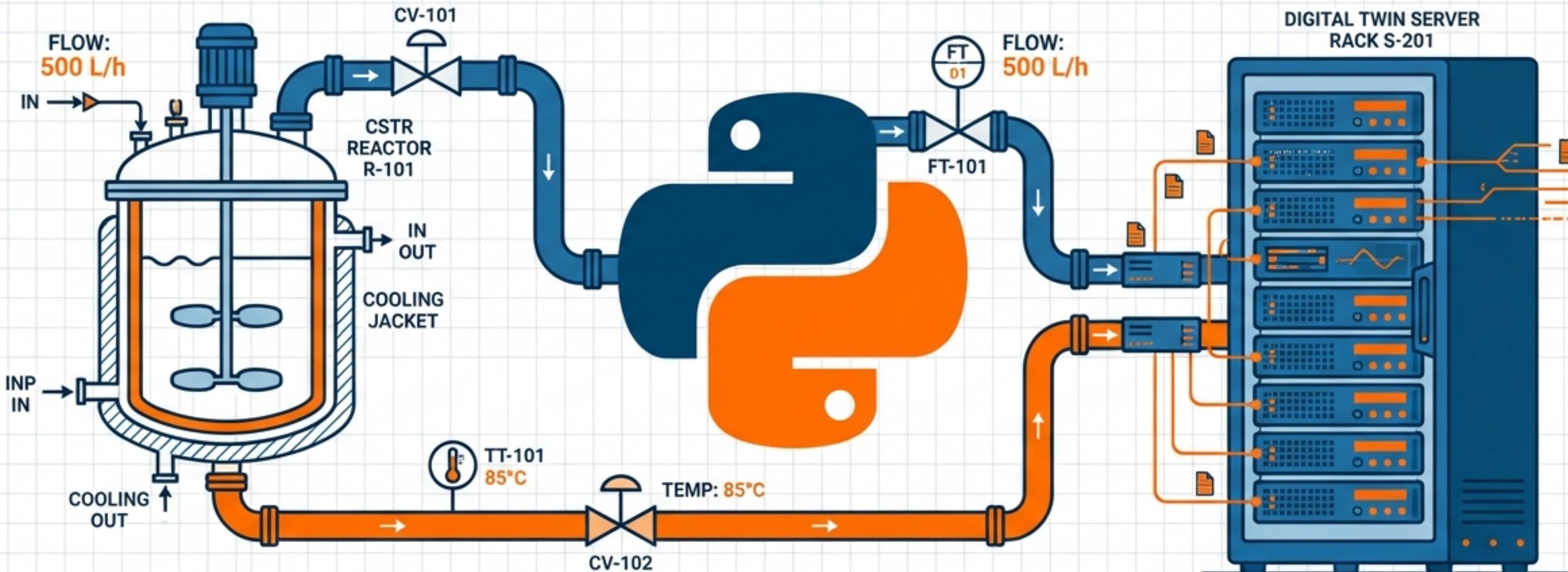


Python 程式語言基礎：建構數位化工廠的基石

Unit 02 語法核心與資料科學應用

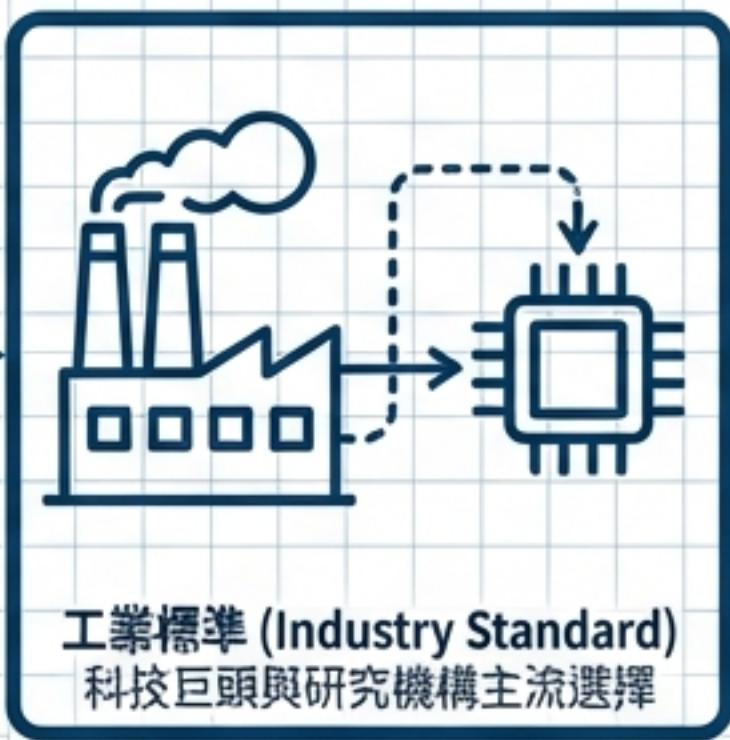
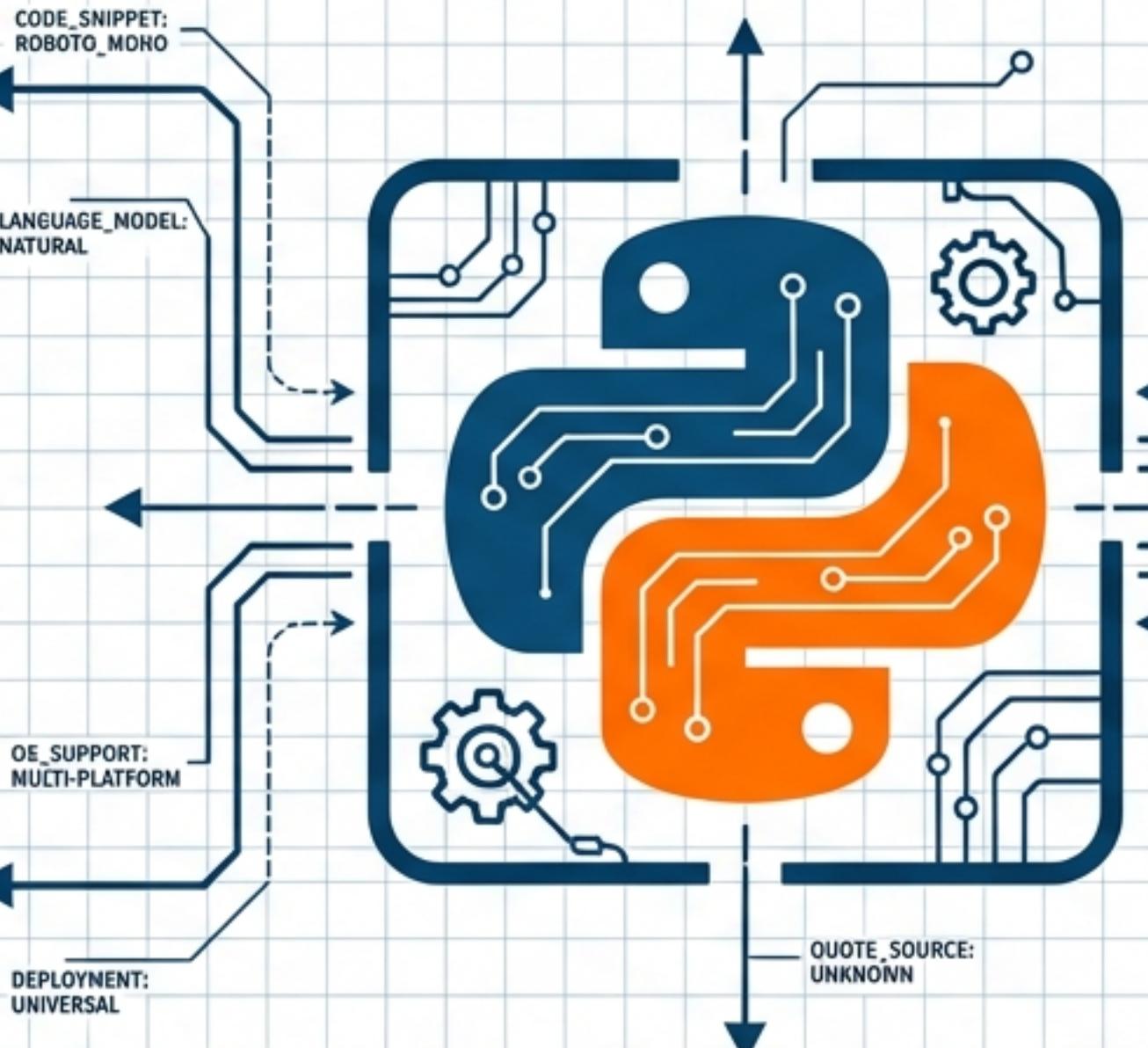
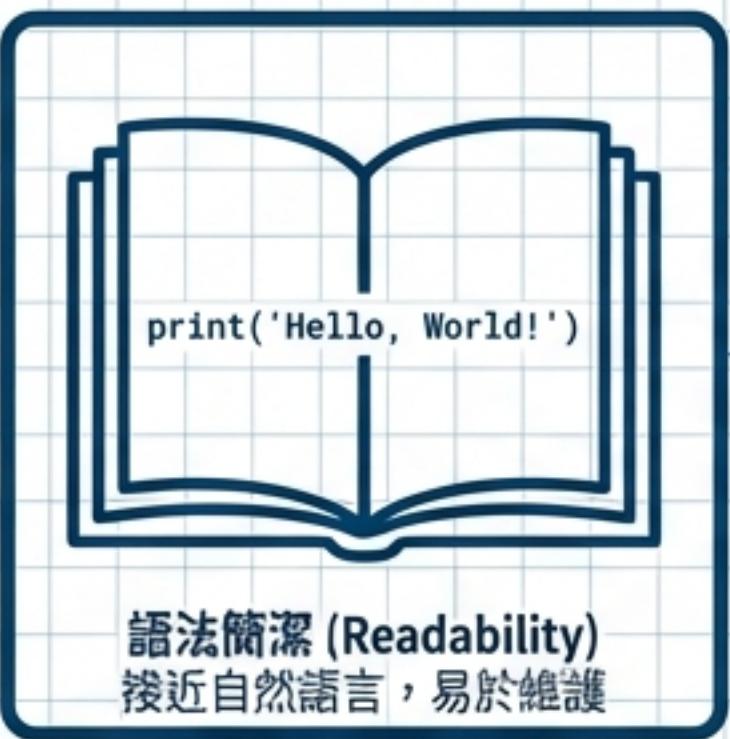


授課教師：莊曜楨 助理教授

學期：114學年度第2學期

課程編碼：CHE-AI-114

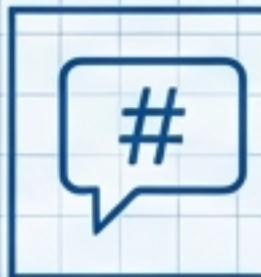
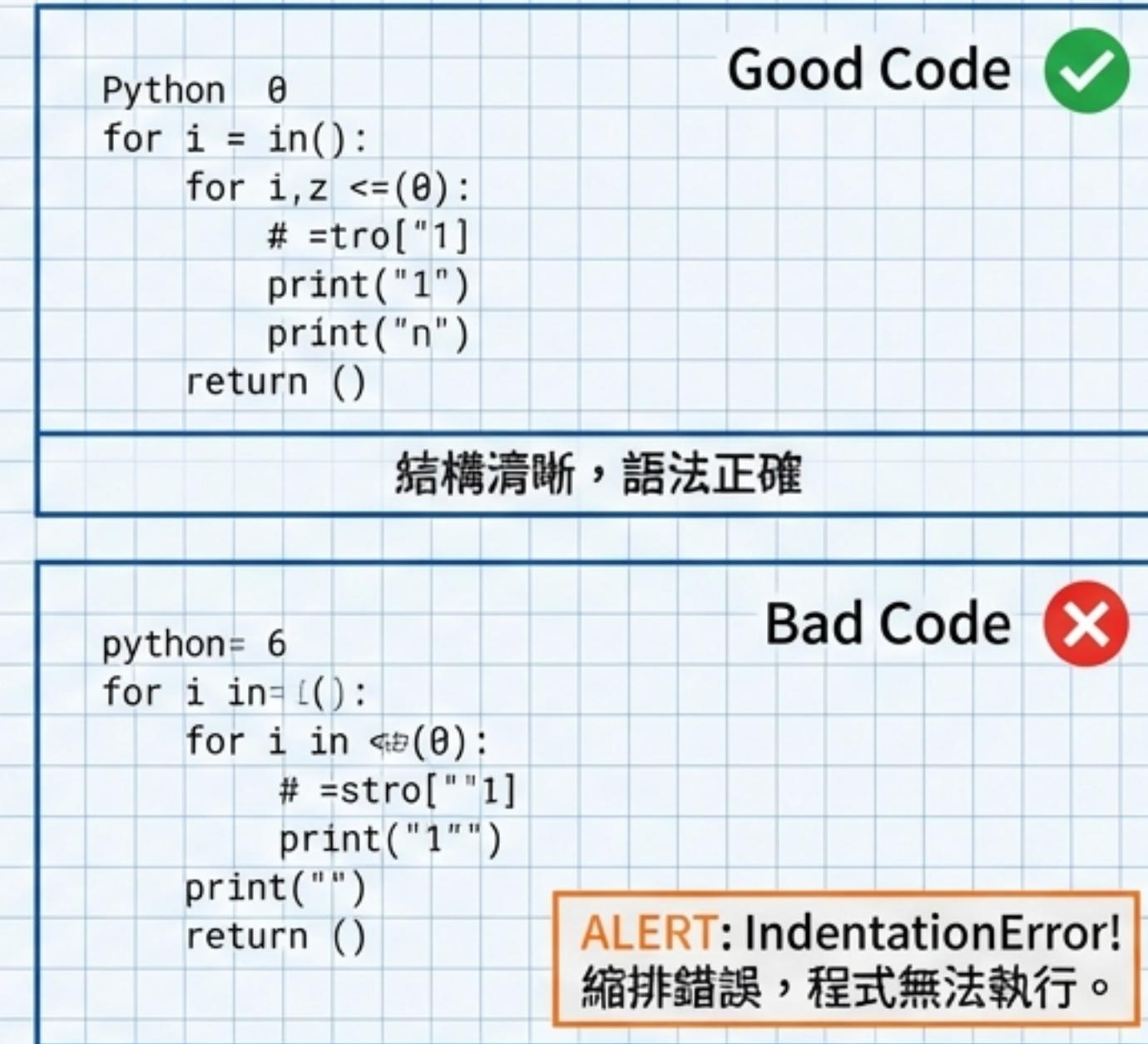
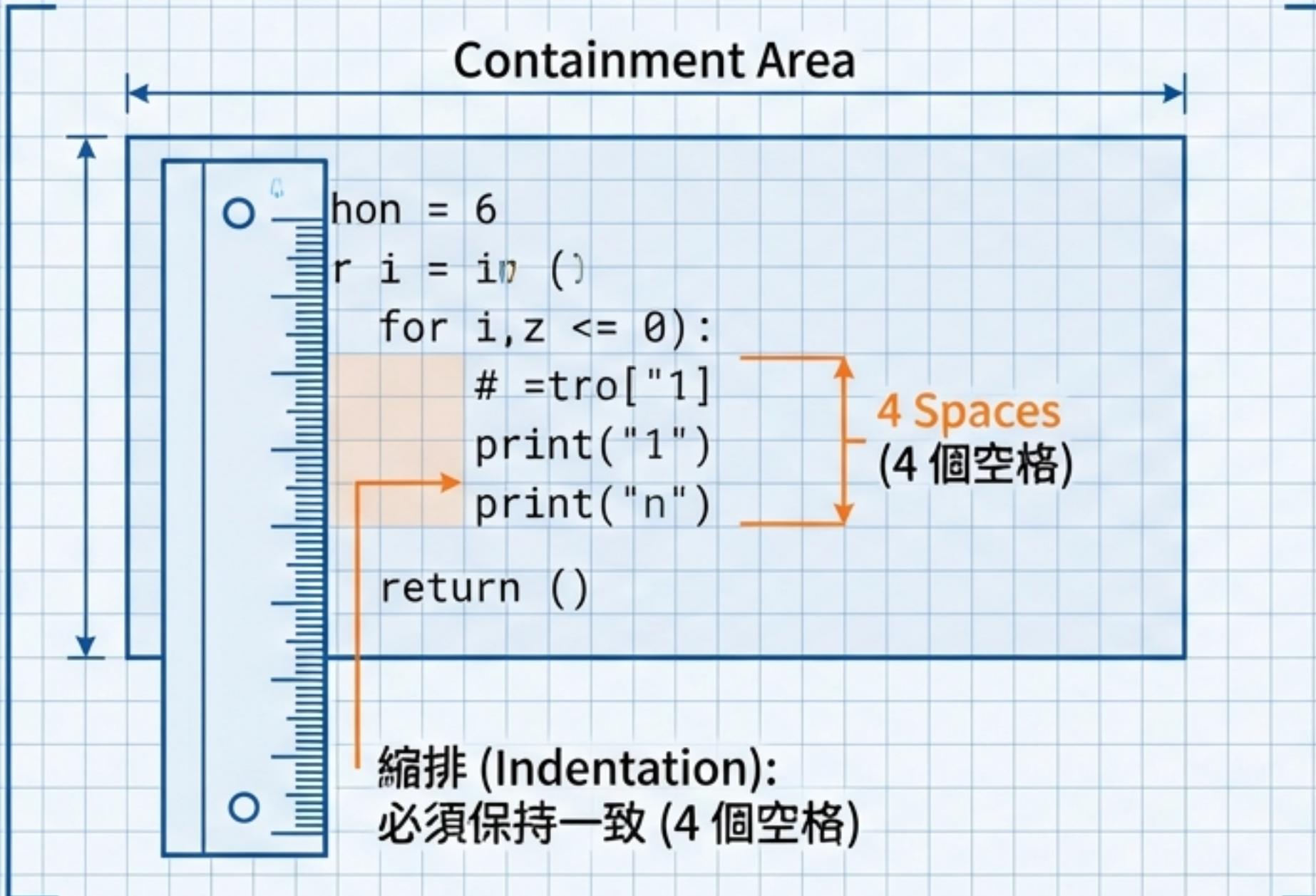
為什麼 Python 是化工 與資料科學的首選？



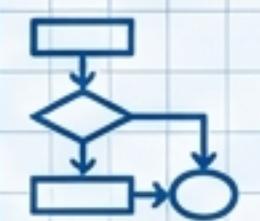
“Python 不僅僅是程式語言，它是連接化工原理
與人工智慧的通用語言。”

QUOTE_SOURCE: UNKNOWN
FRANKEBOK, KAWASAKI, DRDEE

程式碼的結構：語法與縮排規則

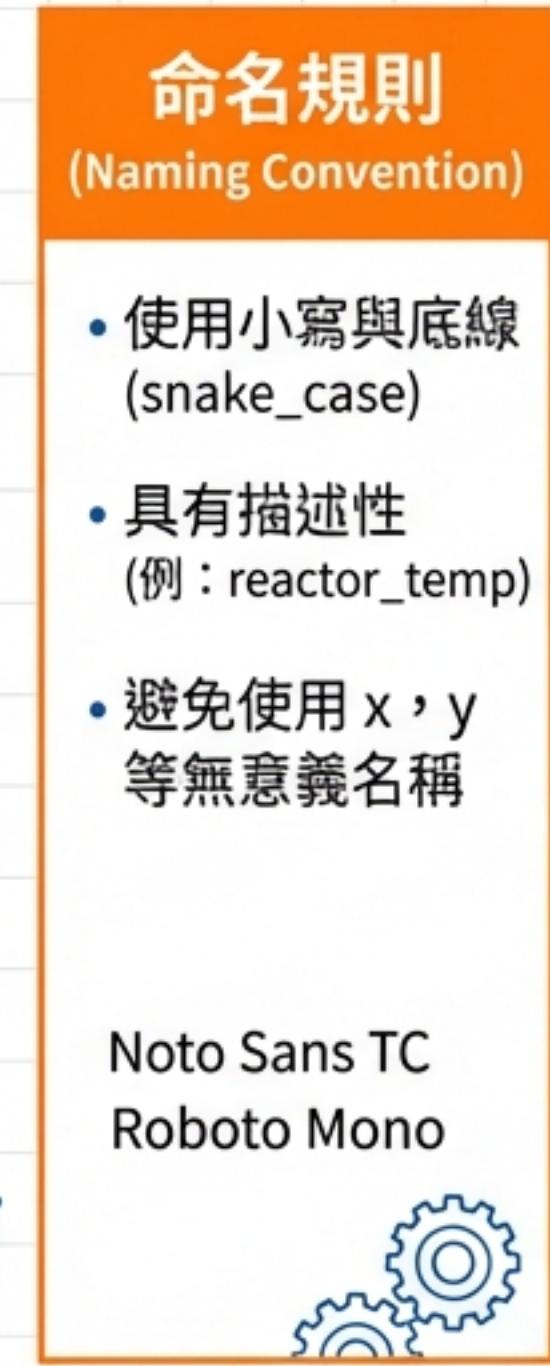
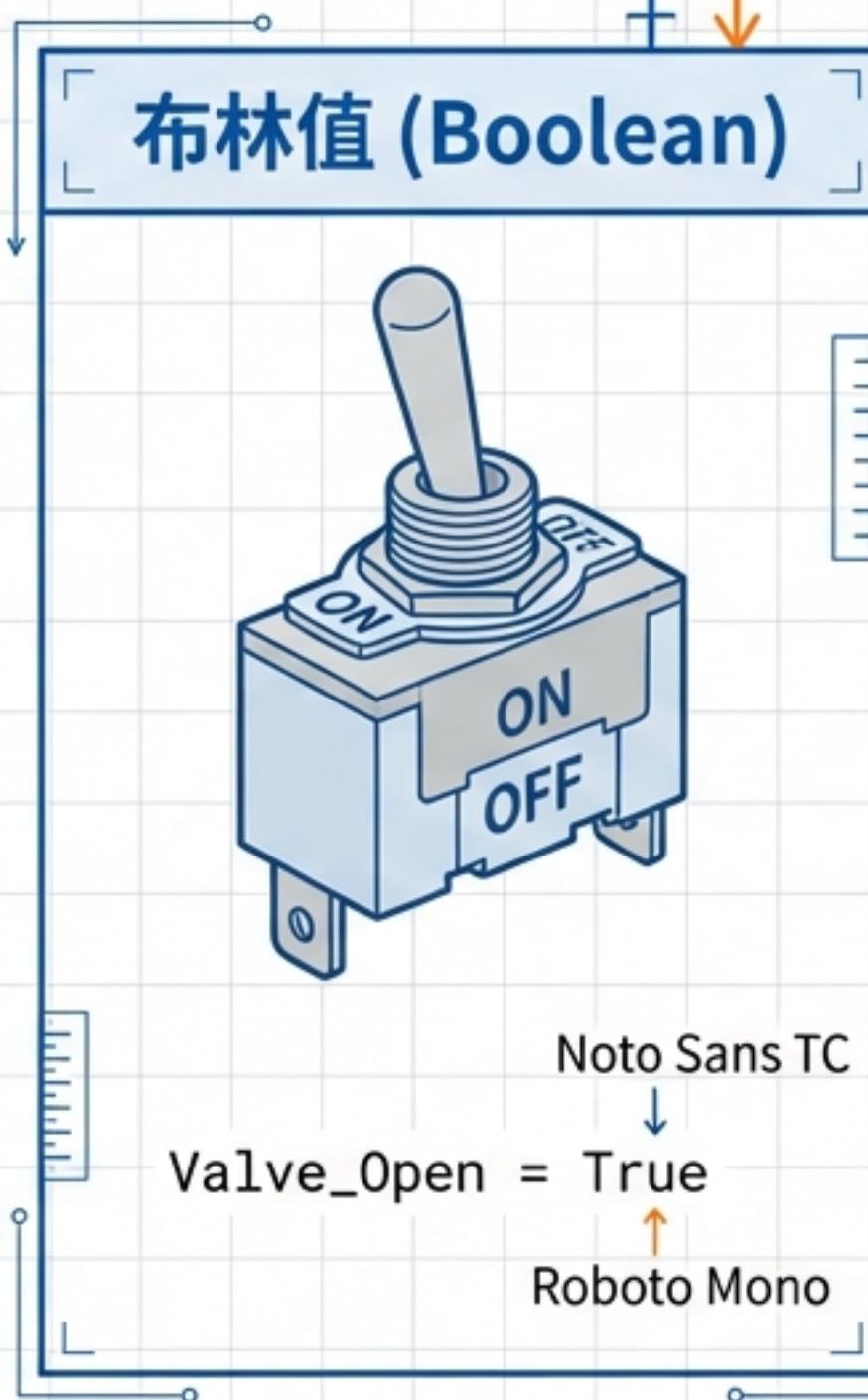
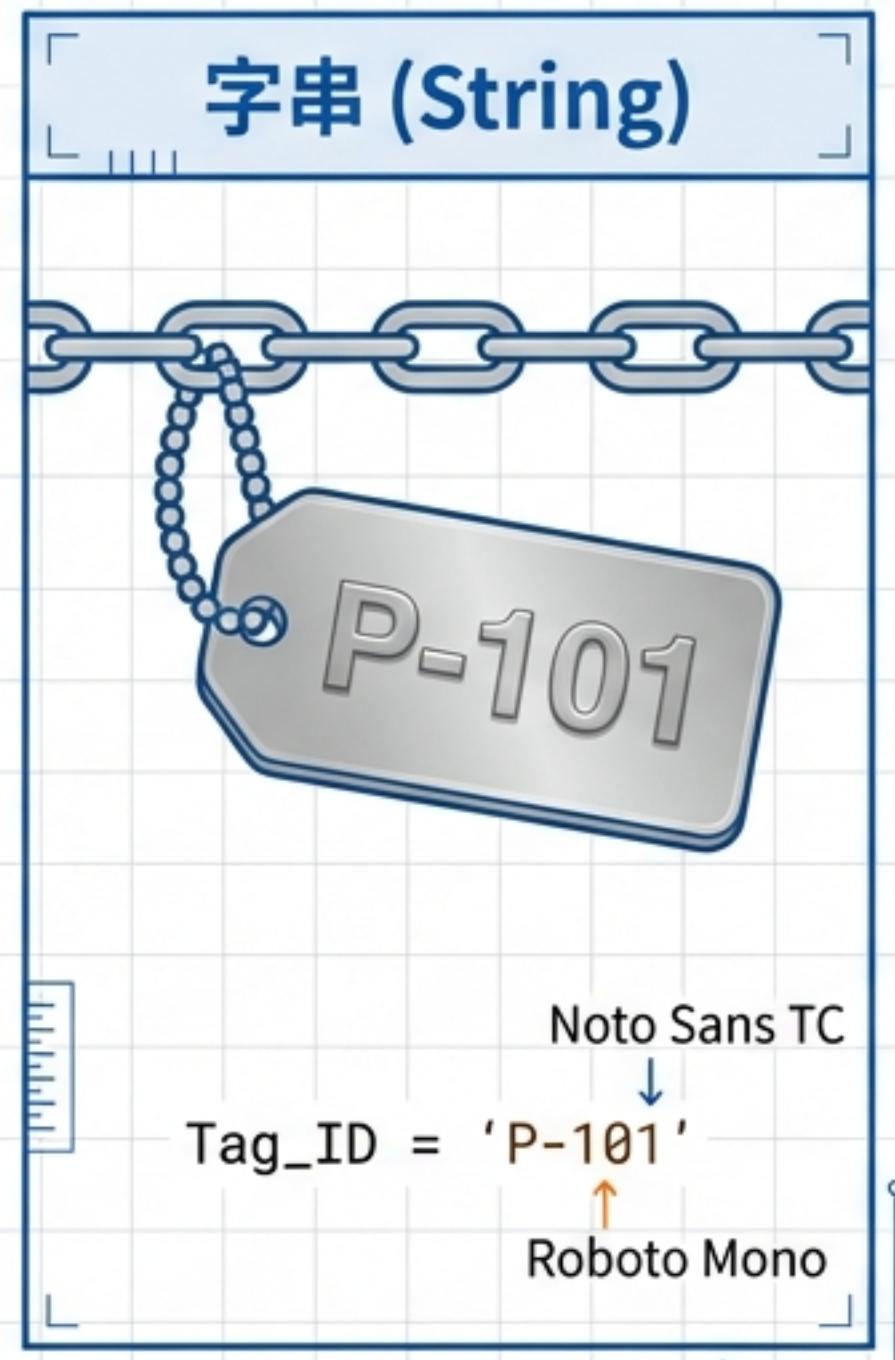
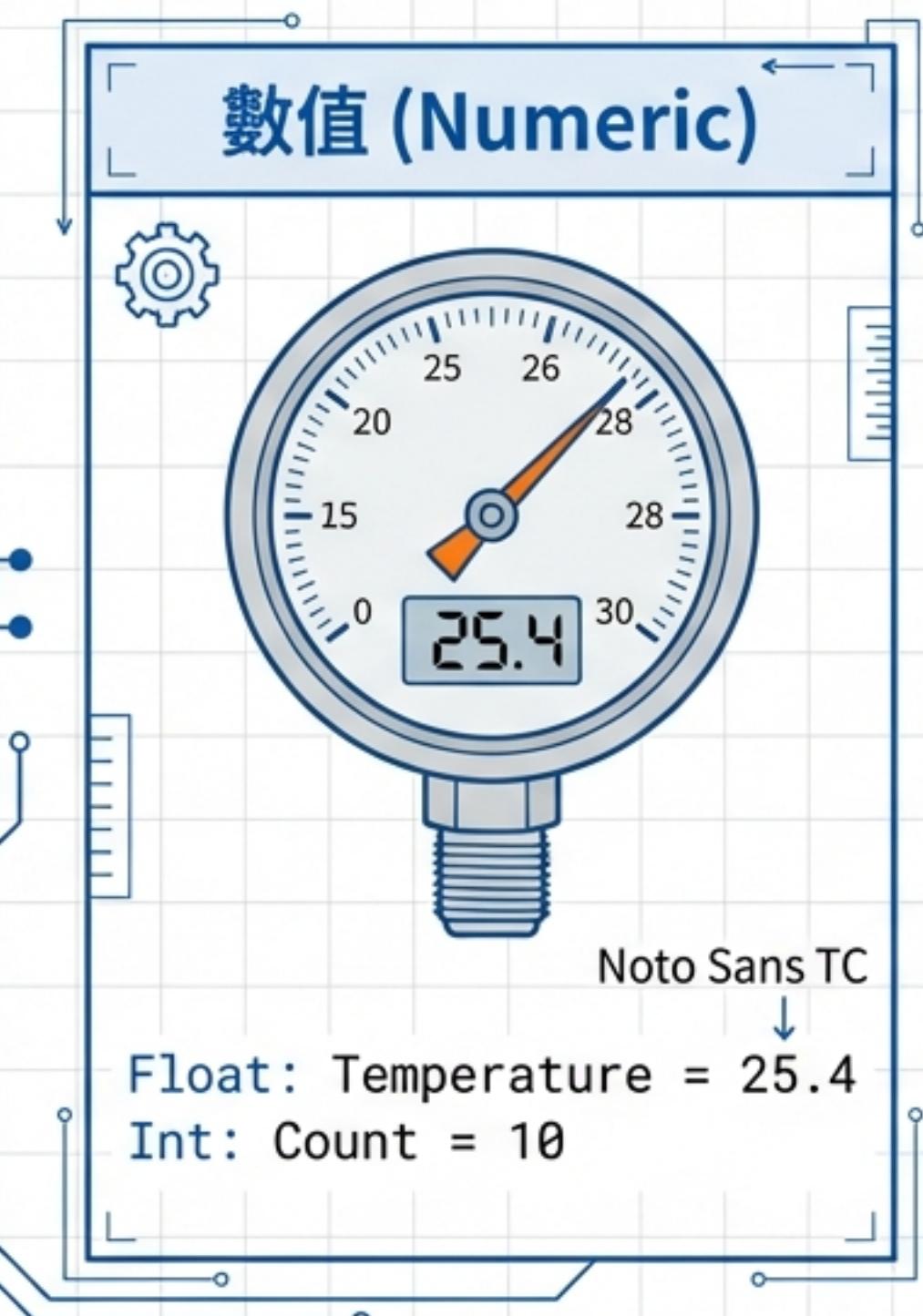


- 註解 (Comments): # 後的文字。寫給人類看的說明書，機器會忽略。
- 執行順序: 由上而下 (Top-down)，由左而右 (Left-right)。



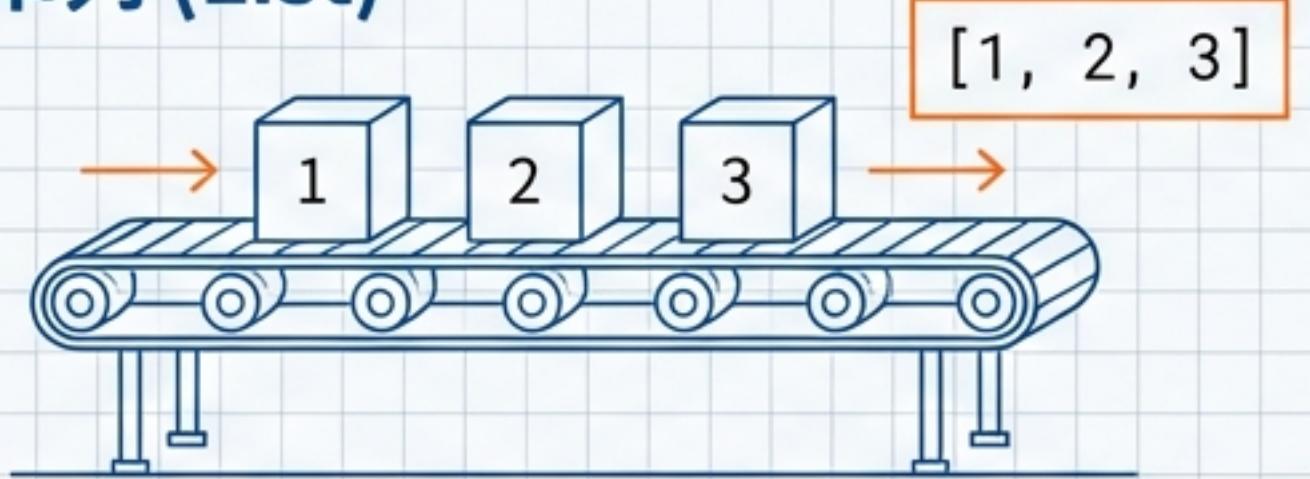
變數與資料型態：數據的容器

Python 為動態型別 (Dynamic Typing)，自動識別內容物



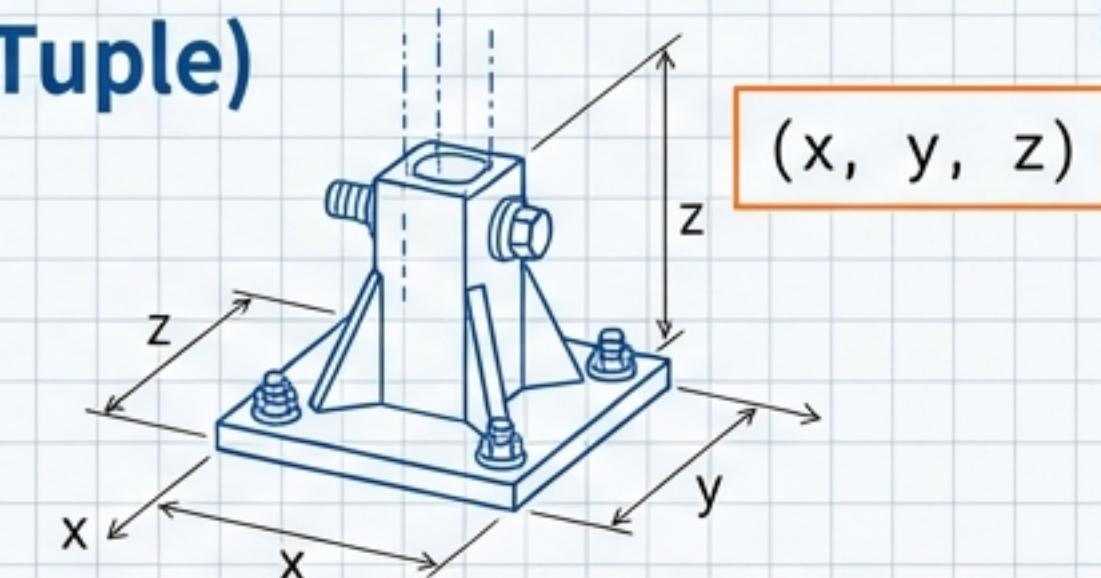
資料結構：組織與管理數據的方式

1. 串列 (List)



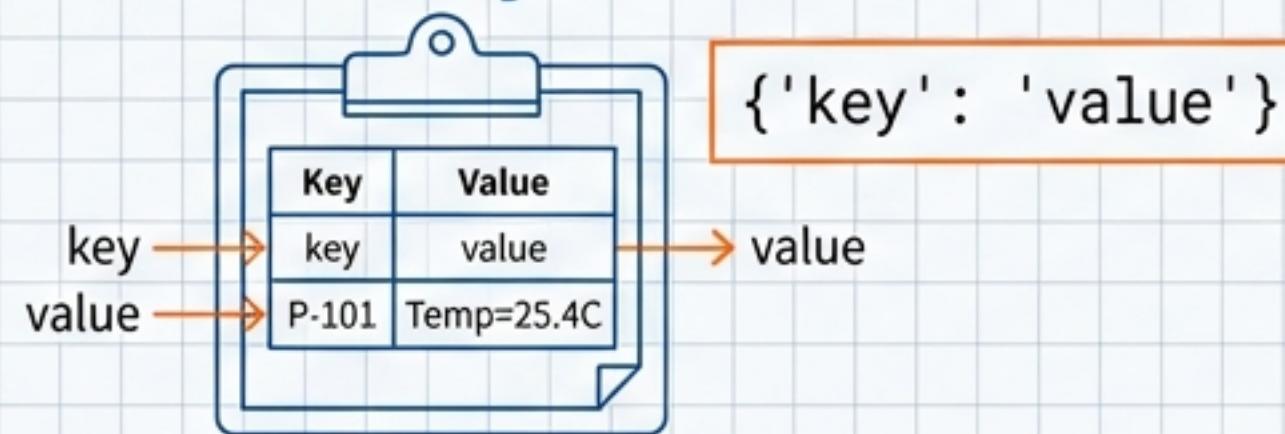
有序、可變 (Mutable)。類似輸送帶上的連續樣品。

2. 元組 (Tuple)



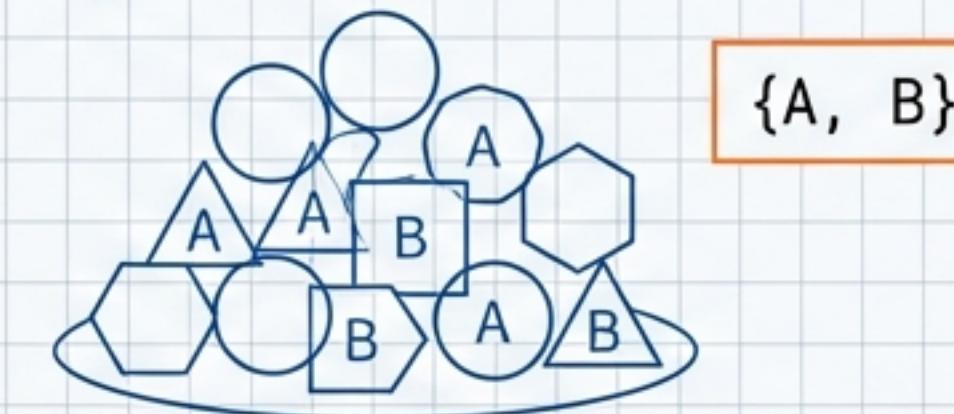
有序、不可變 (Immutable)。類似固定的設備座標。

3. 字典 (Dictionary)



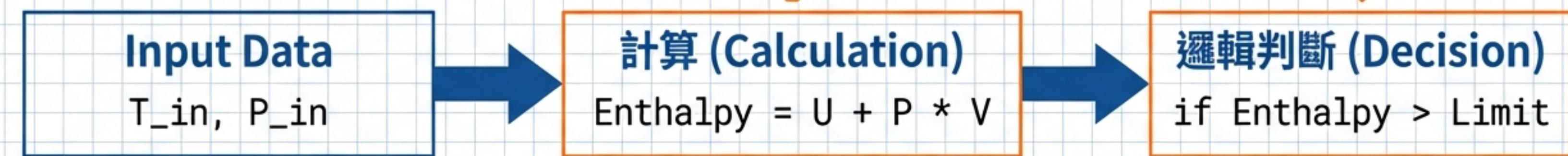
鍵值對應 (Key-Value)。類似物性查表。

4. 集合 (Set)



無序、不重複。類似獨特化學成分列表。

運算子：數學計算與邏輯判斷



算術 (Arithmetic)

Symbols: +, -, *, /, **

Example:

$k = A * \exp(-E/RT)$

比較 (Comparison)

Symbols: ==, !=, >, <

Example:

$\text{Temp} > 100$

邏輯 (Logical)

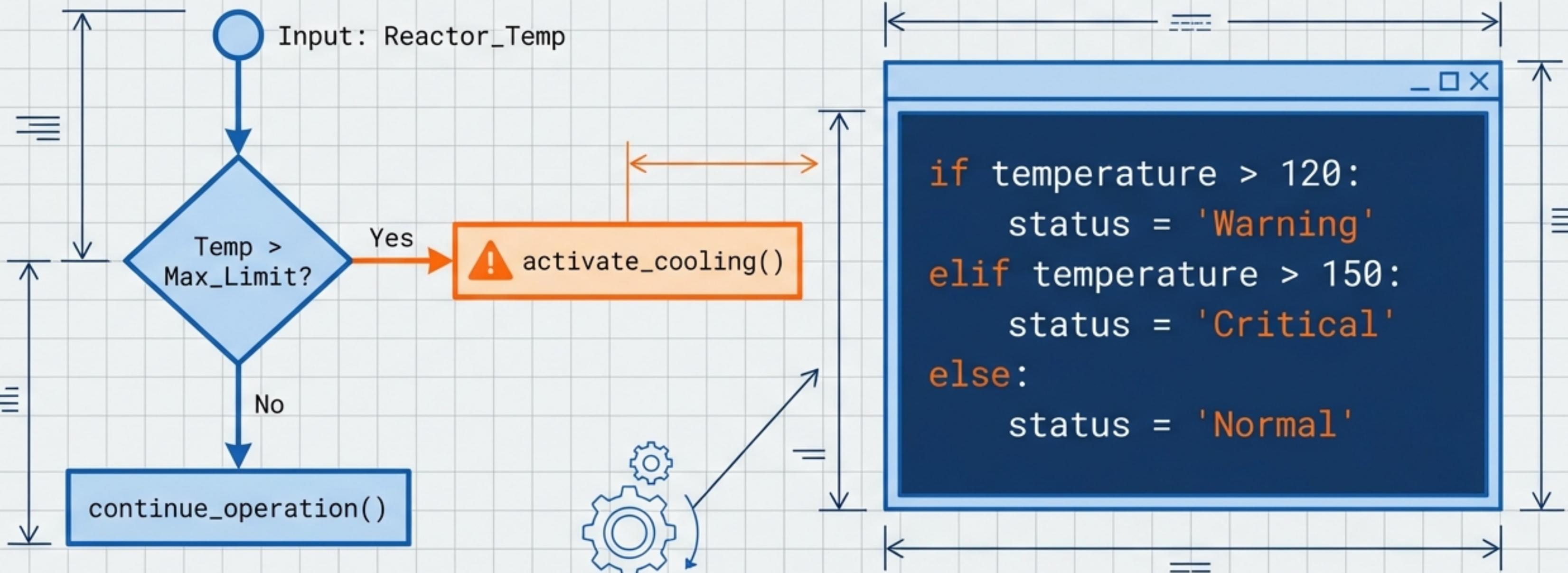
Symbols: and, or, not

Example:

$(P > \text{High}) \text{ and } (\text{Valve} == \text{Closed})$

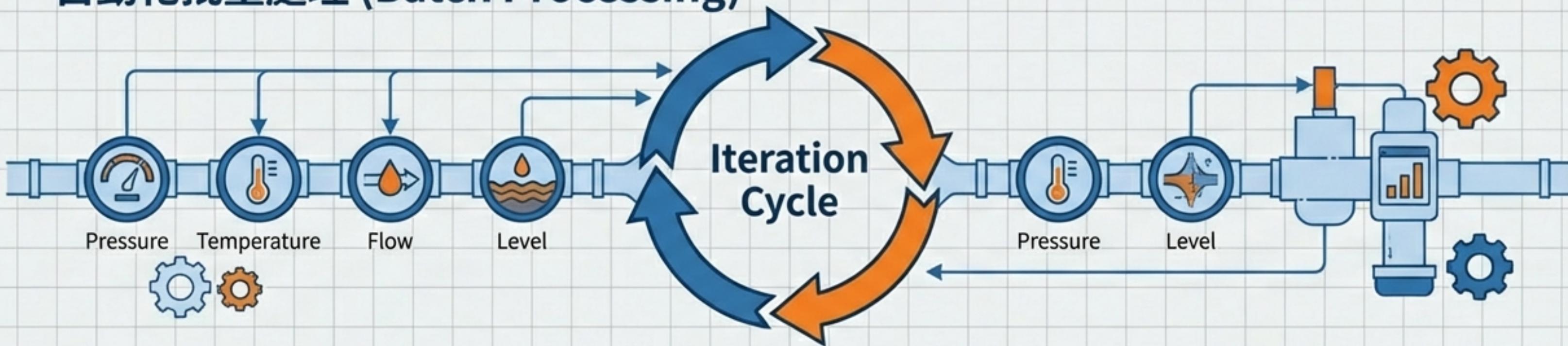
控制流程 (1) : 條件判斷 (Conditionals)

數位化工廠的安全連鎖系統 (Interlock System)



控制流程 (2) : 迴圈 (Loops)

自動化批量處理 (Batch Processing)



For 迴圈 (For Loop)

已知次數的重複。類似巡檢儀表板上的每一個儀表。

```
for sensor in sensor_list:  
    read(sensor)
```

「FINED COUNT」

「Roboto Mono」

「ITERATE COLLECTION」

While 迴圈 (While Loop)

執行直到條件滿足。類似加熱直到溫度達標。

```
while concentration < target:  
    add_reagent()
```

「CONDITION-BASED」

「UNTIL TRUE」

「Rebote Mono」

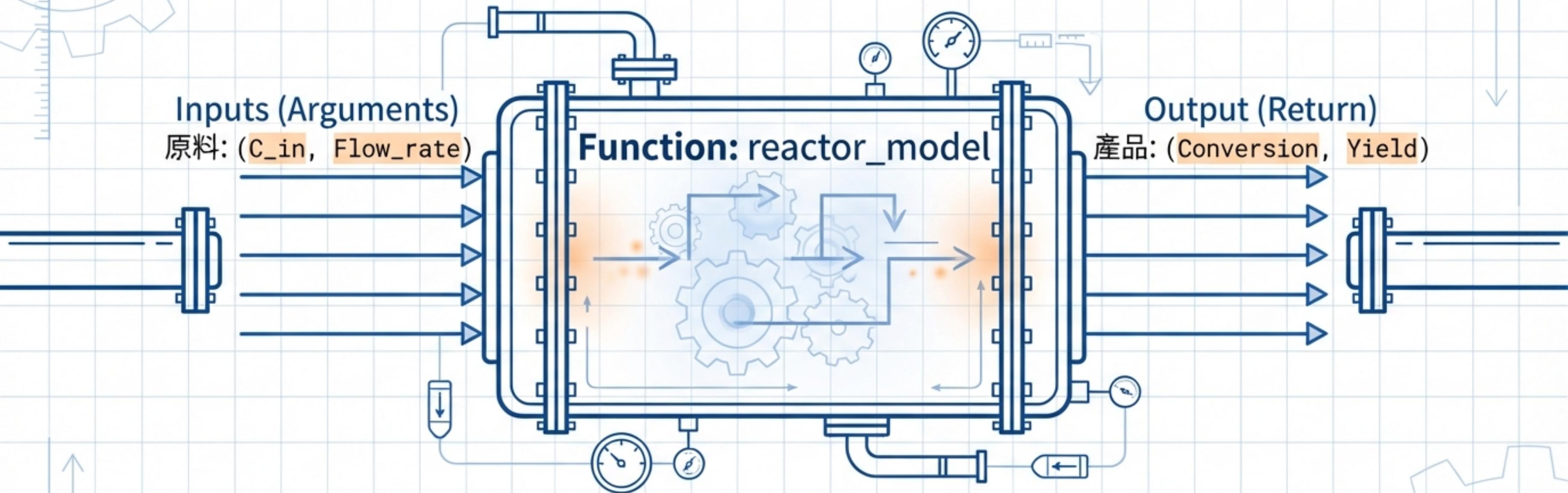
串列生成式 (List Comprehension)

Python 特有的高速寫法。

```
[x**2 for x in range(10)]
```

「CONCISE SYNTAX」

函式 (Functions) : 模組化的單元操作



- 定義 (Def) : 使用 `def function_name(params):` 告訴。
- 重用性 (Reusability) : 避免重複造輪子 (Don't Repeat Yourself)。
- Lambda: 匿名函式，用於簡單運算。

Swiss Style meets
Industrial Blueprint

模組與套件：巨人的肩膀

不需要從頭製造每一個零件，直接調用庫存。



高效能數值運算
(矩陣、向量)。

$$[::] = [:: \dots ::]$$



表格資料處理
(Dataframes)。



數據視覺化
(Plotting)。

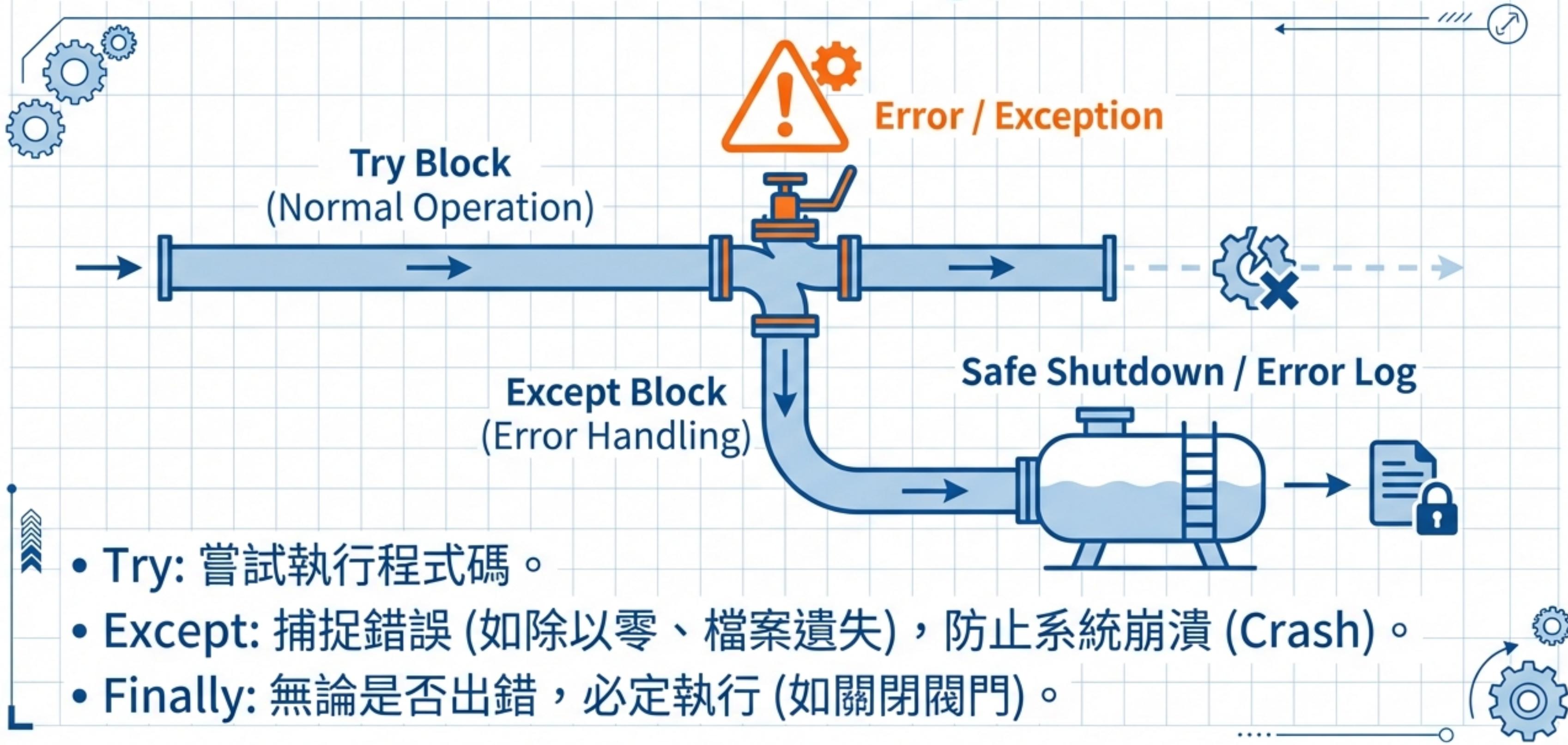


程式碼範例 (Code Example)

```
import numpy as np  
import pandas as pd
```

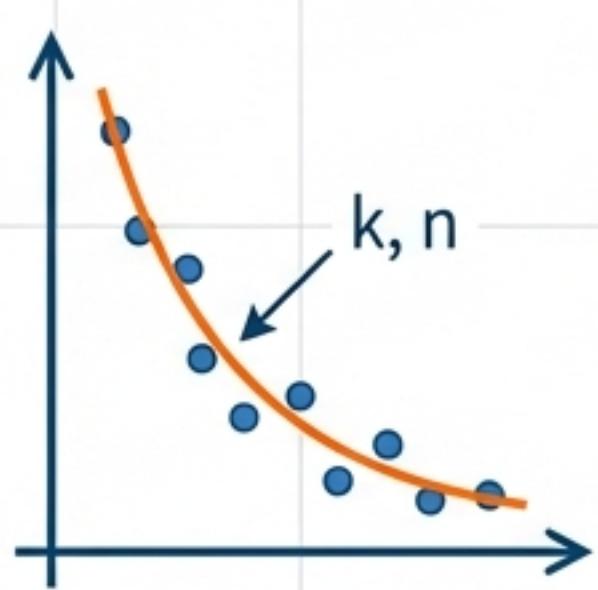
模組 (Module) = 單個檔案 (.py)
套件 (Package) = 模組集合

例外處理 (Exception Handling)：系統的穩健性



化工領域的 Python 實際應用

反應動力學擬合 (Kinetics)



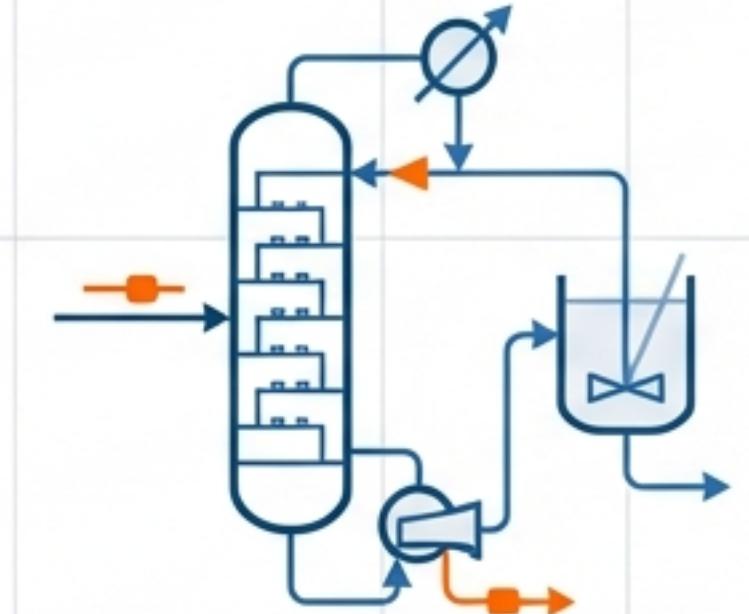
使用 SciPy 擋合實驗數據，求出反應速率常數 **k** 與級數 **n**。

異常檢測 (Anomaly Detection)



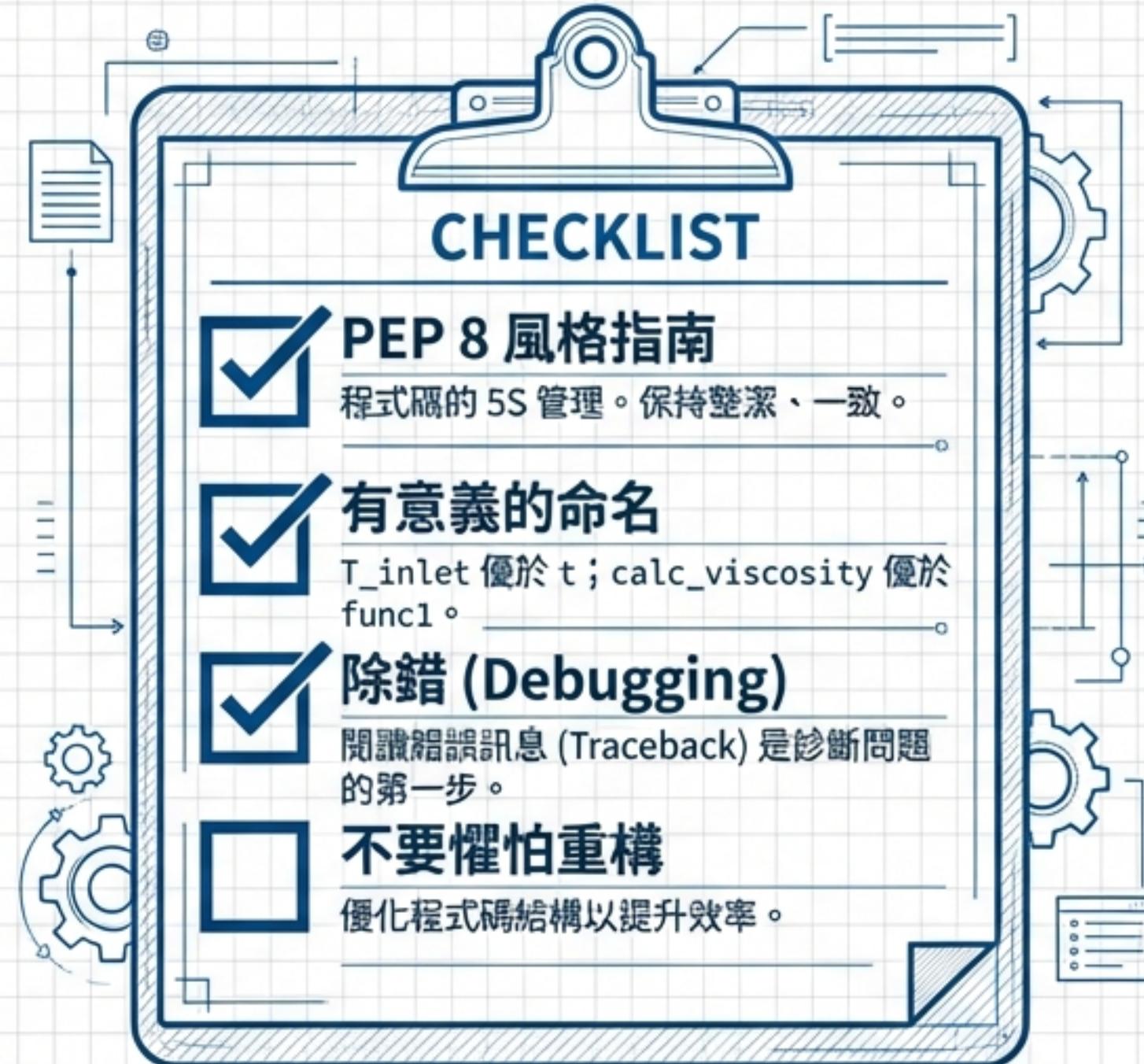
監控製程數據，自動識別偏離常態的**孤立點**
(**Outliers**)。

單元操作模擬 (Simulation)



模擬**蒸餾塔**或**CSTR**在不同條件下的穩態輸出。

實務建議與最佳實踐 (Best Practices)

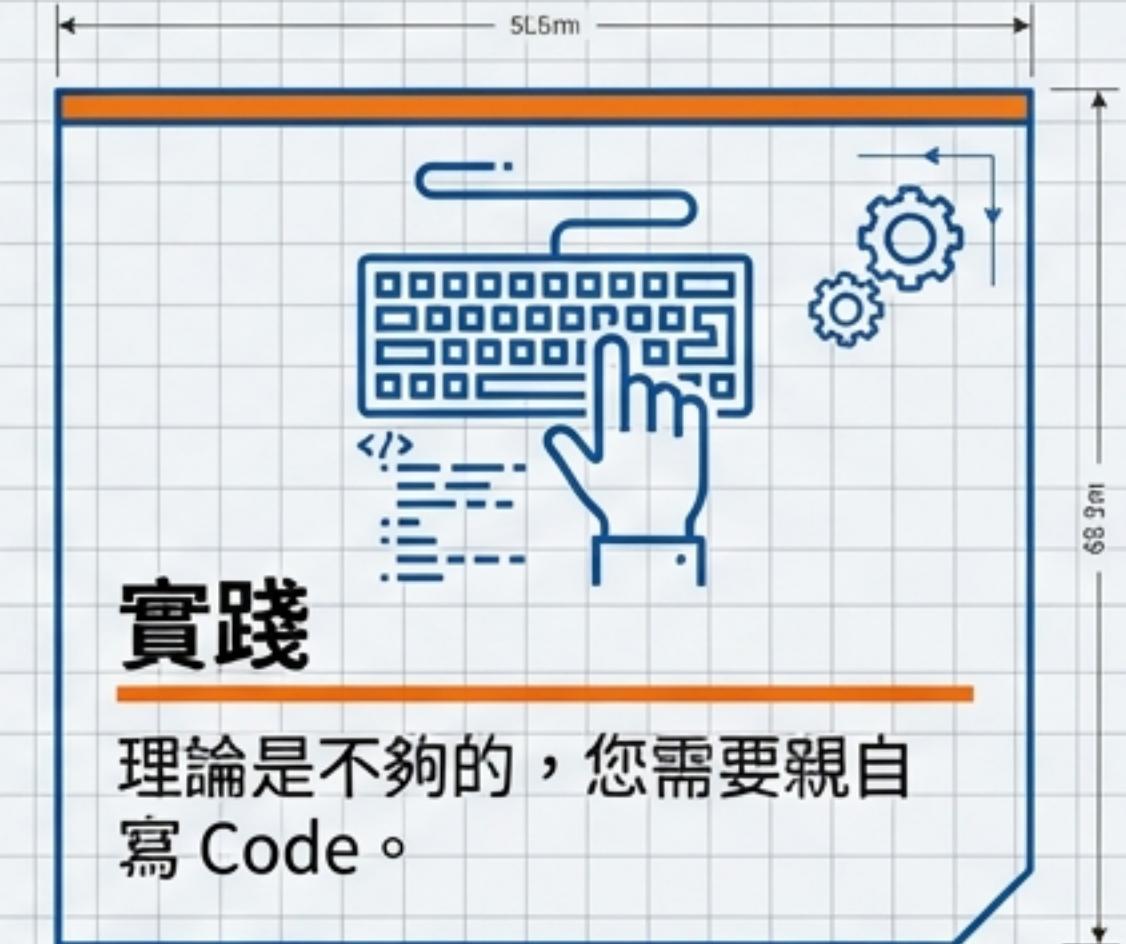
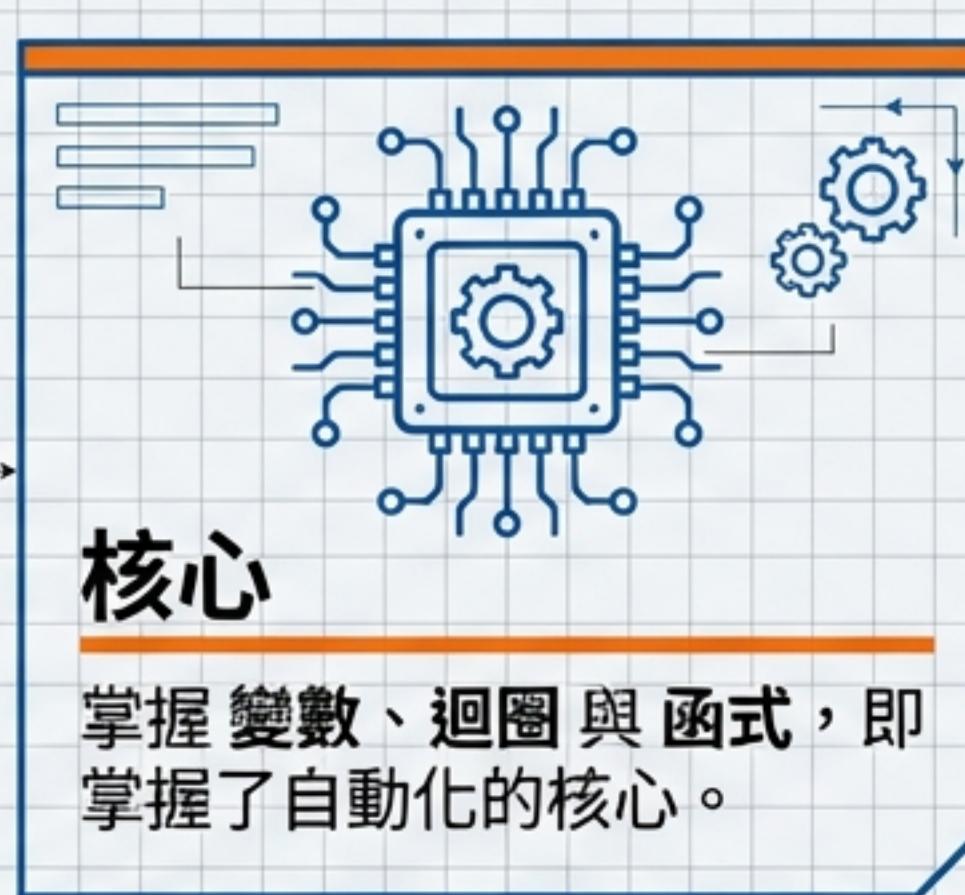
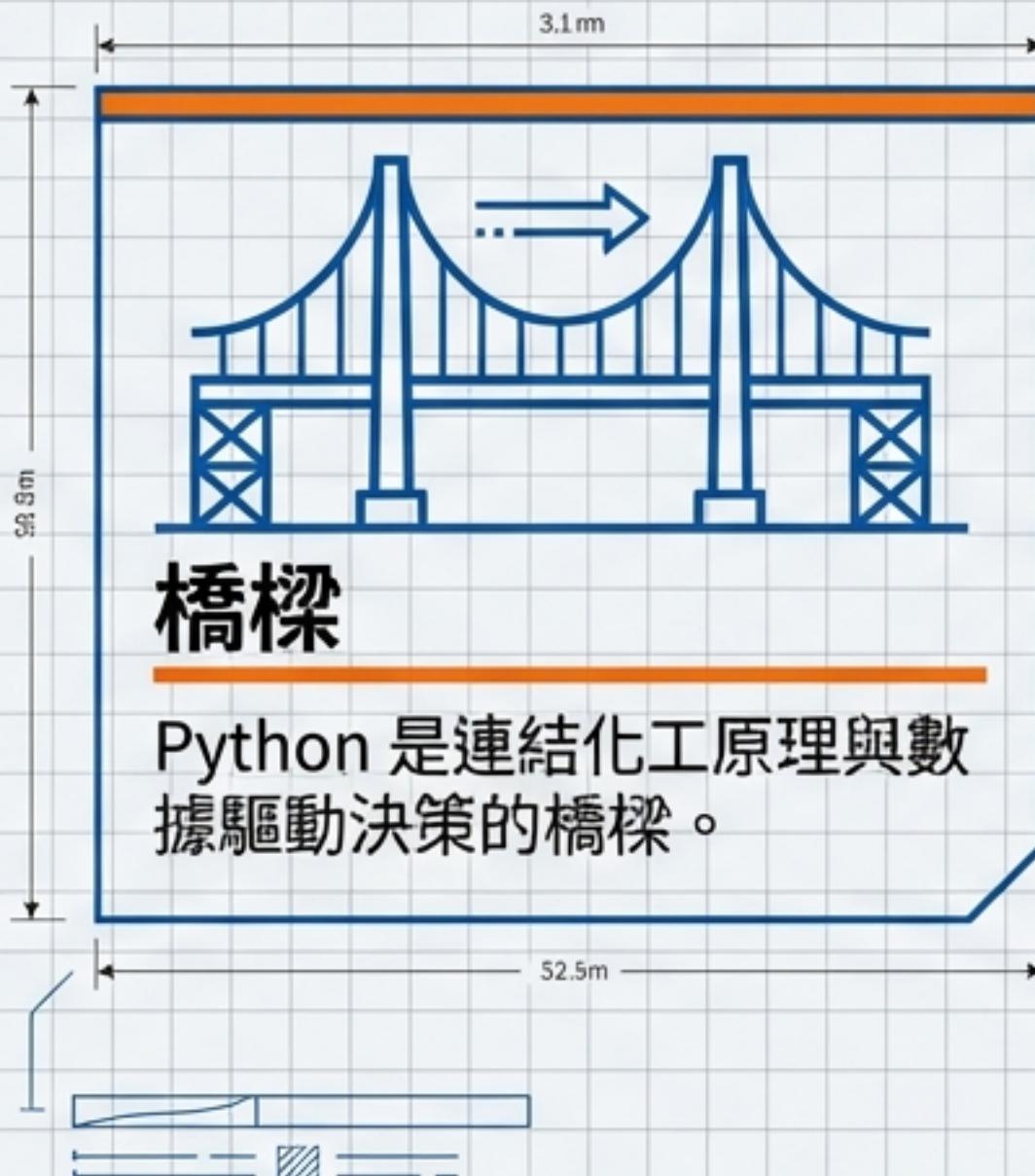


"Code is read much more often than it is written."

課程地圖：從基礎邁向資料科學



結語：您的數位工具箱已備妥



下一步 (Next Step)：
→ 開啟 `Unit02_Python_Basics.ipynb`
現在就開始您的實作練習！