

單元 1 | VIBE CODING 入門

打造你的第一個 AI 協作專案

拆解問題＋環境建置＋驗收思維 (1/6)



講者：Adam & Hao

七號演算股份有限公司





本堂課的冒險目標



“Vibe Coding 的核心不是拼命寫 Code，
而是精準提問與驗收。”



思維升級

學會將模糊需求拆解為IPO
(輸入 → 處理 → 輸出)。



建立標準

寫程式前，先定義「怎
樣才算成功」(驗收思
維)。



動手實作

完成環境建置，跑通
一個最小案例。



VIBE CODING 開發流派



WEB 免安裝

ChatGPT, Claude, Gemini,
Google AI Studio



IDE 介面

Cursor, Antigravity



命令列工具

Claude CLI, Gemini CLI

實戰：機台異常通報 MVP

情境 (SCENARIO)

當機台溫度過高時，通知工程師。



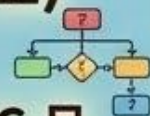
Input (輸入)

設備狀態數據
(溫度值)、
警報代碼。



Process (處理)

判斷邏輯
(例如：溫度 $> 80^{\circ}\text{C}$ 且
持續 5 秒)。



Output (輸出)

通知內容
(Line / Email / 戰情室)。



核心心法 2：驗收思維 (ACCEPTANCE)

為什麼要先寫驗收條件？



- 避免 AI 產生幻覺 (Hallucination)。
- 確保功能符合航道 (實際需求)。

黃金法則：3 X 3



實戰演練：定義驗收標準

機台異常通報驗收清單



成功 (Success)

- 輸入 85°C → 觸發警報，輸出 "溫度超過警戒值"。
- 輸入 60°C → 不觸發警報，輸出 "正常"。
- 輸入警報碼 E01 → 查表對應，輸出 "馬達過載"。



失敗 (Failure)

- 輸入空值 (Null) → 回傳 "No Data"。
- 輸入非數字 → 程式報錯並提示格式錯誤。
- 網路斷線 → 無法發送通知時，將錯誤寫入 Log 檔。



WANTED

AI 時代的開發迴圈



Prompt (提示)
提供 IPO 與驗收標準



Generate (產出)
AI 生成程式碼



Iterate (迭代)
修正錯誤，優化代碼



Verify (驗證)
人工檢查 6 個驗收案例



實作任務清單

將大目標切成 30 分鐘的小任務

1 環境準備

1. 安裝 VS Code / Claude Code。
2. 建立專案資料夾結構。

2 模擬機台

- 產生隨機溫度與錯誤碼。
- 驗收：**能印出運轉數據。

3 邏輯與輸出

- 加入 If Temp > Threshold。
- 驗收：**通過 3x3 測試案例。

今日總結 & 作業



總結



需求來了，先畫 IPO。



寫 Code 前，列 3 好 3 壞。



任務切小，確保能
驗證。



課後作業

挑選工作中的一個小痛點，寫下它的 IPO
拆解與 3x3 驗收案例。

