

單元 2 | Web 小工具快速上線

從 IPO 邏輯驗證到 Git 版本控管 (1/13)

講者/單位：Adam & Hao / 七號演算股份有限公司

課程目標

 **核心目標**：透過 AI (Vibe Coding) 快速解決工作上的「痛點」。

 **本週產出**：一個專屬的 Web 小工具，並學會如何管理它、分享它。

 **關鍵技術**：HTML/CSS/JS, Git, GitHub Pages, Vibe Coding, TDD。



Web 工具的兩種型態



靜態網頁 (Static)

特點：HTML+CSS+JS 在瀏覽器端執行，免後端。

優勢：部署簡單 (GitHub Pages)、零成本、快。

適用：計算機、檢核表。



動態網頁 (Dynamic)

特點：需要資料庫與後端伺服器 (PHP, Node.js 等)。

適用：會員系統、大型電商、大量資料儲存。

別急著寫 Code！先定義 IPO 與驗收標準

Vibe Coding 守則：AI 是工程師，你是 PM。

 I (Input)：輸入什麼？(如：價格、優惠碼)

 P (Process)：邏輯是什麼？(如：總價 = 單價 * 數量)

 O (Output)：結果呈現？(如：顯示金額)

 AC (驗收標準)：明確定義「怎樣才算做完」。



導入 TDD (測試驅動開發) 思維

- 💡 什麼是 TDD？先寫測試 (Test)，再寫程式 (Code)。
- 💡 為什麼需要？避免 AI 產出邏輯有洞的程式碼。

實戰：電商折扣計算器

Input: 購物車總額 \$1200，會員等級 "VIP"，優惠券代碼 "SUMMER_SALE"。

Process (規則):

1. VIP 會員打 9 折。
2. "SUMMER_SALE" 優惠券折抵 \$100。
3. 規則：先計算會員折扣，再扣除優惠券。
4. 規則：折扣後金額若低於 \$0，則為 \$0。

Output: 最終金額

驗收標準：

case 1 : $1200 * 0.9 - 100 = 980$ (正確) ✓

case 2 : $1200 - 100 * 0.9 = 990$ (錯誤邏輯，要防範) ✗





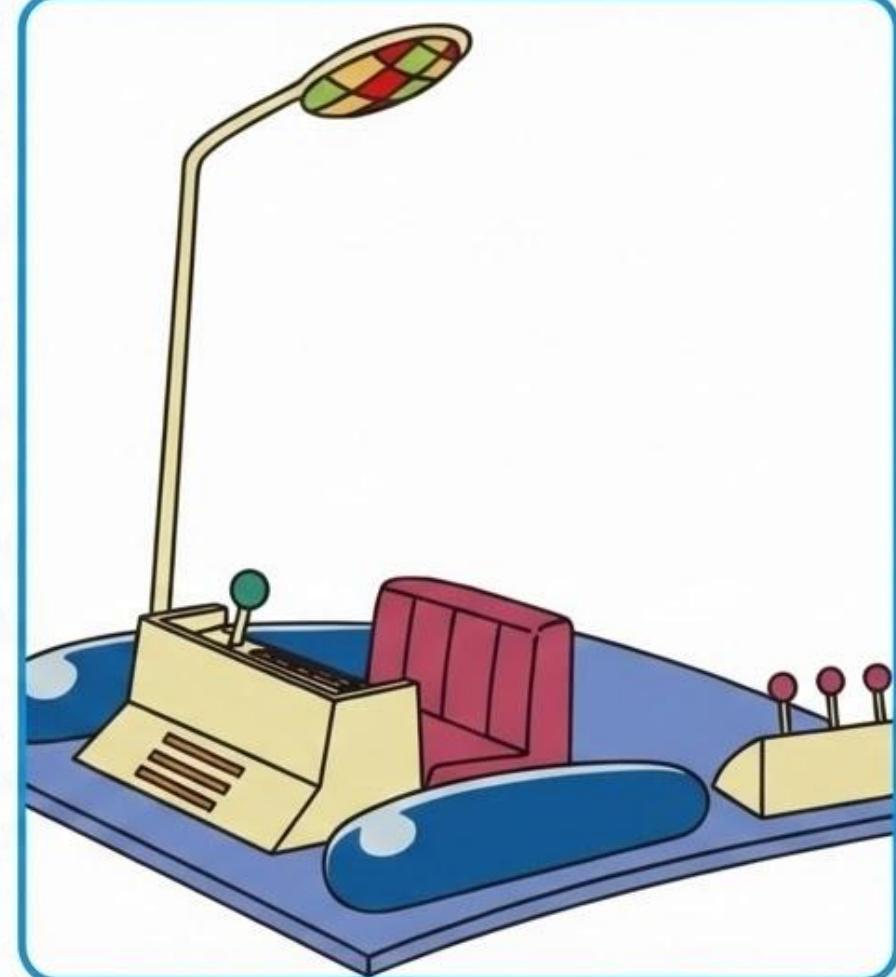
你的時光機：Git 與 GitHub Desktop

為什麼需要 Git？

- 紀錄每次修改的「原因」。
- 最重要的功能：當你（或 AI）把程式改壞時，
可以一鍵回到過去。

實作步驟：

Init (初始化) → Commit (提交) → Push (推送)。



版本的迭代 (Iteration) - 新增需求

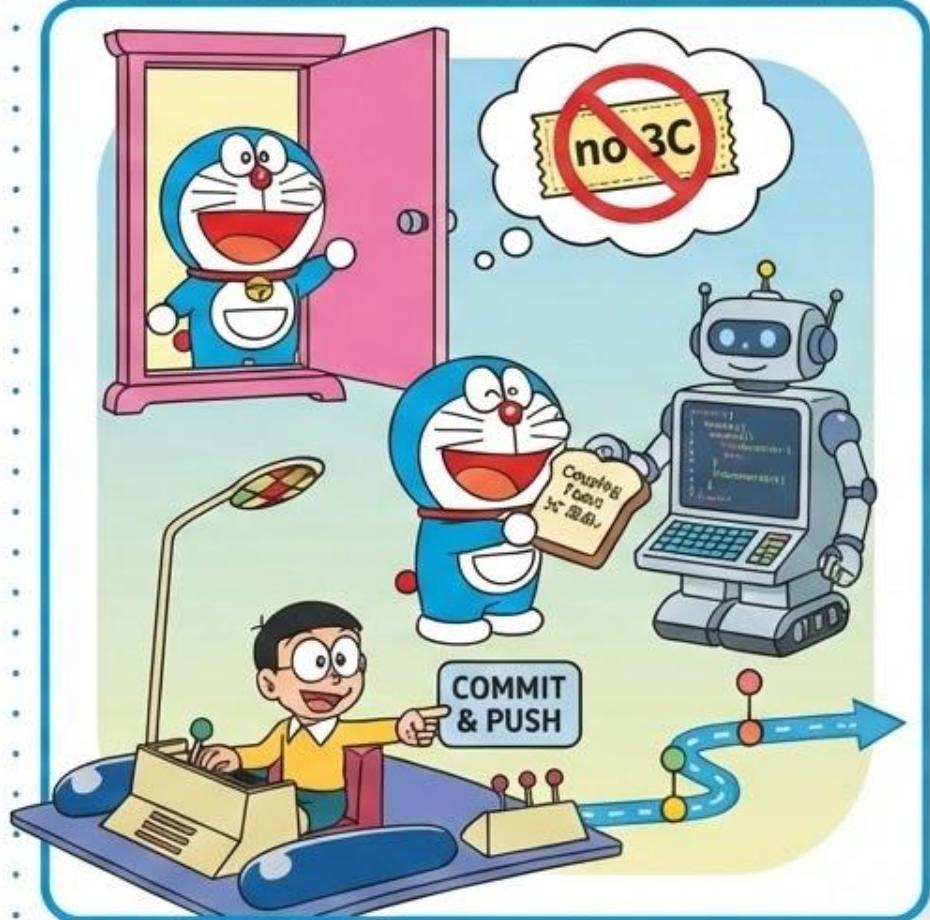
情境：老闆追加需求 → 「優惠券不能用於 3C 產品」。

💡 Vibe Coding 流程：

- 💡 將新規則餵給 AI。
- 💡 AI 更新 JS 邏輯。

💡 Git 動作：

Commit 訊息寫 Feat : 新增 3C 產品排除優惠券邏輯 → Push。



錯誤演示與「退版」救援

情境 (故意犯錯) :

- 讓 AI 加功能，結果所有價格變 0 元 (Bug)。
- 不小心 Commit 並 Push 了壞掉的版本。

救援行動：

- GitHub Desktop 找上一個正常 Commit。
- 右鍵 "Revert changes in commit"。

學習點：有版本控制，大膽嘗試！



從計算器到生產力工具 - 巡檢表單

痛點：紙本易遺失、難統計、
計、作蘿擣、字跡潦草。



解法：
響應式 (RWD) Web 表單。



Vibe Coding 技巧 (Input)：
直接拍照或複製貼上表格欄位給 AI。

Prompt : 「請將這份巡檢項目的文字，轉換為 HTML 表單，需有
checkbox、簽名欄位，及『匯出為文字』按鈕。」



整合功能與部署

⚙ 功能整合：加入「一鍵複製」或「產生 Line 訊息格式」(方便手機回報)。

☁ 部署到 GitHub Pages：

1. Settings → Pages。
2. Source 選 main branch。
3. 獲得專屬網址，發送給同事測試。



為什麼要變成 Chrome Extension ?

Web 工具的缺點：

每次都要開分頁、找網址、
切換視窗。

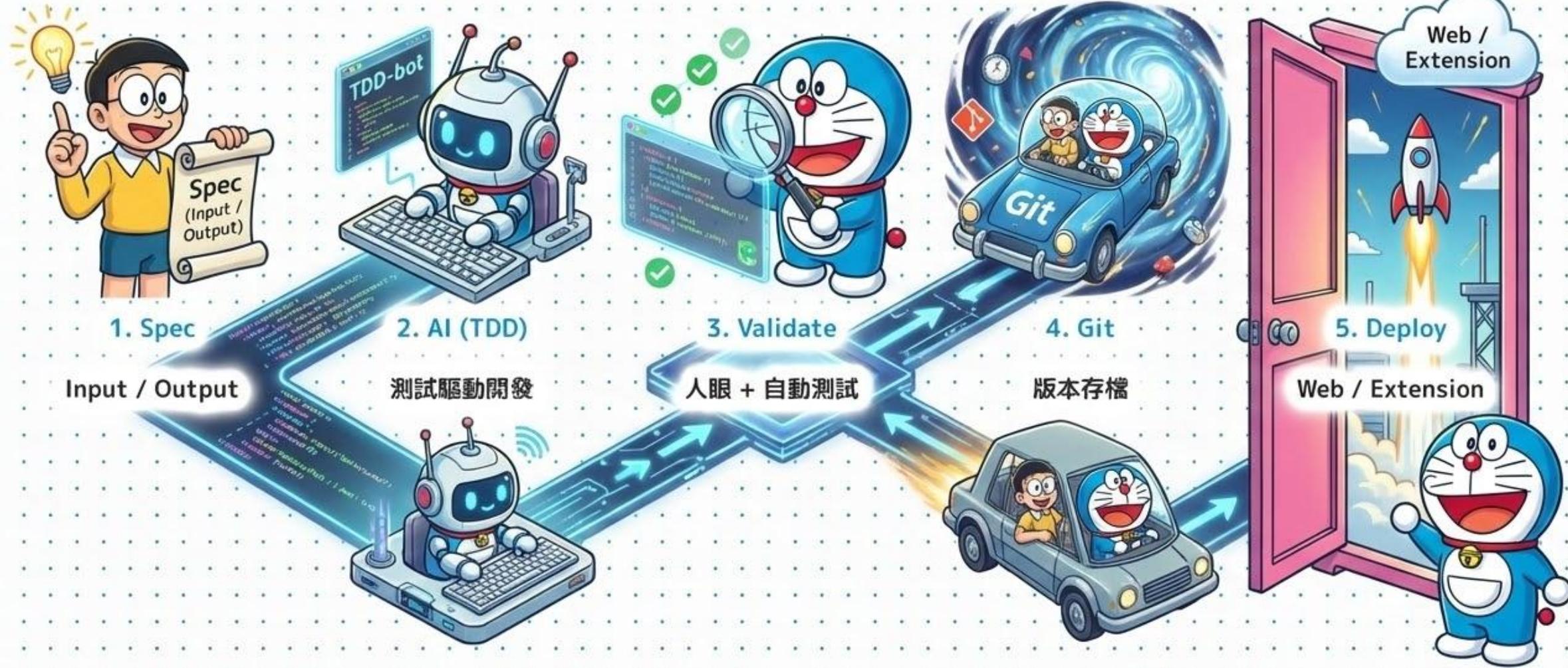


Extension 的優點（任意門）：

- 隨手可得：釘選在瀏覽器右上角，隨點隨用。
- 無縫整合：可以直接讀取
當前網頁內容（例：自動
抓取頁面價格填入計算器）。



Vibe Coding 開發流程全圖



說明書 (README) 是專案的封面

為什麼要寫 README？

- 💡 紿未來的自己看。
- 💡 紿接手的人看。
- 💡 最重要：給 AI 看

「下次丟回給 AI 時，它讀 README 就能秒懂專案架構，像記憶麵包一樣！」



總結

今日成就：

- 用 TDD 鎖死了業務邏輯。
- 做出了手機可用的 Web App。
- 學會用 Git (時光機) 存檔與切換時空。
- 產出了 README 文件。



課後作業

活用目前學過的技巧，做一個可以解決工
作痛點的 Web 工具。

(請提供開發歷程的 prompt)