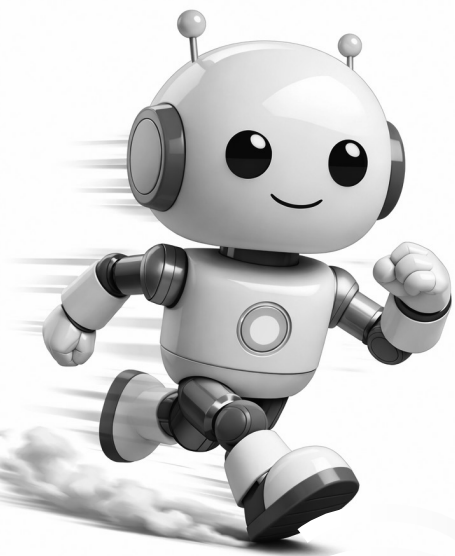


4

你的第一個 Vibe Coding 專案



親愛的讀者朋友，終於來到動手做的時刻了！

在前面的章節中，我們認識了 Vibe Coding 的概念，也瀏覽了各種好用的 AI 工具。現在，讓我們捲起袖子，從零開始打造你人生中的第一個 AI 輔助程式作品——一個個人待辦清單 App。

為什麼選擇待辦清單（to-do list）？因為它夠簡單，每個人都知道它該怎麼運作；但它又有足夠的深度，可以讓你練習到 Vibe Coding 的核心技巧，包括：描述需求、與 AI 對話、測試與迭代。不管你過去有沒有寫過任何一行程式碼，只要跟著這一章的步驟走，你都能做出一個真正可以使用的 App。

接下來，我們會用三種不同的方式來完成這個專案。首先，用最簡單的 ChatGPT 聊天介面來暖身，讓你體驗用對話方式寫出程式碼的感覺。接著，我們會使用 Google 的 Antigravity——一個零程式碼的 AI 開發平臺，讓你在瀏覽器裡就能打造完整的 Web App。最後，我們會進入 Claude Code 的世界，體驗更進階的 AI 輔助開發流程。

準備好了嗎？現在就讓我們開始吧！

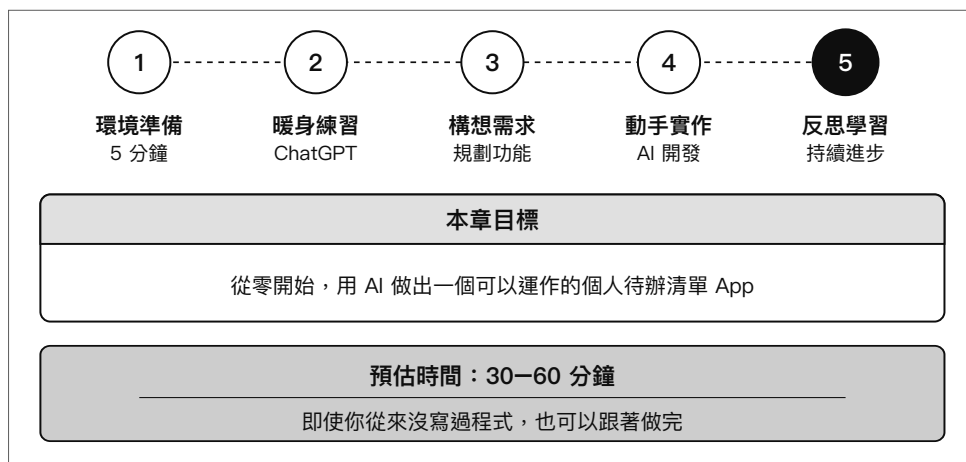


圖 4-1：第一個專案旅程地圖



4.1

準備出發：你需要什麼？

好消息：你需要的東西比你想像的少很多。不需要買新電腦、不需要安裝複雜的軟體，只要你有一臺能上網的電腦和一個瀏覽器，五分鐘內就能做好所有準備。

▮ 硬體需求：你已經有了

只要是能正常上網的電腦就行。Windows、Mac 或 Linux 都可以，不需要高階規格。你平常用來看影片、收信或瀏覽網頁的那臺電腦，就完全夠用了。手機和平板電腦雖然也能跟 ChatGPT 聊天，但因為我們後面要操作 Antigravity 和 Claude Code，建議你還是坐到電腦前面操作。

▮ 軟體需求：一個現代瀏覽器

推薦使用 Google Chrome，因為它跟 Antigravity 的相容性最好。如果你習慣用 Edge、Firefox 或 Safari 也沒問題，這些瀏覽器都能正常使用我們要用到的 AI 工

具。唯一要注意的是，請確保你的瀏覽器是最新版本——這不只是為了功能，也是為了資訊安全。

■ 帳號需求：三個免費帳號

我們在這一章需要使用三個服務，恰巧都有提供免費方案：

1. ChatGPT 帳號：到 chatgpt.com 註冊，可以用 Google 帳號直接登入。免費版就夠用了。
2. Antigravity 帳號：到 Google 的 Antigravity 網站（antigravity.google）註冊。這是 Google 推出的 AI 開發平臺，使用 Google 帳號登入即可。
3. Claude 帳號：到 claude.ai 註冊。如果你想使用 Claude Code，需要訂閱 Claude Pro 或 Max 方案。但別擔心，我們在這章會先用 Antigravity 帶你完成主要的開發流程。

💡 先從免費開始

三個帳號的免費版本，足以讓你完成本章所有練習。等你更熟悉之後，再考慮是否需要升級到付費方案。學習階段，不一定馬上需要花錢。

■ 心態準備：最重要的一件事

技術上的準備很簡單，但最關鍵的其實是心態。請記住這三點：

- 不怕犯錯：AI 不會嘲笑你，你可以反覆嘗試。請謹記，每一次失敗都是 AI 幫你學習的機會。
- 不急著完美：第一版一定不夠好，這其實很正常。Vibe Coding 的精神就是先做出來，再慢慢改進。
- 享受過程：你即將體驗一種全新的創造方式。保持好奇心，把它當成一場有趣的實驗。

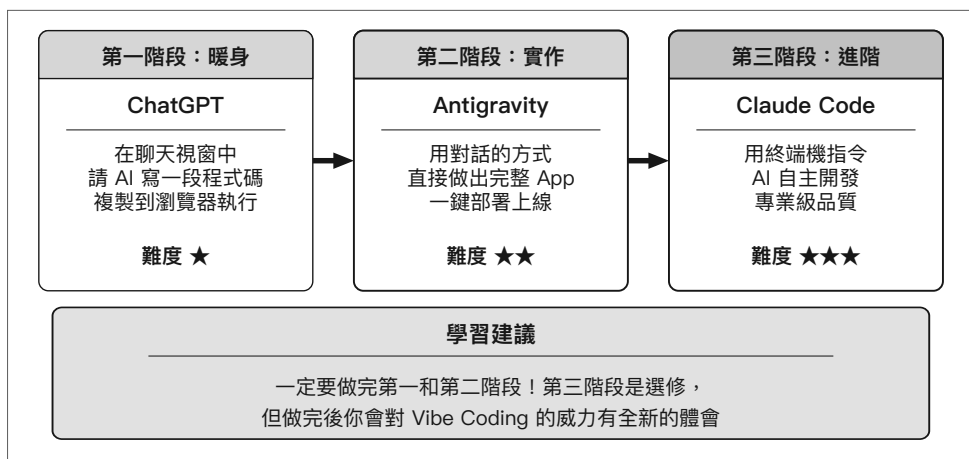


圖 4-2：三階段學習路徑



4.2

暖身練習：用 ChatGPT 寫你的第一段程式碼

在正式開始做待辦清單 App 之前，讓我們先用最簡單的方式暖暖身。這個練習的目的不是做出一個完整的產品，而是讓你親身感受用對話方式就能產生程式碼的這件看似不可能的事情。

步驟一：打開 ChatGPT，說出你的需求

打開瀏覽器，進入 chatgpt.com，登入你的帳號。在對話框中，直接用中文輸入：

幫我寫一個簡單的 HTML 網頁，顯示一個標題「我的第一個網頁」，
下面有一段文字「這是用 AI 幫我寫的！」，
再加上一個按鈕，按下去會跳出一個對話框說「恭喜你成功了！」

按下送出，等待幾秒鐘。ChatGPT 會回覆你一段程式碼，看起來可能像這樣：

```
<!DOCTYPE html>  
<html lang="zh-TW">
```

```
<head>
  <meta charset="UTF-8">
  <title> 我的第一個網頁 </title>
</head>
<body>
  <h1> 我的第一個網頁 </h1>
  <p> 這是用 AI 幫我寫的！ </p>
  <button onclick="alert('恭喜你成功了！')"> 點我試試 </button>
</body>
</html>
```

別被這些看起來奇怪的文字嚇到。你不需要理解每一行的意思，這正是 Vibe Coding 的精神。AI 幫你寫好了程式碼，你只需要知道怎麼讓它跑起來就好。



圖 4-3：ChatGPT 對話示範——請 AI 寫 HTML

步驟二：複製貼上，讓它跑起來

現在，讓我們把這段程式碼變成一個真正可以打開的網頁。這個步驟非常簡單，只需要三個動作：

1. 複製程式碼：在 ChatGPT 的回覆中，找到程式碼區塊，點擊右上角的「複製」按鈕（通常是一個小方塊圖示）。

2. 貼上並儲存：打開你電腦上的「記事本」（Windows）或「文字編輯器」（Mac），貼上剛才複製的程式碼。接著，把這個檔案儲存到桌面上，檔名取為 `my-first-page.html`。注意副檔名一定要是 `.html`，不是 `.txt`。
3. 打開檢視：到桌面上找到你剛儲存的 `my-first-page.html`，用瀏覽器打開它（雙擊通常就會自動用瀏覽器開啟）。你會看到你的第一個網頁出現了！試著點擊那個按鈕，應該會跳出「恭喜你成功了！」的對話框。

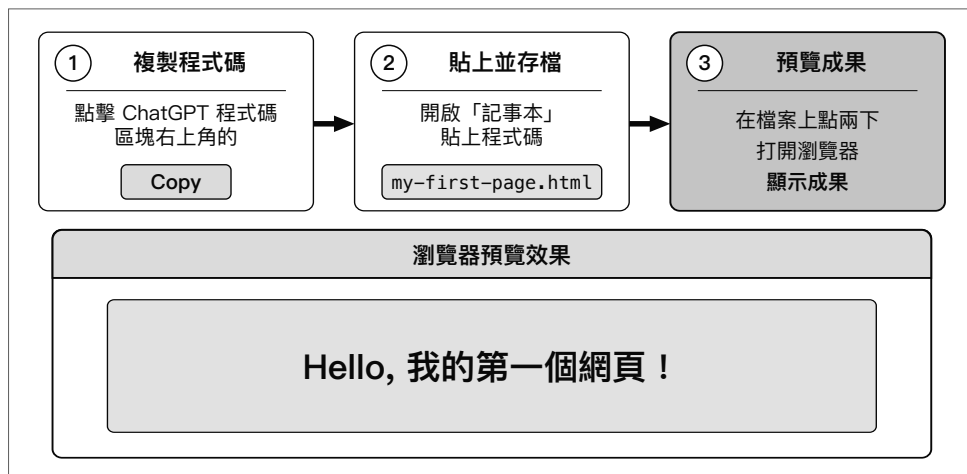


圖 4-4：三步驟執行 ChatGPT 的程式碼

💡 Mac 使用者注意

如果你用的是 Mac 內建的文字編輯 App，請先到「格式」選單，選擇「製作純文字」，再貼上程式碼。否則它可能會加入不必要的格式，導致程式碼無法正常運作。或者，你也可以使用免費的 Visual Studio Code 編輯器。

▮ 步驟三：嘗試修改，讓 AI 幫你加功能

做出第一個網頁後，讓我們更進一步。回到 ChatGPT，在同一個對話中繼續輸入：

很好！請幫我加上以下功能：

1. 把背景改成淺藍色
2. 標題改成置中對齊
3. 加上一個文字輸入框和一個「送出」按鈕，
按下送出時，在頁面上顯示使用者輸入的文字

ChatGPT 會生成一段更長的程式碼，包含了你要求的所有功能。再一次，複製、貼上、儲存、打開——你會看到一個有背景色、有輸入框、能互動的網頁。

這就是 Vibe Coding 的基本循環：你說出想要什麼，AI 寫出程式碼，你執行並檢查結果，然後繼續提出新的需求。在一來一回之間，程式就慢慢成形了。

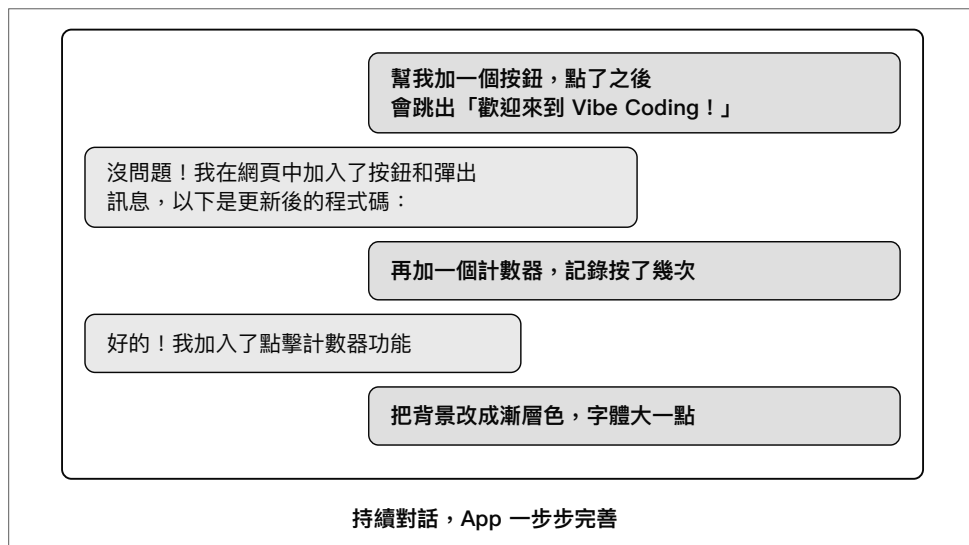


圖 4-5：ChatGPT 進階對話——加上互動功能

▮ 暖身練習的收穫

透過這個簡單的練習，你已經學到了三個重要的東西：

1. 用自然語言就能寫程式：你沒有寫任何一行程式碼，但你所下的指令讓 AI 幫你寫出了一個完整的互動網頁。

2. 「複製貼上」是一個合法的技能：在 Vibe Coding 的世界裡，知道怎麼使用 AI 產生的程式碼，本身就是一種能力。
3. 迭代的力量：你不需要一開始就想好所有功能。先做一個最簡單的版本，再一步步加上新功能。

不過，你也可能注意到了一些不方便的地方，像是：每次修改都要重新複製貼上、整個流程太手動了、而且做出來的東西只在自己電腦上，沒辦法輕鬆分享給別人。

這些問題，正是我們接下來要用 Antigravity 和 Claude Code 來解決的。



4.3

構想你的待辦清單 App

在開始跟 AI 對話之前，我們需要先想清楚一件事：你到底要做什麼？這聽起來很基本，但卻是很多人容易忽略或迅速跳過的一步。就像你去餐廳點菜，如果你只說「我要好吃的」，服務生會不知道該端什麼好菜上來。AI 也是一樣——當你需求描述得愈清楚，它給你的成果就會愈接近你想要的樣子。

▮ 定義功能：你的待辦清單需要什麼？

一個基本的待辦清單 App，通常需要哪些功能呢？讓我們逐一列出來：

1. 新增待辦事項：使用者可以輸入文字，新增一個新的待辦事項。
2. 標記完成：點一下就能把事項標記為已完成，會有刪除線的視覺效果。
3. 刪除事項：不需要的事項可以刪除。
4. 資料保存：關掉瀏覽器再打開，之前加的事項還在（使用 `localStorage`）。
5. 簡潔美觀：乾淨的介面，讓人看了舒服，使用也很方便。

上述的這五個功能，構成了我們的「最小可行產品」（MVP, Minimum Viable Product）。在 Vibe Coding 的世界裡，我們不會一開始就追求完美，而是先做出一個夠用的版本，再慢慢加上更多功能。

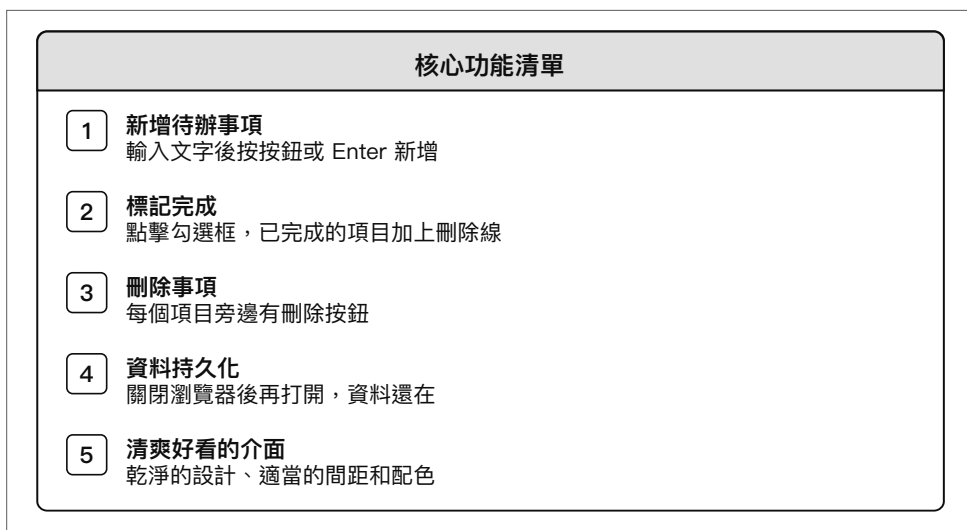


圖 4-6：待辦清單 App 功能規劃

畫出草圖：你的 App 長什麼樣子？

在腦中或筆記本上畫一張簡單的草圖，能幫助你更清楚地描述需求。舉例來說，我們的待辦清單 App 可能是長這樣的：

- 最上方是一個標題「我的待辦清單」
- 標題下面是一個輸入框和一個「新增」按鈕
- 下方是待辦事項的列表，每一項旁邊有一個完成勾選框和一個刪除按鈕
- 最下方顯示「共 x 項，已完成 y 項」的統計

嗯，你不需要畫得很漂亮。即使是在紙上隨手畫幾個方塊和文字，都比什麼都不畫要好得多。這張草圖會幫助你在跟 AI 對話時，更精確地描述你想要的介面佈局。



圖 4-7：待辦清單 App 介面草圖

■ 用自然語言描述需求：好的描述 vs. 不好的描述

現在我們有了功能清單和草圖，讓我們練習把它變成 AI 能理解的自然語言描述。沒錯，這是 Vibe Coding 中最重要的技能之一。

比較不理想的描述方式：

幫我做一個待辦清單

相信你也看得出來，類似這樣的描述太模糊了。畢竟 AI 不會通靈，它不知道你真的想要什麼功能、什麼樣子的介面跟風格。它還是可能做得出來，只是多半會做出一個跟你想像完全不同的東西。

比較好的描述方式，如下所示：

請幫我做一個個人待辦清單的 Web App，需要以下功能：

1. 有一個輸入框和新增按鈕，可以新增待辦事項
2. 每個事項可以點擊標記為已完成（顯示刪除線效果）
3. 每個事項右邊有刪除按鈕可以移除

4. 資料用 `localStorage` 儲存，關閉瀏覽器後不會消失
5. 底部顯示「共 X 項，已完成 Y 項」的統計

介面風格：簡潔現代，白色背景搭配淺灰色卡片，主色調用藍色。置中顯示，最大寬度 500px。

請用 HTML + CSS + JavaScript 做成單一檔案。

嗯，看到差別了嗎？好的描述通常包含了三個要素，像是：功能需求（它要做什麼）、外觀需求（它長什麼樣子）和技術需求（用什麼方式做）。你不需要懂程式語言才能寫出好的需求描述，只需要把「如果我是使用者，我希望這個 App 怎麼運作」說清楚就好。

需求描述的黃金原則

想像你在跟一個很聰明，但完全不了解你需求的助手說話。當你說得愈具體、愈有條理，他做出來的成果就會愈接近你所要的。條列是個好方法，能幫助你和 AI 都不遺漏任何細節。



4.4

方法一：用 Antigravity 打造待辦清單

Antigravity 是 Google 推出的 AI 程式開發平臺。它最大的特色是：你完全不需要離開瀏覽器，就能從對話、生成、預覽到部署一氣呵成。如果你是第一次嘗試 Vibe Coding，Antigravity 是目前最友善的入門工具之一。

▮ 第一步：進入 Antigravity

打開瀏覽器，前往 Antigravity 的網站。用你的 Google 帳號登入後，你會看到一個簡潔的首頁，上頭有一個文字輸入區域，邀請你描述想要做的東西。

Antigravity 的介面設計相當直覺，核心由 Agent Manager 與 Editor 兩個主要視圖組成。你可以在 Agent Manager 中輸入需求、發起任務，並追蹤代理的執行

狀態；也可以切換到 **Editor** 查看工作區檔案、編輯內容，並即時檢視 AI 產生的結果。這種一邊下指令、一邊看成果、一邊調整的體驗，正是 Vibe Coding 很吸引人的地方。



圖 4-8：Antigravity 開啟畫面

第二步：輸入你的需求

還記得我們在上一節寫的那段需求描述嗎？現在，直接把它貼進 Antigravity 的對話框裡：

請幫我做一個個人待辦清單的 Web App，需要以下功能：

1. 有一個輸入框和「新增」按鈕，可以新增待辦事項
2. 每個事項可以點擊標記為已完成（顯示刪除線效果）
3. 每個事項右邊有刪除按鈕可以移除
4. 資料用 `localStorage` 儲存，關閉瀏覽器後不會消失
5. 底部顯示「共 X 項，已完成 Y 項」的統計

介面風格：簡潔現代，白色背景搭配淺灰色卡片，主色調用藍色。置中顯示，最大寬度 500px。

按下送出，然後觀察右邊的預覽區——在短短幾秒鐘內，一個完整的待辦清單 App 就出現了。是的，就是這麼快！

Step 2：輸入你的需求

聊天區

請幫我做一個個人待辦清單 App：

1. 可以新增待辦事項
2. 可以標記完成（加刪除線）
3. 可以刪除事項
4. 資料存在本地
5. 介面乾淨好看

送出 (Enter)

輸入完整需求後按 Enter 送出

預覽區

等待 AI 生成中...

提示：需求描述愈具體，AI 生成的結果就愈接近你想要的

圖 4-9：Antigravity 輸入需求

第三步：檢查成果，開始迭代

老實說，AI 生成的第一版通常已經很不錯了，但可能不會百分之百符合你的預期。現在，讓我們來當一個挑剔的使用者，仔細檢查每一項功能：

- 試著輸入幾個待辦事項，看看新增功能是否正常？
- 點擊事項，看看是否出現刪除線效果？
- 點擊刪除按鈕，看看事項是否被移除？
- 重新整理頁面，看看資料是否還在（localStorage 測試）？
- 看看底部的統計數字是否正確？

嗯，假設你發現了一些想修改的地方。比如說，你覺得字體太小、按鈕不夠明顯，或者你想加一個篩選功能……沒關係，請直接在對話框中告訴 AI：

請幫我做以下修改：

1. 字體放大一點，主要文字用 16px。
2. 「新增」按鈕改成藍色背景、白色文字，更醒目一些。
3. 完成的事項移到列表最下方。
4. 加上一個清除全部已完成事項的按鈕。

Antigravity 會根據你的新指令，修改現有的程式碼。你不需要知道它改了哪些程式碼，你只需要在預覽區確認修改結果是否符合預期。當然，如果你有興趣研究程式碼，那是再好不過的了！

Step 3：看到成果！

AI 回應	我的待辦清單
✓ 已建立待辦清單 App ✓ 新增 / 刪除 / 完成功能 ✓ localStorage 儲存 ✓ 響應式設計 需要我調整什麼嗎？	輸入新事項... 新增 ☐ 買牛奶 ☒ 回覆信件 ☐ 準備週報 可即時互動操作 預覽區的 App 是真的可以操作的！

從輸入需求到看到成果，整個過程不到 2 分鐘

圖 4-10：Antigravity 生成結果

這個來回修改的過程，就是 Vibe Coding 的核心循環：描述 → 生成 → 檢查 → 再描述。透過每一輪對話，你的 App 就離完美更近一步了！



圖 4-11：Antigravity 迭代修改

■ 第四步：部署上線，分享給全世界

當你對結果滿意之後，Antigravity 通常可協助你進一步分享或部署應用程式。依照專案設定與當前支援方式，你可能可以透過介面中的部署或分享功能，將 App 發布到可供他人存取的網址，再把連結分享給朋友、家人或同事使用。

想一想：從開始到現在，你可能只花了十幾分鐘，就做出了一個可以分享的、真正能用的 Web App。如果用傳統的程式開發方式，即使是一個有經驗的工程師，這樣的功能也至少需要一兩個小時，甚至半天的時間。

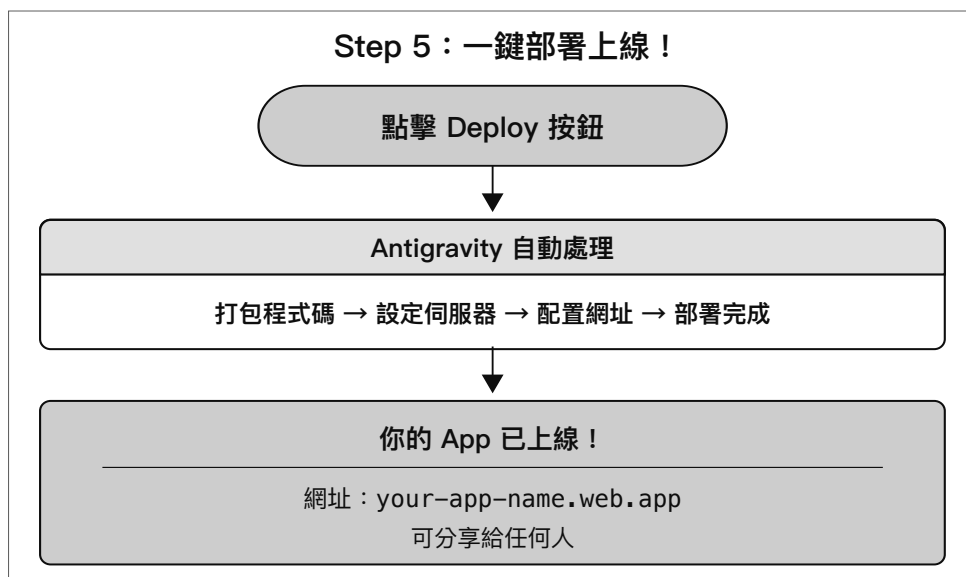


圖 4-12：Antigravity 部署上線

Antigravity 的優勢

Antigravity 最大的優點，是所見即所得的開發體驗。你與 AI 對話，立刻可以看到結果。對於第一次嘗試 Vibe Coding 的人來說，這是比較低門檻的選擇。



4.5

方法二：用 Claude Code 打造待辦清單

如果你體驗過 Antigravity，也許會想再試試 Claude Code。Claude Code 是 Anthropic 推出的 AI 程式開發助手，可在終端機、IDE、桌面版與瀏覽器中使用。它能讀取程式碼庫、建立與修改專案檔案、執行指令，並協助你完成功能開發、除錯與各種自動化開發任務。

使用 Claude Code 的起步門檻比一般圖形化工具高一些，但一旦上手，你會得到更高的操作自由度與更深的專案控制能力。若你希望真正理解 AI 如何協助你建立、修改與管理完整專案，Claude Code 會是一個很值得學的工具；而 Antigravity 則更偏向代理式、多工作面協作的體驗，適合不同類型的使用者。

■ 第一步：建立專案資料夾並啟動 Claude Code

讓我們先建立一個專門的資料夾來放你的待辦清單專案，然後啟動 Claude Code：

```
mkdir my-todo-app
cd my-todo-app
claude
```

輸入 `claude` 後，終端機的介面會變成一個對話模式。你會看到一個提示符號，等著你輸入指令。這就是 Claude Code 的工作環境——看起來像是在跟 AI 聊天，但它能直接操作你電腦上的檔案。

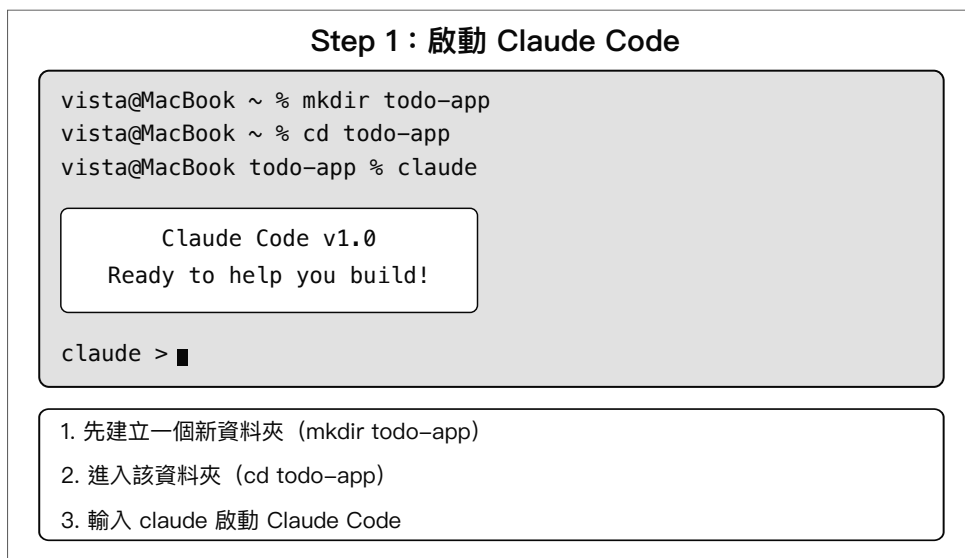


圖 4-13：Claude Code 啟動畫面

■ 第二步：用自然語言描述你的需求

在 Claude Code 的對話模式中，直接輸入你的需求：

請幫我建立一個待辦清單 Web App 的專案。需求如下：

- 使用 HTML + CSS + JavaScript
- 功能：新增事項、標記完成、刪除事項、localStorage 儲存
- 底部有統計（共 X 項，已完成 Y 項）
- 介面簡潔現代，白色背景、藍色主調
- 響應式設計，手機也能正常使用
- 請建立好完整的專案結構

按下 Enter，然後看著 Claude Code 開始工作。它會：

1. 分析你的需求，理解你要做什麼。
2. 自動建立專案資料夾和檔案結構。
3. 寫出 HTML、CSS 和 JavaScript 程式碼。
4. 把所有檔案存到你的電腦上。

整個過程你只需要等待，偶爾看一下 Claude Code 在終端機上的輸出，了解它正在做什麼。

Step 2：用自然語言描述需求

claude >幫我用 HTML / CSS / JavaScript 做一個待辦清單網頁，
需要以下功能：

1. 新增待辦事項（輸入框 + 按鈕）
2. 標記完成（打勾 + 刪除線）
3. 刪除事項
4. 用 localStorage 存資料
5. 乾淨好看的介面

Thinking...

- ✓ 建立 index.html
- ✓ 撰寫 style.css 樣式
- 撰寫 script.js 邏輯...

Claude Code 會自動規劃、建立檔案、撰寫 HTML / CSS / JavaScript 程式碼
你只需要等待，大約 2-3 分鐘

圖 4-14：Claude Code 輸入需求

第三步：查看成果

Claude Code 完成工作後，你會在 my-todo-app 資料夾裡看到它建立的所有檔案。通常會是這樣的結構：

```
my-todo-app/
  index.html    ← 主頁面
  style.css     ← 樣式檔案
  script.js     ← 功能邏輯
  README.md    ← 專案說明
```

你可以直接用瀏覽器打開 `index.html` 來查看結果。你可以用任何編輯器打開它們，完全掌控你的專案。

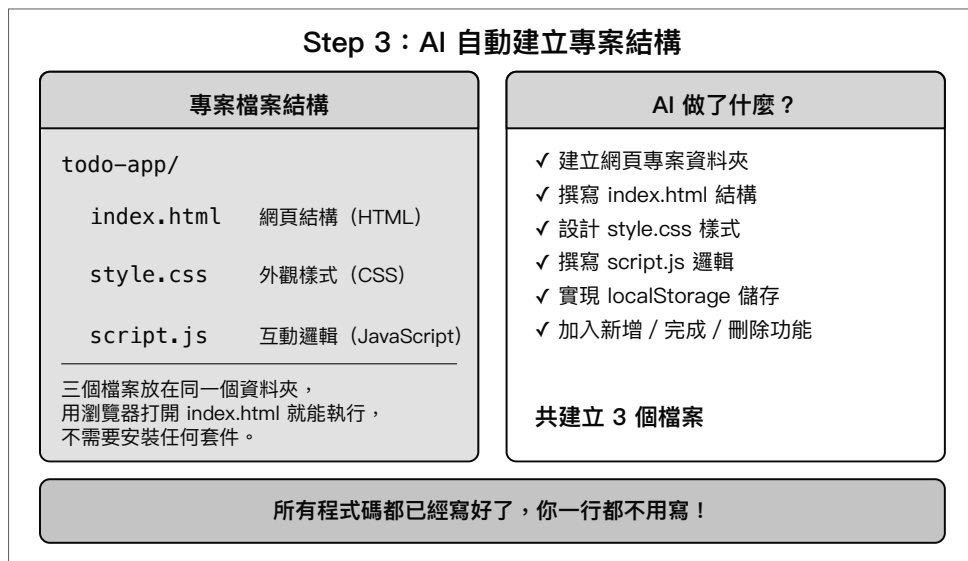


圖 4-15：Claude Code 自動建立專案結構

■ 第四步：預覽和修改

打開 `index.html` 後，測試一下所有功能。如果有需要修改的地方，回到 Claude Code 的對話中，直接說：

我測試了一下，有幾個地方想修改：

1. 新增事項後，輸入框的文字沒有自動清空。
2. 希望按 `Enter` 鍵也能新增事項，不只是點按鈕。
3. 加上日期顯示，每個事項旁邊顯示新增的日期。

Claude Code 會直接修改你電腦上的檔案。你不需要做任何複製貼上的動作——它直接幫你改好了。重新整理瀏覽器，就能看到更新後的結果。

Step 4：預覽和修改

```
✓ All files created successfully!
✓ Dependencies installed
Dev server running at http://localhost:5173

claude >把主題色改成深藍色，加上篩選功能

Updating App.jsx and App.css...
```

修改指令範例

```
「把背景改成淺灰色」 → AI 自動修改 CSS
「加上拖拉排序功能」 → AI 安裝新套件 + 修改元件
「加入截止日期欄位」 → AI 修改資料結構 + UI
「修好那個 Bug」 → AI 自動讀取錯誤、分析、修復
```

圖 4-16：Claude Code 預覽和修改

■ Antigravity vs. Claude Code：怎麼選？

到這裡，你已經用兩種方式做出了待辦清單 App。它們各有優勢，適合不同的場景：

比較面向	Antigravity	Claude Code
產品定位	Google 的 agent-first 開發平臺，強調讓代理幫你規劃、執行、驗證與迭代任務，不只是補 code。	Anthropic 的 AI coding assistant，可讀整個 codebase、改檔、執行指令、協助修 bug 與開發功能。
核心操作方式	比較像任務指揮中心：代理可規劃工作、寫程式、甚至瀏覽網頁協助完成任務。	比較像深入專案的 AI 開發助手：直接內專案內讀檔、改檔、執行命令，偏向實戰開發流。
適合的使用心智	想把 AI 當成半自動團隊成員來指揮的人。	想把 AI 當成貼身工程助手來一起寫與管理專案的人。
新手友善度	通常較直觀。若重點是先體驗代理替你完成任務，學習門檻相對低。	有一定門檻。常見用法仍包含 terminal / CLI 操作，但長期可獲得更高控制力。
主要使用環境	以 agent-first 工作流為核心，常結合瀏覽器、編輯器與代理協作體驗。	可在 terminal、IDE、desktop app、browser 中使用，不只限於終端機。

比較面向	Antigravity	Claude Code
多代理 / 任務編排	這是強項之一，偏向 agent manager、team workflow 與技能編排。	也支援 skills、subagents、routines，但核心敘事仍偏向專案內的高效開發。
擴充能力	可透過 Agent Skills 擴充能力。	可透過 skills、custom commands、subagents、routines 延伸使用情境。
部署導向	部署到 Google Cloud 的學習路徑相對鮮明。	重點不是特定雲端部署流程，而是協助你在既有開發流程中完成建置、維護與自動化。
比較適合誰	想快速體驗 AI 代理自己規劃與分工、偏好多代理協作感的人。	想更深入掌控專案、檔案、命令列與真實 codebase 的人。

如果你是完全新手，重點是先快速做出 demo、感受 agent-first 開發流程，Antigravity 往往比較容易上手。倘若你希望真正掌握專案結構、檔案管理、命令列操作與實際 codebase 工作流，Claude Code 的長期價值通常更高。

如果你完全沒有技術背景，建議先從 Antigravity 開始。等你對 Vibe Coding 有了信心和手感之後，再試試 Claude Code，你會發現更多可能性。

比較項目	Antigravity	Claude Code
準備時間	1 分鐘（開瀏覽器）	10 分鐘（安裝軟體）
開發時間	8—10 分鐘	3—5 分鐘
程式碼品質	良好	非常好
擴展性	中等	非常高
部署上線	一鍵完成	需自行處理
費用	免費	US\$ 20 / 月起
適合誰	零基礎初學者	追求品質和彈性的人

建議：先用 Antigravity 入門，熟悉後再嘗試 Claude Code

圖 4-17：Claude Code vs. Antigravity 比較

不管你選擇哪一種方式，最重要的是：你已經做出了一個真正可以使用的 App。這不再只是模擬、練習，而是你用 AI 親手打造的第一個作品。



圖 4-18：Claude Code 完成的專案成果



4.6

從構想到成品的完整流程

現在，讓我們退後一步，看看剛才我們經歷了什麼？從一個模糊的想法「我想做一個待辦清單 App」，到一個可以分享給朋友的成品。整個過程，可以拆解成五個階段：

第一階段：構想（2 分鐘）

這一切的起點。你不需要寫出完美的產品企劃書，只需要回答幾個簡單的問題：

- 我要做什麼？（一個待辦清單 App）
- 它有哪些功能？（新增、完成、刪除、儲存）

遇到問題怎麼辦？常見狀況排解

在你的第一個專案中，你可能會遇到一些常見的問題。別擔心，這些都是正常的，而且都有簡單的解決方法：

常見問題	可能原因	解決方法
AI 生成的結果跟想像不同	需求描述不夠具體	補充更多細節，重新描述一次
功能運作不正常	程式碼可能有 Bug	具體描述問題現象，請 AI 修復
頁面顯示空白	可能有語法錯誤	打開瀏覽器的開發者工具 (F12)，查看錯誤訊息，然後告訴 AI
localStorage 沒有生效	可能是隱私模式	確認不是用無痕瀏覽模式
手機上排版亂掉	缺少響應式設計	請 AI 加上類似「手機版介面也要好看」的要求

💡 遇到 Bug 不要怕

Bug 是程式開發中完全正常的一部分，即使是最資深的工程師也時常在處理 Bug。在 Vibe Coding 中，修 Bug 變得相對簡單：把問題的現象告訴 AI，讓它幫你找出原因並修復。

最重要的是把 Bug 的現象或症狀描述清楚：什麼時候發生、做了什麼操作、期望看到什麼、實際看到什麼？上述這四項資訊，足以讓 AI 幫你解決大部分問題。



4.7

讓你的作品更好：迭代最佳化

恭喜你做出了第一版的待辦清單 App！但故事還沒結束。Vibe Coding 中最重要的技能之一，就是「迭代」——讓你的作品一次比一次更好。在這一節，我們要學習怎麼有系統地最佳化你的 App。

▮ 迭代的心法：每次只做一類修改

很多人在想到要改善 App 的時候，會一口氣列出十幾個需要修改的重點，然後通通丟給 AI。這樣做的問題是：如果結果出了差錯，你很難知道是哪個修改造成的。

更好的做法是，把修改分成幾類，每次對話只處理一類：

1. 第一輪：功能修正。把所有不能正常運作的地方修好。
2. 第二輪：外觀最佳化。調整色彩、字體、間距、排版。
3. 第三輪：功能增強。加上新的功能，比如篩選、排序。
4. 第四輪：細節打磨。加上動畫效果、鍵盤快捷鍵等。

每一輪修改後，都要仔細測試，確認沒有問題再進入下一輪。這種有紀律的迭代方式，會讓你的開發過程更穩定、更有效率。

▮ 功能擴展的想法

當基本功能都穩定之後，你可以開始想：「還能加上什麼讓它更好用？」以下是一些適合新手嘗試的功能擴展：

1. 分類功能：讓使用者為事項加上標籤（例如「工作」、「生活」、「學習」），然後可以依分類篩選。
2. 優先順序：為事項設定高、中、低優先順序，用不同的顏色來區分。
3. 到期日：為事項加上到期日期，快要到期的事項用紅色提醒。
4. 拖曳排序：讓使用者可以拖動事項來改變順序。
5. 深色模式：加上一個開關，讓使用者可以在亮色和暗色主題之間切換。
6. 匯出功能：讓使用者可以把待辦清單匯出為文字檔。

每一個功能擴展，都可以當成一個獨立的迭代練習。一次做一個，做好了再做下一個。你會發現，每加一個功能，你對 Vibe Coding 的掌握度就更高一些。

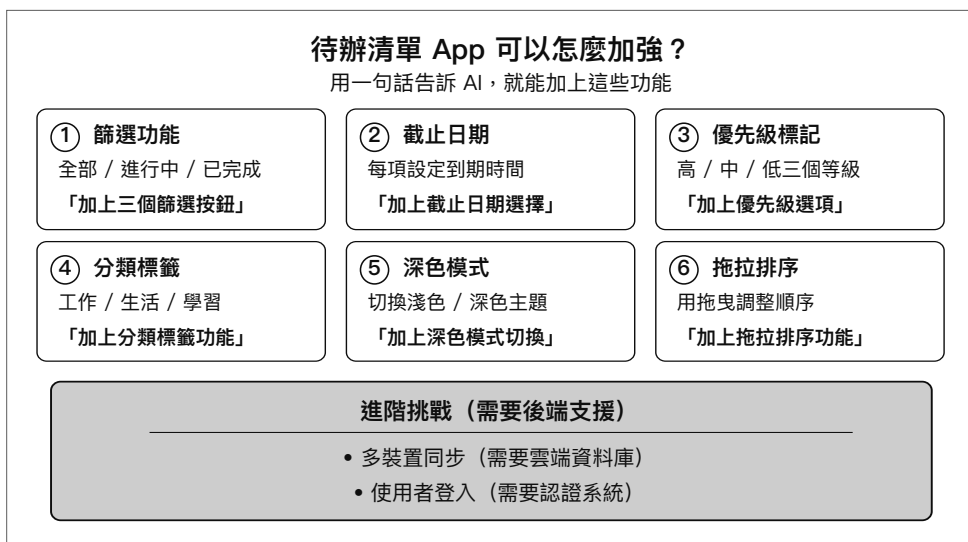


圖 4-19：功能擴展想法清單

與 AI 對話的進階技巧

經過前面的練習，你應該已經掌握了描述需求的基本功。現在，讓我們學幾個能讓你跟 AI 溝通更有效率的進階技巧：

1. 給 AI 看到你的錯誤訊息。如果你的 App 出了問題，打開瀏覽器的開發者工具（通常按 F12），看看 Console 面板有沒有紅色的錯誤訊息。把這些訊息複製給 AI，它就能更精確地幫你找到問題。
2. 告訴 AI 你目前的程式碼狀態。如果你已經做了很多修改，AI 可能會搞不清楚你目前的程式碼長什麼樣子。在 Claude Code 中，AI 可以直接讀取你的檔案，這個問題比較不嚴重。但在 ChatGPT 或其他對話式工具中，你可能需要偶爾把目前的程式碼整段貼給 AI，讓它知道你現在的狀態。
3. 善用角色設定。你可以在對話開始時告訴 AI：「請扮演一個資深的前端工程師，幫我最佳化這個待辦清單 App。請用初學者能理解的方式，說明你做了什麼修改？」如此一來，AI 會用更專業但同時更易理解的方式來回應你。

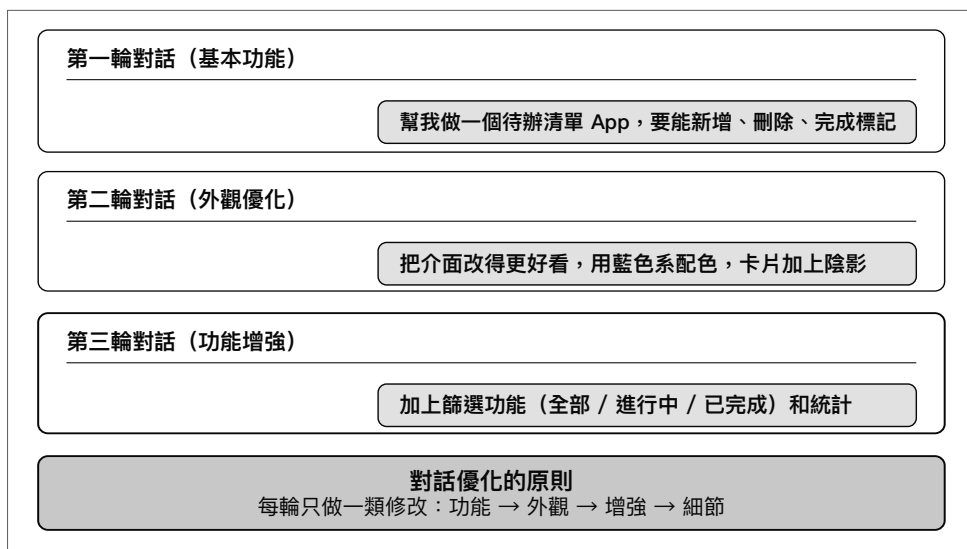


圖 4-20：與 AI 對話最佳化的範例

Debug 的藝術：怎麼跟 AI 報告 Bug

在 Vibe Coding 中，Debug（除錯）是一個你會經常用到的技能。別擔心，你不需要自己看懂程式碼來找 Bug——你只需要學會怎麼把問題說清楚，AI 就能幫你解決。

一個好的 Bug 報告，通常包含四個要素：

1. 你做了什麼操作：例如「我在輸入框中打了一段文字，然後按下 Enter 鍵」。
2. 你期望看到什麼：例如「我期望文字被加入到待辦清單中，輸入框清空」。
3. 實際上發生了什麼：例如「文字沒有被加入，輸入框也沒有清空，頁面完全沒有反應」。
4. 額外線索：例如「瀏覽器 Console 顯示了一個紅色錯誤訊息：TypeError: Cannot read property 'push' of undefined」。

把這四個要素組合成一段描述，AI 就能快速幫你定位並修復問題。根據我自己的經驗，隨著你做愈多的專案，你描述 Bug 的能力也會愈來愈好。

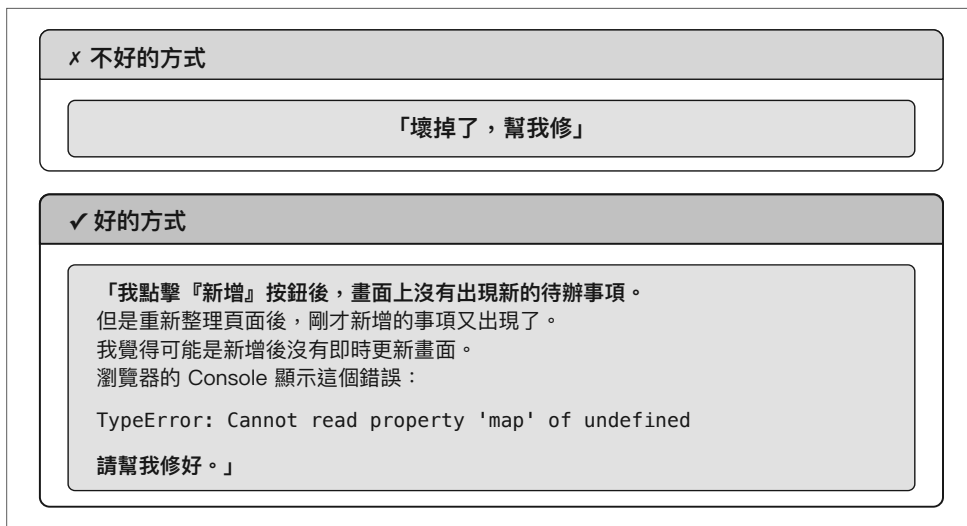


圖 4-21：Debug 對話範例

💡 Debug 小祕訣

如果 AI 修了一次但問題還在，別氣餒。試著提供更多細節，或者換個方式描述問題。有時候，擷取一張畫面給 AI 看（如果工具支援的話），比文字描述更有效。請謹記：跟 AI 溝通就像跟人溝通，多給一些背景資訊跟脈絡，它就能更精準地幫助你。



4.8

第一個專案的反思與學習

嘿，你做到了！不管你用的是 ChatGPT、Antigravity 還是 Claude Code，你已經從零開始，用 AI 打造出了一個真正可以使用的待辦清單 App。在我們結束這一章之前，讓我們回顧一下你在這趟旅程中學到了什麼。



學習重點

讓我們快速整理一下這一章的重點：

1. 環境準備：一臺電腦 + 瀏覽器 + 三個帳號（ChatGPT、Antigravity、Claude），五分鐘搞定。
2. 三種開發方式：ChatGPT（暖身體驗）→ Antigravity（視覺化開發）→ Claude Code（進階開發），你可以選擇最適合自己的方式。
3. 核心流程：構想 → 描述需求 → AI 生成 → 測試 → 迭代修改。整個過程只需要自然語言，不需要寫程式。
4. 溝通技巧：描述愈具體愈好、每次只改一類東西、Bug 報告要包含四要素。
5. 心態：不追求一次完美、擁抱迭代、不怕犯錯。

你的下一步

完成了第一個專案，你已經跨過了最重要的一道門檻——從旁觀者變成了親自動手的創造者。接下來，你可以：

1. 繼續最佳化你的待辦清單 App。試著加上前面提到的擴展功能，把它變成一個你會真正每天使用的工具。
2. 嘗試一個不同類型的專案。比如做一個簡單的記帳本、讀書筆記管理工具、或是運動習慣追蹤器。每一個新專案都會帶給你新的學習。
3. 分享你的作品。把你做的 App 的網址分享到社群媒體上，或是拿給朋友看。來自真實使用者的回饋，是最好的學習資源。
4. 繼續閱讀本書的後續章節。我們會教你更多進階的 Vibe Coding 技巧，讓你能夠挑戰更複雜、更有趣的專案。

請謹記，Vibe Coding 最美妙的地方在於：你的能力只受限於你的想像力。只要你能想到、能說出來，AI 有很高的機率能夠幫你做出來。所以，大膽去想、勇敢去做吧！