**🚀 Vibe Coding 技巧全攻略（Vibe Coding Skills）**

分享一個更輕鬆、但極具實用性的話題 —— **Vibe Coding**。

這個詞由 AI 大師 **Andrej Karpathy** 提出，他形容這種編碼方式就像是在和大型語言模型（LLM）一起「有感覺地」創作程式碼 —— 自然流暢、邊試邊修，快速前進。

**🎧 什麼是 Vibe Coding？**

Vibe Coding 的核心精神是：

讓 **LLM** 幫你**生成程式碼** → 微調它 → 再生成下一段 → 一步步組合出成品

它非常適合探索新框架、新想法，而且樂趣十足、生產力爆表。

但！若使用不當，也容易迷失方向、卡關或產出過於冗長的程式碼。

因此，我整理了 **5 個關鍵技巧**，幫助你用對方式 Vibe Coding：

* **Good vibes** – prompt well – ask for short answers and latest APIs for today's date  
  **良好的感覺** – 撰寫良好的提示語 – 要求簡潔答案並使用最新的 API（註明今天日期）
* **Vibe but verify** – ask 2 LLMs the same question  
  **感覺可以，但要驗證** – 問兩個 LLM 相同的問題
* **Step up the vibe** – ask to break down your request into independently testable steps  
  **升級你的 vibe** – 請求將任務拆分為可獨立測試的步驟
* **Vibe and validate** – ask an LLM then get another LLM to check  
  **感覺之後要驗證** – 問一個 LLM，然後請另一個 LLM 幫你檢查
* **Vibe with variety** – ask for 3 solutions to the same problem, pick the best  
  **多樣 vibe** – 要求同一問題提供三種解法，選擇最佳方案

**✅ 技巧一：Good Vibes – 編寫高品質 Prompt**

* 投入時間設計可重用的提示語（Prompt）
* 請求「簡潔、乾淨的程式碼」，避免預設冗長風格
* 明確指定「截至今天的 API」，避免使用過時語法
* 例子提示：
  + 請以簡潔方式撰寫此功能，僅使用 2025 年 6 月最新可用 API

**✅ 技巧二：Vibe but Verify – 交叉驗證，避免誤導**

* 同樣問題不要只問一個模型，多問**兩個或三個**（如 ChatGPT + Claude）
* 比較不同回答，有助於補強理解
* 常見情況：一個模型答錯或表達模糊，另一個卻直擊核心

**✅ 技巧三：Step Up the Vibe – 拆小問題、逐步前進**

* \*\*錯誤示範：\*\*一次生成 200 行程式碼
* \*\*正確做法：\*\*讓 LLM 一次只生成 10 行、單一功能、可測試的模組

🔧 如果你不會拆解問題？**問 LLM 幫你拆 break-down**！

* 我想實作 XXX 功能，請幫我拆解成 4~5 個簡單步驟，
* 每步都能獨立測試，請勿產生程式碼。
* 接著，對每個步驟再請 LLM 逐一產生程式碼與測試方式。

**✅ 技巧四：Vibe and Validate – Another LLM 幫你驗 LLM**

* 拿第一個模型產出的程式碼
* 再問另一個模型：

這段程式碼是否正確？有沒有更簡潔或更清楚的寫法？

這就是「Evaluator → Optimizer」設計模式的手動實作方式。

**✅ 技巧五：Vibe with Variety – 一題多解，多樣性促進創意 Pick up the best**

* 要求 LLM 提供 3 種不同的解法（如果可能）
* 強迫模型思考不同策略
* 也請它**解釋每個解法的思路與選擇理由**

好處：  
你不只能選出最佳解法，還能**理解為什麼**這樣做比較好。

**📌 補充提醒：請求解釋，完全理解**

Vibe Coding 很有趣沒錯，但如果你不理解生成程式碼在做什麼，  
問題發生時會非常痛苦。

所以請務必：每次讓 LLM **生成程式碼**後，都**要求它詳細說明**每一行的意義。

**🎉 總結：Vibe Coding 是一門技能！**

Vibe Coding 不只是跟 LLM 聊天寫程式，它是一種結合直覺、技巧與思辨的創作模式。

✔ 會讓你學得更快  
✔ 能提高開發效率  
✔ 還能享受創造的樂趣！