

Vibe Coding - Lập Trình Theo Cảm Hứng với AI

Vibe Coding là gì?

- **Khái niệm:** Do **Andrej Karpathy** (nhà nghiên cứu AI huyền thoại) đặt ra, mô tả phong cách lập trình "**cảm hứng**" với sự trợ giúp của AI.
- **Cách hoạt động:** Bạn để AI (như ChatGPT, Claude) **tạo code**, sau đó chỉnh sửa, lặp lại và tiến triển dự án một cách linh hoạt.
- **Ưu điểm:**
 - Tốc độ phát triển nhanh, đặc biệt khi làm việc với framework/công nghệ mới.
 - Giảm áp lực "viết từng dòng code" truyền thống.

Tuy nhiên, nếu không cẩn thận, bạn dễ bị AI "dẫn lối sai" → code rối, khó debug.

5 Bí Kíp "Vibe Coding" Hiệu Quả

1. "Good Vibes" - Prompt Chất Lượng

- **Viết prompt rõ ràng, tái sử dụng được.** Ví dụ:
 - ❌ "Viết code để gọi API" → Mơ hồ.
 - "Viết hàm Python ngắn gọn gọi OpenAI API (phiên bản 2024), dùng thư viện `requests`, xử lý lỗi cơ bản".
- **Yêu cầu code ngắn gọn:** AI thường sinh code dài dòng, nhiều exception không cần thiết.
- **Nhắc ngày hiện tại:** Tránh dùng API lỗi thời (do AI được train trên dữ liệu cũ).

2. "Vibe but Verify" - Kiểm Tra Chéo

- Đừng chỉ hỏi **1 AI**. Hãy thử cùng lúc **ChatGPT + Claude + Gemini** để so sánh kết quả.
- **Ví dụ:**
 - ChatGPT có thể giải thích chi tiết, trong khi Claude cho code tối ưu hơn.
 - Kết hợp ưu điểm của cả hai.

3. "Step Up the Vibe" - Chia Nhỏ Vấn Đề

- **Đừng để AI sinh 200 dòng code rồi mới kiểm tra** → Dễ thành "mớ hỗn độn".
- **Giải pháp:**
 - Yêu cầu AI **chia task thành 5-10 bước nhỏ**, mỗi bước ~10 dòng code.
 - Test từng phần trước khi ghép lại.

- **Ví dụ:**

- "Hãy chia bài toán train model AI thành 4 bước: (1) Load data, (2) Tiền xử lý, (3) Xây dựng model, (4) Đánh giá".

4. "Vibe and Validate" - Kiểm Tra Logic

- Sau khi có code, hỏi AI khác: "Code này có bug không? Có cách nào tối ưu hơn?".
- **Áp dụng pattern "Evaluator-Optimizer"** trong thiết kế AI Agent:
 - Một Agent sinh code, Agent khác kiểm tra chất lượng.

5. "Vibe with Variety" - Đa Dạng Giải Pháp

- Yêu cầu AI **đề xuất 3 cách giải quyết** khác nhau → Tăng khả năng tìm ra phương án tốt nhất.
- **Ví dụ:**
 - "Hãy viết 3 phiên bản hàm tính toán này: dùng vòng lặp, đệ quy, và thư viện numpy"
 - .
- **Bonus:** Yêu cầu AI **giải thích từng cách** → Hiểu sâu hơn.

Lời Khuyên

- **Luôn hiểu code AI sinh ra:** Đừng chỉ "copy-paste" mù quáng. Nếu không rõ, hãy hỏi AI giải thích.
- **Vibe Coding là công cụ tuyệt vời**, nhưng **tư duy phân tích** của bạn mới là yếu tố quyết định!

Phiên bản #1

Được tạo 29 tháng 4 2025 16:25:22 bởi Đỗ Ngọc Tú

Được cập nhật 2 tháng 5 2025 07:45:21 bởi Đỗ Ngọc Tú