1. **Tab "Code" là gì?**
   * **Mục đích chính:** Tab "Code" thường được sử dụng để hiển thị và chỉnh sửa mã nguồn của dự án.
   * **Hình ảnh minh họa:** Có thể có biểu tượng hình cây cúc (icon of a tree structure) hoặc một biểu tượng chữ "Code" để chỉ ra rằng đây là nơi chứa mã nguồn.
2. **Giao diện chính của Tab "Code":**
   * **Editor:** Trên tab "Code," bạn thường sẽ thấy một trình soạn thảo mã nguồn với khả năng làm nổi bật cú pháp (syntax highlighting).
   * **Cấu trúc thư mục:** Nếu dự án lớn, có thể có một cửa sổ thể hiện cấu trúc thư mục để dễ dàng chuyển giữa các tệp và thư mục.
   * **Thanh công cụ:** Thanh công cụ có thể chứa các nút chức năng như lưu, mở, chạy mã, và kiểm tra lỗi.
3. **Các tính năng tương tác:**
   * **Chỉnh sửa:** Người sử dụng có thể nhập và chỉnh sửa mã nguồn trực tiếp trong trình soạn thảo.
   * **Di chuyển giữa các tệp và thư mục:** Nếu có cấu trúc thư mục, người dùng có thể chọn tệp và thư mục để xem nội dung.
   * **Lưu và chạy:** Các nút hoặc tùy chọn để lưu thay đổi và chạy mã.
4. **Hiển thị lỗi và cảnh báo:**
   * **Cảnh báo và lỗi:** Nếu có lỗi hoặc cảnh báo trong mã nguồn, chúng thường được hiển thị dưới dạng thông báo hoặc điều kiện đặc biệt.
5. **Tích hợp công cụ phụ trợ:**
   * **Auto-complete:** Gợi ý từ khóa và biến khi người dùng đang nhập.
   * **Debugging tools:** Nếu có, có thể có các công cụ hỗ trợ kiểm tra lỗi và gỡ lỗi.

Nếu bạn có thêm chi tiết hoặc câu hỏi cụ thể về một môi trường phát triển hoặc một tab "Code" cụ thể, hãy mô tả thêm để tôi có thể cung cấp thông tin chính xác hơn.



**You**

Tìm hiểu các lệnh được sử dụng trong git.

ChatGPT

**ChatGPT**

Git là một hệ thống quản lý phiên bản phổ biến được sử dụng để theo dõi sự thay đổi trong mã nguồn của dự án phần mềm. Dưới đây là một số lệnh quan trọng trong Git:

1. **git init:**
   * Tạo một kho lưu trữ Git mới hoặc khởi tạo một kho lưu trữ Git trong thư mục hiện tại.
2. **git clone <repository\_url>:**
   * Sao chép một kho lưu trữ từ một địa chỉ URL.
3. **git add <file> (hoặc git add .):**
   * Thêm một tệp hoặc tất cả các tệp thay đổi vào chỉ mục để chuẩn bị cho commit.
4. **git commit -m "Commit message":**
   * Lưu trạng thái hiện tại của chỉ mục vào kho lưu trữ với một thông điệp chú giải.
5. **git status:**
   * Hiển thị trạng thái của thư mục làm việc, bao gồm các tệp đã sửa đổi, đã thêm và đã xóa.
6. **git pull:**
   * Cập nhật thư mục làm việc với các thay đổi mới từ kho lưu trữ từ xa và thực hiện merge.
7. **git push:**
   * Đẩy các commit đã thực hiện lên kho lưu trữ từ xa.
8. **git branch:**
   * Liệt kê tất cả các nhánh có trong dự án.
9. **git checkout -b <branch\_name>:**
   * Tạo và chuyển đến một nhánh mới.
10. **git merge <branch\_name>:**
    * Kết hợp các thay đổi từ một nhánh khác vào nhánh hiện tại.
11. **git log:**
    * Hiển thị lịch sử commit của dự án.
12. **git diff:**
    * Hiển thị sự khác biệt giữa các commit hoặc giữa chỉ mục và thư mục làm việc.
13. **git remote -v:**
    * Hiển thị các kho lưu trữ từ xa đã được cấu hình.
14. **git fetch:**
    * Lấy các thay đổi từ các kho lưu trữ từ xa mà không thực hiện merge.
15. **git reset:**
    * Hủy bỏ các thay đổi trong chỉ mục hoặc thư mục làm việc.

Đây chỉ là một số lệnh cơ bản trong Git. Có nhiều lệnh và tùy chọn khác, tùy thuộc vào nhu cầu cụ thể của dự án và quy trình làm việc. Để biết thêm chi tiết về mỗi lệnh, bạn có thể sử dụng **git help <command>** hoặc xem tài liệu trực tuyến của Git.