Hướng dẫn Sử dụng Git

Git là một hệ thống quản lý phiên bản phân tán được sử dụng rộng rãi trong phát triển phần mềm. Dưới đây là một hướng dẫn chi tiết và đầy đủ về việc sử dụng Git.

1. Cài đặt Git:
   * Tải xuống Git từ trang web chính thức (<https://git-scm.com/downloads>) và cài đặt vào máy tính của bạn.
   * Sau khi cài đặt xong, mở terminal và thiết lập thông tin cá nhân của bạn bằng cách chạy các lệnh sau:

**git config --global user.name "Tên của bạn"**

**git config --global user.email "Địa chỉ email của bạn"**

1. Khởi tạo kho lưu trữ (repository):
   * Mở terminal và di chuyển đến thư mục chứa dự án của bạn bằng lệnh **cd <đường-dẫn-thư-mục>**.
   * Sử dụng lệnh **git init** để khởi tạo một kho lưu trữ Git mới trong thư mục hiện tại.
2. Thao tác cơ bản:
   * Sử dụng lệnh **git add <tên-tập-tin>** để đánh dấu tập tin đã thay đổi để được theo dõi bởi Git. Bạn cũng có thể sử dụng **git add .** để thêm tất cả các tập tin thay đổi trong thư mục hiện tại.

**git add file1.txt** # Đánh dấu tập tin file1.txt

**git add .** *# Đánh dấu tất cả các tập tin thay đổi trong thư mục hiện tại*

* + Sử dụng lệnh **git commit -m "<thông-điệp-commit>"** để lưu lại các thay đổi đã đánh dấu trong kho lưu trữ. Thông điệp commit nên mô tả ngắn gọn về những thay đổi đã được thực hiện trong commit đó.

git commit -m "Thêm chức năng x vào ứng dụng"

* + Sử dụng lệnh **git status** để xem trạng thái hiện tại của kho lưu trữ. Git sẽ hiển thị các tập tin đã thay đổi và các tập tin đã được đánh dấu sẵn sàng cho commit.

**git status**

* + Sử dụng lệnh **git log** để xem lịch sử commit trong kho lưu trữ. Git sẽ hiển thị danh sách các commit theo thứ tự thời gian, bao gồm thông tin về tác giả, thời gian và thông điệp commit.

**git log**

1. Branch và Merge:
   * Sử dụng lệnh **git branch** để xem danh sách các nhánh trong kho lưu trữ và biết nhánh hiện tại.

**git branch** *# Liệt kê danh sách các nhánh*

**git branch -v** *# Liệt kê danh sách các nhánh kèm theo thông tin commit cuối cùng trên mỗi nhánh*

* + Sử dụng lệnh **git branch <tên-branch>** để tạo một nhánh mới.

**git branch feature** *# Tạo nhánh mới có tên là "feature"*

* + Sử dụng lệnh **git checkout <tên-branch>** để chuyển đổi sang một nhánh khác.

**git checkout feature** *# Chuyển sang nhánh "feature"*

* + Sau khi làm việc trên một nhánh, bạn có thể thực hiện các thay đổi và commit như bình thường.
  + Để gộp các thay đổi từ một nhánh vào nhánh hiện tại, sử dụng lệnh **git merge <tên-branch>**.

**git merge feature** *# Gộp các thay đổi từ nhánh "feature" vào nhánh hiện tại*

1. Đồng bộ với kho lưu trữ từ xa:
   * Liên kết kho lưu trữ của bạn với một kho lưu trữ từ xa (remote repository) bằng cách sử dụng lệnh **git remote add origin <URL-kho-lưu-trữ>**.

**git remote add origin https://github.com/username/repo.git**  *# Liên kết kho lưu trữ với kho lưu trữ từ xa*

* + Sử dụng lệnh **git push -u origin <tên-branch>** để đẩy các thay đổi lên kho lưu trữ từ xa.

**git push -u origin main** *# Đẩy các thay đổi từ nhánh "main" lên kho lưu trữ từ xa*

* + Sử dụng lệnh **git pull** để kéo các thay đổi mới nhất từ kho lưu trữ từ xa về kho lưu trữ cục bộ của bạn.

**git pull** *# Kéo các thay đổi mới nhất từ kho lưu trữ từ xa*

1. Xử lý xung đột (conflicts):
   * Trong trường hợp xảy ra xung đột khi hợp nhất thay đổi từ hai nhánh khác nhau, Git sẽ thông báo về xung đột và dừng quá trình hợp nhất.
   * Để giải quyết xung đột, bạn cần mở các tập tin bị xung đột trong trình soạn thảo và sửa xung đột theo ý muốn.
   * Sau khi sửa xung đột, sử dụng lệnh **git add <tên-tập-tin>** để đánh dấu tập tin đã được sửa đổi.
   * Tiếp theo, sử dụng lệnh **git commit** để hoàn thành quá trình hợp nhất và tạo một commit mới.