

Để nắm vững kiến thức cơ bản về GitHub và Git, bạn cần tìm hiểu về các khái niệm, lệnh, và thực hành. Dưới đây là danh sách các nội dung quan trọng cần học:

1. **Khái niệm Cơ bản**:

- Repository (Kho lưu trữ): Đây là nơi lưu trữ mã nguồn và tài liệu của dự án.
- Commit: Thay đổi hoặc bản ghi trong kho lưu trữ.
- Branch (Nhánh): Phiên bản riêng của dự án có thể phát triển độc lập.
- Merge (Hợp nhất): Kết hợp thay đổi từ một nhánh vào nhánh khác.
- Pull Request: Yêu cầu hợp nhất các thay đổi từ một nhánh vào nhánh chính.

2. **Cài Đặt Git**:

- Học cách cài đặt Git trên máy tính của bạn.

3. **Cấu Hình Git**:

- Thiết lập tên và email cho các commit của bạn bằng lệnh `git config`.

4. **Tạo Repository**:

- Tạo một kho lưu trữ trên GitHub hoặc trên máy tính cá nhân của bạn bằng lệnh `git init`.

5. **Thêm và Commit Thay Đổi**:

- Sử dụng `git add` để thêm thay đổi vào vùng chờ (staging area).
- Sử dụng `git commit` để lưu trữ các thay đổi đã thêm vào trong kho lưu trữ.

6. **Branching và Merging**:

- Tạo và quản lý các nhánh bằng các lệnh `git branch`, `git checkout`, và `git merge`.

7. **Remote Repository**:

- Liên kết kho lưu trữ cục bộ của bạn với kho lưu trữ từ xa trên GitHub bằng `git remote`.
- Sử dụng `git push` để đẩy các thay đổi lên kho lưu trữ từ xa.

- Sử dụng `git pull` để cập nhật kho lưu trữ cục bộ với thay đổi từ kho lưu trữ từ xa.

8. **Pull Requests**:

- Tạo và quản lý pull requests trên GitHub để đề xuất sự thay đổi vào dự án.

9. **Resolving Conflicts**:

- Hiểu cách giải quyết xung đột (conflicts) khi hợp nhất thay đổi từ các nhánh khác nhau.

10. **Logs và Lịch Sử**:

- Sử dụng `git log` để xem lịch sử commit.

11. **Undoing Changes**:

- Sử dụng `git reset` và `git revert` để hoàn tác các thay đổi.

12. **.gitignore File**:

- Tạo và quản lý tệp `.gitignore` để loại trừ các tệp không cần thiết khỏi kho lưu trữ.

13. **Git Hooks** (Tùy chọn):

- Hiểu cách sử dụng các Git hooks để tự động hóa các hành động trước hoặc sau khi thực hiện các lệnh Git.

14. **GitHub Actions hoặc CI/CD** (Tùy chọn):

- Hiểu cách sử dụng GitHub Actions hoặc các dịch vụ CI/CD khác để kiểm tra và triển khai mã nguồn tự động.

15. **Git Clients** (Tùy chọn):

- Nắm vững việc sử dụng các công cụ Git client như GitHub Desktop hoặc Sourcetree.

Hãy bắt đầu bằng việc đọc tài liệu chính thống của Git và thực hành thông qua các dự án nhỏ để làm quen với các lệnh và quy trình. Sau đó, bạn có thể xây dựng và quản lý dự án thực tế trên GitHub để cải thiện kỹ năng của mình.