**Quy trình làm việc với Git và GitHub từ đầu đến cuối**

**1. Chuẩn bị môi trường**

1. **Cài đặt Git** (nếu chưa có):
   * Tải từ [git-scm.com](https://git-scm.com/downloads)
   * Cài đặt theo hướng dẫn của trình cài đặt
2. **Cài đặt và mở VS Code** (hoặc IDE ưa thích):
   * Tải từ [code.visualstudio.com](https://code.visualstudio.com/download)
   * Cài đặt và mở phần mềm

**2. Thiết lập Git ban đầu (chỉ cần làm 1 lần)**

bash

git config --global user.name "TenTaiKhoanGitHub"

git config --global user.email "email@example.com"

**3. Tạo repository mới trên GitHub**

1. Đăng nhập vào [GitHub](https://github.com)
2. Nhấn vào nút "+" ở góc trên bên phải, chọn "New repository"
3. Đặt tên repository (ví dụ: BMTT\_NC\_HUTECH\_2280600686)
4. Thêm mô tả (tùy chọn)
5. Chọn quyền riêng tư (Public hoặc Private)
6. Nhấn "Create repository"

**4. Làm việc với mã nguồn trên máy tính**

1. **Tạo thư mục dự án mới**:

bash

mkdir TenThuMuc

cd TenThuMuc

1. **Khởi tạo Git repository**:

bash

git init

1. **Viết mã nguồn**:
   * Mở VS Code với thư mục hiện tại:

bash

code .

* + Tạo và chỉnh sửa các file Python của bạn

**5. Đẩy code lên GitHub**

1. **Xem trạng thái file**:

bash

git status

1. **Thêm file vào staging area**:

bash

git add . *# Thêm tất cả file*

*# HOẶC*

git add TenFile.py *# Thêm từng file cụ thể*

1. **Commit các thay đổi**:

bash

git commit -m "Mô tả thay đổi của bạn"

1. **Liên kết repository local với GitHub**:

bash

git remote add origin https://github.com/TenTaiKhoan/TenRepository.git

1. **Đẩy code lên GitHub**:

bash

git push -u origin master

*# HOẶC*

git push -u origin main *# Tùy theo nhánh mặc định*

**6. Cập nhật code khi có thay đổi**

1. **Thực hiện các thay đổi trong code**
2. **Kiểm tra trạng thái**:

bash

git status

1. **Thêm thay đổi vào staging area**:

bash

git add .

1. **Commit thay đổi**:

bash

git commit -m "Mô tả thay đổi mới"

1. **Đẩy lên GitHub**:

bash

git push

**7. Tải code từ GitHub về máy tính (khi làm việc trên máy khác)**

1. **Clone repository**:

bash

git clone https://github.com/TenTaiKhoan/TenRepository.git

1. **Chuyển đến thư mục dự án**:

bash

cd TenRepository

1. **Cập nhật code mới nhất từ GitHub**:

bash

git pull

# Cách đổi tên nhánh master thành Lab1

Có hai phần trong quá trình này: đổi tên nhánh trên máy tính local của bạn và cập nhật tên nhánh trên GitHub.

## 1. Đổi tên nhánh master thành Lab1 trên máy tính local

### Nếu bạn đang ở nhánh master:

bash

*# Đổi tên nhánh hiện tại từ "master" thành "Lab1"*

git branch -m master Lab1

### Nếu bạn không ở nhánh master:

bash

*# Đổi tên nhánh "master" thành "Lab1"*

git branch -m master Lab1

## 2. Cập nhật thay đổi lên GitHub

### Đẩy nhánh Lab1 mới lên GitHub

bash

*# Đẩy nhánh Lab1 lên GitHub và thiết lập theo dõi*

git push -u origin Lab1

### Xóa nhánh master trên GitHub (tùy chọn)

bash

*# Xóa nhánh master trên GitHub*

git push origin --delete master

## 3. Cập nhật nhánh mặc định trên GitHub (qua giao diện web)

1. Truy cập vào repository của bạn trên GitHub
2. Nhấn vào tab "Settings"
3. Trong menu bên trái, chọn "Branches"
4. Ở phần "Default branch", nhấn vào biểu tượng bút chì (edit)
5. Chọn "Lab1" từ danh sách
6. Nhấn "Update"
7. Xác nhận thay đổi bằng cách nhấn "I understand, update the default branch"

## 4. Cập nhật cấu hình local (tùy chọn)

bash

*# Cập nhật upstream cho nhánh Lab1*

git branch --set-upstream-to=origin/Lab1 Lab1

## 5. Kiểm tra kết quả

bash

*# Liệt kê tất cả các nhánh local và remote*

git branch -a

## Lưu ý quan trọng:

* Đổi tên nhánh có thể gây ra vấn đề cho các thành viên khác trong nhóm nếu họ đang sử dụng nhánh master
* Nếu đây là dự án nhóm, hãy thông báo cho tất cả thành viên về việc đổi tên nhánh
* Các thành viên khác cần thực hiện những bước sau để đồng bộ với thay đổi:

bash

git fetch --all

git checkout Lab1

git branch -d master *# Xóa nhánh master cũ trên máy local*

* Một số tích hợp liên tục (CI) hoặc các dịch vụ khác có thể vẫn đang sử dụng nhánh master làm mặc định và cần được cập nhật

Sau khi hoàn thành các bước trên, nhánh mặc định của bạn sẽ được đổi tên từ "master" thành "Lab1" cả trên máy local và trên GitHub.

Retry

X

THƯ MỤC CÓ CẬP NHẬT ĐỔI TÊN FILE CÁC THỨ LÀM SAO ĐỂ CẬP NHẬT THAY DOODI

Edit

# Cách cập nhật thay đổi tên file và cấu trúc thư mục lên GitHub

Khi bạn thay đổi tên file, di chuyển file giữa các thư mục, hoặc thực hiện các thay đổi về cấu trúc trong dự án của mình, đây là quy trình để cập nhật những thay đổi đó lên GitHub:

## 1. Thay đổi tên file hoặc cấu trúc thư mục

### Cách 1: Thông qua giao diện hệ thống

* Thay đổi tên file, tạo thư mục mới, di chuyển file trực tiếp qua File Explorer (Windows), Finder (Mac), hoặc trình quản lý file trên hệ điều hành của bạn.

### Cách 2: Sử dụng lệnh Git

bash

*# Đổi tên file*

git mv TenFileCu.py TenFileMoi.py

*# Di chuyển file vào thư mục*

git mv TenFile.py TenThuMuc/TenFile.py

*# Đổi tên thư mục*

git mv TenThuMucCu TenThuMucMoi

Sử dụng git mv có lợi thế là Git sẽ tự động theo dõi việc đổi tên/di chuyển thay vì coi đó là xóa file cũ và tạo file mới.

## 2. Kiểm tra trạng thái thay đổi

bash

git status

Git sẽ hiển thị các thay đổi về tên file hoặc cấu trúc thư mục. Nếu bạn đã sử dụng git mv, các thay đổi sẽ được hiển thị dưới dạng "renamed". Nếu bạn thực hiện thay đổi qua hệ thống file, Git có thể xem đó là "deleted" và "new file".

## 3. Thêm thay đổi vào staging area

bash

*# Thêm tất cả thay đổi*

git add .

*# Hoặc thêm từng file cụ thể*

git add TenFileMoi.py

git add TenThuMuc/

## 4. Commit thay đổi

bash

git commit -m "Cập nhật tên file và cấu trúc thư mục"

## 5. Đẩy thay đổi lên GitHub

bash

git push

## Các trường hợp đặc biệt:

### Nếu bạn đổi tên file nhưng Git không nhận ra đó là đổi tên:

Git thường khá thông minh trong việc phát hiện đổi tên/di chuyển file, nhưng nếu có quá nhiều thay đổi nội dung, nó có thể không nhận ra. Trong trường hợp đó:

bash

*# Buộc Git nhận ra đó là đổi tên file*

git rm TenFileCu.py

git add TenFileMoi.py

git commit -m "Đổi tên từ TenFileCu.py thành TenFileMoi.py"

### Nếu bạn cần đồng bộ với các thay đổi từ GitHub:

Nếu có người khác đã đổi tên file trên GitHub và bạn muốn cập nhật local:

bash

git fetch

git pull

# Hướng dẫn cập nhật code lên GitHub

## Quy trình cơ bản

### 1. Kiểm tra trạng thái hiện tại

bash

git status

Lệnh này hiển thị các file đã thay đổi, thêm mới, hay xóa trong thư mục làm việc của bạn.

### 2. Thêm thay đổi vào staging area

bash

*# Thêm tất cả các thay đổi*

git add .

*# Hoặc thêm từng file cụ thể*

git add TenFile.py

*# Hoặc thêm nhiều file cùng lúc*

git add TenFile1.py TenFile2.py

### 3. Kiểm tra lại trạng thái sau khi thêm

bash

git status

Các file đã được thêm vào staging area sẽ hiển thị trong phần "Changes to be committed".

### 4. Commit thay đổi

bash

git commit -m "Mô tả ngắn gọn về thay đổi"

Thay "Mô tả ngắn gọn về thay đổi" bằng thông điệp mô tả thay đổi của bạn.

### 5. Đẩy thay đổi lên GitHub

bash

git push

Nếu đây là lần đầu đẩy một nhánh mới lên GitHub:

bash

git push -u origin TenNhanh

## Các trường hợp cập nhật phổ biến

### Cập nhật khi thay đổi code trong file hiện có

bash

git add TenFile.py

git commit -m "Sửa đổi chức năng XYZ"

git push

### Cập nhật khi thêm file mới

bash

git add TenFileMoi.py

git commit -m "Thêm chức năng ABC"

git push

### Cập nhật khi đổi tên file

bash

*# Cách 1: Sử dụng git mv (khuyến nghị)*

git mv TenFileCu.py TenFileMoi.py

git commit -m "Đổi tên file từ TenFileCu thành TenFileMoi"

git push

*# Cách 2: Đổi tên thủ công qua hệ thống file*

*# (sau khi đổi tên thủ công)*

git add .

git commit -m "Đổi tên file từ TenFileCu thành TenFileMoi"

git push

### Cập nhật khi xóa file

bash

*# Cách 1: Sử dụng git rm*

git rm TenFile.py

git commit -m "Xóa file không cần thiết"

git push

*# Cách 2: Xóa thủ công*

*# (sau khi xóa file thủ công)*

git add . *# hoặc git add -u*

git commit -m "Xóa file không cần thiết"

git push

### Cập nhật khi cấu trúc lại thư mục

bash

*# Di chuyển file vào thư mục khác*

git mv TenFile.py ThuMuc/TenFile.py

git commit -m "Di chuyển file vào thư mục mới"

git push

*# Đổi tên thư mục*

git mv ThuMucCu/ ThuMucMoi/

git commit -m "Đổi tên thư mục"

git push

## Xử lý tình huống đặc biệt

### Khi có xung đột khi pull code mới về

bash

git pull *# Gặp xung đột*

*# Mở file xung đột và sửa*

git add .

git commit -m "Giải quyết xung đột"

git push

### Khi cần hoàn tác thay đổi trước khi commit

bash

*# Hoàn tác thay đổi trong một file*

git checkout -- TenFile.py

*# Hoàn tác tất cả thay đổi chưa được staged*

git checkout -- .

*# Hoàn tác thay đổi đã được staged*

git reset TenFile.py

git checkout -- TenFile.py

### Khi cần hoàn tác commit gần nhất

bash

*# Hoàn tác commit nhưng giữ lại thay đổi*

git reset --soft HEAD~1

*# Hoàn tác commit và bỏ thay đổi*

git reset --hard HEAD~1

## Thực hành tốt

1. **Commit thường xuyên**: Thực hiện các commit nhỏ, tập trung vào một thay đổi cụ thể.
2. **Viết thông điệp commit rõ ràng**: Thông điệp commit nên ngắn gọn nhưng mô tả đầy đủ các thay đổi.
3. **Pull trước khi push**: Luôn git pull trước khi git push để tránh xung đột.
4. **Sử dụng branch**: Tạo nhánh mới cho các tính năng lớn hoặc thử nghiệm.

bash

git checkout -b TenNhanhMoi

*# Thực hiện thay đổi...*

git push -u origin TenNhanhMoi

1. **Kiểm tra trạng thái thường xuyên**: Sử dụng git status để theo dõi các thay đổi trong thư mục làm việc.
2. **Xem lịch sử commit**: Sử dụng git log để xem lịch sử commit.