========== Cài đặt tài khoản lần đầu =========

git config --global user.name "{Tên}"

git config --global user.email "{Tên email}"

============ Learn git ==================

Cac buoc tao mot git moi:

B1: Vào bên trong thư mục muốn tạo ta chạy lệnh này (chỉ cần làm một lần khi chọn nơi dùng git)

--> git init

B2: Để kiểm tra trạng thái các file bên trong thư mục, chạy lệnh này:

--> git status

B3: Để xác nhận file (được tạo mới, có thay đổi nội dung bên trong hoặc bị xóa) trong thư mục, ta chạy lệnh này

--> git add {Tên file}

VD: git add abc.txt

Để xác nhận nhiều file cùng lúc, dùng lệnh: git add .

B4: Để commit (đóng gói và lưu lên git -> lưu lại các thay đổi trong thư mục), ta chạy lệnh này:

--> git commit -m "câu mô tả cho sự thay đổi"

============= Một số thuật ngữ =============

Working dir: vùng làm việc với git trên máy (khu vực thư mục được chọn làm việc với git)

Staging area: Là khu vực chứa các file đã được xác nhận (khi người dùng dùng lệnh git add {tên file})

Git repository: khu vực lưu lại vào git sau khi người commit

=============== Một số lệnh git khác ================

git show : dùng để xem những thay đổi của những file

git log : xem lại nhật ký commit

git diff : dùng để xem thay đổi của file (chỉ dùng được với file được thay đổi \_ modified)

git checkout -- {tên file} : dùng để xóa bỏ những thay đổi đối với file đã được thay đổi nhưng chưa được xác nhận (file chưa được đưa vào staging area)

git reset : dùng để bỏ xác nhận đối với file (đưa file ra khỏi staging area)

============== branching and merging ================

git branch: Xem các branch hiện có

gitk : Để mở giao diện git tại đó lên

git checkout -b <tên branch> : Tạo branch và chuyển sang branch đó

git checkout <branch> : Chuyển sang <branch>

git merge : Trộn dữ liệu trong branch này -> branch kia

Lưu ý: Ví dụ trộn branch B --> branch A

B1: Chuyển sang branch A

B2: git merge <branch B>

git branch -D <branch> : Xóa branch

============== Quay trở lại commit trước ===========

git reset --soft <SHA1 ID> : các commit sau của commit này đều bị bỏ qua, các thay đổi trong file của các commit bị bỏ qua được đưa về staging area (SHA1 ID là id của commit trong giao diện git khi dùng lệnh gitk)

git reset --mixed <SHA1 ID> : Quay lại commit trước, các thay đổi trong file của các commit bị bỏ qua được đưa về working dir

git reset --hard <SHA1 ID> : Quay lại commit trước, các thay đổi trong file của các commit bị bỏ qua sẽ bị xóa hoàn toàn

========= Cách tạo một repository trên github ======

- Gần biểu tượng avatar, chọn new repository, nhập thông tin

- GitHub sẽ hiện hướng dẫn dưới đây, bạn cần làm theo hướng dẫn theo đúng trường hợp của bạn

-----------------------------------------

…or create a new repository on the command line

echo "# learn-git" >> README.md

git init

git add README.md

git commit -m "first commit"

git branch -M master

git remote add origin https://github.com/TanThinh99/learn-git.git

git push -u origin master

…or push an existing repository from the command line

git remote add origin https://github.com/TanThinh99/learn-git.git

git branch -M master

git push -u origin master

…or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

------------------------------------------

git push : để đưa commit mới lên github (chỉ dùng được với branch mặc định là branch master)

========== Cách lưu thông tin đăng nhập của github trên máy =====

git config --global credential.helper "cache --timeout=18000" : Lưu thông tin đăng nhập trong RAM thời gian lưu là 18000 giây

======== Cách clone và pull về từ github ============

git clone

- Trong dự án trên github có nút "Có hình tải xuống và có chữ Code", nhấn vào đó và copy link dự án

- Vào thư mục chứa dự án, gõ lệnh: git clone <link dự án>

git pull : Để tải thay đổi trong dự án về

======= Push với branch mới =================

- Tạo branch mới trong command line, lệnh đó là: git checkout -b <tên branch>

- Để push lên github ở branch phụ, ta dùng lệnh: git push origin <branch>, chỉ có branch mặc định là branch master ta mới có thể dùng lệnh "git push"

======== Tạo Pull request ====================

- Tạo pull request để người khác check code và merge branch đó với branch master

+ Nhấn pull request

+ Sau đó đặt tên, và viết mô tả

+ Nhấn create pull request

+ Nếu có xung đột giữa 2 branch, người tạo pull request cần giải quyết xung đột đó (cách giải quyết xem phần bên dưới)

+ Người khác sẽ vào review lại code

+ Khi review xong và không có xảy ra xung đột giữa các branch, họ sẽ nhấn Merge pull request

========== Conflict khi merge branch ===========

- Thường xảy ra conflict khi:

+ Branch A và branch B cùng thay đổi một file trên cùng dòng

+ Branch A xóa file ABC, nhưng branch B chỉnh sửa file ABC

- Để giải quyết xung đột, ta Merge branch master vào branch phụ:

+ Vào branch master, dùng git pull để lấy toàn bộ code ở branch master về

+ Chuyển qua branch phụ

+ git merge master

+ Vào file bị conflict và sửa lại nó

+ Add file đó (Đưa file vào staging area), sau đó commit thay đổi đó

+ Push lên lại: git push origin <branch>

+ Vào pull request, vào lúc này không còn conflict nữa

+ Nhấn Merge pull request.

========== Cấp quyền cho người khác tham gia vào reponsitory ===========

- Vào settings

- Nhấn Manage access, gõ mật khẩu

- Nhấn Invite a collaborator, nhập email hoặc username để mời người đó vào

- Ở phần thông báo của người đó, Nhấn xác nhận tham gia.

============= Thoát tài khoản git trên desktop ====================

Go to: Control Panel -> User Accounts -> Manage your credentials -> Windows Credentials -> under Generic Credentials there are some credentials related to Github, click on them and click "Remove".

It signed me out so the next time I do a push, it asks me to enter my username and my password.