**Thiết lập thông tin user.**

Nếu không thiết lập thì sau này sẽ không biết ai là người đã commit code.

git config --global user.name “Your\_username”

git config --global user.email “Your\_email”

**Thiết lập editor cho git**

git config –global core.editor “**’**Đường\_dẫn\_tơi\_file\_exe\_cua\_editor**’**” -multilnst –nosession

**Kiểm tra các setting**

git config --list //kiểm tra tất cả các cài đặt.

git config user.name //kiểm tra username.

git config user.email //kiểm tra email.

git config user.editor //kiểm tra editor.

**Khởi tạo một repository**

git init

**Kiểm tra xem file nào chưa được untracked trong nhánh hiện tại.**

git status

**Để thêm file vào nhánh hiện tại**

git add ten\_file.xx

**Để commit gõ lệnh**

git commit –m “Nội dung commit”

**Để xem log commit gõ lệnh**

git log

**Để clone một project gõ lệnh**

git clone url\_của\_project.

**The lifecycle of the status of your files in git.**



**Để kiểm tra status của files gõ lệnh.**

git status

git status --short// Để hiển thị theo dạng ngắn.

**Để xem stage và unstage.**

git diff –staged

**Remove một file từ staged sang untracked.**

git rm --cached tên\_file.xx

**Để add hết files vào stage.**

git add --all (hoặc -A)

**Để commit all files.**

git commit -a -m “mô\_tả\_commit”

**Để xóa hết tất cả các file log trong thư mục log.**

git rm log^\*.log

git rm –cached ten\_thu\_muc/\\*.loai\_file

(chí remove ra khỏi git, file vẫn nằm trên ổ cứng)

git rm ten\_thu\_muc**/\\*.**loai\_file -f

(xóa tất cả file nằm trong Ten\_thuc\_muc trên ổ cứng)

**Để đổi tên trong git**

git mv file1.xx file 2.xx (đổi tên từ file1 sang file2)

**Để xem log trong git.**

git log –p -2

**Trong đó**

-p là cho thấy thự khác biệt trong mỗi commit.

-2 là muốn lấy bao nhiêu commit

**Để xem log chi tiết trong git gõ.**

git log –stat

(muốn thoát để gõ lệnh tiếp theo thì di chuyển đến cuối và nhấn phím q).

**Để xem log trên một dòng gõ.**

git log --pretty=oneline

**Để xem log từ khoảng thời gian nào gõ.**

git log --since=2.weeks //Lấy log từ 2 tuần đổ lại.

Xem thêm tại: <https://git-scm.com/docs/git-log>

**Để undu trong git gõ.**

git commit --amend

ví dụ:

Ta muốn add 2 file trong một commit nhưng lúc commit lại chỉ add có một file.

* gõ lệnh **git add** để add file đó vào stage.
* gõ lệnh **git commit --amend** (một màn hình editor xuất hiện để ta chỉnh sửa lại nội dung commit, ta chỉnh sửa xong thì lưu lại).
* ta gõ lệnh **git log --stat** để xem chi tiết các commit, xem đã đúng như ý muốn chưa.

**Để chuyển một file về chế độ unstaged.**

git reset HEAD ten\_file.xx

**Để bỏ thử thay đổi một file nào đó trước khi commit.**

git checkout – ten\_file.xx

**Để xem tên của các remote gõ.**

git remote –v

**Để thêm một remote repositories.**

*Đây là một lệnh dùng để đặt lại tên cho một repo(mặt định có tên là origin)*

*Ví dụ lệnh* ***git remote add cplus*** [***https://github.com/timnguyen90/cplus.git***](https://github.com/timnguyen90/cplus.git)

*sẽ đặt lại repo origin trên có tên là cplus, và từ sau này các lệnh push, fetch, pull sẽ sử dụng tên là cplus.*

git remote add <ten\_muon\_dat> <url\_den remote\_repositories>

(lưu ý là sau khi gõ lệnh trên thì muốn xem nó thay đổi như thế nào thì gõ lệnh

git remote -v).

Ví dụ: ***git pull cplus master --allow-unrelated-histories***

cplus là một repo đã được đặt tên lại.

Tùy chọn ***--allow-unrelated-histories*** dùng để force cho việc merge, nếu không có sẽ báo lỗi*.*

**Để xem những gì mới nhất trên repository.**

git fetch remote\_name.

**Để lấy dữ liệu mới nhất (download dữ liệu) và merge dữ liệu về.**

git pull. remote\_name

**Để show thông tin của một remote repo.**

git remote show ten\_repo

**Để đổi tên một remote repo.**

git remote rename ten\_cu ten\_moi

**Để upload source lên remote repo.**

git push ten\_remote ten\_cua\_nhanh\_local (thường là master)

**Để tạo annotated**

git tag -a Ten\_tag -m “Mo\_ta\_cho\_tag”

**Để tạo một tag cho một commit**

git tag -a ten\_tag Id\_cua\_commmit.

**Để show tag.**

git show Ten\_tag

**Xem log trên một dòng.**

git log -pretty==oneline

**Để push tag lên remote repo**

git push ten\_repo ten\_tag.

**Để push tất cả tag lên**

git push ten\_remote --tags

**Để chuyển tag sang một tag mới.**

git checkout -b ten\_nhanh\_moi ten\_tag

**Ví dụ:** git checkout -b version1.0 v1.0

**Để quay về một nhánh.**

git checkout ten\_nhánh

Ex: git checkout master //quay về nhánh master.

**Để xóa tag.**

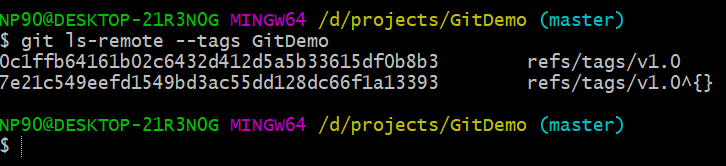
git tag -d ten\_tag.

Ex. git tag -d v1.1

**Để xem tất cả các tag trên remote repo.**

git ls -remote --tags ten\_cua\_remote

Ví dụ:



**Để xóa tag trên remote.**

git push ten\_remote --delete ten\_tag.

**Tạo alias để gõ lệnh nhanh trong git.**

git config --global alias.Tên\_tắt lệnh\_mà\_được\_thay\_thế\_trong\_git.

**Ví dụ**:

1. git config --global alias.st status **//st thay cho lệnh status trong git**

* Để gọi ra sử dụng thì ta gõ **git st**

1. git config --global alias.unstate ‘reset HEAD --’ //unstate thay thế cho việc gõ lệnh ‘reset HEAD --ten\_file\_can\_unstate.

* Gọi ra sử dụng: git unstate --ten\_file

**Nhánh chính (master) là cho việc release sản phẩm, nếu ta muốn thử nghiệm tính năng mới thì tạo thêm một nhánh khác dựa trên nhánh chính (master).**

**HEAD là một con trỏ trỏ tới một nhánh**

**Liệt kê nhánh đang có**

git branch

**Để tạo nhánh mới.**

git branch ten\_nhanh

**Để xem con trỏ HEAD đang trỏ tới đâu**

git log --oneline --decorate

**Để chuyển sang một nhánh khác**

git checkout ten\_nhanh

**Khi tạo một nhánh mới từ nhánh master thì nó sẽ thừa hưởng hết tất cả những gì của master, khi ta tạo file mới từ nhánh mới đó, nhánh master sẽ không có. Nhánh master muốn có gì của nhánh tách ra đó ta phải merge nhánh đó với nhánh master.**

**Để merge một nhánh sang một nhánh hiện tại.**

git merge ten\_nhanh\_merge.

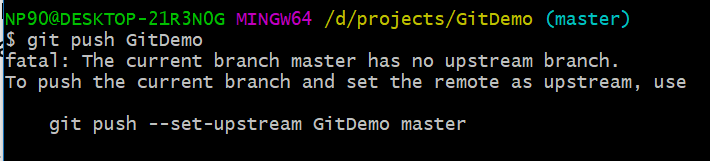
**Ví dụ:** ta đang ở nhánh master, khi ta gõ lệnh **git merge issue53** thì nó sẽ merge nhánh issue53 sang nhánh master (việc này đồng nghĩa là cập nhật lại dữ liệu của nhánh master bằng dữ liệu của nhánh issue53).

**Để kết nối nhánh master trên local và nhánh master trên remote repo.**

git push --set-upstream Ten\_repo master

**Ví dụ:** git push --set-upstream GitDemo master

Ta thực thi lệnh này khi bị lỗi sau:



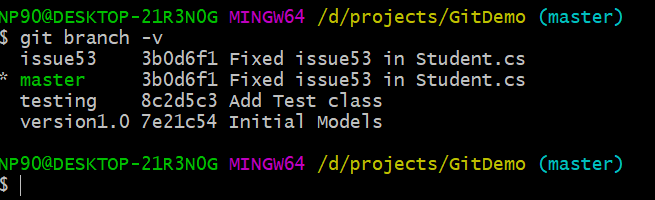
**Sau khi merge ta cần push sự thay đổi lên repo bằng lệnh**

git push ten\_repo

Nếu 2 người ở 2 máy khác nhau cùng sửa một đoạn code thì git không thể nào tự động merge được. Do vậy, ta phải merge bằng tay hoặc dùng tooll (difmerge).

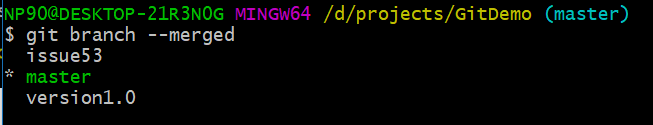
**Để xem commit gần đây nhất của mỗi nhánh**

git branch -v



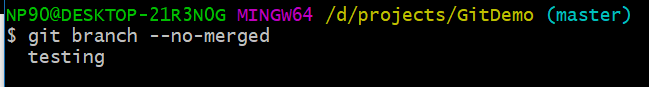
**Để xem nhánh nào đã được merged vào nhánh hiện tại.**

git branch --merged.



**Để xem những nhánh nào chưa merge.**

git branch --no-merged



**Để xóa môt nhánh**

git branch -D ten\_nhanh. (thay -D thành -d khi nhánh cần xóa đã được merge).

