# I Lý thuyết

## Điều quan trọng : Khác biệt giữa Index và WorkingTree

* + Index chính là khi bạn add . ( trạng thái chuẩn bị commit)
  + WorkingTree là toàn bộ file git quản lý hiện tại

## .gitignore ( cái này bắt buộc phải học )

* + Git rm -r --cached . ( xóa toàn bộ những thứ đã đưa vào index)
  + Git restore (--stage) tên\_file (khôi phục 1 file như ban đầu)
  + Tạo file gitignore.txt hoặc dùng lệnh **touch .gitignore**
  + Đưa file(yc đầy đủ đuôi file) /tên thư mục

## Làm sao có thể xem url của 1 remote

Gõ lệnh

* git remote get-url origin

## Làm sao để đổi tên

1.1 Đứng ở branch khác gõ lệnh này

* + - git branch -m <oldname> <newname>

<https://stackoverflow.com/questions/6591213/how-do-i-rename-a-local-git-branch>

1.2 Nếu đang đứng ở branch hiện tại thì

* + - git branch -m <newname>

2. đẩy nhánh có tên mới lên remote

- git push origin -u <newname>

3. Xóa nhánh với tên cũ đi ở remote đi

- git push origin --delete <oldname>

Done đổi tên nhánh!.

Tạo branch mới ko checkout

* + - Git branch <tên branch mới>

Còn tạo và checkout thì

* + - Git checkout –b <tên branch mới>

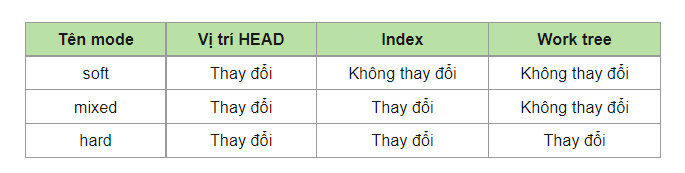
1. Làm sao để xóa 1 branch trên remote và local
   * + git push -d <remote\_name> <branchname>
     + git branch -d <branchname>

## Làm thế nào để hoàn tác 1 giao tác chưa commit lên remote

Khi bạn commit lên **Local Repository** . Nhưng muốn xóa commit gần nhất trước khi push lên **Remote Repository.**

**Trả lời:** Sử dụng lệnh Reset: Để hủy commit

Cú pháp: git reset – soft/hard/mixed HEAD ~<vitri commit>



## Làm thế nào để hoàn tác 1 commit đã push lên Remote?

Khi bạn commit lên **Local Repository** và đã push lên **Remote Repository** nhưng muốn xóa 1 commit trước đó

**Trả lời :** Sử dụng lệnh Revert

**Cú Pháp :** git revert HEAD ~<vitri commit>

## Làm thế nào để apply từ 1 commit branch này sang 1 banch khác

**Sử dụng:** Git cherry-pick

B1: Chuyển đến branch cần apply commit

B2: Thêm thay đổi và commit từ banch này.

Ví dụ

$ **git checkout master**

Switched to branch 'master'

$ **git cherry-pick 99daed2**

$ **git add sample.txt**

$ **git commit**

.

## Bạn đang phát triển chức năng trên branch-freature làm thế nào để có những thay đổi mới nhất từ branch- master về branch hiện tại?

**Sử dụng:** pull , Ta pull từ branch master về branch hiện tại

**Cú pháp :** git pull origin

Thì nó tải về thông tin và ngay lập tức merge cho nhánh đang làm việc, nó tương đương với lệnh

git fetch origin

git merge origin/master

## Conflict xảy ra khi nào ?

Khi 2 hay nhiều người cùng thay đổi 1 file

## Nêu ít nhất 1 điểm giống nhau giữa Merge và Rebase

Giống: Đều để tích hợp các nhánh với nhau

Khác:

merge: hợp nhánh thành 1 ở vị trí branch đang làm việc (hợp commit theo đúng thứ tự tg)

rebase: hợp nhánh trên cơ sở của nhánh khác. ( cái này nếu khác biệt nó sẽ tạo 2 file giống nhau)

## Fetch và Pull khác gì nhau ?

Trả lời:

• git fetch $remote\_origin tải dữ liệu từ các branch của repository trên remote\_origin và cập nhật dữ liệu này với dữ liệu của branch trên local repository.

• git fetch không cập nhật dữ liệu của working copy

• git pull $remote\_origin $branch\_name sẽ tải về từ $branch\_name sau đó merge các thay đổi từ remote này vào repository dưới local.

## Tôi vừa thay đổi file trong dự án và đã commit, bây giờ tôi lại phát hiện ra 1 lỗi nhỏ và muốn sửa file mà không thêm một commit nữa thì làm thế nào?

Trả lời: • Sử dụng git commit –amend để chỉnh sửa lại commit trước đó.

## Bạn đang đứng ở branch master làm thế nào để tạo ra một branch mới base trên branch feature/a.

Trả lời: • Sử dụng git checkout feature/a để chuyển sang branch mới ở đây là branch feature/a

## Tôi vừa thay đổi file trong dự án và đã commit nhưng chưa push lên remote, bây giờ tôi muốn hoàn tác thao tác commit và chuyền những thay đổi trong commit về lại working tree thì phải làm thế nào?

Trả lời:

• Sử dụng git reset --soft HEAD~

• Lệnh này chỉ di chuyển HEAD về vị trí commit reset. Trạng thái của stage và tất cả sự thay đổi của file sẽ được giữ nguyên

• --soft có nghĩa là bỏ commit đi nhưng giữ nguyên những thay đổi chưa được commit trong working space.

• --soft hữu dụng khi muốn giữ lại những thay đổi chưa commit cho lần commit tiếp theo

# II Tình Huống

## Exercise 1:

B1: Clone//pull về máy

git clone https://gitlab.com/tubcvt/git\_basic\_ex1.git

git pull origin master

B2: Tạo Remote

git remote add ex1 https://github.com/dungnguyen150699/Ex1.git

trong đó ex1 là tên remote

B3: Tạo branch tên f1\_NguyenTienDung // hoặc ý bài tạo branch trên remote thì:

(local)git checkout -b f1\_NguyenTienDung

(remote) git push ex1 f1\_NguyenTienDung

B4: Tạo file t1.txt "hello sep duc"

B5: Thêm vào index,và commit

git add . ; git commit -m "ex1"

B6: Thay đổi git branch f1\_NguyenTienDung trên remote

(remote) git push ex1 f1\_NguyenTienDung

Link : https://github.com/dungnguyen150699/Ex1.git

## Exercise 2:

B1: Clone//pull về máy

git clone https://gitlab.com/tubcvt/git\_basic\_ex2.git

git pull origin master

B2: Tạo nhánh trên local

git checkout –b feature\_xxxx

B3: Thêm nội dung : rocket 01h vào file rocket-01.txt

B4: index và commit

git add .

git commit –m”ex2”

B5 push lên hệ thống

git push origin –all

Link : https://github.com/dungnguyen150699/Ex2.git

## Exercise 3:

B1: Clone//pull về máy dữ liệu

git clone https://gitlab.com/tubcvt/git\_basic\_ex3.git

git pull origin master

B2: Từ nhành feature\_awesome\_cherry thực hiện lệnh stash để lưu tạm vào bộ nhớ

git stash

B3: Chuyển sang nhánh hot-fix

git checkout hot-fix

B4: Thêm dòng hot-fix-here vào file f.txt sau đó index và commit chúng

git add .

git commit -m "ex3"

B5: Trở về nhánh feature\_awesome\_cherry thực hiện lấy dữ liệu và tiếp tục làm việc

git stash pop

git add .

git commit -m "ex3"

Link : <https://github.com/dungnguyen150699/Ex3.git>

## Exercise 4:

B1: Clone//pull về máy dữ liệu

git clone https://gitlab.com/tubcvt/git\_basic\_ex4.git

git pull origin master

git pull origin feature-rocket-util

B2: Cập nhật từ nhánh master dữ liệu của nhánh feature-rocket-util bằng rebase

git rebase origin feature-rocket-util

Link : <https://github.com/dungnguyen150699/Ex4.git>

## Exercise 5:

B1: Clone//pull về máy dữ liệu

git clone https://gitlab.com/tubcvt/git\_basic\_ex5.git

git pull origin

B2: Từ nhành muốn cập nhật ( feature\_t) dùng meger

git merge origin master

Link: https://github.com/dungnguyen150699/Ex5.git

## Exercise 6:

B1: Clone//pull về máy dữ liệu

git clone https://gitlab.com/tubcvt/git\_basic\_ex6.git

git pull –all (ở trên chỉ pull branch master TT)

B2: Commit gì đó vào branch master

git commit -m "Something terribly misguided"

B3: Sử dụng reset soft để undo commit và vẫn giữ đc index và treeset

git reset solf HEAD~

## Exercise 7:

B1: Clone//pull về máy dữ liệu

git clone https://gitlab.com/tubcvt/git\_basic\_ex7.git

git pull –all

B2: Chuyển sang nhánh f1

Git checkout f1

B3:Sửa file t.txt nôi dung bất kì sau đó commit 🡪 comment lại “ Sếp Duc bong to xau trai”

B4: Sử dụng reset để xóa commit đó

Git reset hard head~

B5 Ghi lại file t.txt và commit lại

Link : https://github.com/dungnguyen150699/Ex7.git

## Exercise 8:

B1: Clone//pull về máy dữ liệu

git clone https://gitlab.com/tubcvt/git\_basic\_ex8.git

git pull –all

B2: Sau khi pull về ta sử dụng câu lệnh sau để check các commit

Git log –oneline

Hiện ra 5 commit

B3 Sử dụng câu lệnh để xóa 3 bad commit

Git reset –hart soft head~3 để xóa 3 bad commit gần nhất

Link : https://github.com/dungnguyen150699/ex8.git

## Exercise 9:

B1: Clone//pull về máy dữ liệu

git clone https://gitlab.com/tubcvt/git\_basic\_ex8.git

git pull –all

B2:từ nhánh f1 merge nhánh f2

Git merge f2

B3 xảy ra conflict thì phai vào file utils.txt sửa nội dung ( cụ thể là xóa comment thông báo conflict của tool git). Cuối cùng add . và commit lại

Link : https://github.com/dungnguyen150699/ex9.git