# GIT TUTORIAL

## Command Prompt/Terminal:

### Tạo thư mục abc:

mkdir abc

### Chuyển dấu nháy lệnh vào thư mục abc:

cd abc

## Git command:

### Khởi tạo một git repository trên local hoặc khởi tạo lại một git repository (repos) đã có:

git init

### Cấu hình thông tin cho git user:

Ví dụ: git config user.name “Le Minh Tri” => cấu hình tên người dùng Git

Ví dụ: git config user.email “ansidev@gmail.com” => cấu hình email người dùng Git

### Add một git remote vào dự án hiện tại

git remote add [short-name] [repos-url]

[short-name]: Tên rút gọn cho remote, tùy bạn đặt

[repos-url]: url của git repos muốn add

### Set remote mặc định khi push code bằng lệnh git push

git branch --set-upstream-to=[remote-name]/[branch-name]

Ví dụ: git branch --set-upstream-to=origin/master sẽ thiết lập push code lên branch master của remote origin.

### Lấy code từ remote về máy tính, có 2 cách:

Cách 1:

git pull [remote-name] [branch]

Các tham số trong dấu [ ] không nhất thiết phải cung cấp. Khi chạy lệnh trên không có tham số thì git sẽ sử dụng remote mặc định mà bạn thiết đặt. Thường thì chúng ta hay dùng git pull (không có tham số)

Tác dụng của lệnh git pull là nó sẽ lấy code về máy tính và thực hiện merge các thay đổi vào code hiện tại trên máy tính của bạn.

Cách thứ 2 sử dụng để lấy code về máy tính là:

git fetch [remote-name]

Lệnh trên cũng lấy code từ git repos về máy tính nhưng nó sẽ không cập nhật các thay đổi vào code hiện tại.

Bạn muốn merge các thay đổi này thì dùng:

git rebase

Tác dụng của lệnh này là sẽ cập nhật các thay đổi trên máy tính của bạn kể từ lần cuối bạn lấy code về máy tính lên top.

Để hiểu rõ hơn, xem <http://git-scm.com/docs/git-rebase>,

### Làm việc với code trên local:

Sau khi lấy code từ git repos về máy tính, bạn sửa code gì đó, bạn muốn xem trạng thái code hiện tại, sử dụng lệnh:

git status

Màn hình sẽ hiển thị tình trạng code hiện tại của bạn.

Những dòng màu đỏ, file/folder bạn tạo mới: => unstaged.

Bạn muốn cập nhật các thay đổi của mình để push code lên git repos, dùng lệnh

git add . (thêm tất cả các file đã được thêm/xóa/sửa)

Nếu muốn thêm từng file, thực hiện lệnh:

git add [path-to-file]

Đối với file bạn xóa thì dùng lệnh

git rm [path-to-file]

Những dòng màu xanh lá cây: những file bạn thêm/xóa/sửa nhưng đã được add. => staged

Chuyển file từ trạng thái staged sang unstaged:

git checkout . hoặc git checkout [path-to-file]

Commit file: (áp dụng với staged file)

git commit –m ‘Message của bạn (mô tả về thay đổi của commit)’

Sau khi đã commit thì khi chạy git status bạn sẽ không thấy file đó xuất hiện trong thông báo trên màn hình.

Nếu đã commit hết tất cả các file thì lúc này branch của bạn đang ở trạng thái commit.

Undo commit, dùng lệnh (khuyến cáo cẩn thận khi dùng vì có thể công sức cả mấy tiềng đồng hồ sẽ bay theo gió)

git reset

Chi tiết các tham số và cách dùng: <http://git-scm.com/docs/git-reset>.

### Backup code

Giả sử bạn đang chỉnh code nhưng chưa commit nhưng lại muốn lấy code về để merge, nhưng nếu làm vậy thì thì những thay đổi chưa commit sẽ mất hết. Giải pháp cho tình huống này:

git stash

Đại khái tác dụng của lệnh này là backup những thay đổi của code trên máy tính của bạn vào một vùng riêng để sau này phục hồi lại sau khi đã lấy code từ git repos về và merge vào. Có thể có nhiều tình huống bạn muốn dùng lệnh này.

Sau khi dùng git stash, lấy code về bằng lệnh git fetch và git rebase. Lúc này thay đổi code trên git repos (do các thành viên khác trong team push lên) đã được cập nhật vào code của bạn. Lúc này bạn sẽ phục hồi những thay đổi lúc nãy khi dùng git stash bằng lệnh:

git stash apply

### Clear code trong stash (nên làm khi branch đang ở trạng thái commit, tránh phát sinh lỗi sau này)

git stash clear

Nếu xảy ra conflict (thường khi bạn và một thành viên nào đó cùng chỉnh sửa một file, git sẽ tự động merge theo cơ chế auto, nhưng nếu auto-merge không thành công, lúc này sẽ xảy ra conflict code, màn hình sẽ thông báo những file nào bị conflict), bạn chỉnh sửa những file bị conflict sau đó thực hiện add file

git add <file vừa sửa>

Nếu không xảy ra conflict thì commit luôn sau đó push thay đổi lên git repos

git commit -m "<message>"

### Tách branch:

Branch mặc định của một git repos khi khởi tạo là master

Ví dụ bạn được giao phát triển tính năng A, một thành viên khác được giao phát triển tính năng B, nếu làm trên cùng một branch thì sẽ dễ xảy ra lỗi, giải pháp lúc này là tách ra một nhánh mới để làm riêng các tính năng, sau khi đã ổn định thì merge lại vào nhánh chính

git branch -b [branch-name]

### Chuyển nhánh

git checkout [branch-name] (yêu cầu branch bạn đang làm việc phải ở trạng thái commit)

Đang đứng ở branch nào thì có thể merge branch khác vào bằng lệnh :

git merge [tên branch muốn merger vào branch hiện tại]

hoặc dùng

git rebase [tên branch muốn merger vào branch hiện tại]

Mình thường dùng lệnh số 2, và để chắc ăn nên back up code của bạn trước khi thực hiện vào stash sau đó restore

### Clone source: Lấy source của người khác

git clone [repos-url]

Tác dụng: clone project về và add 1 remote là nơi mình clone.

Còn nhiều lệnh khác nữa, càng làm nhiều thì bạn sẽ khám phá ra chúng.

Tài liệu có thể đọc thêm tại: <http://git-scm.com/doc>