

Sổ tay Git

Đây là một hướng dẫn đơn giản cho người mới học ;)

Tweet

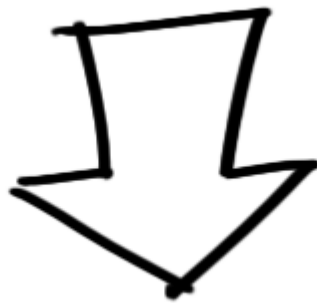
by Roger Dudler

credits to @tfnico, @fhd and Namics

in deutsch, español, français, indonesian, italiano, nederlands, polski, português, русский

မြန်မာ, 日本語, 中文, 한국어 Vietnamese

vui lòng báo cáo vấ'n đề` trên github



cài đặt

Tải git về cho OSX

Tải git về cho Windows

Tải git về cho Linux

tạo một repository mới

Để tạo 1 repository mới, bạn hãy mở cửa sổ lệnh và gõ dòng lệnh sau

```
git init
```

sao chép (clone) một repository

để clone 1 repository có sẵn ở trên máy cục bộ, bạn hãy sử dụng dòng lệnh sau:

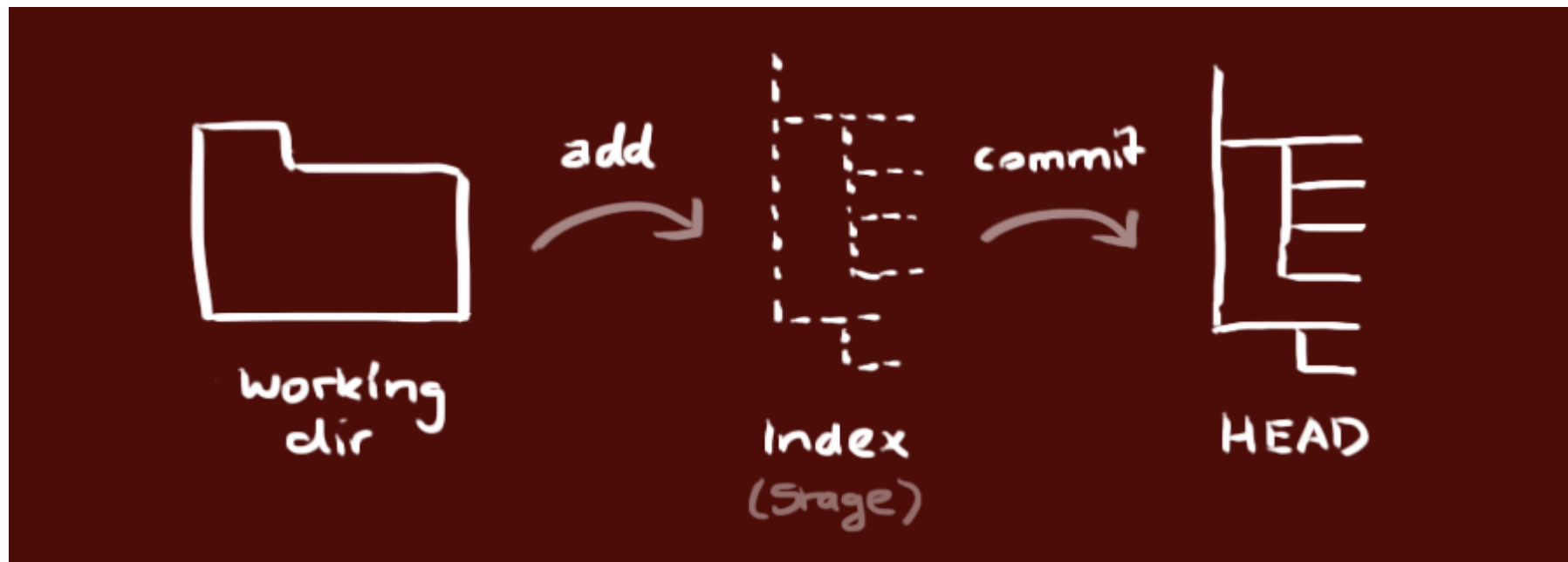
```
git clone /đường-dẫn-đến/repository/
```

Nếu repository đó ở máy chủ khác thì bạn hãy gõ dòng lệnh sau:

```
git clone tênusername@địa chỉ máy chủ:/đường-dẫn-đến/repository
```

quy trình làm việc

thư mục cục bộ của bạn bao gồm ba "trees" được duy trì bởi git. đầu tiên là **Thư Mục Đang Làm Việc (Working Directory)** có chứa các tập tin hiện tại. cái thứ hai là **Chỉ Mục (Index)** đóng vai trò như staging area và cuối cùng là **HEAD** trỏ đến commit gần đây nhất của bạn.



thêm (add) & commit

Bạn có thể đề xuất thay đổi (thêm nó vào chỉ mục **Index**) bằng cách

```
git add <tên-tập-tin>
```

```
git add *
```

Đây là bước đầu tiên trong quy trình git cơ bản. Để thật sự commit những thay đổi, bạn sử dụng

```
git commit -m "Ghi chú Commit"
```

Bây giờ thì tập tin đã được commit đến **HEAD**, nhưng chưa phải trên thư mục remote.

đẩy (push) các thay đổi

Thay đổi của bạn hiện đang nằm tại **HEAD** của bản sao cục bộ đang làm việc. Để gửi những thay đổi đó đến repository remote, bạn thực thi

```
git push origin master
```

Thay đổi *master* bằng bất cứ nhánh nào mà bạn muốn đẩy những thay đổi đến.

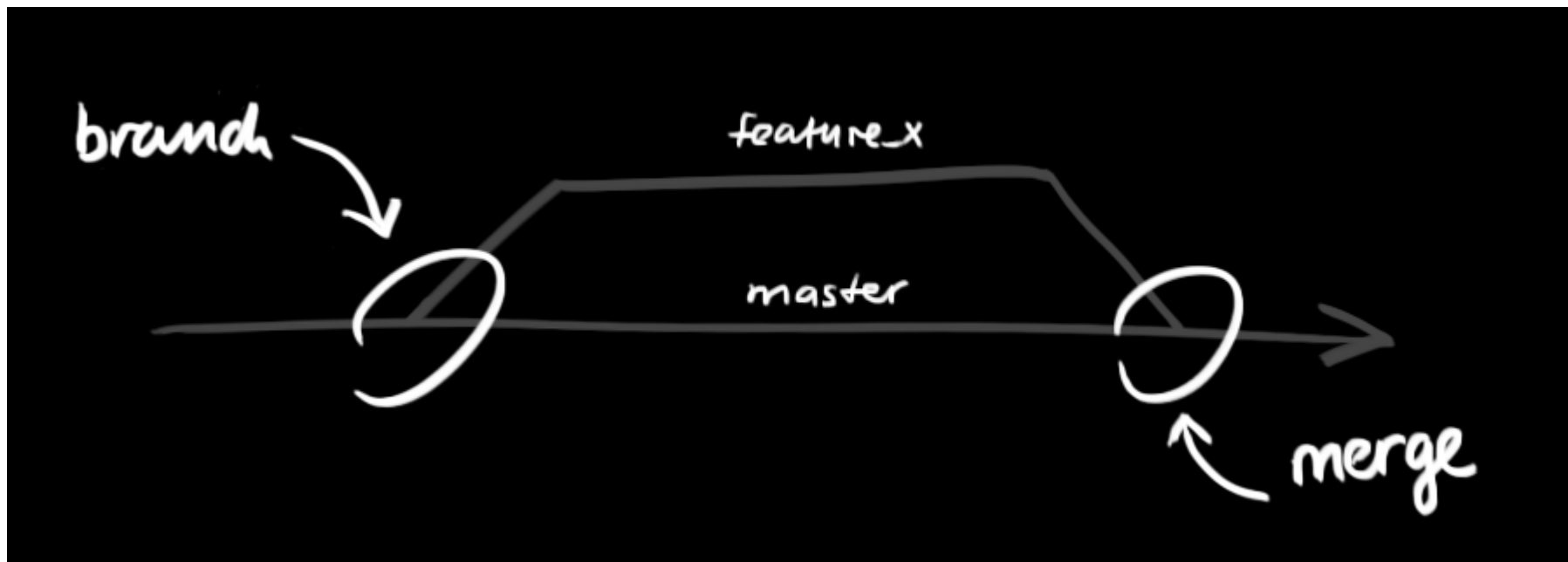
Nếu bạn chưa clone một repository hiện có và muốn kết nối repository của bạn đến máy chủ remote, bạn phải thêm nó với

```
git remote add origin <máy-chủ>
```

Bây giờ bạn đã có thể đẩy các thay đổi của mình vào máy chủ đã chọn

nhánh

Các nhánh (branches) được dùng để phát triển tính năng tách riêng ra từ những nhánh khác. Nhánh *master* là nhánh "mặc định" khi bạn tạo một repository. Sử dụng các nhánh khác tri đang trong giai đoạn phát triển và merge trở lại nhánh master một khi đã hoàn tất.



tạo một nhánh mới và đặt tên là "feature_x" và chuyển qua nhánh đó (từ master) bằng cách

`git checkout -b feature_x`

trở lại nhánh master

`git checkout master`

và xóa nhánh feature_x đó lên nữa

```
git branch -d feature_x
```

một nhánh *không có giá trị với các nhánh khác* trừ khi bạn đẩy nhánh đó về remote repository

```
git push origin <nhánh>
```

cập nhật & trộn (update & merge)

để cập nhật repository cục bộ của bạn và commit mới nhấ t, thực thi

`git pull`

trong thư mục đang làm việc để *lấ y về` (fetch)* và *trộn (merge)* các thay đổi ở remote.

để trộn một nhánh khác vào nhánh đang hoạt động (vd: master), sử dụng

`git merge <nhánh>`

trong cả hai trường hợp, git cố gắ ng trộn tự động (auto-merge) các thay đổi. Không may, điề u này không phải lúc nào cũng làm được và thường dẫn đế n *xung đột*. Trách nhiệm của bạn là trộn *các xung đột* đó thủ công bằ ng cách chỉnh sửa các tập tin được hiển thị bởi git. Sau khi thay đổi, bạn phải đánh dấ u chúng là đã được trộn (merged) với lệnh

`git add <tên-tập-tin>`

trước khi trộn các thay đổi, bạn có thể xem trước chúng bằng các

```
git diff <nhánh_nguồn> <nhánh_mục_tiêu>
```

gắn nhãn (tagging)

người ta khuyên nên tạo nhãn (tags) khi phát hành phần mềm. đây là khái niệm được biết đến, đã từng có trên SVN. Bạn tạo tag mới tên là

1.0.0 bằng cách

```
git tag 1.0.0 1b2e1d63ff
```

chuỗi *1b2e1d63ff* là 10 ký tự đầu tiên của mã commit (commit id) mà bạn muốn tham chiếu đến bằng nhãn của bạn. Bạn có thể lấy mã commit với lệnh

git log

bạn cũng có thể sử dụng ít ký tự hơn từ mã commit, nó chỉ cần phải là duy nhất.

thay thế các thay đổi cục bộ

Trong trường hợp bạn làm sai điều gì đó, bạn có thể thay thế các thay đổi cục bộ bằng lệnh

```
git checkout -- <tên-tập-tin>
```

lệnh này thay thế những thay đổi trong "tree" đang làm việc với nội dung mới nhất của HEAD. Các thay đổi đã được thêm vào chỉ mục, kể cả các tập tin mới, điều này sẽ được giữ lại.

Nếu bạn muốn hủy tất cả thay đổi và commit cục bộ, lấy về (fetch) lịch sử gần đây nhất từ máy chủ và trở nhánh master cục bộ vào nó như sau

```
git fetch origin
```

```
git reset --hard origin/master
```

các gợi ý hữu ích

git GUI tích hợp sẵn

`gitk`

sử dụng kết quả git với nhiều màu

`git config color.ui true`

hiện log trên chỉ một dòng mỗi commit

`git config format.pretty oneline`

sử dụng thêm tập tin tương tác

`git add -i`

các liên kết và tài nguyên

trình sử dụng có đồ họa

GitX (L) (OSX, open source)

Tower (OSX)

Source Tree (OSX, free)

GitHub for Mac (OSX, free)

GitBox (OSX, App Store)

hướng dẫn

Git Community Book

Pro Git

Think like a git

GitHub Help

A Visual Git Guide

Learn Git Branching

More Git and GitHub Secrets

Understanding Git Conceptually

misc

ungit - The easiest way to use git. On any platform. Anywhere.

bình luận

0 Comments

git - the simple guide

 Login ▾ Recommend 8 Tweet Share

Sort by Newest ▾



Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Be the first to comment.

 Subscribe Add Disqus to your siteAdd DisqusAdd Disqus' Privacy PolicyPrivacy PolicyPrivacy