Git Beginner Guide

Jaewoong Lee

Ulsan National Institute of Science and Technology

jwlee230@unist.ac.kr

February 3, 2020

Contents

1. Introduction
2. Basic Step
3. Advanced Step

Introduction

In this guide, we will discuss about followings:

1. Git
2. GitHub
3. GitHub Desktop (`https://desktop.github.com`)

As the other program does, Git is basically controlled CLI. But, I don't want to go harder. In this guide, I will use GUI mainly.

Git?

Figure: Git

Git is Version Control System (VCS) made by Linus Torvalds.

VCS?

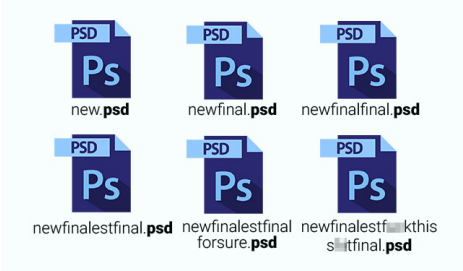


Figure: Without VCS

VCS? *(Cont.)*

With VCS, you can get advantages like:

- Revision Control
- Version Control
- Backup & Restore
- Collaboration

GitHub?

"GitHub is how people build software."
Advantages:

- Free to personal usage.
- Many open source programs are managed by GitHub.
- Issue tracker: you can track the issue of your program

Git Repository

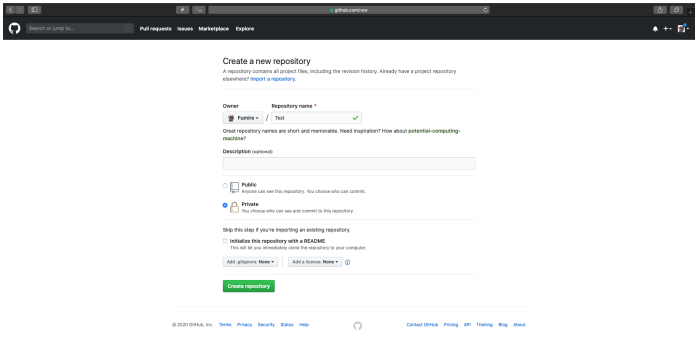
Git Repository means where git save files. There are two types of repository:

1. Remote Repository
2. Local Repository

Usually,

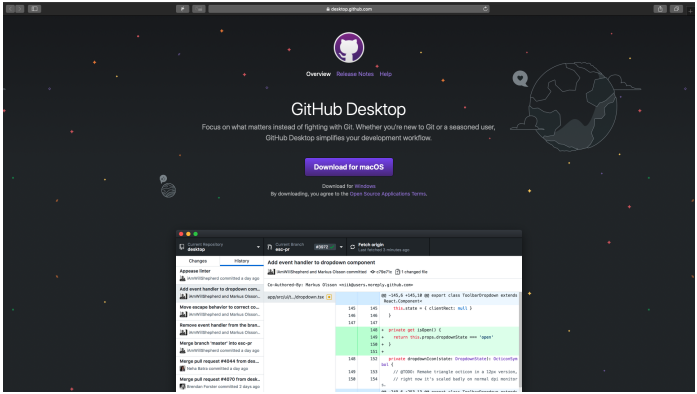
1. you *clone* (download) files from remote repository;
2. edit files on local repository;
3. and, *push* (upload) to remote repository.

Practice 01



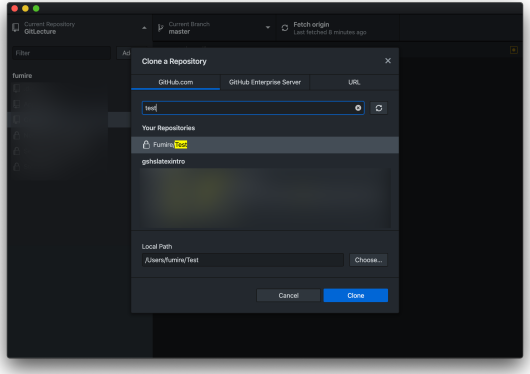
Register **GitHub**, and make a repository with named 'Test'.

Practice 02



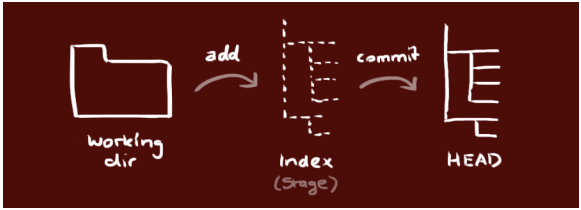
Download & Install 'GitHub Desktop' which gives GUI control with git.

Practice 03 (Clone)



Clone the repository from GitHub as figure.

Trees

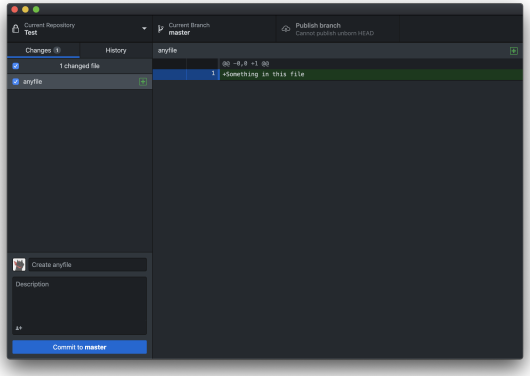


There are three tree which managed by git.

1. Working Directory: which consist of real files
2. Index: staging area (ready area)
3. HEAD: the final files

You can *add* any files from working directory to index.
Also, you could *commit* changes from index to HEAD.
You could add *tag* to commit.

Practice 04

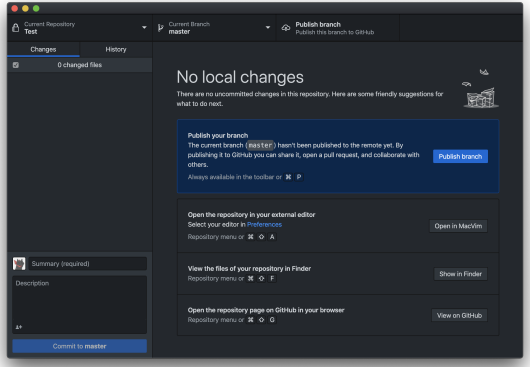


Add any file to working directory, then GitHub Desktop automatically finds the changes as figures. Commit the changes.

Push

However, even you commit the changes, the changes are not applied to remote repository.
The changes are only in local repository.
To apply changes, you should *push* the changes to remote repository.

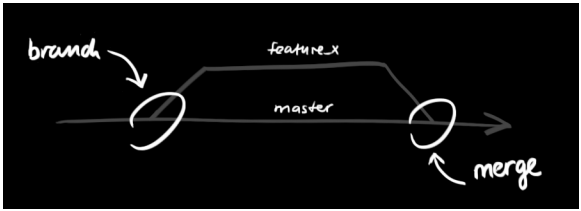
Practice 05



Let's push the changes to remote repository.

Branch / Merge

You can *branch & merge* the changes.
The **master** branch will be automatically generated when creating repository.



You can add/delete branches; and move among the branches.

Conflict

Git automatically try to merge changes.
However, sometimes the *conflict* occurs; in other words, you should solve the twisted.
After you solve the twisted, add/commit the solved as other changes.

Pull

For update as remote directory, you should *pull* the repository.
With *pull* command, the changes of remote directory are *fetched* and *merged*.
Sometimes, as *merging*, conflict can be occurred, and you should solve this.

Advanced Step

After this page, you will get advanced step for git.

.gitignore

You can specify the files which you do not want to upload to git.

```
# : comments

# no .a files
*.a

# but do track lib.a, even though you're ignoring .a files above
!lib.a

# only ignore the TODO file in the current directory, not subdir/TODO
/TODO

# ignore all files in the build/ directory
build/

# ignore doc/notes.txt, but not doc/server/arch.txt
doc/*.txt

# ignore all .pdf files in the doc/ directory
doc/**/*.pdf
```

`https://www.gitignore.io` gives the basic .gitignore file.

.gitkeep

Similar with .gitignore.
Git does not track empty directory, so add empty file to track/*keep* the directory.

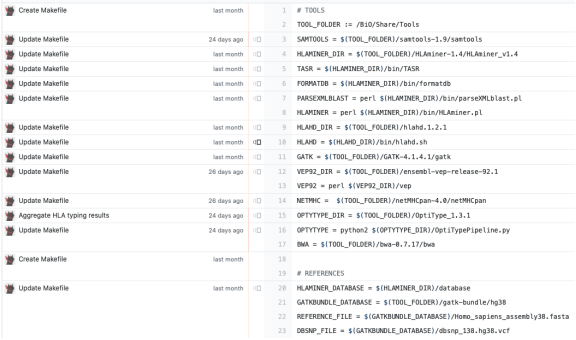
Pull Request

When you are in collaboration and you have developed an important feature, you can send PR to your teammates to use the important features.

Fork

You can fork(copy) other's work to your repository.
Fork is similar to Clone; however, you cannot change directly to original repository with fork.
Clone: Two-way $\Leftrightarrow$ Fork: One-way
Still, you can send PR to forked repository.

Blame



With Blame, you can find who write the code line-by-line.

Blame *(Cont.)*

You easily find who will be blamed or have responsibility to make the bugs.

Git - Large File Storage

References

- Git - The Simple Guide: `https://github.com/rogerdudler/git-guide`
- Apply Git .gitignore: `https://nesoy.github.io/articles/2017-01/Git-Ignore`