



2021/08/02

# A Short Session on Git

DreamOnline



Git is the most commonly used version control system. Git tracks the changes you make to files, so you have a record of what has been done, and you can revert to specific versions should you ever need to. Git also makes collaboration easier, allowing changes by multiple people to all be merged into one source.

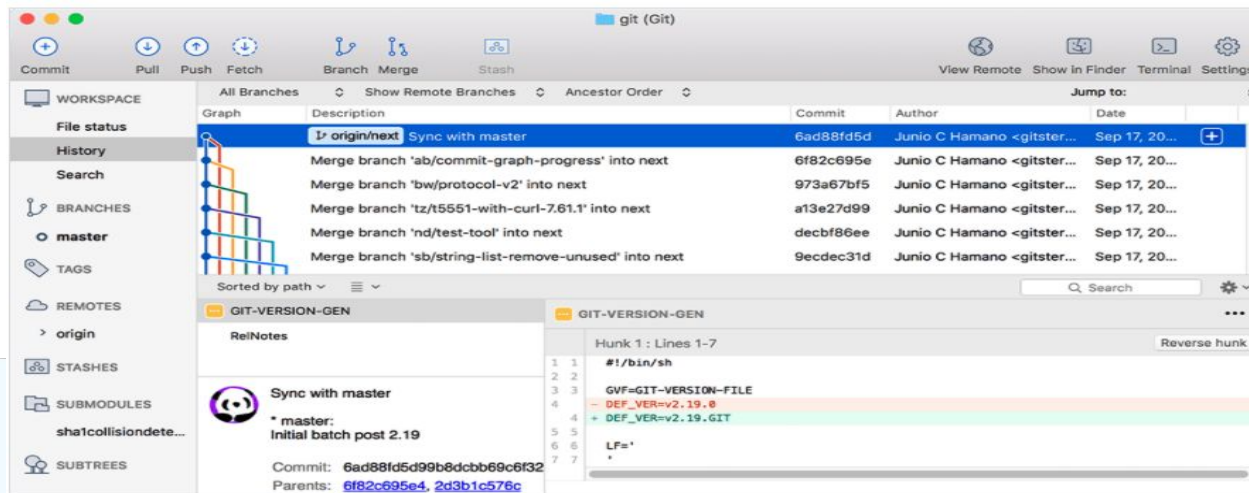
Git is software which need to be installed in your system.  
And you need to configure your git account.

Configuration:

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"  
git config --global user.password "your password"
```

you can access via

1. A command line (terminal),
2. Desktop app that has a GUI (graphical user interface) such as [Sourcetree](#) shown below.



A Git repository (or repo for short) contains all of the project files and the entire revision history. Git initialization command creates a folder named `.git`, which contains all of the Git metadata for tracking changes.

Command : **git init**

Think of Git as keeping a list of changes to files. Each recorded change to a file or set of files is called a commit.

Before we make a commit, we must tell Git what files we want to commit. This is called staging and uses the **add** command.

Commands: **git add fileName / git add .**

## Why can't we just commit the file directly?

Let's say you're working on a two files, but only one of them is ready to commit. We add files to a staging area, and then we commit the files that have been staged.

Storing a copy of your Git repo with an online host (such as Backlog, [GitHub](#) or [Bitbucket](#)) gives you a centrally located place where you can upload your changes and download changes from others, letting you collaborate more easily with other developers.

Commands:

```
git remote add origin https://github.com/taruncse/GitExperiment.git  
git push -u origin master
```

A **branch** is a version of the repository that diverges from the main working project.  
The default **branch** name in **Git** is **master**

Create a branch : **git branch branch\_name**

Checkout a branch : **git checkout branch\_name**

1. Git Conflict
2. Git merge



????