# Git and Github

project with the latest technology

**Lee Hae Joon**

# Contents

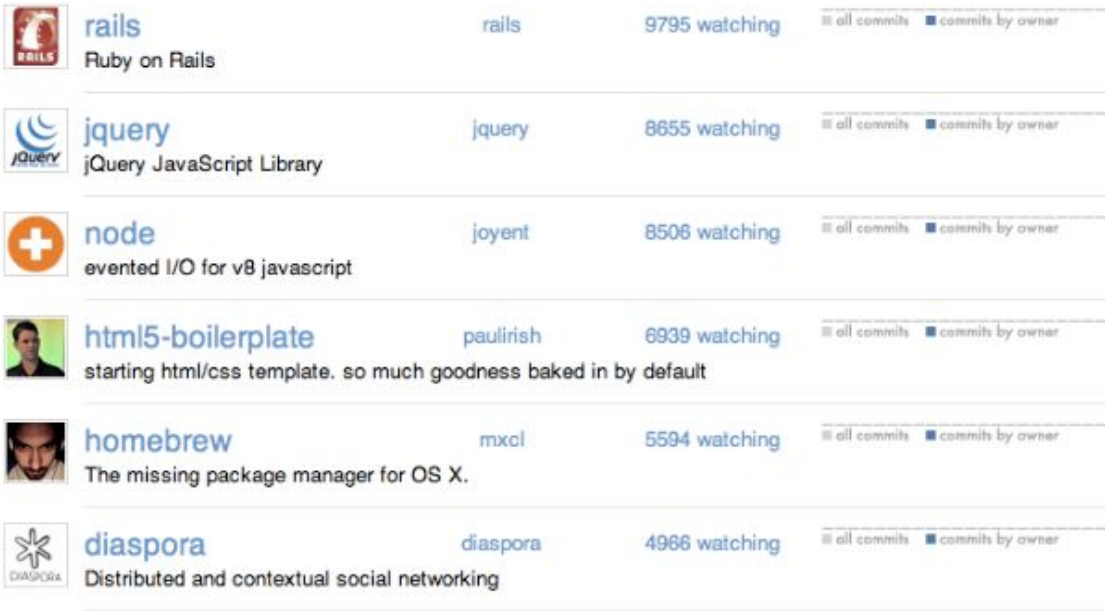
- Motivation
- Git
- Github
- Hack it

# Motivation

Open Source ?

Version Control ?

# Open Source ?




**Bootstrap, Hadoop, Spark, Flink, Rails, jQuery ..**

# Go to Open Source Repository

## Your preference?

Web Server ? - Node JS, jQuery, Rails..

Distributed System or Big Data - Hadoop, Zeppelin, Spark

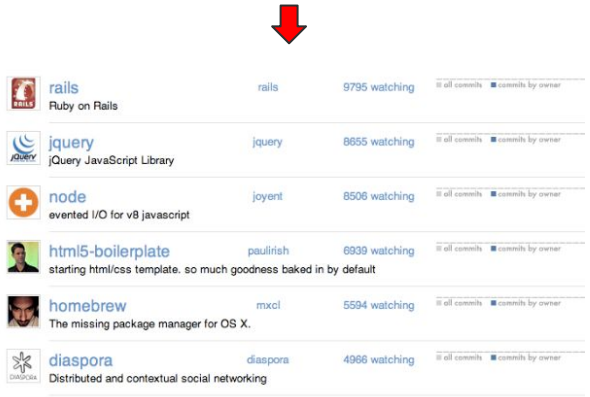
Image Processing - GPUImage

.. Search for everything

# Open Source ?

Most popular projects

Belong to Github !

# Saying ...

If you want to make successful projects with your team, you must handle **Github and Git**

# What is Version Control ?

**5 minutes Talks ..**

**Have you ever handled version control for your project before ?!**

# Version Control ?

Probably, you used to share your projects files via emails or cloud system, e.g., dropbox, google drive

# Version Control ?

## What about these cases ?

- Your team modified a few of files while you are coding the files.
- You want to make a snapshot of changed files just in case.
- You crashed important codes without noticing.
- You want to look at the changed codes your team made.

etc … ?

# Version Control in the situations ?

- Version control systems keep **revisions straight**, and **store** the modifications in a central repository.
- This allows developers to easily **collaborate**, as **downloading** a new version of the software, make **changes**, and **upload** the newest revision.

# Now,
# Understand Version Control ?

# Saying ...

If you already use **Subversion** ?

read this article - http://www.slideshare.net/einsub/svn-git-17386752


vs

# Git ? Github ?

# Git ?

## History

- Git is an open-source version control system that was started by Linus Trovalds – the same person who created “ ? ”.
- Changed files can be stored in **local** file system.

# Github ?

**Git** is a command-line tool.

**GitHub** is a web-based **Git repository** hosting service, where you can store projects and network with likeminded people.

# Github ?

Github isn’t just for developers !

- GitHub can be used for any types of files

  if you have a team that is constantly making changes to a **word document**.

  E.g., project proposal

# Hack Git

## BTW

# sudo apt-get install zsh
# sudo apt-get install vim build-essential

curl -L https://raw.github.com/robbyrussell/oh-my-zsh/master/tools/install.sh | sh

>> zsh

>> vim ~/.zshrc

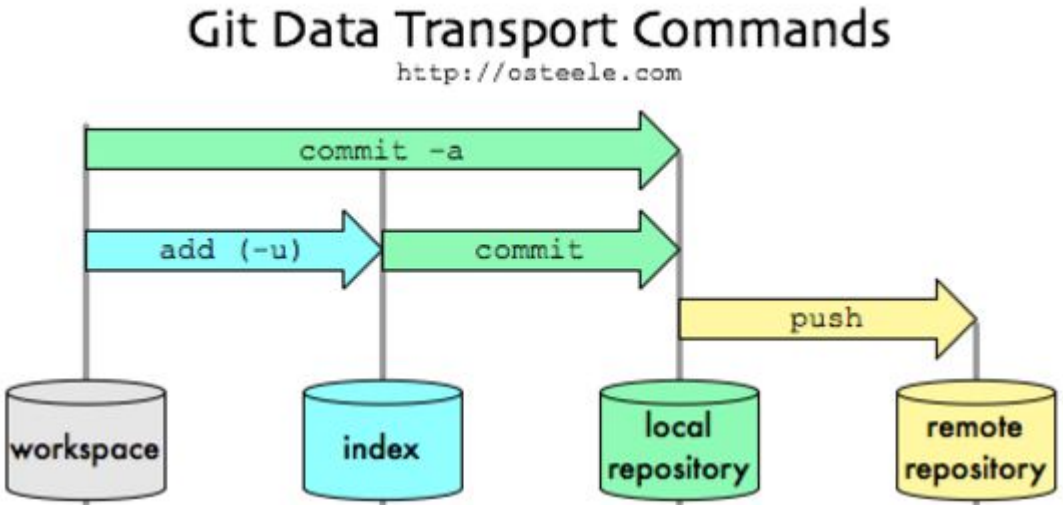
ZSH_THEME="pygmalion" | You can choose your theme up to your favor

https://github.com/robbyrussell/oh-my-zsh/wiki/themes

Reference : https://github.com/robbyrussell/oh-my-zsh

# Hack Git

## 3 STEPS



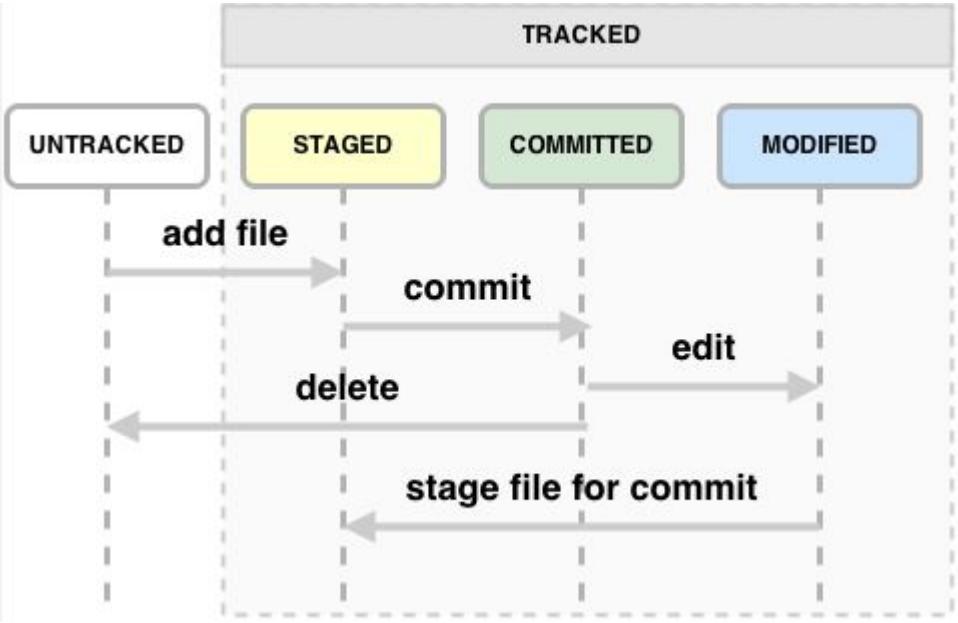
# Hack Git

Tracked Files

**Staged** - modified file has been marked to go into next commit

**Committed** - file's safely stored in your local database

**Modified** - file's been changed, but not committed



# Hack Git

Sudo apt-get update

Sudo apt-get install build-essential git

# Hack Git

>> git clone **GITHUB_URL**

Cloning some repository to your local system

# Hack Git

Make your own repository in **KHUHUB**

# Hack Git

>> git init

>> vim temp_file

In temp file, type “Blah” :wq >> git status >> git add temp_file >> git status

>> git commit -m “first commit” >> git log

Initializes a new Git repository. You run this command inside a repository or directory. Only after you input this does it accept further Git commands

# Hack Git

>> git add

This does not add new files to your repository. Instead, it brings new files to Git’s attention. After you add files, they’re included in Git’s “snapshots” of the repository.

# Hack Git

>> git commit

Git’s most important command. After you make any sort of change, you input this in order to take a “snapshot” of the repository.

# Hack Git

>> git push

If you’re working on your local computer, and want your commits to be visible online on GitHub as well, you “push” the changes up to GitHub with this command.

# Hack Git

>> git pull

If you’re working on your local computer and want the most up-to-date version of your repository to work with, you “pull” the changes down from GitHub with this command.

# Hack Git

>> git checkout

This is a navigational command that lets you move to the repository you want to check. You can use this command as ‘git checkout master’ to look at the master branch.

# Hack Git

>> git branch

This command will let you build a new branch, or timeline of commits, of changes and file additions that are completely your own. Your title goes after the command.

# Hack Git

>> git merge

When you’re done working on a branch, you can merge your changes back to the master branch, which is visible to all collaborators.

# Hack Git

>> git log

Browse your commit files which are snapshots stored in your local repository

# Hack Github

Goto > www.github.com

Sign up !!

# Hack Github

## Try Forking !

is when you create a new project based off of another project that already exists. This is an amazing feature that vastly encourages the further development of programs and other projects.

## GITHUB_URL

# Hack Github

Pull Request !

After **forking** a repository, make a **great revision** to the project, and want it to be **recognized** by the original developers, included in the official repository.

By creating a pull request, the original authors can see your work, and then can accept it into the official project.

**COMMITER**

# Hack Github

Make Issue !

**GITHUB_URL**

# Hack Github

Make Label !

**GITHUB_URL**

# Practice

1. Collaborate your team in **Github**
2. Look at your commit files in **Source Tree**
3. Play **git game**

# PRACTICE 1 (group work)

1. Group 4 students surrounding you
2. **One Stud** make a public repository for your team
3. Collaborate your members
4. **One Stud** make README.md and make issue
5. Git clone ‘your team repository’
6. Vim a file named ‘studentid_number’
7. Commit and Push
8. Git pull again

# PRACTICE 1

9. Make Your Branch and Vim a file named ‘english_name’

   - Git checkout -b ‘studentid_number’ & Vim Philip

10. Upload your branch to remote repository for your team

   - Git push -u origin ‘studentid_number’

11. Check your repository in
    your **Github and git pull and checkout your team branch**

# PRACTICE 1- 2

1. **One Stud** make Issue for your team
2. Use **Markdown**
   https://guides.github.com/features/mastering-markdown/
3. Leave commit message about what you've done
   - check your commit hash value >> git log

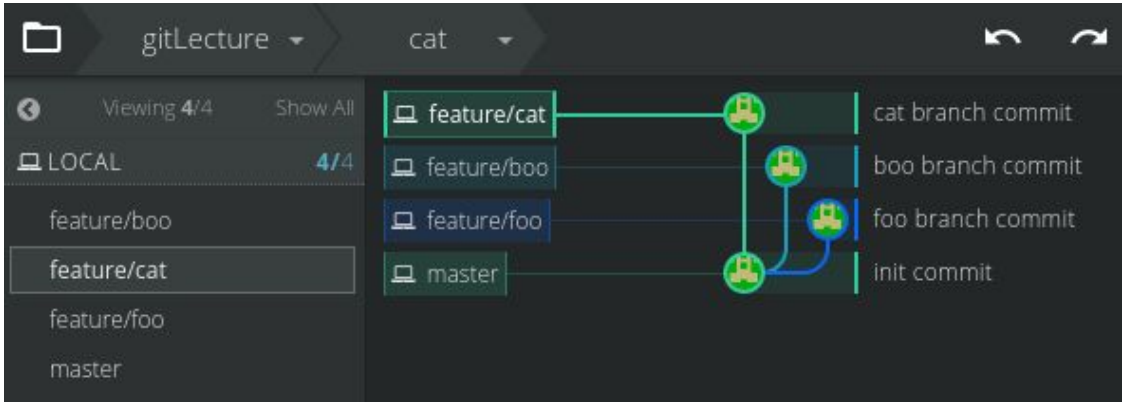
# PRACTICE 2 (individual work)

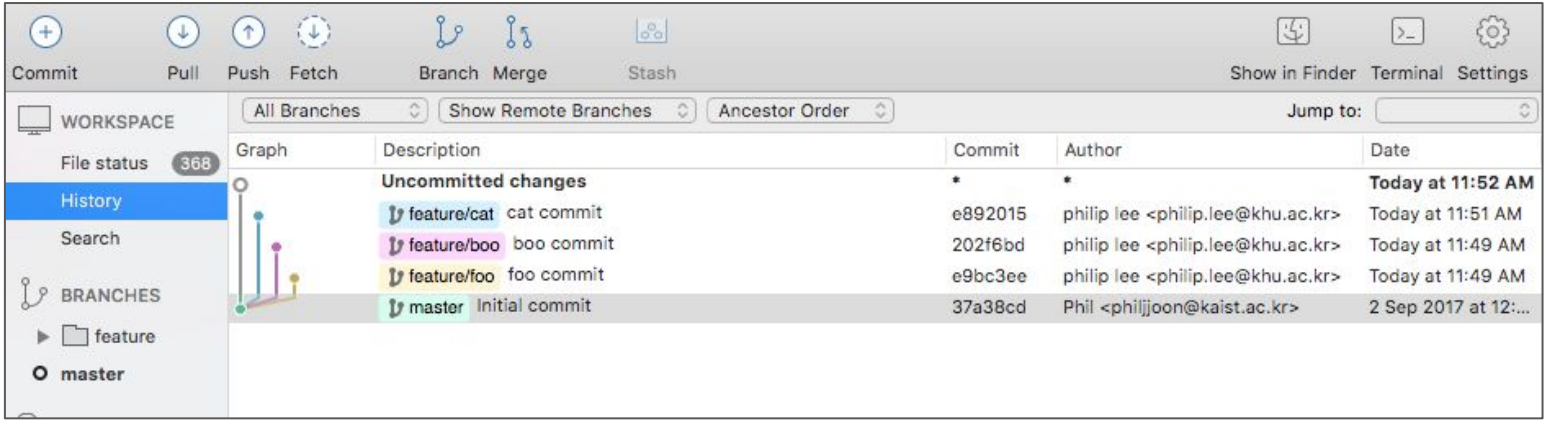
Look at your commit files in **Source Tree**

# PRACTICE 2


1. setup source tree
   - https://www.gitkraken.com/
2. make your own repository in Github & Git clone it
3. (git commit random_file# / git branch ) * 3
   - e.g., feature/foo ,  feature/boo, feature/cat
   - **ATTENTION) move to master**
4. check your source tree

# PRACTICE 2


gitLecture
cat
Viewing 4/4
Show All
LOCAL
4/4
feature/boo
feature/cat
feature/foo
master
feature/cat
feature/boo
feature/foo
master
cat branch commit
boo branch commit
foo branch commit
init commit


Commit
Pull
Push
Fetch
Branch
Merge
Stash
Show in Finder
Terminal
Settings
WORKSPACE
File status 368
History
Search
BRANCHES
feature
master
All Branches
Show Remote Branches
Ancestor Order
Jump to:
Graph
Description
Commit
Author
Date
Uncommitted changes
feature/cat cat commit e892015 philip lee <philip.lee@khu.ac.kr> Today at 11:51 AM
feature/boo boo commit 202f6bd philip lee <philip.lee@khu.ac.kr> Today at 11:49 AM
feature/foo foo commit e9bc3ee philip lee <philip.lee@khu.ac.kr> Today at 11:49 AM
master Initial commit 37a38cd Phil <philjjoon@kaist.ac.kr> 2 Sep 2017 at 12:...

# Hack Git

## git branch -r | git branch -a

```
phil@Philip-Pro:test/python_machine_learning <master>$ git remote update
Fetching origin
phil@Philip-Pro:test/python_machine_learning <master>$ git branch
* master
phil@Philip-Pro:test/python_machine_learning <master>$ git branch -r
  origin/HEAD -> origin/master
  origin/feature/boo
  origin/feature/cat
  origin/feature/foo
  origin/master
phil@Philip-Pro:test/python_machine_learning <master>$ git branch -a
* master
  remotes/origin/HEAD -> origin/master
  remotes/origin/feature/boo
  remotes/origin/feature/cat
  remotes/origin/feature/foo
  remotes/origin/master
```

# PRACTICE 3

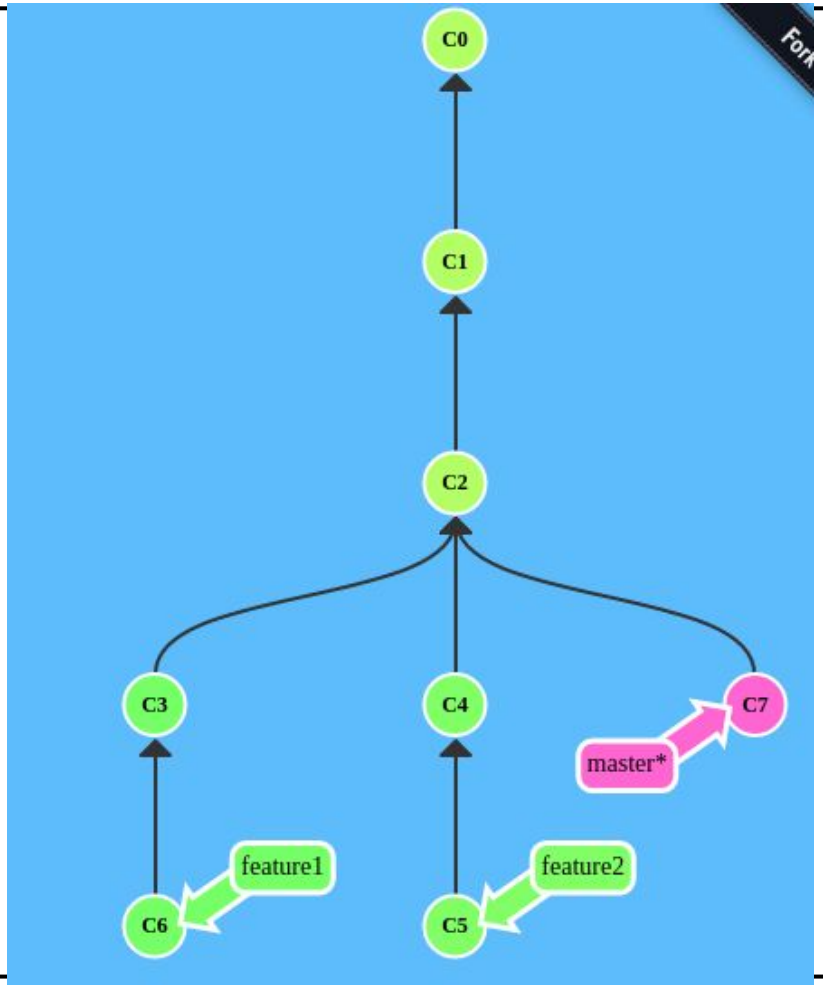
Easily to be familiar **Git Command**

http://learngitbranching.js.org/

# PRACTICE 3

Use only cmd

commit and branch and checkout

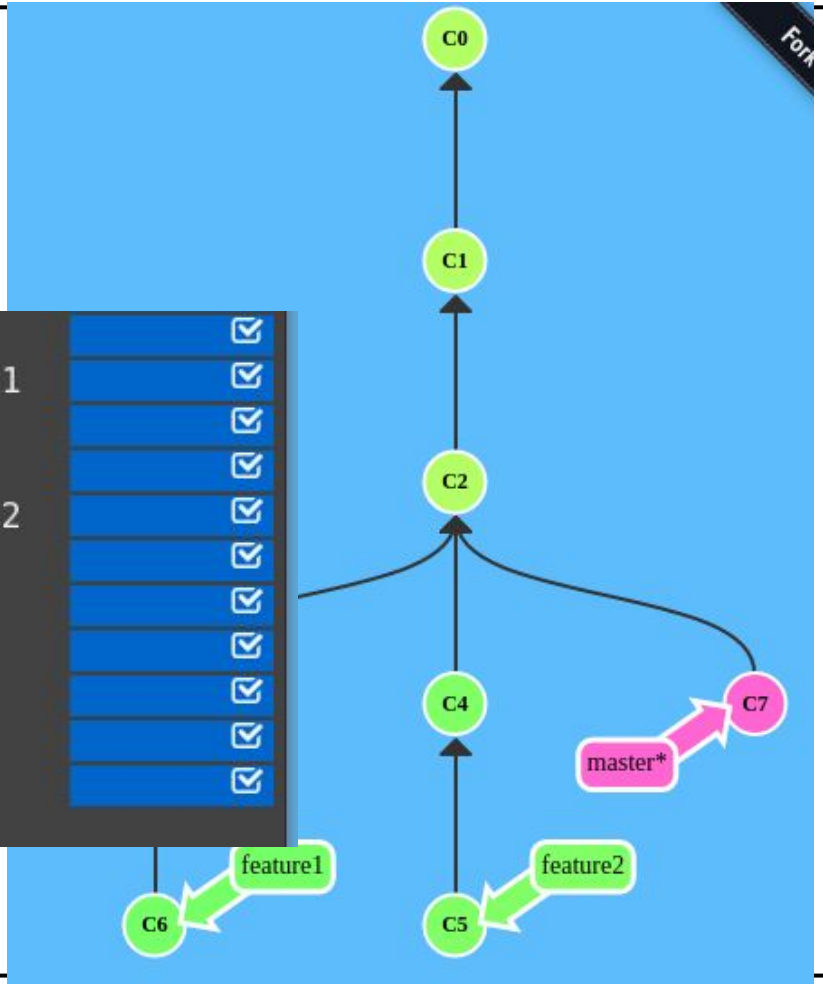


# PRACTICE 3

Use only cmd

commit and bra

```
$ git commit
$ git checkout -b feature1
$ git commit
$ git checkout master
$ git checkout -b feature2
$ git commit
$ git commit
$ git checkout feature1
$ git commit
$ git checkout master
$ git commit
```



# Appendix 1

***IMPORTANT***

TRY MERGE

>> git checkout master

>> git merge feature1

# Assignment

1. 자신의 실습코드 **KHUHUB** 가이드에 따라 관리
2. **SCRUM** 개념 익히기 (https://www.youtube.com/watch?v=9TycLR0TqFA)
3. 개인 프로젝트 준비하기
4. **github username** 등록하기