CSE2019: Open-source Software Development

Lab 1: Git

Software Engineering Lab

lab(se);

Git

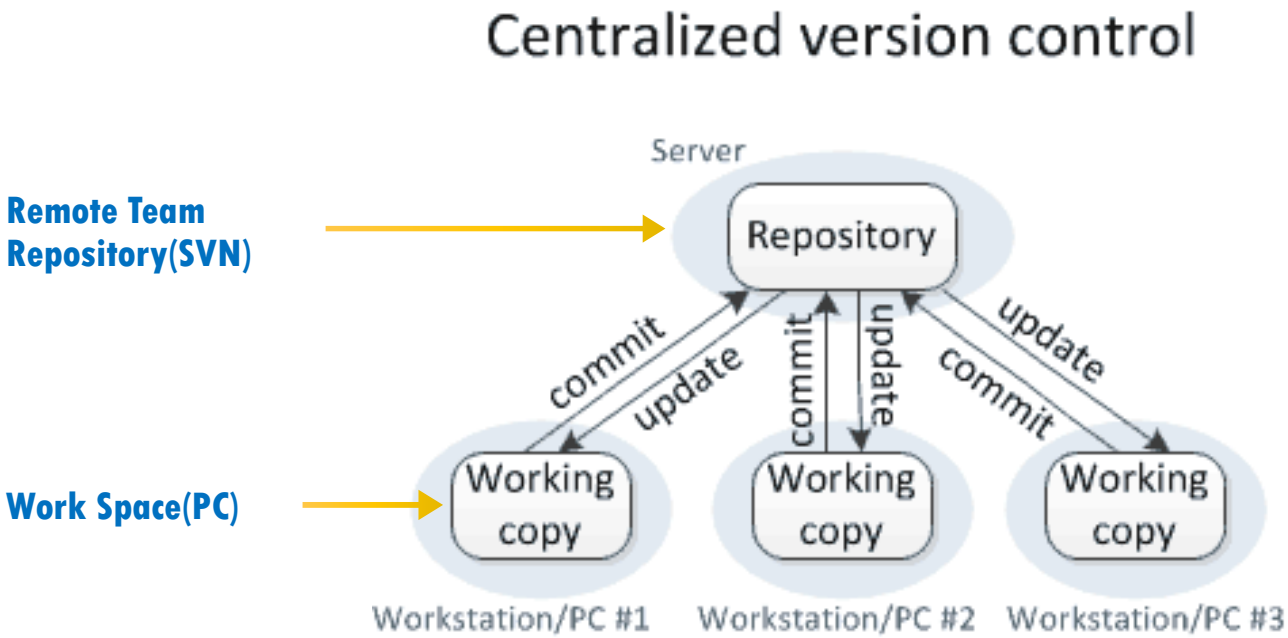
VCS(Version Control System)

Database for 'Version'

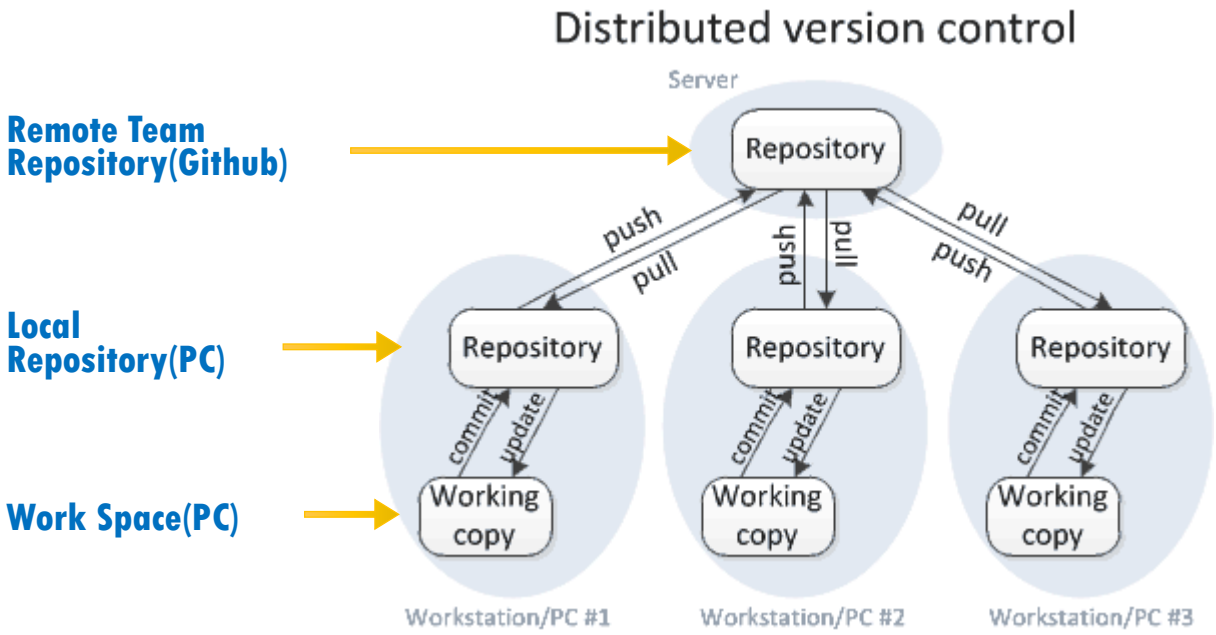
Why use VCS?

- Collaboration
- Backup
- Tracking source code
- Tracking what happened
- Reduce side-effect
  when you develop a new module

Centralized Version Control

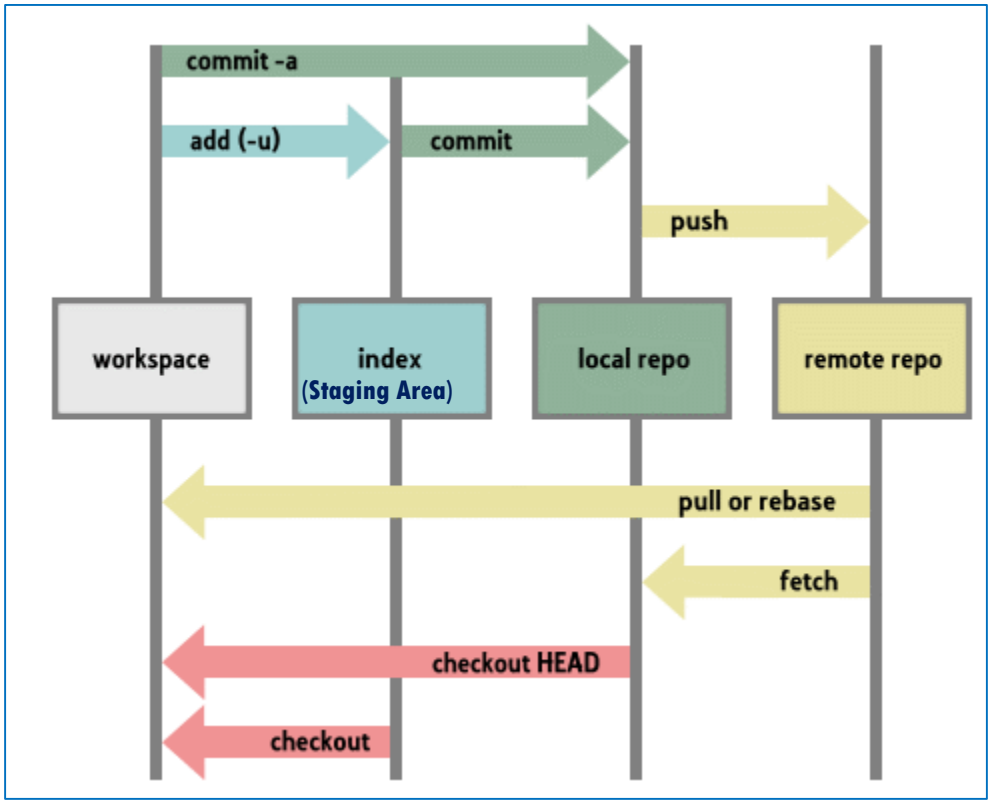


Distributed VC



lab(se);

Git Workflow



lab(se);

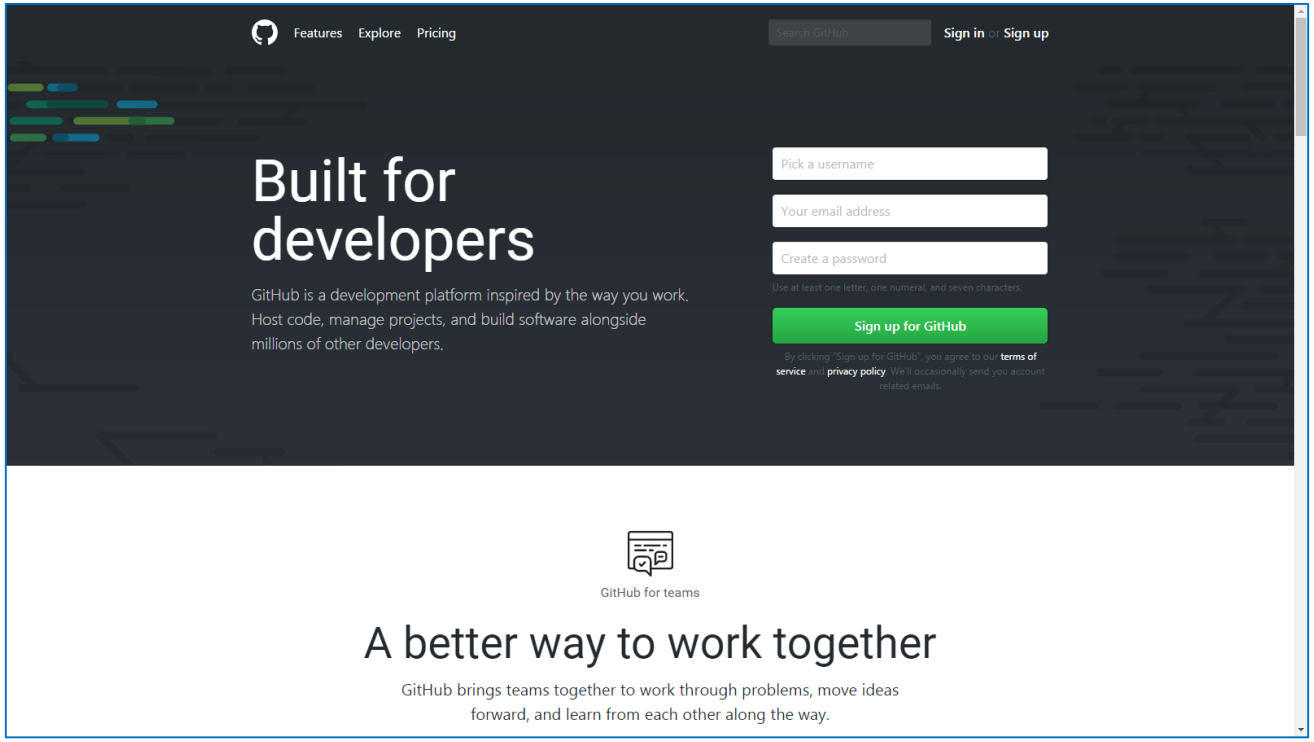
Git Install

- OS X
  - http://sourceforge.net/projects/git-osx-installer/
- Windows
  - http://msysgit.github.com/
- Linux
  - $ apt-get install git

lab(se);

Github

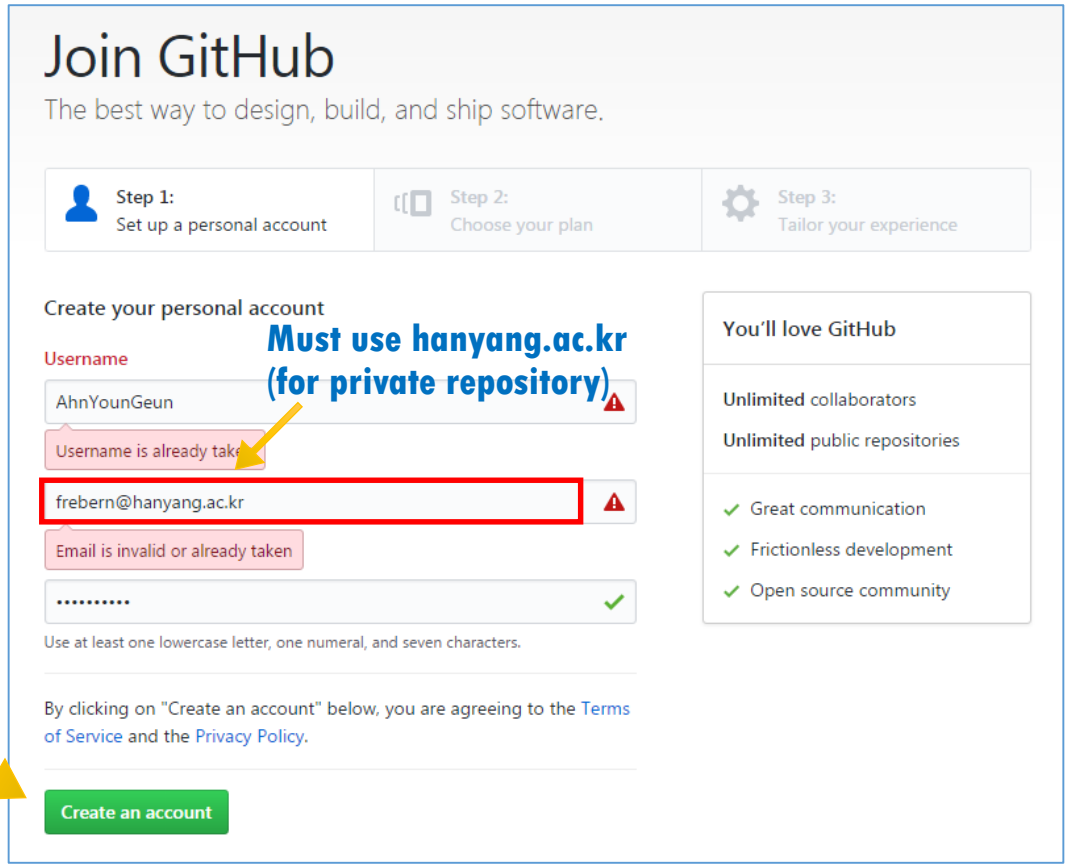
- **Sign up https://github.com**



lab(se);

Github

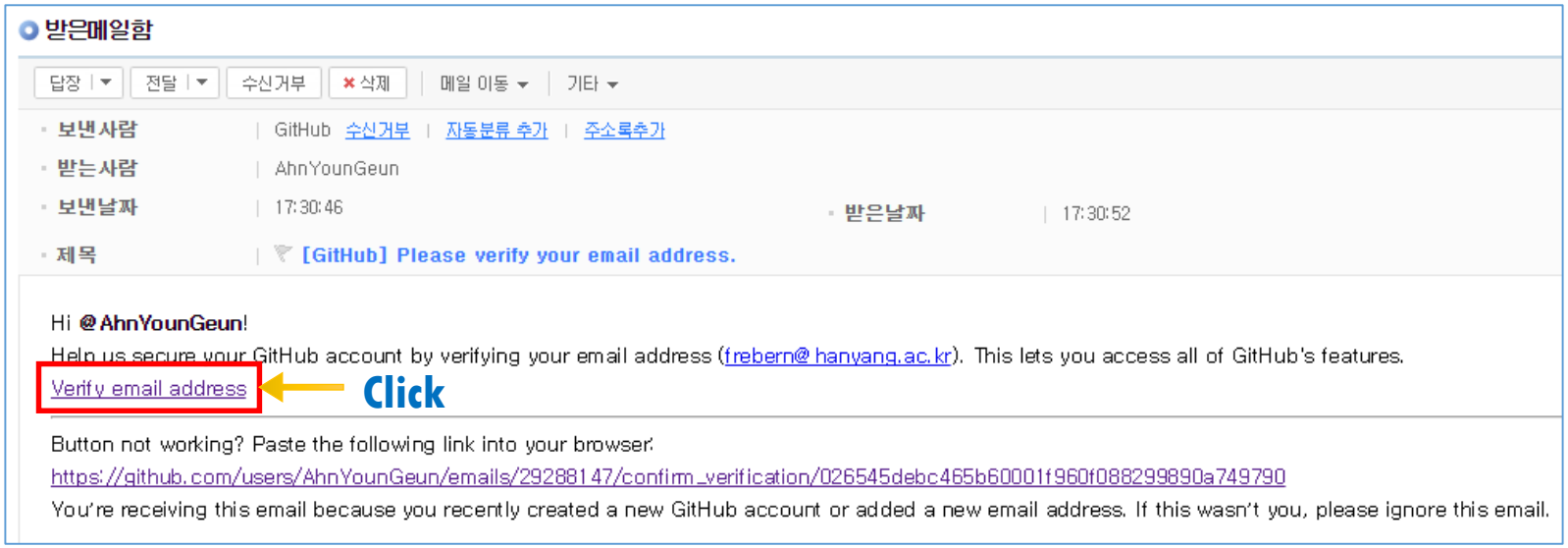
- **Sign up**



Click

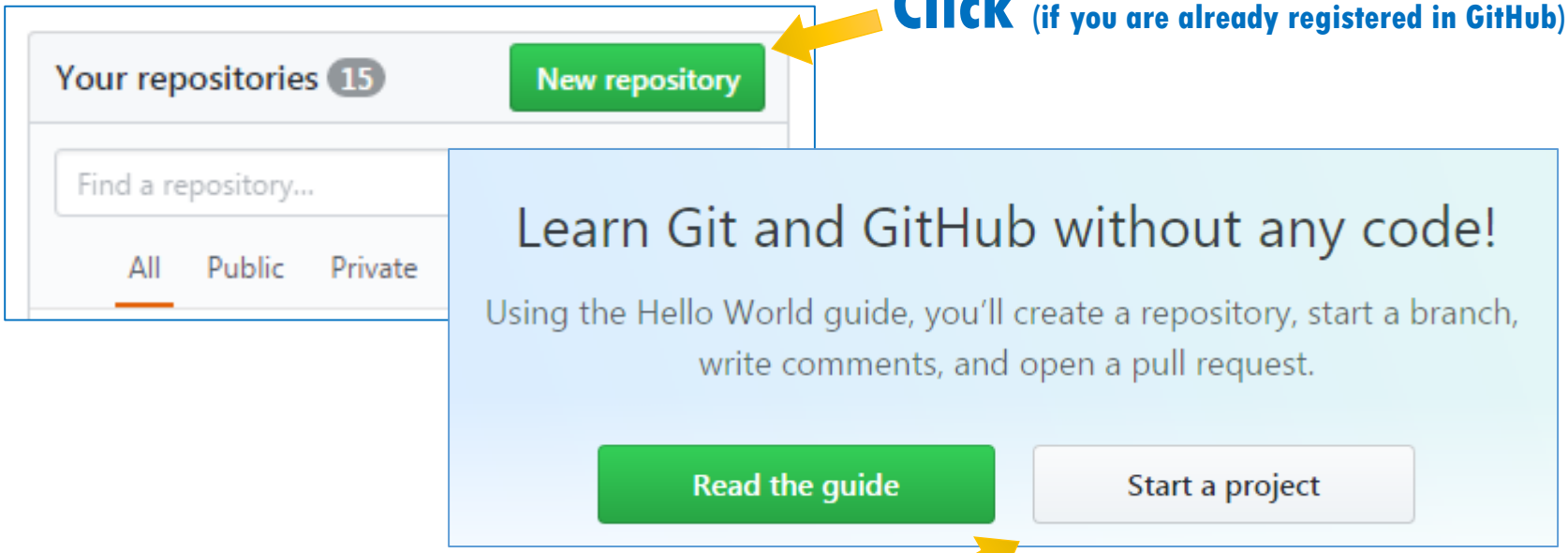
Github

- **Sign up**



Github

- **Create new remote repository**

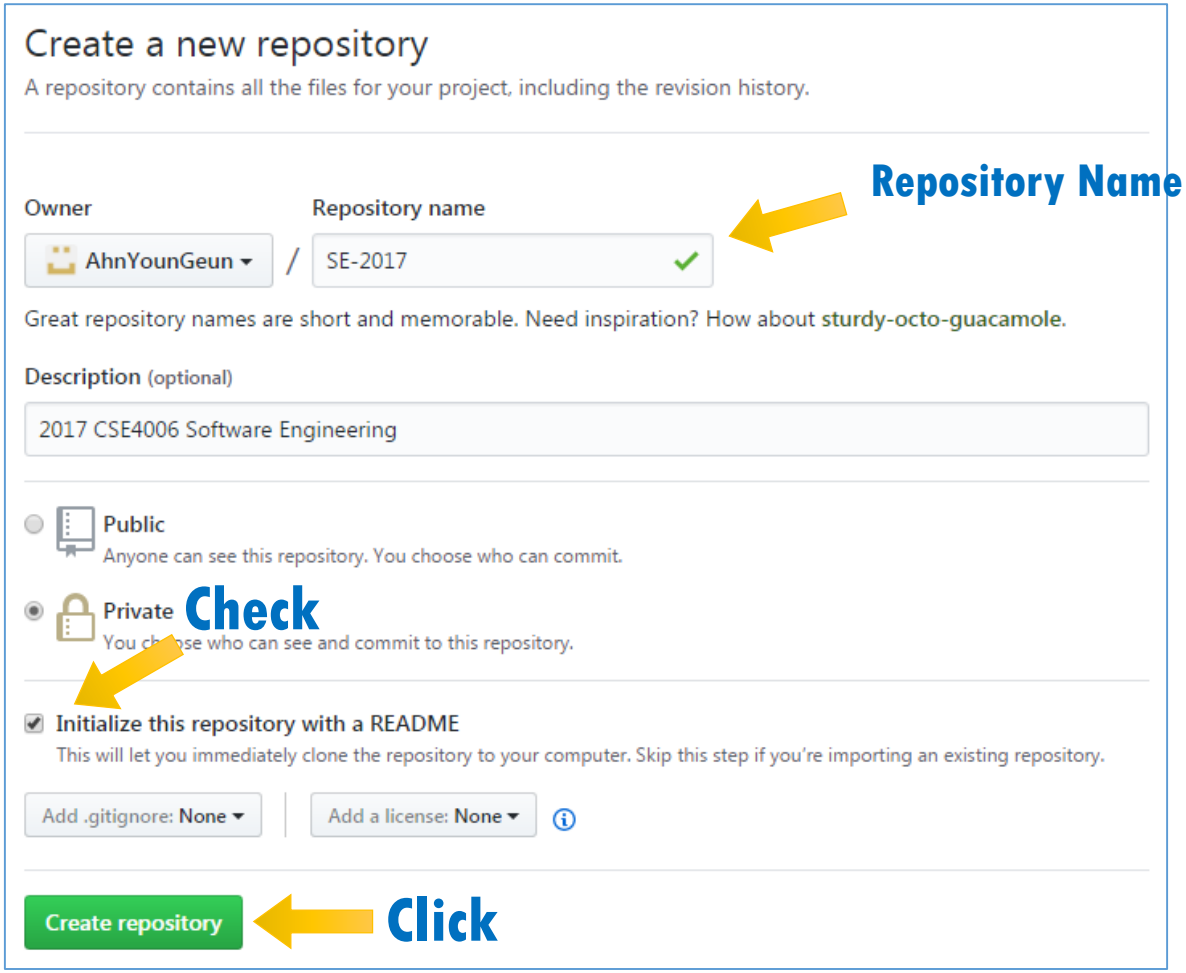


Click (if you are already registered in GitHub)

Click (if you are new GitHub user)

lab(se);

Github



lab(se);

Github

This repository Search Pull requests Issues Gist

AhnYounGeun / SE-2017 Private

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

2017 CSE4006 Software Engineering

Edit

New Add topics

Remote Repository URL

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

AhnYounGeun Initial commit

README.md Initial commit

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/AhnYounGeun/SE-2017.git

Open in Desktop Download ZIP

README.md

SE-2017

2017 CSE4006 Software Engineering

lab(se);

Git bash

MINGW64:/c/Users/user/Desktop/opensource/a

```
user@sec MINGW64 ~/Desktop/opensource
$ ls
a/  b/  c/  d/

user@sec MINGW64 ~/Desktop/opensource
$ pwd
/c/Users/user/Desktop/opensource

user@sec MINGW64 ~/Desktop/opensource
$ cd a/

user@sec MINGW64 ~/Desktop/opensource/a
$
```

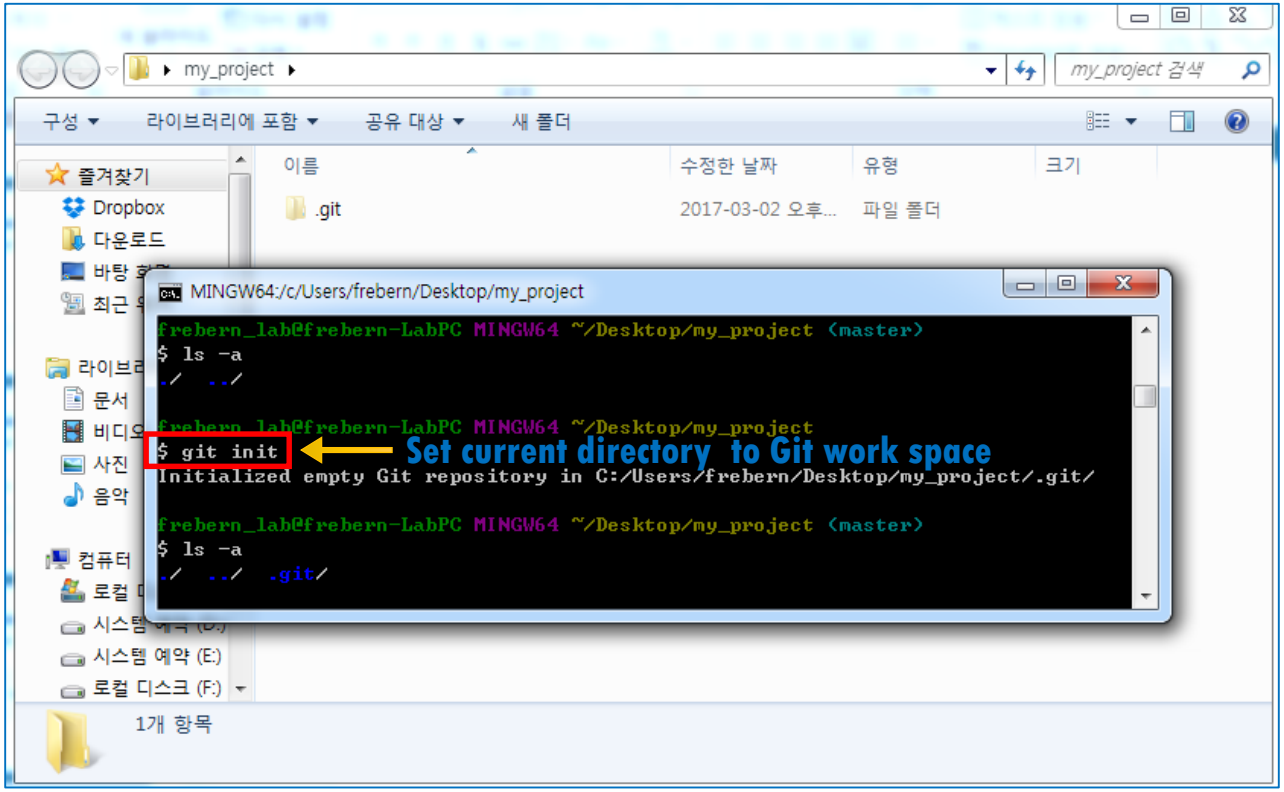
lab(se);

Linux – basic command

- ls (ls –al)
- cd (.. , ./)
- pwd
- mkdir
- rm
- cp
- mv
- history
- clear

Git - Work Space

git init



lab(se);

Git - Work Space

git init

MINGW64:/c/Users/user/Desktop/opensource

```
user@sec MINGW64 ~/Desktop/opensource
$ ls -a
./  ../  a/  b/  c/  d/

user@sec MINGW64 ~/Desktop/opensource
$ git init
Initialized empty Git repository in C:/Users/user/Desktop/opensource/.git/

user@sec MINGW64 ~/Desktop/opensource (master)
$ ls -a
./  ../  .git/  a/  b/  c/  d/

user@sec MINGW64 ~/Desktop/opensource (master)
$
```

Git - Work Space

git config

```
frebern_lab@frebern-LabPC MINGW64 ~/Desktop/my_project (master)
$ git config --global user.name "frebern"

frebern_lab@frebern-LabPC MINGW64 ~/Desktop/my_project (master)
$ git config --global user.email frebern@naver.com
```

lab(se);

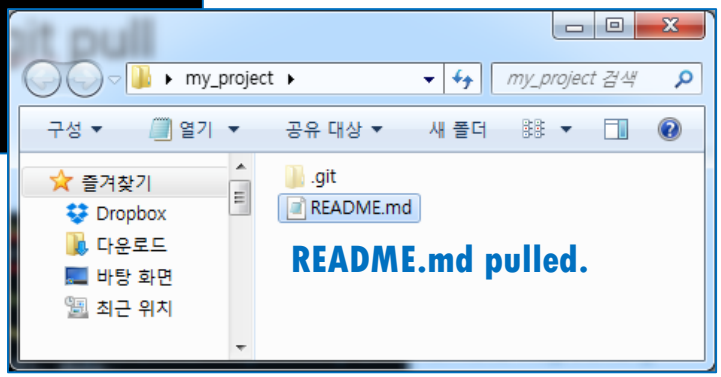
Git - Work Space

git remote add / git pull

Naming Remote Repository
(my_remote_repo ← https://github.com/frebern/SE-2017.git)

```
frebern_lab@frebern-LabPC MINGW64 ~/Desktop/my_project (master)
$ git remote add my_remote_repo https://github.com/frebern/SE-2017.git

frebern_lab@frebern-LabPC MINGW64 ~/Desktop/my_project (master)
$ git pull my_remote_repo master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/frebern/SE-2017
 * branch            master     -> FETCH_HEAD
 * [new branch]      master     -> my_remote_repo/master
```

Pull from my_remote_repo's master branch



README.md pulled.

* Pull = Fetch + Merge
(i.e. Receive current remote repository
+ Merge them to my local repository)

lab(se);

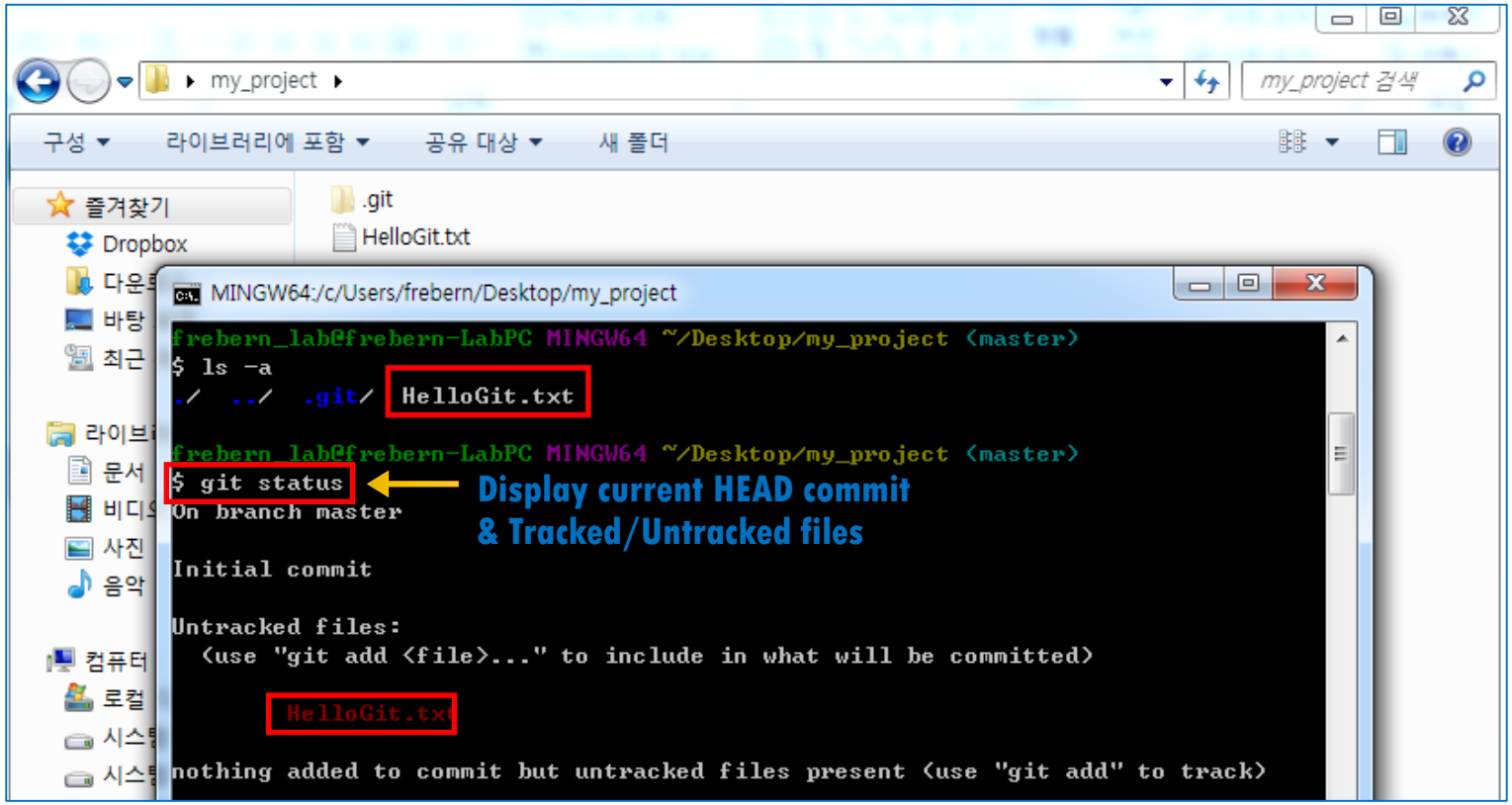
Git - Staging Area

.git

'.git' has index of all files and changes.

Git - Staging Area

git status



lab(se);

Git - Staging Area

git add

```
frebern_lab@frebern-LabPC MINGW64 ~/Desktop/my_project (master)
$ git add HelloGit.txt

frebern_lab@frebern-LabPC MINGW64 ~/Desktop/my_project (master)
$ git status
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   HelloGit.txt
```

Add file to Staging Area(index)

lab(se);

Git - Local Repository

git commit

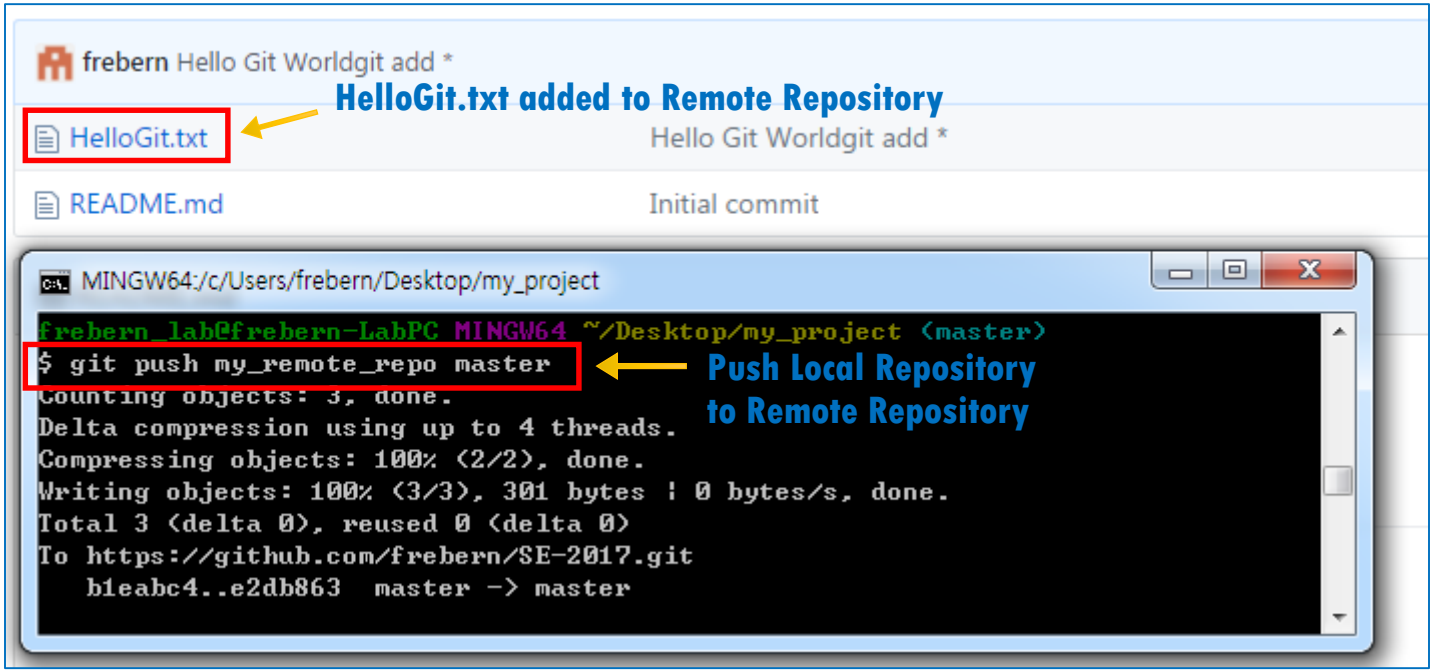
```
frebern_lab@frebern-LabPC MINGW64 ~/Desktop/my_project (master)
$ git commit -m "Git Start!"
[master e2e856b] Git Start!
 1 file changed, 1 insertion(+), 1 deletion(-)

frebern_lab@frebern-LabPC MINGW64 ~/Desktop/my_project (master)
$ git status
On branch master
nothing to commit, working tree clean
```

Commit all added changes
& save to Local Repository

lab(se);

Git - Remote Repository

git push



Basic Work Flow

1. git init
2. git remote add [repoName] [url]
3. git pull [repoName] [branch(e.g. master)]
4. Working…
5. git add *
6. git commit –m “commit message”
7. git push [repoName] [branch(e.g. master)]
8. Repeat 3~7

lab(se);

Exercise

1. Create an remote repository
2. Pull remote repository to your workspace.
   (if you can't pull them, try --allow-unrelated-histories option)
   It's double dash - -
3. Create a program
   if you input "1" then print your name. (any programming language is ok)
4. Add & Commit to local repository
5. Push them to remote repository
   (if you can't push them, try -f option)
6. Add function
   which print your introduce when you input "2"
7. Update your remote repository
8. Send direct message to T.A with screenshot(github , 'history' command)

lab(se);