

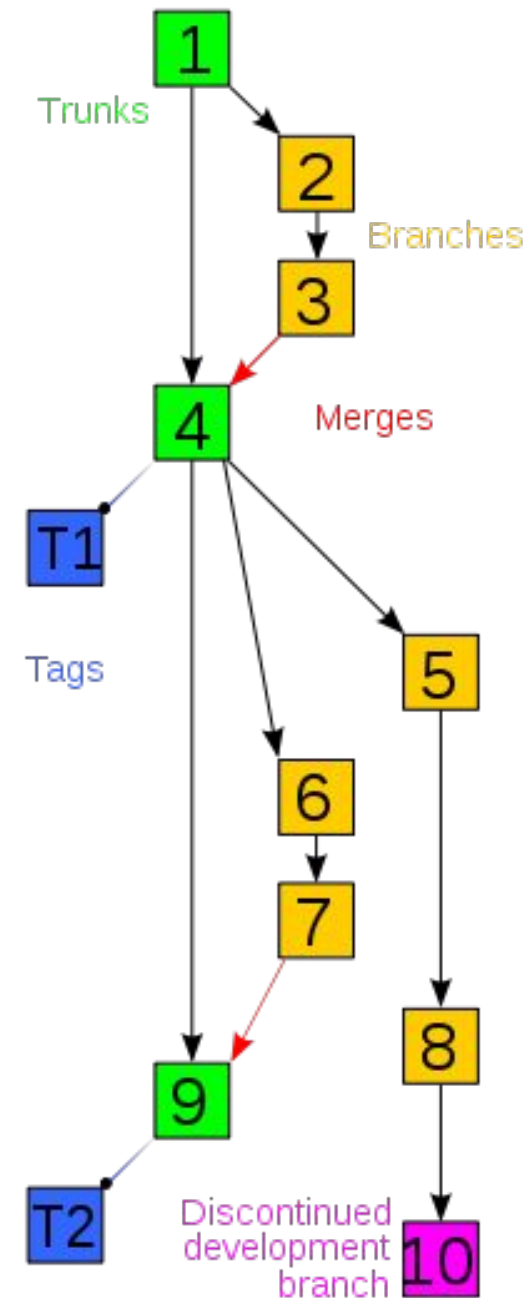
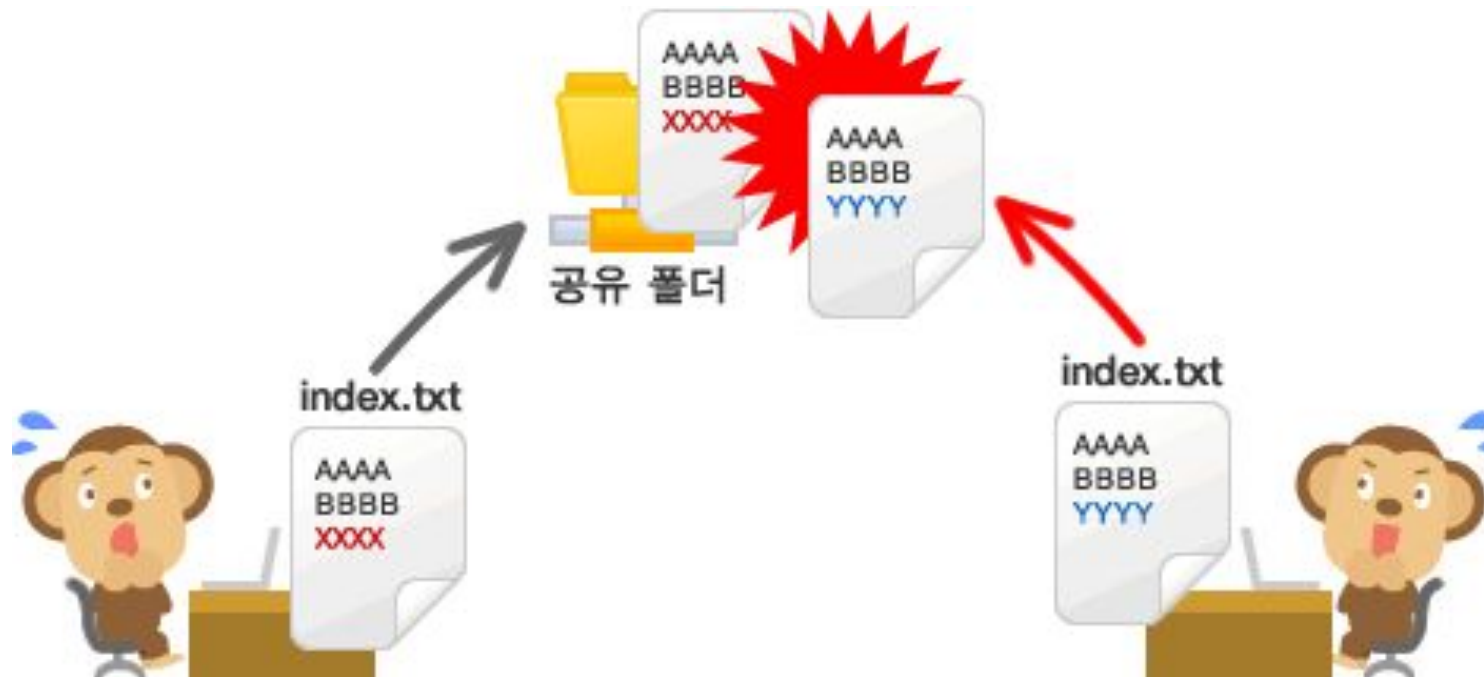
Git

*

저녁이 있는 프로젝트
오상훈
6 Hours, 1 Month

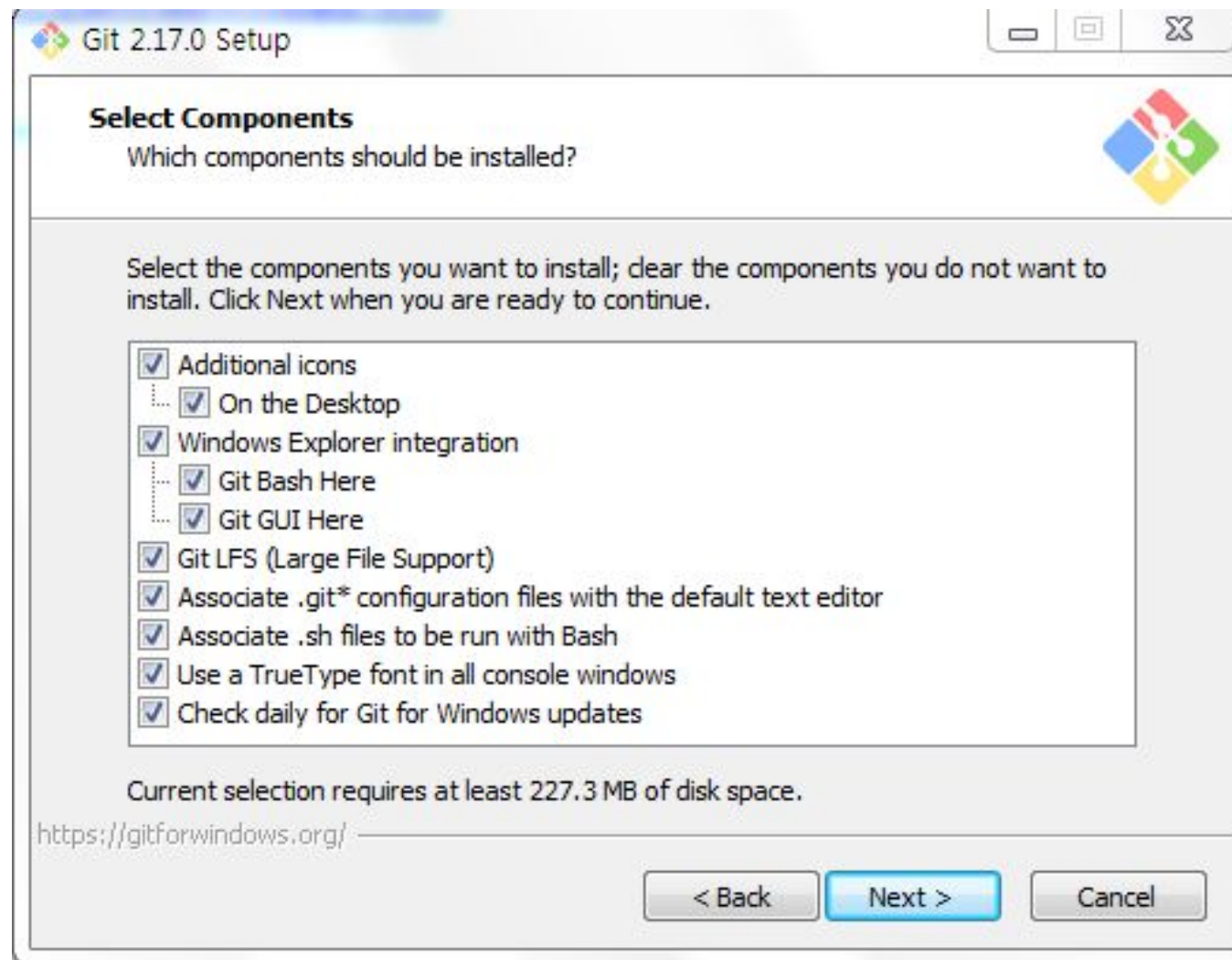
Why VCS(biz30.timedoctor.com/git-mecurial-and-cvs-comparison-of-svn-software/)

- ❖ VCS(Version Control System)
 - 도둑(?) 잡기 시스템 : 안전한 문서 관리 필요
 - 버전 관리(Software Version Management)와 형상 관리(Software Configuration Management)
 - 대표적 종류
 - Apache Subversion (SVN), Git, Mercurial
- ❖ 필요성
 - 2명 이상 동시 작업 시 필요.



Usage Git

- ❖ 가입 : Username, Email, Password 기억 필요.
 - <https://github.com/>
- ❖ Install Git on Local
 - <https://git-scm.com/download/win> : 아래 화면 외엔 'Next' or 'Yes' 클릭



Git - clone

~\$ **sudo apt-get install git gitk** → **c:\Program Files\Git>git-cmd**

~\$ **git --version**

~\$ **cd ?** → move worked folder

~\$ **git clone https://github.com/SanghunOh/lectureHtml.git**

~\$ **git log --graph --oneline**

~\$ **rm -rf *** → 탐색기로 선택 삭제

~\$ **git log --graph --oneline**

~\$ **rm -rf .*** → 탐색기로 상위 폴더 삭제

~\$ **git log --graph --oneline**

❖ 알아보기

➤ 이미 프로젝트 중인 Download(From Github) 이해

➤ 각각 저장소 폴더 위치 확인

■ Remote (repository) : web

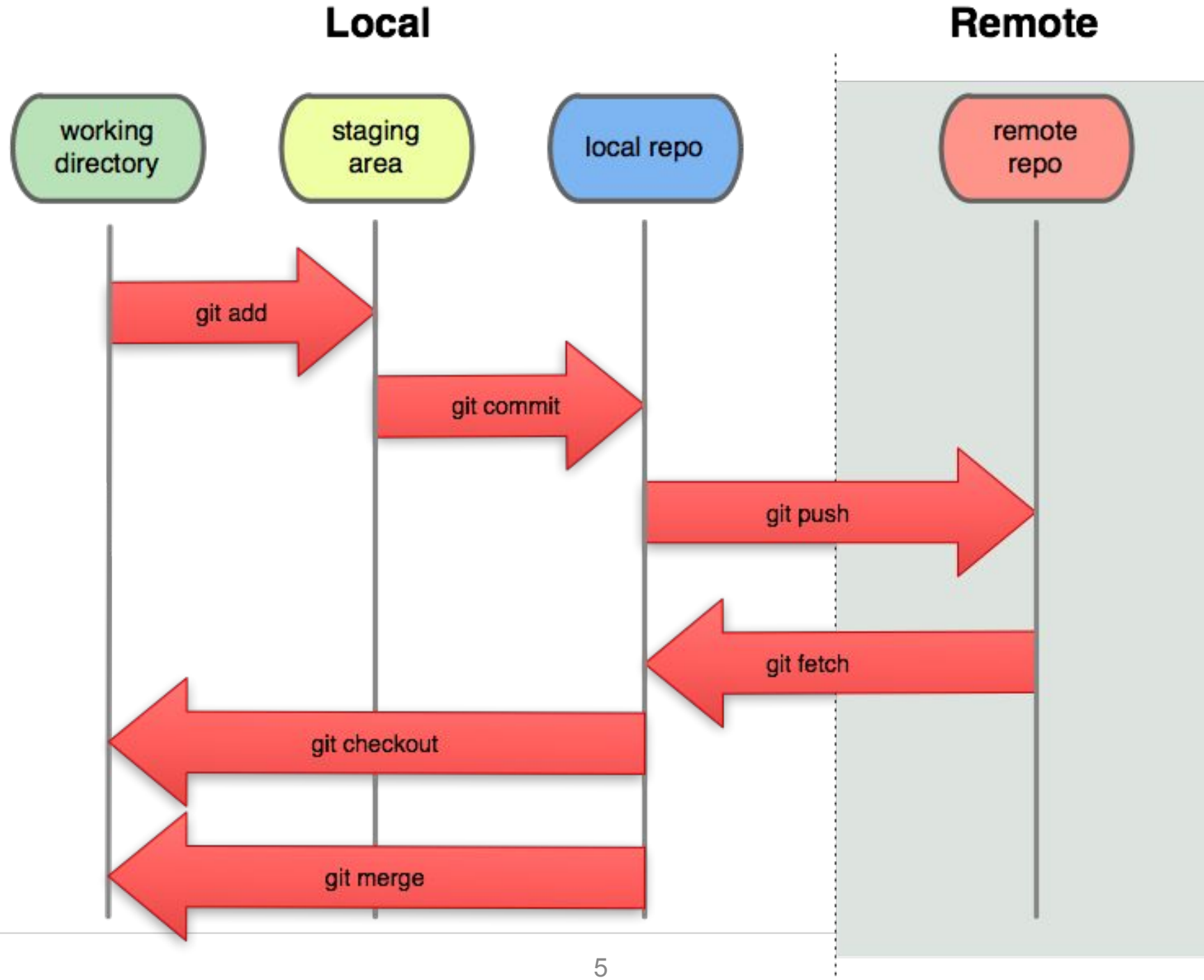
■ Local(stage) :/.git → OS 마다 다름.

■ Working Folder : ../.git

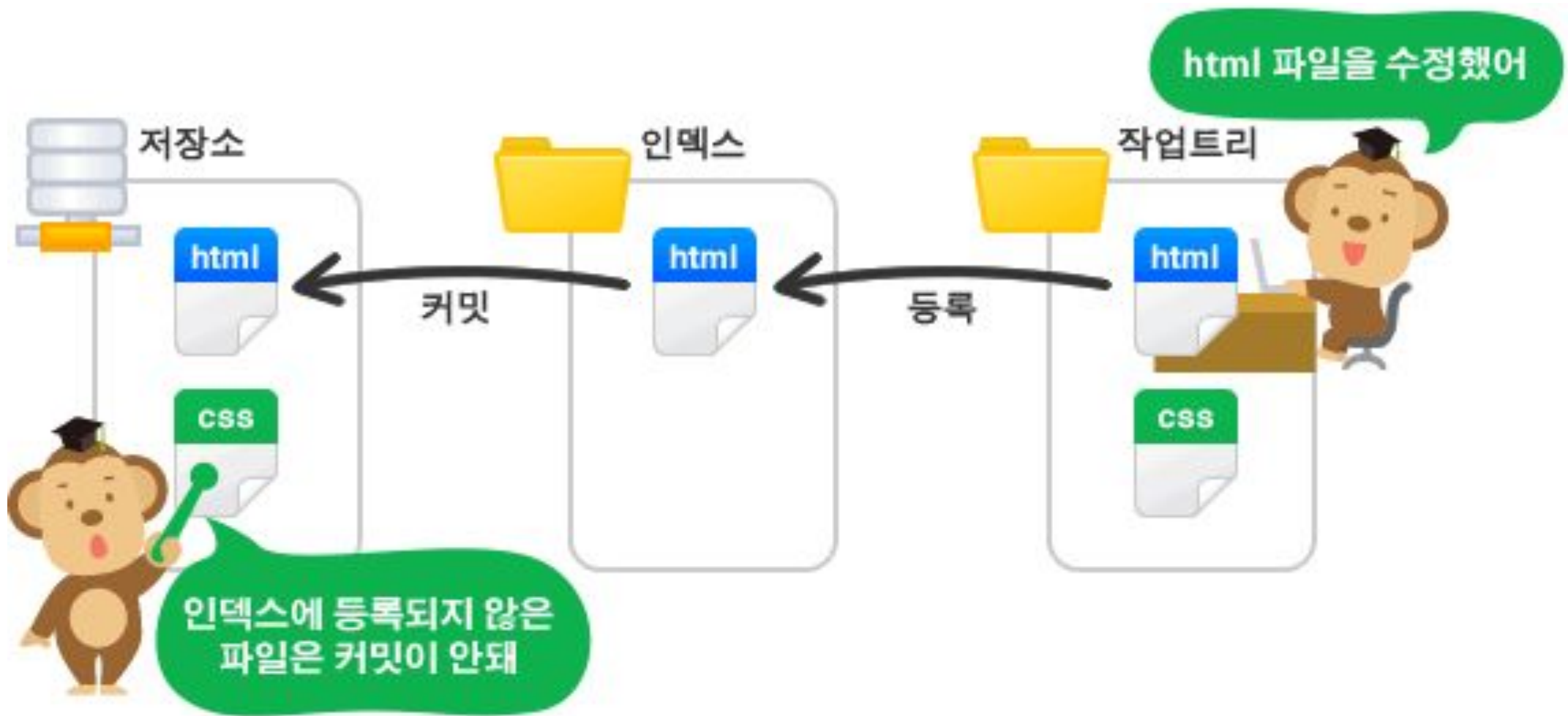
❖ 해보기

➤ git clone과 Download From Github 각각 방식 실행.
ex) https://github.com/SanghunOh/?

Git 저장소(repository) 방식



stage : 인덱스 구간.



Git - Local Commit & Stash

```
~$ mkdir eclipse-workspace_html
~$ cd eclipse-workspace_html
~$ git init                                → git 초기화
~$ git status                             → 상태 확인
~$ git config --global user.email "otter.oh@gmail.com" → Remote Account
~$ git config --global user.name "SanghunOh"
~$ git add .                              → 변경사항 index 공간 등록
~$ git commit -m "first commit"           → Apply Local Worktree
~$ git log                                → 등록 내용
~$ touch second.txt                       → echo . > second.txt
~$ git add .
~$ git commit -m "second commit"          → Apply Local Worktree
~$ touch my_stash.txt                     → echo . > my_stash.txt
~$ git add .
~$ git stash save "temp message"          → 임시 저장
~$ git stash list
~$ ls                                     → dir
~$ git stash pop
~$ git stash drop stash@{0}
~$ ls                                     → dir
```

Try - Git(Local Commit & Stash)

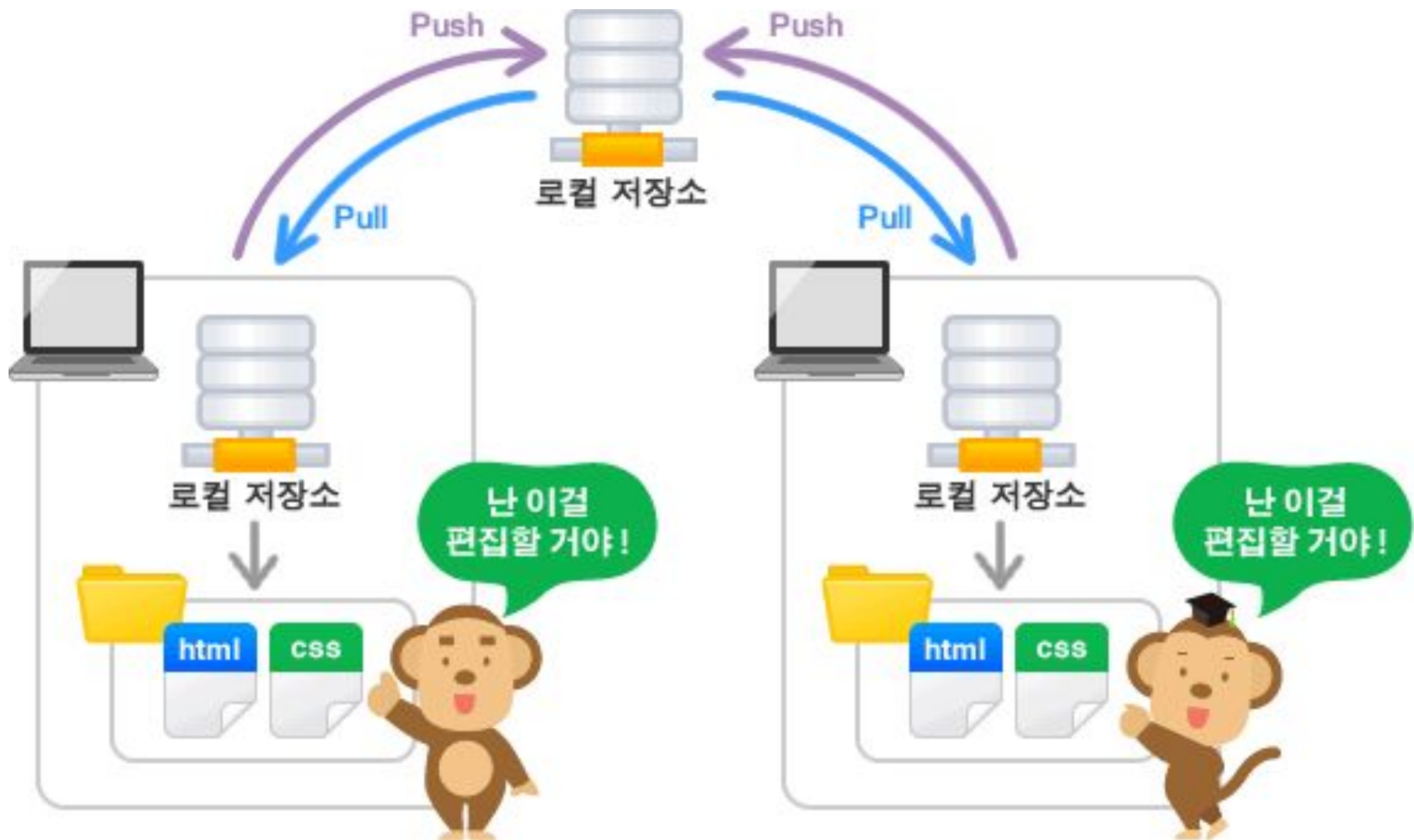
❖ 참조 명령어

- git init
- git add .
- git commit -m
- git stash save
- git stash list

❖ 해보기

- 각각 별도 진행
 - first_01.html 저장, first_02.html 임시저장, first_03.html 작성, first_04.html 저장
 - second_01.html 저장, second_02.html 임시저장, second_03.html 작성, second_04.html 저장, second_02.html 임시저장 꺼내어 저장
- 진행 결과 공유

Git - Remote



GitHub - Create Git Server

- ❖ <https://github.com/> → Login
- ❖ Click 'New Repository' -> Repository name : gitTest > Click 'Create repository'
 - Settings -> Click 'Delete this repository' -> Click 'I understand ...'

GitHub, Inc. [US] | <https://github.com/new>

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner

SanghunOh ▾

Repository name

gitRepo ✓

Great repository names are short and memorable. Need inspiration? How about [bug-free-octo-meme](#).

Description (optional)

Test Git Repository

☒ **Public**
Anyone can see this repository. You choose who can commit.

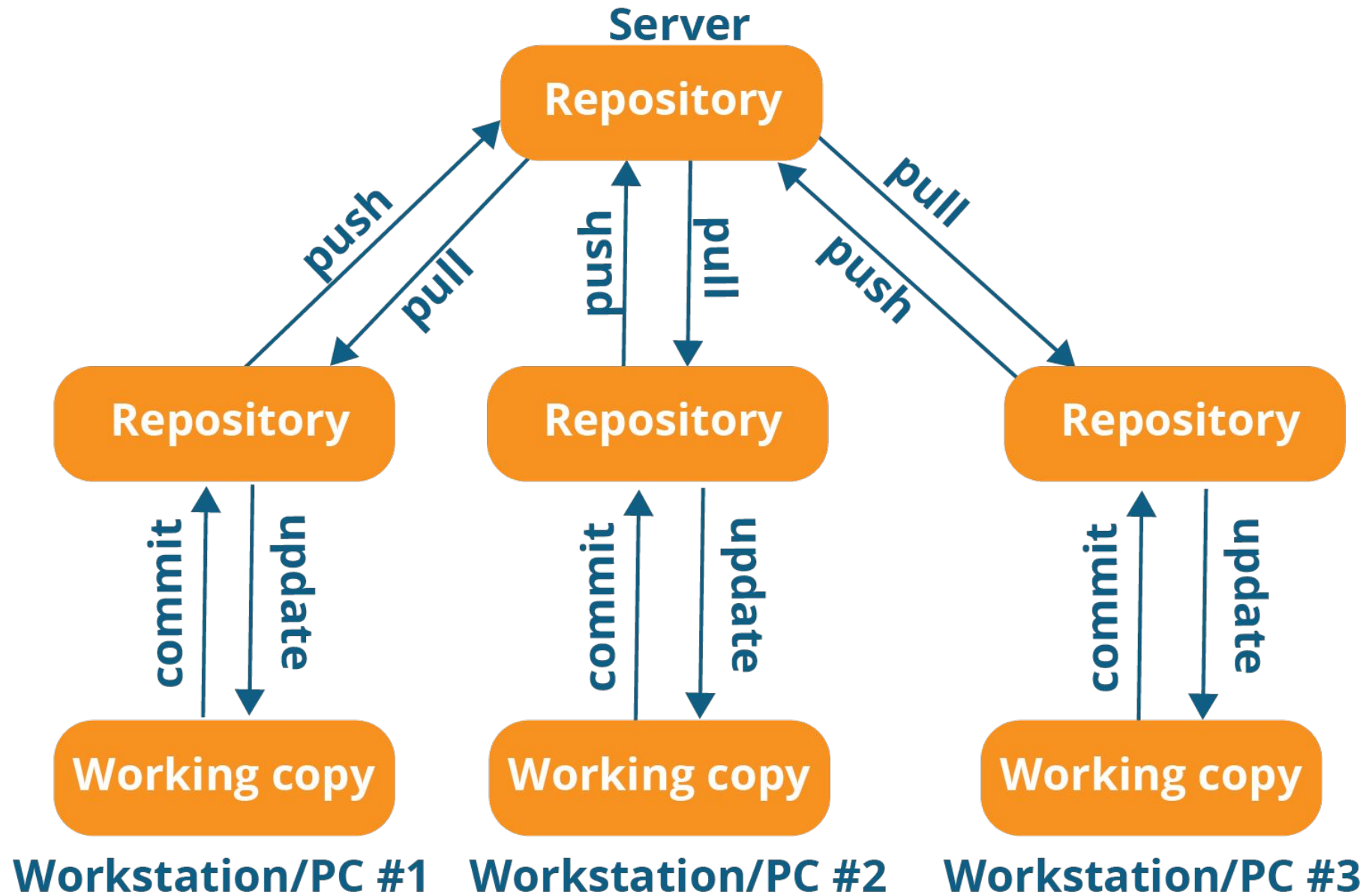
☐ **Private**
You choose who can see and commit to this repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: **None** ▾ | Add a license: **None** ▾ ⓘ

Create repository

Distributed version control system



Git - Remote, push/pull Source

```
~$ mkdir temp
~$ cd temp
~$ touch temp.txt
~$ git init
~$ git add .
~$ git commit -m "first commit"
~$ git config --global user.email "otter.oh@gmail.com"
~$ git config --global user.name "SanghunOh"
```

→ echo . > temp.txt
→ git 초기화
→ 변경사항 **index** 공간 등록
→ Apply Local Worktree
→ Remote Account

❖ Copy URI in TextBox on Github
ex) <https://github.com/SanghunOh/gitTest.git>

```
~$ git remote add original https://github.com/SanghunOh/gitTest.git
~$ git push -u original master
```

→ Error 시 동일 Stage 삭제 필요

❖ <https://github.com/SanghunOh/gitTest> 변경 확인

```
~$ touch sample.txt
~$ git add sample.txt
~$ git commit -m "add sample.txt"
~$ git push
```

→ echo . > sample.txt

❖ Modify sample.txt on Github

```
~$ git pull
```

Try - Git(Remote, push/pull Source)

- ❖ 참조 명령어
 - git remote add original
 - git push
 - git pull
- ❖ 해보기
 - Git server 만들고, 파일 올리기
 - Git server 명 : gitTemp
 - temp.html, sample.html To Git server
 - Git server 파일 temp.html 수정
 - pull temp.html
 - 진행 결과 공유

Git - Conflict

❖ 대상 Github : <https://github.com/SanghunOh/gitTest.git>

❖ Modify first line in sample.txt on Github

~\$ vi sample.txt → Notepad 사용 첫 라인 수정

~\$ git add sample.txt

~\$ git commit -m "modify first line in sample.txt"

~\$ git push

... hint: (e.g., 'git pull ...') before pushing again.

~\$ git pull original master

Conflict on Message

Auto-merging sample.txt

CONFLICT (content): Merge conflict in sample.txt

Automatic merge failed; fix conflicts and then commit the result.

~\$ vi sample.txt

→ Notepad 사용 불필요한 부분 삭제

~\$ git add sample.txt

~\$ git commit -m "clean up mass in sample.txt"

~\$ git pull

Already up to date.

~\$ git push

~\$ git log --graph --oneline

Try - Git(Conflict)

- ❖ 참조 명령어
 - git push
 - git pull
- ❖ 해보기
 - Git server 파일 편집 충돌 연습하기
 - push index.html
 - Modify first line in index.html on Github
 - vi index.html → or Notepad 사용 첫 라인 수정
 - 수정 부분 정상 push
 - 진행 결과 공유

GitHub - Collaborator

- ❖ <https://github.com/> → Login
- ❖ Click 'New Repository'
- ❖ Setting -> Collaborators
 - Typing ID Name > Add Collaborators
 - 대상자 Email -> Click 'View invitation' > Login on GitHub > Click 'Accept'
 - 대상자 Github 엔 표시 안됨, 'Star★' 기능 활성화로 추적

Code Pull requests 0 Projects 0 Wiki Insights Settings

Options
Collaborators
Branches
Webhooks
Integrations & services
Deploy keys

Collaborators Push access to the repository

Awaiting SanghunOh's response Copy invite link Cancel invite

Search by username, full name or email address
You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

Sanghun Add collaborator

sanghun
sanghun-han sanghun han

Git - Collaborator

~\$ git clone <https://github.com/otter35/ServletJSPExampleForElementCourse.git>

❖ 대상자 Github 활용

~\$ touch collaborator.txt

→ echo . > collaborator.txt

~\$ git add .

~\$ git commit -m 'collaborator files'

~\$ git push

❖ 대상자 Github 상태 확인

❖ Setting -> Collaborators > 대상자 삭제

~\$ touch wrong.txt

→ echo . > wrong.txt

~\$ git add .

~\$ git commit -m 'wrong files'

~\$ git push

remote: Permission to ...

...: The requested URL returned error: 403

Try - Github(Collaborator)

❖ 해보기

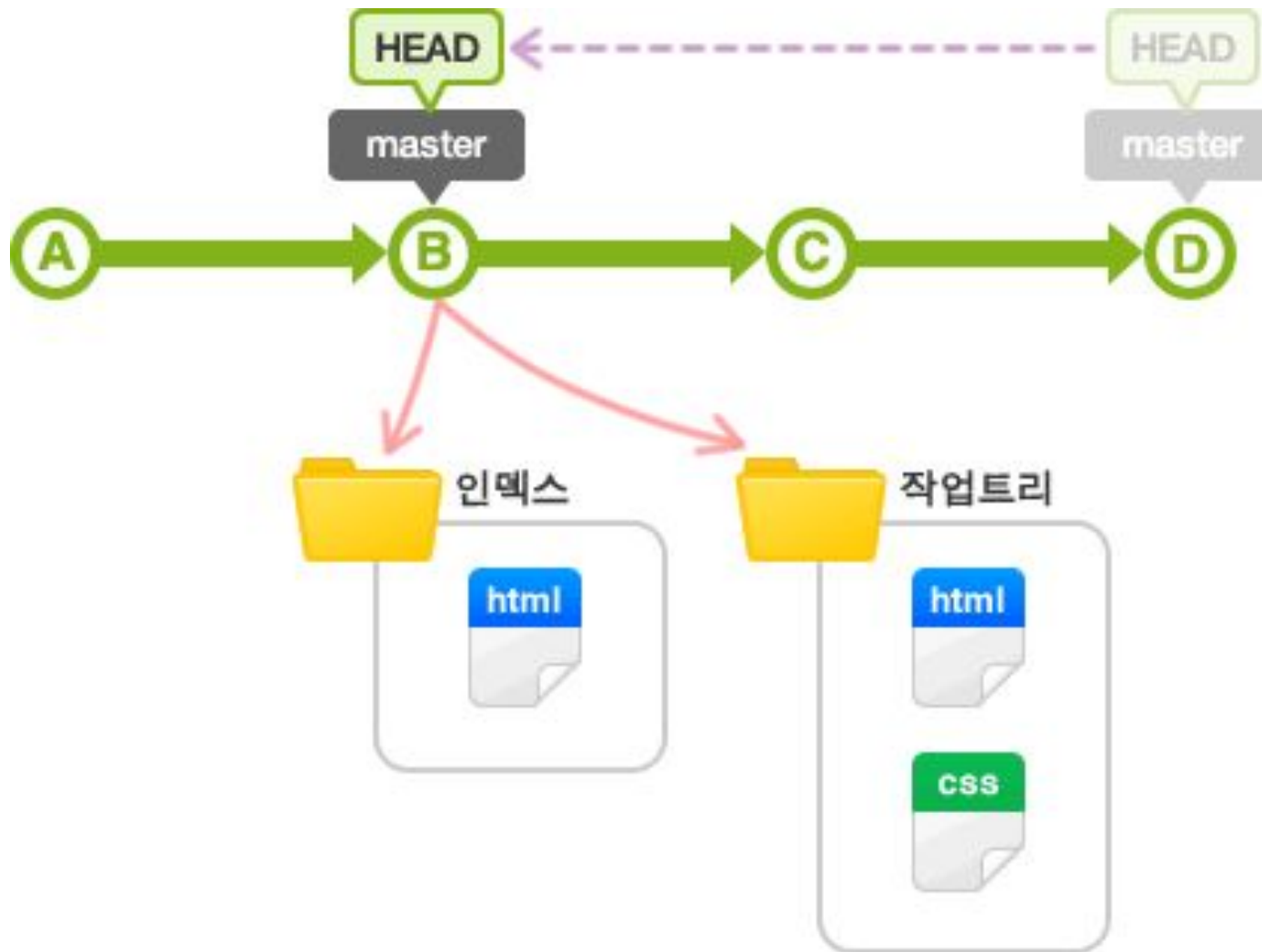
➤ 협력해 개발하기

- Github에 **Collaborator** 대상자 등록
- `git clone` 이용 상대 프로젝트 **Download**
- `push collaborator.html`
- `collaborator.html` 협력자 모두 같은 라인 편집 후 충돌 해결.(각각 마지막 해결자 롤로 하기)

➤ 진행 결과 공유

Git - Revert

- ❖ **revert** : 특정 **commit** 편집 삭제, 하나의 완전한 편집 이력 추가.
- ❖ **reset** : 이전 commits 완전 Rollback
 - 대상 : soft-commit, mixed-index, hard-worktree)
- ❖ **cherry-pick** : 다른 브랜치에서 지정한 커밋을 현재 브랜치로 복사



Git - Revert(simple version)

❖ 대상 Github : <https://github.com/SanghunOh/gitTest.git>

~\$ git push -u origin master

~\$ touch first.txt

→ echo . > first.txt

~\$ git add .

~\$ git commit -m "first.txt commit"

~\$ **git revert HEAD**

→ type ': q' in 편집기

~\$ ls

→ dir

~\$ git log --graph

~\$ touch second.txt third.txt

→ echo . > second.txt

~\$ git add .

~\$ git commit -m "files commit"

~\$ **git reset --hard HEAD~**

ex) 특정 HEAD명 가능

git reset --hard HEAD~~

git reset --hard ORIG_HEAD

~\$ ls

→ dir

~\$ git log --graph

Try - Git(Revert)

- ❖ 참조 명령어
 - git revert HEAD
 - git reset --hard HEAD~
- ❖ 해보기
 - Local Stage Rollback
 - commit 파일 first.html
 - commit 파일 second.html
 - revert HEAD
 - commit 파일 third.html
 - reset HEAD
 - 진행 결과 공유

Git - ignore

❖ 대상 Github : <https://github.com/SanghunOh/gitTest.git>

~\$ touch login.txt

→ echo . > login.txt

~\$ git add .

~\$ git commit -m "login.txt"

~\$ **git rm --cached** login.txt

→ commit 파일 제외 시

~\$ git commit -m "untrack login.txt"

~\$ vi .gitignore

→ Notepad 편집

login.txt # 파일 추가

→ commit 이전 사전 제외 시

subfolder/ # 폴더 추가

→ subfolder/ 하위폴더 모두 시

...

ex) **/subfolder/

~\$ git push

❖ Github 상태 확인

~\$ git commit -am 'ignore login.txt'

→ '-a' : add 명령 옵션

~\$ mkdir subfolder

~\$ cd subfolder

~\$ touch enroll.txt

→ echo . > enroll.txt

~\$ git add .

~\$ git commit -m "login.txt"

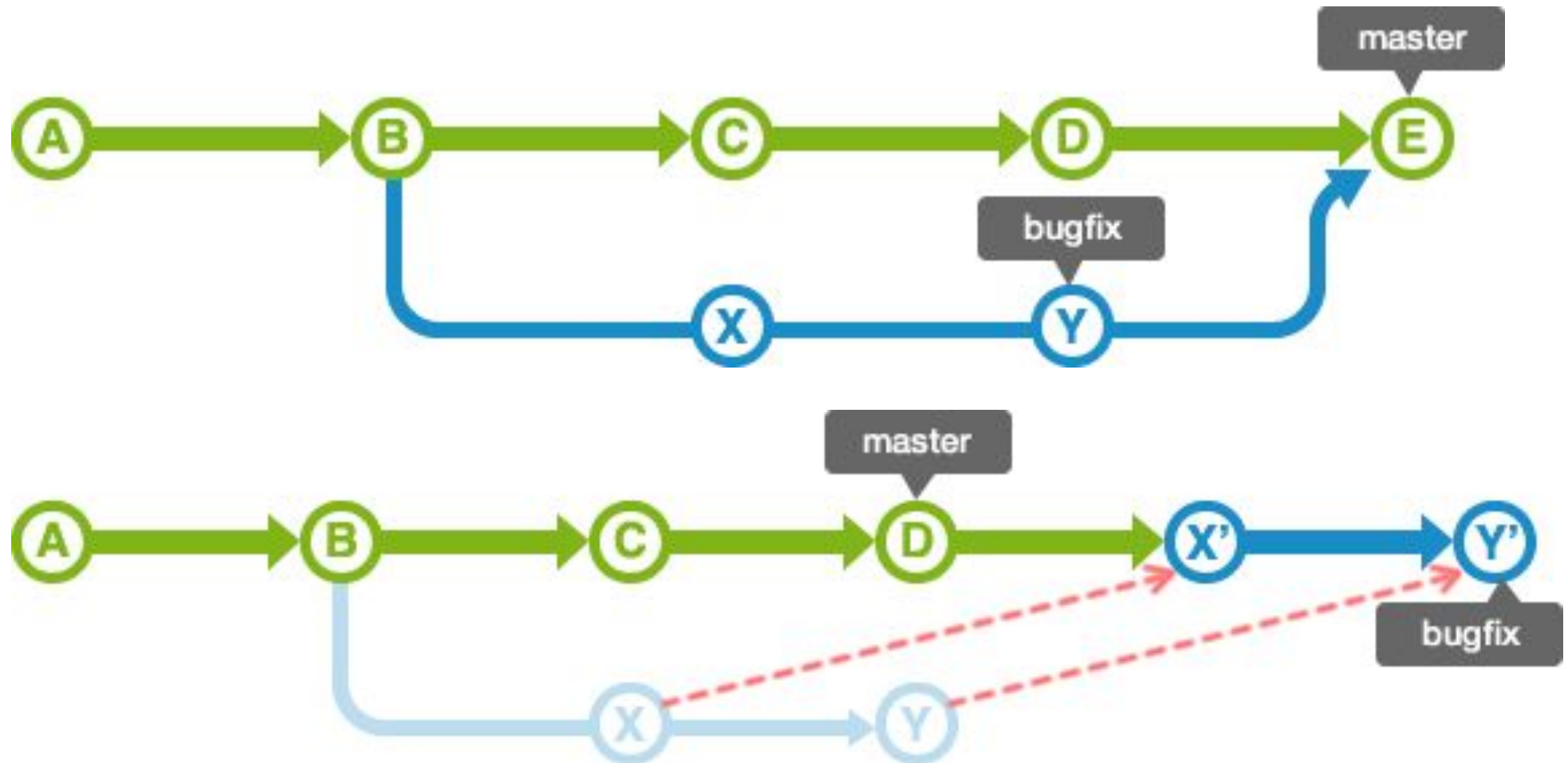
... nothing to commit, working tree clean

Try - Git(ignore)

- ❖ 참조 명령어
 - `git rm --cached`
 - `.gitignore` file
- ❖ 해보기
 - 올리지 않는 파일/폴더 작성하기
 - `index.html`, `login.html`, `temp.html`
 - `push *`
 - 이후 `temp.html` 파일만 `push` 없음.
 - `member` 폴더, `member/enroll.html`, 각각 위치 맞게 생성
 - `push *`
 - 이후 `member` 폴더 전체 `push` 없음.
 - 진행 결과 공유

Git - Branch(1)

- ❖ Main branch : master(배포 가능본), develop(개발본)
- ❖ Feature branch : 새로운 기능 개발 및 버그 수정본.
- ❖ Release branch : 정상동작 확인 후 master와 병합, prefix 'release-'
- ❖ Hotfix branch : 배포 후 긴급 수정본, prefix 'hotfix-'



Git - Branch(2)

```
~$ git branch feature
~$ git branch
~$ git checkout feature          → or git checkout -b <branch>
~$ touch feature_file.txt       → echo . > feature_file.txt
~$ git add .
~$ git commit -m 'add files'
~$ git remote add feature https://github.com/SanghunOh/gitTest.git
~$ git push --set-upstream feature feature    → 이후 git push 사용
~$ ls
~$ git checkout master
~$ git merge feature              → delete : ~$ git branch -d feature
~$ ls
~$ git add .
~$ git commit -m 'merge anther branch'
~$ git push
~$ git reset --hard HEAD~
~$ git tag -a ver0.1 -m 'version 0.1'
~$ git tag -n                    → delete : ~$ git tag -d version01
```

Try - Git(Branch)

- ❖ 참조 명령어
 -
- ❖ 해보기
 - Branch 생성과 이용하기
 - branch develop-version 생성
 - push feature_file.html
 - master branch 기준 merge
 - push master branch
 - 진행 결과 공유

GitHub - Drop Repository

- ❖ <https://github.com/> → Login
- ❖ Click 'Repository'
- ❖ Settings and Scroll Down

Danger Zone

Make this repository private

Please [upgrade your plan](#) to make this repository private.

Transfer ownership

Transfer this repository to another user or to an organization where you have the ability to create repositories.

Transfer

Archive this repository

Mark this repository as archived and read-only.

Archive this repository

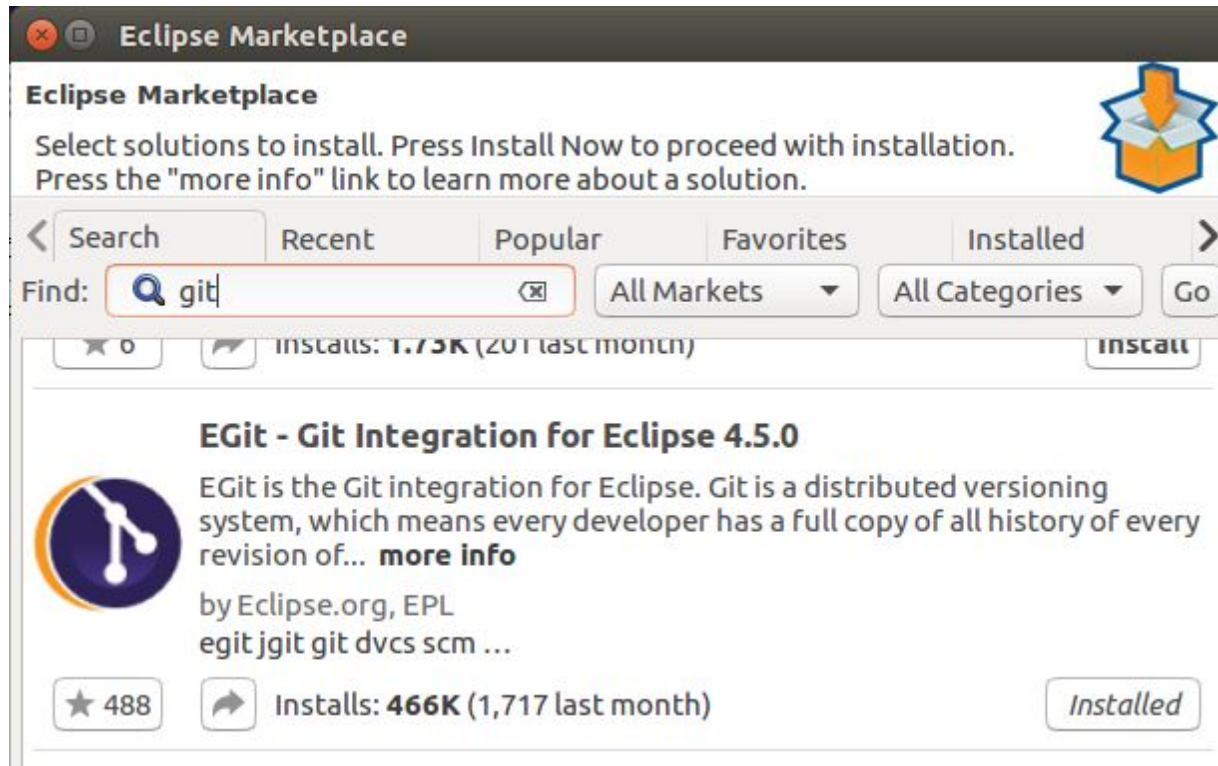
Delete this repository

Once you delete a repository, there is no going back. Please be certain.

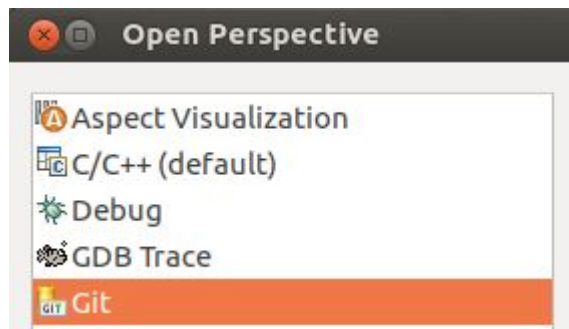
Delete this repository

Eclipse - plugin Egit

❖ PlugIn EGit

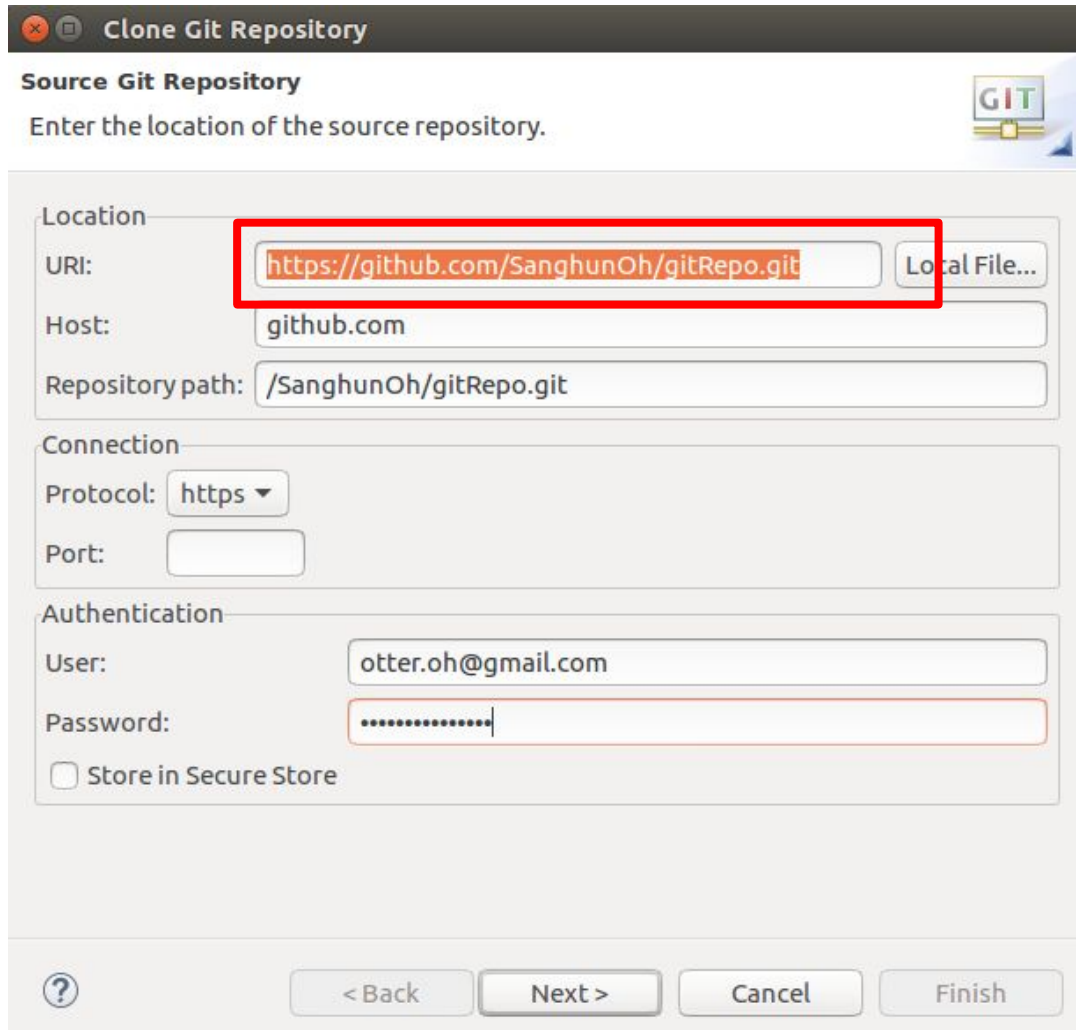


❖ Open Git From Perspective



Eclipse - Clone Project(1)

- ❖ Window > Perspective > Open Perspective > Other
- ❖ Open Perspective -> Click 'Git'



Clone Git Repository

Source Git Repository

Enter the location of the source repository.

Location

URI: Local File...

Host:

Repository path:

Connection

Protocol:

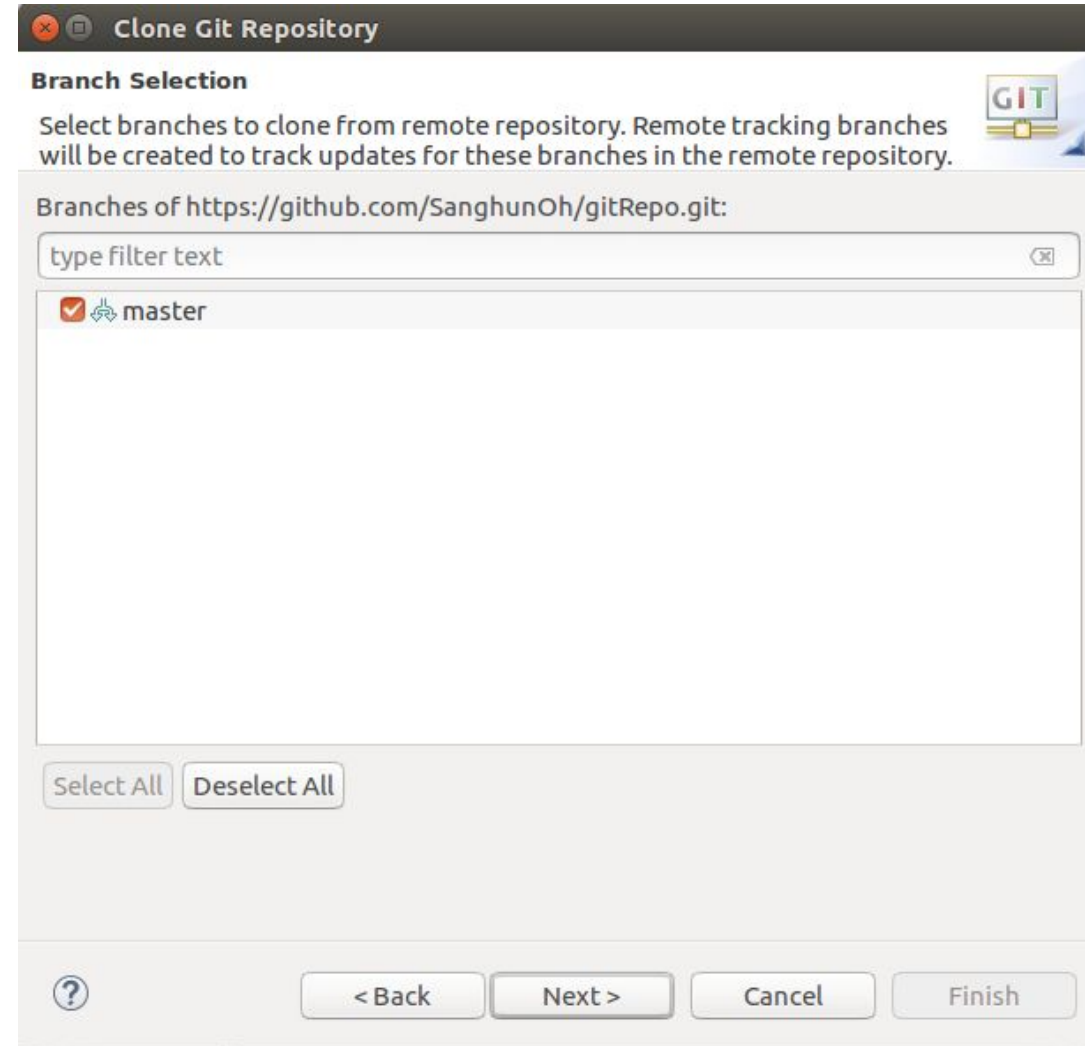
Port:

Authentication

User:

Password:

☐ Store in Secure Store



Clone Git Repository

Branch Selection

Select branches to clone from remote repository. Remote tracking branches will be created to track updates for these branches in the remote repository.

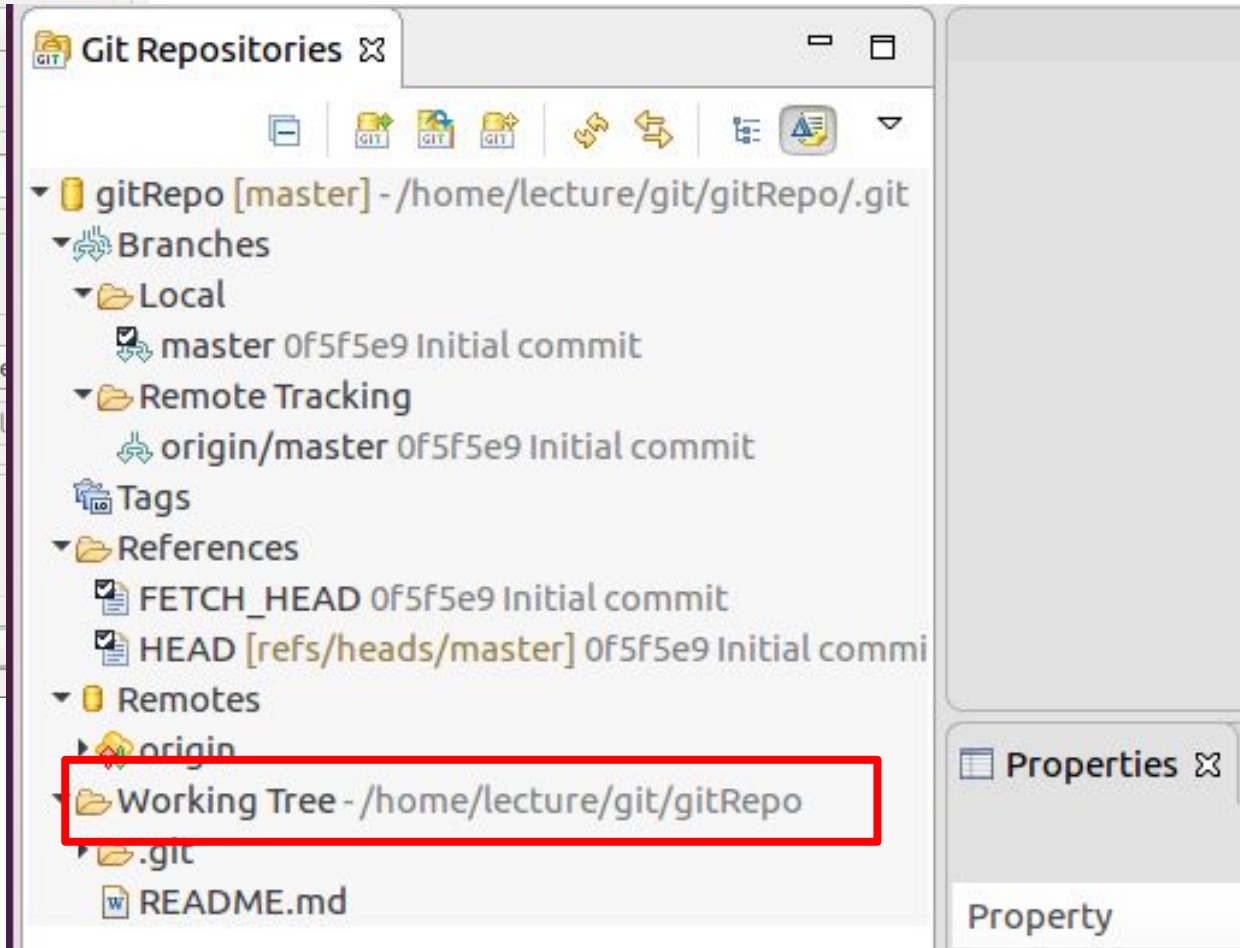
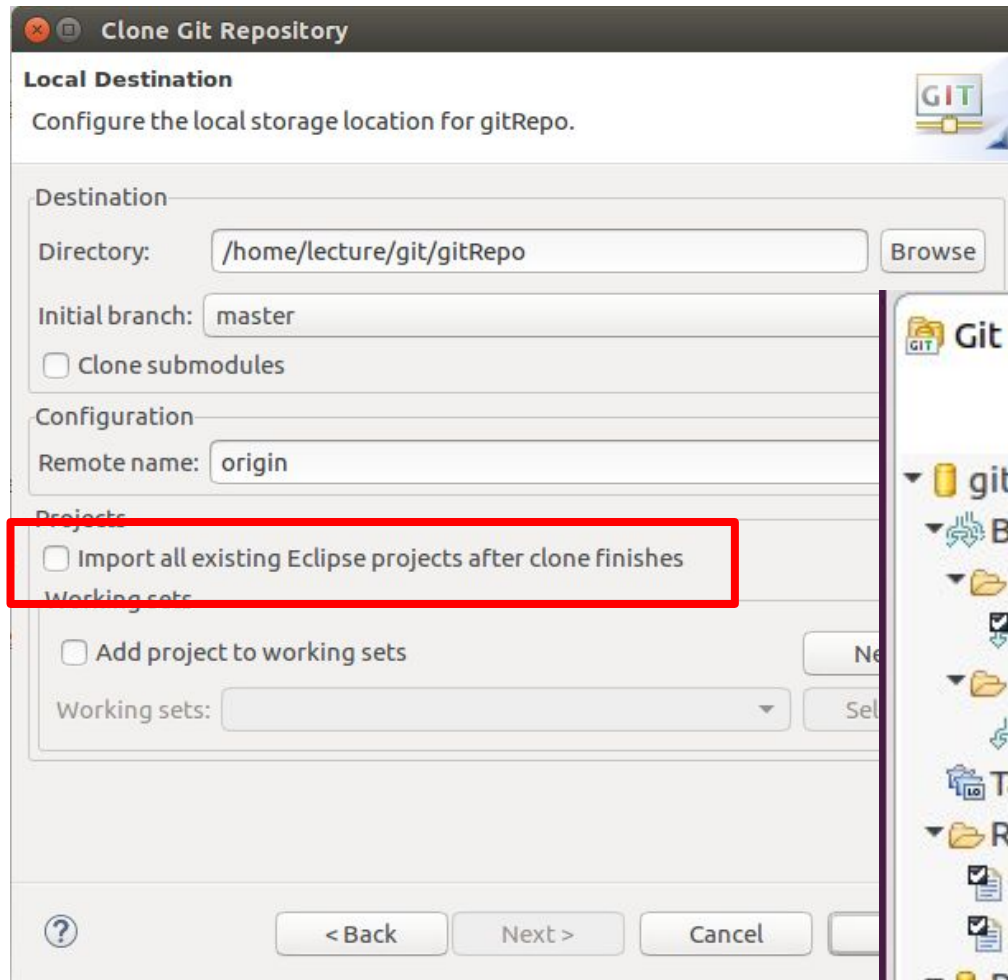
Branches of https://github.com/SanghunOh/gitRepo.git:

type filter text

☒ master

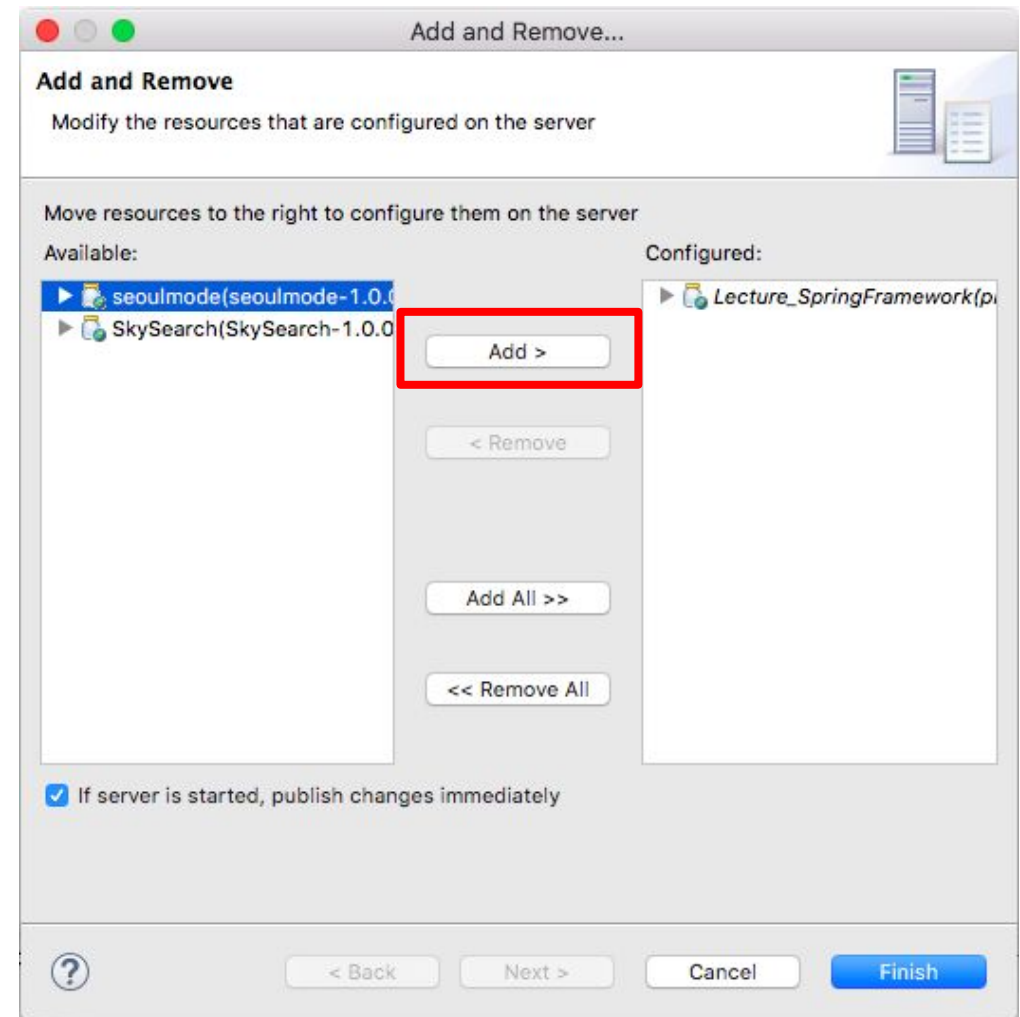
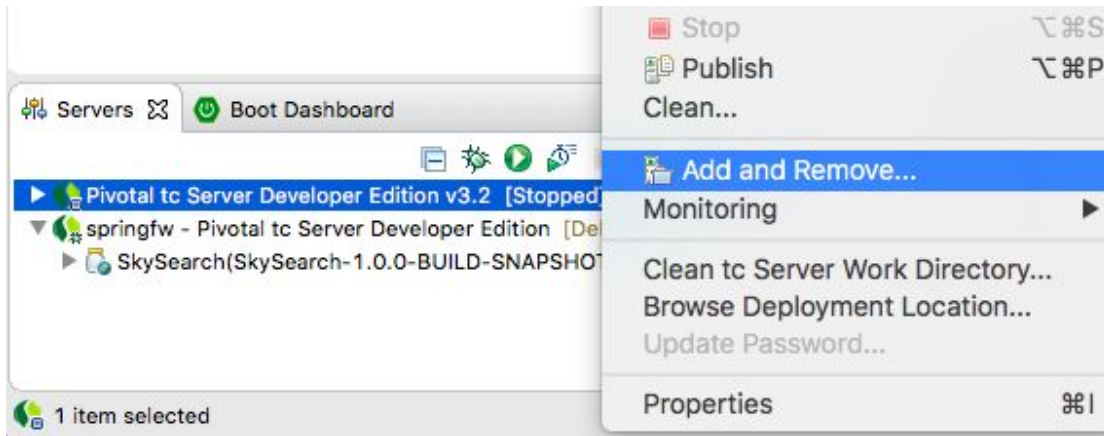
Eclipse - Clone Project(2)

❖ Check 'import all existing ...'



Eclipse - Deploy with Server

- ❖ Server Perspective
- ❖ AS Deploy problem
 - Shortcut Menu > Maven > Update Project



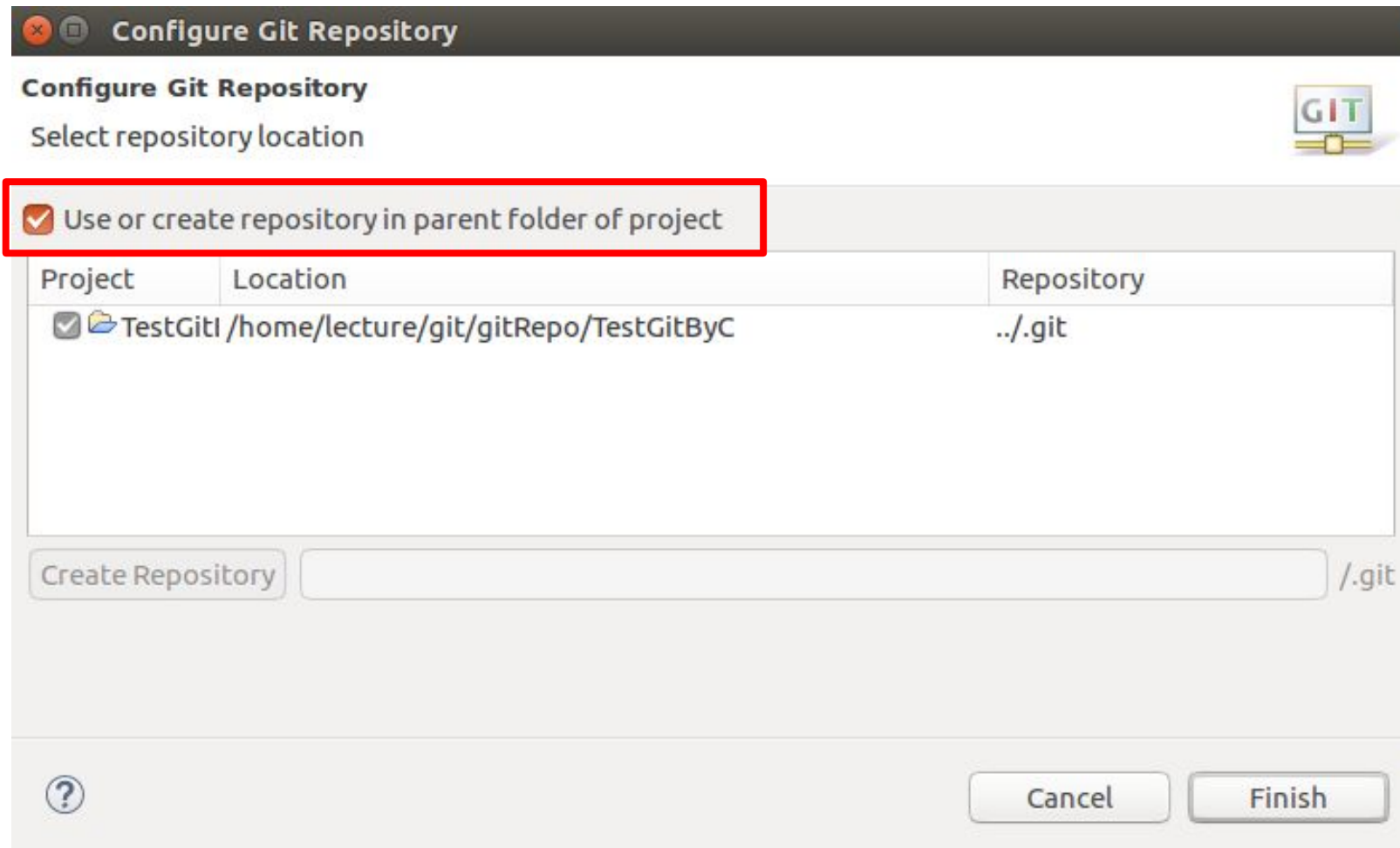
Eclipse - initial Project(1)

❖ Create Project



Eclipse - initial Project(2)

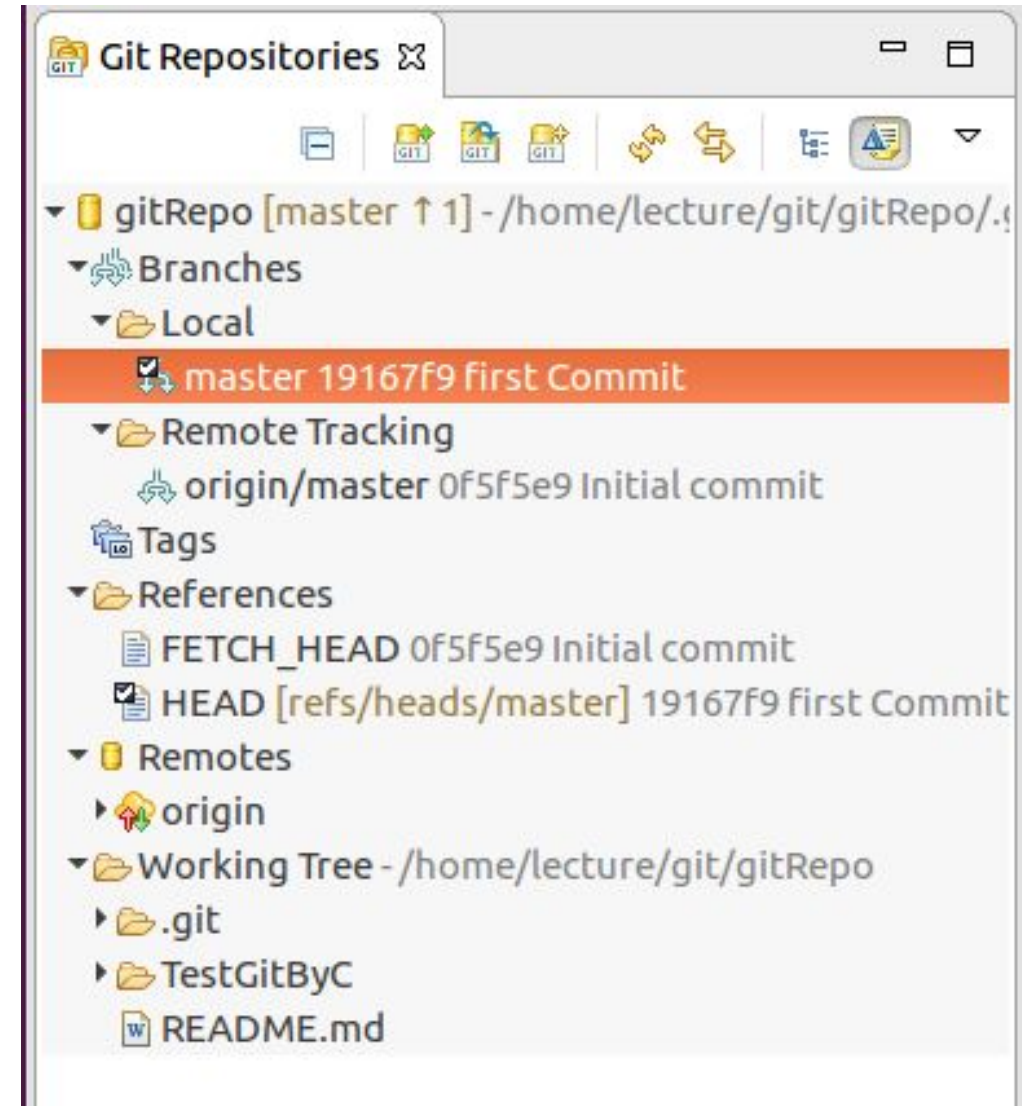
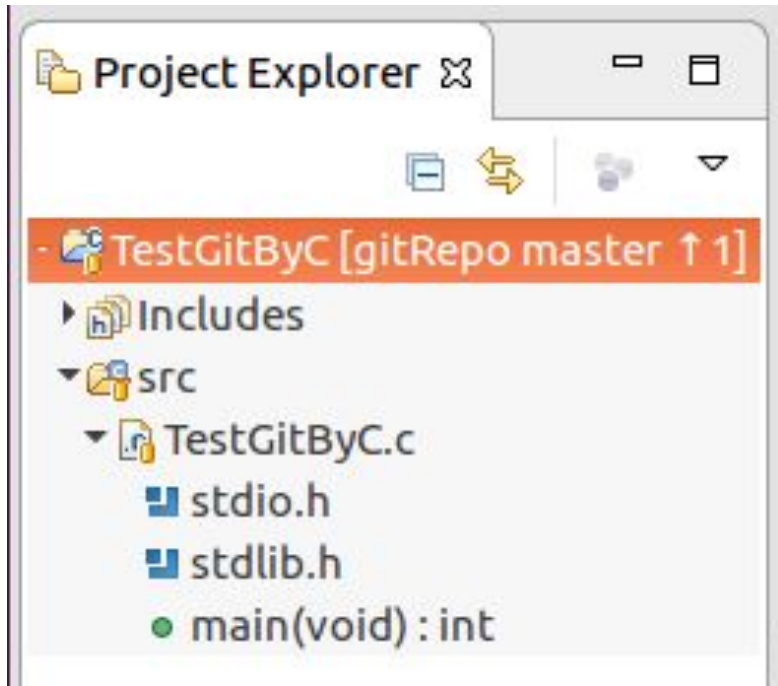
- ❖ Team > Share Project
- ❖ Check 'Use or create repository ...'
- ❖ Click 'Create Repository'
- ❖ Click Finish



Eclipse - Commit & push(1)

- ❖ <https://github.com/> → Login
- ❖ Create New Repository on Github
- ❖ Commit To Local Git
- ❖ Drag files To Staged Panel
- ❖ Click 'Commit & Push'
→ 다음 장 참조.

- ❖ Push To Remote Github As Click 'Commit'



Eclipse - Commit & push(2)

❖ Enroll Remote Git URI

Push Branch master

Destination Git Repository
Enter the location of the destination repository.

Remote name:

Location

URI:

Host:

Repository path:

Connection

Protocol:

Port:

Authentication

User:

Password:

☐ Store in Secure Store

Push Branch master

Push to branch in remote
Select a remote and the name the branch should have in the remote.

Source:
master 07fcbfd second two

Destination:

Remote:

Branch:

☒ Configure upstream for push and pull

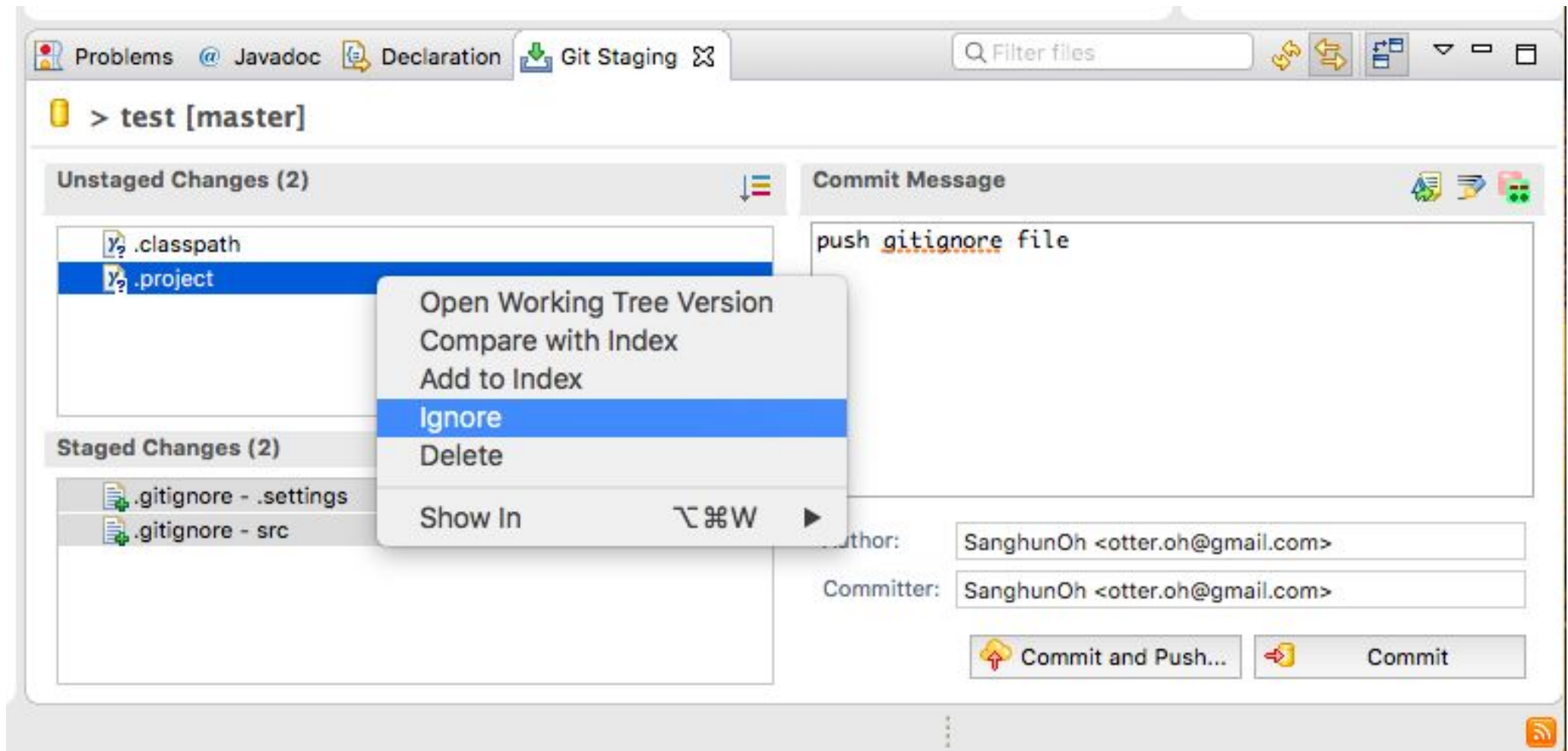
When pulling:

☐ Force overwrite branch in remote if it exists and has diverged

[Show advanced push dialog](#)

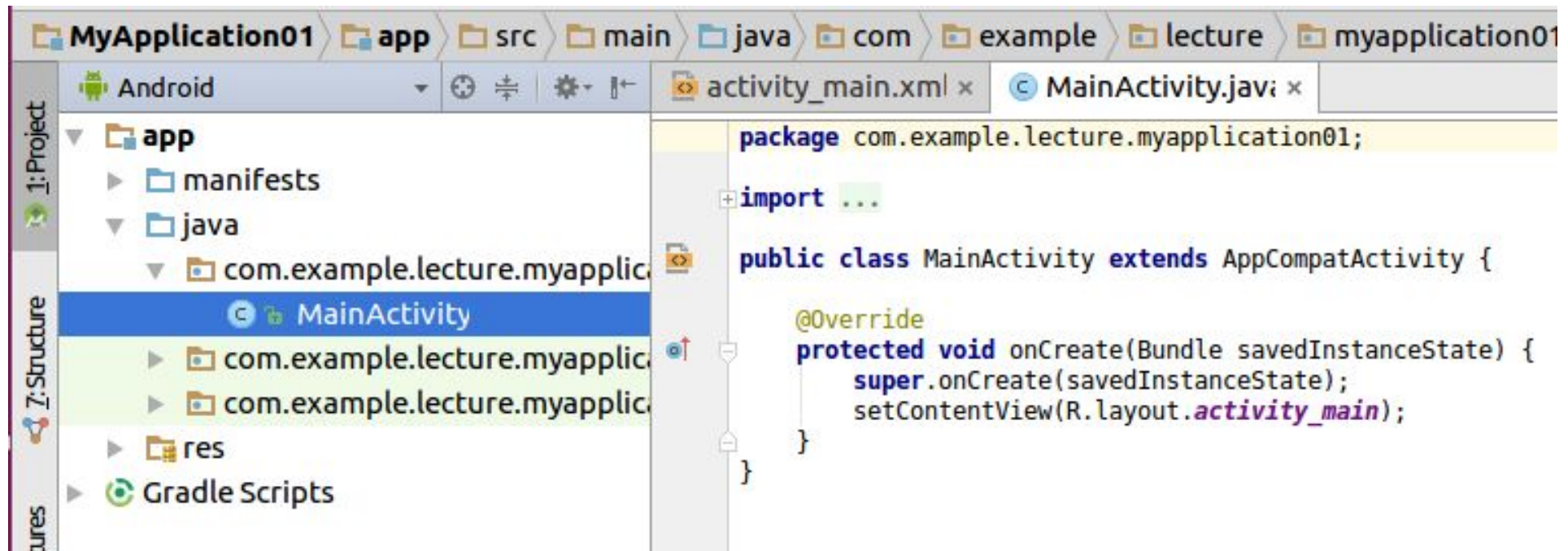
Eclipse - ignore(1)

- ❖ Enroll Remote Git URI
- ❖ ignore Spring files : /target/, /bin/, /.settings, config 파일



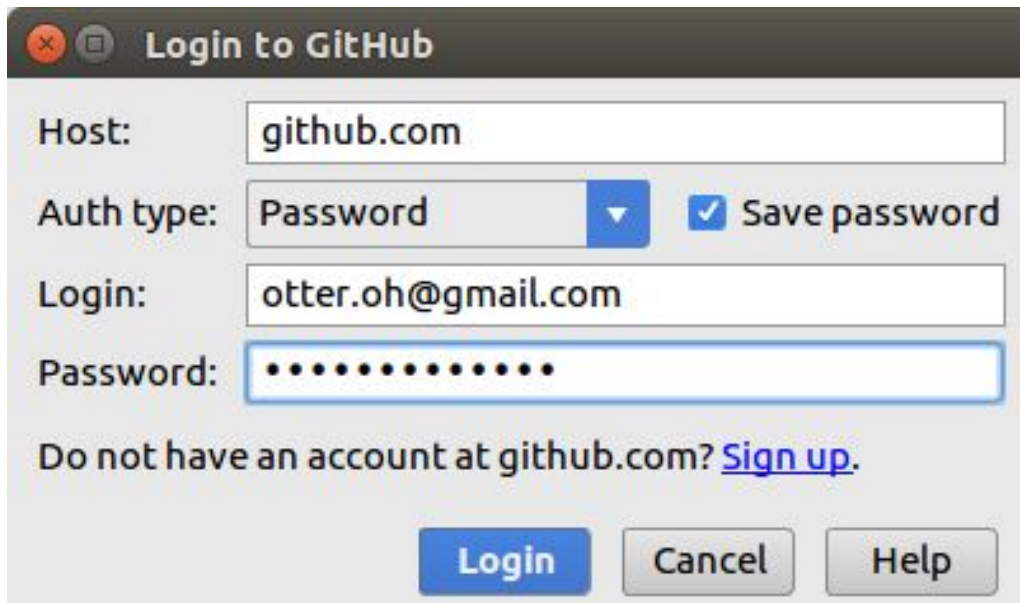
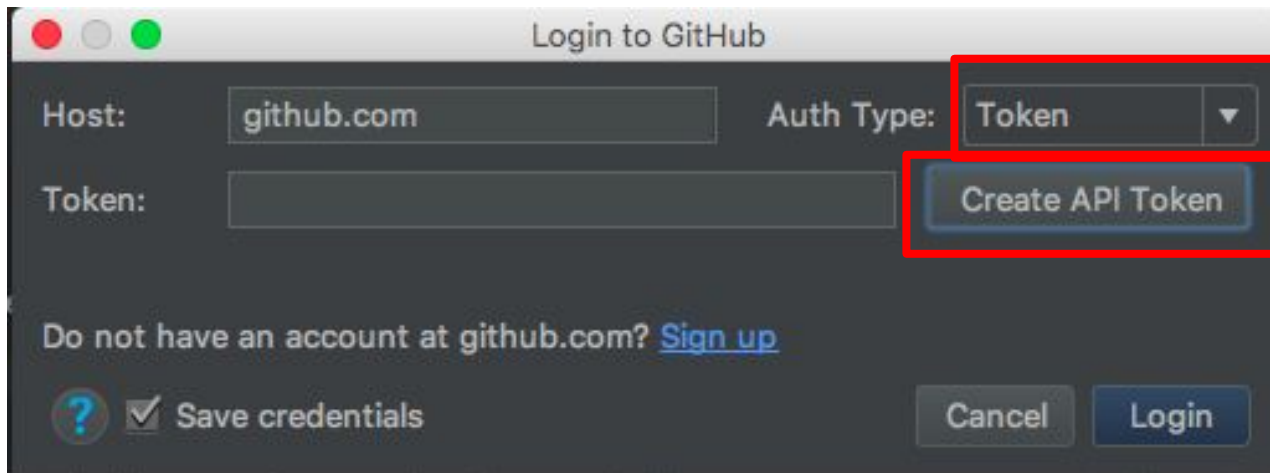
Android Studio(1)

- ❖ Install git
 - Linux OS
 - `sudo apt -y install git`
 - Windows OS
 - <https://git-scm.com/download/win>
- ❖ Create Project



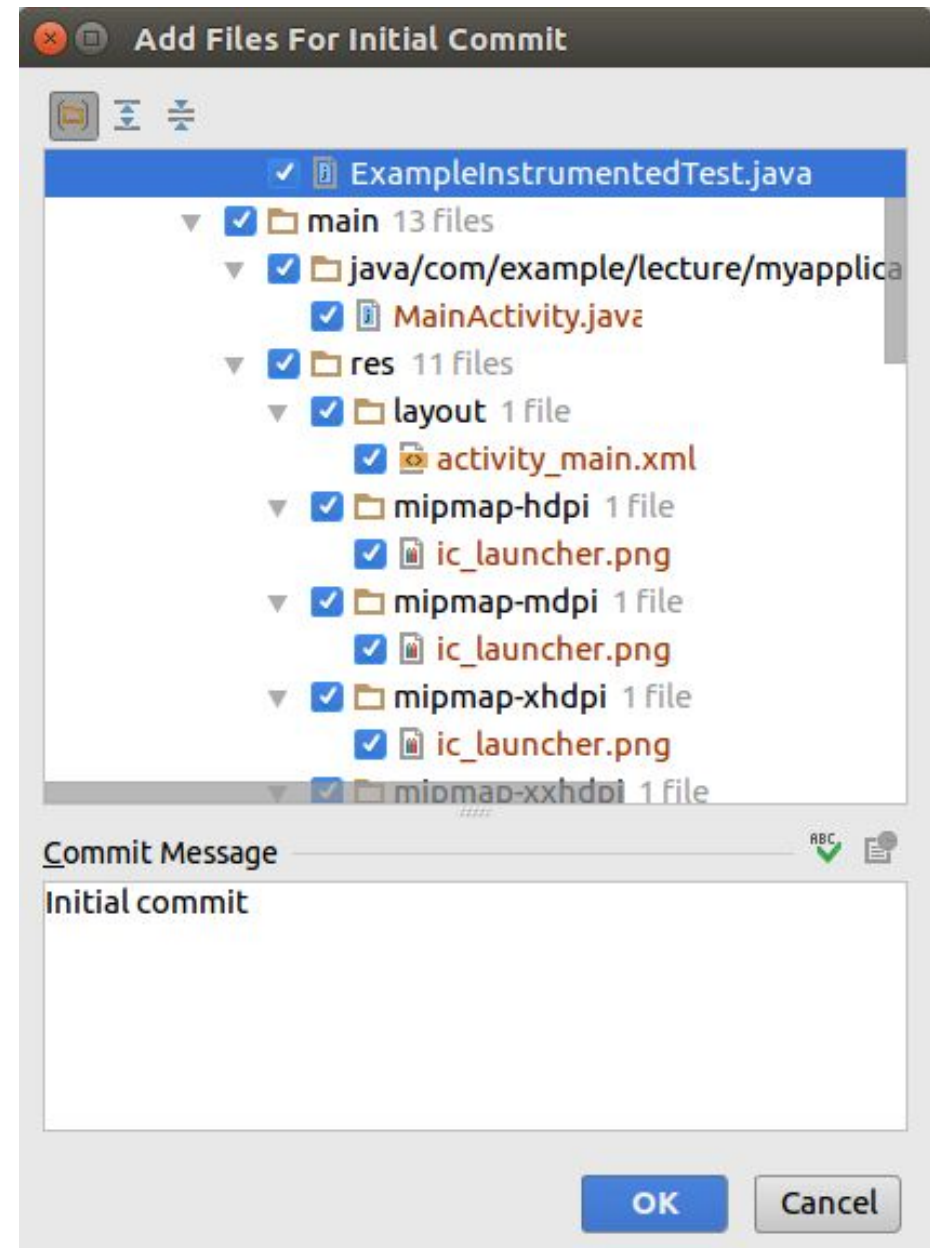
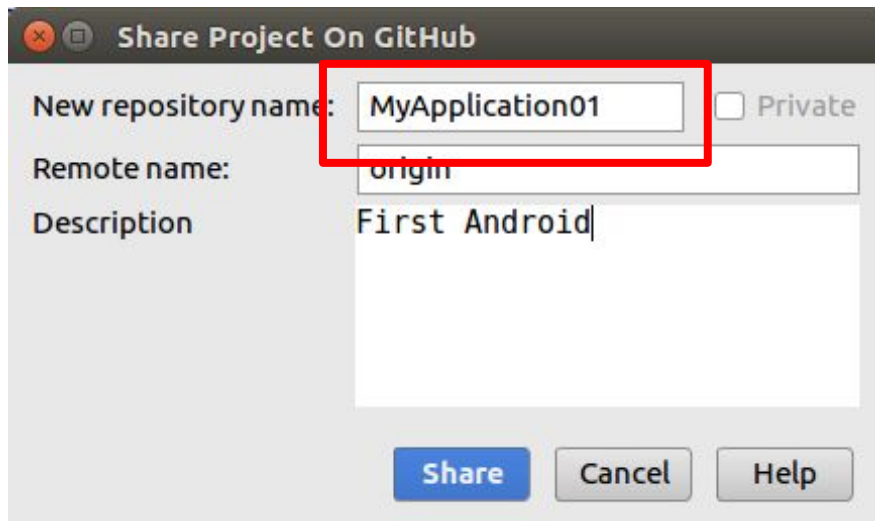
Android Studio(2)

- ❖ Shared Project
 - VCS > Import into Version Control > Share Project on GitHub
- ❖ Login to GitHub (Auth Type : Password 변환)



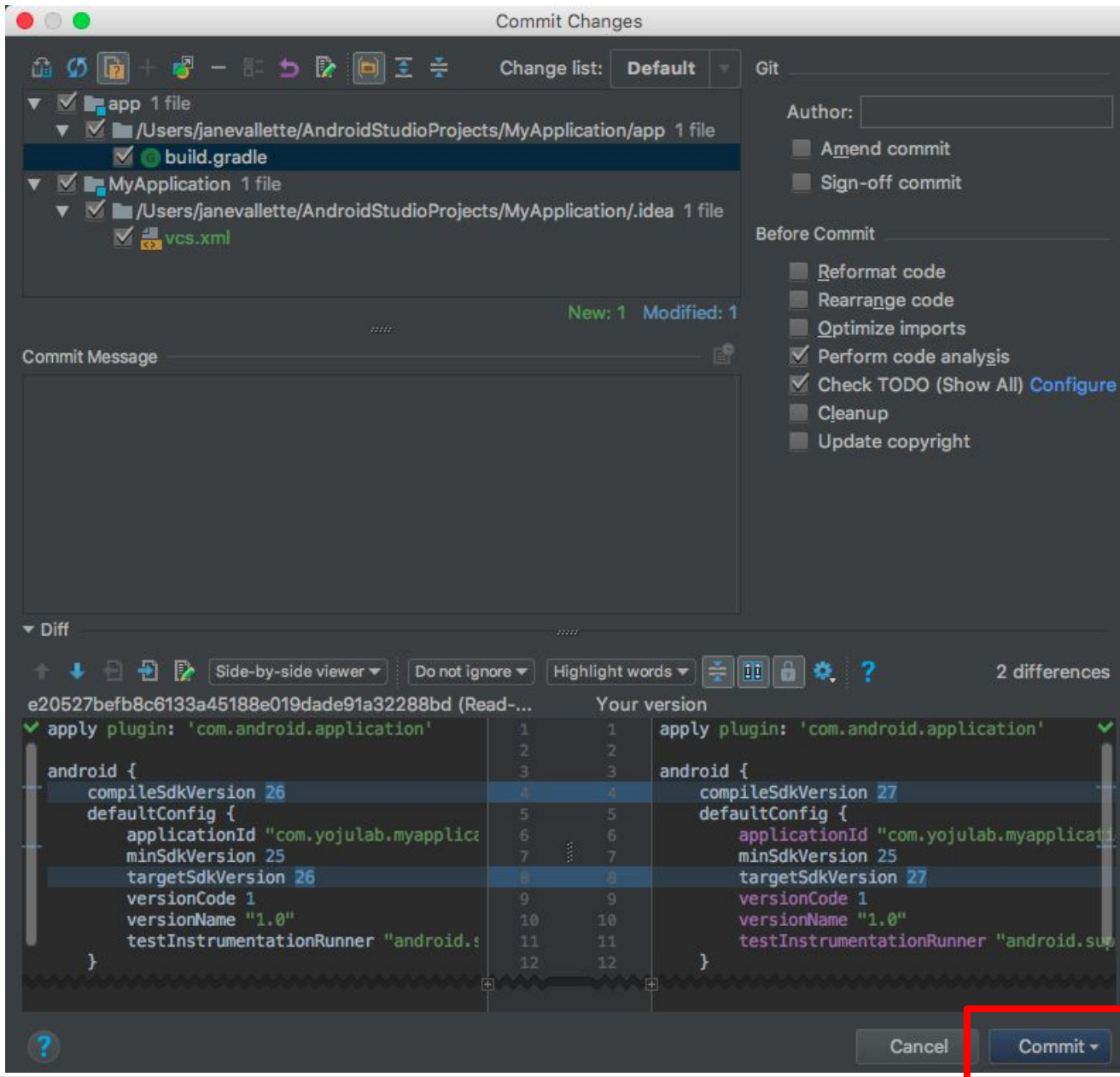
Android Studio(3)

❖ Add Files For Initial Commit (Need Repository Name)



Android Studio(4)

- ❖ commit Project (click Menu > VCS > Commit Changes...)



Android Studio(5)

❖ SVN Ignore

File -> Settings -> Version Control -> Ignored Files

File : local.properties

→ ignore specified file

Mask : *.iml

→ ignore all files matching

Directory : .gradle

→ ignore all files under

Directory : .idea/

→ ignore all files under

Directory : build/

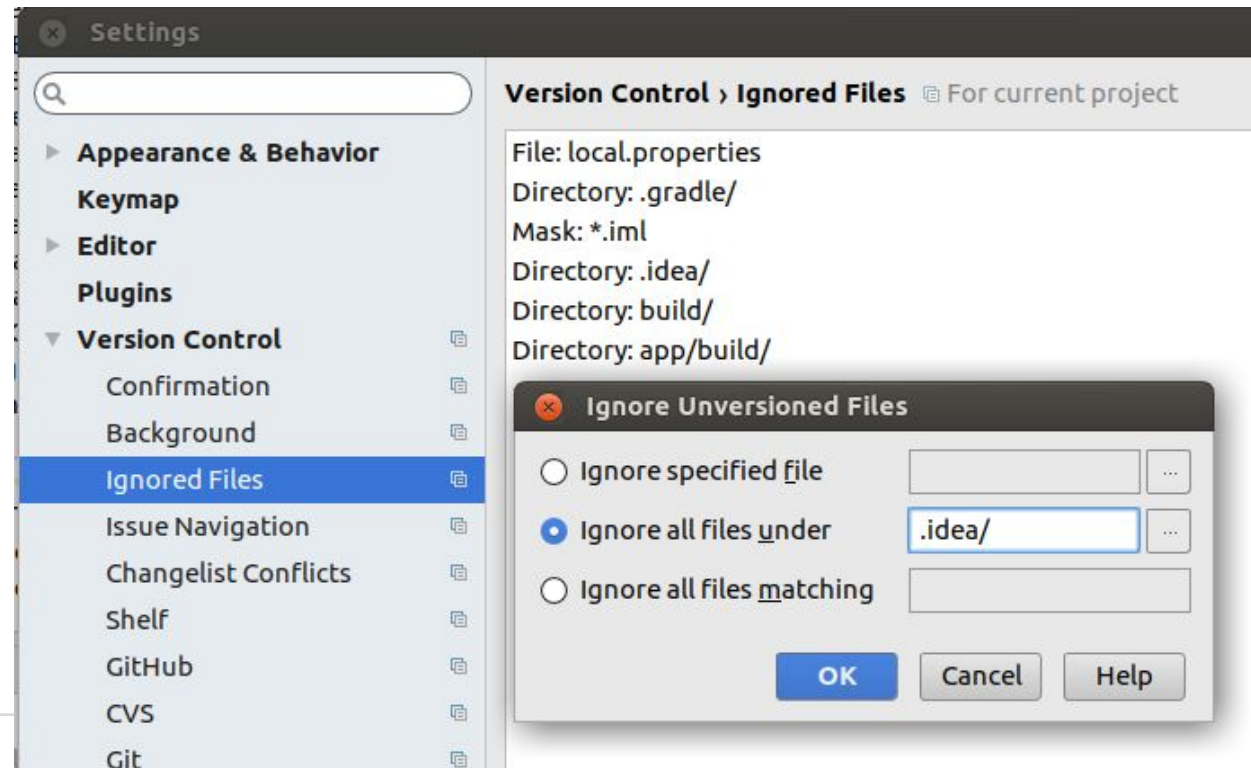
→ ignore all files under

Directory : app/build/

→ ignore all files under

Directory : app/release/

→ ignore all files under

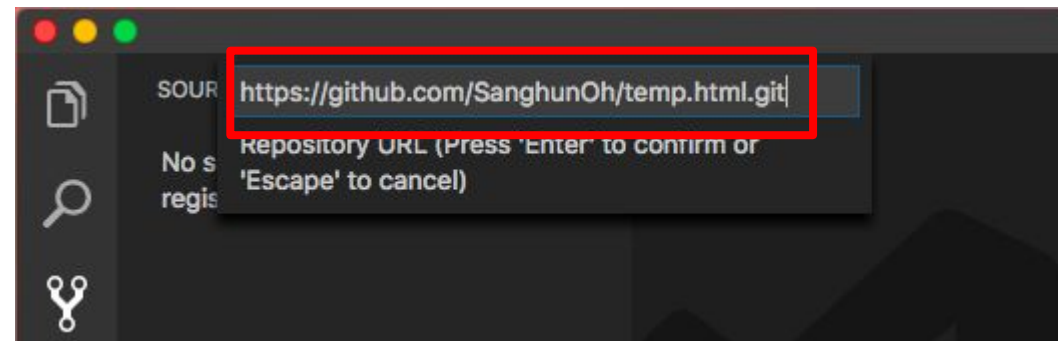
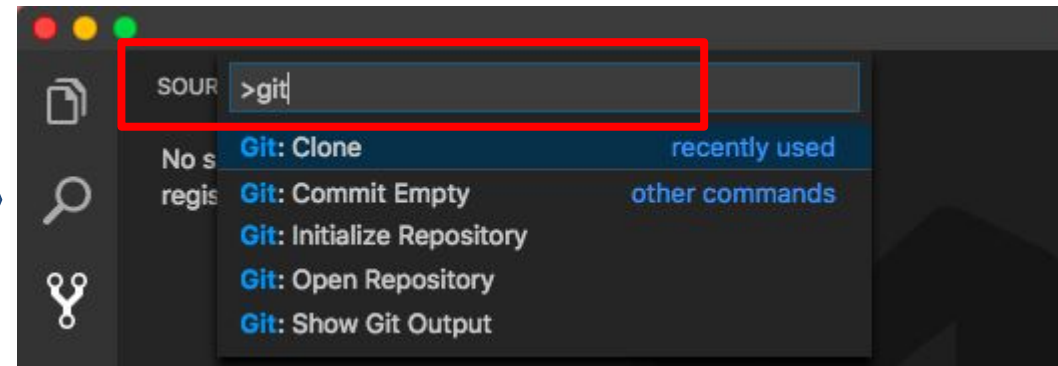
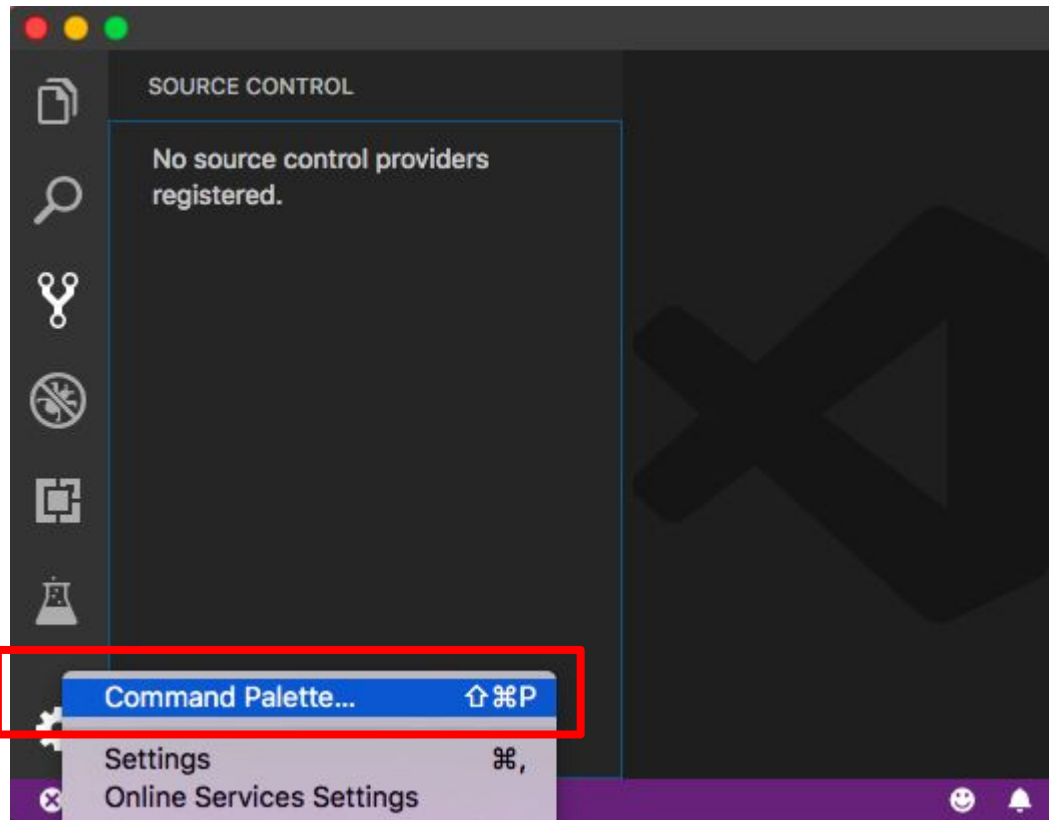


❖ clone Project



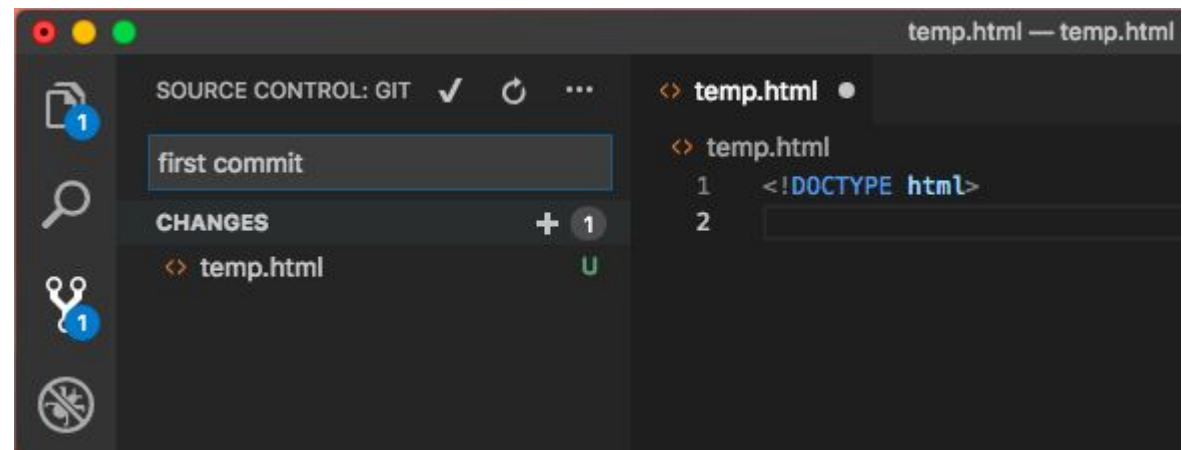
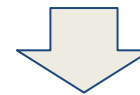
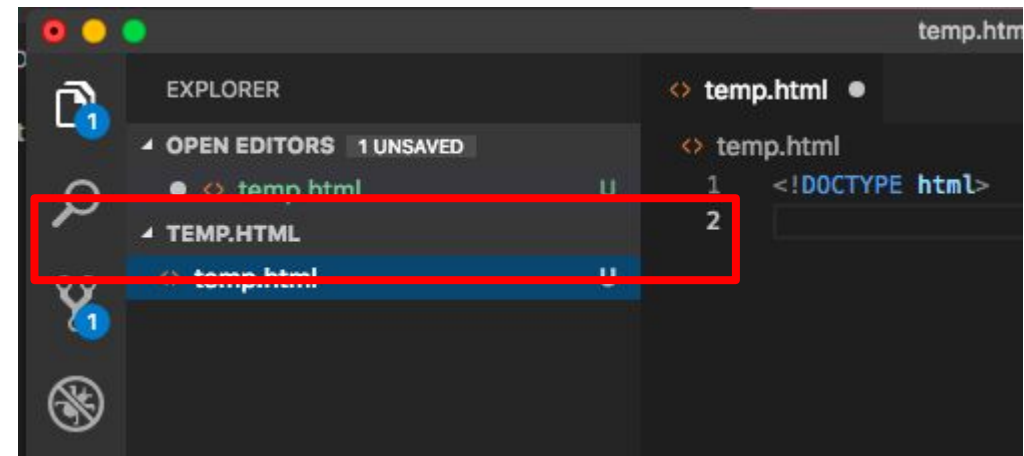
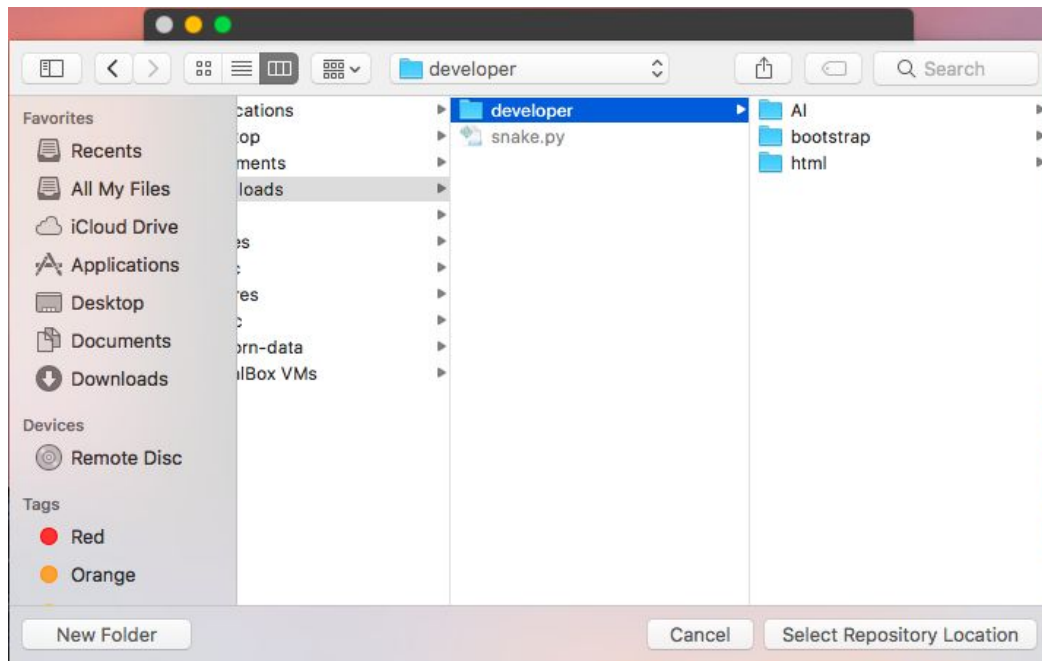
Visual Studio Code(1)

❖ Git Clone



Visual Studio Code(2)

❖ Commit





수고하셨습니다.