

Git 교육자료

2019.01 제작

목차

- Git 이란?
- Git 기초
- Git 브랜치
- Git 고급
- GitHub

Git이란?

- 버전 관리 시스템(VCS, Version Control System)의 일종
: 소스코드 관리, 협업할 때 유용하게 사용

Git의 구조

- Repository: 파일의 정보를 갖고 있는 저장소.
실제 파일들보다 작은 용량을 가짐

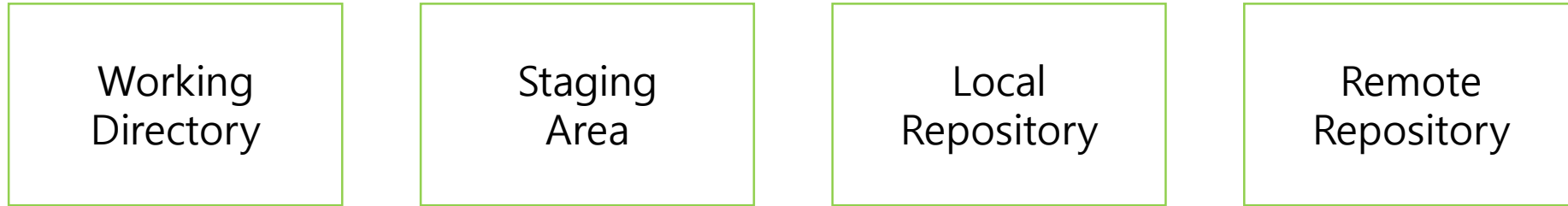
Working
Directory

Staging
Area

Local
Repository

Remote
Repository

Git의 구조



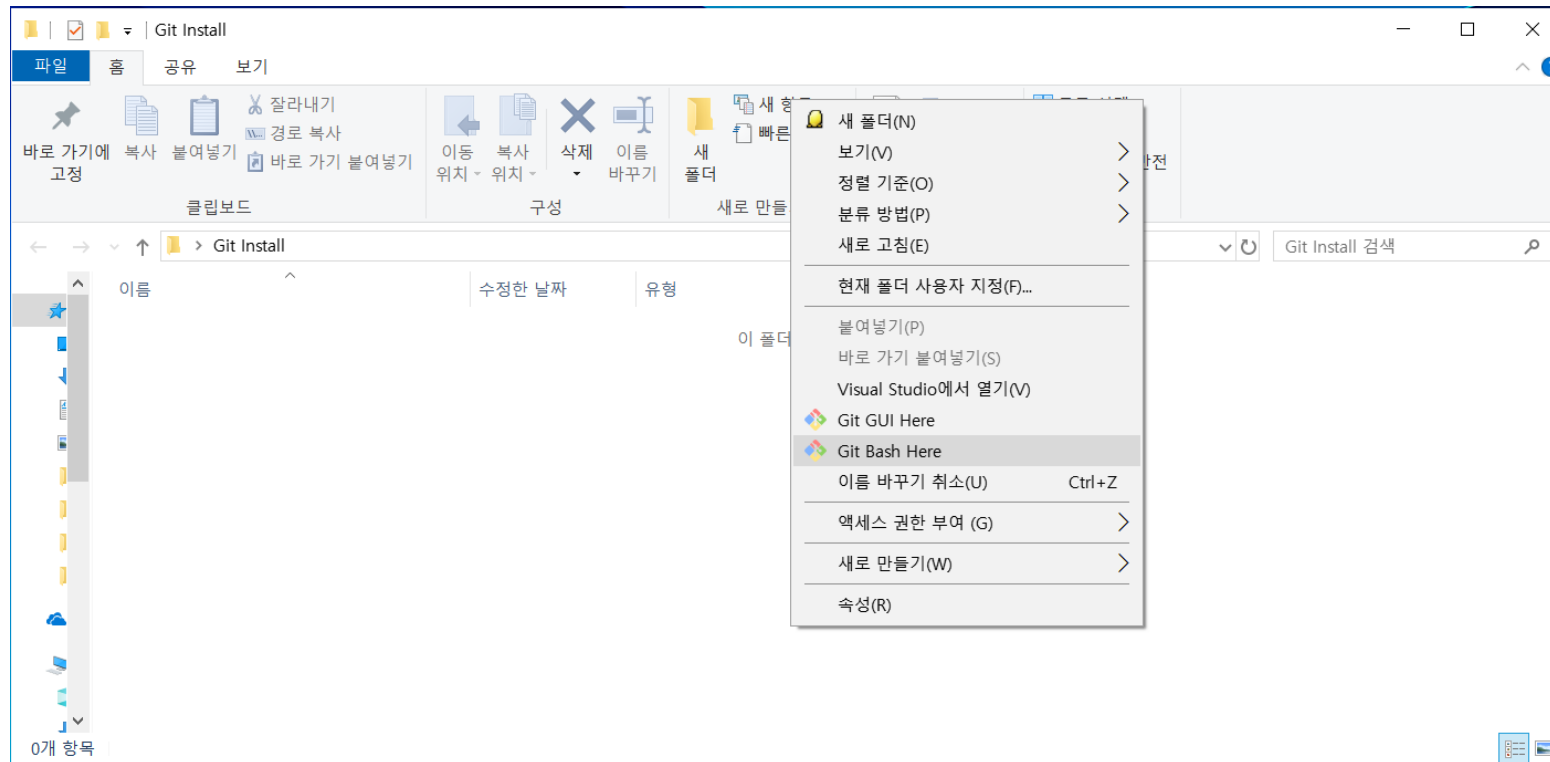
- Working Directory: 현재 작업공간. 파일을 생성, 수정, 삭제하는 것은 Working Directory에서 일어나는 일.
- Staging Area: Working Directory에서 작업한 것을 Local Repository에 올리기 전. 어떤 파일을 올릴 것인지 선택함.
- Local Repository: 내 컴퓨터상의 repository
- Remote Repository: GitHub와 같이 외부에 있는 repository

Git 설치

- <https://www.git-scm.com/downloads> 접속
- 자신의 운영체제에 맞게 다운로드.
- Next 계속 누르면 설치 완료.

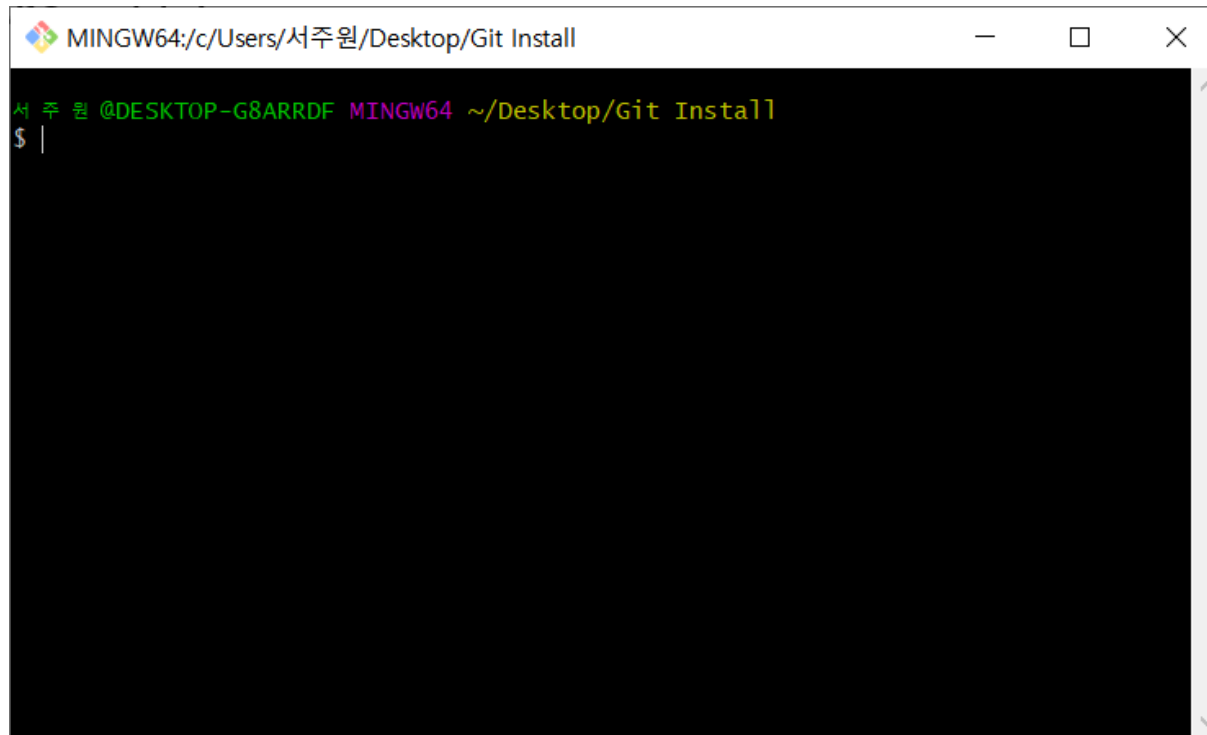
Git 설치

- 폴더 하나 생성 후, 탐색기 안에서 마우스 우클릭 후 Git Bash Here 클릭.



Git 설치

- 다음과 같이 나오면 성공. Git bash에서는 쉘 명령어 사용 가능.



A screenshot of a Windows command prompt window. The title bar reads "MINGW64:/c/Users/서주원/Desktop/Git Install". The command prompt shows the user "서 주 원" at "DESKTOP-G8ARRDF" in a "MINGW64" environment, with the current directory being "~/Desktop/Git Install". The prompt is "\$ |", indicating that the shell is ready for input.

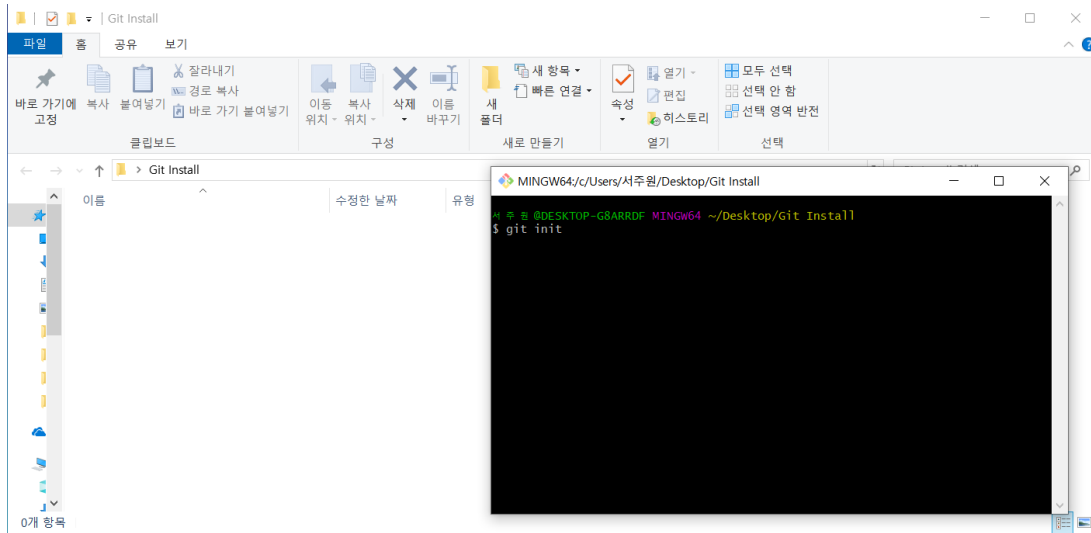
Git 기초 – 사용자 정보 등록

- 사용자 이름과 이메일 주소를 설정해야한다.
- `git config --global user.name "홍길동"`
- `git config --global user.email "GilDong@khu.ac.kr"`
- `git config --list` : 설정한 정보 확인 가능

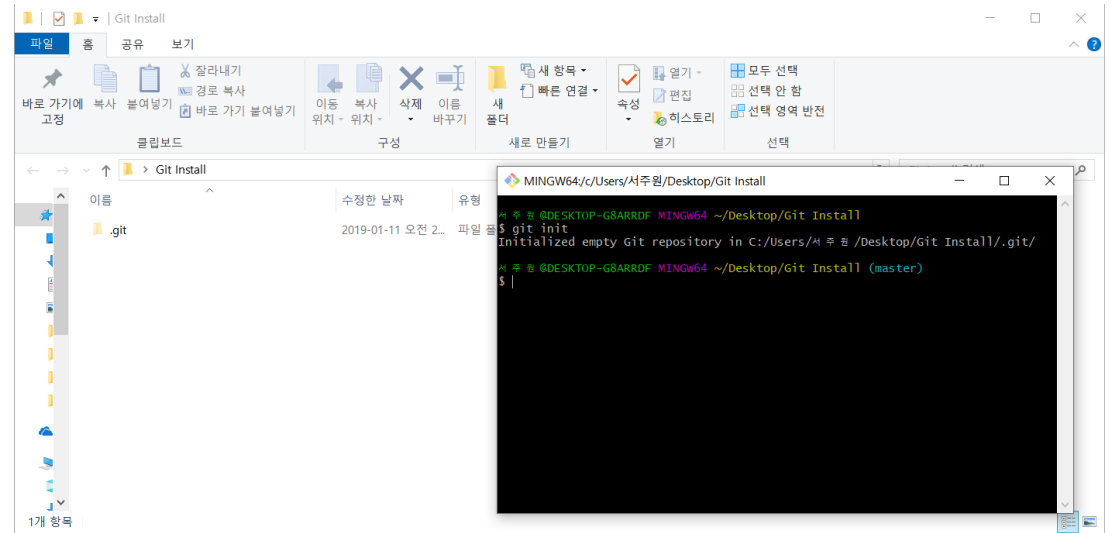
Git 기초 – git init

- git init: 현재 폴더에서 git 시작
- .git(숨겨진 폴더)가 생성됨. 이 폴더에 들어갈 일은 없음.

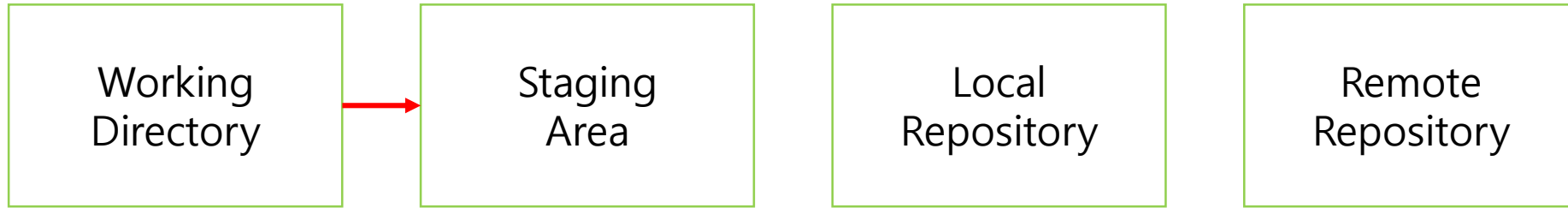
before



after

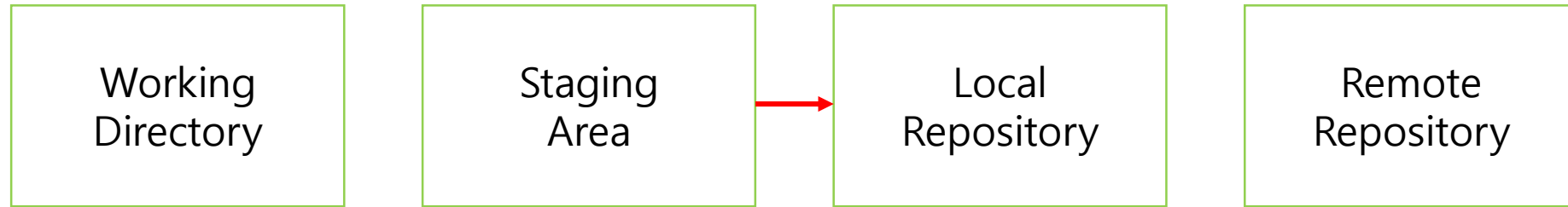


Git 기초 – git add



- git add 파일명: 해당 파일을 staging area에 올림
- git add . : 해당 디렉토리 내의 모든 파일을 git add함

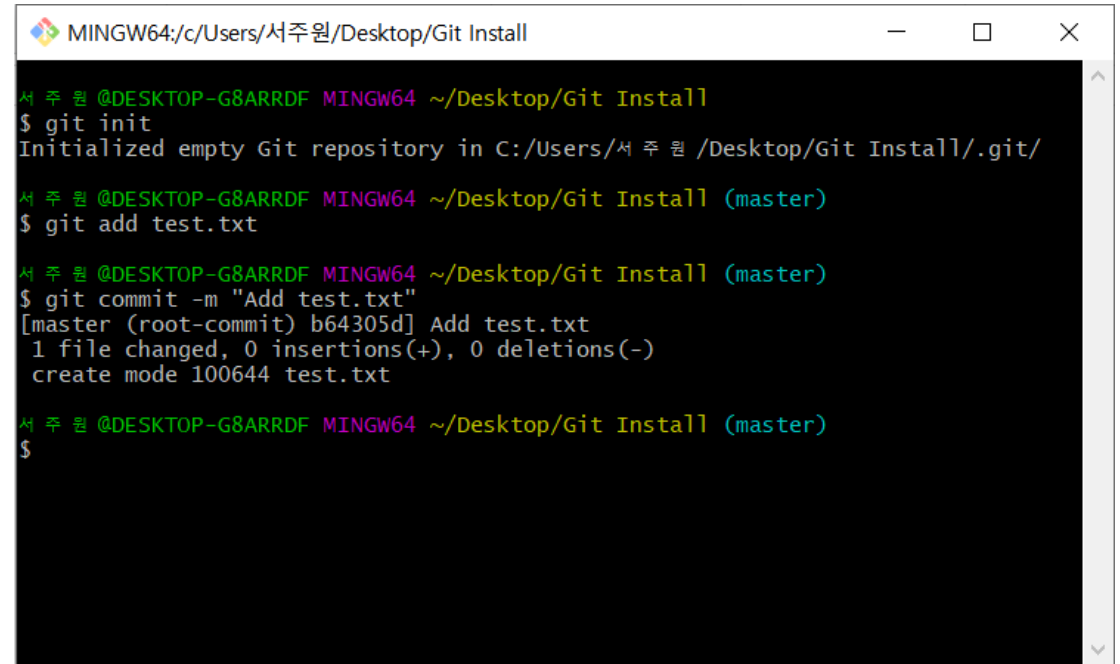
Git 기초 – git commit



- 파일의 현재 상태를 저장
- `git commit -m "commit message"`: staging area에 있던 파일들을 Local Repository에 올림.
- 이때 커밋 메시지는 보통 무엇을 했는 지, 신경써서 영어로 적어 주는 것이 중요.

Git 실습 1 – git add & git commit

- test.txt라는 새 텍스트 파일을 생성
- git add test.txt
- git commit -m "Add test.txt"



```
MINGW64:/c:/Users/서주원/Desktop/Git Install
서 주 원 @DESKTOP-G8ARRDF MINGW64 ~/Desktop/Git Install
$ git init
Initialized empty Git repository in C:/Users/서 주 원 /Desktop/Git Install/.git/

서 주 원 @DESKTOP-G8ARRDF MINGW64 ~/Desktop/Git Install (master)
$ git add test.txt

서 주 원 @DESKTOP-G8ARRDF MINGW64 ~/Desktop/Git Install (master)
$ git commit -m "Add test.txt"
[master (root-commit) b64305d] Add test.txt
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 test.txt

서 주 원 @DESKTOP-G8ARRDF MINGW64 ~/Desktop/Git Install (master)
$
```

Git 실습 2 – git add & git commit

파일의 수정, 삭제도 같은 방식으로 진행

- 파일 수정

1. 파일을 수정한다.
2. git add <파일명>
3. git commit -m <commit message>

Git 실습 3 – git add & git commit

- 파일 삭제

1. 파일을 삭제한다.
2. `git add <파일명>`
3. `git commit -m <commit message>`

또는

1. `git rm --cached <파일명>` : 실제로는 삭제되지 않았지만 삭제되었다고 함.
2. `git commit -m <commit message>`

Git 기초 – git status

- git status: working directory에서 파일의 생성, 수정, 삭제가 발생했지만, staging area에 반영되지 않은 경우. (git add를 안함)

또는, staging area에 올라와 있지만, commit 되지 않은 경우를 알 수 있음.

Git 실습 4 – git status

- test.txt를 수정해보자.
- git status

```
서주원 @DESKTOP-G8ARRDF MINGW64 ~/Desktop/Git Install (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   test.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

- test.txt가 수정되었고, staging area에 반영되지 않았다고 알려준다.

Git 실습 4 – git status

- git add test.txt
- git status

```
서주원@DESKTOP-G8ARRDF MINGW64 ~/Desktop/Git Install (master)
$ git status
On branch master
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   test.txt
```

- 이제 commit 해주면 된다.

Git 기초 - .gitignore

- 모든 파일을 커밋하지는 않는다.
- 빌드 파일과 같이, 크게 중요하지 않고, 재생성 가능한 파일들은 commit할 때 제외하는 것이 편하다.

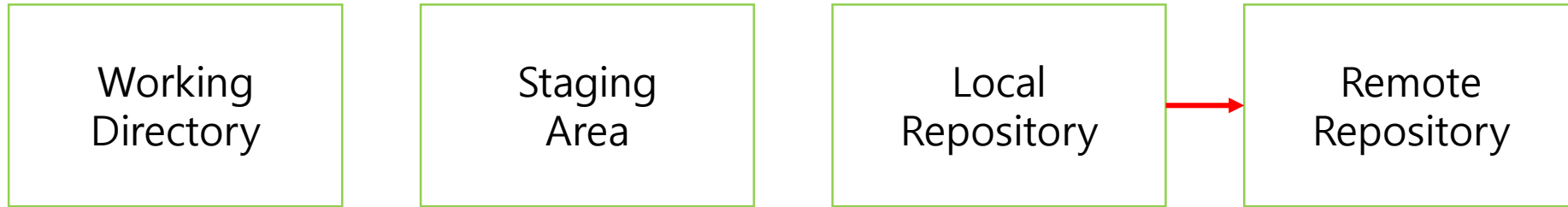
Git 기초 - .gitignore

- touch .gitignore : .gitignore 생성.
- .gitignore을 열고, 제외하고자 하는 파일명 또는 폴더이름, 또는 파일 확장자(*.exe와 같은 형식) 작성.

Git 기초 – git clone

- GitHub와 같은 remote repository에서 그대로 가져와 시작 할 수 있다. git init 불필요.
- git clone <HTTPS 주소 or SSH주소>

Git 기초 – git push



- `git push origin <branch name>` : origin이라는 remote repository에 해당 브랜치의 데이터를 전송한다. 브랜치는 추후에 다룰 예정

Git 실습 5 – remote repository 생성

- GitHub(<https://github.com>)에 가입.
- 로그인 후, 좌측의 new repository 클릭.



Git 실습 5 – remote repository 생성

- Repository name을 입력하고 create repository 클릭

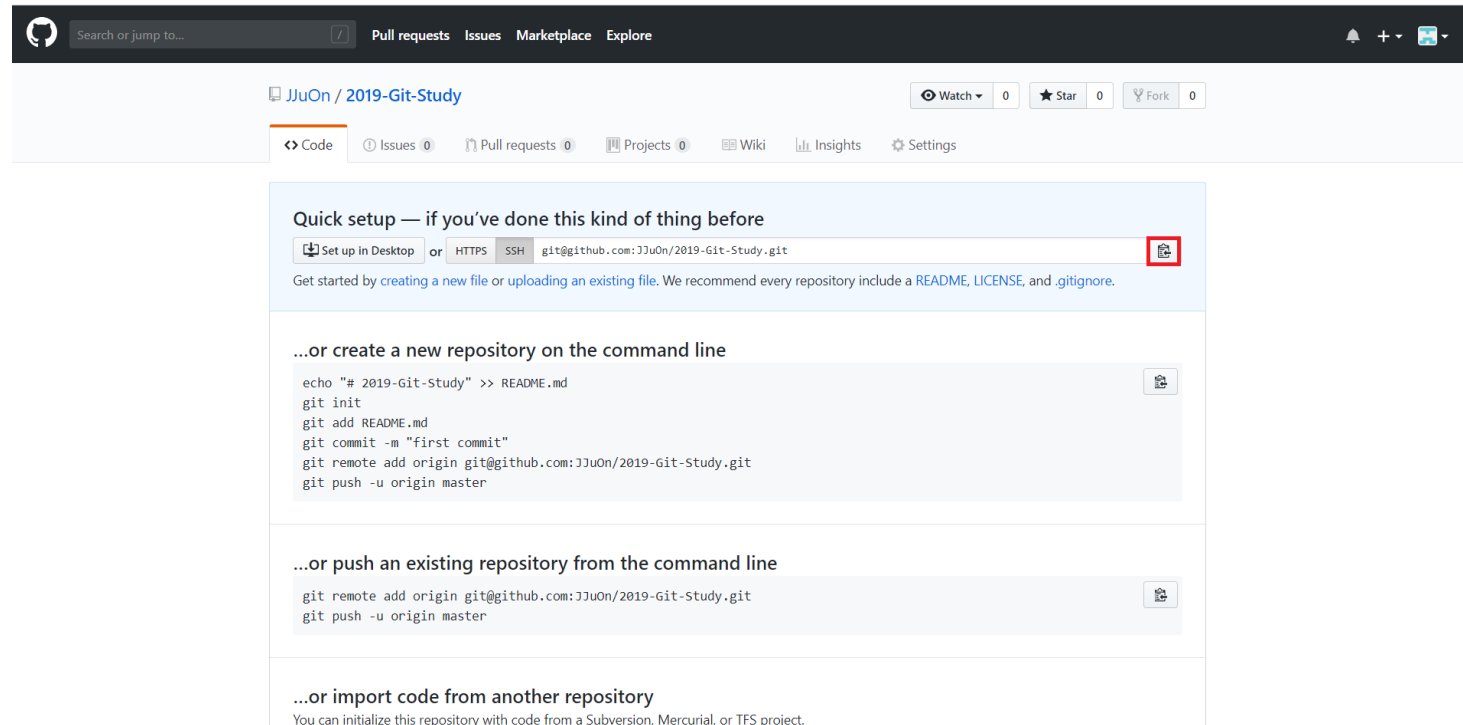
- 중간의 public vs private에 대해,

GitHub에서 생성되는 repository는 기본적으로 오픈소스이다.

이전에는, private는 다른 사람에게 소스코드를 공개하지 않지만 유료 이용자만 선택할 수 있었다.

하지만 2019년 1월자로, 무료로 private를 선택할 수 있게 되었다. 단, 협업은 3인까지만 가능하다.

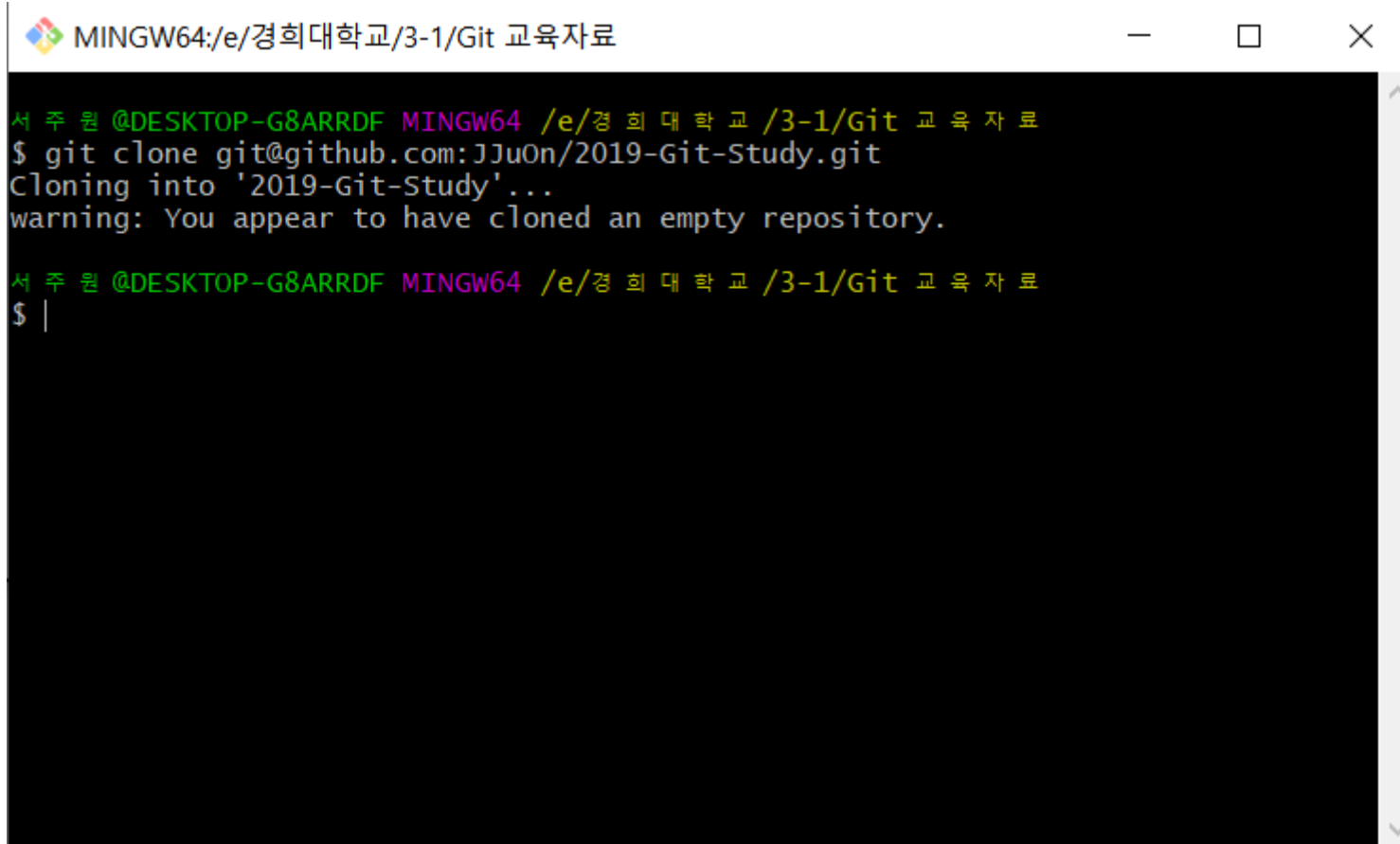
Git 실습 5 – remote repository 생성



- 빨간 부분을 클릭하면, 해당 repository의 SSH주소가 복사된다.

Git 실습 5 – git clone & git push

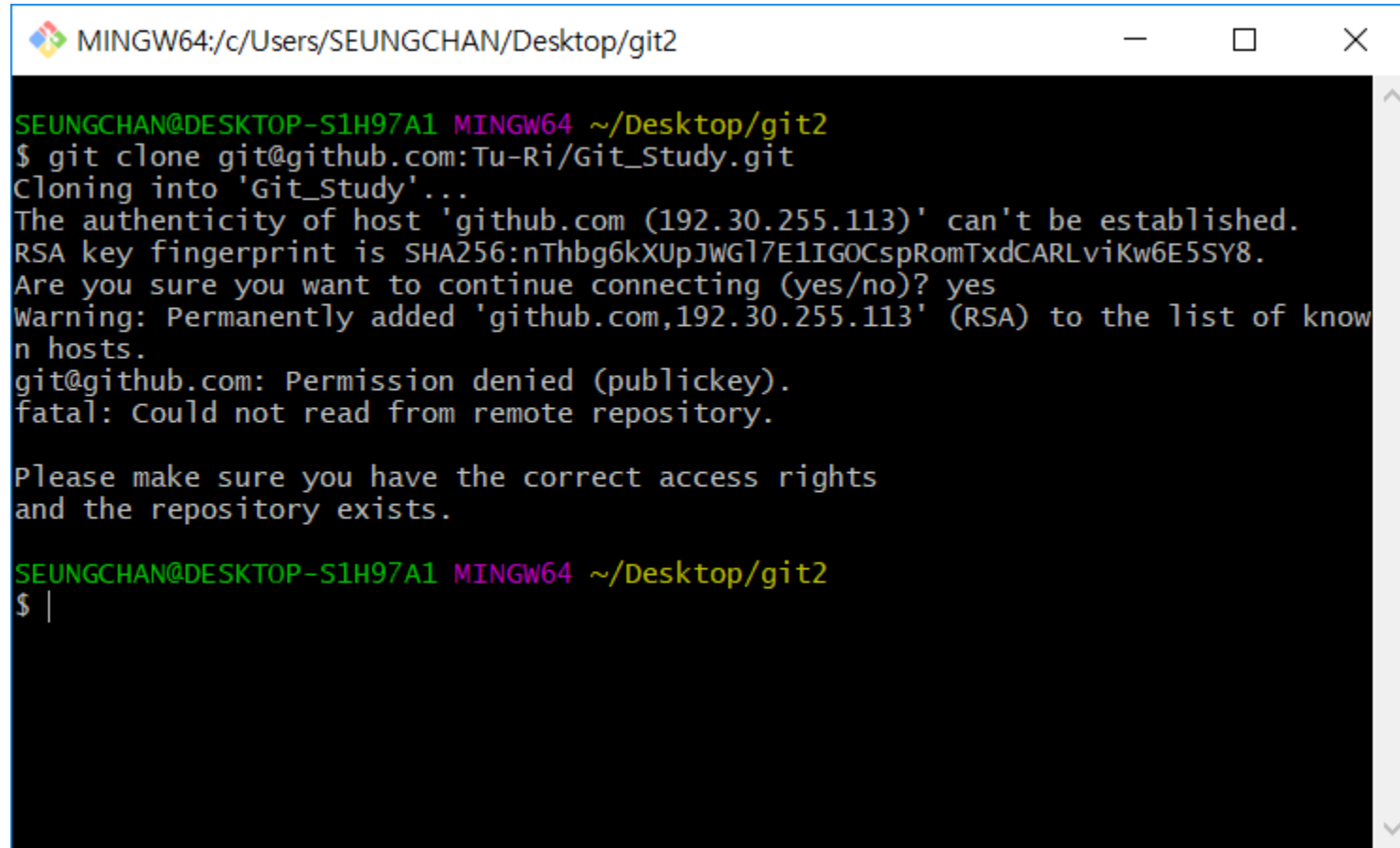
- git clone <복사한 주소>



```
MINGW64:/e/경희대학교/3-1/Git 교육자료
서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료
$ git clone git@github.com:JJuOn/2019-Git-Study.git
Cloning into '2019-Git-Study'...
warning: You appear to have cloned an empty repository.

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료
$ |
```

Git 실습 5 – git clone & git push



```
MINGW64:/c/Users/SEUNGCHAN/Desktop/git2

SEUNGCHAN@DESKTOP-S1H97A1 MINGW64 ~/Desktop/git2
$ git clone git@github.com:Tu-Ri/Git_Study.git
Cloning into 'Git_Study'...
The authenticity of host 'github.com (192.30.255.113)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWG17E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.255.113' (RSA) to the list of known hosts.
git@github.com: Permission denied (publickey).
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

SEUNGCHAN@DESKTOP-S1H97A1 MINGW64 ~/Desktop/git2
$ |
```

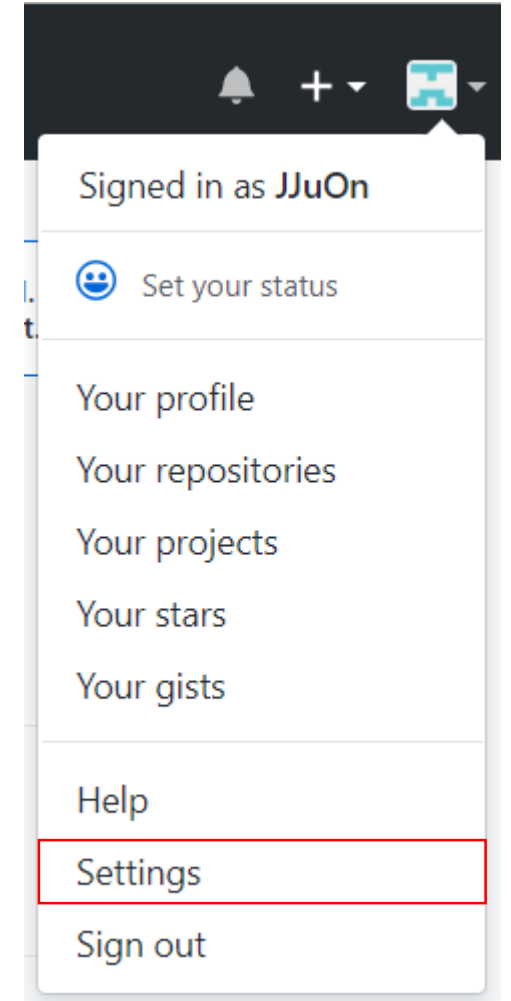
- 처음 git clone을 하게 되면 인증 오류가 뜨게 된다.

Git 실습 5 – git clone & git push

- `ssh-keygen -t rsa -C "your_email@example.com"`
- 계속 엔터를 누르고
- `C:\Users\사용자\ssh` 에서 `id_rsa.pub`의 내용 전부 복사(워드패드, 노트패드 등 이용.)

Git 실습 5 – git clone & git push

- GitHub의 우측 상단 Settings 클릭



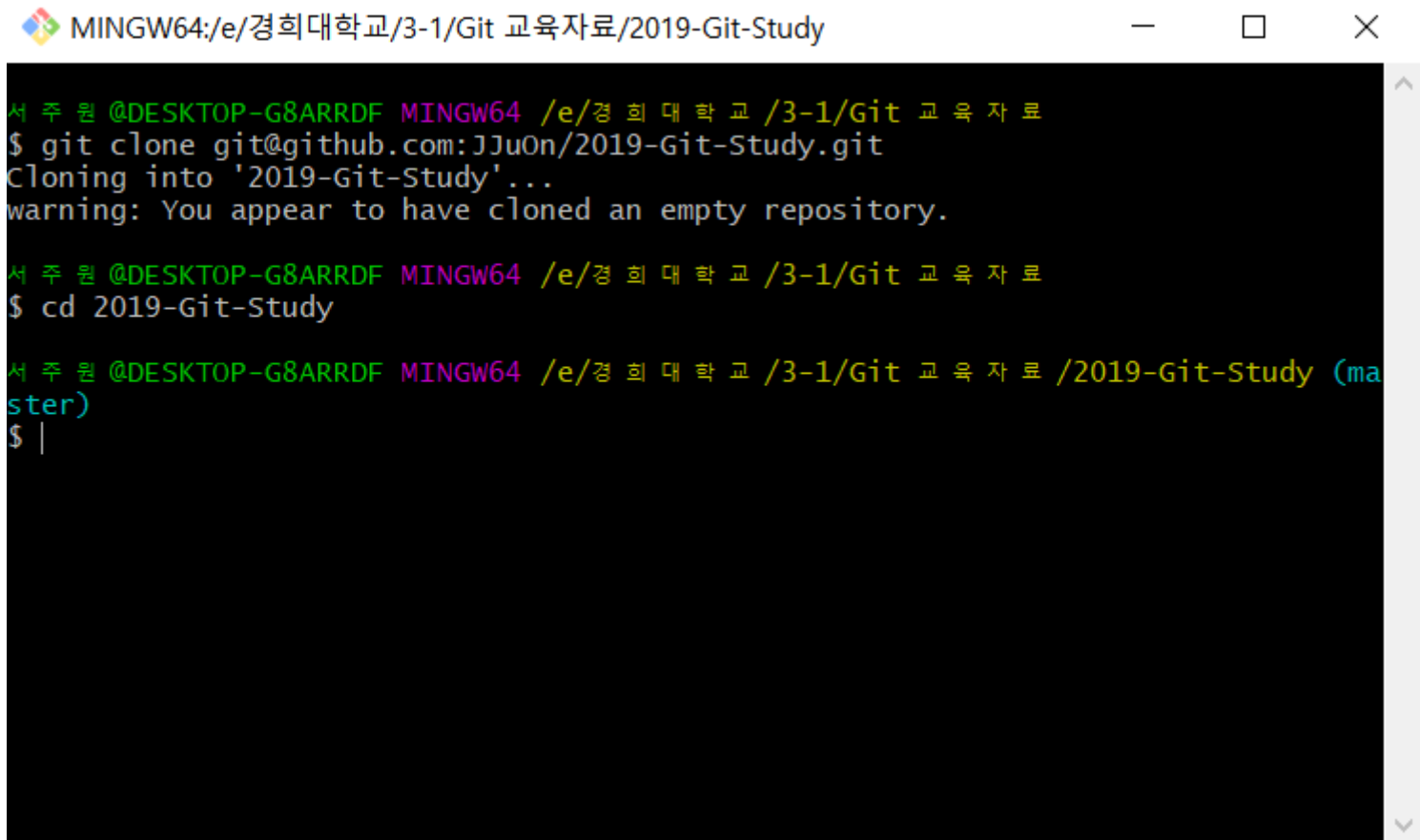
Git 실습 5 – git clone & git push

- 좌측 SSH and GPG keys 클릭
- New SSH key 클릭 후 복사했던 것 붙여넣기
- 다시 git clone <HTTPS or SSH 주소>

Personal settings
Profile
Account
Emails
Notifications
Billing
SSH and GPG keys
Security
Sessions
Blocked users
Repositories
Organizations
Saved replies
Applications

Git 실습 5 – git clone & git push

- cd <git clone을 통해 생성된 폴더>: 현재 위치를 이동한다.



```
MINGW64:/e/경희대학교/3-1/Git 교육자료/2019-Git-Study
서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료
$ git clone git@github.com:JJuOn/2019-Git-Study.git
Cloning into '2019-Git-Study'...
warning: You appear to have cloned an empty repository.

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료
$ cd 2019-Git-Study

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (ma
ster)
$ |
```

Git 실습 5 – git clone & git push

- touch README.md : README.md 파일 생성
- README.md 파일 수정
- git add README.md
- git commit -m "Add README.md"

※ .md 파일은 마크다운 파일로, jupyter notebook에서 처럼 꾸밀 수 있다.

Git 실습 5 – git clone & git push

```
MINGW64:/e/경희대학교/3-1/Git 교육자료/2019-Git-Study
Cloning into '2019-Git-Study'...
warning: You appear to have cloned an empty repository.

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료
$ cd 2019-Git-Study

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (ma
ster)
$ touch README.md

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (ma
ster)
$ git add README.md

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (ma
ster)
$ git commit -m "Add README.md"
[master (root-commit) bf83dcc] Add README.md
1 file changed, 1 insertion(+)
create mode 100644 README.md

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (ma
ster)
$ |
```

Git 실습 5 – git clone & git push

- git push origin master

```
MINGW64:/e/경희대학교/3-1/Git 교육자료/2019-Git-Study
서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (ma
ster)
$ git add README.md

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (ma
ster)
$ git commit -m "Add README.md"
[master (root-commit) bf83dcc] Add README.md
1 file changed, 1 insertion(+)
create mode 100644 README.md

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (ma
ster)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 273 bytes | 273.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:JJuOn/2019-Git-Study.git
 * [new branch]      master -> master

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (ma
ster)
$
```

Git 실습 5 – git clone & git push

- GitHub가서 확인

The screenshot shows the GitHub interface for a repository named 'JJuOn / 2019-Git-Study'. The repository has 1 commit, 1 branch, 0 releases, and 1 contributor. The main branch is 'master'. The repository description is 'No description, website, or topics provided.' The repository contains a file named 'README.md' which was added by 'jwseo001' 3 minutes ago. The content of the README.md file is '2019학년도 git스터디를 위한 repository입니다.' The footer of the page shows the GitHub logo and links to 'Contact GitHub', 'Pricing', 'API', 'Training', 'Blog', and 'About'.

Search or jump to... Pull requests Issues Marketplace Explore

JJuOn / 2019-Git-Study Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. Edit

Manage topics

1 commit 1 branch 0 releases 1 contributor

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

jwseo001 Add README.md Latest commit bf83dcc 3 minutes ago

README.md Add README.md 3 minutes ago

README.md

2019학년도 git스터디를 위한 repository입니다.

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About

Git 기초 – git remote add

- git clone에서는 git init을 따로 하지 않았음을 알 수 있었다.
- 그렇다면 이미 local에서 작업하고 있었다면, remote와 연결할 수 없는가? => X
- git remote add origin <HTTPS 주소 or SSH 주소>: origin이라는 remote repository에 연결한다.

Git 기초 – git fetch & git pull

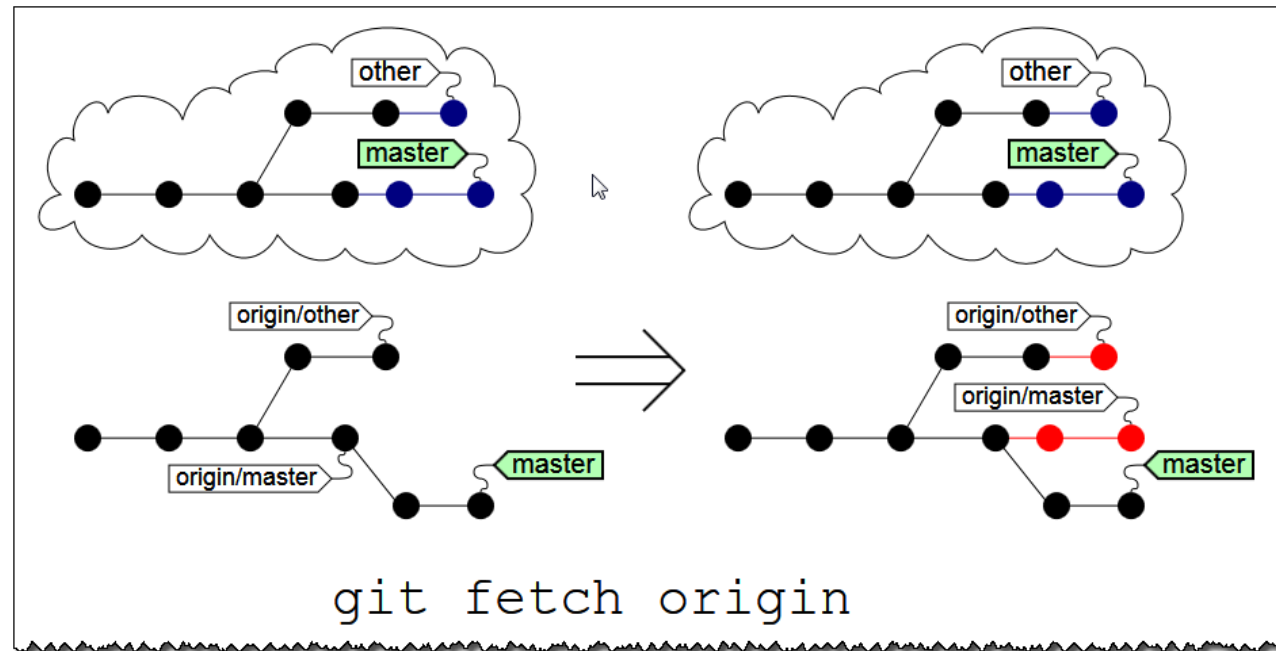
- git clone은 remote repository를 통째로 가져왔다.
- 이미 작업하던 local repository에서 remote repository를 가져오고 싶다면 => git fetch, git pull

Git 기초 – git fetch & git pull

- `git fetch origin <branch>`: origin에서 해당 브랜치의 데이터를 가져온다.
- `git pull origin <branch>`: origin에서 해당 브랜치의 데이터를 가져온 후, local과 merge시킨다. Merge는 추후에 설명.
- 주로 `git pull`을 사용

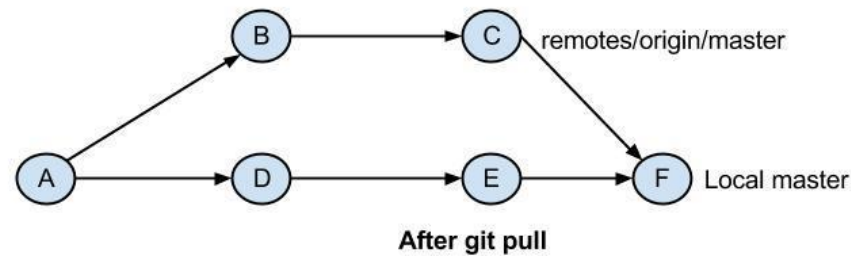
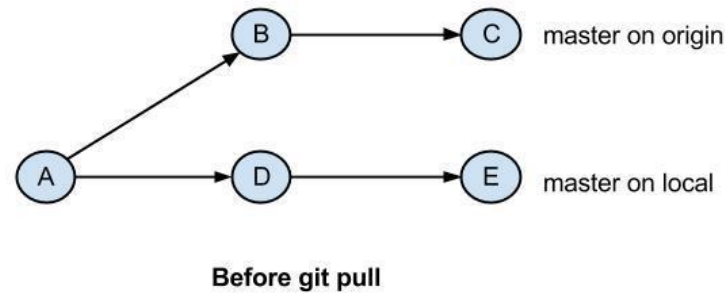
Git 기초 – git fetch & git pull

- git fetch vs git pull



Git 기초 – git fetch & git pull

- git fetch vs git pull



Git 실습 6 – git remote & git pull

- git init
- GitHub에서 이전에 생성했던 repository 주소 복사
- git remote add origin <복사했던 주소>
- git pull origin master

Git 기초 – git remote & git pull

```
MINGW64:/e/경희대학교/3-1/Git 교육자료/2019-Git-Study

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study
$ git init
Initialized empty Git repository in E:/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study/.git/

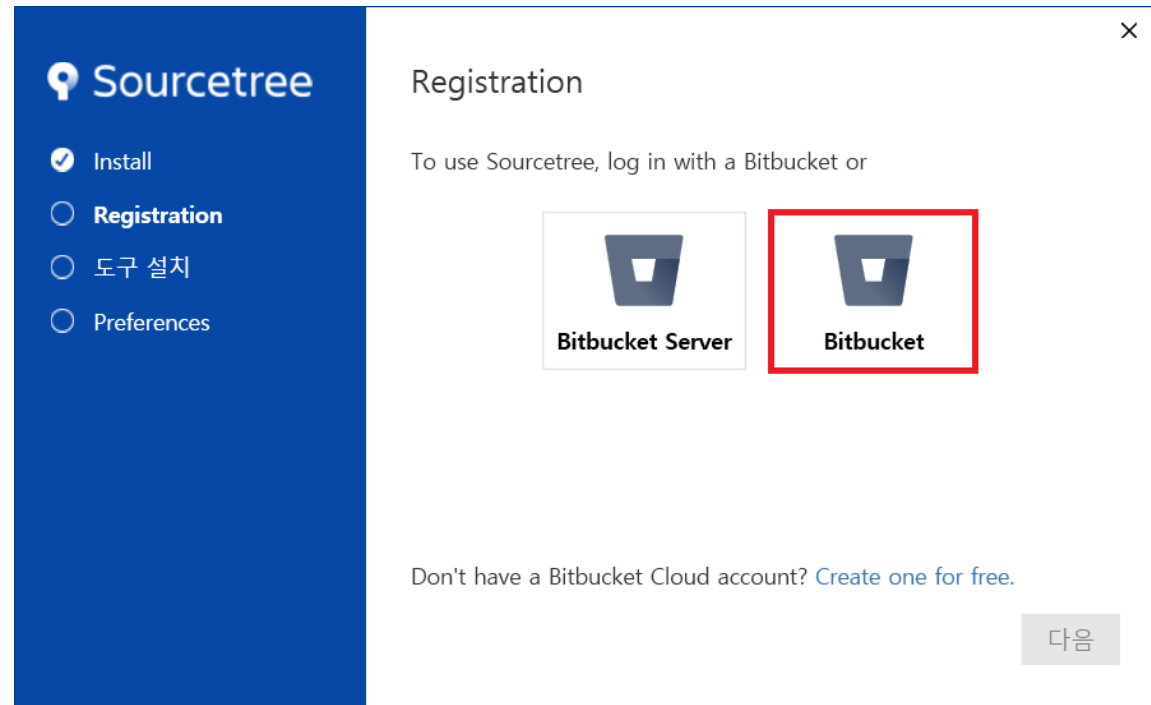
서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (master)
$ git remote add origin git@github.com:JJuOn/2019-Git-Study.git

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (master)
$ git pull origin master
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From github.com:JJuOn/2019-Git-Study
 * branch          master      -> FETCH_HEAD
 * [new branch]     master      -> origin/master

서 주 원 @DESKTOP-G8ARRDF MINGW64 /e/경 희 대 학 교 /3-1/Git 교 육 자 료 /2019-Git-Study (master)
$ |
```

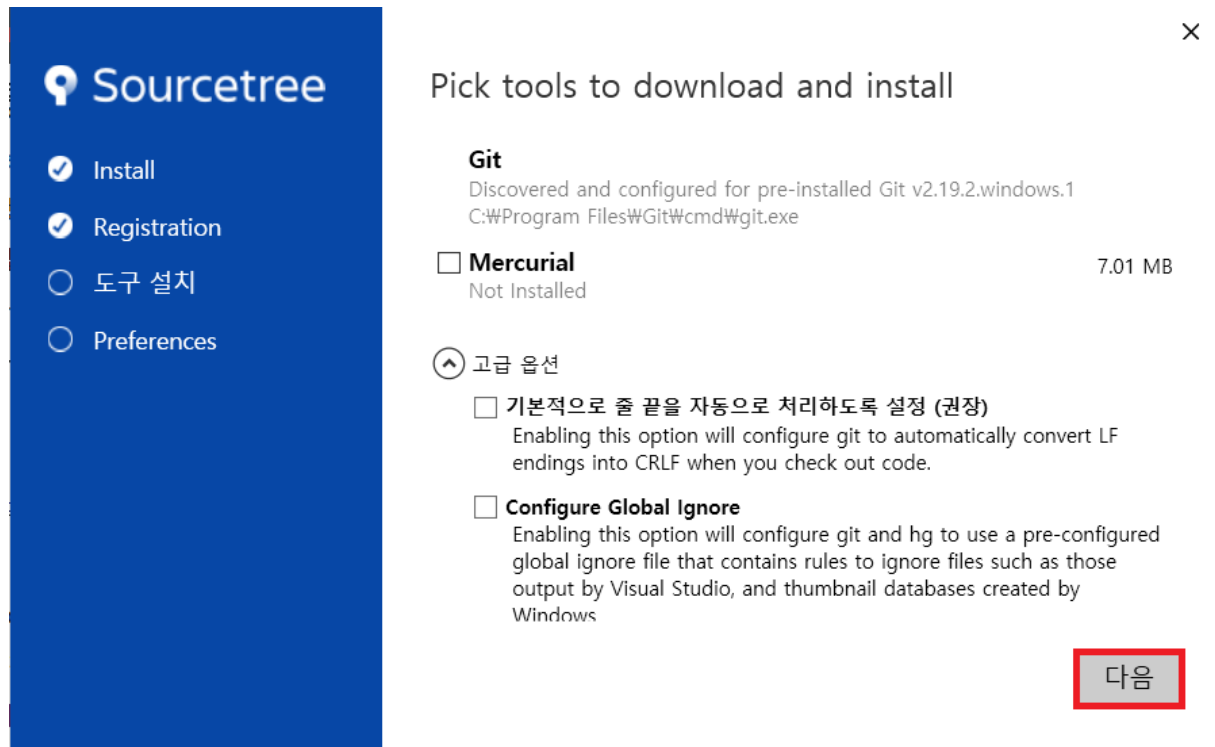
Git 브랜치 – sourcetree 설치

- 커밋 상황을 GUI로 보여주는 프로그램.
- <https://www.sourcetreeapp.com/> 에서 다운로드



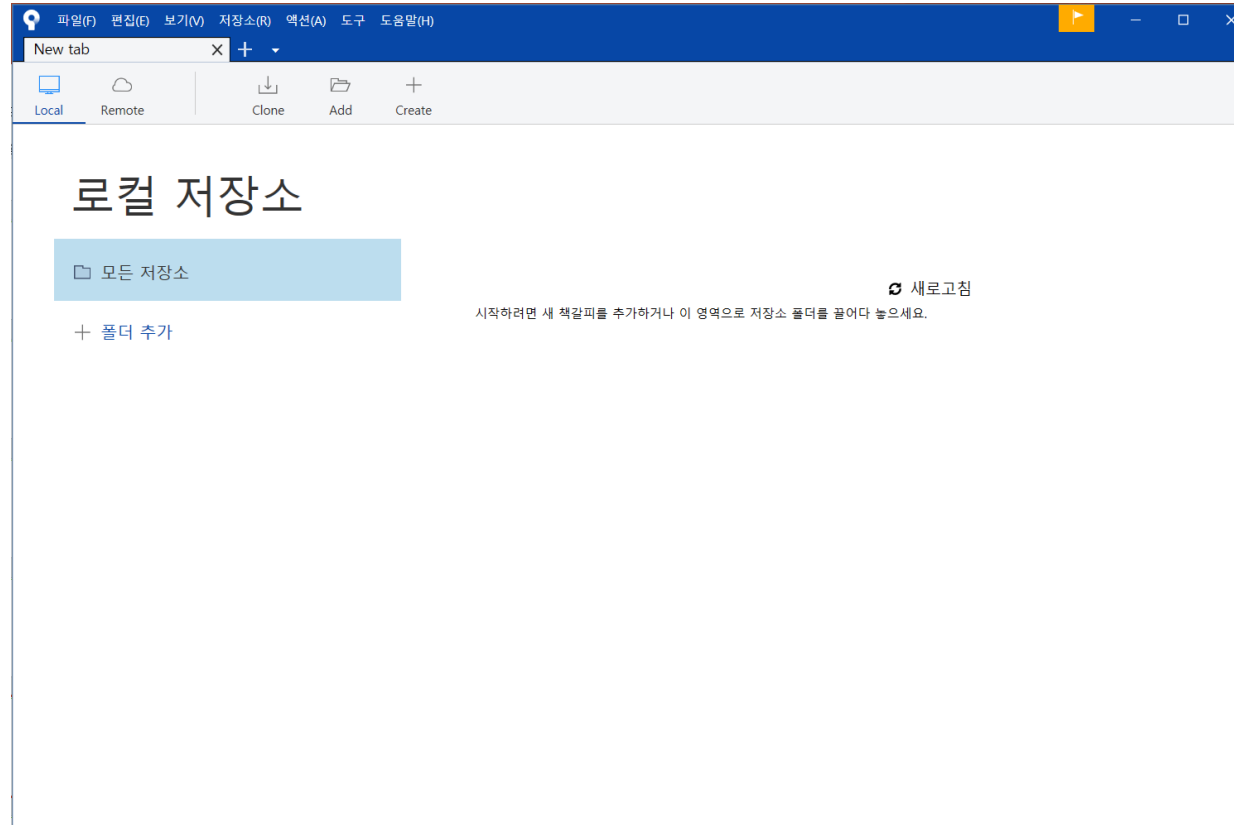
Git 브랜치 – sourcetree 설치

- 구글 계정으로 로그인

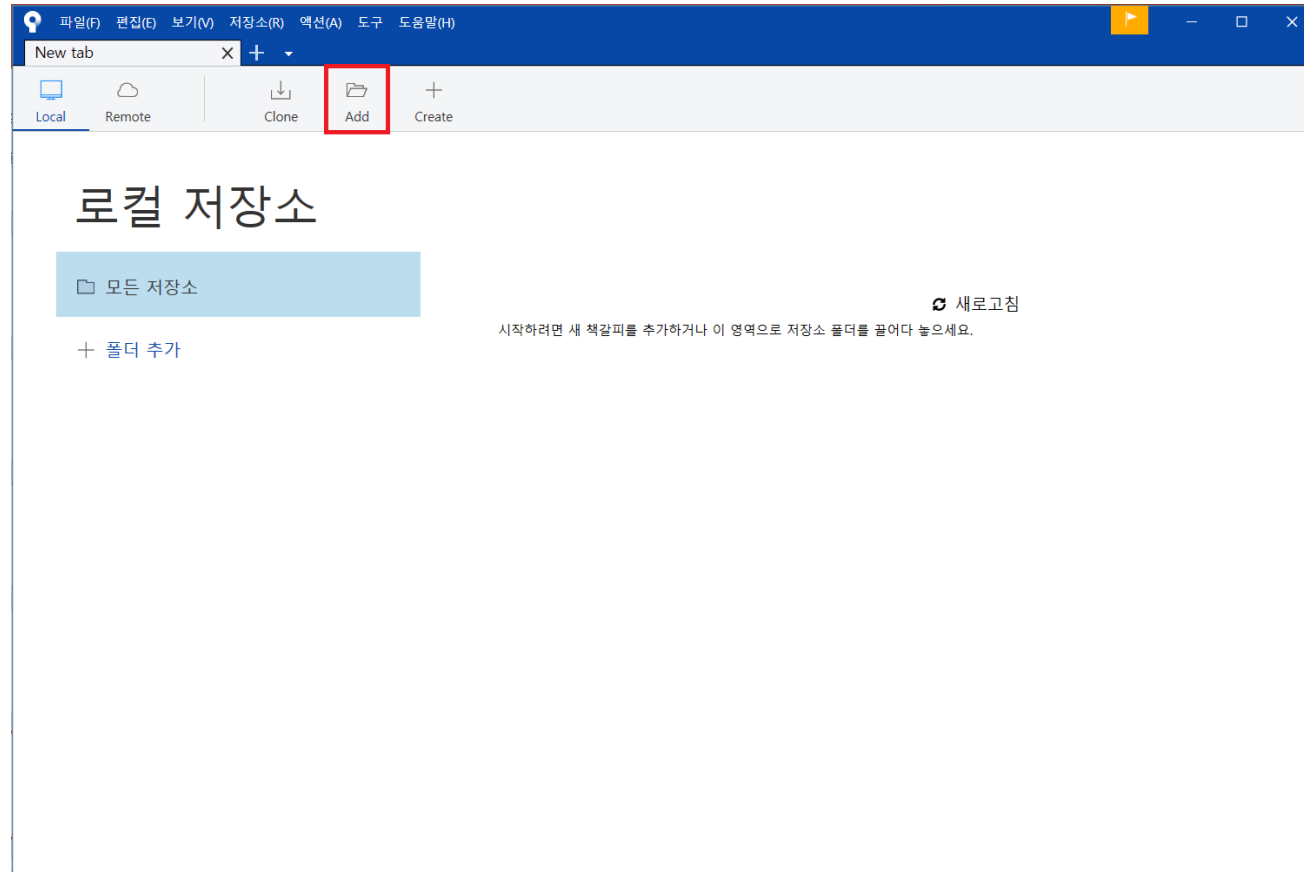


Git 브랜치 – sourcetree 설치

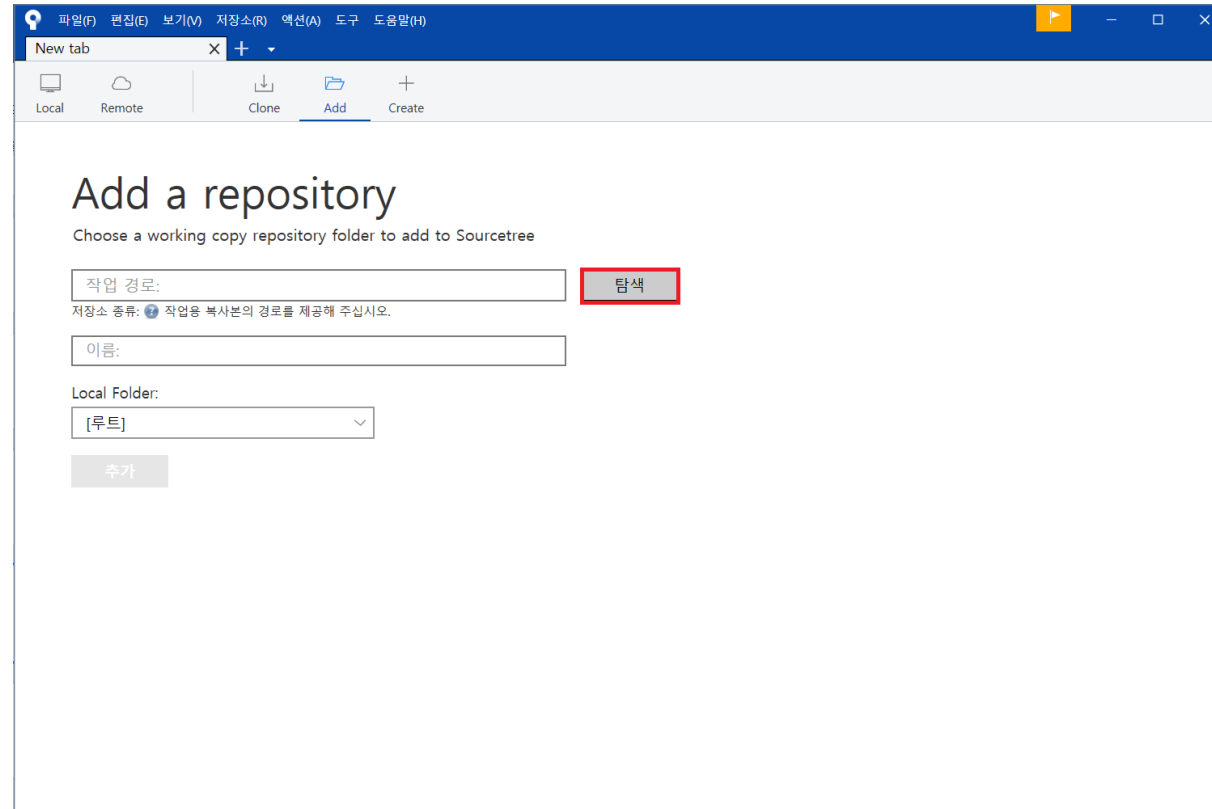
- 다음 계속 눌러주면 설치 완료.



Git 브랜치 – sourcetree 설치



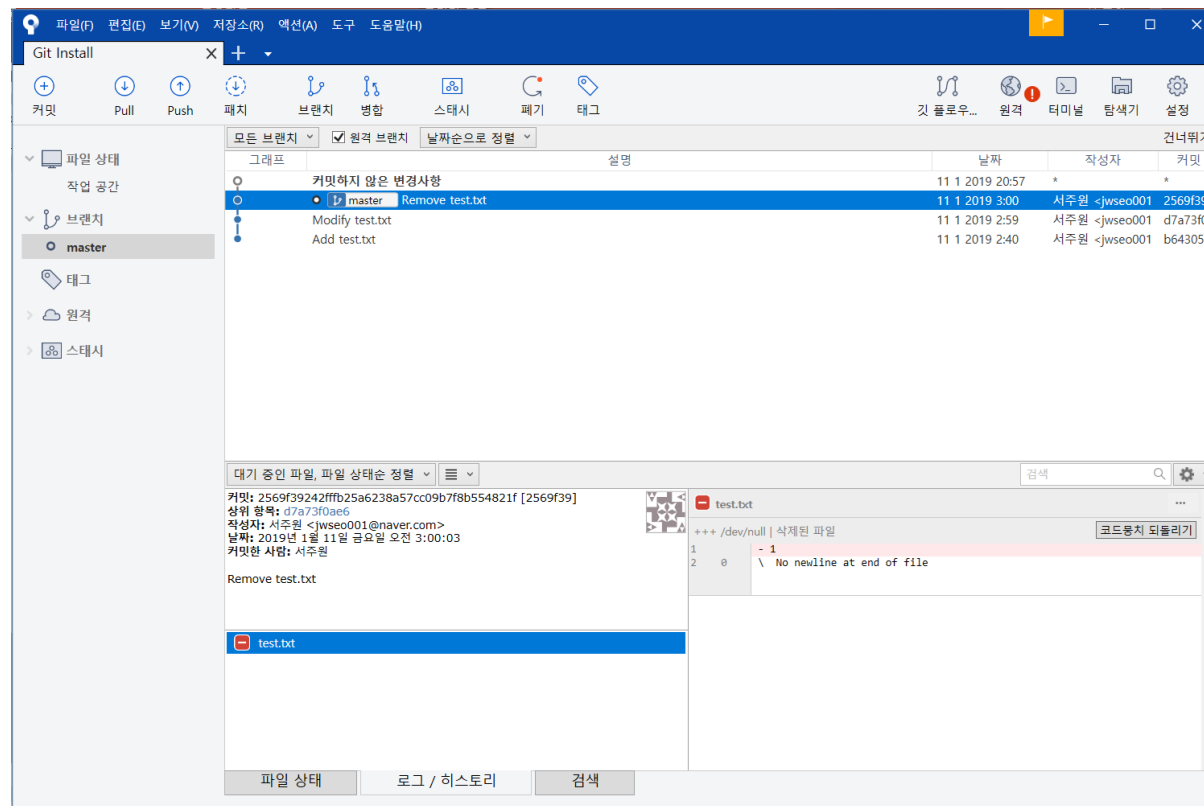
Git 브랜치 – sourcetree 설치



- 탐색 버튼을 누른 후, .git이 들어있는 폴더 선택

Git 브랜치 – sourcetree 설치

- 예시로 이전에 git 기초에서 사용했던 폴더를 선택하고 좌측의 브랜치(master)를 클릭하면 다음과 같이, 커밋했던 내용이 나옴.

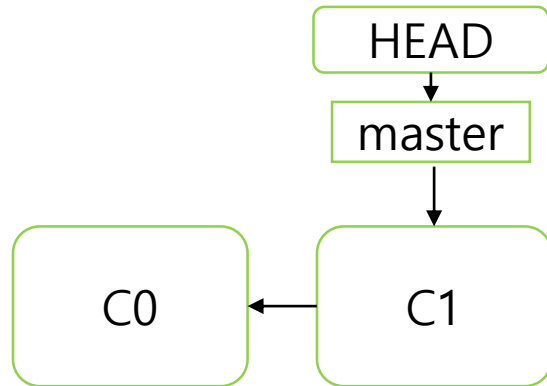


Git 브랜치 – git branch & git checkout

- 주로 다른 사람과의 협업할 때 주로 사용.
각자 다른 부분을 작업하고 있을 때, 브랜치를 판다.
- `git branch <branch name>` : 현재 위치에서 브랜치를 하나 생성함.
- `git checkout <branch name>` : 현재 작업 위치(HEAD)를 변경함.

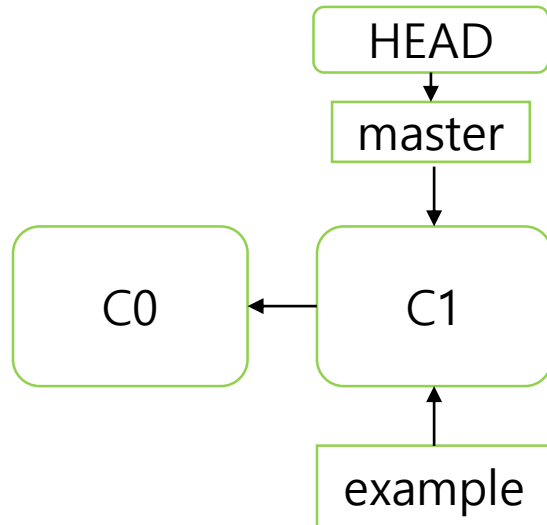
Git 브랜치 – git branch & git checkout

- 초기 상태 (C0~C1은 커밋 된 순서, 예시에서 git add와 commit message는 생략)



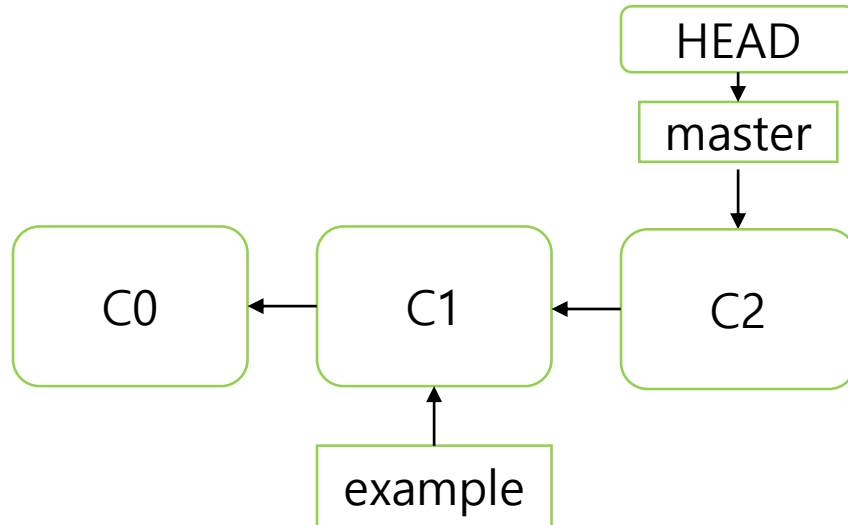
Git 브랜치 – git branch & git checkout

- git branch example



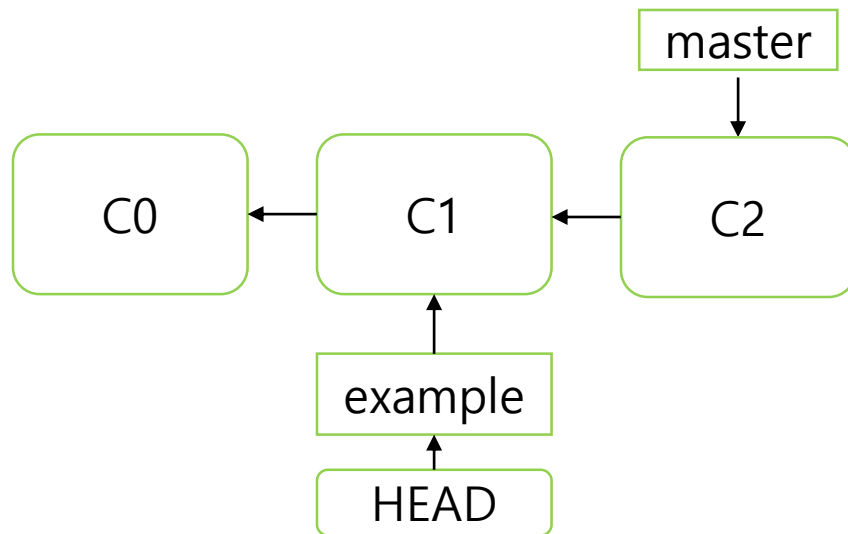
Git 브랜치 – git branch & git checkout

- git commit



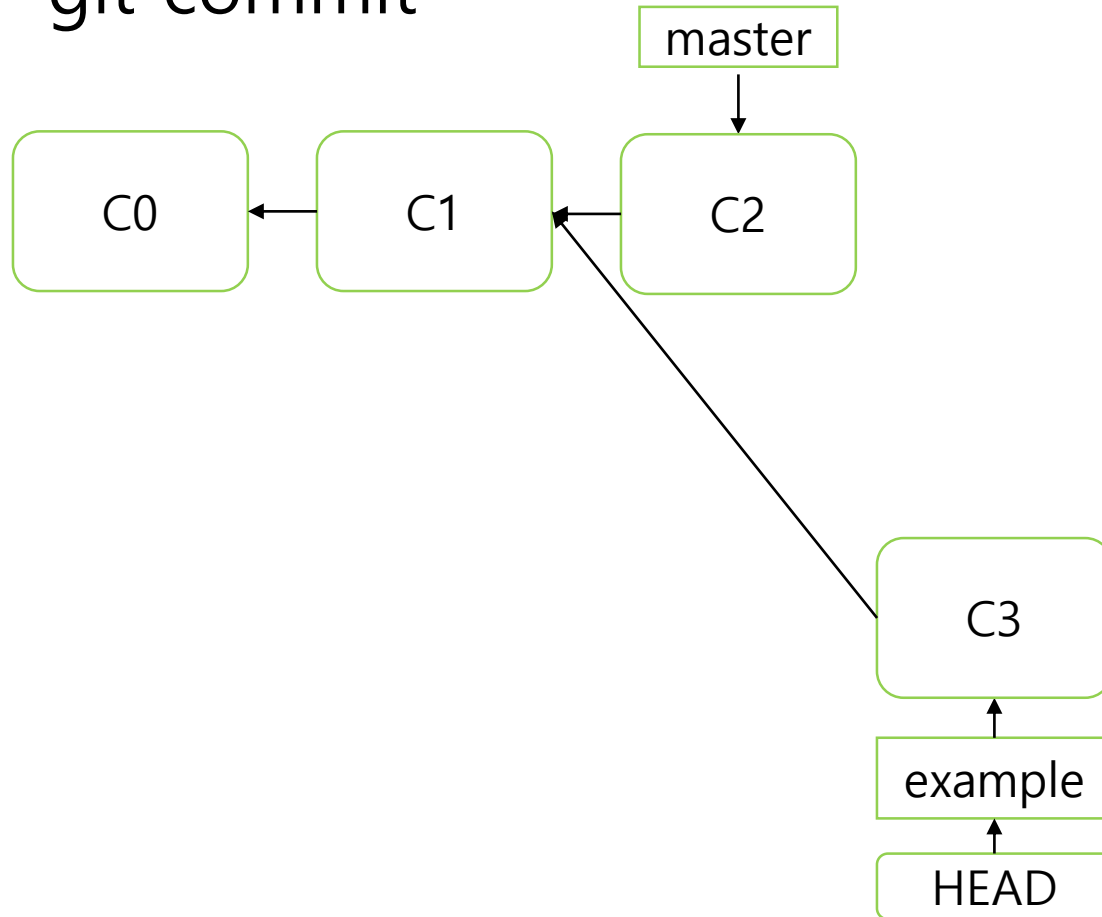
Git 브랜치 – git branch & git checkout

- git checkout example



Git 브랜치 – git branch & git checkout

- git commit



Git 브랜치 – git merge

- 각각의 브랜치에서 작업한 내용을 한데 모아야함.
- git merge <branch name> : 해당 브랜치를 HEAD에 합친다.

Git 브랜치 – git merge

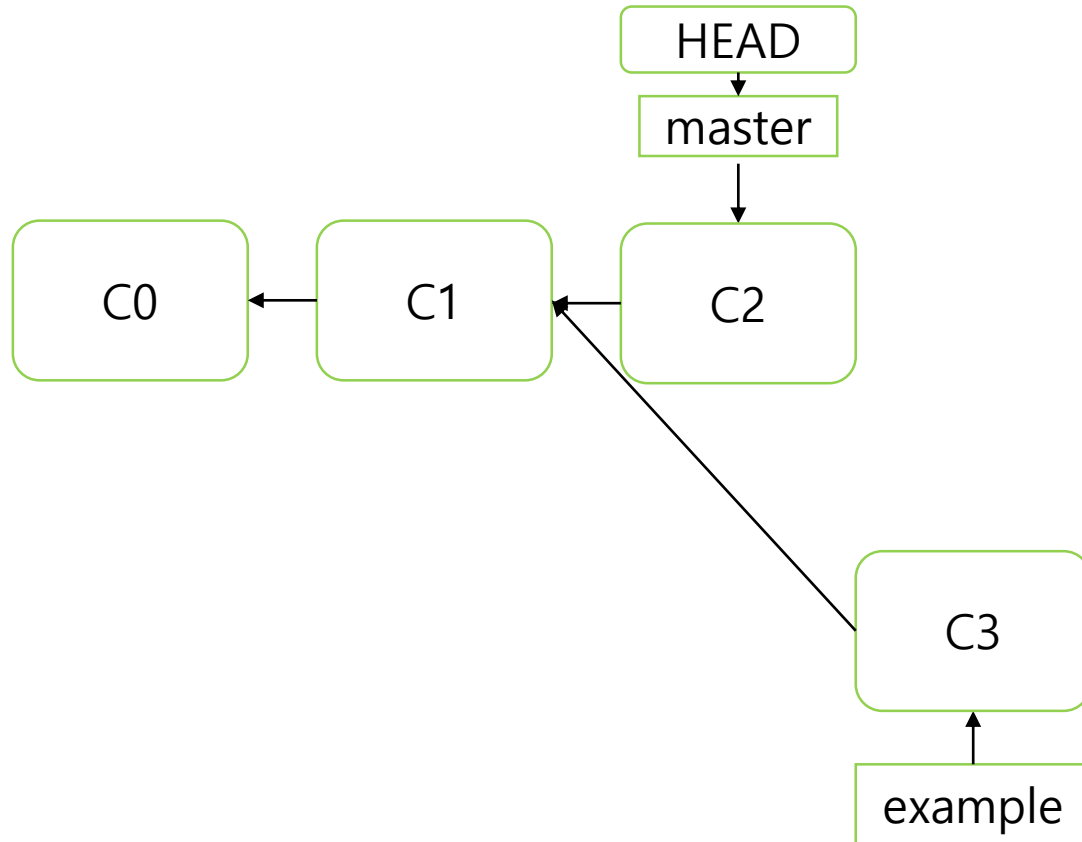
- 만약 merge도중, 두 브랜치 간에 코드가 겹치는 부분이 있어 충돌이 발생 할 수 있다. 이때는

1. 충돌한 파일 수정
2. git add <충돌했던 파일명>
3. git commit -m <commit message>

를 해주면 된다.

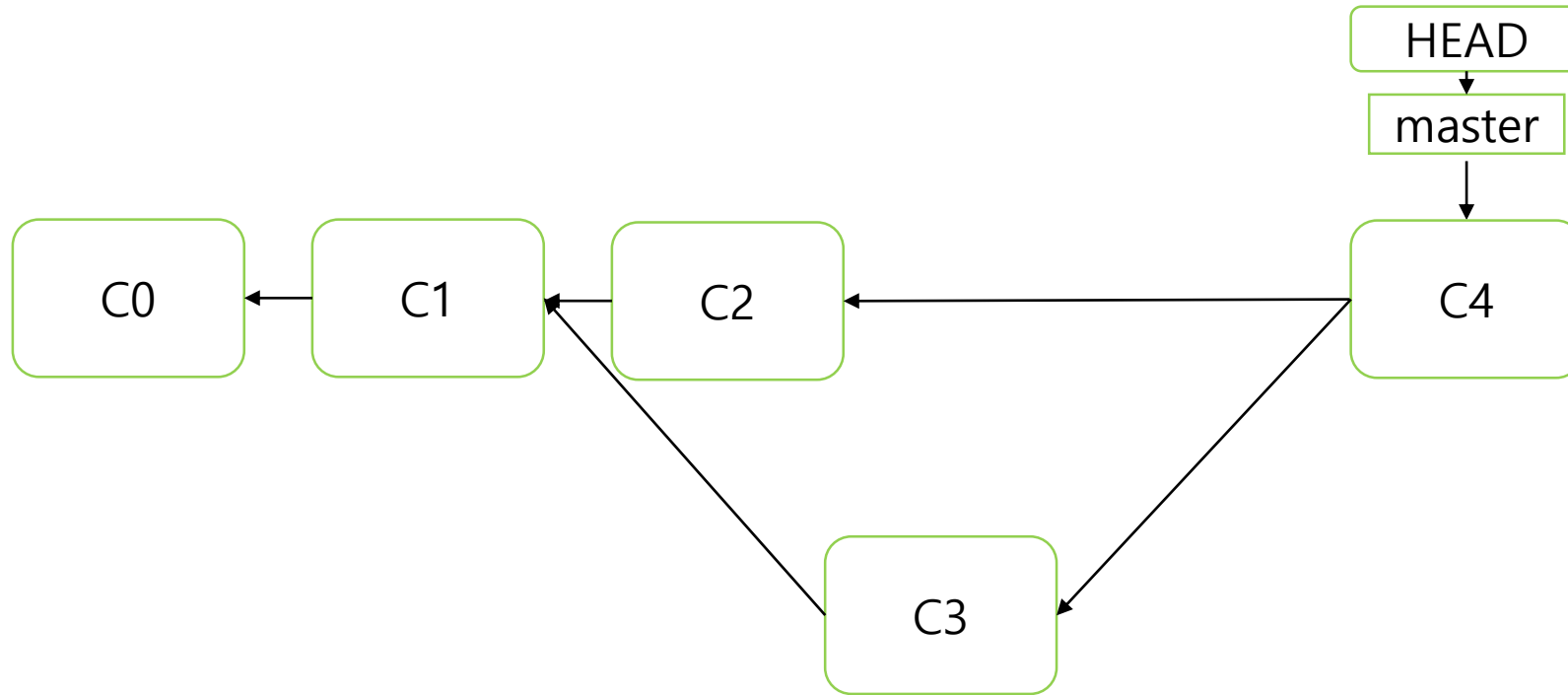
Git 브랜치 – git merge

- git checkout master



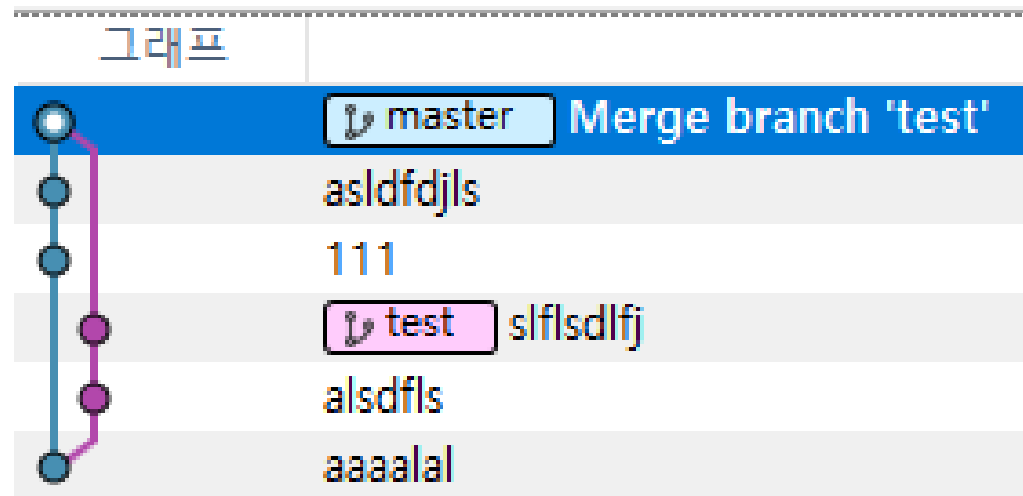
Git 브랜치 – git merge

- git merge example



Git 실습 7 - branching

- git 명령어들을 이용하여 해당 그림처럼 구성. Commit 메시지는 중요하지 않으나, commit간의 순서는 중요.



- 아직 reset을 배우지 않은 시점이니, 잘못하면 지우고 다시시작

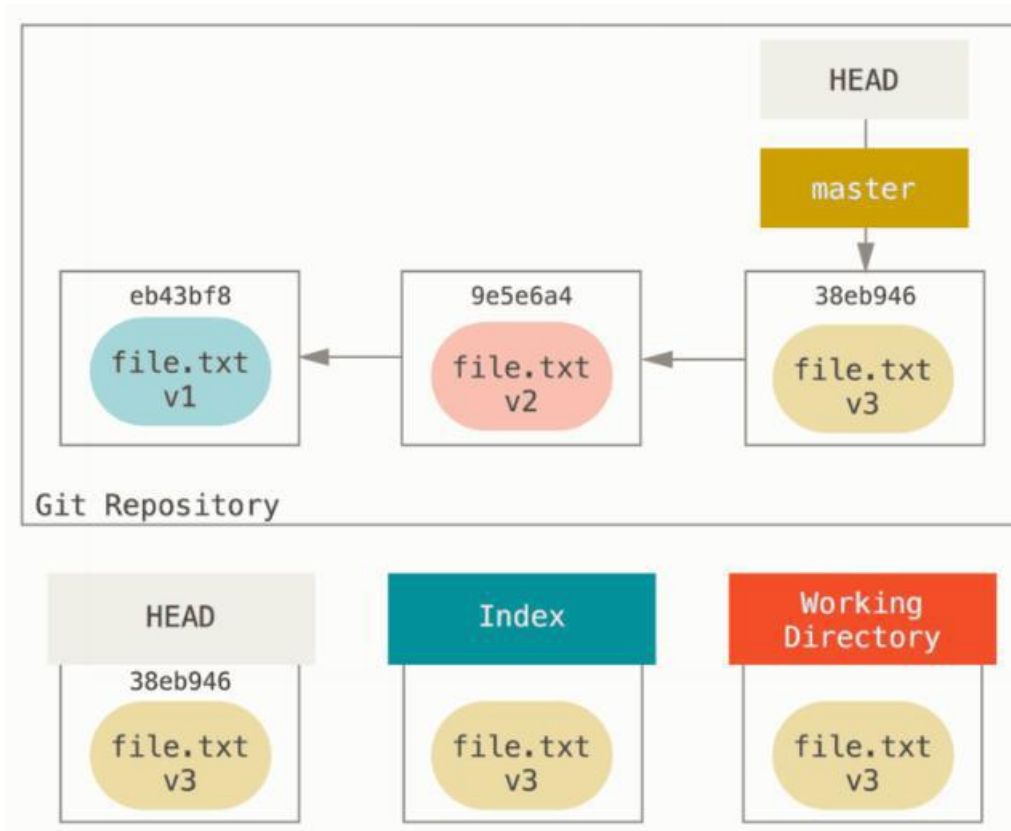
Git 고급

- 오픈소스SW개발 수업에서는 git advanced에서 reset, tag, stash, rebase, cherrypick 등을 다루지만

이 강의자료에서는 reset만 다룰 예정.

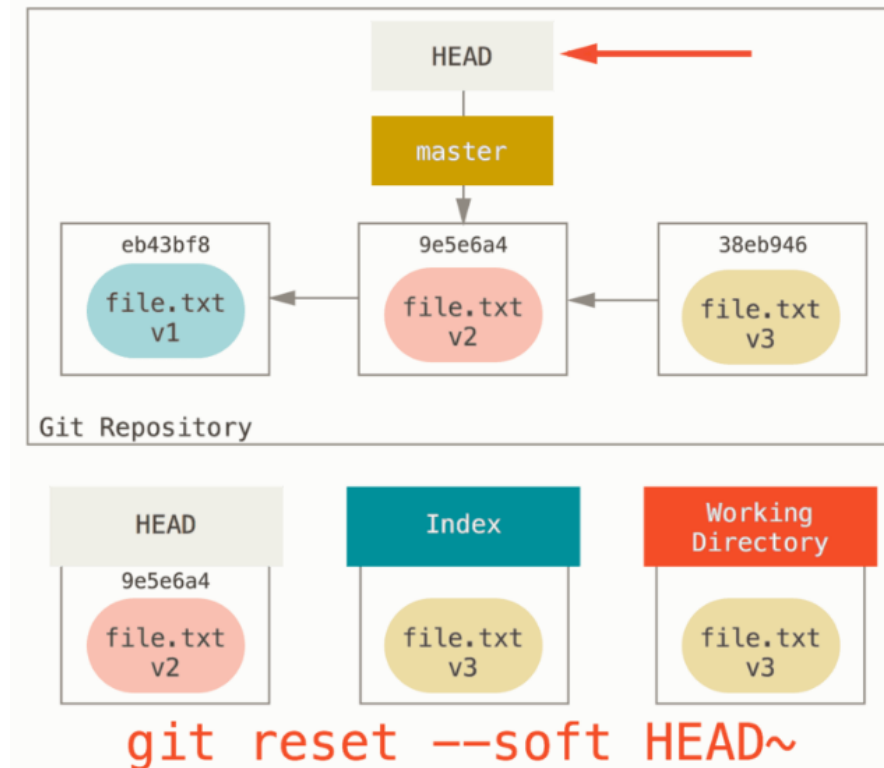
Git 고급 – git reset

- 초기 상태(index=staging area)



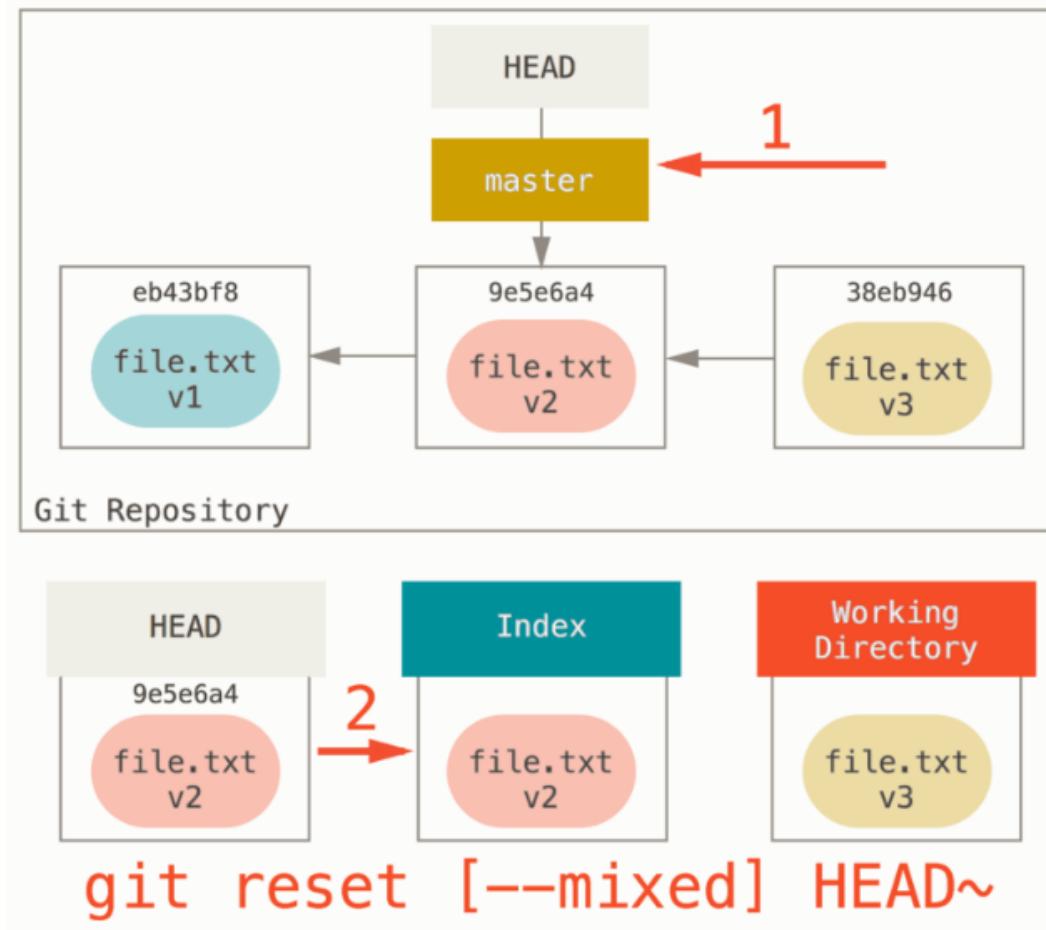
Git 고급 – git reset

- `git reset --soft HEAD~n` : HEAD포함 n개 소프트 리셋.
git commit과 반대 역할
n이 없다면 1로 간주



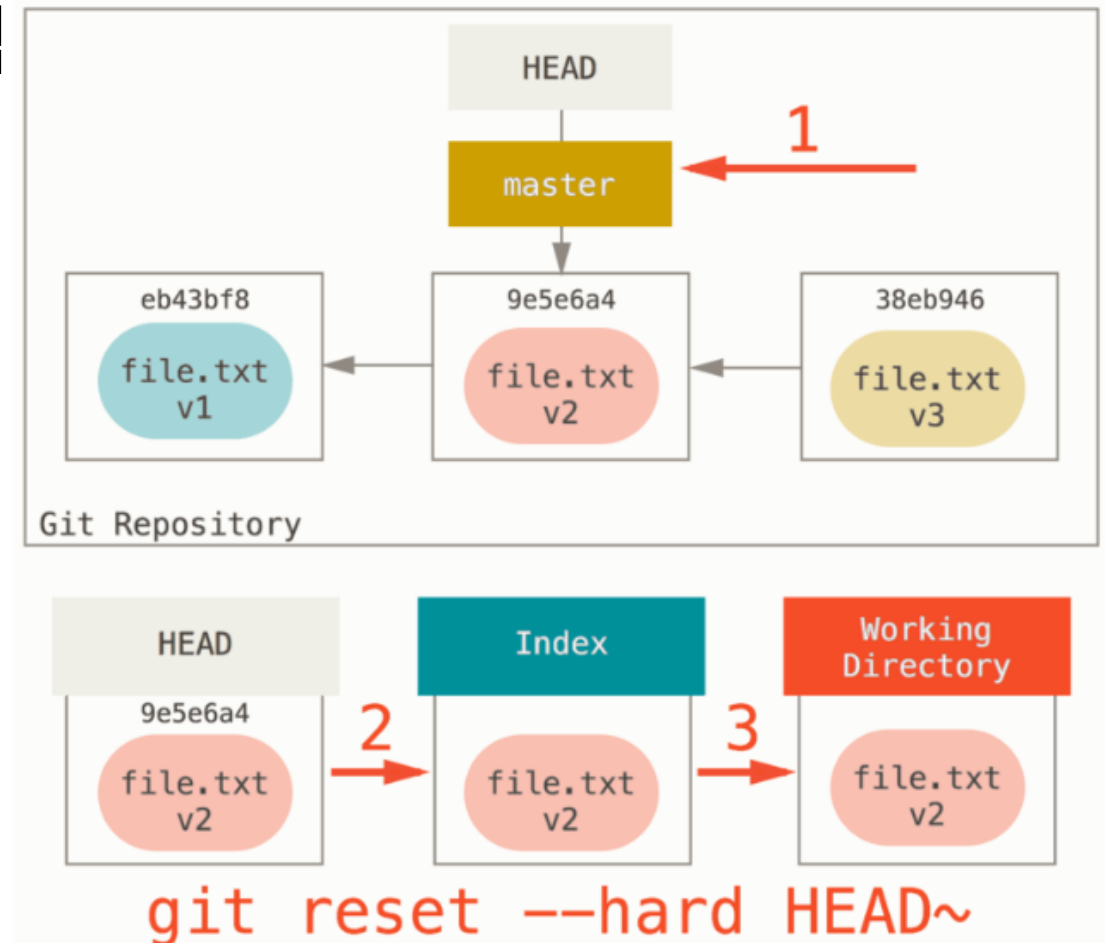
Git 고급 – git reset

- git reset (--mixed) HEAD~n : HEAD포함 n개 리셋.
git add과 반대 역할



Git 고급 – git reset

- `git reset --hard HEAD~n` : HEAD포함 n개 리셋.
파일을 수정하기 전으로 되돌림



Git 실습 8 – git reset

1. Local repository를 하나 만든다.
2. 3개의 커밋(C0, C1, C2)을 생성한 다음
3. C1의 상태로 돌아간다. (파일도 C1의 상태로)

GitHub - fork

- Repository의 owner나 contributor가 아니면, 해당 repository를 수정할 수 없음. => fork
- Fork를 통해 다른 사람의 프로젝트를 복사해 올 수 있음.
- 원본 repository에 적용하기 전, 여러 아이디어를 검토하고 점검하는 목적으로도 사용됨.

GitHub – pull request

- Fork된 repository를 수정한 후, 원본 repository에 반영하기 위해 pull request 사용.
- Pull request 하는 도중, 소스코드에 대한 검토가 이루어짐.

Git 실습 9 - GitHub

1. 2인이 짝을 이룬다.
2. 각자 GitHub에서 새 repository를 생성하고, 텍스트 파일 하나를 commit한 다음 push도 한다.
3. 서로의 repository를 fork한 후, 텍스트 파일을 수정한 다음 commit, push한다.
4. 서로의 원본 repository로 pull request를 하고, 확인 후 confirm merge한다.