



# Round 7

**PRESS  
START**



## << Round 7 >>

- Git 개요
- GitHub
- Git 실습



New  
Assignment



## << Round 7 >>

- Git 개요 <<
- GitHub
- Git 실습



Let's  
Go



# Git?



- 분산형 버전 관리 시스템
- 소스 코드 관리에 주로 사용

## Why use it?

- 여러 사람이 동일한 코드에 대해 동시에 작업을 할 때
  - 상대방의 작업을 방해하지 않으면서
    - 변경 이력을 남기면서
      - 효율적으로

프로그래밍이 가능해진다.

# Why use it?

## Project

# 쇼핑몰 웹 사이트 제작

...

# 각자 할 일

나 + 팀원1 : 로그인, 메인페이지 로직

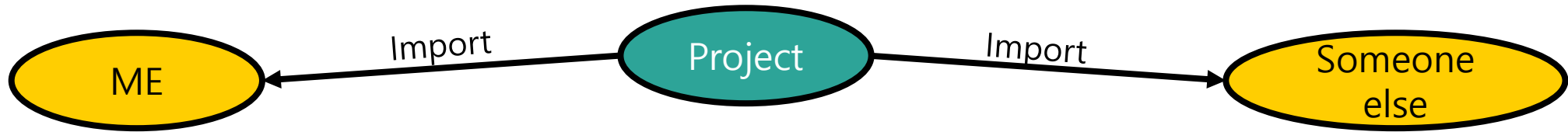
팀원2 + 팀원3 : 로그인 관련 DB

팀원1 + 팀원3 : 로그인, 메인페이지 프론트엔드

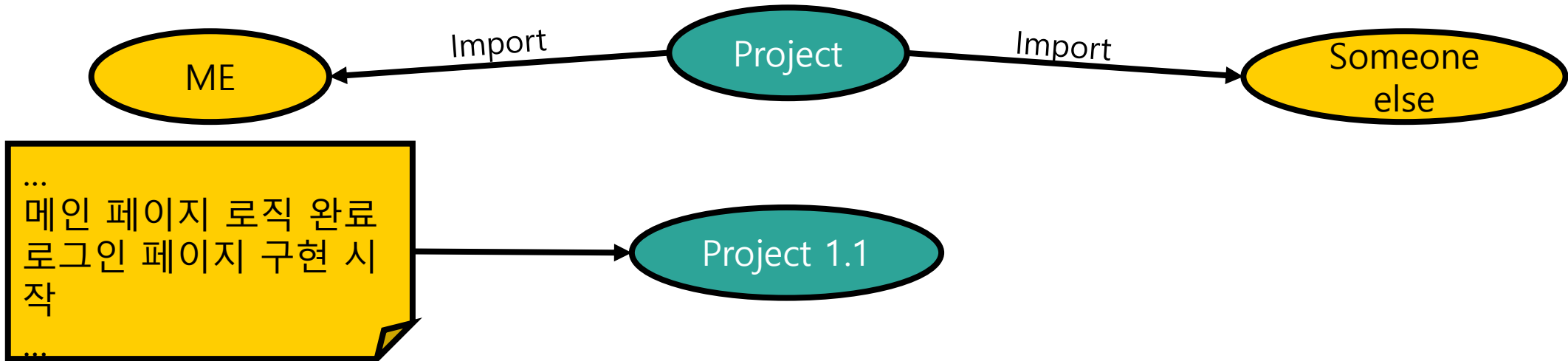
...

# 중간 검토 기간 : 이틀 뒤...

# Why use it?

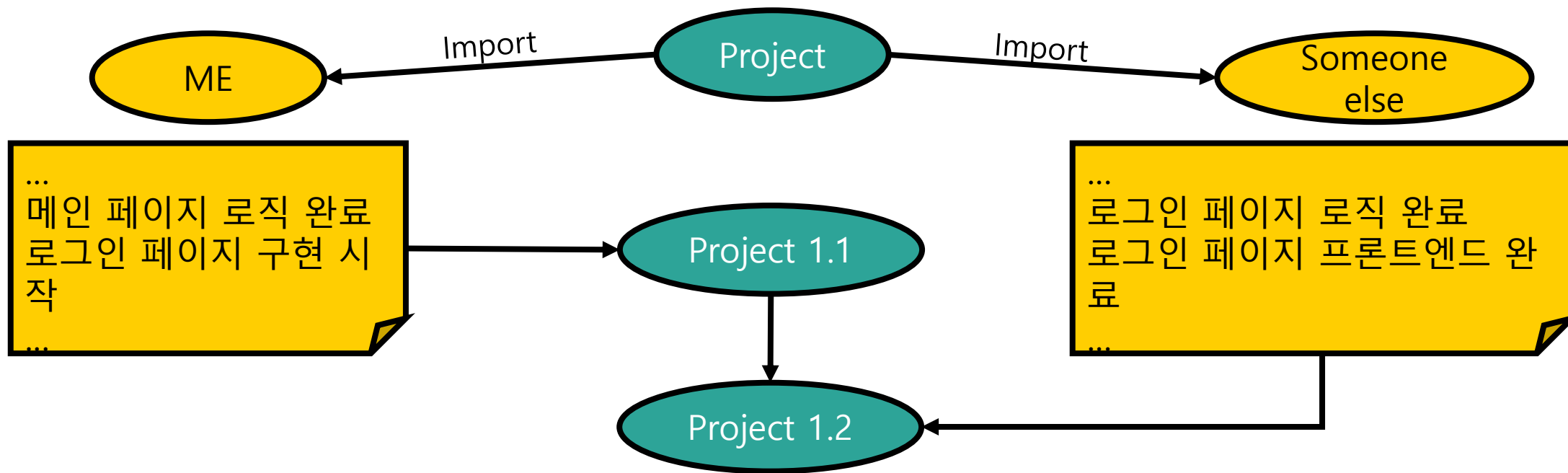


# Why use it?

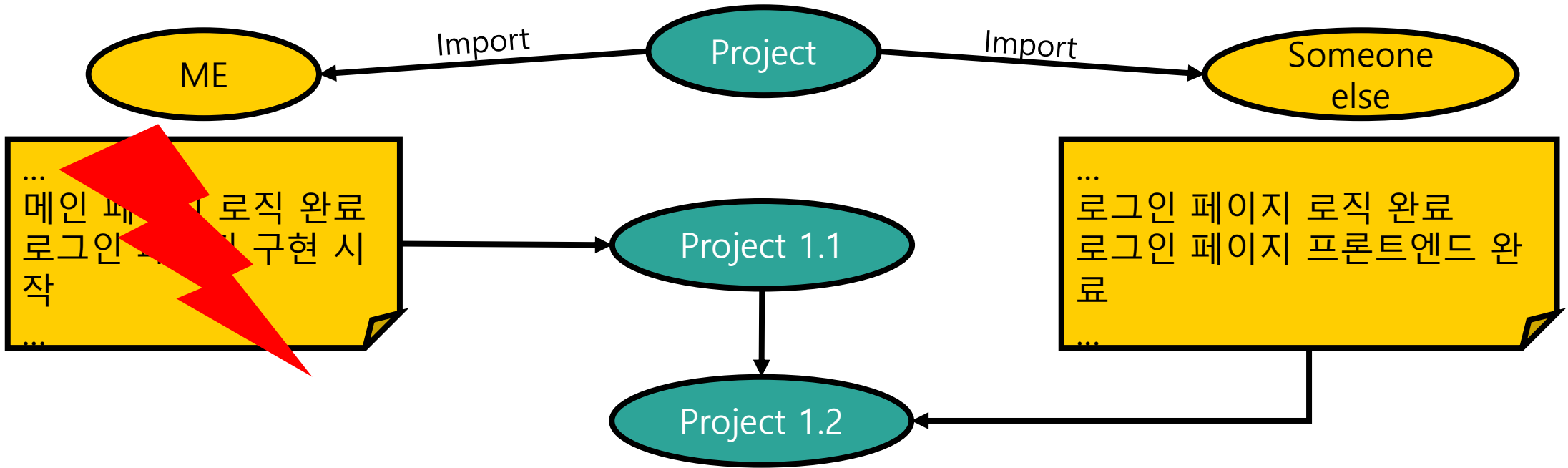




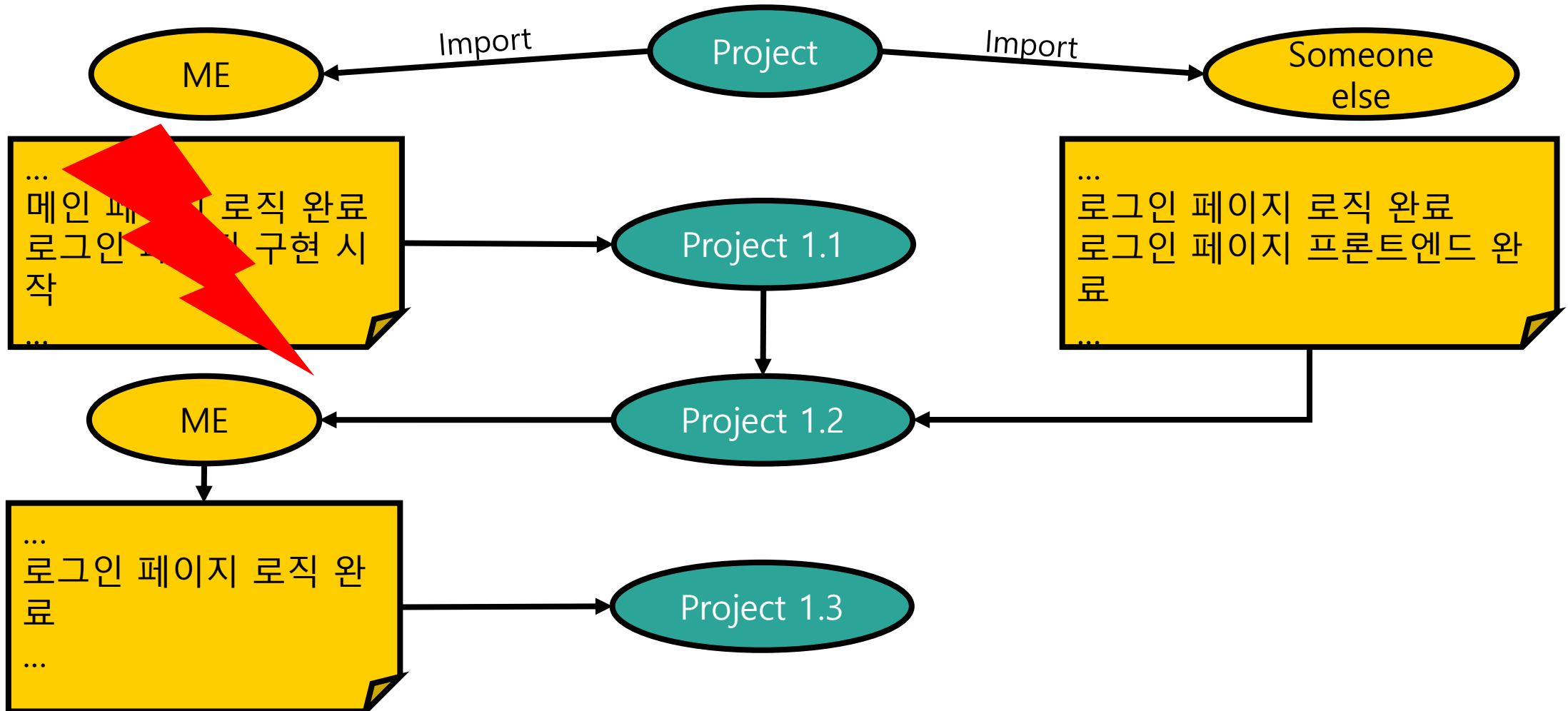
# Why use it?



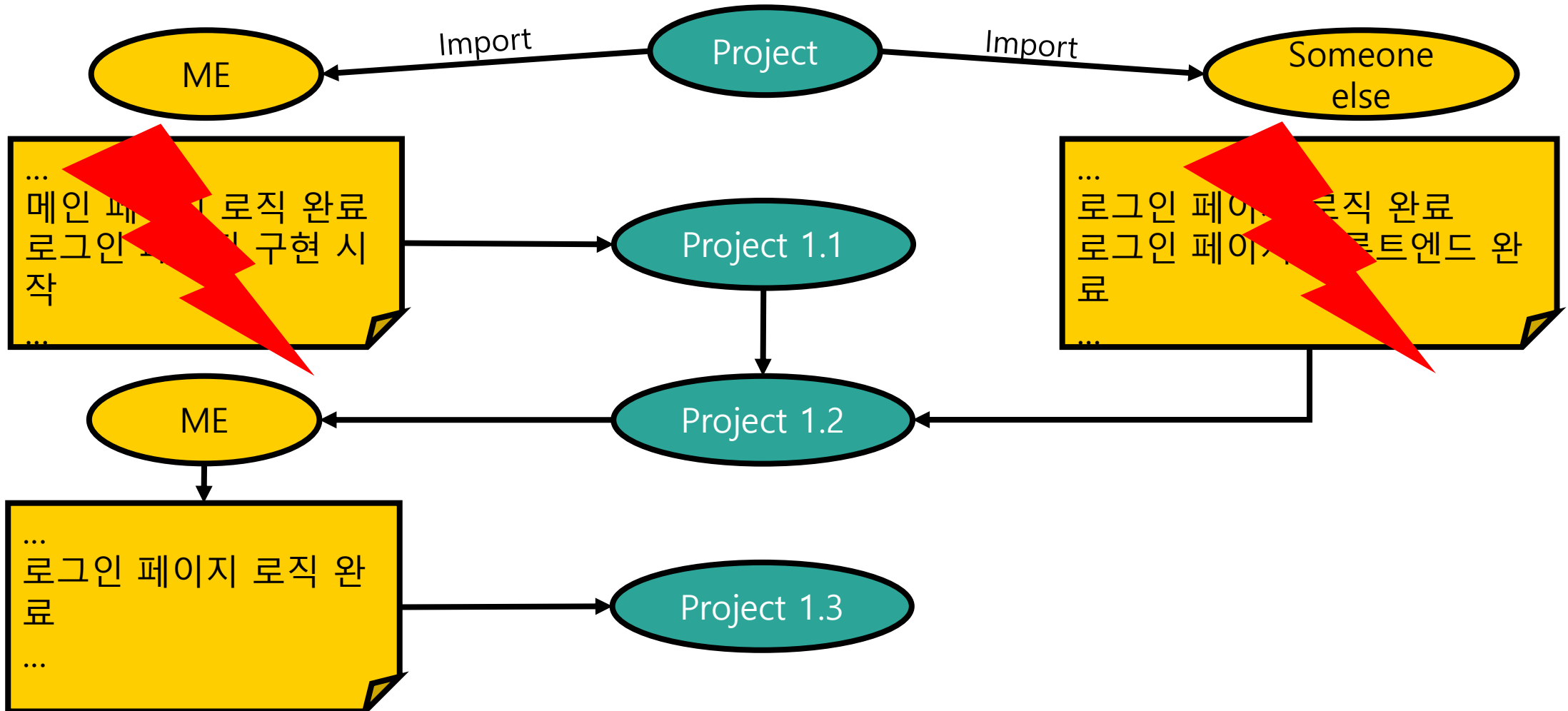
# Why use it?



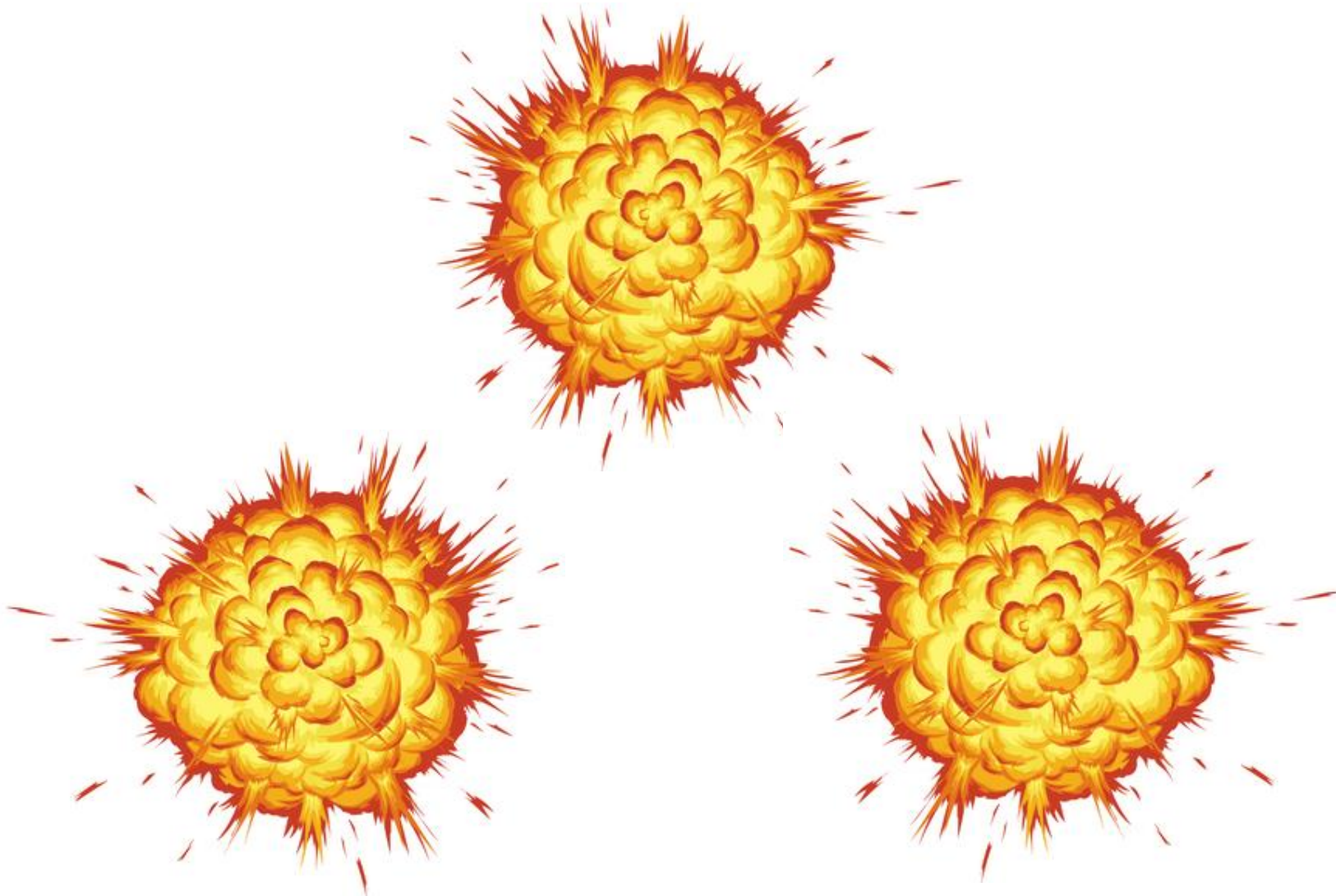
# Why use it?



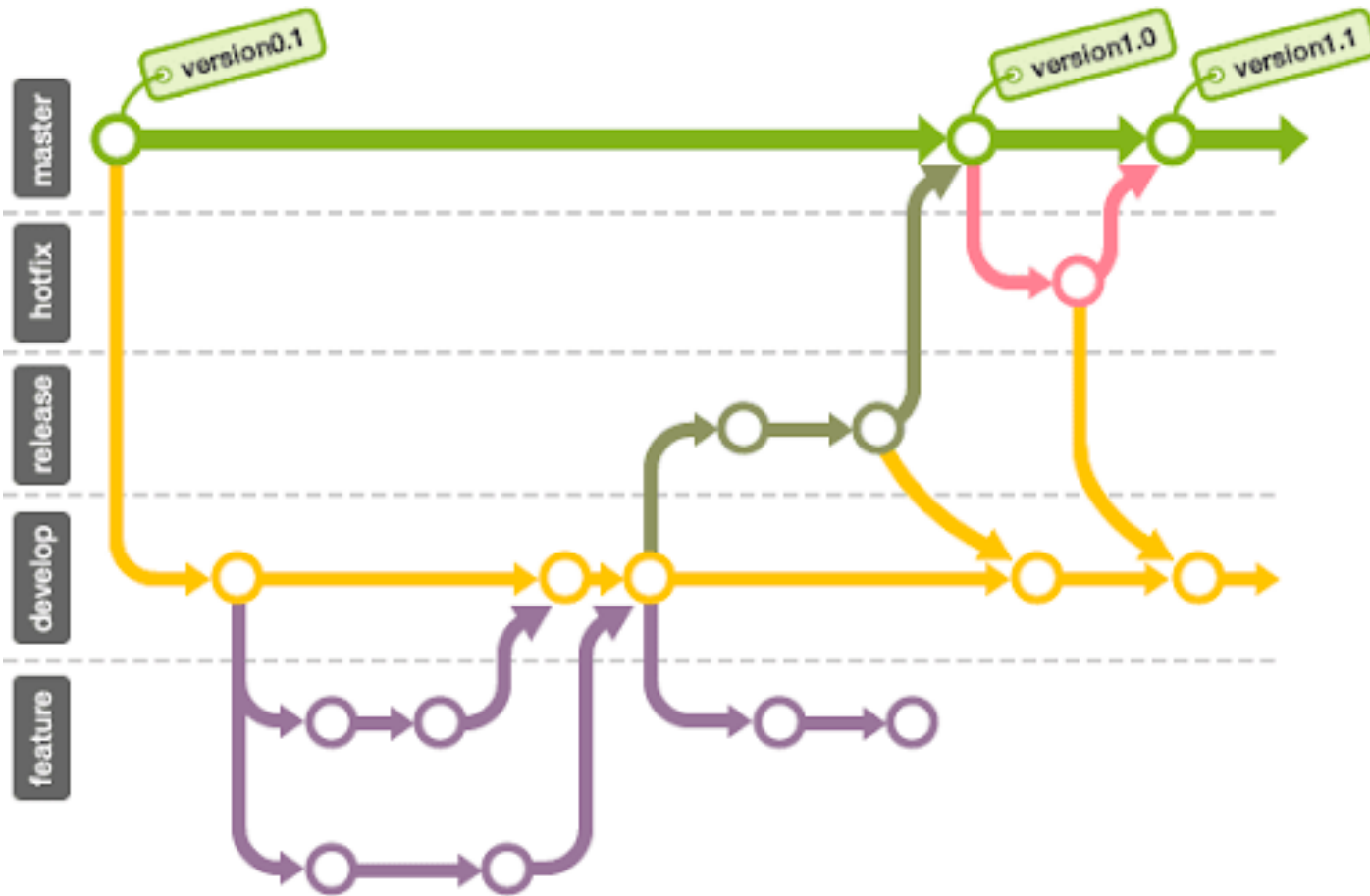
# Why use it?



# Why use it?




# Why use it?





# Install git

<https://git-scm.com/downloads>

## Downloads

 **Mac OS X**

 **Windows**

 **Linux/Unix**

Older releases are available and the [Git source repository](#) is on GitHub.

### GUI Clients

Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

[View GUI Clients →](#)

### Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.


[View Logos →](#)

### Git via Git

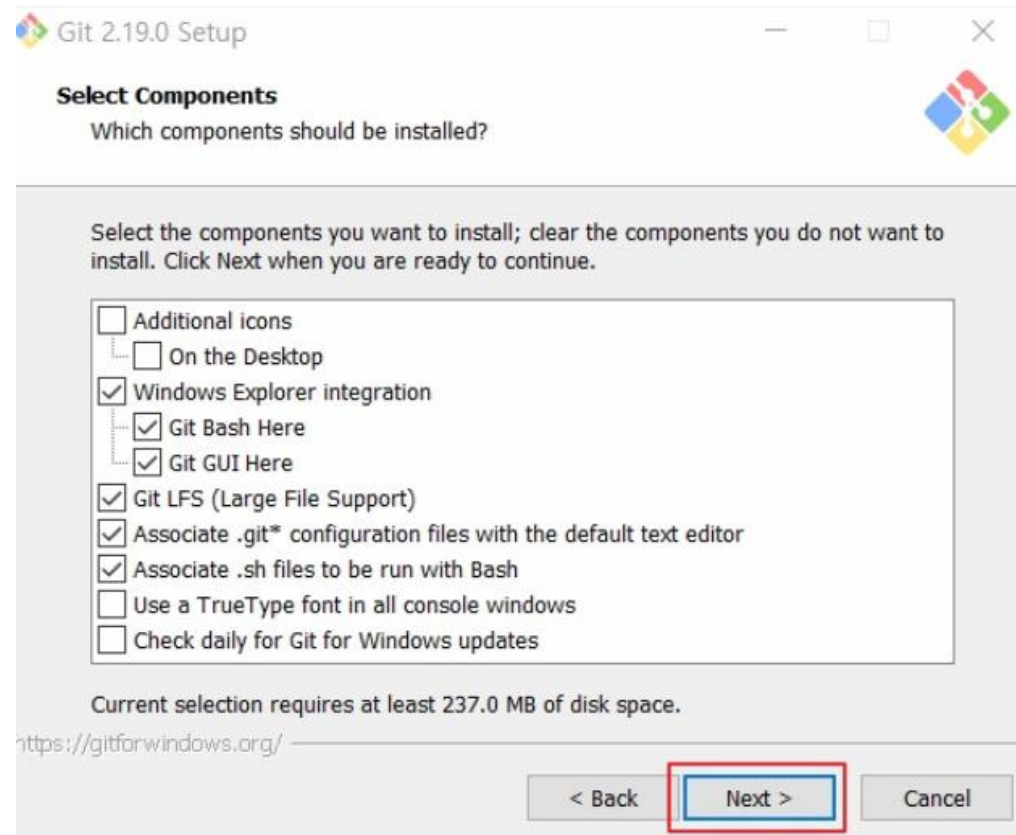
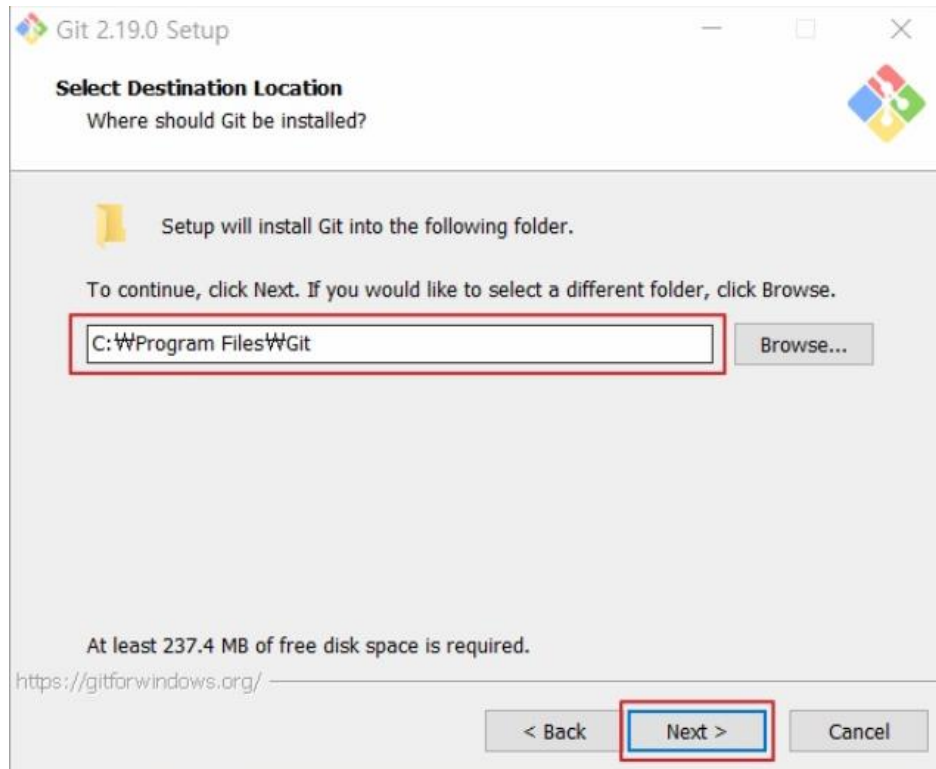
If you already have Git installed, you can get the latest development version via Git itself:

```
git clone https://github.com/git/git
```

You can also always browse the current contents of the git repository using the [web interface](#).

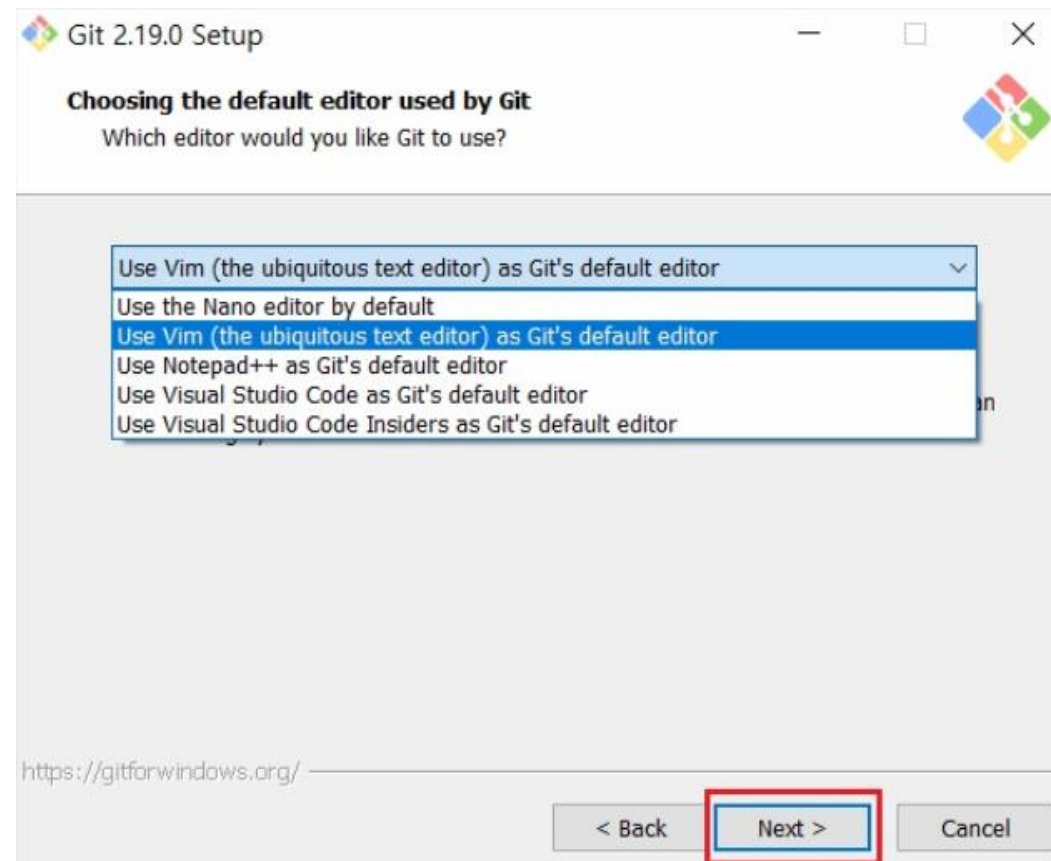
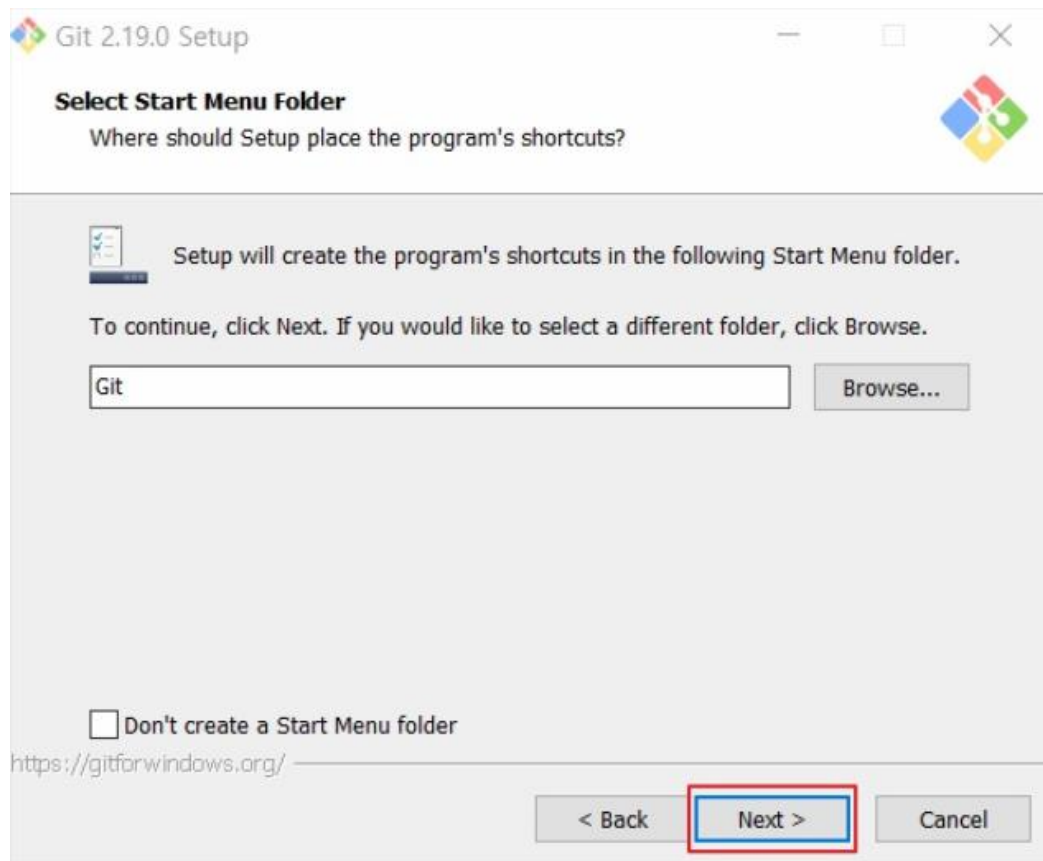


# Install git

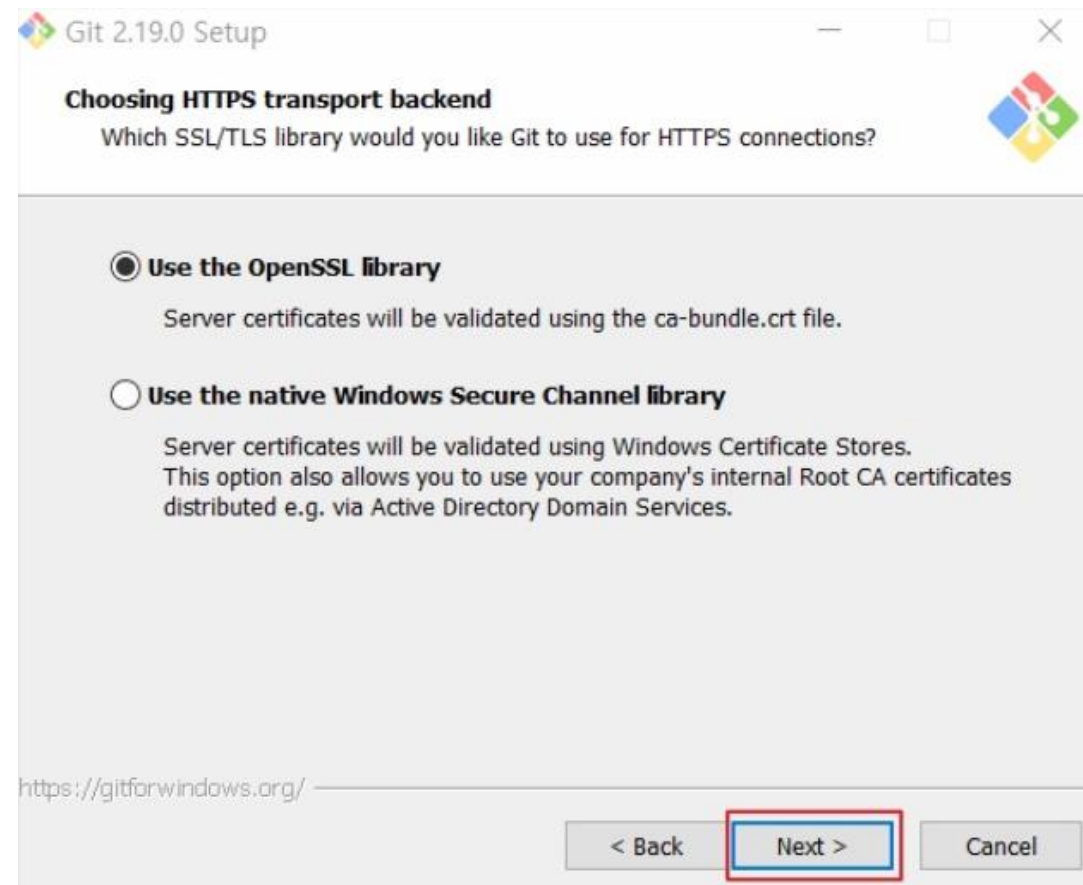
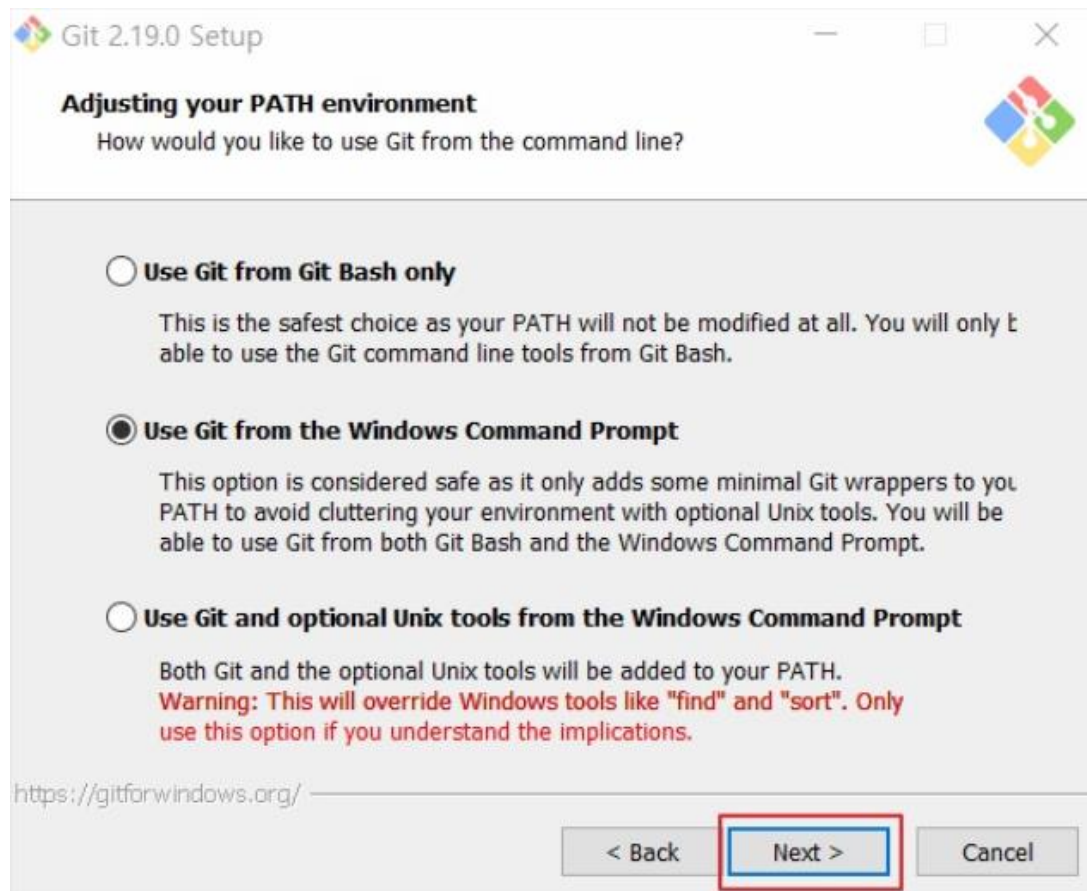




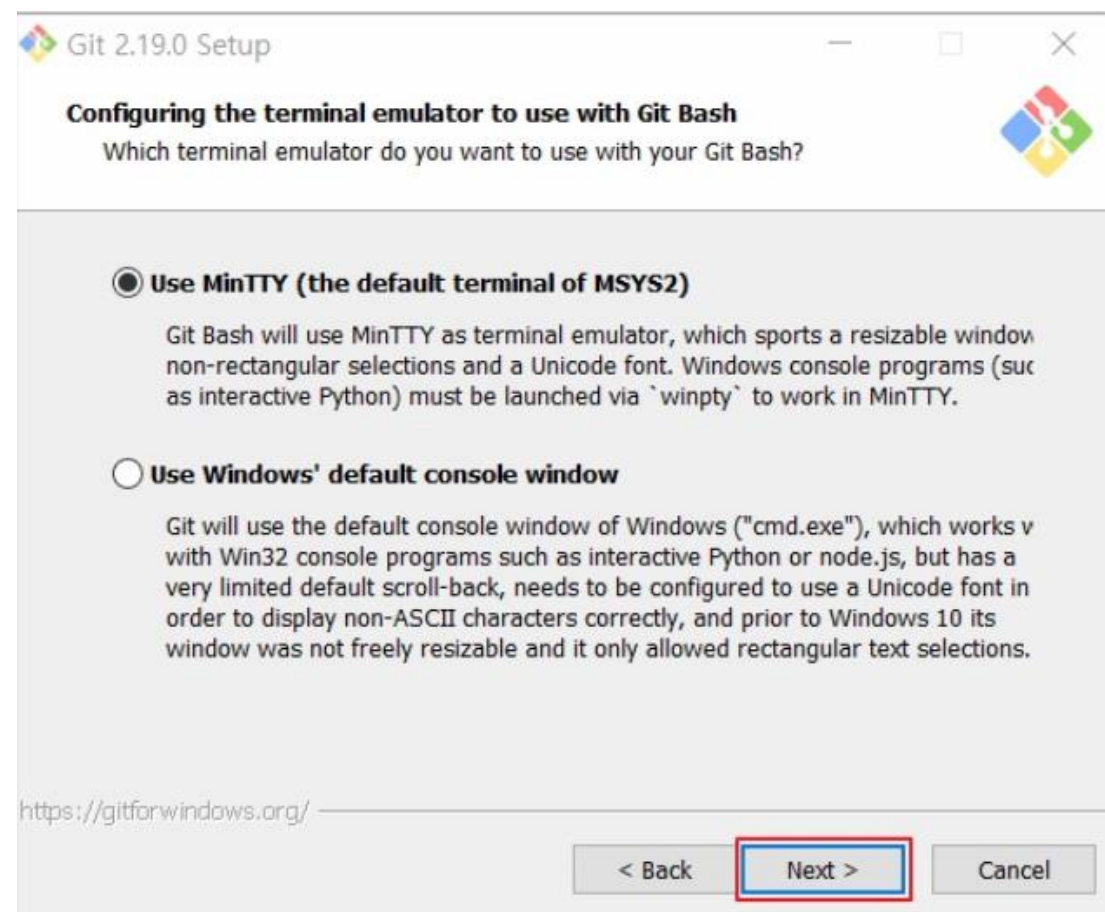
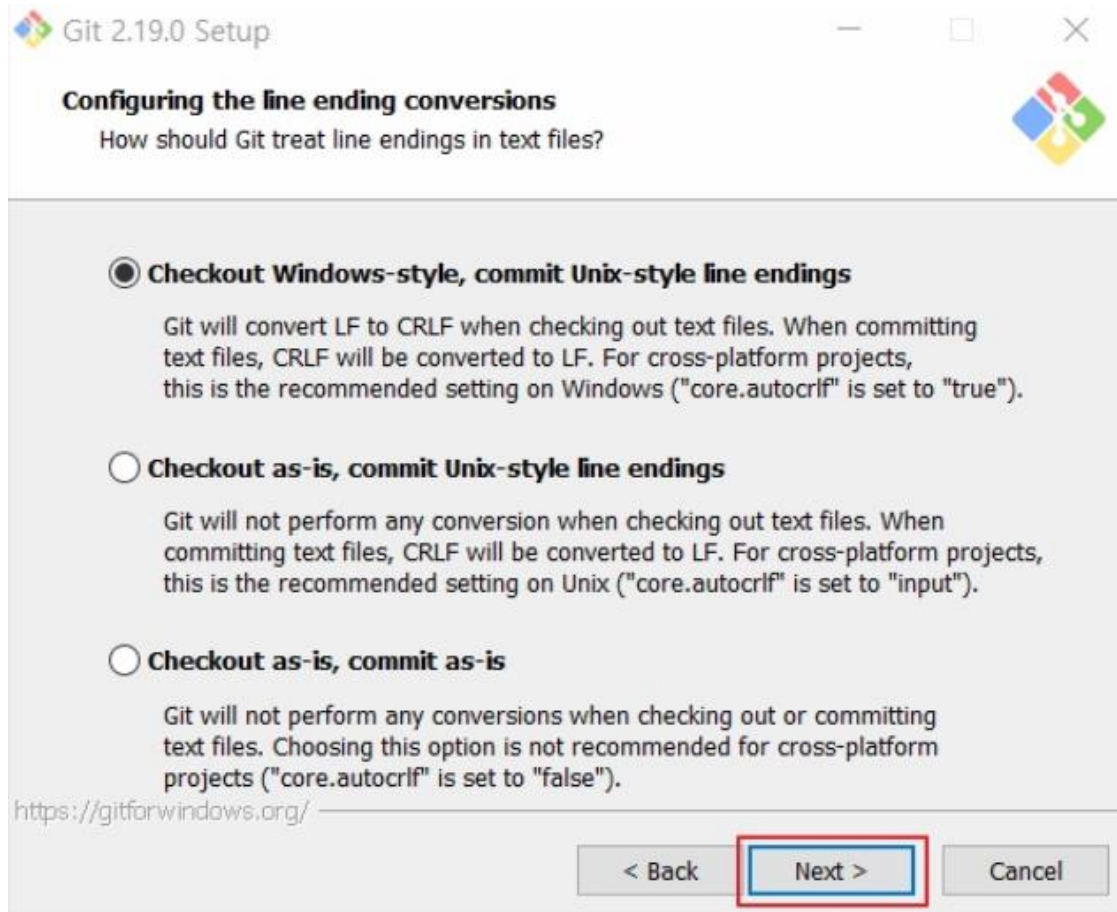
# Install git



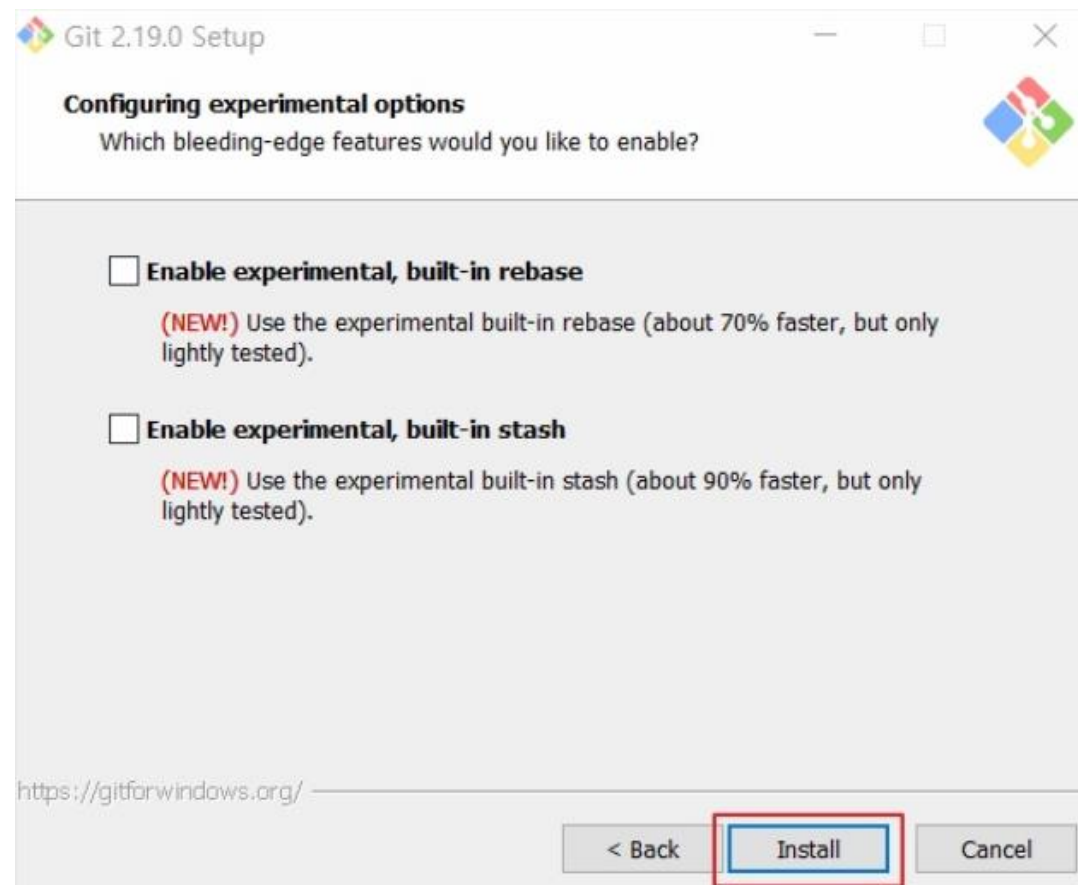
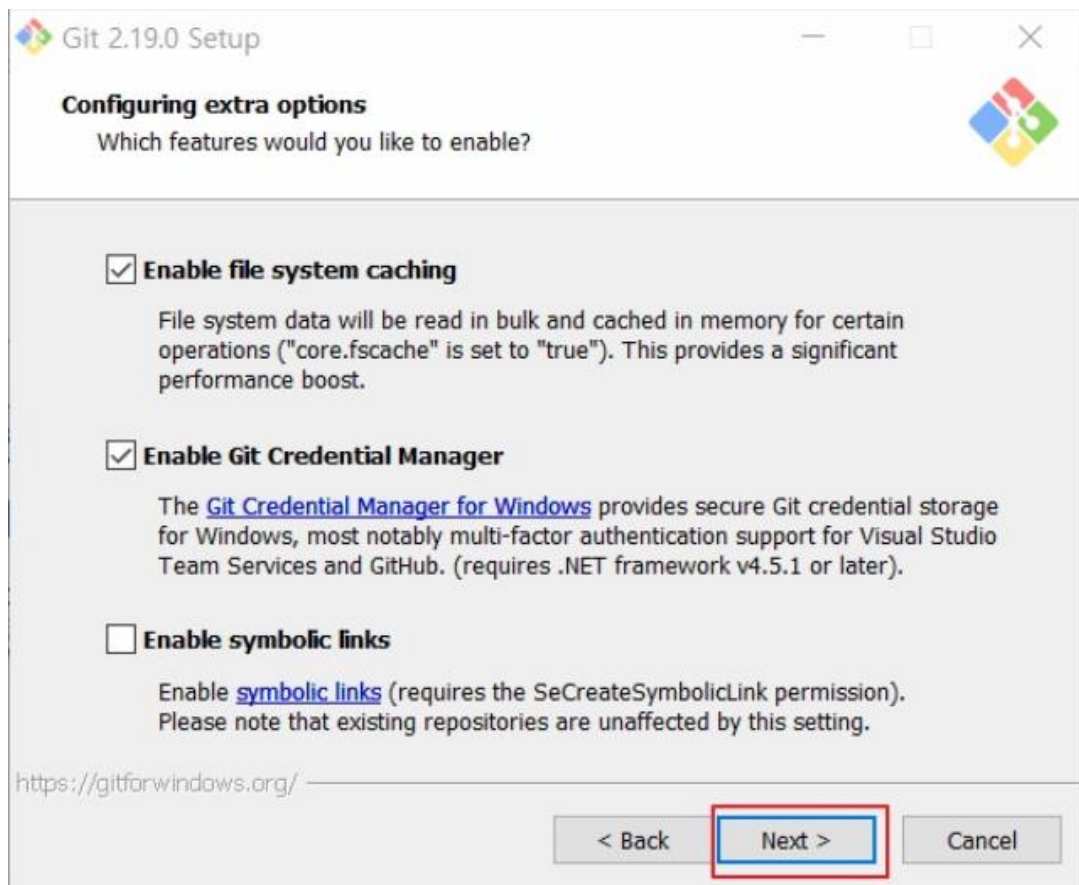
# Install git



# Install git



# Install git





# Git status

Working  
Directory

편집된 파일이 저장되어 있음

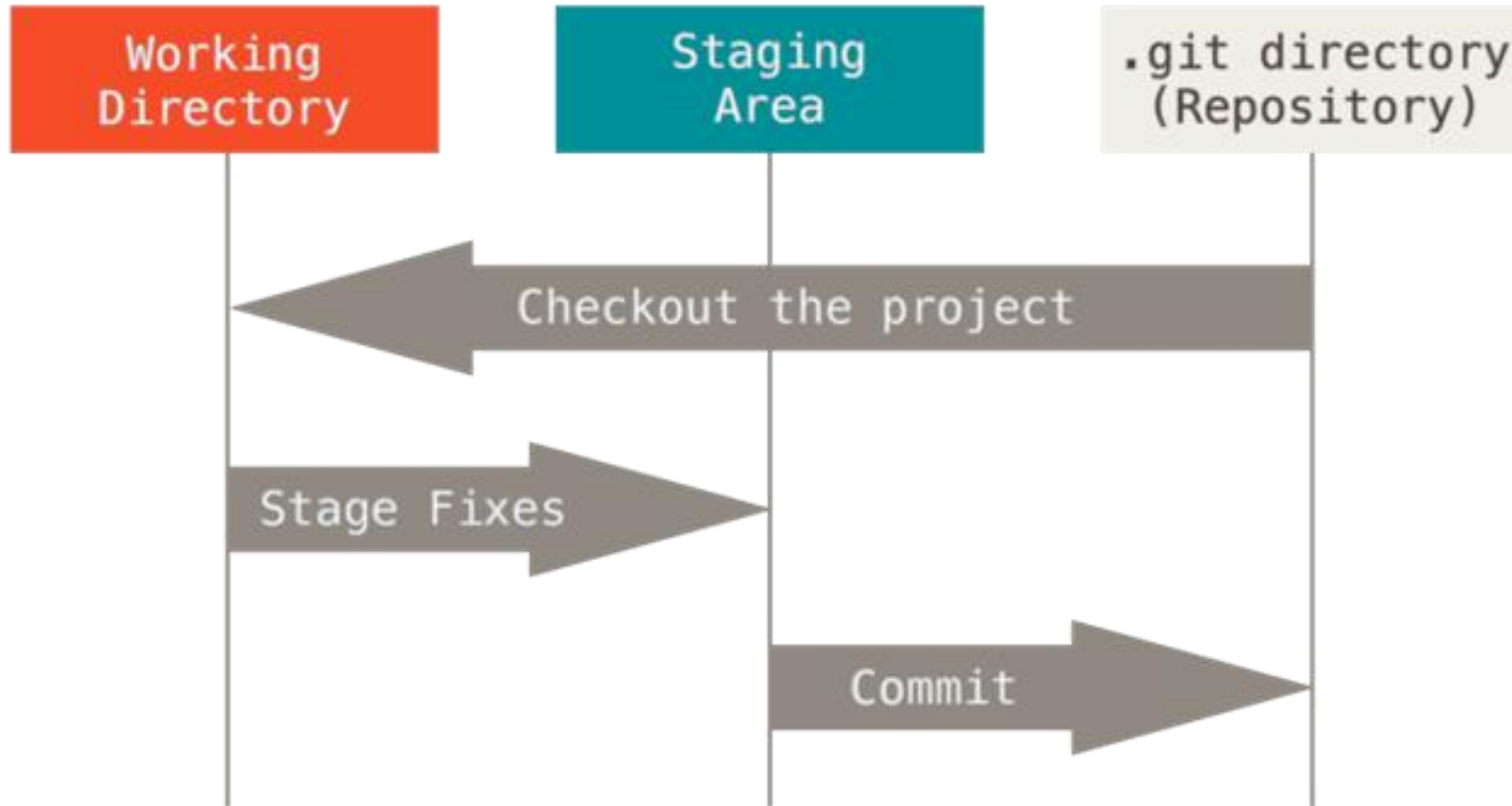
Staging  
Area

Repository 로 변경내역을 저장하기 위한 파일들의 목록

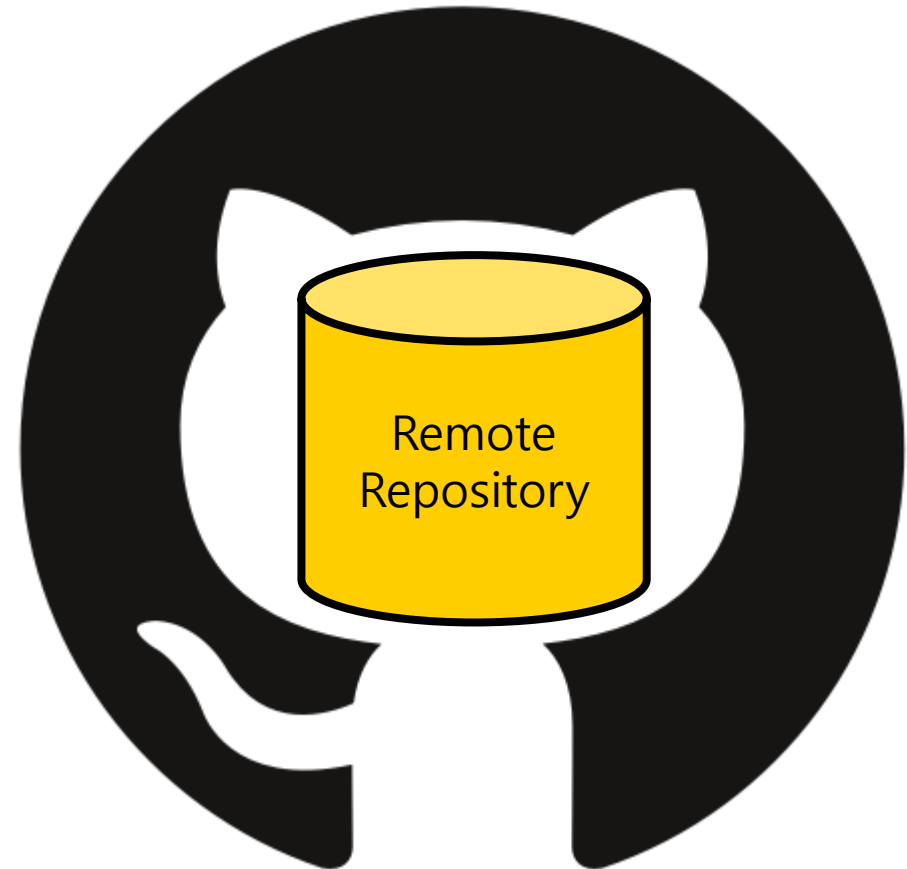
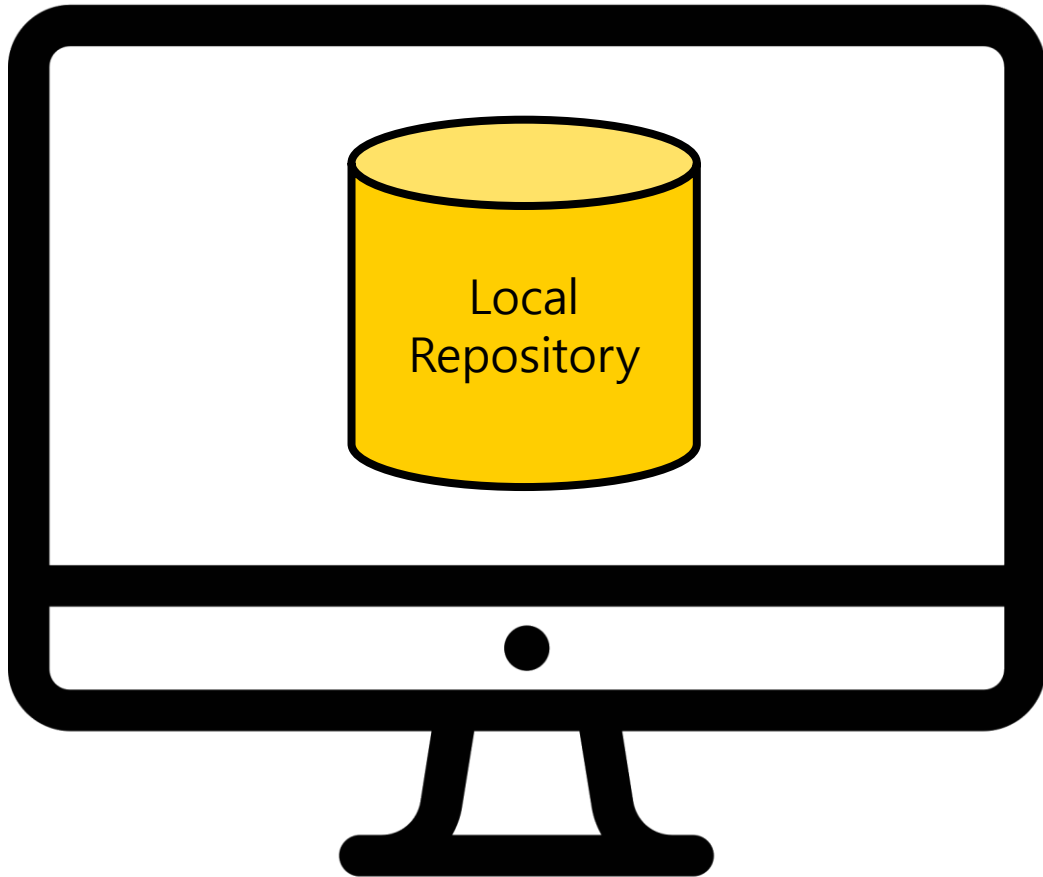
.git directory  
(Repository)

변경 내역이 저장되는 곳

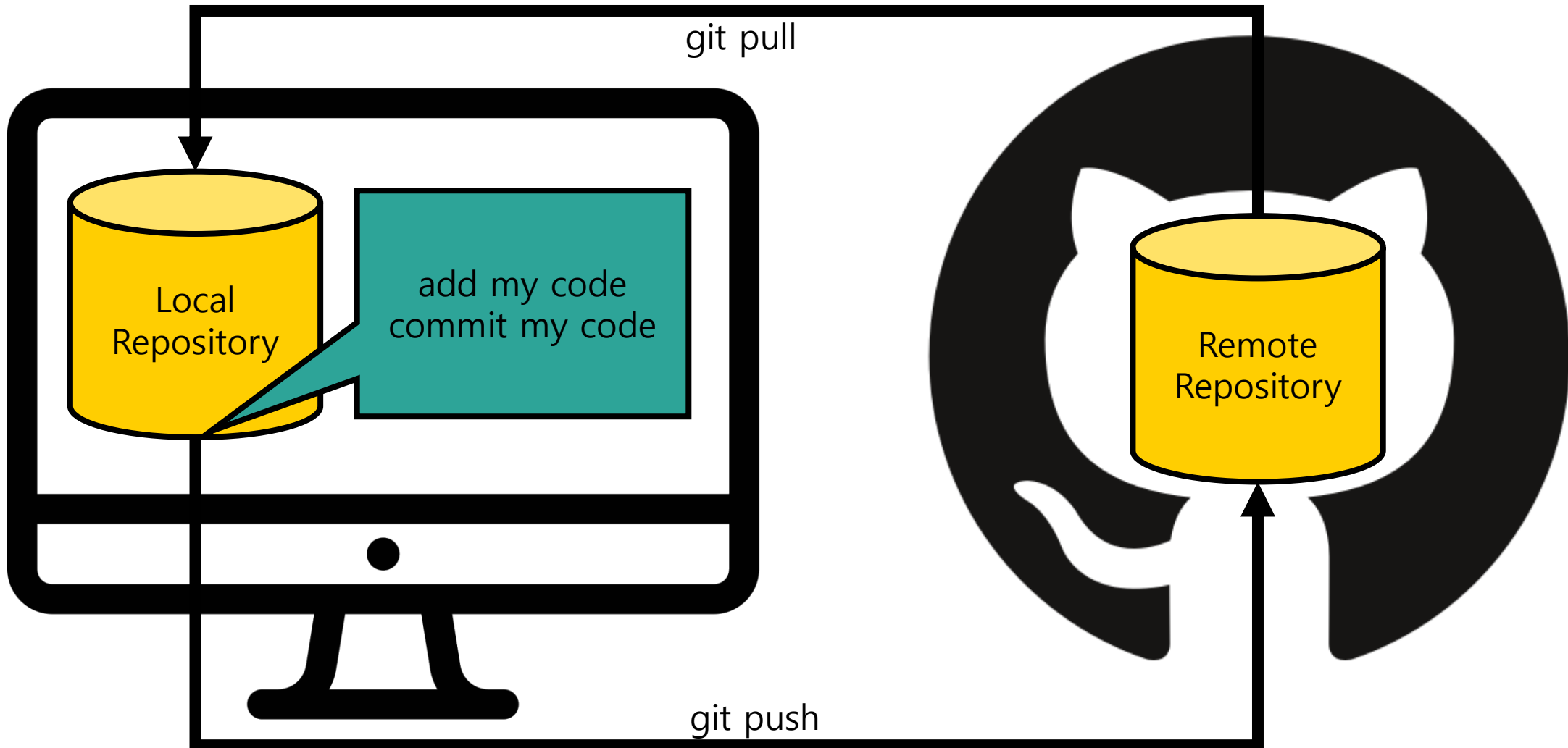
# Git status



# Local & Remote Repository



# Workflow





## << Round 7 >>

- Git 개요 - complete
- GitHub <<
- Git 실습



Let's  
Go



# Github??



Open Source Software를 위한 최고의 Remote Repository 사이트

# Github??

- 무료!
- 모든 소스코드 공개!!  
(돈 내면 Extended Private Repository 생성 가능)
- Github Pages라는 자체 웹 호스팅 서비스 제공  
(무료로 프로젝트 홈페이지 제작 가능)
- Open Source Software의 성지!!!

# Make a Github repository

Overview **Repositories 14** Projects 0 Packages 0 Stars 11 Followers 0 Following 6

Find a repository... Type: All Language: All **New**

---

**my\_GA\_studio** ★ Star  
Study GA(Genetic Algorithm) with python  
Python Apache License 2.0 Updated 1 hour ago

---

**Python\_Breakers** ★ Star  
파뿌리(파이썬 뿌시는 이십대들) 강의자료  
Python Updated 2 hours ago

---

**kakaotalk\_chatbot\_sandol** ★ Star  
kakao openbuilder로 만든 한국산업기술대학교용 카카오톡 챗봇 -산돌이-  
Python ★ 1 GNU General Public License v3.0 Updated 10 days ago

# Make a Github repository

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner



Repository name \*

git\_prac



Great repository names are short and memorable. Need inspiration? How about [supreme-engine](#)?

Description (optional)

test push



Public

Anyone can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ Initialize this repository with a README

This will let you immediately clone the repository to your computer.

Add .gitignore: None

Add a license: MIT License



Create repository

# Make a Github repository

KGJsGit / **git\_test** Unwatch 1 Star 0 Fork 0

[Code](#) [Issues 0](#) [Pull requests 0](#) [Actions](#) [Projects 0](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

To my git testing Repository Edit

[Manage topics](#)

1 commit 1 branch 0 packages 0 releases 1 contributor GPL-3.0

Branch: master [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)


KGJsGit Initial commit Latest commit fe97191 on 3 Jan

<a href="#">.gitignore</a>	Initial commit	2 months ago
<a href="#">LICENSE</a>	Initial commit	2 months ago
<a href="#">README.md</a>	Initial commit	2 months ago

[README.md](#)

## git\_test

To my git testing Repository

© 2020 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

## << Round 7 >>

- Git 개요 - complete
- GitHub - complete
- Git 실습 <<



Let's  
Go

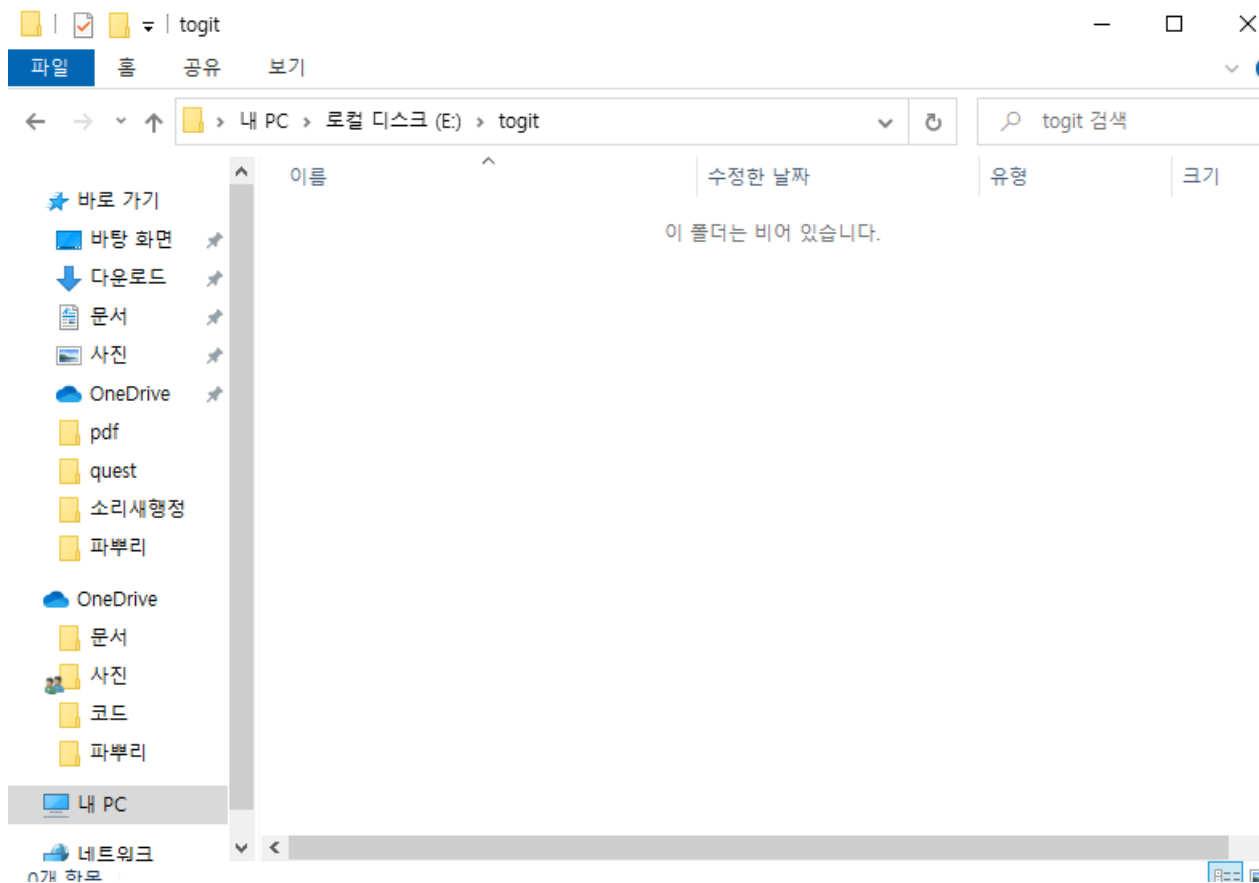


# Git keyword

- git init  
: git repository 생성
- git add "file name"  
: 해당 파일 stage
- git commit -m "comment"  
: staged 파일들 commit(repository에 반영)
- git pull "remote repository address"  
: 해당 remote repository를 받아와서 내 local repository에 합병하기
- git push origin master  
: local repository를 remote repository로 밀어넣기



# Git training



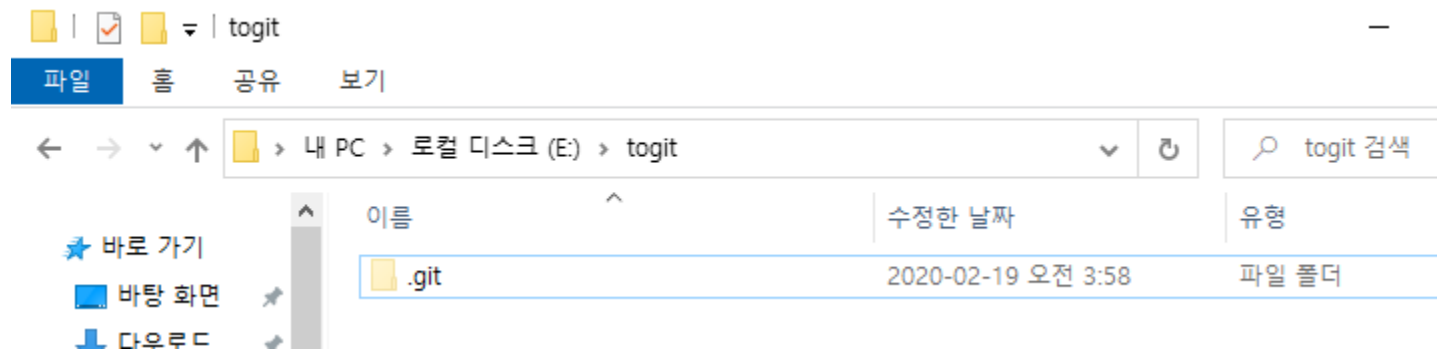
make a directory (anywhere)

# Git training

C:\> 명령 프롬프트

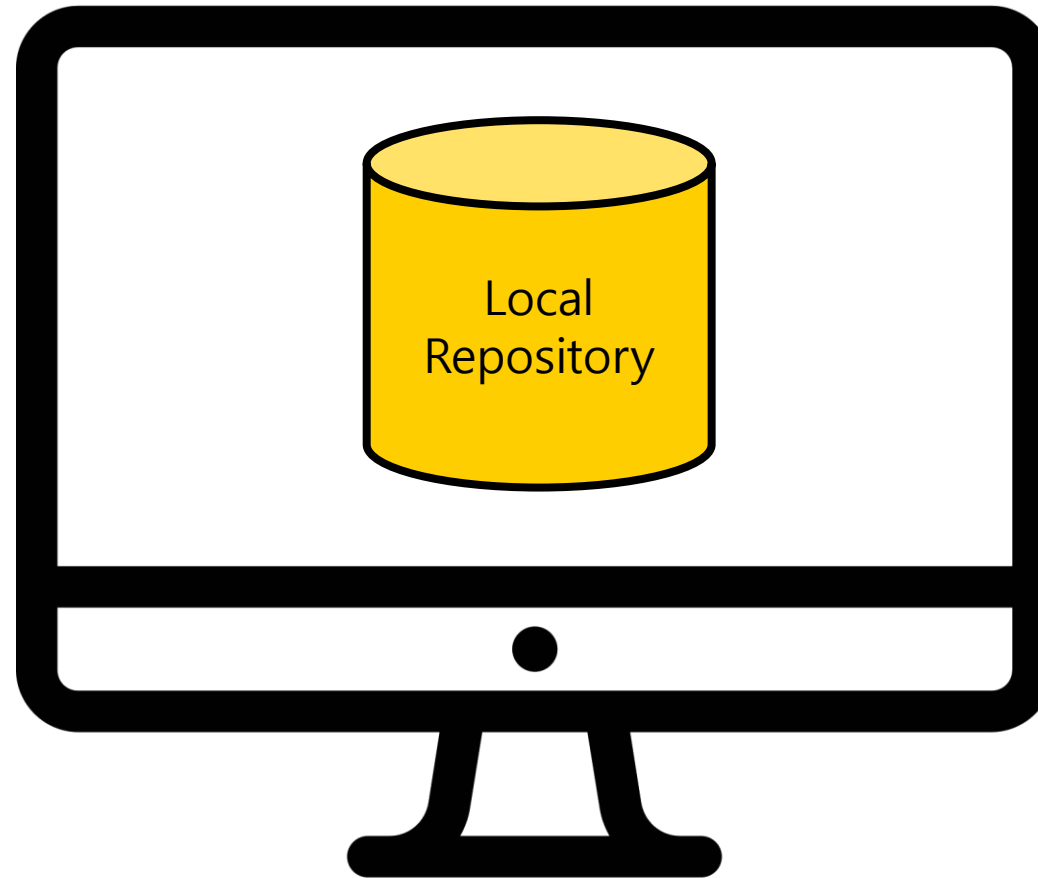
```
E:\wtogit>git init
Reinitialized existing Git repository in E:/togit/.git/
E:\wtogit>
```

Go to directory & "git init"



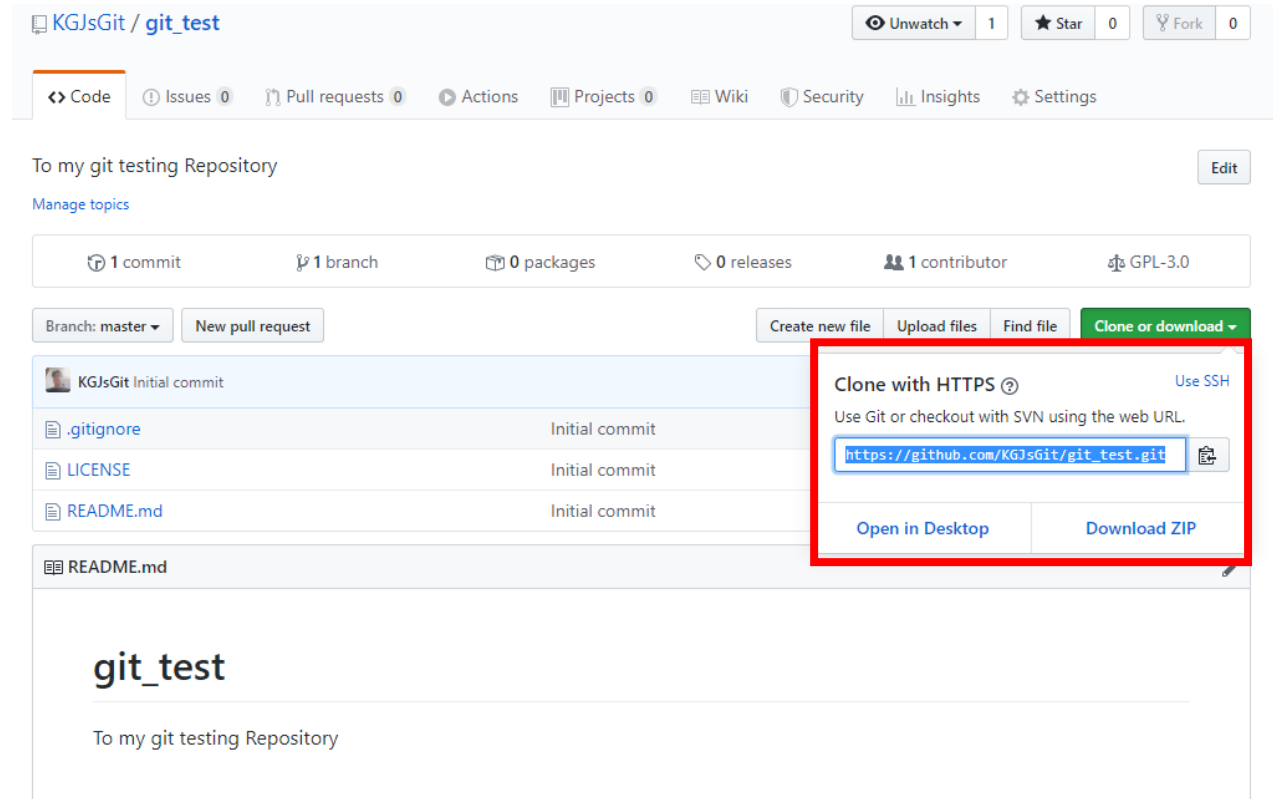
Check your local repository

# Git training



"git init"

# Git training



KGJsGit / git\_test

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

To my git testing Repository [Edit](#)

[Manage topics](#)

1 commit 1 branch 0 packages 0 releases 1 contributor GPL-3.0

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

KGJsGit Initial commit

<a href="#">.gitignore</a>	Initial commit
<a href="#">LICENSE</a>	Initial commit
<a href="#">README.md</a>	Initial commit

README.md

## git\_test

To my git testing Repository

Clone with HTTPS Use SSH

Use Git or checkout with SVN using the web URL.

[https://github.com/KGJsGit/git\\_test.git](https://github.com/KGJsGit/git_test.git)

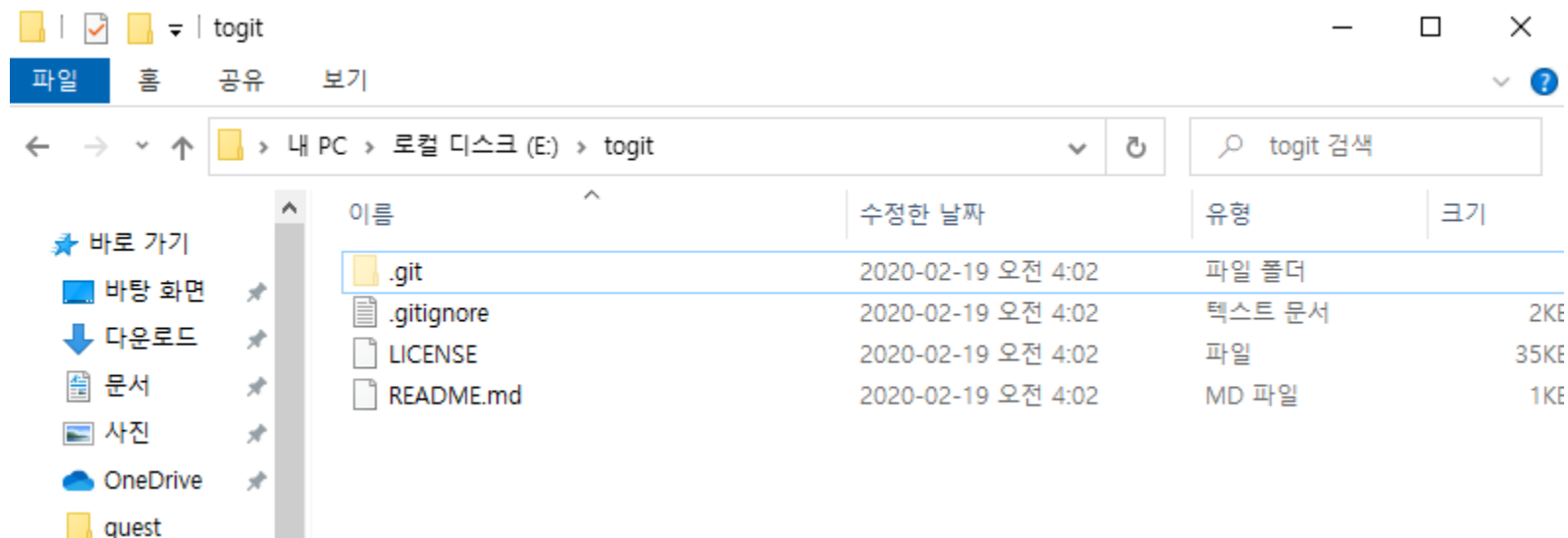
Open in Desktop Download ZIP

Copy the remote repository URL

# Git training

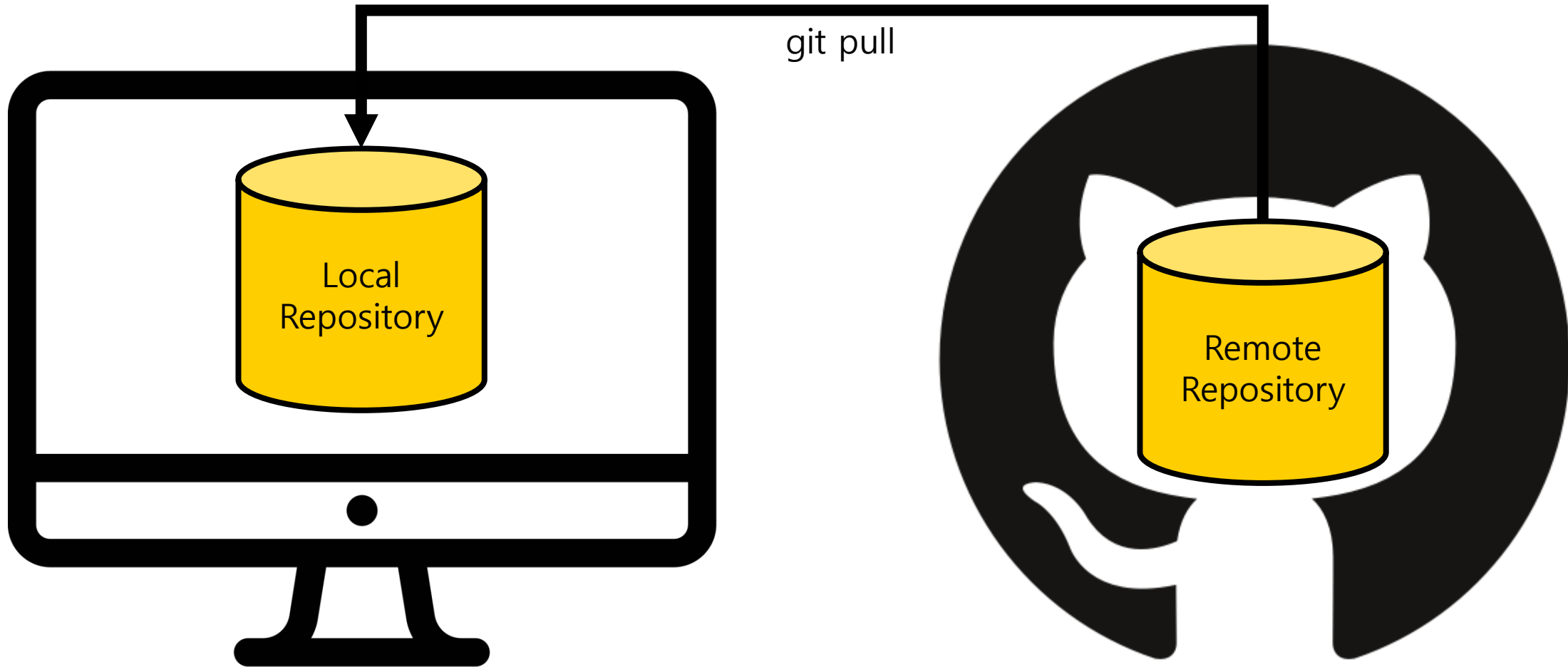
```
E:\tgit>git pull https://github.com/KGJsGit/git_test.git
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 13.45 KiB | 57.00 KiB/s, done.
From https://github.com/KGJsGit/git_test
* branch HEAD -> FETCH_HEAD
E:\tgit>
```

"git pull 'remote repository URL' "



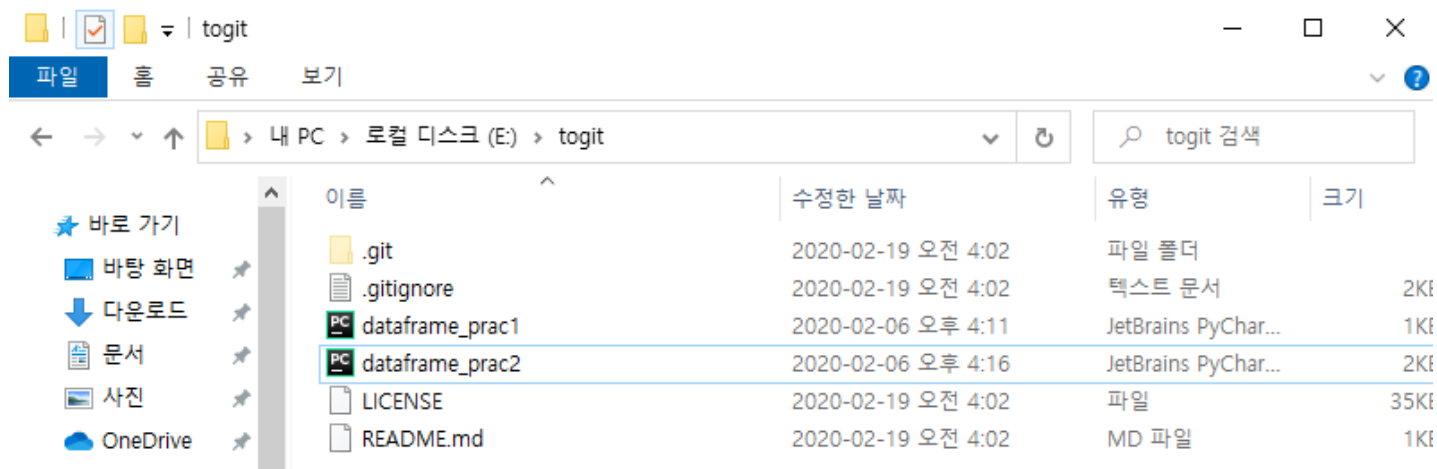
check your fetched local repository

# Git training



"git pull 'remote repository URL' "

# Git training



make files on the directory

```
E:\wtogit>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        dataframe_prac1.py
        dataframe_prac2.py

nothing added to commit but untracked files present (use "git add" to track)
E:\wtogit>
```

check the git status, "git status"

# Git training



```
E:\wtogit>git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    dataframe_prac1.py
    dataframe_prac2.py

nothing added to commit but untracked files present (use "git add" to track)
E:\wtogit>
```

check the git status, "git status"  
(2 files are untracked)



# Git training

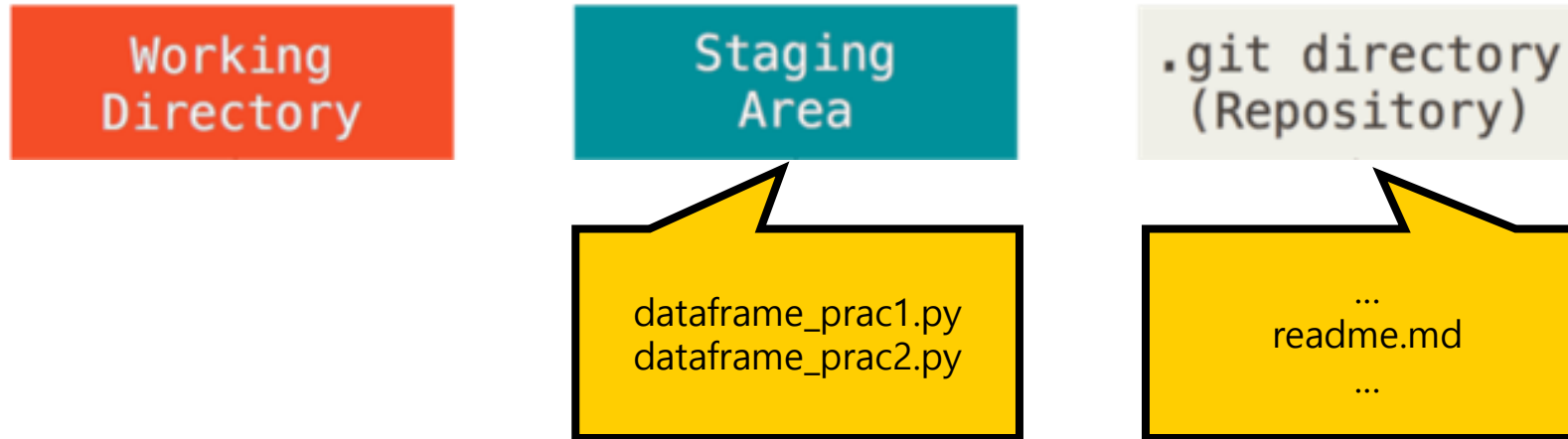
```
E:\wtogit>git add dataframe_prac1.py  
E:\wtogit>git add dataframe_prac2.py
```

"git add 'file name' "

```
E:\wtogit>git status  
On branch master  
Changes to be committed:  
  (use "git restore --staged <file>..." to unstage)  
    new file:   dataframe_prac1.py  
    new file:   dataframe_prac2.py  
  
E:\wtogit>
```

check the git status, "git status"

# Git training



```
E:\wtogit>git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   dataframe_prac1.py
        new file:   dataframe_prac2.py

E:\wtogit>
```

check the git status, "git status"  
(2 files are staged)



# Git training

Working  
Directory

Staging  
Area

.git directory  
(Repository)

...  
readme.md  
dataframe\_prac1.py  
dataframe\_prac2.py  
...

```
E:\wtogit>git status
On branch master
nothing to commit, working tree clean
E:\wtogit>
```

check the git status, "git status"

# Git training

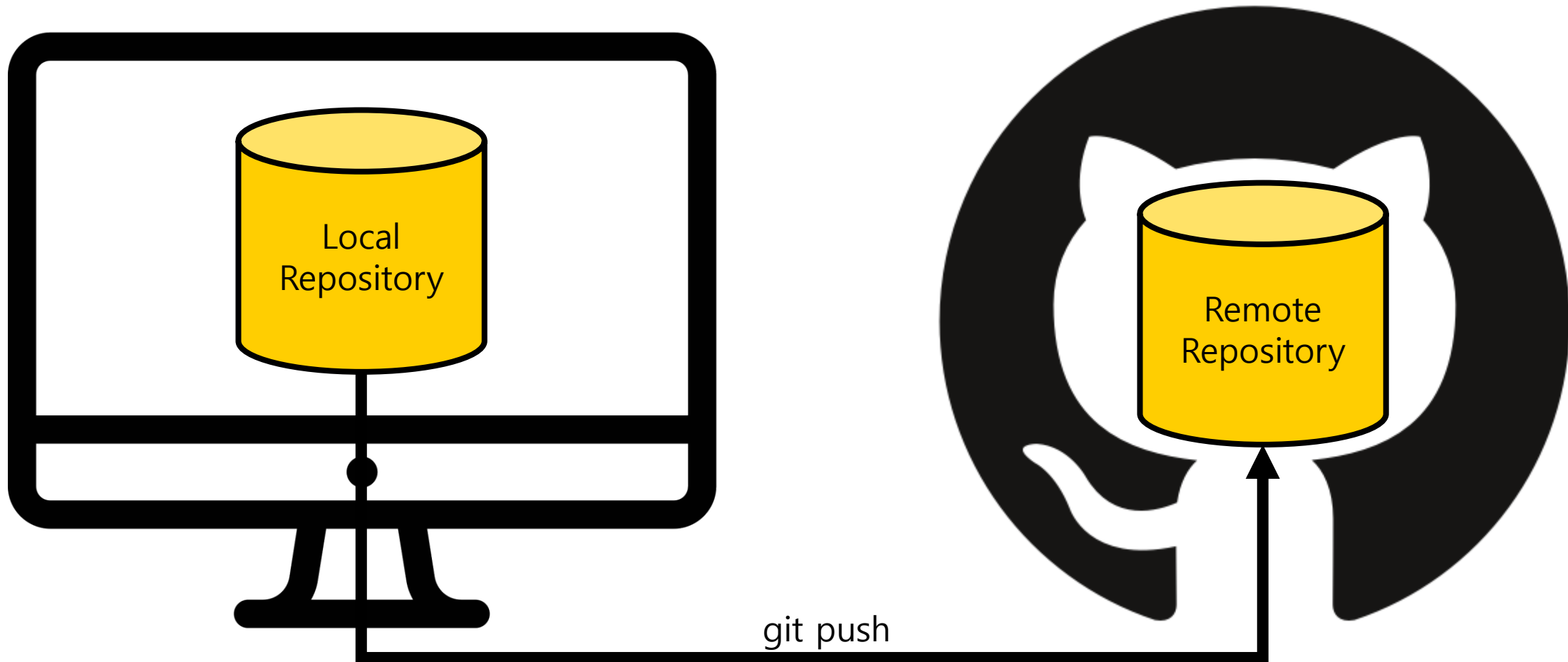
```
E:\wtogit>git remote add origin https://github.com/KGJsGit/git_test.git
```

"git remote add origin 'remote repository URL'"  
(push해줄 remote repository의 URL 명시)

```
E:\wtogit>git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 1.31 KiB | 1.31 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/KGJsGit/git_test.git
   fe97191..9c4acc0  master -> master
```

"git push origin master"

# Git training



"git push origin master"

# Git training

KGJsGit / `git_test`
Unwatch 1
Star 0
Fork 0

[Code](#)
[Issues 0](#)
[Pull requests 0](#)
[Actions](#)
[Projects 0](#)
[Wiki](#)
[Security](#)
[Insights](#)
[Settings](#)

To my git testing Repository Edit

[Manage topics](#)

2 commits
1 branch
0 packages
0 releases
1 contributor
GPL-3.0

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

KGJsGit add new files
Latest commit 9c4acc0 6 minutes ago

<code>.gitignore</code>	Initial commit	2 months ago
<code>LICENSE</code>	Initial commit	2 months ago
<code>README.md</code>	Initial commit	2 months ago
<code>dataframe_prac1.py</code>	add new files	6 minutes ago
<code>dataframe_prac2.py</code>	add new files	6 minutes ago

README.md

## git\_test

To my git testing Repository

## Additional Git keyword

- git clone

: remote repository의 모든 데이터를 복사, push시 원격저장소 자동 지정

- git rm 'file name'

: 깃의 파일을 삭제, working directory에서도 삭제됩니다. commit필요.

- git push -u origin master

: push의 -u 옵션은 remote repository로부터 업데이트 받은 후 push하겠다는 의미

- git commit --amend -m 'comment'

: 최근 실수한 commit을 덮어 쓸 수 있음. 다만 되돌린 commit을 다시 되돌릴 수는 없음.

- git reset HEAD 'file name'

: 실수로 staging area에 올린 파일을 unstaged로 바꿔 줌

- git reset hard 'commit num'

: 해당 commit으로 git을 되돌리고 그 이후 모든 이력을 삭제합니다. hard대신 다양한 옵션 사용 가능



