



Round 4

**PRESS
START**



《 Round 4 》

- Git과 Github
- 가상환경
- Jupyter Lab



New
Assignment



《 Round 4 》

- Git과 Github 《
- 가상환경
- Jupyter Lab



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

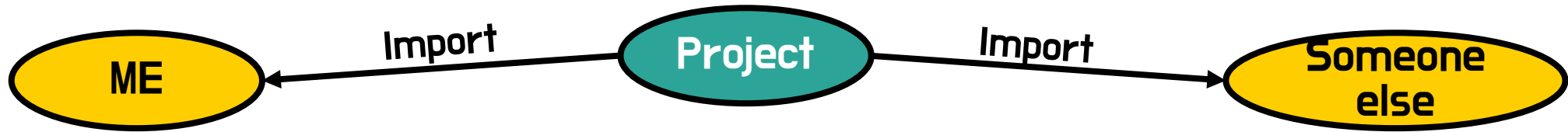
팀원2 + 팀원3 : 로그인 관련 DB

팀원1 + 팀원3 : 로그인, 메인페이지 프론트엔드

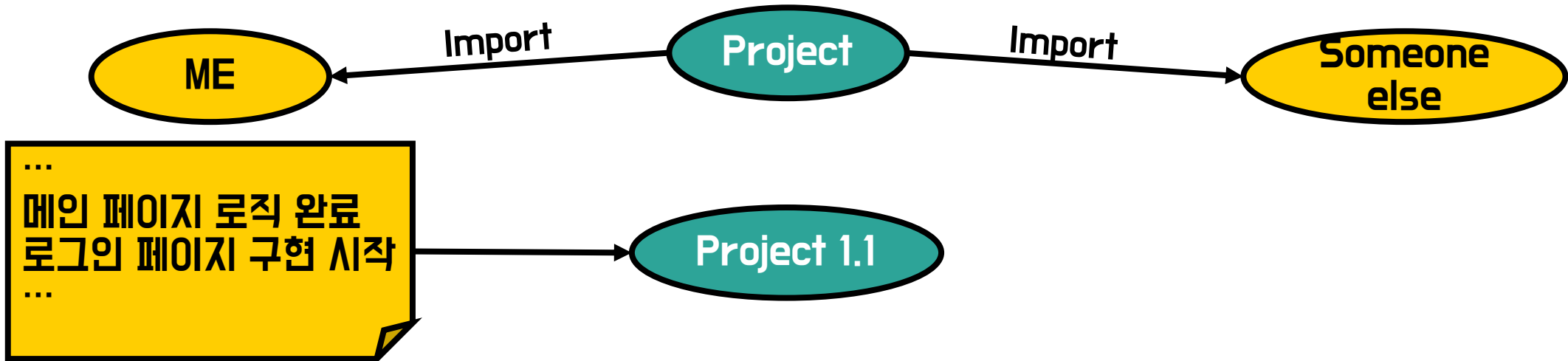
...

중간 검토 기간 : 이를 뒤...

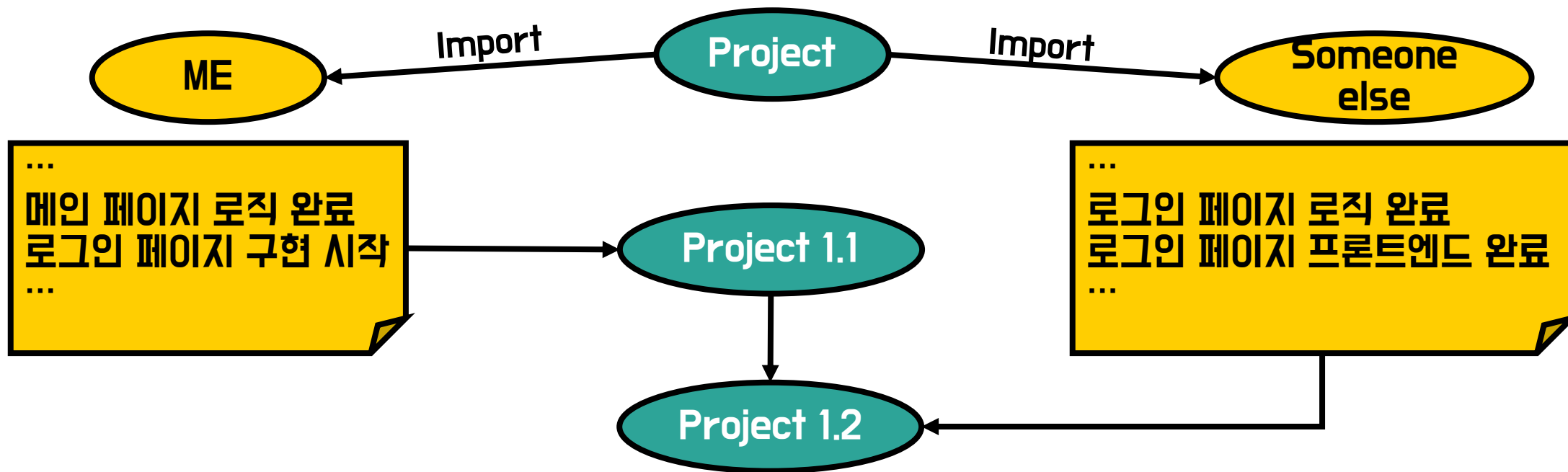
Why use it?



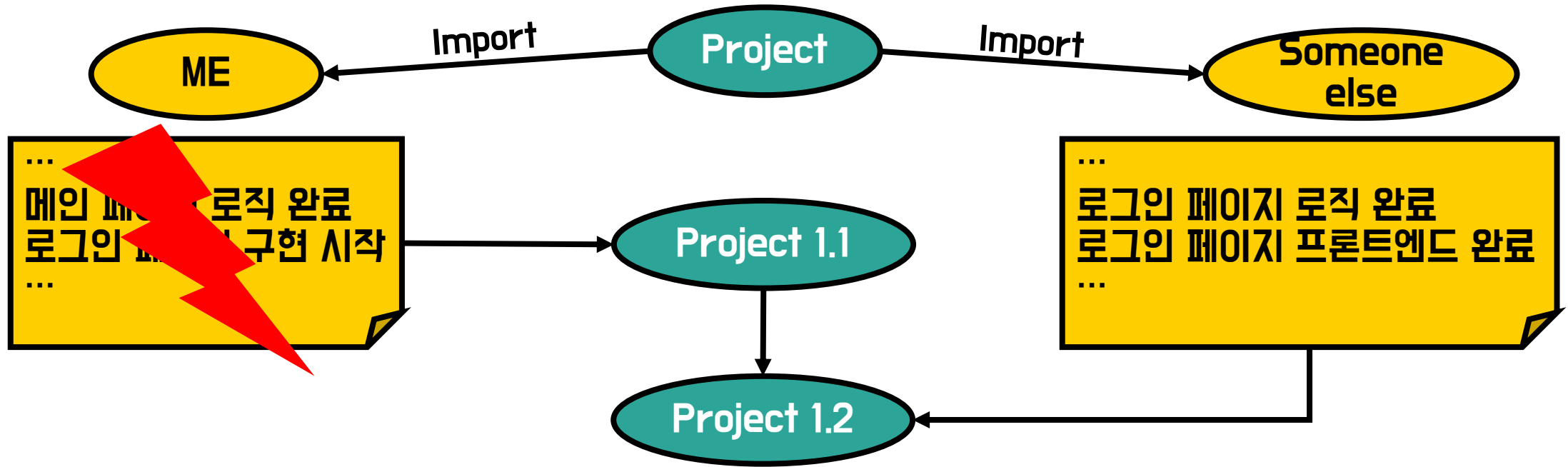
Why use it?



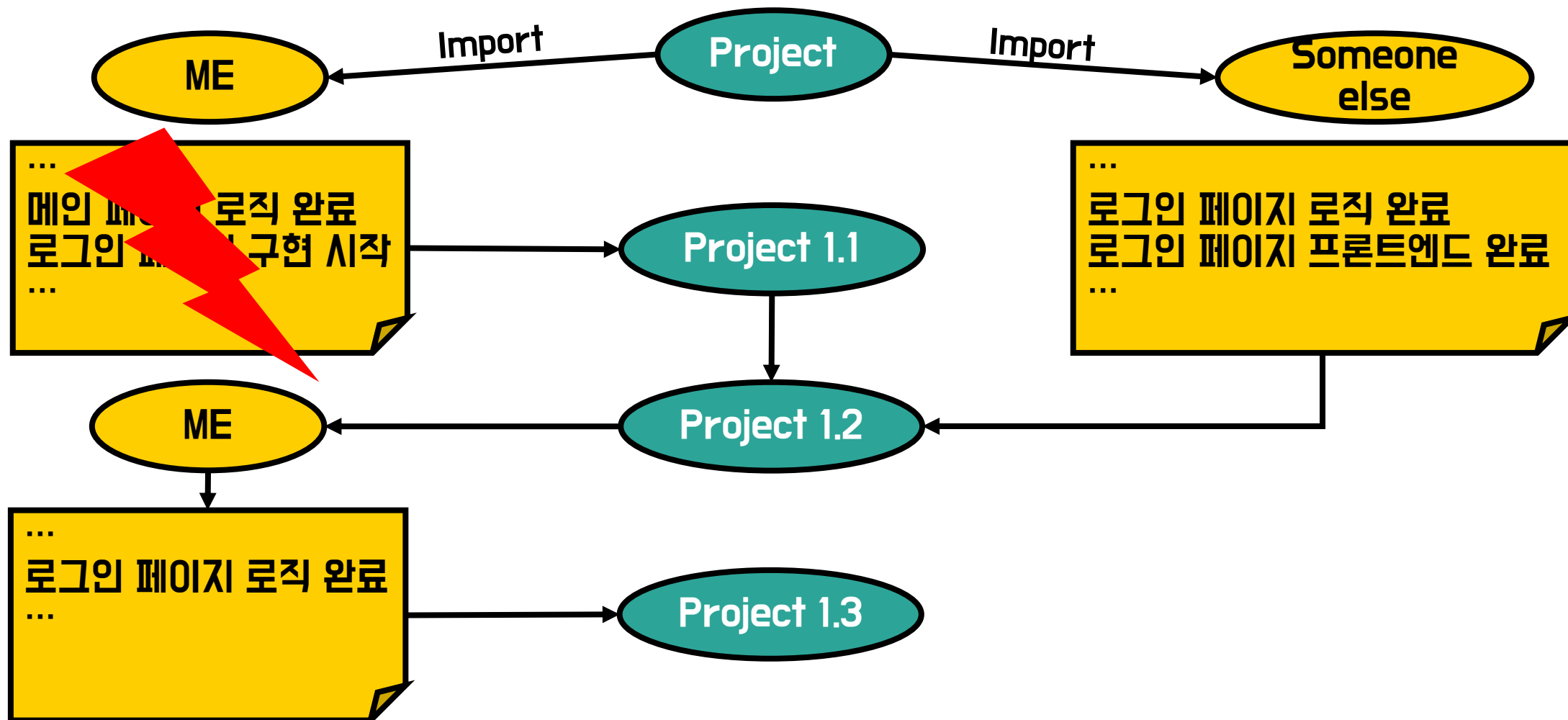
Why use it?



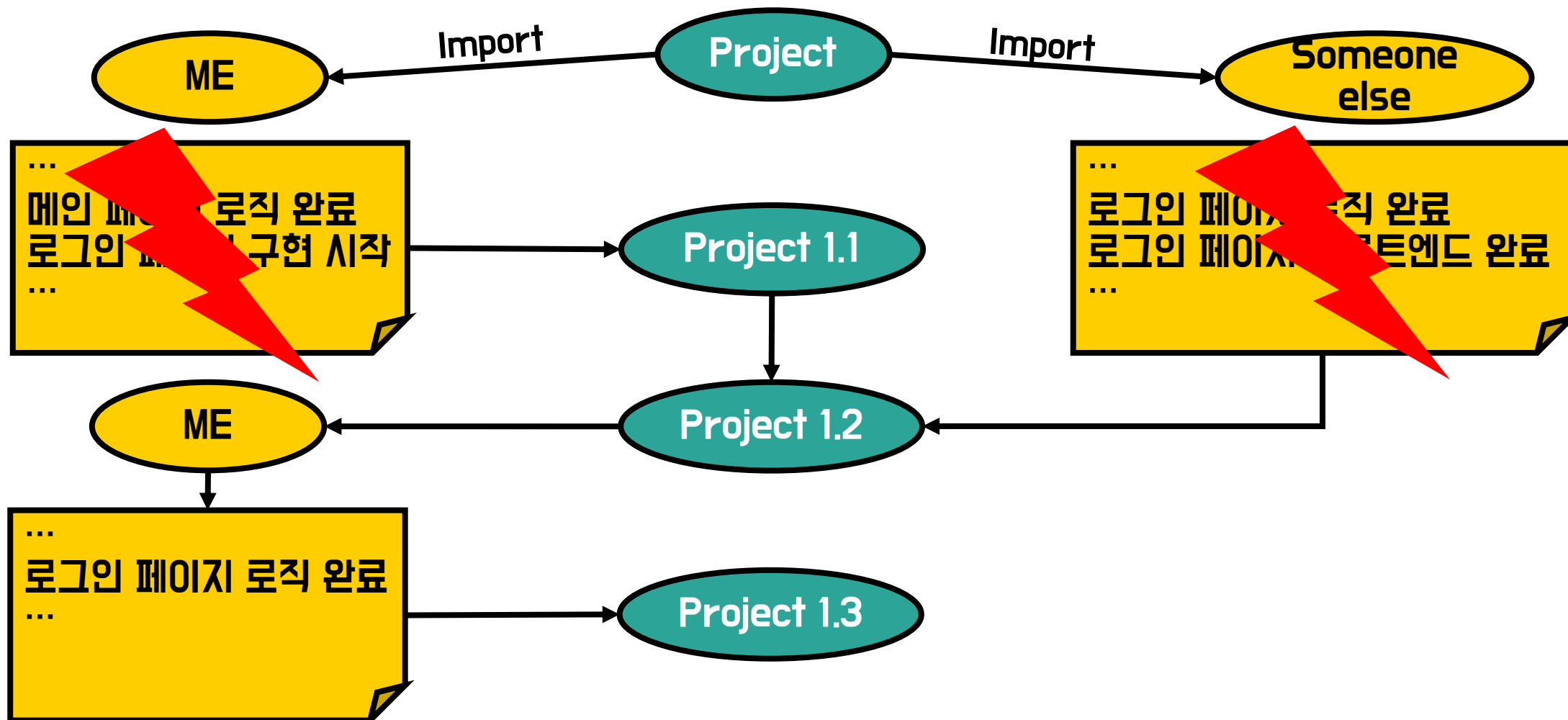
Why use it?



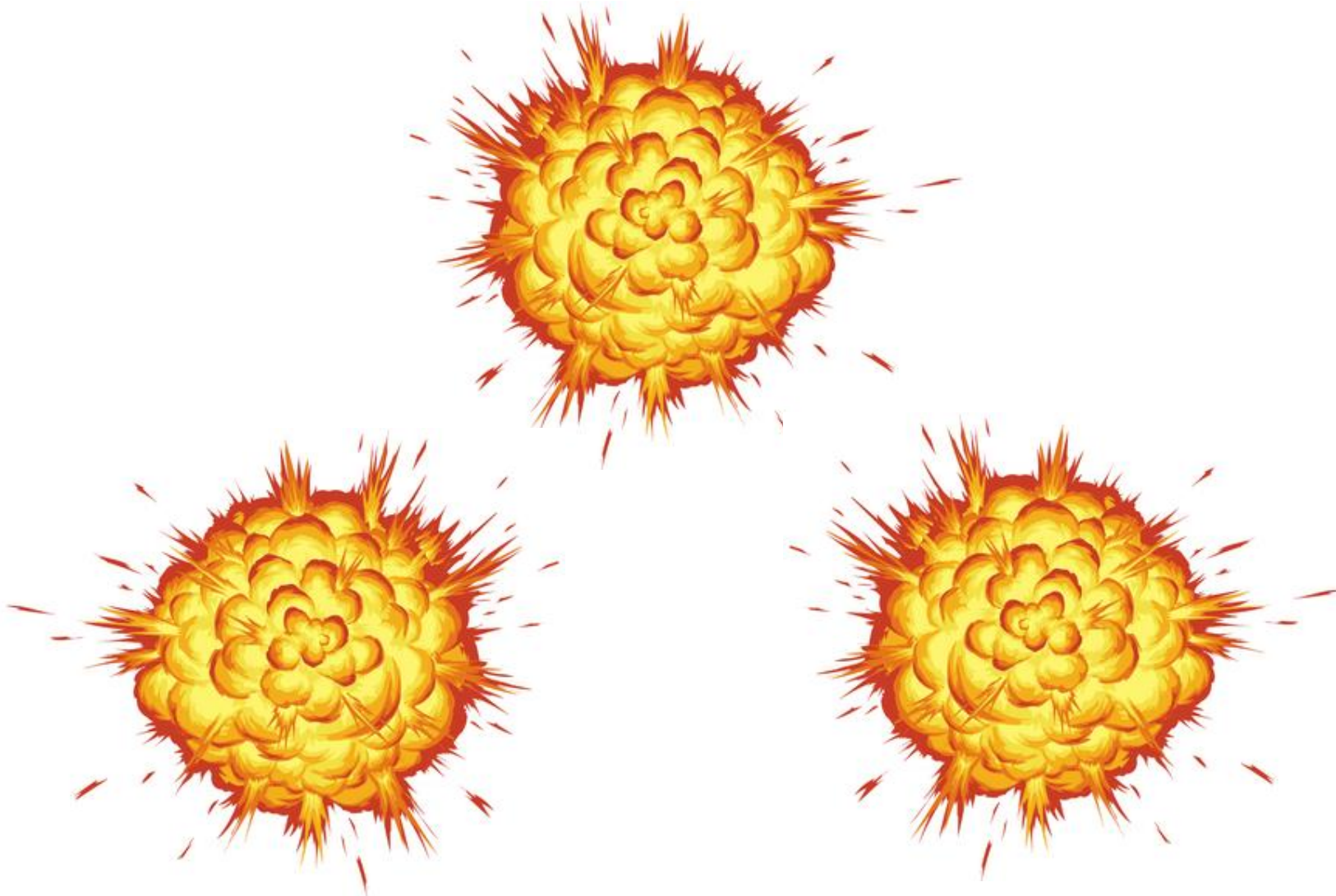
Why use it?



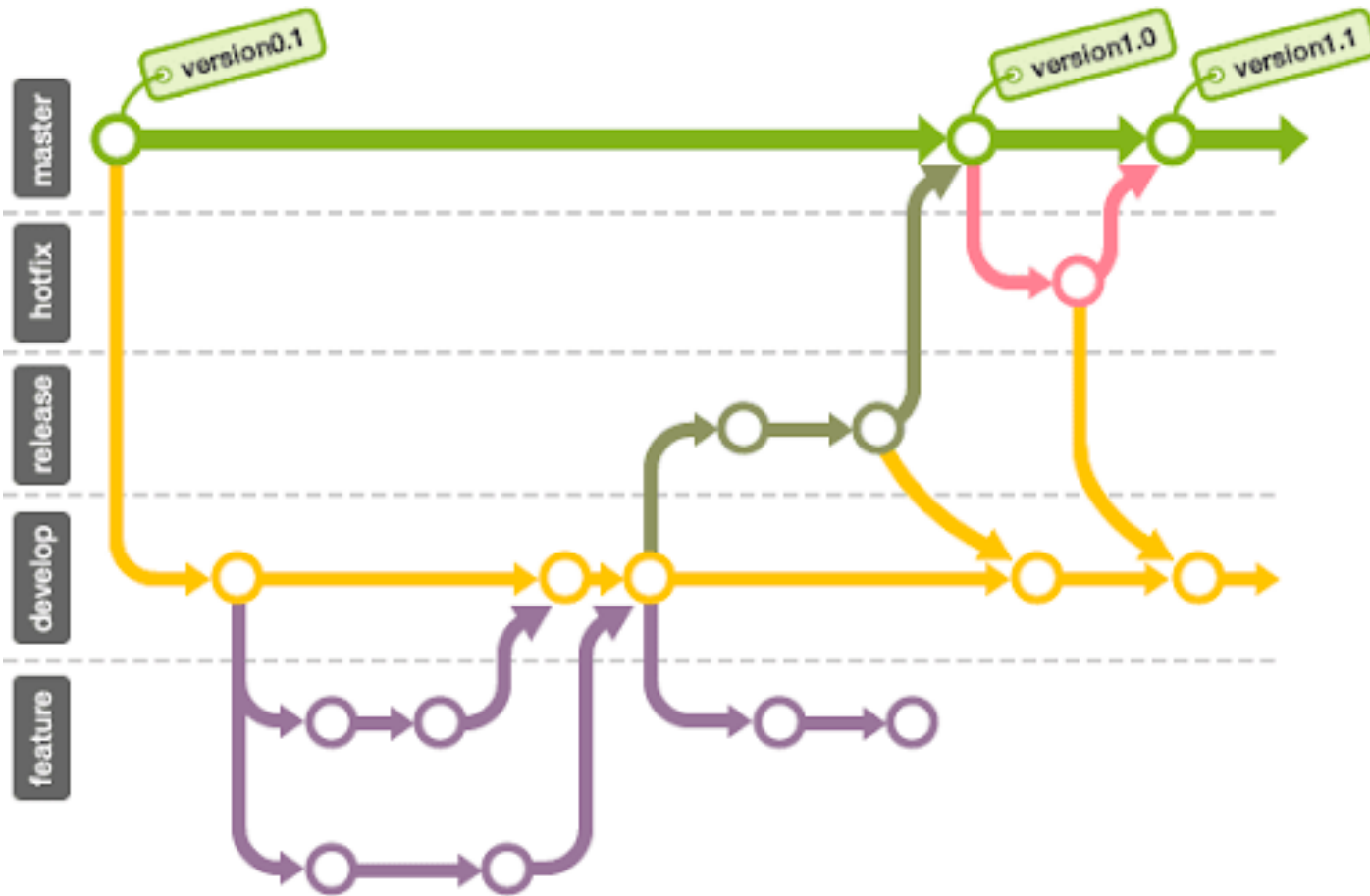
Why use it?



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 →](#)

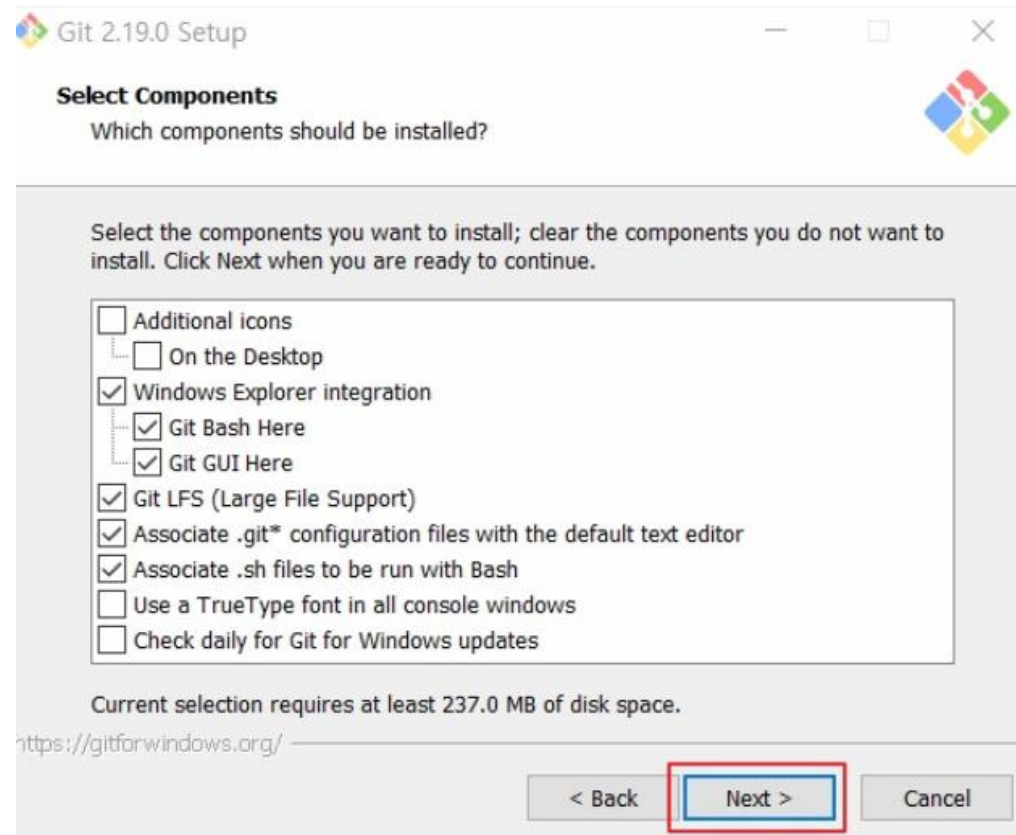
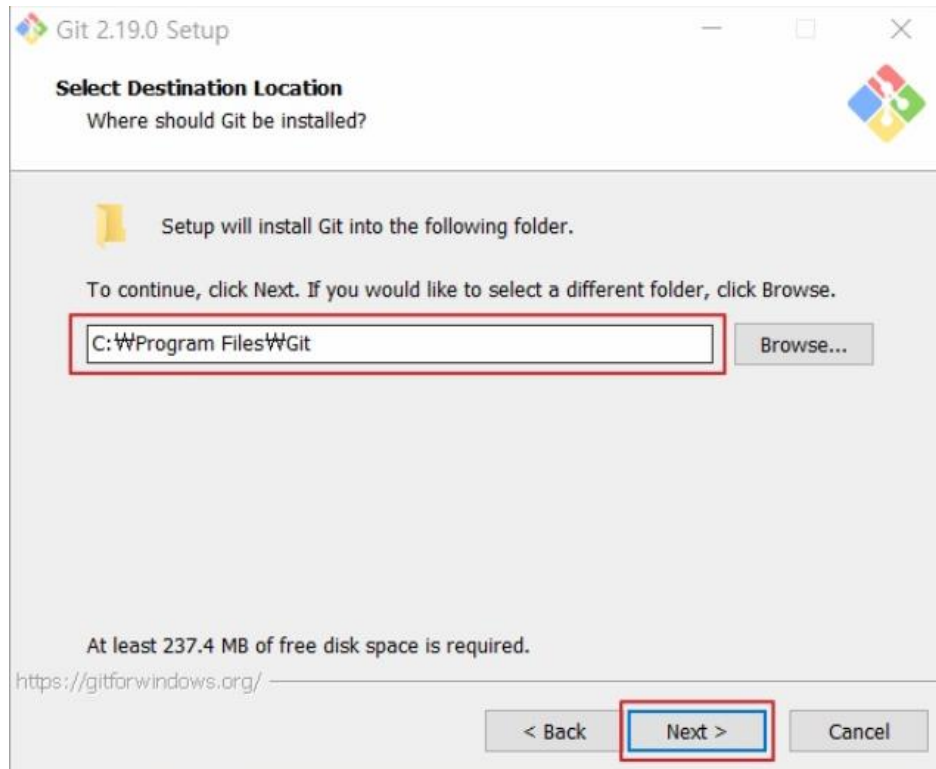
Git via Git

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

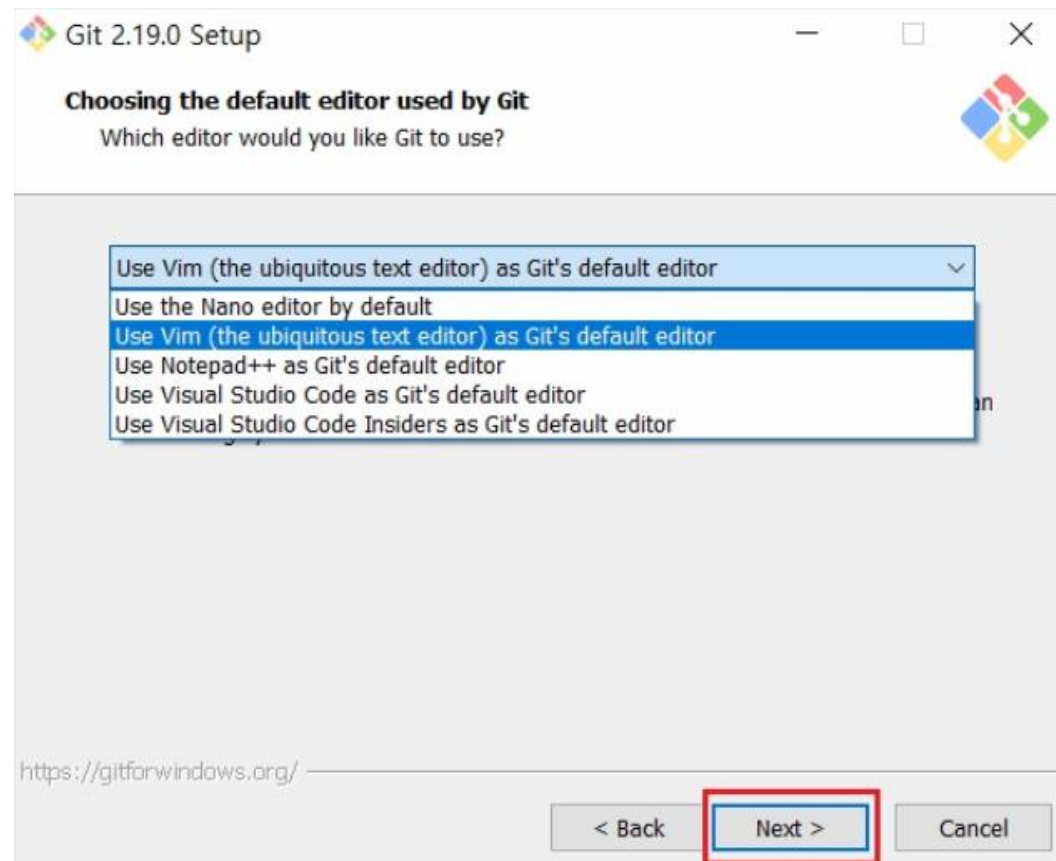
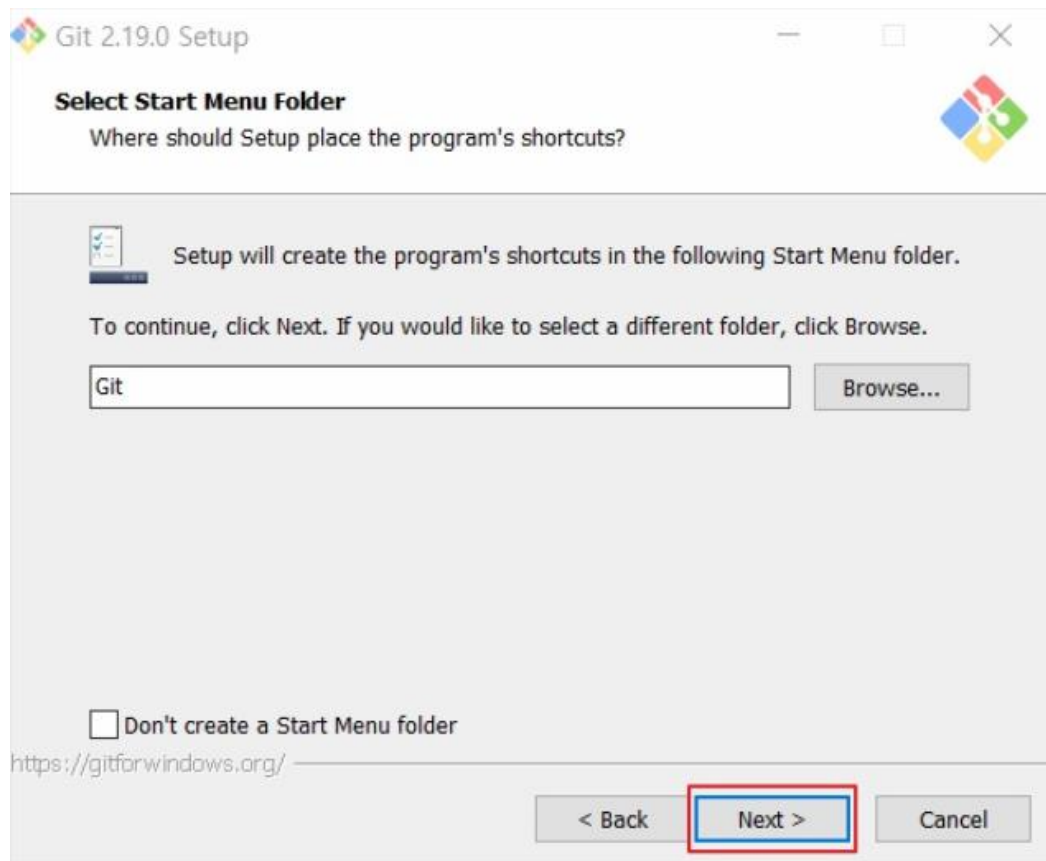
```
git clone https://github.com/git/git
```

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

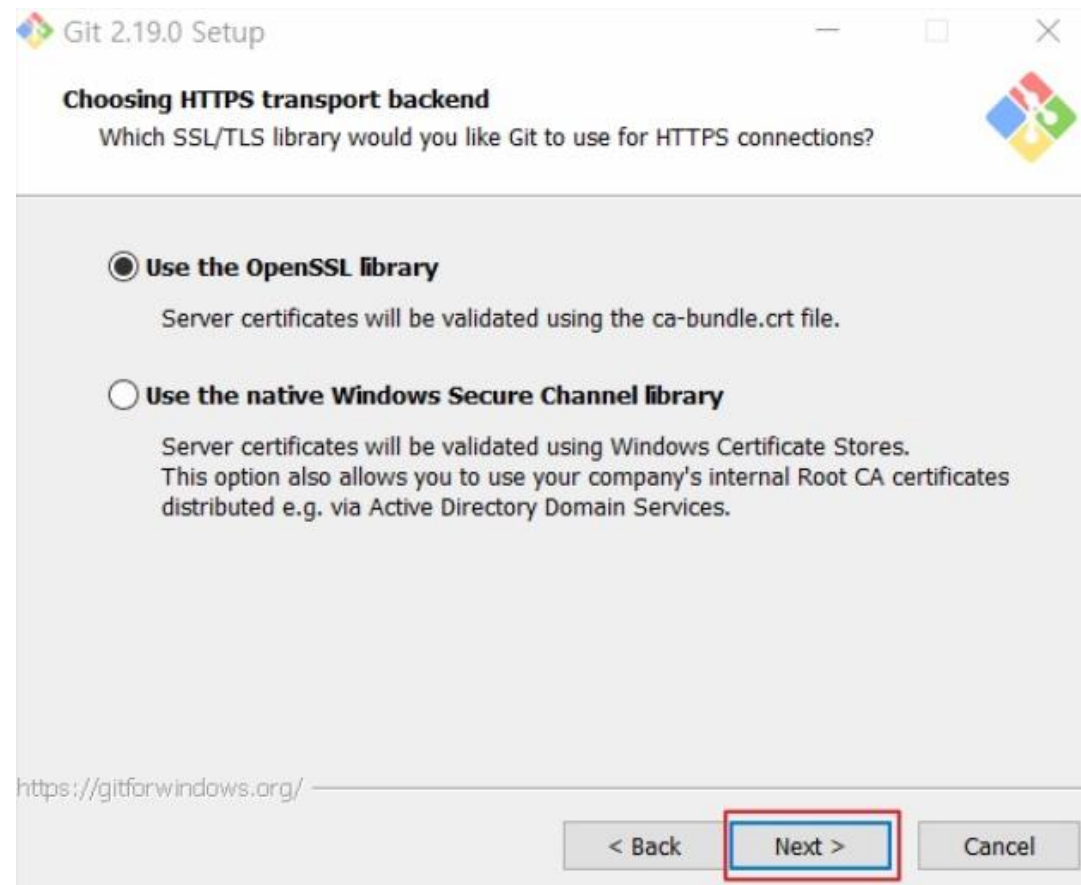
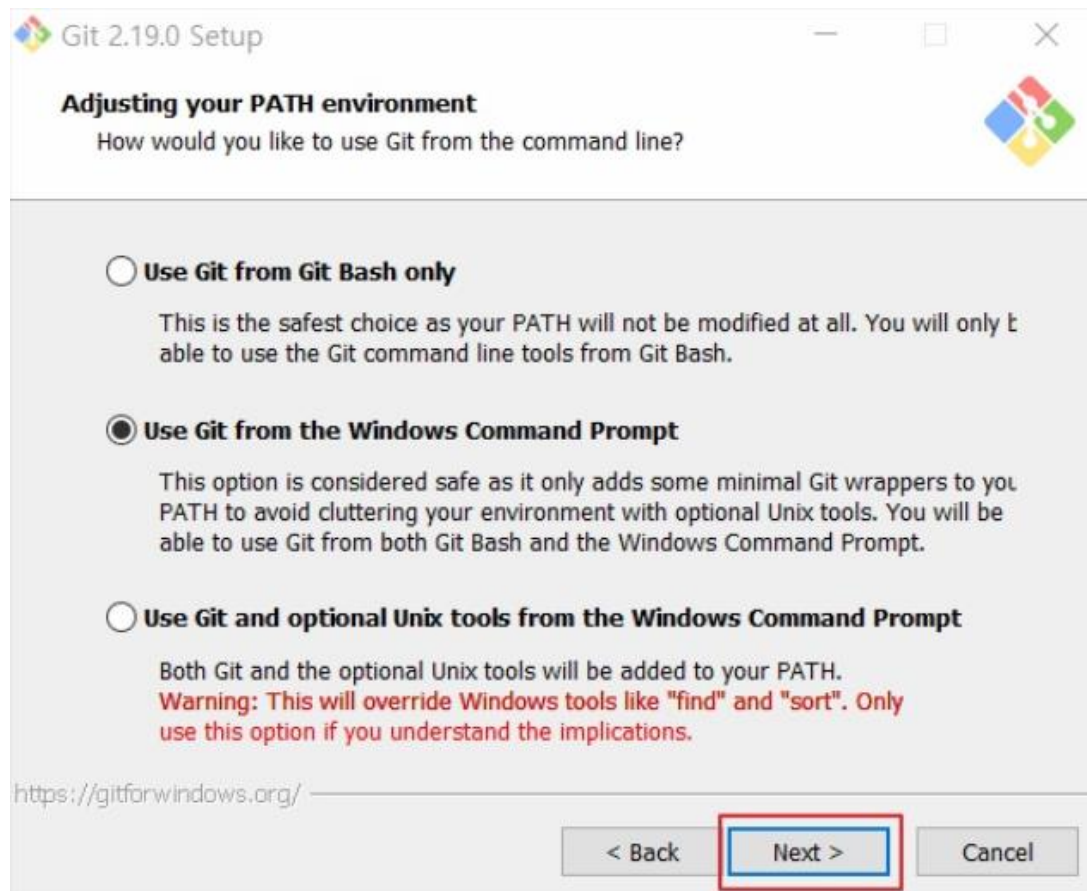
Install git



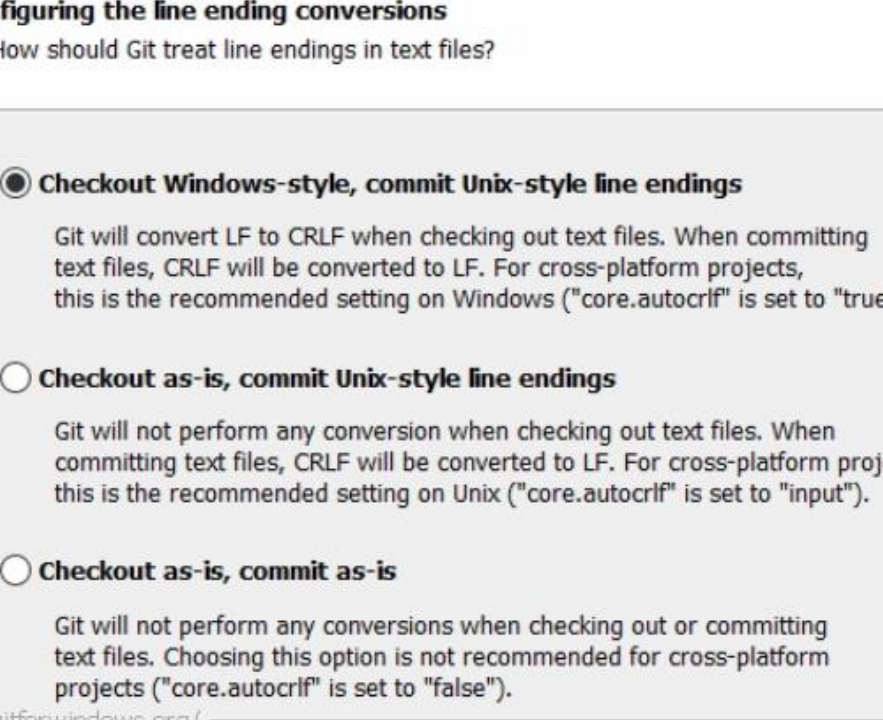
Install git



Install git



Install git



Git 2.19.0 Setup

Configuring the line ending conversions

How should Git treat line endings in text files?

☒ **Checkout Windows-style, commit Unix-style line endings**

Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").

☐ **Checkout as-is, commit Unix-style line endings**

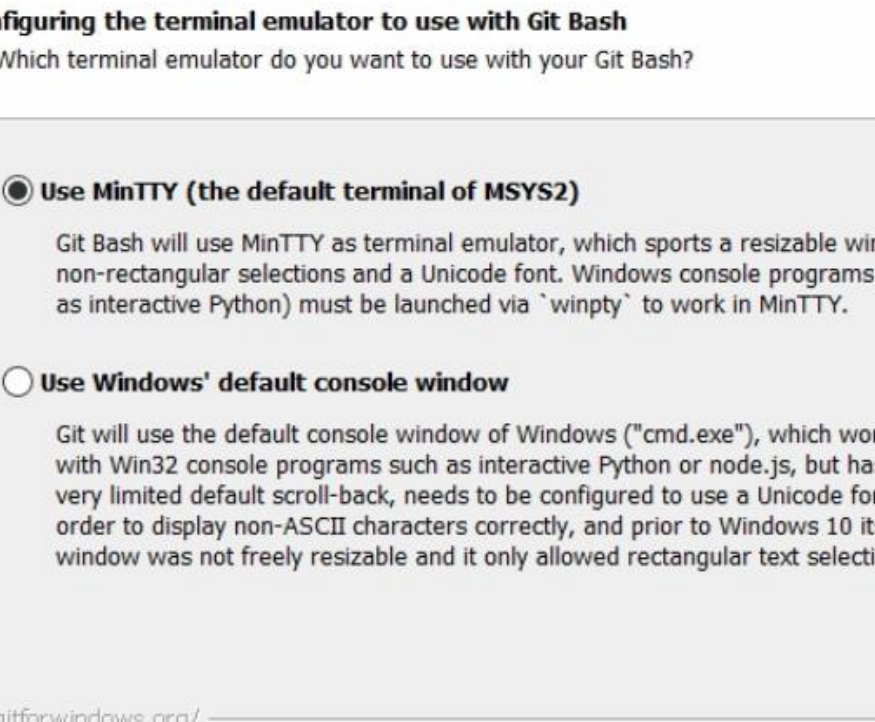
Git will not perform any conversion when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Unix ("core.autocrlf" is set to "input").

☐ **Checkout as-is, commit as-is**

Git will not perform any conversions when checking out or committing text files. Choosing this option is not recommended for cross-platform projects ("core.autocrlf" is set to "false").

<https://gitforwindows.org/>

< Back **Next >** Cancel



Git 2.19.0 Setup

Configuring the terminal emulator to use with Git Bash

Which terminal emulator do you want to use with your Git Bash?

☒ **Use MinTTY (the default terminal of MSYS2)**

Git Bash will use MinTTY as terminal emulator, which sports a resizable window non-rectangular selections and a Unicode font. Windows console programs (such as interactive Python) must be launched via ``winpty`` to work in MinTTY.

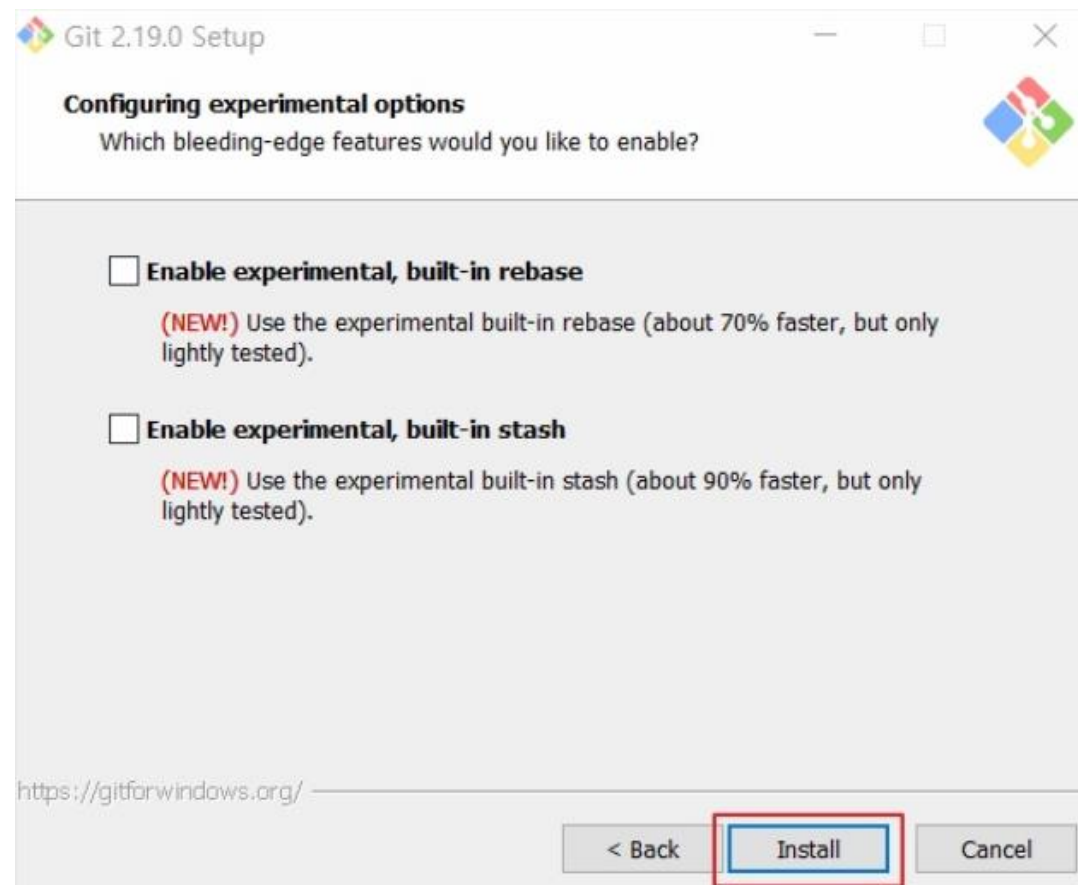
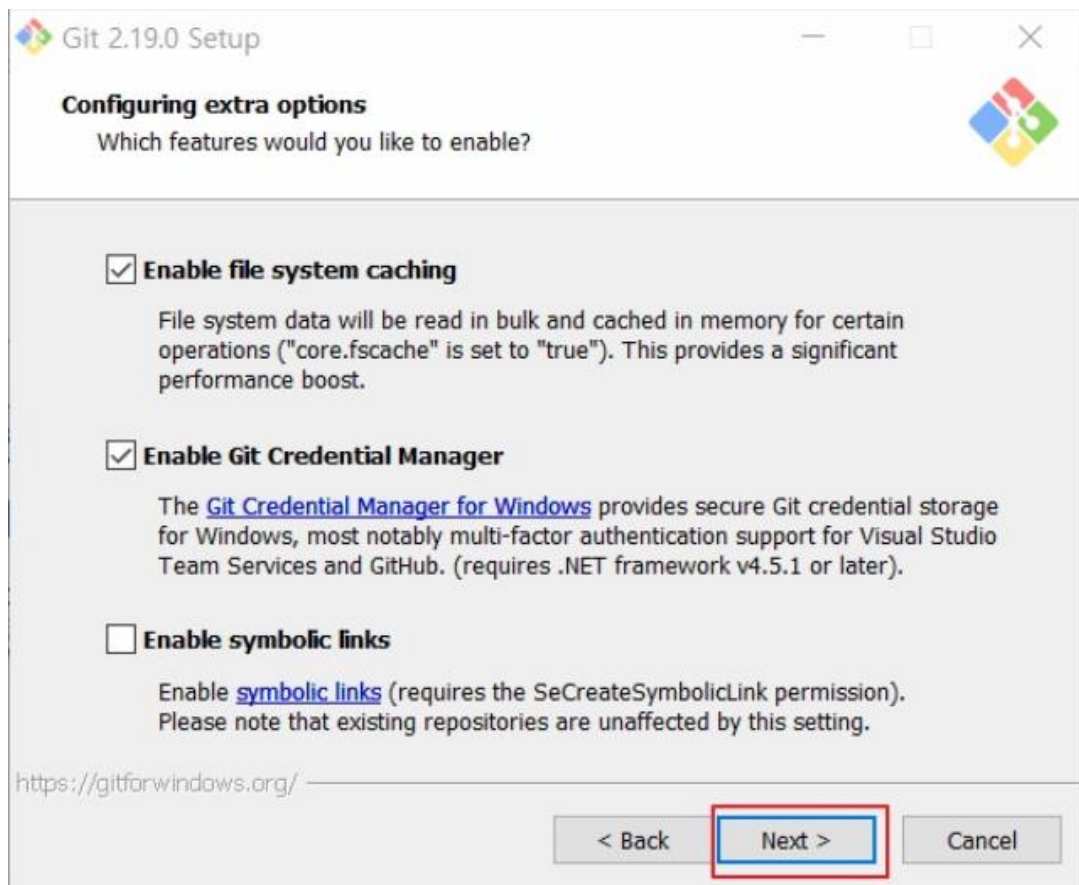
☐ **Use Windows' default console window**

Git will use the default console window of Windows ("cmd.exe"), which works with Win32 console programs such as interactive Python or node.js, but has a very limited default scroll-back, needs to be configured to use a Unicode font in order to display non-ASCII characters correctly, and prior to Windows 10 its window was not freely resizable and it only allowed rectangular text selections.

<https://gitforwindows.org/>

< Back **Next >** Cancel

Install git



Git status

Working
Directory

편집된 파일이 저장되어 있음

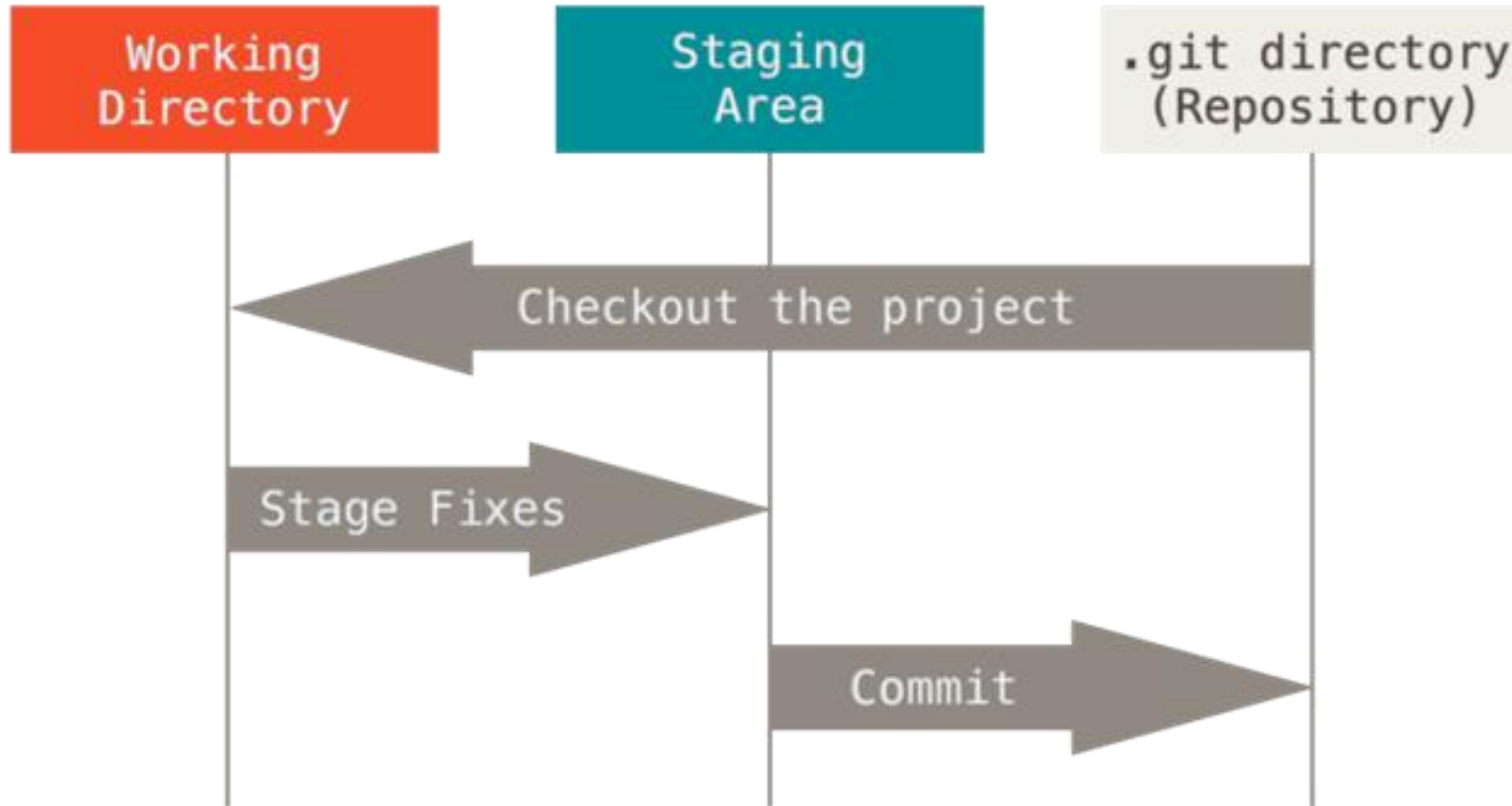
Staging
Area

Repository 로 변경내역을 저장하기 위한 파일들의 목록

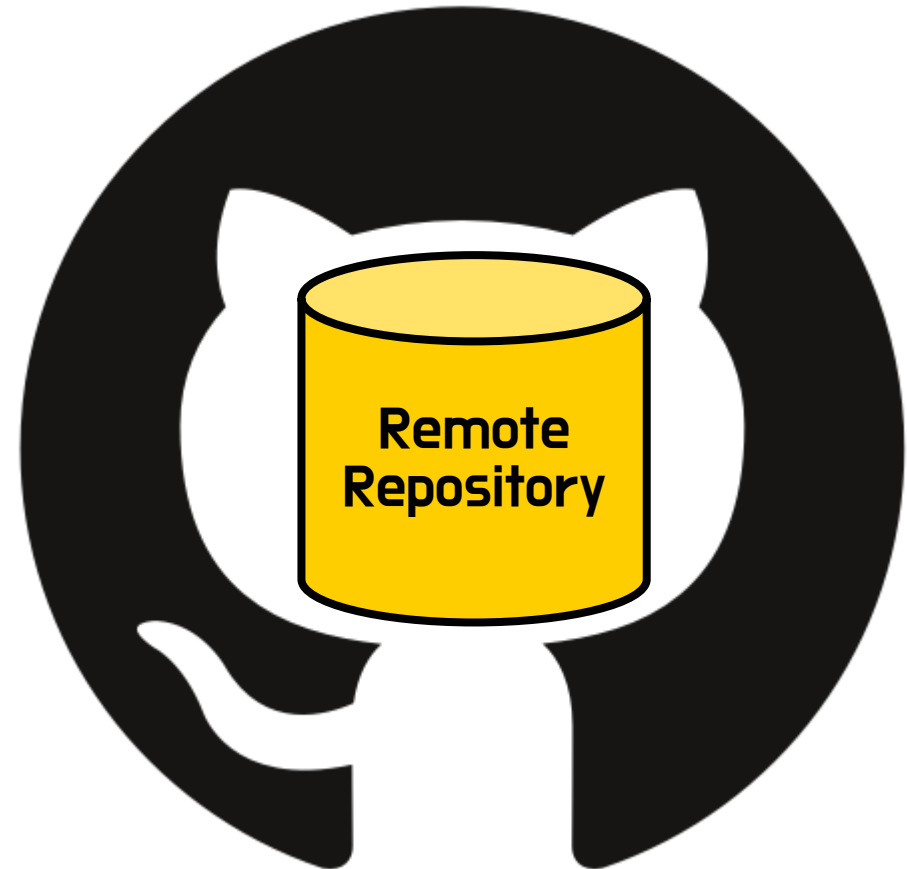
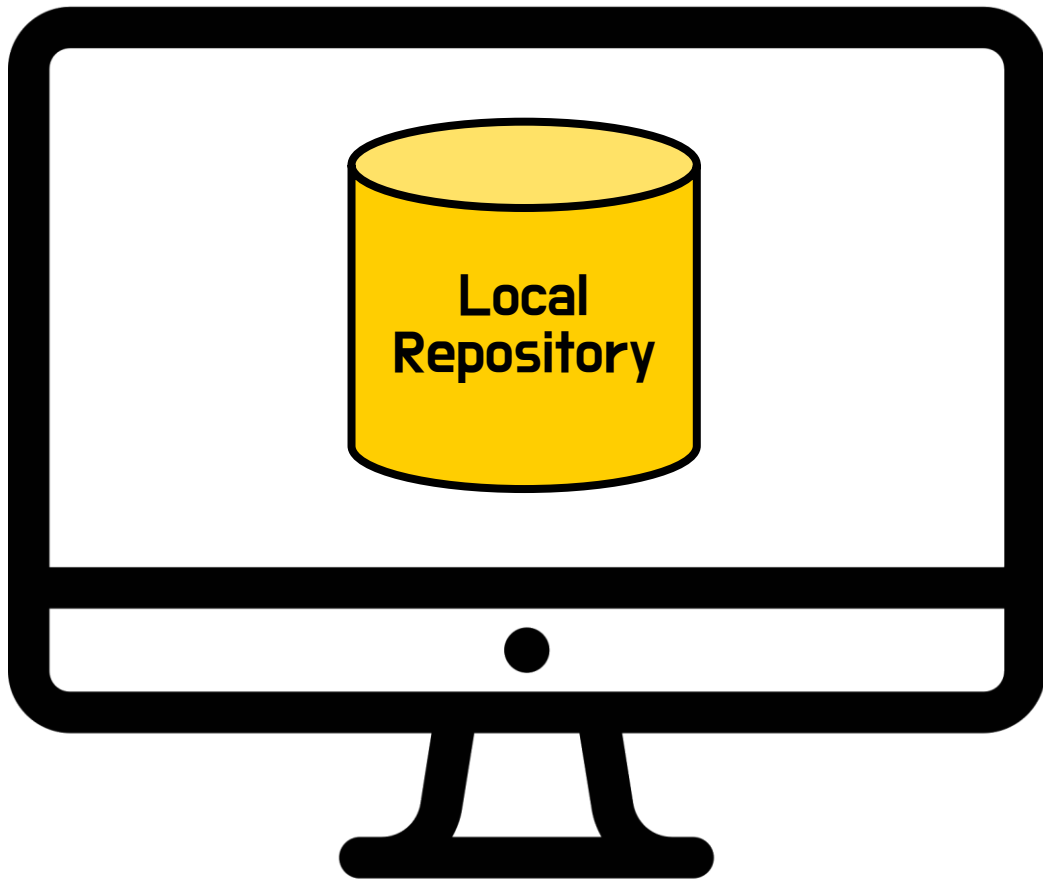
.git directory
(Repository)

변경 내역이 저장되는 곳

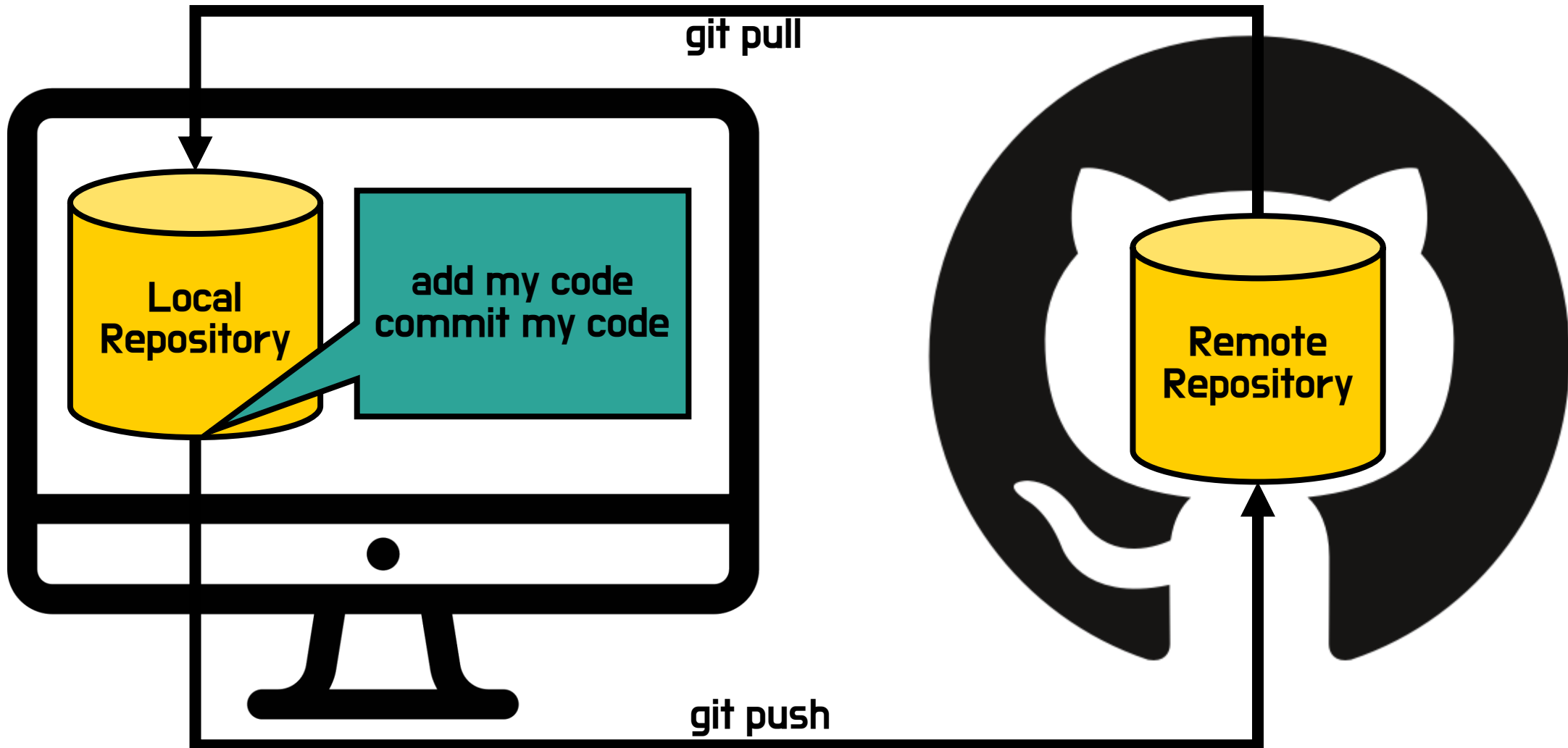
Git status



Local & Remote Repository



Workflow



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

.gitignore	Initial commit	2 months ago
LICENSE	Initial commit	2 months ago
README.md	Initial commit	2 months ago

README.md

git_test

To my git testing Repository

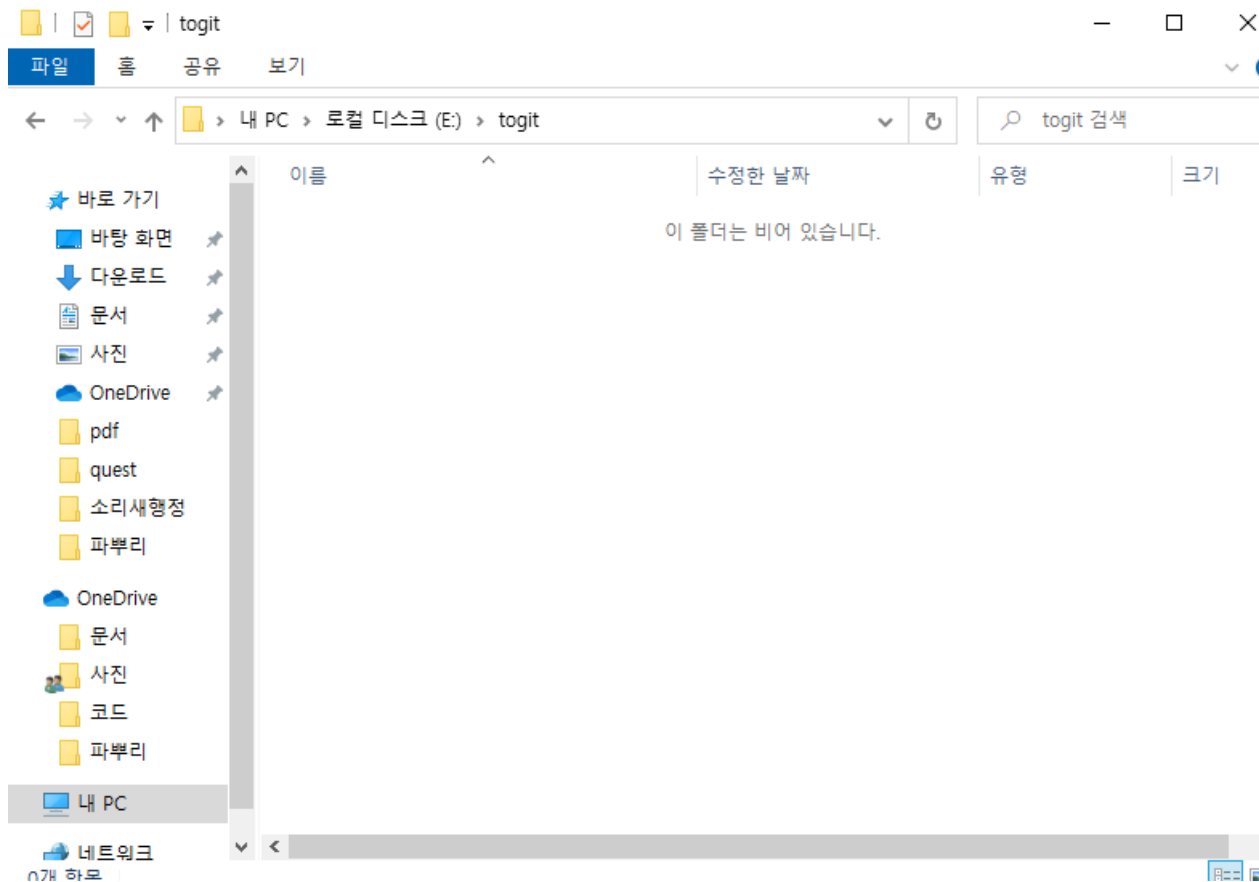
Git settings

- `git config --global user.name [자기 이름]`
- `git config --global user.email [자기 이메일]`

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



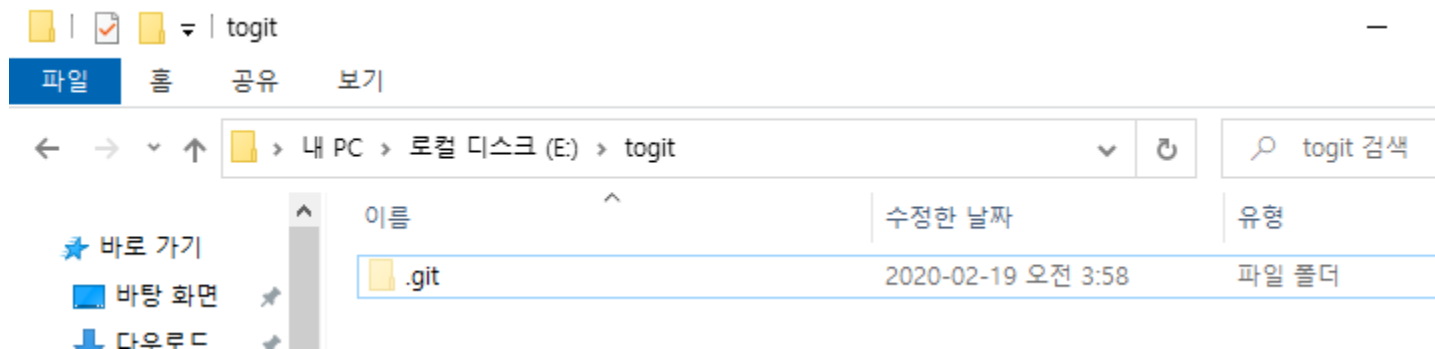
make a directory (anywhere)

Git training

명령 프롬프트

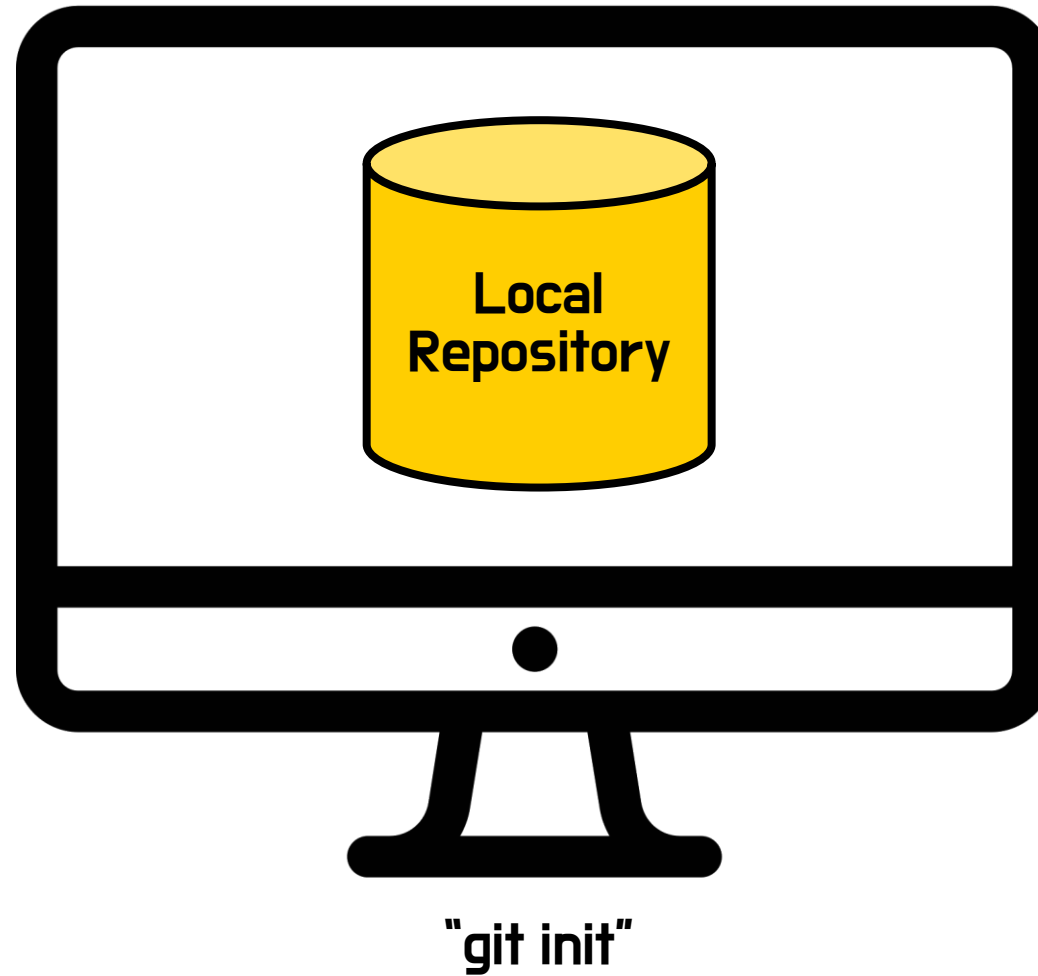
```
E:\wtogit>git init
Reinitialized existing Git repository in E:/togit/.git/
E:\wtogit>
```

Go to directory & "git init"

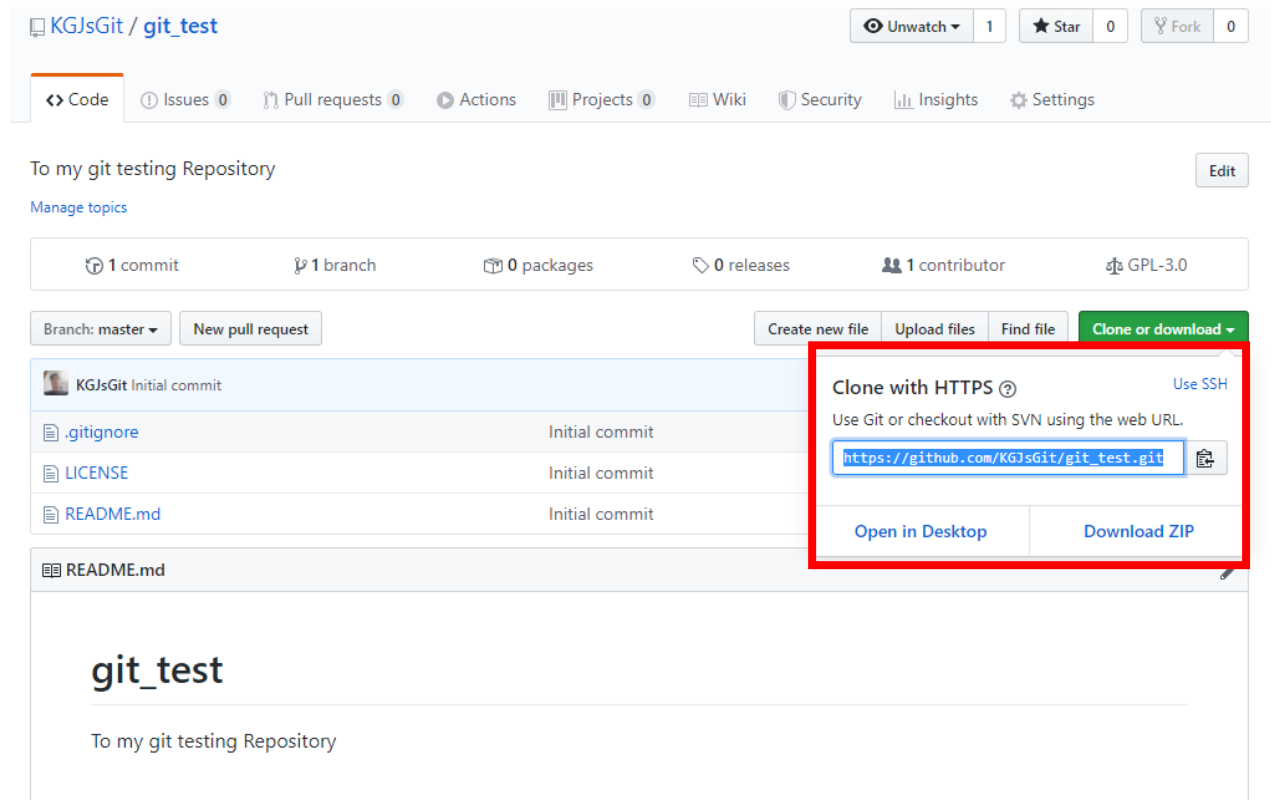


Check your local repository

Git training



Git 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

.gitignore	Initial commit
LICENSE	Initial commit
README.md	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

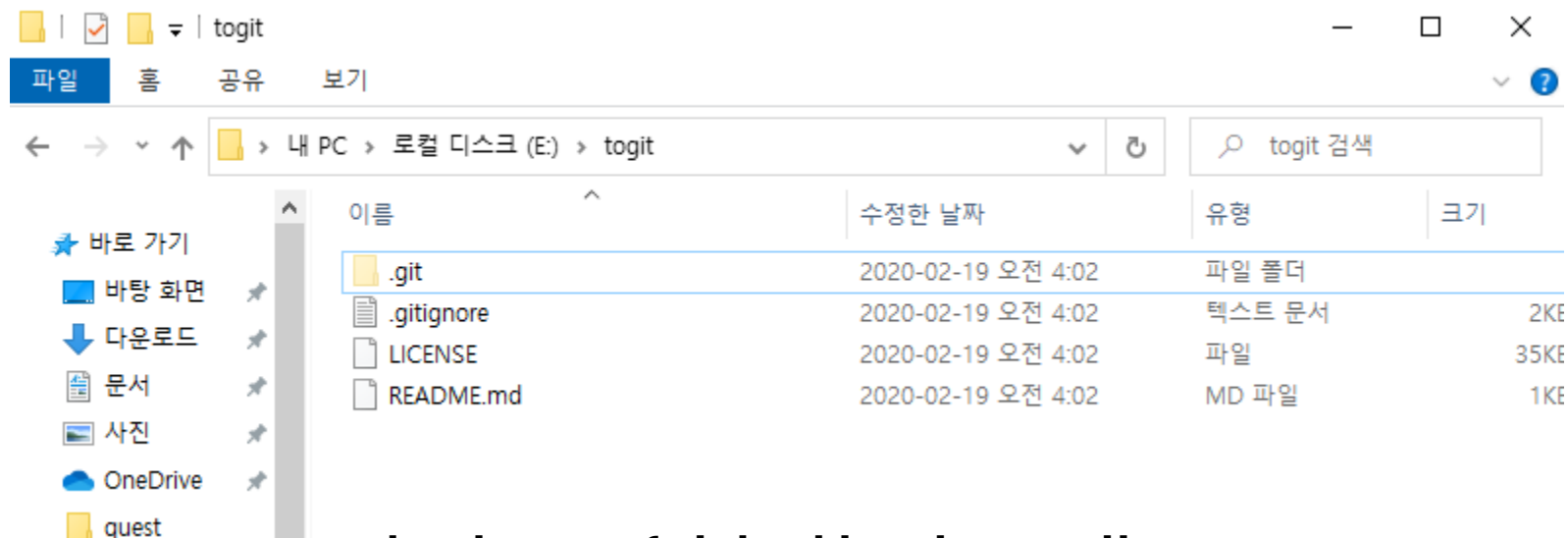
[Open in Desktop](#) [Download ZIP](#)

Copy the remote repository URL

Git training

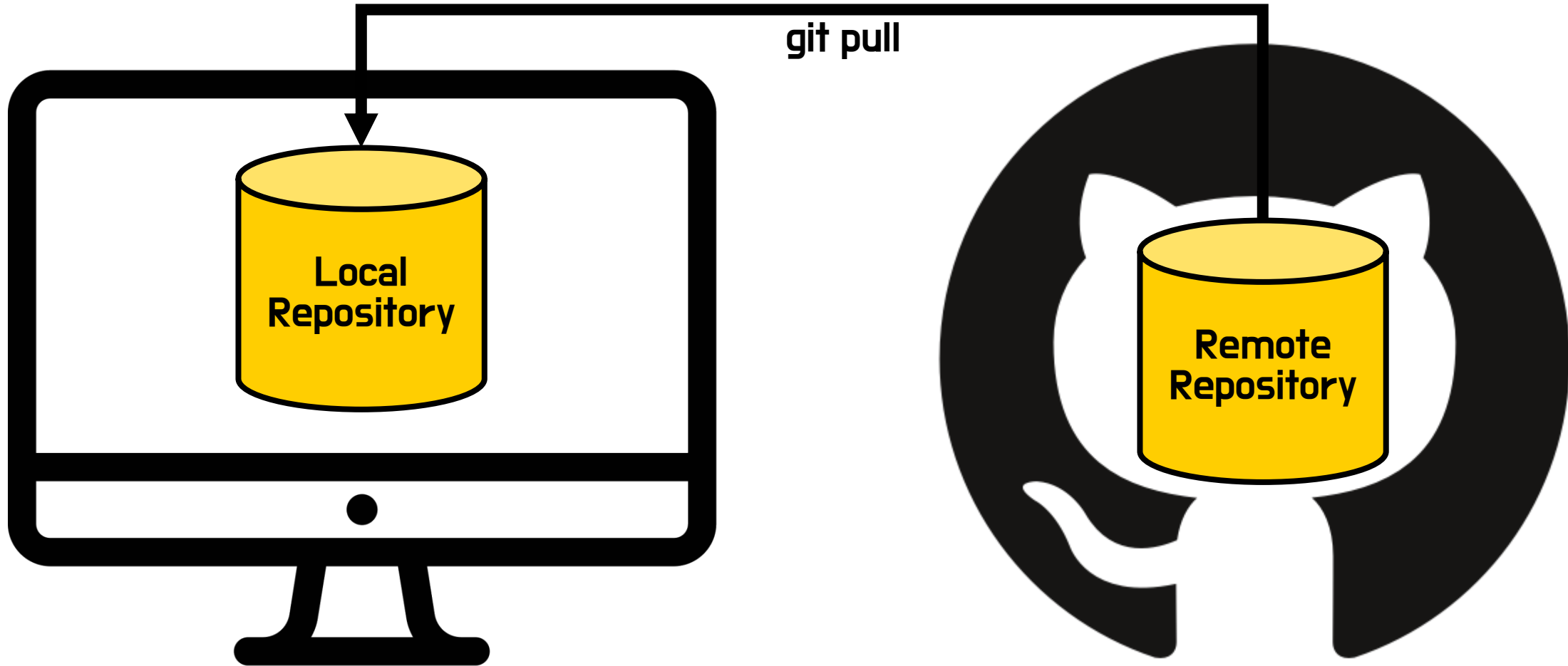
```
E:\wtogit>git pull https://github.com/KGJsGit/git_test.git
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (5/5), 13.45 KiB | 57.00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

"git pull 'remote repository URL' "



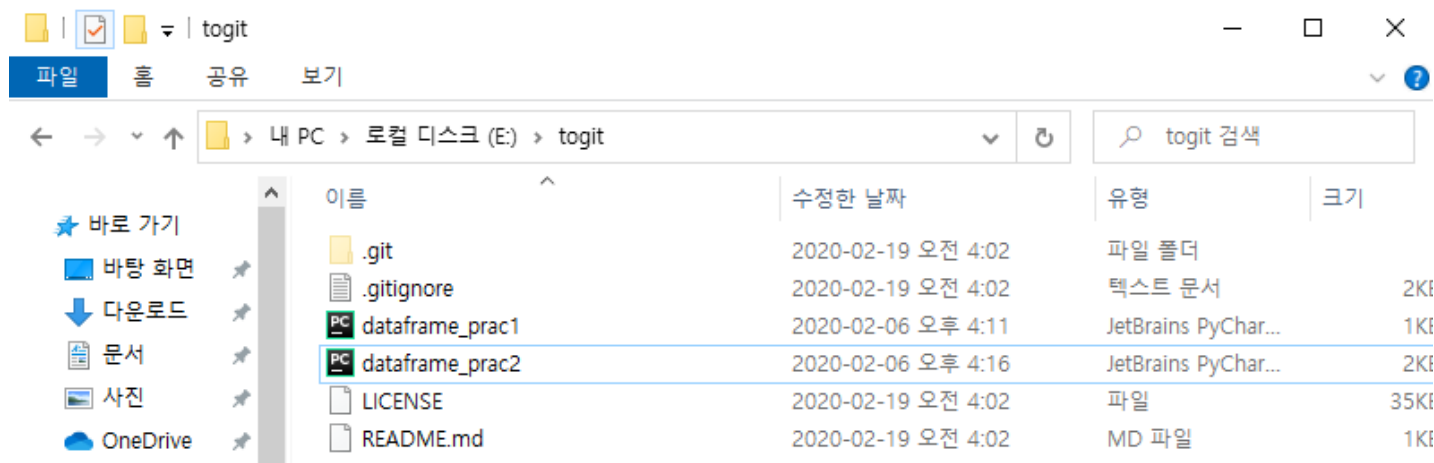
check your fetched local repository

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
```

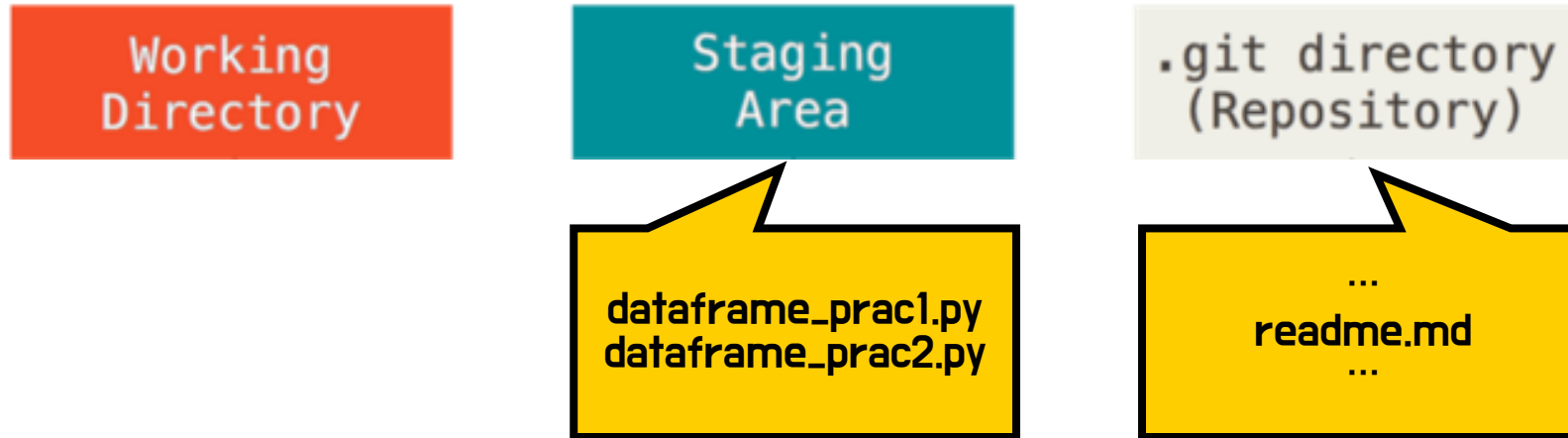
"git add 'file name' "

```
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

```
E:\wtogit>git commit -m "add new files"  
[master e7c7219] add new files  
2 files changed, 56 insertions(+)  
create mode 100644 dataframe_prac1.py  
create mode 100644 dataframe_prac2.py
```

"git commit -m "Add new files"
(commit message는 명료하고 보기 좋게 쓸 것)

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

check the git status. "git status"

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

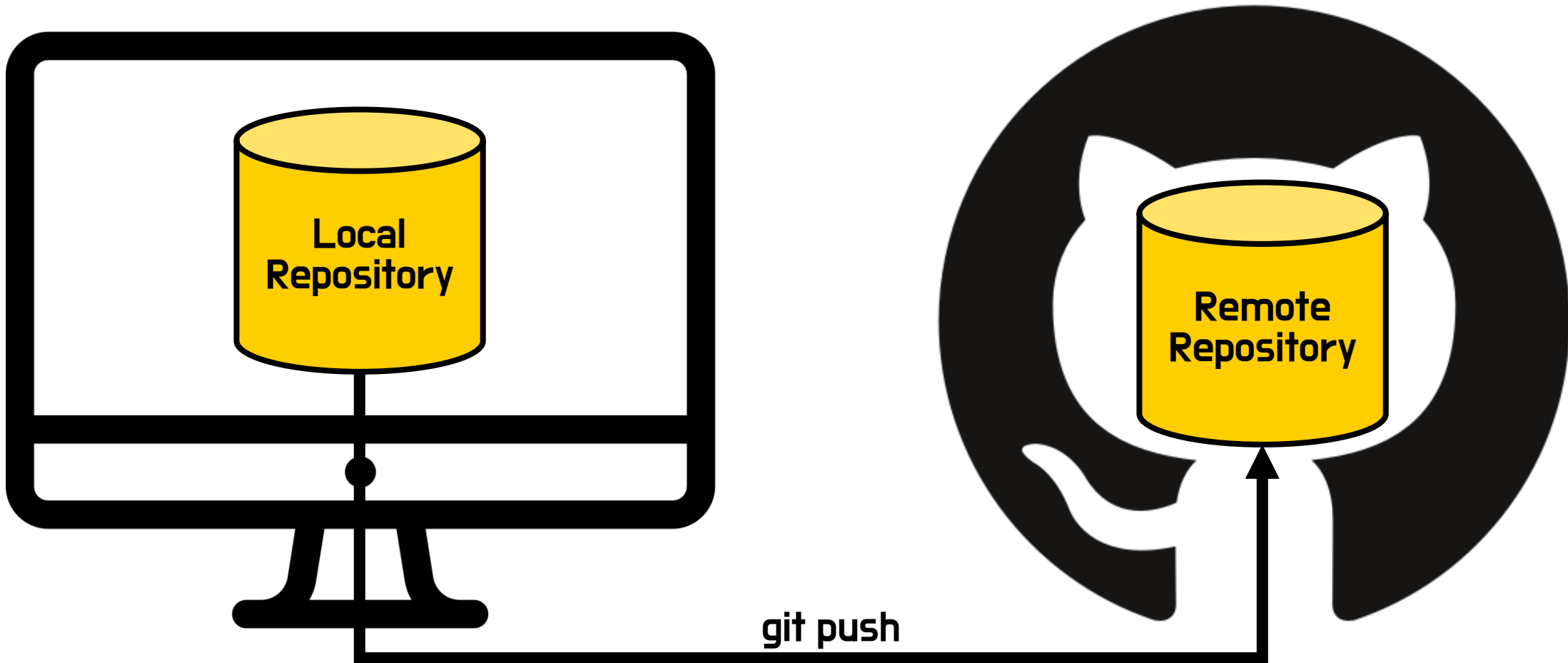
```
E:\wtogit>git remote add origin https://github.com/KGJsGit/git_test.git
```

"git remote add origin 'remote repository URL'"
(push할 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

.gitignore	Initial commit	2 months ago
LICENSE	Initial commit	2 months ago
README.md	Initial commit	2 months ago
dataframe_prac1.py	add new files	6 minutes ago
dataframe_prac2.py	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'

: git의 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 바뀜 줌

- git reset hard 'commit num'

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


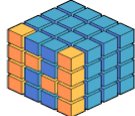



《 Round 4 》

- Git과 Github - complete
- 가상환경 《
- Jupyter Lab

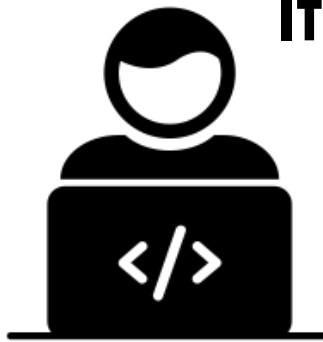


Let's
Go



Data Science Project_1 =  +    

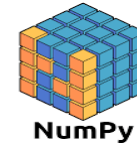
It's OK, but...



Data Science Project_1 =



+



SciPy

Pandas



Seaborn

Deep Learning Project_1 =



+



TensorFlow

matplotlib

Data Science Project_2 =



+

BeautifulSoup



K Keras

⋮

So sad...





- Anaconda는 파이썬의 배포판
- 특정 용도의 소프트웨어 패키지를 묶어서 제공
 - 각종 가상환경 구축 가능






ANACONDA

가상 환경 1




ANACONDA

가상 환경 2




ANACONDA

가상 환경 3

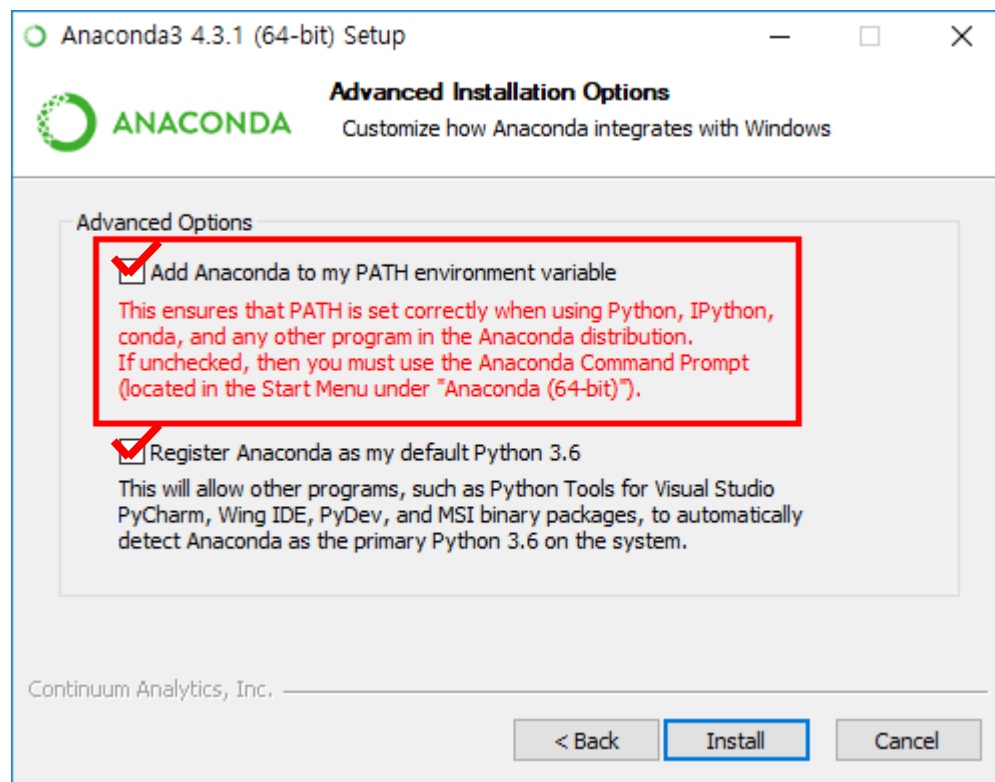


...

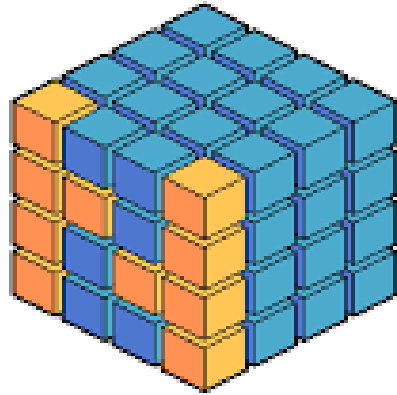
1. Go to

<https://www.anaconda.com/distribution/#download-section>

2. check this box



3. how to check my Anaconda is installed well?



NumPy

- 대표적인 파이썬 기반 선형대수 라이브러리
- 다양한 행렬과 배열 연산 함수 지원, 다양한 라이브러리에 기본 내장
- 데이터는 대부분 행렬의 형태를 띄기 때문에 데이터처리에 필수적

matplotlib

- 자료를 차트나 플롯으로 시각화(visualization)하는 패키지
- 판다스와 연계되어 있으며, EDA를 위한 시각화에 필수적임.



- 다양한 분석과 머신러닝 학습에 사용될 수 있는 파이썬 라이브러리
 - numpy와 Scipy와 상호 운영되도록 디자인 됨



- 데이터 분석 및 조작을 위한 라이브러리
- R과 비슷한 자료형을 사용함(Dataframe)

**이렇게 다양한 라이브러리를
하나하나 관리해야 한다면...?**



So sad...



- 파이썬 라이브러리 패키지를 손쉽게 관리 할 수 있게 해주는 시스템
 - 파이썬 3.4이후 버전부터 기본 탑재!
 - 짧은 명령어 한 줄로 손쉽게 설치 가능

1. 패키지를 설치해줄 가상환경 활성화

```
C:\Users\User>conda activate test
```

2. pip install numpy로 numpy 설치

```
(test) C:\Users\User>pip install numpy
Collecting numpy
  Using cached https://files.pythonhosted.org/packages/a9/3f/numpy-1.18.1-cp37-cp37m-win_amd64.whl
Installing collected packages: numpy
Successfully installed numpy-1.18.1
```

3. pip freeze로 현재 설치된 패키지 확인

```
(test) C:\Users\User>pip freeze
certifi==2019.11.28
numpy==1.18.1
winertstore==0.2
```

《 Round 4 》

- Git과 Github - complete
- 가상환경 - complete
- Jupyter Lab 《



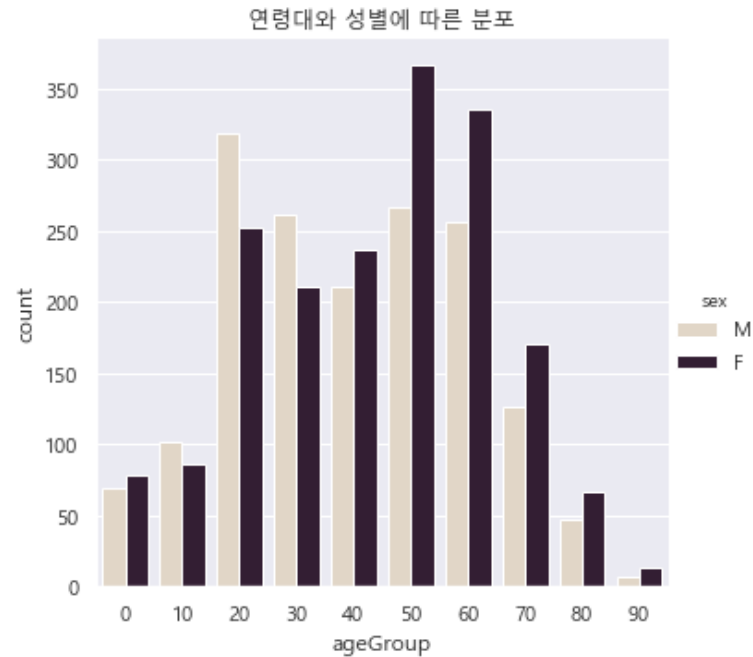
Let's
Go



Jupyter Lab의 장점





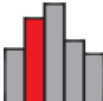


- 코드를 블록 단위로 실행시킬 수 있고 결과에 코멘트를 달 수 있음.

```
sns.catplot(x="ageGroup", kind="count", hue="sex", palette="ch:.25", data=cov_df)
plt.title("연령대와 성별에 따른 분포")
plt.show()
```



- 20대와 50대, 60대가 월등히 많았다.
- 내 생각엔 20대는 놀기 위해서 많이 돌아다녔기 때문이고, 50~60대는 교회때문이 아닐지??

Applications on base (root) Channels

 CMD.exe Prompt 0.1.1 Run a cmd.exe terminal with your current environment from Navigator activated Launch	 JupyterLab 1.1.4 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Launch	 Jupyter Notebook 6.0.1 Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis. Launch	 Powershell Prompt 0.0.1 Run a Powershell terminal with your current environment from Navigator activated Launch
 Glueviz 0.15.2 Multidimensional data visualization across files. Explore relationships within and among related datasets. Install	 Orange 3 3.26.0 Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox. Install	 RStudio 1.1.456 A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks. Install	

Notebook



Console



Shift + Enter

Untitled.ipynb



Code



```
[1]: print("Hello world")
```

Hello world

Untitled.ipynb



Markdown



```
[1]: print("Hello world")
```

Hello world

첫 주피터 노트북 사용하기

- Hello world 출력

Untitled.ipynb



Markdown



```
[1]: print("Hello world")
```

Hello world

첫 주피터 노트북 사용하기

- Hello world 출력

Jupyter의 확장자는 .ipynb

여러 탭을 띄울 수 있고 여러 창을 띄울 수 있음

Directory

The screenshot shows the JupyterLab interface with the following components:

- Directory (Left Panel):** A file browser showing a directory structure. The path `/ ... / 공모전 / 경기코로나예측` is highlighted. The file `cocoa_1.ipynb` is selected.
- Code Editor (Center Panel):** The `cocoa_1.ipynb` file is open, showing Python code for data analysis and plotting. The code includes imports for `pandas`, `numpy`, `matplotlib.pyplot`, `matplotlib`, `seaborn`, `datetime`, `folium`, and `json`. It also sets the plot style to `darkgrid` and configures the font and figure size.
- Table (Right Panel):** A table titled `확진일-증상발현일` is displayed. It has two columns: `확진일` and `증상발현일`. The table contains 20 rows of data.

Red lines indicate the following connections:

- A line from `Jupyter의 확장자는 .ipynb` points to the `cocoa_1.ipynb` file in the directory and the code editor tabs.
- A line from `여러 탭을 띄울 수 있고 여러 창을 띄울 수 있음` points to the multiple open tabs at the top of the interface.
- A line from `Directory` points to the file browser on the left.

Jupyter Tip

- 코드 작성 중 Tab을 누르면 code navigation 작동
- Jupyter kernel을 바탕화면이나 상태표시줄에 바로가기 등록 가능
- command/edit mode가 존재, 이를 이용해서 단축키를 활용한 코딩 가능
 - 필요한 패키지는 conda prompt를 켜고 pip install로 설치!
 - 저장(Ctrl + s)을 생활화 합시다...!