

# SKKU Overflow 전체 총회

개발자가 되기 위한 발걸음, 제 1차 GIT 교육



## 목 차

01 버전관리 시스템이란?

02 Git이란?

03 기본적인 사용법

- 몇가지 Linux 명령어
- git 기본 사용법
- 원격저장소 연결하기

04 Branch 사용하기

- Branch란?
- Merge & Rebase

00

사전준비

| vscode |



| git bash |



00

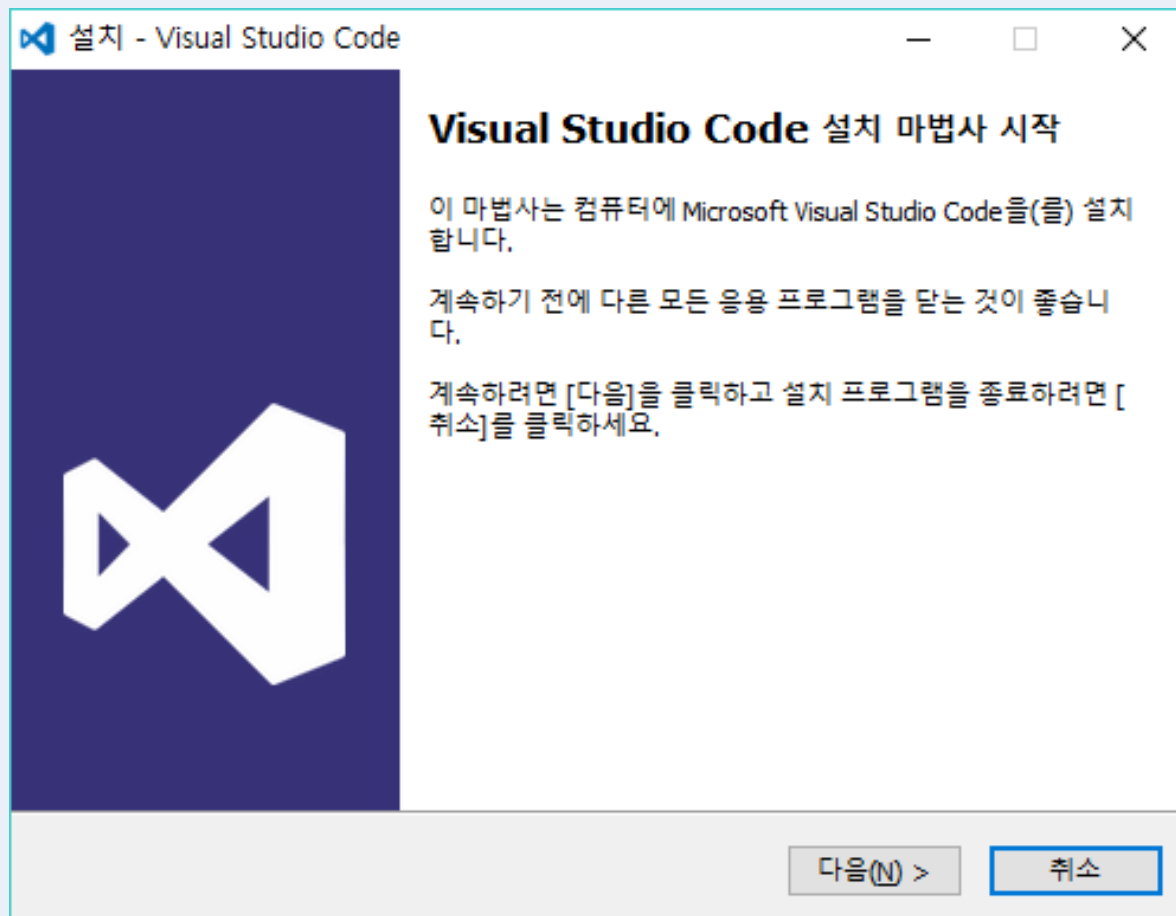
사전준비 - Visual Studio Code 설치

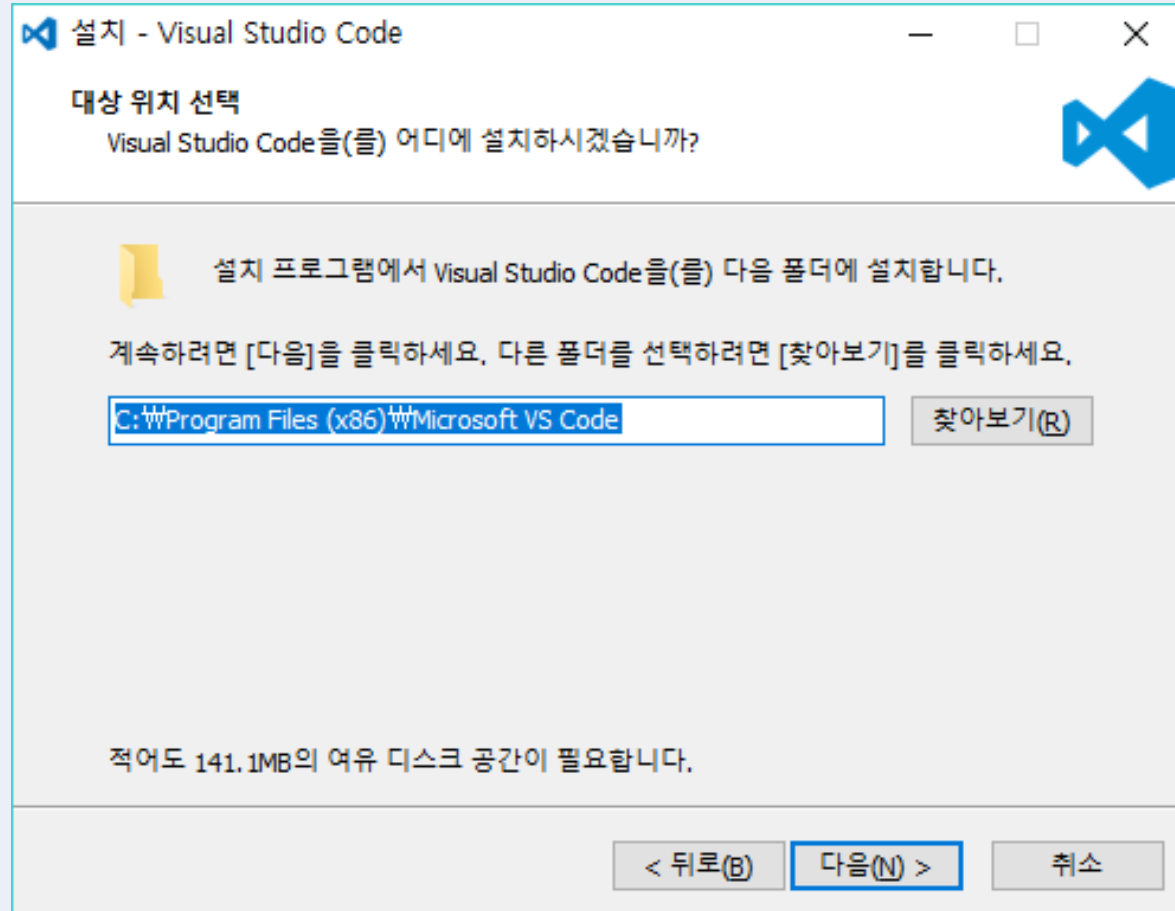


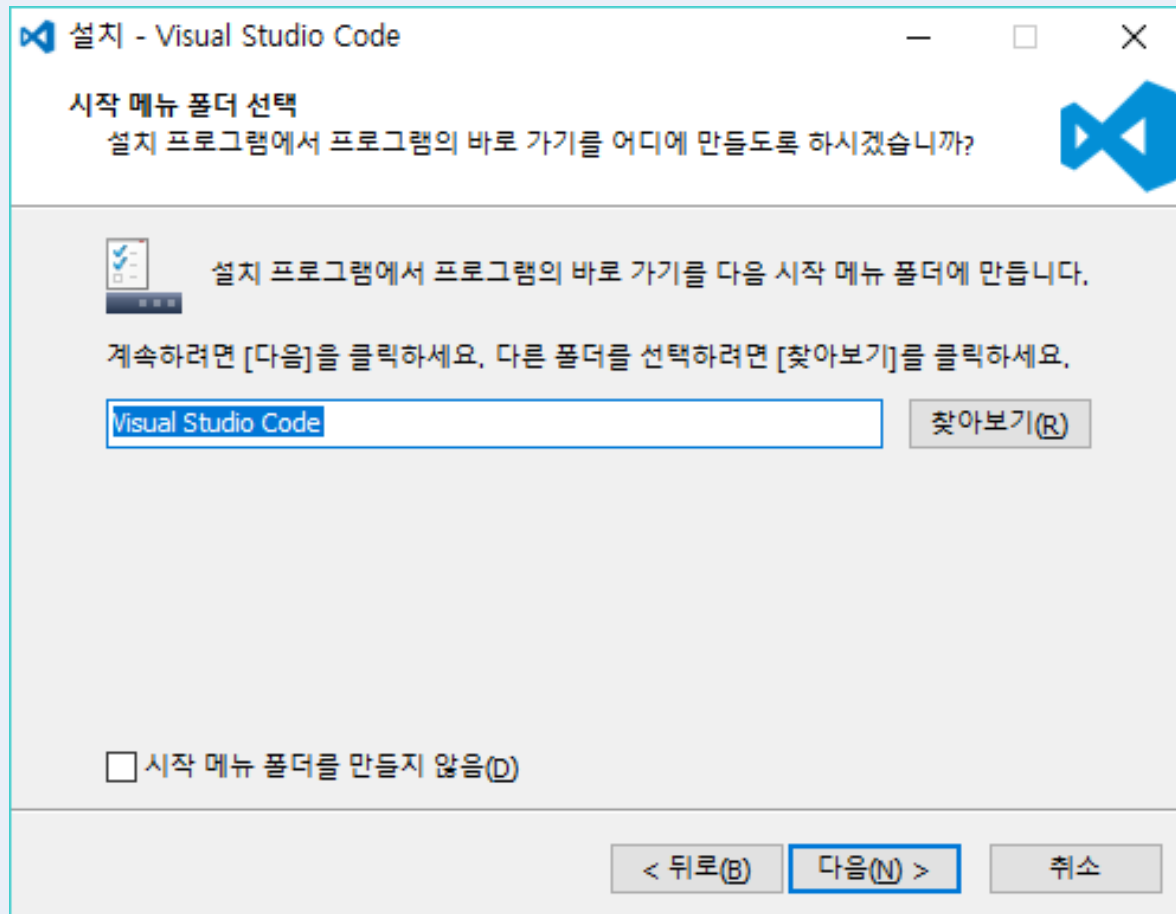
# Visual Studio Code

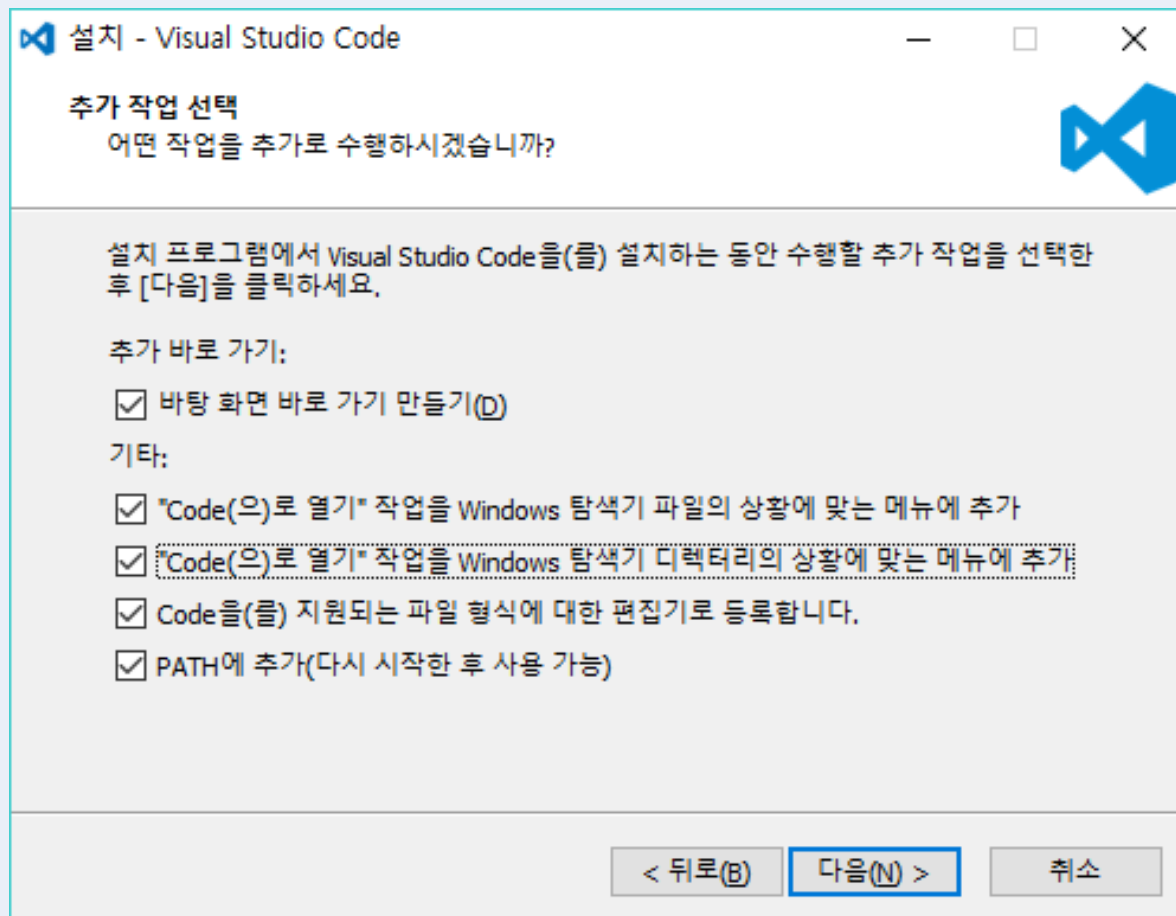
<https://code.visualstudio.com/download>













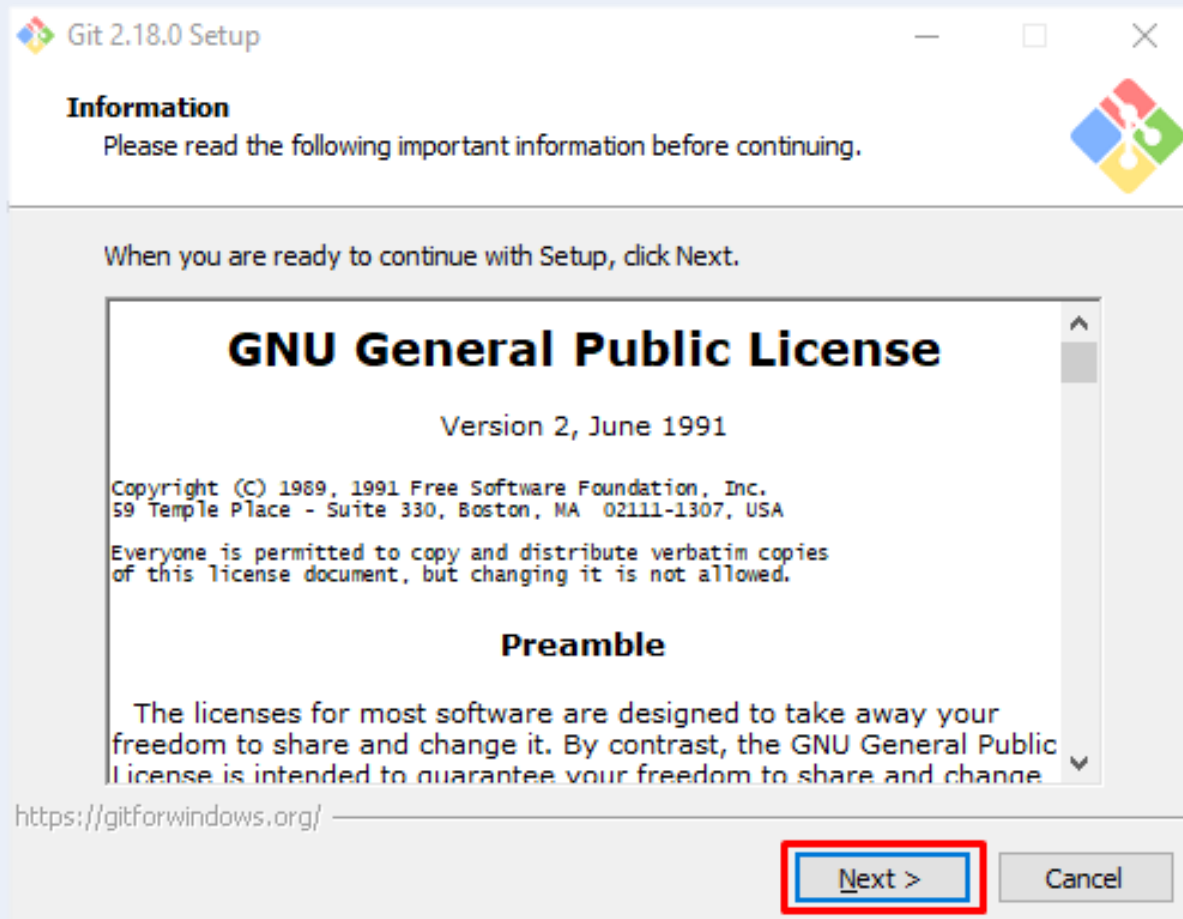
00

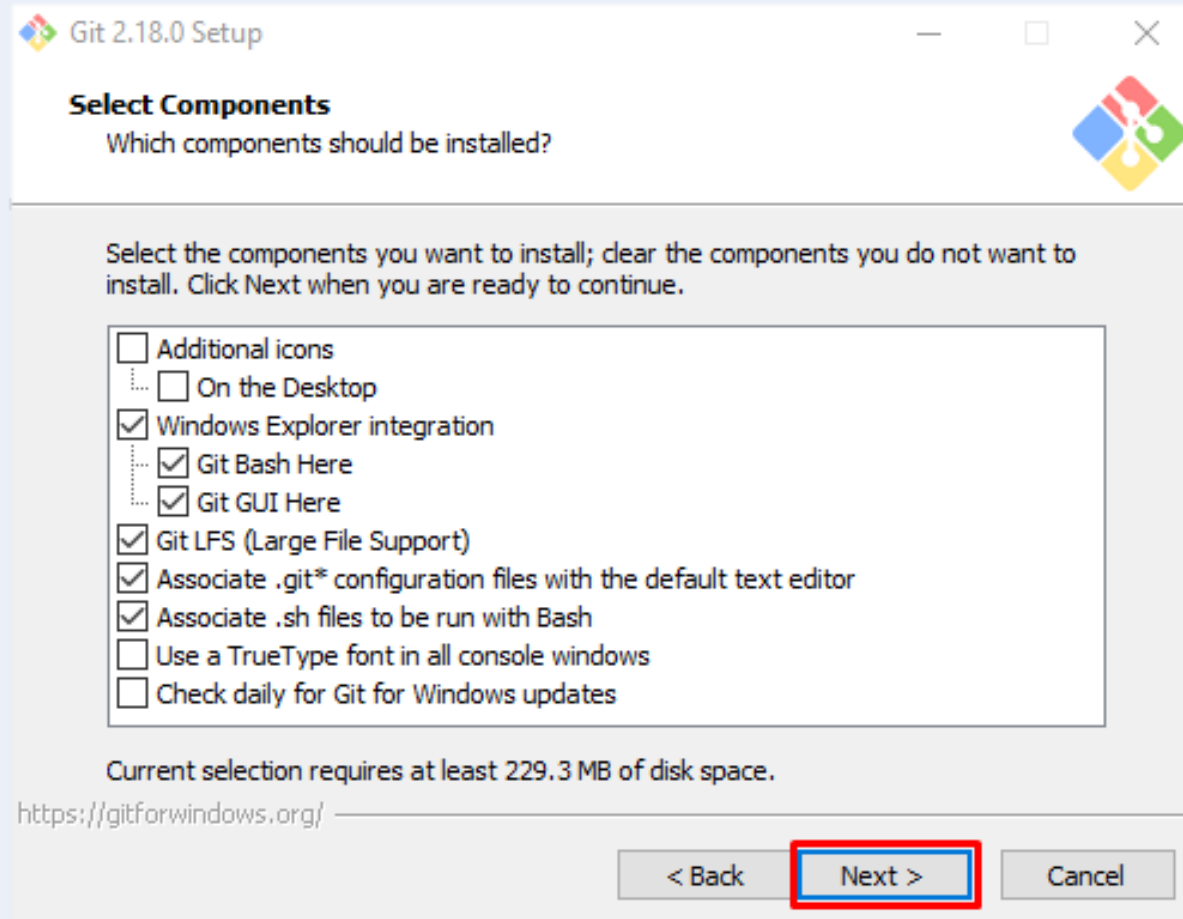
사전준비 - Git Bash 설치



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

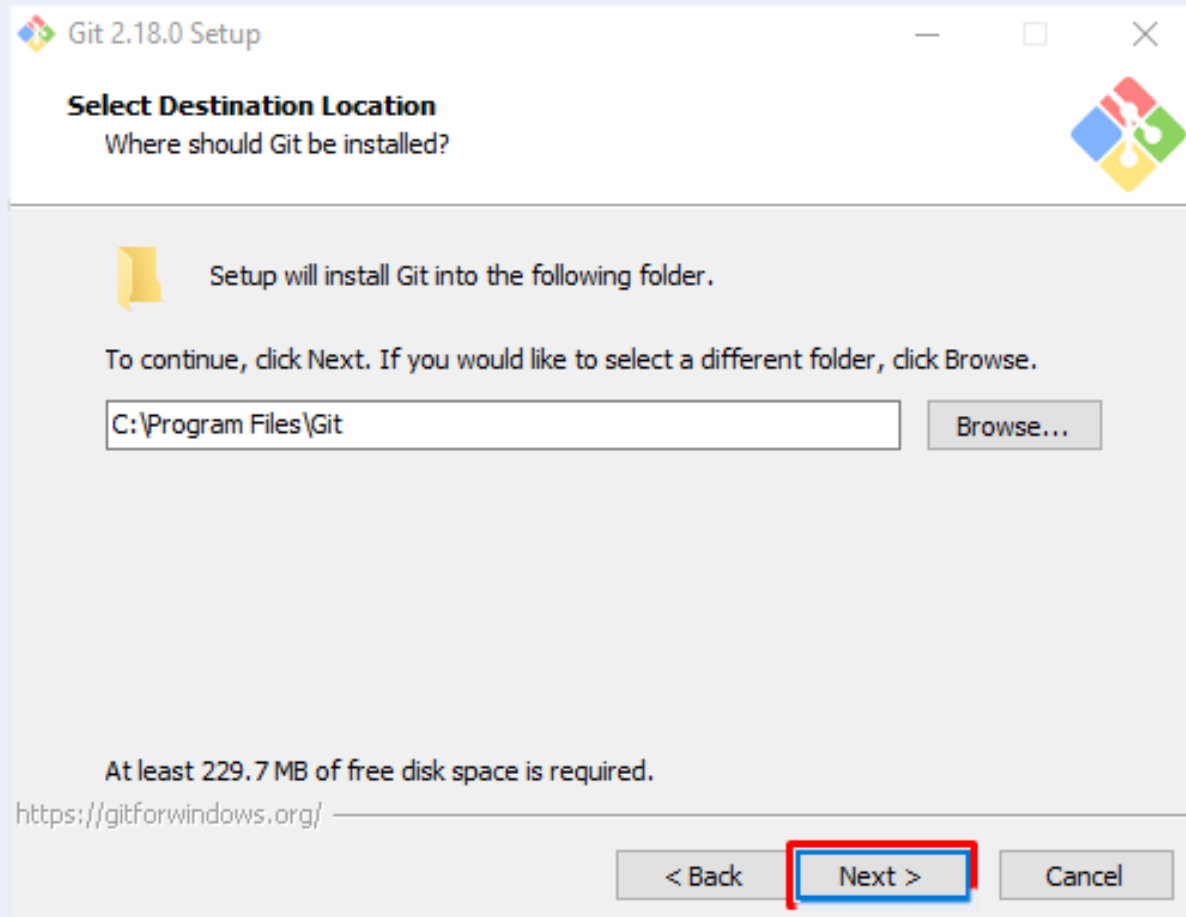




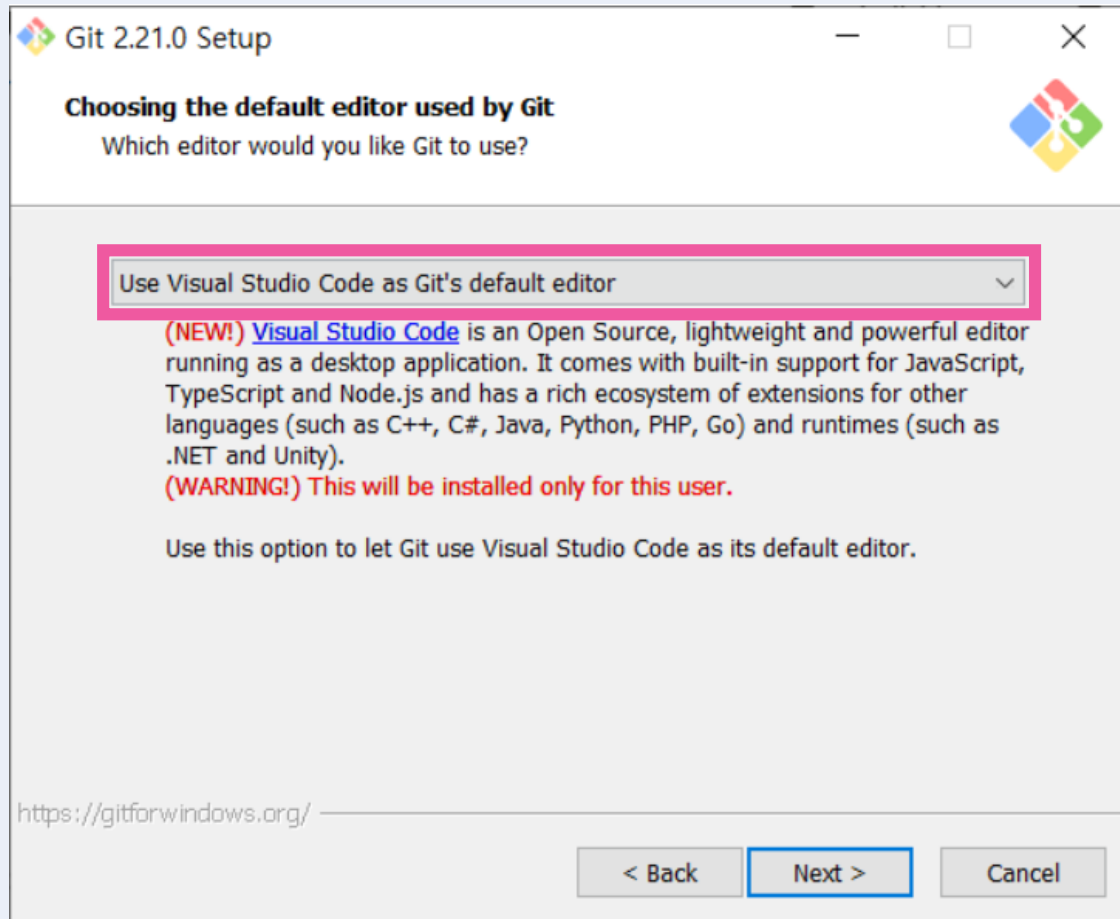


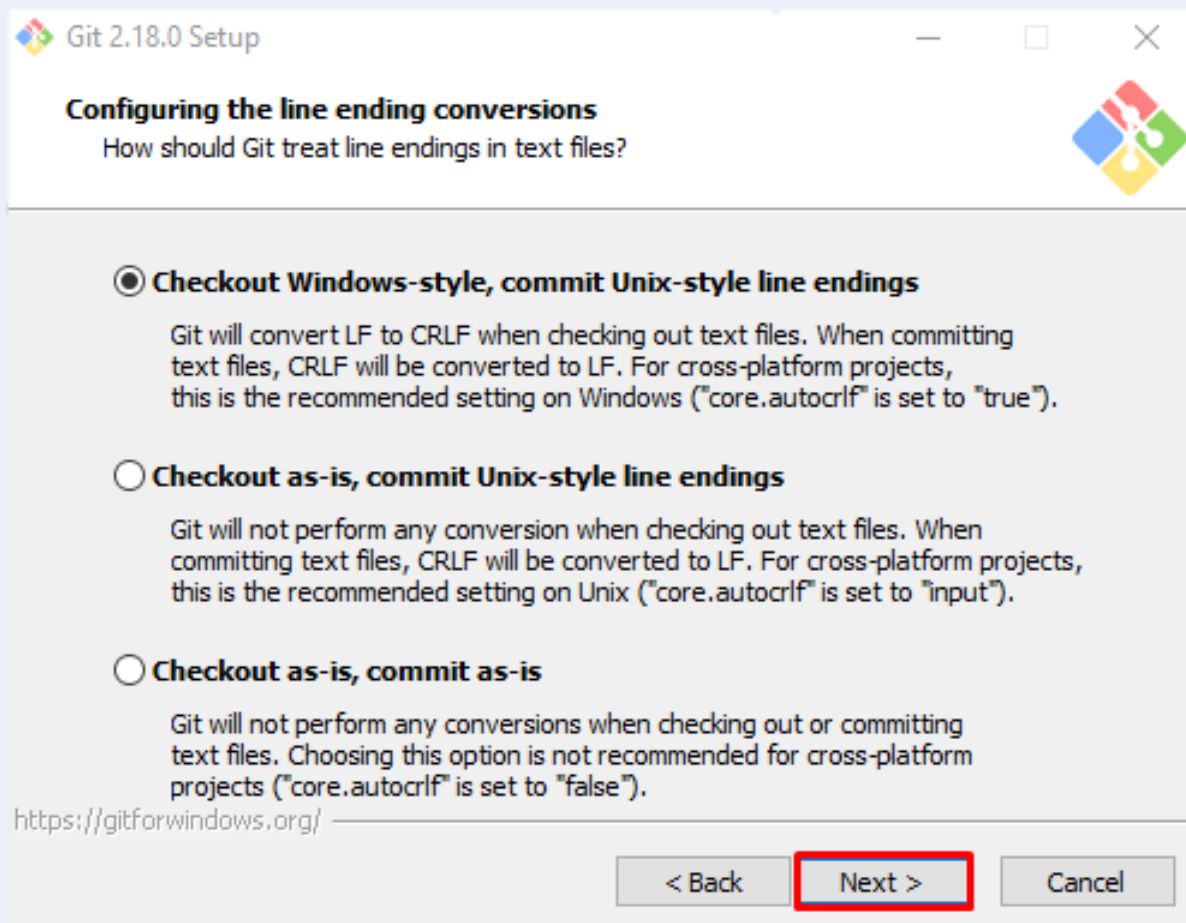
00

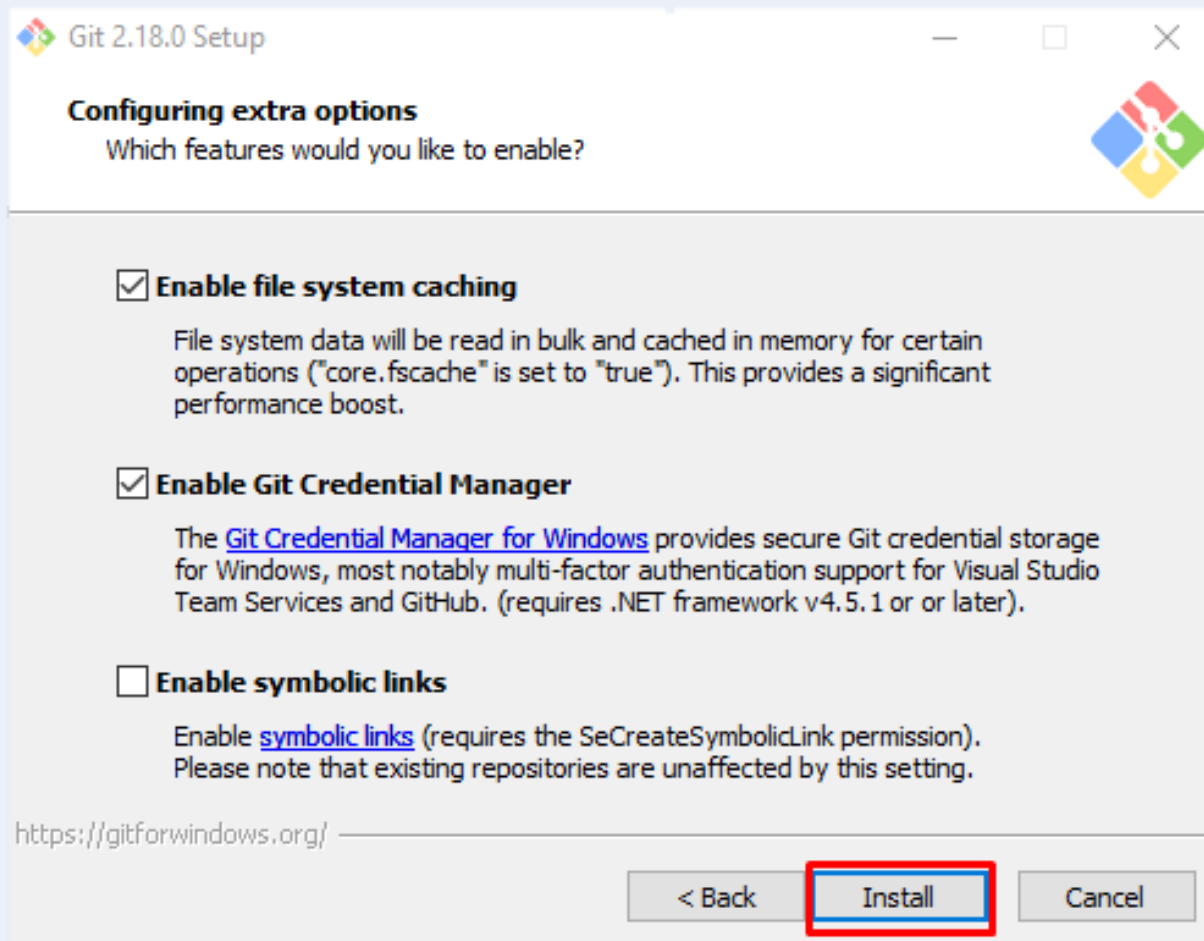
사전준비 - Git Bash 설치



Use Visual Studio Code as Git's default editor









설치할 때 VS Code를 Git 기본 editor로 설정하지 못한 분들을 위한 슬라이드

```
$ git config --global core.editor 'code --wait'
```

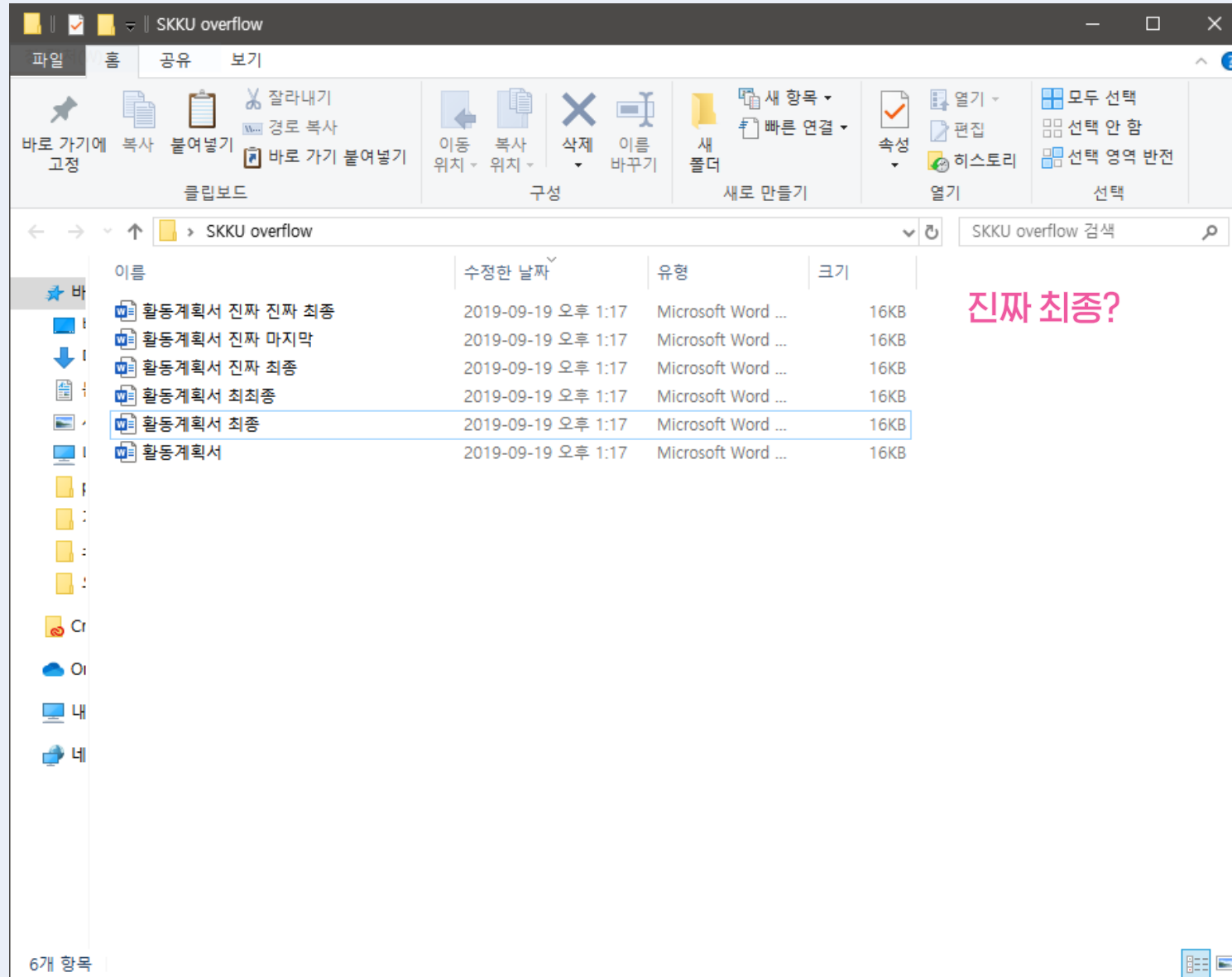
: VS Code를 Git 에디터로 사용하기

커밋 메시지 입력 시 : \$ git commit만 입력하고 Enter

Merge, Rebase 충돌 해결 인터페이스가 편리해짐

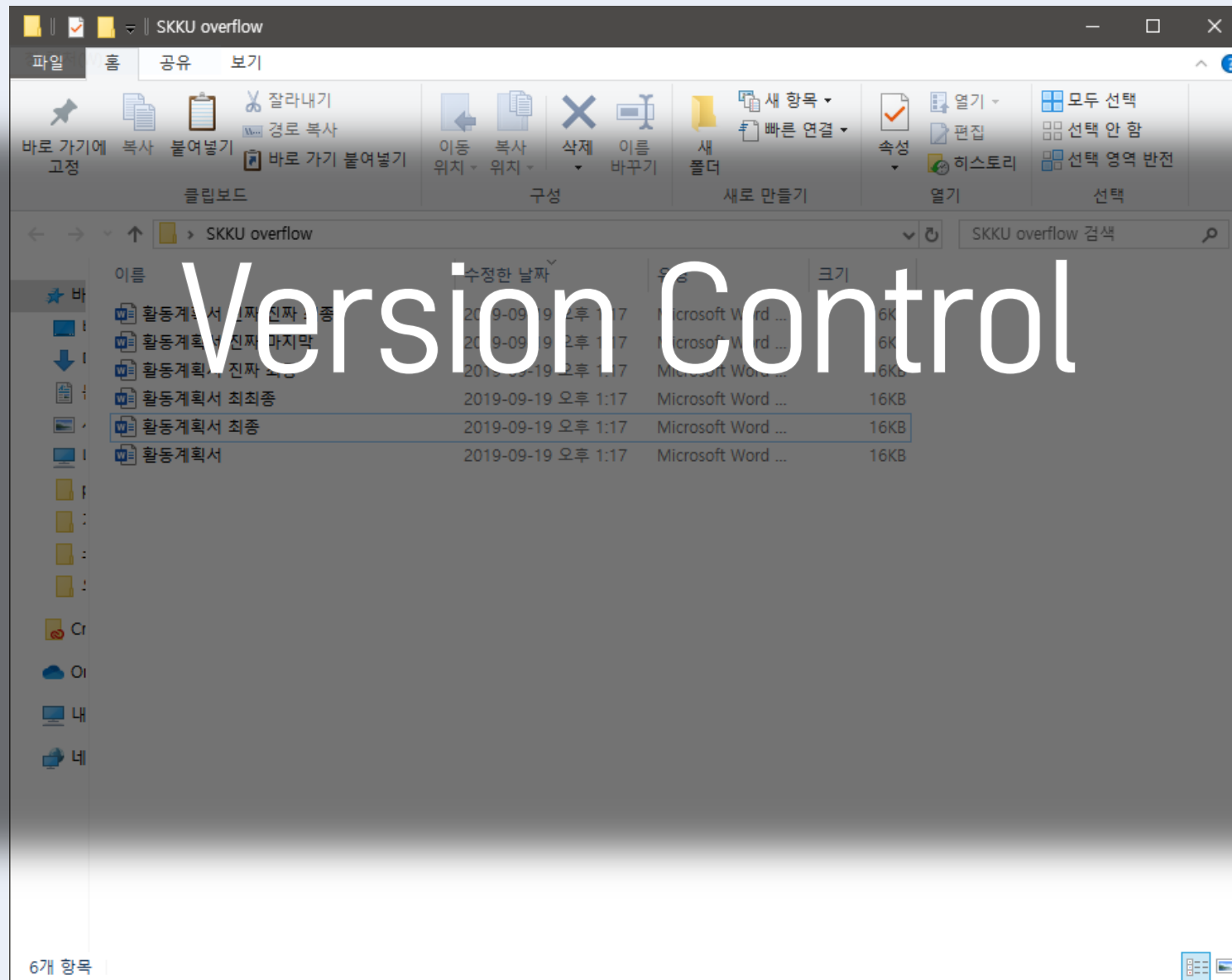






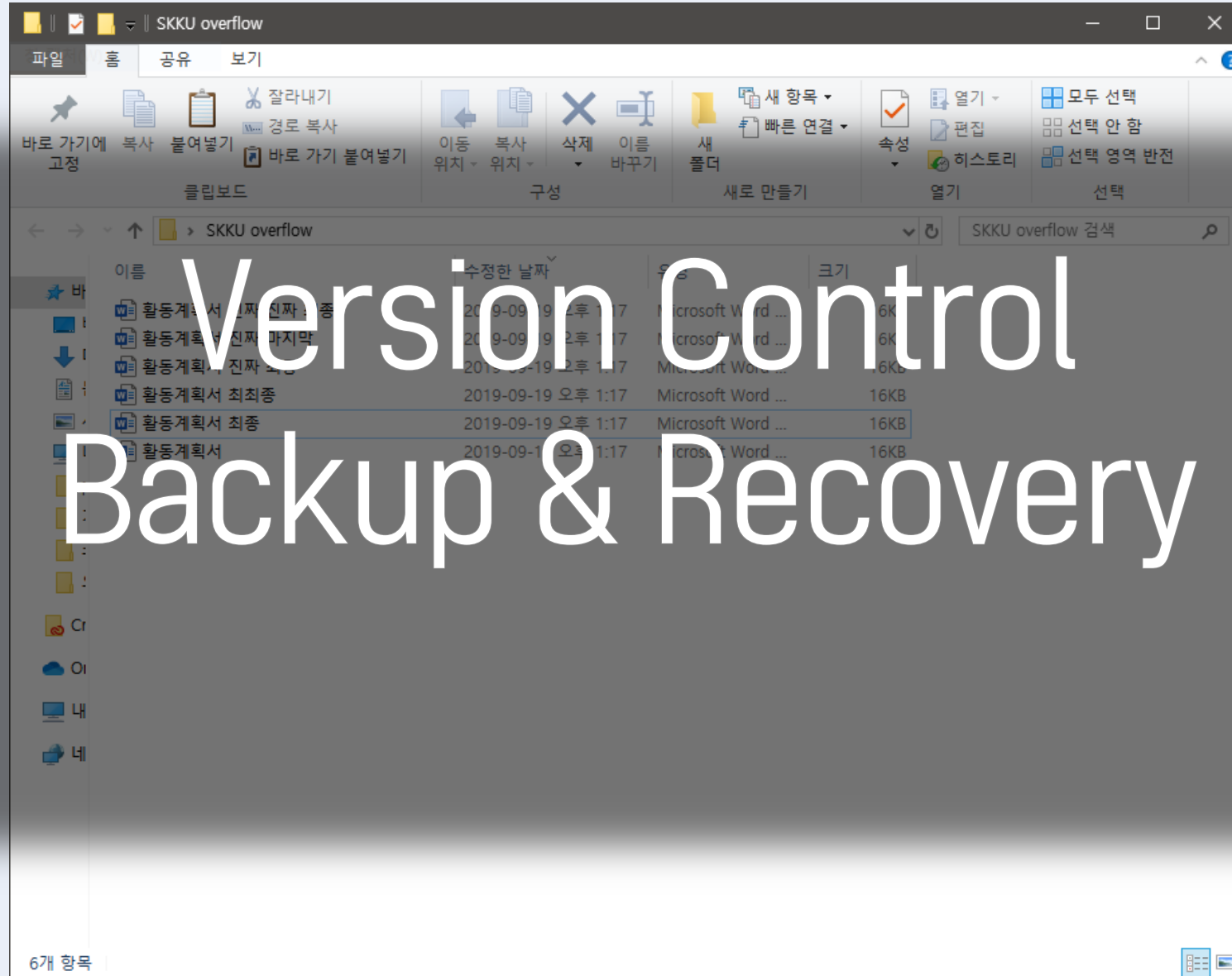
01

버전관리 시스템이란?



01

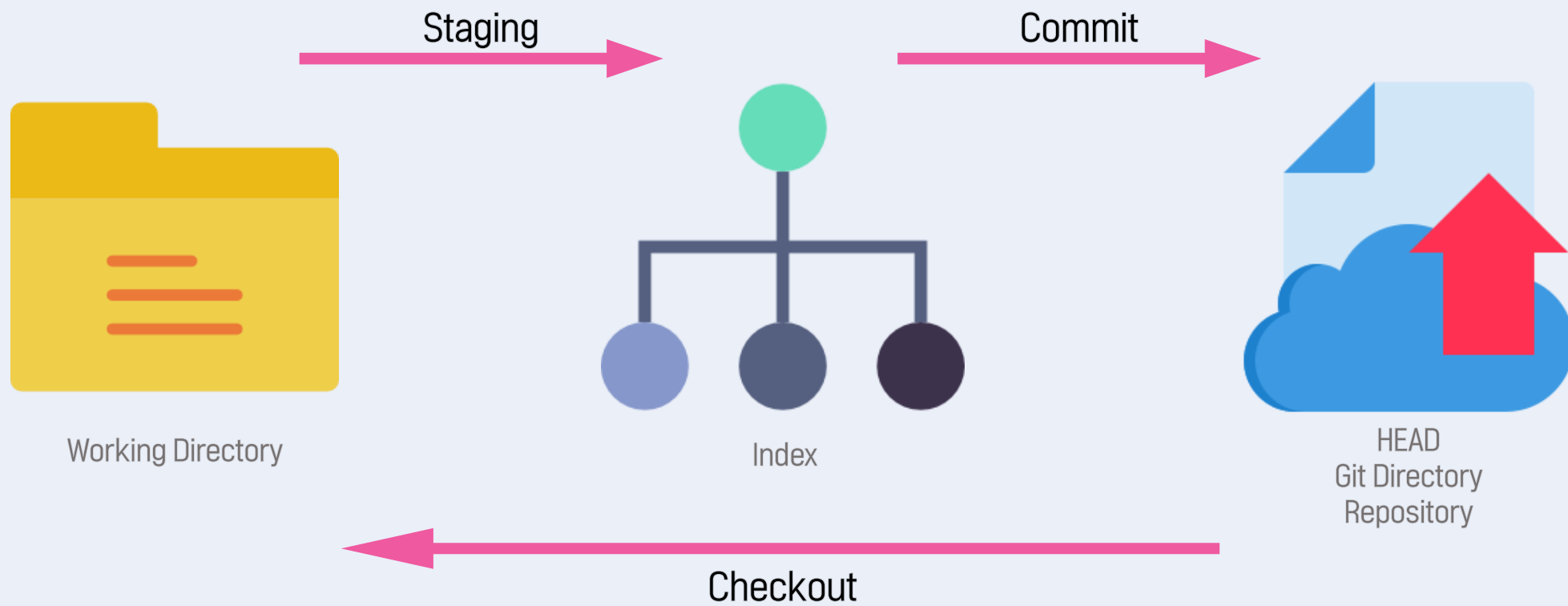
버전관리 시스템이란?



01

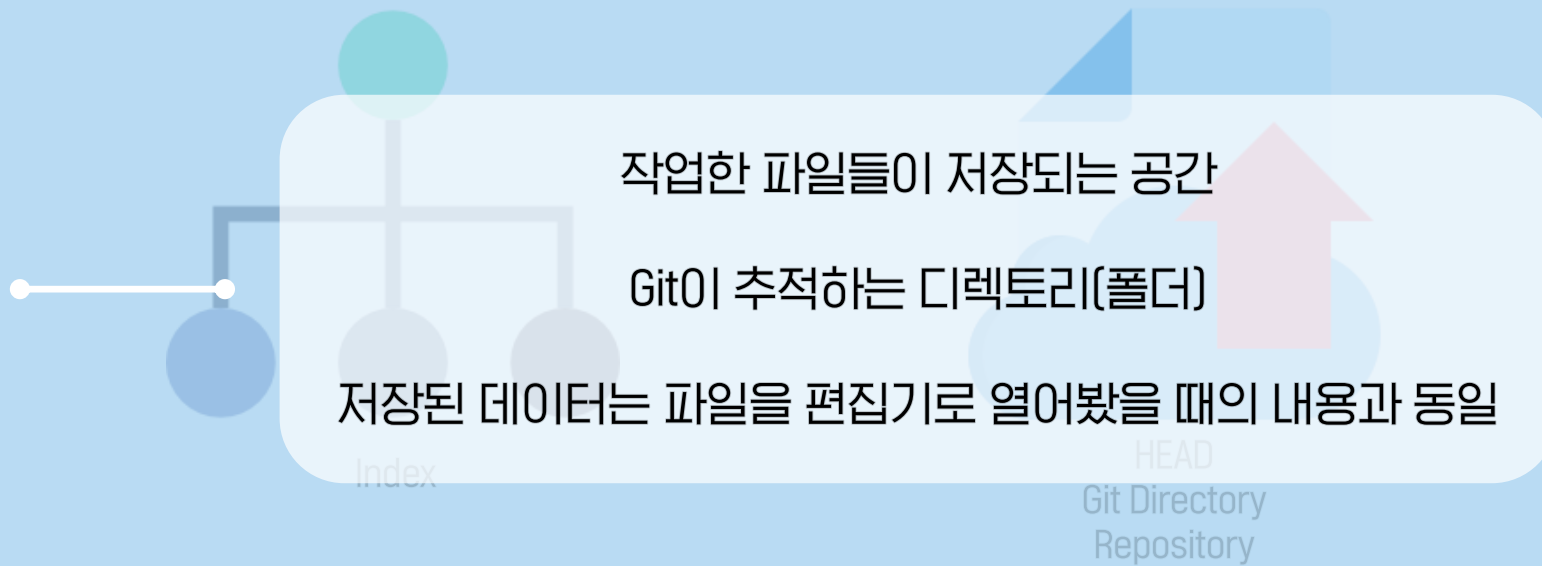
버전관리 시스템이란?







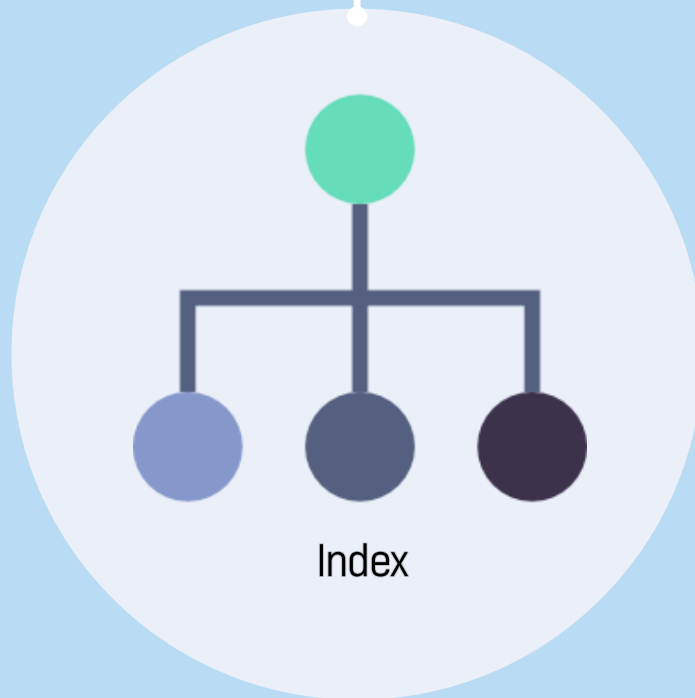
Working Directory



Commit 대상 목록들을 담아두는 공간  
(Files are Staged / Staging 된 파일)



Working Directory



Index



HEAD  
Git Directory  
Repository



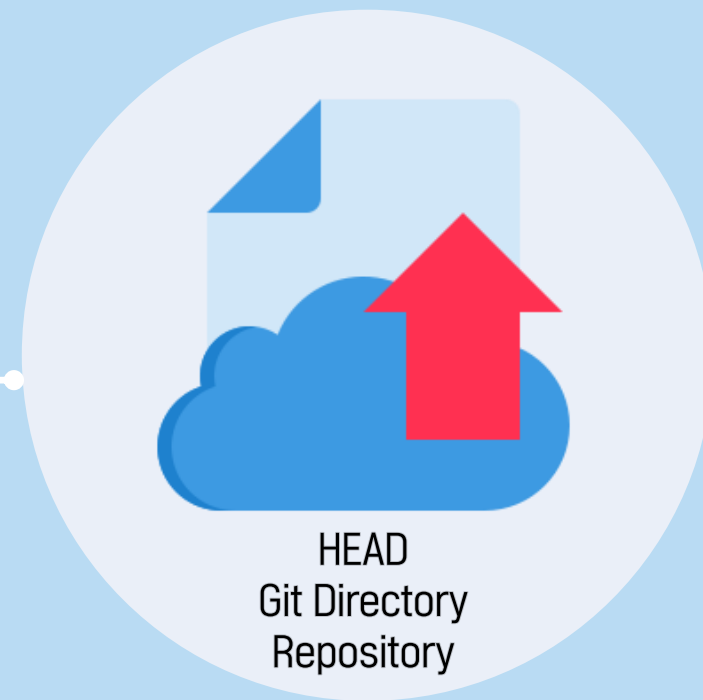
Commit된 파일들이 저장되는 공간

HEAD는 일종의 포인터

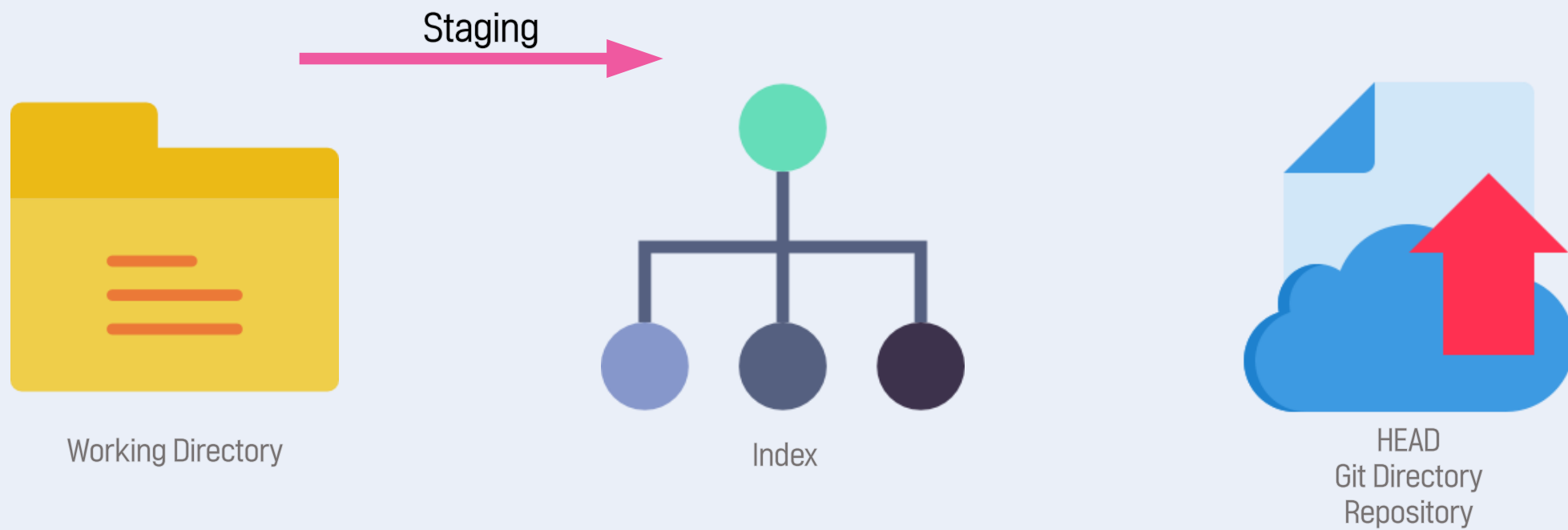
특정 커밋 혹은 브랜치를 가리킴

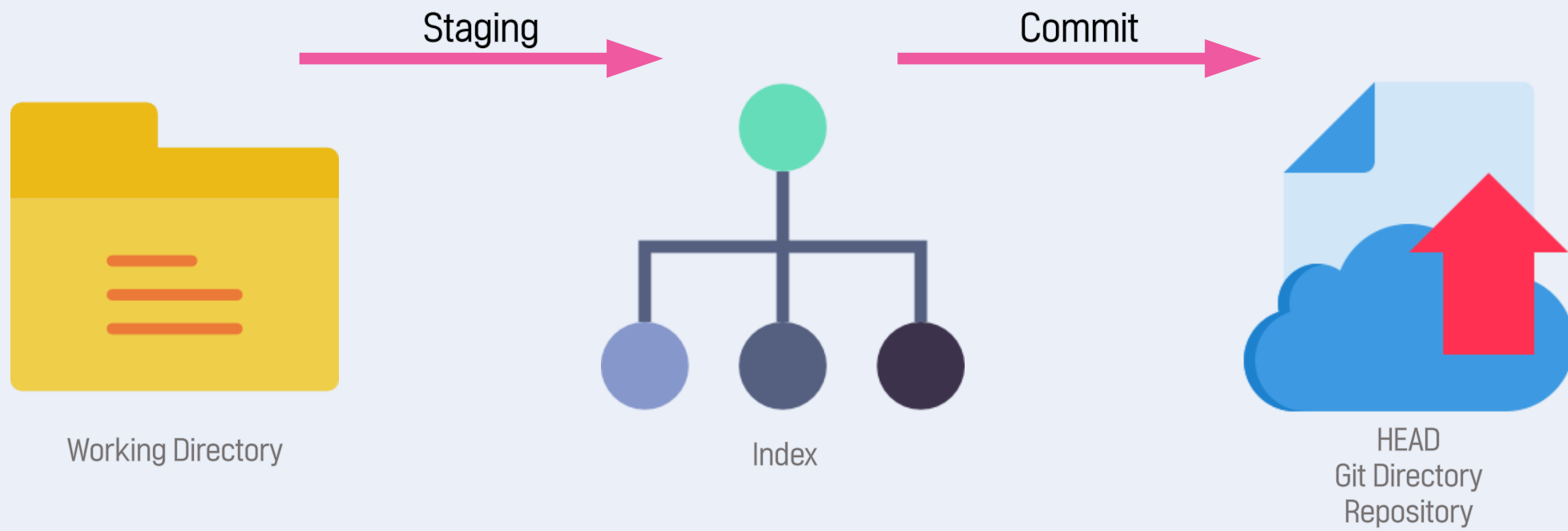
Working Directory

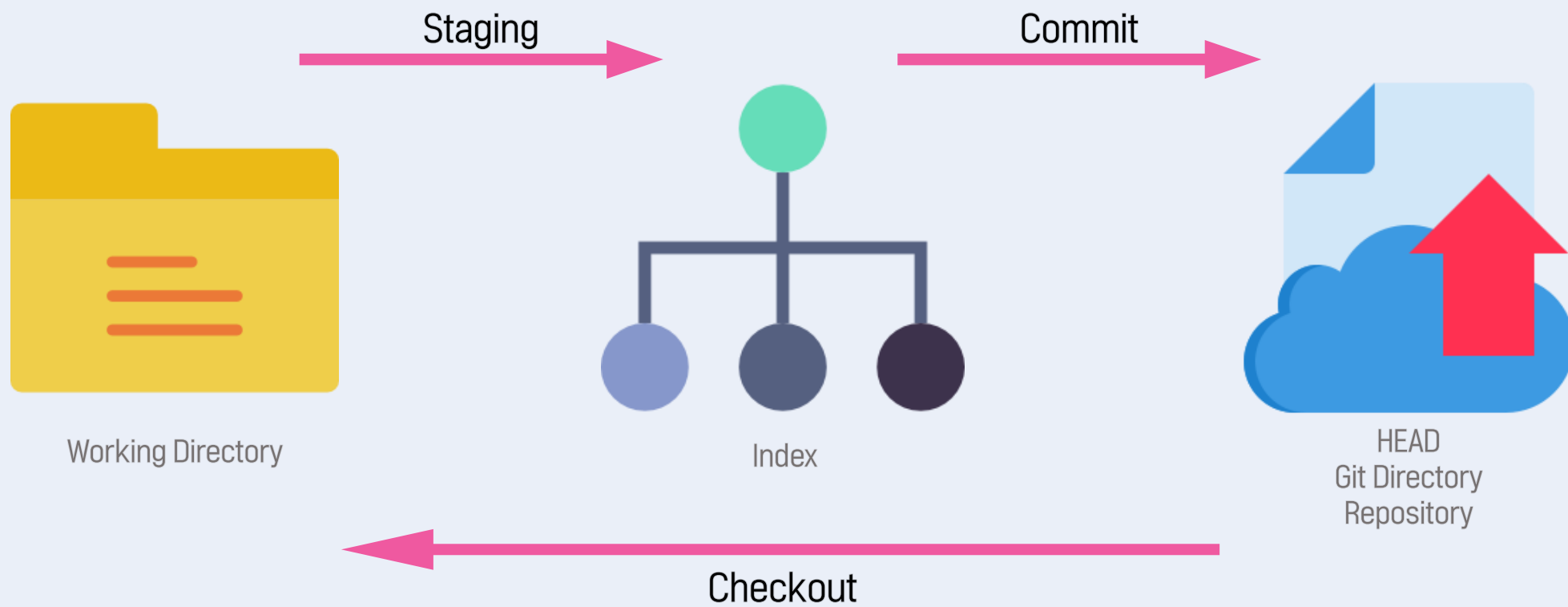
Index





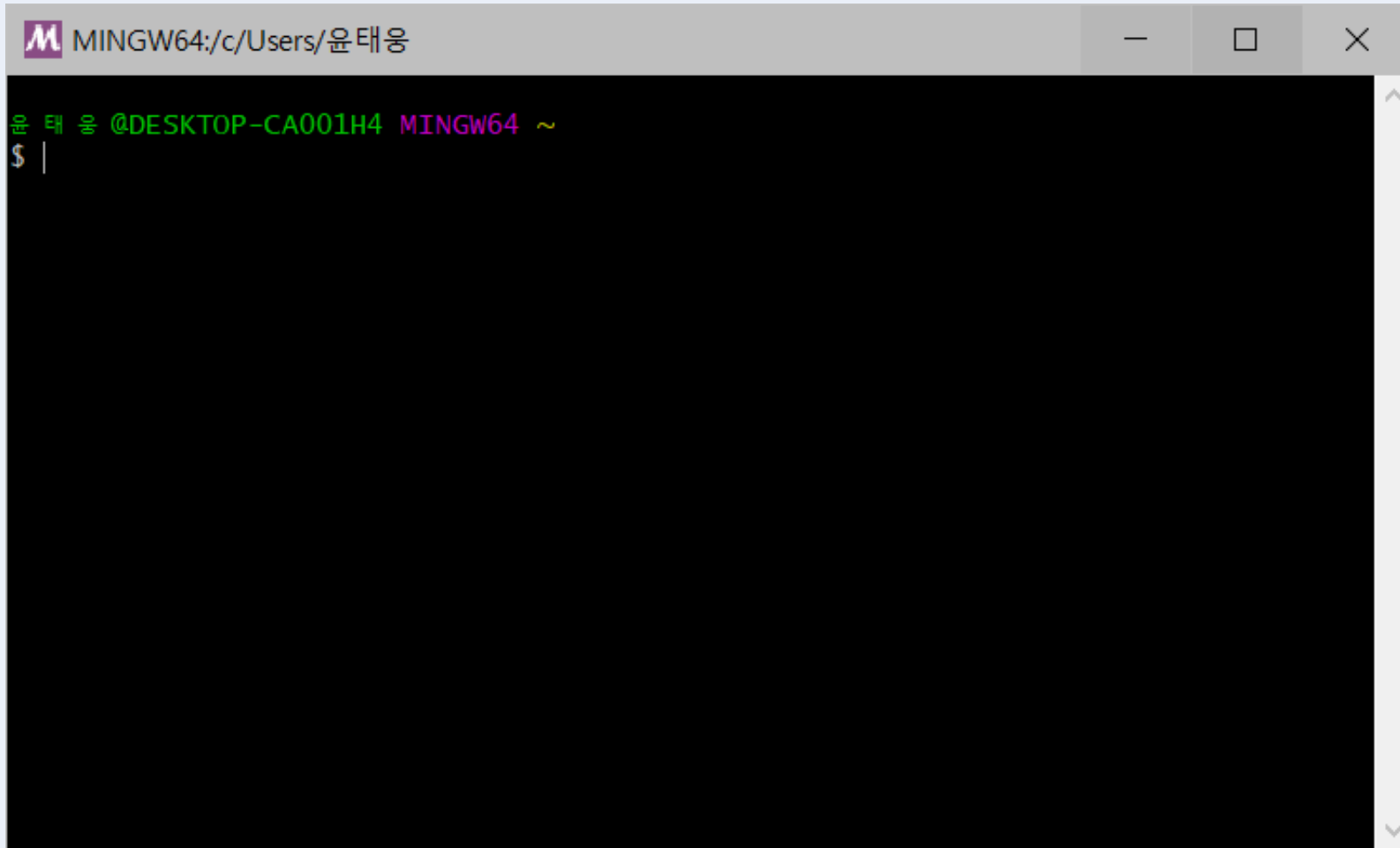






03

기본적인 사용법 - 몇가지 Linux 명령어

A screenshot of a MINGW64 terminal window. The title bar shows the MINGW64 logo and the path 'MINGW64:/c/Users/윤태웅'. The terminal content shows the prompt '윤 태 웅 @DESKTOP-CA001H4 MINGW64 ~' followed by a dollar sign '\$' and a vertical bar '|' on the next line, indicating a ready state for a command.

```
MINGW64:/c/Users/윤태웅
윤 태 웅 @DESKTOP-CA001H4 MINGW64 ~
$ |
```



```
$ pwd
```

현재 디렉토리 출력

```
$cd dir  
$cd /dir
```

cd : 디렉토리 변경

cd dir : 상대경로

cd/dir : 절대경로

cd : 최상위 디렉토리 [c/User/사용자]

```
$mkdir  
$mkdir -p
```

mkdir : 디렉토리 생성

mkdir -p : 하위 디렉토리까지 생성



```
$ ls -a
```

```
$ ls -l
```

```
$ ls -R
```

```
$ ls -F
```

ls : 파일 및 디렉토리 목록 출력

-a : 숨겨진 파일까지 모두 출력

-l : 리스트 형태로 자세히 출력

-R : 하위 디렉토리까지 출력

-F : 확장자도 출력

```
$ touch
```

touch : 파일 생성

```
$ rm
```

rm : 파일 삭제

rm -r : 디렉토리까지 삭제

● rm, rmdir 등의 삭제 명령어는 사용시 주의해야함 ●



## | 사용 예시 |

```
MINGW64:/c/Users/윤태웅/Desktop
윤 태 웅 @DESKTOP-CA001H4 MINGW64 ~
$ cd /c/Users/

윤 태 웅 @DESKTOP-CA001H4 MINGW64 /c/Users
$ ls
'All Users'@  Default/  'Default User'@  desktop.ini  Public/  윤 태 웅 /

윤 태 웅 @DESKTOP-CA001H4 MINGW64 /c/Users
$ cd 윤 태 웅

윤 태 웅 @DESKTOP-CA001H4 MINGW64 ~
$ cd Desktop

윤 태 웅 @DESKTOP-CA001H4 MINGW64 ~/Desktop
$ ls
'~$CoMit 무심코 명단_최종.xlsx'
'~$CoMit 부원 관리(최종).xlsx'
'~$tlin - Class.docx'
'~$개강총회 자료.pptx'
'~$출석부.xlsx'
1.pptx
12.pptx
a/
asdf/
```



## | 사용 예시 |

```
MINGW64:/C/Users/윤태웅/Desktop
윤 태 웅 @DESKTOP-CA001H4 MINGW64 /C/Users/윤 태 웅 /Desktop
$ mkdir asdf

윤 태 웅 @DESKTOP-CA001H4 MINGW64 /C/Users/윤 태 웅 /Desktop
$ mkdir asdfg/aaa/bbb
mkdir: cannot create directory 'asdfg/aaa/bbb': No such file or directory

윤 태 웅 @DESKTOP-CA001H4 MINGW64 /C/Users/윤 태 웅 /Desktop
$ mkdir -p asdfg/aaa/bbb
```





윤태웅@DESKTOP-CA001H4 MINGW64

```
$git config --global user.name <이름>
```

```
$git config --global user.email <이메일>
```

```
$git config --global core.autocrlf true
```

Git 초기설정

Git 로그에 표시되는 이름 설정

Git 로그에 표시되고, github에서 contribution을 세는 기준

여러 운영체제에서 프로젝트를 진행할 때 사용



# \$ git init

: 현재 디렉토리를 git 디렉토리 (HEAD)로 지정



```
$ git add <파일 이름>  
$ git add -A
```

```
$ git status  
$ git status -s
```

```
$ git commit -m '<메시지>'  
$ git commit
```

```
$ git log  
$ git log -p
```

add : 지정한 파일을 index에 staging

status : 작업 디렉토리의 상태를 확인

commit : index의 내용을 HEAD로

-m : commit 메시지를 입력  
옵션없이 Enter 누르면 editor 실행  
i 입력, esc :wq

log commit 내역 확인

-p : 시간 역순으로 확인



## | 파일 삭제 절차 |

1. (파일을 삭제)

2. `$ git add <파일명>`  
또는 `$ git add -A`

3. `$ git commit`

`$ git revert <hash>`

작업 디렉토리에서 파일 삭제

파일 삭제 기록을 add

파일 삭제 기록을 commit

Commit을 이미 push한 경우 사용하는 명령어



# .gitignore?

: git이 추적해서는 안되는 파일들이 존재함 → 추적하지 않을 파일 [/private, .apk, ...]을 직접 지정

## | .gitignore 파일 사용하기 |

<https://nesoy.github.io/articles/2017-01/Git-Ignore>



## | 규칙 |

#으로 시작

주석

\*.ipr

확장자가 ipr인 모든 파일

filename

이름이 filename인 파일

dir/

Dir 디렉토리의 모든 파일과 폴더 무시 (/는 선택)

!dir/file

dir/file은 무시하지 않음

a/\*\*/b

a폴더에 속한 모든 하위 폴더의 파일 중 이름이 b인 파일 무시

a/b

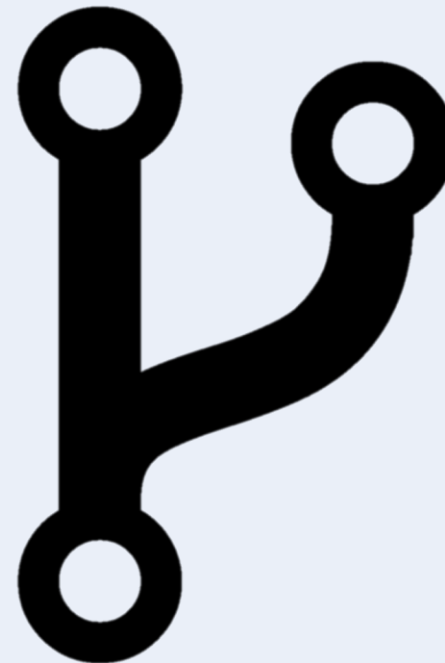
a/b 무시



<https://github.com>



<https://github.com/skku-overflow/Spoon-Knife>



Github 원격저장소 연결하기 위해서는 계정이 필요

HTTPS

SSH key





Github 원격저장소 연결하기 위해서는 계정이 필요

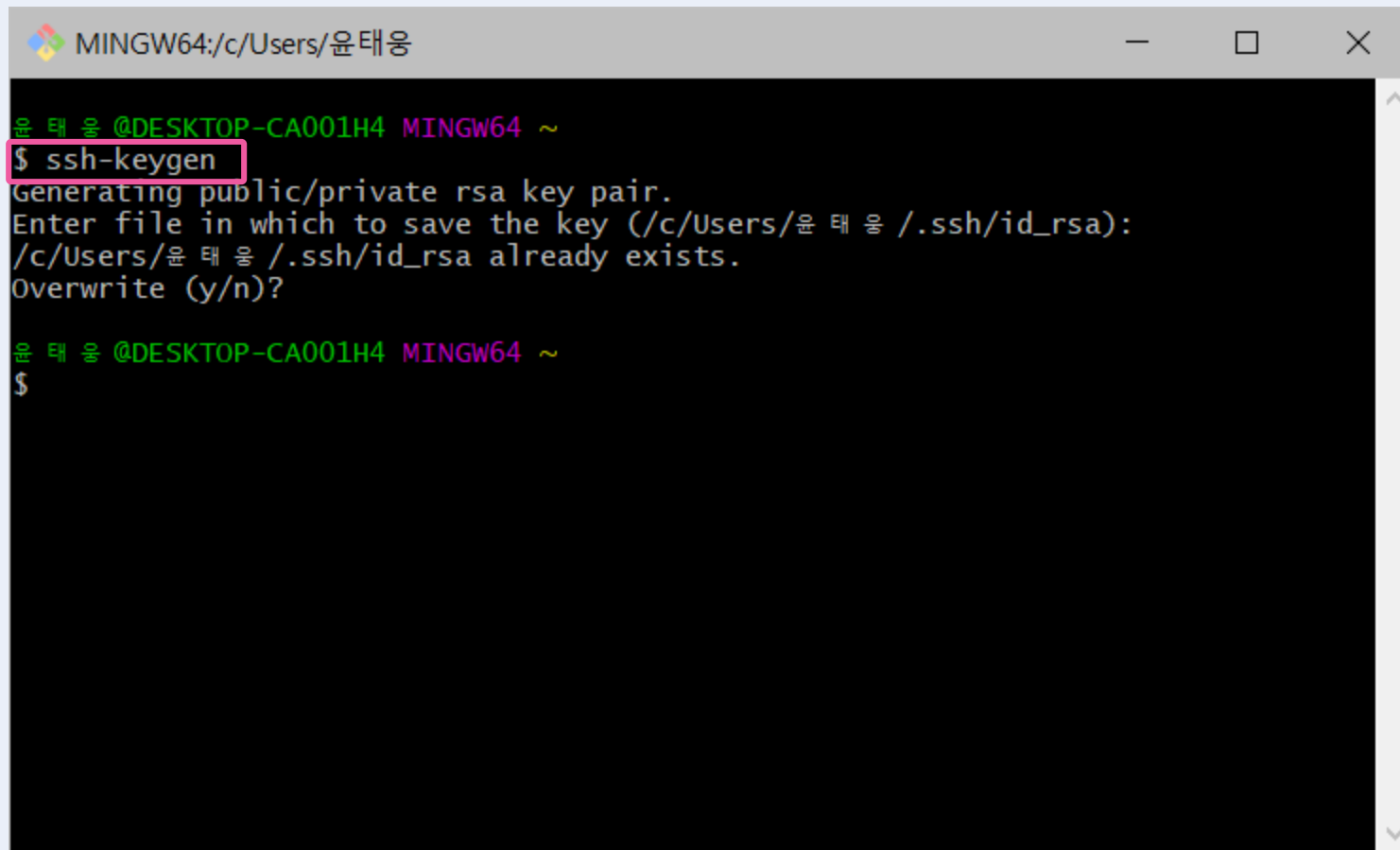
Push할 때 로그인이 필요 없어 편리하다!

HTTPS

SSH key



## 1. SSH key 생성



```
MINGW64:/c:/Users/윤태웅

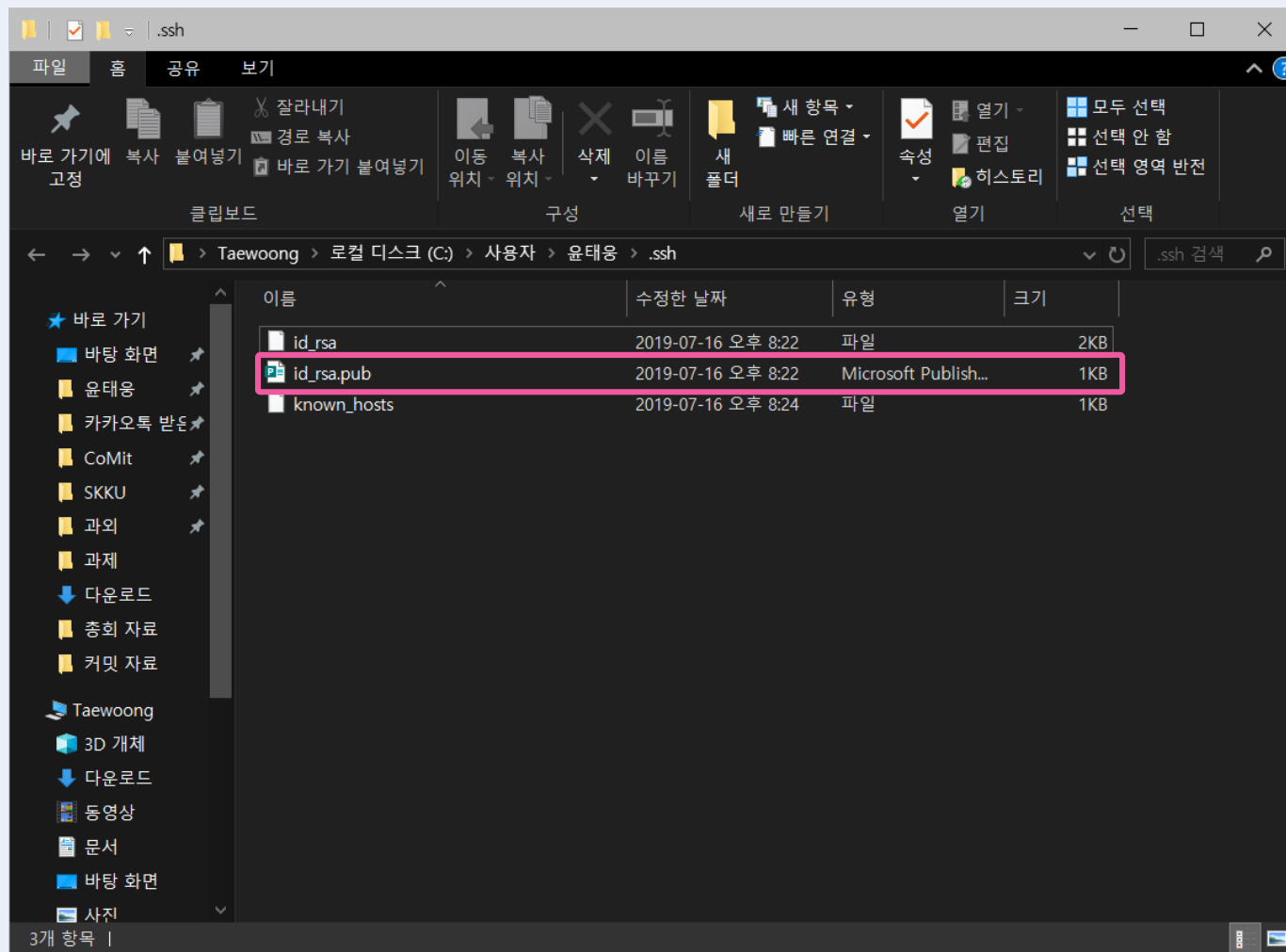
윤 태 웅 @DESKTOP-CA001H4 MINGW64 ~
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c:/Users/윤 태 웅 /.ssh/id_rsa):
/c:/Users/윤 태 웅 /.ssh/id_rsa already exists.
Overwrite (y/n)?

윤 태 웅 @DESKTOP-CA001H4 MINGW64 ~
$
```

Git bash 창에 ssh-keygen 입력하고  
계속 엔터를 누른다.



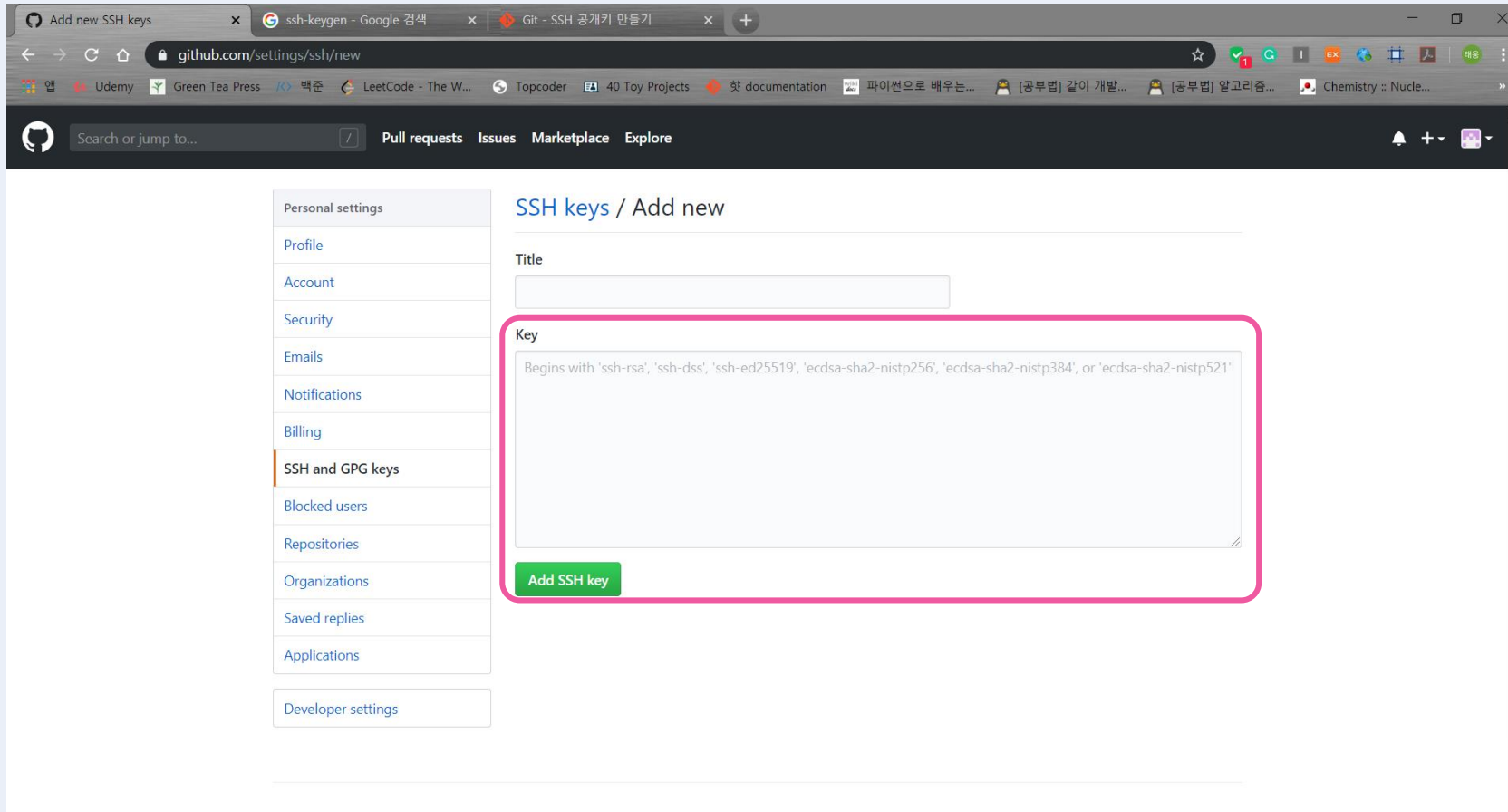
## 2. SSH key 복사



users/.ssh/id\_rsa.pub 파일을 VS Code로 열어서 값을 복사해두기



### 3. 원격 저장소에 키 입력



Github 사이트에서 [프로필] > [Settings] > [SSH and GPG keys] key에 붙여넣기



## 4. 로컬 저장소를 원격 저장소와 연결하기

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 \*:  ✓

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

Description (optional):

☒ 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.

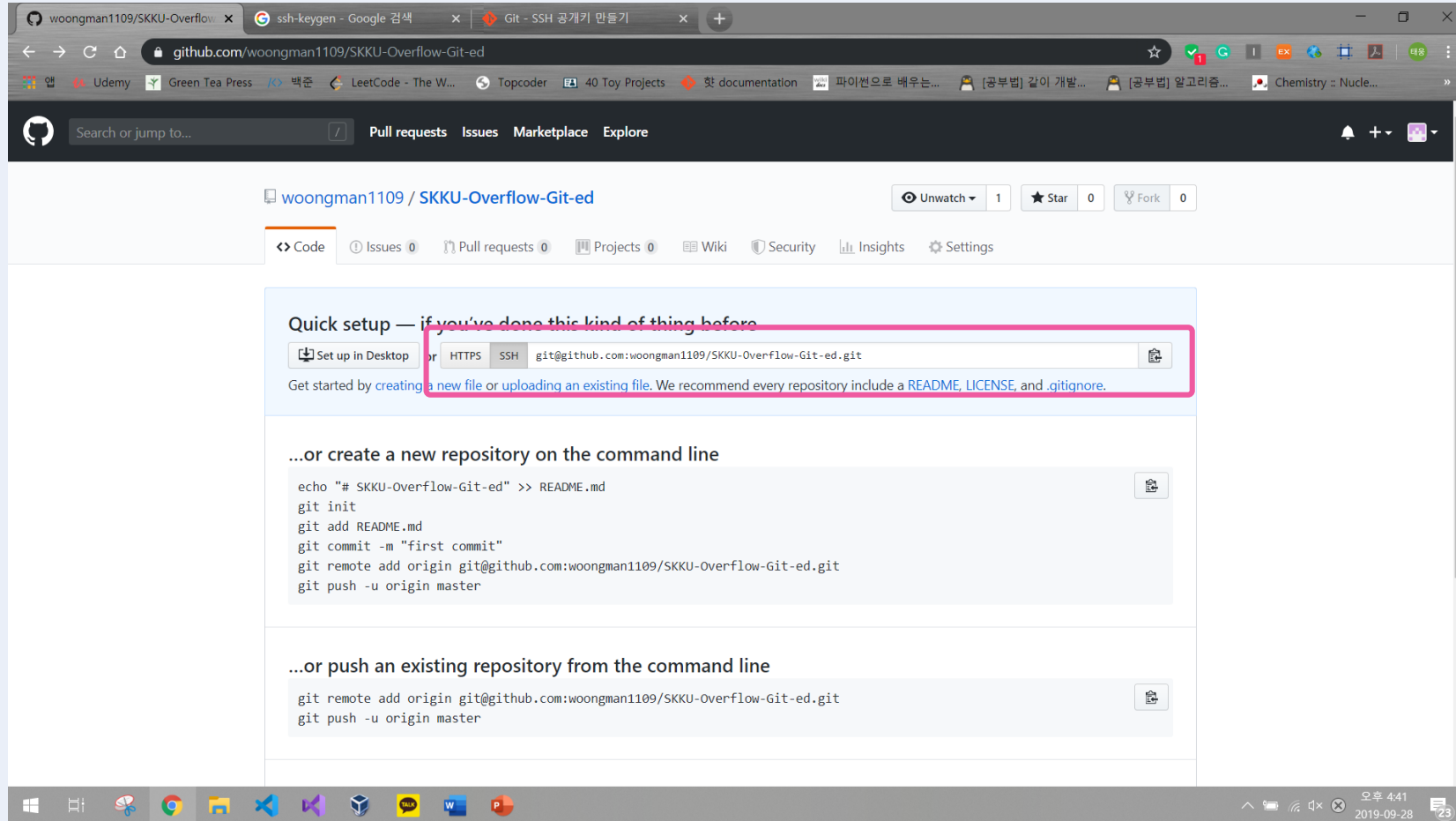
☐ Initialize this repository with a README  
This will let you immediately clone the repository to your computer.

Add .gitignore:  Add a license:  ⓘ

Github 사이트에서 [프로필] > [Your repositories] > [New] 에서  
이름 설정만 하고 Create Repository



## 5. 원격저장소의 URL 복사하기



woongman1109/SKKU-Overflow-Git-ed

Unwatch 1 Star 0 Fork 0

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

Quick setup — if you've done this kind of thing before

Set up in Desktop or HTTPS SSH git@github.com:woongman1109/SKKU-Overflow-Git-ed.git

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# SKKU-Overflow-Git-ed" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin git@github.com:woongman1109/SKKU-Overflow-Git-ed.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin git@github.com:woongman1109/SKKU-Overflow-Git-ed.git
git push -u origin master
```

다음 화면의 원격저장소 URL을 복사한다  
[SSH 버튼 눌린 거 확인하고 오른쪽에 있는 복사버튼 클릭]



```
$ git remote add origin <서버 URL>
```

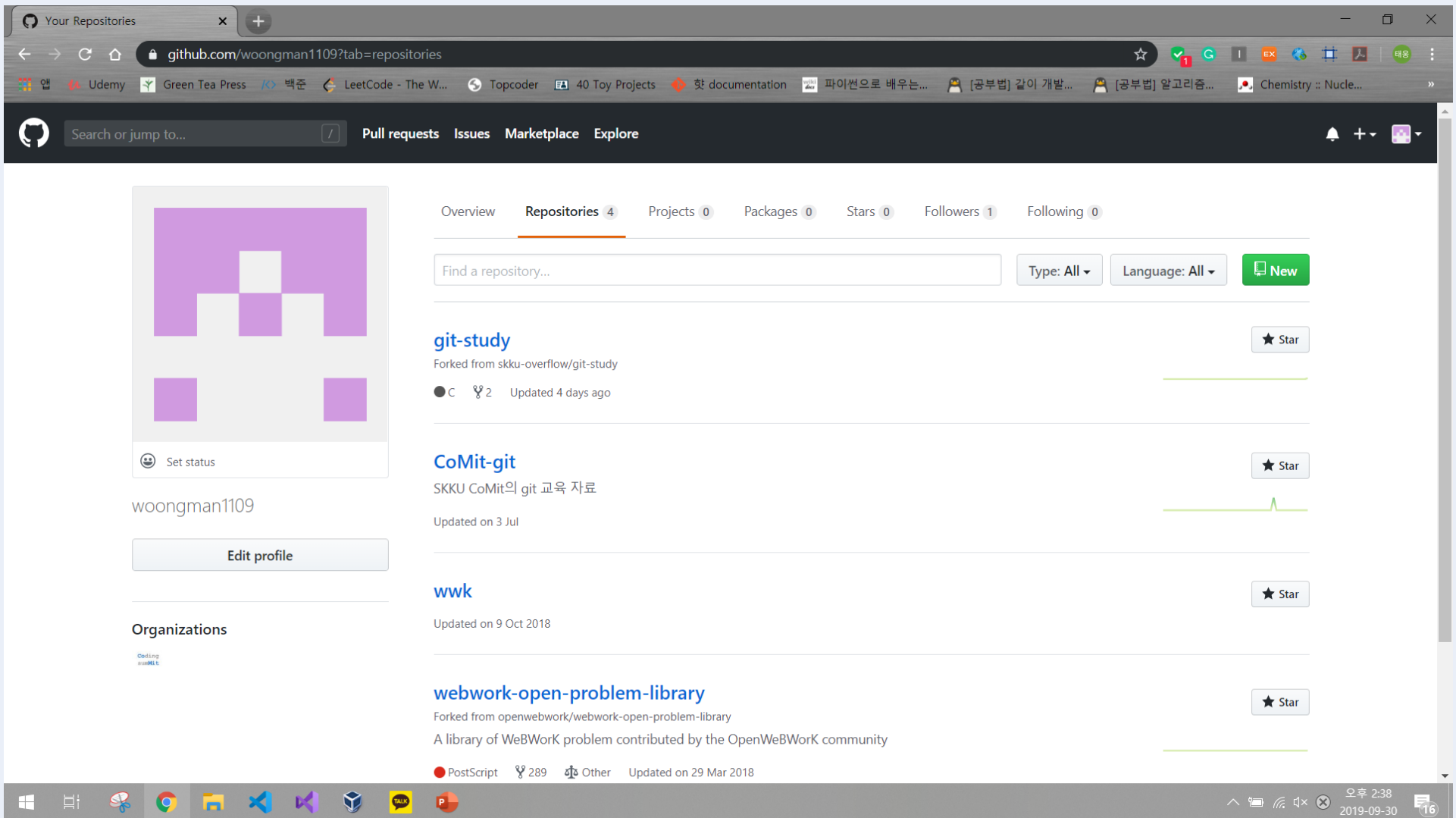
복사한 주소의 원격 저장소를 현재 디렉토리의 원격 저장소로 지정 (origin)

```
$ git push -u origin <branch 이름>  
$ git push
```

연결된 원격저장소(origin)의 branch를 현재 로컬 branch의 upstream으로 설정하고,  
이와 동시에 해당 branch의 HEAD 내용을 지정된 원격저장소에 push

처음 push하는 branch는 upstream 설정을 위해 -u 옵션을 준다.  
이후에는 다른 옵션이나 branch 지정 없이 push만 하면 됨





The screenshot shows a web browser window displaying the GitHub profile of user **woongman1109**. The browser's address bar shows the URL `github.com/woongman1109?tab=repositories`. The GitHub navigation bar at the top includes links for Pull requests, Issues, Marketplace, and Explore. The user's profile section on the left shows a placeholder for a profile picture, the username **woongman1109**, and an "Edit profile" button. Below this, there is a section for "Organizations" with a small "Pending" status indicator. The main content area displays the user's repositories, with tabs for Overview, Repositories (4), Projects (0), Packages (0), Stars (0), Followers (1), and Following (0). The repositories listed are:

- git-study**: Forked from `skku-overflow/git-study`. Language: C. 2 forks. Updated 4 days ago.
- CoMit-git**: SKKU CoMit의 git 교육 자료. Updated on 3 Jul.
- wwkw**: Updated on 9 Oct 2018.
- webwork-open-problem-library**: Forked from `openwebwork/webwork-open-problem-library`. A library of WeBWorK problem contributed by the OpenWeBWorK community. Languages: PostScript (289), Other. Updated on 29 Mar 2018.

The browser's taskbar at the bottom shows various application icons, including Windows, File Explorer, Google Chrome, and others. The system clock in the bottom right corner indicates the time is 2:38 PM on 2019-09-30.





The screenshot shows a web browser window displaying the GitHub repository page for 'woongman1109/git-study'. The browser's address bar shows the URL 'github.com/woongman1109/git-study'. The repository page includes a header with the repository name, a 'Watch' button, and statistics for stars (0) and forks (2). Below the header, there are tabs for 'Code', 'Pull requests', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The main content area shows a message 'No description, website, or topics provided.' followed by a 'Manage topics' link. A summary bar indicates '1 commit', '1 branch', '0 releases', and '1 contributor'. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find File', and 'Clone or download'. A commit history section shows a single commit by 'kdy1' with the message 'first commit' and a file named 'main.c'. At the bottom, there is a prompt to 'Add a README'.

woongman1109/git-study

github.com/woongman1109/git-study

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

woongman1109 / git-study  
forked from skku-overflow/git-study

Watch 0 Star 0 Fork 2

Code Pull requests 0 Projects 0 Wiki Security 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

This branch is even with skku-overflow:master. Pull request Compare

kdy1 first commit Latest commit 6910501 4 days ago

main.c first commit 4 days ago

Help people interested in this repository understand your project by adding a README. Add a README

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



# \$ git clone /dir [newdir]


/dir를 /newdir (기본값: 현재 폴더) 에 그대로 복제  
복제한 디렉토리를 git 디렉토리 (HEAD)로 지정

복제할 디렉토리 (/dir) 설정 방법

-로컬 저장소 : 직접경로입력-

-원격저장소 : [github의 clone URL](#) 붙여넣기-



 [skku-overflow](#) / [git-study](#)

Watch 1

Star 0

Fork 2

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Security

Insights

*No description, website, or topics provided.*

1 commit

1 branch

0 releases

1 contributor

Branch: master ▾


New pull request

Create new file


Upload files

Find File

Clone or download ▾

 kdy1 first commit


Latest commit 6910501 4 days ago

 [main.c](#)

first commit

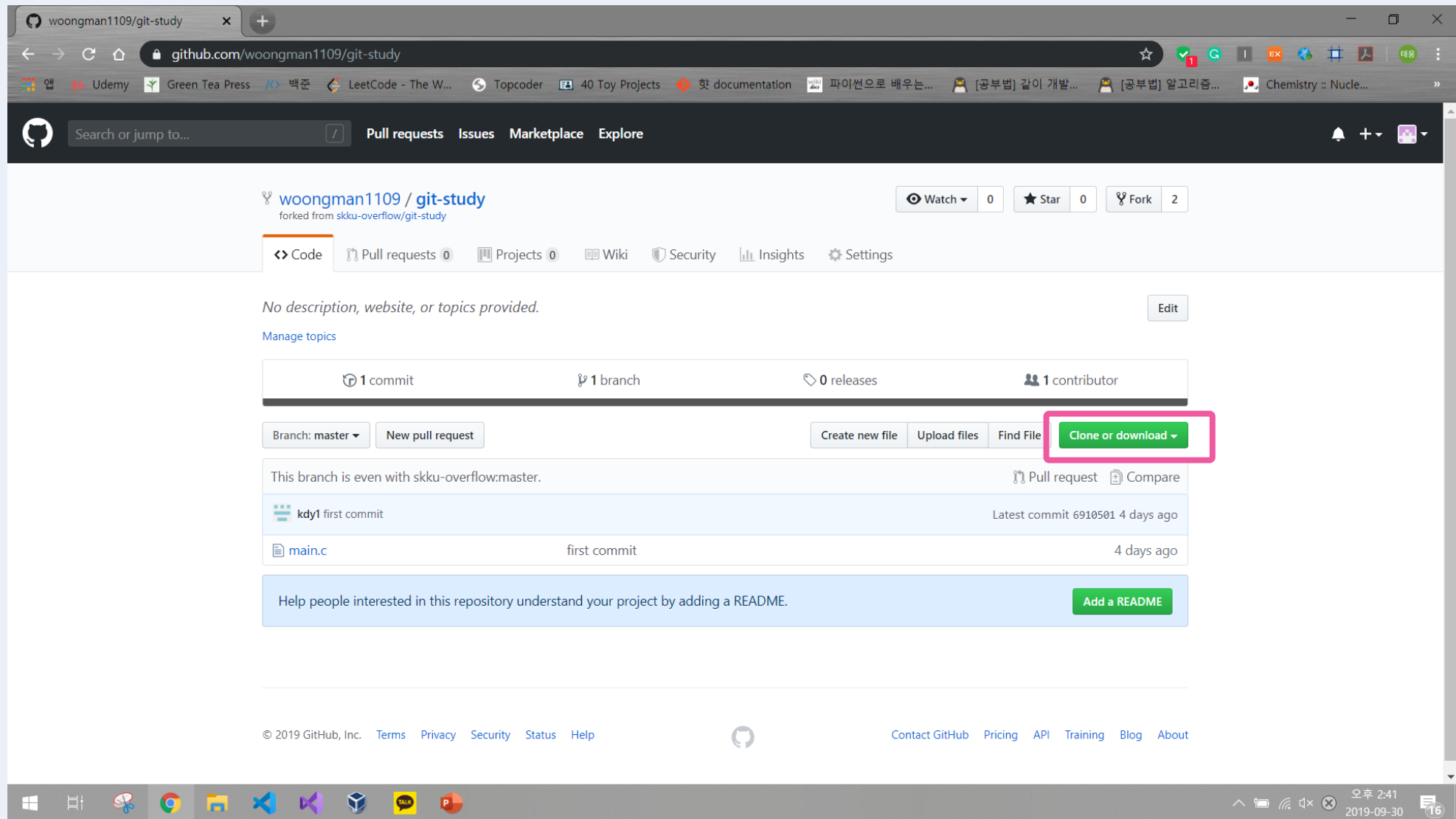
4 days ago

© 2019 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)



[Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

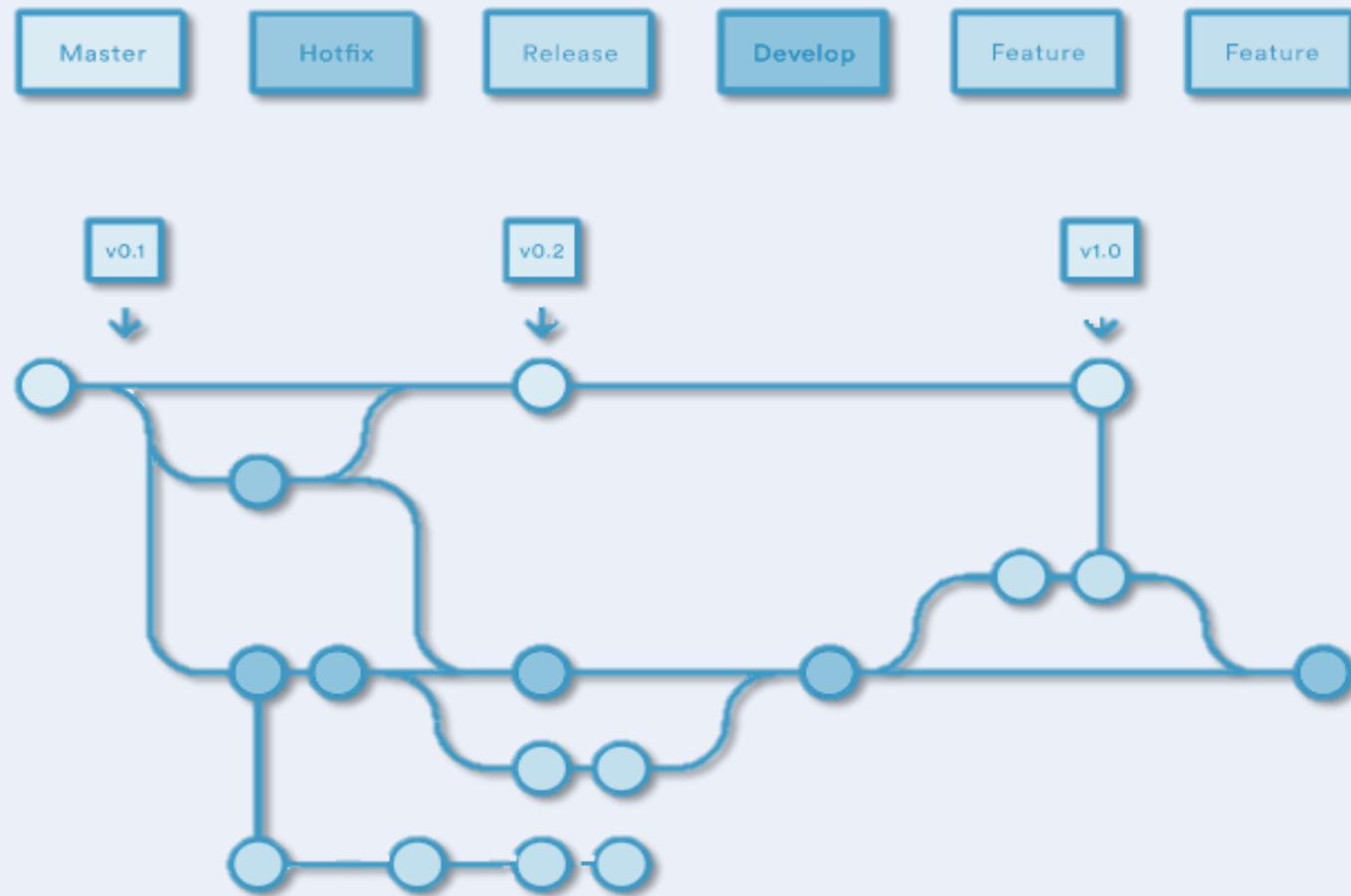




**\$ git push**

git clone 한 경우 원격저장소가 연결되며 upstream도 자동으로 설정되므로  
git push만 하면 충분하다!  
(commit들을 업로드)





## Local

```
$ git checkout -b <branch 이름>
```

```
$ git checkout <branch 이름>
```

```
$ git branch -d <branch 이름>
```

```
$ git branch -D <branch 이름>
```

## Remote

```
$ git push origin <branch 이름>
```

## Local

지정한 이름으로 branch 생성

지정한 branch로 이동

지정한 branch 삭제

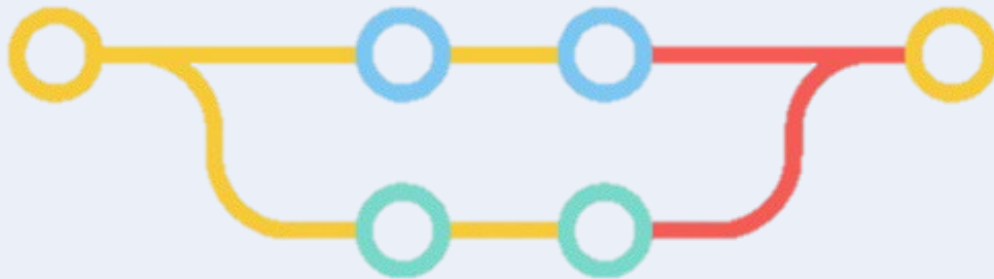
지정한 branch 강제삭제

## Remote

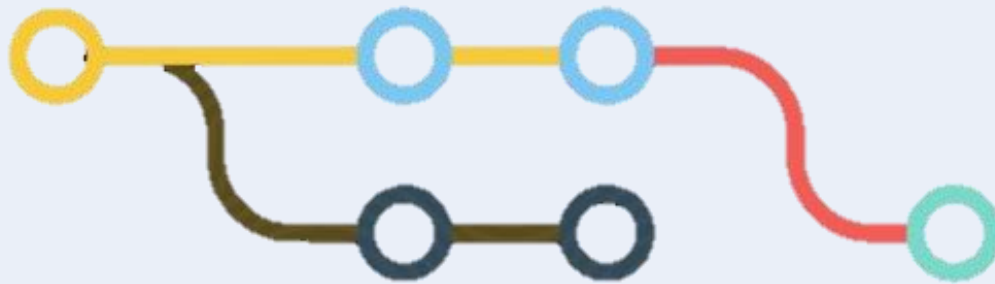
지정된 원격 저장소에 입력한 이름으로  
branch 현재 HEAD 내용을 push



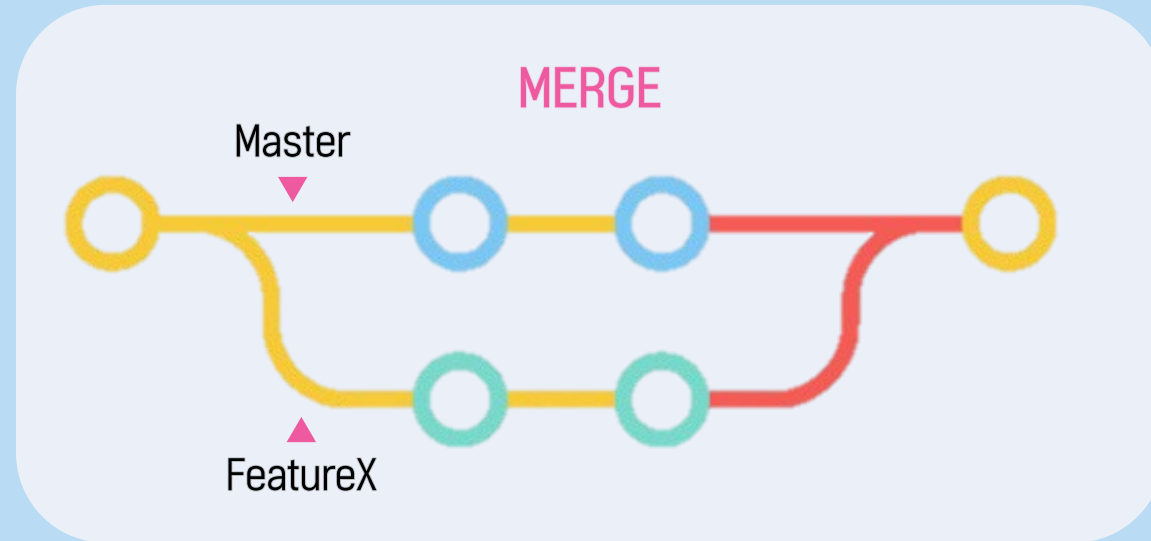
## MERGE



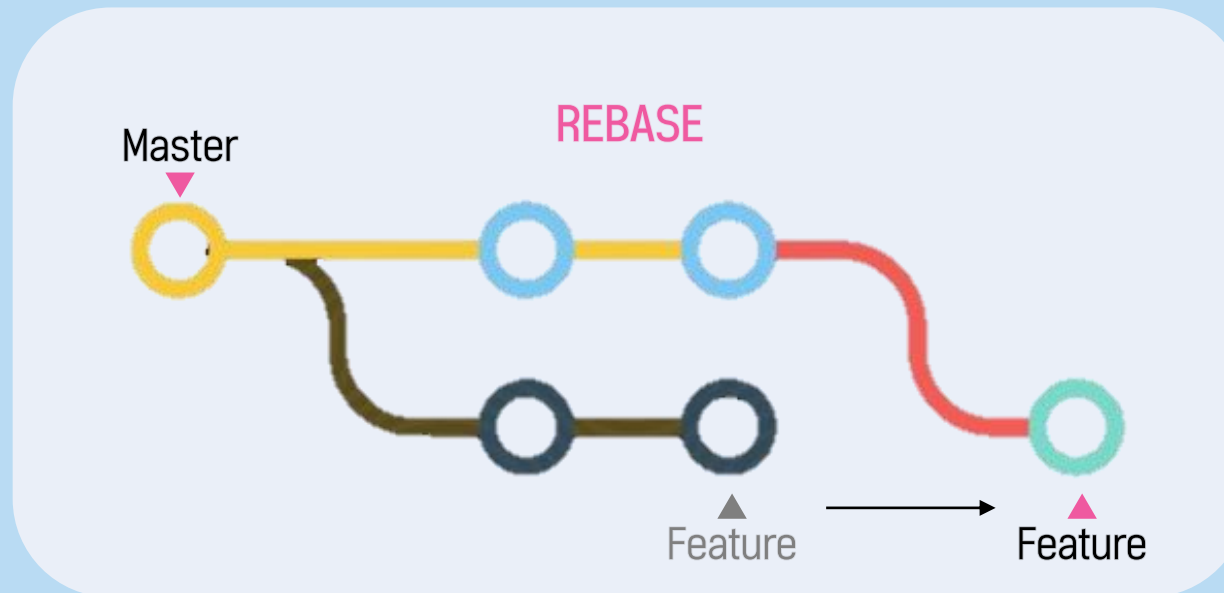
## REBASE







Base branch의 수정내역(commit)을  
최신 내역까지 업데이트하여  
현재 branch의 base 를 재설정(rebase)



```
$ git merge <branch 이름>
```

```
$ git merge feature_x
```

지정한 branch를 현재 위치한 branch에 merge

### 충돌(conflict)이 발생하면?

충돌 부분을 직접 수정하여 병합이 가능하도록 해야함  
이후 해당 파일들은 \$ git add하여 staging

### fast-forward merge가 실패하면?

직접 병합을 해줘야함  
[vscode로 열어서 간편하게 병합 가능]

```
$ git add <파일>
```

merge 중일 경우

```
$ git merge --continue
```

rebase 중일 경우

```
$ git rebase --continue
```



## 1. 원격저장소에 branch 생성, push

```
윤 태 웅 @DESKTOP-CA001H4 MINGW64 /  
1)  
$ git add origin branch1
```

## 2. Branch 목록 보기

A screenshot of the GitHub interface showing a branch list with a blue icon and the text "2 branches".

## 3. New pull request

A button with the text "New pull request" in a light gray box.

## 4. Merge 경로 확인



base: master ▼



compare: branch1 ▼

## 5. Merge에 대한 코멘트 작성 후 create pull request

A green button with the text "Create pull request" and a small downward arrow icon.

## 6. Pull request 확인

A notification box showing a pull request icon and the text "Pull requests 1".

## 7. merge, commit 메시지 작성, confirm

A green button with the text "Merge pull request" and a small downward arrow icon.

감사합니다 :)

