

Git 서브 모듈

컴소과 동아리 REFERENCE

제작자: 김명준, 노종현

목 차

- 서브 모듈이란?
- 활용 방법
- 사용 방법
- 주의할 점

서브 모듈이란?

Git 저장소 안에 다른 Git 저장소를
디렉토리로 분리해 넣는 것

저장소

저장소

ngjun / Git-Practice

Issues Pull requests Actions Projects Wiki Security Insights Settings

Git-Practice Public

main 1 branch 0 tags Go to file

Mangjun update Sub1 version 2 4a61ef8 49 m

lib @ 3f742c1	update Sub1 version 2
.gitmodules	init submodule
README.md	Git과 Github 사용하는 방법

README.md

Git과 Github에 대해 공부

- 목표: 협업을 위한 기능들을 학습

학습하기 전 준비

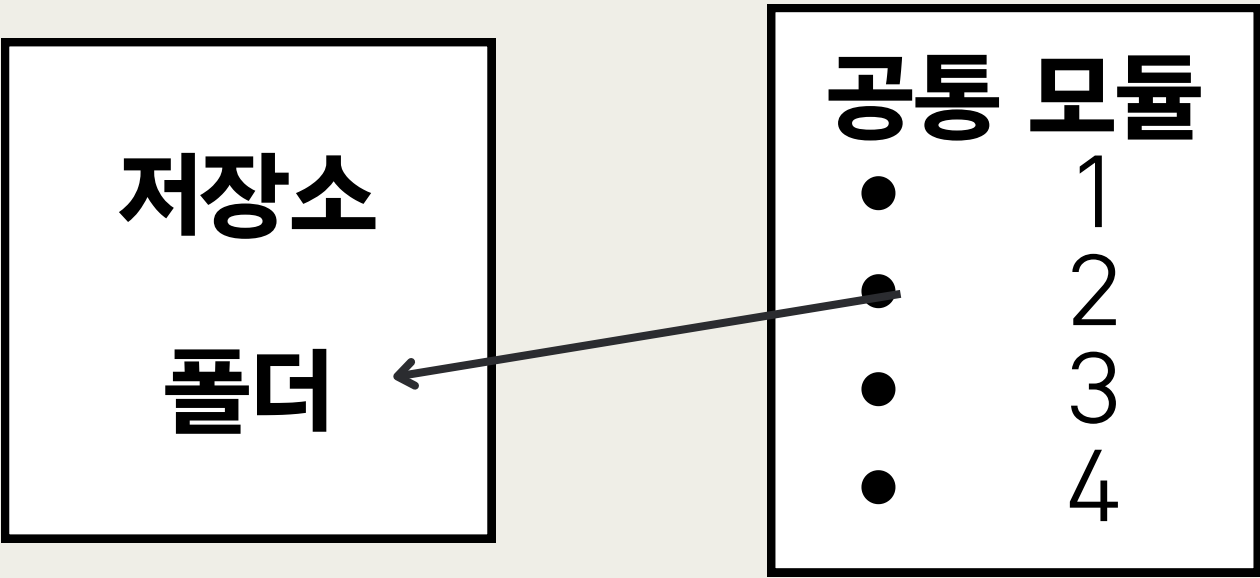
- [Git 설치](#)
- [Github](#)

활용 방법

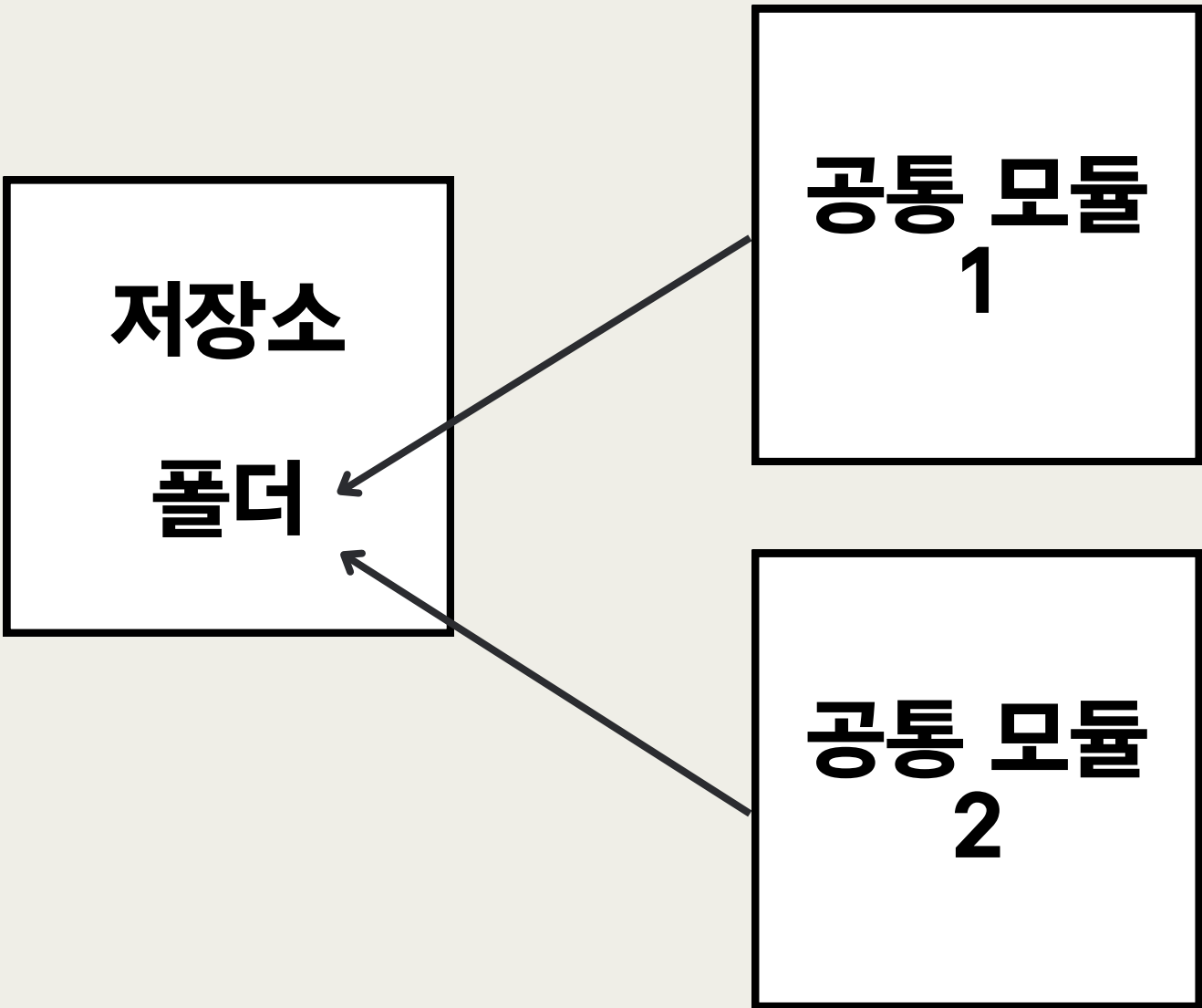
여러 프로젝트에서 사용되는 공통 모듈이 있을 때



공통 모듈이 여러 버전이 있을 때



공통 모듈이 필요할 때 복제



사용 방법

1. Repository 2개 이상을 준비


submodule을 가진 Repository


angjun / Super

IssuesPull requestsActionsProjectsWikiSecurityInsightsSettings


SuperPublic

PinUnwatch

**Set up GitHub Copilot**
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

**Add collaborators to this repository**
Search for people using their GitHub usernames
[Invite collaborators](#)

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

 Set up in Desktop

 or

HTTPS

SSH

<https://github.com/Mangjun/Super.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 "# Super" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Mangjun/Super.git
```


submodule


Mangjun / Sub

IssuesPull requestsActionsProjectsWikiSecurityInsightsSettings


SubPublic

PinUnwatch

**Set up GitHub Copilot**
Use GitHub's AI pair programmer to autocomplete suggestions as you code.
[Get started with GitHub Copilot](#)

**Add collaborators to this repository**
Search for people using their GitHub usernames
[Invite collaborators](#)

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

 Set up in Desktop

 or

HTTPS

SSH

<https://github.com/Mangjun/Sub.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 "# Sub" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Mangjun/Sub.git
```

사용 방법

2. submodule 생성

명령어: git submodule add "Sub git 주소" 디렉터리

```
myoun@Mangjun MINGW64 /d/Git/Super (main)
$ git submodule add https://github.com/Mangjun/Sub.git lib
Adding existing repo at 'lib' to the index
warning: in the working copy of '.gitmodules', LF will be replaced by CRLF the n
ext time Git touches it

myoun@Mangjun MINGW64 /d/Git/Super (main)
$ git status
On branch main

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   .gitmodules
    new file:   lib

myoun@Mangjun MINGW64 /d/Git/Super (main)
$
```

git 주소를 알려주는 명령어: git remote -v

```
myoun@Mangjun MINGW64 /d/Git/Sub (main)
$ git remote -v
origin  https://github.com/Mangjun/Sub.git (fetch)
origin  https://github.com/Mangjun/Sub.git (push)

myoun@Mangjun MINGW64 /d/Git/Sub (main)
$
```

사용 방법

3-1. submodule pull

submodule에 파일 생성 및 원격에 올림

```
myoun@Mangjun MINGW64 /d/Git/Sub (main)
$ touch "version 1"

myoun@Mangjun MINGW64 /d/Git/Sub (main)
$ ls -al
total 4
drwxr-xr-x 1 myoun 197609 0 Aug 28 13:55 ./
drwxr-xr-x 1 myoun 197609 0 Aug 28 13:36 ../
drwxr-xr-x 1 myoun 197609 0 Aug 28 13:36 .git/
-rw-r--r-- 1 myoun 197609 0 Aug 28 13:55 'version 1'

myoun@Mangjun MINGW64 /d/Git/Sub (main)
$ git add .;git commit -m "version 1";git push
[main (root-commit) e4cd795] version 1
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 version 1
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 210 bytes | 210.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Mangjun/Sub.git
 * [new branch]      main -> main

myoun@Mangjun MINGW64 /d/Git/Sub (main)
$ git log --oneline
e4cd795 (HEAD -> main, origin/main) version 1
```

Super의 submodule에서 pull

```
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
      lib/

nothing added to commit but untracked files present (use "git add" to track)

myoun@Mangjun MINGW64 /d/Git/Super (main)
$ cd lib/

myoun@Mangjun MINGW64 /d/Git/Super/lib (main)
$ git pull
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), 190 bytes | 8.00 KiB/s, done.
From https://github.com/Mangjun/Sub
 * [new branch]      main      -> origin/main

myoun@Mangjun MINGW64 /d/Git/Super/lib (main)
$ git log --oneline
e4cd795 (HEAD -> main, origin/main) version 1

myoun@Mangjun MINGW64 /d/Git/Super/lib (main)
$
```

사 용 방 법

3-2. submodule pull

여러 개의 submodule이 존재한다면 일일이 pull해주기 귀찮음

→ `git submodule update --remote`

(Super 디렉터리에서)

주의할 점

서브 모듈이 있는 repository clone 시
서브 모듈의 디렉터리는 비어있음

→ `git submodule init`

→ `git submodule update --remote`

감사합니다

컴소과 동아리 REFERENCE