

1/12 git

Remote Repository

로컬 & 원격 저장소

gitignore

github활용

원격 저장소

- 코드와 버전 관리 이력을 온라인 상의 특정 위치에 저장하여 여러 개발자가 협업하고 코드를 공유할 수 있는 저장 공간
- 원격 저장소 서비스 > GitLab, GitHub등
- git과 github은 각각의 서비스이다

git remote add origin <remote_repo_url>

github에서 원격 저장소인 repository 생성

Create a new repository

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

Required fields are marked with an asterisk (*).

Owner *

 JAMIEKIMHS ▾

Repository name *

/

Great repository names are short and memorable. Need inspiration? How about **turbo-sniffle** ?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:



Add a README file

This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

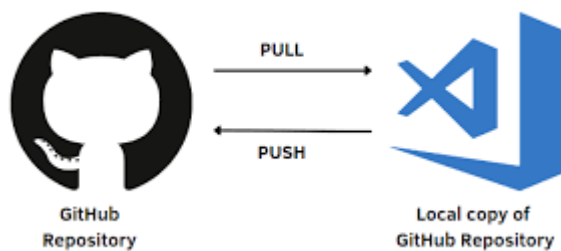
License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)



You are creating a public repository in your personal account.

Create repository

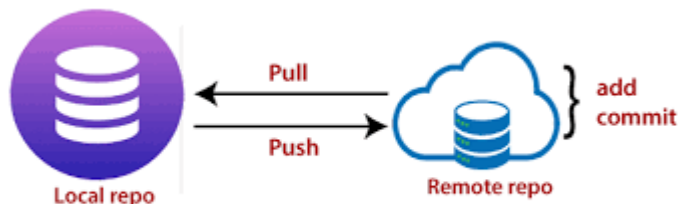


```
git remote add origin <remote_repo_url>
# 로컬 저장소에 원격 저장소 주소 추가
# origin은 remote_repo_url에 대한 별칭이다
```

```
SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git remote add origin https://github.com/JAMIEKIMHS/github-practice

SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git remote -v
origin https://github.com/JAMIEKIMHS/github-practice (fetch)
origin https://github.com/JAMIEKIMHS/github-practice (push)
```

git push -u origin master



```
git push
git pull # 변경된 버전만 가져옴
git clone # 전체를 가져옴
```

```
git push -u origin master
# 원격 저장소에 commit 목록을 업로드
# origin이라는 이름의 원격 저장소에 master라는 이름의 브랜치를 push하라.
```

```
SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git status
On branch master

No commits yet

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

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

SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git add .

SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   README.md
```

```
SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git commit -m "first commit"
[master (root-commit) f4babf9] first commit
1 file changed, 7 insertions(+)
create mode 100644 README.md

SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git status
On branch master
nothing to commit, working tree clean

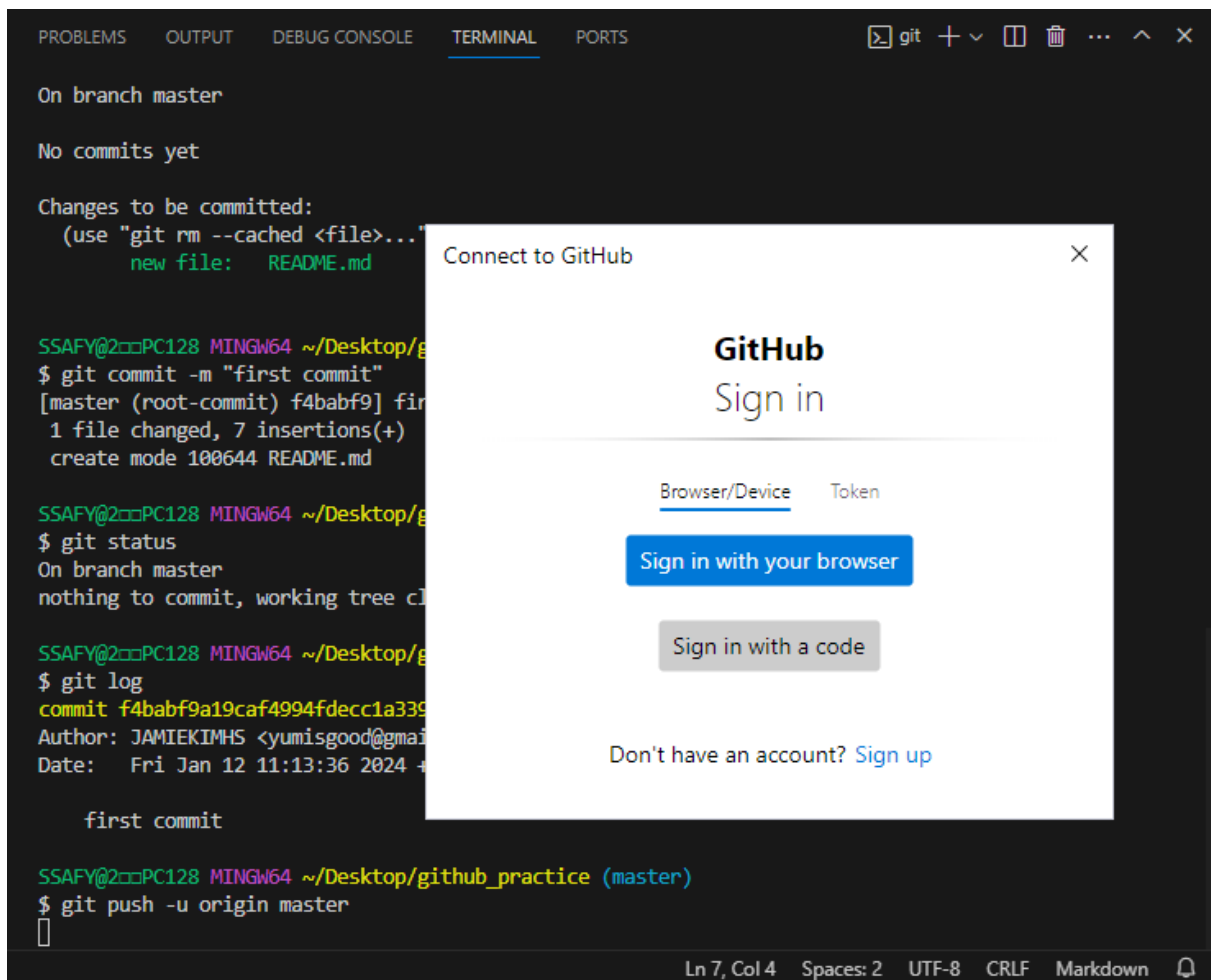
SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git log
commit f4babf9a19caf4994fdecc1a3399734be62db4ab (HEAD -> master)
Author: JAMIEKIMHS <yumisgood@gmail.com>
Date:   Fri Jan 12 11:13:36 2024 +0900

    first commit
```

push를 해보기 위해서 commit까지 올린 상태임

```
git push -u origin master
```

위의 명령어를 쳤더니



Connect to GitHub라는 창이 뜬다

그냥 Sign in with your browser를 클릭하면 내 컴퓨터에서 깃허브 로그인이 되어있으면 push가 된다

```
SSAFY@2024PC128 MINGW64 ~/Desktop/github_practice (master)
$ git push -u origin master
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 273 bytes | 273.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/JAMIEKIMHS/github-practice
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

JAMIEKIMHS / github-practice

Type to search

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

github-practicePublic

master1 Branch0 Tags

Go to file

Add file

Code

JAMIEKIMHSfirst commitf4babf9 · 11 minutes ago1 Commits

README.mdfirst commit11 minutes ago

README

github practice

- 목록1
- 목록2

```
print(1)
```

```
SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

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

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

SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git add .
warning: in the working copy of 'test.py', LF will be replaced by CRLF the next time Git touches it

SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
       new file:   test.py
```

```
SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git commit -m "second commit"
[master e453b86] second commit
1 file changed, 1 insertion(+)
create mode 100644 test.py

SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

SSAFY@200PC128 MINGW64 ~/Desktop/github_practice (master)
$ git push -u origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 291 bytes | 291.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/JAMIEKIMHS/github-practice
f4babf9..e453b86 master -> master
branch 'master' set up to track 'origin/master'.
```

처음에 인증을 받아놔서 두 번째 push때는 인증하라는 창이 다시 뜨지 않는다

github-practice Public

Pin Unwatch 1

master 1 Branch 0 Tags

Go to file t Add file <> Code

JAMIEKIMHS second commit e453b86 · 1 minute ago 2 Commits

README.md	first commit	15 minutes ago
test.py	second commit	1 minute ago

README


github practice

- 목록1
- 목록2

```
print(1)
```

다음과 같이 push가 잘 된 것을 확인할 수 있다

Commits

 master

All usersAll time

Commits on Jan 12, 2024

second commit

JAMIEKIMHS committed 2 minutes ago

e453b86

first commit

JAMIEKIMHS committed 16 minutes ago

f4babf9

Commit

second commit

master

JAMIEKIMHS committed 2 minutes ago

1 parent f4babf9 commit e453b86

Browse files

Showing 1 changed file with 1 addition and 0 deletions.

WhitespaceIgnore whitespaceSplitUnified

test.py










...
...
1 + print(1)

0 comments on commit e453b86

Lock conversation

Write

Preview

H B I         

Leave a comment

Markdown is supported Paste, drop, or click to add files

Comment on this commit

그러면 git이 준 인증키는 어디에 있는가?

자격 증명 관리자

제어판 > 사용자 계정 > 자격 증명 관리자

제어판 홈

자격 증명 관리

웹 사이트, 연결된 응용 프로그램 및 네트워크에 대해 저장된 로그인 정보를 보고 삭제합니다.

웹 자격 증명

Windows 자격 증명

웹 암호

웹 암호가 없습니다.

참고 항목

사용자 계정

자격 증명 관리

웹 사이트, 연결된 응용 프로그램 및 네트워크에 대해 저장된 로그인 정보를 보고 삭제합니다.



웹 자격 증명



Windows 자격 증명

자격 증명 백업(B) 자격 증명 복원(R)

Windows 자격 증명

Windows 자격 증명 추가

Windows 자격 증명이 없습니다.

인증서 기반 자격 증명

인증서 기반 자격 증명 추가

인증서가 없습니다.

일반 자격 증명

일반 자격 증명 추가

git:https://github.com

수정한 날짜: 오늘 (v)

virtualapp/didlogical

수정한 날짜: 2023-12-29 (v)

SSO_POP_Device

수정한 날짜: 2024-01-11 (v)

git:https://github.com 증명서가 일반 자격 증명에 추가된 것을 확인할 수 있다

만약 컴퓨터가 바뀌면 기존의 증명서를 지우고 다시 추가해야 할

원격 저장소에는 commit이 올라가는 것 → commit 이력이 없다면 push 할 수 없다

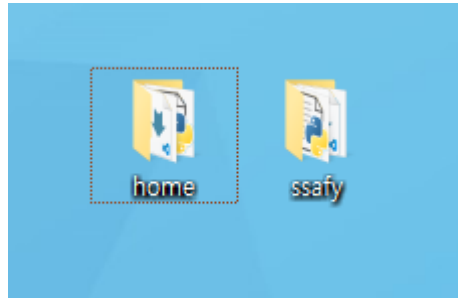
그래서 항상 git status로 확인을 해주자

git pull origin master

- 원격 저장소의 변경사항만을 받아옴(업데이트)
- origin과 같이 별칭을 안 지정하면 계속해서 레포지토리 주소를 입력해야해서 번거롭다

git clone <remote_repo_url>

- 원격 저장소 전체를 복제(다운로드)
- clone으로 받은 프로젝트는 이미 git init이 되어있다



위 두 개의 디렉토리를 각각 home컴퓨터, ssafy컴퓨터로 가정하겠다.

현재 두 디렉토리는 같은 상태이다.

그리고 연결된 하나의 github repository가 있다.

ssafy컴퓨터에서 변경사항을 만들겠다.

```
SSAFY@2020PC128 MINGW64 ~/Desktop/ssafy (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   README.md

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

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

ssafy컴퓨터의 변경사항을 원격 레포지토리에 push하겠다

```
SSAFY@200PC128 MINGW64 ~/Desktop/ssafy (master)
$ git commit -m "20240112"
[master be05501] 20240112
2 files changed, 4 insertions(+), 1 deletion(-)
create mode 100644 a.txt

SSAFY@200PC128 MINGW64 ~/Desktop/ssafy (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

SSAFY@200PC128 MINGW64 ~/Desktop/ssafy (master)
$ git push -u origin master
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 16 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 381 bytes | 381.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/JAMIEKIMHS/github-practice.git
e453b86..be05501 master -> master
branch 'master' set up to track 'origin/master'.
```

```
SSAFY@200PC128 MINGW64 ~/Desktop/ssafy (master)
$ git log --oneline
be05501 (HEAD -> master, origin/master, origin/HEAD) 20240112
e453b86 second commit
f4babf9 first commit
```

home컴퓨터의 commit상태를 보면

```
SSAFY@200PC128 MINGW64 ~/Desktop/home (master)
$ git log --oneline
e453b86 (HEAD -> master, origin/master) second commit
f4babf9 first commit
```

ssafy컴퓨터에서 commit상태를 확인한 것과 다르다

git pull을 하고 다시 확인해보면 같아진 것을 확인할 수 있다

```
SSAFY@2-PC128 MINGW64 ~/Desktop/home (master)
$ git pull origin master
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 4 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (4/4), 361 bytes | 36.00 KiB/s, done.
From https://github.com/JAMIEKIMHS/github-practice
* branch                master      -> FETCH_HEAD
   e453b86..be05501      master      -> origin/master
Updating e453b86..be05501
Fast-forward
 README.md | 5 ++++
  a.txt    | 0
 2 files changed, 4 insertions(+), 1 deletion(-)
 create mode 100644 a.txt

SSAFY@2-PC128 MINGW64 ~/Desktop/home (master)
$ git log --oneline
be05501 (HEAD -> master, origin/master) 20240112
e453b86 second commit
f4babf9 first commit
```

실습

- 현재까지는 로컬에서 origin이라는 원격 저장소 하나에만 보냈는데, 새로운 github repository를 만들고 origin이 아닌 다른 이름으로 원격 저장소를 추가하고 커밋 목록을 push해보자.

```
SSAFY@2-PC128 MINGW64 ~/Desktop/ssafy (master)
$ git remote add second https://github.com/JAMIEKIMHS/second-repo.git

SSAFY@2-PC128 MINGW64 ~/Desktop/ssafy (master)
$ git remote -v
origin  https://github.com/JAMIEKIMHS/github-practice.git (fetch)
origin  https://github.com/JAMIEKIMHS/github-practice.git (push)
second  https://github.com/JAMIEKIMHS/second-repo.git (fetch)
second  https://github.com/JAMIEKIMHS/second-repo.git (push)
```

원격 저장소를 하나 더 만들고 second라는 별칭으로 연결했다

현재 원격 저장소에는 commit이 없고 로컬 저장소에는 commit이 있는 상태이다. 그래서 push를 진행해서 같은 상태로 만들겠다.

```
SSAFY@2-PC128 MINGW64 ~/Desktop/ssafy (master)
$ git log
commit be055015e74ad9009a27bb0ba335256c4622fe46 (HEAD -> master, origin/master, origin/HEAD)
Author: JAMIEKIMHS <yumisgood@gmail.com>
Date:   Fri Jan 12 12:04:19 2024 +0900

    20240112

commit e453b86cc01915c6f4422cb9240c47ceb1a0a6c7
Author: JAMIEKIMHS <yumisgood@gmail.com>
Date:   Fri Jan 12 11:27:55 2024 +0900

    second commit

commit f4babf9a19caf4994fdecc1a3399734be62db4ab
Author: JAMIEKIMHS <yumisgood@gmail.com>
Date:   Fri Jan 12 11:13:36 2024 +0900

    first commit

SSAFY@2-PC128 MINGW64 ~/Desktop/ssafy (master)
$ git push -u second master
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (10/10), 834 bytes | 834.00 KiB/s, done.
Total 10 (delta 1), reused 5 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/JAMIEKIMHS/second-repo.git
 * [new branch]      master -> master
branch 'master' set up to track 'second/master'.
```

gitignore

- Git에서 특정 파일이나 디렉토리를 추적하지 않도록 설정하는 데 사용되는 텍스트 파일
- 프로젝트에 따라 공유하지 않아야 하는 것들도 존재해서

▼ 순서

1. .gitignore 파일 생성(파일명 앞에 '.' 입력, 확장자 없음)

2. b와 c 이름을 가진 텍스트 파일 생성
3. gitignore에 b.txt 작성
4. git init
5. git status

- .gitignore 파일에 작성해둔 파일은 git의 추적을 받지 않는다

```
SSAFY@2024PC128 MINGW64 ~/Desktop/ssafy (master)
$ git status
On branch master
Your branch is up to date with 'second/master'.

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

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

- 이미 git의 관리를 받은 파일이나 디렉토리는 나중에 gitignore에 넣어도 소용이 없다
- 보통은 git init하고 바로 .gitignore를 만든다

gitignore 설정 서비스

- gitignore.io
- gitignore 목록을 만들어주는 사이트

gitignore.io

자신의 프로젝트에 꼭 맞는 .gitignore 파일을 만드세요

운영체제, 개발 환경(IDE), 프로그래밍 언어 검색

생성

[소스 코드](#) | [커맨드라인 문서](#)