GitHub에 올리기 (협업의 시작)

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

뜻 : 현존하는 repository에 올리겠습니다

git remote add origin https://github.com/GitWoogeun/gitpractice.git

```
이 repository에 origin이란 이름의 원격 저장소로 설정하겠다는 의미
origin은 기본값으로 사용(default)

출처:github repository에 있음
```

git remote add origin (해당 Repository URL) == git이랑 github연결

```
git remote add https://github.com/GitWoogeun/Spring.git
```

```
user@SIAT MINGW64 /c/gitproject (master)
$ git remote add origin https://github.com/GitWoogeun/Spring.git
```

git remote -v (Git이랑 원격저장소인 GitHub와의 연결확인)

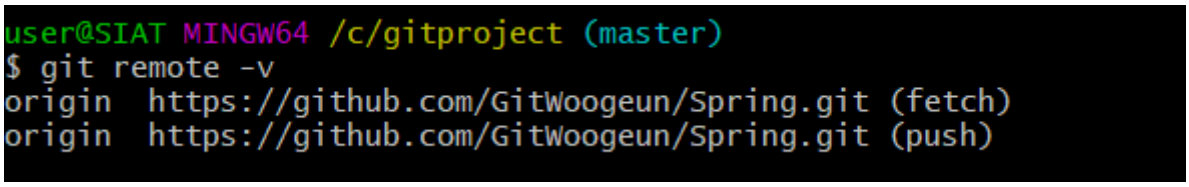
```
git remote -v 연결 확인

git init한 폴더에서 git remote -v로 연결 확인을 하면된다.
밑에 문장처럼
origin https://github.com/GitWoogeun/Spring.git(fetch)
origin https://github.com/GitWoogeun/Spring.git(push)
이런식으로 뜬다면 연결 성공!
```

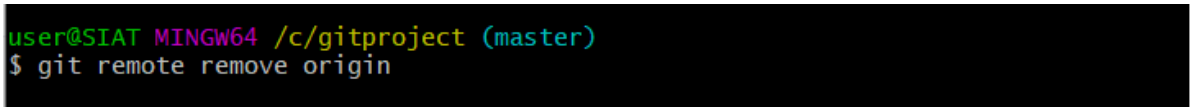
```
user@SIAT MINGW64 /c/gitproject (master)
$ git remote -v
```

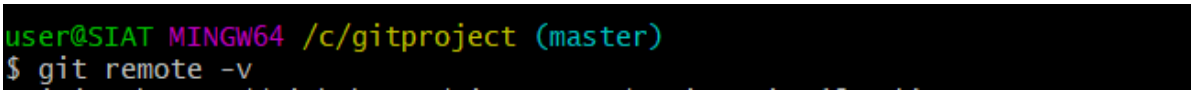
```
user@SIAT MINGW64 /c/gitproject (master)
$ git remote -v
origin  https://github.com/GitWoogeun/Spring.git (fetch)
origin  https://github.com/GitWoogeun/Spring.git (push)
```

git remote remoe origin = 현재 연결된 repository를 삭제

```
git remote remove origin
순서
1). 현재 연결된 github의 repository확인
2). 현재 연결된 github의 repository 삭제
3). 잘 삭제 되었는지 확인

지우는 이유 : 다른 repository에 새로 연결해서 파일을 업로드 하기위해서
```







push 명령어

```
git push -u origin master

현 브랜치에 컷된 내용들을 이 이름의 원격, 즉 이 레파지토리의 이 이름의 브랜치에 올리겠다 거에요~
```

컴퓨터가 처음으로 git-hub에 파일을 업로드 할때는 user.name과 github ID를 물어볼껀데 입력하면된다.

그런 다음에 다시 명령어를 입력해주면된다.

```
git remote add origin https://github.com/GitWoogeun/gitTest.git
git branch -M main
git push -u origin main
```

입력하고 난뒤 이런 메세지가 뜨면 업로드 성공!

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (main)
$ git push -u origin main
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 12 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (18/18), 1.72 KiB | 588.00 KiB/s, done.
Total 18 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/GitWoogeun/gitpractice.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

위에 처음 컴퓨터가 push를 하고 난뒤는 부터는 위에 처럼 설정하면서 올리는것이 아닌 이제 기본 push명령어로 올릴수있음

기본 push 명령어

```
git push origin master

master부분이 main으로 되어있다면 branch main이 main으로 되어있기때문에
master로 바꿔어 주어야한다.

git branch -M master  (브랜치의 메인이 master라고 선언하는 명령어)
```

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ git push origin master
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 309 bytes | 309.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:      https://github.com/GitWoogeun/gitpractice/pull/new/master
remote:
To https://github.com/GitWoogeun/gitpractice.git
 * [new branch]      master -> master
```

.gitignore = Git에 올리지않을 파일들

```
제일먼저 .gitignore파일을 하나 만든다

echo 올리지않을파일>.gitignore

그 다음 올리지 않을 파일을 .gitignore파일에 집어 넣는다. (ex: 올리지않을파일.txt)

echo 올리지않을파일.txt>>.gitignore

이렇게하면 올리지않을파일.txt파일은 .gitignore로 들어가기 때문에 git에서 사라지고 commit을 할수없게 된다.
```

.gitignore파일생성

```
user@DESKTOP-D3NC14R MINGW64 /d/project/git (master)
$ echo d.txt>.gitignore
```

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ git status
On branch master
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

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

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

올리지않을 파일 생성

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ echo 올리지않을파일>올리지않을파일.txt

user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ git status
On branch master
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        "\354\230\254\353\246\254\354\247\200\354\225\212\354\235\204\355\214\21
4\354\235\274.txt"
```

.gitignore에 올리지않을파일 집어넣기

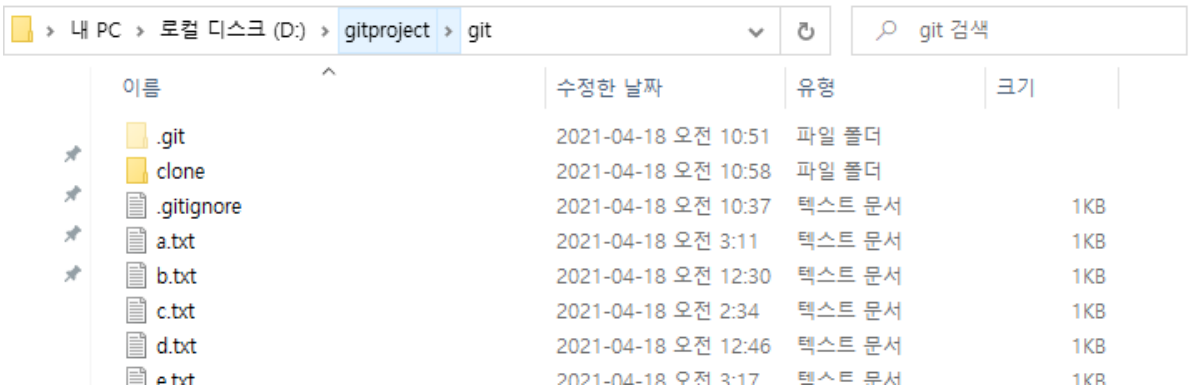
```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ echo 올리지않을파일.txt>>.gitignore

user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ git status
On branch master
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

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

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

git clone url(repository) = 해당 폴더와 똑같은 파일들을 복사

먼저 repository의 (복사)클론파일 저장할 폴더를 하나 만들고

그 폴더로 들어가서 git clone repository(주소)하게 되면 해당 repository를 복사해서 가져올 수있음

먼저 clone할 폴더 생성 (폴더 이름 : clone으로 지정했음)



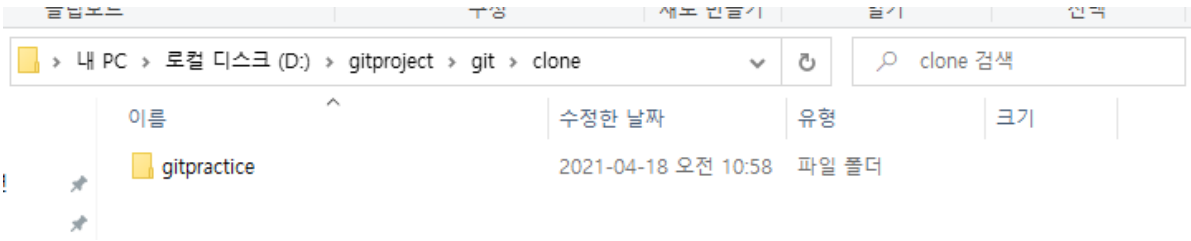
clone폴더로 들어가기

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ cd clone
```

clone폴더에 들어가서 clone작업하기

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git/clone (master)
$ git clone https://github.com/GitWoogeun/gitpractice.git
Cloning into 'gitpractice'...
remote: Enumerating objects: 27, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 27 (delta 5), reused 26 (delta 4), pack-reused 0
Receiving objects: 100% (27/27), done.
Resolving deltas: 100% (5/5), done.
```

clone폴더 안에 repository가져오기 성공!!



내 repository에 누군가 파일을 올렸다!! (받을파일.txt파일 누군가 올렸다.) (협업의 시작)

git fetch : github(내 repository)에 새로운 파일이 올라왔는지 확인!

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ git fetch
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (1/1), done.
remote: Total 3 (delta 1), reused 3 (delta 1), pack-reused 0
Unpacking objects: 100% (3/3), 289 bytes | 2.00 KiB/s, done.
From https://github.com/GitWoogeun/gitpractice
   7d9892b..0946845  master     -> origin/master
```

그 다음 git status로 확인 하면된다.

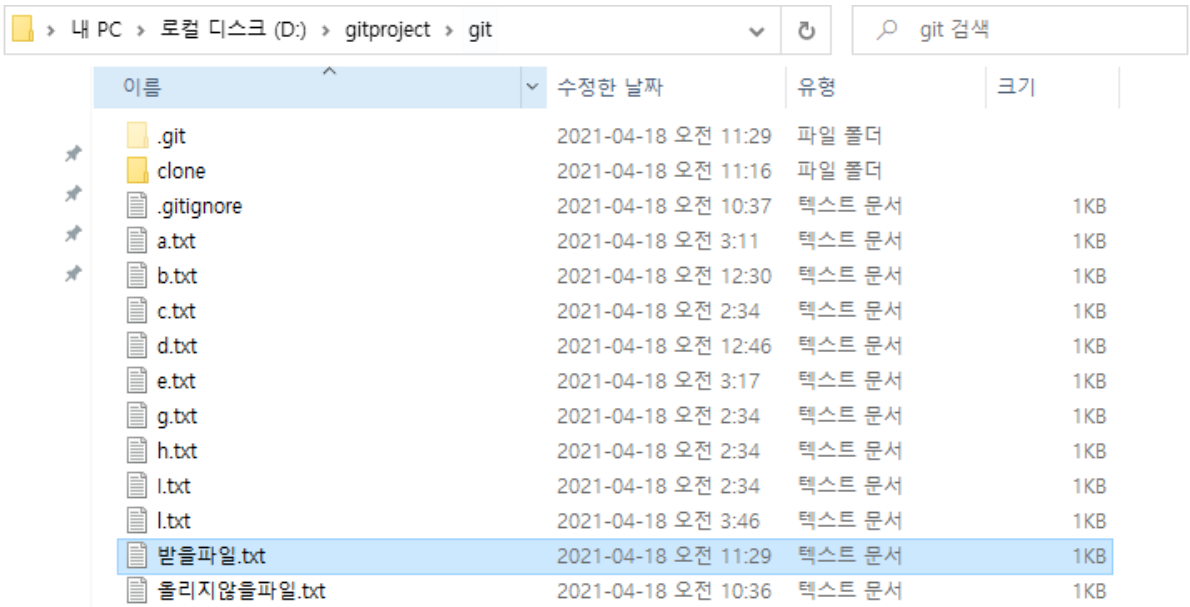
```
git status

Your branch is behind 'origin/master' by1 commit, add can be fast-forwarded. 라고 뜰것이다.
이 브랜치가 원격 origin의 마스터에 커밋 하나가 뒤쳐져 있다고 말해주는 것이다.
GitHub에서 다운 받아야 할 사항이 있다는 얘기에요
```

git pull origin master (다른쪽에서 내 repository에 올린 파일을 내 pc에 저장)

```
git pull (원격명) (브랜치명)을 입력합니다.
그럼 뒤쳐진 커밋파일을 새로 다운받아진다. (내 ropository에 다른사람이 올린파일)
```

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ git pull origin master
From https://github.com/GitWoogeun/gitpractice
 * branch            master     -> FETCH_HEAD
Updating 7d9892b..0946845
Fast-forward
 "\353\260\233\354\235\204\355\214\214\354\235\274.txt" | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 "\353\260\233\354\235\204\355\214\214\354\235\274.txt"
```

그럼 이렇게 내 컴퓨터에 받을파일.txt 이 저장이 될것이다.



브랜치 주고 받기 (엄청 심화버전)

git checkout -b 브랜치명

```
git checkout -b developking
이렇게하면 바로 developking 브랜치가 만들어져서 체크아웃까지 됩니다.
```

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ git checkout -b developking
Switched to a new branch 'developking'

user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$
```

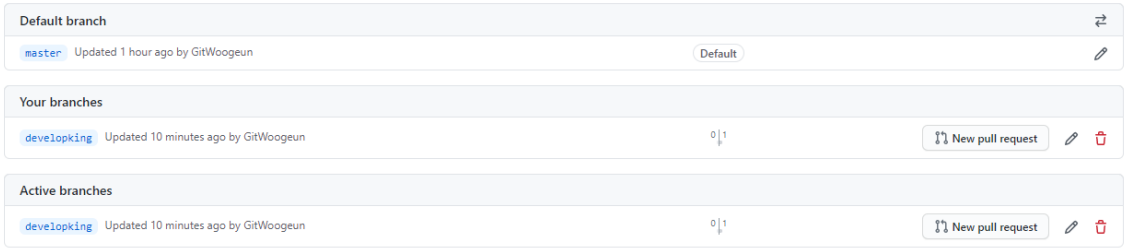
developking이라는 브랜치에서 clone폴더 저장

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$ git status
On branch developking
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore
        clone/

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

user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$ git add clone/
warning: adding embedded git repository: clone/gitpractice
hint: You've added another git repository inside your current repository.
hint: Clones of the outer repository will not contain the contents of
hint: the embedded repository and will not know how to obtain it.
hint: If you meant to add a submodule, use:
hint:
hint:   git submodule add <url> clone/gitpractice
hint:
hint: If you added this path by mistake, you can remove it from the
hint: index with:
hint:
hint:   git rm --cached clone/gitpractice
hint:
hint: See "git help submodule" for more information.

user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$ git status
On branch developking
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   clone/gitpractice

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


user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$ git commit -m "clone폴더저장"
[developking 806b278] clone폴더저장
 1 file changed, 1 insertion(+)
 create mode 160000 clone/gitpractice

user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$ git status
On branch developking
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .gitignore

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

user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$
```

banch = developking을 github에 만들어서 파일을 올리겠다.

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$ git branch
  another
* developking
  master
  my-another-idea
  my-idea
  my-ideaTwo

user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$ git push origin developking
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 12 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 317 bytes | 317.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'developking' on GitHub by visiting:
remote:      https://github.com/GitWoogeun/gitpractice/pull/new/developking
remote:
To https://github.com/GitWoogeun/gitpractice.git
 * [new branch]      developking -> developking

user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$
```

developking으로 파일 올라감



git branch 와 git branch -a의 차이

```
git branch = 로컬저장소에있는 branch만 볼수 있지만
git branch -a = 로컬저장소와 원격저장소에 저장된 branch를 볼수 있습니다
```

git branch -a = 원격저장소(github)에 저장된 branch 보기

```
git branch -a
```

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$ git branch -a
  another
* developking
  master
  my-another-idea
  my-idea
  my-ideaTwo
  remotes/origin/developking
  remotes/origin/main
  remotes/origin/master

user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (developking)
$
```

다른쪽에서 저장된 git branch로 이동하기

```
user@DESKTOP-D3NC14R MINGW64 /d/gitproject/git (master)
$ git checkout -b developking origin/developking
```

충동 해결하기 (이건 나중에 센터에서 해보기로 컴퓨터 하나로는 힘듬..)