



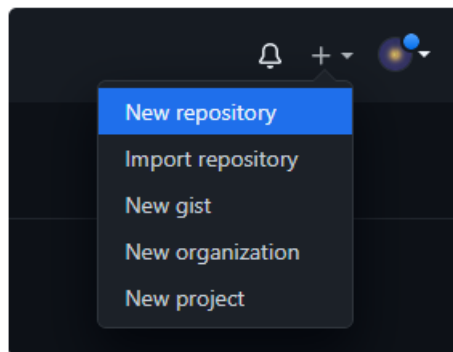
Git

파일 업로드 하기

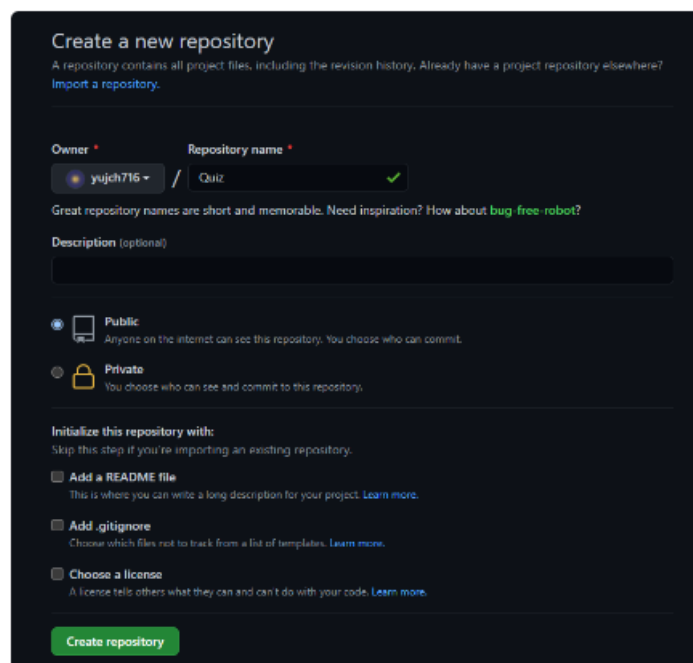
1. git 설치

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

2. GitHub에 새 저장소를 만든다.

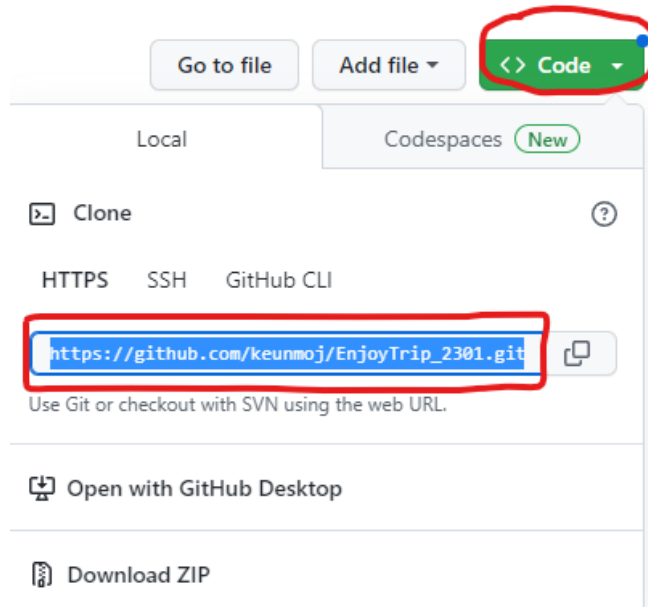


왼쪽 상단에서 클릭

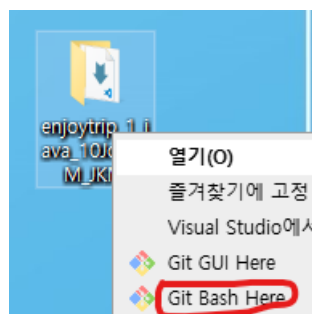


저장소 이름을 정하고 생성한다.

3. 생성된 저장소의 주소를 기억한다.



4. 업로드 하고 싶은 프로젝트 폴더 우클릭 → Git Bash Here



5. 초기 설정



git config --global user.name "유저이름" (나는 keunmoj)

git config --global user.email "유저 이메일"(나는 rmsah0612@gmail.com)

```
SSAFY@DESKTOP-DOGVPU MINGW64 ~/Downloads/enjoytrip_1_java_10Jo_LSM_JKM (main)
$ git config --global user.name "keunmoj"

SSAFY@DESKTOP-DOGVPU MINGW64 ~/Downloads/enjoytrip_1_java_10Jo_LSM_JKM (main)
$ git config --global user.email "rmsah0612@gmail.com"
```

6. 파일 준비 및 commit



git init #.git 파일 생성

git add . #선택한 프로젝트 폴더 내의 모든 파일 관리

-> 특정파일만 하고 싶다면 git add 파일이름.파일형식 ex) git add a.txt

git status #상태확인

git commit -m "주석" #커밋

```
SSAFY@DESKTOP-DOGVPU MINGW64 ~/Downloads/enjoytrip_1_java_10Jo_LSM_JKM (main)
$ git init
Reinitialized existing Git repository in C:/Users/SSAFY/Downloads/enjoytrip_1_java_10Jo_LSM_JKM/.git/

SSAFY@DESKTOP-DOGVPU MINGW64 ~/Downloads/enjoytrip_1_java_10Jo_LSM_JKM (main)
$ git add .

SSAFY@DESKTOP-DOGVPU MINGW64 ~/Downloads/enjoytrip_1_java_10Jo_LSM_JKM (main)
$ git commit -m "first upload"
On branch main
nothing to commit, working tree clean
```

- On branch main ⇒ branch가 main으로 설정되어 있다는 뜻.
- nothing to commit, working tree clean ⇒ 업로드 하려고 하는 파일에 변경사항이 없어 commit할 내용이 없다는 뜻

7. 업로드 하기(push)



git remote add origin {위 3번에서 저장한 깃허브 저장소 주소}

git push -u origin master

```
SSAFY@DESKTOP-DOGVPU MINGW64 ~/Downloads/enjoytrip_1_java_10Jo_LSM_JKM (main)
$ git remote add origin https://github.com/keunmoj/EnjoyTrip_2301.git


SSAFY@DESKTOP-DOGVPU MINGW64 ~/Downloads/enjoytrip_1_java_10Jo_LSM_JKM (main)
$ git push -u origin main
Enumerating objects: 188, done.
Counting objects: 100% (188/188), done.
Delta compression using up to 12 threads
Compressing objects: 100% (167/167), done.
Writing objects: 100% (188/188), 2.92 MiB | 5.63 MiB/s, done.
Total 188 (delta 28), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (28/28), done.
To https://github.com/keunmoj/EnjoyTrip_2301.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```


오류 및 해결

1. Git Remote origin already exists 에러 해결방법

```
SSAFY@DESKTOP-DOGVPU8 MINGW64 ~/Downloads/enjoytrip_1_java_10Jo_LSM_JKM (main)
$ git remote add origin https://github.com/keunmoj/EnjoyTrip_2301.git
error: remote origin already exists.
```

- 의미: 이미 등록된 origin이 있다. 그리고 해당 origin과 다른 origin을 입력했을 때 발생한다.
- 해결방법 : 이미 등록된 origin을 삭제하고 새로 등록한다.

 \$ git remote remove origin ⇒ 등록된 origin 삭제


 \$ git remote add origin {주소} ⇒ 새로운 origin 등록


```
SSAFY@DESKTOP-DOGVPU8 MINGW64 ~/Downloads/enjoytrip_1_java_10Jo_LSM_JKM (main)
$ git remote remove origin

SSAFY@DESKTOP-DOGVPU8 MINGW64 ~/Downloads/enjoytrip_1_java_10Jo_LSM_JKM (main)
$ git remote add origin https://github.com/keunmoj/EnjoyTrip_2301.git
```

2. error:failed to push some refs to 에러 해결방법

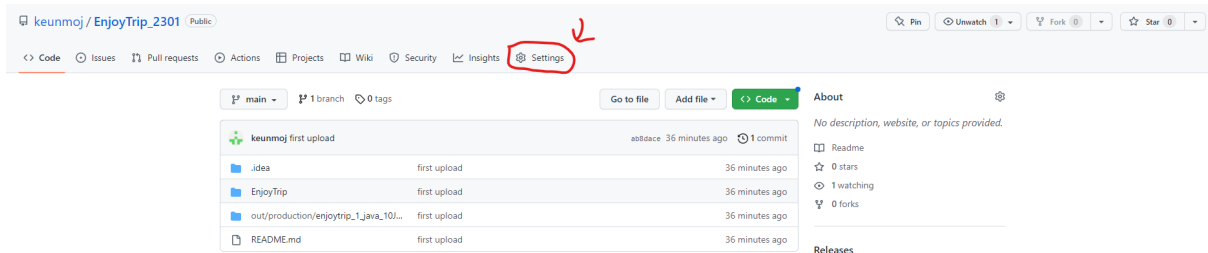
- 의미: push하려고 하는 github에 내 local에는 없는 파일이 존재하는 경우 (내 local에는 test1.java가 없는데 팀원이 github에 해당 파일을 업로드 했을 경우 발생)
- 해결방법: github 파일을 내 local로 pull하고 다시 push한다.

 \$ git pull {origin 원격 저장소 주소} {branch (main, master...)} ⇒ 원격저장소의 파일을 내 local로 pull

 \$ git push {origin 원격 저장소 주소} {branch (main, master...)} ⇒ 내 local 파일을 원격저장소에 push

레파지토리 삭제하기

1. repository 내부의 settings를 클릭



2. 스크롤을 내려서 Danger Zone 찾기

Danger Zone

Change repository visibility
 This repository is currently public.
 Change visibility

Transfer ownership
 Transfer this repository to another user or to an organization where you have the ability to create repositories.
 Transfer

Archive this repository
 Mark this repository as archived and read-only.
 Archive this repository

Delete this repository
 Once you delete a repository, there is no going back. Please be certain.
 Delete this repository

3. Delete this repository 클릭

Danger Zone

Change repository visibility
 This repository is currently public.
 Change visibility

Transfer ownership
 Transfer this repository to another user or to an organization where you have the ability to create repositories.
 Transfer

Archive this repository
 Mark this repository as archived and read-only.
 Archive this repository

Delete this repository
 Once you delete a repository, there is no going back. Please be certain.

Delete this repository

4. 삭제 완료