

Git & Github

전범위 강의자료

<https://github.com/2tle>

이 자료는 <https://aonee.tistory.com/15> 를 참고하여 제작됨.

Git이란?

코드 저장소

이력을 관리하는 저장소

Version Control System

코드 저장 및 저장시점으로 돌아갈 수 있음

분산 버전관리시스템



Git을 사용하는 이유

과제_완성_최종_최종_진짜최종_이게최종

- 최종파일을 만들었지만 수정해야 할 일이 생김
- 파일명에 최종을 계속 붙이면 어느 파일이 최종인지 확인하기 어려움

파일 하나로 합치기

- 하나의 파일로 합치기 힘들고 그 과정에서 오류나기 쉬움
- 무엇보다 내가 하면 시간이 아까움

Github란?

Git을 웹에 옮겨 놓은 원격저장소(Remote Repository)

=> 협업 가능

=> 코드 복구 가능

버전관리가 뭐죠?

버전 관리

- 파일 변화를 시간에 따라 기록했다가 특정 시점의 버전을 다시 가져올 수 있음

장점

- 소스코드 변경이력 쉽게 확인 가능
- 특정 시점으로 돌아갈 수 있음
- 편집 내용 충돌시 덮어쓰기 위험에서 벗어나 해결 가능

Local? Remote? Repository?

Remote Repository

- 원격 저장소
- 파일이 서버에서 관리되며 여러 사람이 함께 공유 가능

Local Repository

- 로컬 저장소
- 내PC에 파일이 저장되는 개인 전용 저장소

Github에서 저장소를 만들어보자!

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

Overview **Repositories 38** Projects Packages Stars 46

Hyunju Yang
2tle

Edit profile

10 followers · 12 following

Sunrin Internet High School
Yongsan-gu, Seoul
iam@2tle.io
https://2tle.io

Achievements

2tle / README.md

I am a highschool student studying and developing back-end and native app.
And my real name is Hyunjun Yang, not Hyunju Yang...

Stack
Stats
devLog by 현주(blog)
hits 27 / 255

Pinned

2tlebot Public
JavaScript ☆ 1

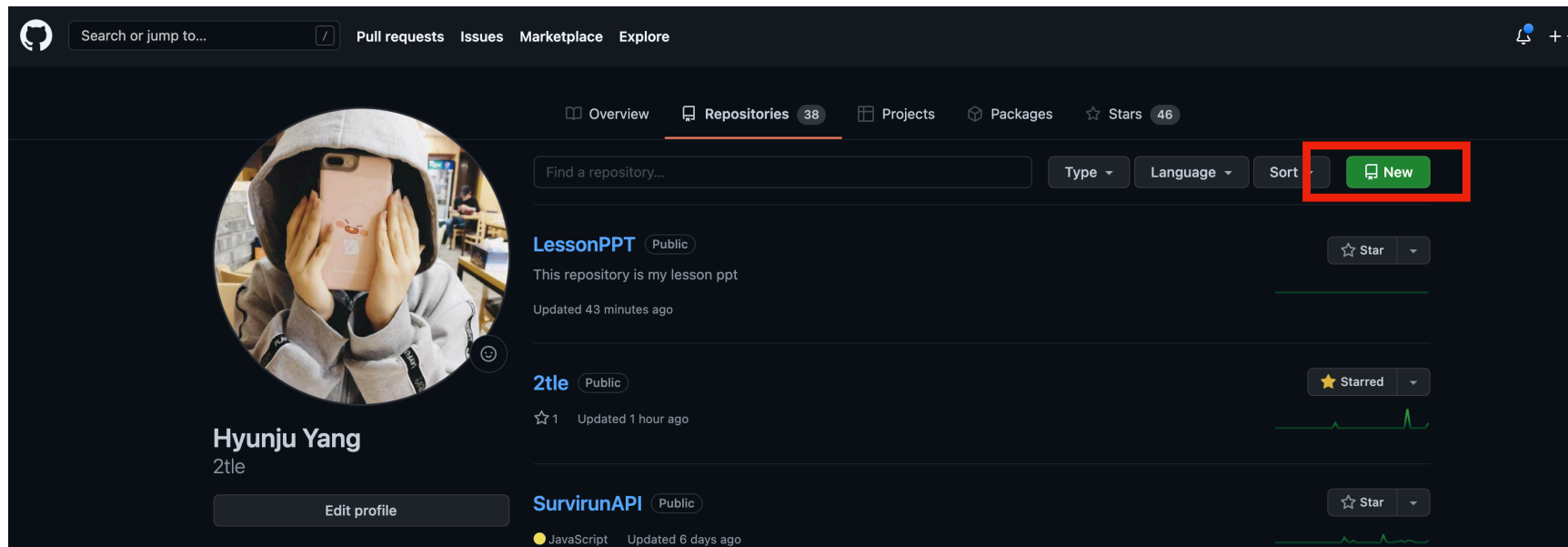
blacklamp Public
PHP ☆ 1

miniDB-for-JS Public
JavaScript

Dalvodre Public
Kotlin

Customize your pins

Github에서 저장소를 만들어보자!



Github에서 저장소를 만들어보자!

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 *

2tle /

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

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:

Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Owner : 소유자

Repo. name: 저장소 이름

Description: 설명

Public: 내 저장소를 모두에게 공개

Private: 내 저장소를 볼 사람을 선택해서 공개

README.md: 프로젝트에 대한 설명

.gitignore: 깃에서 커밋하지 않을(무시할) 파일을 정하는 파일

license: 이 프로젝트에 대한 라이선스

Github에서 저장소를 만들어보자!

Git을 설치하고 나서 가장 먼저 해야하는 것!!

```
git config --global user.name "GITHUB_NAME"
```

- 커밋시 보여줄 사용자 이름 설정

```
git config --global user.email "GITHUB_EMAIL"
```

- 커밋시 보여줄 사용자 이메일 설정

Github에서 저장소를 만들어보자!

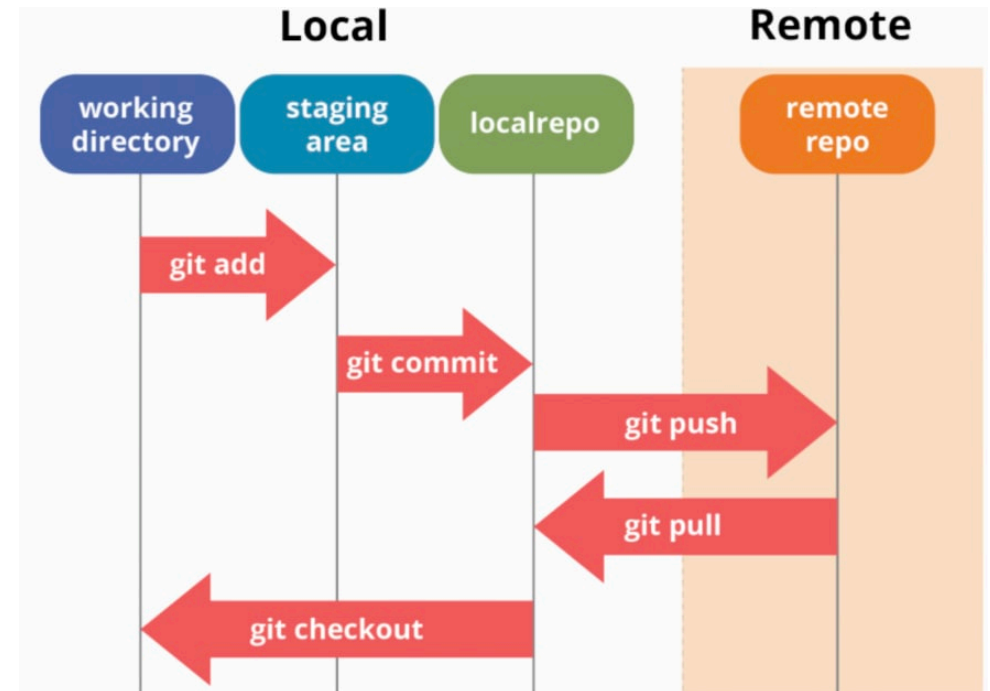
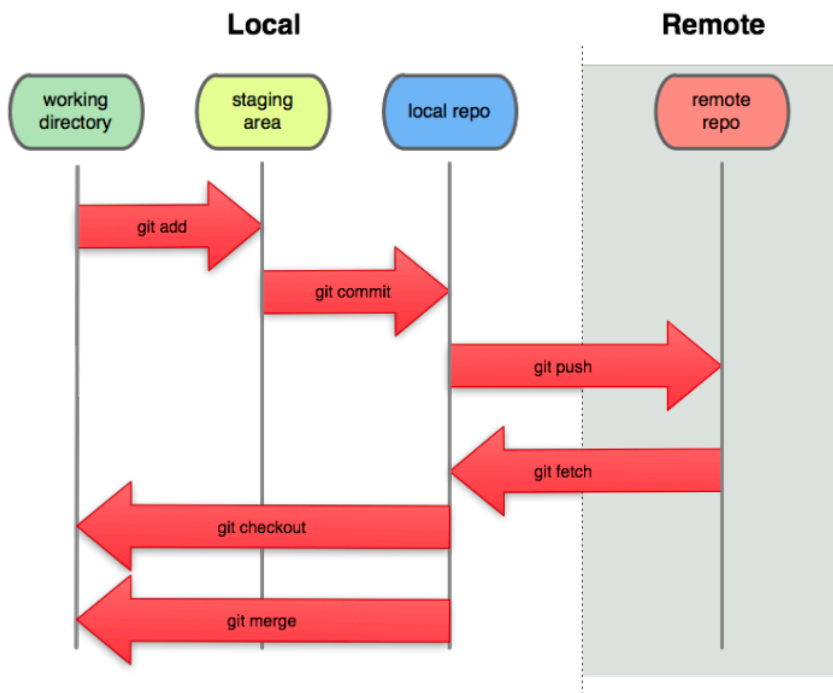
git init

- Git 초기화

git remote add origin 원격저장소_주소

- 원격 저장소와 연결
- ex) git remote add origin <https://github.com/2tle/SomeRepo.git>

git 돌아가는 과정?



add

저장소에 파일을 추가할 때 사용

git add .

- 디렉터리 내에 모든 파일 추가(스테이징 영역으로 넘김)

git add 경로/파일이름

- 특정 파일만 추가(스테이징 영역으로 넘김)

commit

스테이징 영역에 있는 내용을 로컬 저장소에 저장

```
git commit -m "commit message"
```

- 커밋
- 커밋메시지는 좀 간략하고 알아보기 쉽게!!

push

로컬 저장소에 있는 내용을 원격 저장소에 저장

git push 원격_저장소_이름 원격_저장소_브랜치_이름

- ex) git push origin master

pull 과 fetch

pull과 fetch는 모두 원격 저장소에서 로컬 저장소로 가져오는 역할
차이점

- pull은 병합(merge)까지 해줌
- fetch는 직접 확인을 하고 본인이 직접 병합을 해야함

git pull origin main

git fetch origin main

Fork

Fork란?

- 타깃 레포지토리를 자신의 레포지토리로 가져오는 것

Pull Request

Pull Request란?

- 누군가 수정한 브랜치를 가져가 검토후 main 브랜치에 병합하는 것
- 코드 충돌 최소화 할 수 있음
- Push 권한이 없는 오픈소스 프로젝트에 기여할 때 많이 사용됨

Pull Request 사용하는 방법

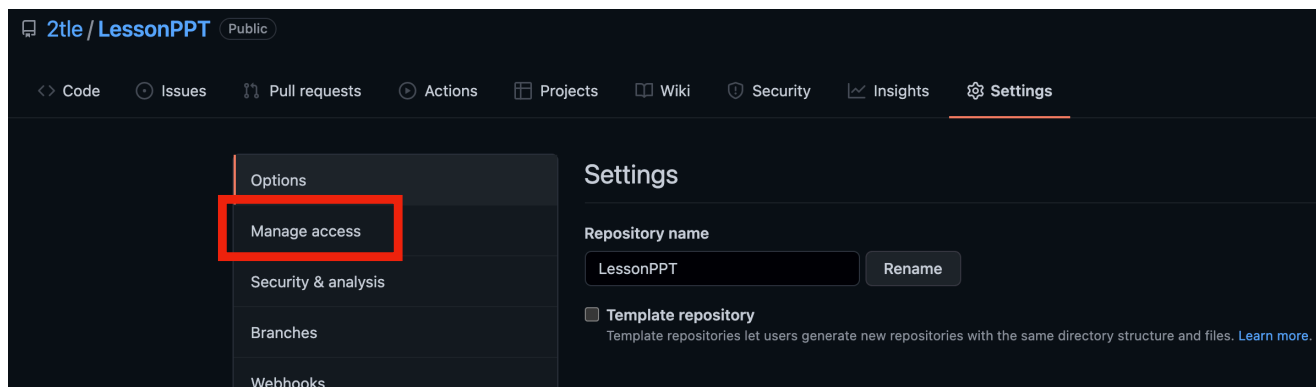
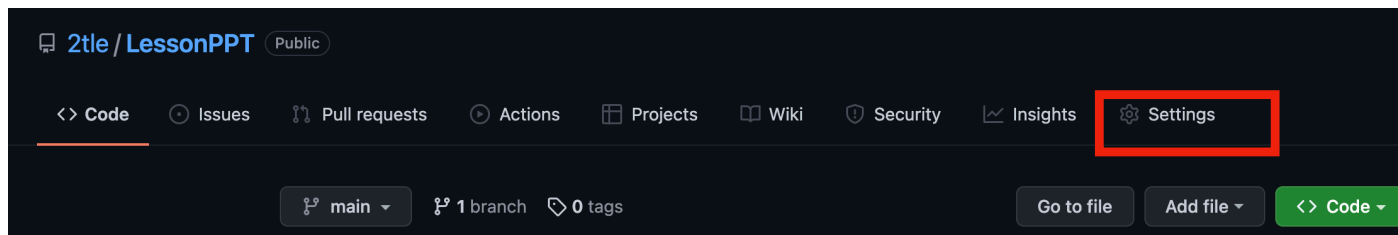
Fork -> 새 브랜치 생성 -> 작업(add,commit,push...)

-> PR생성 -> 병합(Merge Request)

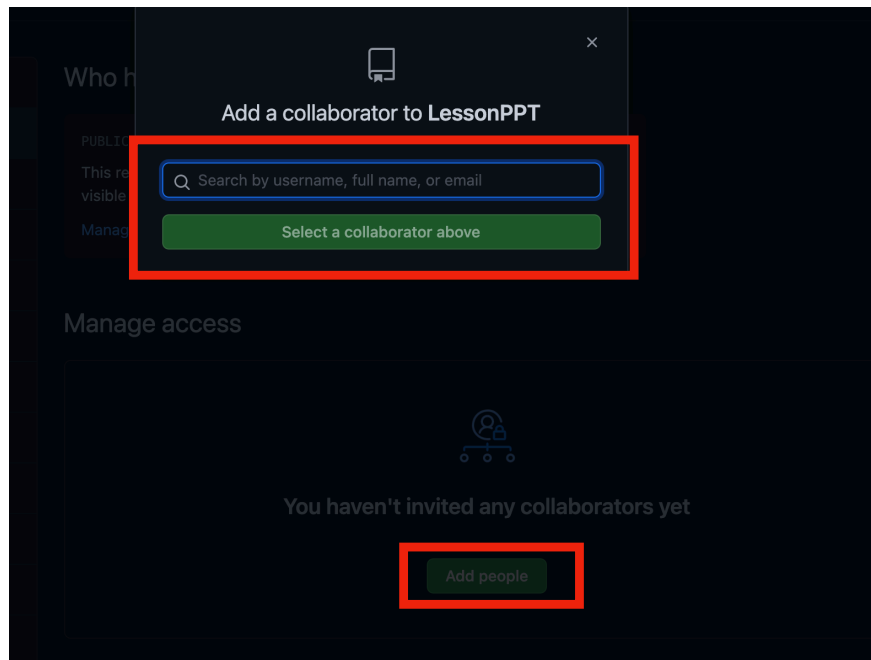
-> 병합 이후 동기화 및 작업한 브랜치 삭제

Github 협업 하는 방법

저장소에 유저 초대하기



저장소에 유저 초대하기



이메일에서 수락해야함