

Intro



Git & Github 세미나

클라이언트

Android
iOS
Vue.js
React.js
Unity3D

클라우드

Docker , Heroku , Python AnyWhere, AWS, Azure

서버사이드

Node & Express
Node & Koa
Rust & Rocket
Go & Echo
Go & Revel
Python & Django
Python & Flask

디자인

Ai & Ps
Sketch3
AfterEffects
Cinema4D

Git

Git 만 먼저 알아봅시다

Git

형상관리툴 또는 버전관리툴

Save 포인트를 만드는 것!

Git

최초버전(v0.1)

뷰 추가(v0.2)

뷰 간신

기본

네트워크

결제모듈

결제모듈
추가
(v0.4)

네트워크 추가(v0.3)

Git

init : 해당 프로젝트에 git을 사용함. init과 동시에
Git은 로컬에 저장소를 생성해줌 (세이브 폴더)

로컬 git 저장소

Git

commit : 로컬 저장소로 변경된 내용을 보냄 (세이브)

로컬 git 저장소

추가함

프로젝트

a.txt
b.txt

Git

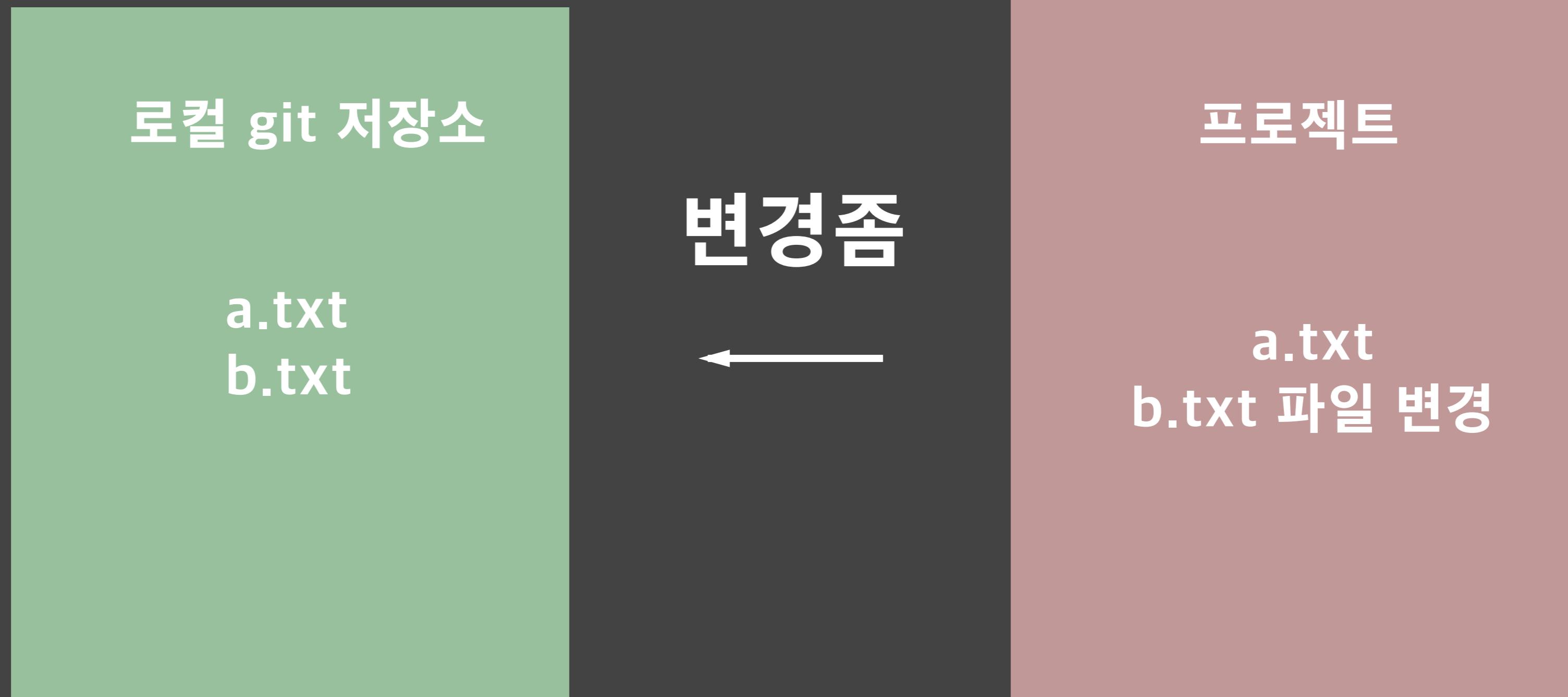
커밋 완료 후

로컬 git 저장소

a.txt
b.txt

Git

새로 추가되는 내용뿐만 아니라, 변경사항이 있다면 변경됨



Git

커밋 완료 후

로컬 git 저장소

a.txt

b.txt

(변경된 파일이
저장됨)

Git

!!!!문제점!!!!

저장소(세이브 파일) 이 한개임

두 가지 문제가 발생

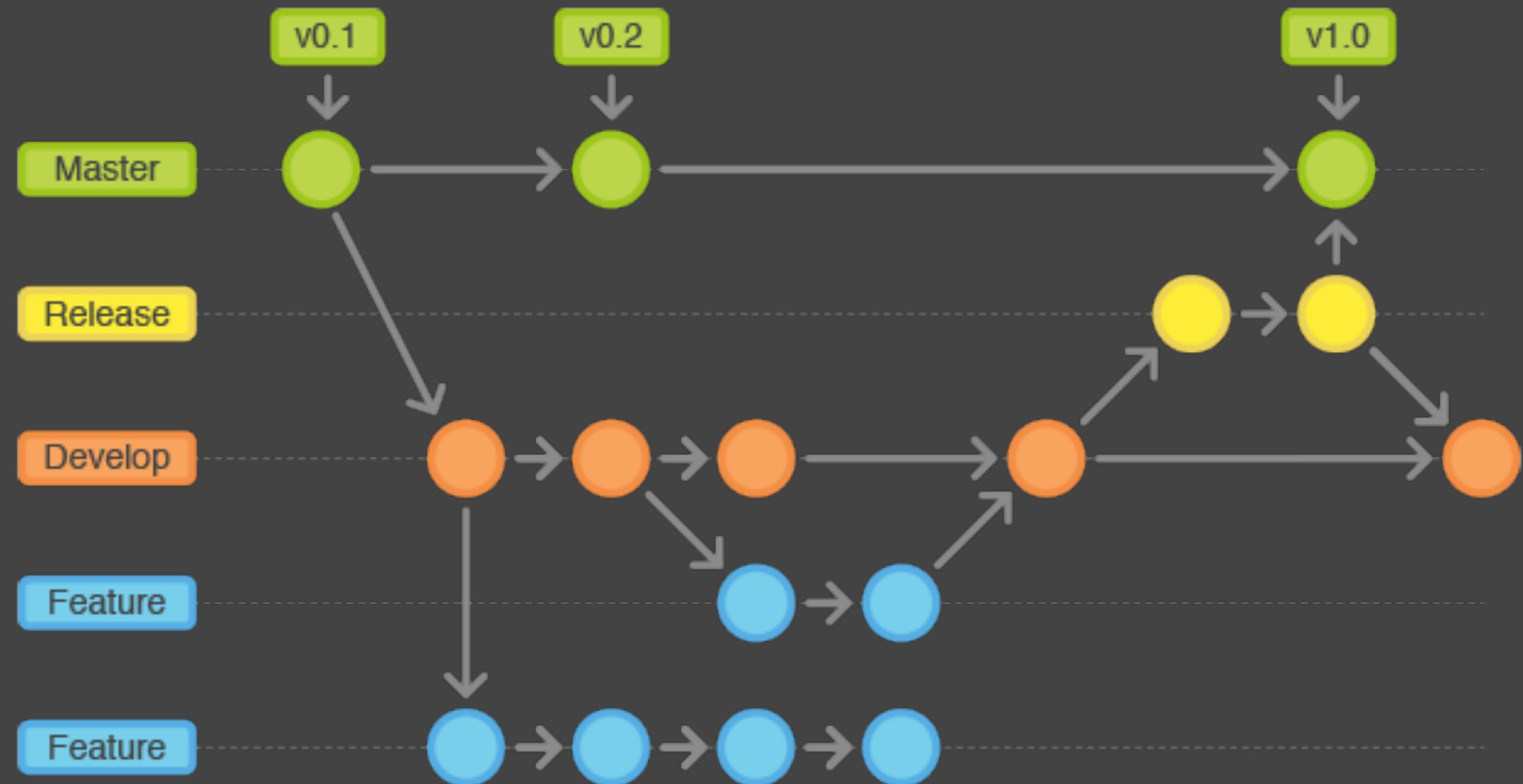
실수(잘못 저장) 했을 때.. 돌아갈 수 없다



저장 이력을 볼 수 없다. (이 또한 돌아갈 수 없다)



브랜치 (나뭇가지) : 새로운 저장공간을 만듬!

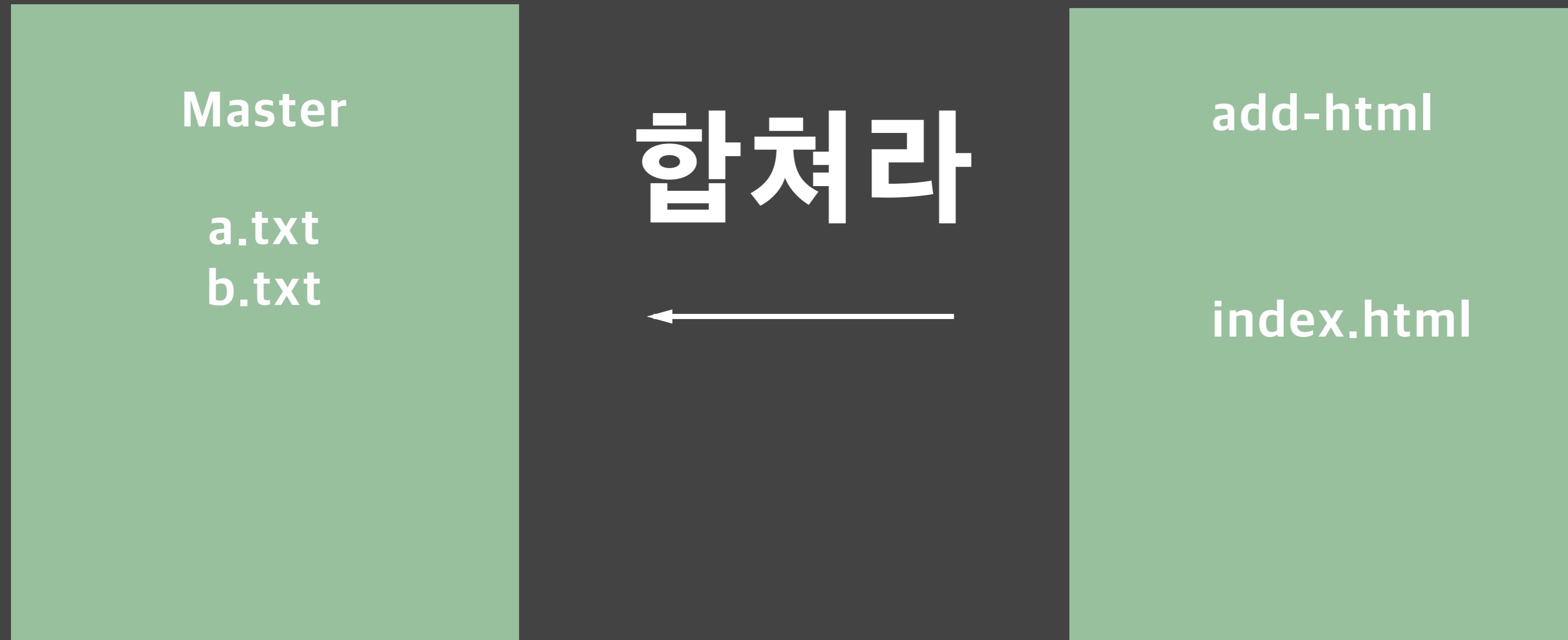


**마스터 브랜치 : 중앙 저장소, 모든 브랜치는 결국 중앙저장소에 저장되어야 반영됨
(Android, iOS 의 Main Thread 개념)**

그냥 브랜치 : 추가 저장소, 모든 브랜치는 작업 완료 후, 마스터 브랜치에 합쳐야함

Git

merge : 브랜치를 합병함 (세이브 파일 합침)



Git

merge 완료후

Master

a.txt
b.txt
index.html

Git



Github

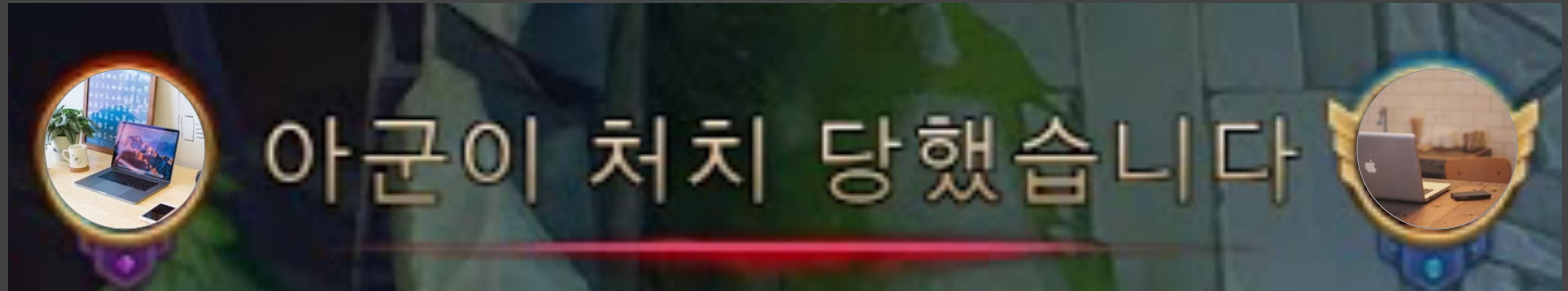
브랜치도 잘 만들었고, Merge도 잘했고, 완벽해!

이제 프로젝트에 개짓거리를 해도 문제없어!



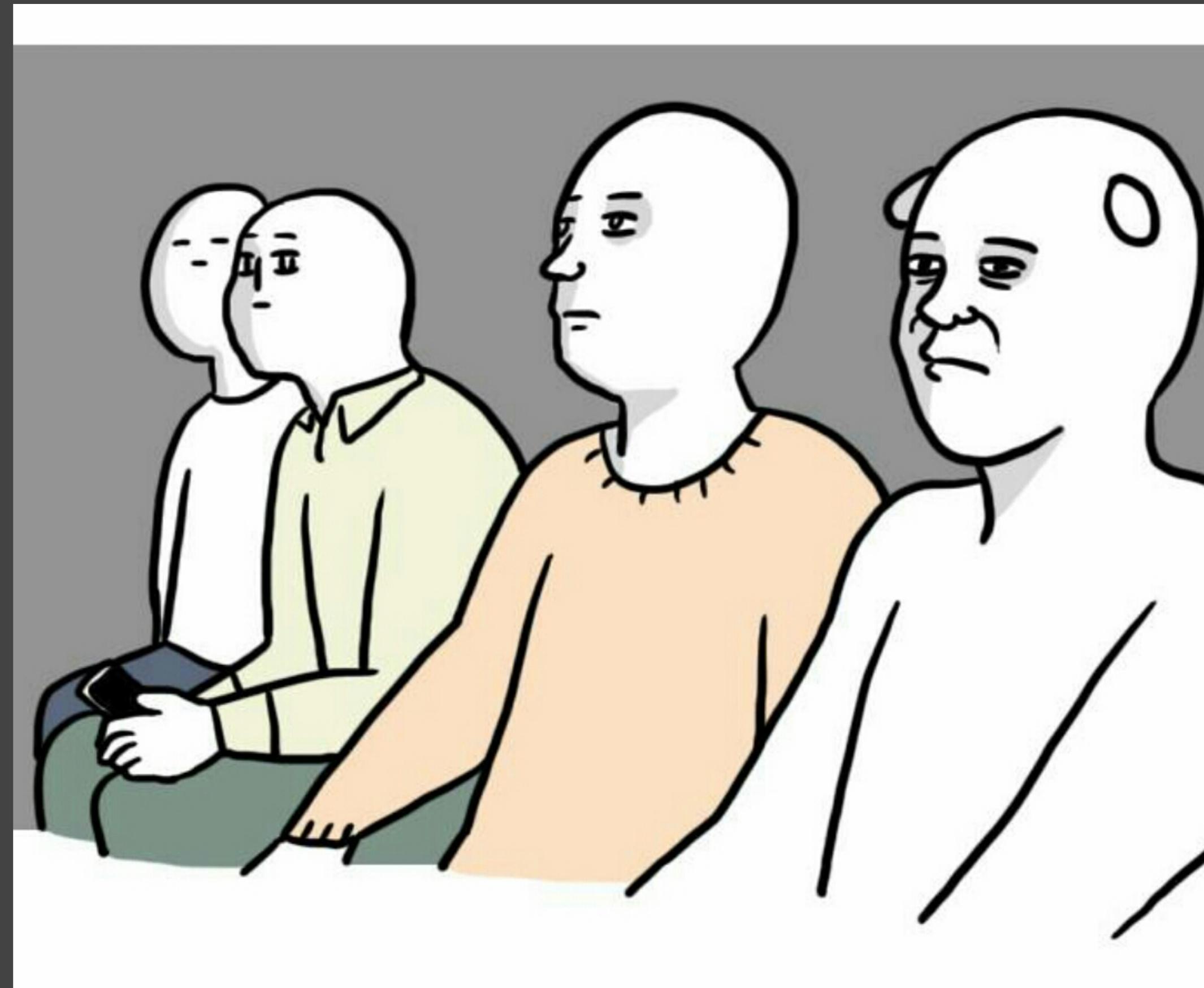
Git

근데 노트북을 잃어버림 ㅋ



Git

?



Github : 온라인 저장소(원격, remote)를 이용할 수 있도록 해주는 서비스 (+ SNS)



Github 말고도, bitbucket 같은 서비스도 있음

Git

깃허브 사용시 장점

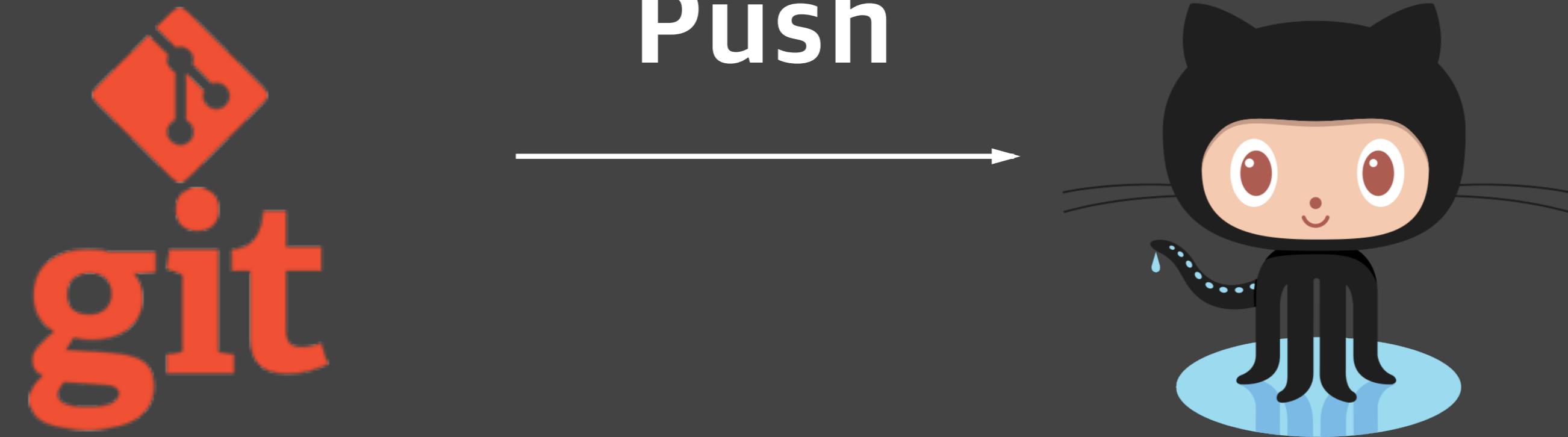
온라인에 올렸으니 안전함!

기기에 제한 받지 않음
(다른컴퓨터에서 해도됨)

여러사람과 하나의 저장소를 공유할 수 있음(*)

Git

Push : 로컬 저장소를 원격 저장소에 올림



Git

Pull , Fecth : 원격 저장소를 로컬 저장소에 가져옴



Fetch, Pull



Clone : 원격 저장소를 내 컴퓨터에 복사함



pull, fetch는 local 저장소와 remote 저장소가
연동이 되어있어야함.
Clone은 없어도댐.

Git

Github로 협업하기 실습

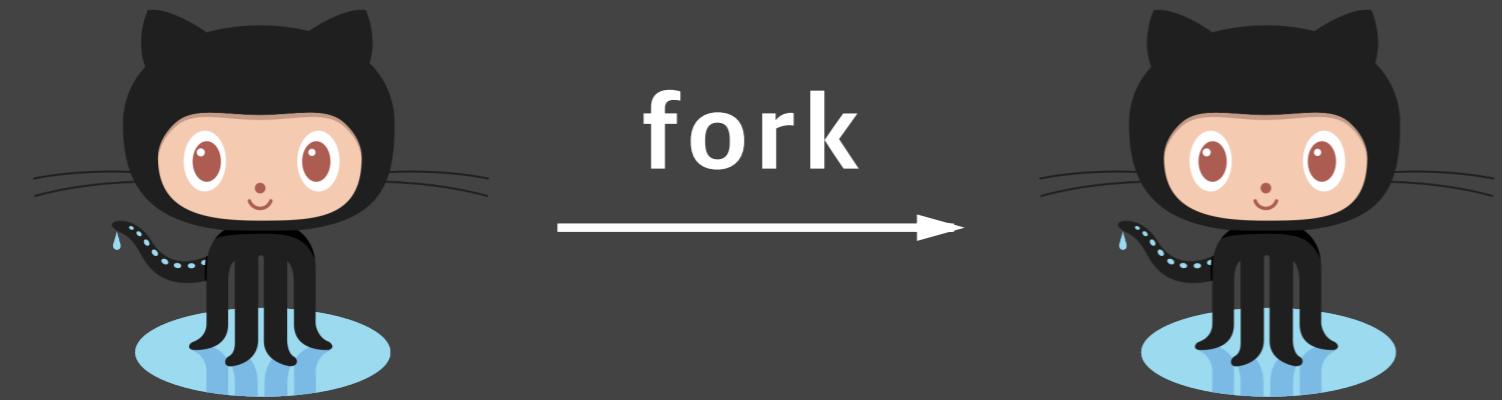
프로세스 이해하기

Git



1. Github 저장소를 만듬

Git



1. Github
저장소를 만듬
(origin)

2. 원저장소를
내 원격에 복사

Git



1. Github
저장소를 만듬
(origin)

2. 원저장소를
내 원격에 복사

3. 내 원격 저장소를
로컬로 복사

Git



1. Github
저장소를 만듬
(origin)

2. 원저장소를
내 원격에 복사

3. 내 원격 저장소를
로컬로 복사



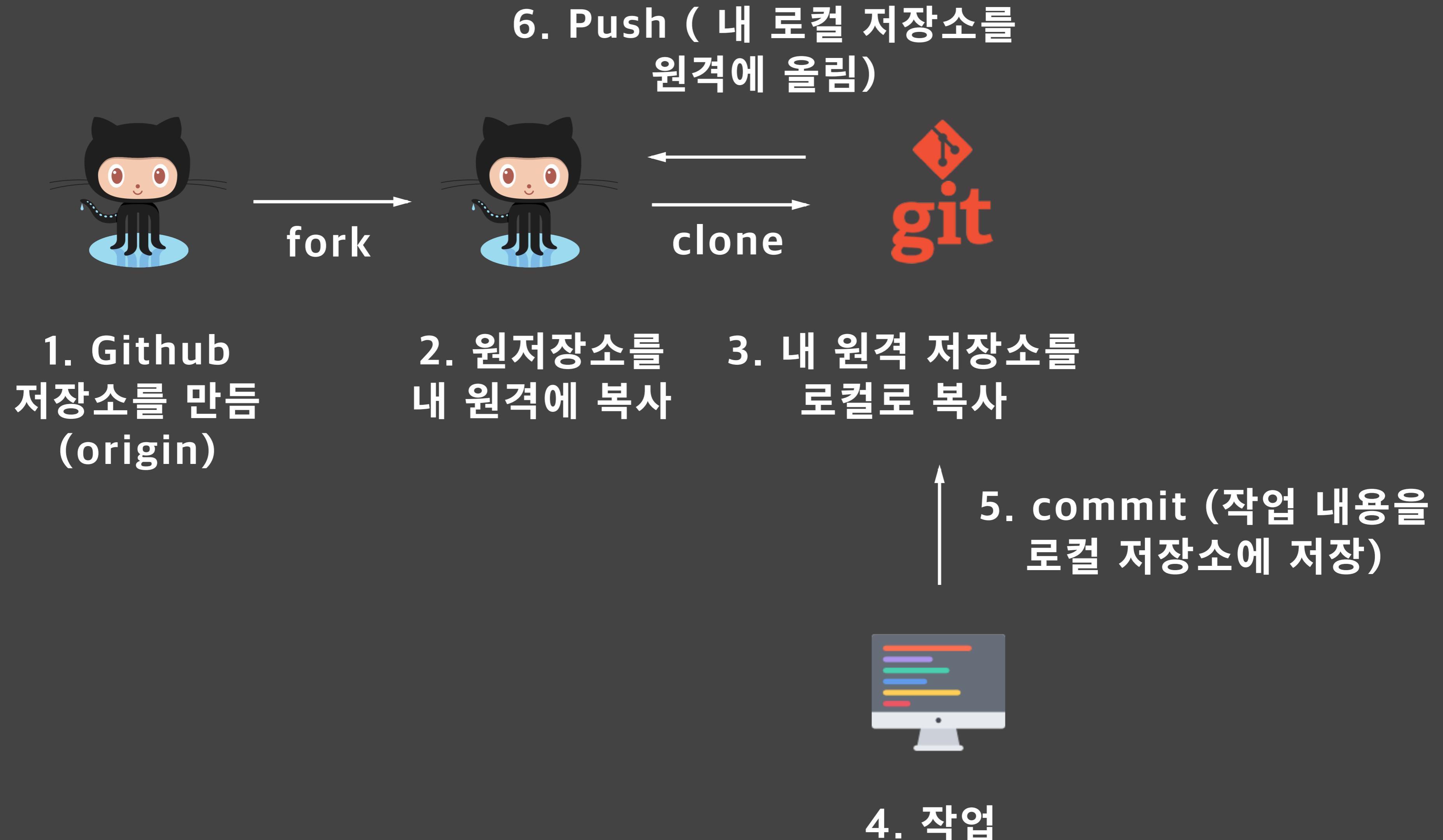
4. 작업

Git



Git





7. Pull Request :

origin 저장소의 유저에게
변경 사항 반영을 요청

6. Push (내 로컬 저장소를
원격에 올림)

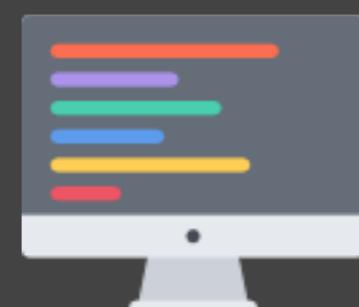


1. Github
저장소를 만듬
(origin)

2. 원저장소를
내 원격에 복사

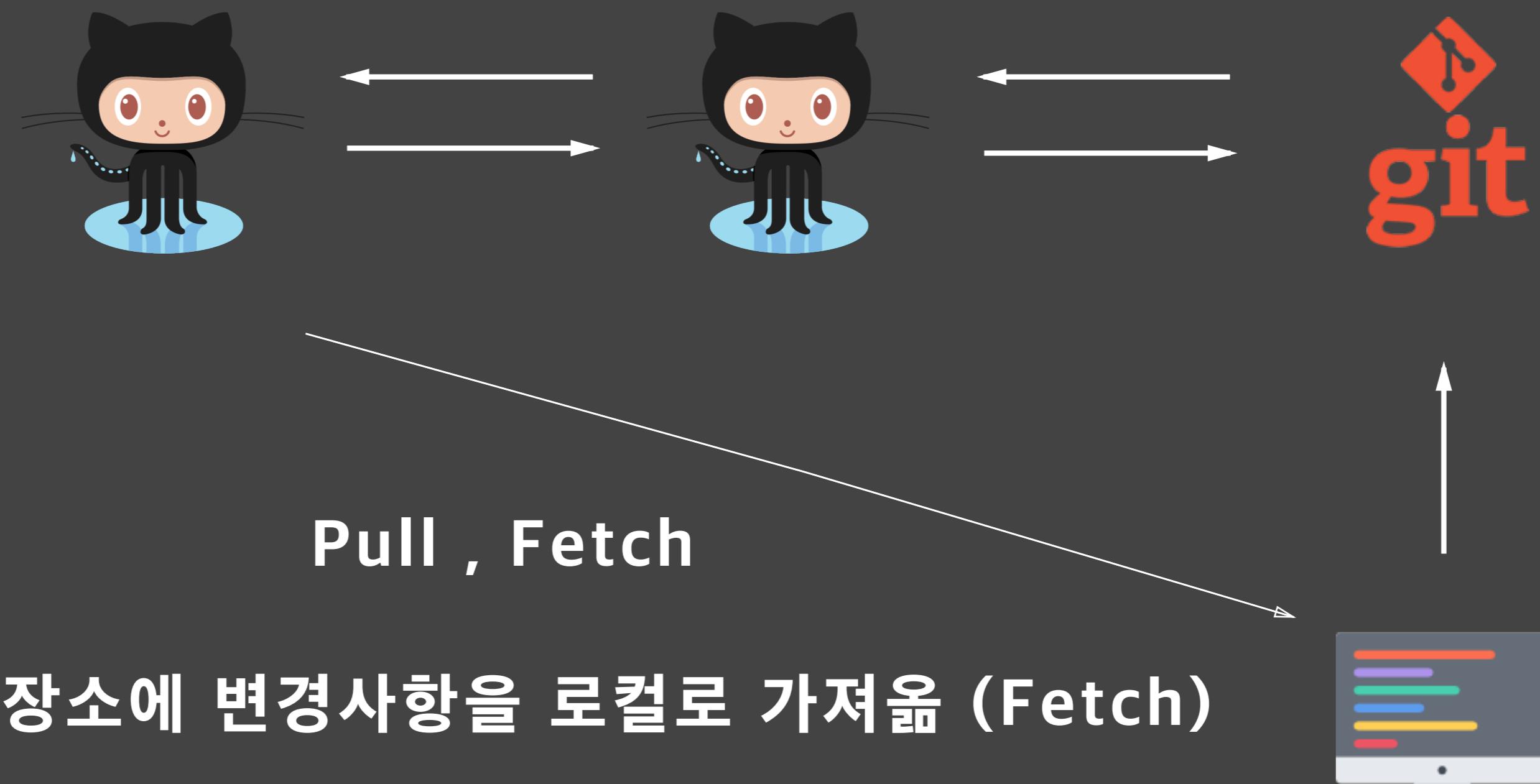
3. 내 원격 저장소를
로컬로 복사

5. commit (작업 내용을
로컬 저장소에 저장)



4. 작업

만약 원 저장소에
다른 사람이 Pull Request 하면!



원격 저장소에 변경사항을 로컬로 가져옴 (Fetch)

가져와서 합치기 까지함 (Pull = Fetch + Merge)

Git

.gitignore 파일

올리지 않을 파일들을 지정함
(충돌방지)

Git

.gitignore 파일 팁

```
/*
!/src/index.html
```

Git

앱잼 팁..?

폴더링을 해두면 정말 편함

```
/*
!/src/ui/main/*
```

Git

앱잼 팀..?

처음 앱잼 팀이 결성되면,

가장 잘하는 사람 한명을 선출 (Main Dev)

Main Dev가 저장소를 만들고

Sub Dev는 Fork 함

Git

앱잼 팁..?

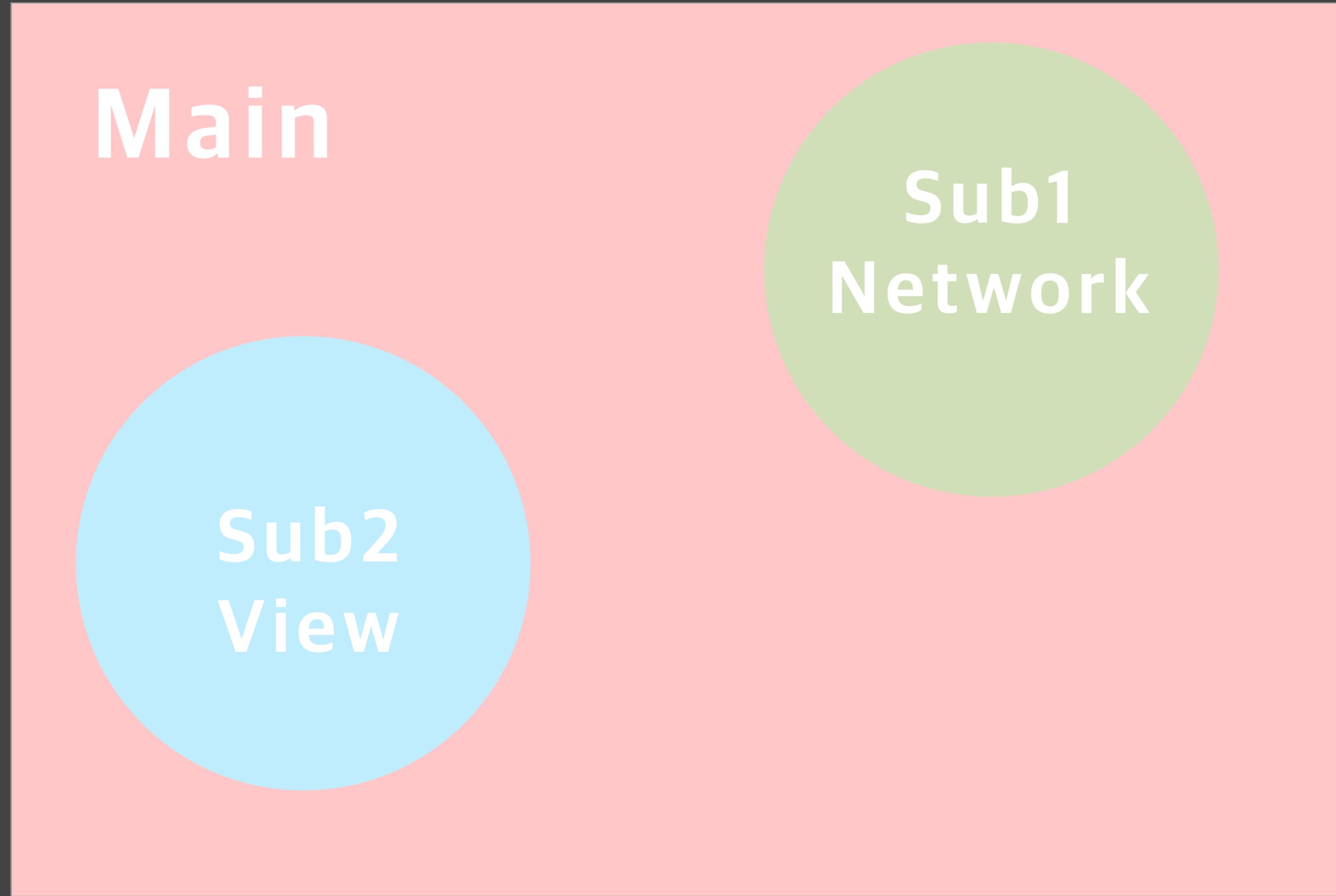
Main Dev는 .gitignore 파일을 작성할때

무시할 파일을 명시하고

Sub Dev는

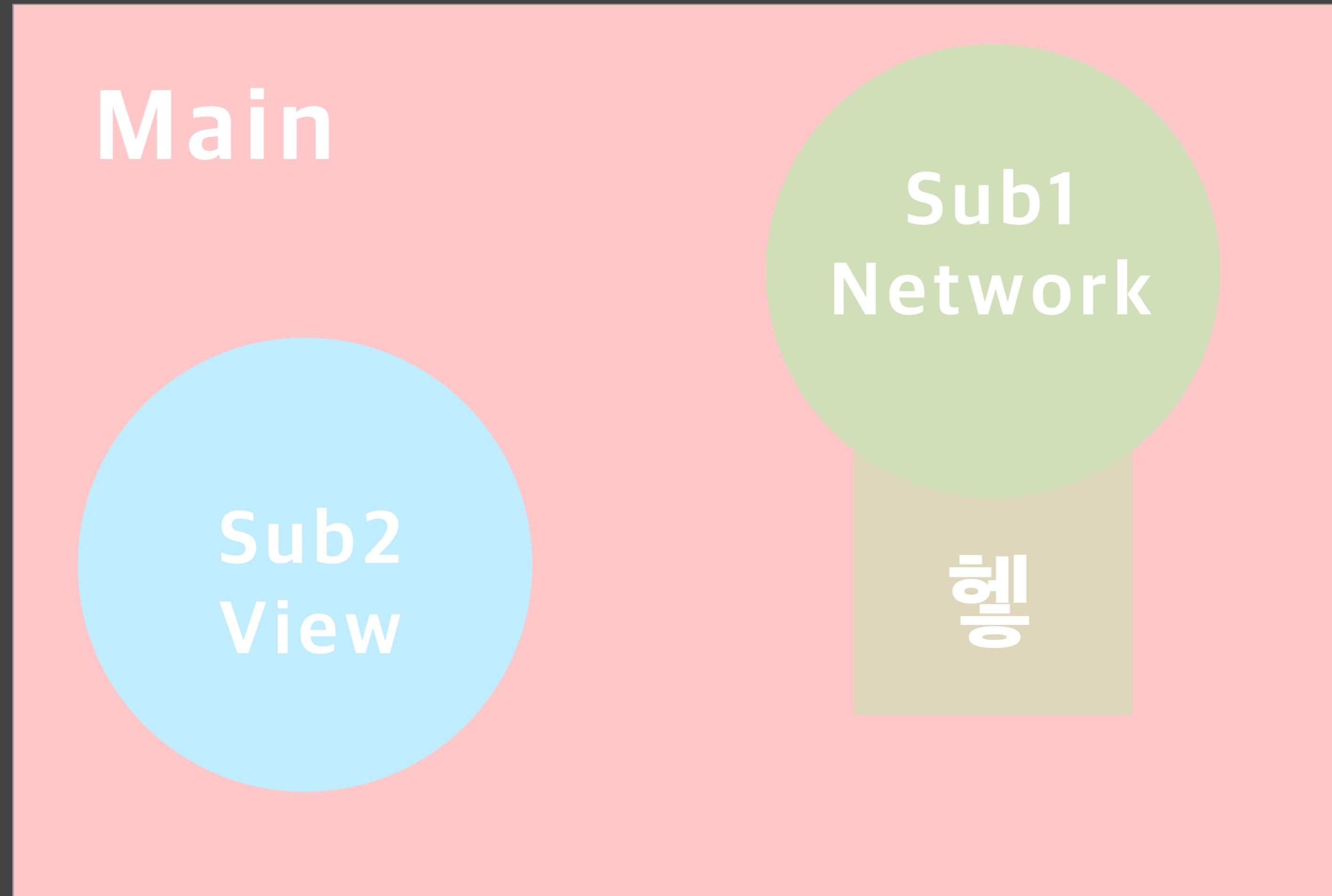
올릴 파일을 명시함

이렇게하면 서로의 영역을 침범하지 않아
충돌이 일어나지 않음



Git

맘대로 Pull Request 했다가 충돌이 일어나면?



Git

현실에서 충돌남 ㅋ



팝콘 들고 구경 가즈아아아아아아아!!

충돌이 안나게 Main Dev 말을 잘따르시길 바랍니다



(꼭 팀마다
git 잘다루는 갓개발자가 있길 기도..)

Git

앱잼 다들 화이팅하고
기말고사 얼마 안남았는데
잘 보시길 바랍니다!

Git

끝