

# Git 심화3-2 (fork 사용한 협업)

📎 자료	<u>DB</u>
☰ 구분	DB
⋮ 과목	

## Fork란?

다른 사람의 GitHub 저장소를 내 GitHub 계정으로 복사해오는 것을 말한다.

- 원본 프로젝트는 그대로 두고
- 내 계정에서 독립적인 복사본을 만들어서
- 마음껏 수정, 실험, 개발할 수 있도록 만든 것이 Fork이다.

즉, "나도 이 프로젝트로 놀아보고 싶은데, 내 버전으로 가져가서 작업해볼게!" 라는 의미다.

이러한 Fork는 오픈소스 프로젝트에 기여할 때 가장 많이 사용하고

혼자 실습용으로 남의 코드를 수정해보고 싶을 때 사용한다.

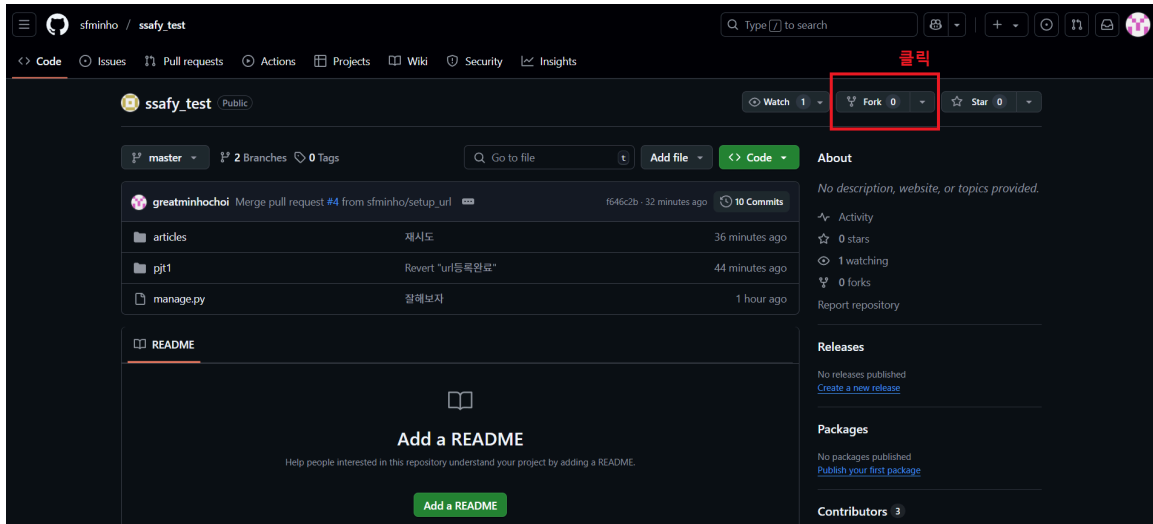
참고로 Fork는 오픈소스처럼 권한 없는 사람 여러명이 공동으로 작업하는 경우 유효하겠지만

우리처럼 팀원이 한 프로젝트에 2~3인팀이 사용할 필요는 없다.

팀원이 적은 경우 브랜치 전략이 Fork를 이용한 협업보다 업무 효율이 높을 것이다.

Fork의 작업 흐름을 살펴보자.

1. 소유권이 없는 원격저장소를 Fork를 통해 복제한다.



그러면 내 깃허브 계정에 원본 원격저장소를 복제한 (Fork한) 저장소가 자동으로 생성된다.

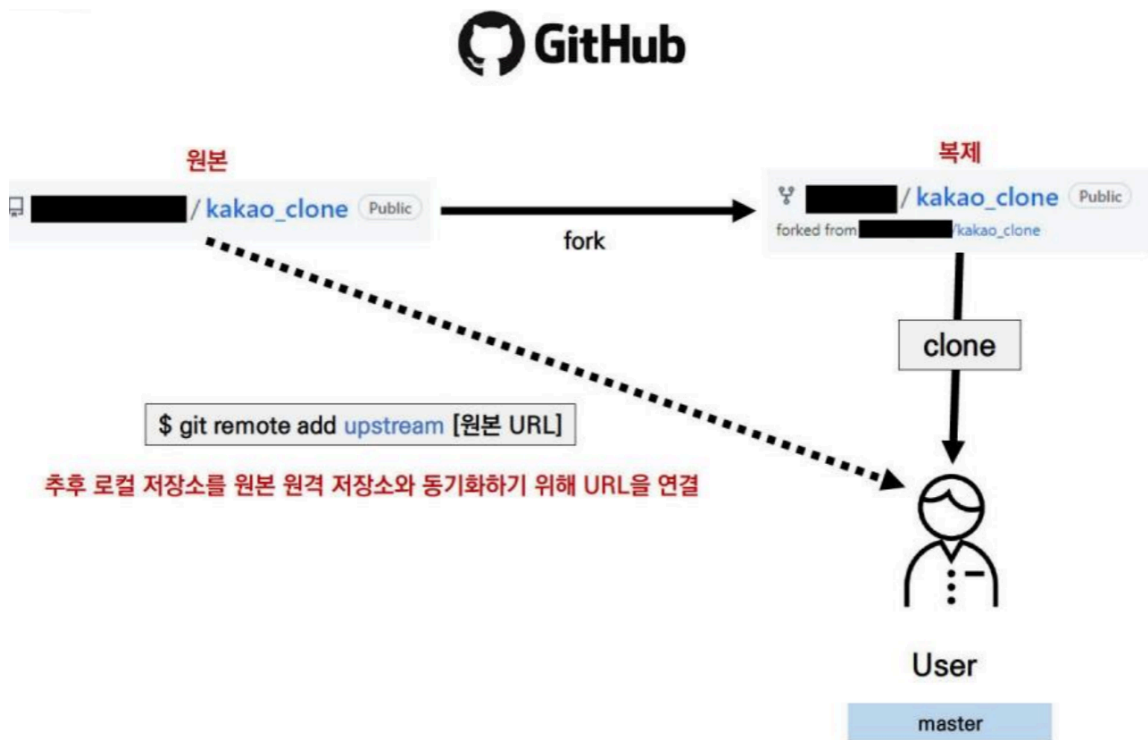
(원격저장소 소유자는 클릭이 불가함. 따라서 팀장은 그냥 브랜치 만들어서 작업한다)

## 2. Fork한 저장소를 로컬에 clone한다.



## 3. 원본 원격저장소와 Master branch를 연결한다.

그리고 원본 원격저장소의 버전 업데이트를 적용시킬 때 fetch 또는 pull을 통해 가져온다.



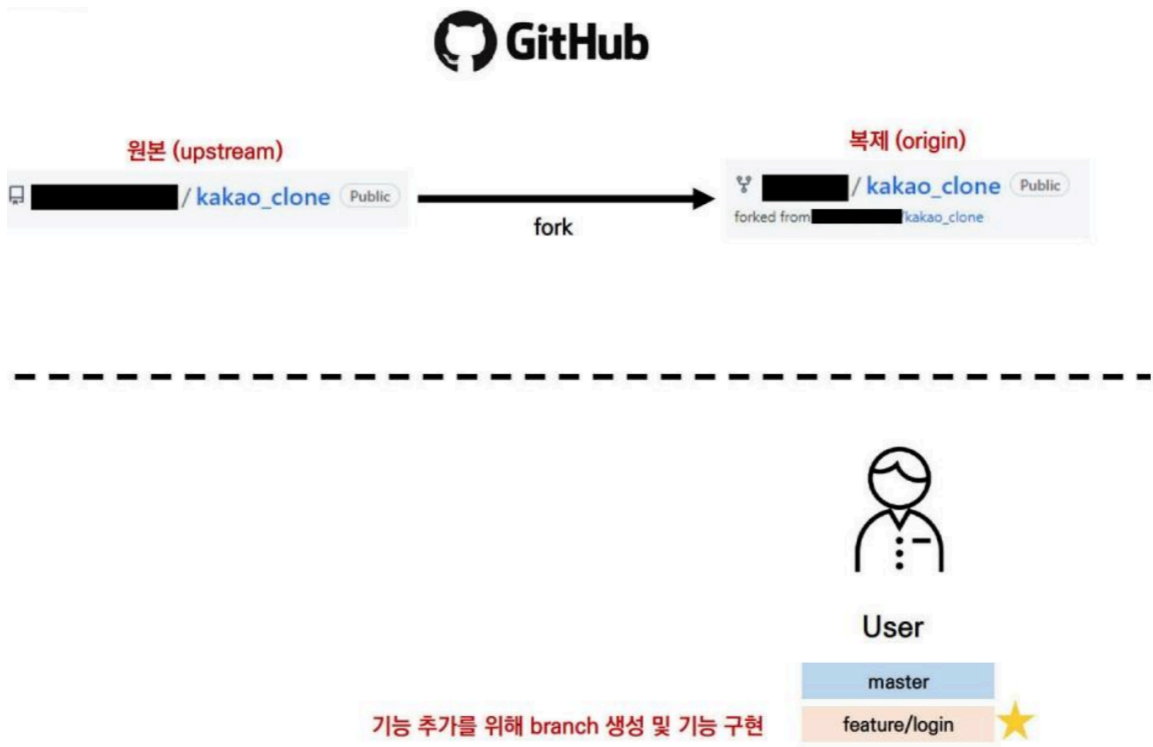
```
# 원본 연결 (처음 1회)
git remote add upstream [원본 원격저장소URL].git

# 최신 원본 코드 가져오기
git fetch upstream or git pull upstream master

# 내 로컬 master에 병합 (fetch 했을경우 시행)
git checkout master
git merge upstream/master # 또는 rebase

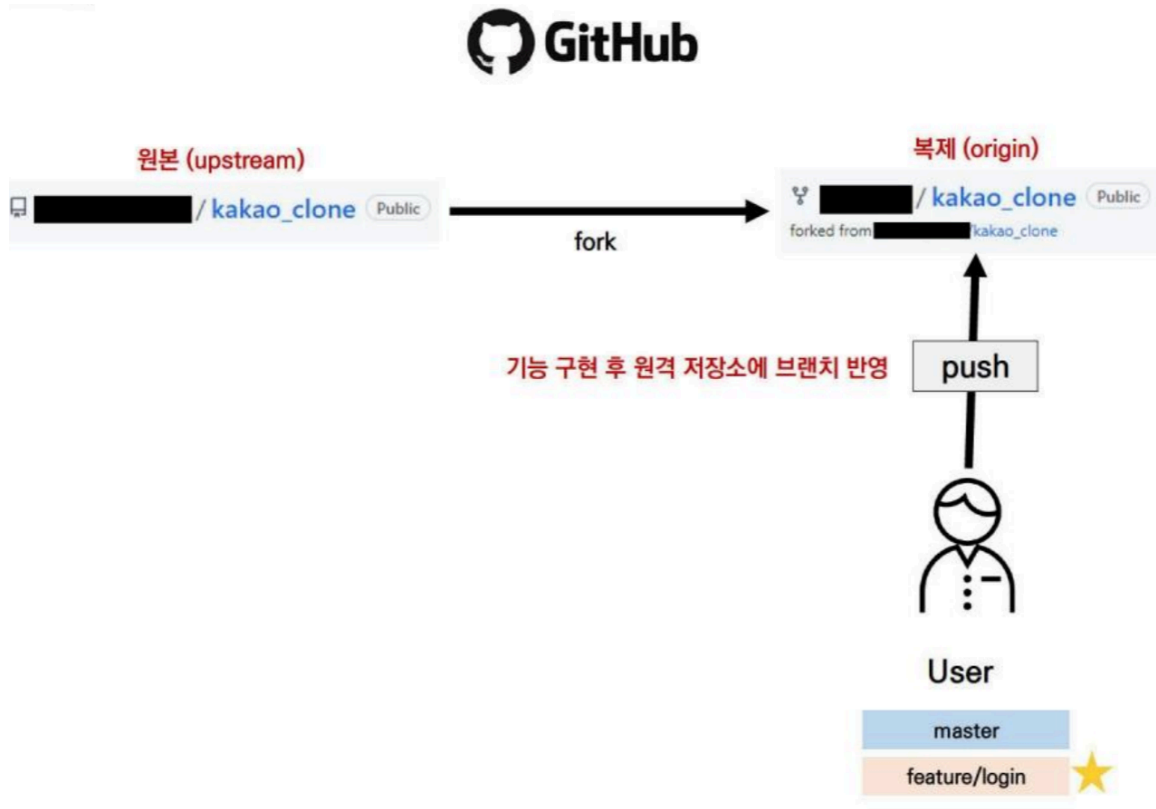
# 내 GitHub(Fork)에 반영
git push origin master
```

4. 브랜치 생성 후 내 마음대로 작업한다.



5. Branch 전략과 마찬가지로 기능구현을 모두 마쳤으면 원격저장소에 반영한다.

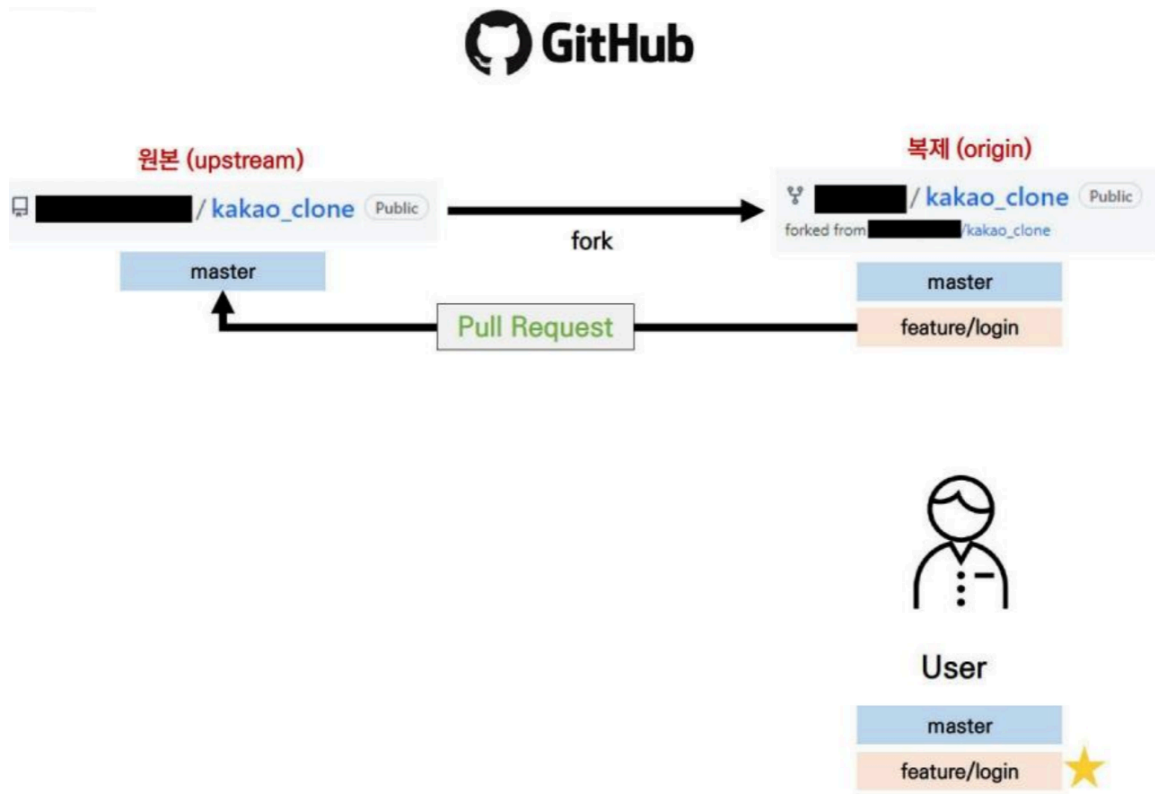
```
git push origin [브랜치이름]
```



6. Fork한 원격저장소로 PUSH가 완료 되었다면

원본 원격저장소에 깃허브를 통해서 PR을 보내자.

아까와 마찬가지로 "Compare & pull request" 버튼을 통해 PR을 생성하면 된다.



7. 병합(Merge)가 완료 되었다면 앞에서 브랜치 전략과 동일하다.

- master 브랜치로 switch해서 원본 원격저장소를 pull 한번 하고
- 다 사용한 브랜치는 삭제한다.

8. 지금까지와 과정을 반복한다.

끝으로

브랜치 이름을 지을 때 조금 신경을 써 주는 것이 좋겠다.

일반적으로 추가될 기능이나 이슈를 나타낼 때에는 `feature/user-auth`, `bugfix/index.html`, `hotfix/refactor_URL` 와 같이 이름만 봐도 대략적으로 추측이 가능한 브랜치명이 좋을 것 같다.