

[Git-Flow 협업 방법 참고](#)
[게시물 링크](#)

최초 1회 사이클

Pull
Push
Branch
Pull Request
Fork
Clone

Remote 구분
* Upstream
* Origin

Remote
Upstream

master

1 Organization
프로젝트
저장소
생성

Branch

3 develop
브랜치
생성

2

Fork
해오기

develop

8 코드 리뷰 후
Merge

Remote
Origin

master

Pull Request

7

프로젝트의
develop
브랜치로
Pull Request
생성
->
Pull Request
생성 완료 후
Origin에
feature 브랜치
생성 됨

4

Clone

Local로
Clone

feature

Local

master

Branch

feature

6 구현 완료 후
동일한 이름의
새로운 브랜치
생성하면서
Push

9 브랜치 삭제

(8) Pull Request 생성하자마자
바로 Merge 창으로 이동 되니
무의식 중에 Merge 하지 않게 주의

(7) Pull Request 생성하는 브랜치가
프로젝트의 develop 브랜치가 맞는지
반드시 더블 체크

->
default 브랜치를 develop으로 하자

(6) 새로운 브랜치 생성하면서 Push
git push origin [feature 브랜치명]

(6) 습관처럼
git push origin master
를 하지 않도록 주의

(9) Merge 후 브랜치 삭제
git checkout master
git merge [feature 브랜치명]
git branch -d [feature 브랜치명]

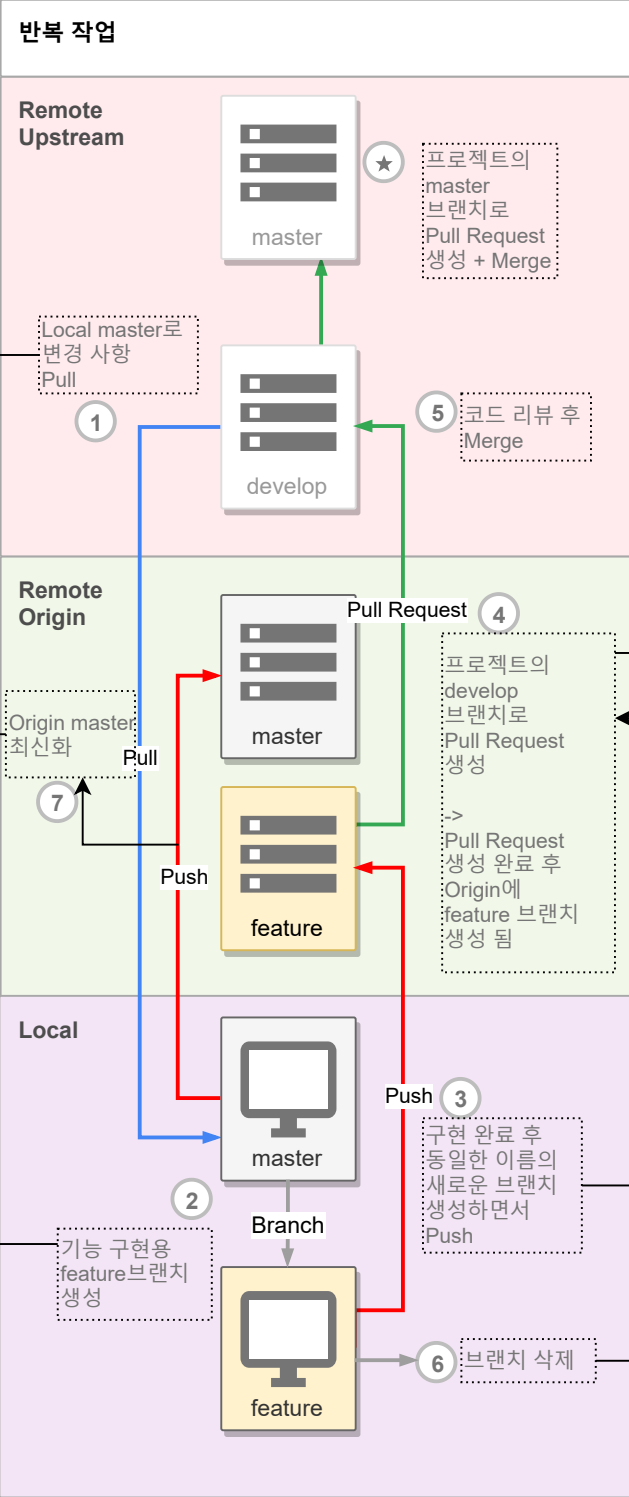
(4) Clone하기
git clone [개인 저장소 주소]

(4.5) 원격 저장소 추가
git remote add origin [개인 저장소 주소]
git remote add upstream [프로젝트 저장소 주소]

(5) 브랜치 생성
git branch [feature 브랜치명]

(5.5) 브랜치 이동
git checkout [feature 브랜치명]

- * 원격 저장소 리스트 확인
git remote
- * 원격 저장소 삭제
git remote remove [저장소이름/ex:origin]
- * 브랜치 리스트 확인
git branch
- * 브랜치 이름 수정
git branch -m [기존 브랜치명] [수정 브랜치명]



(1) upstream develop 브랜치에서 pull하기
git pull upstream develop

(7) Origin에 Push
git push origin master

(2) 브랜치 생성
git branch [feature 브랜치명]

(2.5) 브랜치 이동
git checkout [feature 브랜치명]

(5) Pull Request 생성하자마자
바로 Merge 창으로 이동 되니
무의식 중에 Merge 하지 않게 주의

(4) Pull Request 생성하는 브랜치가
프로젝트의 develop 브랜치가 맞는지
반드시 더블 체크
-> default 브랜치를 develop으로 하자

(3) 새로운 브랜치 생성하면서 Push
git push origin [feature 브랜치명]

(6) 습관처럼
git push origin master
를 하지 않도록 주의

(6) Merge 후 브랜치 삭제
git checkout master
git merge [feature 브랜치명]
git branch -d [feature 브랜치명]