

git 명령어 정리 (진영님것)

git clone

- 생성하거나 fork한 원격 저장소를 내 컴퓨터(local)와 연결하여 데이터를 복사
- fork하지 않은 저장소를 가져올 수는 있지만, 사용에 제한적

git remote

- git remote add origin <https://github.com/>아이디/repository명.git
 - 로컬 저장소와 원격 저장소 연결
- git remote -v
 - 원격 저장소와 연결 확인

git status

- 파일의 상태를 의미함
- 크게 Untracked, Tracked으로 분류됨
- add, commit을 통해 각각 파일의 상태가 Staged, Unmodified가 되므로 add, commit 상황이 아니어도 자주 쓰는걸 권장

git add

- 내 컴퓨터(local)의 데이터를 중간 저장소(Staging Area)로 복제

git commit -m "커밋메세지"

git push

- 원격 저장소에 로컬 저장소의 내용을 올림

git push origin --all

- master이외 브랜치 존재시 로컬의 모든 브랜치를 push

git fetch

- 로컬 저장소와 원격 저장소의 변경 사항이 다를 때 이를 비교 대조하고 git merge 명령어와 함께 최신 데이터를 반영하거나 충돌 문제 해결

git pull

- git remote(연결) 명령을 통해 서로 연결된 원격 저장소의 최신 내용을 로컬 저장소로 가져오면서 병합(push의 반대 성격)
- fetch와 merge를 자동 수행한다
- 협업시 권고되지 않음
- fetch와 pull 구분하기

git merge origin/master

- 로컬 저장소의 master branch에 원격 저장소의 origin/master branch를 병합

git diff

- 로컬 저장소의 branch와 원격 저장소의 branch 사이에 어떤 차이점이 있는지 확인
- '<<<<<HEAD'
- code1
- =====
- code2
- '>>>>>origin/master'
- 충돌 사항은 수동으로 수정하는걸 권장

[중요] 협업시 용어 및 개념

pull request

- 원본 저장소(fork의 출처)의 소유자가 포크한 다른 저장소의 변경 내역을 병합
 1. fork한 사용자는 Create pull request
 2. 원본 저장소에서는 pull request를 승인/거절
- 현업에서는 pull request를 통해 저장소간 병합을 진행하고, 변경사항에 대해 토론을 진행하거나 코드 리뷰를 할 수 있다

fork한 저장소 동기화

1. 주소 확인
 - git remote -v
2. 원본 저장소의 주소가 없을 시 추가
 - git remote add upstream <https://github.com/아이디/repository명.git>
3. 추가된 주소 확인
 - git remote -v
4. 원본 데이터 fetch
 - git fetch upstream
5. local 저장소 branch와 병합
 - git merge upstream/master
6. fork한 저장소에 push
 - git push origin master