

김준범의 git init 주소 : /C/Users/user/Desktop/database/databasegit

아이디 : doodoodi (log는 junbeom으로 바꿈)

email : wnsqja0219@naver.com

pwd : 현재 위치 알려줌

cd : 원하는 곳으로 이동할 때 쓰는 명령어

mkdir : 디렉토리 생성

-----github 기록 남기기-----

git init한 주소로 가서

git add 파일명 으로 추가 or **git add .** (변경된 모든 파일 추가)

git commit -m "주석줄 내용" 으로 커밋 (-m은 주석 줄 내용을 넣기 위한 옵션)

git push -u origin main : 로컬에서 변경사항을 원격 저장소로 보냄(git으로 보냄)

(-u는 맨처음에 원격이랑 로컬 두 브랜치를 연결해주는 역할)

git pull origin main : 원격 저장소에서 로컬로 가져옴(내 드라이브로 가져옴)

-----브랜치 생성, 삭제, 병합, 확인 -----

git branch 브랜치이름 : 새로운 브랜치 생성

git branch : 프로젝트에 존재하는 모든 브랜치 목록 확인(현재 사용되는 브랜치 확인가능)

git branch -r : 원격 브랜치 목록 확인

git checkout 브랜치이름 : 새로운 브랜치로 이동

git checkout -b 브랜치이름 : 브랜치 생성 및 이동

git push origin 브랜치이름 : 브랜치를 원격 저장소에 push (여기까지 해야 github에 브랜치 생김)

(branch의 내용은 로컬에서의 가장 최신 commit 내용을 반영한다.)

(만든 branch 내용을 commit하고 **git checkout main**으로 **main branch**로 이동한 후 **git merge 새브랜치이름**을 하면 그 내용이 병합된다. 그후 **git push -u origin main**을 해서 **github main**에 반영하면 된다.)

git merge new-feature : new-feature 브랜치를 main에 병합

git branch -d new-feature : 브랜치를 삭제

git push origin --delete new-feature : 원격 브랜치 삭제

(로컬이랑 github 브랜치랑은 따로 삭제를 해줘야함)

-----git log사용하기-----

`git log`나 `git reflog`로 내가 commit했던 내용들로 되돌아 갈 수 있다.

예를 들어 1, 2, 3, 4라는 commit 기록이 있을 때, 내가 `git checkout 3(commit-hash)`로 가서 추가 작업한 후 commit을 하면 4가 사라지는 것은 아니지만 4 commit과는 연결되지 않는다. 3과 연결된 새로운 5 commit 내용이 생겨나는 것. 4와 합치려면 4 브랜치로 이동해서 5 브랜치를 merge하면 된다.

1, 2, 3, 4의 commit이 있고, 각각 a, b, c, d라는 내용을 작업했을 때, 3으로 돌아가서 push를 하면 github에서는 4에서 작업한 d의 내용이 사라진다. 그러나 `git checkout 4`로 가서 다시 push를 하면 그 내용을 되돌릴 수 있다.