git 사용법

폴더 우클릭 🡪 Open git bash here 선택 🡪터미널 열림

$git config --global user.name “<닉네임>”

🡪유저 이름 설정

$git config --global user.email “<메일 주소>”

🡪유저 메일 주소 설정

$ls

🡪폴더 내의 가시 파일 정렬

$ls –al

🡪폴더 내의 숨김 폴더까지 함께 보여줌

$git init

🡪깃 저장소 초기화

$touch <파일명>.<파일 형식>

🡪해당 레포지토리에 파일 생성

$git status

🡪폴더 내 파일들의 현재 상태를 확인 가능

$git add <파일명>.<파일 형식>

🡪깃이 관리하는 가상 영역에 해당 파일을 전달

(커밋 전 단계, Untracked file에서 벗어난 상태)

$git add .

🡪현재 워킹 디렉토리에 있는 모든 untracked file을 전부 올린다.

$git commit

🡪현재 add된 파일들을 전부 커밋함

shift+i를 통해 해당 커밋에 대한 코멘트 작성 가능(insert 표시가 나와 있을 때)

esc를 통해 insert 모드 나옴

esc+shift+: 🡪 wq 작성 🡪 enter

커밋 후 해당 파일에 변경 사항이 있다면, Untracked file로 돌아가게 됨

(add, commit 과정 반복)

$git commit -m “<커밋 메시지>”

🡪위에서 insert모드에 관한 내용 전부 생략하고 바로 커밋 메시지 작성

$git log

🡪현재까지 commit 이력을 보여줌

$git .gitignore

🡪git에 감지되지 않는(나만 볼 수 있는) 파일 생성

gitignore파일을 텍스트 파일로 연 후에 숨기고 싶은 파일을 작성한다면, 해당 파일이 git에 반영되지 않는다.

$nano .gitignore

🡪바뀐 인터페이스에서 숨기고 싶은 파일 작성 후 ^x 🡪절차를 따라가면 .gitignore 파일 내에 숨기고 싶은 파일이 작성되어 같은 결과를 얻게 된다.

해당 작업 후에 git status를 해 보면 숨기고 싶은 파일을 사라지지만, .gitignore 파일이 untracked file로 나온다. .gitignore 파일을 commit 하면 해결된다.

$git log -p

🡪로그의 변경 사항까지 함께 출력됨 (exit하려면 q)

$git log -<int>

🡪숫자만큼의 최신 로그를 보여줌

$git log --oneline

🡪각 로그를 한 줄씩만 간단하게 보여줌

$git checkout <해시 앞7자리>

🡪해당 로그의 래포지토리 상태로 롤백함

해시 대신 master을 작성하면 master로 돌아감

$git reflog

🡪헤드의 이력(로그) 확인 가능

$git reset <파일명>

🡪해당 파일을 git의 가상 공간에서 untracked로 옮김

$git reset <해시 7자리> --<soft> or <mixed> or <hard>

🡪해당 해시의 상태로 reset

🡪reset까지만 작성하면 mixed

🡪soft: 리포지토리만 바꿈

🡪mixed: working directory만 냅둠

🡪hard: 리포지토리, staging area, working directory 전부 변경

$git revert <해시 7자리>

🡪git reset과 비슷하지만, 커밋 수정 사항을 되돌렸다는 내용까지 log에 이력으로 남기는 명령어이다. 또한, 해시 7자리로 돌아가는게 아니라 해당 해시 7자리의 내용을 지우는 것이다.

#발생 상황 및 대응 방법

1. 스테이징 실수 되돌리기

git reset <파일명>

2. 커밋 실수를 되돌리기

git reset <해시 7자리> --option

3. 커밋 실수 되돌리되 이를 기억하기

git revert <해시 7자리>

깃 프로젝트가 처음 만들어지면 master라는 이름의 기본 브랜치 하나가 만들어진다.

이를 바탕으로 여러 브랜치를 분기해가며 작업 가능

$git branch

🡪현재 가지고 있는 브랜치 목록을 보여줌 (현재 브랜치에 \*이 붙음)

$git branch <새로운 브랜치 이름>

🡪새로운 브랜치 생성

$git checkout ‘<브랜치 이름>’

🡪해당 브랜치로 이동

$git merge <브랜치 이름>

🡪현재 브랜치와 해당 브랜치를 병합한다.

$git branch -d <브랜치 이름>

🡪해당 브랜치를 삭제한다.

깃허브

깃 저장소 호스팅을 지원하는 웹 서비스

온라인 상에 저장소를 만들어 원격으로 이를 관리

내 PC안에서(git)에서 작업할 경우에는 로컬,

깃 프로젝트를 공유할 수 있는 깃허브를 사용할 경우에는

$git remote add <저장소 이름> <원격 저장소 주소>

🡪해당 원격 저장소에 현재 저장소를 저장소 이름으로 하여 연결한다.

$git remote -v

🡪현재 연결되어 있는 원격 저장소를 출력

fetch: 원격 저장소에 있는 내용을 내려받을 때에 사용하는 주소

puch: 원격 저장소에 로컬의 내용을 올릴 때에 사용하는 주소

$git push -u <저장소 이름> <브랜치 이름>

🡪해당 저장소의 해당 브랜치를 github에 push 한다.

$git clone <원격 저장소 주소>

🡪해당 주소의 github 내용을 가져옴(merge)

$cd <option>

🡪option에 ..을 입력하면 상위 폴더로, 폴더명을 입력하면 해당 폴더로 이동한다.

$git pull

🡪연결된 깃헙의 내용을 현재 branch에 merge

$