명령어 정리

리눅스 명령어

pwd : 현재 폴더 위치 확인

ls -a : 현재 폴더 파일의 목록 (-a 옵션으로 숨겨진 파일도 확인)

cd : 폴더 이동 기본은 홈 폴더로 이동 .. – 상위 폴더 폴더이름 – 해당 폴더

mkdir : 폴더 생성

echo : 화면에 글자 표현

깃 명령어

git init : 깃 초기화(깃 생성) .git파일 생성

git status : 깃 상태 확인

옵션

-s : 짧게 요약해서 상태를 보여준다

git hash-object 파일명 : 일반 파일의 체크섬을 확인 내용이 같다면 체크섬도 같음

git show 체크섬 : 해당 체크섬을 가진 객체의 내용을 표시함

git ls-files --stage : 스테이지 파일의 내용을 표시

git cat-file -t 체크섬 : 해당 파일의 타입을 알려줌

git cat-file 객체타입 체크섬 : 파일 내용 확인

git의 4대 오브젝트

blob : 바이너리 데이터 자체 저장

tree : 디렉토리 내용을 저장

commit : working tree의 snapshost

tag

git clone 원격장소URL 폴더이름 : 원격저장소 클론

git add 파일명 : 파일 추적(등록)

옵션

--patch 파일 내용 분리 해서 수정 후 올릴 수 있음

git commit : 현재 상태 커밋

옵션

-a : 등록과 커밋을 동시에 (이미 등록이력이 있는 파일만)

-m : 커밋 메시지 작성

--amend : 커밋 내용 수정(이전 커밋 메시지 수정 해쉬 값 바뀜)

git log : 깃 로그 확인

옵션

--all : 모든 로그 출력

--graph : 로그 그래프로 보여줌

--oneline : 로그 간단하게 출력

-- decorate : 브랜치와 태그등의 참조를 간결하게 표현

git diff : 커밋 비교 working tree와 stage의 차이(추적 안된 파일 안됨)

옵션

--check 공백문자 확인 (git checkout -- 파일명 -> 수정된 내용 돌리기)

--staged : commit stage 비교

git remote : 원격 저장소 별칭 확인

옵션

-v : 별칭과 주소

add origin : 원격저장소별칭 원격저장소URL 원격저장소와 연결

rm : 원격저장소별칭 등록한 원격저장소 삭제

git clone 저장소주소 폴더명 : 원격저장소 복사

git pull : 커밋 가져오기 (git fetch + git merge)

git fetch : 커밋 가져오기

git push 원격저장소별칭 브랜치이름 : 커밋 전송하기 -u 옵션으로 다른 브랜치도 같이 올라감

옵션

--all : 모든 브랜치

--tags : 모든 태그

git clean -f -d : 추적 되지 않은 파일을 삭제 (-f 파일 -d 디렉토리)

git branch : 브랜치 확인, 뒤에 브랜치이름 입력시 새로운 브랜치 생성

옵션

-v : 마지막 커밋도 같이 보여줌

-f : HEAD로부터 브랜치를 생성(이동) 체크섬 추가시 해당 체크섬부터

-r : 원격 저장소의 브랜치 확인

-b : 브랜치 삭제

-D : 병합하지 않는 브랜치를 강제 삭제

git checkout 브랜치이름 :

옵션

-b : 브랜치 생성과 이동

커밋 체크섬 : HEAD와 브랜치가 분리

git stash : 스태시 저장

옵션

-u : 추적 안된 파일도 저장

pop : 스태시 읽고 삭제

apply : 스태기 읽기 삭제 X

list : 변경사항 목록 확인

drop : 지정 스태시 삭제

clear : 모든 스태시 삭제

branch : 새로운 브랜치를 만들어서 스태시 저장

git cherry-pick : 특정 커밋 가져오기

git merge : 브랜치 병합

병합시에 파일의 충돌이 없다면 fast-forward 병합

git rebase : 리베이스 병합

--continue로 충돌 해결 후 진행

git reset 옵션 커밋아이디 : 리셋

옵션

파일명(스테이지에 올라가있는) : 스태이지 내리기 (rm으로도 가능)

-hard 브랜치를 지정한 커밋으로 옮김

HEAD~숫자 : 이전 커밋을 가르킴

HEAD^숫자 : 해당 번째 부모를 가르킴

git revert 커밋아이디 : 리버트 취소 커밋

git tag : 태그 목록

옵션

태그이름 : 태그 추가

-a : 주석

-m : 메시지

git push 원격저장소별칭 태그이름 : 태그 전송

git reflog : 참조기록(몇일간 저장)

git clean 옵션 : 추적되지 않은 파일 삭제

git config : 옵션 설정

옵션

--global : 전역 옵션

--local : 지역 옵션

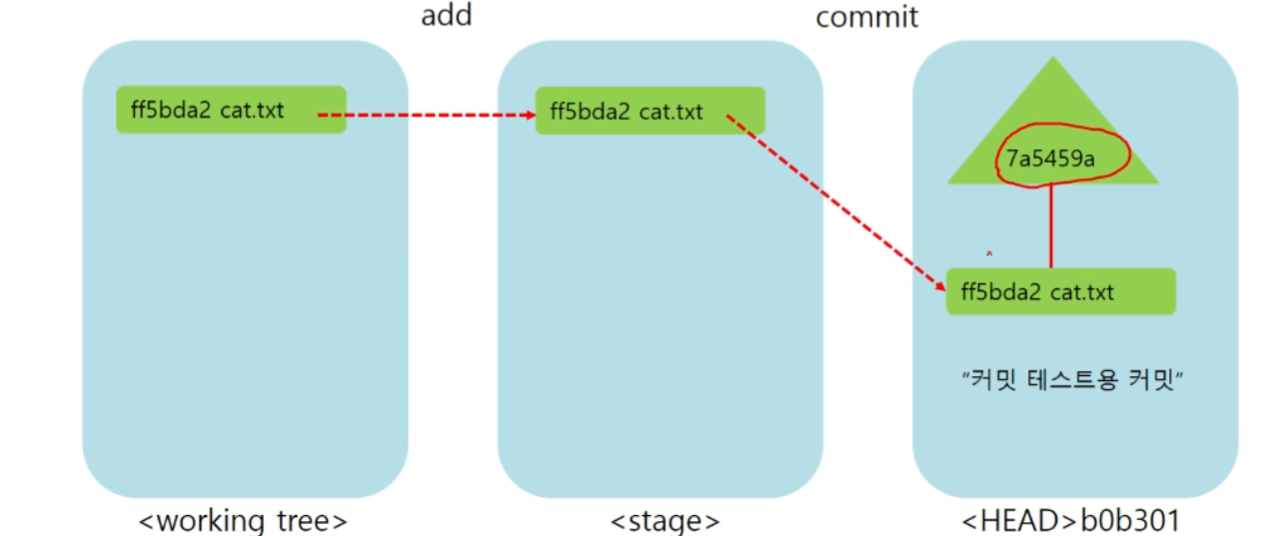
--system : 시스템 옵션

옵션 --unset : 옵션 삭제

--list : 모든 옵션 리스트 출력

기본 에디터 확인

git config –global core.editor (“core --wait” vscode로 설정)



수동 커밋

git write-tree : 트리 생성 (tree 해쉬 생성)

git ls-tree 해쉬값 : 스테이지에 있는 blob객체와 같은 지 확인

$echo “ 커밋 메시지 ” | git commit-tree 해쉬값 -p HEAD : 커밋 생성 (-p HEAD) -> 헤드 지정 (commit 해쉬 생성)

수동 커밋은 헤드에대한 갱신이 안되서 해줘야 한다

git update-ref refs/heads/master 해쉬로 갱신

.git/refs/heads -> 브랜치 저장 장소

.git/HEAD -> 현재 헤드 위치

테이블이(가) 표시된 사진

자동 생성된 설명

git rev-parse : 포인터의 해쉬값 반환