

# 2주차

📅 날짜 @2025년 9월 8일

## #01~02

커밋(Commit): 적어두다

버전관리의 커밋: 저장소의 현 상태를 저장하는 행위  
파일 집합의 변경 내용을 깃 저장소에 기록하는 작업

- 커밋

: 저장소의 현재 상태를 저장

- 저장소

: 연속된 커밋으로 관리, 파일이 달라지지 않았으면 파일을 새로 저장하지 않음

- 원격 저장소(Remote Repository)

: 파일이 원격 저장소 전용 서버에서 관리, 여러 사람이 함께 공유하기 위한 저장소

- 지역 저장소(Local Repository)

: 내 PC에 파일이 저장되는 개인 전용 저장소

- 커밋: 저장소의 현 상태를 저장하는 행위이자 깃 명령어

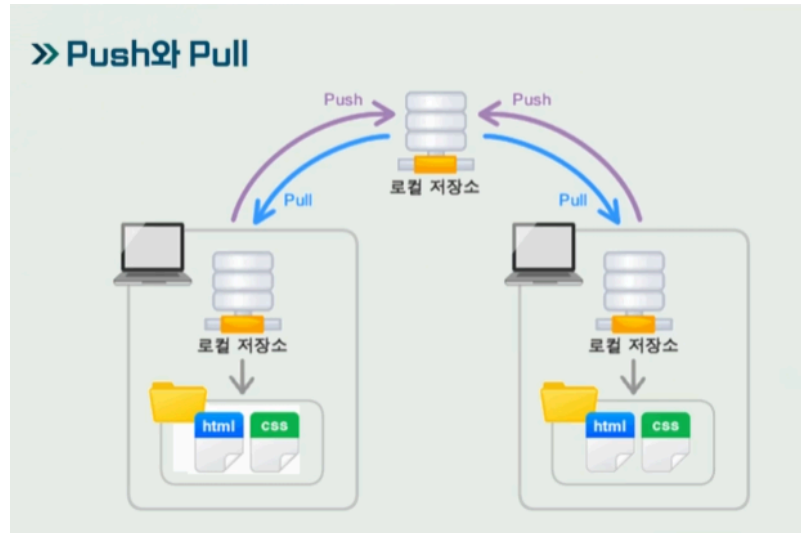
- HEAD: 가장 최근의 커밋을 가리키는 포인터

- 저장소(git repository): 파일이나 폴더, 버전 관리를 위한 관련 파일을 저장해 두는 곳

- PUSH: 지역에서 원격 저장소로 자료를 올림

- PULL: 원격 저장소에서 지역 저장소로 자료를 내림

- CLONE: 서버의 원격 저장소를 지역 저장소에 복사



버전 관리: 시간의 흐름에 따라 파일 집합에 대한 변경 사항을 추적, 관리

필요성: 과거 지점의 버전으로 돌아가 누가, 무엇을 수정했는지 파악 가능

버전 관리 도구의 인터페이스 방식: CLI와 GUI 존재

CLI: 깃

GUI: 소스 트리

## #03

깃GIT

2005년, 리누스 토발즈가 개발

주니어 하마노가 소프트웨어를 유지보수중

깃 정의

: 컴퓨터 파일의 변경을 추적하는 데 사용되는 버전 관리 시스템

기능

- 여러 개발자가 함께 작업
- 소스 코드의 변경 사항을 추적하는 데 사용
- 소스 코드 관리에 분산 버전 제어 도구가 사용
- 여러 개의 병행 분기를 통해 비선형 개발을 지원

깃 사용의 장점

: 소스 코드에 대한 모든 변경 사항을 다른 사용자가 추적 가능

git bash: 명령 행 인터페이스

git gui(graphic user interface)

깃의 구조(4개)

- 작업 디렉토리, 작업 공간, 작업 트리
- 스테이징 영역
- 깃 저장소
- 임시 저장소(stash)

깃에서 상태 간의 이동 명령

work space → (add) → Index(stage) → (commit) → local repository

↑↓

pull    push

↑↓

local repository