# 01~03

| 상태 | 진행 중 |
| --- | --- |

< 내용 정리 >

## 버전관리의 정의

시간 흐름에 따라 파일 집합에 대한 변경 사항을 추적, 관리

- 누가 __ 어떤 저장소에서 __ 어떤 파일을 __ 언제 __ 어디를 __ 어떻게(추가/수정/내용)

## 버전관리의 필요성

과거 지점의 버전으로 돌아가 누가,무엇을 수정했는지 파악 가능

## 버전관리 도구의 인터페이스방식

### CLI(ex.Git Bash)

- Command  Line  Interface
- 명령 행 인터페이스
- Mac의 Terminal, Windows의 CMD나 Powershell

### GUI(ex.소스트리, Git GUI)

- Graphical  User  Interface
- 대부분 일부의 기능만 구현 → 비교적 단순

## 커밋(commit)

저장소의 현 상태를 저장하는 행위이자 깃 명령어

- 파일 집합의 변경 내용을 깃 저장소에 기록하는 작업
- 어느 시점의 추가/변경 사항 → 이전 커밋 상태~현재 커밋 상태 의 커밋 생성

## HEAD

가장 최근의 커밋을 가리키는 포인터

### 저장소(Git repository)

파일이나 폴더, 버전관리를 위한 관련 파일을 저장해 두는 곳

- 원격 저장소 : 원격 저장소 전용 서버에서 관리, 여러 사람이 함께 공유하기 위한 저장소
- 지역 저장소 : 내PC에 파일이 저장되는 개인 전용 저장소

### 원격 저장소와 지역 저장소의 명령

- clone : 원격저장소 → 지역저장소 [*복사*]
- push : 지역저장소 → 원격저장소 [*올리기*] (작업 내용을 공개하고 싶을 때)
- pull : 원격저장소 → 지역저장소 [*내리기*] (여러 사람이 작업한 파일을 가져올 때)

### Git

파일 및 프로젝트의 시간 경과에 따른 변경 사항을 추적할 수 있는 버전 관리 시스템

### Github

버전관리를 위한 서버 저장소 및 프로젝트 개발을 위한 협업 관리 서비스

개발자들이 코드를 더 쉽게 공유할 수 있도록 온라인으로 Git저장소를 호스팅하는 웹사이트

### 깃 저장소 구조

- 작업 디렉토리(Working derectory) - 작업 폴더(Working folder), 작업 트리(Working tree), 작업 공간(Work space) — modified,untracked
- 스테이징 영역(Staging area) — stages,indexed
- 깃 저장소(Git Repository) — committed
- 임시 저장소 — Stash

### Add 명령

작업 디렉토리 → 스테이징 영역

## Commit 명령

스테이징 영역 → 깃 저장소

## 깃 + 깃허브

- Open Source Software || Development
- 깃과 깃허브를 통해 오픈소스 소프트웨어를 개발

## 버전관리 시스템의 종류

CVS, SVN, GIT, Mercurial, Bazarr