

버전관리와 깃과 깃허브 개요

Version Control and
Git & Github basics

강환수 교수








AI Experts
Who Lead
The Future

01

깃, 깃허브 버전관리 이해



- 이름으로 관리의 문제
 - 이름을 수정하면서 관리
 - 관리할 파일이 많다면
 - 여러 이름의 새로운 폴더에 여러 파일을 복사해 관리

Name	
	120525_문서_업데이트.txt
	120604_문서.txt
	120605_문서_수정판.txt
	120605_문서_수정판2.txt
	120605_문서_최신 복사.txt
	120605_문서_최신.txt
	120605_문서.txt
	1200602_문서.txt
	문서_회의용.txt

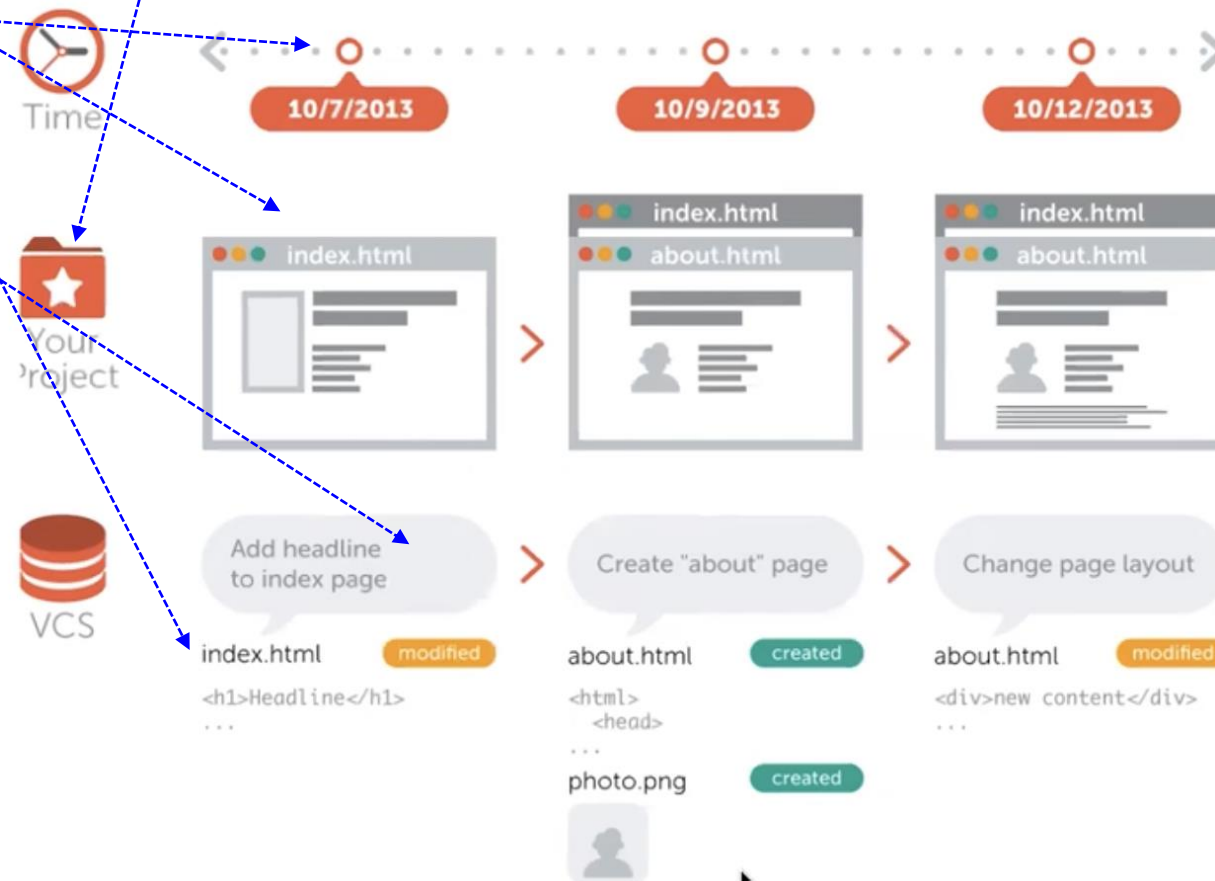


- 파일 버전 관리
 - 파일의 수정 이력 관리
 - 언제, 무엇을

• 파일의 추가 및 수정 이력(추적) 관리

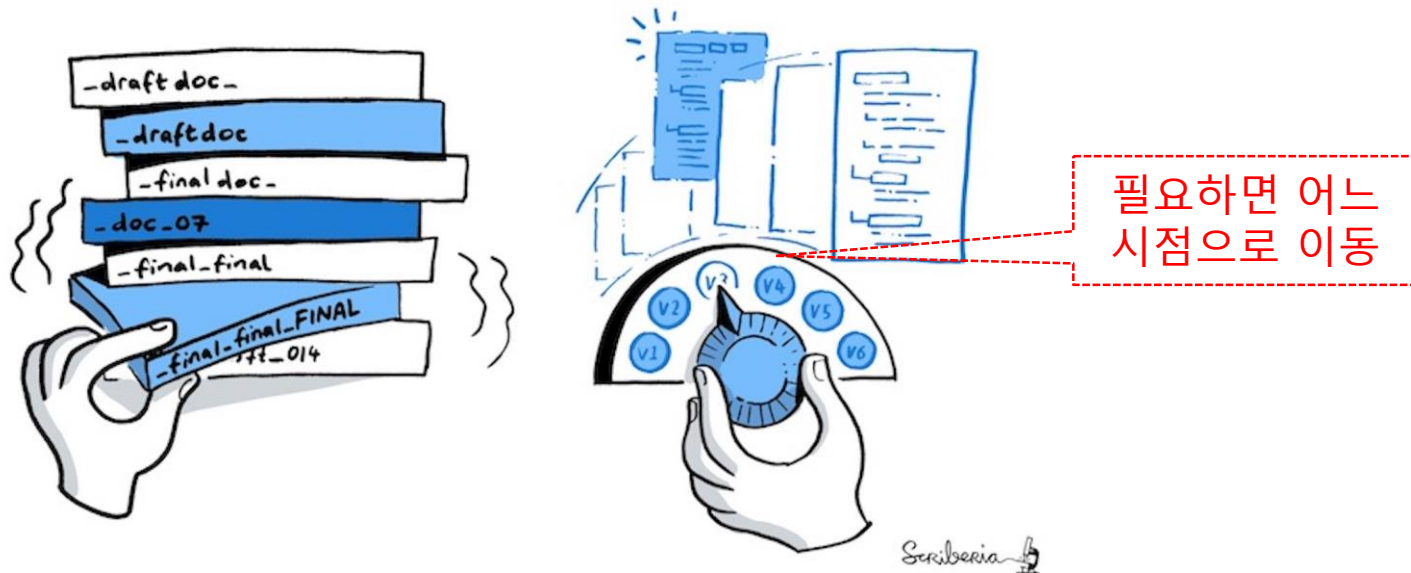
- 누가(다중 사용자)
- 저장소가 여러 개인 경우(어느 저장소에서)
- 어느 파일을
- 언제
- 어디를
- 어떻게

- 추가
- 수정
- 내용

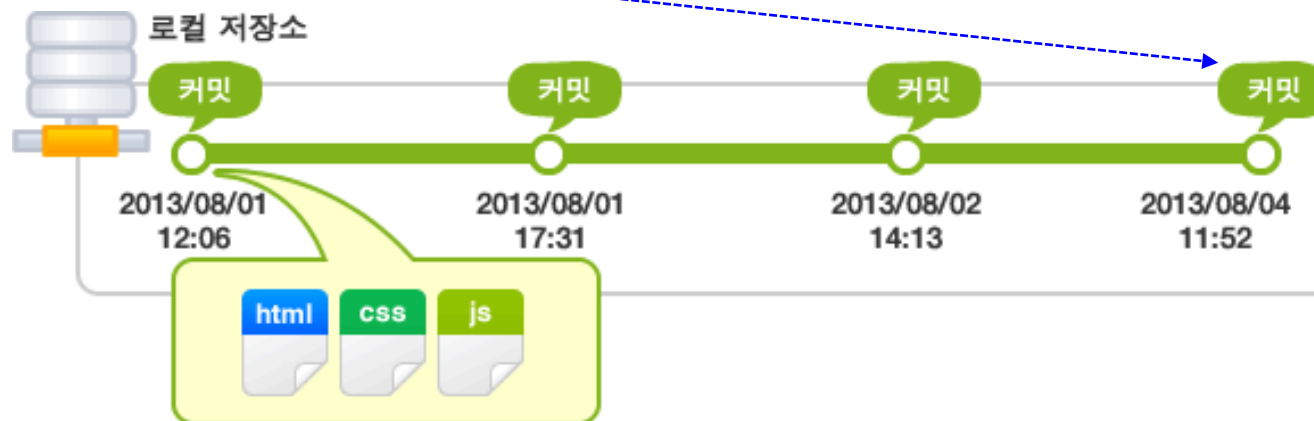


- 시간 흐름에 따라 파일 집합에 대한 변경 사항을 추적, 관리
 - the version control system tracks changes to a file or set of files over time
- 버전의 저장과 백업
 - 필요하면 이전의 버전으로 되돌림
 - 변경사항의 자세한 확인
- 여러 사용자에게 버전 이력 추적관리
 - 소스 내용의 충돌에 대한 처리도 필요
 - 어떤 파일이 언제 어떻게 삭제되고 추가 됐는지 확인이 가능

TRACK PROJECT HISTORY



- 커밋(commit): 사전적 의미: ~적어두다
- 버전관리의 커밋
 - 저장소의 현 상태를 저장하는 행위
 - 현 상태를 담은 스냅샷 사진을 찍는 것에 비유
 - 파일 집합의 변경 내용을 깃 저장소에 기록하는 작업
 - 어느 시점의 파일 집합(폴더)의 추가/변경 사항을 저장소에 기록
 - 이전 커밋 상태부터 현재 상태까지의 변경 이력이 기록된 커밋이 생성
- 4번의 커밋
 - 시간순으로 저장
 - 최근 커밋부터 거슬러 올라가면 과거 변경 이력과 내용을 알 수 있음



- 커밋
 - 프로젝트의 상태를 저장
- 저장소 관리
 - 스냅샷의 연속(연속된 커밋)으로 관리
 - 파일이 달라지지 않았으면 파일을 새로 저장하지 않음
 - 단지 이전 상태의 파일에 대한 링크만 저장

가장 최근의 커밋을
가리키는 포인터:
HEAD

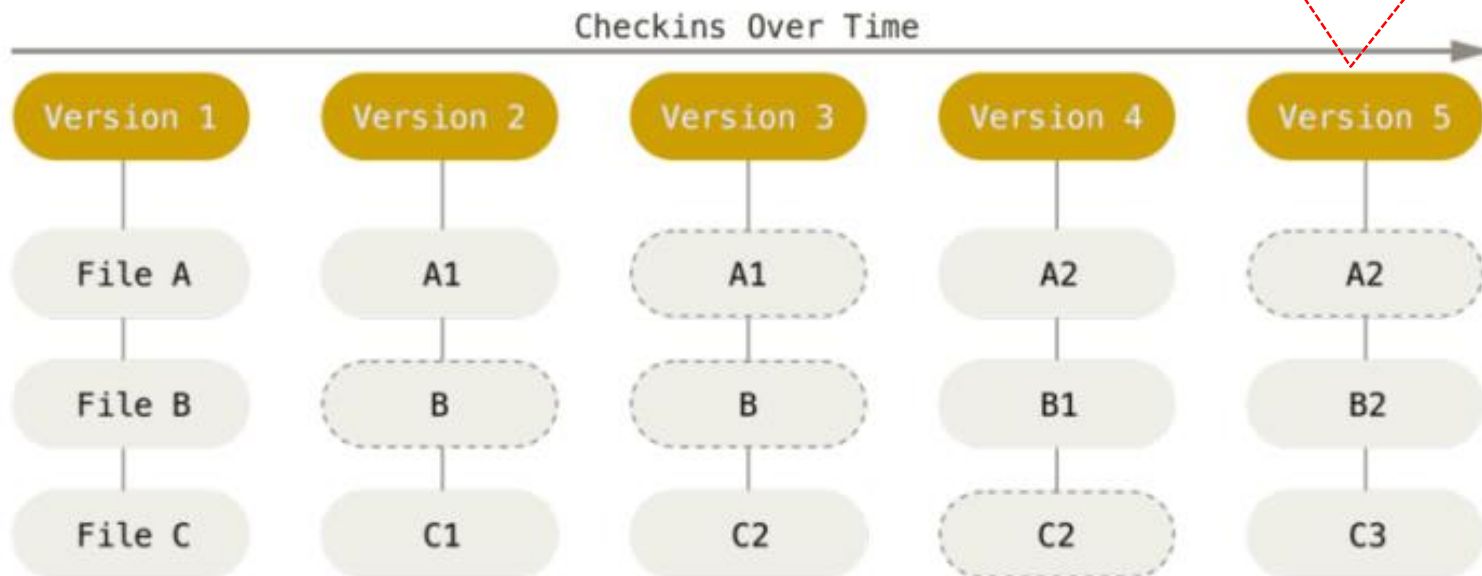


그림 5. 시간순으로 프로젝트의 스냅샷을 저장.

- 저장소(Git repository)

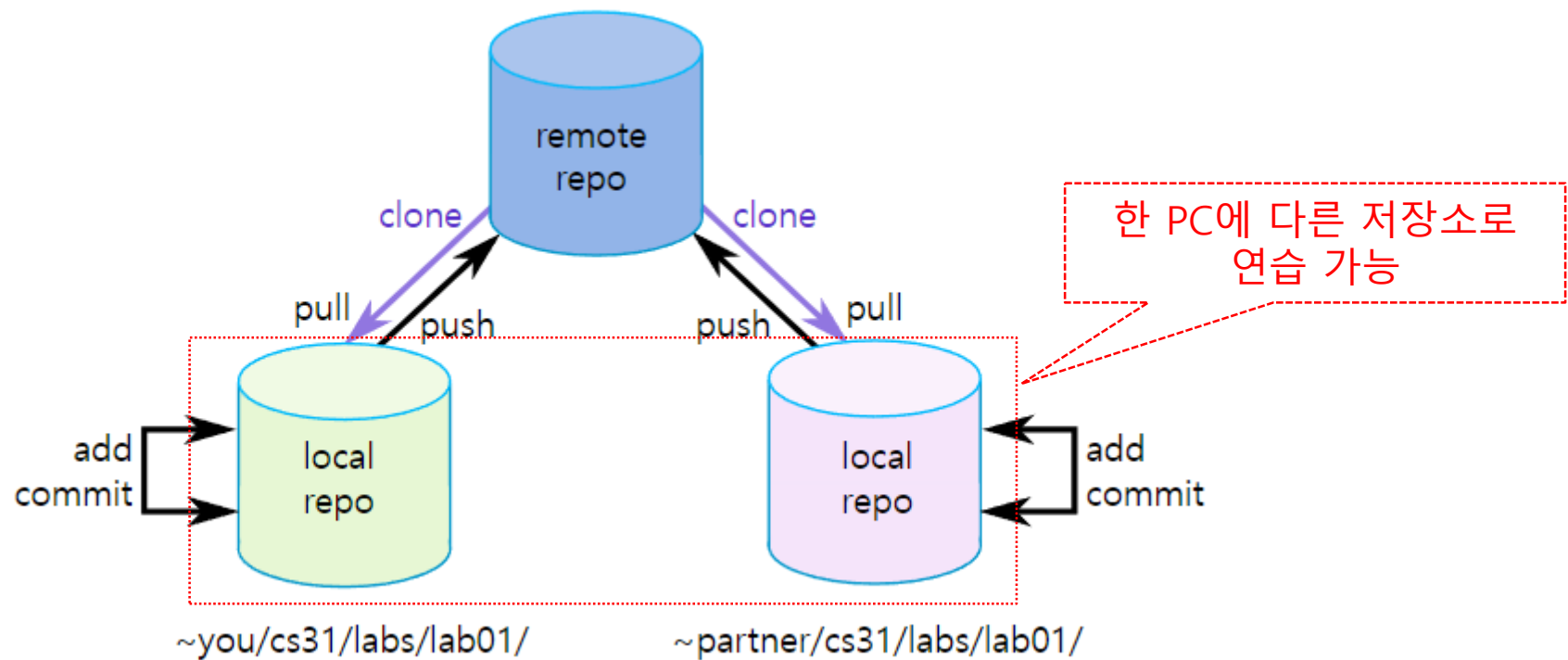
- 파일이나 폴더를 저장해 두는 곳
 - 파일이 변경 이력 별로 구분되어 저장



- 일반적으로 두 종류의 저장소를 제공

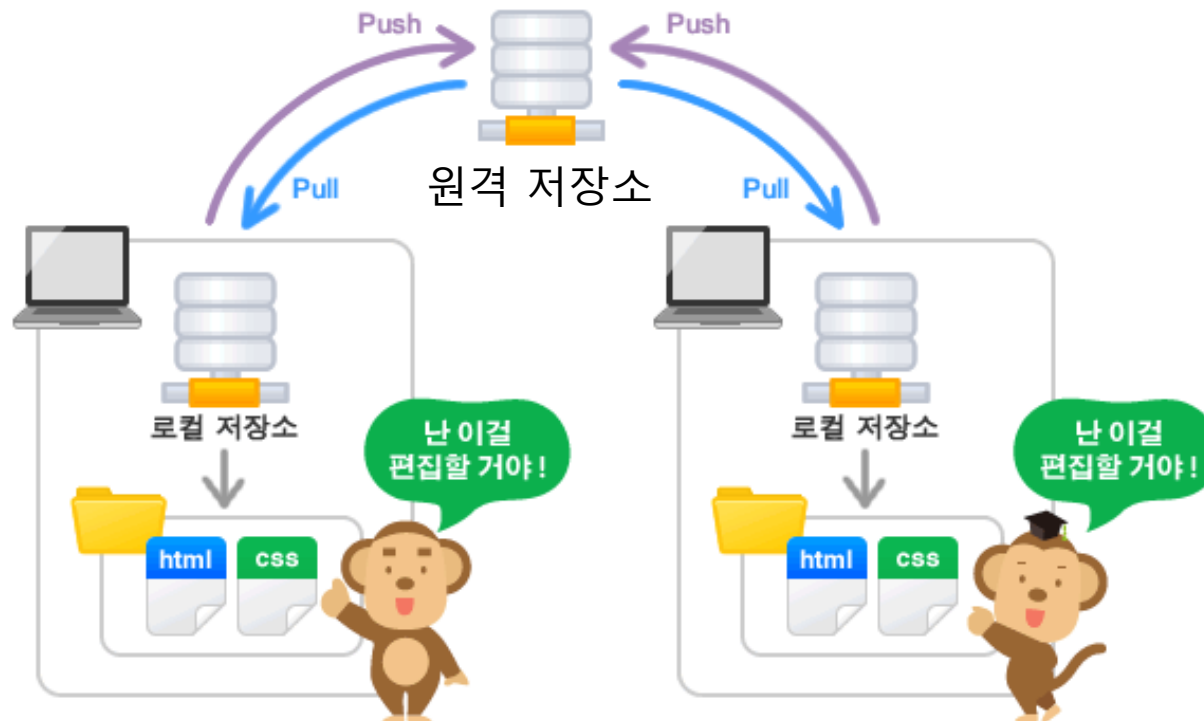
- 원격 저장소(Remote Repository)
 - 파일이 원격 저장소 전용 서버에서 관리되며 여러 사람이 함께 공유하기 위한 저장소
- 지역(로컬) 저장소(Local Repository)
 - 내 PC에 파일이 저장되는 개인 전용 저장소

- **Push**
 - 서버로 올리기
- **Pull**
 - 지역 저장소로 내리기
- **파일을 저장소에 저장하는 명령어**
 - add
 - commit



- Push와 pull 활용

- Push: 원격저장소 올리기
 - 평소에는 내 PC의 로컬 저장소에서 작업하다가
 - 작업한 내용을 공개하고 싶을 때에 원격 저장소에 업로드(push)
- Pull: 원격 저장소에서 지역 저장소로 내리기
 - 물론 원격 저장소에서 다른 사람이 작업한 파일을
 - 로컬 저장소로 가져(pull)올 수도 있음



- **Git**
 - 가장 많이 활용되는 분산 버전 관리 시스템

Top 5 Open Source Version Control Tools for System Admins



CVS



SVN



GIT



Mercurial



Bazaar

+1 (888) 571-6480

getfilecloud.com

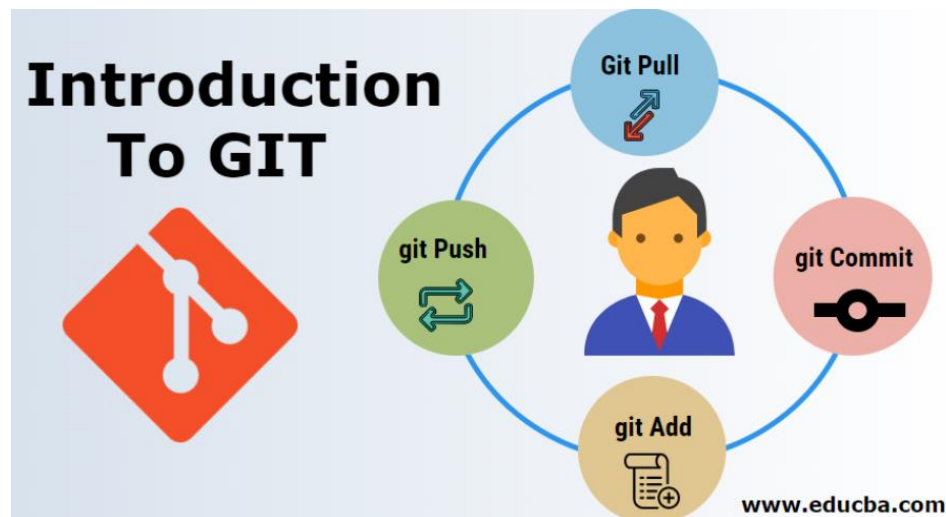
AI Experts
Who Lead
The Future

02

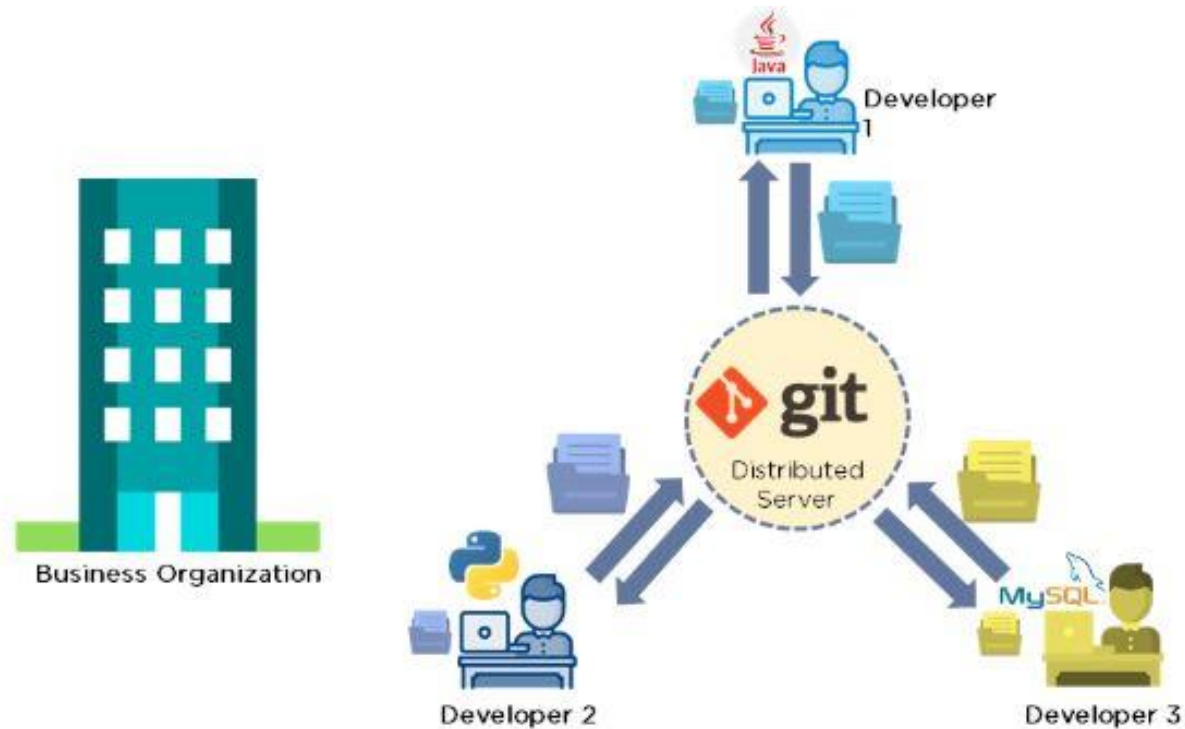
깃과 깃허브 개요



- 2005년, 리누스 토발즈가 개발
 - 주니오 하마노(Junio Hamano)가 소프트웨어의 유지보수
- 대표적인 오픈 소스 소프트웨어 프로젝트
 - <https://github.com/git>
 - <https://git.kernel.org/pub/scm/git/git.git>



- 모든 개발자는 지역 시스템에 코드의 전체 사본을 소유
 - 소스 코드에 대한 모든 변경 사항은 다른 사용자가 추적 가능



- 정의

- 컴퓨터 파일의 변경을 추적하는 데 사용되는 버전 관리 시스템

- 기능

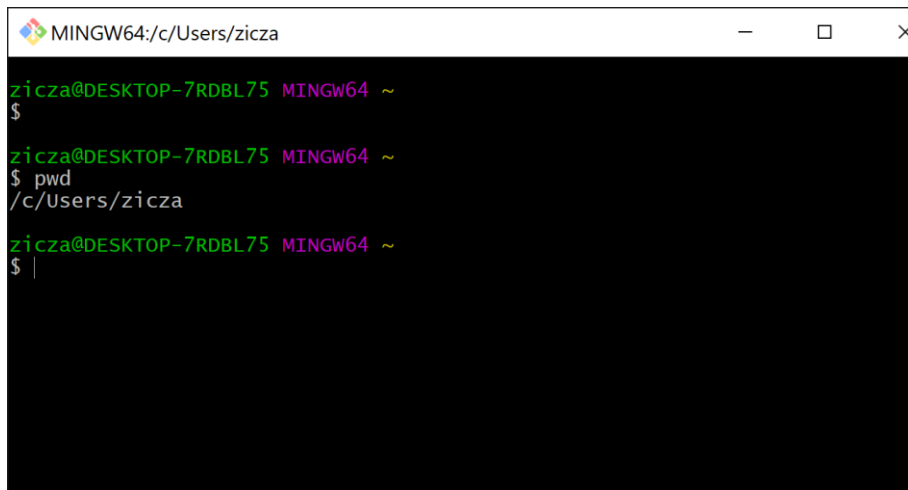
- 여러 개발자가 함께 작업
- 소스 코드의 변경 사항을 추적하는 데 사용
- 소스 코드 관리에 분산 버전 제어 도구가 사용
- 여러 개의 평행 분기를 통해 비선형 개발을 지원

- 특징

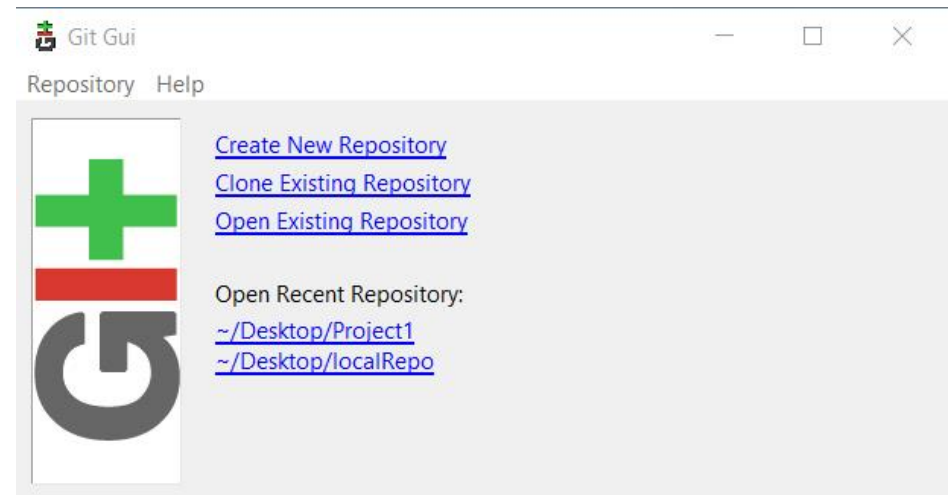
- 기록 추적
- 백업 생성
- 협업 지원
- 분산 개발
- 비선형 개발 지원
 - 브랜치 지원
- 자유 및 오픈 소스에 적합



- **Git Bash: CLI(Command Line Interface)**
 - 명령 행 인터페이스
 - 처음엔 어렵지만
 - CLI를 사용할 줄 알면 GUI도 사용할 수 있지만
 - 반대는 성립하지 않음
 - Mac의 Terminal
 - Windows의 CMD나 Powershell
- **Git GUI: GUI(Graphical user interface)**
 - GUI 프로그램의 대부분은 Git 기능 중 일부만 구현하기 때문에 비교적 단순



```
MINGW64: c/Users/zicza
zicza@DESKTOP-7RDBL75 MINGW64 ~
$
zicza@DESKTOP-7RDBL75 MINGW64 ~
$ pwd
/c/Users/zicza
zicza@DESKTOP-7RDBL75 MINGW64 ~
$ |
```

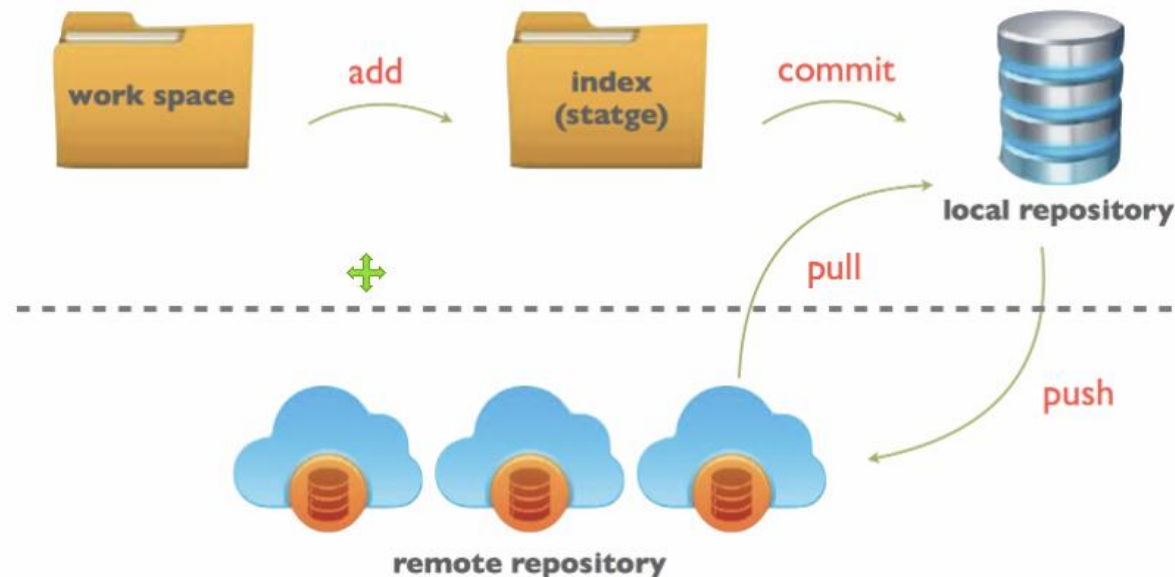


- 깃 3 단계와 3 상태

- 작업 디렉토리(working directory, 작업 공간 work space, 작업 트리 working tree)
 - modified, untracked
- 스테이징 영역(staging area, stage area, index)
 - staged, indexed
- 깃 저장소(git repository, repository, .git directory)
 - committed

- 상태 간의 이동 명령

- add
- commit



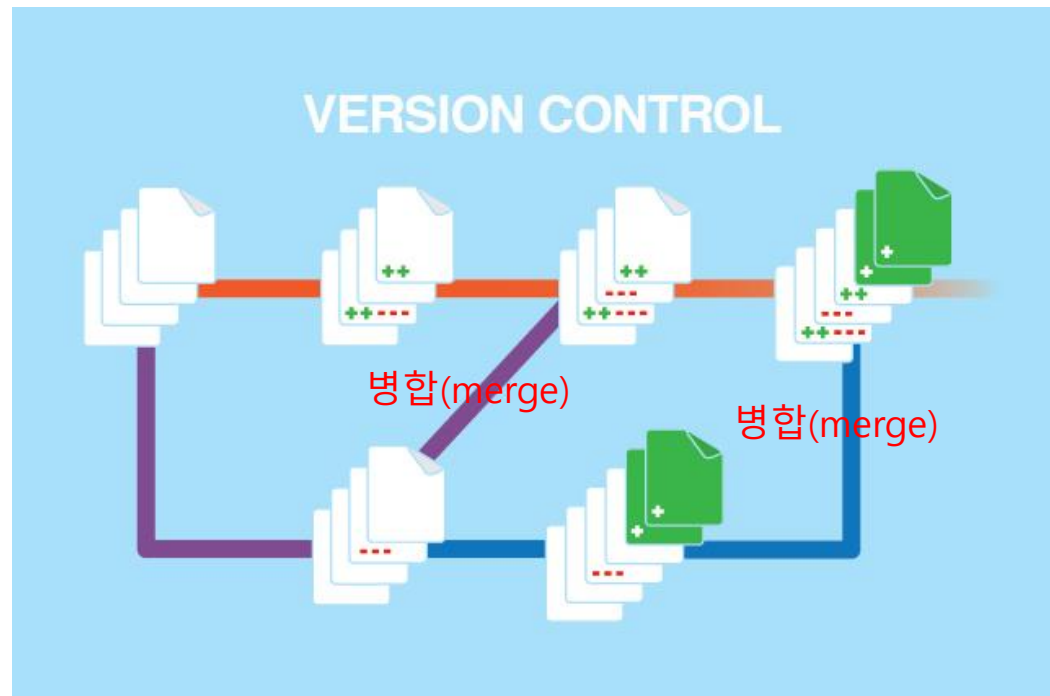
- 나무 가지



- 지점

- 분기

- **우리의 실 생활**
 - 여러 파일을 제목의 변형으로 버전을 관리하다가
 - 파일 집합을 다시 버전 시작
 - 새로운 폴더에 복사해서 사용하는 것과 비슷
- **분기, 가지(branch) 개념**
 - 새로운 수정을 할 수 있는 또 다른 버전의 작업 흐름(workflow)



- Github

- 버전 관리를 위한 서버 저장소 및 프로젝트 개발을 위한 협업 관리 서비스
 - 시스템 개발자와 운영을 담당하는 정보기술 전문가 사이의 소통, 협업, 통합 및 자동화 지원
 - 전 세계 개발자가 함께 개발 지원
- 프로젝트 소스 공유와 협업 소프트웨어 빌드 플랫폼
 - 개발자들의 소셜 네트워크 서비스 SNS

Git (깃) ?
분산 버전 제어 SW



Github (깃허브) ?
Git을 위한 웹 저장소(Repository)
+ 커뮤니티 협의공간

+ 시스템 개발자와 운영을 담당하는 정보기술 전문가 사이의
소통, 협업, 통합 및 자동화 지원

Github

분산 환경 소스코드 버전 관리 도구인
Git의 글로벌 웹 호스팅 서비스


세계 최대의 오픈 소스 공유 플랫폼

- ▶ 공개/비공개 저장소, 코드 리뷰, 문서화, 커뮤니티 등 제공
- ▶ 무료 서비스 사용 시 공개 저장소만 사용 가능
 - ✓ 모두에게 소스 코드 내용 오픈
 - ✓ 비공개 저장소 사용 시 유료 서비스 가입




Github

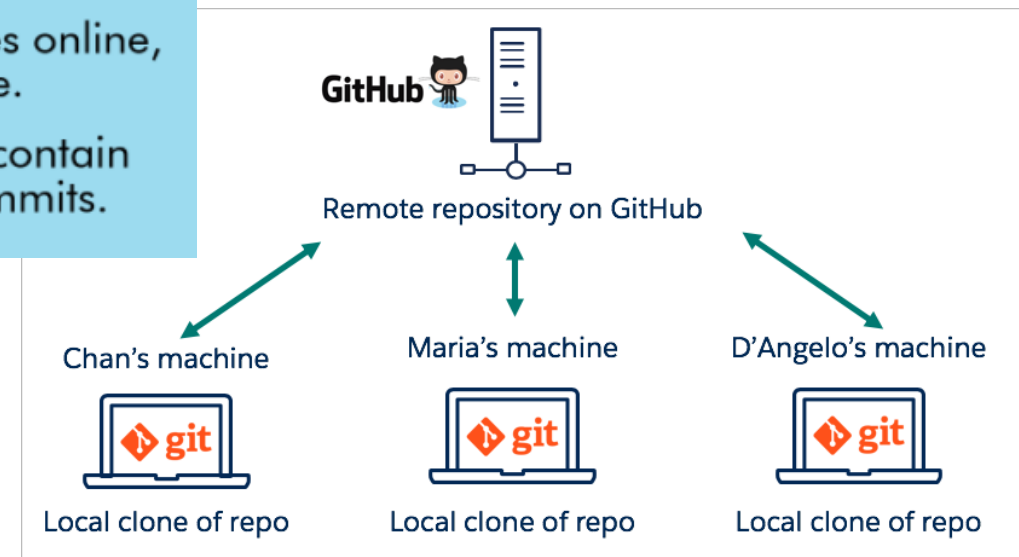
분산 환경 소스코드 버전 관리 도구인
Git의 글로벌 웹 호스팅 서비스

- ▶ 2018년 8월 기준 9600만 개 이상의 프로젝트 호스팅
3100만 명 이상의 개발자 참여
- ▶ 2018년 6월 4일 마이크로소프트사 75억 달러 인수 발표



WHAT ARE GIT & GITHUB?

-  **Git** is a version control system to keep track of changes to files and projects over time.
-  **GitHub** is a website that hosts Git repositories online, making it easier for developers to share code.
-  **Repositories** (or "repos") are folders which contain intentional snapshots of progress called commits.



<https://www.coursereport.com/blog/what-is-github>



- 문서 형식화 언어

- 마크다운은 일반 텍스트 문서에 형식 요소를 추가하는 데 사용할 수 있는 경량 마크업 언어
- 현재 세계에서 가장 인기 있는 마크업 언어 중 하나

- 2004년 존 그루버가 개발

- <https://www.markdownguide.org/>

Markdown Cheat Sheet

A quick reference to the Markdown syntax.

Overview

This Markdown cheat sheet provides a quick overview of all the Markdown syntax elements. It can't cover every edge case, so if you need more information about any of these elements, refer to the reference guides for [basic syntax](#) and [extended syntax](#).

Basic Syntax

These are the elements outlined in John Gruber's original design document. All Markdown applications support these elements.

Element	Markdown Syntax
Heading	# H1 ## H2 ### H3
Bold	bold text