

ROLO - NOMAD CODERS NAPOLITAN

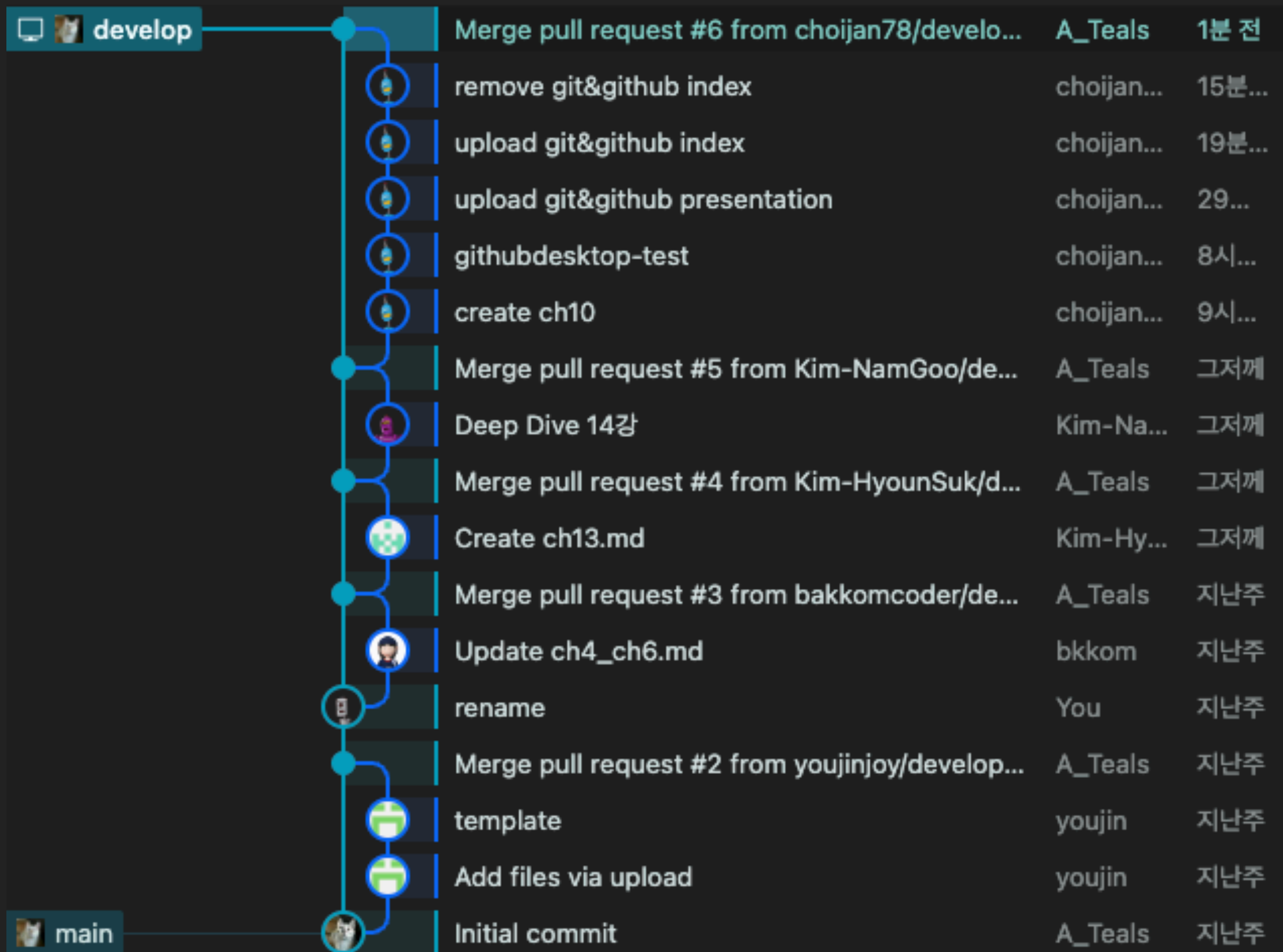
GIT & GIT HUB

용어정리 및 커밋방법 상세설명

작성일 : 2023.05.17

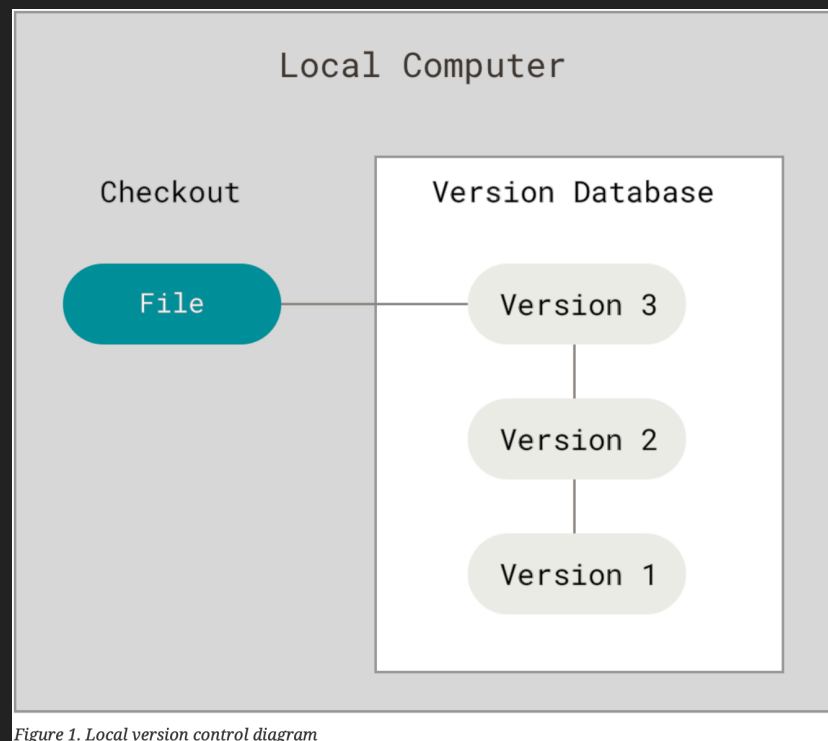
출처 : progit.pdf, 위키백과, Doeveloper님 명강의(5/16)

히스토리 GUI

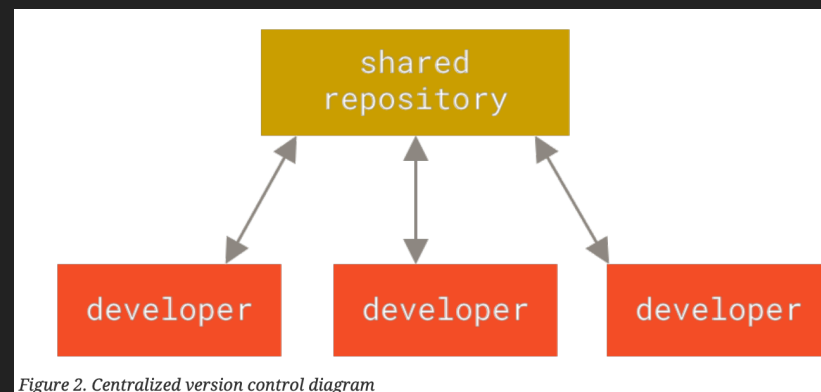


VERSION CONTROL SYSTEM 의 종류

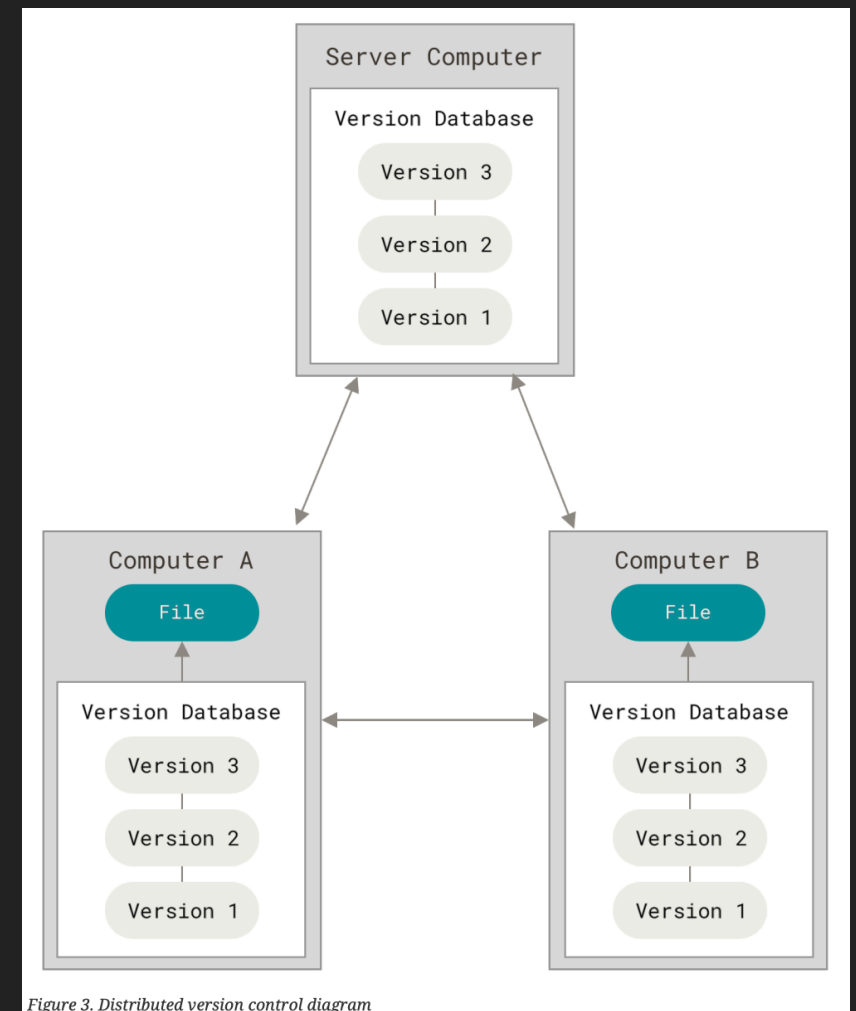
LOCAL



CENTRALIZED



DISTRIBUTED



단점 : 한곳에 저장된 파일이 날아가면, 모든 프로젝트를 망친다

- ▶ 마치 파일 버전1, 버전2 저장하듯, 따로 저장을 하되, patch set을 이용
- ▶ 대표프로그램 : RCS
- ▶ 몇년 간 기준이 되어 이용됨, 1개의 중앙 저장소에 모든 사람이 접속하여 파일을 업데이트함.
- ▶ 대표프로그램 : SVN, Perforce
- ▶ 최근 가장 핫함. 파일의 마지막 상태만 가져오는 것이 아니라, 모든 히스토리를 가져온다. 클론 레파지토리는 풀백업이다.
- ▶ 대표프로그램 : Git, Mercurial, Bazaar or Darcs)

깃의 탄생배경

- ▶ 2002년 : 리눅스 커널 프로젝트가 BitKeeper 라는 분산 버전 관리 시스템을 사용
- ▶ 2005년 : 리눅스 커널 프로젝트가 더이상 BitKeeper를 사용하지 못하게됨에따라 리눅스 창시자 리누스 토르발드가 자신만의 버전관리시스템을 개발
- ▶ 리누스 토르발드의 새로운 버전관리 시스템의 개발목표
 - ▶ 빠른속도
 - ▶ 간단한 디자인
 - ▶ 수천개의 브랜치를 사용할 수 있는 병렬적인 개발지원
 - ▶ 완전히 분산된
 - ▶ 리눅스 커널 프로젝트같은 아주 큰 프로젝트를 다룰 수 있도록 (속도와 용량면에서)

깃이 갖는 이유 (깃의 장점)

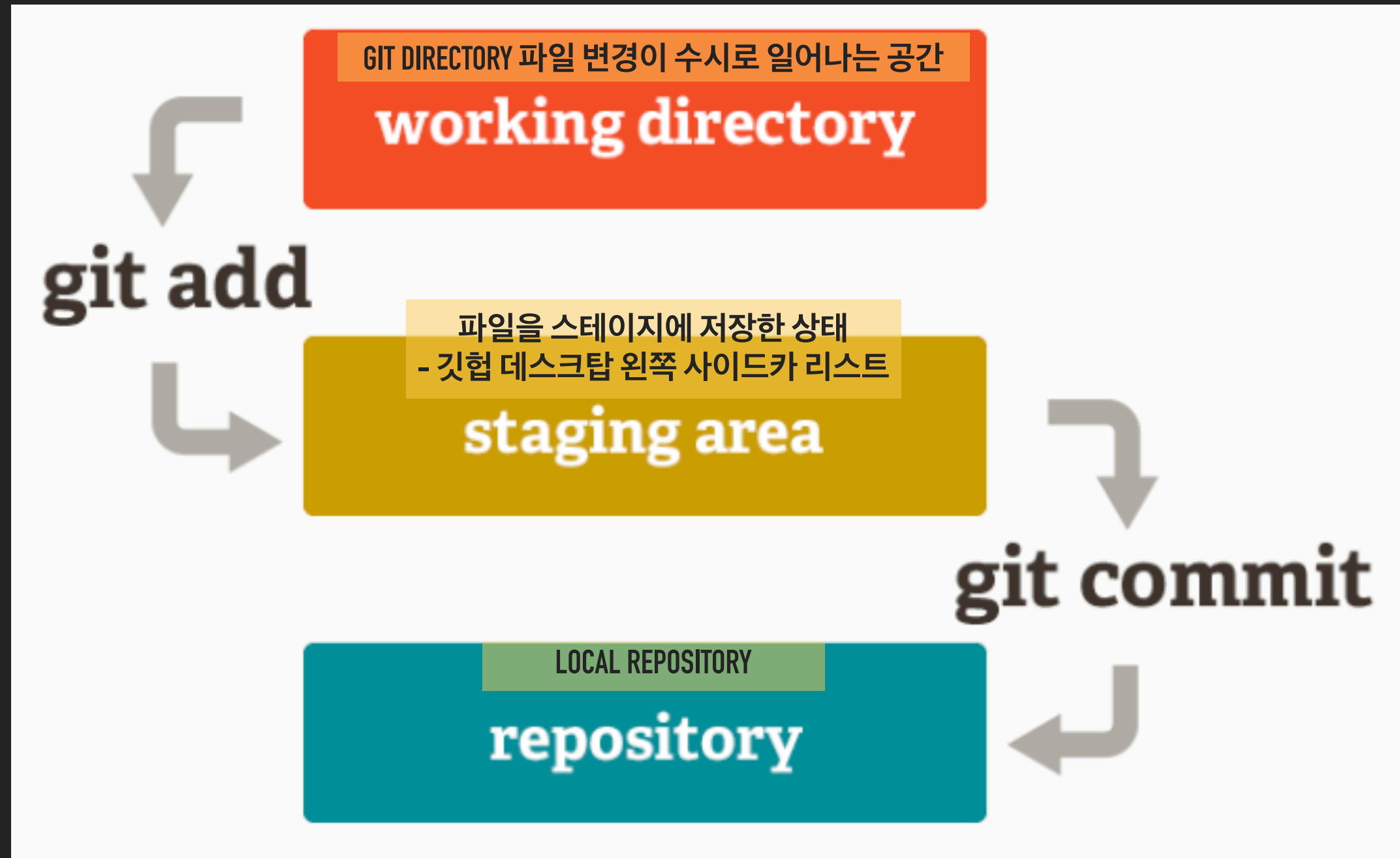
1. 파일의 저장시점만 저장하는 것이 아니라, 변경사항만 저장한다. 변경된 것이 없으면 덮어쓰기같은 작업을 하지 않는다. 그냥 저장을 안함.(변경사항없음)
2. 저장 할때마다 현재 프로젝트의 파일 전체를 스냅샷을 찍는다고 생각하면된다.
3. 이전버전과 이후버전 비교가 용이하다.
4. 이전버전데이터를 당겨올 (pull) 수 있다.
5. 로컬 컴퓨터에서 작업한다.
 1. 인터넷없는 곳 (비행기, 시골)에서도 커밋을 할 수 있다.
 2. VPN이 없을때도 커밋을 할 수 있다.
 3. 히스토리 확인 속도가 매우 빠르다. (로컬에 모든 프로젝트의 히스토리가 저장되어 있기 때문에)
6. 정보를 잃기가 어렵다.

모든 정보를 추가로 저장하고 추가 데이터를 집어 넣는 개념이기때문에

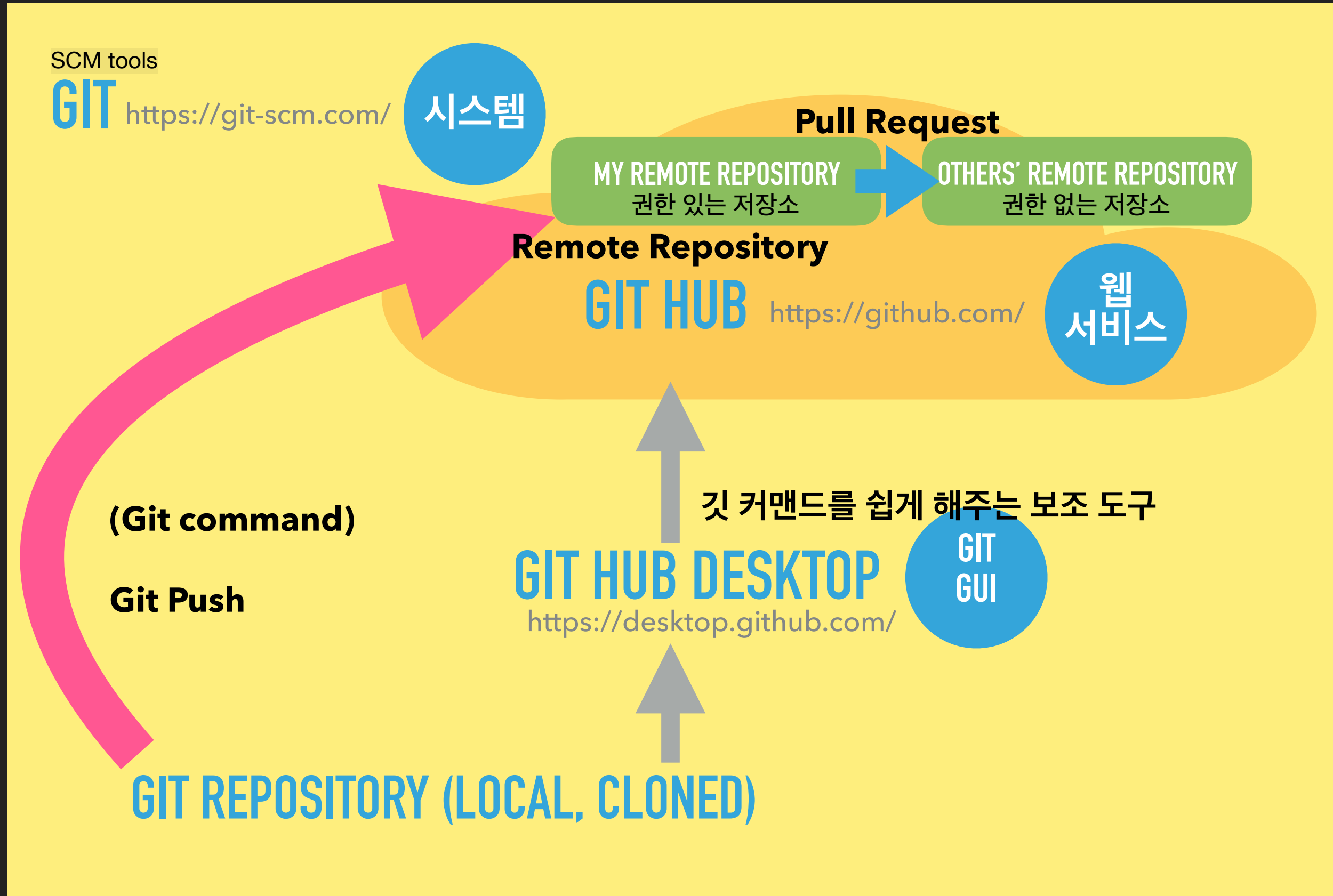
깃의 저장방법 및 온전함 (INTEGRITY)

- ▶ 체크섬을 이용해 저장
(체크섬: 나열된 데이터를 더하여 체크섬 숫자를 얻고, 정해진 비트수의 모듈로 정해진 비트수로 재구성 한다)
- ▶ 이로 인해, 깃이 모르게 파일을 저장하고 변경하고 삭제할 수가 없다. 깃이 체크섬 방법으로 저장을 했기때문에, 변경시에 "해독"을 하는 작업이 필요하므로, 깃이 그 변경사항을 지나칠 수가 없다.
- ▶ 깃의 메커니즘 : SHA-1 hash
->SHA-1 hash의 포맷 (16진수를 이용한 40자리 String)
24b9da6552252987aa493b52f8696cd6d3b00373

3단계 영역 (AREAS, STATES)



GIT & GIT HUB DESKTOP & GIT HUB 의 관계



Q&A