[GIT] ⚡ git 개념 & 원리

Git 이란 무엇인가?

Git이란 분산형 버전 관리 시스템(Version Control System) 의 한 종류이며, 빠른 수행 속도에 중점을 둔다.

우리가 레포트를 제출한다고 가정했을때, 처음에 저장했을때 'report.txt'라고 저장을 했다가 수정을 하면서 'report_final.txt'로 저장하고 또 수정을 거치면서 'report_final_final.txt'...로 수정을 하게 되는 경험이 있었을 것이다. 여기서 이 파일들을 복사, 백업, 저장 등을 한것이고, 이러한 것을 버전 관리라고 부른다.

버전 관리란?

버전관리 시스템은 파일변화를 시간에 따라 기록했다가 나중에 특정시점의 버전을 다시 꺼내올 수 있는 시스템이다.

- 각 파일을 이전 상태로 되돌릴 수 있다.
- 프로젝트를 통째로 이전 상태로 되돌릴 수 있다.
- 시간에 따라 수정 내용을 비교해 볼 수 있다.
- 누가 문제를 일으켰는지도 추적할 수 있다.
- 누가언제만들어낸이슈인지도알수있다.
- 파일을 잃어버리거나 잘못 고쳤을 때도 쉽게 복구할 수 있다.

Git 의 필요성

1. 나와 내 동료가 같은 웹사이트에서 동시에 같은 'A'페이지를 업데이트 하고 있다고 하자.
2. 나는 무언가를 변경하고 저장한 다음 웹사이트에 'A'페이지를 업로드 한다.
3. 그런데 이때 동료가 동시에 'A'페이지에서 작업을 할 때 문제가 발생된다.
4. 만약 확인하지 않고 동시에 작업을 한다면 누군가의 작업은 겹쳐 쓰여질 것이고 지워질 것이기 때문이다.
5. Git은 이와 같은 일을 사전에 방지해준다.
6. 나와 동료는 같은 페이지에 각자 수정사항을 업로드 할 수 있고, 두개의 복사본을 저장한다.

즉, 모두 같은 환경에서 개발하여 불필요한 시간을 없애고 서로 주고 받는 와중에 일어나는 충돌을 최소화 하는 것이라고 말할 수 있다.

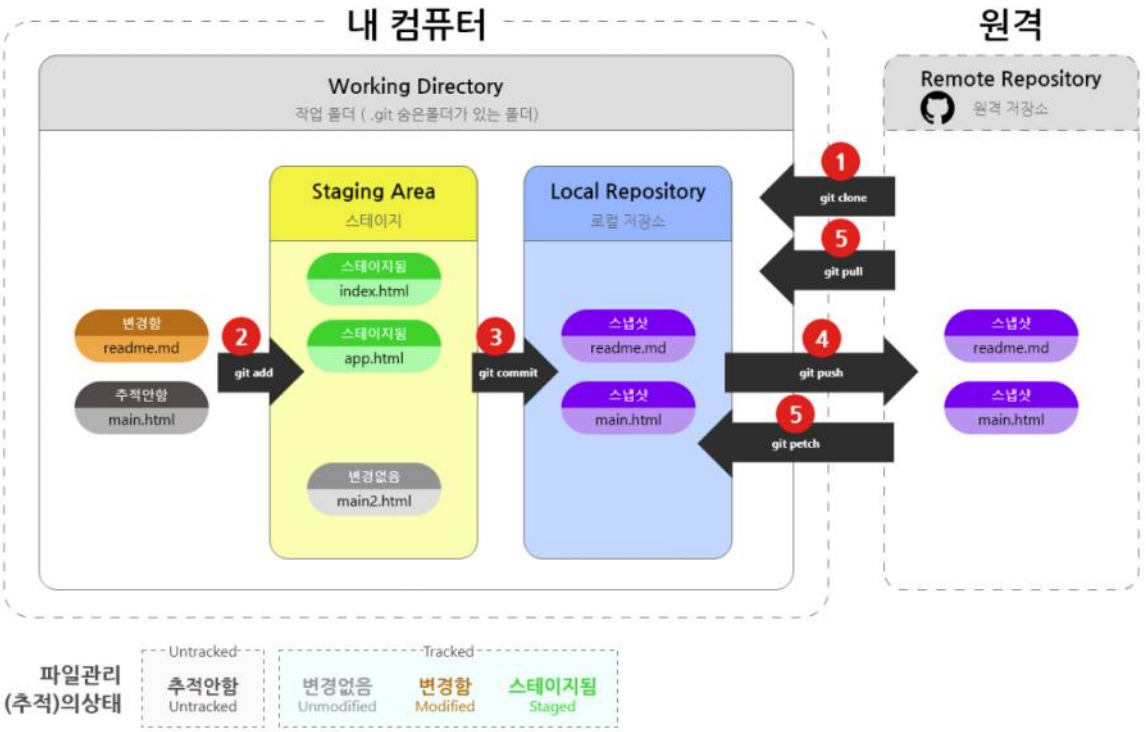
프레임워크를 갖추고 공통으로 쓰는 코드들을 미리 넣어두고, 테스트 코드나 샘플코드들을 넣어주어서 프로젝트 개발 시작하기전에 팀원 모두 같은 코드로 테스트할 수 있는 환경도 제공할 수 있다.

Git 의 장점

- 소스 코드를 주고 받을 필요 없이, 같은 파일을 여러 명이 동시에 작업하는 병렬 개발이 가능하다.
  (브랜치를 통해 개발한 뒤, 본 프로그램에서 합치는 방식(Merge)으로 개발을 진행할 수 있다.)
- 분산 버전 관리이기 때문에 인터넷이 연결되지 않은 곳에서도 개발을 진행할 수 있고, 중앙 저장소가 날라가 버려도 원상복구할 수 있다.
- 팀 프로젝트가 아닌, 개인 프로젝트일지라도 Git을 통해 버전 관리를 하면 체계적인 개발이 가능해지고, 프로그램이나 패치를 배포하는 과정도 간단해진다.

Git 작동 구조

7. 내가 작업한것을(로컬저장소)
8. 원격저장소로 밀어내거나(push)
9. 원격저장소에서 최신 형상을 내 작업공간으로(로컬저장소) 당겨온다(pull)

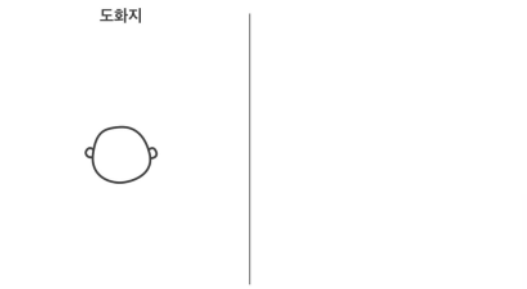


Git 버전 관리 원리 이해하기

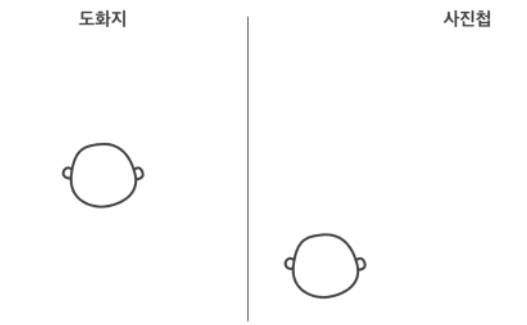
그림으로 이해하는 버전 관리 원리

만일 도화지에 그림을 그리고, 각 그림 완성도 마다 버전 관리를 한다고 가정하자.

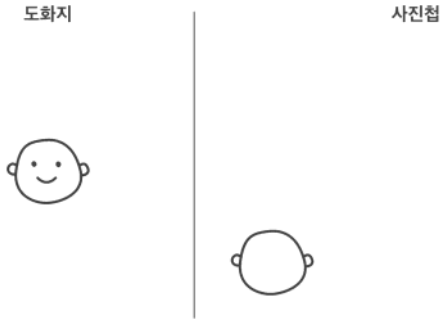
먼저 그림을 그리고,



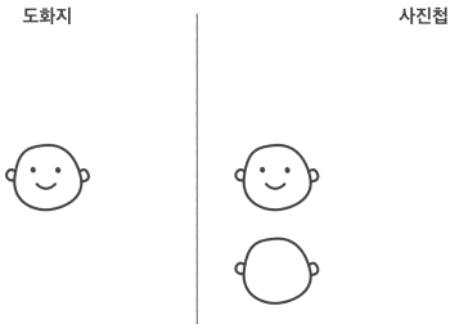
이렇게 사진첩 현재 상태를 저장을 한다.



그리고 내일 작업을 이어나가 눈코입을 그렸다고 하자.



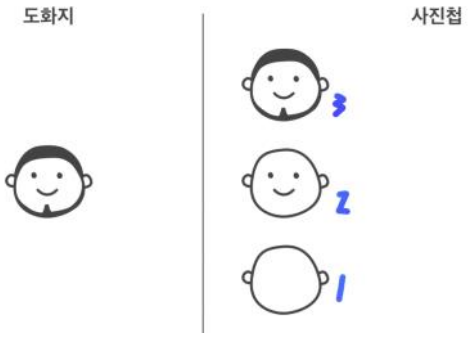
다시 작업한 내용을 사진첩에 추가로 저장했다.



머리와 수염을 그리고 이 역시 사진첩에 추가로 저장을 했다.

그러면 사진첩에는 총 3개의 작업내용이 버젼별로 저장된 것을 확인 할 수 있다.

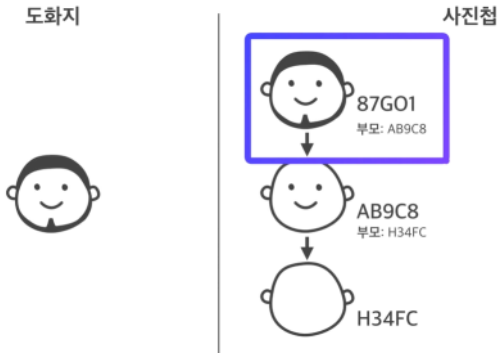
그리고 각 작업들의 순서를 매겨줘서 버젼 순서를 기억한다.



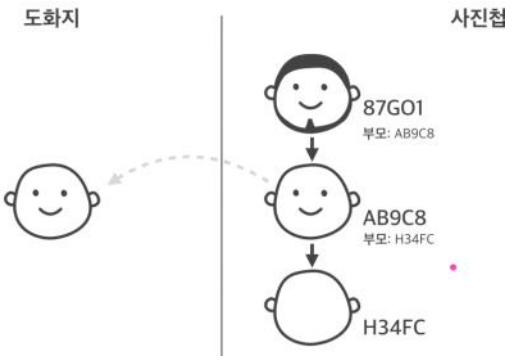
하지만 git은 숫자로 인덱싱하지않고, 해시값으로 인덱싱을 한다.

해시값은 중복이 안된다고 봐도 무방하다.

단 해시값으로 순서를 알수없으니, 트리 처럼 부모를 각 노드마다 기록해주어 진행순서를 알수있게 한다.

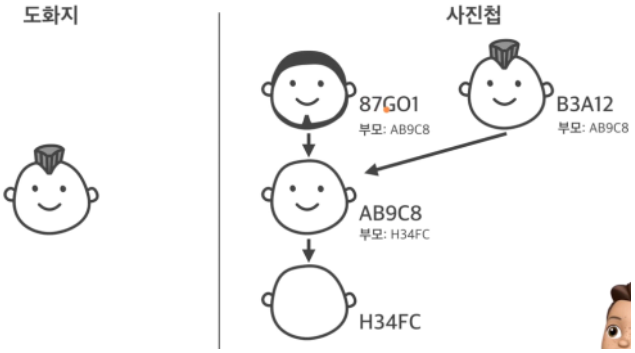


만일 머리를 다시 그리고 싶다면, 이전 버젼인 해시값 AB9C8을 참조하여 다시 가져오면 된다.



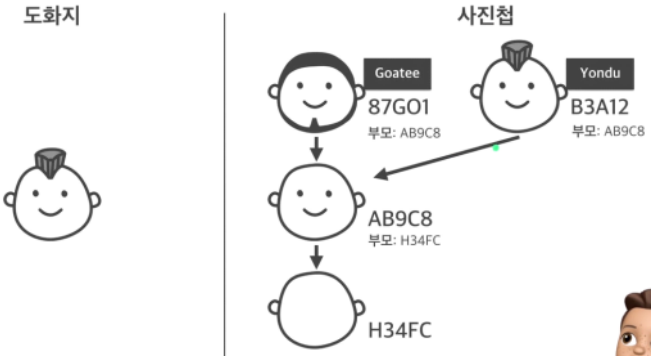
그리고 머리를 다시 그리고, 사진첩에 추가로 저장한다. (B3A12)

그리고 작업순서를 기억하기 위해 부모를 기록한다.



그리고 작업이 분기로 두가지로 나뉘어지니까,

어느작업이 무엇인지 알기 쉽게 작업명을 붙여준다. (Goatee / Yondu)



Git 기본 용어

Git을 사용하기 위해 알아야 할 용어 몇가지가 있다.

Repository

스테이지에서 대기하고 있던 파일들을 버전으로 만들어 저장하는 곳이다.

Git은 원격 저장소와 로컬 저장소 두 종류의 저장소를 제공한다.

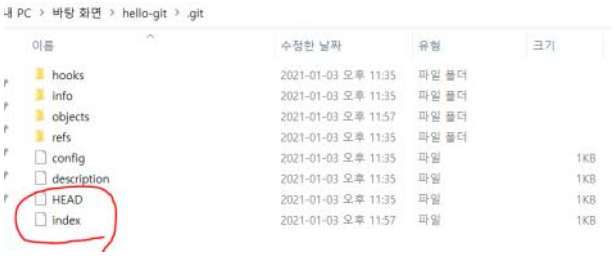
원격 저장소(Remote Repository)

파일이 원격 저장소 전용 서버에서 관리되며 여러 사람이 함께 공유하기 위한 저장소다.

로컬 저장소(Local Repository)

내 PC에 파일이 저장되는 개인 전용 저장소다.저장소를 만드는 방법은 두 가지가 있다.

아예 저장소를 새로 만들거나, 이미 만들어져 있는 원격 저장소를 로컬 저장소로 복사해 올 수 있다.



스테이지 내용은 .git/index 파일에 저장되고, 저장소의 내용은 .git/HEAD 파일에 저장된다.

Working Tree (Working Directory)

저장소를 어느 한 시점을 바라보는 작업자의 현재 시점.

파일 수정, 저장 등의 작업을 하는 디렉터리로, '작업 디렉터리(working directory)'라고도 한다.

SnapShot

특정 시점에서 파일, 폴더 또는 워크스페이스의 상태를 의미.

스냅샷을 통해 특정 시점에 어떤 파일에 어떤 내용이 기록되어 있었는지, 폴더 구조는 어떠했는지, 어떤 파일이 존재했는지 등 저장소의 모든 정보를 확인할 수 있다.

Git에서는 새로운 버전을 기록하기 위한 명령인 커밋(commit)을 실행하면 스냅샷이 저장된다

10. 개발자는 작업 디렉토리에 있는 파일을 수정
11. 수정된 파일을 모아 정리하여 만든 Snapshot을 Staging 디렉토리에 추가하고 저장
12. GIT 디렉토리에 저장 (Staging 디렉토리에 저장된 파일을 앞으로 영구 불변의 상태를 유지하는 Snapshot 으로서 git 디렉토리에 저장하는 것)

Checkout

이전 버젼 작업을 불러오는것.

Staging Area

저장소에 커밋하기 전에 커밋을 준비하는 위치.

예를 들어 작업 트리에서 10개의 파일을 수정했는데 4개의 파일만 버전으로 만들려면 4개의 파일만 스테이지로 넘겨주면 된다. 즉, 로컬 스테이지에 올려둔 파일만 원격 저장소에 커밋할 자격이 있는 것이다.

Commit

현재 변경된 작업 상태를 점검을 마치면 확정하고 저장소에 저장하는 작업.

Head

현재 작업중인 Branch를 가리킨다.

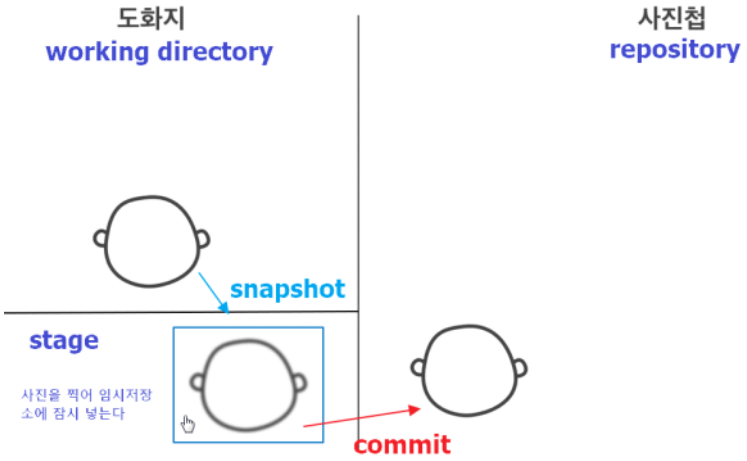
Branch

가지 또는 분기점을 의미하며, 작업을 할때에 현재 상태를 복사하여 Branch에서 작업을 한 후에 완전하다 싶을때 Merge를 하여 작업을 한다.

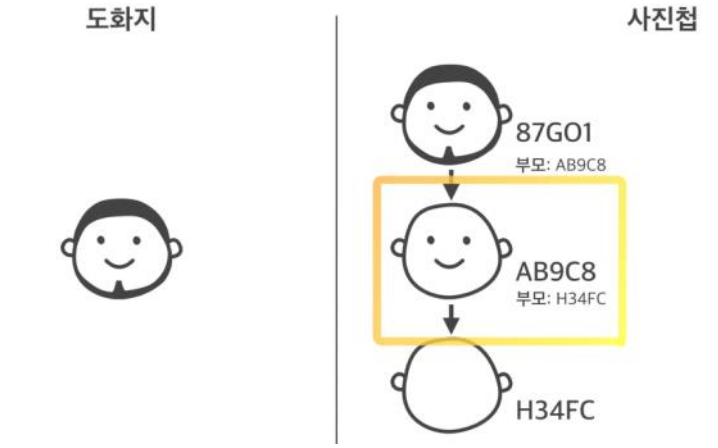
Merge

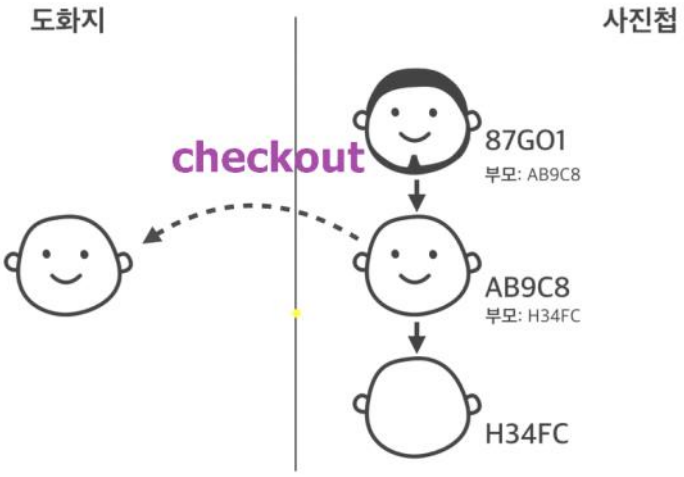
다른 Branch의 내용을 현재 Branch로 가져와 합치는 작업을 의미한다.

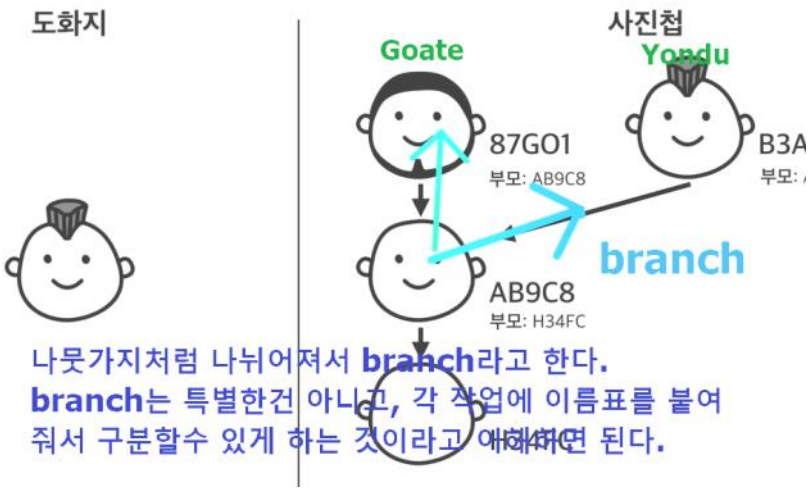
이제 이 용어들을 위의 사진첩에 빗대어 다시 보자.



만일 이전 버젼 파일이 필요하다면..







Git 호스팅 웹 서비스

깃 저장소 서버를 대신 유지 및 관리해주는 서비스 다.

다른 유저들과 함께 온라인으로 하나의 프로그램을 제작하는것도 가능하며 많은 오픈소스 프로그램들이 Github를 통해서 전 세계 개발자들에 의해 제작되고 있다. 주로 개인은 GitHub, 회사는 GitLab를 사용하는 편이다.

GitHub

GitLab

Gitea

Github vs GitLab

깃허브는 세계최대 규모의 Git 저장소로 무료 서버저장소를 지원하기때문에, 초급~고급 개발자 모두 이용하며, 수많은 개인 개발자들, 팀개발자들이 이용하는 서비스이다.

깃허브는 무료로 이용하는대신 자신의 소스코드가 오픈되어 수많은 사람들이 보며 활용가능하고, 익명의 개발자와 함께 작업할수있도록 하여 프로그래밍을 더욱 확산시켜주는 환경으로 자리하고있다.

반면 GitLab(깃랩)은 수많은 기업에서 보안성을 중시하는 프로그램 코드를 올려 함께 협업하는 툴로 애용하므로, 사용자가 기업들이 되며, 자신의 서버에 설치하여 서버내 프라이빗한 Git 원격 저장소를 만들수있는 서비스이다.