

버전관리시스템

- 구체적인 이점
 - 변경이력을 기록을 통해서 변경된 내용을 공유 가능
 - 타인이 작업한 내용을 쉽게 병합
 - 과거 상태로 쉽게 복구 가능
 - 여러 분기(브랜치)를 통해 병렬 관리 가능
 - 파일의 변경이력을 기록하여 관리를 용이하게 해주는 것

GIT

- 복잡한 branch 관리에 적합
- 심플하지만 핵심적인 기능이 강력
- 로컬 저장소와 원격저장소의 분리
- 빠른 속도
- 다양한 서비스 업체
- 다양한 보조 툴
- git에서의 작업 흐름
 - working directory : 실제 작업하고 있는 공간
 - staging area(index) : git이 변경이력을 관리하는 부분, working directory에서 git 명령어를 통해서 추가 가능, 이 곳에 올라와 있는 파일만 저장소에 추가 및 수정 가능
 - 일종의 준비구역
 - repository : 변경이력을 저장한 저장소
 - local repository : 외부에 위치하지 않고 작업하고 있는 컴퓨터에 존재, 인터넷을 이용하지 않기 때문에 매우 빠른 속도, 인터넷이 연결되지 않아도 작업이 가능, 잦은 저장소 처리요청에도 부담이 없음, 외부 저장소 손실이 발생하더라도 빠르게 복구 가능
 - remote repository : 전통적인 관점에서의 저장소의 개념, 외부 서버에 위치하여 변경이력을 기록하는 부분, 인터넷을 이용하여 접근가능, 다중 사용자로부터 관리되는 각 로컬 저장소의 접점
 - 저장소의 위치만 내 컴퓨터에 있느냐, 서버에 있느냐의 차이
- git ignore - git에서는 git ignore 파일을 통해서 해당 파일은 stage영역에 올라가지 않도록 자동으로 관리해줌
- SVN저장소 - 기본적으로 수정된 모든 파일이 commit의 대상이 되기 때문에 선택적으로 파일 개개를 따로 commit을 해줘야 한다. 전체를 커밋할 경우 신규기능과 관련된 파일 b,c,d를 전체 내용을 저장소에 제출
- GIT저장소 - 두 기능을 따로 commit 하여 배포에서 일부 수정내용만 반영하여 배포하면 되며, 별도의 백업이나 복구의 단계가 필요없음