2023-03-03

Git & GitHub

Git개요

분산 버전 관리 시스템

- Distributed Version Control System(DVCS)
- 여러 사람이 협동 작업하는 환경에서 문서 변경사항을 관리하는 시스템

▼ 특징

- 변경사항을 적절히 저장했다가 필요한 시점으로 돌리기 가능
- 서로 다른 변경사항들을 쉽게 합칠 수 있음
- 저장소가 로컬(내컴퓨터)에 존재해 네트워크가 끊어져도 작업 가능
- 다른 버전 관리 시스템보다 빠름
- 원격 저장소를 연결해 협동 작업 가능
- 변경사항을 관리할 대상을 스테이지(Stage)관리 가능

▼ 파일 관리 상태

- Modified : 수정한 파일을 아직 로컬 데이터베이스에 커밋하지 않은것을 의미
- Staged : 현재 수정한 파일을 곧 커밋할 것이라고 표시한 상태를 의미
- Committed : 데이터가 로컬 데이터베이스에 안전하게 저장됐다는 것을 의미
- **Git 디렉토리는 메타데이터와 객체 데이터베이스가 저장되는 곳. 저장소를 Clone 할 때 생성**

▼ 파일 저장 방식

- Git은 데이터를 파일 시스템 스냅샷의 연속으로 취급하고 크기가 아주 작다
  - 커밋된 시점의 저장소 상태가 하나의 버전이 된다
- 파일이 달라지지 않으면 Git은 성능을 위해 파일을 새로 저장하지 않고, 이전 상태의 파일에 대한 링크만 저장
- Git은 데이터를 스냅샷 스트림으로 취급

- Git은 커밋 혹은 프로젝트의 상태를 저장할 때마다 파일이 존재하는 그 순간을 중요하게 여김

▼ 저장소에서의 세가지 파일상태

- Hisory
- Stage(Index)
- Working Directory

▼ 로컬 & 원격 저장소

- 로컬 저장소(Local Repository)
  - 내 PC에 파일이 저장되는 개인 전용 저장소
- 원격 저장소(Remote Repository)
  - 파일이 원격 저장소 전용 서버에서 관리되며 여러 사람이 함께 공유하기 위한 저장소
- 로컬 저장소에서 커밋한 작업 내용을 공개하고 싶을 때 원격 저장소에 Push한다.
- 원격 저장소에서 다른 사람이 작업한 파일을 로컬 저장소로 가져올 때 Pull한다.

▼ 버전 관리 시스템 (VCS- Version Control System)

- 파일 변경을 기록했다가 나중에 특정 시점의 버전의 코드를 체크 아웃할 수 있는 시스템
- 각 파일을 이전 상태로 되돌릴 수 있으며 버전의 수정 내용을 비교해 볼 수 있음
- 파일의 버전을 수정한 팀원을 찾을 수 있으며 특정 시점으로 돌아가거나 복구 가능
- 소프트웨어 개발에서 개인 또는 팀 단위로 개발중인 소스 코드나 메뉴얼 등의 디지털 문서를 관리하는데 사용

▼ 버전 관리 시스템 (LVCS)

- 로컬버전관리(LVCS)는 아주 간단한 데이터베이스를 사용해서 파일의 번경 정보를 관리
- 많이 쓰는 VCS 도구중에 RCS(Revision Control System)라고 있다
  - RCS는 기본적으로 Path Set(파일에서 변경되는 부분)을 관리
  - 이 Patch Set은 특별한 형식의 파일로 저장
  - 일련의 Patch Set을 적용하여 모든 파일을 특정 시점으로 되돌릴 수 있다

▼ 버전 관리 시스템 (CVCS)

- 중앙집중식 버전 관리(CVCS)는 파일을 별도로 관리하는 서버가 있다
  - 프로젝트에서 팀원들이 소스코드를 공유하기 위해 개발
  - CVS, Subversion, Perforce등이 대표적인 CVCS 도구
  - 변경된 파일을 저장 순서로 관리
  - 파일을 관리하는 서버가 별도로 존재하고, 클라이언트가 중앙 서버에서 파을을 Check out 하여 사용
  - 서버다운 혹은 하드디스크크래시와 같은 물리적인 문제 발생시에 프로젝트의 모든 히스토리가 없어질 수 있음

▼ **Git - 버전 관리 시스템 (DVCS)**

- 분산버전 관리시스템(DVCS)은 클라이언트에 단순히 마지막 버전의 소스를 체크아웃하지 않고 히스토리와 더불어 모두 복제
  - Git, Mecurial, Bazaar, Darcs등이 있다
  - 서버에 문제가 생기더라도 이 복제물로 작업가능
  - DVCS환경에서는 리모트 저장소(클라우드 원격 저장소)가 존재하여 다양한 그룹과 다양한 방법으로 협업 가능

소프트웨어 형상관리 → 소프트웨어 생명주기 및 유지보수 과정에서 만들어지는 각 단계별 산출물을 체계적으로 관리

GitHub

▼ Branch : 새 원격 저장소

- 커밋의 묶음이며 , 특정 커밋을 기준으로 Branch를 생성할 수 있다.
- Branch를 이용하면 여러 사람이 함께 진행하는 프로젝트에서 다른 Branch의 커밋에 영향을 받지 않고 작업할 수 있다.
- 이슈가 생겨서 작업할 경우 Branch를 생성하여 개발한다.
- 개발이 진행중인 상황에 배포된 버전에 문제가 생겼을 때 이슈를 개발중인 Branch는 유지하고, 배포된 커밋에 수정용 Branch를 생성하여 문제를 해결한다.
- 다양한 상황에서 유연하게 버전을 관리할 수 있다.

Git main Branch는 특별한 의미는 없고, git init 명령으로 초기화할 때 자동생성된 기본이름

---

cmd에서 새로운 vite 생성하기

```
D:\sini> npm create vite@latest
Need to install the following packages:
  create-vite@4.1.0
Ok to proceed? (y) y
√ Project name: ... mydktecinvue
√ Select a framework: » Vue
√ Select a variant: » JavaScript

Scaffolding project in D:\sini\mydktecinvue...

Done. Now run:

  cd mydktecinvue
  npm install
  npm run dev

D:\sini>
```