**Git**

Git이란 분산 관리 시스템으로써 파일의 변경 사항 추적, 사용자들 간의 해당 파일들의 작업을 조율하기 위한 소프트웨어이다. 분산 관리 시스템이란 여러 사용자가 서로 협업하여 파일을 업데이트 시키고 버전을 관리할 수 있는 시스템을 말한다.

**1. Git의 데이터 저장 방식**

Git은 중앙 서버 컴퓨터와 여러 개의 컴퓨터들이 연결되어 모두 같은 버전의 데이터베이스를 유지한다. 이 버전은 업데이트가 될 때마다 자동으로 생성된다. 따라서 이 파일들은 최신 버전으로 모든 컴퓨터에서 유지될 수 있다.

예를 들어 컴퓨터 A에서 업데이트한 것을 중앙 서버에 올리면 버전이 업데이트 되고, 컴퓨터 B는 이를 중앙 서버 컴퓨터로부터 최신화시키면서 버전과 파일을 모두 컴퓨터 A와 동일하게 유지할 수 있다. 이와 같은 방식을 분산 버전 관리 시스템이라 한다.

**2. Git의 특징**

1) 분산 개발환경

2) 비선형 개발 공간을 위한 지원

3) 대형 프로젝트의 효율적인 관리가 가능

4) 히스토리의 암호화 인증 제공

5) 다양한 툴킷 제공

**3. GitHub와 GitLab**

Git의 기능 데이터를 원격 저장소에 저장하는 기능을 제공해주는데, 여기서 원격 저장소 역할을 하는 것이 GitHub이다. 하지만 이를 비공개로 이용하고 싶다면 월정액을 지불해야하기 때문에 비공개 프로젝트도 무료로 지원하는 GitLab도 많이 사용되고 있다.

**4. 관련 용어**

- Repository : 저장소를 의미한다. 히스토리, 태그, 소스 등의 Branch에 따라 버전을 저 장할 수 있다. 또한 저장소를 통해 지금까지의 히스토리를 확인할 수 있다.

- Working Tree : 저장소의 어느 한 시점을 바라보는 작업자의 현재 시점을 의미한다.

- Staging Area : 저장소에 커밋하기 전에 준비하는 위치

- Commit : 현재 변경된 작업 상태를 점검하고 확정한 후 저장소에 저장하는 작업.

- Branch : 분기점을 의미한다. 현재 상태를 복사하여 Branch에서 작업을 한 후에 작업을 확정하고 싶을 때 Merge를 하여 저장할 수 있다.

- Merge : 다른 Branch의 내용을 가져와 현재 Branch에 합치는 작업

- Head : 현재 작업 중인 Branch를 말한다.

**5. Git을 사용하기 전에 알아야할 기초 용어**

* Git init : 버전을 관리하고 싶은 폴더에서 초기화를 하는 작업
* Git Branch

- 독립적인 공간을 만드는 작업. 특정 작업을 수행하였을 때 그것이 성공했다면 Master Branch에 합치는(Merge) 작업을 한다. 실패하였다면 실패한 Branch를 삭제하면 된다.

* CheckOut : 독립된 작업 공간인 Branch를 자유롭게 이동할 수 있게 한다.
* Git Commit : 수정 작업이 끝났을 때 마침을 알리는 작업을 말한다.
* pull : 원격 저장소의 변경된 내용을 로컬 저장소에 적용하는 작업
* Master(Master Branch) : Git init을 했을 때 default로 만들어지는 Branch.