**Git 이란?**

한 줄로 정리하자면 Git은 버전 관리 시스템이고, GitHub는 Git으로 관리하는 프로젝트를 올려둘 수 있는 사이트이다. Git은 데이터를 저장할 공간만 있다면 어디서나 사용할 수 있다. 이러한 언제 어디서든 버전 관리를 할 수 있다는 장점 때문에 전 세계의 수많은 개발자가 Git으로 협업하는 이유이다.

**GitHub 란?**

사용자에게 무료로 계정과 저장소를 제공하며, 분산형 버전 관리 서비스로 서버 장애 시 데이터 복원력이 뛰어나다. 여기서 버전 관리란 프로그램의 소스코드, 문서, 그래픽 및 관련 파일들을 대단위 소프트웨어로 관리하는 것을 말한다.

**분산형 버전 관리 시스템을 사용하는 이유**

분산형 버전 관리 시스템 (DVCS)은 중앙 집중 방식(CVCS)의 문제를 해결한다. 중앙 집중 방식의 버전 관리 시스템은 중앙 서버에 모든 파일의 버전을 저장 및 관리하고 다수 클라이언트가 서버에서 파일들을 가져와서 사용하는 방식으로 서버에 장애가 발생하면 사용할 수 없을 뿐 아니라 데이터 복구도 할 수 없다. 이와 달리 DVCS에서는 클라이언트가 저장소를 통째로 복제하여 사용하기 때문에 서버에 문제가 발생해도 클라이언트는 복제된 저장소를 다시 서버에 복사하여 서버 내 데이터를 복원할 수 있다.

**Git을 사용하는 또 다른 GitHub?**

그것은 다름아닌 GitLab이다. 소스코드 저장소 호스팅의 양대산맥인 GitHub 와 GitLab은 그 기술적 기반을 Git으로 삼는다는 점에서 큰 차이를 보이지 않는다. 기본 기능은 같지만 서비스를 만든 철학과 초점에 차이를 갖기 때문에, 사용자는 목적과 업무 체계를 고려해 적합한 서비스를 골라야 한다. 젯브레인스에 의하면, 개발자 77%가 GitHub 를 정기적으로 사용한다. GitLab의 경우 40% 정도 사용한다.

**GitHub 와 GitLab 뭐가 다를까?**

GitHub 와 GitLab의 가장 큰 차이점은 데브옵스 요소에 있다. GitLab은 지속적 통합/지속적 전달(CI/CD)와 데브옵스 워크플로우를 내장했다. 반면, GitHub 를 사용하면 사용자가 원하는 CI/CD 도구를 직접 통합해야 한다. 또 다른 차이점은 철학이다. GitHub 가 속도를 우선시 한다면, GitLab은 안정성에 중점을 둔다.

**GitHub의 장점**

개발자의 77%가 사용하는 GitHub. 사람들은 왜 GitHub를 사용할까? GitHub는 버전 관리 시스템을 사용하기 때문에 롤백하는 것에 용이하고, 코드 검토 및 이슈 추적과 같은 기능을 통해 개발자가 프로젝트에 쉽게 협업할 수 있게 한다. 또한 개발자들이 메인 코드 베이스에 병합되기 전에 코드 변경 사항을 검토하고 팀과 논의하여 잠재적인 버그를 방지할 수 있도록 한다. 이외에도 오픈 소스 프로젝트, 커뮤니티, 통합, 보안 등과 같은 이유가 있다.

**GitHub의 단점**

GitHub의 단점은 무엇이 있을까? GitHub의 무료 요금제는 1GB의 스토리지만 포함되어 있어 대규모 프로젝트나 팀에는 충분하지 않을 수 있고, 다운타임 또는 운영 중단의 위험이 있으며 이로 인해 작업이 중단되고 지연이 발생할 수 있다. 또한 GitHub는 보안을 심각하게 생각하지만, 과거에 일부 중요한 보안 이슈가 있었다.

<출처>

**GitHub란?**

**-> 한빛출판네트워크** [[Git, GitHub] 깃, 깃허브란? - 한빛출판네트워크 (hanbit.co.kr)](https://www.hanbit.co.kr/channel/category/category_view.html?cms_code=CMS2036561776)

**GitHub 와 GitLab 뭐가 다를까?**

**-> ZDNET Korea** [깃허브 vs. 깃랩, 무엇을 써야할까 - ZDNet korea](https://zdnet.co.kr/view/?no=20220804135812)

**GitHub의 장단점**

**-> 대두코기** [[GitHub] 깃허브란? 깃허브의 장점과 단점 :: 대두코기 (tistory.com)](https://hoohaha.tistory.com/104)