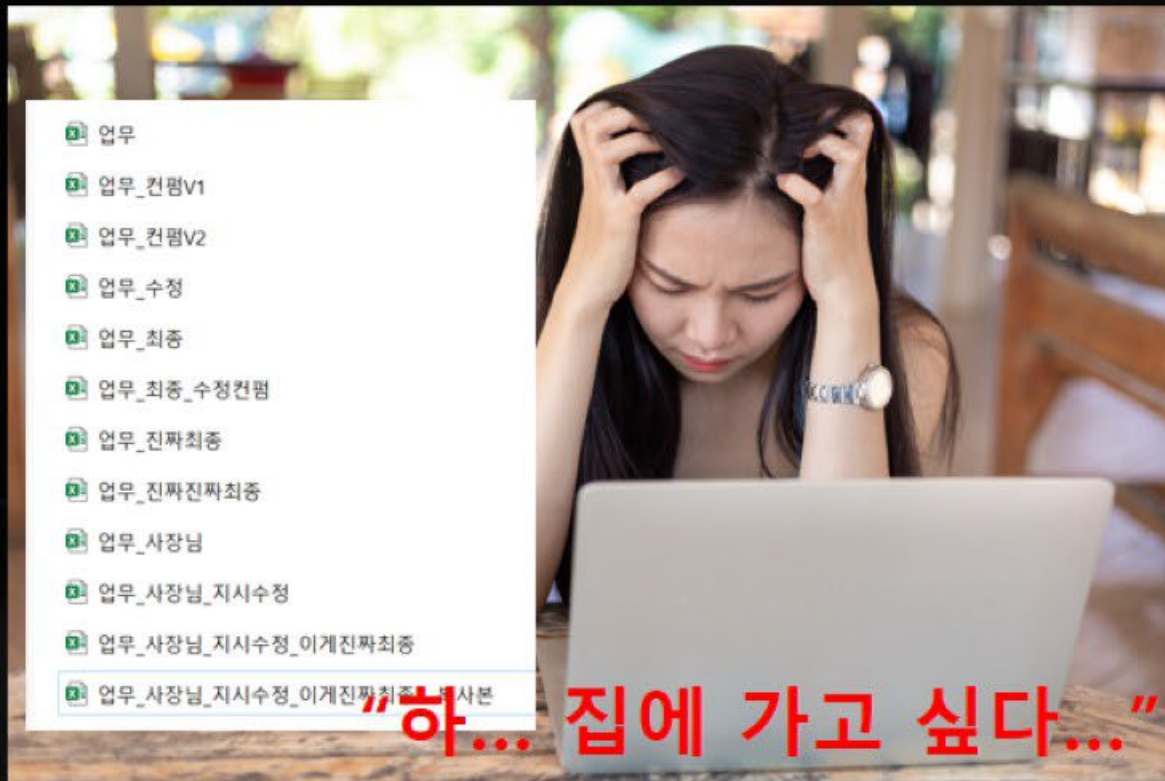


# 클라우드 네이티브에 적합한 형상관리(GIT)

# 형상관리란?

- 형상관리는 변경사항을 체계적으로 추적, 통제하는 것을 말합니다.

[ 형상관리가 없을 때 ]



어떤 버전이 진짜 최종인지 알 수 없음

[ 형상관리가 있을 때 ]



모든 변경사항이 기록되고, 원하는 버전으로 롤백가능

# 형상관리 솔루션의 종류

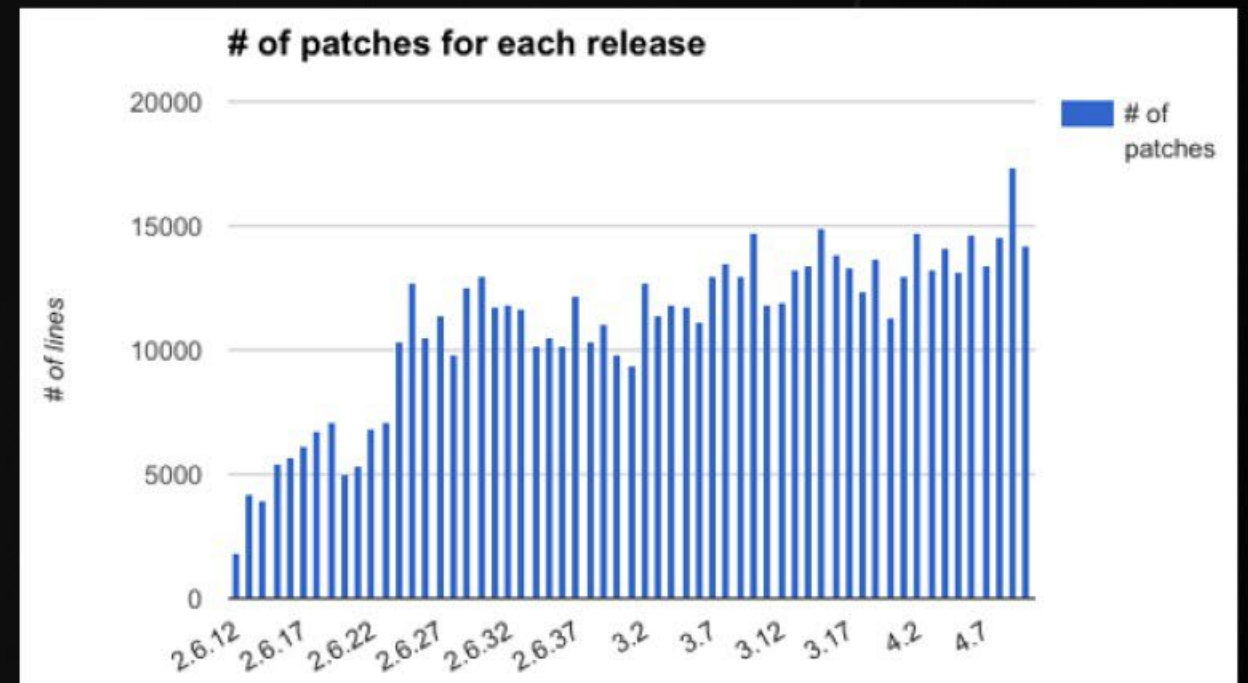
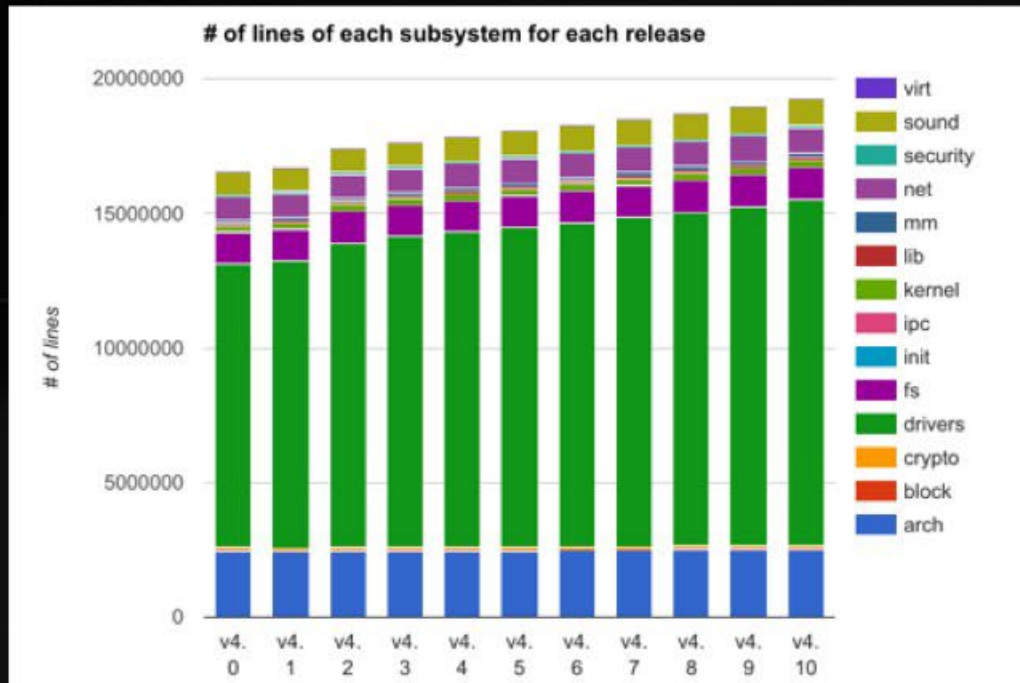


항목	GIT	Subversion	CVS
관리방식	분산형 버전관리	중앙집중형	중앙집중형
충돌가능성	Branch와 Merge를 통해 충돌 가능성 낮음	동시 업로드시 충돌 가능성 많음	동시 업로드시 충돌 가능성 많음
속도	빠름 작업은 로컬에서 업로드만 네트워크 이용	느림 모든 작업이 네트워크를 사용	느림 모든 작업이 네트워크를 사용
네트워크	옵션	필수	필수
최초개발	2005년	2000년대 초	1990년대 초



# 리눅스 커널의 소스

- 리눅스는 1991년 리누스 토발스가 개발함
- 현재 퍼블릭 클라우드 서버의 90%, 스마트폰 82%, 임베디드 기기 66%, 슈퍼컴퓨터 99%가 리눅스 사용 중
- 커널 소스는 4.10 버전의 경우 2천만라인의 코드로 약 14,000명의 프로그래머가 참여하고 있음
- 패치 버전마다 약 15,000라인이 변경됨

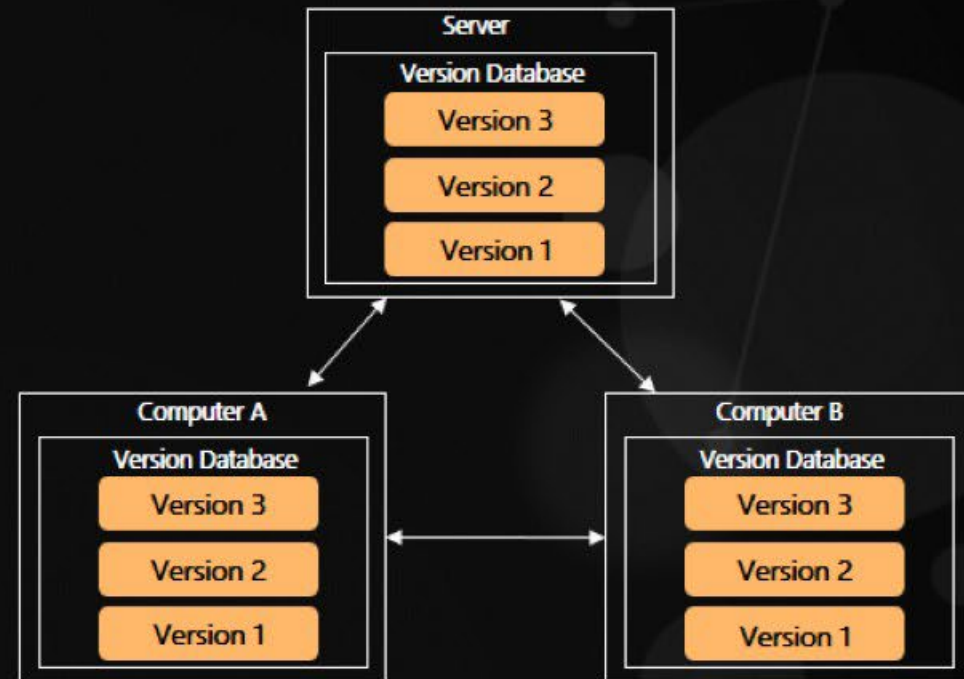


# GIT은 누가 왜 만들었는가?

- 리눅스 커널은 가장 규모가 큰 오픈소스 프로젝트 중의 하나
- 1991~2002년 까지는 patch와 단순 압축 파일로 소스 관리
- 분산 버전관리시스템 BitKeeper를 사용하다가 문제가 발생함
- 리눅스 커널을 관리하기 위한 버전관리 시스템으로 2005년 리누스 토발즈가 2주만에 개발
- 버전관리 시스템의 사실상의 표준

## 주요 특징과 목표

- 빠른 속도
- 단순한 구조
- 비선형적 개발(전세계 개발자 동시개발)
- 완벽한 분산
- 대형 프로젝트에서도 유용할 것

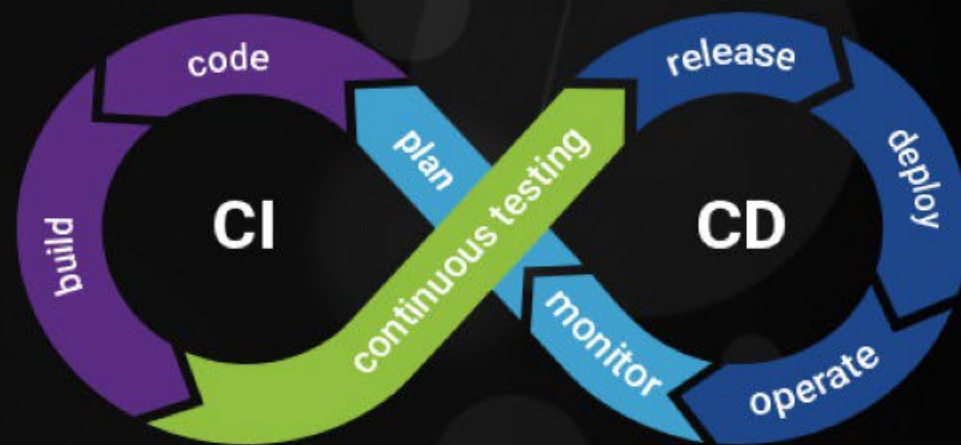
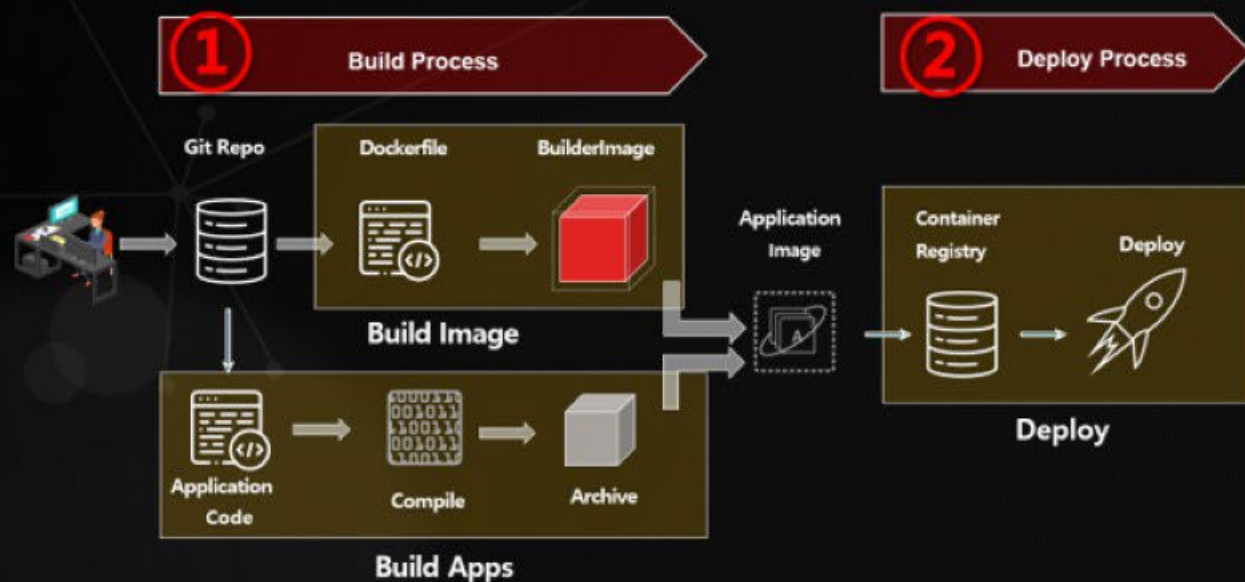


분산형 버전관리 시스템 GIT



# 클라우드 네이티브의 DevOps(CI/CD) 기본 환경

- 클라우드 네이티브 환경에서 DevOps를 구축하기 위해서는 CI/CD 환경이 필수
- CI/CD(Continuous Integration / Continuous Delivery)는 애플리케이션 개발단계를 자동화하여 더욱 짧은 주기로 고객에게 개선된 서비스를 제공하는 방법
- 지속적인 통합, 지속적인 배포가 이루어지기 위해서는 소스 형상관리의 소스로 부터 시작
- CI/CD 도구들은 대부분의 오픈소스 프로젝트가 사용하는 Git 저장소와 연결을 자동화
- 클라우드 네이티브 환경의 CI/CD 절차
  - GIT 소스 → 빌드 → 컨테이너 이미지 생성 → 클라우드 네이티브 환경에 배포



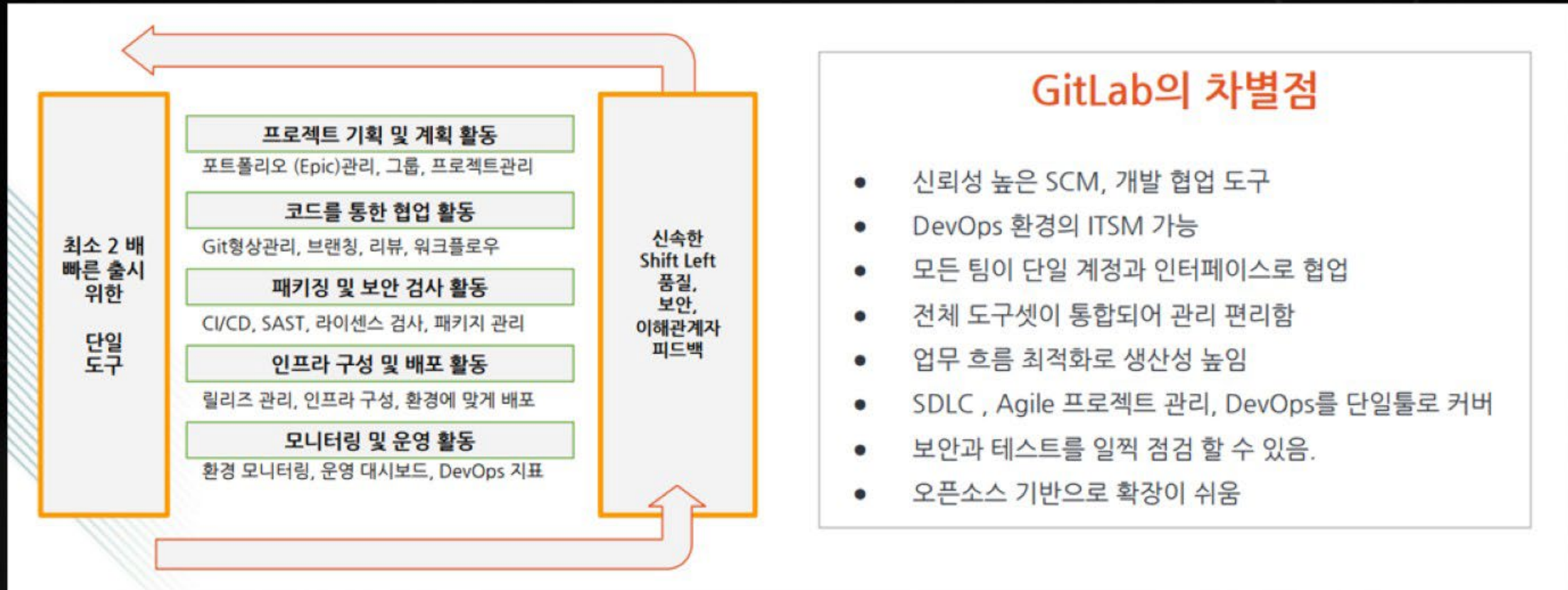
# Git vs Github vs Gitlab



항목	GIT	Github	GitLab
주요특징	버전관리 소프트웨어	Git 서비스(SaaS) 플랫폼	Github과 유사한 설치형 서비스 플랫폼
목적	분산 버전관리 시스템으로 프로젝트 소스를 효과적으로 관리하기 위한 도구	Git 저장소를 호스팅하고 온라인에서 소스 코드를 관리하기 위한 서비스 플랫폼	설치형으로 사용할 수 있는 Git 저장소 호스팅, 관리 플랫폼(Github 과 유사)
특징	<ul style="list-style-type: none"> <li>로컬에서 작업하고 변경사항을 추적하기 위해서 사용됨. 네트워크 연결이 필요치 않음</li> </ul>	<ul style="list-style-type: none"> <li>웹기반 인터페이스를 통해 저장소 관리</li> <li>이슈 추적, 협업 기능 제공</li> <li>Pull Request 및 코드 리뷰 등 협업지원</li> <li>Github Action CI/CD 자동화 제공</li> <li>오픈소스는 무료 / 저장소 별 유료</li> </ul>	<ul style="list-style-type: none"> <li>웹기반 인터페이스를 통해 Git 저장소 관리</li> <li>이슈트래킹, 코드리뷰, CI/CD, 협업도구, 프로젝트 관리 기능을 제공</li> <li>GitLab Runner로 CI/CD 자동화 제공</li> <li>오픈소스, 설치형, 서비스형 모두 제공</li> </ul>
기술지원	<ul style="list-style-type: none"> <li>없음(오픈소스)</li> </ul>	<ul style="list-style-type: none"> <li>유료 서비스</li> </ul>	<ul style="list-style-type: none"> <li>설치형에 대한 Enterprise 기술지원 가능</li> </ul>



- 개발(Dev)에서 운영(Ops)에서 고객까지 기술 가치 흐름 협업을 단일툴로 제공!
- SCM, 프로젝트 관리, 지속적인 탐색(CE), 통합(CI), 테스트 및 배포(CD)를 빠르고 고품질로 제공함!



## GitLab의 차별점

- 신뢰성 높은 SCM, 개발 협업 도구
- DevOps 환경의 ITSM 가능
- 모든 팀이 단일 계정과 인터페이스로 협업
- 전체 도구셋이 통합되어 관리 편리함
- 업무 흐름 최적화로 생산성 높임
- SDLC , Agile 프로젝트 관리, DevOps를 단일툴로 커버
- 보안과 테스트를 일찍 점검 할 수 있음.
- 오픈소스 기반으로 확장이 쉬움





openmaru