

변경 저장하기

- [.gitignore](#)
- [git add](#)
- [git commit](#)
- [git diff](#)
- [git stash](#)

'체크인'은 중앙 서버로 원격 푸시하는 작업입니다. 이것은 SVN 커밋이 프로젝트 변경 사항을 완전히 '저장'하기 위해 인터넷 액세스가 필요하다는 것을 의미합니다. Git 커밋은 로컬에서 캡처 및 빌드 한 다음 필요에 따라 `git push -u origin master` 명령을 사용하여 원격 서버에 푸시 할 수 있습니다. 두 가지 방법의 차이점은 아키텍처 설계 간의 근본적인 차이입니다. Git은 분산 애플리케이션 모델이지만 SVN은 중앙 집중식 모델입니다. 분산 응용 프로그램은 중앙 집중식 서버와 같이 단일 지점에서 장애가 발생하지 않으므로 일반적으로 더 강력합니다.

`git add`, `git status`, `git commit` 명령은 모두 Git 프로젝트의 현재 상태의 스냅 샷을 저장하는 데 사용됩니다.

Git은 'stash'라는 추가 저장 메커니즘을 가지고 있다. stash는 커밋 할 준비가되지 않은 변경 사항을 임시로 저장하는 영역입니다. stash는 세 개의 트리 중 첫 번째 트리 인 작업 디렉토리에서 작동하며 광범위한 사용 옵션이 있습니다. 자세한 내용은 `git stash` 페이지를 참조하십시오.

Git 저장소는 특정 파일이나 디렉토리를 무시하도록 설정할 수 있습니다. 이것은 자식이 무시한 내용의 변경 사항을 저장하지 못하도록합니다. 힘내는 무시 목록을 관리하는 여러 가지 설정 방법을 가지고있다. Git ignore configure에 대해서는 `git ignore` 페이지에서 자세히 설명합니다.

Git 가이드

1. Git 시작하기

- [Git 저장소](#)
 - [git init](#)
 - [git clone](#)
 - [git config](#)
- [변경 저장하기](#)
 - [git add](#)
 - [git commit](#)
 - [git diff](#)
 - [git stash](#)
 - [.gitignore](#)
- [저장소 점검하기](#)
 - [git status](#)
 - [git log](#)
 - [git tag](#)
 - [git blame](#)
- [변경 취소하기](#)
 - [git 실행 취소](#)
 - [git clean](#)
 - [git revert](#)
 - [git reset](#)
 - [git rm](#)
- [Rewriting history](#)
 - [git commit --amend](#)
 - [git rebase](#)
 - [git reflog](#)

2. Git 협업하기

- [동기화하기](#)
 - [git remote](#)
 - [git fetch](#)
 - [git pull](#)
 - [git push](#)
- [브랜치 사용하기](#)
 - [git branch](#)
 - [git checkout](#)
 - [git merge](#)
 - [병합 충돌 해결하기 \(Merge conflicts\)](#)
 - [병합 전략 \(Merge strategies\)](#)
- [Pull request 만들기](#)