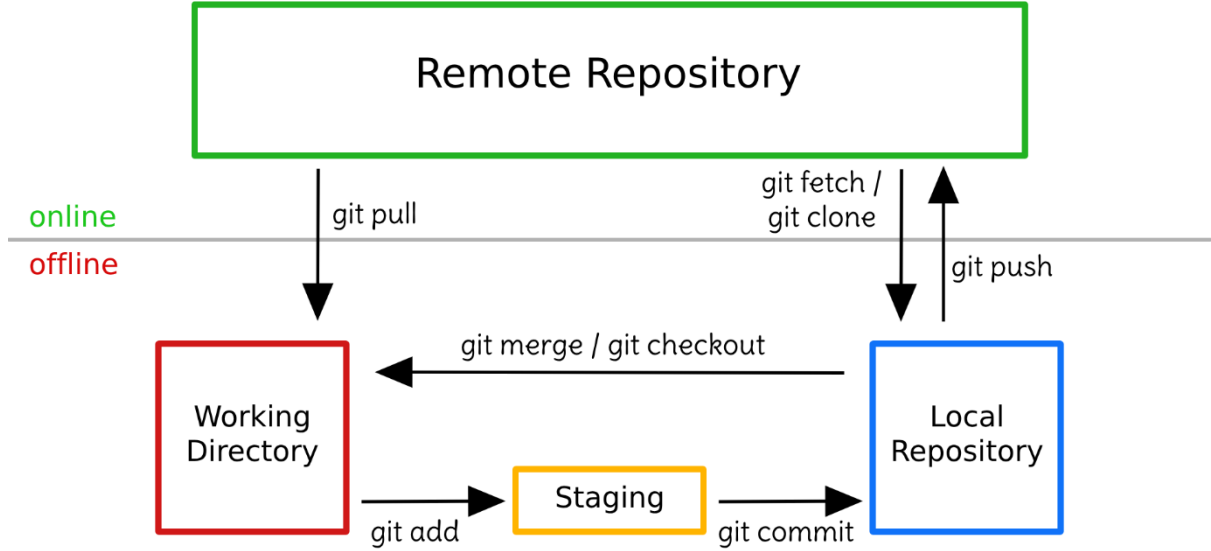


# GIT MANUAL



## Important Notes

- Remote Repository'ye erişebilmek için, <https://git-scm.com/> linki üzerinden **git** indirmeniz gerekiyor.
- Kurulum esnasında, **command line** ve **Vim** seçeneklerini seçmeniz gerekiyor.
- Komutlar, **büyük** ve **küçük** harflere duyarlı.
- Komut sonunda **nokta** kullanımı **bütün** dosyaları ifade eder.
- Tekli dosya belirtmek için sadece **<file name>** olarak belirtmek gerekir.
- **.gitignore** projenin başında oluşturulmalı.
- Dev branchi, development dalı olarak görülür ve test amaçlı kullanılır
- Master branch, ana dal olarak görülür ve test edilmiş kodlar içindir.
- **Readme** dosyası Remote Repository kısmında oluşturabilir ya da Local Repository kısmında oluşturulup git push komutu ile Remote Repository'ye gönderilebilir.
- Readme dosyası için web üzerinde github readme editor araçları kullanılabilirç

## Commands

- **git config** komutu ile projenin sahibinin ismi ve mail adresi belirleniyor.
  - `git config --global user.name "<name>"`
  - `git config --global user.email "<mail>"`
- **pwd** komutu şu anda bulunduğunuz dosya yolunu gösterir.
- **mkdir <folder name>** komutu, bulunduğunuz yolda, belirttiğiniz isimde klasör oluşturur.
  - `mkdir folder`

- ***touch <file name>*** komutu, bulunduğunuz yolda, belirttiğiniz isimde ve uzantıda bir dosya oluşturur.
  - *touch test.txt* ya da *touch test.html*
- ***cd..*** komutu, bir üst dosya yoluna geçiş yapar yani geri gider.
- ***cd <folder name>*** komutu, bulunduğunuz dosya yolunun içindeki klasörlere geçiş yapmanızı yani ileri gitmenizi sağlar.
- ***ls*** komutu, bulunduğunuz dosya yolundaki dosyaları ve klasörleri gösterir.
- ***ls -a*** komutu, bulunduğunuz dosya yolundaki gizli dosyaları dahil bütün dosya ve klasörleri gösterir.
- ***git init*** komutu, .git dosyasını projemize ekler ve projeyi başlatır.
- ***git status*** komutu, proje dosyasındaki değişiklikleri gördüğümüz ama henüz git commit komutu ile Local Repository'e çekmediğimiz yerdir.
- ***git add <file name>*** komutu, belirtilen dosyayı Staging kısmına gönderir.
- ***git add .*** komutu, klasördeki bütün dosyaları Staging kısmına gönderir.
- ***git commit -m "<message>"*** komutu, Staging kısmındaki dosyaları Local Repository kısmına gönderir. Mesaj kısmı, herkes tarafından anlaşılabilir olmalı.
  - *git commit -m "<file.txt changed>"*
- ***git log*** komutu, yapmış olduğumuz değişiklikleri, işlemleri gösterir.
- ***git log --oneline*** komutu, yapılan işlemleri daha sade ve estetik bir şekilde gösterir fakat detayları göremezsiniz.
- ***git checkout <commit id>*** komutu, proje dosyasından silinen bir dosyayı geri getirmek için kullanılır. git log komutu sayesinde geri getirmek istediğin dosyanın commit id'sini bulabilirsin. Bu sayede silinen dosyayı tekrar getirebilirsin ancak dosyayı geri getirdikten sonra dosya için *git status* ve *git commit* komutlarını tekrar yazman gerekli.
- ***git reset HEAD*** komutu, Staging kısmındayken, Working Directory kısmına getirmek için kullanılır.
- ***git checkout <file name>*** komutu, belirtilen dosya içindeki silinen bilgileri geri getirmek için kullanılır.
- ***git diff*** komutu, içerisinde değişiklik yapılmış olan dosyaları, yapılan değişikliklerle beraber gösterir.
- ***git diff <file name>*** komutu, sadece seçili olan dosyadaki yapılmış olan değişiklikleri, yapılan değişikliklerle beraber gösterir.

- ***git rm <file name>*** komutu, řu anda bulunduęun dosya yolundaki, belirtilen dosyayı siler.
- ***git rm -r <folder name>/*** komutu, řu anda bulunduęun dosya yolundaki, belirtilen klasör içindeki dosyalar ile beraber siler.
- ***git mv <current file name> <new file name>*** komutu, mevcutta bulunan dosya ismini, uzantısı aynı kalacak řekilde ismini deęiřtirir.
- ***git mv <file name> <folder name>*** komutu, belirtilen dosyayı, belirtilen klasöre taşıır.
- ***git mv <current file name> <folder name> / <new file name>*** komutu, belirtilen dosyayı, belirtilen klasöre taşıır ve taşıma işlemi sırasında dosyanın ismini deęiřtirir.
- ***git config --global alias.<kısaltma> <kısaltmak istenen komut>*** komutu, mevcutta bulunan komutları, kiřinin kendi isteęine göre kısaltmasını saęlar.
- ***git remote add origin <remote repository link(https)>*** komutu, projeye baęlanmanızı saęlar.
- ***git remote*** komutu, baęlı olduęunuz klasörü gösterir.
- ***git push -u origin <branch name>*** komutu, Local Repository kısmında bulunun dosyaları Remote Repository'e göndermek için kullanılır.
- ***git pull origin <branch name>*** komutu, dosyaları Remote Repository kısmından, Working Directory kısmına çekmenizi saęlar.
- ***git ignore*** komutu, göz ardı edip kullanamayacaęımız dosyalar için kullanılır. Göz ardı edeceęimiz dosyaları, projenin içinde bulunan ".gitignore" dosyası içinde belirtmemiz gerekli.
  - ***<folder name>/\**** komutu, hepsini dahil eder.
  - ***!<folder name>/<harię tutmak istenilen dosya>*** komutu, klasör içindeki belirtilen dosya harię bütün klasörü göz ardı eder.
- ***git branch*** komutu, baęlı olduęunuz dalı gösterir.
- ***git branch --all*** komutu, baęlı olduęunuz Local Repository ve Remote Repository dallarını gösterir.
- ***git fetch*** komutu, Remote Repository kısmında oluřturduęunuz dalları Local Repository'e aktarmak için kullanılır
- ***git branch dev*** komutu, git fetch ile Locale Repository'e aktardıęımız dalları ayrıca Local Repository için tekrar oluřturmak için gerektięi için kullanılır.
- ***git checkout <branch name>*** komutu, Local Repository kısmında dal deęiřtirmek için kullanılır

- ***git merge <branch name>*** komutu, Working Directory ve Local Repository arasında dalları eşitler. Sonrasında git push -u origin <branch name> komutu ile Remote Repository'e aktarılır. Aynı işlem Remote Repository kısmında pull request ve merge butonları ile yapılabilir.
- ***git fetch -p*** komutu, Remote Repository kısmında dalları kontrol eder. Eğer Remote Repository kısmında silinmiş bir dal varsa aynı dalı Local Repository'den siler.
- ***git branch -D <branch name>*** komutu, Local Repository kısmında dal silmek için kullanılır.