

# Git Komutlar

- komutlar

```
git init
```

```
vim .gitignore
```

```
git add file(s)
```

```
git add .      # butun degisen ve yeni dosyalari guncellenecekler listesine ekle
```

```
git add -u     # degisiklik olmus veya silinmis dosyalari guncellenecekler listesine ekle
```

```
git commit -m 'not'  # guncelle
```

```
git diff      # son guncelleme ve mevcut durum arasindaki farklar
```

```
git status    # degisen dosyalar
```

```
git log       # guncelleme gecmisi
```

- örnek

```
mkdir src
```

```
cd src
```

```
touch file1
```

```
touch file2
```

```
touch file3
```

```
git init
```

```
echo file3 > .gitignore
```

```
git status
```

```
git add .
```

```
git status
```

```
git commit -m 'ilk commit notu'
```

```
git status
```

```
git log
```

```
echo 'xxx' >> file1
```

```
git diff HEAD  (veya sadece git diff)
```

```
git add file1
```

```
git commit -m 'ikinci commit notu'
```

```
git log
```

- uzak depo

- ana depoyu oluşturmak

-bare ile oluşturulan depoda doğrudan değişiklik yapılamaz.

cd

mkdir depo

cd depo

git -bare init

- depoyu klonlamak

cd

mkdir yerel

cd yerel

git clone ~/depo proje1 # aynı dosya sisteminde

git ssh://adres:port/~/depo proje1 # ssh ile

- son değişiklikleri ana depodan almak

cd ~/yerel/proje1

git pull

- değişiklik yapıp ana depoya yükleme

cd ~/yerel/proje1

(değişiklikleri yap)

git add .

git commit -m 'degisiklik yapildi'

git pull

# sorun yoksa

git push origin master

- eski sürümlere ulaşmak

git log file1

git show 0323e5918144ec29a069c7ab23cac40ed921d31d:file1

git show HEAD:file1

git show HEAD~2:file1

git tag vo.1 0323e5918144ec29a069c7ab23cac40ed921d31d

git show vo.1:file1

```
git show v0.1:file2
```

```
vim file1
```

```
git reset file1
```

```
rm file1
```

```
git checkout -f file1
```

- Yapılan değişikliği geri almak

- Son değişiklikleri iptal edip HEAD'e dönmek.

```
git reset --hard HEAD
```

- Sadece bir dosyayı, depodaki haline geri döndürmek

```
git checkout -- dosya
```

```
git checkout HEAD dosya
```

- Son yapılan commit'i geri alma.

```
git revert HEAD
```

- branch ile çalışma

- branch oluşturma

```
git branch test
```

- branch listesi

```
git branch
```

- branch değiştirme

```
git checkout test
```

```
git checkout master
```

- branch merge

```
git checkout test
```

```
git commit -a
```

```
git checkout master
```

```
git commit -a
```

```
git merge test
```

- branch silme

```
git branch -d test
```

- depoyu klonlamak (klasör henüz yok, emc2-dev adıyla oluşacak)

```
git clone git://git.linuxcnc.org/git/emc2.git emc2-dev
```

- depodan son değişiklikleri almak

```
git pull --rebase
```

- kendi değişikliklerimizi depoya yüklemek

(yaptıklarımız yerel depoya commit edilecek)

```
git pull --rebase
```

(çakışma varsa düzeltilecek)

```
git push
```

- henüz yayınlanmaya hazır değişikliklerimiz varken depodan güncelleme

```
git stash save
```

```
git pull --rebase
```

```
git stash apply
```

- yerel depo

- komutlar

```
git init
```

```
vim .gitignore
```

```
git add file(s)
```

```
git add .          # bütün değişen ve yeni dosyaları güncellenecekler  
listesine ekle
```

```
git add -u         # değişiklik olmuş veya silinmiş dosyaları  
güncellenecekler listesine ekle
```

```
git commit -m 'not'  # güncelle
```

```
git diff           # son güncelleme ve mevcut durum arasındaki farklar
```

```
git status         # değişen dosyalar
```

```
git log            # güncelleme geçmişi
```

- örnek

```
mkdir src
```

```
cd src
```

```
touch file1
```

```
touch file2
```

```
touch file3
```

```
git init
echo file3 > .gitignore

git status
git add .
git status
git commit -m 'ilk commit notu'
git status
git log

echo 'xxx' >> file1
git diff HEAD      (veya sadece git diff)
git add file1
git commit -m 'ikinci commit notu'
git log
```

- uzak depo

- ana depoyu oluşturmak

--bare ile oluşturulan depoda doğrudan değişiklik yapılamaz.

```
cd
mkdir depo
cd depo
git --bare init
```

- depoyu klonlamak

```
cd
mkdir yerel
cd yerel
git clone ~/depo proje1      # aynı dosya sisteminde
git ssh://adres:port/~/depo proje1  # ssh ile
```

- son değişiklikleri ana depodan almak

```
cd ~/yerel/proje1
git pull
```

- değişiklik yapıp ana depoya yükleme

```
cd ~/yerel/proje1
```

```
(değişiklikleri yap)
git add .
git commit -m 'degisiklik yapildi'
git pull
# sorun yoksa
git push origin master
```

- eski sürümlere ulaşmak

```
git log file1
git show 0323e5918144ec29a069c7ab23cac40ed921d31d:file1
git show HEAD:file1
git show HEAD~2:file1

git tag v0.1 0323e5918144ec29a069c7ab23cac40ed921d31d
git show v0.1:file1
git show v0.1:file2

vim file1
git reset file1
rm file1
git checkout -f file1
```

- Yapılan değişikliği geri almak

- Son değişiklikleri iptal edip HEAD'e dönmek.

```
git reset -- hard HEAD
```

- Sadece bir dosyayı, depodaki haline geri döndürmek

```
git checkout -- dosya
git checkout HEAD dosya
```

- Son yapılan commit'i geri alma.

```
git revert HEAD
```

- branch ile çalışma

- branch oluşturma

```
git branch test
```

- branc listesi

```
git branch

- branch değiştirme
  git checkout test
  git checkout master

- branch merge
  git checkout test
  git commit -a
  git checkout master
  git commit -a
  git merge test

- branch silme
  git branch -d test

- depoyu klonlamak (klasör henüz yok, emc2-dev adıyla oluşacak)
  git clone git://git.linuxcnc.org/git/emc2.git emc2-dev

- depodan son değişiklikleri almak
  git pull --rebase

- kendi değişikliklerimizi depoya yüklemek
  (yaptıklarımız yerel depoya commit edilecek)
  git pull --rebase
  (çakışma varsa düzeltilecek)
  git push

- henüz yayınlanmaya hazır değişikliklerimiz varken depodan güncelleme
  git stash save
  git pull --rebase
  git stash apply
```