

①

GIT

Bir Version Control System (VCS)

Version Control System'in bir tool'u (aracı)

Open-source ve free.

Linus Torvalds olusturdu.

Distributed (Kaynak dosya herkesin bilgisayarına iner.)

Working Directory → Bilgisayarında isinde calis-
tirmaya dayalı.

Repository (Depo) → .git klasörü
Kayıtları tutar

Local
(?internetiz?)

Remote
(?internetli?)



git --version

git config --global user.name "Your name"

git config --global user.email "Your email"

git config --global core.editor "Vim" → Text editör
(Not defteri?)

git config --list
(Metin yazmaya
müsade eder)

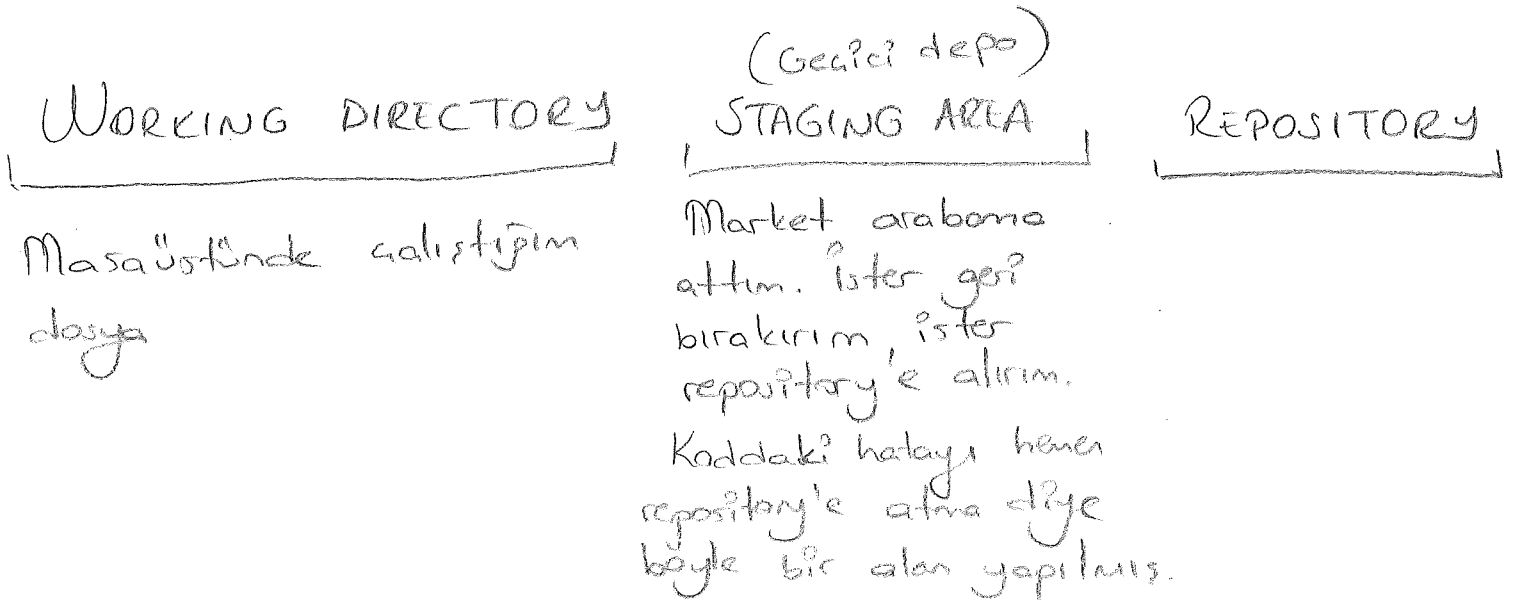
- ! vim yaz metin dosyası gibi bir şey açılacak
ESC'ye bas
Çıkmak için :q! Enter

(*) cd Desktop/ (Masaüstüne gitmek için Eng. yazmak zorundasın)

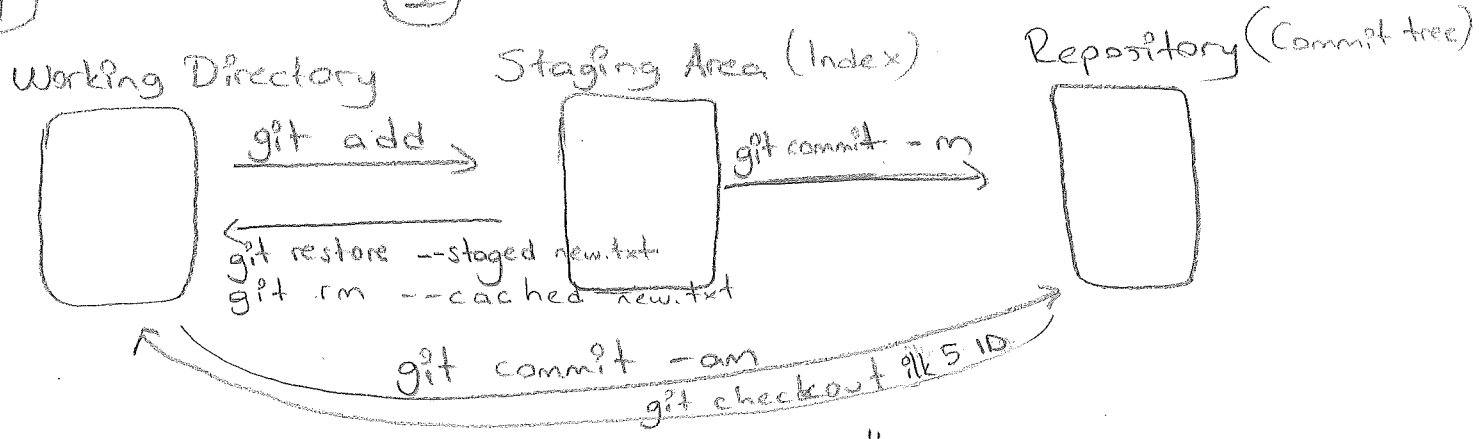
(*) git init → git initialize
Takibi başlat veya dosyada sağ tıklayıp
"git bash here" de.
git dosyası oluşur.
Bu bizim reponuz.
Takip edeceğimiz dosyaların bilgileri burada tutulur.
git görürsen gitile takip edilen dosya demek

(*) git status → Version Control System yapılan dosyanın durumunu gösterir.

(*) git clone adres → Uzaktaki kaynak dosyaları kendi bilgisayarımıza indirmek.



(2)



(*) `git commit -m "first commit"`
(Yaptığım değişikliği kalıcı hale getiririm.)

(*) `git log` \Rightarrow Yaptığım değişiklikleri gösterir.
(Bulduğundan önceki halini gösterir.)

(*) `git checkout` \Rightarrow İleri-geri gitmek (Zamanda yolculuk)
 \downarrow `git log`'da çıkan satırların ilk 5'i yazılır.
Yani 1 ay önce ne yaptım gibi gibi.

(*) `git commit --amend` \Rightarrow Yapılan değişiklikler yazmak için sayfa mı açıyor ne.

(*) `git checkout master` \Rightarrow Bütün commitler görünür.
Master branch'e gider

(*) `git branch` \Rightarrow Branch'leri görünüm

`git branch new-file` \Rightarrow Yeni branch yapı

`git checkout new-file` \Rightarrow Branchler arası geçiş

`git commit -m "master first commit"` \Rightarrow Branch'in üzerine küçük asılları yazıyor.

(*) `git checkout -b new-branch` \Rightarrow Yeni bir branch yapar ve ona gider.

(*) `git branch -a` \Rightarrow Her anaktaki hem bendeki branch'leri göster.

(*) `git checkout [branch name]` \Rightarrow Switch to a branch

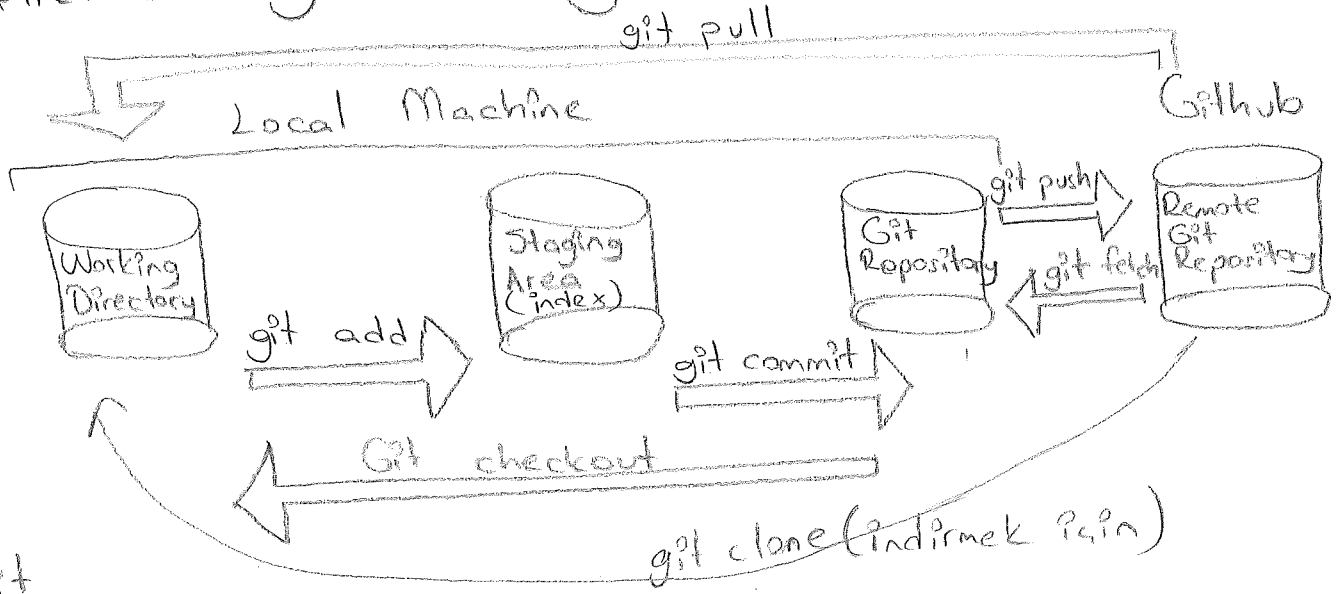
(*) `git branch -r` \Rightarrow Uaktaki branch'leri göster.

(*) git branch → new-branch ⇒ Branch siler

(*) git merge new ⇒ fast forward merge
Master ve head? başka yere götürür.
Yani senin master'in artık değişir.

MERGE CONFLICTS

Bir dokümanın üzerinde birden fazla kişi çalışma yaptı.
Hangi değişikliğin kabul edileceğine proje yöneticisi karar verir.
Bu conflictlerin giderilmesi gerekir.



git init

git status ⇒ Working dir. ile Staging area arasındaki farkı gösterir.

git add.

git rm --cached ⇒ Stagingden → Working'e geri alır.

git commit -m "abc" ⇒ Yaptığın değişiklikleri yaz

git log

git checkout commitID ⇒ Commitler arası geçiş

git commit -am "abc" ⇒ ekle ve mesaj yaz aynı seye
add
staging
areaya ekle
message

(*) git branch \Rightarrow Branch ③

(*) git branch yeni.txt \Rightarrow yeni branch

(*) git checkout yeni.txt \Rightarrow Branchler arası geçiş

(*) git branch -r \Rightarrow Uzaktaki branchler görünür.

(*) git branch -a \Rightarrow Hem uzaktaki + hem bizdeki branchler görünür.

(*) git checkout -b yeni.txt \Rightarrow Branch yap + Ona geç
 \hookrightarrow İkisi bir arada

(*) git branch -d yeni.txt } Branch siler.
git branch -D yeni.txt } (Ama önce git checkout la
başka yere git. Üzerindeki
dalı silmez.)

(*) Aynı satırda farklı değişiklik yapılırsa CONFLICT olur. Bu conflict'in giderilmesi gerekir.

vi new-file.txt yazdım. Sonra i'ye bastım. Sayfada değişikliği yaptım. ESC'ye bastım. :wq koydum. ENTER Yönetici olarak farklı kişilerin aynı satırda yaptığı değişiklikleri ben düzenledim.

! Origin kelimesini görürsen "github'da bulunan repo" demek.

(*) Uzaktaki dosyayı kendi repoma indirme \Rightarrow Klonlama

GITHUB'dan kendi repoma dosya indirme

Git bash'i aç

Bilgisayarda hangi dosyanın içine kaydetmek istiyorsan
cd komutu ile oraya git.

git clone web arder kopyala

origin → Uzakdaki repomuz

touch hello-world.py

git status

git add.

git status (Üstöring areada commit edilmeği bekliyor)

git commit -m "hello-world.py dosyası olusturuldu."

echo "merhabo dünya" > hello-world.py (1.commit)

git status

git add.

git commit -m "hello-world.py dosyası güncellendi." (2.commit)

git status

git log (Commitimiz 2 commit önce orijinden)

git push (Localimizdeki github'a gönderiyoruz)

git clone link ⇒ Github'dan locale almak için

git fetch ⇒ Uzak repodaki dosyaları bilgisayara indirir

git log ⇒ git log --oneline

git diff ⇒ github-bilgisayarım arasındaki farkları gösterir.

localdeki github'a göndermek ⇒ git push

(4)

git fetch → Github'dan locale dosya indir.

git diff main origin/main ⇒ Github - local farkı ne?

git merge → Uzakdaki repodaki branch'le birleştirdi.

git fetch + git merge = git pull

! Önce github'da reponu oluşturun.
Sonra localine klon et. Hocanın önerisi

git remote add origin Repo address ⇒ connect to remote repo

git push -u origin master ⇒ first push

git remote rm origin ⇒ remove remote origin

⑤
(*) whoami \Rightarrow Hangi kullanıcıyım?

Sony

(*) hostname \Rightarrow Bilgisayarın ismi
DS-PC

(*) hostname -i \Rightarrow Local IP'yi söyler.

(*) head \Rightarrow İlk 10 satırı gösterir.

(*) tail \Rightarrow Son 10 satırı gösterir.

(*) head -3 \Rightarrow ---.txt \Rightarrow 3 satır yazdırır.

(*) tail -4 \Rightarrow ---.txt \Rightarrow Son 4 satırı yazdırır.

(*) touch \Rightarrow Birder fazla dosya oluşturmak için bunu kullan.

! touch {a..d}{1..20} \rightarrow a, b, c, d için 20'er dosya oluşturur.

(*) rm * \Rightarrow Bütün hepsini siler

rm a*.txt \Rightarrow Sadece içinde a olanları sil.

(*) touch {a..c}{1..20}{x..z} \rightarrow a1x, a1y, a1z, b1x, b1y, b1z, ...

ls a*x \Rightarrow içinde a x olanları listele demek

ls ??? \Rightarrow Sadece 3 karakterliyi çıkarır.

* Bastaki . gizli dosyaları ifade eder.

(*) man ls ls --help

info ls'den daha kapsamlı bir komut

*cat \Rightarrow Dosya içeriği okunur. Dosyanın içindeki yazılar görünür.

! cat yeridebysya.txt (Bu da ikinci görevi.)
Burda içinde yazanlar çıkar. (Echo'ya benzer.)

Cont + d

*echo \Rightarrow Eko yapar. Sana geri dönüş verir.
Yazı ekleyebilir.

! echo "merhaba dünya"
Burda "merhaba dünya" yazar.

echo "safi" > yeridebysya.txt \rightarrow içeriği sildi yerine "safi" koydu.

echo "safi" >> yeridebysya.txt \rightarrow içeriği koruyup altına "safi" yattı.

*touch file1 file {1..10}
 \rightarrow file1 dosyasının yerine 10 dosya yapar.

*ls *f*2* \rightarrow sonuna yıldız koyarsam sonu 2 ile biter.
Çiçinde f ve 2 olanları göster.

ls f?le
Başı f ile başlasın ortası önemli değil son da önemli değil.
Dosyalarda fildirekime için kullanılır.

*ls ?????
5 karakterli olanları getirir.

? \rightarrow Tek bir karakteri yeri tutar.

* \rightarrow Birden fazla karakteri yeri tutar.

6

* explorer.exe \Rightarrow Bilgisayarındaki dosyaları açar.

* pwd \Rightarrow Çalıştığım klasörü göster.

* ls \Rightarrow Listelemek için.
(Basınca nokta olan her klasör girilir.)

* ls -l \Rightarrow Detaylı olarak listeler

* ls -a \Rightarrow Gizliler de dahil listeler

* ls -all \Rightarrow Hem gizli + hem detaylı liste.

* cd \Rightarrow Klasörü değiştirir. (change directory)

! cd klasöradı/ \Rightarrow O klasörün içine gider.

* cd .. \Rightarrow Bir üst klasöre çık.

* cd / \Rightarrow En tepedeki klasöre gider. (Windows'ta C)

* cd - \Rightarrow Bir önceki klasöre git.

* cd ~ \Rightarrow Home klasörü (Home directory'e götürür.)

* mkdir \Rightarrow Dosya oluşturur. (Windows'ta sağ tıkla yeni klasör.)

! mkdir KlasörAdı (Klasör oluşturdu)

* rm dir \Rightarrow Boş dosyayı siler

! rm dir KlasörAdı/ (sildi?)

* touch \Rightarrow Yeni dosya, metin bel... oluşturur. (Win'ta sağ + yeni?)

! touch DosyaAdı
• touch DosyaAdı. sa fi uzantısı

⊗ rm ⇒ Dosyaları siler

! rm DosyaAdı

⊗ cp ⇒ Bir dosyayı kopyala - başka bir dosyanın içine yapıştır.

! touch newfile (Yeni dosya yaptık.)

cp newfile clarusway/

Bu dosyanın içine kopya yaptı.

mkdir linux-lesson → Dosya oluşt

ÖRNEK

cd linux-lesson/ → Dosyanın içine girdik

mkdir folder1 folder2 → Dosya içinde aynı anda iki dosya oluşt

touch folder1/newfile → folder1'in içinde yeni bir dosya oluşt.

(farklı bir yöntem)

cp folder1/newfile folder2 (Folder1'in içine girmeden yaptık)

Aynı! cp folder1/newfile folder2
Bunu al Buna at

cd folder1 → içine gir

cp newfile ../folder2/newfile-copy
Bir üst klasöre çık folder2'ye ismini değiştirerek kopyaladım.

⊗ mv ⇒ Tasıma ve dosya ismi değiştirmek.
✓ Kopya değil tasıma

! mv newfile-copy ../folder2/newfile-moved
bunu kes üst dosyaya ismi değiştirerek tasımak
çık

! mv newfile.txt yeni-dosya.txt
bu ismi buna değiştir

⑦
① * more istiklal.txt ⇒ Yücede olarak metnin neresinde olduğunu

① * less istiklal.txt
→ Çıkmak için q'ya bas

