

Git

Nedir, Ne değildir?

Git bir versiyon kontrol sistemidir.

Kurulum

- Debian tabanlı bir GNU/Linux dağıtımı için:
-
- `$ apt-get install git`

Kimlik ve Editör

- `$ git config --global user.name "Dora Uzunsoy"`
- `$ git config --global user.email dora@aucc.org`
- `# Kullanıcı adı ve e-posta adresi tanımlama`
- `$ git config --global core.editor vim`
- `# Metin editörü ayarlama`
-
- `$ git config --list`
- `# Ayarları listeler`

Repo Edinmek

- \$ git init
- # Sistemdeki bir dizini repoya çevirir.
-
- \$ git clone github.com/github_kullanici_adi/repo-adi
- # Bir Github reposunu bilgisayara klonlar.

Ekleme-Kontrol Etme-Notlama

- `$ git add dosya_adi`
- `# Dosyaların git tarafından takibini başlatır.`
-
- `$ git status`
- `# Yapılan ve commitlenmemiş değişiklikleri gösterir.`
-
- `$ git commit -m „Değişiklikler ile ilgi not.“`
- `# Commitler ve değişiklikler kaydedilir.`

Silme-Yeniden İsimlendirme

- `$ git rm`
- `# silme işlemi yapar.`
- `-f(--force)` `#git silme hakkında hata verirse kullanılır.`
- `--cached` `# git takibi bırakır ama sistemden silinmez.`
-
- `$ git mv eski_isim yeni_isim`
- `# İsim değiştirir.`
-

Kayıt Tarihçesi

- `$ git log # commit geçmişini gösterir.`
-
- `$ git reset --hard master~2 #2 commit geriye götürür.`
-
- `$ git checkout <commit'in ilk 8 hanesi> # O adıma döner ama kayıp olmaz. <$ git checkout kullanılan_branch> diyerek son hale geri dönebiliriz.`
-
- `$ git commit --amend # Son comitte bir eksiklik varsa comitten sonra eksiklik giderilir ve bu komut çalıştırılır. Bu commit kalabalığını engeller.`
-
- `$ git reset HEAD <file> # Commit öncesi add ile eklenen bir dosya üzerinde tekrar değişiklik yapabilmek için kullanılır.`

Dallar

- \$ git branch <branch_adi> # branch oluşturur.
-
- \$ git checkout <branch_adi> # branch'e geçer.
-
- -b (checkout ile kullanılır ve dalı yaratır yaratmaz ona geçer) ve -d(branch ile kullanılır ve dalı siler.)

Birleştirme

- \$ git rebase <branch> # merge aksine ,merge commit' oluşturmaz ve tarihsel bir iz kalmaz.
-
- \$ git merge <branch> # yazılan branch'i kullanılan branch ile birleştirir.

Uzak Repo Dalları

- `$ git remote (-v) # uzak repo adreslerini ve ismini görmek için kullanılır.`
-
- `$ git push origin master # origin reposunun master dalına push işlemi uygular.`
-
- `$ git push origin :repo_adi # repo_adi verilen uzak repoyu siler.`

Önemli olan Github Şeyleri

- Fork → Başkasının projesi üzerinde bağımsız çalışmak için Github tarafından sağlanan bir imkan.
-
- Pull-Request → Birinin projesini onunla beraber geliştirmek için yapılan bir teklif işlemi.
-