



BATCH : BATCH 48
LESSON : GIT
DATE : 21.03.2022
SUBJECT : GIT & GITHUB



techproeducation



techproeducation



techproeducation



techproeducation



techproedu



techproeducation.com



info@techproeducation.com



+1 (917) 768-7466

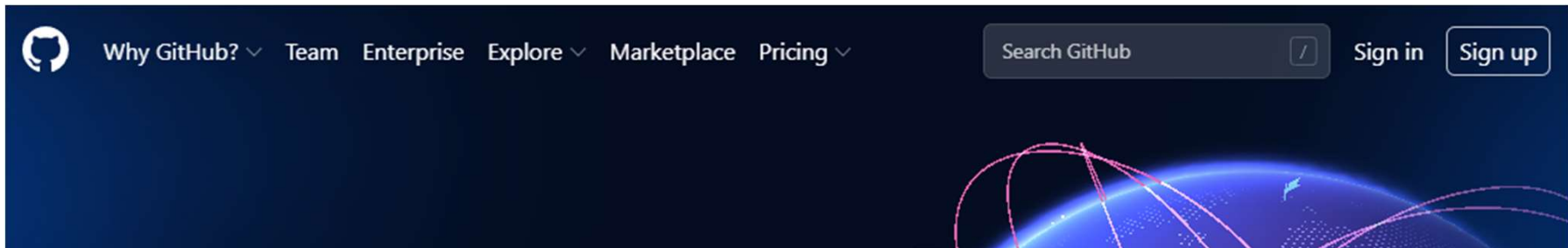


Github hesabı oluşturma



› Github hesabı oluşturma kısaca şudur:

<https://docs.github.com/en/get-started/signing-up-for-github/signing-up-for-a-new-github-account>



Github.com



Github repo oluşturma



Overview Repositories 15 Projects Packages

Find a repository...

Type

Language

Sort

New

1

New butonuna basılır

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner *

techproeducation1

Repository name *

Great repository names are short and memorable. Need inspiration? How about [scaling-meme](#)?

Description (optional)

☒



Public

Anyone on the internet can see this repository. You choose who can commit.

☐



Private

You choose who can see and commit to this repository.

2

Repository için bir isim veriyoruz

3

Herkes tarafından ulaşılabilir mi olsun, yoksa sadece bizim belirlediğimiz kullanıcılar mı ulaşabilsin

4

Create repository butonuna basılır

Create repository



GitHub Lokaldeki Veriyi Remote' a Yüklemek İçin

git init

git add .

git status

git commit -m
"mesaj"

git log

git remote add
origin
[REMOTE_URL]

git push --set-
upstream origin
master



Local Repo'nun Github'a Yükleneceği



- Local 'de en az 1 commit oluşturduktan sonra, **her proje için bir kereye mahsus olmalı** üzere aşağıdaki komutlar kullanılır. Böylece local repo ile Github ilişkilendirilmiş olur.

1

`git remote add origin [REMOTE_URL]`

Remote url, Github sitesi üzerinden ilgili repo sayfasında code bölümünden elde edilir

2

`git push --set-upstream origin master`

Local master branch, remote repo ya gönderilmiş olur.



Git push origin master komutu kullanıldığında eğer daha önceden Github hesabına login olunmamışsa login ekranı açılabilir veya kullanıcı adı şifre/ssh key istenebilir.

Ssh key şuradan oluşturulabilir:
<https://github.com/settings/keys>
Key oluşturulduktan sonra bir yere kaydedilmelidir!!



Local Repo nun Github a Yükleneesi



- › İlk yüklemeden sonra yapılacak yeni local commit'leri Github a göndermek için aşağıdaki komutun kullanılması yeterlidir.

```
git push
```

* Sonrasında shell'de çıkan url'deki branches seçeneğine tıklanarak push kontrol edilir.



Github Çalışma Prensipleri

Main Branch çalışmanın son halidir. Bu yüzden main'deki belgeler her zaman önceden gözden geçirilmiş ve onaylanmış olmalıdır. Bütün kullanıcılara 'main' e push etme yetkisi vermek best practice değildir.

MAIN BRANCH



GitHub Remote'daki Veriyi Lokal'e Indirmek için

```
git clone  
[git url]
```

```
git pull
```




Cloning

Github'daki bir reponun lokale indirilerek geliştirilmesi için öncelikle klonlanmalıdır. Bunun için aşağıdaki komut kullanılır. Yani burada remote'daki repo'yu local'e indiririz.

```
git clone [git url]
```

Directory'miz neresiyse remote'daki proje oraya indirilir.



Github dan commit çekme



- › Github üzerinden bir commit ile **local repo güncellenmek istenirse** aşağıdaki komutlar kullanılır

`git fetch`

Değişiklikleri remote'dan local'e indirir

`git merge`

İndirilen değişiklikleri local repoya uygular

VEYA

`git pull`

fetch & merge



.gitignore

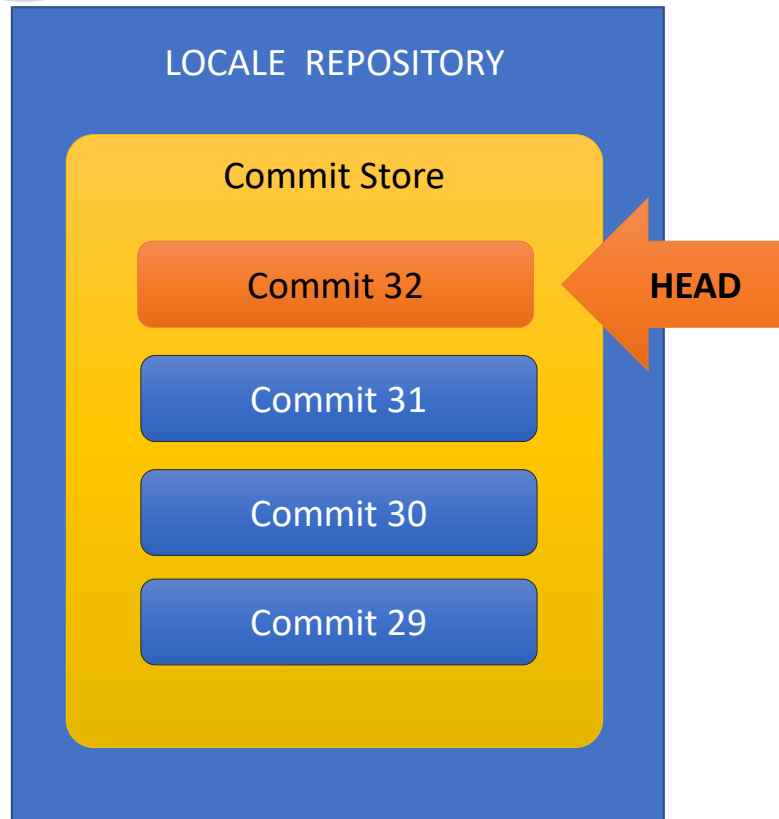
Staging area' ya gitmesini istemediğimiz, yani versiyon kontrol sistemine dahil etmek istemediğimiz dosya ve klasörlerimizi tanımladığımız özel bir dosyadır. Bu dosya .git klasörünün bulunduğu dizinde olmalıdır.

```
out/  
.idea/  
.idea_modules/  
*.iml  
*.ipr  
*.iws
```

.gitignore



Commit Store



- › Bir repo içinde birden fazla commit olabilir. Bunlardan en son alınan commit' e **HEAD** denir.
- › Bu HEAD değiştirildiğinde önceki versiyonlara dönüş yapılabilir.

```
C:\Users\sariz\Desktop\test>git log
commit c417dfe1afa5deac505808a0a2c8ba05afc8e86d (HEAD -> master)
Author: Your Name <you@example.com>
Date: Sat Aug 7 23:49:17 2021 +0300

    1 satır eklendi

commit 5e063d211454b3bc8846bc0720aef4895b1fdbff
Author: Your Name <you@example.com>
Date: Sat Aug 7 23:40:18 2021 +0300

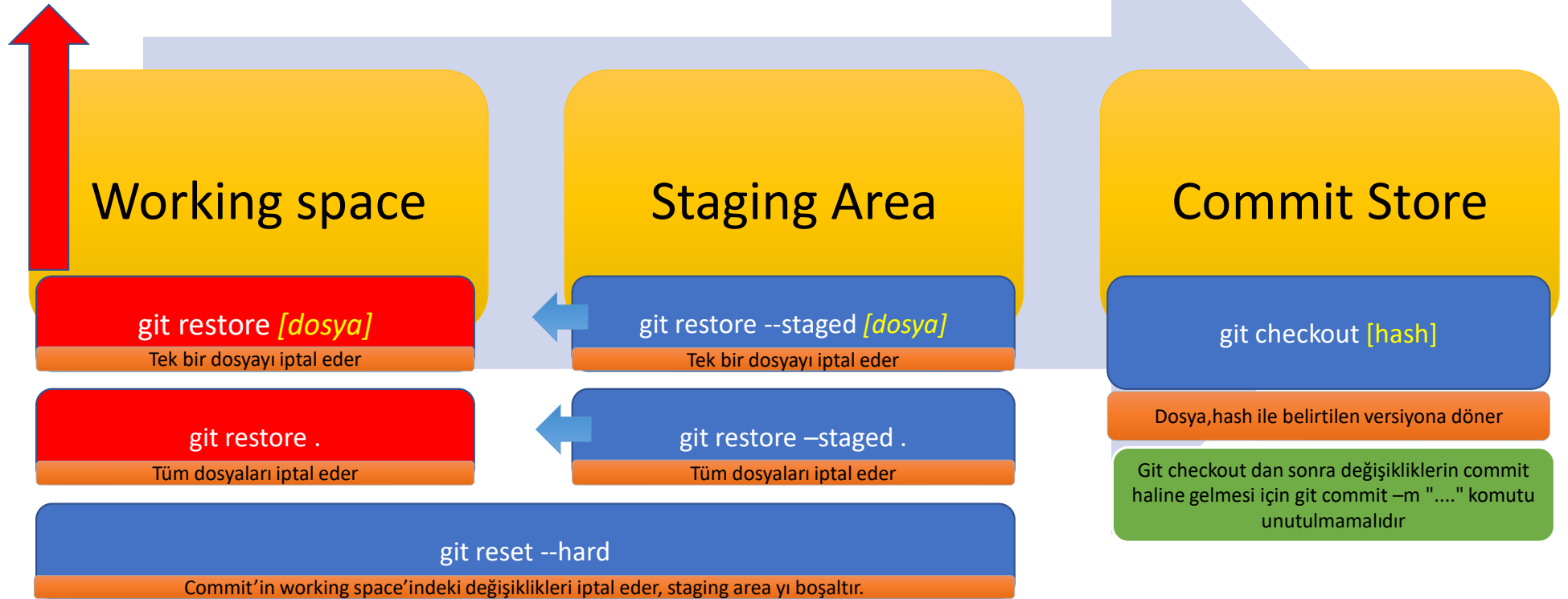
    first commit
```



Değişiklikleri iptal etmek

Hala tamamlanmış bir komut değil, bilin ama kullanmayın:

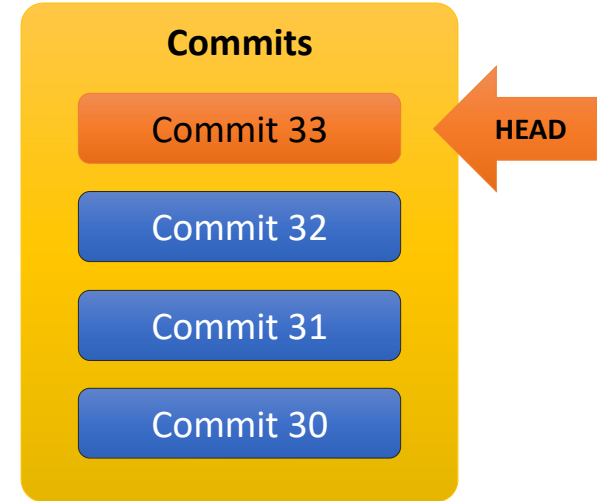
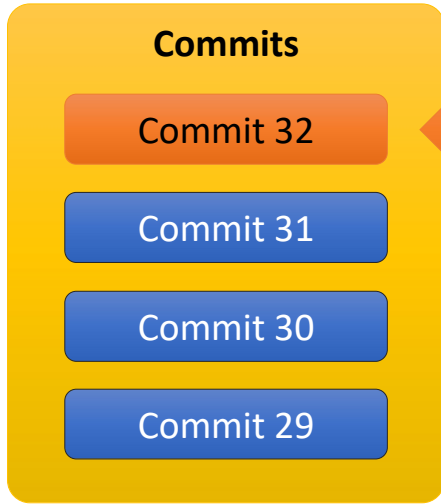
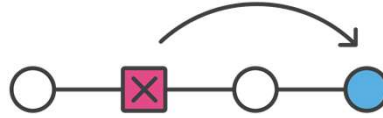
<https://stackoverflow.com/questions/48508799/how-to-reset-all-files-from-working-directory-but-not-from-staging-area/57066072#57066072>





Önceki versiyonlara dönmek

1.Yöntem: CHECKOUT

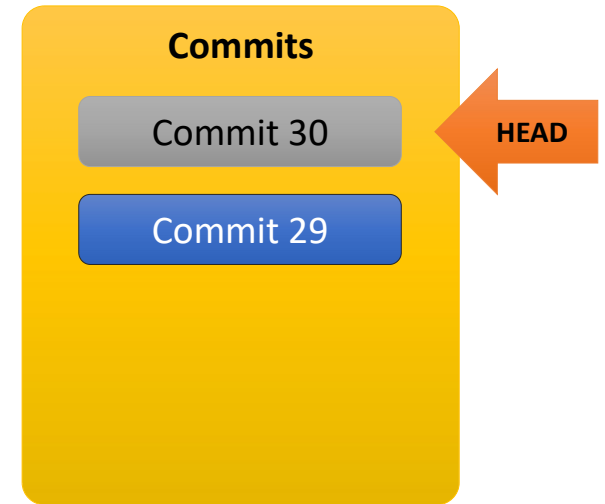
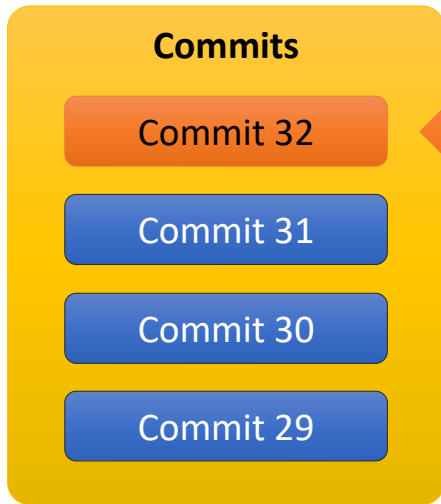


İstenilen versiyonu geri alır. Ancak bunun için sadece HEAD hareket ettirilir. Yapılan değişiklikler silinmez. Bu işlemin de bir versiyon haline gelmesi için commit oluşturmak gerekir.



Önceki versiyonlara dönmek

2.Yöntem: RESET

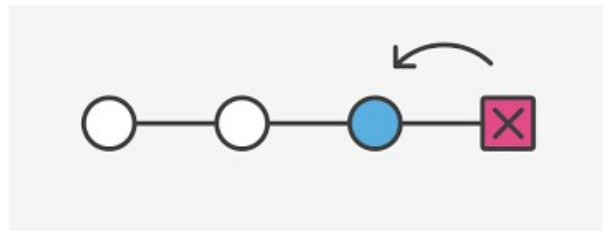


İstenilen versiyona geri döner, bu versiyondan daha sonra yapılan tüm commit ler ve içerdiği değişiklikler geri alınamayacak şekilde iptal edilir.



Git Reflog

- Git te yapılan bütün işlemlerin çıktısıdır.
- git reflog



- git reflog kullanılarak geçmiş işlemlerin izi sürülebilir, gerektiği durumlarda bu işlemler modifiye edilebilir.



Git Reflog

- ▶ Mesela **git reflog + git reset --hard <hashkodun 4 hanesi>** ile hash kodunun 4 hanesi verilen state'e geri dönülebilir.

```
bcb8fb1 (deneme1) HEAD@{4}: checkout: moving from deneme1 to master
bcb8fb1 (deneme1) HEAD@{5}: Branch: renamed refs/heads/deneme2 to refs/heads/deneme1
bcb8fb1 (deneme1) HEAD@{7}: checkout: moving from master to deneme2
bcb8fb1 (deneme1) HEAD@{8}: checkout: moving from deneme1 to master
d86e0ad HEAD@{9}: checkout: moving from d86e0addb245baf67d8d03e877a0f08c3a421344 to deneme1

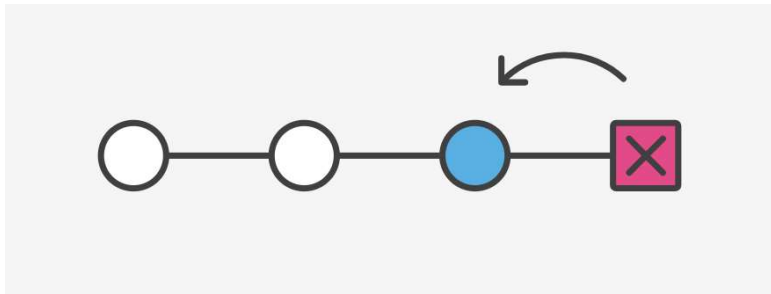
talfi@DESKTOP-CK670UE MINGW64 ~/devops/Git (master)
$ git reset --hard bcb8fb1
HEAD is now at bcb8fb1 Revert "Geri aldık" reverts commit d86e0addb245baf67d8d03e877a0f08c3a421344.

talfi@DESKTOP-CK670UE MINGW64 ~/devops/Git (master)
$
```



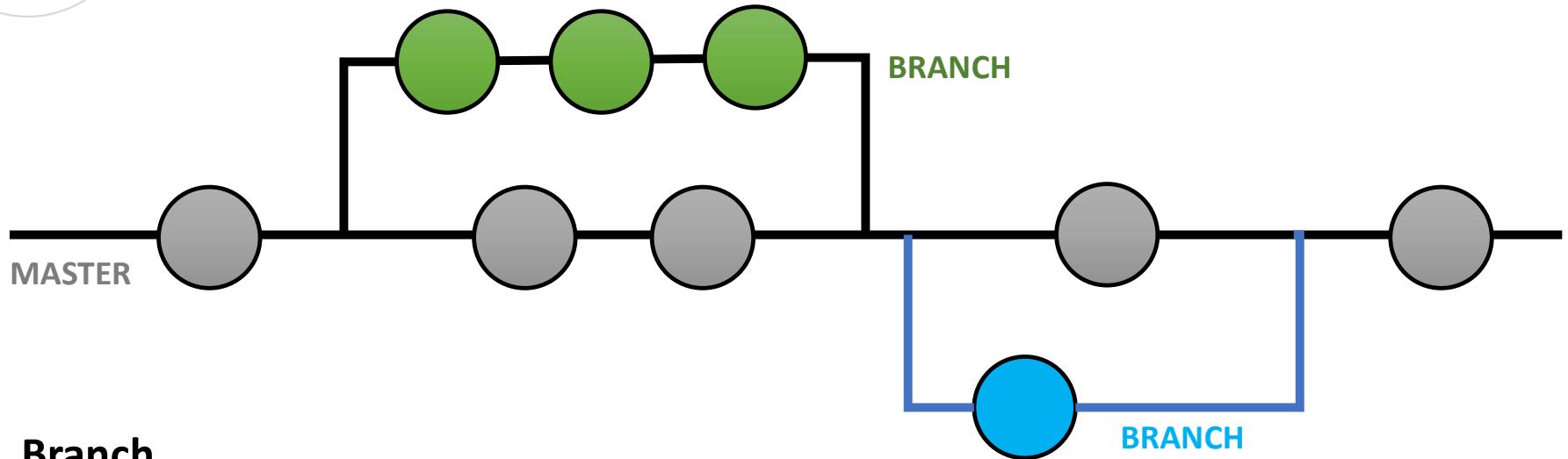
Git Reflog

- **Not!:** **git log** sadece commit'lerle ilgili bilgileri temel bilgileri verirken **git reflog** repo'nun geçmiş durumuyla ilgili kullanıcıyı daha detaylı bilgilendirir.





Branch (Dal)



Branch

Master branch, projemizin ana yapısıdır. Zaman zaman bu ana yapıyı bozmadan bazı denemeler yapmak ve gerekirse kolaylıkla bu denemeleri iptal etmek ya da master ile birleştirmek için branch ler kullanılır. Branch ler içindeki değişiklikler master dan bağımsız olarak saklanır.



Branch Komutları

`git branch [isim]`

Yeni branch oluşturur

`git branch -d [isim]`

Branch i siler

`git branch -m [isim]`

Branch ismini değiştirir
Hangi branch'taysanız onun ismini
değiştirir.

`git branch`

Mevcut branch leri listeler

`git checkout [isim]`

Branch aktif hale gelir
Çalışmak istediğin branch'ı seçersin

`git branch -a`

Local ve remote daki bütün
branch'leri görmek için kullanırız