



BATCH :
LESSON :
DATE :
SUBJECT :

130-131-138

Git-Github

22.02.2023

Git

ZOOM GİRİŞLERİNİZİ LÜTFEN **LMS** SİSTEMİ ÜZERİNDEN YAPINIZ



TECHPROEDUCATION



Git - Github



Git - Github

Fundamentals

- Git - Github nedir?
- VCS Nedir?
- Ne amaçla kullanılır



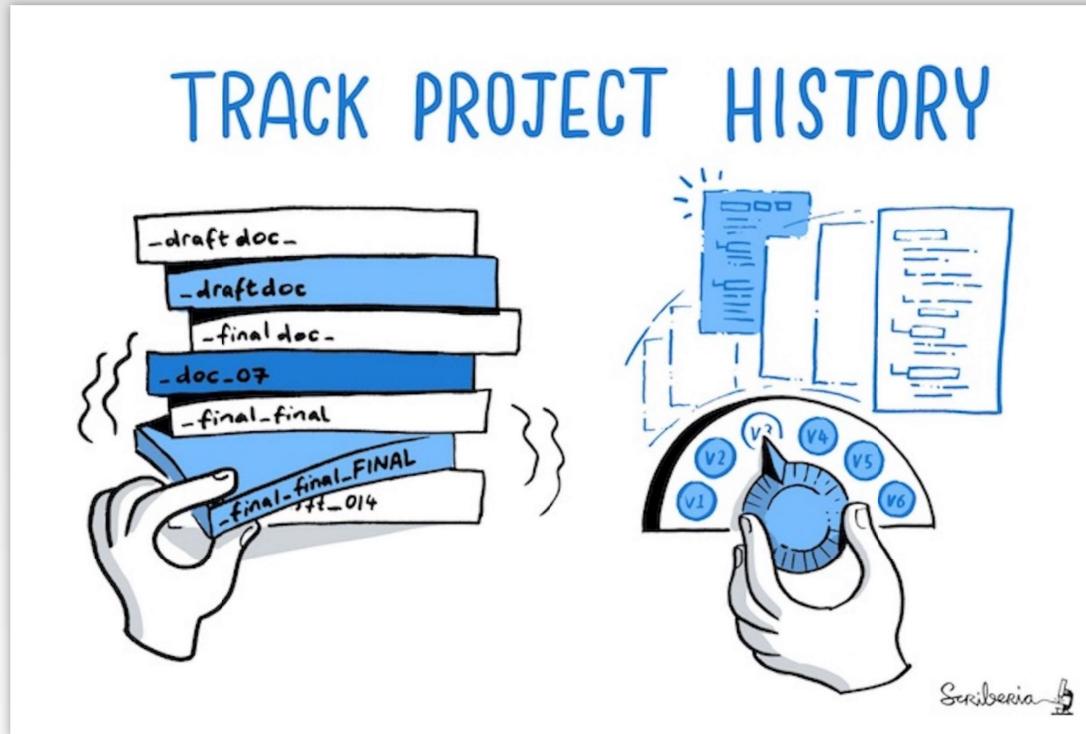
Git-Github nedir

Git versiyon kontrol sistemidir,
Github Git versiyon kontrol sistemini kullanan depolama servisleri diyebiliriz.





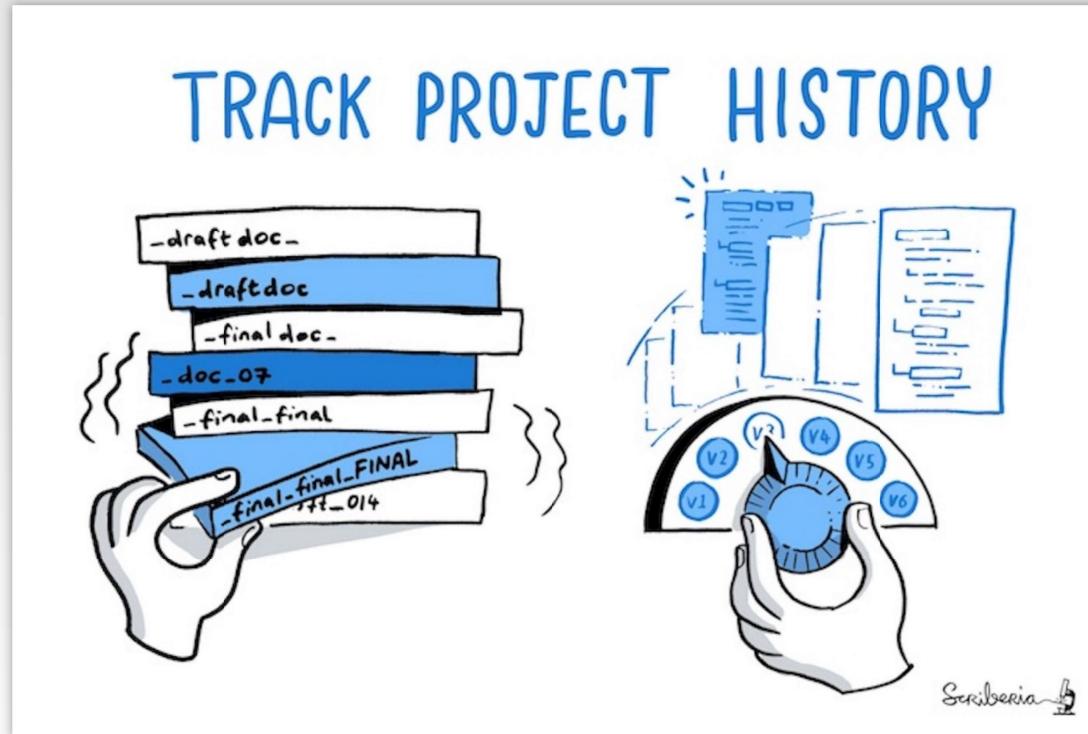
VKS nedir



Versiyon kontrol sistemi, dosya ve klasör yapısında değişiklikleri kaydeden bir sistemdir.



VKS nedir



- Bazı dosyaların veya proje tamamının bir önceki versiyonuna döndürülmesi,
- Zaman içerisinde yapılan değişikliklerin karşılaştırılması,
- Probleme neden olabilecek değişikliklerin en son kimin tarafından yapıldığının tespiti,



VKS çeşitleri

| 3 tip Versiyon Kontrol Sistemi vardır. |

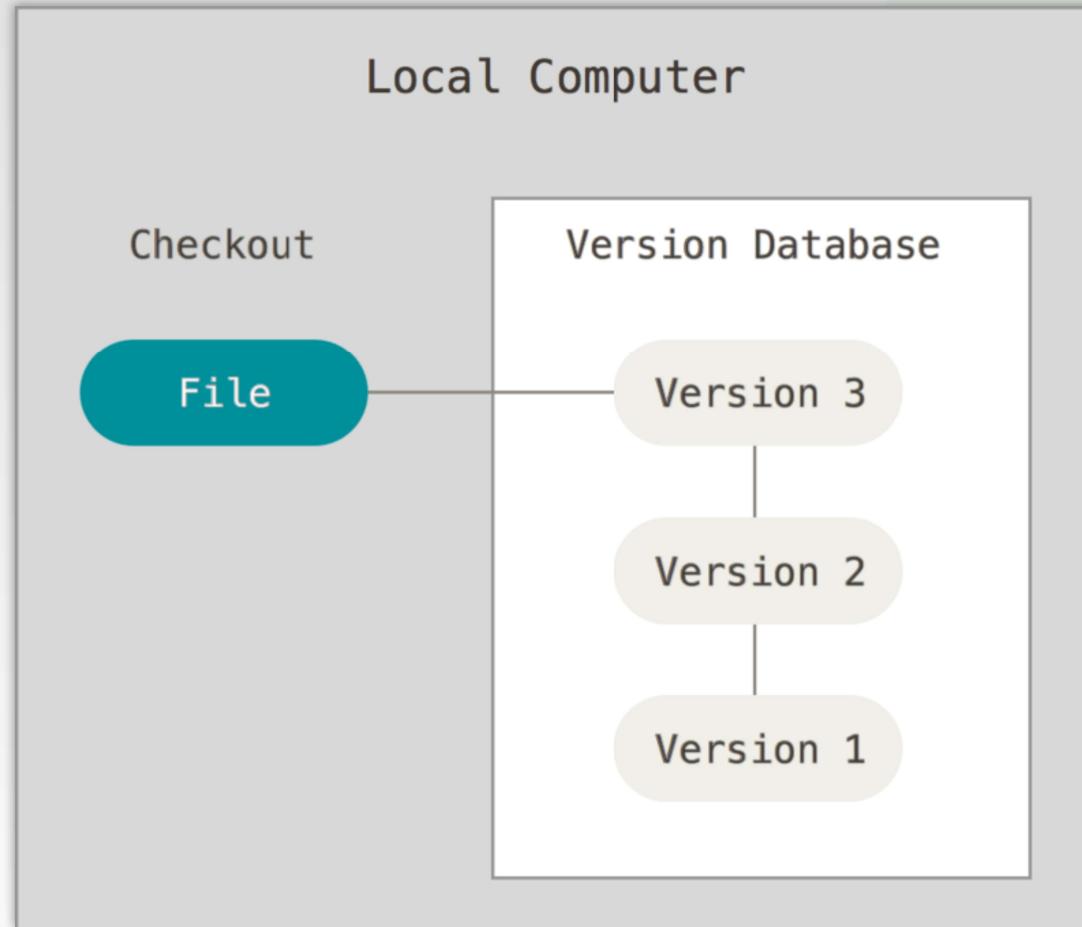
YEREL

MERKEZİ

DAĞITİK



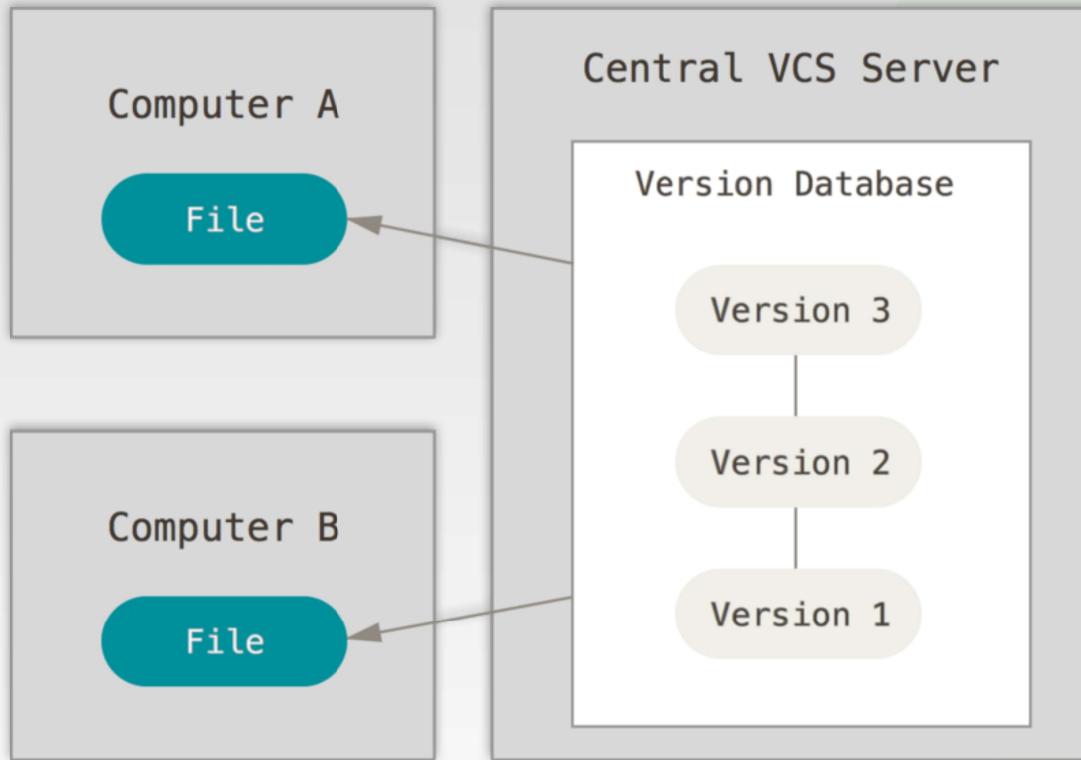
Yerel VKS nedir



<https://git-scm.com/book/en/v2/Getting-Started>



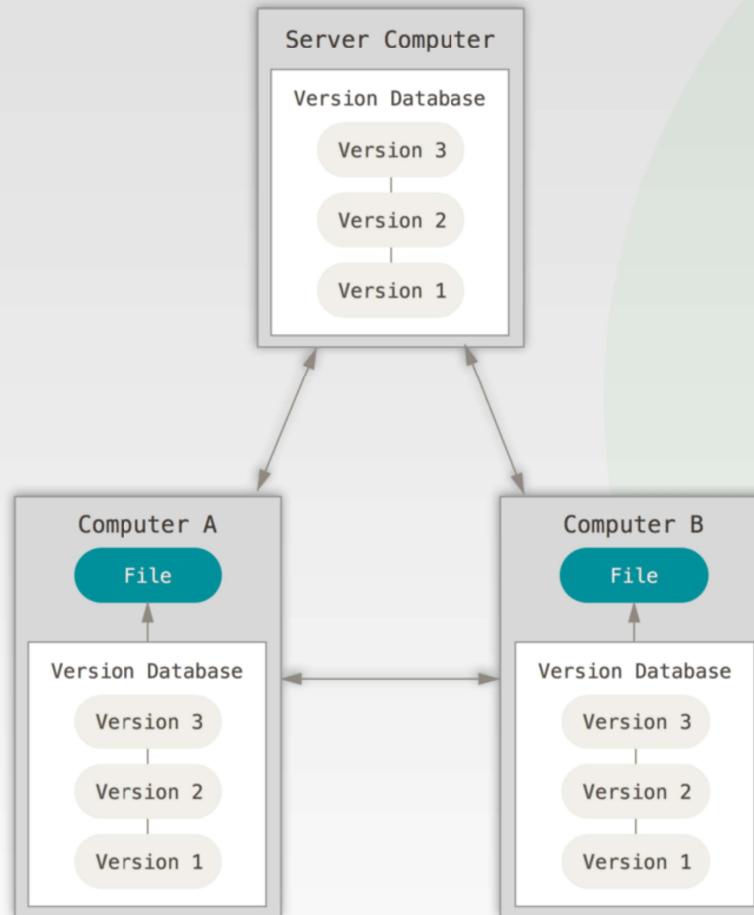
Merkezi VKS nedir



<https://git-scm.com/book/en/v2/Getting-Started>



Dağıtık VKS nedir



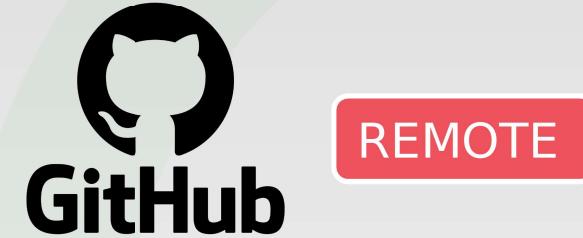
<https://git-scm.com/book/en/v2/Getting-Started>



Git-Github Ne Amaçla Kullanılır



- Lokalde versiyon yönetimi yapmak
- Offline çalışabilmek
- Hataları geri alabilmek
- Versiyonlar arasında geçiş yapabilmek



- Yedekleme (backup)
- Proje paylaşımı (sharing)
- Proje yayınılama (deployment)
- Ortak çalışma (collaboration)



Git - Github



- Kurulum ve ilk ayarlar
- Repository
- Lokal repo oluşturma
- Working space, staging, commit store
- Değişiklikleri iptal etme
- Önceki versiyonlara geri dönme
- Branches



Kurulum ve İlk Ayarlar



Version Control System

- Git altyapısını oluşturmak ve git komutlarını kullanabilmek için Git kütüphanesi kurulması gerekmektedir
[<https://git-scm.com/downloads>]
- *git --version*



Kurulum ve İlk Ayarlar

Git configuration

```
git config --global user.name "Ali Gel"  
git config --global user.email "ali@gel.com"
```

```
git config --global color.ui true
```

Yapılan commit leri bu isim ve eposta ile ilişkili olarak çalışan diğer kişilere epostayı gönderebilirsiniz.

Terminal de komutları renklendirilmesi.

- **System** parametresi kullanıldığında tüm kullanıcılar ve tüm repolar üzerinde etkili olur
- **Global** parametresi geçerli kullanıcının tüm repolar üzerinde etkili olur
- **Local** parametresi ise sadece geçerli repo üzerinde etkili olur



Genel Kavramlar

| Repository |

Versiyon kontrol ve birlikte çalışma altyapısını ayrı tutmak istediğimiz her bir bağımsız yapıya **repository** denir. Genellikle her proje için ayrı bir repository tanımlanır.



Local repo oluşturma

| git init |

Local bilgisayarımızda bir projeyi versiyon sistemine alabilmek için **git init** komutu kullanılır. Bu komut kullanılıncı proje klasöründe .git klasörü oluşturulur ve repomuzu saklayacaktır.



Genel Kavramlar



Working Space

.git klasörünün bulunduğu çalışma alanıdır. Klasörler ve dosyalar üzerinden değişiklik burada yapılır.



Staging Area

Versiyon oluşturulacak olan dosya veya klasörlerin geçici olarak toplandığı yerdir. Versiyon (commit) oluşturulduktan sonra otomatik olarak staging area boşaltılır



Comm

Git her bir ayrı bir ve olarak tut. Böylece çeşitli değişiklikler sonra projeler sorunlar olsa çıkarsa bir commit e dönülebilir.



Local versiyonlar oluşturma

Working Space veya Staging area'nın durumunu görmek için kullanılır.

Oluşturulan versiyonlar görmek için bu komutlar kullanılır

git status

Working Space

git add

Staging Area

git commit

git add dosya_adi
veya
git add .

git commit -m ''bir mesaj''
git commit -amend -m ''mesaj''



Versiyon detaylarını görme

git show

Bir versiyon içinde, hangi değişikliklerin olduğunu görmek için kullanılır.

git show *[hash kodun ilk 7 karakteri]*

git log --oneline

```
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
```



Versiyon oluşturmak için kodlar

Ana komutlar

git init

git add .

git commit -m "versiyon metni"

Repo oluşturur. Her projede en az bir kere kullanılır.

Dosyaları staging area ya getirir.

Versiyon oluşturur.

Yardımcı komutlar

git status

git log

git show [hash_kodu]

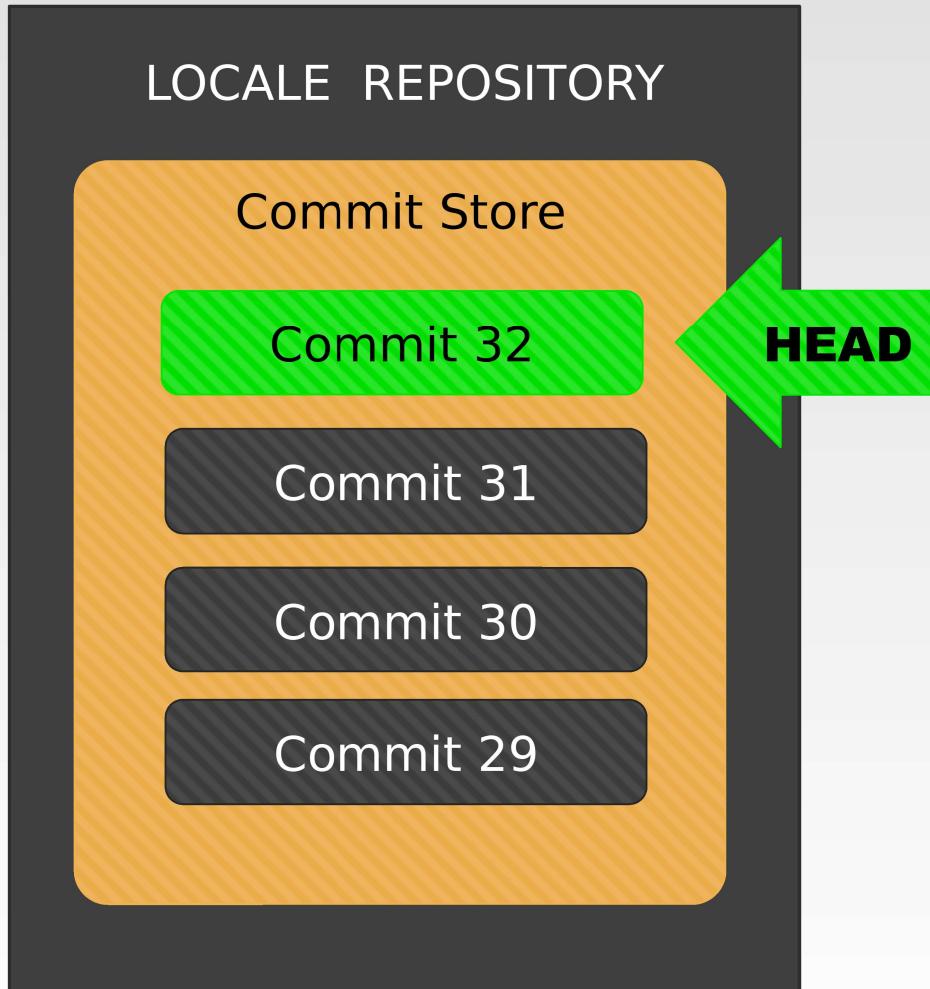
Genel durum ile ilgili bilgi verir.

Versiyonların listesini ve detaylarını gösterir.

Versiyondaki değişiklikleri gösterir.



Commit Store & Head



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

```
C:\Users\sariz\Desktop\test>git log
commit c417dfe1afa5deac505808a0a2c8ba05afc8e86d
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

Working space

git restore [dosya]

Tek bir dosyayı iptal eder

git restore .

Tüm dosyaları iptal eder

git reset --hard

Working space deki değişiklikleri iptal eder, staging area yi boşaltır.

Stage Area

git restore --staged [dosya]

Tek bir dosyayı iptal eder

git restore --staged .

Tüm dosyaları iptal eder

Comm

git checkout [

Dosya,hash ile belir

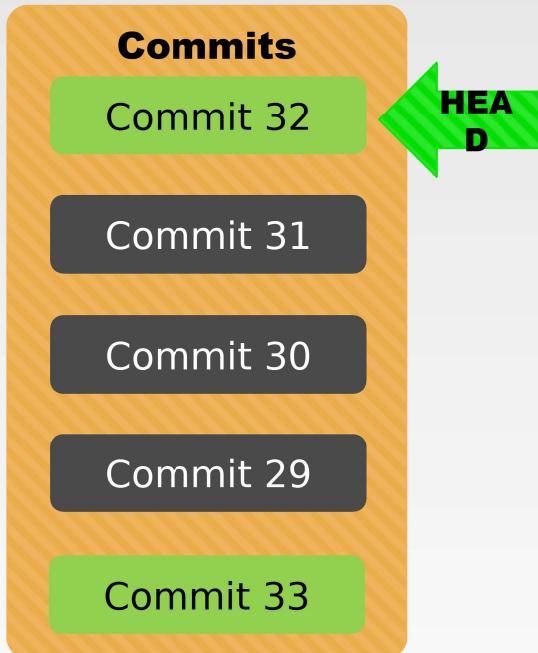
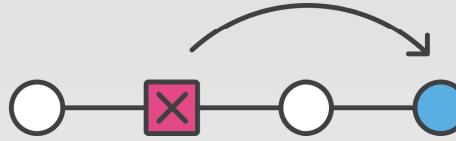
git checko

Hash değeri veril



Önceki versiyonlara dönmek

1.Yöntem: CHECKOUT



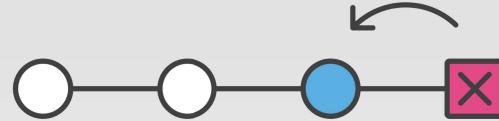
Önceki veriyonu incelemek için
git checkout [hash].

Bu işlemi kalıcı hale getirmek için
git commit -m"..."



Önceki versiyonlara dönmek

2.Yöntem: RESET

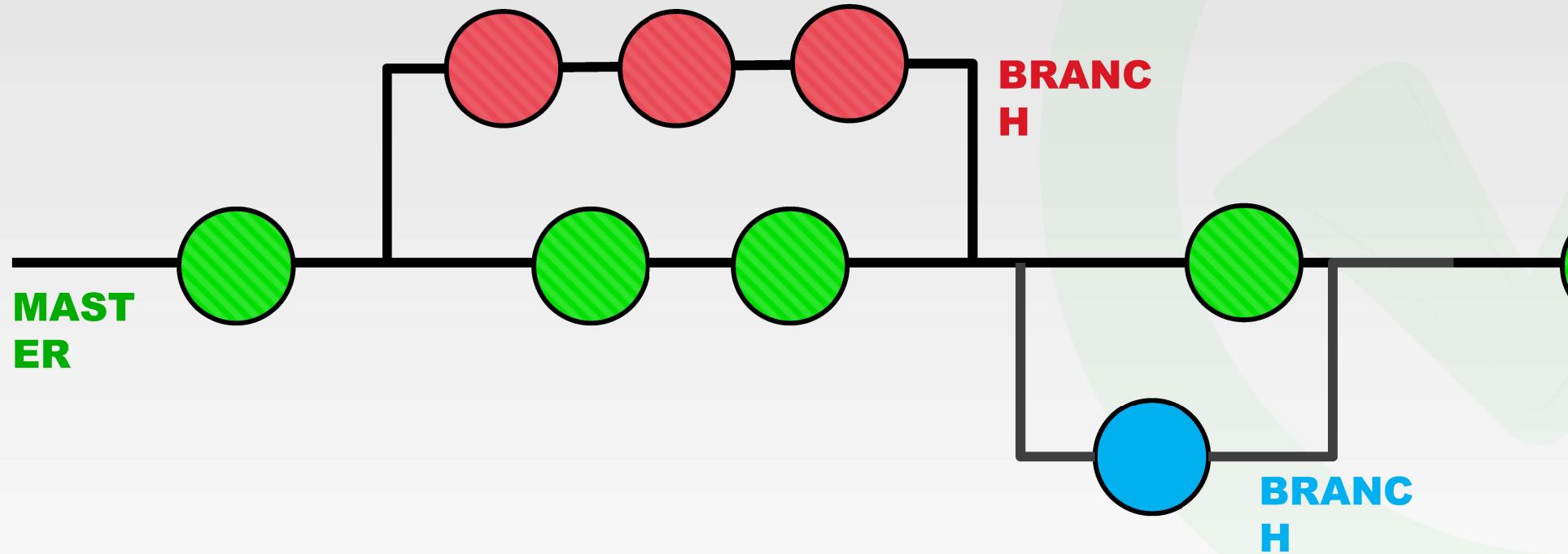


Geri alınamayacak şekilde önceki ve
dönmek

git reset --hard [hash]

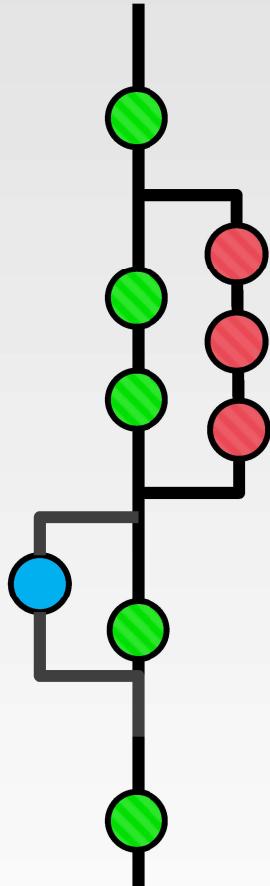


Branch (Dal)





Branch lerin faydaları



- Orginal kodların güvenliği sağlanır
- Her developer kendi bölümünden sorumlu olur
- Daha hızlı geliştirme yapılır
- Daha az hata oluşur
- Sorunlar daha hızlı düzelttilir.
- Organize kod yapısı sağlanır
- Kaos olmaz



Branch Komutları

`git branch [isim]`

Yeni branch oluşturur

`git checkout [isim]`

Branch aktif hale gelir

`git branch`

Branch ismini listeler

`git branch`

Mevcut branch leri listeler

`git merge [isim]`

İki branch i birleştirir

`git branch`

Branch ismini siler



Stashing

Working space ve staging area daki –henüz commit haline gelmemiş- değişikliklileri **geçici olarak geri alınması** için stashing işlemi yapılır.

git stash

Working space ve staging area daki değişiklikleri geçici olarak hafızaya alır ve bu bölgeleri temizler

git stash list

Hafızaya alınan değişiklikleri görmek için kullanılır

git stash po

Hafızaya alınan değişiklikleri geri uygulamak için kullanılır.



Git - Github



- Hesap oluşturma
- Repo oluşturma
- Genel kavramlar
- Github çalışma pren
- Clone
- Push & Pull
- Gitignore
- Merge & Conflicts



Github hesabı oluşturma

Github.co
m

The screenshot shows the GitHub homepage with a dark blue header. On the left is the GitHub logo. To its right are navigation links: "Why GitHub? ▾", "Team", "Enterprise", "Explore ▾", "Marketplace", and "Pricing ▾". To the right of these is a search bar with the placeholder "Search GitHub" and a magnifying glass icon. On the far right is a "Sign in" button. The background features a stylized globe with glowing blue and pink lines.



Github hesabı oluşturma

Welcome to GitHub!
Let's begin the adventure

Enter your email
✓ techproed11@gmail.com

Create a password
✓*

Enter a username
✓ techproed11

Would you like to receive product updates and announcements via email?
Type "y" for yes or "n" for no
✓ n

Eposta adresi

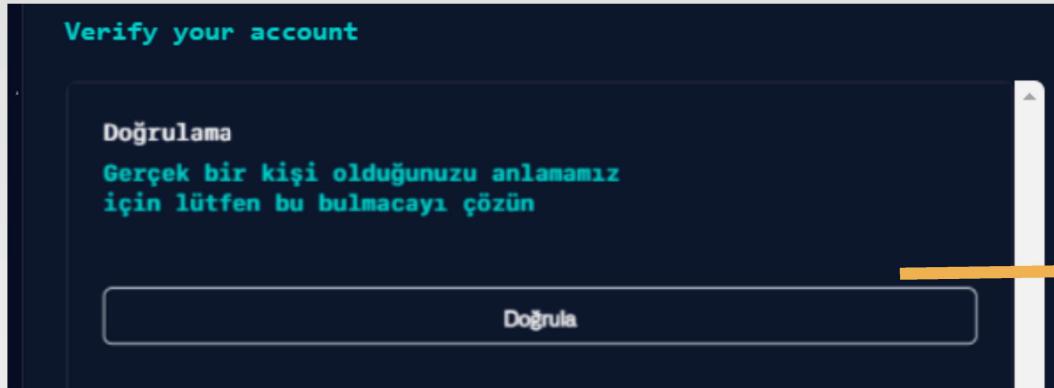
Sifre belirle

Bir kullanıcı adı

Ürün güncelleştirmeleri ve tanıtılardan en hızlı haberdar olmak isteyen yazılımcılar için yazıyo



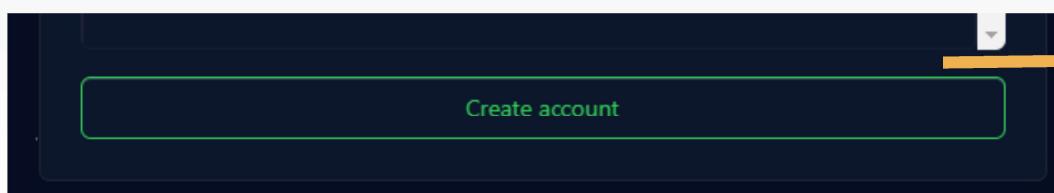
Github hesabı oluşturma



Doğrula butonu



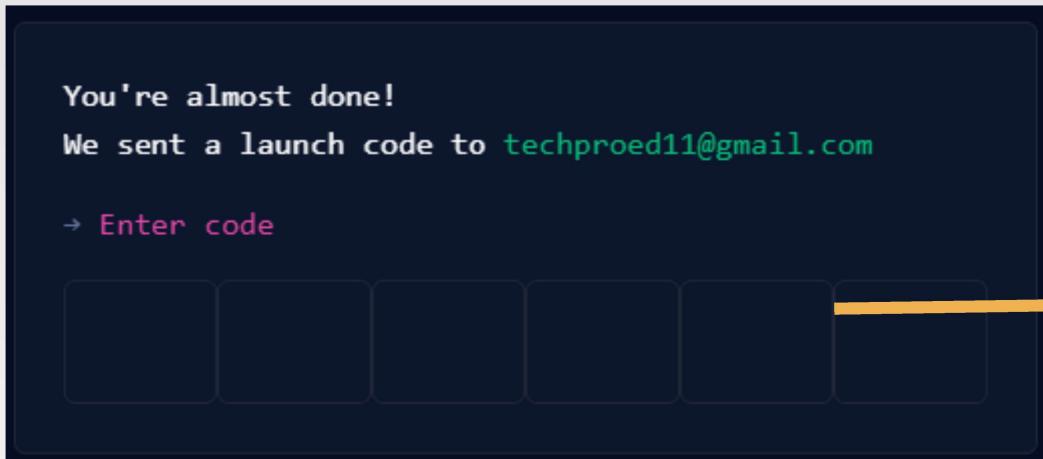
Doğrulama adı
geçiyor



Create Account
basıyor



Github hesabı oluşturma



Eposta adresine
kodu girerek
tamamlay



Github repo oluşturma

The screenshot shows the GitHub dashboard with the 'Repositories' tab selected. A large green button labeled 'New' is highlighted. Below it, the 'Create a new repository' form is displayed. The 'Owner' field is set to 'techproeducation1'. The 'Repository name' field is empty. A placeholder text 'scaling-meme?' is shown below the name field. The 'Description (optional)' field is empty. Under 'Visibility', the 'Public' option is selected, with a note that anyone on the internet can see the repository. The 'Private' option is also available. At the bottom is a green 'Create repository' button.

Pull requests Issues Marketplace Explore

Overview Repositories 15 Projects Packages

Find a repository... Type Language Sort New

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.

Create repository

1

New

2

Repository için b

3

Herkes tarafından ulaşılabilir, yoksa sadece bizim kullanıcılarımız.

4

Create repository



Kavramlar

Clone

Github daki bir repoyu lokale indirme işlemidir

Push

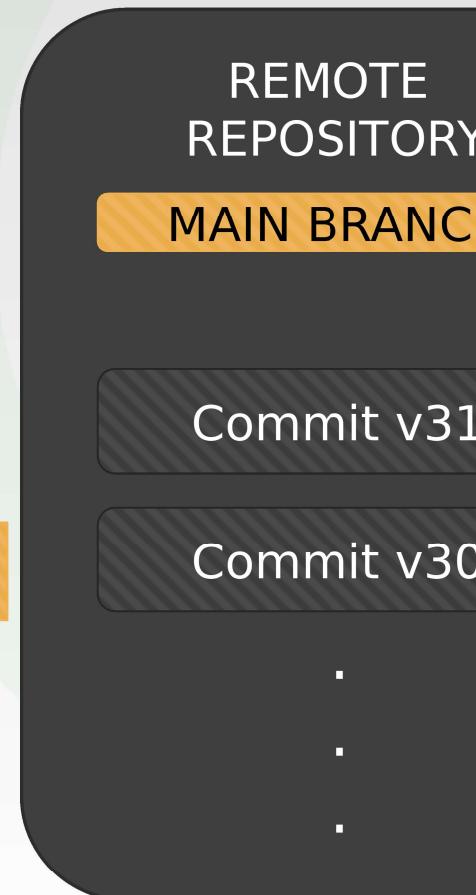
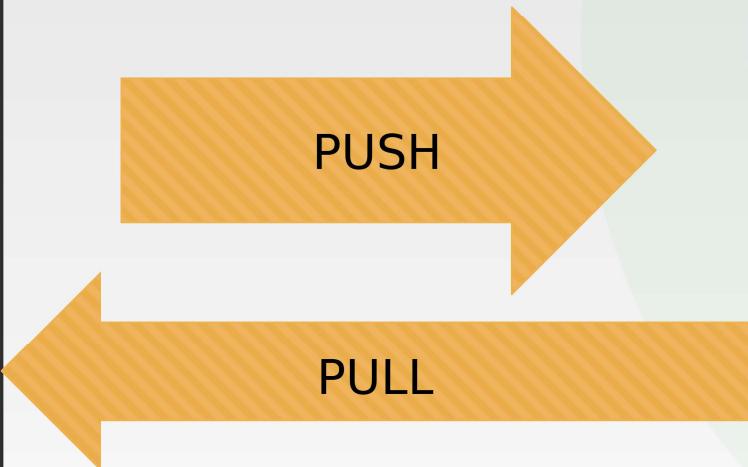
Lokalde oluşturulan commit lerin github a gönderilmesi işlemidir.

Pull

Fetch ve Merge işlemini tek başına yapar



Github Çalışma Prensibi





Cloning

| **git clone** |

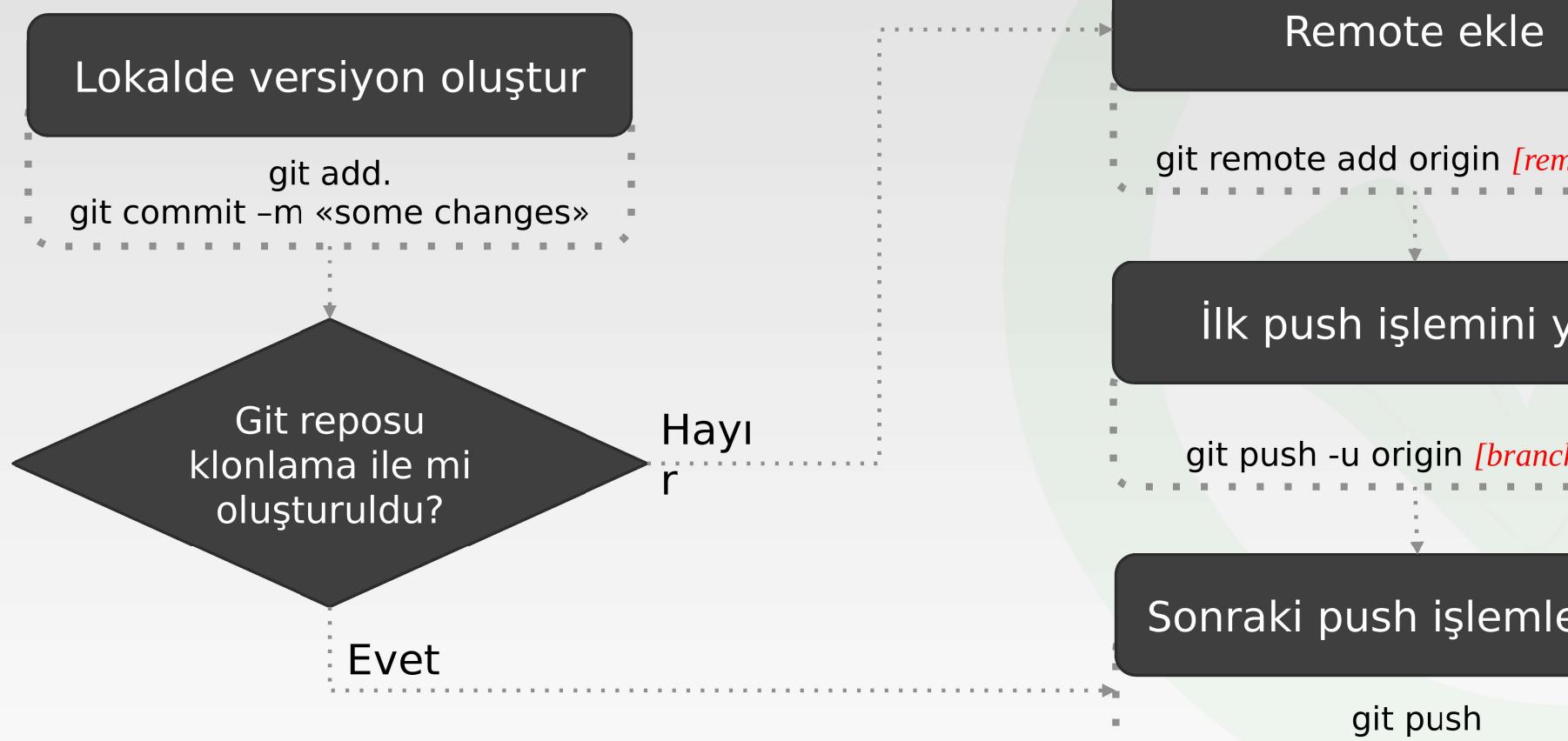
Github daki bir reponun lokale indirilmesi işlemine klonlama denir. Pub veya gerekli izinlere sahip olunan private repolar klonlanabilir.

Bunun için **git clone** komutu kullanılır.

```
git clone https://github.com/techproeducation-batchs/B-71-FED-TR.git
```



Github'a yükleme (pushing)





Github dan commit çekme (pull)

Github üzerinden local repo güncellenmek istenirse aşağıda komutlar kullanılır

git fetch

Değişiklikleri remote'dan local'e indirir

git merge

İndirilen değişiklikleri local repoya uygular

veya
A

git pull

fetch & merge



.gitignore

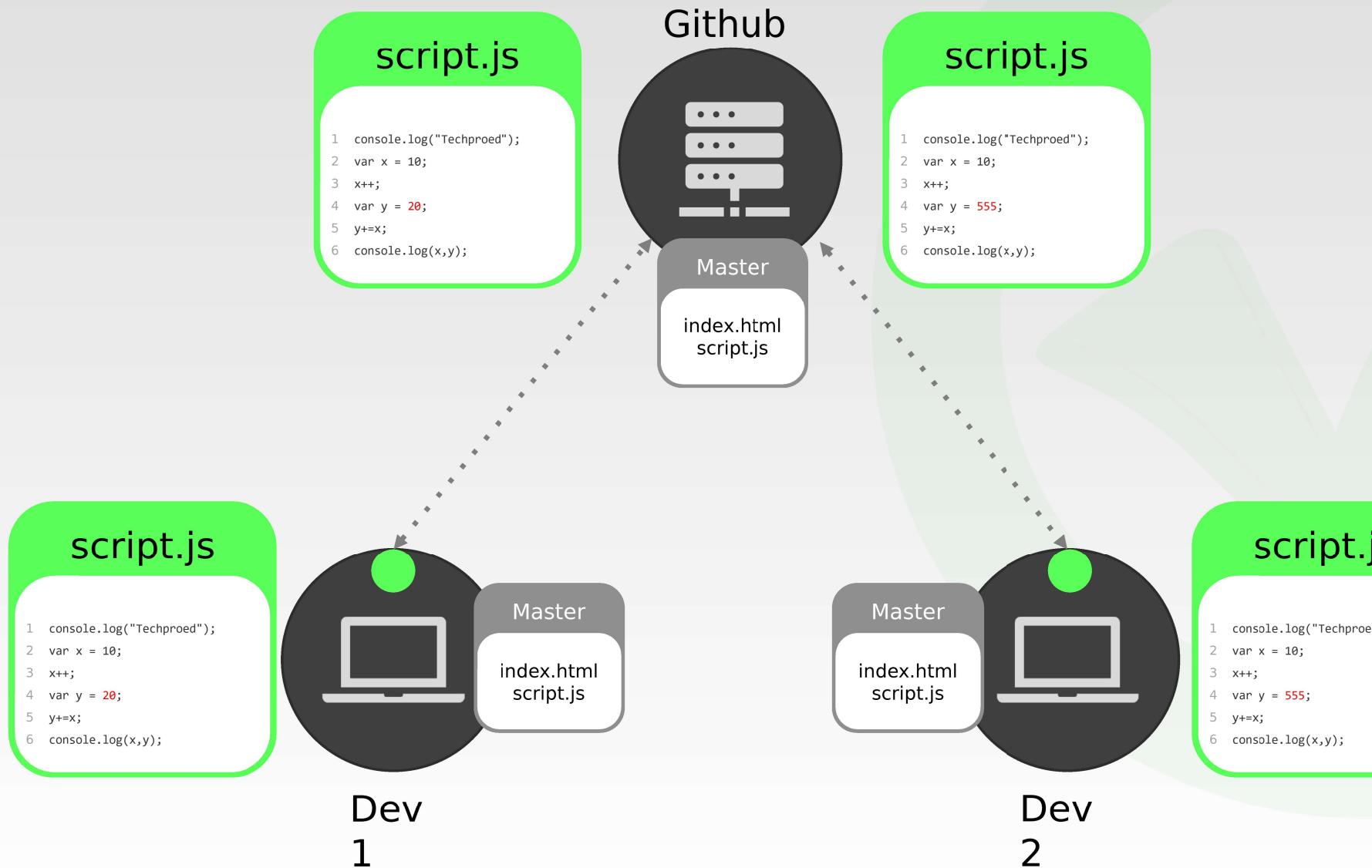
Staging area' ya gitmesini istemediğimiz, yani versiyon kontrol sistemine dahil etmek istemediğimiz dosya ve klasörlerimizi tanımladığımız özel bir dosyadır.

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

.gitignor
e



Merge Conflict





Merge Conflict

script is

1 consol

2 var x = 10;

3 x++;

4 var y

4 var y

$$5 \quad y=x;$$

6 console

Auto-merging script.js

CONFLICT (content): Merge conflict in

Automatic merge failed; fix conflict
commit result.

<<<<<<<<<<<< HEAD (Current Change)

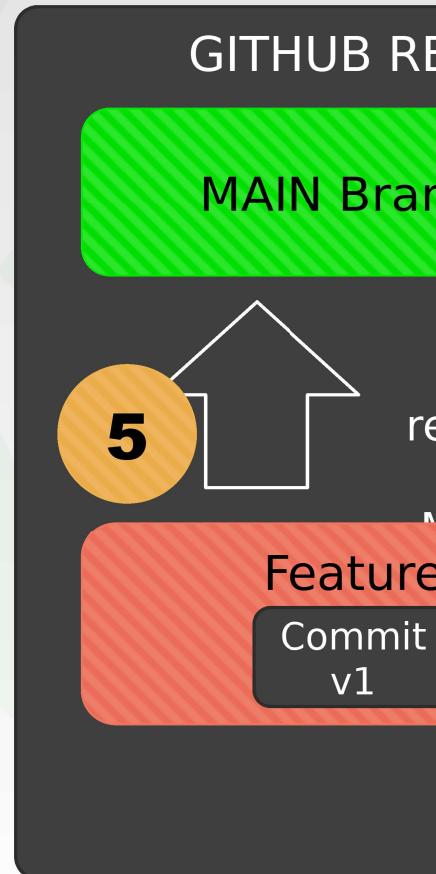
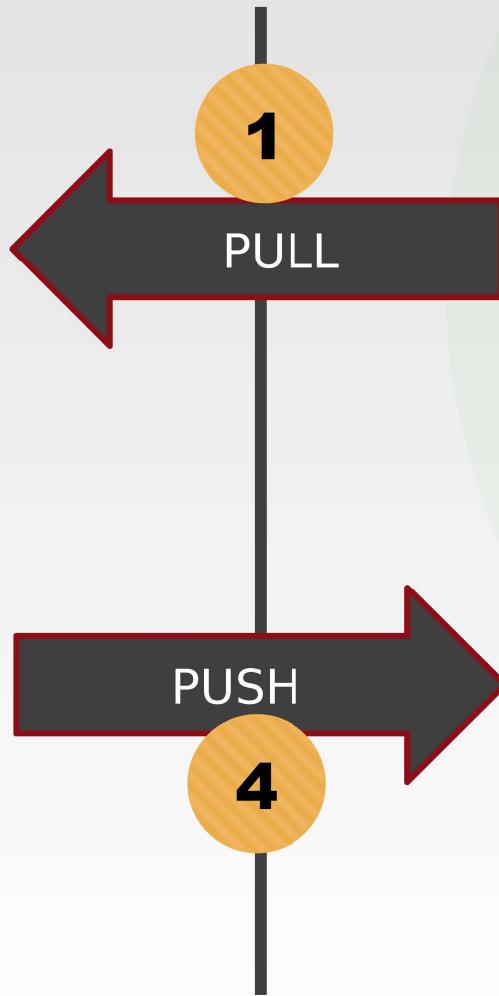
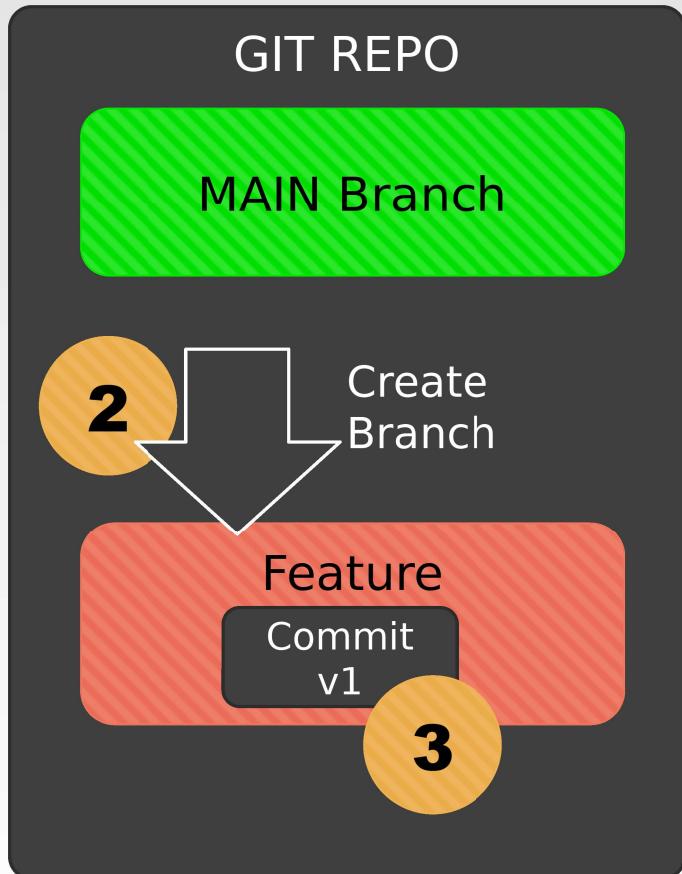
```
var y = 20;
```

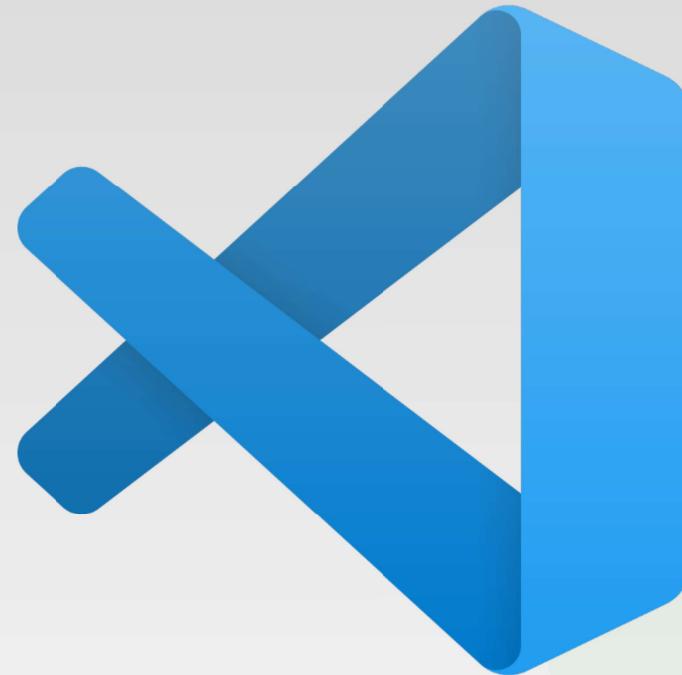
䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗ ䷗

```
var y = 555;
```



Git – Github Çalışma döngüsü





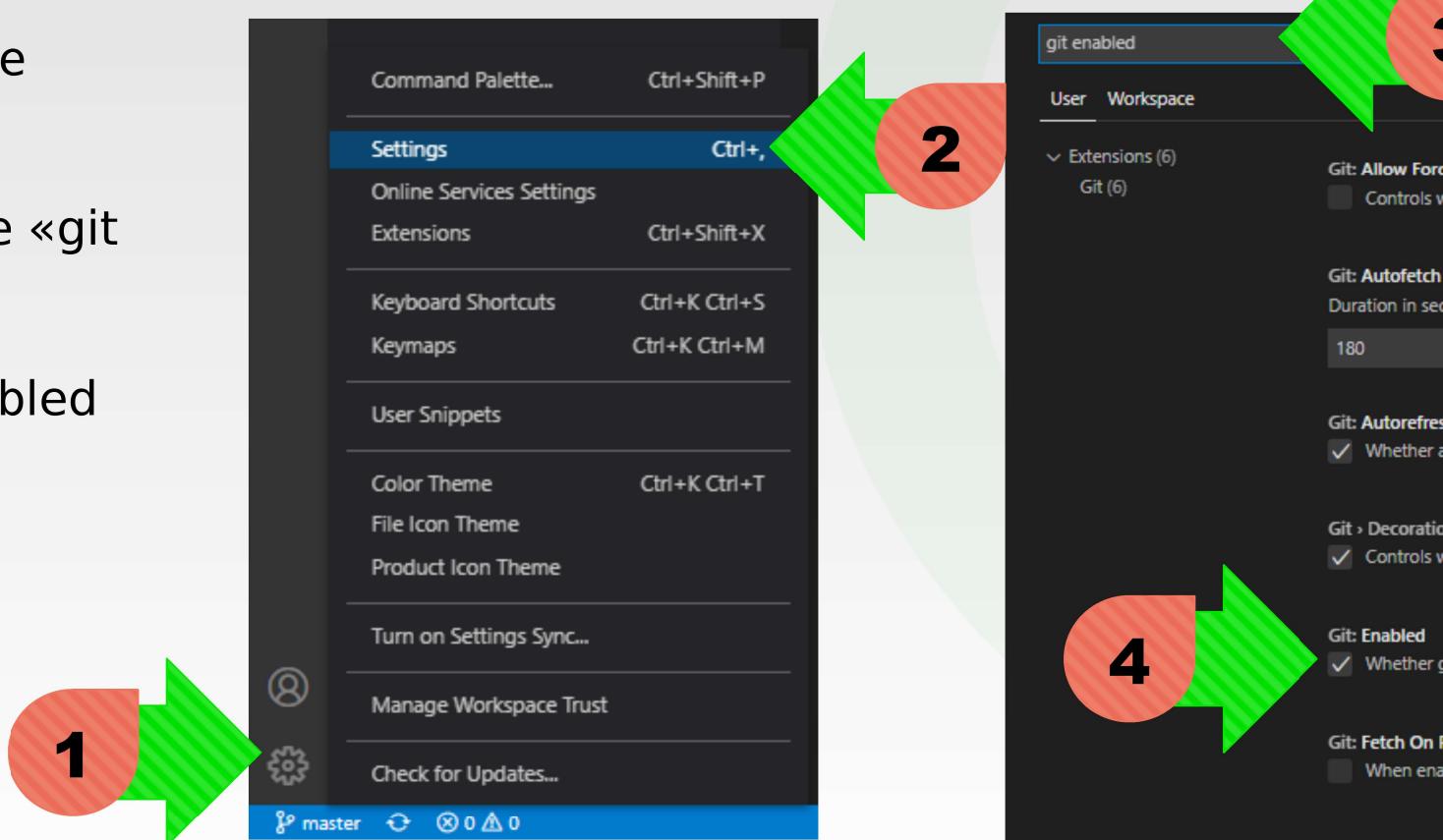
Visual Studio Code ile Git – Github Entegrasyonu



VS Code & Git-Github

VS Code' da Git in Aktif Hale Getirilmesi

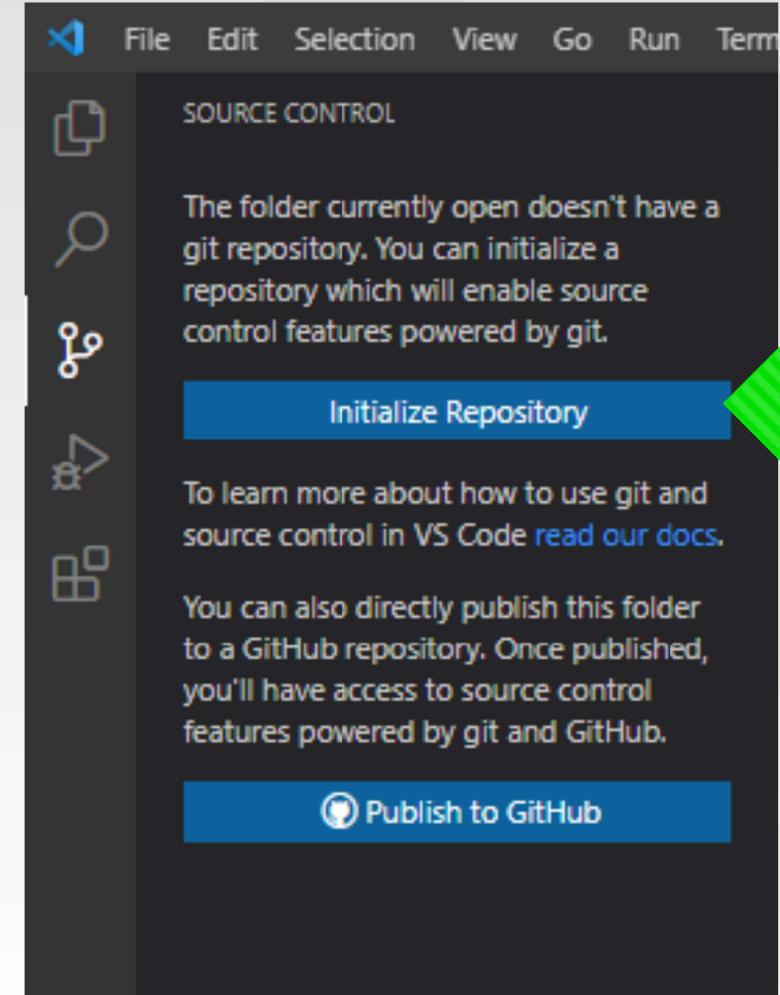
- 1- Ayarlar simgesine tıklanır
- 2- Settings seçilir
- 3- Arama bölümüne «git enabled» yazılır
- 4- Alt tarafa gelen ayarlardaki Git Enabled kutusu işaretlenir.





VS Code & Git-Github

5- Bir proje VS Code ile açıldıktan sonra Initialize Repository ile repository oluşturulur. Burada VS Code **git init** komutunu çalıştırır. Bu aşamada local version kontrol sistemi kurulmuş olur.





VS Code & Git-Github

The screenshot shows the Visual Studio Code interface with a floating settings sidebar for 'Settings - xxx - Visual Studio Code'. A green arrow points from the top right towards the sidebar, which contains sections for 'User' and 'Workspace' settings, including 'Git' (6) and 'Autorefresh' (8). Another green arrow points from the bottom left towards the main workspace, where the 'SOURCE CONTROL' view is open. In the 'Remote' section of the source control menu (7), a red circle with the number 6 highlights the 'Add Remote...' option. A third green arrow points from the bottom right towards the bottom of the screen, where a modal dialog box is displayed for entering a 'Remote name' (10). The dialog also includes a message: 'Please provide a remote name (Press 'Enter' to confirm or 'Escape' to cancel)'.

6- Üç basılı
7- Re
8- Ad
9- Açı
githu
alınan
yapış
10- R
yazılı
basılı

File Edit Selection View Go Run Terminal Help

SOURCE CONTROL

Message (Ctrl+Enter to commit on 'ma')

Changes index.html

View & Sort

Pull

Push

Clone

Checkout to...

Commit

Changes

Pull, Push

Branch

Remote

Stash

Tags

Show Git Output

index.h Provide repository URL or pick a repository source.

Add remote from GitHub

User Workspace

Extensions (6)

Git (6)

Git: Allow Force Push

Controls whether force push (with or without lease) is enabled.

Git: Auto Fetch Period

Duration in seconds between each automatic git fetch, when Git: Autorefresh is enabled.

180

Autorefresh

Whether auto-refreshing is enabled.

Clone

HTTPS SSH GitHub

https://github.com/cehproed-project/test.

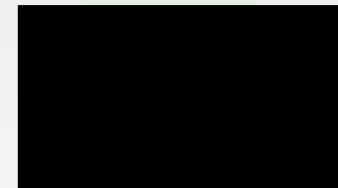
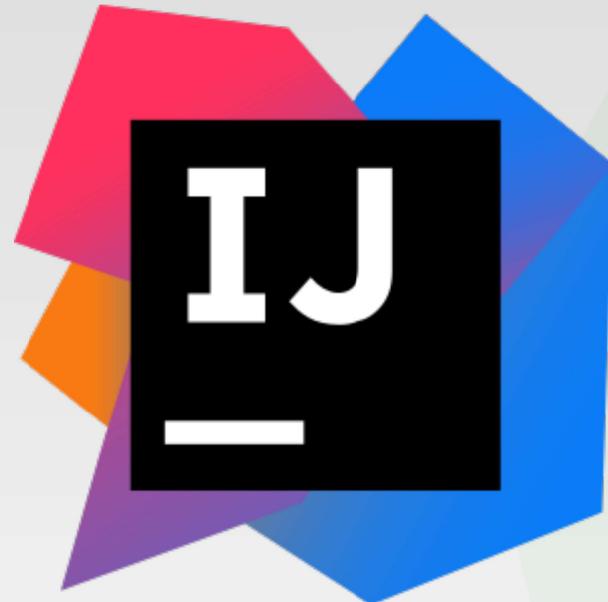
Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

10

Remote name

Please provide a remote name (Press 'Enter' to confirm or 'Escape' to cancel)



IntelliJ ile Git – Github Entegrasyonu

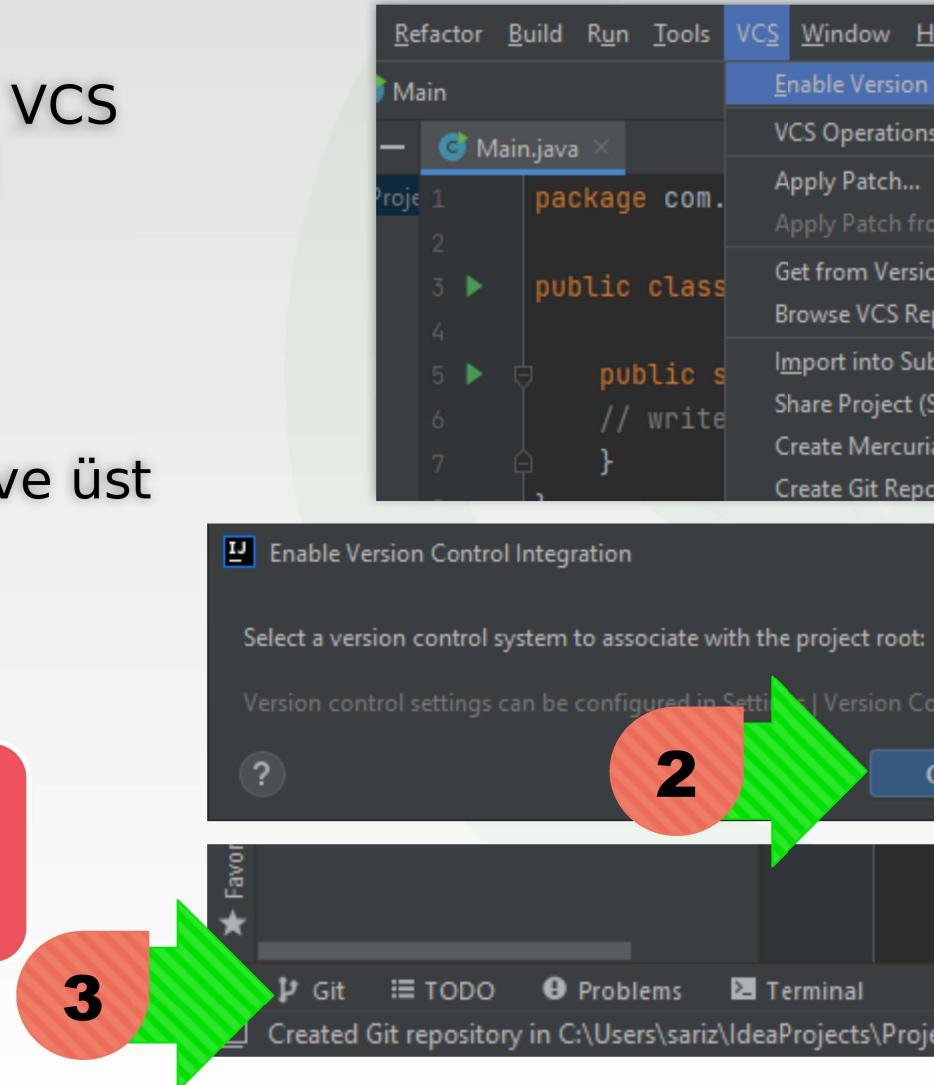


IntelliJ ile Git Github

- 1- IntelliJ üzerinde VCS kullanmak için VCS menüsündeki «Enable Version Control Integration» komutu seçilir.
- 2- Açılan pencerede Git seçilerek OK butonuna basılır
- 3- Böylece sol alt tarafta Git sekmesi ve üst tarafta Git menüsü görülecektir. (Bu aşamada git init çalıştırılmış oldu)



IntelliJ, üzerinde git ile ilgili olası sorunlarda File/Invalidate Caches komutu kullanılabilir





IntelliJ ile Git Github

Github bağlantısı

4- Git menüsüne tıklanır

5- Manage Remotes seçilir

6- Açılan pencerede + butonuna basılır

7- Github adresi yapıştırılır ve OK butonuna basılır

Ardından github hesabı için authorization işlemleri başlayacaktır.

