



# Git ve GitHub: Zaman Makinesi ve Sosyal Ağ

Hiç bilmeyenler için modern yazılım geliştirmenin temelleri.

Kodunuzun tarihçesini yönetin (Git) ve dünyaya paylaşın (GitHub).  
Bu sunumda, yazılım dünyasının 'Kaydet' butonu ve 'Sosyal Medyası'nı keşfedeceğiz.



# Neden Buna İhtiyacımız Var?

## Sorun:

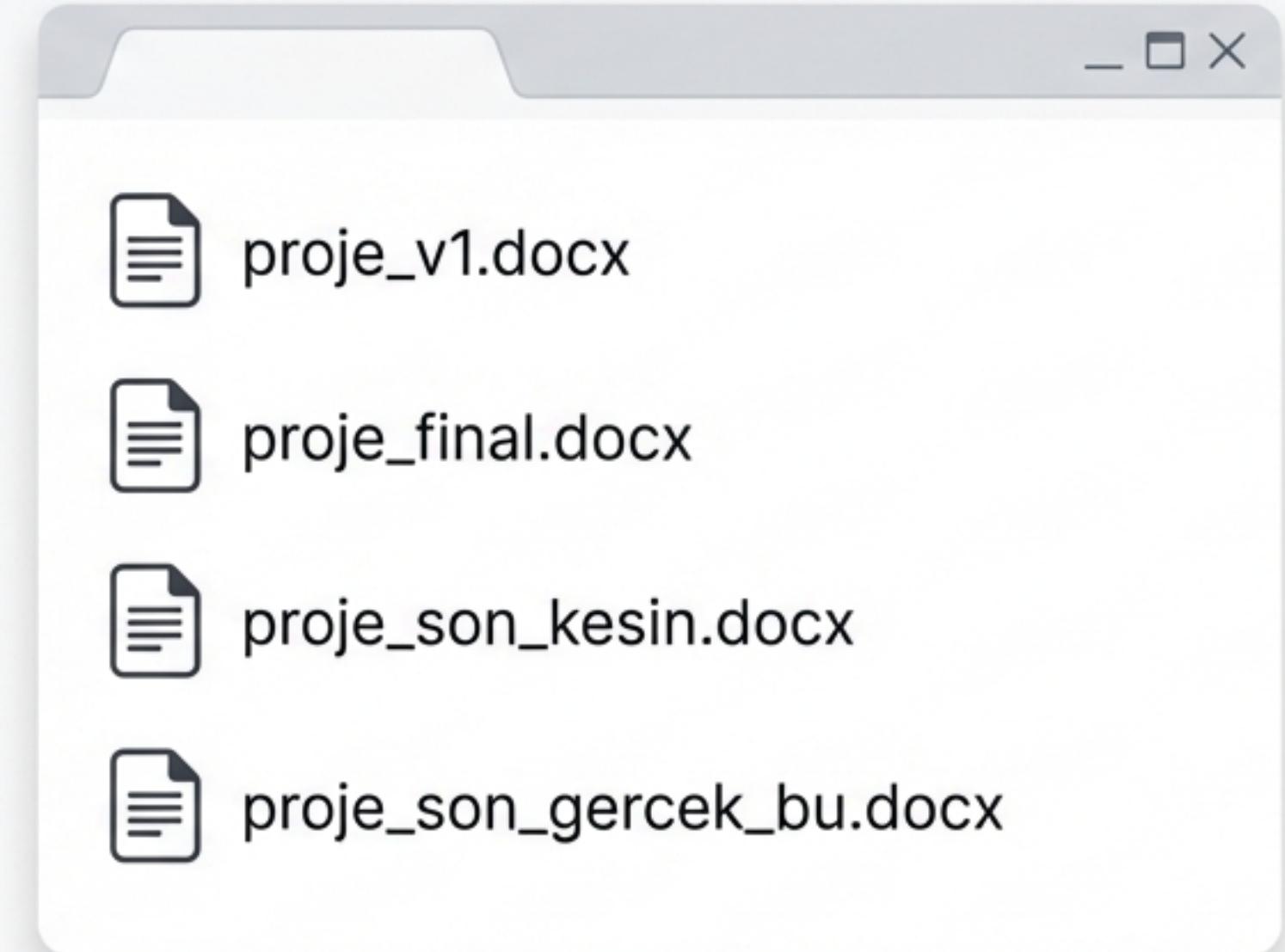
Manuel dosya yedekleme hataya açıktır, sürdürülemez ve ekip çalışmasını imkansız kılar.

## İhtiyaç:

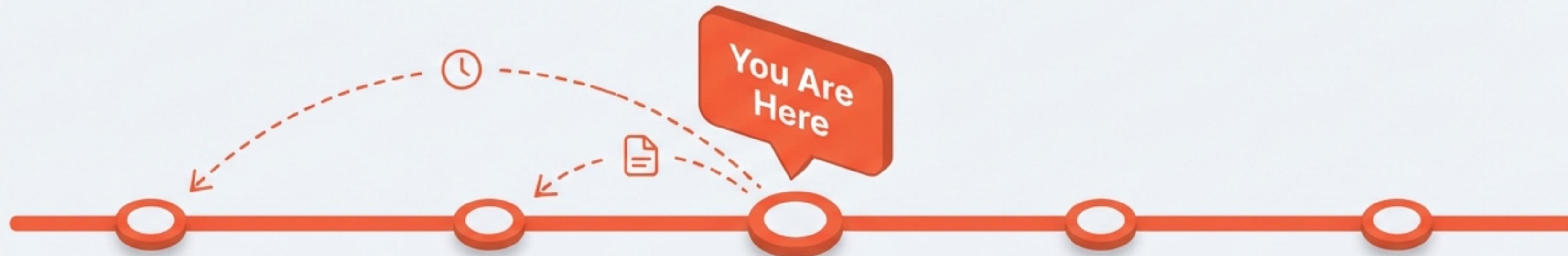
Bir hata yaptığınızda, projenin çalışan son haline dönebilmek istersiniz.

## Çözüm:

**Versiyon Kontrol Sistemi**. "Keşke 1 saat öncesine dönebilsem" dediğiniz anlar için tasarlandı.



# Git Nedir? — Kişisel Zaman Makineniz



## Tanım

Git, bilgisayarlarınızda yerel olarak çalışan, hız odaklı ve dağıtık bir Versiyon Kontrol Sistemi'dir.

## Oyun Analojisi

Bilgisayar oyunlarındaki "Save Point" (Kayıt Noktası) mantığı gibidir. Canavarın karşısına çıkmadan önce oyunu kaydederiniz; ölürseniz o noktadan tekrar başlarsınız. Git, kodunuz için tam olarak bunu yapar.

## Kökeni

2005 yılında Linux çekirdeğini geliştirmek için Linus Torvalds tarafından yaratılmıştır.

# Git Nasıl Çalışır? — 3 Temel Aşama



## 1. Working Directory (Çalışma Dizini)

Masanızın üzeri. Dosyaları oluşturduğunuz ve düzenlediğiniz dağınık alan.

## 2. Staging Area (Sahne/Hazırlık)

Fotoğraf çekilmeden önceki hazırlık. Hangi değişikliklerin kaydedileceğini burada seçersiniz.

## 3. Repository (Depo)

.git klasörü. Fotoğrafın çekilip albüme kalıcı olarak eklendiği yer.

Akış: Değişiklik Yap -> Sahneye Al (add) -> Kalıcı Kaydet (commit)

# GitHub Nedir? — Kodların Sosyal Medyası



## Tanım

Git dosyalarınızı (repository) internet üzerinde saklamamanızı sağlayan bulut tabanlı bir hizmettir. Git projeleriniz için bir Google Drive gibidir.

## Yedekleme & İşbirliği

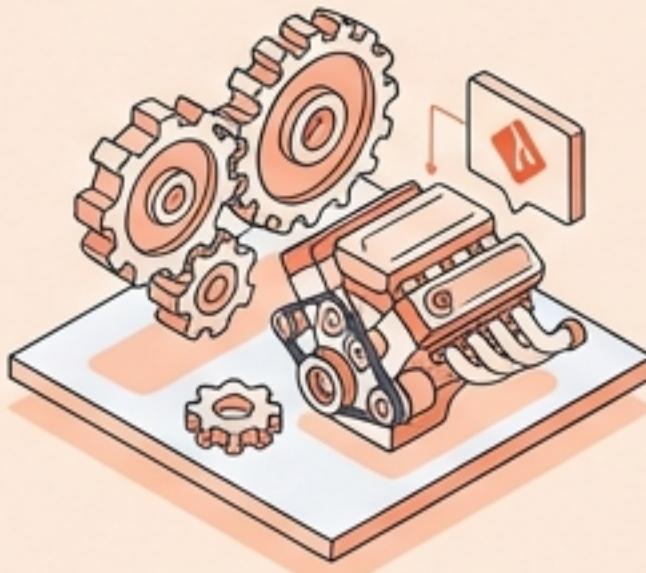
Bilgisayarınız bozulsa bile kodunuz güvendedir. Takım arkadaşlarınızla aynı dosya üzerinde aynı anda çalışabilirsiniz.

## Portfolyo

Yazdığınız kodları dünyaya sergileme alanınızdır. Geliştiriciler için bir LinkedIn karışımıdır.

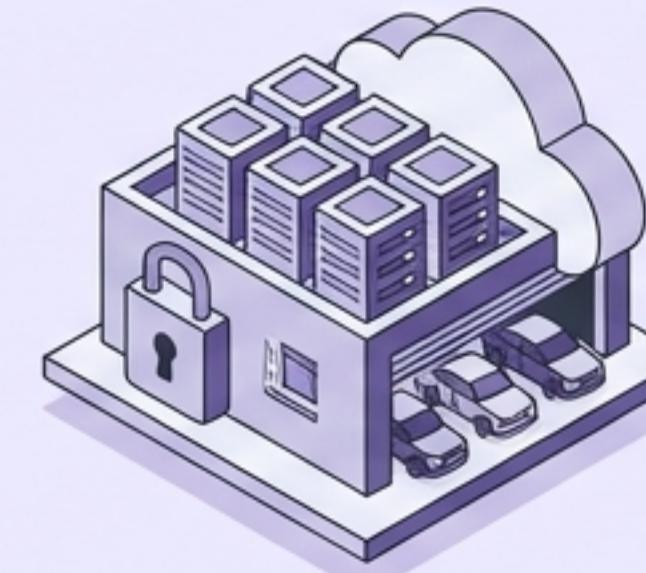
# Git vs. GitHub: Fark Nedir?

## Git



- **Araçtır (Tool):** Arabanın motoru gibidir.
- **Yereldir (Local):** İnternet olmadan çalışır.
- **Tarihçe:** Linus Torvalds tarafından geliştirildi (Açık Kaynak).
- **İşlev:** Versiyon kontrolünü yapar.

## GitHub



- **Hizmettir (Service):** Arabayı park ettiğiniz dev garajdır.
- **Buluttadır (Remote):** İnternet gerektirir.
- **Sahiplik:** Microsoft tarafından satın alınmıştır.
- **İşlev:** Barındırma ve paylaşımı sağlar.

Özet: Git kullanmak için GitHub'a mecbur değilsiniz, ama GitHub kullanmak için Git'e ihtiyacınız vardır.

# Kurulum ve Kimlik — ‘Bunu Kim Yazdı?’

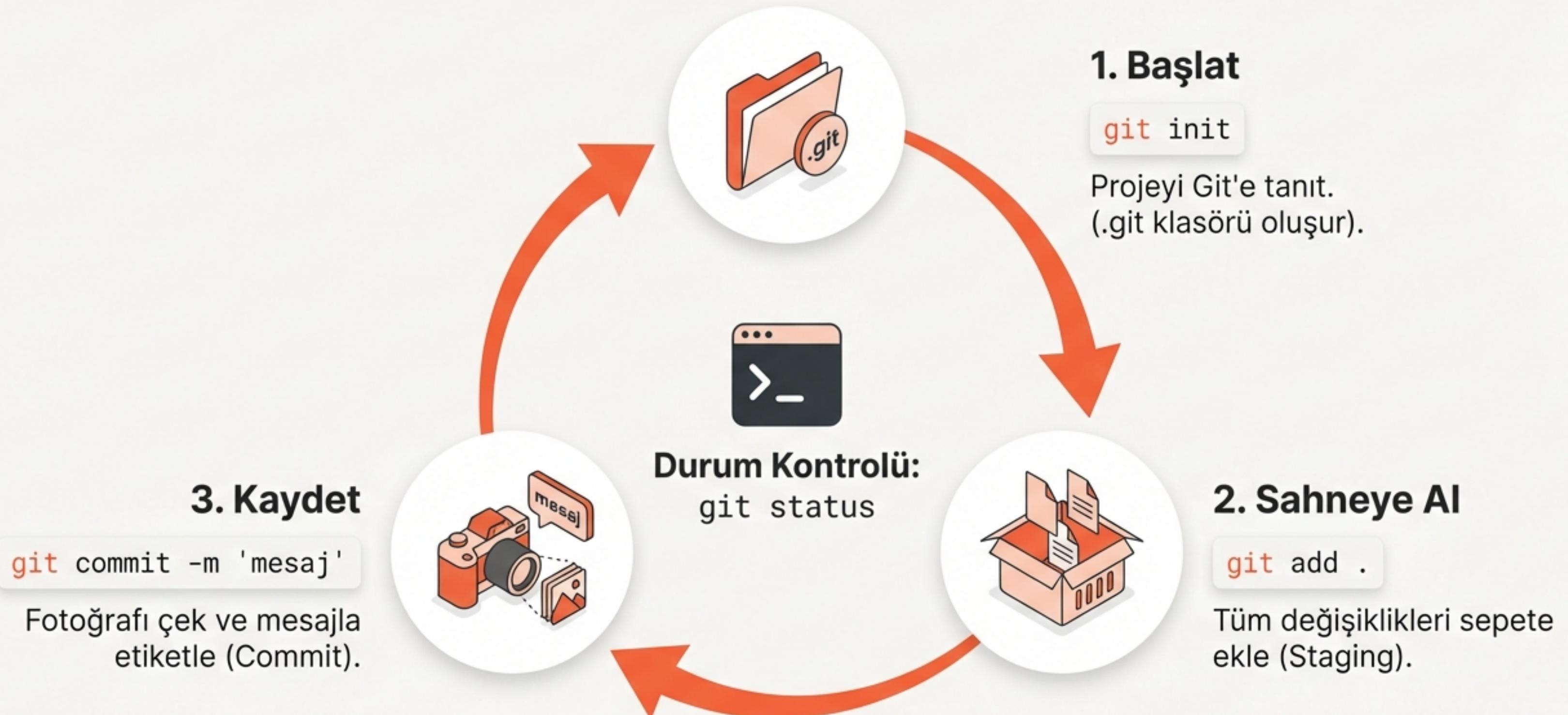


Git, yapılan her değişikliğin kime ait olduğunu bilmek ister. Bu ayarı sadece bir kez yapmanız yeterlidir.

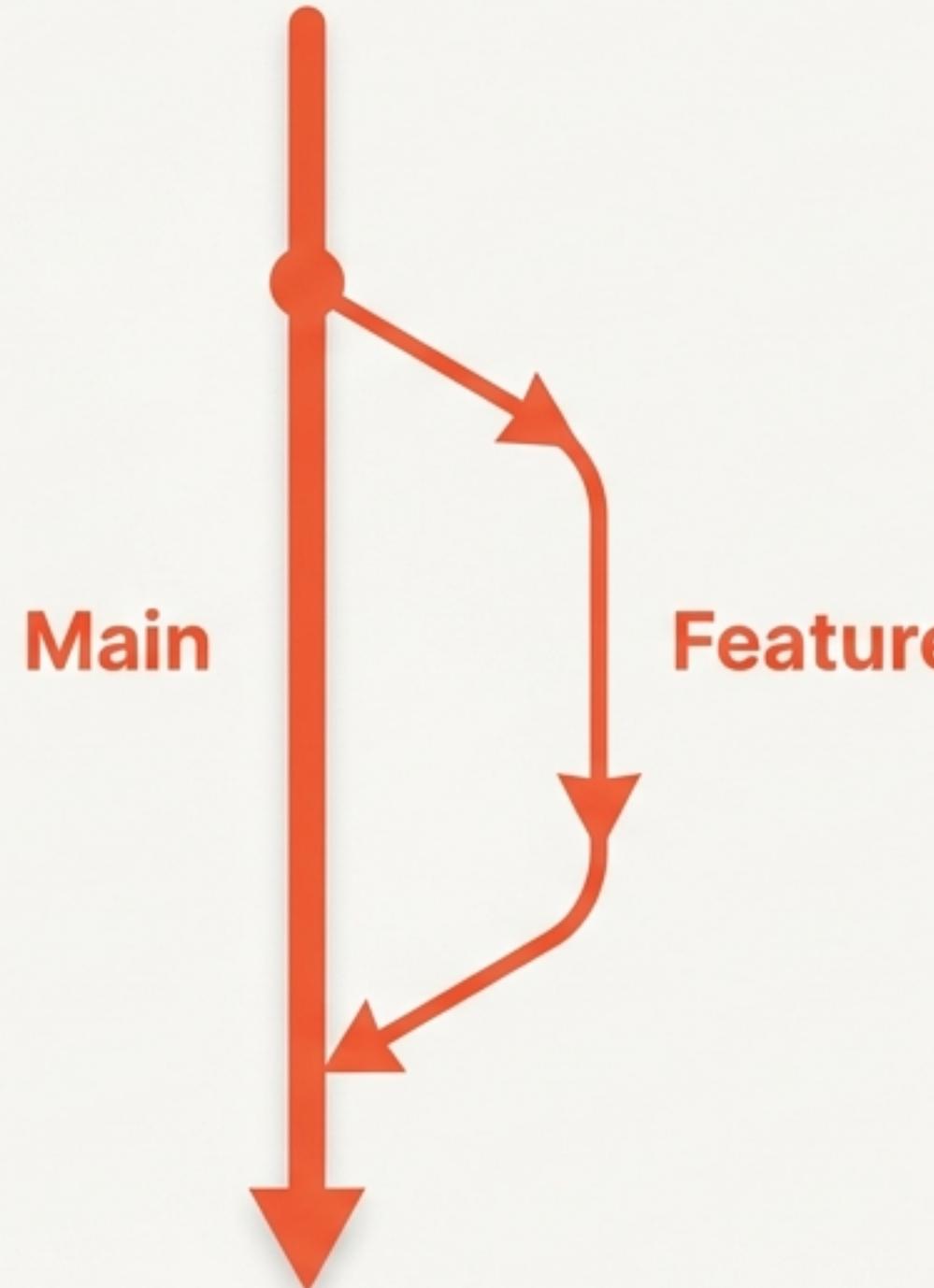
A screenshot of a terminal window titled "Terminal". The window has three colored window control buttons (red, yellow, green) at the top left. The title bar says "Terminal". Inside the terminal, there are two lines of text in white on a black background:

```
$ git config --global user.name 'Adınız'  
$ git config --global user.email 'mail@adresiniz.com'
```

# Temel Komutlar 1: Yerel Kayıt Döngüsü



# Dallanma (Branching) — Paralel Evrenler



## Kavram

Ana projeyi bozmadan riskli denemeler yapmak.  
Ana koddan bir kopya alıp 'paralel bir evrende'  
çalışmak gibidir.

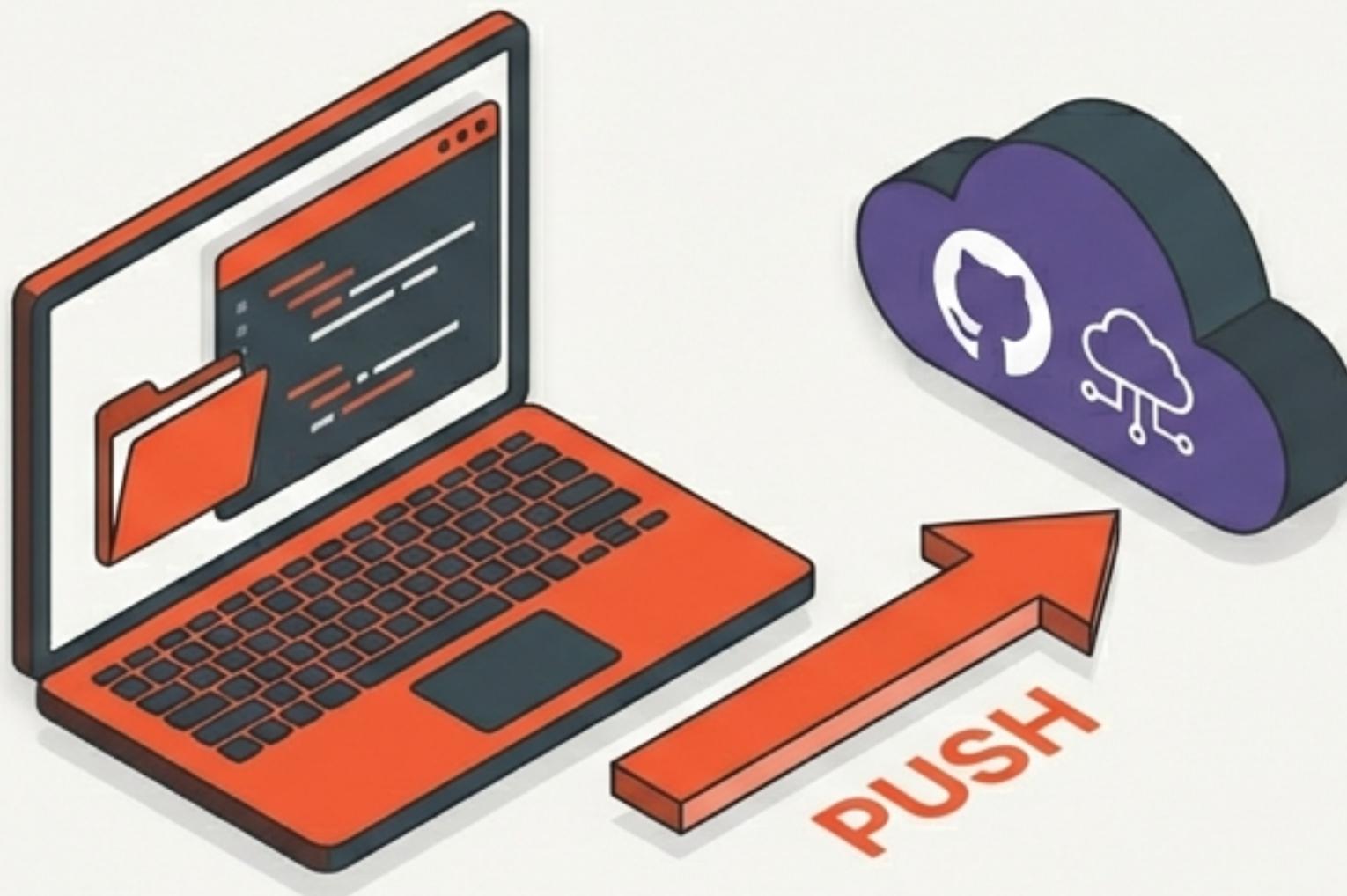
## Komutlar

`git branch [isim]` (Yeni evren yarat)  
`git checkout [isim]` (Evrene geçiş yap)

## Merge

İşiniz bittiğinde bu paralel evreni ana evrenle  
birleştirirsiniz.

# Buluta Göndermek — Push



Bilgisayarınızdaki (Local) değişiklikleri GitHub'a (Remote) gönderme işlemidir.

## Remote Ekleme (Hedefi Göster)

• • •

```
git remote add origin [GitHub Linki]
```

## Gönderme (Upload)

```
git push -u origin main
```

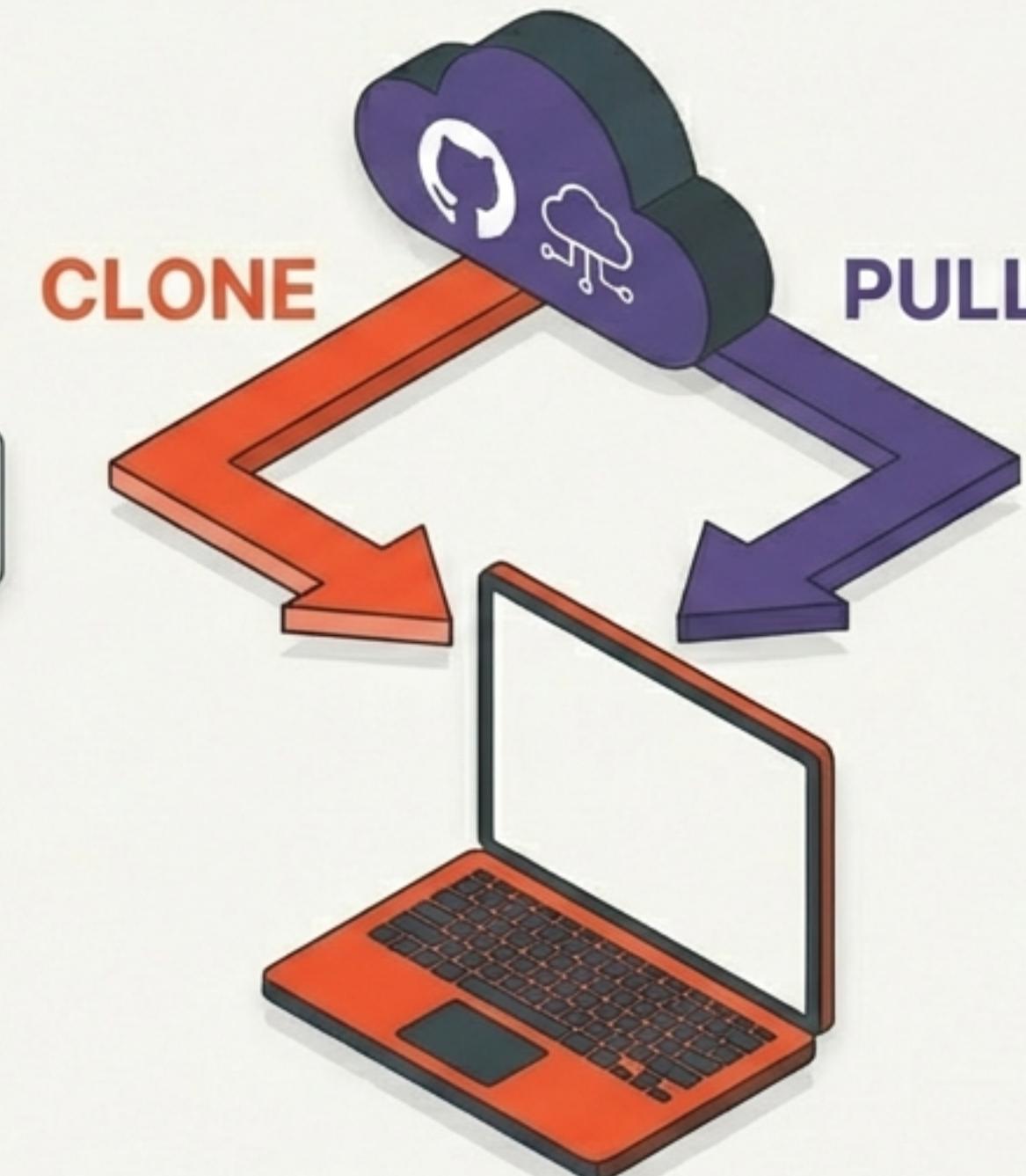
Not: İlk defa yapıyorsanız GitHub kullanıcı adı ve şifrenizi (veya Token) soracaktır.

# Verileri Çekmek — Clone ve Pull

## Clone (Kopyalamak)

```
git clone [Link]
```

GitHub'daki bir projeyi tüm tarihçesiyle bilgisayarınıza indirir. (Sıfırdan indirme).



## Pull (Güncellemek)

```
git pull
```

Mevcut projenizde çalışırken, başkasının yaptığı değişiklikleri çekip günceller.

Fetch (Getir) + Merge (Birleştir) = Pull

# Neleri Yüklememeliyiz? — .gitignore

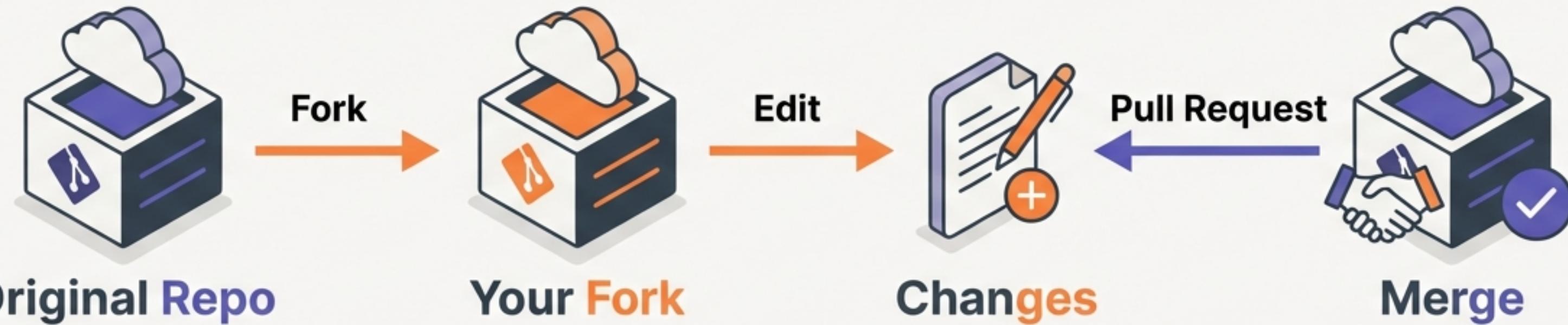


## .gitignore Dosyası

Git'e "Bunu takip etme" dediğimiz listedir. Her dosya GitHub'a yüklenmez.

- Bağımlılıklar (örn: node\_modules - çok ağırdir)
- Gizli Anahtarlar ve Şifreler (örn: .env)
- Sistem Dosyaları (örn: .DS\_Store)

# İşbirliği Kültürü — Pull Request (PR)

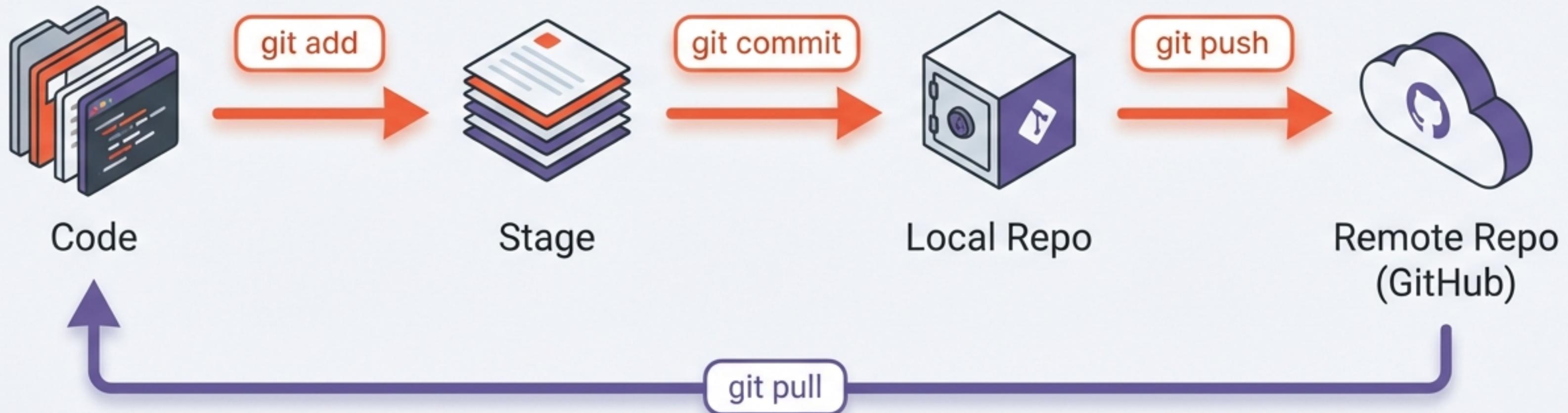


**Fork:** Projeyi kendi hesabınıza kopyalamak.

**Pull Request (PR):** "Ben bir değişiklik yaptım, inceleyip ana projeye ekler misin?" isteğiidir.

Bu yöntem, açık kaynak dünyasının kalbidir.

# Büyük Resim: Git Döngüsü



Kodla -> Sahnele -> Kaydet -> Paylaş

# Kodlarınız Güvende, İşbirliğiniz Sınırsız Olsun

## Nasıl Başlıyım?

1. Git'i indirin ([git-scm.com](http://git-scm.com)).
2. Bir GitHub hesabı açın.
3. Terminalden korkmayın, hata yapmaktan çekinmeyin.



## Kaynaklar

[Patika.dev & BTK Akademi](#)  
[GitHub Docs](#)  
[Learn Git Branching](#)

"Bir şeyi bozmaktan korkmayın, Git varsa her zaman geri dönebilirsiniz."