

- [Çözümlerimiz](#)
 - [Self Servis Teknolojiler](#)
 - [Mobil Teknolojiler](#)
 - [Web Teknolojileri](#)
- [Başarı Hikayeleri](#)
- [Değerlerimiz](#)
 - [Değerlerimiz](#)
 - [Ekibimiz](#)
- [Blog](#)
- [İletişim](#)
- [Akademi](#)



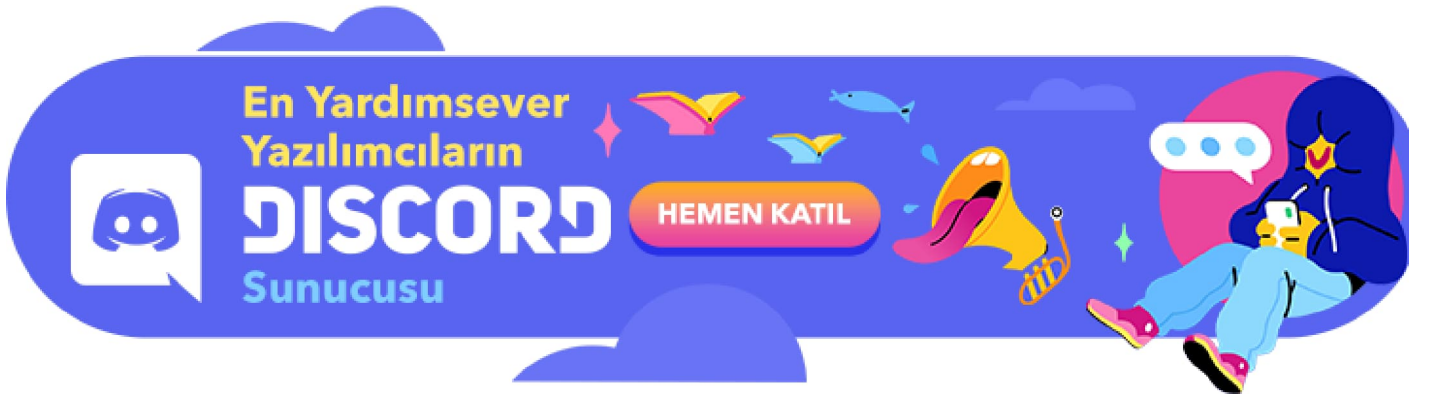
- [Çözümlerimiz](#)
 - [Self Servis Teknolojiler](#)
 - [Mobil Teknolojiler](#)
 - [Web Teknolojileri](#)
- [Başarı Hikayeleri](#)
- [Değerlerimiz](#)
 - [Değerlerimiz](#)
 - [Ekibimiz](#)
- [Blog](#)
- [İletişim](#)
- [Akademi](#)

[İnsan Kaynakları Yönetimi](#)[Argenova](#)[Yazılım Geliştirme](#)[Girişimcilik](#)[Proje Yönetimi](#)[Müşteri Hizmetleri](#)[Teknoloji](#)

En Çok Kullanılan Git Komutları (Örnekli)



[Caner BULUT • 18 Temmuz 2021](#)



Git ile komut satırında çalışmak göz korkutucu olabilir. Ama yazılım geliştirmede versiyon kontrol sistemi kullanmak kritik derecede önemlidir.

Eğer versiyon kontrol sistemi konusunda yeni iseniz [Versiyon Kontrol Sistemi Nedir?](#) başlıklı yazımızı incelemenizi öneriyoruz.

Yardımcı olmak için, ortak **Git komutlarının** bir listesini, her birinin ne anlama geldiğini ve nasıl kullanılacağını bir araya getirdik.

Bu yazımızda aşağıdaki git komutlarını ele alacağız.

- git config
- git init
- git clone
- git add
- git commit
- git diff
- git reset
- git status
- git rm
- git log
- git show
- git tag
- git branch
- git checkout
- git merge
- git remote
- git push
- git pull
- git stash

Git Komutları



git config

Bu komut, verdiğiniz değerler ile birlikte kullanılacak yazar adını ve e-posta adresini sırasıyla ayarlar.

Kullanımı

```
git config -global user.name "[isim]"  
git config -global user.email "[e-posta adresi]"
```

git init

Bu komut yeni bir depo başlatmak için kullanılır.

Kullanımı

```
git init [repository adı]
```

git clone

Bu komut, mevcut bir URL'den bir havuz elde etmek için kullanılır.

Kullanımı

```
git clone [url]
```

git add

Repoya bir dosya eklemek için kullanılır.

Kullanımı

```
git add [dosya-adi]
```

git commit

Bu komut, dosyayı sürüm geçmişinde kalıcı olarak kaydeder veya anlık görüntüler.

Kullanımı

```
git commit -m "[Commit mesajınız]"
```

git diff

Bu komut, henüz aşamalı olmayan dosya farklılıklarını gösterir.

Kullanımı

```
git diff
```

git reset

Bu komut dosyanın aşamasını kaldırır, ancak dosya içeriğini korur.

Kullanımı

```
git reset [dosya-adi]
```

git status

Bu komut, işlenmesi gereken tüm dosyaları listeler.

Kullanımı

```
git status
```

git rm

Bu komut, dosyayı çalışma dizininizden siler ve silme işlemini gerçekleştirir.

Kullanımı

```
git rm [dosya-adi]
```

git log

Bu komut, geçerli dalın sürüm geçmişini listelemek için kullanılır.

Kullanımı

```
git log
```

git show

Bu komut, belirtilen taahhüdün meta verilerini ve içerik değişikliklerini gösterir.

Kullanımı

```
git show
```

git tag

Bu komut, belirtilen işleme etiket vermek için kullanılır.

Kullanımı

```
git tag [commitID]
```

git branch

Bu komut, geçerli depodaki tüm yerel dalları listeler.

Kullanımı

```
git branch
```

Aşağıdaki komut yeni bir dal oluşturur.

```
git branch [branch name]
```

git checkout

Bu komut bir daldan diğerine geçmek için kullanılır.

Kullanımı

```
git checkout [branch name]
```

git merge

Bu komut, belirtilen dalın geçmişini geçerli dalla birleştirir.

Kullanımı

```
git merge [branch name]
```

git remote

Bu komut, yerel deponuzu uzak sunucuya bağlamak için kullanılır.

Kullanımı

```
git remote add [variable name] [Uzak Sunucu Linki]
```

git push

Bu komut, ana dalın taahhüt edilen değişikliklerini uzak havuzunuza gönderir.

Kullanımı

```
git push [variable name] master
```

git pull

Bu komut, uzak sunucudaki değişiklikleri çalışma dizininize getirir ve birleştirir.

Kullanımı

```
git pull [Repository Link]
```

git stash

Bu komut, değiştirilen tüm izlenen dosyaları geçici olarak saklar.

Kullanımı

```
git stash save
```

[Versiyon Kontrol Sistemi GitOps DevOps GitHub](#)

Daha Fazla Kaynak

- [GitHub Alternatifi 6 En İyi Kaynak Kod Yönetim Uygulaması](#)
- [Kodlama İçin En İyi 3 IDE](#)
- [Önemli Temiz Kod Yazma Pratikleri](#)
- [Versiyon Kontrol Sistemi Nedir?](#)

- [GitHub Nedir? \(Yeni Başlayanlar İçin\)](#)

Görüşlerini Paylaş

Görüşlerin ve yorumların bizim için çok değerli. Yorumunu aşağıdaki formu doldurarak bize gönderebilirsin.

Ad Soyad *

E-Mail *

Yorum *

E-mail adresiniz gösterilmeyecektir.

Gönder



Yenilikçi ve girişimci ruhumuz ile hareket ediyor, her işimizi değerlerimiz ışığında en doğru şekilde yapıyoruz.

Web Teknolojileri

- [Kurumsal Web Tasarım](#)
- [E-Ticaret](#)
- [Web Uygulamaları](#)
- [Seo](#)
- [Dijital Pazarlama](#)
- [Sunucu ve Barındırma](#)
- [Teknik Destek ve Bakım](#)

Mobil Teknolojiler

- [IOS Uygulama Geliştirme](#)
- [Android Uygulama Geliştirme](#)
- [Bakım ve Teknik Destek](#)

Self-Servis Teknolojiler

- [Ziyaretçi Yönetim Sistemi](#)
- [Anket ve Eğitim Sistemleri](#)
- [Toplantı Odası Bilgilendirme Ekranı](#)
- [Multitouch Uygulamalar](#)
- [Digital Signage](#)
- [İnteraktif Stand Çözümleri](#)
- [Özel Kiosk Çözümleri](#)

- [Temizlik Kontrol Sistemi](#)

Argenova

- [Hakkımızda](#)
- [Blog](#)
- [İletişim](#)
- [Argenova Yazılım Stajı 2022](#)
-
-

Copyright © 2022. [Argenova Teknoloji](#). All rights reserved.