

Git'e Giriş ve Temel Kavramlar

Dolunay Yazılım

Başlıklar

- Git Nedir?
- GitHub Hesabı Oluşturma
- Temel Git Terimleri
- Repository Yönetim Temelleri

Git Nedir?

Git, yazılım geliştiricilerin kod değişikliklerini takip etmelerine, işbirliği yapmalarına ve projelerini etkin bir şekilde yönetmelerine olanak tanıyan opensource bir araçtır.



git

Neden Git Kullanmalıyız?

- **Versiyon Kontrolü**

Git, versiyon kontrolü sağlayarak projelerinizi adım adım takip etmenizi ve değişiklikleri geri almanızı kolaylaştırır.

- **İşbirliği ve Takım Çalışması**

Projelerinizi başkalarıyla paylaşabilir, katkıda bulunmalarını sağlayabilir ve takım üyeleriyle etkileşimde bulunabilirsiniz.

- **Pull Request ve Değerlendirme**

Pull Request, değişikliklerinizi paylaşmanız ve diğerlerinin incelemesi için sunmanızı sağlar.

- **Issues ve Sorun Takibi**

Hataları, geliştirmeleri ve görevleri izlemek için Issues özelliği sayesinde sorunları organize edebilirsiniz.

GitHub Hesabı Oluşturma

<https://github.com/>

Let's build from here

The world's leading AI-powered developer platform.



Email address

Sign up for GitHub

Start a free enterprise trial >

Trusted by the world's leading organizations

3M

KPMG

Mercedes-Benz

SAP

P&G

TELUS



Productivity

Accelerate innovation

Our AI-powered platform increases the pace of software development.

Temel Git Terimleri

- **Repository**

Proje veya çalışmanın dijital depolama alanıdır. İçerisinde kod, dosyalar, belgeler ve tarihçe (history) bulunur.

- **Branch**

Ana kod tabanından ayrılan ve bağımsız olarak geliştirilebilen bir parça. Değişiklikleri izole etmek ve test etmek için kullanılır.

- **Clone**

Bir repoyu lokal bilgisayarınıza kopyalama işlemidir. Bu, repo üzerinde çalışırken lokaldeki değişiklikleri uygulama imkanı sağlar.

Repository Terimleri

- **Stage**

Belirtilen dosyaları "staging area"ya ekler. Bu dosyanın değişiklikleri bir sonraki committe yer alacaktır.

- **Commit**

Değişiklikleri repoya eklemek veya kaydetmek için kullanılan işlem. Genellikle bir değişiklik setini veya düzeltmeyi temsil eder.

- **Push**

Bu komut, yereldeki commitleri remote bir repoya gönderir. Bu, yapılan değişikliklerin paylaşılmasını sağlar.

- **Fetch**

Remote repodaki yeni branch'leri ve commit'leri lokale getirir, ancak çalışma dizinindeki dosyaları değiştirmez veya güncellemez.

Repository Terimleri

- **Pull**

Bu komut, remote repodaki değişiklikleri alır ve mevcut çalışma branch'imizi günceller.

- **Merge**

Bu komut, iki veya daha fazla branch'in değişikliklerini birleştirir ve birleşik değişiklikleri kaydeder.

- **Pull Request (PR)**

GitHub üzerinde bir geliştiricinin yaptığı değişiklikleri ana projeye eklemek istediğini belirten bir istektir.

Teşekkürler!

Terminal komutlari:

