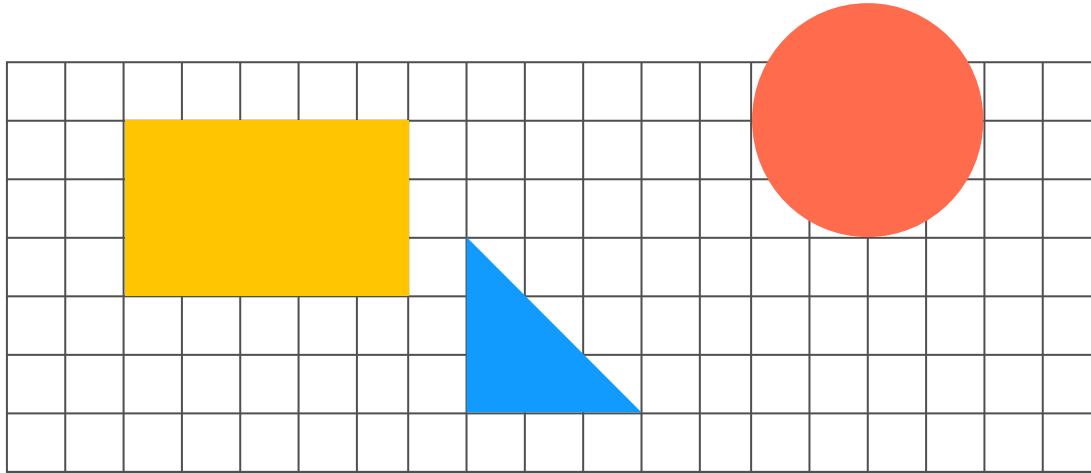


GİT VE GİTHUB



Git ve Github



Git ve Github

Git Nedir?

Git, yazılım projelerinde yapılan değişiklikleri kaydetmeye ve yönetmeye yarayan **bir versiyon kontrol sistemidir**.

Geri alma: Yanlış bir değişiklik yaptığında önceki sürümlere dönebilirsin.

Paralel çalışma: Bir ekip içinde farklı özellikler aynı anda geliştirilebilir.

Değişiklik takibi: Hangi kodun, ne zaman ve kim tarafından değiştirildiğini görebilirsin.

Git ve Github



GitHub Nedir?

GitHub, Git ile çalışan projelerini **internette saklayabileceğin ve başkalarıyla paylaşabileceğin** bir platformdur.

Kod paylaşımı: Açık kaynak projelerini yayınlayabilir veya özel projelerini yedekleyebilirsin.

Ekip çalışması: Takım üyeleri aynı proje üzerinde rahatça çalışabilir.

Git ve Github



Temel Git Komutları

Komut	Açıklama
<code>git init</code>	Yeni bir Git deposu oluşturur
<code>git clone <URL></code>	Mevcut bir depoyu bilgisayarına indirir
<code>git add .</code>	Değişiklikleri sahneye ekler
<code>git commit -m "Mesaj"</code>	Değişiklikleri kaydeder
<code>git push</code>	Güncellemeleri GitHub'a gönderir
<code>git pull</code>	Uzaktaki değişiklikleri çeker

Git ve Github Farkı



1. It is a service
2. It is hosted on Web
3. It provides a graphical interface
4. It is a space to upload a copy of the Git repository
5. It provides functionalities of Git like VCS, Source Code Management as well as adding few of its own features

1. It is a software
2. It is installed locally on the system
3. It is a command line tool
4. It is a tool to manage different versions of edits, made to files in a git repository
5. It provides functionalities like Version Control System Source Code Management

Git ve Github Farkı



Git ve GitHub Farkı

- ♦ **Git:** Yerel bilgisayarında çalışır, sürüm takibini yapar.
- ♦ **GitHub:** Git projelerini çevrimiçi depolar, iş birliğini kolaylaştırır.



Örnek Kullanım:

1. Projeyi başlat (**git init**)
2. Değişiklik yap ve kaydet (**git add .** → **git commit -m "Açıklama"**)
3. GitHub'a yükle (**git push**)
4. Güncellemeleri al (**git pull**)

GitHub Desktop Nedir ve Neden Kullanılır?



Git komut satırı kullanmadan GitHub işlemlerini yapmak



Daha görsel bir çalışma ortamı sunarak özellikle yeni başlayanlar için kolaylık sağlamak

GitHub Desktop Kurulumu:

- [GitHub Desktop](<https://desktop.github.com/>) resmi sitesinden indirme ve kurulum
- GitHub hesabıyla oturum açma
- Var olan bir repository'yi GitHub Desktop'a ekleme veya yeni bir repo oluşturma

Branch (Dal) Kavramı ve Branch Yönetimi

Branch Nedir?

- Ana kodu bozmadan yeni özellikler geliştirmek veya hata düzeltmek için kullanılan Git yapısı

Ana (main) branch ile çalışma mantığı

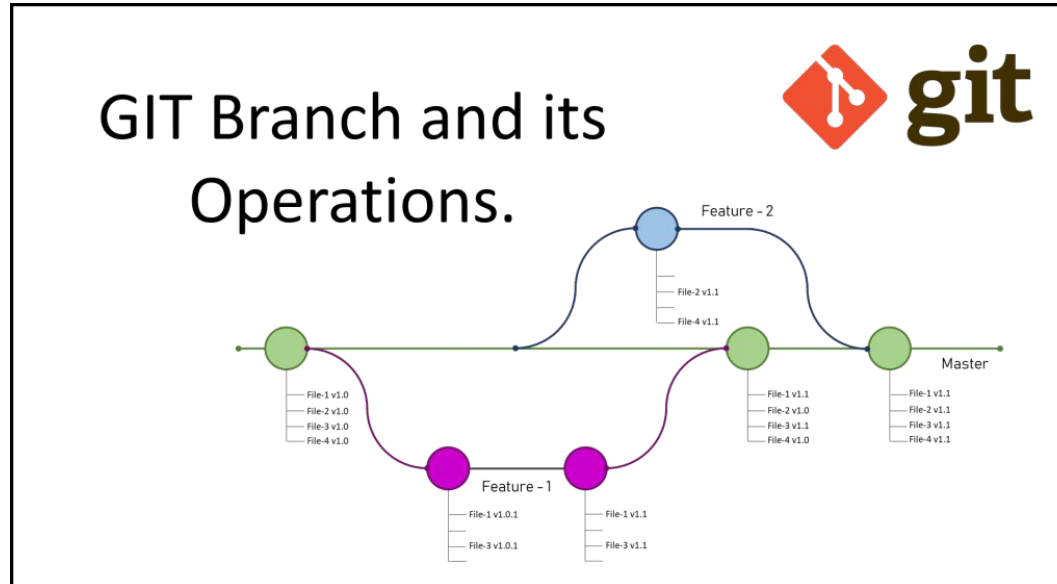
Yeni Bir Branch Oluşturma ve Geçiş Yapma:

```
# Yeni bir branch oluştur ve o branch'e geç  
git branch yeni-ozellik  
git checkout yeni-ozellik
```

GitHub Desktop ile Branch Kullanımı:

Yeni bir branch oluşturma ve branch'e geçiş yapma

Branch'te değişiklikler yapma ve commit atma



Branch'leri Birleştirme (Merge) ve Çatallaşma Çözme

Branch Birleştirme (Merge) Nedir?

 Yeni özellik veya hata düzeltmesini ana branch'e eklemek için kullanılır

Git Komut Satırı ile Merge Yapma:

```
# Ana branch'e geri dön  
git checkout main  
# Yeni branch'i ana branch'e birleştir  
git merge yeni-ozellik
```

Branch'leri Birleştirme (Merge) ve Çatallaşma Çözme

GitHub Desktop ile Merge Yapma:

GitHub Desktop'ta branch'i seçerek kolayca birleştirme

Olası merge çakışmalarının görsel olarak çözülmesi



Peki Merge Nedir ?