

Git Konfigürasyonu

```
PS C:\Users\Barış\.git> git config --global user.name "Adınız Soyadınız"
```

```
PS C:\Users\Barış\.git> git config --global user.email "email@example.com"
```

Yeni Bir Git Deposu Oluşturma

```
PS C:\Users\Barış> git init  
Initialized empty Git repository in C:/Users/Barış/.git/
```

Mevcut Bir Depoyu Klonlama

```
PS C:\Users\Barış\.git> git clone https://github.com/kullanıcı_adı/proje_adı.git
```

'git status'

'git status' komutu, çalışmakta olduğunuz dizindeki dosyaların durumunu gösterir. Hangi dosyaların değiştirildiğini, hangi dosyaların izlenmediğini ve hangi değişikliklerin commit edilmek üzere olduğunu görebilirsiniz

'git add' komutu, değişikliklerinizi bir sonraki commit'e ekler.

```
PS C:\Users\Barış\antalyabootcamp> git add style.css  
PS C:\Users\Barış\antalyabootcamp> git add index.html
```

'git commit' komutu, eklenen değişiklikleri kaydeder. Commit mesajı ekleyerek yapılan değişiklikleri açıklayabilirsiniz

```
PS C:\Users\Barış\antalyabootcamp> git commit -m "ikinci commit"
```

'git push' komutu, yerel depo değişikliklerini uzak depoya gönderir.

```
PS C:\Users\Barış\.git> git push origin master
```

'git pull' komutu, uzak depodaki de i iklikleri yerel depoya çeker ve günceller.

```
PS C:\Users\Barış\.git> git pull origin master
```

'git branch' komutu, mevcut branch'leri listelemek ve yeni bir branch olu turmak için kullanılır.

```
PS C:\Users\Barış\.git> git branch #branch listesini gösterir
PS C:\Users\Barış\.git> git branch branch_adı # yeni bir branch oluşturur
```

Branch Nedir?

Git'te branch, kod tabanının farklı versiyonlarını aynı anda geli tirmek için kullanılan bir yöntemdir. Her geli tirici, projede farklı özellikler veya düzeltmeler üzerinde çalı rken kendi branch'ini olu turabilir ve bu branch üzerinde çalı abilir. Ana branch (genellikle master veya main olarak adlandırılır) projedeki ana kod tabanını temsil ederken, yan branch'ler bu ana branch'ten ba ımsız olarak geli tirilebilir.

Branch'ler, aynı proje üzerinde birden fazla geli tiricinin paralel olarak çalı masını kolayla tırır. Örne in, bir geli tirici yeni bir özellik eklerken, bir di eri hata düzeltmeleri yapabilir. Bu sayede, her iki geli tirici de ana kod tabanına zarar vermeden kendi çalı malarını sürdürebilir.

'git checkout' komutu, belirli bir branch'e veya commit'e geçi yapmanızı sa lar.

```
PS C:\Users\Barış\.git> git checkout branch_adı #branch' e geçmek için kullanılır
```

```
PS C:\Users\Barış\.git> git checkout -b branch_adı #Yeni bir branch açmak için kullanılır
```

```
PS C:\Users\Barış\antalyabootcamp> git checkout ilkbranch
Already on 'ilkbranch'
PS C:\Users\Barış\antalyabootcamp> git branch
* ilkbranch
  main
PS C:\Users\Barış\antalyabootcamp> git checkout -b ilkbranch^C
PS C:\Users\Barış\antalyabootcamp> git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
PS C:\Users\Barış\antalyabootcamp> git branch
  ilkbranch
* main
PS C:\Users\Barış\antalyabootcamp> 
```