

Git Nedir? Nasıl Kullanılır?

Git, dağıtık bir versiyon kontrol sistemi olarak kullanılan bir araçtır. Projelerinizi takip etmek, değişiklikleri yönetmek ve işbirliği yapmak için kullanılır. Bu kısa kurs, Git'i kullanmaya başlamak için temel işlevleri açıklamak ve örneklerle anlatmak amacıyla hazırlanmıştır.

Git Kurulumu:

Git'i indirmek için resmi web sitesi olan <https://git-scm.com> adresini ziyaret edin. İndirme sayfasından işletim sisteminize uygun olan sürümü seçin ve indirin. İndirme tamamlandıktan sonra kurulum sihirbazını takip ederek Git'i bilgisayarınıza kurun.

Terminal Kullanımı:

"pwd" (Print Working Directory): Bu komut, kullanıcının mevcut çalışma dizinini (yani bulunduğu konumu) ekrana yazdırır.

"cd" (Change Directory): Bu komut, kullanıcının çalışma dizinini değiştirmesini sağlar. Kullanıcı, "cd" komutunu takip eden dizin yolunu belirterek hedeflenen dizine geçebilir.

"mkdir" (Make Directory): Bu komut, yeni bir dizin (klasör) oluşturmak için kullanılır. Kullanıcı, "mkdir" komutunu takip eden dizin adını belirterek yeni bir dizin oluşturabilir.

"touch": Bu komut, boş bir dosya oluşturmak için kullanılır. Kullanıcı, "touch" komutunu takip eden dosya adını belirterek yeni bir dosya oluşturabilir.

```
user@DESKTOP-HMONPQO MINGW64 ~ (master)
$ pwd
/c/Users/user

user@DESKTOP-HMONPQO MINGW64 ~ (master)
$ mkdir deneme2

user@DESKTOP-HMONPQO MINGW64 ~ (master)
$ cd deneme2

user@DESKTOP-HMONPQO MINGW64 ~/deneme2 (master)
$ touch python.py

user@DESKTOP-HMONPQO MINGW64 ~/deneme2 (master)
```

Bu örnekte "pwd" komutu ile içinde bulunduğumuz klasörü görüntüledik , bu klasörün içinde "mkdir" komutu ile deneme2 adında bir klasör oluşturduk , "cd" komutu ile deneme2 klasörünün içine girdik ve "touch" komutu ile de deneme2 klasörünün içinde bir python dosyası oluşturduk.

Yeni Bir Git Deposu Oluşturma:

Projelerinizi saklamak için Git deposu (repository) oluşturmanız gerekmektedir. Depo oluşturmak için komut satırını açın ve projenizin bulunduğu dizine gidin. Ardından **git init** komutunu kullanarak yeni bir Git deposu oluşturun:

```

user@DESKTOP-HMONPQ0 MINGW64 ~/deneme (master)
$ git init
Initialized empty Git repository in C:/Users/user/deneme/.git/

user@DESKTOP-HMONPQ0 MINGW64 ~/deneme (master)
$ ls -la
total 16
drwxr-xr-x 1 user 197121 0 May 29 11:58 ./
drwxr-xr-x 1 user 197121 0 May 29 11:54 ../
drwxr-xr-x 1 user 197121 0 May 29 11:58 .git/ *
-rw-r--r-- 1 user 197121 0 May 29 11:54 deneme.txt

```

Değişiklikleri İzleme:

Git, projenizdeki değişiklikleri izlemek için "**git status**" komutunu kullanır.

```

user@DESKTOP-HMONPQ0 MINGW64 ~/deneme (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    deneme.txt

nothing added to commit but untracked files present (use "git add" to track)

```

(Bu örnekte deneme klasörünün içinde deneme.txt adında bir dosya oluşturulmuştur.)

Komutu çalıştırdığınızda, değişiklikleri yapan dosyaların bir listesi görüntülenir.

Değişiklikleri eklemek için "**git add**" komutunu kullanabilirsiniz. Örneğin:

```

user@DESKTOP-HMONPQ0 MINGW64 ~/deneme (master)
$ git add deneme.txt

```

git add dosya1.txt dosya2.txt veya **git add .** ile tüm değişiklikleri ekleyebilirsiniz.

Değişiklikleri Kaydetme:

Eklediğiniz değişiklikleri kaydetmek için "git commit" komutunu kullanmanız gerekmektedir.

Aşağıdaki örnek komutu kullanarak değişiklikleri bir açıklama ile kaydedebilirsiniz:

```

user@DESKTOP-HMONPQ0 MINGW64 ~/deneme (master)
$ git commit -m "deneme.txt oluşturuldu"
[master (root-commit) 15fa379] deneme.txt oluşturuldu
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 deneme.txt

```

git commit -m "Yeni özellik eklendi"

"**git log**" komutu commit işlemleri ile ilgili özet bilgileri görmek için kullanılır.

```
user@DESKTOP-HMONPQ0 MINGW64 ~/deneme (master)
$ git log
commit 91dcc3b20e8e5731d97ce7b4d42fd9581df69723 (HEAD -> master)
Author: Ali Ure <ureali90@gmail.com>
Date: Mon May 29 12:13:38 2023 +0300

    python dosyası oluşturuldu

commit 15fa3791ef88480310dc07048e84cc7e88d940c0
Author: Ali Ure <ureali90@gmail.com>
Date: Mon May 29 12:10:51 2023 +0300

    deneme.txt oluşturuldu
```

(Bu örnekte iki defa yapılan commit işlemi mesajlarıyla birlikte görüntülenir.)

Uzak Bir Sunucuya Bağlanma:

Projelerinizi paylaşmak ve işbirliği yapmak için uzak bir Git sunucusuna bağlanmanız gerekmektedir.

GitHub gibi popüler bir Git sunucusu kullanabilirsiniz.

Projeyi uzak sunucuya göndermek için "git push" komutunu kullanabilirsiniz. Örneğin:

git push origin ana-dal

Dal (Branch) Oluşturma ve Birleştirme:

Dal, projenizi farklı parçalara bölerek ayrı ayrı çalışmanıza olanak sağlar. Mevcut dalları görüntülemek veya Yeni bir dal oluşturmak için "git branch" komutunu kullanabilirsiniz. Örneğin:

```
user@DESKTOP-HMONPQ0 MINGW64 ~/deneme (master)
$ git branch dal2

user@DESKTOP-HMONPQ0 MINGW64 ~/deneme (master)
$ git branch
dal2
* master
```

(Bu örnekte dal2 adında ayrı bir dal oluşturulmuş ve mevcut dallar görüntülenmiştir.)
git branch dal2

"git switch" komutu ile farklı bir dala geçiş yapabilirsiniz.

```
user@DESKTOP-HMONPQ0 MINGW64 ~/deneme (master)
$ git switch dal2
Switched to branch 'dal2'

user@DESKTOP-HMONPQ0 MINGW64 ~/deneme (dal2)
$ git branch
* dal2
master
```

(Bu örnekte git switch kullanılarak farklı bir dala geçilmiş ve git branch ile görüntülenmiştir.)

Farklı dalları birleştirmek için "git merge" komutunu kullanabilirsiniz. Örneğin:
git merge dal2

```
user@DESKTOP-HMONPQO MINGW64 ~/deneme (master)
$ git merge dal2
Updating 91dcc3b..491e8da
Fast-forward
 dal2.txt | 0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 dal2.txt
```

(Bu örnekte dal2 adındaki dal anadal ile birleştirilmiştir.)

Geçmişteki Değişiklikleri İnceleme:

- Geçmişteki değişiklikleri görmek için "git log" komutunu kullanabilirsiniz.
- Değişiklikleri detaylı olarak incelemek için "git show" komutunu kullanabilirsiniz.