

En sık kullanılan ve en temel Git işlemleri nelerdir bir örnek üzerinden inceleyelim. Bilgisayarınıza Git'i kurmuş olmanız gerekiyor. [link](#) üzerinden indirebilirsiniz. Bu yazının içeriği şöyle

1. İlk Commit : Yeni Bir Dosya Ekleyelim
2. İkinci Commit : Dosyada Değişik Yapıp Versiyonlayalım
3. Git Dosyaları Nasıl Takip Ediyor?

İlk Commit : Yeni Bir Dosya Ekleyelim

- Bir klasör açın mesela 'myrepo' isminde ve içine bir txt dosyası oluşturun mesela 'program.txt'. Dosyaya bir metin yazıp kaydedin.

```
print("merhaba")
```

- Git Bash'i klasör içinde sağ tıklayarak başlatın ve aşağıdaki komutla Git'i başlatın

```
git init
```

Böylece .git isminde bir klasör oluşacak ve Git'e dair ne varsa bu klasör üzerinden yürütülecek.

- Şimdi aşağıdaki komutu çalıştırarak dosyamızı commitlemek üzere ekleyelim.

```
git add .
```

- Eklediğimiz dosyayı bir commit mesajı yazarak commitleyelim.

```
git commit -m "ilk commit: program.txt oluşturuldu."
```

- Yaptığımız değişiklik artık Git tarafından kaydedildi, görmek için aşağıdaki komutu kullanabiliriz.

```
git log
```

```

MINGW64/c/Users/canen/Desktop/enes/gitDoc/myrepo
can@DESKTOP-CHER608 MINGW64 /c/Users/canen/Desktop/enes/gitDoc/myrepo
$ git init
Initialized empty Git repository in C:/Users/canen/Desktop/enes/gitDoc/myrepo/.git/

can@DESKTOP-CHER608 MINGW64 /c/Users/canen/Desktop/enes/gitDoc/myrepo (master)
$ git add .

can@DESKTOP-CHER608 MINGW64 /c/Users/canen/Desktop/enes/gitDoc/myrepo (master)
$ git commit -m "ilk commit: program.txt oluřturuldu."
[master (root-commit) b9c4cbd] ilk commit: program.txt oluřturuldu.
1 file changed, 1 insertion(+)
create mode 100644 program.txt

can@DESKTOP-CHER608 MINGW64 /c/Users/canen/Desktop/enes/gitDoc/myrepo (master)
$ git log
commit b9c4cbd518468dfedeeb2451a2a448de35791809 (HEAD -> master)
Author: H Enes Simsek
Date: Thu Jun 2 16:55:18 2022 +0300

    ilk commit: program.txt oluřturuldu.

```

İkinci Commit : Dosyada Deęişik Yapıp Versiyonlayalım

- program.txt dosyasındaki metni deęiřtirelim kaydedelim.

```
print("merhaba dünya")
```

- Őimdi **git status** ile durumuna bakalım. Git, bize program.txt'nin deęiřtięini sylyor. Dosyamız **kırmızı fontla modified** olarak iřaretlenmiř.

```

can@DESKTOP-CHER608 MINGW64 /c/Users/canen/Desktop/enes/gitDoc/myrepo (master)
$ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
       modified:   program.txt

no changes added to commit (use "git add" and/or "git commit -a")

```

- Deęiřiklikleri eęer commitlemek istiyorsak dosyayı commite gndermek zere hazırlamalıyız. Daha sonra tekrar son duruma bakalım. Dosyamız Őimdi **yeřil fontla modified** olarak iřaretli.

```
git add .
git status
```

```
can@DESKTOP-CHE8608 MINGW64 /c/Users/canen/Desktop/enes/gitDoc/myrepo (master)
$ git status
On branch master
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   program.txt
```

- Dosyamız commit için hazır, şimdi commitleyelim. Ve versiyon geçmişine bakalım.

```
git commit -m "merhaba dünya değişikliği yapıldı."
git log
```

```
can@DESKTOP-CHE8608 MINGW64 /c/Users/canen/Desktop/enes/gitDoc/myrepo (master)
$ git log
commit 16513a3be61d816eed6b8ae4d2c4b70198c7f946 (HEAD -> master)
Author: H Enes Simsek <enes1406@gmail.com>
Date: Thu Jun 2 17:18:56 2022 +0300

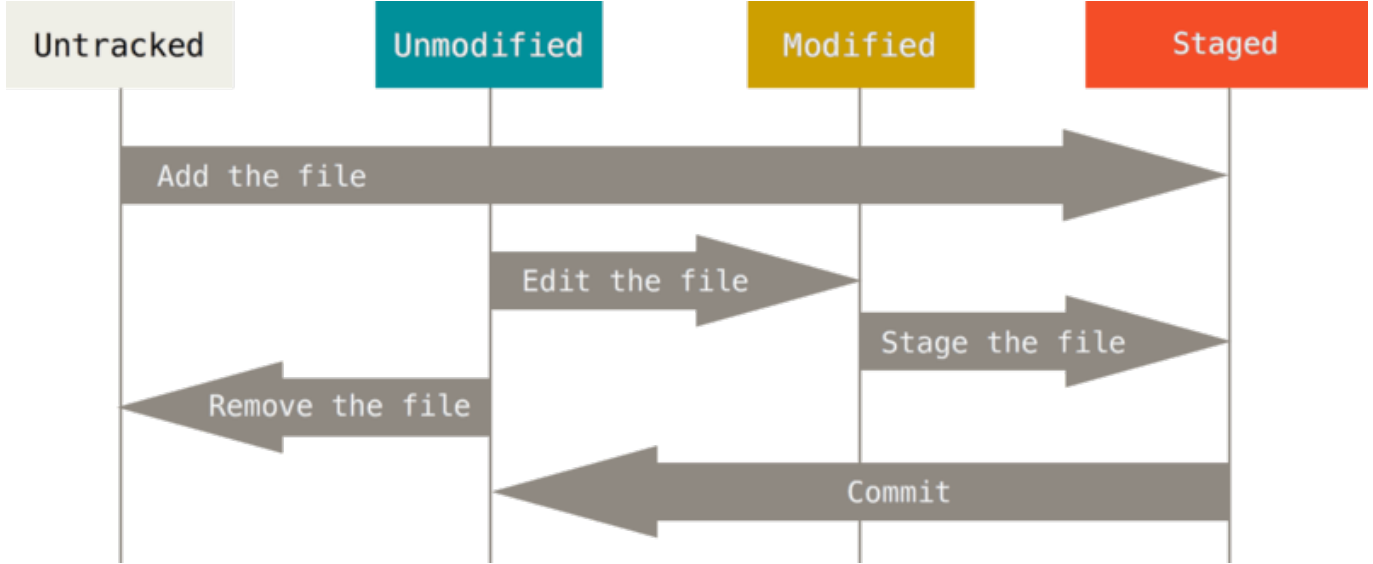
    merhaba dünya değişikliği yapıldı.

commit 51c09bc2654894141f449a31402cd001b43921f3
Author: H Enes Simsek <enes1406@gmail.com>
Date: Thu Jun 2 17:12:33 2022 +0300

    ilk commit: program.txt oluşturuldu.
```

Git Dosyaları Nasıl Takip Ediyor?

İki commit yaptık ancak gerçekleşen olayı daha detaylı inceleyelim. Git çalışma alanımızdaki dosyalar Git tarafından takip edilir. Dosyanın ilk oluşturulduğu andan commite kadar geçen döngüsü aşağıdaki gibidir.



[kaynak](#)

Yeni bir dosyayı ilk oluşturduğumuzda git tarafından **untracked** statüsünü alır.

Dosyayı Git'in takip etmesi için **git add** komutunu kullanırız. Daha sonra **git commit** ile dosyaları commitleriz.

İlk committen sonra dosyamız artık git tarafından takip edilir ve **unmodified** statüsündedir. Dosya üzerinde değişiklik yaptığımızda otomatik olarak **modified** statüsüne geçer.

Ancak dosya üzerinde değişiklik yapsak bile bunu commitlemek istemeyebiliriz. Bundan dolayı commite gidecek dosyaları tek tek eklemek gerekir. **git add** ile commite gidecek dosyaları seçeriz. Seçilen dosyalar **staged** statüsünü alır.

git commit işlemi yaptığımızda staged statüsündeki dosyalar versiyonlanır. Artık yeni versiyona geçtiğimiz için staged statüsündeki dosyalar unmodified statüsüne geçer.

Bu döngü içerisinde dosyalar versiyonlanır.

Not: **git add program.txt** sadece ismi verilen dosyayı staged kısmına alır. **git add .** ise tüm dosyaları staged kısmına alır.