Bu yazımda versiyon değiştirme işlemi yapacağız daha daha önceki yazımda oluşturmuş olduğumuz “lokaldepo” klasörü üzerinden ilerleyeceğiz.Terminal üzerinden “lokaldepo” dizinine gidelim.

nano ile bir dosya daha oluşturalım.bunun adı “dosya.txt” olsun.

```
nano dosya.txt
```

“dosya.txt” dosyasının içine “test1” yazısını girip , ctrl+s ile kaydedip ctrl+x ile nano dan çıkıyoruz.

şimdi sırası ile daha önceki derslerimizde görmüş olduğumuz adımları tekrar edip değişiklikleri commit edelim.

```
git status
```

komutu ile değişiklikleri görelim.

```
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        dosya.txt
nothing added to commit but untracked files present (use "git add" to
track)
```

Sonuç yukarıdaki gibi “dosya.txt” dosyasının **index** bölümüne eklenmesi gerektiğini

anlatmakta.Index bölümüne klasörümüzde ki tüm değişiklikleri aktarmasını sağlayalım ;

```
git add .
```

Daha sonra dilerseniz “git status” komutu ile değişiklikleri görebilirsiniz.Şimdi anlatımlarım boyunca 2. commit işlemimizi gerçekleştirelim.

```
git commit -m "ikinci versiyonlama - dosya.txt oluşturuldu"
```

sonucu görelim ;

```
[master 8eef6e6] ikinci versiyonlama - dosya.txt oluşturuldu
 1 file changed, 1 insertion(+)
 create mode 100644 dosya.txt
```

Commit işlemimiz başarılı.Şimdi loglara bakalım ve commit ettiklerimizi görelim ;

```
git log
```

Sonuç ;

```
Author: Serkan Kaya <serkankayatr@hotmail.com>
Date:   Fri Feb 15 13:09:16 2019 +0300
    ikinci versiyonlama - dosya.txt oluşturuldu
commit 3b87c818004c43f4a3674e774c457b6102ea838a
Author: Serkan Kaya <serkankayatr@hotmail.com>
Date:   Fri Feb 15 11:05:17 2019 +0300
    ilk sürüm
```

Sonuca baktığımızda daha önce de bahsettiğim üzere 2 adet commit işlemi tamamlanmış

yani versiyonlanmış çalışma görmekteyiz …

Şimdi versiyon değiştirme amacı ile bulunduğumuz klasörde bulunan “dosya.txt” dosyamızı silelim ve tekrar commit edelim … sırası ile aşağıdaki komutları girelim ;

```
rm dosya.txt
git add .
git commit -m "3. versiyon - dosya.txt dosyası silindi"
```

Yukarıdaki komutlar ile dosyamızı sildik ve versiyonladık.

“dosya.txt” dosyasını geri getirmek istediğimizi düşünelim … bu durumda checkout komutu imdadımız yetişiyor.versiyon değiştirme sırasında **commit id** sine ihtiyacımız var.Bu id nin ister ilk 7 karakterini(en az) istersek tamamını girerek eski versiyonlardan birine geçiş yapabiliriz.
Geçiş yapmak istediğimiz commit id sini öğrenmek için aşağıdaki kodu yazalım.

```
git log
```

komut sonucu ;

```
Author: Serkan Kaya <serkankayatr@hotmail.com>
Date:   Fri Feb 15 13:19:28 2019 +0300

    3. versiyon - dosya.txt dosyası silindi

commit 8eef6e64f465b2454798a16fe4c14032520d8f68
Author: Serkan Kaya <serkankayatr@hotmail.com>
Date:   Fri Feb 15 13:09:16 2019 +0300

    ikinci versiyonlama - dosya.txt oluşturuldu

commit 3b87c818004c43f4a3674e774c457b6102ea838a
Author: Serkan Kaya <serkankayatr@hotmail.com>
Date:   Fri Feb 15 11:05:17 2019 +0300
```

```
    ilk sürüm
```

loglardan ikinci versiyonu seçelim ve buradaki **commit** ile başlayan commit id sini kopyalayalım.Şimdi **checkout** komutunu kullanalım;

```
git checkout 8eef6e64f465b2454798a16fe4c14032520d8f68 -- .
```

"lokaldepo" klasörüne gittiğinizde "dosya.txt" dosyasının geri geldiğini göreceksiniz git versiyonlama sisteminin versiyon değiştirme işlemini de görmüş olduk

Ek bilgi : istersek sadece tek bir dosyayı eski versiyonumuzdan çekebiliriz.Örnek olarak aşağıdaki kodu uygulayabiliriz ;

```
git checkout -- dosya.txt
```

dosya.txt gibi bir boşluk bırakarak birden fazla belirlediğimiz dosyayı da eski versiyondan çekebiliriz …

Umarım faydalı olmuştur

Serkan Kaya
Full Stack Java Developer