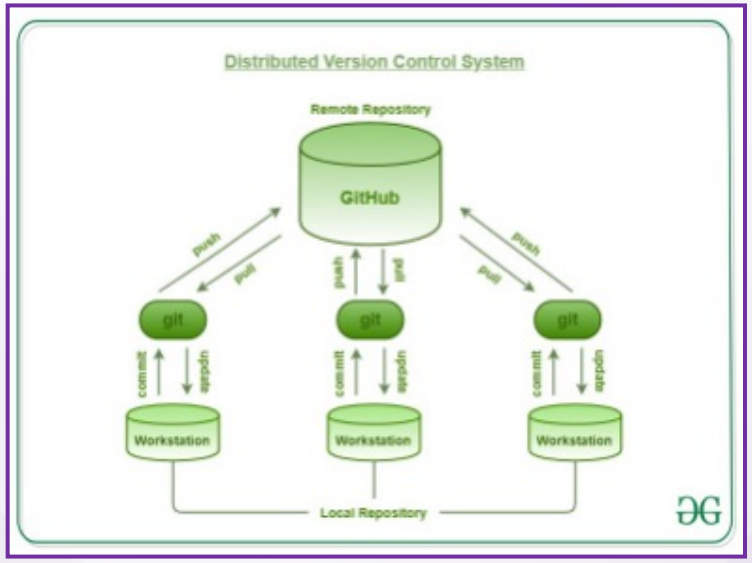
Wise Quarter
first class IT courses

Git – GitHub

+1 912 888 1630
www.wisequarter.com
/wisequarter

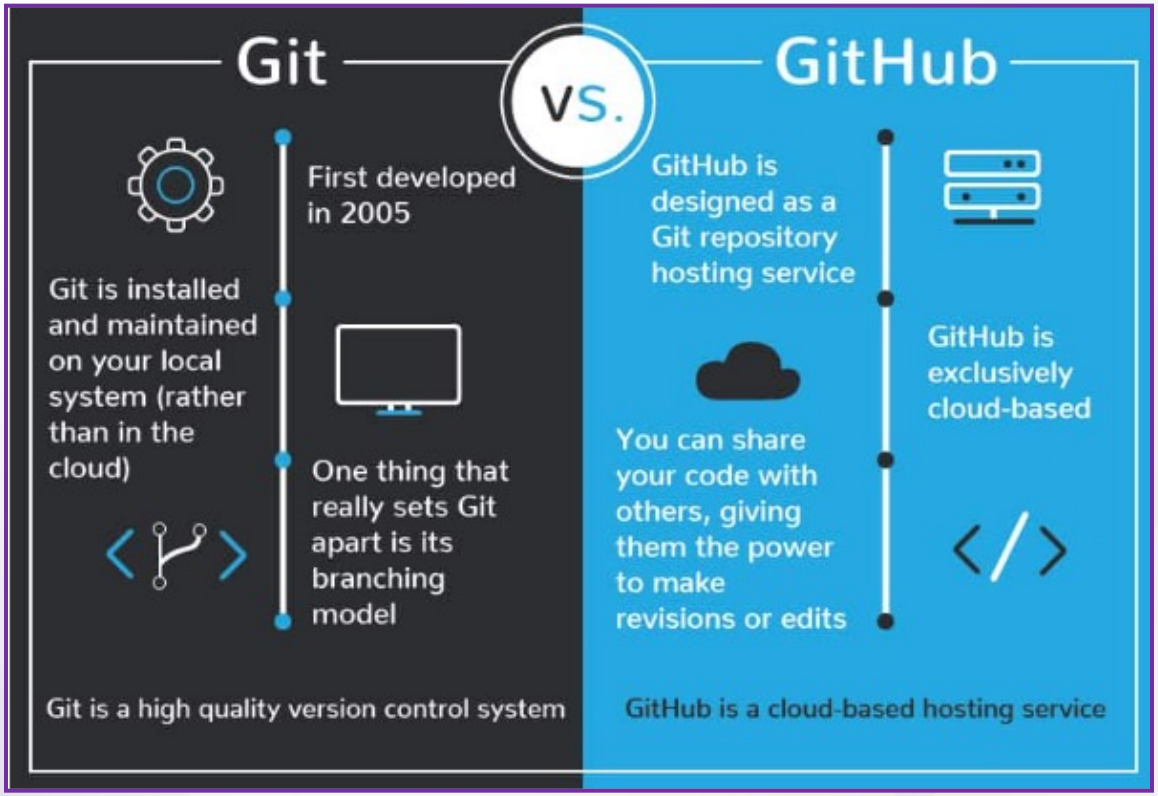
GitHub

GitHub, Git isimli bir version control system(sürüm kontrol sistemi) barindiran cloud tabanli bir uygulamadir.



Git, bireysel küçük projelerden, ekip olarak calisilan çok büyük projelere kadar her şeyi hızlı ve verimli bir şekilde yonetmek için tasarlanmış ücretsiz ve açık kaynaklı, dağıtılmış bir sürüm kontrol sistemidir(distributed version control system).

Git & GitHub



Git, kaynak kod geçmişinizi yönetmenize ve takip etmenize izin veren bir version kontrol sistemidir.

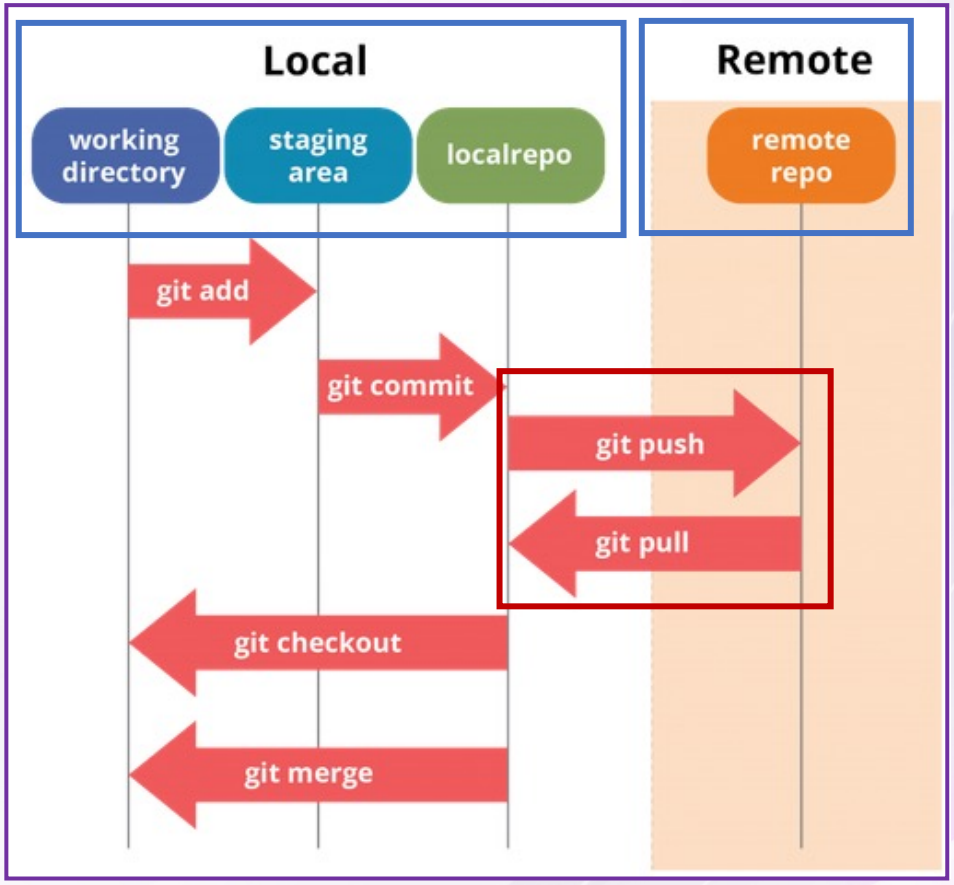
GitHub ise, Git depolarını(repo) yönetmenize izin veren bulut tabanlı bir barındırma(hosting) hizmetidir.

Tekli Kullanim

Tekli kullanimda her bir projemiz icin GitHub'da bir repo olusturup, bunu istedigimiz araliklarla bilgisayarimiz veya GitHub'daki hali uzerinden guncelleyebiliriz.

Github'da yaptigimiz degisiklikleri her kaydettigimizde (commit) Git yeni bir version olusturur.

Versiyon kontrol sistemlerinin en buyuk avantaji, istedigimiz bir anda versiyonlara ulasabilmektir.

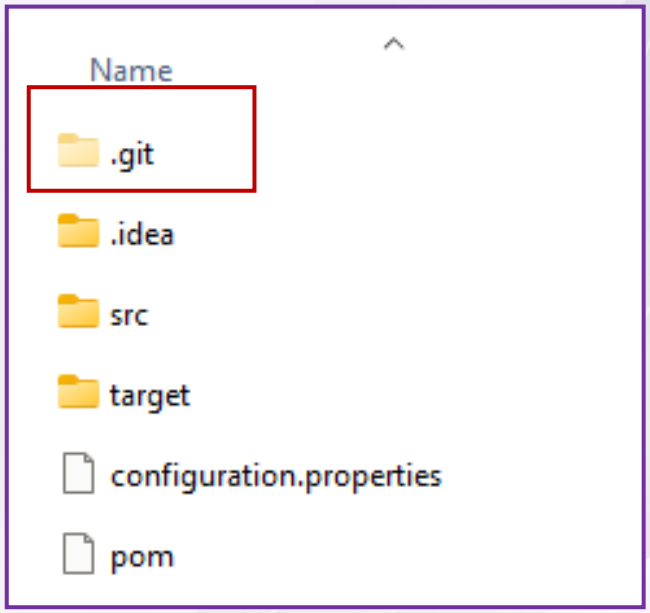


Tekli Kullanim

Her degisiklik yapildiginda tum projenin bir kopyasini alip kaydetmek cok buyuk storage gerektirir.

Git her version'da projenin tamamini degil, bir onceki versiyona gore degisen kodlari store eder.

Git version kontrolunu klasor uzerinden yapar, biz IntelliJ gibi bir ide'de calissak da git o projenin bilgisayarda kaydedildigi klasor uzerinden calisir.



Git version kontrolunu yapmak icin Git'e tanittigimiz (init) her klasore .git isminde bir klasor olusturur ve tum versiyonlar bu dosyada tutulur. Bu dosya silinirse tum versiyonlar da silinir.

.git dosyasini gormek icin bilgisayarinizda "Gizli dosyalari goster" seceneginin acik olmasi gerekir. (Mac icin yukari ok + command + .)

Git Komutlari

Git bilgisayarlarimizdaki windows benzeri uygulamalarla degil, bilgisayarlarin ilk kullanilmaya basladigi donemdeki gibi, siyah ekrana(command promt / terminal) yazilan komutlarla calisir.

Gunumuzde GitHub Desktop gibi uygulamalar Command Prompt ihtiyaci olmadan da calismaniza izin verse de version kontrolu icin temel git komutlarini kullanmayi bilmelisiniz.

Git ve command prompt'daki dosya kullanim komutlarini ezberlemek zorunda degilsiniz. Google ile basit bir arama yaptiginizda tum komutlar ve ne ise yaradiklari karsiniza cikacaktir.

Cmd / Terminal Komutlari

Git bilgisayarlarimizdaki windows benzeri uygulamalarla degil, bilgisayarlarin ilk kullanilmaya basladigi donemdeki gibi, siyah ekrana(command promt / terminal) yazilan komutlarla calisir.

	Windows	Mac
Islem	Komut Istemi	Terminal
Klasor hareketleri	cd	
Listeleme	dir	ls
Ekran temileme	cls	clear
Klasör Oluşturma	mkdir	
Klasör Silme	rmdir	
Dosya oluşturma	echo > dosyaAdi.uzantisi	
Dosyanın içini görme	more	cat
Dosya silme	del	rm
Klasör ve dosya ismi değiştirme	ren	mv

Temel Git Komutlari

git --version : Bilgisayarlarimizda git kurulu ise versiyonunu verir

git config --global color.ui true : Terminalde kullanilan komutlari renklendirir.

git config --global user.name “John Doe”: Yaptiginiz islemlerde gorunecek ismi tanimlar.

git config --global user.mail “a@b.com”: Yaptiginiz islemlerde gorunecek email’i tanimlar.

NOT : -’den sonra global kullanilirsa o kullanicinin tum repolari icin tanimlar
system kullanilirsa, tum kullanicilar ve tum repolar icin tanimlar
local kullanilirsa, sadece o an kullanilan repo icin tanimlar

Local Repo Olusturma

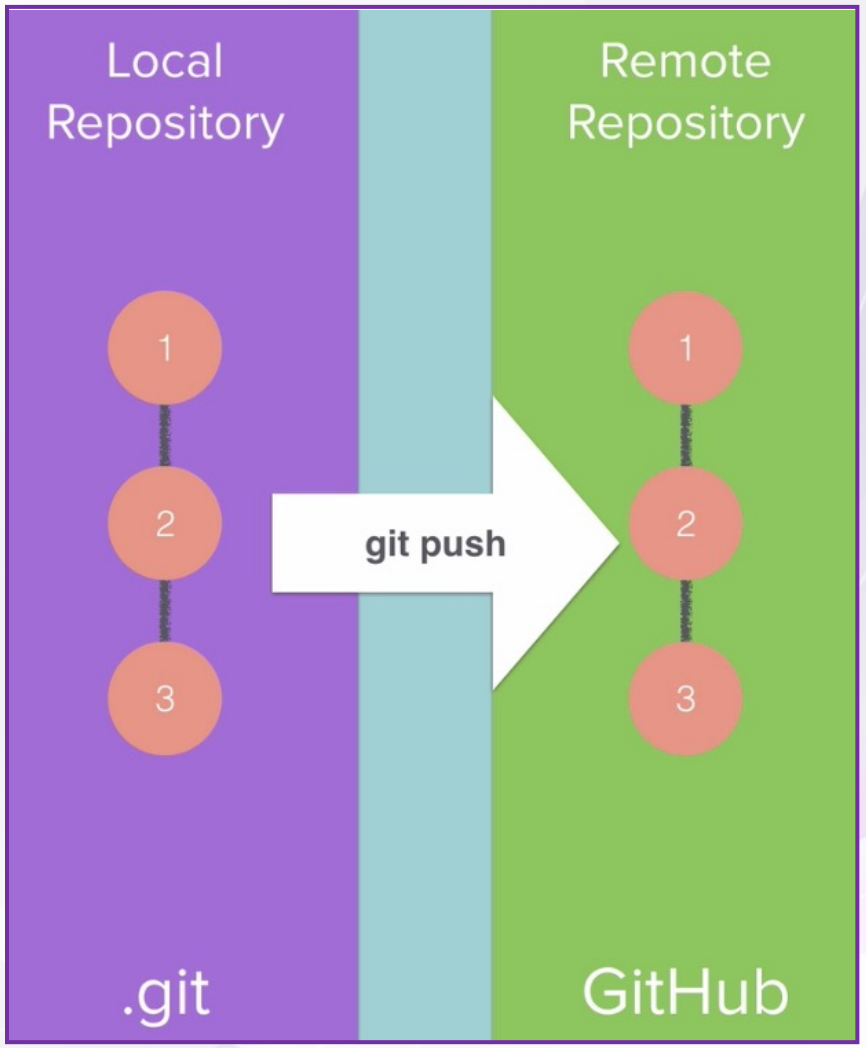
GitHub kullanmasak bile Git ile tum projelerimizi local'de versiyon kontrol sistemi ile kullanabiliriz.

Bir klasorde git ile versiyon kontrolu yapmak istersek, terminalde o klasore gidip

```
git init
```

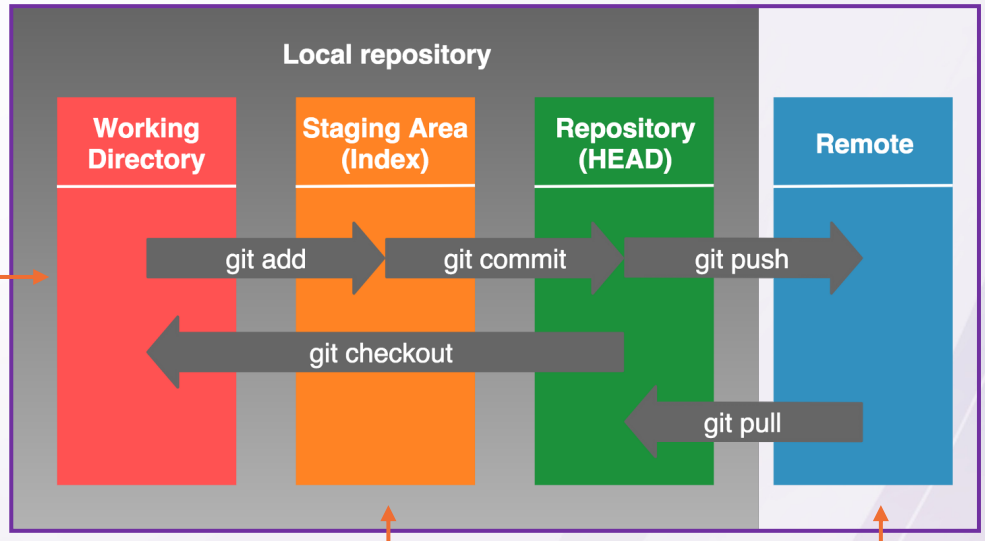
komutu calistirilmalidir.

git init komutu calistirilinca git o klasorde hidden .git klasoru olusturur ve bu klasor uzerinden versiyon kontrolu yapar.



Remote Repo'ya Yollama

Working Space (git init)
.git dosyasinin bulundugu klasordur.



Working space'de yapilan degisiklikler git add komutu ile Staging Area'ya yuklenir.

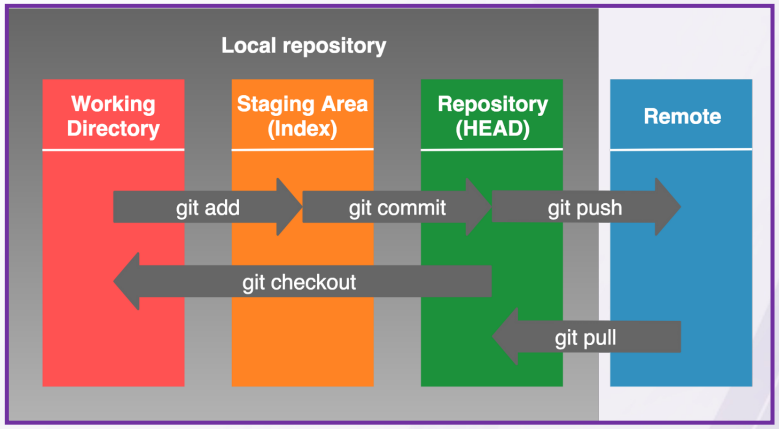
Staging Area'daki dosyalar git commit komutu ile gonderilmeye hazir hale getirilir

Remote repo'ya ilk yukleme icin, gitHub'da remote repo olusturulmali ve local ile remote repo eslestirilmelidir.

Daha once gitHub'daki remote repo ile eslestirilmis olan projeler icin, versiyonu remote repo'ya yollamak istedigimizde git push yeterli olur.

Remote Repo'ya Yollama

Kisisel kullanimda yapacagimiz tum islemler, kendi bilgisayarimizdaki local repository ile GitHub'daki Remote Repository arasinda senkronizasyondan ibarettir.



1. Bilgisayarimizdaki bir dosyayi ilk defa Git'e tanitip, Github'da olusturacagimiz bir repo'ya yukleme

- dosyalar git'e ekleyip working space olusturma git init
- Dosyalari staging area'ya gonderme git add.
- Versiyon olusturup, local repository'e gonderme git commit –m "commit ismi"

GitHub'da remote repo olustur.

- Remote repo'daki branch'i tanimla git branch –M main
- Remote repo ile local repoyu eslestirme git remote add origin "remote repo url"
- local repodaki dosyalari remote repoya yollama git push –u origin main

Wise Quarter

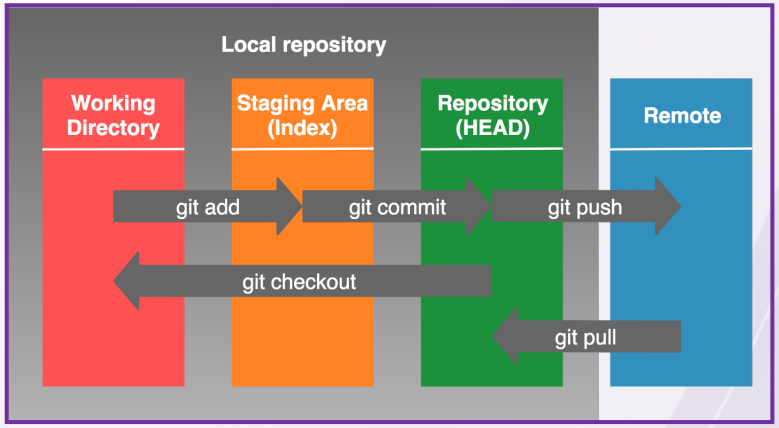
first class IT courses

Git – GitHub 2

+1 912 888 1630
www.wisequarter.com
/wisequarter

Remote Repo'ya Yollama

2. GitHub'da begendigimiz ve erisim yetkimiz olan bir repoyu bilgisayarimiza indirme



- Dosyalari yuklemek icin bir klasor olusturup, git'e ekleme git init
- GitHub'da begendigimiz reponun url'ini kopyalayip, remote repoyu local repoya indirmek icin git pull remoteRepoUrl

  IntelliJ'de bu islem icin new project menusunden url ile proje acma secilir.

NOT : Eger olusturdugumuz local repo ile remote repoyu bundan sonra da senkronize edeceksek git pull yaptiktan sonra local'de commit olusturup, remote repoya baglayalim.

- Remote repo'daki branch'i tanimla git branch –M main
- Remote repo ile local repoyu eslestirme git remote add origin "remote repo url"
- local repodaki dosyalari remote repoya yollama git push –u origin main

Local Repo'da Commit Gecmisi

Git ile yaptigimiz tum version takibi local repoda tutulmaktadir.

Commit yaptigimiz versiyonlar arasinda gezinebilir, istedigimiz commit'e gecebilir ve istersek o versiyon'u projemizin son hali sekline getirebiliriz.

- Local repoda yaptigimiz tum commit'eri listelemek icin git log

```
commit 01867b98e9cb888ba78dd16dd52e65abd2cf55d3 (HEAD -> main)
Author: ahmet bulutluoz <abbulutluoz@gmail.com>
Date:   Sun Sep 25 07:06:03 2022 +0200

    C03

commit 03c8adde7006b96293a52990e03831af7c8bc73b
Author: ahmet bulutluoz <abbulutluoz@gmail.com>
Date:   Sun Sep 25 07:05:22 2022 +0200

    C02

commit 90bd42f51e7692bfc3bd79ca3f06b3aef8d6dd57
Author: ahmet bulutluoz <abbulutluoz@gmail.com>
Date:   Sun Sep 25 07:04:41 2022 +0200

    C01
```

- Local repoda yaptigimiz tum commit'eri, her commit bir satir olacak sekilde listelemek icin git log --oneline

Log listesinde son version HEAD olarak belirtilir.

```
670da98 (HEAD -> main) D03
3f61cc8 D2
5ad0480 (deneme) D01
01867b9 C03
03c8add C02
90bd42f C01
```

Istenen Commit'e Donme

Git ile projeyi istedigimiz bir commit'e dondurebiliriz.

Eski bir commit'e gitmek icin git checkout 01867b9

```
01867b9 (HEAD) C03
03c8add C02
90bd42f C01
```

```
670da98 (HEAD -> main) D03
3f61cc8 D2
5ad0480 (deneme) D01
01867b9 C03
03c8add C02
90bd42f C01
```

Eski commit'e gittikten sonra, o versiyon'u projemizin son hali yapmak icin yeniden versiyon yapmaliyiz.

git add .

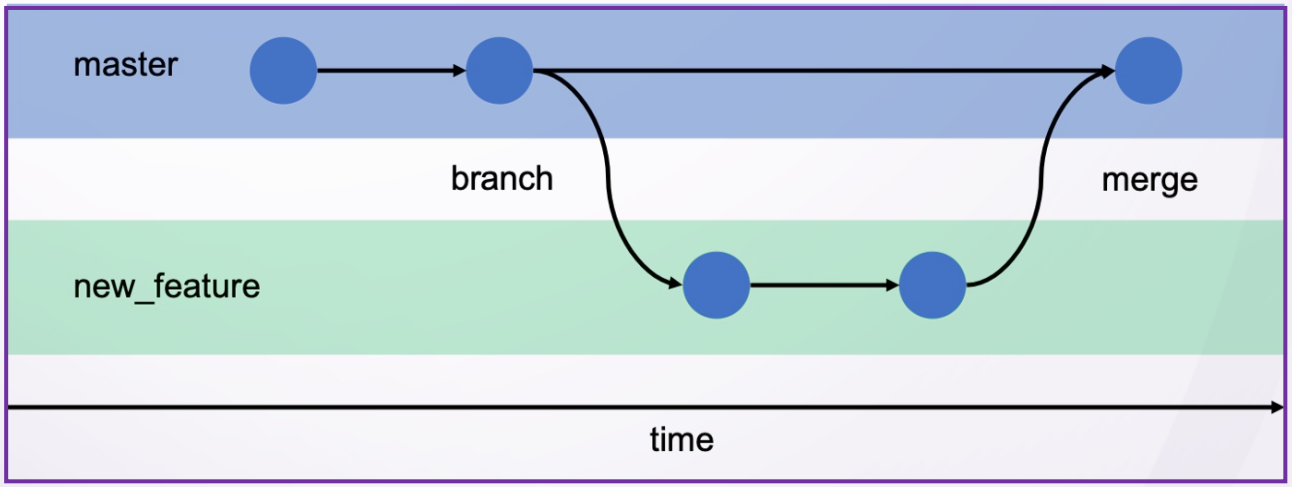
git commit –m "commit ismi"

Ancak git bu istegimize uyari mesaji verecektir. Burada sorumlulugu almamiz ve eski versiyona donmek istedigimizi belirtmemiz gerekir. Bunun icin terminale

git revert 01867b9

:wq komutlari yazilmalidir.

Local Repo'da Branch Kullanimi



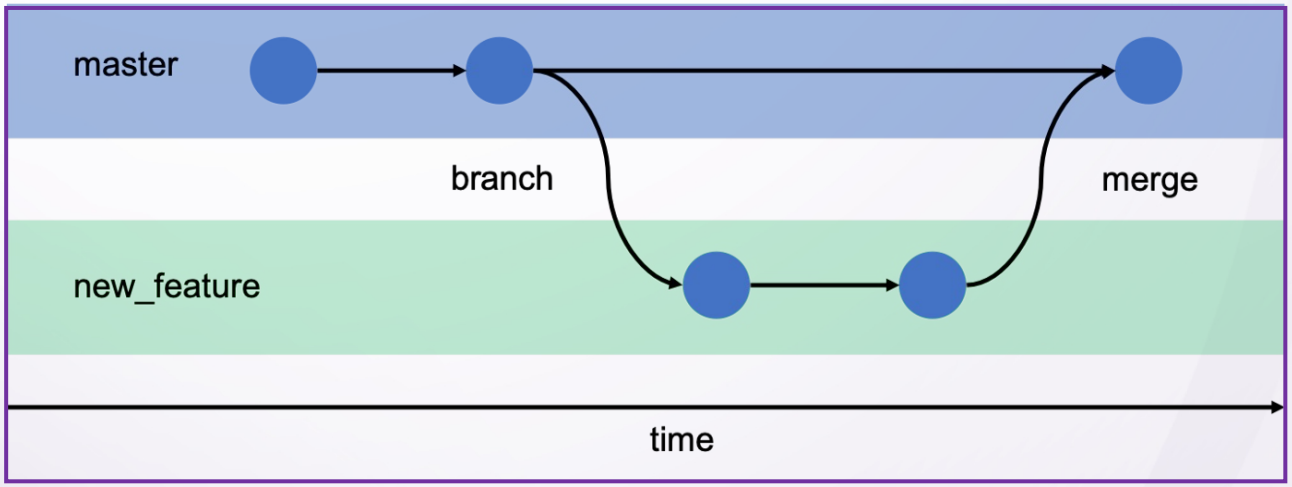
GitHub dagitilmis merkezli bir version kontrol sistemidir.

Bu ozelligi, bir projede calisan pek cok kisinin ayni anda degisiklikler yapabilmesi, kendi kodlarini ana projeye gondermeleri ve yeni kodlar ile ana projeyi guncellemelerine imkan tanir.

Branch (dal) yapisi local repo icin de mumkundur.

Projemizde sonucu belirli olmayan bir sey denemek istiyorsaniz veya projenizde calismaya devam ederken biryandan da bazi degisikliklere calismak istiyorsak main branch disinda bir branch olusturup onunla calisabiliriz.

Local Repo'da Branch Kullanimi



- Branch listesini gormek icin git branch
- Yeni bir branch olusturmak icin git branch branchIsmi
- Istenen branch'e gecmek icin git checkout branchIsmi
- Baska bir branch'i kullandigimiz branch ile birlestirme git merge branchIsmi

NOT : Merge yaptigimiz branchlerde ayni kod satirinda farkli kodlar varsa conflict olusur. Git iki kodu da gosterir ve bizden manuel olarak bunlari duzenlememiz istenir.