
Sayfa 110**Git ve GitHub****1. GitHub nedir?**

- Sürüm kontrol sistemi
- Dokümanların yeni / eski versiyonunu takip eder
- Dosya kümesini yönetir / depolar

2. Depo nedir?

- Dosyaların kaydedildiği klasör ve
- Tek, dosya koleksiyonları veya tek projeler içerebilir.

3. Uzak ve Yerel Depo nedir?

- *Uzak Depo*: Sunucuda ana bilgisayar (GITHUB) Değişikliklerimiz yerelden uzak depoya gider
- *Yerel Depo*: Genellikle bilgisayarınızda -Değişikliklerimiz burada Çalışma Dizini, dizin ve HEAD'den oluşur.

4. Git komutları nelerdir?

- **Ekle** : hazırlık alanına ekleyin
- **Kaydetme** : çalışma dizininden ve yerel depodan ekleyin
- **Gönder** : uzak depoya ekle
- **Pull** : değişiklikleri uzaktan çalışma dizinine alın
- url ile klonla: url'yi dizine kopyalar
- Git sürümü: size git sürümünü verir
- Git durumu: hangi daldaki olduğunuzu ve izlenmeyen tüm değiştirilmiş dosyaları gösterir
 - Menşei: uzaktan kumanda adı
 - Master: şubenin adı
- Git ekle:
 - Evreleme alanına ekleme
 - Özyinelemeli ekleme
 - Her şeyi ekler
- **git commit -m** : "mesaj tüm dosyalar için geçerli olacak"
- **git push** : origin nameOfBranch
- **git ignore** :
 - Notepad.gitignore → Not defterinde, hazırlama alanına eklemek istemediğiniz dosyaları ekleyin
 - YOK ETMEK İSTEDİĞİNİZ DOSYALARIN SİPARİŞ EDİLMESİ İÇİN .GITIGNORE DOSYASINI BASMALISINIZ
 - GIT'TE YOK EDİLDİ
 - Bazı dosyalar önemli değildir ve git'e gönderilmemelidir

- Dosyayı kaldır → GIT EKLE POM'U KALDIR → BUNU İŞLEMLE → VE İTME Kendi şubesini oluştur
- checkout branch -git → Git checkout -b nameOfBranch master

5. Git'i terminalde nasıl kullanırım?

- yeni repo-git oluştur
- echo "# SqlMentor" >> README.md


```
git init
git README.md ekle
git commit -m "ilk commit"
git uzaktan kaynak ekle
https://github.com/Andylam224/SqlMentor.git git push -u kaynağı
usta
```
- mevcut bir repo-git'i aktar
 - git uzaktan kaynak ekle https://github.com/Andylam224/SqlMentor.git
 - git push -u kaynak yöneticisi
- Varsayılan düzenleyici

110

Sayfa 111

6. GIT Komutları?

```
git init
git ekle.
git commit -m "yorumum"
-----
git bilgisi
git günlüğü
git push -u kaynak yöneticisi
git itme
-----
git init
git uzaktan kaynak URL ekle // kopyala yapıştır https:// url'yi URL yerine
git add src // sadece bu klasörü eklemek istersem
git commit -m "yorumum"
git günlüğü
git push -u kaynak yöneticisi
```

7. En son sürüme mi dönüyorsunuz?

```
// ikisini de yazmamız gerekiyor
git getirme kaynağı
git sıfırlama - sabit kökeni / ana
```

8. Bir seferde çift dosya mı ekliyorsunuz?

```
git dosya1 dosya2 dosya3 ekle //
```

9. GIT Şube şubeleri?

```
git rm file.java
git commit -m "kaldırma"
git itme kaynağı yöneticisi
```

git şube BranchName	: - Dal oluşturma
git şubesi	: - şube yöneticisini kontrol etme
git checkout BranchName	: - ad, geçiş yapmak istediğiniz şube adıdır
git şube -d BranchName	: - yerelde ayrıca siliniyor
git itme kaynağı: deleteBranchName	: - Uzak (gitHub WebSite) yerel (intellij) üzerinde şube deteled siliniyor
git şube -a	: - Yerelde silinmiş olsa bile tüm şubeleri kontrol etme (ancak uzaktan değil)
git checkout -b BranchName	: - Dal oluşturma ve yeni şubeye geçiş
git merge BranchName	: - Birleştirme şubesi
git push --set-upstream origin BranchName: - Şubeyi yerelden uzak (gitHub WebSite) (intellij) itme	
git kaynak BranchName getir	: - Uzaktan yerel (intellij) şubeye çekme (github WebSite)
git itme kaynağı branch1: branch2	
git çekme kaynağı branch1: branch2	

111

Sayfa 112

10. Şubeyi ana ile birleştirme

```
--ikinci şubenize gidin sonraki adımları uygulayın
git ekle.
git commit -m "yorumunuz"
- ana şubenize gidin
git merge "branchName"
- eğer birleşmiyorsa git commit -m "yorum" u ana şubeden tekrar yapmalıyız
```

11. Kod GIT Komutlarındaki Değişikliklerle GitHub deposundan Yerel ana sunucuya yeni Branch birleştiriliyor mu?

```
git kaynak BranchName getir
git checkout BranchName
git şubesi // yeni şubede olmalısın
git ödeme yöneticisi
git şubesi // ana dalda olmalısın
git merge BranchName // birleşecek. herhangi bir çatışma varsa düzeltmeniz gerekiyorsa, birleştirme çatışmanız yoksa geçecek

git şube -a // hem yerel hem de uzak dalları görebiliyoruz
klonlamak istediğiniz şeyin git clone URL'si // ve bağlantıyı kopyaladıktan sonra
git getir // ctobi obnovit obnovleniya v glavnom
git birleştirme
git log - grafik // neyin taahhüt edildiğini ve ne olduğunu göstermek
git log --graph - çevrimiçi // neler olduğunu tek satırda gösterme
- Çakışmanız varsa projeye gidin sağ tıklayın -> git -> çakışmayı çöz -> birleştir:
wq ve kaç
```

12. GITHUB URL'SİNİ KONTROL EDİN?

```
git uzaktan -v
git config --get remote.origin.url
git uzak gösteri kaynağı
git config --get remote.origin.url
```

13. Çekme talebi nedir?

```
git merge fetch_head --allow-unrelated-geçmişleri
• İleri sarma sorunu olmayan bir sorunu çözdü
• Escape tuşuna ve ardından
  o ": x!" → Kaydeder ve çıkar
  o ": q!" → Kaydetme ve çıkma yok
```

14. Çekme talebi nedir?

```
• Git merge fetch_head --allow-unrelated-geçmişleri
  o İleri sarma olmayan bir sorunu çözme sorunu çözüldü
```

15. Git'teki çatışmayı nasıl çözersiniz?

```
• deponuz → cd ~/ <repo_directory>
• Son sürüm deposunu çek → git çekme
• Kaynak şubeye göz atın → git checkout <feature_branch>
• Hedef dalı kaynak dala çekin → git çekme orijini <hedef_branch>
• Çakışmaları düzeltin ve ardından sonucu kesin.
```

112

Sayfa 113

16. Git ve SVN komutları

Git-Subversion komutlarının karşılaştırma tablosu

Komut	Operasyon	Yıkım
git klon	Depoyu kopyala	svn ödeme
git commit	Dosya geçmişindeki değişiklikleri kaydedin	svn commit
git göster	Kaydetme ayrıntılarını görüntüleyin	svn kedi
git durumu	Durumu onaylayın	svn durumu

git fark	Farklılıkları kontrol edin	svn fark
git günlüğü	Günlüğü kontrol et	svn günlüğü
git ekle	İlave	svn ekle
git mv	Hareket	svn mv
git rm	Silme	svn rm
git checkout	Değişikliği iptal et	svn geri döndürme 1
git sıfırla	Değişikliği iptal et	svn geri döndürme 1
git şubesi	Bir şube yap	svn kopyası 2
git checkout	Şube değiştir	svn anahtarı
git birleştirme	Birleştirmek	svn birleştirme
git etiketi	Bir etiket oluşturun	svn kopyası 2
git çekme	Güncelleme	svn güncellemesi
git getir	Güncelleme	svn güncellemesi
git itme	Uzaktan kumandaya yansıtılır	svn commit 3
git görmezden gel	Dosya listesini yoksay	svn görmezden gel

1. SVN'de geri dönme, değişikliğin iptalidir, ancak Git'te Geri Döndür, olumsuzlamanın kesinleştirilmesidir. Revert'in anlamları farklıdır.

2. Dal ve etiket SVN'deki yapıda aynıdır, ancak Git'te açıkça farklıdır.

3. SVN, yerel depo / uzak depo kavramına sahip değildir, bu nedenle commit doğrudan uzaktaki depoya yansıtılır.

Ancak Git, yerel depoya yansıtma ve uzak depoya yansıtma için farklı yansıtma yöntemlerine sahiptir.