**Git Notları**

İçindekiler

1. 1. Git Nedir?
2. 2. Git Kurulumu
3. 3. Git Temel Komutları
4. 4. Git Branch (Dal) Yapısı
5. 5. Git Merge ve Rebase
6. 6. GitHub ile Entegrasyon
7. 7. Gitignore Dosyası
8. 8. Git Log ve Commit İnceleme
9. 9. Git Stash ile Değişiklikleri Saklama
10. 10. Git Reset ve Revert
11. 11. Sık Karşılaşılan Git Hataları ve Çözümleri

**1. Git Nedir?**

Git, geliştiricilerin dosyalar üzerinde versiyon kontrolü yapabilmesini sağlayan dağıtık bir versiyon kontrol sistemidir. Linus Torvalds tarafından Linux çekirdeği geliştirilirken yaratılmıştır.

# 2. Git Kurulumu

Git'i [https://git-scm.com](https://git-scm.com) adresinden indirip yükleyebilirsiniz. Kurulum sırasında `git bash` adlı bir terminal aracı da yüklenir.

# 3. Git Temel Komutları

- `git init` : Yeni bir git deposu başlatır.  
- `git clone <repo\_url>` : Var olan bir repoyu bilgisayarına indirir.  
- `git status` : Deponun mevcut durumunu gösterir.  
- `git add <dosya>` : Dosyayı sahneye ekler.  
- `git commit -m "mesaj"` : Sahnedeki dosyaları commit eder.  
- `git push` : Değişiklikleri uzak sunucuya gönderir.  
- `git pull` : Uzak depodaki değişiklikleri alır.

# 4. Git Branch (Dal) Yapısı

Git ile aynı proje üzerinde farklı dallar oluşturarak çalışabilirsiniz. Bu sayede yeni özellikler geliştirirken ana dal etkilenmez.  
- `git branch <isim>` : Yeni bir dal oluşturur.  
- `git checkout <isim>` : Dal değiştirir.  
- `git merge <isim>` : İlgili dalı mevcut dala birleştirir.

# 5. Git Merge ve Rebase

Merge, iki dalı birleştirirken her iki dalın da geçmişini korur. Rebase ise bir dalın geçmişini başka bir dalın sonuna ekler.  
- `git merge <dal>`  
- `git rebase <dal>`

# 6. GitHub ile Entegrasyon

GitHub, Git için bir uzak sunucu görevi görür. Reponuzu GitHub'a yüklemek için şu adımları izleyin:  
- GitHub’da yeni bir repo oluştur.  
- `git remote add origin <repo\_url>`  
- `git push -u origin main`

# 7. Gitignore Dosyası

.gitignore dosyası, hangi dosya ve klasörlerin Git tarafından izlenmeyeceğini belirtir.  
Örnek:  
```  
node\_modules/  
.env  
\*.log

# 8. Git Log ve Commit İnceleme

- `git log` : Commit geçmişini gösterir.  
- `git log --oneline` : Kısa şekilde commit’leri listeler.  
- `git show <commit\_id>` : Bir commit’in detaylarını gösterir.

# 

# 9. Git Stash ile Değişiklikleri Saklama

- `git stash` : Çalışmanızı saklar.  
- `git stash list` : Saklanan değişiklikleri listeler.  
- `git stash apply` : En son saklanan değişikliği geri getirir.

# 10. Git Reset ve Revert

- `git reset` : Commit’i geri alır (farklı modları vardır).  
- `git revert` : Yeni bir commit ile değişikliği geri alır.

# 11. Sık Karşılaşılan Git Hataları ve Çözümleri

- `fatal: not a git repository` : Yanlış klasörde çalışıyorsunuzdur.  
- `merge conflict` : Aynı satırda iki farklı değişiklik varsa çıkar. Elle çözülüp commit edilir.