

Git ve GitHub - Haftalık Çalışma Planı

Hafta Hedefi

Git ve GitHub temellerini öğrenmek, pratik yaparak versiyon kontrol sistemini kullanabilir hale gelmek.

Günlük Plan

Gün 1: Kurulum ve İlk Adımlar ⚡

Süre: 2-3 saat

Yapılacaklar:

- Git kurulumu (işletim sistemine göre)
- Git konfigürasyon (`git config --global user.name` ve `user.email`)
- Git'in ne olduğunu ve neden kullanıldığını öğren
- Terminal/Command Prompt'a alış

Pratik:

```
bash
git --version
git config --global user.name "Adın Soyadın"
git config --global user.email "email@example.com"
git config --list
```

Ödev: Git nedir, neden önemli - 10 dakikalık araştırma yap

Gün 2: Yerel Repository İşlemleri 🏠

Süre: 2-3 saat

Yapılacaklar:

- `git init` ile repo oluşturma
- `git add` ve `git commit` komutları
- `git status` ve `git log` ile durumu kontrol etme
- Working directory, staging area, repository kavramları

Pratik:

```
bash
```

```
mkdir my-first-repo
cd my-first-repo
git init
echo "Merhaba Git!" > readme.txt
git add readme.txt
git commit -m "İlk commit"
git log
```

Mini Challenge: 5 farklı dosya oluştur ve her birini ayrı commit'lerle ekle

Gün 3: GitHub ve Uzak Repository 🌐

Süre: 2-3 saat

Yapılacaklar:

- GitHub hesabı oluşturma (yoksa)
- Yeni repository oluşturma
- `git remote`, `git push`, `git pull` komutları
- SSH key oluşturma ve GitHub'a ekleme

Pratik:

```
bash

# SSH key oluştur
ssh-keygen -t ed25519 -C "email@example.com"

# GitHub'a public key'i ekle

# Repo bağlantısı
git remote add origin git@github.com:username/repo-name.git
git push -u origin main
```

Ödev: Dün oluşturduğun local repo'yu GitHub'a push et

Gün 4: Branch ve Merge İşlemleri 🌿

Süre: 3 saat

Yapılacaklar:

- Branch nedir, neden kullanılır?
- `git branch`, `git checkout`, `git switch` komutları
- Branch'lar arası geçiş

- `git merge` ile birleştirme

Pratik:

```
bash

git branch feature-1
git checkout feature-1
# Dosya deęişiklikleri yap
git add .
git commit -m "Feature-1 geliřtirmeleri"
git checkout main
git merge feature-1
```

Challenge: 3 farklı feature branch'i oluřtur, her birinde farklı özellikler geliřtir ve main'e merge et

Gün 5: Conflict Çözme ve Best Practices ✂

Süre: 3 saat

Yapılacaklar:

- Merge conflict'leri anlama
- Conflict çözme teknikleri
- `.gitignore` dosyası oluřturma ve kullanma
- Commit mesaj standartları öğrenme

Pratik:

```
bash

# Kasıtlı conflict oluřtur
git checkout -b conflict-branch
# Aynı dosyada aynı satırı farklı şekillerde düzenle
git merge conflict-branch
# Conflict'i çöz
```

Best Practices Öğren:

- Conventional Commits formatı
 - `.gitignore` template'leri
 - Küçük ve anlamlı commit'ler
-

Gün 6: Mini Proje Bařlangıcı 🚀

Süre: 3-4 saat

Proje: Basit Hesaplama Aracı (Python)

Yapılacaklar:

- Proje klasörü ve repo oluştur
- Temel Python hesaplama scripti yaz
- `.gitignore` dosyası ekle (Python için)
- İlk versiyon commit'le

Proje Özellikleri:

```
python

# calculator.py
def topla(a, b):
    return a + b

def cikar(a, b):
    return a - b

def carp(a, b):
    return a * b

def bol(a, b):
    if b != 0:
        return a / b
    return "Sıfıra bölme hatası!"

# Ana menu
if __name__ == "__main__":
    print("Basit Hesap Makinesi")
    # Kullanıcı arayüzü ekle
```

Gün 7: Proje Tamamlama ve Review

Süre: 2-3 saat

Yapılacaklar:

- Proje özelliklerini genişlet
- Test dosyası ekle
- README.md dosyası yaz
- Son commit'leri at

- GitHub'a push et

Proje Geliřtirmeleri:

- Geliřmiř hesaplama fonksiyonları
- Hata kontrolü
- Kullanıcı arayüzü
- Test dosyası

README.md İerięi:

- Proje tanımı
- Kurulum talimatları
- Kullanım örnekleri
- Özellikler listesi



Hafta Sonu Self-Check

Bilmen Gereken Komutlar:

- ☐ git init, add, commit, status, log
- ☐ git push, pull, clone
- ☐ git branch, checkout, merge
- ☐ git remote add origin
- ☐ SSH key kurulumu

Kavramlar:

- ☐ Working Directory vs Staging Area vs Repository
- ☐ Branch ve merge mantığı
- ☐ Conflict özme
- ☐ .gitignore kullanımı
- ☐ Commit mesaj standartları

Pratik Beceriler:

- ☐ GitHub'da repo oluşturabilme
- ☐ Local ve remote repo senkronizasyonu
- ☐ Branch ile paralel geliştirme
- ☐ Merge conflict özebilme



İpuları

Her Gün İin:

- Öğrendiklerin üzerinde en az 30 dakika pratik yap
- Hata yapmaktan korkma, hata yaparak öğren
- Terminal komutlarını ezberlemek yerine anlayarak kullan
- Her commit'te anlamlı mesaj yaz

Takıldığında:

- Git documentation'ı oku
- `git help <komut>` kullan
- Stack Overflow'da ara
- Bana sorabilirsin!

Motivasyon:

- Bu hafta sonunda Git ile rahatça çalışabiliyor olacaksın
- İlk projen GitHub'da live olacak
- Yazılım geliştirme sürecine profesyonel bir adım atmış olacaksın