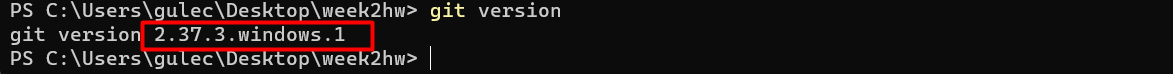
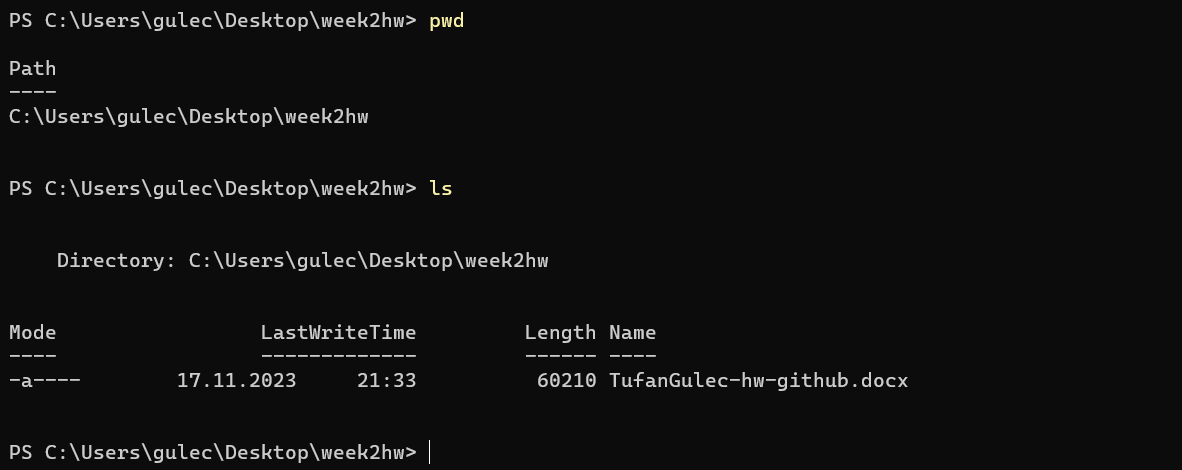
GitHub Ödev Dökümanı

1. Bir projede çalışırken temel Git komutlarını kullanarak değişiklikleri takip etme ve paylaşma becerisini ölçün.

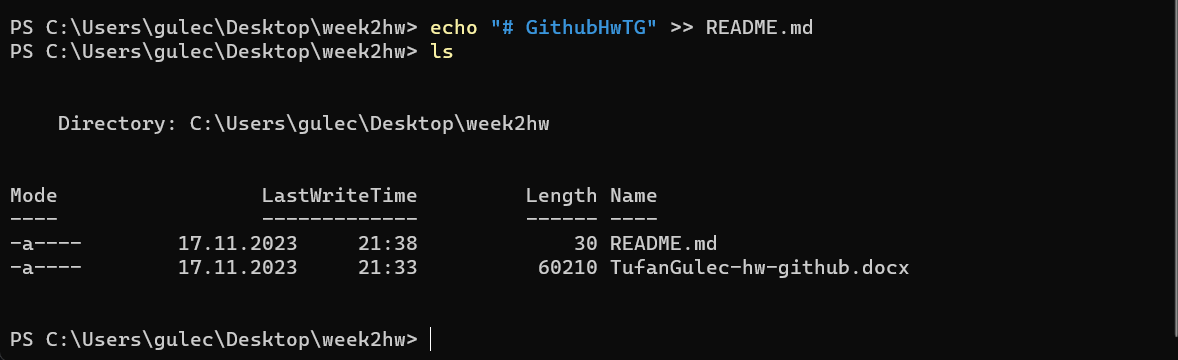
-> git versiyonunu öğrenme



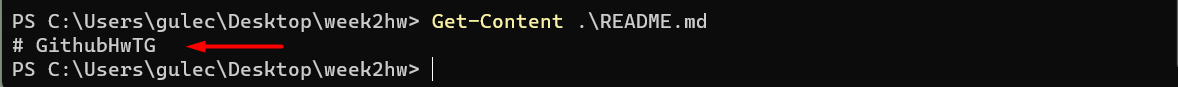
- > masaüstünde “week2hw” adında klasör oluşturdum ve ödev dokümanımı içerisine koydum.



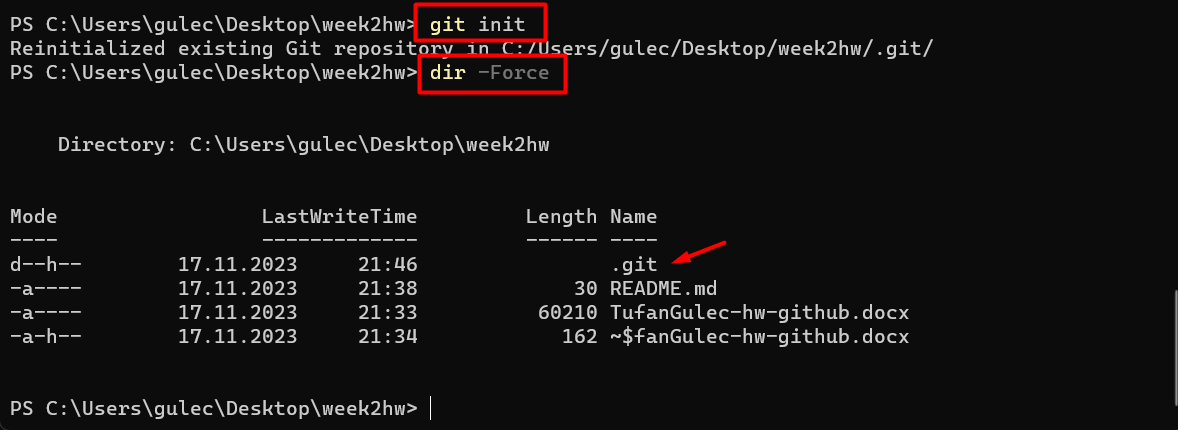
-> README.md dosyası oluştur ve içine



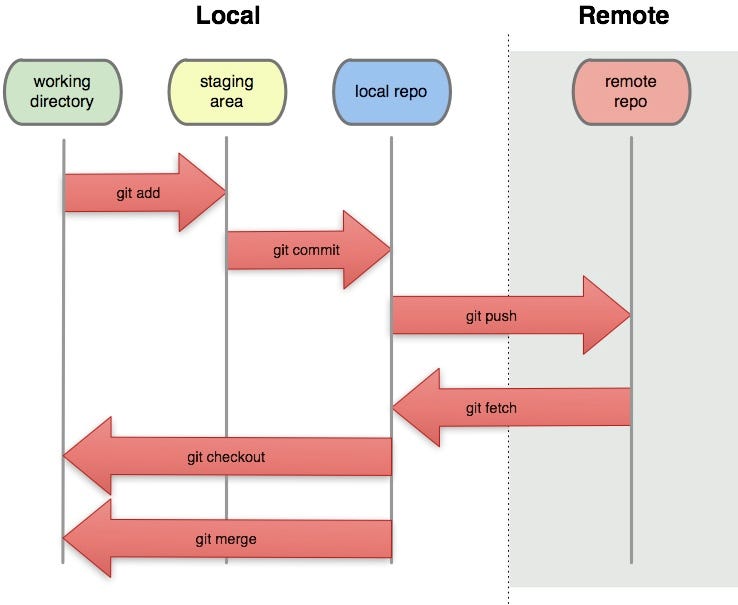
-> Get-Content ile README.md’nin içeriğini okuduğumuzda # GithubHwTG yazdığını görebiliriz.



-> git init ile klasörümüze git tanımalması yapıyoruz. Böylece .git dosaysının oluştuğunu görmemiz gerekir.

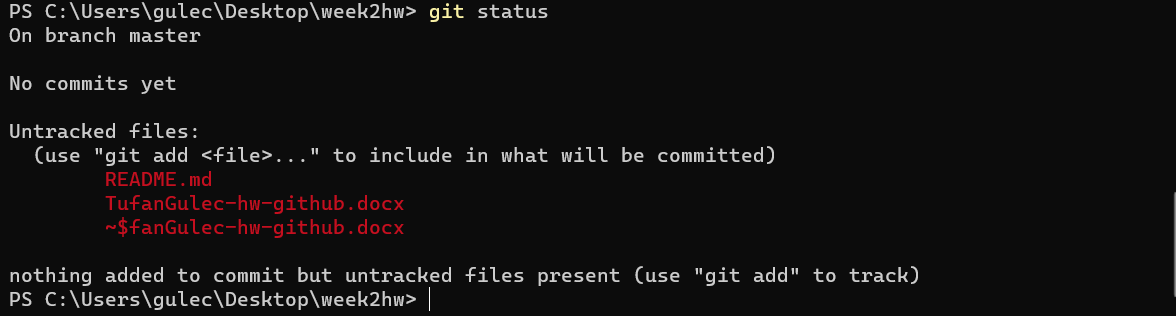


-> Git dosyalarımızı



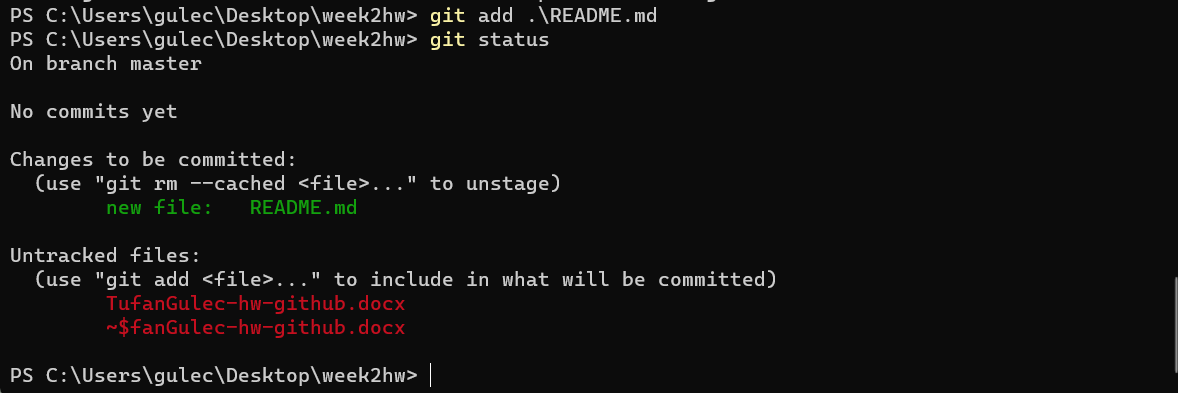
-> git status ile durumu kontrol ediyoruz.

- henüz commit yapılmamış dosyaları gösterdi.

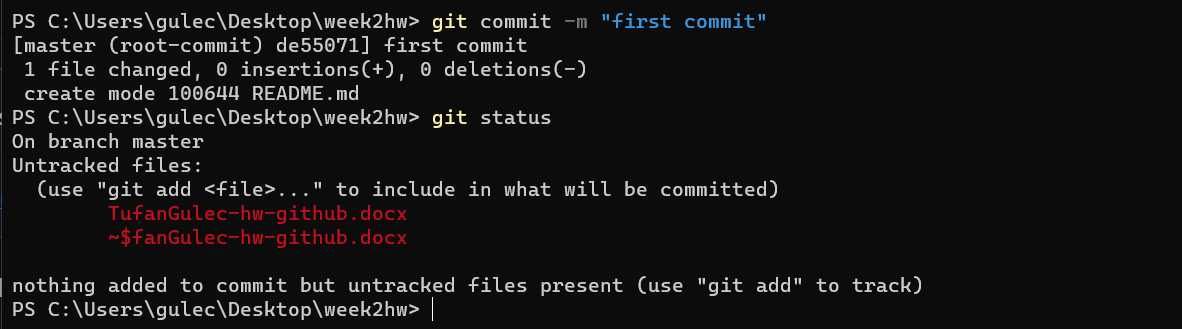


-> git klasörümüzde yaptığımzı değişiklikleri “staged” ortama göndermek için add komutu kullanırız.

- README.md dosyasını staged ortama aldığımızda status’ünü kontrol edersek, README.md dosyasını commit edilebilir dosya oalrka listelediğini görürüz.

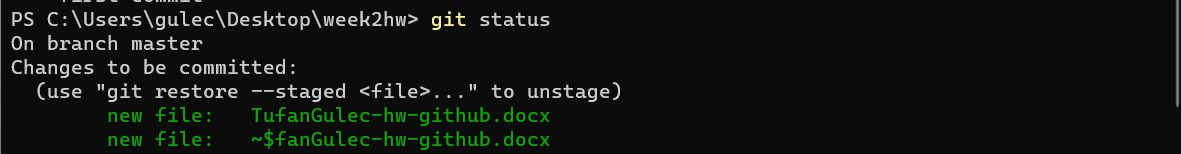


-> staged area’den README.md dosyasını commit’leyelim artık staged area’da olmadığını göreceğiz.

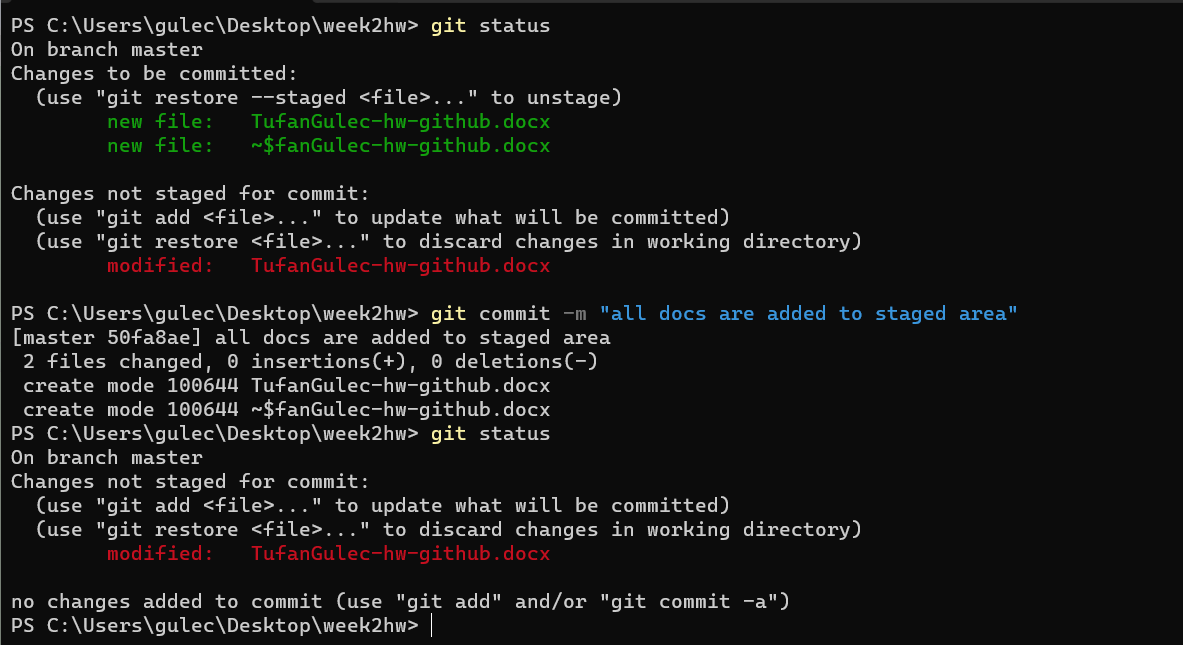


-> working directory’de yer alan tüm dosyaları birlikte staged area’ya alalım.

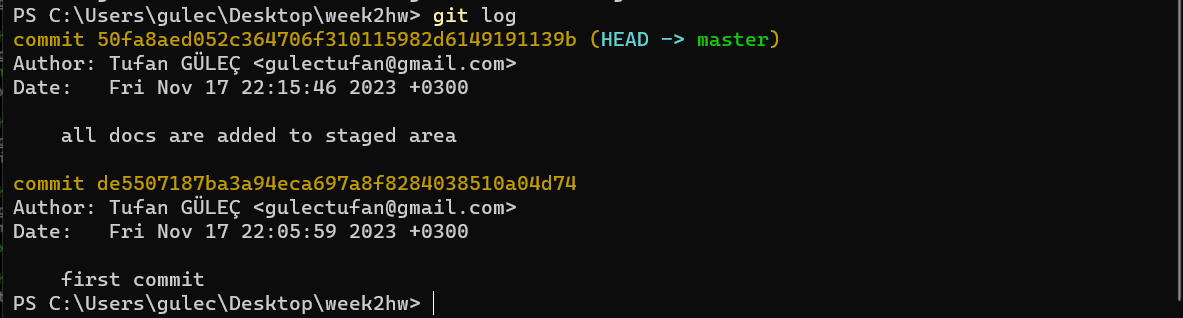




-> staged area’daki dosyalardan birinde değişiklik yapalım ve status’üne bakalım. Ardından commitleyip tekrar status’üne bakalım. Dosyanın staged area’ya alındığı haliyle commitlendiğini, değişiklik yapılan halininse staged area’ya gitmek için beklediğini görebiliriz.

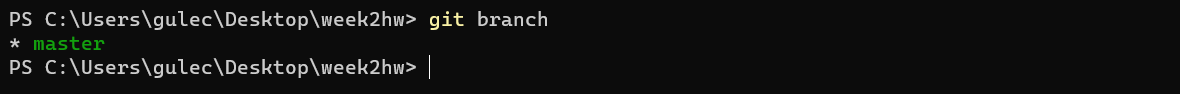


-> git log ile commit’lerimizi takip edebiliriz.



1. Projelerini geliştirmek için branch oluşturma, değişiklik yapma ve bu değişiklikleri ana projeye birleştirme konularını kullanın

-> branch’ları listeleme



-> dev, uat, pre-prod ve prod isimlerinde branch’ler oluşturdum.



-> dev branch’ine geçiş yap

