Kurulum
Git üzerinde temel işlemleri yapmak için Github aşağıdaki adresler de kullanıcılara görsel bir arayüz sunmaktadır.

Windows
https://windows.github.com
Mac
https://mac.github.com
Lİnux
https://git-scm.com

Başlangıç Ayarları
Git kullanıcı bilgileri aşağıdaki şekilde ayarlanmaktadır.
$ git config --global user.name “[name]”
Git üzerinde yaptığınız işlemler için görünen isim bu şekilde ayarlanmaktadır.
$ git config --global user.mail “[email]”
Git üzerinde yaptığınız işlemler için görünen mail adresi bu şekilde ayarlanmaktadır.
$ git config --global color.ui auto
Komut istemcisinin renk ayarı için kullanılmaktadır.

Repository Oluşturma
$ git init [project-name]
Kullanıcının kendi localinde repository oluşturur.
$ git clone [url]
Kullanıcı kendi localine verilen adresteki repository i ekler.

Make Changes
$ git status
Repository üzerinde yapılan bütün işlemleri gösterir.
$ git diff
Repository üzerinde yapılan değişikliklerden sonra dosyalar arasında oluşan farklılıkları gösterir.
$ git add [file]
Commit yapmadan önce commite eklenecek dosyaları stage kısmına ekler.
$ git diff –staged
Stagedeki dosyalar ile versiyondaki dosyalar arasındaki farkları gösterir.
$ git reset [file]
Stagedeki dosyaları add edilmemiş konuma getirir.Bu işlem yapılırken dosyaların içerikleri korunur.
$ git commit -m "[descriptive message]"
Stagede bulunan dosyaları verilen tanımlayoco mesaj ile versiyona ekler.

GROUP CHANGES
$ git branch
Repository de bulunan bütün brancleri listeler.
$ git branch [branch-name]
Verilen isimle yeni bir branch oluşturur.
$ git checkout [branch-name]
Verilen isimdeki branch e geçiş yapar.
$ git merge [branch]
Mevcut brach ile verilen branch i merge eder.
$ git branch -d [branch-name]
Verilen braanch merge edilmişse siler.

REFACTOR FILENAMES
$ git rm [file]
Dosyayı hem localden hem version üzerinden siler
$ git rm --cached [file]
Dosyayı version dan siler.Dosya local de kalmaya devam eder.
$ git mv [file-original] [file-renamed]
Commit etmeden önce verilen dosyanın ismini değiştirir.

SAVE FRAGMENTS
$ git stash
Çalışma dizinimizdeki bütün değişiklikleri kaydeder ve clean hale getirir.
$ git stash list
Bütün stash leri bize gösterir
$ git stash pop
En son alınan stash i etkinleştirir
$ git stash drop
En son alınan stash i siler

REVIEW HISTORY
$ git log
Branch üzerindeki son commit leri tanımlayıcı mesajlar ile gösterir.
$ git log --follow [file]
Verilen geçmiş dosyalarına bakar.
$ git diff [first-branch]...[second-branch]
İki branch arasındaki farkları gösterir.
$ git show [commit]
Verilen commit ile ilgili olarak bilgi verir.

REDO COMMITS
$ git reset [commit]
Verilen commit den önceki hale dönüş yapar.Local deki değişiklikleri tutar.
$ git reset --hard [commit]
Verilen commite bütün değişiklikleri göz ardı ederek döner.

SYNCHRONIZE CHANGES
$ git fetch [bookmark]
Repositoryi çalışma dizinine alır.
$ git merge [bookmark]/[branch]
Fetch edilmiş branchi verilen branch ile merge eder.
$ git push [alias] [branch]
Localimizdeki değişiklikleri remote repository e gönderiri.
$ git pull
Remote daki en son güncel olan repository locale alınır.