**GIT**

Inıt : git desteği eklemek

VS’de initlemek = $ git init

Commit : Yapılan her değişiklik

VS’de commitlemek = $ git config --global user.name “Mert Gürşimşir”

Pull request : Projeye bir şey dahil etme isteği (proje sahibine gönderilir)

Branch : Branch oluşturduğumuzda yaptığımız projenin bir kopyasını oluşturmak

Bir test dosyası oluşturup (mkdir test) içine git’i initlemen lazım.

Sonra örneğin dosya içine readme.md dosyası oluşturdun.   
Sonra bunu add’lemen lazım 🡪 git add readme.md  
Ayrıca “git add .” komutu tüm dosyaları ekler.

“git status” ile durumu kontrol edebilirsin.

“git commit -m “commit message”

readme.md dosyasında değişiklik yaparsan “git add readme.md” komutu ve ardından “git commit -m “message” “ komutu ile commitleyebilirsin.

Yeni branch oluşturmak 🡪 “git checkout -b branchName”

Örneğin readme.md dosyanda bir şeyi değiştirdin ve git add ve commit komutlarıyla yeni değişikliği yükledin. Bu değişiklik yeni branchte yapılır.

“git checkout master” komutuyla master branchine geçersin. Yaptığın değişiklik bu branchte yapılmadı.

Örneğin master branchinde yeni bi dosya oluşturdun ve addleyip commitledin. Bu değişiklik tabi ki yeni branchte olmaz. Şimdi diyelim ki bu değişikliği new branche entegre etmek istiyoruz.   
“git merge master” komutuyla iki dosyayı birleştiririz.  
**Böylece masterda yeni eklediğimiz dosyayı yeni branche entegre ettik, ayrıca readme.md dosyası new branchte değişmiş haliyle, master branchte ise eski haliyle kaldı.**

“git remote add origin URL” (origin dediğimizde URL’e referans yap demek istiyoruz) yaptıktan sonra “git push -u origin master” demeden önce master branchinde olduğundan emin ol.

Eğer githuba yükledikten sonra github üzerinde sen veya bir başkası kodda değişiklik yaparsa:

* “git pull origin master” komutu ile yapılan değişiklikleri locale indirirsin.

“git push” komutu ile kodunu tekrar remote’a atarsın.

Eğer remote’ta bir değişiklik yapıldıysa ve sen de yaptığın değişikliği pull etmeden önce push yapmaya çalışırsan hata alırsın.

Eğer remote’ta değişiklik yapıldıysa ve localde aynı dosyada değişiklik yaptıysan pull ettikten sonra MERGE CONFLICT yersin. Conflicten sonra dosyayı düzenleyip commit edip push etmen gerekir.

Yeni branchini yeni branchte olduğundan emin olduktan sonra (“git checkout new”):

* “git push origin new” komutuyla pushlarsın.

Bunu yaptıktan sonra githubta “new” branchinin olduğunu ve compare&pull request yapmak isteyip istemediğini sorar. Çünkü genelde branchleri hazır olduklarında master branchi ile birleştirmek için kullandığımızı bilir.

Branchlere gelip “new pull request” dersen “can’t automatically merge” yazısını düzenleme ekranının üzerinde görürsün. Çünkü conflicting changes vardır. Draft pull request olarak submit edebilirsin. Files changed kısmında yapılan değişiklikler gözükür.

Bir projeye katkı sağlamak istiyorsun:

* Fork ettin.
* “git clone URL”
* Değişiklik yaptın, addledin, commitledin
* “git push origin master” (ya da master değil de üzerinde çalıştığın branchin adı)
* Forkladığın github sayfasından original sayfaya git (“forked from …” kısmındaki …’ya git)
* Pull requests tabına gel ve new pull request de
* Compare across forks de (gerekmeyebilir)
* Açıklamanı yaz ve create a pull request de

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

 🡪 

If you have an image that you want to display:

![Alt text](path to the image -> can be local path or URL from internet)

A picture containing graphical user interface

Description automatically generated 🡪 Table

Description automatically generated

:--- 🡪 left aligned  
---: 🡪 right aligned  
:---: 🡪 center

* `code.representation()`
* ```

code.block();

```

* ```js

code.block(); -----> color according to javascript

```

A picture containing table

Description automatically generated A picture containing graphical user interface

Description automatically generated

Text

Description automatically generated with medium confidence Graphical user interface, text, application

Description automatically generated

Pull request yaparken “closes #39707” gibi bir şey yaz. Verdiğin hash bir issuenin hashi.

Eğer erken feedback almak istiyorsan, pull request yaparken sağda “Convert to draft”a tıkla.

It shows to community and project maintainers that you would like some feedback but it’s not ready for the proper review to get merged in.

Otherwise you get lots of comments of things you might not have quite finished yet. That’s why checklist is important.

When it’s ready to have a proper review and everything’s finished, then you can convert it back to a normal pull request.

<https://blog.devcenter.co/contributing-to-your-first-open-source-project-a-practical-approach-1928c4cbdae>