A GIT és a GitHUB

# Telepítés

A fejlesztéshez szükséges Git telepítése. A telepítő ezen a linken elérhető: <https://git-scm.com/downloads> Telepítéskor fontos, hogy a Git Bash-t ([gitforwindows.org](file:///C:\Users\Katalin\Downloads\gitforwindows.org) ) is telepítsük.

<https://github.com/>

# Tartalomjegyzék

[Telepítés 1](#_Toc63452643)

[Git parancsok 1](#_Toc63452644)

[Tartalomjegyzék 1](#_Toc63452645)

[Tananyag 3](#_Toc63452646)

[Mi az a GIT? 3](#_Toc63452647)

[git –version 3](#_Toc63452648)

[Git Bash 3](#_Toc63452649)

[git config --global –list 4](#_Toc63452650)

[Repository 4](#_Toc63452651)

[git init 4](#_Toc63452652)

[Master a Fork és a Branch 4](#_Toc63452653)

[stage és a git add 5](#_Toc63452654)

[git status 5](#_Toc63452655)

[git commit -m „HTML szerkezet, kiinduló állapot” 5](#_Toc63452656)

[**git log --oneline** 4](#_Toc63452657)

[gitk 5](#_Toc63452658)

[git checkout 6](#_Toc63452659)

[git rm fajlnev 6](#_Toc63452660)

[git mv fájlnév mappaneve 6](#_Toc63452661)

[Mire jó a GitHub? 7](#_Toc63452662)

[git push -u origin main 8](#_Toc63452663)

[Weboldal publikálása a gitHub-on 9](#_Toc63452664)

[A fájlok módosítása 10](#_Toc63452665)

[git pull 11](#_Toc63452666)

[Hogyan publikálom a módosításaimat? 12](#_Toc63452667)

[A fájnak 4 állapota van: 12](#_Toc63452668)

[git clone https://github.com/csefikatalin/Probaweboldal.git 12](#_Toc63452669)

[Mi a conflict és hogyan oldjuk fel? 13](#_Toc63452670)

[Közreműködők létrehozása a gitHub-on 13](#_Toc63452671)

# Git parancsok

[git –version 2](#_Toc63452610)

[git config --global –list 3](#_Toc63452611)

[git init 3](#_Toc63452612)

[stage és a git add 5](#_Toc63452613)

[git status 5](#_Toc63452614)

[git commit -m „HTML szerkezet, kiinduló állapot” 5](#_Toc63452615)

[git log --oneline 4](#_Toc63452616)

[gitk 4](#_Toc63452617)

[git checkout 5](#_Toc63452618)

[git rm fajlnev 5](#_Toc63452619)

[git mv fájlnév mappaneve 5](#_Toc63452620)

[git push -u origin main 7](#_Toc63452621)

[git pull 10](#_Toc63452622)

[git clone https://github.com/csefikatalin/Probaweboldal.git 11](#_Toc63452623)

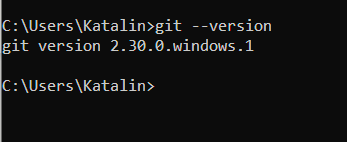
# Tananyag

## Mi az a GIT?

Elosztott verziókezelő rendszer. Támogatja a közös munkát. Ha csoportban dolgozunk, akkor a közösen változtatott fájlokat tudja kezelni, összefésülni.

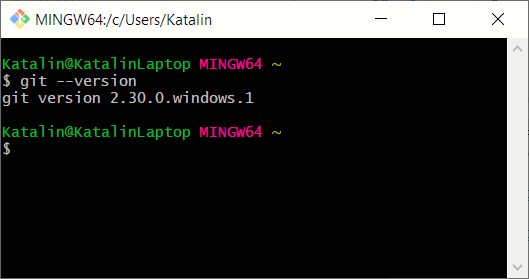
Nyissunk egy cmd ablakot és írjuk be:

#### git –version



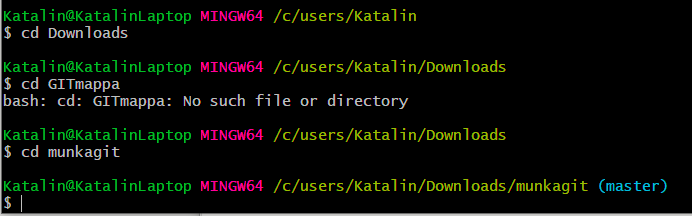
## Git Bash

Ad egy csili vili konzolos felületet a git parancssori használatához.

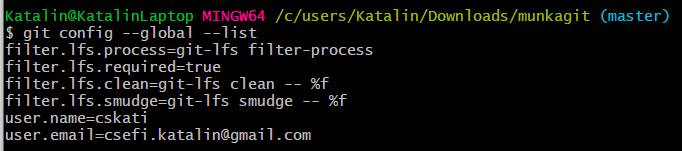


##### Mappaváltás:

CD elérési út.



#### git config --global –list

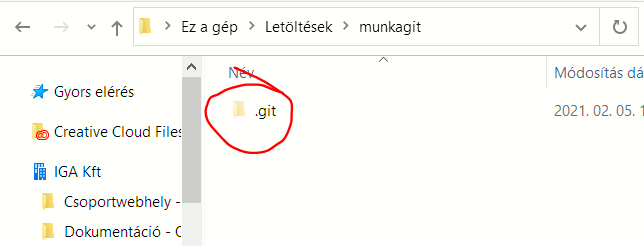


## Repository

Repository egy tárhely, ahol a munkáinkat tároljuk. Nyomon követhetjük a kód változásait és raktározhatjuk a forráskódot

#### git init

Ez a parancs az aktuális mappát git tárhellyé, repository-vá alakítja. Létrehoz egy rejtett mappát. Ez a mappa fogja tárolni a fájljaink állapotait, a különböző verziókat.



A .git rejtett mappa a fájlom történetének oldala.itt tárolok mindent, ami fájlommal történt. Innen bármikor visszaállítható, vagy megnézhető egy korábbi állapot.

### Master a Fork és a Branch

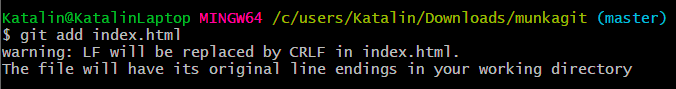
* **Master** – ez egy ág, a fő ág, amiben éppen dolgozunk.
* **Branch** - A fejlesztés során különböző irányban ágazhatunk el, és ezeket az ágakat hasonlíthatjuk össze a master-rel.
* **Fork . villa** – egy másolatot hoz létre a kódból, és onnan folytathatjuk a kódolást.

#### stage és a git add

Hozzunk létre egy fájlt: index.html-t

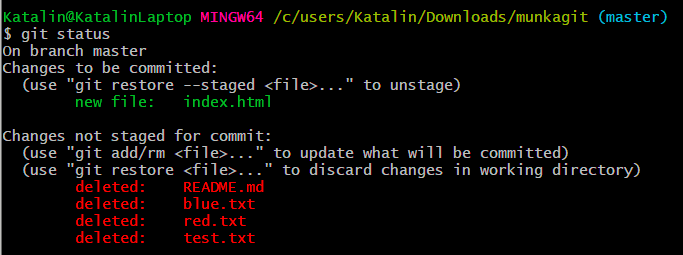
Kitesszük a fájlokat a színpadra a git add index.html, vagy az összes fájlunkat a git add . paranccsal.

Látszólag nem történik semmi, de ezzel előkészítettük a fájlt arra, hogy majd létrehozzunk róla egy állapotot, egy verziót, amit a git tárol majd a fájlunkról.



#### git status

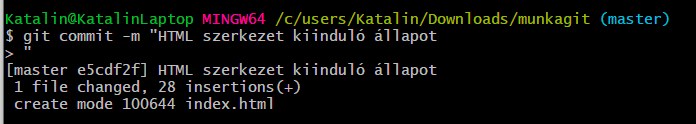
Lekérdezhetjük a git repository-nk aktuális állapotát.



Az látszik, hogy egy (zölddel jelölt) fájl van a színpadon, arra vár, hogy a repo-ba kerüljön.

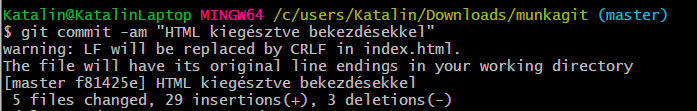
#### git commit -m „HTML szerkezet, kiinduló állapot”

Commitolás = Mérföldkövezés – Akkor használom, ha a fájljaimról szeretnék egy állapotot tárolni.

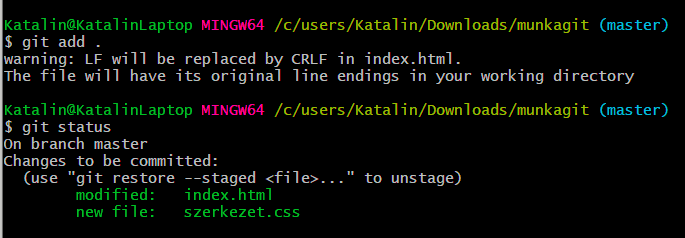


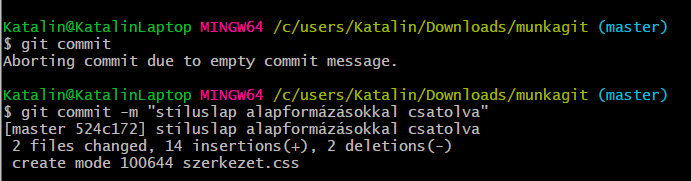
Megtörtént a fájl commitolása, azaz a git repository-ban val elhelyezése, rögzítve lett róla az első állapot.

Ezután módosítsuk az index.html fájlunkat, majd adjuk ki újra a git commit parancsot, egy újabb leíró üzenettel.Ahhoz hogy commitolni tudjunk, először a színpadra kell tenni a fájlt, ezt rövidíthetjük a git commit ‑am „üzenet” paranccsal.



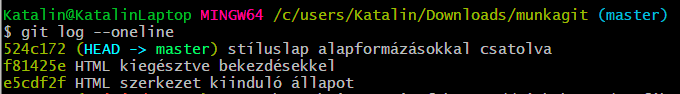
Hozzunk létre egy stíluslapot is, és ezt is adjuk hozzá a repository - hoz az előbbi módon!

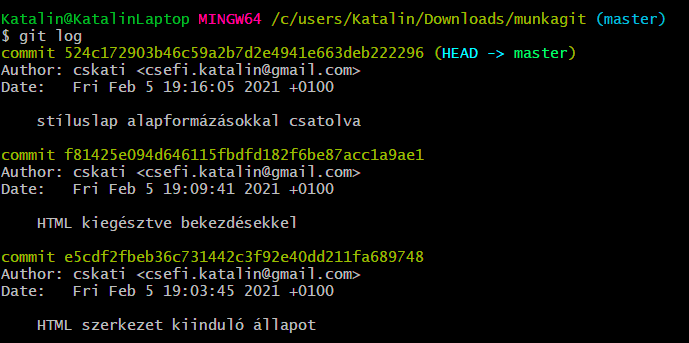




#### git log --oneline

Ezzel a paranccsal megnézhetjük, hogy milyen commitokat hoztunk létre eddig. Ez a naplózás.



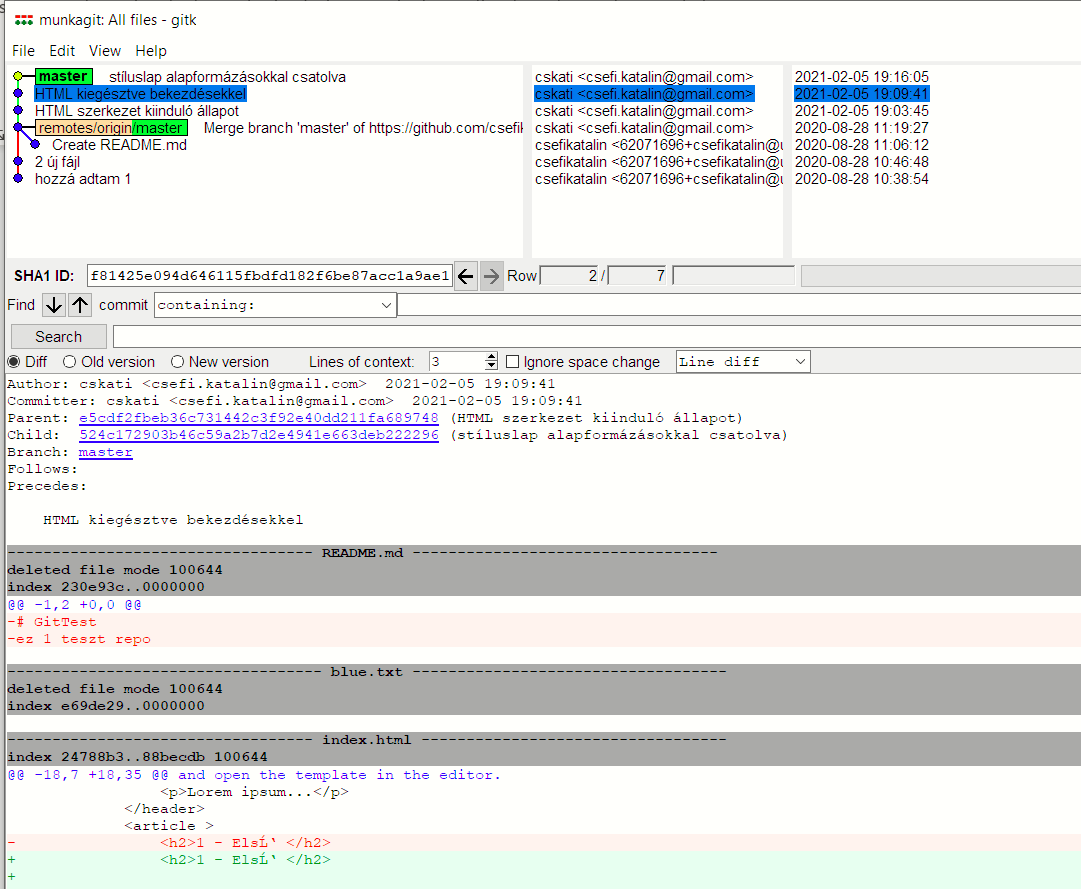


A sárga hexadecimális számok jelzik a különböző commitokat.

A logolásból a q billentyűvel tudunk kilépni.

#### gitk

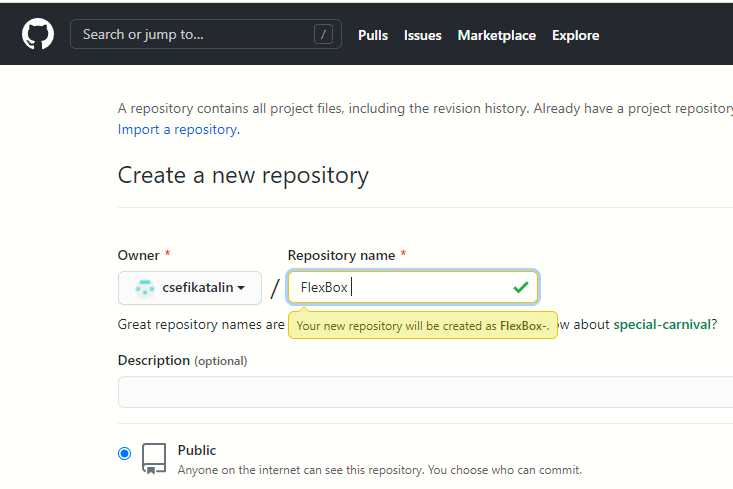
Megnyit egy grafikus ablakot ( git GUI alkalmazás), és megnézhetjük az eddig létrehozott különböző verziókat.



## Mire jó a GitHub?

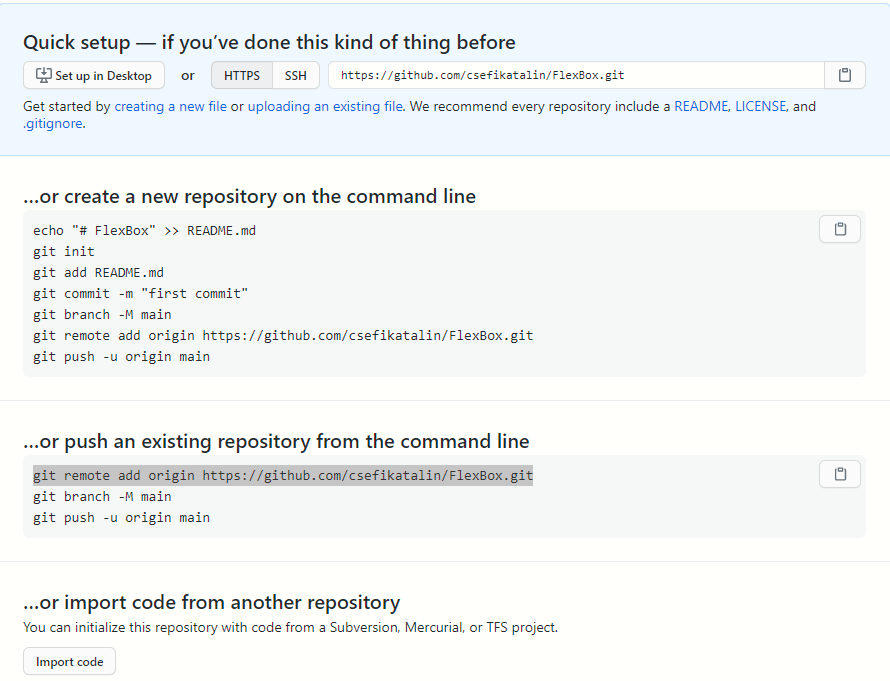
Olyan git repository tároló, aminek van egy webes interface-e.

Regisztráció után létrehozzuk az első repository-nkat.



Ezután érdemes megnézni, hogy mi van a nyitólapon. Itt láthatjuk azokat az utasításokat, amiket parancssorból használhatnánk.

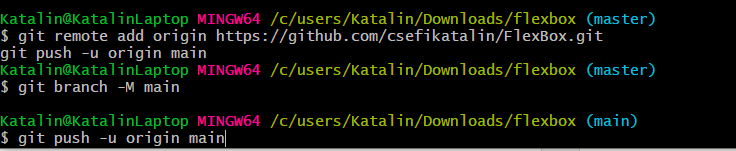
Az új repository-t már nem kell létrehoznunk, azt megtettük korábban.



Most erre van szükségünk, másoljuk vágólapra és illesszük be a parancssorba!

Figyelem, lehet, hogy csak jobb gomb/Paste paranccsal fog működni a beillesztés.

Ezzel most nem kell foglalkoznunk, hiszen pont ezeket az utasításokat használtuk a lokális repo létrehozásához



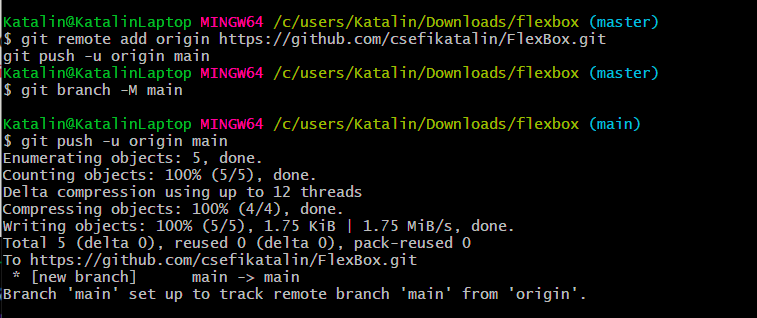
git remote add origin <https://github.com/csefikatalin/FlexBoxAlap.git>

git branch -M main

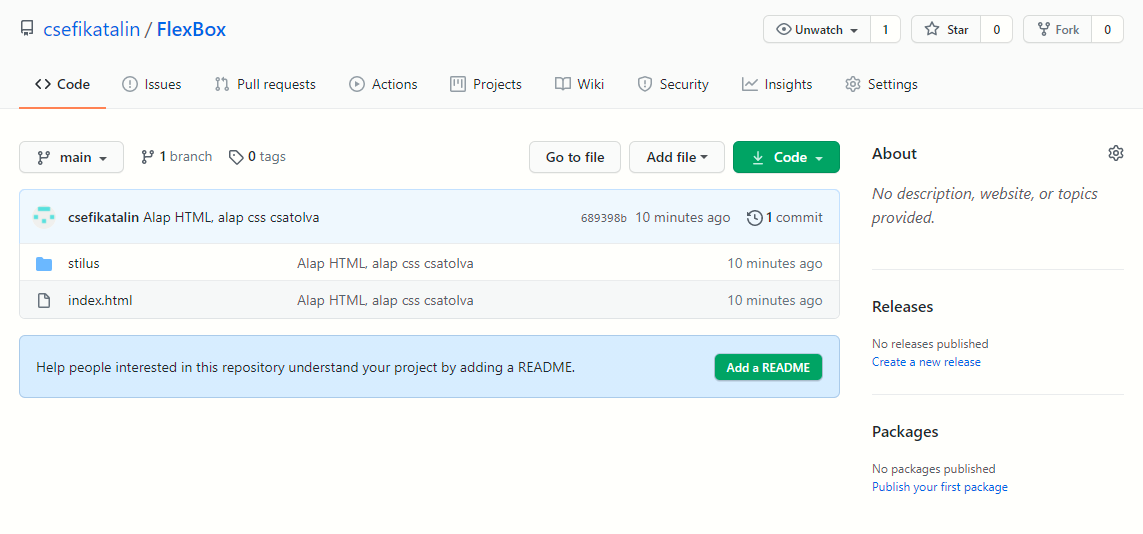
git push -u origin main

#### git push -u origin main

A push paranccsal tudjuk felküldeni a fájljainkat a repo-ba.

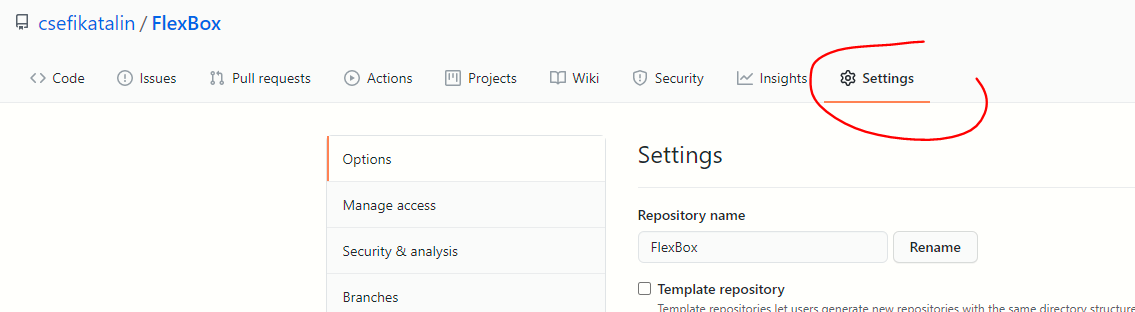


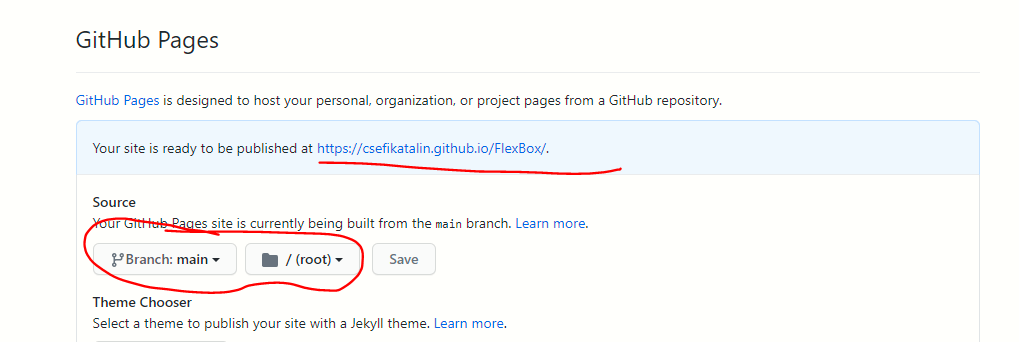
Most ezt látom a github-on:



## Weboldal publikálása a gitHub-on

A repositorynk Settings menüpontjában az option fülön keressük meg a GitHub Pages blokkot.





Állítsuk be a pirossal jelzett részeket, majd megkapjuk a Your site is ready to be published … üzenetet. Ezt a linket másoljuk a böngészőbe, és nézzük meg, mit csináltunk!

Összefoglalás

|  |  |
| --- | --- |
| Hogyan publikálom a módosításaimat? | A fájnak 4 állapota van: |
| 1. **Létrehozom** a fájlt a lokális repo-ban. 2. **git add** hozzáadjuk a szükséges fájlokat a stage-hez 3. **git commit** létrehozzuk a commit-ot, az új állapotot, mérföldkövet. Mindig kell egy megjegyzés, ami leírja az aktuális változtatásokat. 4. **git push** – feltöltjük a fájlokat a távoli repository-ba. | 1. Saját **lokális mappában** a fájlokat módosítom/létrehozom. 2. Az **add** paranccsal az adott állapotból hozom létre a commitot. 3. A **committal** létrejön az állapot. 4. A **push-sal** kerül a felhőbe |