**GitHub, GitHub Desktop tutorial**

**Cél:** Mivel úgyis kell majd Git-et használnunk a verziókezeléshez, szerintem nem árthat már az elején elkezdeni ismerkedni vele. A dokisk és egyéb cuccok kezeléséhez ami az elején kellhet a félévnek, teljesen elég szerintem a GitHub Desktop program. Segít a környezettel és az alapfogalmakkal való megismerkedésben és még hasznos is.

**Tartalom:** GitHub és GitHub Desktop alap megismerése, ehhez szükséges alap fogalmakkal.

1. **GitHub**

Az első dolog, hogy kell egy GitHub fiók. Ezt elég egyszerű megcsinálni, csak fel kell menni a honlpajára :

<https://github.com/>

Ha jól emlékszem, nagyjából minden egyértelmű a regisztrációban, sima ingyenes felhasználó kell, az elég.

Maga a GitHub egy elég nagy és összetett oldal, de így itt az első nekivágásnak nem kell minden részét ismerni.

1. **Alapdolgok**

Csak röviden egy két alapfogalom, ami kell elsőre talán:

Commit

Elemi egység, a fájlokról elmentett pillanatkép. Minél gyakrabban és minél kisebb logikailag összetartozó egységet commitoljunk A commitokank van gyereke és szülője. HEAD: utolsó commit

Branch

Commitok egymásutáni sorozata (commit pötty, branch vonal) master: automatikusan létrejön

Workspace

Fejlesztésben használt fájlok, ezeken dolgozunk

Repository

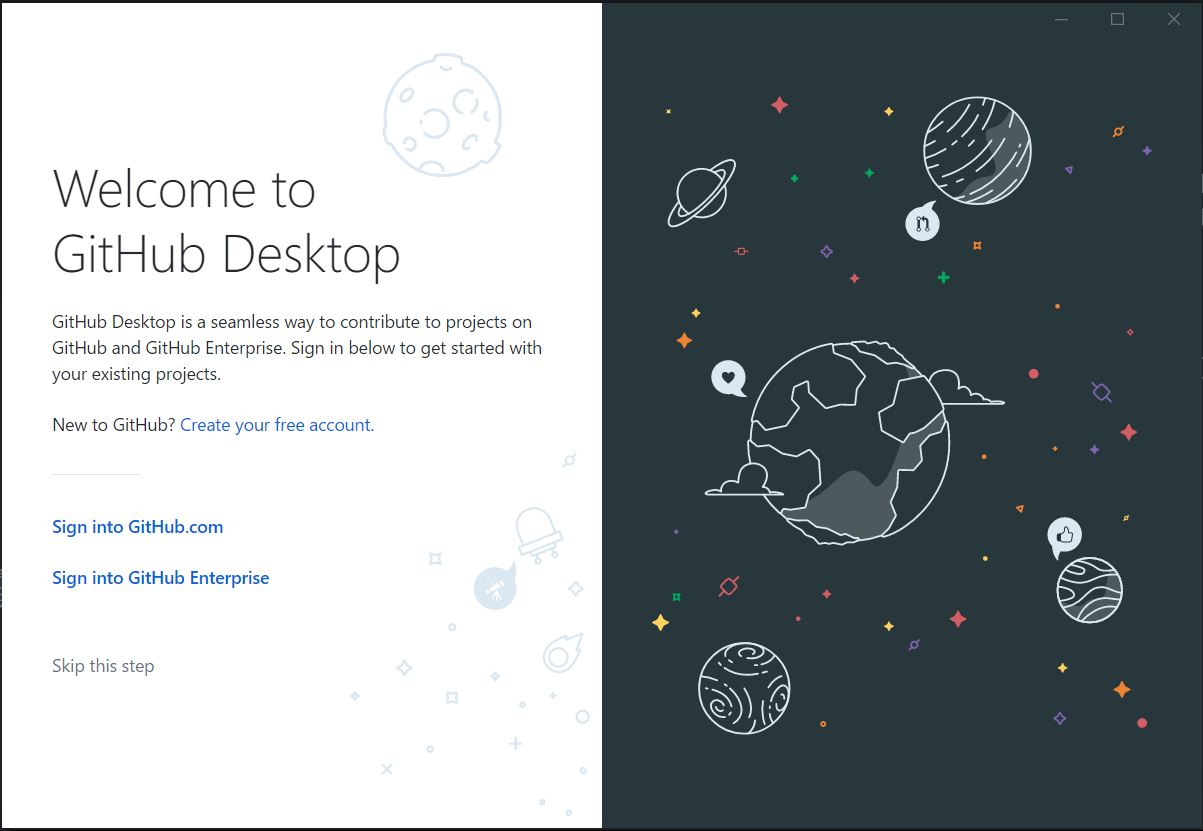
Metadatok tárolása (kommitok, branchek, címkék)

1. **GitHub Desktop**

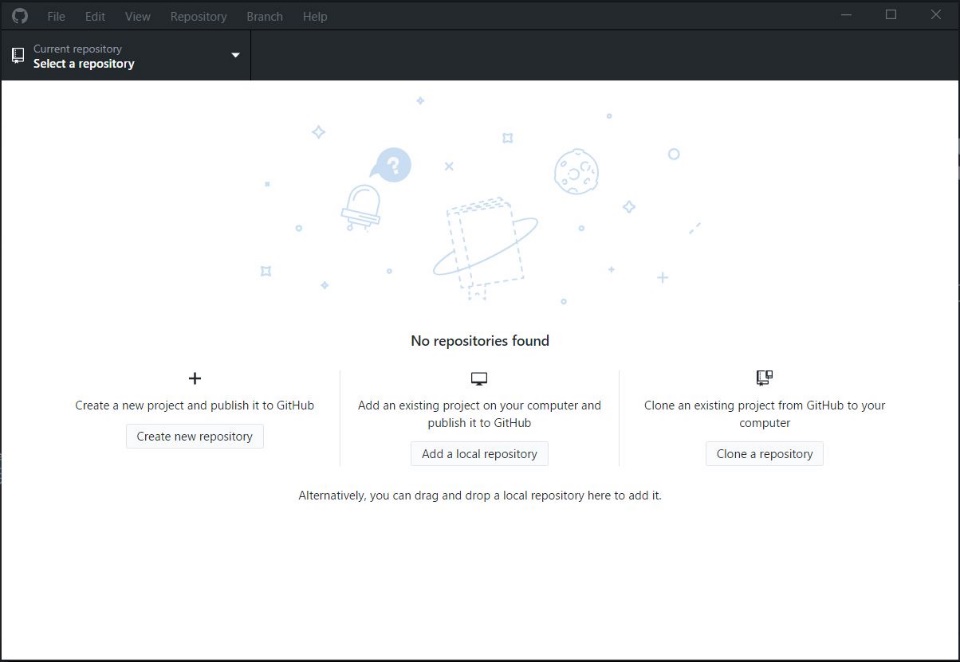
Egy egyszerű, asztali verziója a GitHubnak, elsőre szerintem full elég. Letölteni innen lehet:

<https://desktop.github.com/>

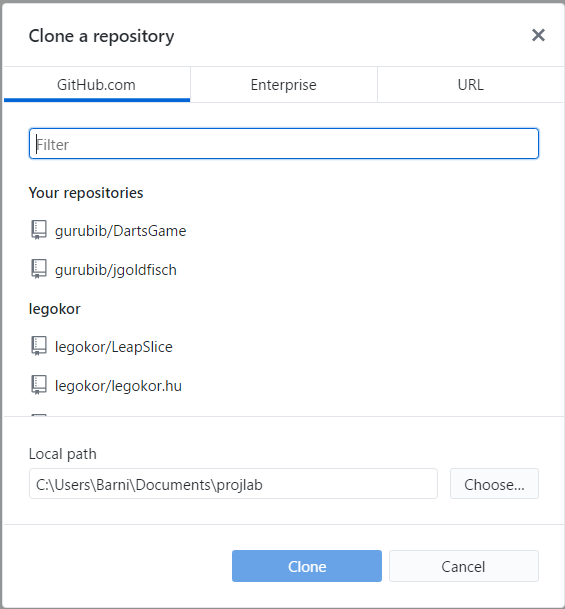
A telepítés elég egyszerű.



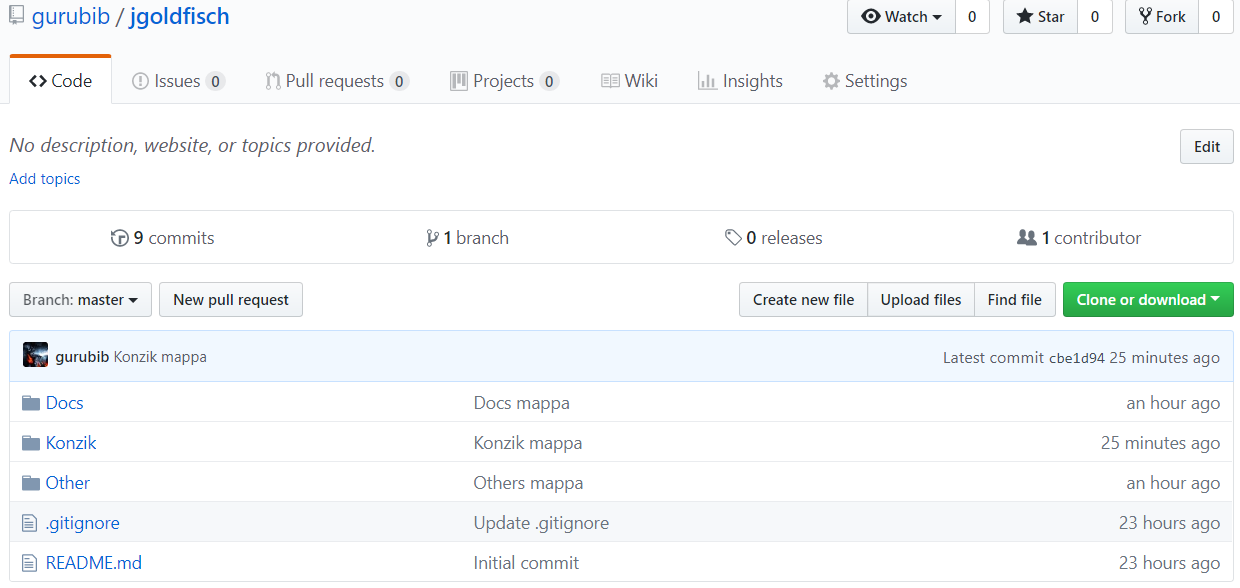
Itt be kell jelentkezni értelemszerűen a sima GitHub.com-os loginnal. Ezután a program még kissé üres állapotban vár.



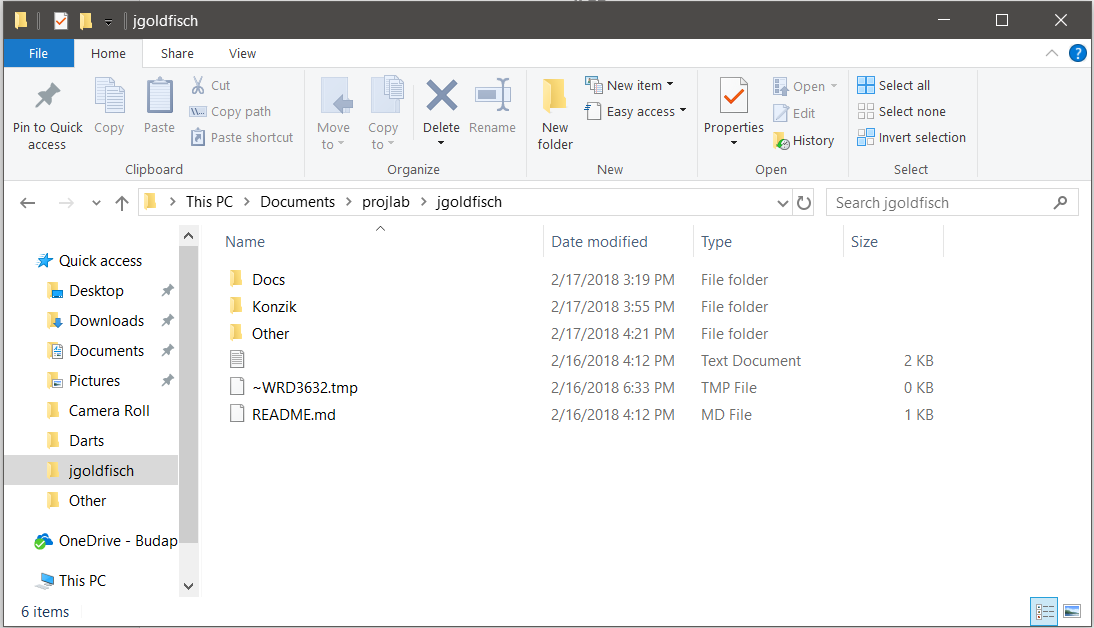
Itt a “Clone a repository” opcióra kell menni. Itt rákeresni a “jgoldfisch” nevűre és kiválasztani



A kiválasztott helyre, le fogja tölteni az egész eddigi repository-t, amit amúgy a GitHub-ról egyszerűen el lehet érni és megnézni. Most így áll:



És a gépemen a clone-ozás után. Nyilván ugyan az.



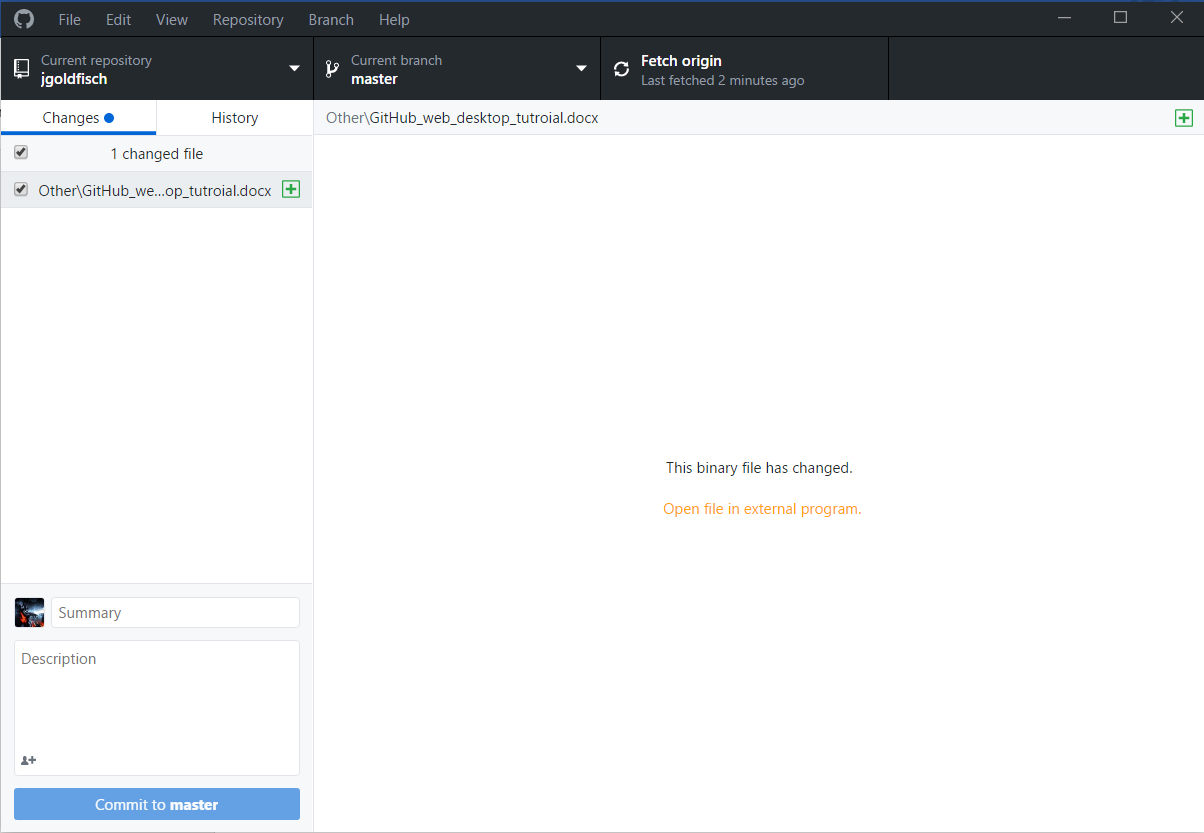
1. **Alap használat**

Első nekiindulásból a használat elve a következő: A gépen lévő repository clone-ba véghezvitt változtatásokat (mappa létrehozása, fájl létrehozása, fájl megváltozása stb…) a program (GitHub Desktop) észreveszi és jelzi. Ha egy változtatást el akarsz menteni a GitHubon, akkor azt Commit-olni kell, és a commit-ot el kell nevezni, valami a változtatást jól tükröző néven (pl. Others mappa létrehozása). Na ekkor ez a változtatás csak a te gépeden él és látszik, ahhoz, hogy ez megjelenjen a közös, neten lévő repo-ban, push-olni kell a változtatásokat. Ha ez megtörténik, akkor már a böngészőből nézve is látszik a változás, sőt a commit is.

A más által véghezvitt változásokat, commitokat is nyilván látni szeretnénk a gépünkön, hiszen ott a repo legfrissebb változatával akarunk dolgozni, ilyenkor szinkronizálni kell a programban a közös netes repo-val (ennek az elnevezése origin).

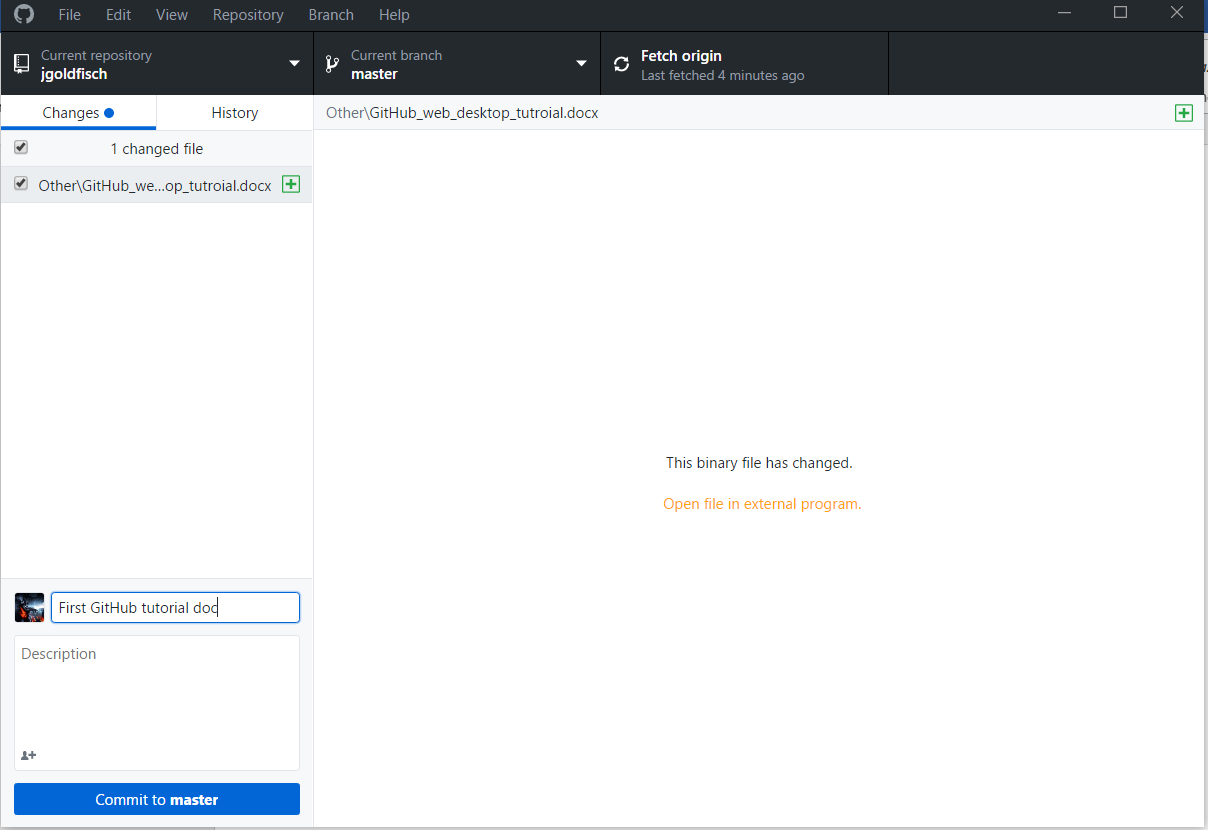
**Egy kis bemutatása ezeknek:**

Létrehoztam ezt a dokumentumot (GitHub\_web\_desktop\_tutorial) a helyi repo Others mappájában. A program ezt reagálta:

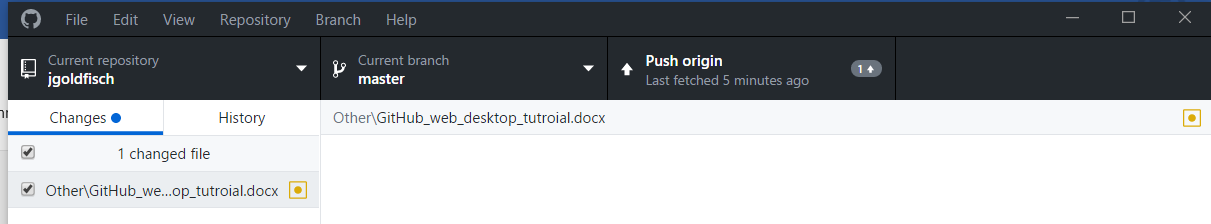


Írja, hogy létrejött a fájl. (amúgy ha sima txt, vagy forráskód, vagy olyan amit egy notepad meg tud nyitni akkor azt a job oldalon ő is megnyitja és mutatja a konkrét változásokat, új sorokat stb…)

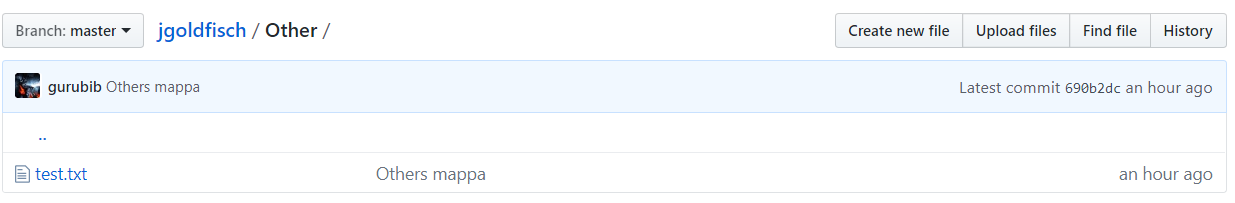
Ezt elmenteni, vagyis commit-olni a bal oldali rész alján tudom, elnevezem a commit-ot és megnyomom a “Commit to master” gombot.



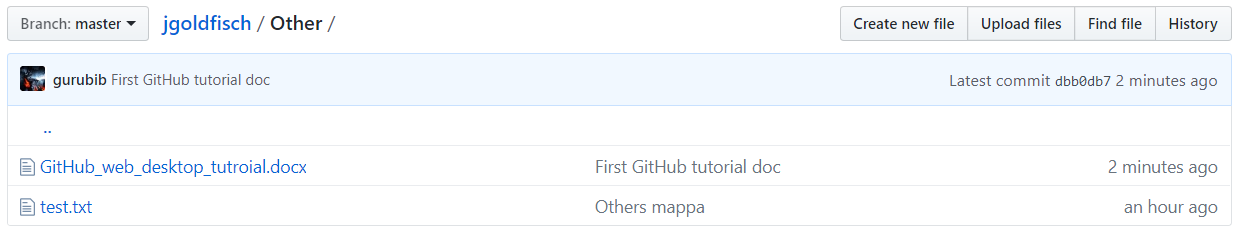
Ekkor ez elmentődik, és jelzi hogy csak itt látszik, az origin-en (tehát a neten még nem).



És ez igaz is mert a neten még sehol semmi:



Azonban megnyomva a “Push origin” gombot, életbelép a változtatás a közös, origin-en:



Ugyanezzel a gombbal lehet, amúgy szinkronizálni a helyi repo-t az origin-el “Fetch origin” szöveg áll rajta ilyenkor. Ha történt változás az origin-en akkor “Pull origin” lesz a szöveg.

**Alap munkafolyamat:**

Eldöntöd, hogy csinálsz valamit -> megnyitod a GitHub Desktopot -> szinkronizálsz egyet az origin-el (Fetch origin gomb) -> ha kell akkor frissítesz róla (Pull origin) -> megcsinálod amit akarsz -> a GH Desktopban látod hogy észlelte a változást -> **mielőtt commit-olsz szinkronizálsz és ha kell frissítesz az originről (Fetch, Pull)** -> megírod a commit-ot, commit-olsz -> frissíted az origin (Push origin)