

GitHub ismertető

0. Bevezető

A következőkben lehetőségetek lesz megismerni a GitHub-ot. Hosszasan, majd röviden is bemutatom hogyan lehet könnyen – mivel több mód is létezik a kezelésére – használni. Ebben az ismertetőben csak addig megyünk el, ameddig az a projektmunkátokban hasznos lesz: letöltés, módosítás, feltöltés.

Mielőtt belevágunk szükséges néhány előkészület, amik nélkül – ezen ismertető alapján – nem tudnátok használni a gitet.

Regisztráció

Ha eddig nem tettétek, akkor feltétlen szükségetek lesz egy fiók regisztrálására. Ezt a következő linken tudjátok megtenni: <https://github.com/signup>

GitHub Desktop letöltése

Ezen ismertető abból „különleges”, hogy a git használatát alkalmazáson keresztül mutatja be. Véleményem szerint ezen az úton a legmegérthetőbb, itt legkönnyebb használni a gitet.

Az alkalmazást innen tudjátok letölteni: <https://desktop.github.com>

Ha regisztráltatok és telepítettétek, akkor el is készültünk, hogy elkezdjük használni a felületet.¹

¹ Miután ez a rész első elolvasás után nem feltétlen lesz annyira hasznos, így betettem egy hivatkozást, amellyel a rövid lépésekhez ugorhattok. Ezt a címre kattintva tudjátok megtenni.

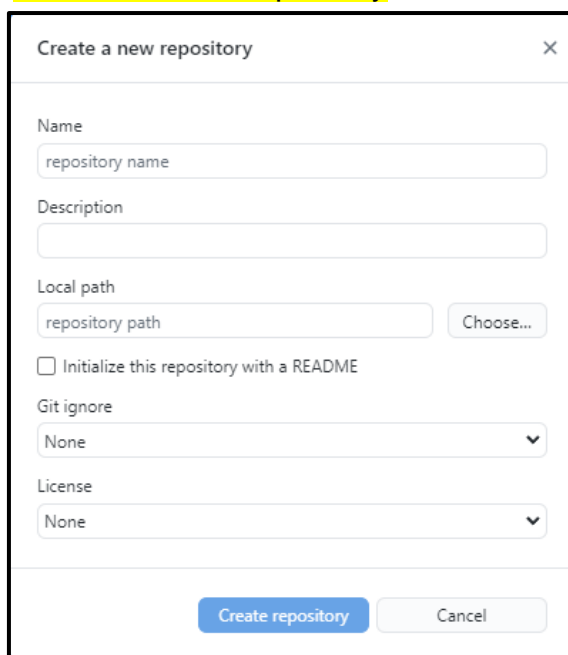
1. Hogyan használjuk a gitet?

A git felületén **repository**-kat (röviden: **git repo**) fogunk létrehozni, majd kezelni. Lefordítva *raktárat* jelent, ebben tároljuk az adatainkat – fájljainkat, néhány beállításainkat, fájlok korábbi mentéseit. Így nem meglepő talán, hogy első lépés egy git repo létrehozása.²

Repository létrehozása

A GitHub Desktop alkalmazást megnyitva a felül található menüsoron a következőkre kattintva létrehozhatok egy git repo-t.

➤ File → New Repository



Az 1. ábrán látható felugró ablakkal találkozhattok. Itt lehetőség van a git repo-t elnevezni, rövid leírást adni, megadni a helyet – ahol a fájlokat szeretnétek tárolni a gépeteken.

Következőként létrehozhattok egy [README.md](#) fájlt is (jelölőnégyzet bepipálásával), ahol általában a következő információkról írunk:

- fájl(ok) használata
- telepítési útmutatók
- lista egyéb fájlokról a repo-ban
- kapcsolattartási infók ... stb.

1. ábra: git repo létrehozása

[Git ignore](#) során megadhatjuk, hogy mely fájlokat, mappákat kerülje ki a Git a projektünkben (repo-nkban). (Ez most nekünk nem lesz fontos. Maradhat *None*.)

Végül beállíthatjuk a repo [License](#)-t, vagyis megadhatjuk, hogy mások mit és mit nem tehetnek meg a forrás kódunkkal. (A Git-et alapjáraton arra használjuk, hogy nyílt forráskódú szoftvereket adjunk ki, viszont ahhoz, hogy ez tényleg, hivatalosan is az legyen, licenszelni kell. Jelenleg ez sem fontos, így nyugodtan maradhat *None*.)

Utolsó sorban pedig a **Create repository** gomb lenyomásával létrehoztuk raktárunkat.

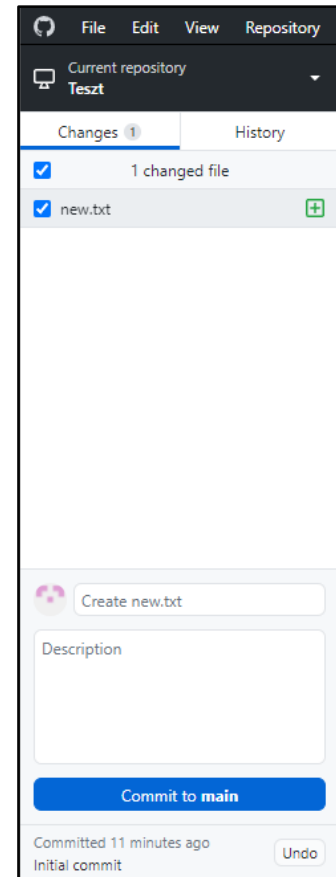
² Ahogy említettem, csak az alkalmazást használva dolgozunk, viszont itt egyszer jelzem, hogy git repo létrehozását böngészőn, parancssoron is meg lehet tenni, ahogy a többi lépést is szintén.

Miután létrehoztunk egy git repo-t, elkezdhetünk fájlokat létrehozni, és tárolni benne. Ehhez nincs másra szükségünk mint, hogy megkeressük a mi kis raktárunkat (git repo-t) és dolgozunk benne. Minden módosítást – fájl létrehozását, annak módosítását, vagy csak akár a README babrálását is – a git (GitHub Desktop esetünkben) érzékelni fog.

Módosítások után hasonló dolgot láthatunk, mint a 2. ábrán (ahol fájlkezelőn keresztül létrehoztam egy new.txt-t).

A git repo neve (Teszt) alatt láthatjuk az előbb említett módosítást a **Changes** fülön. Itt infót kaphatunk arról, hogy mi módosult, jobb oldalt megtekinthetjük ezt a változást is, illetve a Changes felírat mellett látjuk, hogy hány módosítás történt.

Minden módosítások sorozata után, amelyeket szeretnénk rögzíteni, a változtatások kommitolásával tehetünk meg. Egy **commit** egy pillanatképet ad vissza a Git-nek a repo-ban történt változtatásokról.



2. ábra: commit

Commit rögzítése

A 2. ábrán alul látható dobozban van lehetőségünk kommitolásra. Itt megadhatjuk a **commit nevét** – amit fel is ajánl, ez most: *Create new.txt* -, amely nekünk szolgál, ezzel rövid infót adunk magunknak milyen változtatás történt éppen. Illetve **rövid vagy hosszú leírást** adhatunk arról, hogy mi módosítást történt, mi volt ezek célja, avagy ki végezte ezeket.

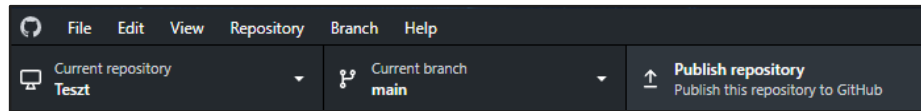
Ha ezeket az információkat megírtuk, akkor a **Commit to main** gomb lenyomásával elkészítettük aktuális pillanatképünket a projektunkról.

*[A "main" az alapértelmezett **branch** (magyarul: ág), ahol a változtatások mentésre kerülnek. Lehetőségetek van új ág létrehozására: ezt a felső menüsor alatt található Current branch: main földre kattintva tehetitek meg, majd a lenyíló ablakon a New branch gombra kattintva létrehozhattok egy új ágot. Új ág létrehozásakor általában akkor kerítünk sort, amikor szeretnénk új funkciókat vagy hibákat kijavítani, viszont nem szeretnénk, ha ezen változtatások miatt módosulna a fő ágon lévő munkánk.]*

Az elmentett változtatásokat a 2. ábrán is látható **History** fülön találjátok a többi commit mellett egy időrendi sorrendben. Itt egy commit-ot kiválasztva tudjátok megtekinteni a korábbi módosításokat.

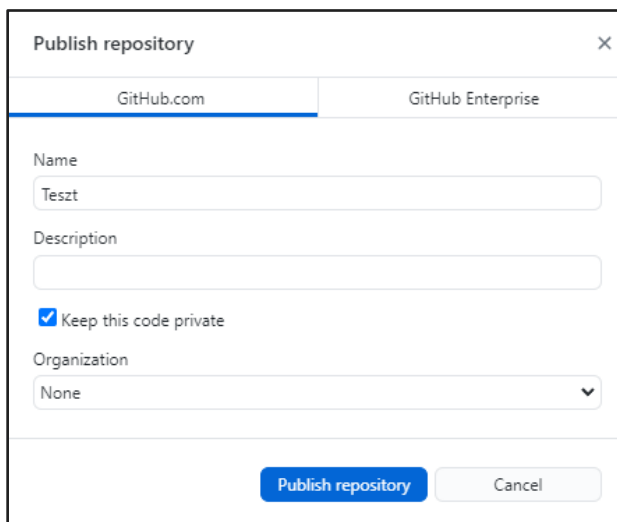
Repo közzététele, avagy push

Miután kommitoltunk – de még a git repo-nkat nem töltöttük fel a Git-re (az internetes szolgáltatásra) – lehetőségünk van a repo közzététele. Ez annyit tesz, hogy a git repo nem csak a gépünkön fog a továbbiakban létezni, hanem a **GitHub-on (böngészőn) is láthatjátok azt.**



3. ábra: publish repository

Ehhez a **Publis repository** fülre kell kattintani, majd az előugró ablakban újból beállíthatjuk a repo nevét, rövid leírását, beállíthatjuk (jelölőnégyzet bepipálásával), hogy a raktárunk maradjon privát, illetve, ha van egy Git Szervezetünk, akkor akként töltjük fel.



4. ábra: publish repository to Git

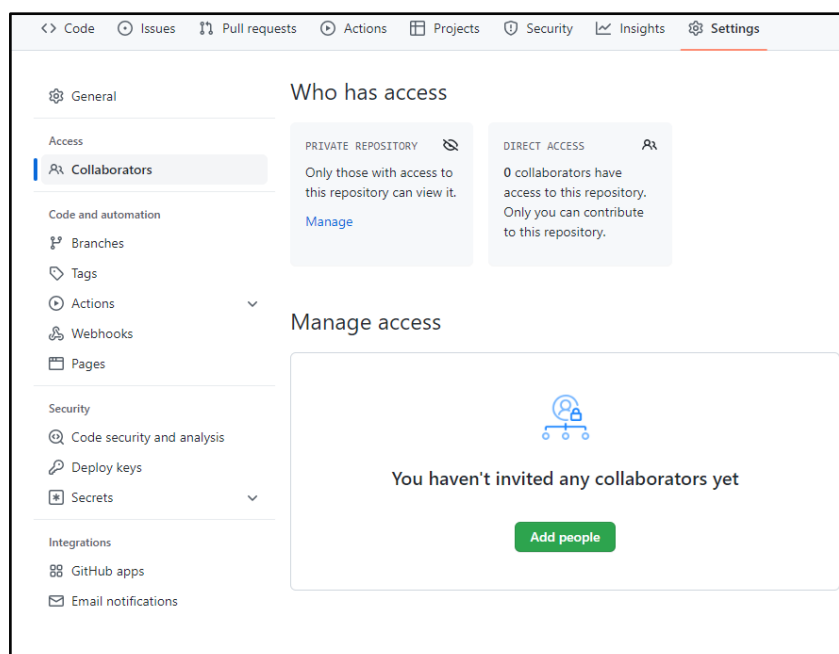
Azzal, hogy a kódunkat privátra állítjuk, úgy csak mi láthatjuk, férhetünk hozzá. Ahhoz, hogy más is lássa ezen úton, más git felhasználót is hozzá kell rendelnünk a projekthez.

Ezt legkönnyebben a github.com oldalon tehetitek meg, ahol megkeresitek a projektet, majd az 5. ábrán látható **Add people** gombra nyomva felveszitek ő(ke)t.

Ide legkönnyebben a **Settings** → **Collaborators** úton juthattok el.

Ellenkező esetben, ha már a git repo feltöltésre került a GitHub Desktop alkalmazásban a **Push origin** szöveg jelenik meg, mellyel szintén feltöltitek a commit-ban rögzült változtatásokat.

5. ábra: Add people



Miután ezt megtettétek, a **Fetch origin** szöveg jelenik meg a fülön, mellyel lehetőségetek van ellenőrizni, hogy a Git-re került-e fel új változat – olyan módosítások sorozata a Git-en, amelyek a saját gépeteken még nincs rajta.

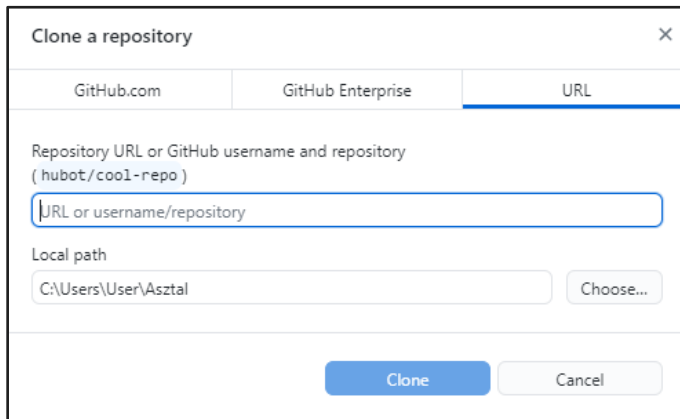
Új commit(ok) letöltése, avagy a pull

Amennyiben a fent említett igaz, a **Pull origin** szöveg fog megjelenni egy Fetch origin lenyomása után. Magyarán letöltitek a legfrissebb változatot a projektből, ami a Git-re felkerült.

Repository másolása

Amennyiben a git repo-t nem te hoztad létre, de kaptál hozzáférést, akkor lehetőség van leklónozni a Git-en megtalálható repo-t.

Ezt a GitHub Desktop menüsorán, a **File → Clone** repository-ra kattintva előugró ablakon tehetitek meg.

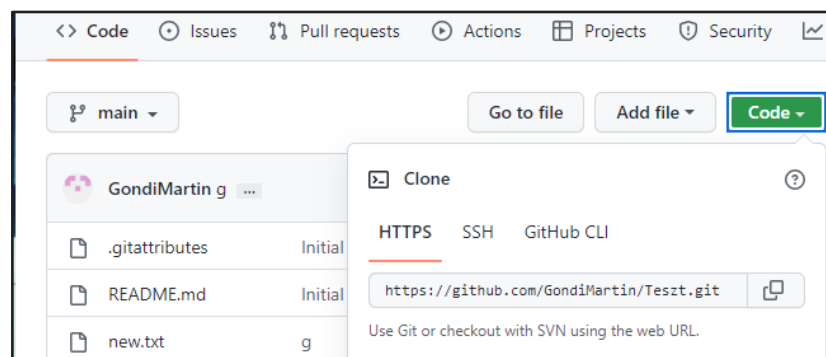


6. ábra: Clone a repository

A 6. ábrához hasonló ablakot kell lássatok, ahol az **URL fület** kiválasztva megadjátok a **másolni kívánt git repo linkjét**, majd a helyet, ahol szeretnétek tárolni a repo-t.

Git repo linket a legkönnyebben a weboldalról másolhattok vágólapra.

Ehhez böngészőn keressétek meg a repo-t, majd a 7. ábrán is látható módon másoljátok vágólapra a repo linkjét.



7. ábra: Repo link

2. Röviden a lépések

1) **Repository létrehozása:** GitHub Desktop → Create repository

Megnyitjuk a GitHub Desktop alkalmazást. A felső menüsoron *File* felíratra kattintunk. Itt kiválasztjuk a *New Repository ...* opciót. A felugró ablakban **elvégezzük a szükséges beállításokat**. Ha mindent jól csináltunk a menüsor alatt található (balról) a harmadik fülön feltehetjük a repo-nkat a Git-re a *Publish repository* lenyomásának segítségével.

2) **Commit:** változások rögzítése GitHub Desktop-on keresztül

Amennyiben végeztünk módosításainkkal az alkalmazás **bal alsó** sarkában elvégezhetjük a kommitolást: commit nevének, majd rövid leírásának megadása, végül a *Commit to [ág_név]* gomb megnyomása.

3) **Push:** commit-ban szereplő módosítások feltöltése a Git-re

A menüsor alatt található három fül közül kiválasztjuk a legutolsót, amin a következő kell, hogy szerepeljen: *Push to origin*

4) **Fetch:** ellenőrizni, hogy történt-e új commit

Ezt az előbbi fülön keresztül érjük el: *Fetch origin*

5) **Pull:** az új commit(ok) letöltése saját gépünkre

Ha rajtunk kívül valaki más is töltött fel commit-ot a Git-re, akkor szintén az előbb meghatározott fülön keresztül érjük el a commit letöltését: *Pull origin*

6) **Clone:** másokkal való megosztáshoz

Menüsorban található *File → Clone repository* opciót kiválasztva, majd az előugró ablakon az **URL fülön keresztül** másolhatjuk gépünkre a kívánt repot. Ehhez adjuk meg a repo linkjét és a letöltési helyünket.

3. Órai feladat (vagy otthoni házi feladat)

A feladatot legalább ketten, maximum hárman kell megoldani. Segítséget nyújtsatok egymásnak. A feladatokat kezdetben (1-5. feladat) egy ember kezdje el csak.

1) Készíts egy új git repo-t.

- a. Neve legyen „**HomeWork**”.
- b. Tartalmazzon README fájlt.

2) Tedd elérhetővé a Git-en.

- a. A git repo legyen nyilvános.

3) Készíts a git repo-ban egy új fájlt.

- a. A fájl neve legyen „**teszt.txt**”.
- b. A fájl tartalmazza első sorban a nevedet.

4) Kommitold a git repo-ban történt változásokat.

- a. A commit neve legyen a következő: „**New .txt file**”.
- b. Rövid leírásként add meg, hogy a fájl a nevedet tartalmazza.

5) Töltsd fel a Git-re az elkommitolt módosításokat.

6) Oszd meg azon társaiddal a git repo-t, akikkel megbeszéltétek, hogy együtt dolgoztok.

- a. Ők is töltsék le a git repo-t saját gépükre.
- b. Módosítsák új a fájlt, hogy a külön sorban az (ők) saját neveiket is beleírják.
- c. Kommitolják a módosításokat.
- d. Töltsék fel a commit-ot.

7) Töltsd le társaid módosítását a Git-ről.

- a. Ellenőrizd, hogy mindenki neve szerepel-e a fájlban.

4. Git bash használata

1) Repository létrehozása:

<u>Parancs</u>	<u>Mit csinál?</u>
<code>cd [könyvtár neve]</code>	Átlépünk a megadott könyvtárba.
<code>git config --global user.name „Saját Név”</code>	GitHub-név megadása és beállítása a repo-hoz.
<code>git config --global user.email „sajat@email.cim”</code>	GitHub-email-cím megadása és beállítása a repo-hoz.
<code>git init [mappa neve]</code>	Létrehoz a jelenlegi könyvtáron belül egy megadott mappát. Ezen mappán belül jön létre a git repo.

2) Commit:

<u>Parancs</u>	<u>Mit csinál?</u>
<code>git add .</code>	A Git figyelmébe ajánljuk a mappában található összes fájlt.
<code>git add file1 file2</code>	A Git figyelmébe ajánlunk a mappában file1-et és file2-öt.
<code>git commit -m „változások leírása”</code>	Kommitolás, a commit nevének megadása.

3) Push:

<u>Parancs</u>	<u>Mit csinál?</u>
<code>git push</code>	Feltölti az elkommitolt módosításokat. Itt továbbiakban meg kell adni a GitHub nevünket és jelszónkat.

4) **Fetch:**

<u>Parancs</u>	<u>Mit csinál?</u>
<code>git fetch</code>	Letölti a Git-en található új változtatásokat, de még nem alkalmazza azokat.

5) **Pull:**

<u>Parancs</u>	<u>Mit csinál?</u>
<code>git pull</code>	Letölti és alkalmazza a Git-en található módosításokat.

6) **Clone:**

<u>Parancs</u>	<u>Mit csinál?</u>
<code>git clone [projekt-link]</code>	Lemásolja a megadott linken található git repo-t.

Tartalomjegyzék

GitHub ismertető	1
0. Bevezető	1
<i>Regisztráció.....</i>	<i>1</i>
<i>GitHub Desktop letöltése.....</i>	<i>1</i>
1. Hogyan használjuk a gitet?	2
<i>Repository létrehozása.....</i>	<i>2</i>
<i>Commit rögzítése</i>	<i>3</i>
<i>Repo közzététele, avagy push</i>	<i>4</i>
<i>Új commit(ok) letöltése, avagy a pull.....</i>	<i>5</i>
<i>Repository másolása</i>	<i>5</i>
2. Röviden a lépések.....	6
3. Órai feladat (vagy otthoni házi feladat).....	7
4. <i>Git bash használata</i>	<i>8</i>