**Készítette**: Dr. Subecz Zoltán

Neumann János Egyetem, GAMF Informatikai és Műszaki Kar, Informatika tanszék

[subecz.zoltan@gamf.uni-neumann.hu](mailto:subecz.zoltan@gamf.uni-neumann.hu)

**GITHUB-GITLAB használat**

Tartalomjegyzék

[Github vs Gitlab – amire nekünk kell mind a kettő jó 2](#_Toc158462594)

[Alapok – GitHub 2](#_Toc158462595)

[Git and GitHub 2](#_Toc158462596)

[Regisztráció 2](#_Toc158462597)

[Két fő használati lehetőség: CLI: Command line interface / Desktop: grafikus interfész: egyszerűbb használat => A Desktop –ot mutatom be 2](#_Toc158462598)

[Egyéb használati lehetőség: a nagyobb IDE alkalmazások is biztosítanak GitHub hozzáférést. pl. IntelliJ, VSCode, PHPStorm… 2](#_Toc158462599)

[Github Desktop telepítés, beállítás 3](#_Toc158462600)

[Első feladat 3](#_Toc158462601)

[Create a New Repository on your hard drive… 3](#_Toc158462602)

[Fájlok készítése, módosítása 4](#_Toc158462603)

[Készítsük el a proba1.txt és a proba2.txt fájlokat rövid tartalmakkal. 4](#_Toc158462604)

[Commit 4](#_Toc158462605)

[Módosítsuk a fájlokat => Commit. 5](#_Toc158462606)

[Töröljük az egyik fájlt => Commit. 5](#_Toc158462607)

[Tegyük láthatóvá (public) a repository-t mindenki számára 5](#_Toc158462608)

[Creating a branch 5](#_Toc158462609)

[Create a branch 6](#_Toc158462610)

[Making and committing changes 6](#_Toc158462611)

[Opening a pull request 7](#_Toc158462612)

[Merging your pull request 8](#_Toc158462613)

[Töröljük a readme-edits branch-ot 8](#_Toc158462614)

[Hozzunk létre egy új branch-ot branch2 néven 8](#_Toc158462615)

[Módosítás a branch-okon 8](#_Toc158462616)

[Második feladat – projektmunka másokkal - Contributing to projects 10](#_Toc158462617)

[Regisztráljanak egy másik email címmel is, egy másik böngészővel. Legjobb, ha egy másik gépen hozzuk létre 10](#_Toc158462618)

[About forking 10](#_Toc158462619)

[Forking a repository 10](#_Toc158462620)

[Cloning a fork 10](#_Toc158462621)

[Making and pushing changes 11](#_Toc158462622)

[Making a pull request 11](#_Toc158462623)

[Közreműködők megtekintése - Contributors 12](#_Toc158462624)

# Github vs Gitlab – amire nekünk kell mind a kettő jó

<https://www.geeksforgeeks.org/difference-between-gitlab-and-github/>

<https://about.gitlab.com/devops-tools/github-vs-gitlab/>

<https://usersnap.com/blog/gitlab-github/>

<https://www.upgrad.com/blog/github-vs-gitlab-difference-between-github-and-gitlab/>

<https://spectralops.io/blog/github-vs-gitlab/>

<https://kinsta.com/blog/gitlab-vs-github/>

<https://www.coscreen.co/blog/gitlab-vs-github/>

<https://rigorousthemes.com/blog/gitlab-vs-github-which-is-better/>

# Alapok – GitHub

<https://docs.github.com/en>

<https://docs.github.com/en/get-started/quickstart/hello-world>

## Git and GitHub

<https://product.hubspot.com/blog/git-and-github-tutorial-for-beginners>

A quick aside: git and GitHub are not the same thing. Git is an open-source, version control tool created in 2005 by developers working on the Linux operating system; GitHub is a company founded in 2008 that makes tools which integrate with git. You do not need GitHub to use git, but you cannot use GitHub without using git. There are many other alternatives to GitHub, such as GitLab, BitBucket, and “host-your-own” solutions such as gogs and gittea. All of these are referred to in git-speak as “remotes”, and all are completely optional. You do not need to use a remote to use git, but it will make sharing your code with others easier.

# Regisztráció

<https://github.com/>

Regisztráljanak egy felhasználót => Sign up

# Két fő használati lehetőség: CLI: Command line interface / Desktop: grafikus interfész: egyszerűbb használat => A Desktop –ot mutatom be

<https://docs.github.com/en/get-started/quickstart/set-up-git>

## Egyéb használati lehetőség: a nagyobb IDE alkalmazások is biztosítanak GitHub hozzáférést. pl. IntelliJ, VSCode, PHPStorm…

<https://www.jetbrains.com/help/idea/github.html>

<https://code.visualstudio.com/docs/editor/github>

<https://www.jetbrains.com/help/phpstorm/github.html>

# Github Desktop telepítés, beállítás

<https://docs.github.com/en/desktop/installing-and-configuring-github-desktop>

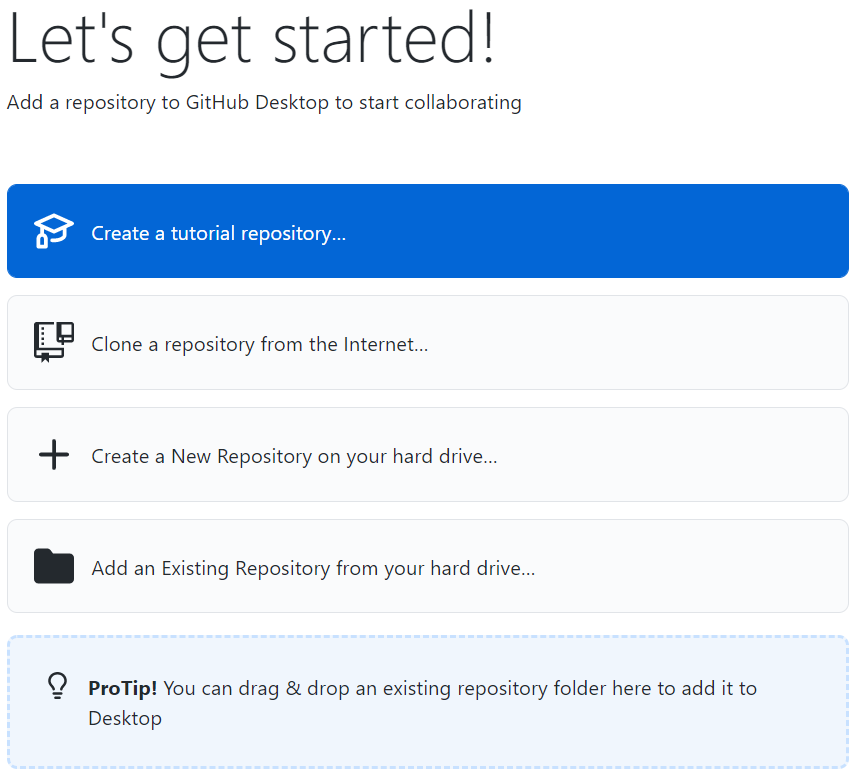
Töltsük le

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

Telepítsük

Indítsák a feltelepített GitHub Desktop programot.

Jelentkezzenek be => Sign in



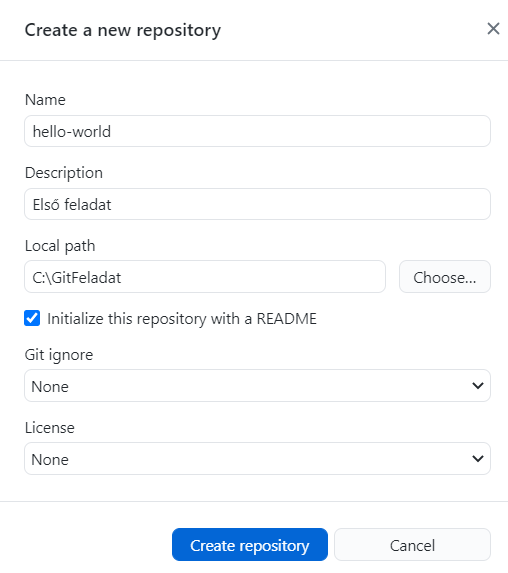
# Első feladat

<https://docs.github.com/en/get-started/quickstart/hello-world>

alapján

## Create a New Repository on your hard drive…

Hozzunk létre egy mappát a Repository-nak: pl. C:\GitFeladat

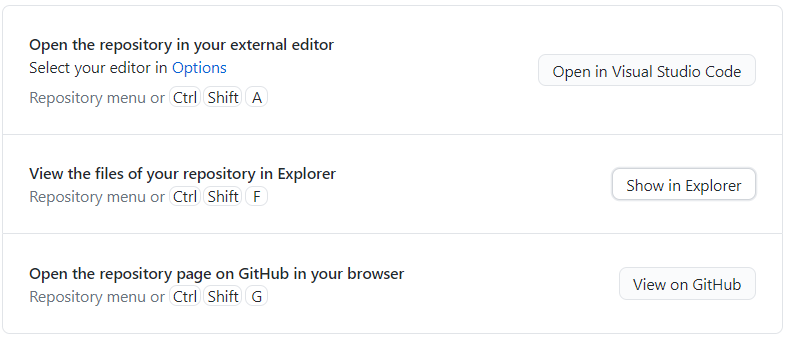


=> Create repository

=> Publish repository

Nézzük meg a **C:\GitFeladat** mappa tartalmát

Néhány lehetőséget felajánl:



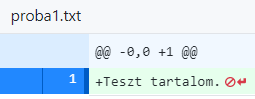
**Próbáljuk ki:**

Open in Visual Studio Code

**View on GitHub** => megnyitja a [https://github.com/\*\*\*\*\*\*/hello-world](https://github.com/******/hello-world) oldalt.

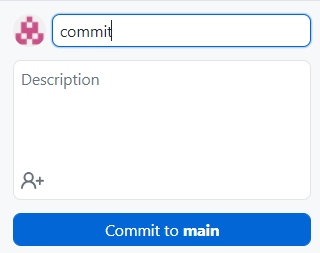
## Fájlok készítése, módosítása

### Készítsük el a proba1.txt és a proba2.txt fájlokat rövid tartalmakkal.

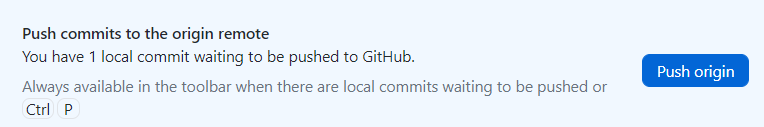


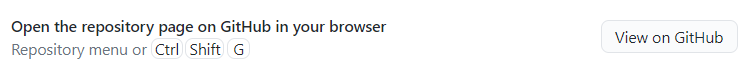
### Commit

A Desktop-ban: **commit:**



majd **Push origin:**



majd: **View on GitHub**:

### Módosítsuk a fájlokat => Commit.

### Töröljük az egyik fájlt => Commit.

## Tegyük láthatóvá (public) a repository-t mindenki számára

[https://github.com/\*\*\*\*/hello-world](https://github.com/****/hello-world) oldalon: Settings: 

=> Danger Zone

Change repository visibility (This repository is currently private.)

Change visibility => Make public

## Creating a branch

Branching lets you have different versions of a repository at one time.

By default, your repository has one branch named main that is considered to be the definitive branch. You can use branches to experiment and make edits before committing them to main.

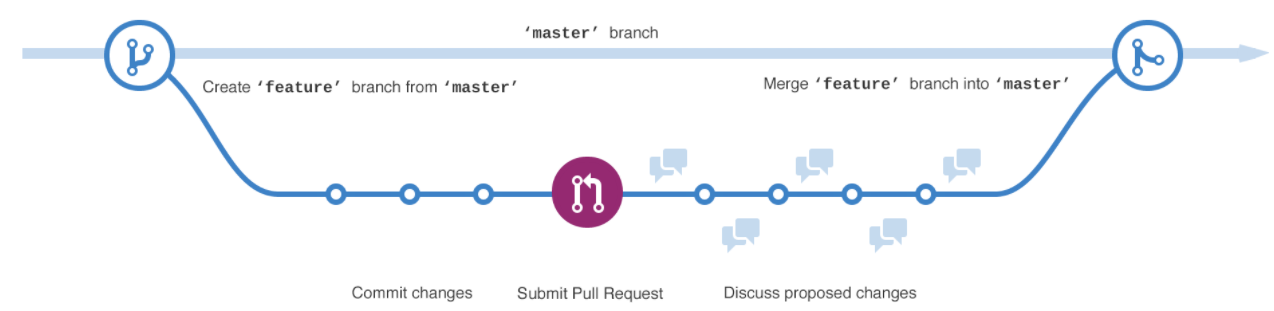
When you create a branch off the main branch, you're making a copy, or snapshot, of main as it was at that point in time. If someone else made changes to the main branch while you were working on your branch, you could pull in those updates.

**This diagram shows:**

The **main** branch

A new branch called **feature**

The journey that **feature** takes before it's merged into **main**



Have you ever saved different versions of a file? Something like:

* story.txt
* story-joe-edit.txt
* story-joe-edit-reviewed.txt

Branches accomplish similar goals in GitHub repositories.

## Create a branch

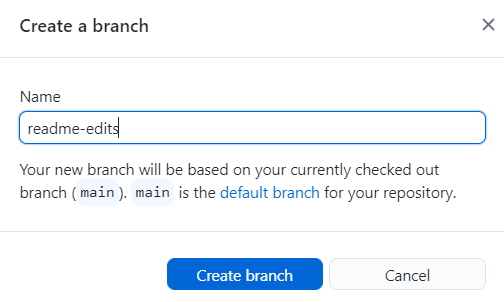
Click the Code tab of your hello-world repository.

Click the drop down at the top of the file list that says main.



Type a branch name, **readme-edits**, into the text box.

Click Create branch: readme-edits from main.

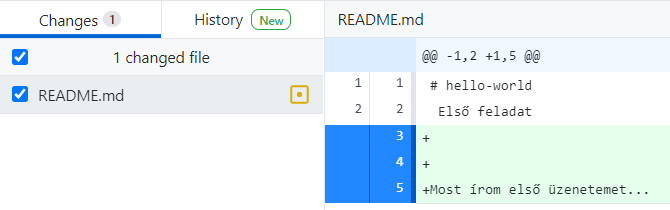
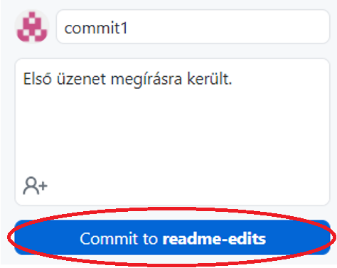
 

Now you have two branches, main and readme-edits. Right now, they look exactly the same. Next you'll add changes to the new branch.

## Making and committing changes

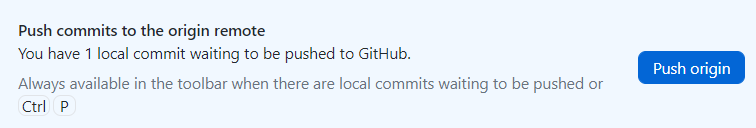
When you created a new branch in the previous step, GitHub brought you to the code page for your new readme-edits branch, which is a copy of main.

You can make and save changes to the files in your repository. On GitHub, saved changes are called **commits**. Each **commit** has an associated **commit message**, which is a description explaining why a particular change was made. **Commit messages** capture the history of your changes so that other contributors can understand what you’ve done and why.

* Click the README.md file.
* Click  to edit the file.  
  Jegyzettömbbel is szerkeszthetjük a fájlt.
* In the editor, write a bit about yourself.
* In the Commit changes box, write a commit message that describes your changes.  
   

Nézzük meg a [https://github.com/\*\*\*\*/hello-world](https://github.com/****/hello-world) oldalon a README.md fájl tartalmát a Commit előtt és után is.

* Click Commit changes.



These changes will be made only to the README file on your **readme-edits** branch, so now this branch contains content that's different from **main**.

## Opening a pull request

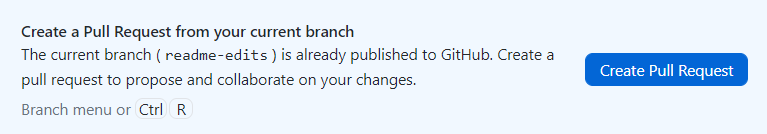
Now that you have changes in a branch off of main, you can open a pull request.

**Pull requests are the heart of collaboration on GitHub**. When you open a pull request, you're proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch. Pull requests show diffs, or differences, of the content from both branches. The changes, additions, and subtractions are shown in different colors.

As soon as you make a commit, you can open a pull request and start a discussion, even before the code is finished.

By using GitHub's @mention feature in your pull request message, you can ask for feedback from specific people or teams, whether they're down the hall or 10 time zones away.

You can even open pull requests in your own repository and merge them yourself. It's a great way to learn the GitHub flow before working on larger projects.



* Click the Pull requests tab of your hello-world repository.
* Click New pull request
* In the Example Comparisons box, select the branch you made, readme-edits, to compare with main (the original).
* Look over your changes in the diffs on the Compare page, make sure they're what you want to submit.



* Click Create pull request.
* Give your pull request a title and write a brief description of your changes. You can include emojis and drag and drop images and gifs.
* Click Create pull request.

## Merging your pull request

In this final step, you will merge your readme-edits branch into the main branch.

* Click Merge pull request to merge the changes into main.
* Click Confirm merge.

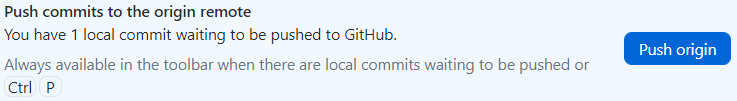
Egyesítés után nézzük meg a main branch-t: a módosítások ott is megjelentek.

### Töröljük a readme-edits branch-ot

* Go ahead and delete the branch, since its changes have been incorporated, by clicking Delete branch.

### Hozzunk létre egy új branch-ot branch2 néven

### Módosítás a branch-okon

* Válasszuk ki a **branch2** branch-ot
* Készítsünk egy fájt rövid tartalommal + Commit
* Nézzük meg a mappa tartalmát: ott van a fájl
* Nézzük meg a távoli szerveren: nincs ott a fájl
* 
* Nézzük meg a távoli szerveren: ott van a fájl
* Nézzük meg a távoli szerveren a **main** branch-nál: nincs ott a fájl
* Menjünk át a **main** branch-ra
* Nézzük meg a mappa tartalmát: nincs ott a fájl
* Menjünk vissza a **branch2** branch-ra
* Create pull request
* Merge pull request
* Nézzük meg a távoli szerveren a **main** branch-nál: ott van a fájl
* Menjünk át a **main** branch-ra
* Nézzük meg a mappa tartalmát: nincs ott a fájl
* 
* Nézzük meg a mappa tartalmát: nincs ott a fájl
* 
* Nézzük meg a mappa tartalmát: ott van a fájl

# Második feladat – projektmunka másokkal - Contributing to projects

Learn how to contribute to a project through forking.

## Regisztráljanak egy másik email címmel is, egy másik böngészővel. Legjobb, ha egy másik gépen hozzuk létre

Azért jobb másik gépen, mert a Desktop programban maradnak beállítások az előző állapotból.

<https://github.com/desktop/desktop/issues/3707>

<https://docs.github.com/en/get-started/quickstart/contributing-to-projects>

## About forking

Creating a "fork" is producing a personal copy of someone else's project. Forks act as a sort of bridge between the original repository and your personal copy. You can submit pull requests to help make other people's projects better by offering your changes up to the original project. Forking is at the core of social coding at GitHub.

## Forking a repository

* Menjen az előzőleg elkészített oldalra: [https://github.com/\*\*\*\*/hello-world](https://github.com/****/hello-world)
* Kattintson a Fork-ra:



* A GitHub készít egy másolatot a hello-world repository-ról
* Elkészül a Fork oldal: [https://github.com/\*\*\*\*\*ÚJ/hello-world](https://github.com/*****ÚJ/hello-world)

## Cloning a fork

A hello-worlds repository másolása (fork) sikerült, de ez még csak a GitHub-on van jelen. Hogy dolgozni tudjunk vele klónozni kell a helyi gépre.

**Ha ugyanazt a gépet használjuk a másik felhasználóval is**

A **GitHub Desktop** programon jelentkezzünk ki és be az új regisztrációs adatokkal.

File menü/ Options / Sign out majd Sign in

Ha nem adja a bejelentkezési oldalt a böngészőben, akkor lépjünk be először azon a böngészőn a Git URL címet megadva, amit megnyitott

**Állítsuk be az új felhasználó adatait a Desktop-ban**

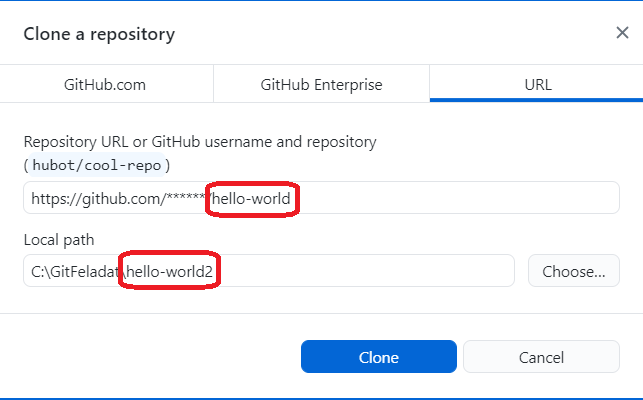
Ha ezt nem állítjuk be, akkor is el tudjuk küldeni az új fájlt, de a [https://github.com/\*\*\*\*/hello-world](https://github.com/****/hello-world) oldalon úgy látjuk, hogy ugyanaz a személy készítette a fájlt.

File menü/ Options / Git / Name és Email –nél már az új felhasználó adatai legyenek.

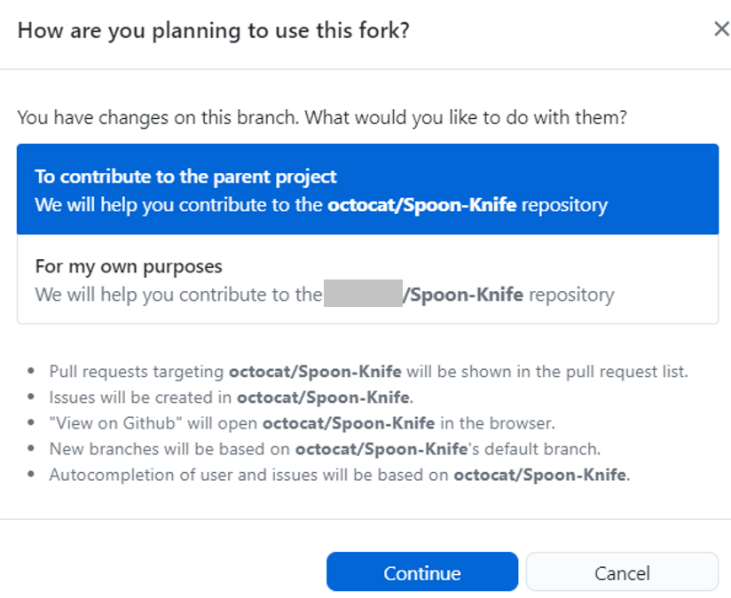
Ha ugyanazt a gépet használjuk a másik felhasználóval is, akkor fogjuk látni az előző felhasználó repository-ait is a Desktop-ban. Ezért jobb ha egy másik gépet használnak a másik felhasználóval. De így is meg tudjuk oldani a feladatot: nem hello-world, hanem hello-world2 néven készítjük itt a repository-t.

* Menjünk az oldalra: [https://github.com/\*\*\*\*\*ÚJ/hello-world](https://github.com/*****ÚJ/hello-world)
* 
*  Desktop-al végezzük el a klónozást
* Engedélyezés

Olyan mappa nevet adjunk, ami még nincs a gépen:

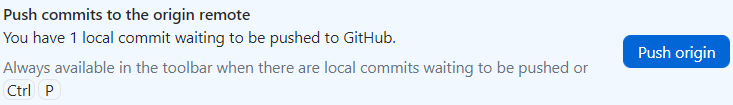


* Clone => Nézzük meg a létrehozott mappát



**A továbbiakban ugyanaz a folyamat, amit az előző fejezetben megismertünk:**

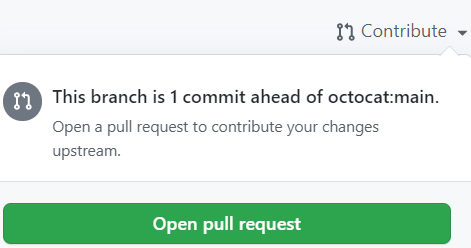
## Making and pushing changes

* Hozzunk létre egy **proba10.txt** fájlt rövid tartalommal
* Commit
* 
* Nézzük meg a fork oldalon: [https://github.com/\*\*\*\*ÚJ/hello-world](https://github.com/****ÚJ/hello-world)

ott is megjelenik a fájl.

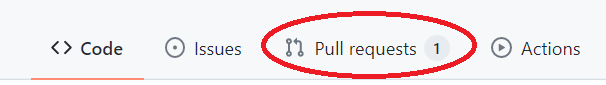
## Making a pull request

At last, you're ready to propose changes into the main project! This is the final step in producing a fork of someone else's project, and arguably the most important. If you've made a change that you feel would benefit the community as a whole, you should definitely consider contributing back.

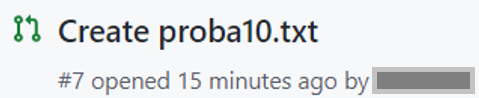
* [https://github.com/\*\*\*\*ÚJ/hello-world](https://github.com/****ÚJ/hello-world) oldalon:
* Contribute: 
* 
* Create pull request

A tulajdonosnak engedélyezni kell az egyesítést:

Menjünk át az eredeti oldalra: [https://github.com/\*\*\*\*/hello-world](https://github.com/****/hello-world)



Kattintsunk a pull request nevére:



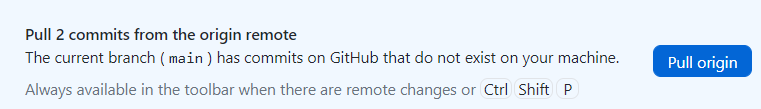


A [https://github.com/\*\*\*\*\*\*/hello-world](https://github.com/******/hello-world) oldalon látszik, hogy megjelent a proba10.txt fájl.

Hogy a helyi c:\GitFeladat\hello-world\ mappában is megjelenjen a fájl:

Desktop-on jelentkezzünk be az első felhasználóval.





## Közreműködők megtekintése - Contributors

A [https://github.com/\*\*\*\*\*\*/hello-world](https://github.com/******/hello-world) oldalon több helyen is meg lehet nézni

* Fő oldal jobb alsó sarok
* egy fájlnévre, vagy mappanévre kattintva
* 
*  / Contributions, Networks