

Új repository létrehozása


Create a new repository


A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * ancsapro / Repository name * Tutorial ✓

Great repository names are short and memorable. Need inspiration? How about [didactic-system?](#)

Description (optional)

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.


Initialize this repository with:
Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☒ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

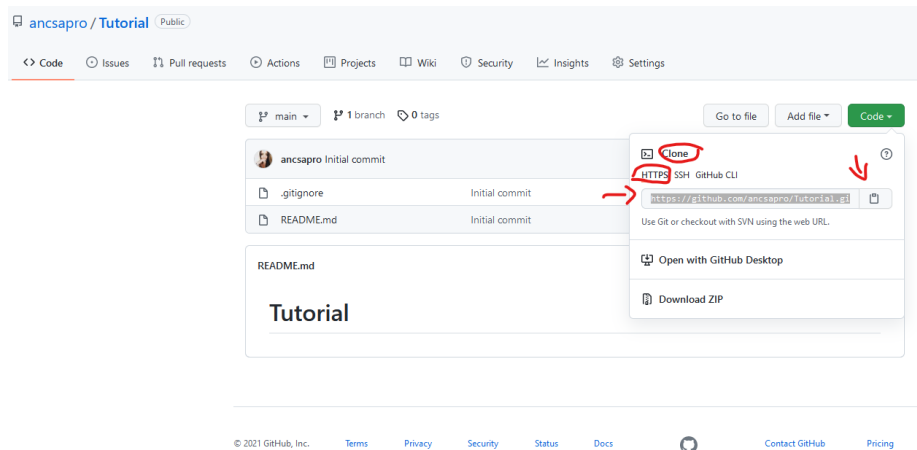
.gitignore template: C++ ←

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

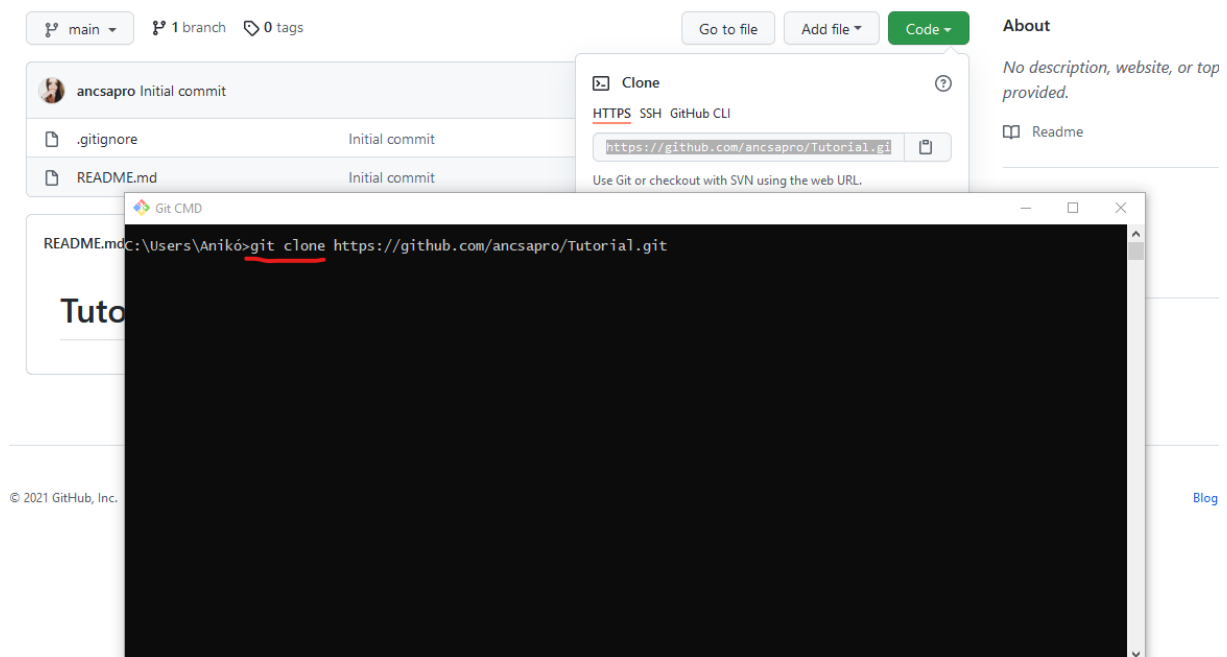
Create repository

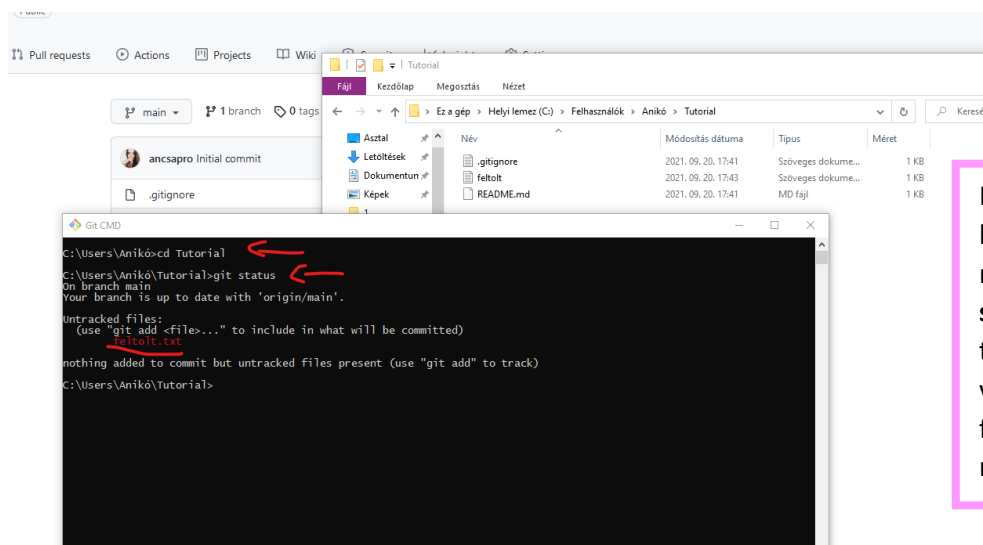
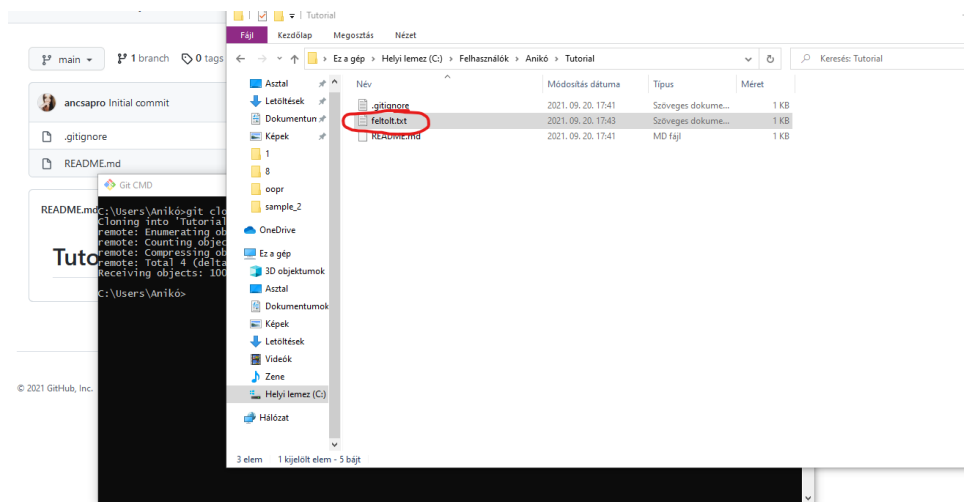
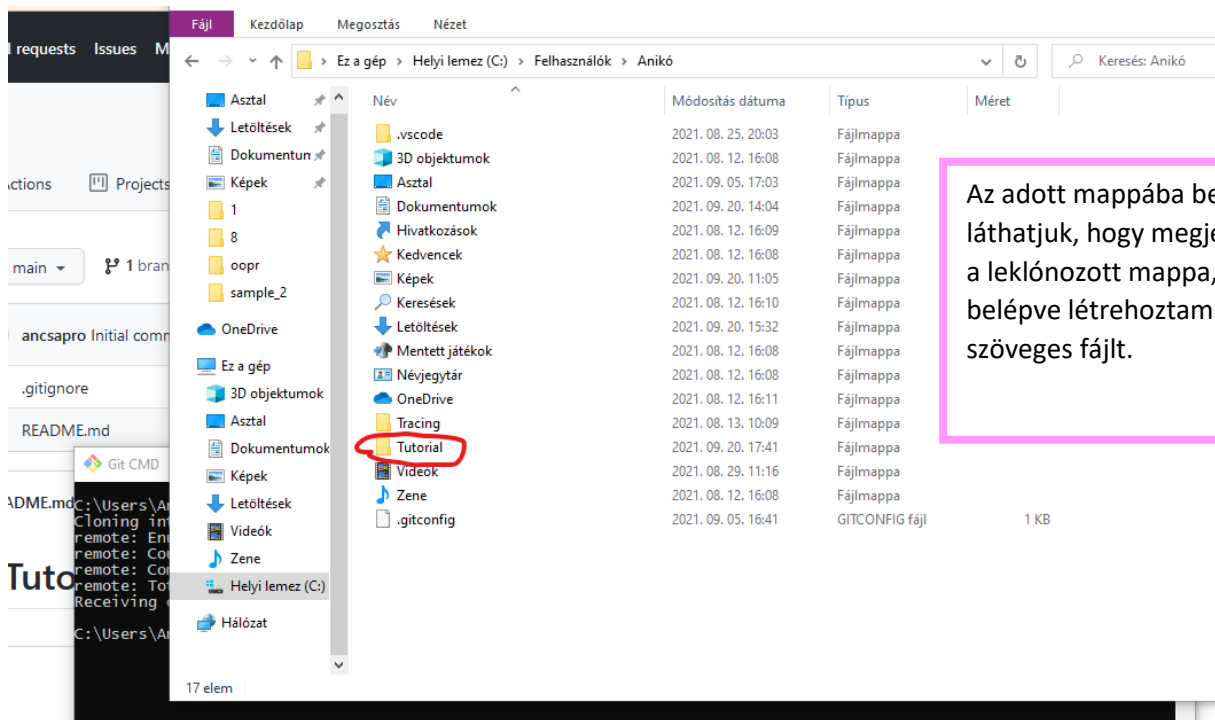
Drillek feltöltéséhez fontos, hogy public legyen, legyen benne README és add gitignore programozási nyelvnek megfelelően



Code fölön a HTTPS alatti linkkel tudjuk leklónozni a repository-t a **git clone** *link* parancssal.

FIGYELJÜNK, HOVA KLÓNOZZUK LE!





Ezután terminálba belépünk a leklónozott mappába, majd a **git status** paranccsal meg tudjuk nézni a változásokat. (megjelent a feltolt.txt állomány, amit még nem töltöttünk fel)

```
C:\Users\Anikó>cd Tutorial
C:\Users\Anikó\Tutorial>git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    feltolt.txt

nothing added to commit but untracked files present (use "git add" to track)
C:\Users\Anikó\Tutorial>git add feltolt.txt
C:\Users\Anikó\Tutorial>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   feltolt.txt

C:\Users\Anikó\Tutorial>
```

A **git add** fájlnev
paranccsal hozzáadjuk
a feltöltéshez. Majd a
git status parancs
újbol kiadása után
láthatjuk, hogy új fájl
adtunk hozzá.

```
C:\Users\Anikó\Tutorial>git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    feltolt.txt

nothing added to commit but untracked files present (use "git add" to track)
C:\Users\Anikó\Tutorial>git add feltolt.txt
C:\Users\Anikó\Tutorial>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   feltolt.txt

C:\Users\Anikó\Tutorial>git commit -m "first upload"
[main 509324e] first upload
1 file changed, 1 insertion(+)
create mode 100644 feltolt.txt

C:\Users\Anikó\Tutorial>
```

git commit -m
"message log"
paranccsal
véglegesítjük a
 tranzakciót, a -m
kapcsolóbal idézőjelek
között feltöltési
üzenetet hagyhatunk.
(erősen ajánlott)

Végül a **git push**
paranccsal (és némi
azonosítás után kerül
fel a fájl a repository-
ba)

A mappa frissebb
tartalmát a terminálba
a mappába belépbe és
git pull parancs
kiadásával tudjuk up-to-
datelni.