



# Git képzés

## Tematika

A kezdő szinten csak lokális repóban dolgozunk és eljutunk a branchek fogalmához, valamint a branchek összeolvasztásának módszereihez.

A haladó szinten megtanuljuk, hogyan dolgozzunk távoli repokkal. Végig vesszük a különböző szervezeti struktúrákhoz illeszkedő folyamatokat. Megnézzük milyen repok/branchek legyenek, ha csak páran dolgozunk egy projekten, ha elosztott teamek vannak, vagy ha külsős céggel kell együtt dolgozni, és ha különböző csoportoknak különböző jogosultságokat szeretnénk kiosztani.

Opcionálisan választható az “interactiv rebase” kiegészítő oktatás. Ez abban segíthet, hogy a fejlesztési folyamat során “összekoszolódott” brancheket képesek legyünk rendbe rakni. Végig vesszük, hogy hogyan tudunk egy korábbi commit-ot módosítani vagy akár törölni. Fel is cserélhetünk commitokat, többet egybe gyúrhatunk, vagy akár egyből csinálhatunk többet. Bármilyen lehetséges!

### 1. Kezdő 1 x 4 óra

- 1.1 Ismerjük meg a 3 fat: repo/stage/workdir
- 1.2 Készítsünk repot: ``git init``
- 1.3 Az ember legjobb barátja: ``git status``
- 1.4 Az első fájlunk: ``git add``
- 1.5 Mentsük el mindent: ``git commit``
- 1.6 Nyissuk ki a szerszámosládát: plumbing parancsok
- 1.7 Túrjunk bele a .git könyvtárba: minden csak egy file
- 1.8 Nézzünk bele a stage-be: ``git ls-files``
- 1.9 Módosítsuk a fileokon
- 1.10 Mit is csináltunk: ``git diff``
- 1.11 Commituljunk meg
- 1.12 Most csináljunk új filet
- 1.13 Most már van egy kis múltunk is: ``git log``
- 1.14 Ismerjük meg a referenciákat: HEAD





- 1.15 Ki volt az őse? Kalap, kalap, kal ... kabat.
- 1.16 Vissza a jövőbe: ``git rest``
- 1.17 Soft/Mixed/Hard reset, melyik mit jelent.
- 1.18 Rakjunk mindenhova cetlit: ``git tag``
- 1.19 Fixáljunk bugot: ``git branch``
- 1.20 Ugráljunk a branchek között: ``git checkout``
- 1.21 Hogyan fésüljük össze brancheket: ``git merge``
- 1.22 Így lesz gyorsan spagettink.
- 1.23 Hogyan lehet fésülni még: ``git rebase``
- 1.24 Mi a fontosabb a történelem, vagy az egyenes út?

## 2. Haladó 1 x 4 óra

- 2.1 Ha máshol van a kód: ``git clone``
- 2.2 Milyen protokollok vannak: file/http/https/ssh/git
- 2.3 Mit tudunk a remote-okrol: név + url
- 2.4 Hosszú branch nevek, mi a szósz az a per jel:  
``origin/master``
- 2.5 Hogyan updateljük a lokál repot? Nem, nem a ``git pull``
- 2.6 Akkor hogyan: ``git fetch``
- 2.7 A relatiGITas elmélete: pull = fetch + merge
- 2.8 Hogyan dolgozunk egyszerre több branchen.
- 2.9 Kódfésülés csapatszinten? Ki a leggyorsabb?
- 2.10 Az utolsó dolgozik a legtöbbet?!
- 2.11 Merge vagy Rebase az örök kérdés
- 2.12 Próbáljuk ki mind a kettőt, mik az előnyök hátrányok.
- 2.13 Mi az a forkolás?
- 2.14 Hogyan küldjük OpenSource projektbe javítást?
- 2.15 Branching stratégiák: gitflow vagy github-flow

## 3. Igény szerinti témakörök testre szabva 1 x 4 óra

