**Alapfogalmak**

**Kommit:**

Elemi egység, a fájlokról elmentett pillanatkép. Minél gyakrabban és minél kisebb logikailag összetartozó egységet kommitoljunk A kommitokank van gyereke és szülője. HEAD: utolsó kommit

**Branch**

Kommitok egymásutáni sorozata (kommit pötty, branch vonal) master: automatikusan létrejön

**Workspace**

Fejlesztésben használt fájlok, ezeken dolgozunk

**Repository**

Metadatok tárolása (kommitok, branchek, címkék)

**Installálás**

apt-get install git-core git-doc git-gui gitk

**Help**

git help

git help <parancs> # pl git help commit

**Beállítások**

git config --global user.name "Simon Balázs"

git config --global user.email "balazs.simon@intren.hu"

git config -l # lista

**Alias**

git config --global alias.co "checkout"

git config --global alias.st "status"

**Alap parancsok**

**Init**

Új git repository létrehozása

git init

**Status**

Változások listázása

git status

**Add, mv, rm**

File hozzáadása, áthelyezése, törlése

git add <filename>

git mv <oldfilename> <newfilename>

git rm <filename>

**Commit**

Változások mentése

git commit

git commit -m 'commit message'

git commit -a # staging area kihagyása

Kimaradt változás hozzáadása az utolsó kommithoz vagy kommit message átírása

git commit --amend

**.gitgnore**

Fájlok listája amit nem akarunk a verziókezelőben tárolni (pl konfig fájlok)

**Log**

Commitok listázása

git log

git log -p # lista diff-el

**Diff**

Változások megtekintése

git diff

git diff -w # whitespace karakterek figyelmen kívül hagyása

git diff <oldhash>..<newhash>

git diff <oldtag>..<newtag> # tag is használható

**Blame**

File változások listázása

git blame <filename>

**Tag**

Címkézés (pl programverzió), kommit azonosítására (checkout, diff)

git tag <cimke> # címke létrehozása

git tag # használt címkék listázása

git tag -d <cimke> # címke törlése

**Reset**

**Nem kommitolt módosítás eldobása**

git reset --hard

**Kommit törlése (utolsó kommit hash-t kell megadni, az előttelevőket törli)**

git reset --hard <hash>

**Stash**

**Módosítás mentése átmeneti tárolóba**

git stash save "message" # mentés

**Stash kommitok megtekintése**

git stash list # stash commitok listázása

git stash show <stash@\{0\}> # commit megtekintése, a kapcsos zárójeleket escape-elni kell!

git stash show -p <stash@\{0\}> # commit megtekintése diff-el

**Változások újraalkalmazása, stash törlése nélkül**

git stash apply <azonosito> # azonosító nélkül az utolsó kommit

**Változások újraalkalmazása, stash törlésével**

git stash pop <azonosito> # azonosító nélkül az utolsó kommit

**Kommit eldobása**

git stash drop <azonosito> # azonosító nélkül az utolsó kommit

**Branch**

**Listázás**

git branch

**Új branch létrehozása**

git branch <branchname>

**Branch átnevezése**

git branch -m <oldbranchname> <newbranchname>

**Törlés**

git branch -d <branchname>

git branch -D <branchname> # nem mergelt branch törlése

**Checkout**

Branchek közötti váltás

git checkout <branchname>

git checkout -b <branchname> # új branch létrehozása és abba átállás

**Merge**

Branchek összeolvasztása

git merge <branchname>

**További parancsok**

**fsck**

Repository metaadatainak ellenőrzése

git fsck

**gc**

Felesleges metaadatok törlése

git gc

**~ ^**

Kommitok elérési útvonala (git help rev-parse) ~ - kommit szülője függőleges irányban ^ - kommit szülője vízszintes irányban

**Cherry-pick**

Kommit másolása másik branchből (csak kicsi, hibajavító kommitot másoljunk!)

git cherry-pick <commithash>

**Staging area**

Workspace és a repository között elhelyezkedő tároló.

* git add-al lehet hozzáadni
* git commit innen kommitolja a változást
* git commit -a kihagyja a staging area-t

**git reset**

Változás törlése a staging area-ból (a változás nem törlődik)

git reset

**add -p**

File egy részletének kommitolása

git add -p

* ? súgó
* s szétdarabolja a módosítást
* n kihagyja a módosítást
* y hozzáadás a staging area-hoz

**Branch kezdőpontja**

git co -b <branchname> <commithash>

**Rebase**

Régi kommitok módosítása, módosítása, törlése (pusholt kommitoknál ne használjuk)

git rebase -i <commithash> # -i interactive

* sor törlése: kommit törlése
* sorrend változtatása: kommitok sorrendjének változtatása
* pick: kommitot a rendszer használja
* edit: kommitot a rendszer használja, de álljon meg a feldolgozás, hogy lehessen (--amend), folytatás: git rebase --continue
* squash: kommit összeolasztása az előző kommittal

**Távoli repository**

**remote**

Távoli repository-k listázása.

git remote

**show**

Távoli repository adatainak megtekintése.

git remote show <name>

**add**

Távoli repository hozzáadása a local repository-hoz. Az alapértelmezet távoli repository-t konvenció szerint *origin* névvel szoktuk létrehozni.

git remote add <name> <url>

**remove**

Távoli repository kapcsolat törlése

git remote rm <name>

**rename**

Távoli repository kapcsolat nevének törlése

git remote rename <oldname> <newname>

**clone**

Távoli repository lemásolása local gépre. Automatikusan hozzádódik a távoli repository *origin* néven.

git clone <url>

**fetch**

Távoli repository változásainak letöltése

git fetch <remotename> <branchname>

A letöltött változásokat nem mergeli bele a local branchbe, kézzel kell mergelni

**pull**

Távoli repository változásainak letöltése és mergelése

git pull <remotename> <branchname>

**push**

Lokális változások feltöltése távoli repository-ba

git push <remotename> <branchname>