

GIT rendszer alapjai

Készítette: Varga Dávid



Bemutakozás

- Varga Dávid
- Műszaki kommunikációs management mérnöke
- QA Software developer – ICB-TECH



A Workshop felépítése

Git teória

- Bevezető
- Történelem
- Fejlesztése
- Felépítése
- Parancsok
- Elméleti példa

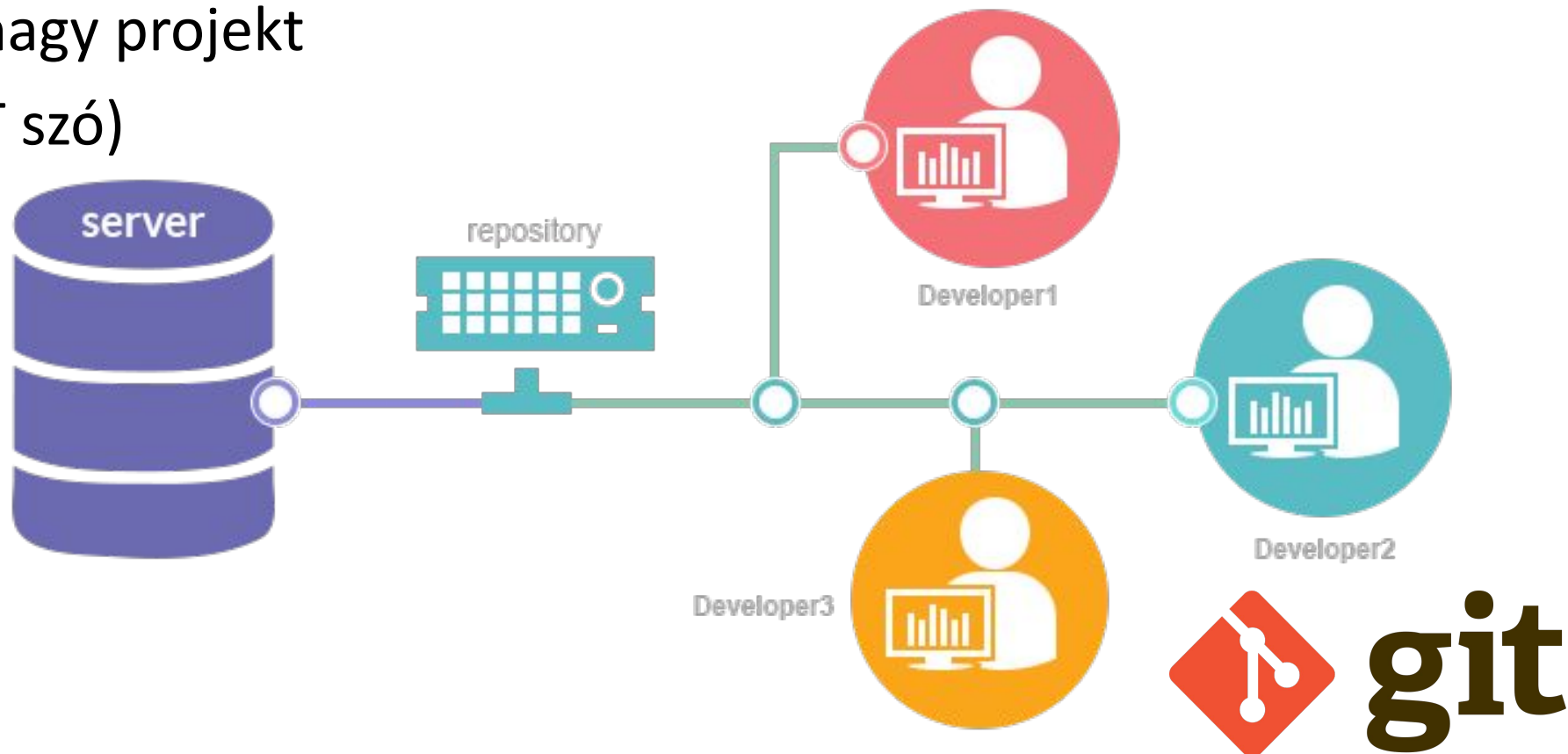
Git gyakorlat

- Git parancsok alkalmazása



GIT bevezető

- verziókezelő rendszer
- kis projekt vs nagy projekt
- piaci oldal (GIT szó)



Milyen lehetőségeket kínál?

- file szerkesztés követése
- verzió listázás
- verzió visszaállítás
- mozgás oda-vissza a verziók között
- logolás



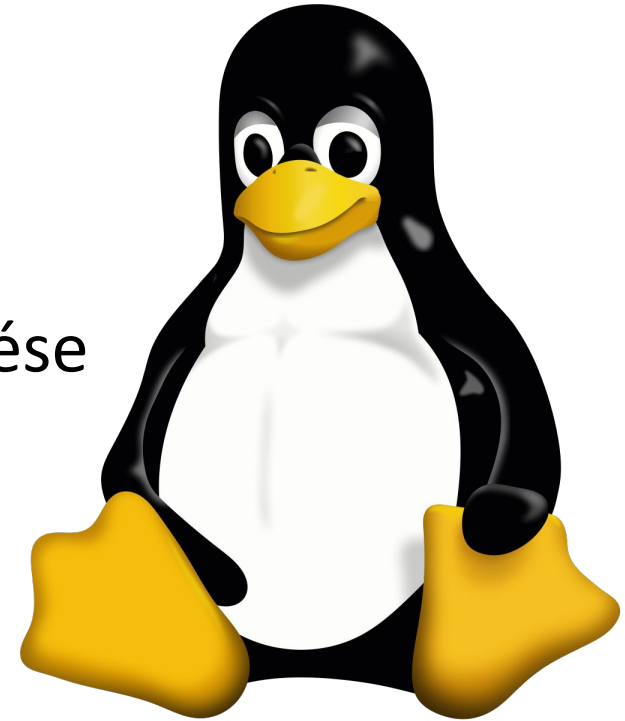
Miért népszerű?

- decentralizált működés
- optimalizált teljesítmény
- nem igényel hálózati hozzáférést
- nem igényel hálózati központi szervert
- lokális könyvtárak:
 - teljes adattárral
 - előzményekkel
 - verzió követéssel



Története

- 2005 – Linus Torvalds
- Linux fejlesztője, megalkotója
- cél: linux kernel verziók tömeges együttműködése
- BitKeeper
- ingyenes majd 2005-től fizető
- más verziókezelő szoftverek:
 - centralizált modell
 - 30s egy commit



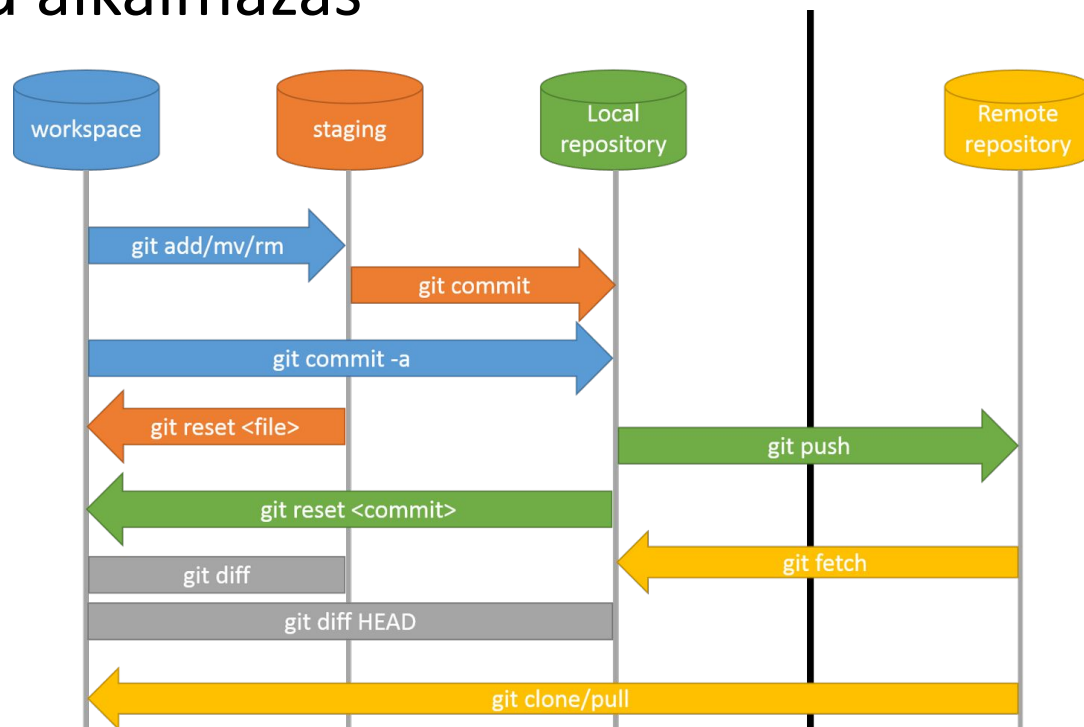
A GIT kifejlesztése

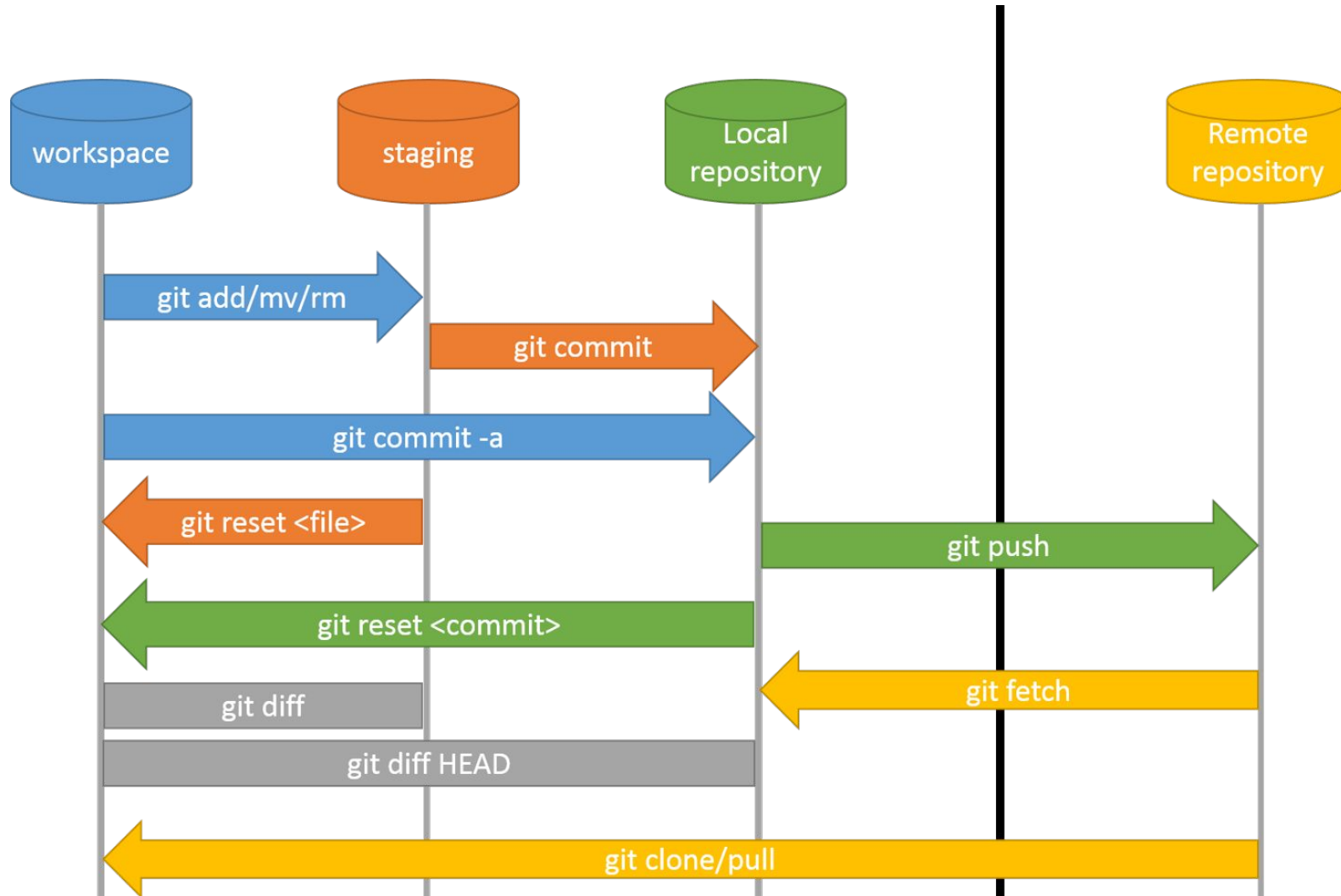
- meggyengült verzió kezelői körülmények
- egy saját megalkotása
- cél: megosztott verziókezelő és ingyenes
- GIT kifejlesztése
 - 6-7s egy commit
 - decentralizált rendszer
 - hálózat és központi szerver független
- A git jelentése: „*kellemetlen ember*”



Felépítése

- komplikált kinézet, de egyszerű alkalmazás
- egyszerű parancsok
- Repository vagy Adatár
 - Az összes file-ok gyűjteménye
 - projekt név alatt





Három fő szakasz:

- Workspace – munkakörnyezet
- Staging - ideiglenes verzió
- Local repository – állandó verzió

Negyedik szakasz:

- Remote repository
- GitHub
- GitLab
- Csapatmunka



Workspace - munkakörnyezet

- file-ok szerkesztése
- törlése
- kódolás kód szerkesztővel
- szabadon alakítható – a GIT nem követi



Staging – átmeneti verzió

- szerkesztés befejezése (vagy annak egy része)
- hozzá lehet adni a szakaszhoz
- átmeneti verzió keletkezik
- átmeneti index



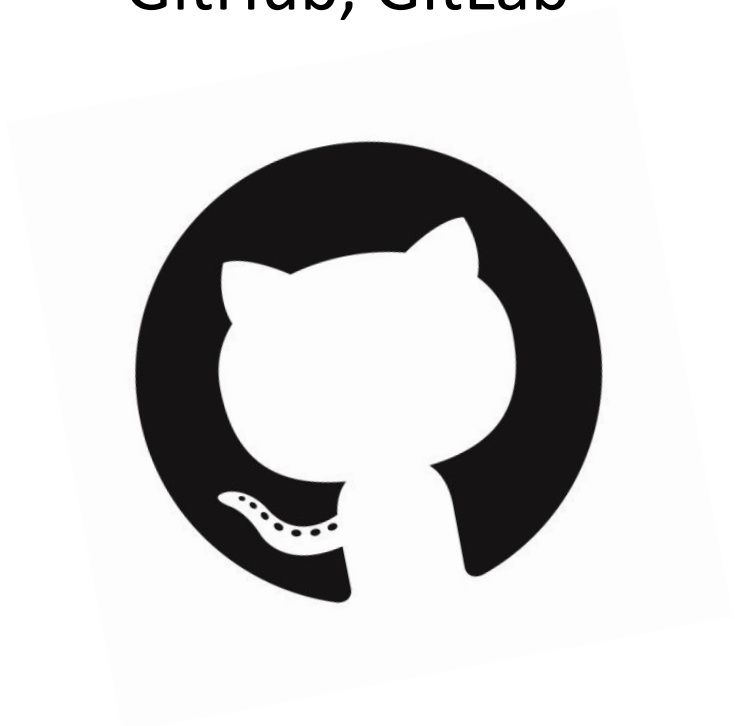
Local repository – állandó verzió

- átmeneti szakasz -> állandó verzió (local)
- file elérí teljes szakaszát (pl. hotfix vagy feature)
- commit -> művelet
- Részletek:
 - Fejlesztő
 - Commit
 - Commit id
 - Dátum
 - Commit üzenet
 - Kriptográfiai hash
- Fontos: commit után az átmeneti verzió (staging) törlődik

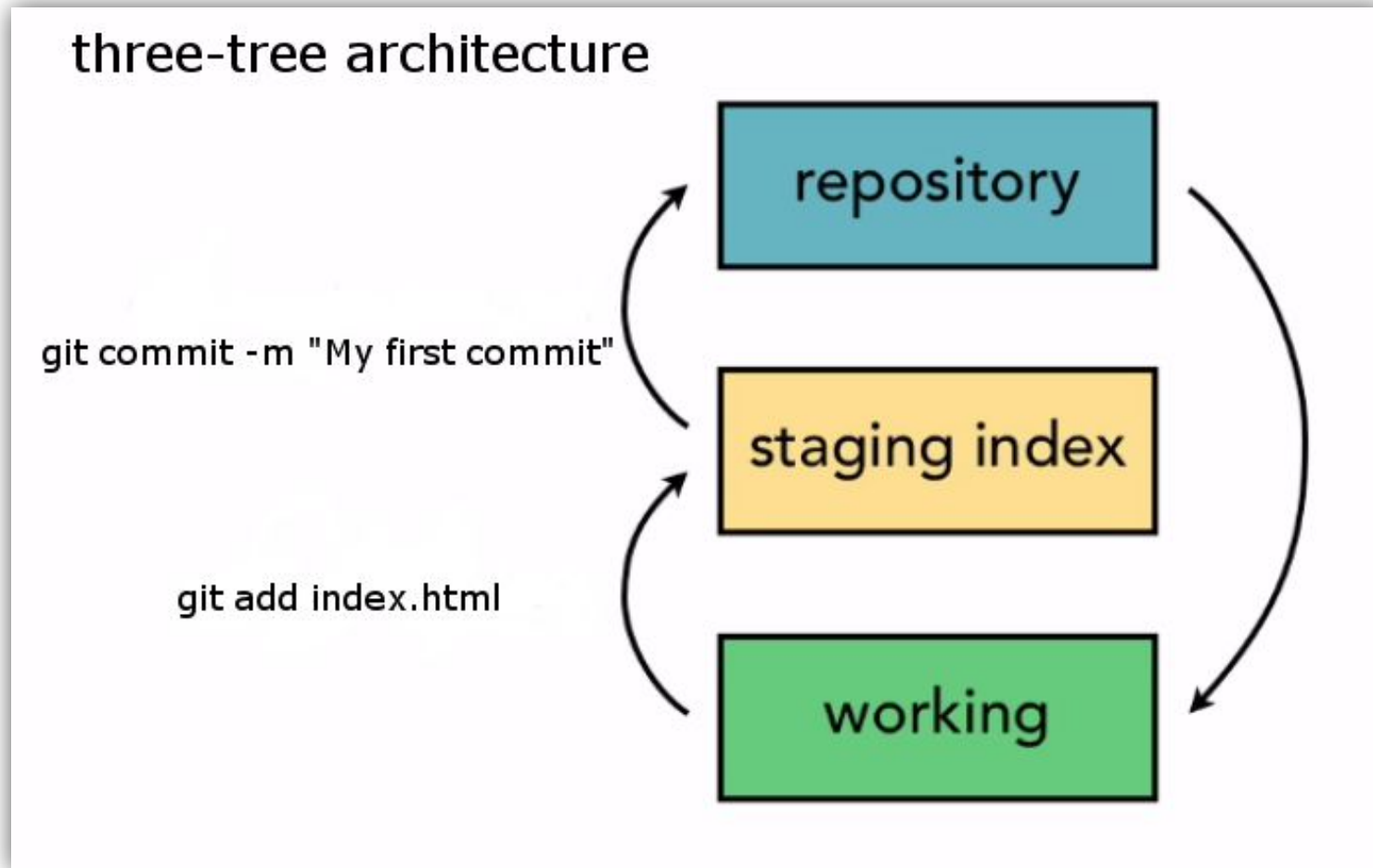


Remote repository – távoli adattár

- ha magad – 3 szakasz
- ha csapatmunka – 4 szakasz
- GitHub, GitLab

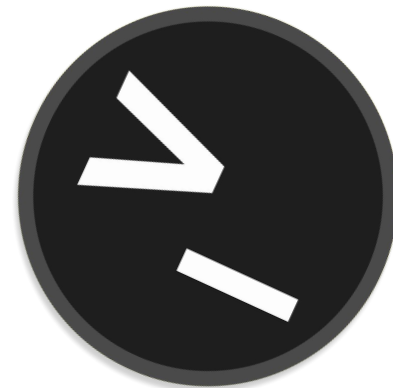


The Three Trees of GIT



Parancsok

- Terminal – Console – Command Prompt
- rengeteg kínálat, parancs
- Linux fejlesztés – UNIX parancsok
- Windows-on – GitBash program



Repo beállítása

- git init
 - mappa átalakítása GIT tárolóvá
- git clone
 - ha már van remote repo, akkor az egész „letöltése” a lokális mappába
- git config
 - különböző konfigurációk beállítása (felhasználónév, email, színek stb.)

```
git init
```

```
git clone
```

```
git config
```



Mentés

```
git add
```

```
git commit
```

```
git diff
```

- git add
 - munkafelületről -> átmeneti verzió
 - egyenként vagy több file-t
- git commit
 - átmeneti verzió -> állandó verzió
 - egyenként vagy több file-t
 - fontos a commit üzenet hozzácsatolása
- git diff
 - a verziók közötti különbségek kezelése, megjelenítése
 - + / -



HEAD

- legnépszerűbb GIT terminológia
- lényege a munkakörnyezet és a legutolsó commit kezelése pl. összehasonlítása
- különbség ellenőrzések:
 - workspace - staging
 - staging - local
 - local - remote
 - 2 commit



Követési parancsok

- git status
 - repo státusza
 - időről – időre
 - file, commit információk
- git log
 - commit előzmények megjelenítése
 - ki, mit, mikor

```
git status
```

```
git log
```



Visszaállítás

- git revert
 - HEAD vagy commit visszaállítás
 - új commit képződik az előző inverz verziójával
 - mintha nem is commiteltünk volna
- git reset
 - teljes visszaállítás parancsa
 - óvatosan kell alkalmazni
 - 3 típus: soft, mixed, hard

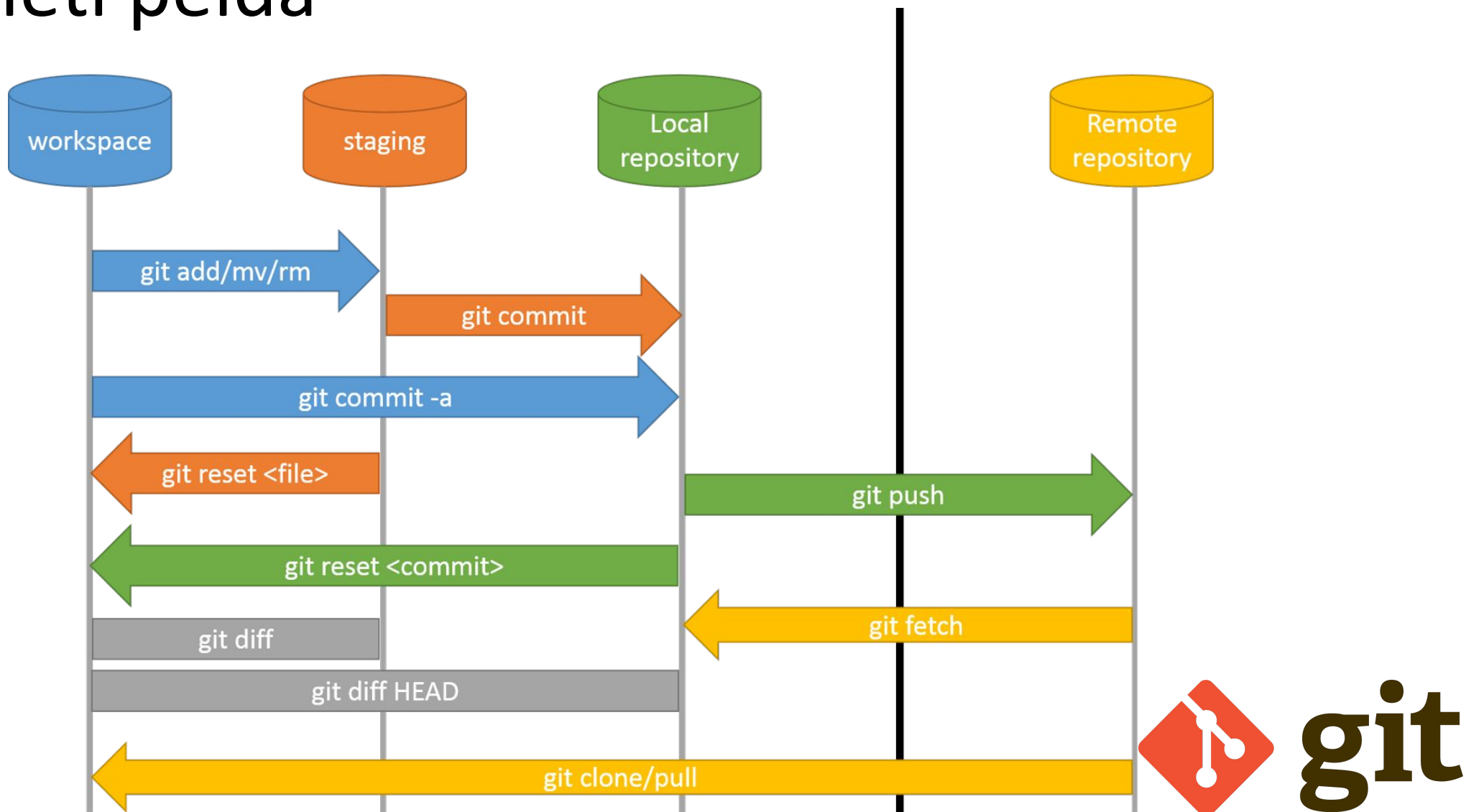
```
git revert
```

```
git reset
```



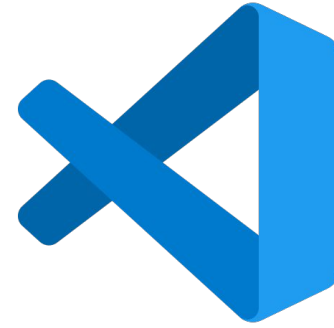
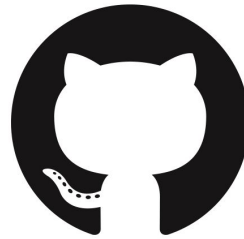
- soft
 - csak a lokális repo az érintett
- mixed
 - local és staging érintett
- hard
 - mind a 3 szakasz érintett
 - minden vissza lesz állítva
 - a nem kezelt változtatások elvesznek

Elméleti példa



Git gyakorlat

- Ami szükséges:
 - GitHub fiók
 - Kódszerkesztő
 - GitBash (Windows)
 - Git parancsok
 - Internet hozzáférés





Thank
you

Köszönöm a figyelmet!

Irodalom

- https://miro.medium.com/max/2900/1*gZX2Cs-To3k1h63hHhPPcw.png
- <https://www.icbtech.rs/wp-content/uploads/2019/07/512-300x300.png>
- <https://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/1200px-Tux.svg.png>
- <https://upload.wikimedia.org/wikipedia/commons/thumb/e/e0/Git-logo.svg/1280px-Git-logo.svg.png>
- <https://avatars2.githubusercontent.com/u/4571183>
- https://image.freepik.com/free-vector/business-workshop-person-man-concept-illustration-design_159757-72.jpg
- https://cdn.icon-icons.com/icons2/2107/PNG/512/file_type_vscode_icon_130084.png
- <https://surva2in1.files.wordpress.com/2016/06/git-architecture.png>
- https://github.githubassets.com/images/modules/logos_page/GitHub-Mark.png
- <https://about.gitlab.com/images/press/logo/png/gitlab-icon-rgb.png>
- <https://cdn1.vectorstock.com/i/1000x1000/59/00/opportunities-icon-design-template-isolated-vector-24295900.jpg>
- <https://www.psdstamps.com/wp-content/uploads/2019/11/round-most-popular-stamp-png-768x512.png>
- <https://macroports.com/wp-content/uploads/2019/05/terminal-app-icon.png>
- https://2.bp.blogspot.com/-K6OX_DJBHl0/Vms2vUM7PII/AAAAAAAAA4M/QCNAHS_RLeY/s1600/Terminal-icon.png
- https://img.freepik.com/free-vector/thank-you-lettering_1262-7412.jpg?size=626&ext=jpg
- <https://bitlogic.dev/static/images/website-development.png>
- <https://programmerblog.net/wp-content/uploads/2021/03/what-is-verion-control-systems-process.png>
- <https://medium.com/@mehulgala77/git-fundamentals-history-architecture-popular-commands-1d3cd8bc1049>