

Git Tahák Pro (Bash)

© 2025 Vaňha Jiří (®). Distribuce povolena pro studijní účely.

Základní práce s repozitářem

```
git init
git clone <url>
git status
git add <soubor>
git add .
git commit -m "popis"
git log -oneline
git diff
git show <hash>
```

Inicializuje nový lokální repozitář.
Naklonuje vzdálený repozitář (např. GitHub).
Zobrazí aktuální stav pracovního adresáře.
Přidá konkrétní soubor do stagingu.
Přidá všechny nové a změněné soubory.
Uloží změny do historie s komentářem.
Stručný výpis commitů.
Zobrazí rozdíly oproti poslednímu commitu.
Zobrazí detailní informace o commitu.

Práce s větvemi

```
git branch
git branch <název>
git checkout <název>
git checkout -b <název>
git branch -m <nový>
git merge <větev>
git merge -abort
git branch -d <větev>
git branch -D <větev>
git log master..bug -oneline
git cherry-pick <hash>
```

Vypíše všechny lokální větve.
Vytvoří novou větev.
Přepne se do dané větve.
Vytvoří novou větev a přepne se do ní.
Přejmenuje aktuální větev.
Sloučí danou větev do aktuální.
Zruší probíhající merge (např. při konfliktu).
Smaže větev (pokud je sloučená).
Násilně smaže větev (i když není sloučená).
Zobrazí commity pouze ve větvi bug.
Přidá konkrétní commit z jiné větve.

Vzdálený repozitář (GitHub, GitLab)

```
git remote -v
git remote add origin <url>
git push -u origin master
git pull
git fetch
git push origin -delete <větev>
git push origin -tags
```

Zobrazí připojené vzdálené repozitáře.
Přidá vzdálený repozitář.
Nahraje větev **master** na server.
Stáhne a sloučí změny z remote repozitáře.
Stáhne změny bez sloučení.
Smaže vzdálenou větev.
Nahraje všechny lokální tagy.

Práce se změnami a zálohování

```
git restore <soubor>
git restore -staged <soubor>
git reset -soft HEAD~1
git reset -hard HEAD~1
git stash push -m "záloha"
git stash list
git stash apply stash@{0}
git diff master..bug
```

Vrátí soubor do posledního commitu.
Odebere soubor ze stagingu.
Vrátí poslední commit, ale ponechá změny.
Vrátí stav repozitáře o jeden commit zpět (destruktivní).
Uloží rozpracované změny bokem.
Zobrazí uložené stash zálohy.
Obnoví konkrétní stash.
Porovná rozdíly mezi dvěma větvemi.

Reverze, obnova a historie

```
git revert <hash>
git reflog
git log -graph -oneline -all
git blame <soubor>
git shortlog -sn
git checkout <hash>
```

Vytvoří nový commit, který vrátí změny z daného commitu.
Zobrazí všechny pohyby HEADu (i smazané commity).
Zobrazí historii commitů s větvemi jako graf.
Zobrazí, kdo upravil který řádek souboru.
Statistika commitů podle autora.
Přepne se do konkrétního commitu (detached HEAD).

Tagy a verze

```
git tag
git tag <název>
git tag -a v1.0 -m "verze 1.0"
git show <tag>
git push origin <tag>
git push origin -tags
git tag -d <tag>
git describe -tags
```

Zobrazí všechny tagy.
Vytvoří lehký tag na aktuální commit.
Vytvoří anotovaný tag s popisem.
Zobrazí commit a info tagu.
Nahraje tag na server.
Nahraje všechny lokální tagy.
Smaže lokální tag.
Zobrazí nejbližší tag k aktuálnímu commitu.

Konfigurace a uživatelské nastavení

```
git config -list
git config user.name "Jméno"
git config user.email "email@example.com"
git config -global user.name "Jméno"
git config -global core.editor "code -wait"
git config -global color.ui auto
git config -list -show-origin
```

Zobrazí aktuální konfiguraci.
Nastaví jméno pro commity.
Nastaví e-mail.
Nastaví jméno globálně pro všechny repozitáře.
Nastaví výchozí editor (např. VS Code).
Zapne barevné výpisy.
Ukáže konfiguraci s cestami k souborům.

Další užitečné příkazy

```
git clean -n
git clean -f
git mv <starý> <nový>
git rm <soubor>
git grep <text>
git archive -o verze1.zip HEAD
git bisect start
```

Ukáže, které nesledované soubory by byly smazány.
Smaže nesledované (untracked) soubory.
Přejmenuje nebo přesune soubor.
Smaže soubor ze sledování i z disku.
Vyhledá text v historii commitů nebo souborech.
Vytvoří ZIP archiv aktuálního stavu.
Zahájí binární hledání chyby mezi commity.

Poznámka: Tento tahák obsahuje nejpoužívanější i rozšířené příkazy Gitu vhodné pro vývojáře, správce kódu a pokročilé uživatele. Kompiluj pomocí `pdflatex -shell-escape Git_tahak_pro.tex` nebo v Overleafu s povoleným **shell escape**.