

# Git Tahák Pro (Bash)

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

---

## Základní práce s repozitářem

<code>git init</code>	Inicializuje nový lokální repozitář.
<code>git clone &lt;url&gt;</code>	Naklonuje vzdálený repozitář (např. GitHub).
<code>git status</code>	Zobrazí aktuální stav pracovního adresáře.
<code>git add &lt;soubor&gt;</code>	Přidá konkrétní soubor do stagingu.
<code>git add .</code>	Přidá všechny nové a změněné soubory.
<code>git commit -m "popis"</code>	Uloží změny do historie s komentářem.
<code>git log -oneline</code>	Stručný výpis commitů.
<code>git diff</code>	Zobrazí rozdíly oproti poslednímu commitu.
<code>git show &lt;hash&gt;</code>	Zobrazí detailní informace o commitu.

## Práce s větvemi

<code>git branch</code>	Vypíše všechny lokální větve.
<code>git branch &lt;název&gt;</code>	Vytvoří novou větev.
<code>git checkout &lt;název&gt;</code>	Přepne se do dané větve.
<code>git checkout -b &lt;název&gt;</code>	Vytvoří novou větev a přepne se do ní.
<code>git branch -m &lt;nový&gt;</code>	Přejmenuje aktuální větev.
<code>git merge &lt;větev&gt;</code>	Sloučí danou větev do aktuální.
<code>git merge -abort</code>	Zruší probíhající merge (např. při konfliktu).
<code>git branch -d &lt;větev&gt;</code>	Smaže větev (pokud je sloučená).
<code>git branch -D &lt;větev&gt;</code>	Násilně smaže větev (i když není sloučená).
<code>git log master..bug -oneline</code>	Zobrazí commity pouze ve věti bug.
<code>git cherry-pick &lt;hash&gt;</code>	Přidá konkrétní commit z jiné větve.

## Vzdálený repozitář (GitHub, GitLab)

<code>git remote -v</code>	Zobrazí připojené vzdálené repozitáře.
<code>git remote add origin &lt;url&gt;</code>	Přidá vzdálený repozitář.
<code>git push -u origin master</code>	Nahraje větev master na server.
<code>git pull</code>	Stáhne a sloučí změny z remote repozitáře.
<code>git fetch</code>	Stáhne změny bez sloučení.
<code>git push origin -delete &lt;větev&gt;</code>	Smaže vzdálenou větev.
<code>git push origin -tags</code>	Nahraje všechny lokální tagy.

## Práce se změnami a zálohování

<code>git restore &lt;soubor&gt;</code>	Vrátí soubor do posledního commitu.
<code>git restore -staged &lt;soubor&gt;</code>	Odebere soubor ze stagingu.

git reset -soft HEAD~1	Vrátí poslední commit, ale ponechá změny.
git reset -hard HEAD~1	Vrátí stav repozitáře o jeden commit zpět ( destruktivní).
git stash push -m "záloha"	Uloží rozpracované změny bokem.
git stash list	Zobrazí uložené stash zálohy.
git stash apply stash@{0}	Obnoví konkrétní stash.
git diff master..bug	Porovná rozdíly mezi dvěma větvemi.

# Reverze, obnova a historie

git revert <hash>	Vytvoří nový commit, který vrátí změny z daného commitu.
git reflog	Zobrazí všechny pohyby HEADu (i smazané commity).
git log -graph -oneline -all	Zobrazí historii commitů s větvemi jako graf.
git blame <soubor>	Zobrazí, kdo upravil který řádek souboru.
git shortlog -sn	Statistika commitů podle autora.
git checkout <hash>	Přepne se do konkrétního commitu (detached HEAD).

# Tagy a verze

git tag	Zobrazí všechny tagy.
git tag <název>	Vytvoří lehký tag na aktuální commit.
git tag -a v1.0 -m "verze 1.0"	Vytvoří anotovaný tag s popisem.
git show <tag>	Zobrazí commit a info tagu.
git push origin <tag>	Nahraje tag na server.
git push origin -tags	Nahraje všechny lokální tagy.
git tag -d <tag>	Smaže lokální tag.
git describe -tags	Zobrazí nejbližší tag k aktuálnímu commitu.

# Konfigurace a uživatelské nastavení

git config -list	Zobrazí aktuální konfiguraci.
git config user.name "Jméno"	Nastaví jméno pro commity.
git config user.email "email@example.com"	Nastaví e-mail.
git config -global user.name "Jméno"	Nastaví jméno globálně pro všechny repozitáře.
git config -global core.editor "code -wa"	Nastaví výchozí editor (např. VS Code).
git config -global color.ui auto	Zapne barevné výpisy.
git config -list -show-origin	Ukáže konfiguraci s cestami k souborům.

# Další užitečné příkazy

git clean -n	Ukáže, které nesledované soubory by byly smazány.
git clean -f	Smaže nesledované (untracked) soubory.
git mv <starý> <nový>	Přejmenuje nebo přesune soubor.
git rm <soubor>	Smaže soubor ze sledování i z disku.
git grep <text>	Vyhledá text v historii commitů nebo souborech.
git archive -o verze1.zip HEAD	Vytvoří ZIP archiv aktuálního stavu.
git bisect start	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**.