

Git Tahák Pro (Bash)

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

Základní práce s repozitářem

<code>git init</code>	Inicializuje nový lokální repozitář.
<code>git clone <url></code>	Naklonuje vzdálený repozitář (např. GitHub).
<code>git status</code>	Zobrazí aktuální stav pracovního adresáře.
<code>git add <soubor></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 <hash></code>	Zobrazí detailní informace o commitu.

Práce s větvemi

<code>git branch</code>	Vypíše všechny lokální větve.
<code>git branch <název></code>	Vytvoří novou větev.
<code>git checkout <název></code>	Přepne se do dané větve.
<code>git checkout -b <název></code>	Vytvoří novou větev a přepne se do ní.
<code>git branch -m <nový></code>	Přejmenuje aktuální větev.
<code>git merge <větev></code>	Sloučí danou větev do aktuální.
<code>git merge -abort</code>	Zruší probíhající merge (např. při konfliktu).
<code>git branch -d <větev></code>	Smaže větev (pokud je sloučená).
<code>git branch -D <větev></code>	Násilně smaže větev (i když není sloučená).
<code>git log master..bug -oneline</code>	Zobrazí history pouze ve věti bug.
<code>git cherry-pick <hash></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 <url></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 <větev></code>	Smaže vzdálenou větev.
<code>git push origin -tags</code>	Nahraje všechny lokální tagy.

Práce se změnami a zálohování

<code>git restore <soubor></code>	Vrátí soubor do posledního commitu.
<code>git restore -staged <soubor></code>	Odebere soubor ze stagingu.
<code>git reset -soft HEAD~1</code>	Vrátí poslední commit, ale ponechá změny.
<code>git reset -hard HEAD~1</code>	Vrátí stav repozitáře o jeden commit zpět (destruktivní).
<code>git stash push -m "záloha"</code>	Uloží rozpracované změny bokem.
<code>git stash list</code>	Zobrazí uložené stash zálohy.
<code>git stash apply stash@{0}</code>	Obnoví konkrétní stash.
<code>git diff master..bug</code>	Porovná rozdíly mezi dvěma větvemi.

Reverze, obnova a historie

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

Tagy a verze

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

Konfigurace a uživatelské nastavení

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

Další užitečné příkazy

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