

Tahák základních příkazů Git (Bash)

Autor: Šéfe | 30. října 2025

Základní práce s repozitárem

<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.

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 merge <větev></code>	Sloučí danou větev do aktuální.
<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, které jsou jen ve věti bug.

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>	Nahráje větev <code>master</code> 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.

Práce se změnami

<code>git restore <soubor></code>	Vrátí soubor do posledního commitu.
<code>git restore -staged <soubor></code>	Odebere soubor ze stagingu.
<code>git reset -hard</code>	Vrátí vše do stavu posledního commitu (pozor!).
<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.

Patche a archivace

<code>git diff master..bug > bug.patch</code>	Vytvoří patch soubor se změnami.
<code>git apply bug.patch</code>	Aplikuje patch zpět do aktuální větve.
<code>git format-patch master..bug -o ./patches</code>	Vytvoří sadu patchů pro každý commit.
<code>git am ./patches/*.patch</code>	Aplikuje všechny patche i s historií.

Užitečné informace a diagnostika

<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 show <hash></code>	Zobrazí detailní informace o commitu.
<code>git reflog</code>	Ukáže i ztracené nebo přepsané commity.

Poznámka: Tento tahák obsahuje nejpoužívanější příkazy Gitu pro Bash. Pro pokročilou práci doporučuji prostudovat příkazy `git rebase`, `git cherry-pick`, `git bisect` a `git blame`.