

# Gitový tahák

## Instalace

**Fedora:** `sudo dnf install git gitk nano`  
**Ubuntu:** `sudo apt-get install git gitk nano`  
**Mac OS X:** Instalace se nabídne při prvním použití  
**Windows:** stáhnout z <http://git-scm.com/>  
"Run Git from the Windows Command Prompt"  
"Use OpenSSH"  
"Use Windows-style Line endings"

## Konfigurace

```
git config --global user.name "Jan Novotný"
git config --global user.email "jnov@example.com"
git config --global ui.color auto
git config --global core.editor nano
```

## Ve Windows navíc:

```
git config --global core.editor notepad
git config --global format.commitMessageColumns 80
```

## Založení repozitáře

```
git init          Založit repozitář v aktuálním adresáři
git clone <url>   Naklonovat repozitář z 'netu'
```

## Přidání souborů

```
git status        Kontrola stavu
git add <soubor>   Přidat soubor.py do Stage
git commit         Vytvořit novou revizi
```

## Prohlížení změn

```
git diff           Vypsát změny (viz diagram)
git diff <soubor>   Vypsát změny v konkrétním souboru
git log            Výpis všech revizí
git show           Zobrazení poslední revize
git show <revize>  Zobrazení konkrétní revize
gitk --all         Grafické klikátko
```

## Příprava revize

```
git add <soubor>   Přidat soubor do Stage
git reset <soubor> Reset Stage (zachová změny v prac. adr.)
git add -p          Vybrat jednotlivé změny k přidání
git rm <soubor>     Smazat soubor
git rm --cached <s> Smazat ze Stage, ale nechat v prac. adr.
git mv <z> <na>     Přejmenovat/přesunout soubor
```

## Větvení

<code>git branch</code>	Vypsát všechny větve
<code>git branch &lt;v&gt;</code>	Založit novou větev
<code>git branch &lt;v&gt; &lt;c&gt;</code>	Založit větev na dané revizi
<code>git checkout &lt;v&gt;</code>	Přepnout na větev
<code>git merge &lt;v&gt;</code>	Sloučit <v> s aktuální větví
<code>git branch -d vevet</code>	Smazat větev

## Ignorování

Podle záznamů v souboru `.gitignore`:

<code>build.log</code>	Daný soubor, i v podadresářích
<code>*.aux</code>	Všechno s danou příponou
<code>/a.out</code>	Abs. cesta z adresáře, kde je <code>.gitignore</code>
<code>__pycache__</code>	Musí být adresář
<code>!good.aux</code>	Negace (tenhle soubor neignorovat)

Osobní záznamy – `git config core.excludesfile`

## Synchronizace

```
git remote add <r> <url> Přidat repo na 'netu'
git remote -v             Vypsát přidávané repa
git fetch <r>             Stáhnout změny
git push <r> <b>          Poslat změny
git pull <r> <b>          fetch+merge najednou
```

## Jména revizí

<code>3abe48df9832</code>	Zkrácené jméno revize
<code>HEAD</code>	Aktuální revize
<code>master</code>	Větev - její poslední revize
<code>HEAD~</code>	"Rodič" aktuální revize
<code>HEAD~3</code>	"Pra-pra-prarodič" aktuální revize
<code>HEAD^</code>	"Rodič" aktuální revize
<code>HEAD^2</code>	"Druhý rodič" aktuální revize
<code>projekt/vetev</code>	Větev z externího repa (po git fetch)
<code>HEAD^{/regex}</code>	Posledí revize s "regex" v popisku

## Archeologie

```
git blame <soubor>   Autorství jednotlivých řádků
git log A...B         Porovnání dvou větví
git log --author <a>  Revize daného autora
git log --oneline     Jeden řádek na revizi
git log --graph       "Nakreslit" graf
git log --all         Vypsát všechny větve
git log --decorate    Popsat větve
git log -7            Posledních 7 revizí
git log -S <regex>   Revize přidávající/odebírající text
```

## Přepisování historie

```
git commit --amend   Přepsat poslední revizi
git rebase -i HEAD~3 Přeskládat poslední 3 revize
```

## Nestačí?

`git help`  
Literatura:  
Scott Chacon – Pro Git  
(ke stažení na [knihy.nic.cz](http://knihy.nic.cz))

