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

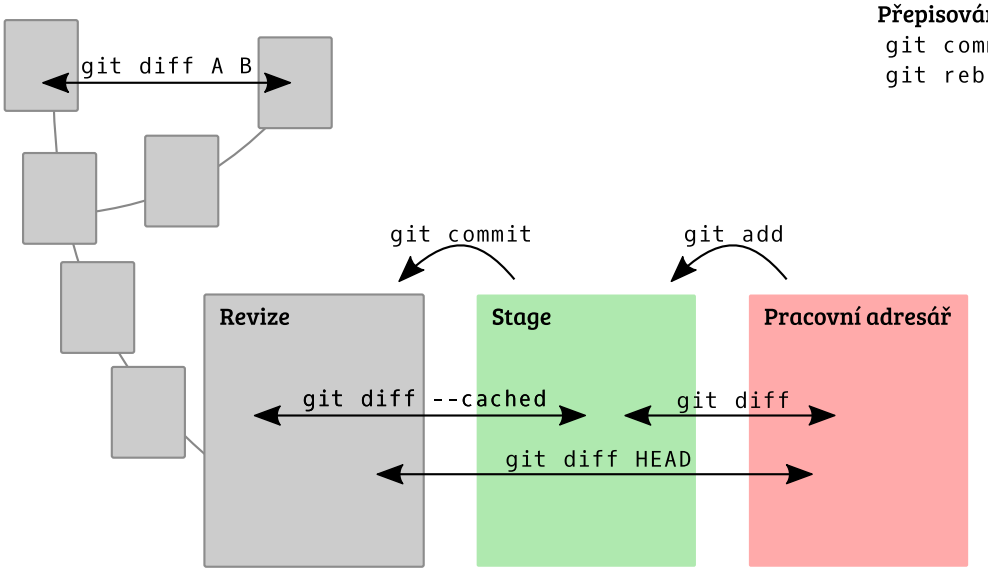
`git status`	Kontrola stavu
`git add <soubor>`	Přidat soubor.py do Stage
`git commit`	Vytvořit novou revizi

Prohlížení změn

`git diff`	Vypsat změny (viz diagram)
`git diff <soubor>`	Vypsat 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

`git branch`	Vypsat všechny větve
`git branch <v>`	Založit novou větev
`git branch <v> <c>`	Založit větev na dané revizi
`git checkout <v>`	Přepnout na větev
`git merge <v>`	Sloučit <v> s aktuální větví
`git branch -d vetev`	Smazat větev

Ignorování

Podle záznamů v souboru .gitignore:

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

Osobní záznamy – `git config core.excludesfile`

Synchronizace

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

Jména revizí

`3abe48df9832`	Zkrácené jméno revize
`HEAD`	Aktuální revize
`master`	Větev - její poslední revize
`HEAD~`	"Rodič" aktuální revize
`HEAD~3`	"Pra-prarodič" aktuální revize
`HEAD^`	"Rodič" akruální revize
`HEAD^2`	"Druhý rodič" akruální revize
`projekt/vetev`	Větev z externího repa (po git fetch)
`HEAD^{/regex}`	Poslední 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`	Vypsat 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)