### <https://git-scm.com/book/cs/>

<http://www.kutac.cz/blog/pocitace-a-internety/jak-na-git-dil-4/>

C:\Program Files\Notepad++

### Seznam základ příkazů

<https://git-scm.com/book/cs/v2/Z%C3%A1klady-pr%C3%A1ce-se-syst%C3%A9mem-Git-Zobrazen%C3%AD-historie-reviz%C3%AD#log_options>

### Přihlášení

https://github.com/new

RomanMacek

Rmacek123

Repository

RomanMacek/Test

Local path: C:\Users\RM\Documents\GitHub

### Stavy

#### Commited

Zapsáno = data jsou uložena v lokální DB.

#### Modified

Změněno = v souboru byly provedeny změny, ale tyto změny ještě nebyly uloženy do DB

#### Staged

K zapsání = změněný sobor v jeho aktuální verzi jsme určili k tomu, aby byl zapsán do dalšího snímku

### Instalace

<https://git-scm.com/book/cs/v2/%C3%9Avod-Instalace-syst%C3%A9mu-Git>

<https://desktop.github.com/>

### Příkazy

#### Editor config

Git config -- global core.editor "'C:/Program Files (x86)/Notepad++/notepad++.exe' -multiInst -nosession"

#### Nápověda

Git help <prikaz>

Git <prikaz> --helo

Man git-<prikaz>

Např: git help config

#### Inicializace repozitáře v existujícím adresáři

##### Zahájit sledování existujícího projektu

Přejít do adresáře a zadat :

Git init

Bližší info o adresářích a souborech které při tomto příkazu vzniknou je

https://git-scm.com/book/cs/v2/Git-Internals-Plumbing-and-Porcelain#\_git\_internals

#### Přidání sledování souborů

Git add \*.c

Nebo git add LICENCE

A potom udělat commit

Git commit –m 'initial project version'

#### Klonování existujícího repozitáře

Git clone <https://github.com/RomanMacek/Test.git>

Příkaz vytvoří i adresář

#### Sledování nových souborů

Git add nazev\_souboru

#### Ignorované soubory

$ cat .gitignore

\*.[oa] … ignorovat všechny soubory končící na o nebo a

\*~ … ignorovat všechny soubory, jejichž jméno končí vlnovkou

Pravidla pro masky, které můžete použít v souboru .gitignore, jsou následující:

* Prázdné řádky nebo řádky začínající znakem # budou ignorovány.
* Používají se standardní masky souborů (glob patterns).
* Masku můžete zahájit lomítkem (/) a tím potlačíte rekurzivitu.
* Masku můžete ukončit lomítkem (/) a tím vyjádříte, že jde o adresář.
* Masku můžete negovat tím, že na začátku použijete vykřičník (!).

Hvězdička (\*) označuje žádný nebo více znaků; [abc] označuje jakýkoli znak uvedený v závorkách (v tomto případě a, b nebo c); otazník (?) označuje jeden znak; znaky v hranatých závorkách oddělené pomlčkou ([0-9]) označují jakýkoli znak v daném rozmezí (v našem případě 0 až 9). Použitím dvou hvězdiček můžeme popsat zanořené adresáře; a/\*\*/z popisuje a/z, a/b/z, a/b/c/z a tak dále.

#### Zobrazení připravených/nepřipravených změn

Git diff

Git diff --staged nebo git diff -- chached

#### GIT status

Git status –s

#### Commit

Git commit

Git commit –m „zprava k commitu“

Git commit –a …………. Git do revize automaticky zahrne každý soubor, který je sledován. Odpadá potřeba zadávat příkat git add

#### Odstranění souboru

<https://git-scm.com/book/cs/v2/Z%C3%A1klady-pr%C3%A1ce-se-syst%C3%A9mem-Git-Nahr%C3%A1v%C3%A1n%C3%AD-zm%C4%9Bn-do-repozit%C3%A1%C5%99e>

#### Přejmenování souboru

<https://git-scm.com/book/cs/v2/Z%C3%A1klady-pr%C3%A1ce-se-syst%C3%A9mem-Git-Nahr%C3%A1v%C3%A1n%C3%AD-zm%C4%9Bn-do-repozit%C3%A1%C5%99e>

#### zobrazení historie revizí

<https://git-scm.com/book/cs/v2/Z%C3%A1klady-pr%C3%A1ce-se-syst%C3%A9mem-Git-Zobrazen%C3%AD-historie-reviz%C3%AD>

git log

git log - -stat

git log - -pretty …… změní výstup logu na jiný než výchozí formát

git log - -pretty = oneline …. Vypíše všechny revize na jednom řádku

git log - -pretty = short …. zkrácený

git log - -pretty =fulll …………. plný

git log - -pretty =fuller ……….. úplný

git log - -pretty =format …… umožní předepsat vlastní formát výstupu logu

git log - -pretty = xxxxx - -graph

git log - -since=2.weeks …. Zobrazí seznam všech revizí pořízených za poslední 2 týdny

Tento příkaz pracuje s velkým množstvím formátů. Můžete zadat konkrétní datum ("2008-01-15") nebo relativní datum, jako je "2 years 1 day 3 minutes ago" (před 2 roky, 1 dnem a 3 minutami).

Xxx –author

Xxx –grep … vezprávách k reizi můžu hledat konkrétní slova

xxx-S … zobrazí se jen ervize se změnami kódu, kdy byl zadaný řetězec přidán nebo ostraněn. Pokud například chcete zjistit poslední revizi, kdy byl přidán nebo odstraněn odkaz na určitou funkci, můžete zavolat:

$ git log -Sjmeno\_funkce