

# Git-Github

## Git a Github

### Git

#### Příkazy command prompt (cmd)

- cd .. : přesun do adresáře o level výše
- dir : vypíše všechny položky v aktuálním adresáři
- D: : změna disku na disk D
- cd <relativní cesta> :  
změna adresáře.
- cd <absolutní cesta> :  
změna adresáře.
- cd / : návrat do kořenového adresáře C:\>
- cls : clear screen
- mkdir : vytvoří nový adresář  
**Příklad.** mkdir novy : vytvoří nový podadresář s názvem "novy" v aktuálním adresáři.
- rmdir : vymaže adresář v aktuálním adresáři
- echo : vytvoří nový soubor v aktuálním adresáři.  
**Příklad.** echo our first file > test.txt : vytvoří v aktuálním adresáři nový soubor s názvem test.txt a obsahující text "our first file"
- type : zobrazí obsah souboru.  
**Příklad.** Type test.txt : zobrazí obsah tohoto souboru, tj, text "our first file"
- del : vymaže soubor v aktuálním adresáři.  
**Příklad.** del test.txt : vymaže se soubor test.txt v aktuálním adresáři

#### Příkazy pro použití gitu z příkazové řádky

- git init : Create an empty Git repository or reinitialize an existing one
- git-status : Show the working tree status,  
Displays paths that have differences between the index file and the current HEAD commit, paths that have differences between the working tree and the index file, and paths in the working tree that are not tracked by Git (and are not ignored by gitignore[5]). The first are what you would commit by running git commit; the second and third are what you could commit by running git add before running git commit.
- git add . : Add file contents to the index (všechno se přidá do staged area)
- git commit -m "added file" :  
uloží změny do repozitáře.
- git config --global --edit : zobrazí jméno uživatele a email (a možná umožňuje tyto osobní údaje přepsat přímo v editoru VSC)
- git config --global user.name : zobrazí jméno "majitele"  
git config --global user.name "Dušan Bednařík"  
git config --global user.email "dusanbednarik@gmail.com" :  
nastaví v počítači jméno a email uživatele. [Viz Lekce 103, 11:07](#)
- git config --global user.email : zobrazí email "majitele"
- git branch : List, create, or delete branches
- git branch -m main : příkazem přejmenujeme master branch na main branch
- git checkout -b feature : založí branch s názvem "feature" a přepne se do této větve feature
- git merge second-feature : vnoří commity z větve "second-feature" do hlavní větve "main".
- git log : Show commit logs
- Vymazání souborů v pracovním adresáři (a které už jsme před tím "commitli") můžeme provést těmito způsoby:
  1. Vymažeme přímo bez použití gitu. Je potom zapotřebí přidat tuto změnu do staged area pomocí příkazu git add . a pak to ještě commitnout.
  2. git rm file2.txt: Vymaže se soubor file2.txt v aktuální větvi. Je také zapotřebí, abychom se nacházel v adresáři, kde je uložen soubor file2.txt. To se například stane, když budeme mazat soubor v nějakém podadresáři. Pak se tyto změny projeví ve staging area a poté je možné tyto změny commitnout a definitivně vymazat dané soubory.
- git reset --hard HEAD~1 : vracíme se jeden commit zpět. Inovuje se jak index, tak path.
- git branch -D feature : vymaže se větev (branch) s názvem "feature". (Asi bychom měli před tím přepnout do jiné větve než kterou chceme smazat.)
- git checkout -- . : tímto způsobem smažeme resetujeme poslední změny, které jsme zatím nepřidali do staged area a přejdeme k poslednímu stavu, který odpovídá poslednímu commitu v aktuaálním branchi.
- git reset file1.txt : tímto příkazem vrátíme poslední verzi souboru file1.txt ze staged area do unstaged area. (Tedy pokud jsme již změny předali do staged area příkazem git add .) Potém můžeme přejít k posledním commitu pomocí příkazu: git checkout -- . nebo více specificky příkazem: git checkout -- file1.txt

### Github

- **git remote add origin https://github.com/dusanpython66/learning-web-dev.git** : vytvoří propojení mezi lokálním repozitářem v gitu a vzdáleném (remote) repozitářem na githubu s uvedenou url adresou. Dále bude jméno *origin* zatupovat uvedenou url adresu.
- To add a new remote, called for example github instead of origin (which obviously already exists in your system), do the following: **git remote add github https://github.com/dusanpython66/learning-web-dev.git** Remember though, everywhere in the tutorial you see "origin" you should replace it with "github". For example git push origin master should now be git push github master. However, if you want to see what that origin which already exists is, you can do a **git remote -v** . If you think this is there by some error, you can update it like so: **git remote set-url origin https://github.com/dusanpython66/learning-web-dev.git**
- **git push -u origin main** : zkopíruje obsah lokálního repozitáře do vzdáleného (remote) repozitáře main.
- **git clone <url repozitáře> .** : skopíruje obsah vzdáleného repozitáře do aktuálního lokálního adresáře. není ani potřeba inicializovat příkazem git init lokální git repozitář neboť naklonujeme celý vzdálený repozitář včetně celé struktury repozitáře.
- **git pull** : stáhne aktuální stav vzdáleného repozitáře do lokálního repozitáře.