



# Source control management

PVA2 Programování a vývoj aplikací

# Obsah

1. Obsah
2. Bez SCM
3. git
4. Co je git
5. Jak se pracuje s git
6. Základy práce s git
7. Repository
8. Repository
9. add - Přidání souborů
10. commit - Záznam změn
11. commit
12. push - Odeslání změn
13. pull - Stáhnutí změn
14. fetch - Stáhnutí změn
15. Git
16. Instalace git
17. GitHub
18. Registrace GitHub
19. Registrace GitHub
20. Chcete se o git dozvědět více?

# Bez SCM

- Změny se ukládají ručně
- Špatná orientace adresáře s projektem
- Není organizace
- Není historie změn
- Není možné se vrátit k předchozí verzi
- Není možné spolupracovat na projektu
- Není možné vytvářet větve

- 📁 projekt
- 📁 projekt - backup-20xy-02-03
- 📁 projekt - backup-20xy-06-13
- 📁 projekt - HOTFIX před změnou fceBY
- 📁 projekt - kopie
- 📁 projekt - kopie (3)
- 📁 projekt - kopie (3) - kopie
- 📁 projekt - nova-fceAB
- 📁 projekt - záloha
- 📁 projekt - změna-fceBY





FREIHEIT  
ANTI-FASCHISTISCH  
ERKÄMPFEN!

CORONAL  
RECHTE HA...

3.1.2021 /

if [www.HAU4.d](http://www.HAU4.d)

Vii.s.d.P. ... le Schiffm...

n:

ar 2021

er P' P

git

mo

o

W

W

photos,  
ge. Let's  
anymore.  
ns have to  
outcome of  
policy.

LEUC  
HEIT

witz/M  
aue

ε

de

# Co je git

git je distribuovaný systém řízení verzí

Řízení verzí je systém, který zaznamenává změny v souboru nebo sadě souborů a pomáhá nám v případě potřeby později vyvolat konkrétní verze. Příklady SCM: GIT, Subversion (SVN), CVS a mnoho dalších

- Řízení verzí nám umožňuje:
  - sledovat změny v kódu
  - spolupráci na projektech
  - ukládat historii změn
  - vrátit se k předchozím verzím
  - vytvářet větve

# Jak se pracuje s git

- GIT je původně aplikací pro příkazový řádek
- Pro zjednodušení ovládání existují GUI aplikace nebo lépe, přímá integrace do IDE

# Základy práce s git

- Repozitáře
- Přidání souborů
- Commit
- Vytvoření větve
- Merge
- Pull
- Push
- Fetch

# Repository

- Základem práce jsou repository (repozitář, repo)
- Repozitář je místo, kde je uložen váš projekt a jeho historie změn.
- Členíme je dle místa uložení na centrální vzdálený a lokální.
- Centrální – všichni spolupracovníci
- Lokální – vaše privátní kopie centrálního repozitáře. Nad lokálním repo provádíte změny.



# Repository

- Pokud plánuji používat Git na projektu sám a pouze na jednom stroji, je možné vytvořit si pouze lokální repository. V tu chvíli stačí Git zprovoznit pouze na daném stroji a vše potřebné je připraveno. **Pozor na zálohy!**
- Bude-li na projektu pracovat více lidí nebo jej chci mít dostupný na více strojích, je nutné vytvořit vzdálené repository. Na výběr jsou dvě možnosti:
  - vlastní server, na kterém si zprovozníme repository
  - služba nabízející hostování repository (Github)

# add - Přidání souborů

```
1 git add <soubor>
```

- Přidání souborů do repozitáře se provádí pomocí příkazu `add`.
- Příkaz `add` přidá soubor do tzv. stage. Stage je místo, kde se nachází soubory, které budou součástí následujícího commitu.
- Lze použít pro jednotlivé soubory nebo pro celé adresáře.
- Je nutné provést před `commitem`. Pokud soubor nebo adresář nebyl přidán do stage, nebude součástí commitu.

# commit - Záznam změn

```
1 git commit -m "Zpráva"
```

- Commit je záznam o změně v projektu. Každý commit obsahuje zprávu, která popisuje, co bylo změněno.
- Commit si představte jako souhrn jednoho bloku práce, která spolu souvisí. Například jedna nová funkce.
- Tzn. budeme-li měnit obsah čtyř souborů v rámci jedné změny, zahrneme všechny čtyři soubory do jednoho commitu.

# commit

- Budeme-li v budoucnu chtít úpravy zrušit, vrátíme celý commit a všechny úpravy commitu se vrátí zpět.
- Je zvykem provádět commit nad funkčním a otestovaným kódem.
- Součástí commitu je zpráva, která pomůže identifikovat obsah změn. Obecný popis, identifikace issue atd.
- Commit nenahrazuje uložení změny souboru.

# push - Odeslání změn

```
1 git push
```

- `commit` máme vytvořen, ale je pouze v lokálním repo. Je nutné jej nahrát na centrální repozitář.
- Důležitou akcí při práci s GITem je tzv. push. Pushem odešleme naše commity, tedy provedené změny, do vzdáleného GIT repozitáře, kde je od této chvíle vidí všichni a mohou s úpravami dále pracovat.
- Každý commit musí být v nějaké větvi. Standardně se hlavní větev jmenuje master.
- Je vhodné commit ihned nahrát `push` na vzdálený repo. Nahráním zpřístupníte svůj commit ostatním vývojářům. Ti si jej mohou ihned stáhnout a navázat na Váš kód.

# pull - Stáhnutí změn

```
1 git pull
```

- `pull` je opakem `push`. Stáhne změny z vzdáleného repozitáře a sloučí je s Vaším lokálním repozitářem.
- Pokud se změny vzdáleného repozitáře neshodují s Vaším lokálním repozitářem, dojde ke sloučení anebo může dojít ke konfliktu. O konfliktech později.
- `pull` je vhodné provádět pokaždé, před začátkem práce, abychom vždy pracovali na aktuální verzi projektu.



# fetch - Stáhnutí změn

```
1 git fetch
```

- `fetch` je podobné jako `pull`, ale neprovádí automatické sloučení změn.
- Stáhne změny z vzdáleného repozitáře, ale neaplikuje je na Vaši lokální větev.
- Tímto způsobem můžete zjistit, co se změnilo v repozitáři, ale nebudete mít změny ve Vašem kódu.
- `fetch` je vhodné provádět před `pull`, abyste měli přehled o změnách, které se chystáte stáhnout.

Git



FREIHEIT  
ANTIFASCHISTISCH  
ERKÄMPFEN!

CRONAL  
ECHTE HA

if [www.HAU4.d](http://www.HAU4.d)

Vii.s.d.P. ... le Schiffm

n:

ar 2021  
er P' P

mo

o

# Instalace git

- Stáhnout a nainstalovat GIT <https://git-scm.com>
- Při instalaci není nutné měnit žádné předvolby
- Po instalaci je nutné nastavit jméno a email

```
1 # Nastaví jméno, které má být připojeno k transakcím odevzdání.  
2 git config --global user.name "Jméno Příjmení"  
3  
4 # Nastaví e-mail, který má být připojen k transakcím odevzdání.  
5 git config --global user.email "skolniEmail@oa-opava.cz"  
6  
7 # Volitelně  
8 # Povolí užitečné obarvení výstupu příkazového řádku  
9 git config --global color.ui auto  
10  
11 # Kontrola nastavení  
12 git config -l
```



photos,  
ge. Let's  
anymore.  
ns have to  
outcome of  
by.

# GitHub

FREIHEIT  
ANTIFASCHISTISCH  
ERKÄMPFEN!

RONALD  
ECHTE HA

3.1.2023 / C

if [www.HAU4.d](http://www.HAU4.d)

Vii.s.d.P. ... le Schiff

n:

ar 2021

er P' P' P'

mo

o

# GitHub

# Registrace GitHub

## 1. Registrace pokud ještě nemáme účet

- Vytvořit registraci na GitHub.com - <https://www.GitHub.com>
- Uživatelské jméno ve tvaru `JmenoPrijmeni` nebo `JmenoPrijmeniPrezdivka`
- Primární emailová adresa – školní/pracovní (ověřuje se vůči OA Enterprise)
- Nutná 2FA autentizace

Po registraci:

- Nastavit sekundární email – soukromý – neztratíte přístup po opuštění organizace
- Nastavte profil – Billing informace, fotografie, bio,...

# Registrace GitHub

## 2. Získání výhod GitHub Education Pack

- <https://education.github.com/pack>
- Při registraci povolit ověření polohy
- Dokládá se fotka ISIC karty



# Chcete se o git dozvědět více?

- <https://skills.github.com/>
- <https://git-scm.com/book/cs/v2> - kniha o GITu
- <https://githubtraining.github.io/training-manual>
- <https://docs.github.com/en/get-started/quickstart/hello-world>
- <https://training.github.com/downloads/github-git-cheat-sheet/>
- <https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet>
- <https://edav.info/github.html>
- <https://docs.microsoft.com/cs-cz/training/modules/introduction-to-github-visual-studio-code/>

Děkuji za pozornost

Otázky?

Repository / Prezentace