Obejrzyj na YouTube

git – wprowadzenie

Kasia Siedlarek
Marcin Sędłak-Jakubowski

PyLight #11, 12.02.2019

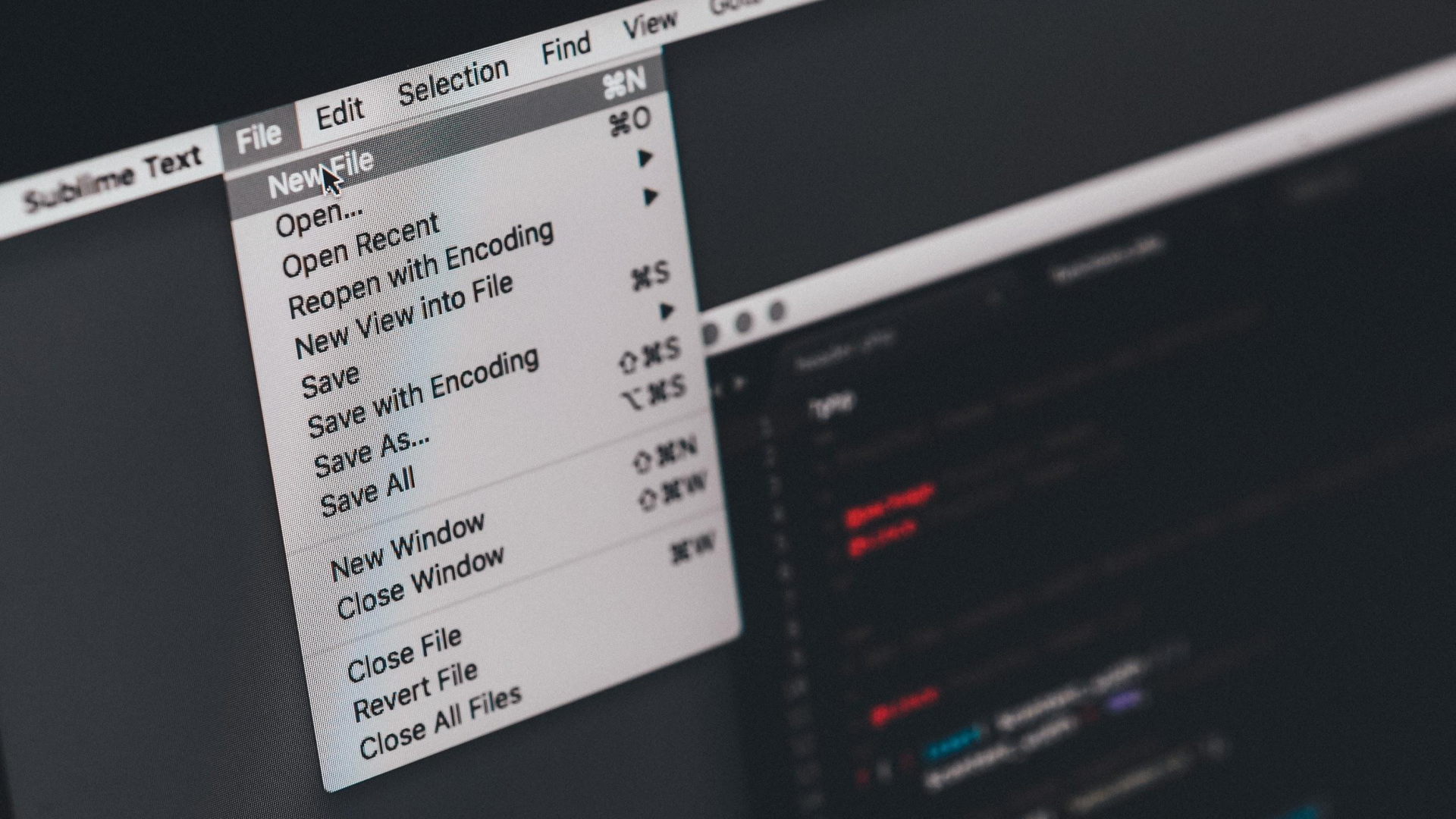
krótko o nas

Kasia Siedlarek

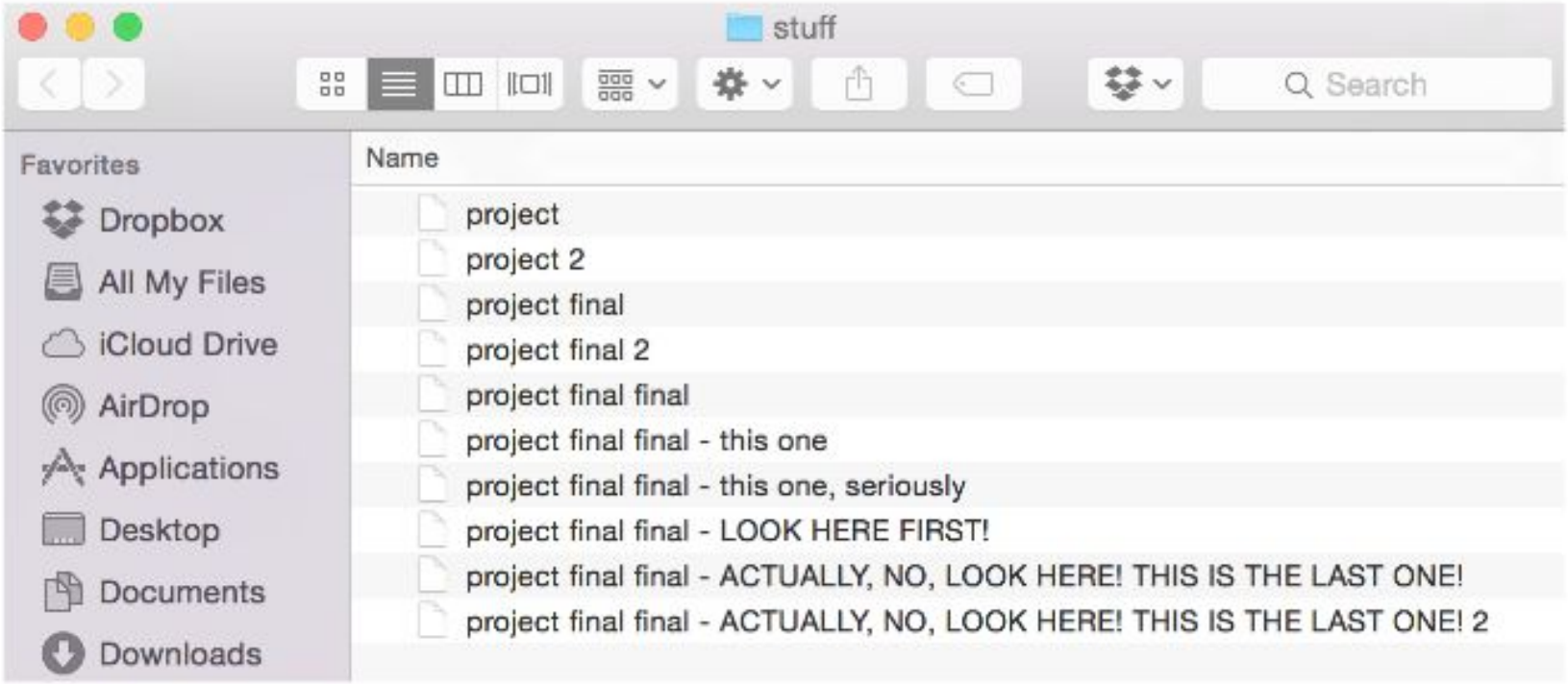
- teraz: virtualisation specialist / devops
- kiedyś i jednak nie: web dev

Marcin Sędłak-Jakubowski

- teraz: technical writer
- kiedyś: nauczyciel angielskiego
- jednak nie: socjolog, Python dev


Sublime Text
File
Edit
Selection
Find
View
New File ⌘N
Open... ⌘O
Open Recent
Reopen with Encoding
New View into File
Save ⌘S
Save with Encoding ⇧⌘S
Save As... ⌥⌘S
Save All ⇧⌘N
New Window ⇧⌘W
Close Window ⌘W
Close File
Revert File
Close All Files

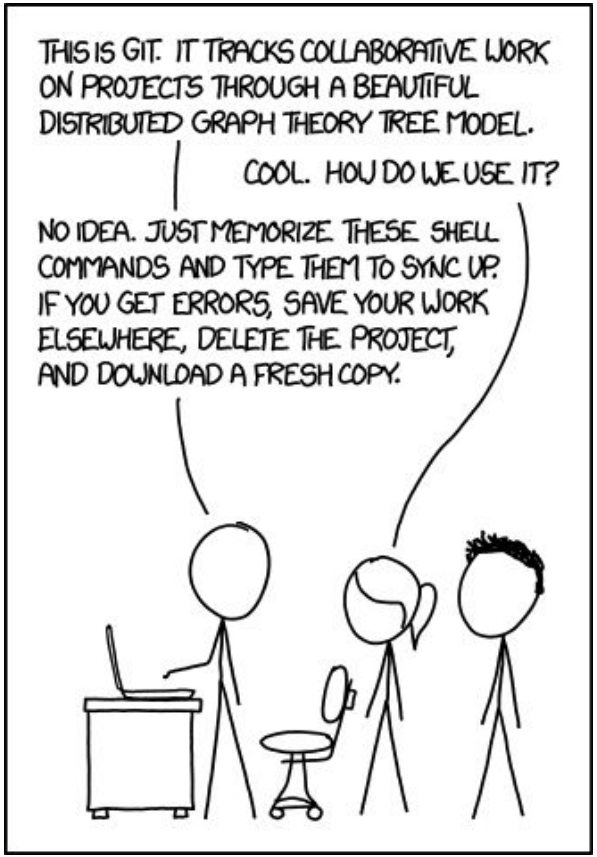
The Only Naming Convention That Works



DOGHOUSEDIARIES

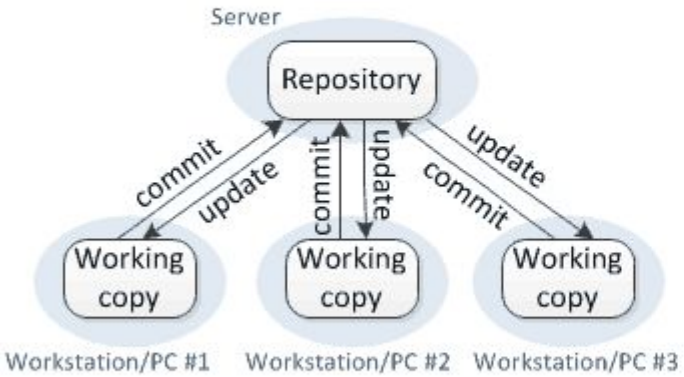
na ratunek: systemy kontroli wersji

- Istnieje wiele sposobów radzenia sobie z problemem zmian w naszych plikach
- Git - Linus Torvalds - 2005 - praca nad rozwojem linuksowego kernela
- Git - narzędzie opensource: https://github.com/git/git

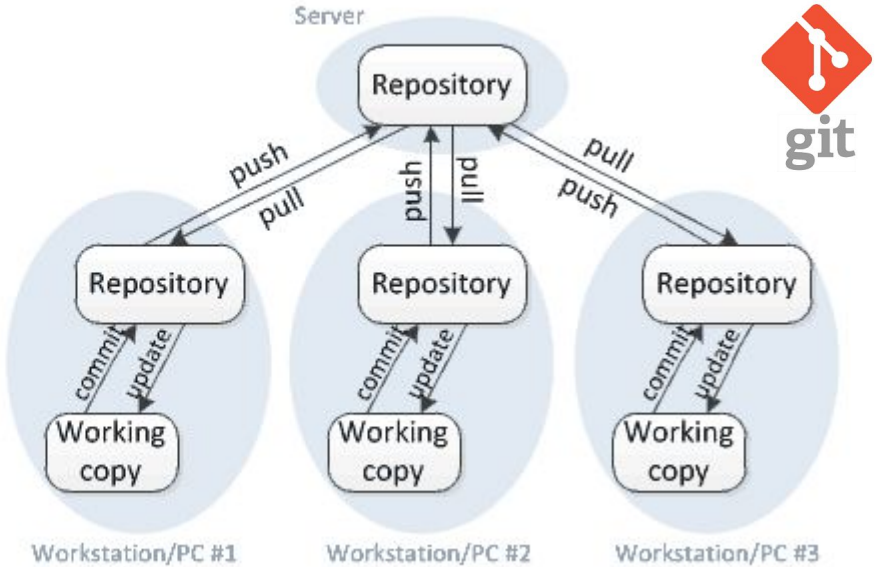


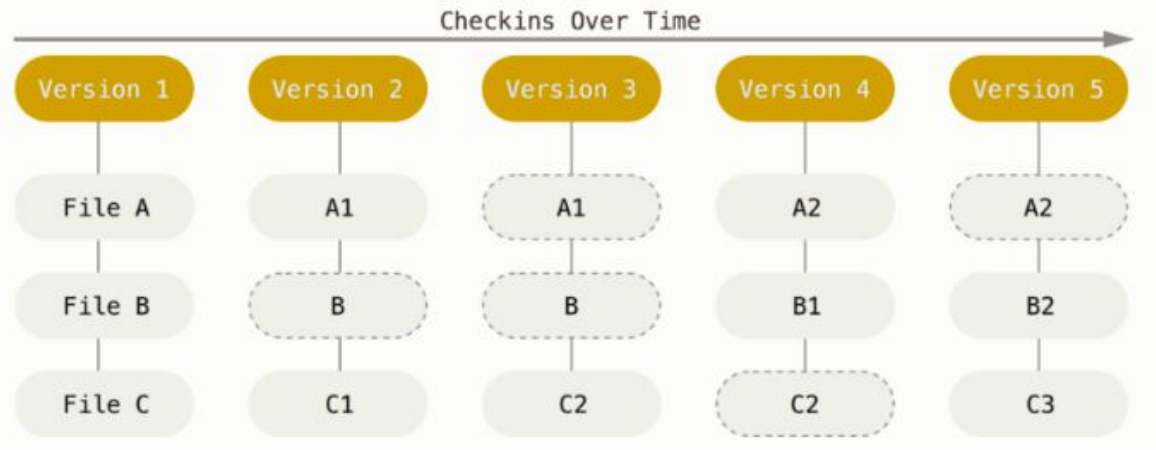
źródło: xkcd.com

Centralized version control

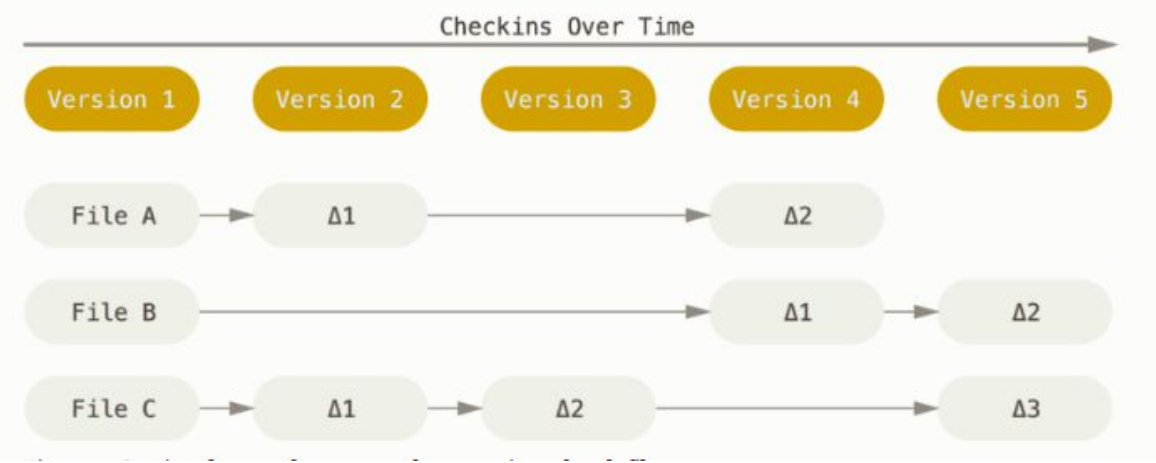


Distributed version control





snapshot całego systemu plików



tylko dane które zostały zmienione

źródło: https://git-scm.com/book/en/v2/

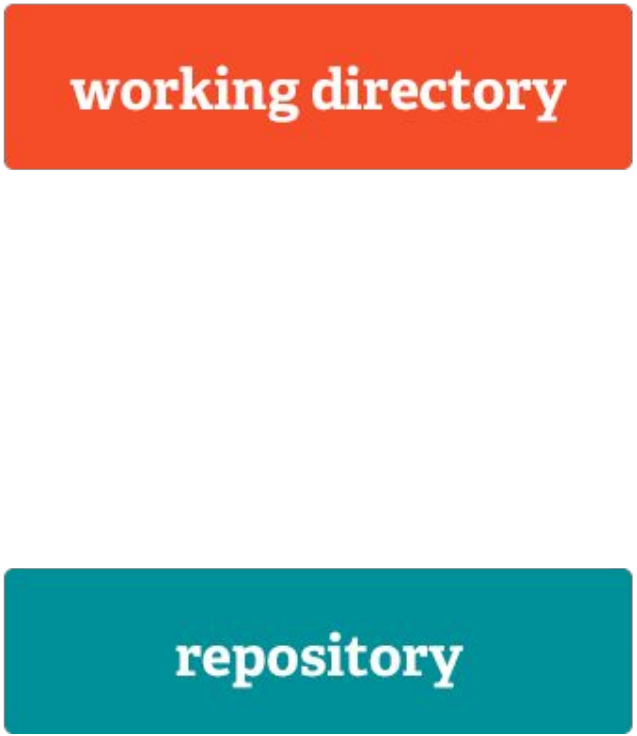
jak działa git?

źródło: Lucas Maurer

demo

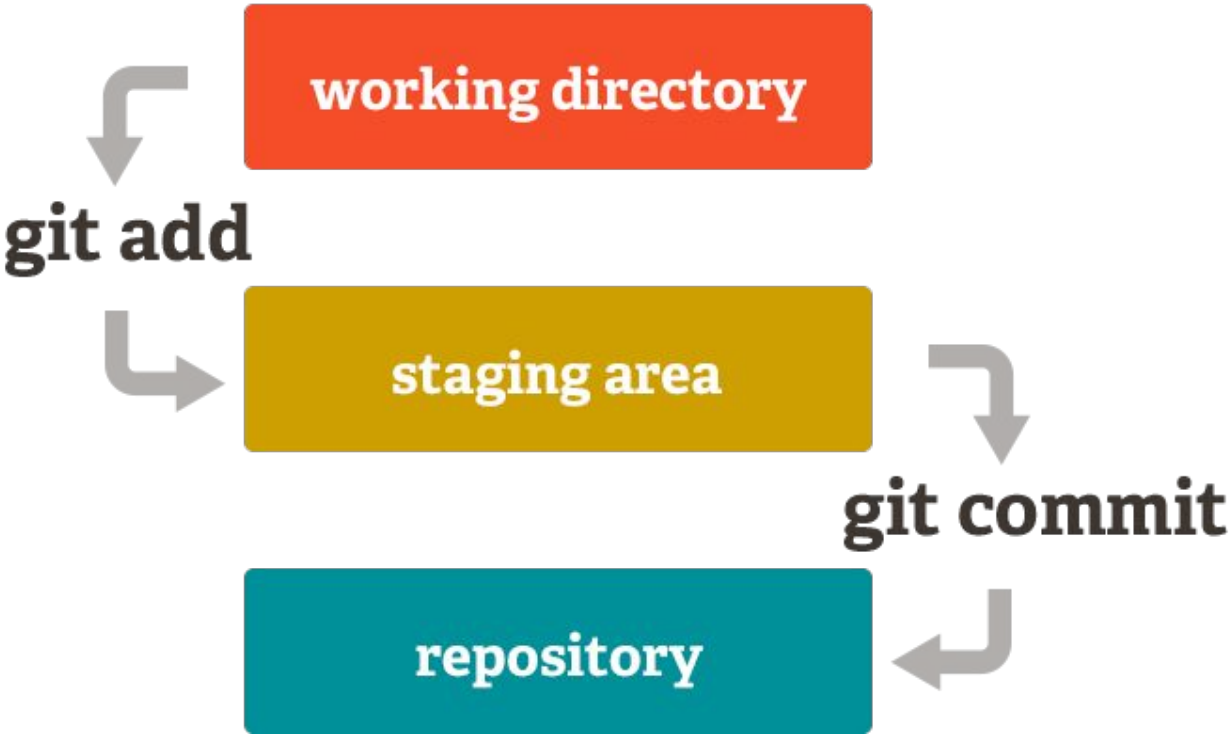
https://drive.google.com/open?id=10jK4scw9XZw0jOSWXA0TxdACck_59WOE

jak działa git?



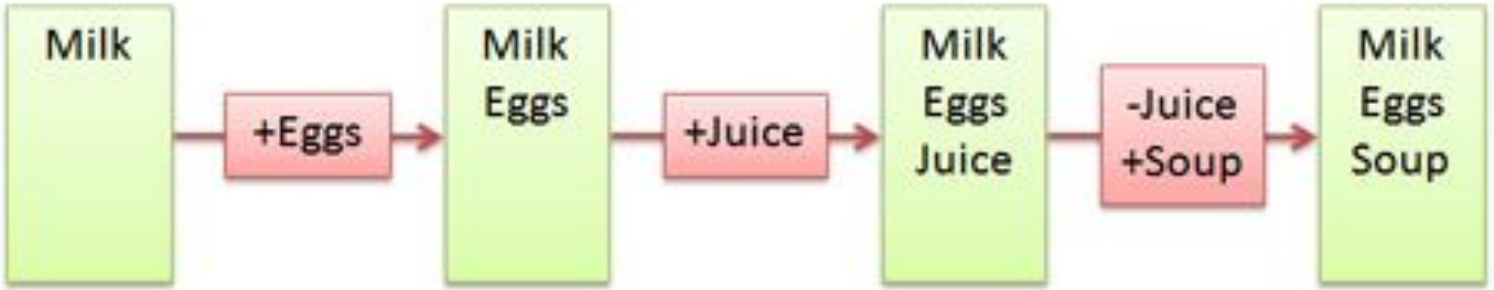
źródło: Lucas Maurer

jak działa git?



źródło: Lucas Maurer

co to jest commit?



5e28a90247
Create shopping list

135a324faa
Add eggs

153c0802a7
Add juice

3901e02e5e
Remove juice and add soup

SHA-1

źródło: Better Explained

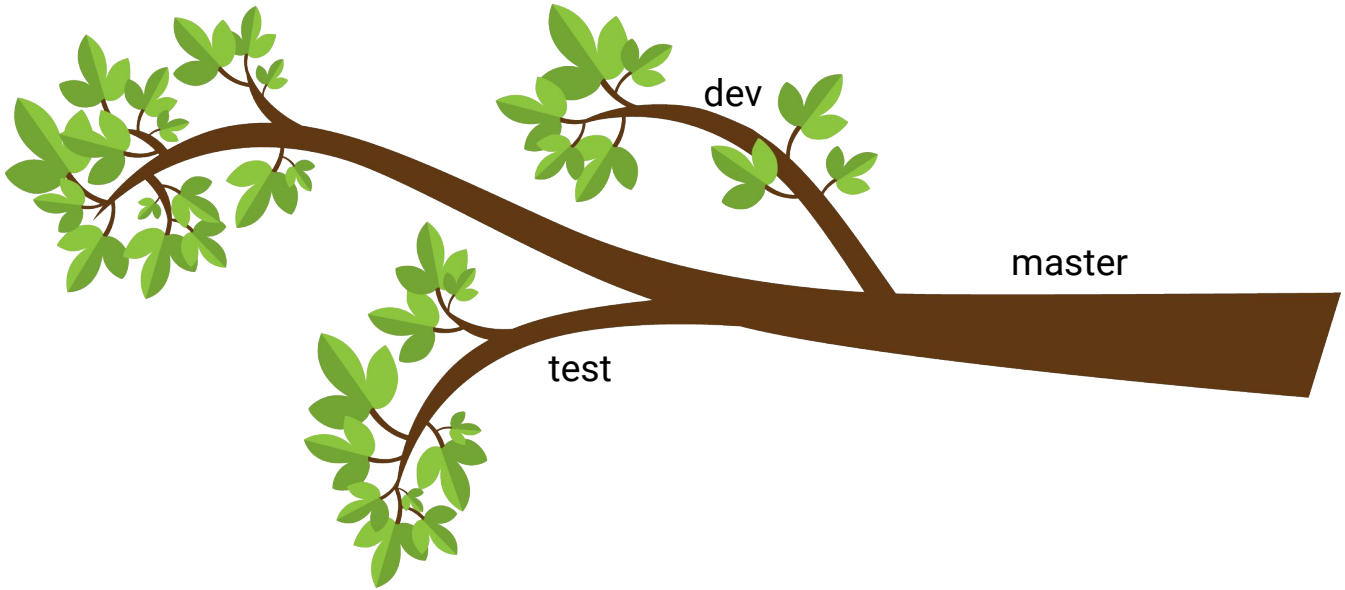

[~/t/.git (GIT_DIR!) ls
COMMIT_EDITMSG ORIG_HEAD config hooks info objects
HEAD branches description index logs refs
[~/t/.git (GIT_DIR!) cd objects/
[~/t/.g/objects (GIT_DIR!) ls
0d 29 2e 3a 3c 4e 55 6d 88 99 a5 a6 b0 b3 b7 b9 bd be c8 e1 info pack
[~/t/.g/objects (GIT_DIR!) ls 0d/
461e5e1bfd3d4d4370e111310fdffdf5b68868
~/t/.g/objects (GIT_DIR!)

HEAD i git checkout

```
[~/t/.git (GIT_DIR!) ls
COMMIT_EDITMSG ORIG_HEAD      config         hooks          info           objects
HEAD           branches       description    index          logs           refs
[~/t/.git (GIT_DIR!) cat HEAD
ref: refs/heads/master
[~/t/.git (GIT_DIR!) ls refs
heads tags
[~/t/.git (GIT_DIR!) ls refs/heads
master
[~/t/.git (GIT_DIR!) cat refs/heads/master
e1cef2f6d8b637ee0ba89bed77754ad0de69c2e6
~/t/.git (GIT_DIR!)
```

```
~/t/.git (GIT_DIR!) ls
COMMIT_EDITMSG ORIG_HEAD      config         hooks          info           objects
HEAD           branches       description    index          logs           refs
~/t/.git (GIT_DIR!) cat HEAD
ref: refs/heads/dev
~/t/.git (GIT_DIR!) cat refs/heads/dev
e1cef2f6d8b637ee0ba89bed77754ad0de69c2e6
~/t/.git (GIT_DIR!)
```

gałąź aka branch



chcę gita! skąd go wziąć?

git-scm.com/downloads

Poza tym:

- Debian/Ubuntu: `sudo apt-get install git`
- RedHat: `sudo yum install git`
- macOS (homebrew): `brew install git`
- Windows (chocolatey): `choco install git`

podstawowa konfiguracja

demo

https://drive.google.com/open?id=1Jp7ujWk-tssPrv1ZwydeGSvsYZRp9Cza

podstawowe operacje na plikach

git add

git rm

.gitignore

git <command> --dry-run

demo: https://asciinema.org/a/226263

więcej: https://www.gitignore.io/

Branch: master ▾ **django** / **.gitignore** Find file | Copy path

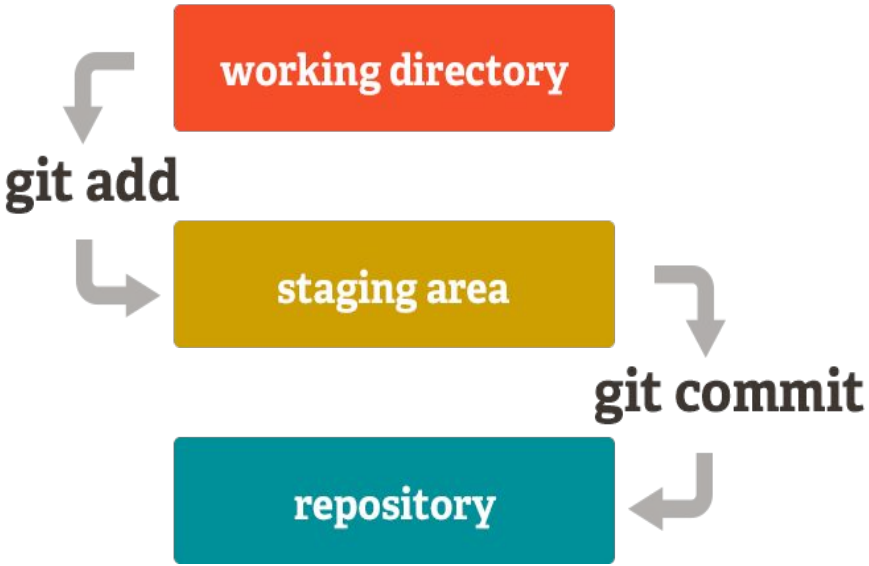
tobiasmcnulty Fixed #22446 -- Added tox.ini to automate pull request checks. 09d3874 on 20 Jul 2016

9 contributors 

19 lines (17 sloc) | 353 Bytes Raw | Blame | History   

```
1  # If you need to exclude files such as those generated by an IDE, use
2  # $GIT_DIR/info/exclude or the core.excludesFile configuration variable as
3  # described in https://git-scm.com/docs/gitignore
4
5  *.egg-info
6  *.pot
7  *.py[co]
8  .tox/
9  __pycache__
10 MANIFEST
11 dist/
12 docs/_build/
13 docs/locale/
14 node_modules/
15 tests/coverage_html/
16 tests/.coverage
17 build/
18 tests/report/
```

podstawowy workflow (lokalnie)



1. > on branch master
2. rób swoje zmiany
3. `git status`
4. `git add xxx.txt`
5. `git commit -m "Testowy commit"`

6. Sprawdź zaktualizowany stan:
   `git status`
   `git log`

źródło: Lucas Maurer

demo

https://drive.google.com/open?id=1MzCktGRCkraADNx0D9oDtBaj7cytrFpU

cofanie zmian

git revert <commit-id>

git reset (<commit-id>) <file>/ git reset --soft / git reset --hard/ git reset --mixed

demo: https://asciinema.org/a/226342

więcej:
https://docs.gitlab.com/ee/topics/git/numerous_undo_possibilities_in_git/

git reset - demystified

za dużo... pomocy!

git help / git help <command-name>

git help tutorial

git log / git log -p / git log --oneline / git log --graph --oneline --decorate

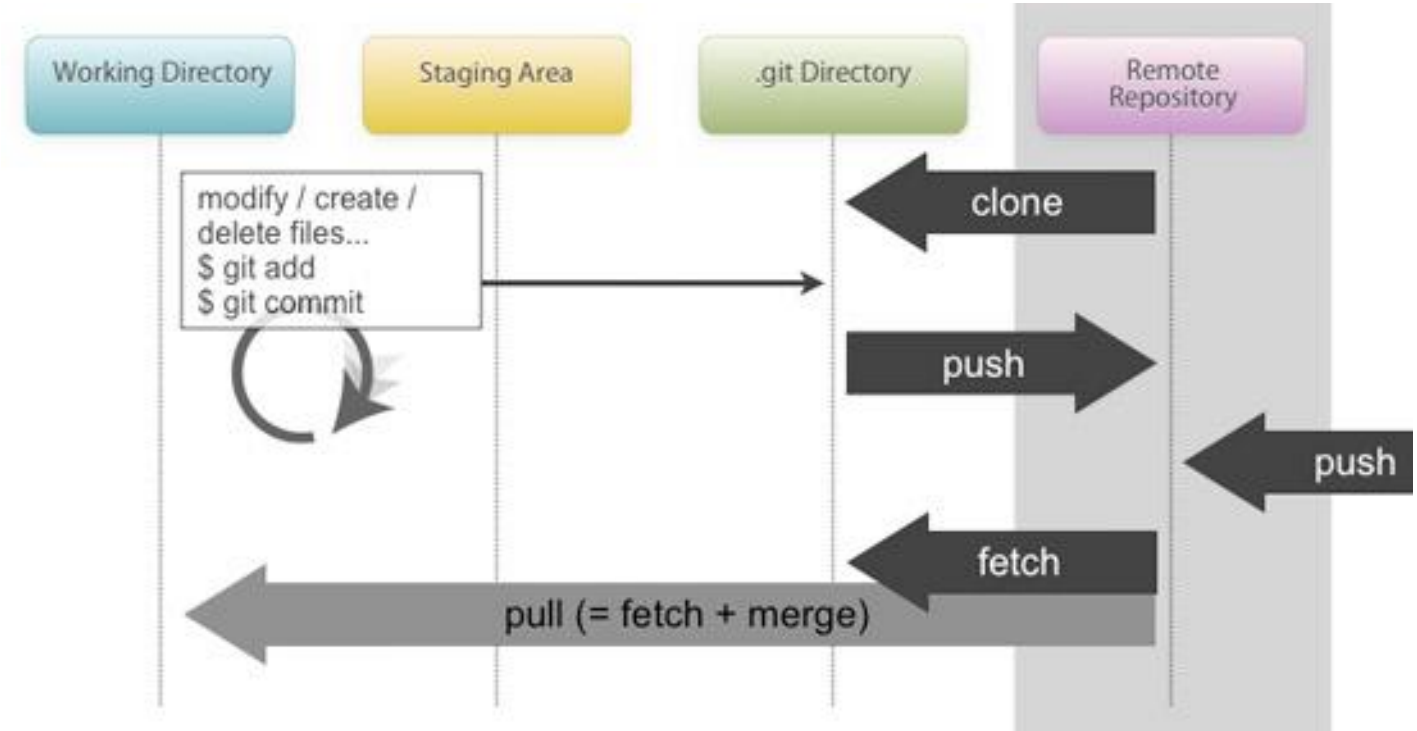
demo: https://asciinema.org/a/226347

więcej:

https://git-scm.com/book/en/v2/Git-Basics-Viewing-the-Commit-History

https://git-scm.com/docs/git-help

mój projekt dostępny w internecie?



demo

https://drive.google.com/open?id=1ZmsKr4bQ7uBgCp8kuC3elcPrOt_MFwjL

konsola albo GUI

demo

https://drive.google.com/open?id=1sOx1uG2fE9haSj2jt5z6bPrbd8khLkSA

co dalej?

- opanować podstawy (!)
- git branch
- git merge
- sposoby pracy z gitem w projektach
- repo w repo (np submodules)
- ...

linki

Pożyteczne:

Git Cheat Sheet | GitHub

Try GitHub

Git Book

Git tutorials | Atlassian

Understanding Git (part 1) — Explain it Like I'm Five | Hackernoon

Śmieszne (być może):

GIT - the stupid content tracker (Linus Torvalds - historyczny opis czym jest git)

git-poem

dziękujemy!