Git Workshop

by Laura Żuchowska 👋

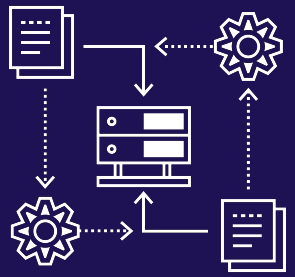
dyvenia

Agenda

Hello 👋

1. What's Git even? 🤔
2. Basics 🏫
3. Pull Request how-to 🔲
4. Practice! 😱

dyvenia



What's Git even?

dyvenia

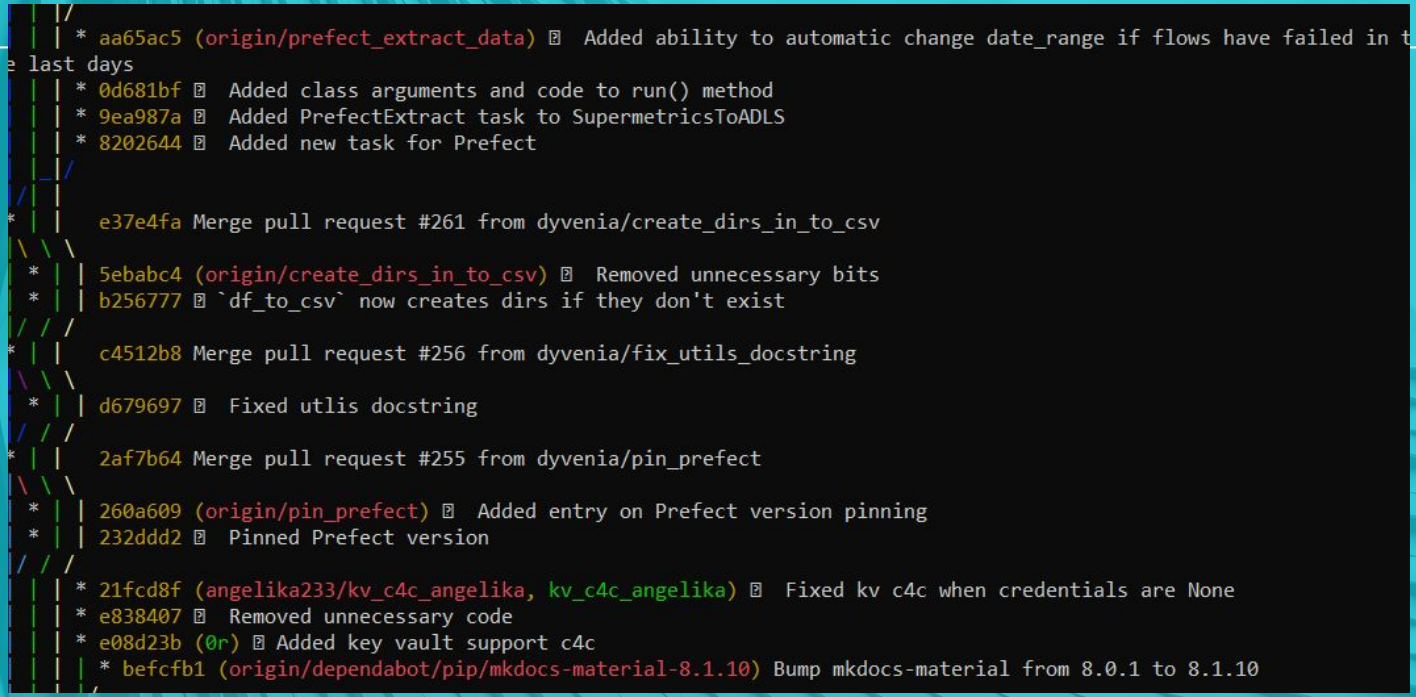
How does git work?

Collaborations on trees

- has a tree-like structure
- every developer has a local copy on their machine
- you can change your local history
- can’t handle big files

Branches

git log --all --decorate --oneline --graph 🤷



dyvenia

Common misconceptions

👀 Fear not 👀

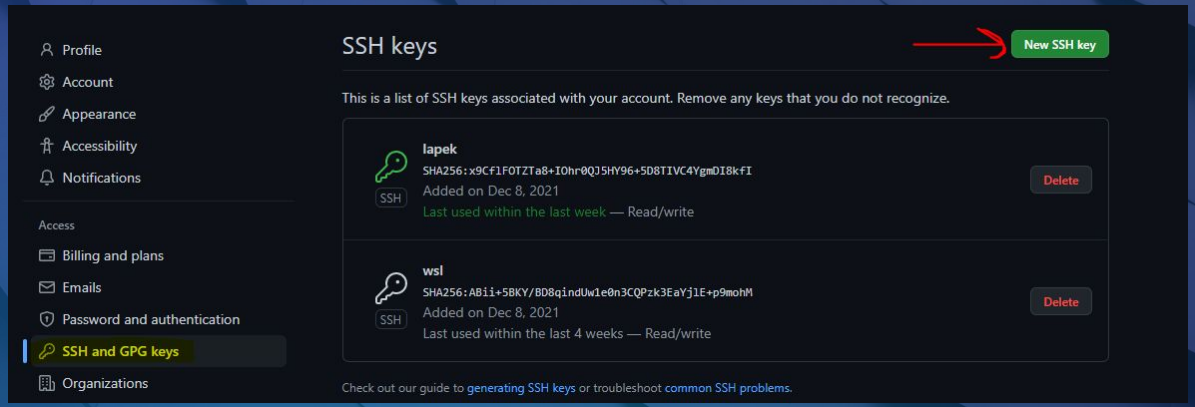
1. You will break the git repository with one push
2. You only commit when the feature is production-ready
3. Merge/rebasing is scary

SSH

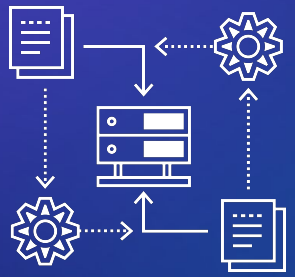
dyvenia

ssh-keygen -t rsa -b 4096 -C "tag-of-your-ssh"

1. Create an SSH key
2. Save it to a file
3. Write passphrase
4. ssh-add -K /path/to/your/file
5. Add SSH on git



dyvenia



Basics

dyvenia

Basic commands

add

git add [file/directory]

Stage files for commit.

Good practice:
Don't use *git add .*

commit

git commit -m [message]

Commit changes with message.

push

git push [from] [to]

Push changes to a branch.

pull

git pull [to] [from]

Pull changes from a branch.

dyvenia

Basic commands

rebase

git rebase [branch]

Update branch based on changes made.

Use if you're the contributor on official repo.

merge

git merge [branch]

Update branch based on changes made.

Use if you're working on a forked repo.

checkout

git checkout [branch]

Change branch.

Add -b to create a new branch.

Add -d to delete a branch.

status

git status

Show current list of files with changes.

dyvenia

Useful commands

stash

git stash show - show latest stash
git stash list - show list of stashes
git stash - save changes into stash (removes from workspace!)
git stash apply/pop - apply stashed changes (pop removes from stash list)

fetch

git fetch [to][branch] - saves a remote work in your local repo (without making actual changes, so you have to git checkout)

restore

git restore [file] - reverse the changes made on a file

Forking

On github repository page, click ‘fork’. Clone the forked repository with:

git clone <repository>

Add the original repository to your remote list

git remote add <name> <url>

Always pull changes from dev before working!

Create a new branch
git checkout -b <name>

~do stuff~

Add and commit
git add <things>
git commit -m “message”

Check for changes (and pull if there are)
git fetch
git pull <remote> <branch>

Push changes and merge
git push origin <name>
git checkout <where do you want to merge>
git merge <branch to merge>

dyvenia

Internal user

Clone the forked repository with:

git clone <repository>

No need to add to remote list

Always pull changes from dev before working!

Create a new branch
git checkout -b <name>

~do stuff~

Add and commit
git add <things>
git commit -m “message”

Check for changes (and pull if there are)
git pull <remote> <branch>

Push changes and merge
git checkout <branch>
git rebase <where to>
git push <origin> <branch>

dyvenia

Merge conflicts

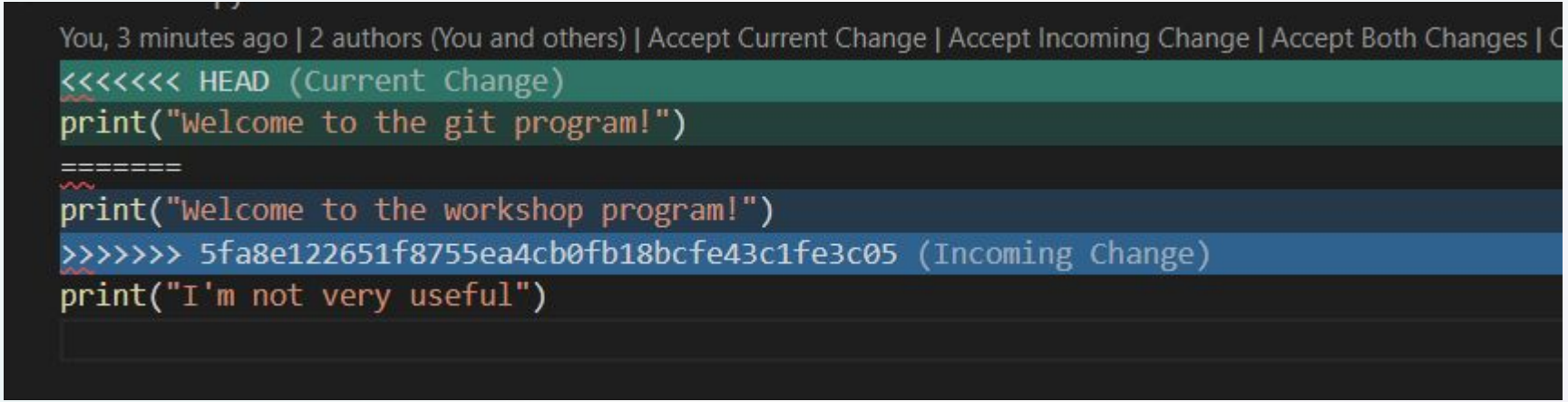
Chill, you're not breaking anything

```
C:\Users\Laura\Documents\Work\git_workshop\forkd\git-workshop>git
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 5 (delta 2), reused 4 (delta 2), pack-reused 0
Unpacking objects: 100% (5/5), 1001 bytes | 66.00 KiB/s, done.
From https://github.com/dyvenia/git-workshop
 * branch            main       -> FETCH_HEAD
   548f0af..5fa8e12  main       -> dyvenia/main
Auto-merging code/welcome.py
CONFLICT (content): Merge conflict in code/welcome.py
Automatic merge failed; fix conflicts and then commit the result.
```

dyvenia

Merge conflicts

Open your VSCode



dyvenia

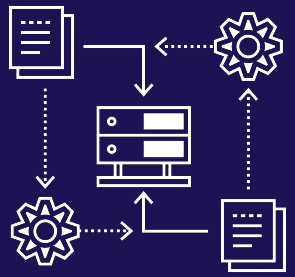
Merge conflicts

Where am i?

Add, commit, push to your remote, merge again.

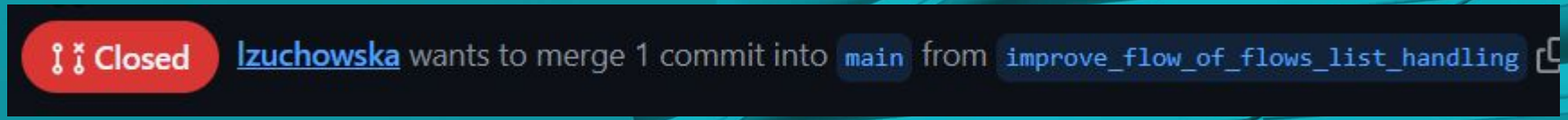
```
C:\Users\Laura\Documents\Work\git_workshop\forkd\git-workshop>git status
HEAD detached from 548f0af
All conflicts fixed but you are still merging.
  (use "git commit" to conclude merge)
```

dyvenia



Pull Request how-to

Never push to main



Don’t be like this gal

Keep it simple

dyvenia

One PR should implement one functionality

Don’t try to push as much as possible - if you notice a bug, have an idea for a new component get a sudden urge to write a test for something not directly connected to what you’re working on - leave it for next PR. Open an issue about it.

Keep the amount of commits per PR as low as possible. Reading through 100 commits is not the most pleasurable thing to do. But so is reading one commit with 20.000 lines of code. It’s all about the balance in the universe.

dyvenia

Naming conventions

:emoji: Verb + functionality

Always begin with a gitmoji (VSCode extension, each emoji specifies a function).

Use verbs such as “Added” or “Fixed”, capitalized.

Specify the functionality, try to keep it short.

If you get lost - naming conventions are usually written down in the documentation.


COMMENT
DATE
CREATED MAIN LOOP & TIMING CONTROL
14 HOURS AGO
ENABLED CONFIG FILE PARSING
9 HOURS AGO
MISC BUGFIXES
5 HOURS AGO
CODE ADDITIONS/EDITS
4 HOURS AGO
MORE CODE
4 HOURS AGO
HERE HAVE CODE
4 HOURS AGO
AAAAAAAA
3 HOURS AGO
ADKFJSLKDFJSDKLFJ
3 HOURS AGO
MY HANDS ARE TYPING WORDS
2 HOURS AGO
HAAAAAAAAANDS
2 HOURS AGO
AS A PROJECT DRAGS ON, MY GIT COMMIT MESSAGES GET LESS AND LESS INFORMATIVE.

dyvenia

Examples

Bad

- Fixed a bug
- 💩 Added a test for *my-thing*
- anna_commit
- ✨ Added the possibility to parse a letter from a non-latin alphabet - a mandarin Chinese symbol for ‘data’, I’ve discovered those by reading a book once mentioned from my father whom...

Better

- 🐛 Fixed a bug in *name* for empty df
- ✅ Added a test for my thing
- 🎨 Updated code structure for *name*
- ✨ Added the possibility to parse a letter from a non-latin alphabet

dyvenia

PRACTISE TIME

Thanks!

Contact us:

ul. Czyżówka 14/0.3,
30-526 Kraków
Poland

hello@dyvenia.com
www.dyvenia.com