

Version Control with Git & GitHub

Toby Hodges

Software Carpentry // CERN // 27 & 28 November 2019

"FINAL".doc



FINAL.doc!



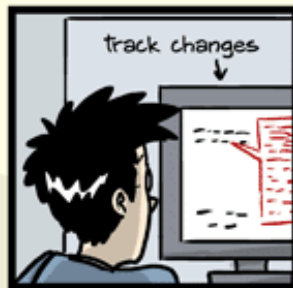
FINAL_rev.2.doc



FINAL_rev.6.COMMENTS.doc



FINAL_rev.8.comments5.
CORRECTIONS.doc



FINAL_rev.18.comments7.
corrections9.MORE.30.doc

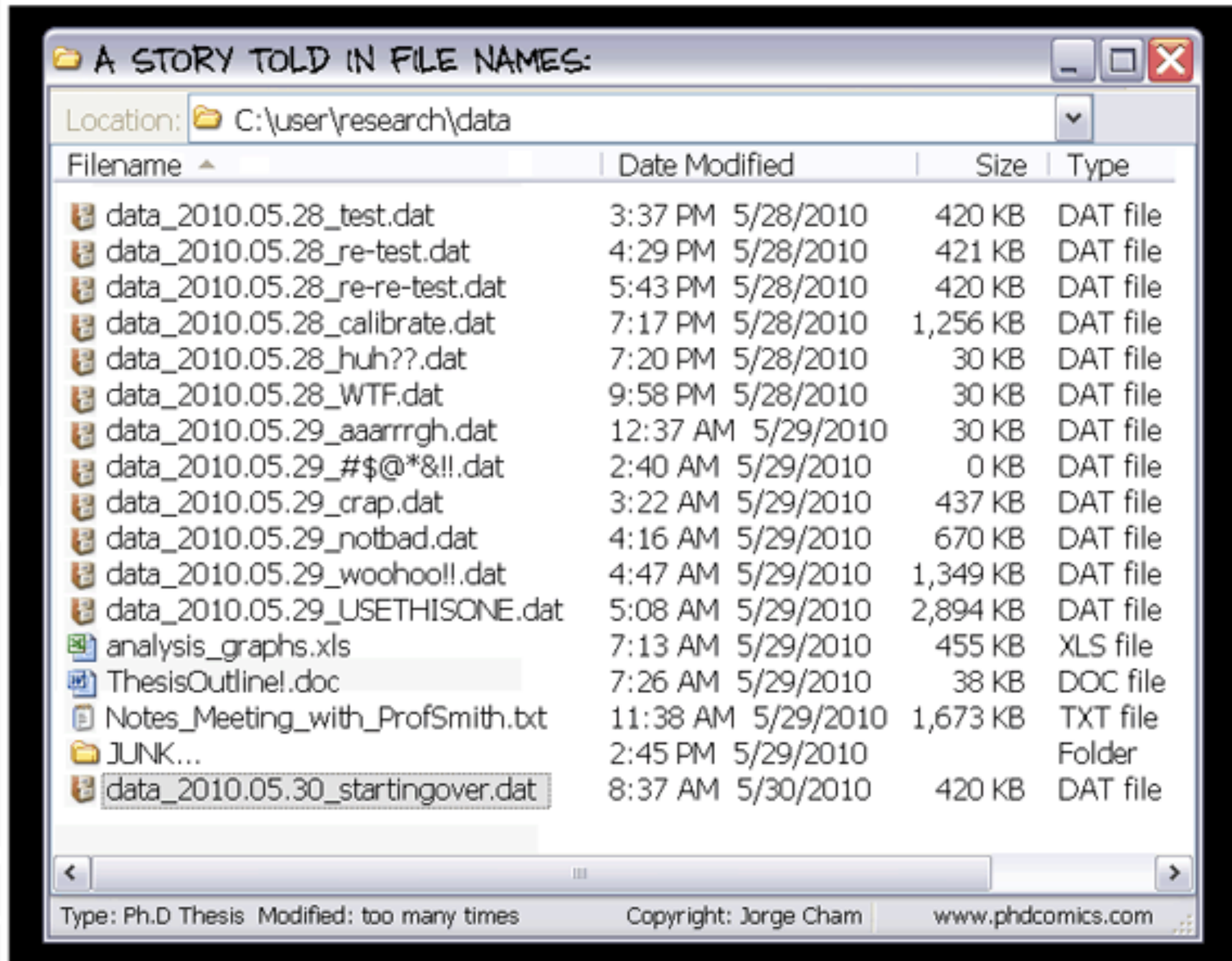


FINAL_rev.22.comments49.
corrections.10.#@\$%WHYDID
ICOMETOGRADSCHOOL?????.doc

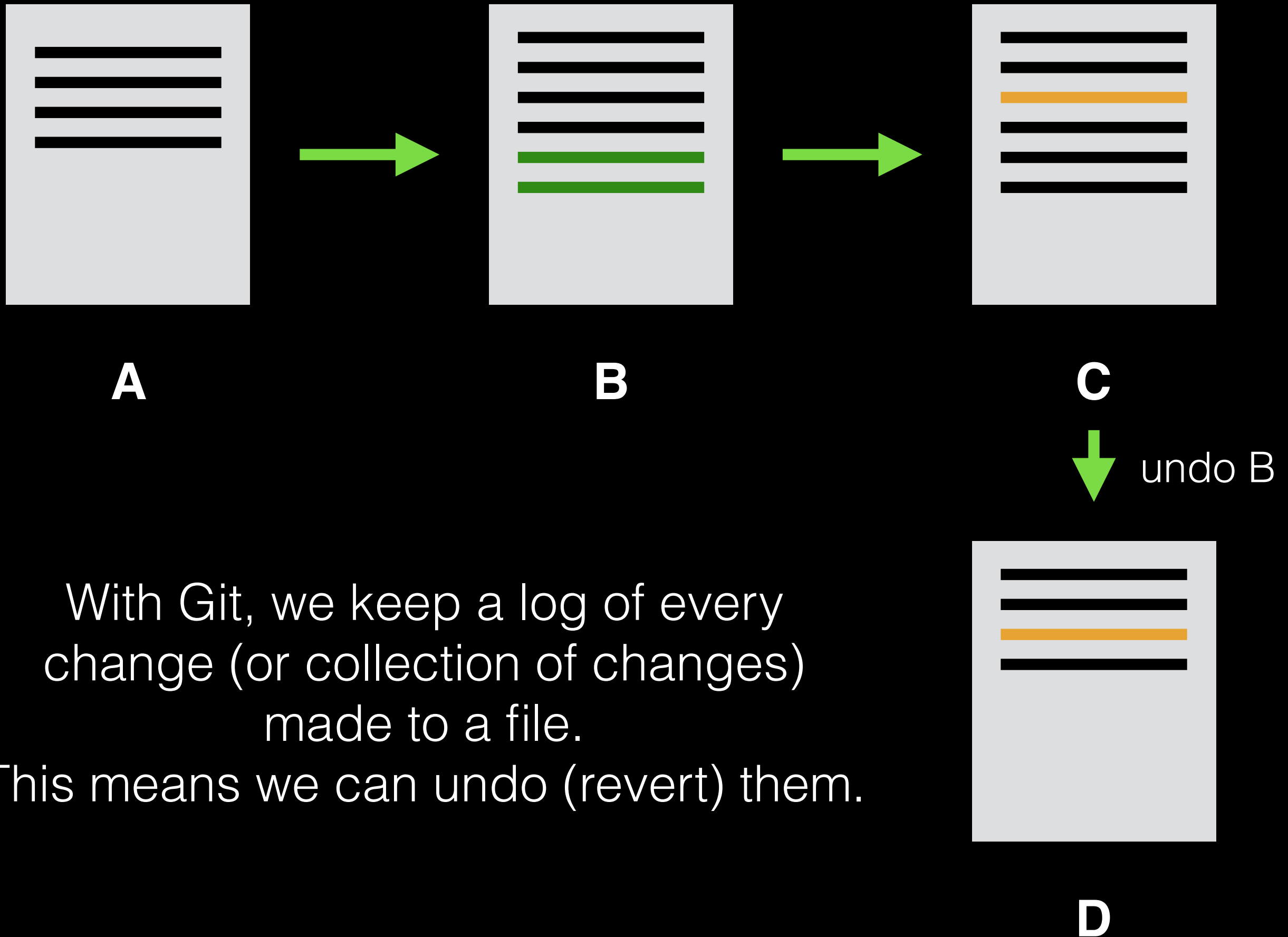


JORGE CHAM © 2012

WWW.PHDCOMICS.COM



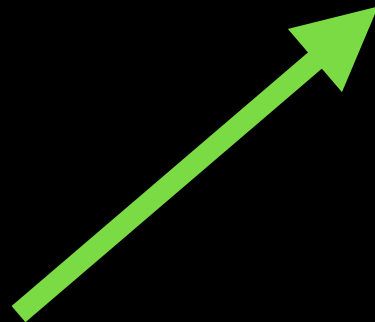
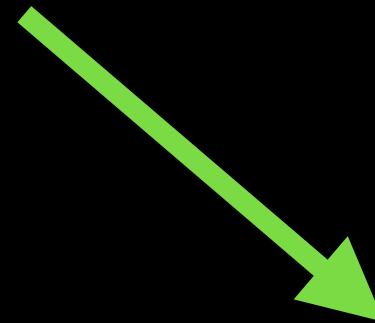
<http://phdcomics.com/comics.php?f=1323>



Changes
from person 1



Changes
from person 2



We can also
(automatically)
merge changes
from multiple
collaborators

Git != GitHub

- Git
 - Command-line software for managing file versions and project development
- GitHub
 - Web service hosting projects and providing extra features e.g. to facilitate collaboration

Today

- Follow along with what I do
- I can't teach you everything in half a day (or even two days)
- I can teach you >90% of what you **need**
- ...and make it less scary along the way

Acknowledgments

- Material inspired by/based on work by
 - Luis Pedro Coelho
 - Holger Dinkel & Grischa Toedt
 - Software Carpentry
 - Code Refinery

Let's get started!

Exercise

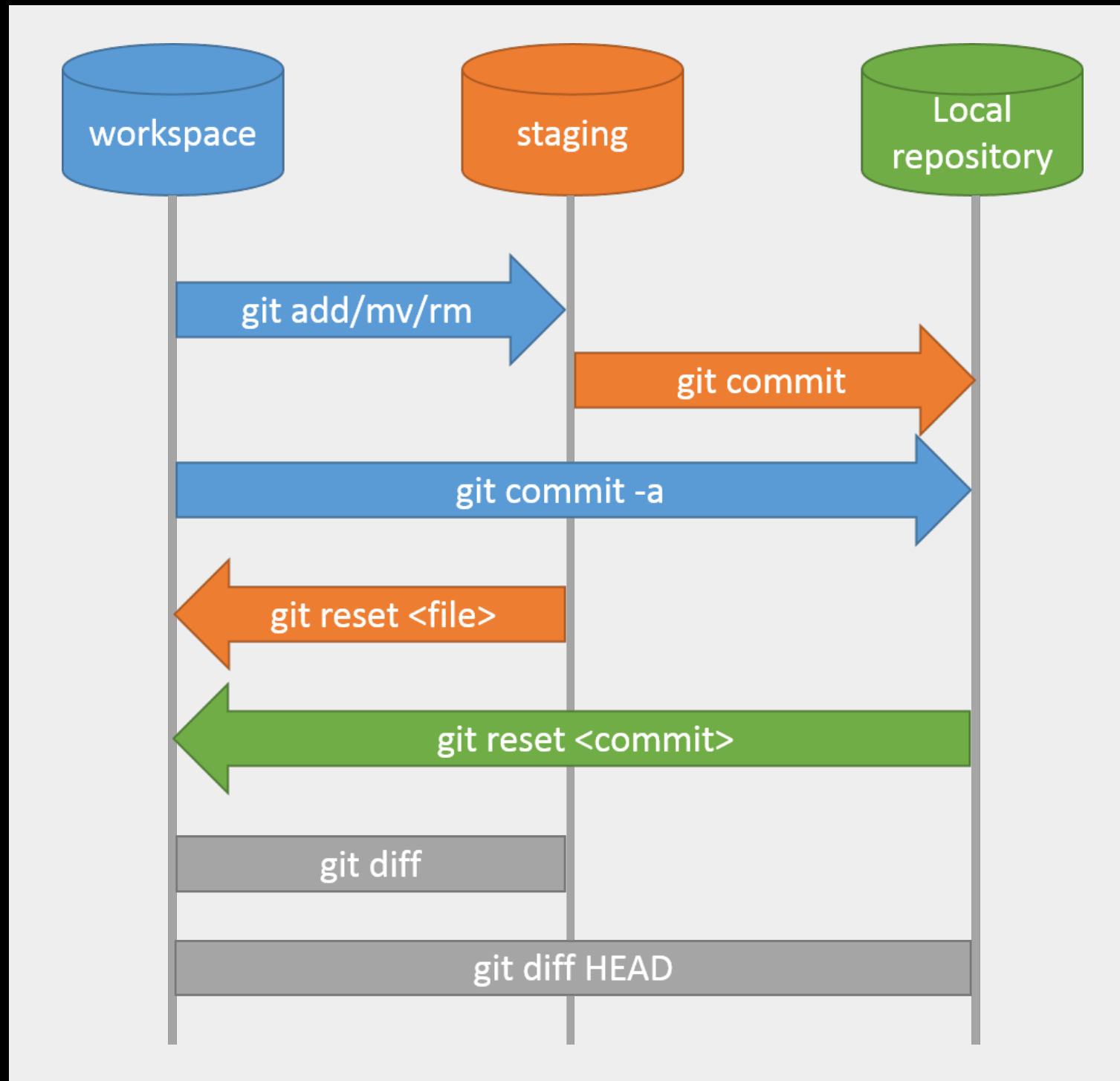
Now it's your turn...

1. add a line, “* enjoy!” to the end of `instructions.txt`
2. add and commit this change to your project history

Exercise

1. make some changes to `README.md`
2. commit them to the repository
3. make some more changes
4. discard the most recent changes to `README.md`
5. revert `README.md` to the state it was in *before* you started this exercise

Okay, so what just happened?



(image courtesy of 'research bazaar')

<https://raw.githubusercontent.com/resbaz/lessons/master/git/git-local.png>

Exercise

1. Split into pairs - A & B
2. A adds B to her repository
3. B clones A's repository
(make sure to keep the folder names different!)
 - `cd ..`
 - `git clone https://github.com/userA/repo.git ~/Desktop/userA-repo`
4. B makes some changes locally, then adds, commits, pushes (during this, A makes **no** changes)

Exercise

1. B changes first line of hummus.md, adds, commits, **pushes before A**
2. A also changes that line, adds, commits
3. A should now try to push the changes...

Conflict resolution

```
$ cat file_with_merge_conflict
<<<<<<< HEAD
this is the file content in the local version
=====
this is the conflicting content that
was retrieved from the remote repository when
we tried to pull changes
>>>>>>> 25484e4dbed3e259db9a64d51ec7a74552998036
```

Exercise

- Swap roles and repeat the previous process, creating and resolving a merge conflict
- remember to commit and push the changes when you're done

Suggested reading

- The Git Parable - <http://tom.preston-werner.com/2009/05/19/the-git-parable.html>
- Git Concepts Simplified - <http://gitolite.com/gcs.html>
- Software Carpentry Git Lesson - <http://swcarpentry.github.io/git-novice/>
- Code Refinery Git Lesson - <https://coderefinery.github.io/git-intro/>
- GitHub Flow - <http://scottchacon.com/2011/08/31/github-flow.html>