



Version Control, MarkDown and Social Coding

Robert Oostenveld
Melanie Ganz
Cyril Pernet

Vresions

version

Version

Versions

How many of you have written a paper that has been peer reviewed and published?

How many versions of the manuscript did you need for that?

What are the similarities and differences in writing code and writing text?

Teacher notes: use some platform for students to share (preferably allow anonymous input for those not feeling comfortable with that - also nothing is mandatory here)

Formatting of information and documents

How do you write a text, i.e. which tools do you use?

- A napkin and a pen
- Word
- Notepad
- Google Doc
- LaTeX

How do you write a text if it includes figures, like a paper?

- Word
- LaTeX
- InDesign

How do you construct something that is mainly figures, like a poster?

- PowerPoint
- Adobe Illustrator

What is Version Control?

- keep track of all the changes to your code/software
- roll back to see how something worked at any point in time
- distributed VCS (either [git](#) or [mercurial](#))
- The more often you commit your code, the more fine-grained is the history you create
- Make sure you spend a little time thinking about this message to describe the change you've made
- Branching, merging for managing concurrent development is a must for collaborative work

<https://www.software.ac.uk/blog/2016-10-06-top-tips-version-control>



git



GitHub

Adopted from Sophie Arana, Kristijan Armeni

IMPRS conference, June 3, 2020

Dondrite git workshop, November 28, 2019

<https://osf.io/zvkyx/>

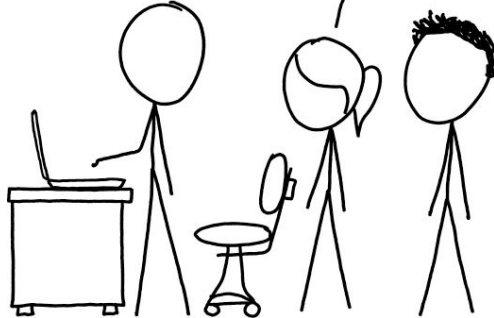
Some preliminaries

- we use Git/GitHub in our **day-to-day work**
- our work involves writing **data analysis scripts**
- **search engines** are our friends

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.



<https://xkcd.com/1597/>

For today

- Quick review of **markdown**
- **try out** working with Git and GitHub
- set you up with the **necessary minimum**
- **minimize the overhead** for your next projects
- inspire you to use VC and **continue learning yourself**

Version control is a system that **records changes to a file** or set of files **over time** so that you can recall specific versions later. [...] you can do this with nearly **any type of file** on a computer.

(Git Pro book)

"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

	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
○	AAAAAAAAA	3 HOURS AGO
○	ADKFJSLKDFJSDKLFT	3 HOURS AGO
○	MY HANDS ARE TYPING WORDS	2 HOURS AGO
○	HAAAAAAAAAANDS	2 HOURS AGO

AS A PROJECT DRAGS ON, MY GIT COMMIT
MESSAGES GET LESS AND LESS INFORMATIVE.

Why bother?



- safe and robust workflow, reduce risk
- reproducibility
- backup
- save time
- flexibility to experiment and prototype
- collaboration



```
modified:   .Rhistory

Untracked files:
(use "git add <file>..." to include in what will be committed)

docs/assets/images/OSCNlogo_inkscape_small.svg
docs/assets/images/OSCNlogo_inkscape_v2.svg
docs/assets/images/OSCNlogo_inkscape_v2_header.svg
docs/assets/images/contributing.png
docs/assets/images/logo_title_web.svg
docs/assets/images/oscn_launch_flyer_.png
src/OSCN_questionnaire.html
src/OSCN_questionnaire.md
src/OSCN_questionnaire.nb.html
src/OSCN_questionnaire_files/
src/devtmp.R
src/figures/

no changes added to commit (use "git add" and/or "git commit -a")

Kristijan@DESKTOP-LN31DSA MINGW64 /c/users/kristijan/documents/git/osc-
nijmegen (master)
$ git remote show
kriarm
origin

Kristijan@DESKTOP-LN31DSA MINGW64 /c/users/kristijan/documents/git/osc-
nijmegen (master)
$ git remote show origin
```

the basics 🧒 ...

git [verb] [--options] [arguments]

Basic workflow

project folder

git repository

tracked file

file with changes

file with committed
changes



git config

git status

git diff
git log

git init
git clone

git add

git add

git commit

intro

version control

working with git

git basics





demos

Demo and exercise 1  ...

Adding remotes to the mix 🌟...



GitHub

 Search or jump to...  [Pull requests](#) [Issues](#) [Marketplace](#) [Explore](#)  

[Radboud-University](#) / [osc-nijmegen](#) [Unwatch](#) 7 [Star](#) 2 [Fork](#) 13


[Code](#) [Issues 11](#) [Pull requests 0](#) [Actions](#) [Projects 2](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)

Open Science Community Nijmegen [Edit](#)

[open-science](#) [open-data](#) [open-access](#) [Manage topics](#)

[372 commits](#) [1 branch](#) [0 packages](#) [0 releases](#) [1 environment](#) [10 contributors](#)

Branch: [master](#) [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

 [KristijanArmeni](#) Update initiatives.md ✓ Latest commit 80c50e6 18 hours ago

docs	Update initiatives.md	18 hours ago
src	push the changes that were used to produce the output for the OSC-NL ...	5 months ago
.Rhistory	first commit from Rstudio Cloud	9 months ago
.all-contributorsrc	docs: update .all-contributorsrc	9 days ago
.gitignore	added <data> folder such that git ignores any data files we put there...	9 months ago
README.md	docs: update README.md	9 days ago

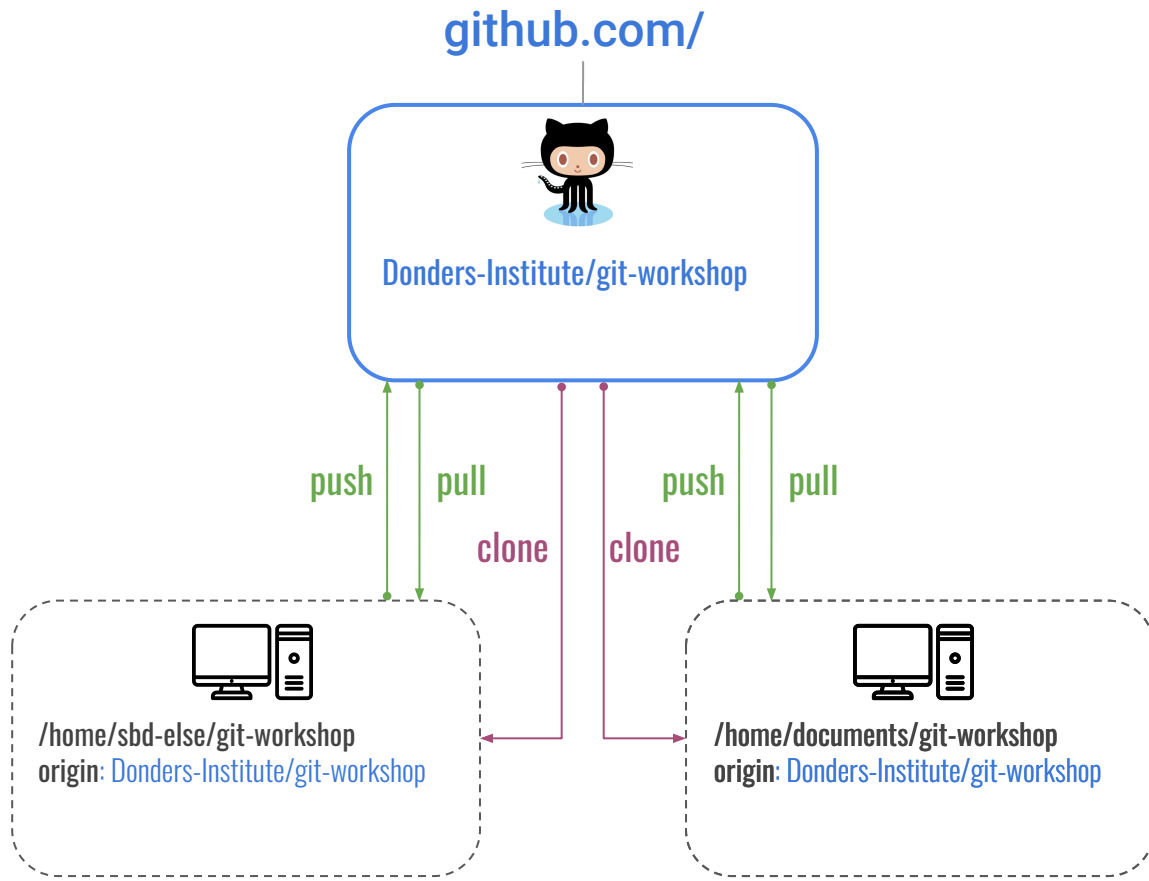
intro

version control

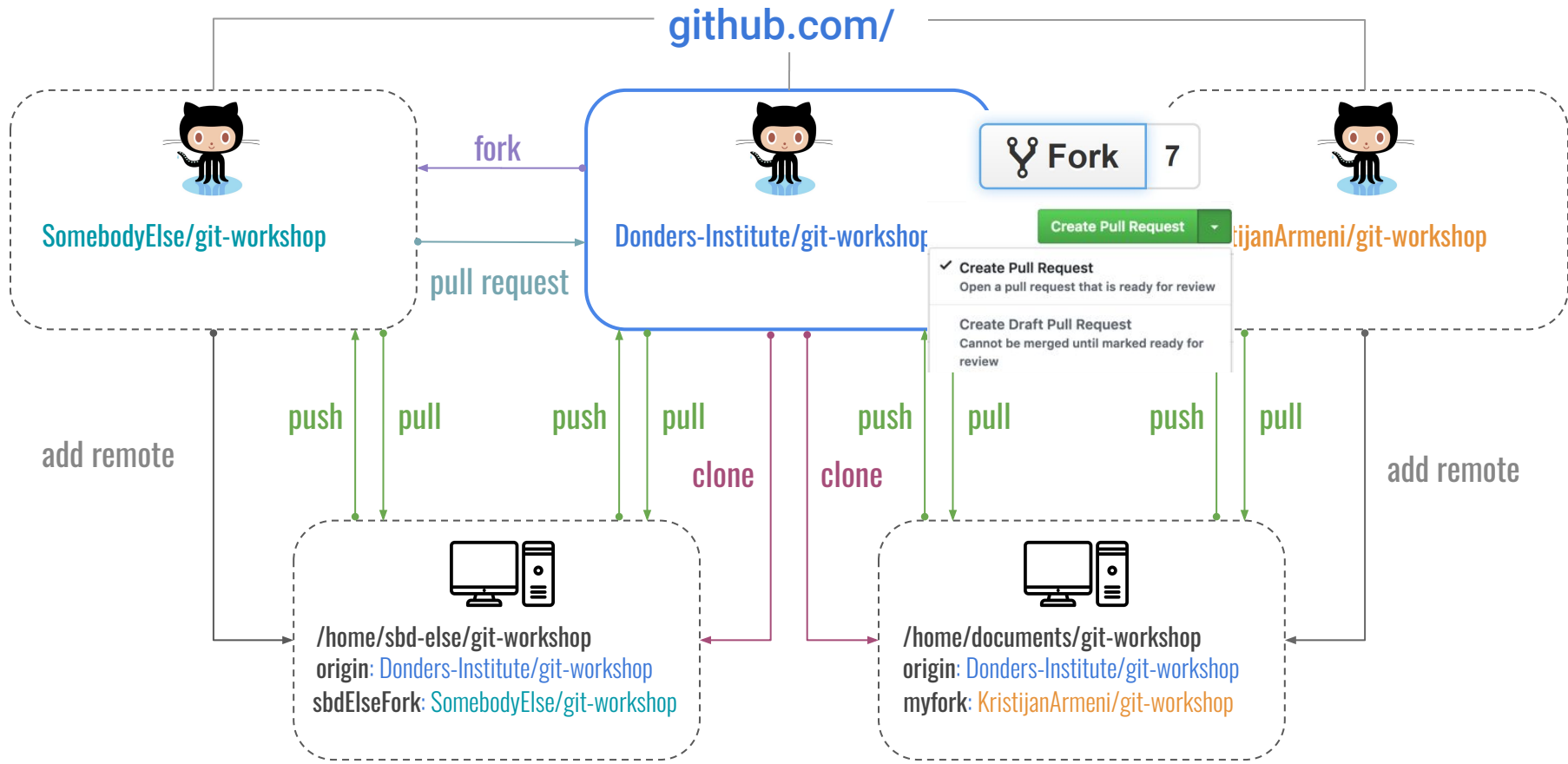
working with git

git basics

demos



Adding forks to the mix 🍴 ...



Hands-on

- we have issues!
- focus on git and GitHub
- explore, try out, ask

Branching  ...

Branches

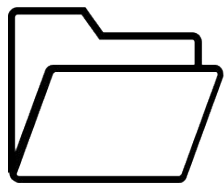
project folder

git repository

tracked file

two branches with changes

file with merged changes



git config

git status

git diff
git log

git init
git clone

git add

git checkout -b
git merge
git branch

git commit

intro

version control

working with git

git basics

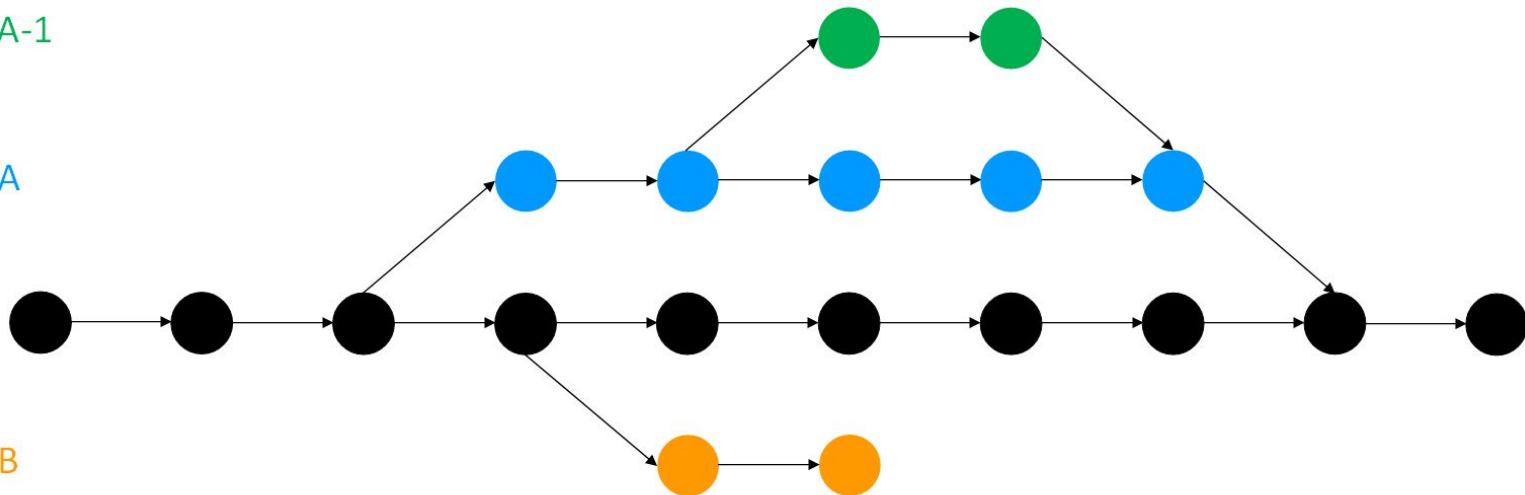
demos

Feature A-1

Feature A

Master

Feature B

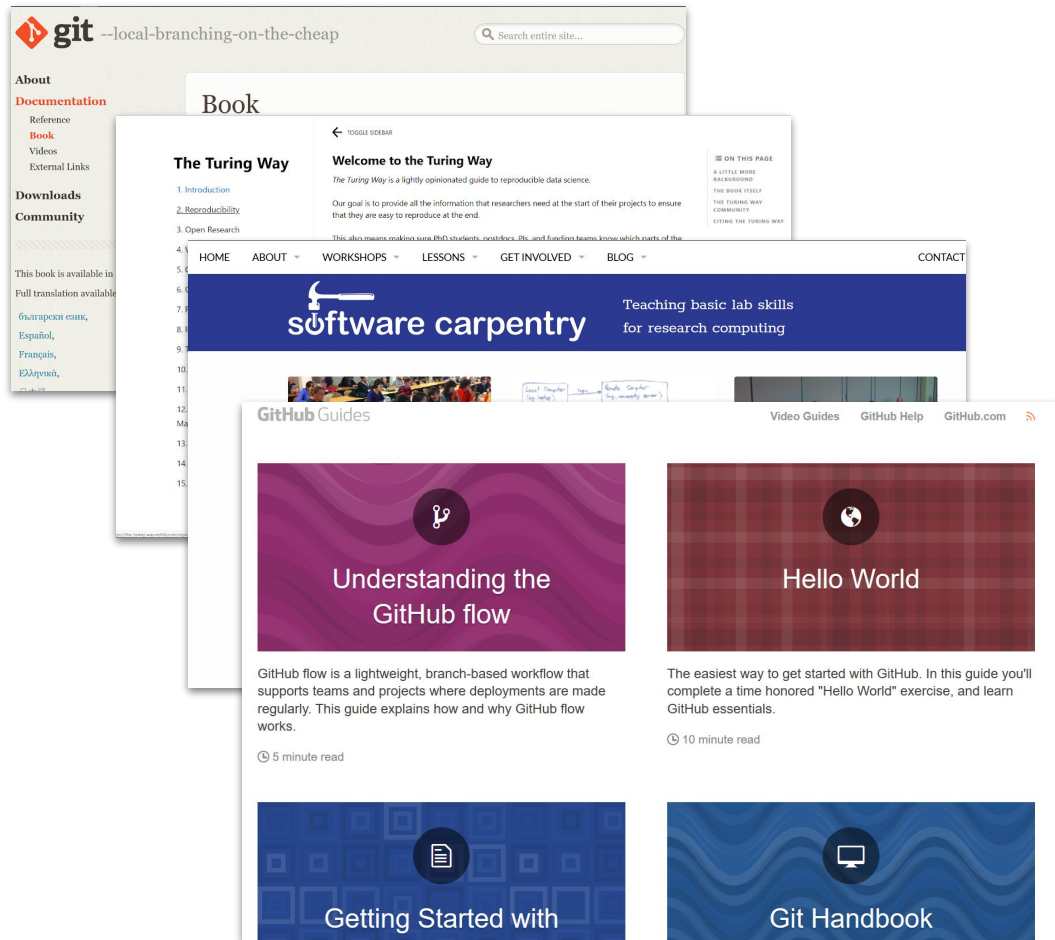


<https://the-turing-way.netlify.com/version-control/version-control.html>



Useful resources

- **Git Pro Book**
<https://git-scm.com/book/en/v2>
- **The Turing Way**
<https://the-turing-way.netlify.com>
- **Software Carpentry**
<https://swcarpentry.github.io/git-novice/>
- **GitHub Guides**
<https://guides.github.com/>



Acknowledgments

Icons and adaptations used in this slideshow designed by Nikita Golubev and Smashicons obtained from www.flaticon.com

Git and GitHub Exercise

Since you need to **clone** and **pull** (which anyone can do) and **push** (which needs permission to write), we need your github accounts.

Please share these on

<https://hackmd.io/@robertostenveld/B17w4SdOh/edit>

Head over to <https://github.com/Donders-Institute/git-workshop> and follow the instructions there.