

GIS analyses with Free and Open-Source Software

Introduction

Christina Ludwig

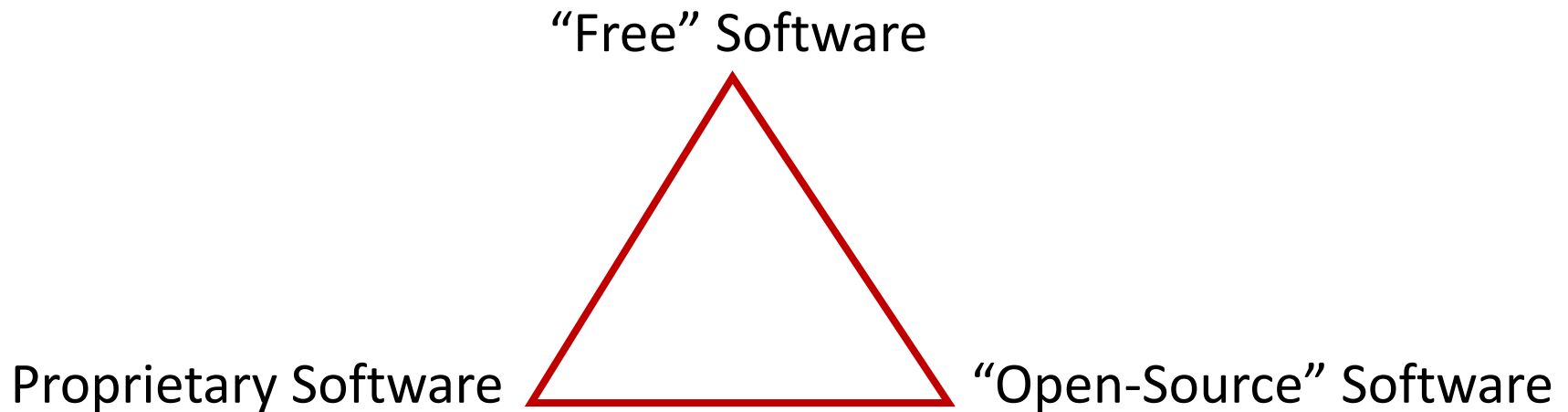
Seminar im Wintersemester 2019/2020

Arbeitsgruppe Geoinformatik,
Geographisches Institut



**UNIVERSITÄT
HEIDELBERG**
ZUKUNFT
SEIT 1386

What is the difference?



2

Think for yourself (1 min),
Discuss with your neighbour (2 min)
Share with the group

Definition FOSS

FOSS: Free and Open-Source Software

- Anyone is freely licensed to use, copy, study, and change the software in any way.
- „free software" does not refer to the monetary cost of the software at all, but rather whether the license maintains the software user's civil liberties.
- "free" as in "free speech," not as in "free beer"
- It is developed by one or several developers that contribute on a voluntary basis (usually without being payed).

GNU-Manifest (1985)

4 degrees of freedom (after R.M. Stallman):

- ❑ Freedom 0: The **freedom to run the program**, for any purpose.
- ❑ Freedom 1: The **freedom to study how the program works**, and adapt it to your needs.
- ❑ Freedom 2: The **freedom to redistribute copies**.
- ❑ Freedom 3: The **freedom to improve the program**, and release your improvements to the public, so that the whole community benefits.

Free Software Foundation (FSF): „Open-source is a development methodology; free software is a social movement.“



What are the advantages and risks of using
Free and Open-Source Software (FOSS)?

Think for yourself (1 min),
Discuss with your neighbour (2 min)
Share with the group

Benefits and risks of using FOSS

- Lower costs or no costs
 - Personal control, customizability and freedom
 - Privacy and Security
 - Linus Throvald: "Given enough eyeballs, all bugs are shallow."
 - Better Collaboration
 - (fast) collaborative development
 - Not dependent on the financial success of the company
- often it depends on the project and its community support

See: [Wikipedia](#) for details

How to Ensure the Openness?

- **Licenses, e.g.,**

- GNU General Public License (GPL)
- www.gnu.de/gpl-ger.html

- **Preamble **GPL-3.0**:**

“The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program--to make sure it remains free software for all its users.

[...]“

OSGeo Foundation / FOSSGIS e.V. ...

- Open Source Geospatial Foundation (**OSGeo**): osgeo.org
- Non-profit, non-governmental organization to promote the development of open geospatial technologies and geodata
- German local group: FOSSGIS e.V. : **F**reie und **O**pen **S**ource **S**oftware für **G**eo**I**nformations**S**ysteme

 **FOSSGIS-Konferenz 2020**
Freiburg
11.–14. März



OSGeo Software Projects



<https://live.osgeo.org/>

- Mix of Desktop GIS, WebGIS, GIS libraries, databases, ...
- We will only work with a selection of these, but you're encouraged to use others, if you need them.
- Plus: Selected github projects from the Heidelberg GIScience group

GDAL and OGR Libraries

- GDAL: **G**eospatial **D**ata **A**bstraction **L**ibrary
- OGR: **O**pen**G**IS Simple Features **R**eference Implementation

„GDAL (and OGR) is a translator library for raster and vector geospatial data formats“ (gdal.org)

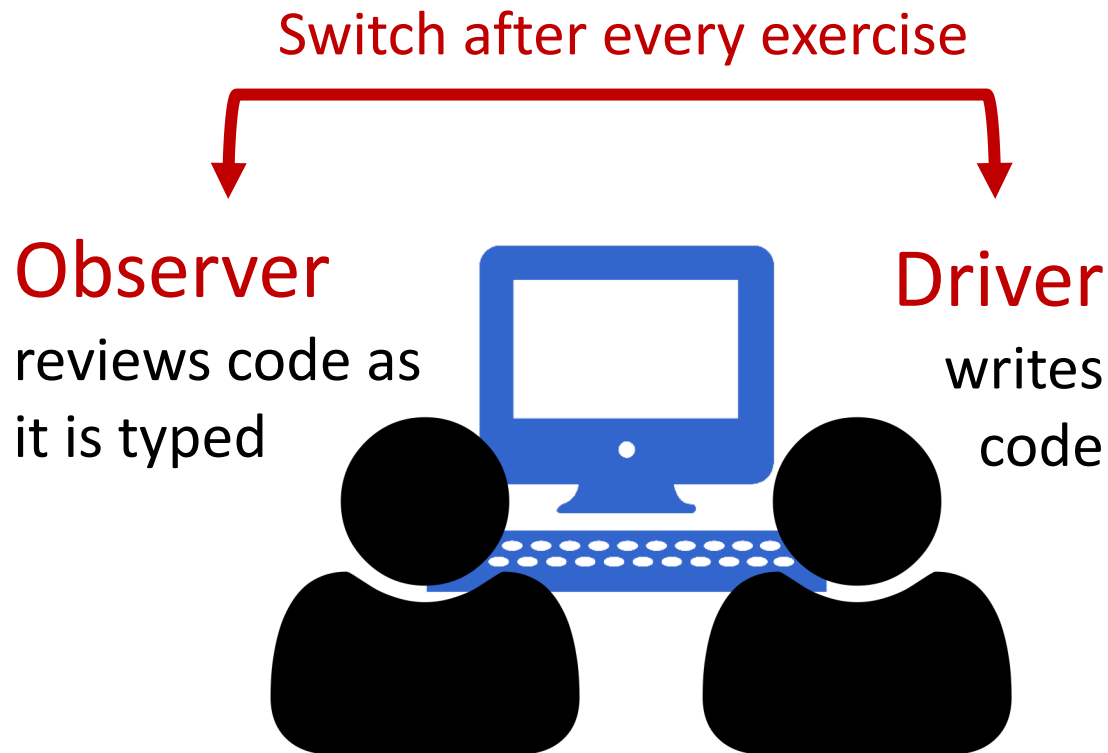
- Used for file format conversion, reprojection, etc.
- It is often used for **pre- or post-processing**
- Available as **command line tools or bindings** to various scripting languages (e.g. Python, R, Java)

Practical Part

1. Introduction to the command line and git
2. GDAL and OGR exercises

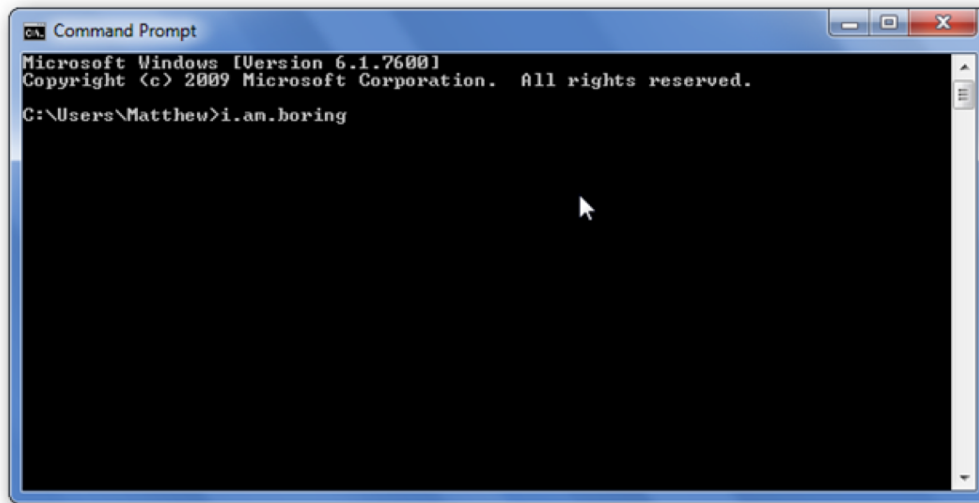
Pair Programming

- Agile Software Development method, but great for courses too

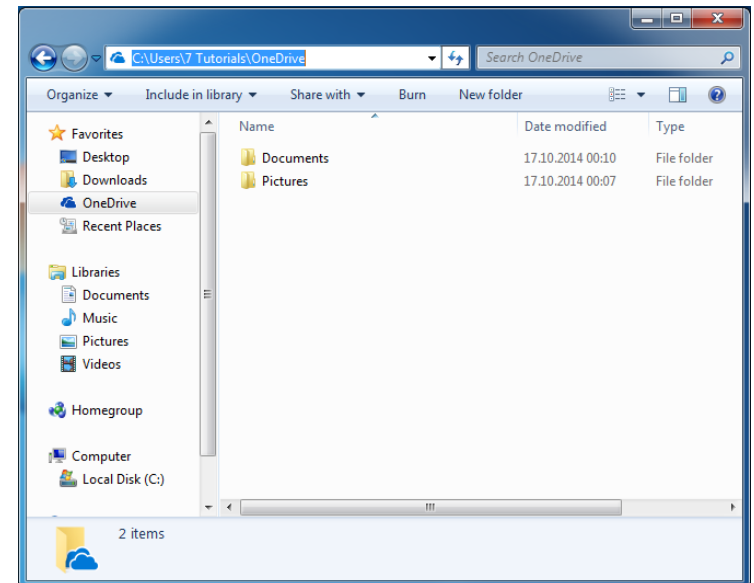


Introduction to the command line

- The **command line** (also called command-line interface, command prompt, shell), is a **text-based** application for **viewing, handling, and manipulating files** on your computer.
- Similar to Windows Explorer, but without graphical user interface



Text-based



"button/click based"

Introduction to the command line

- General Syntax: `command` + `[optional parameters]`

`dir Desktop /b` 

- Reference to current directory is "."
`dir .`
- Reference to parent directory is ".."
`cd ..`
- Write output of command to file e.g.
`dir > contents.txt`
- Print content of file in command line e.g.
`type contents.txt`
- If things go wrong, abort the command:
 - STRG + c

Flag: an option that can be set for a command. For some commands using a "minus" e.g. `git commit -m`

Important commands for the command line

Windows command	Mac OS / Linux command	Description	Example
exit	exit	close the window	exit
cd	cd	change directory, When changing hard drives (C:\ to M:\) execute M:\ afterwards	cd M:\Documents
cd	pwd	print working directory	cd
dir	ls	list directories/files	dir
copy	cp	copy file	copy c:\test\test.txt c:\windows\test.txt
move	mv	move file	move c:\test\test.txt c:\windows\test.txt
mkdir	mkdir	Make (new) directory	mkdir testdirectory
rmdir (or del)	rm	Remove directory	del c:\test\test.txt
rmdir /S	rm -r	Remove directory recursively	rm -r testdirectory

Hands-On

Documenting commands

- List your commands in the order that you execute them
- Add a brief comment that explains what the following command does
 - For this course, a leading “#” indicates a comment (Python style)
 - Generally comments are indicated with different sign depending on the language used

For example:

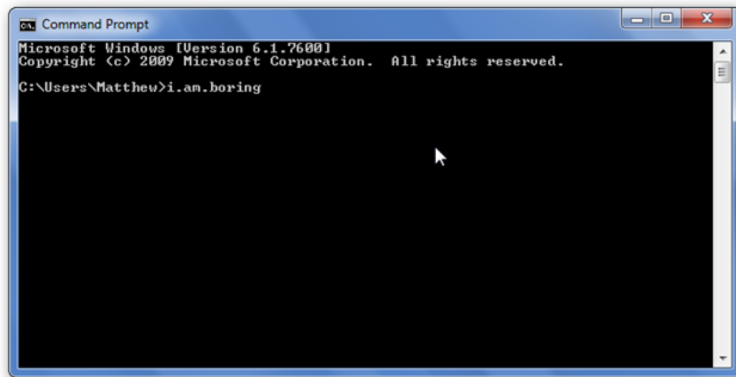
```
# Change to my git directory  
cd ./my_git_repos
```

```
# List all files in this directory  
dir
```

← comment line describing the command below
← This is the command

Normal command line vs. OSGeo Shell

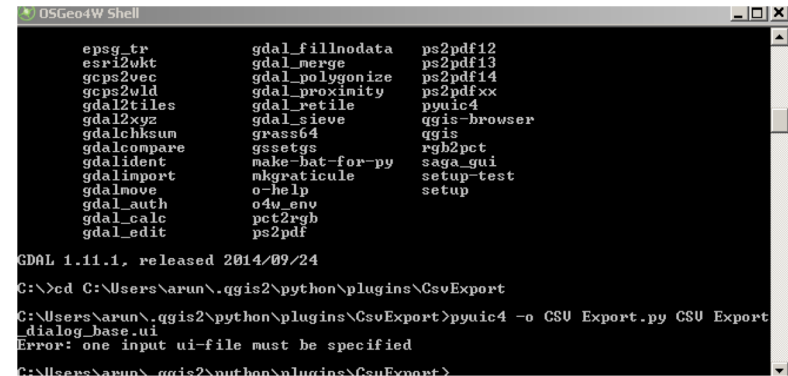
Standard Command line / “Eingabeaufforderung”



```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Matthieu>i.am.boring
```

OSGeo Shell



```
epsg_tr      gdal_fillnodata  ps2pdf12
esri2wkt     gdal_merge       ps2pdf13
gcps2vec     gdal_polygonize  ps2pdf14
gcps2wid     gdal_proximity   ps2pdfxx
gdal2tiles   gdal_retile       pyuic4
gdal2xyz     gdal_rieve        qgis-browser
gdalchksum   grass64           qgis
gdalcompare  gssetgs           rgb2pct
gdalident    make-bat-for-py   saga_gui
gdalimport   mkgraticule       setup-test
gdalmove     o-help            setup
gdal_auth    o4w_env           ps2pdf
gdal_calc    pct2rgb
gdal_edit    ps2pdf

GDAL 1.11.1, released 2014/09/24

C:\>cd C:\Users\arun\.qgis2\python\plugins\CsvExport
C:\Users\arun\.qgis2\python\plugins\CsvExport>pyuic4 -o CSU_Export.py CSU_Export
_dialog_base.ui
Error: one input ui-file must be specified
C:\Users\arun\.qgis2\python\plugins\CsvExport>
```

Available commands	Standard Command line	OSGeo Shell
Standard Windows commands	yes	yes
Git commands	yes	no
OSGeo commands	no	yes

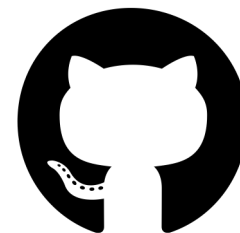
→ You have to switch sometimes between these if you want to execute certain commands

Introduction to git

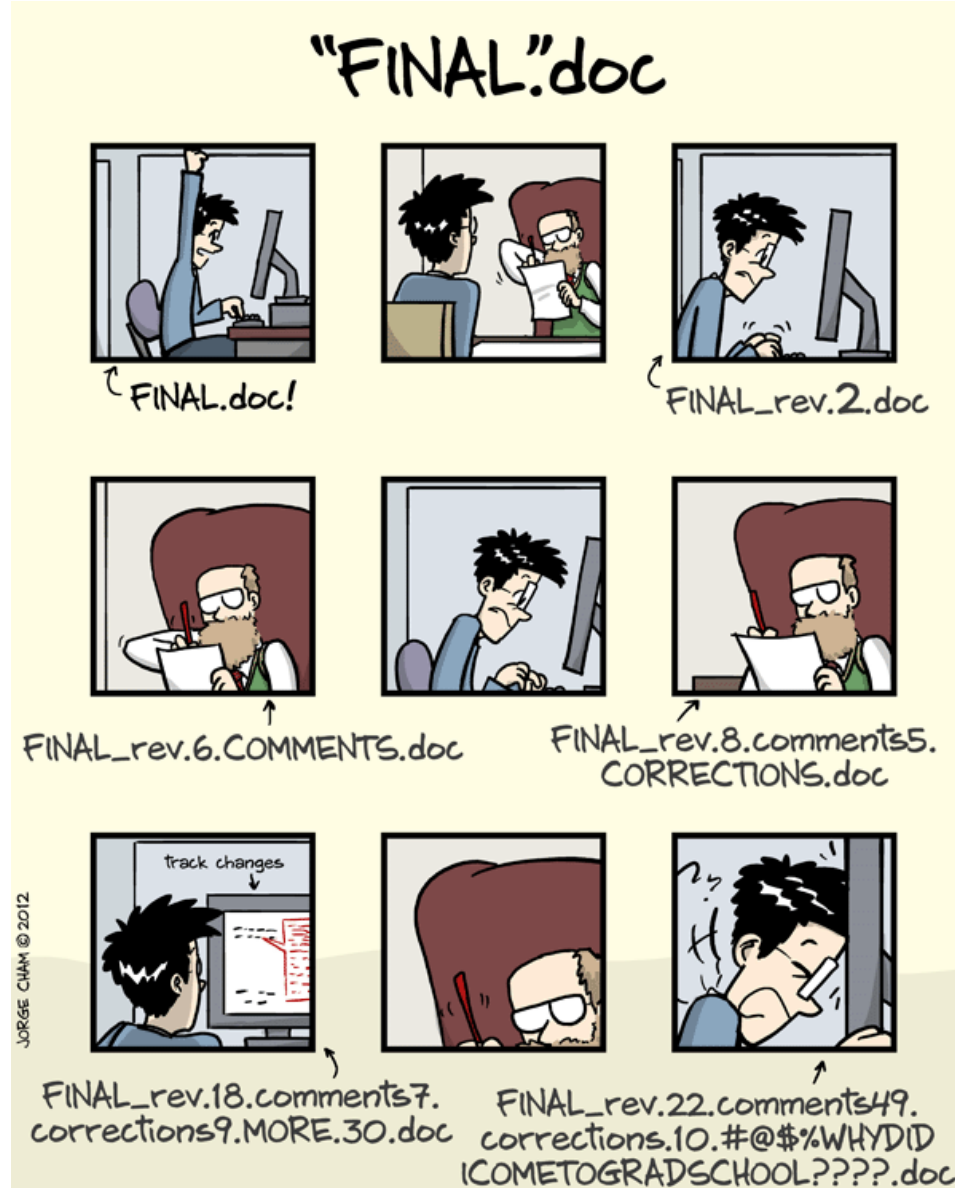
- **Git is a Version Control System:** tracks the changes made to computer files and coordinates work on those files among multiple people.
 - Developed for **collaboration on open source software** by Linus Torvalds (principal developer of Linux)
 - Different implementations e.g. **GitHub**, gitlab, bitbucket, ...
- all your favourite FOSSGIS projects are on GitHub



GitHub

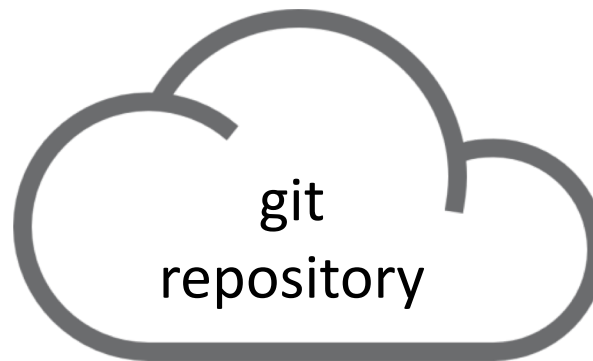


Why would you need git as researchers?



JORGE CHAM © 2012

Concept of git



REMOTE
Server

repository = "repo":
a directory containing
all the files of your
project whose
changes should be
tracked

Synchronizing
changes (git pull)

"clone a repository":
create a *local* copy of a
repository on your
computer and sync
between the two
locations



LOCAL
computer

Local copy of the repository on your computer

Hands-On in Pairs!

Clone this git repository to your computers

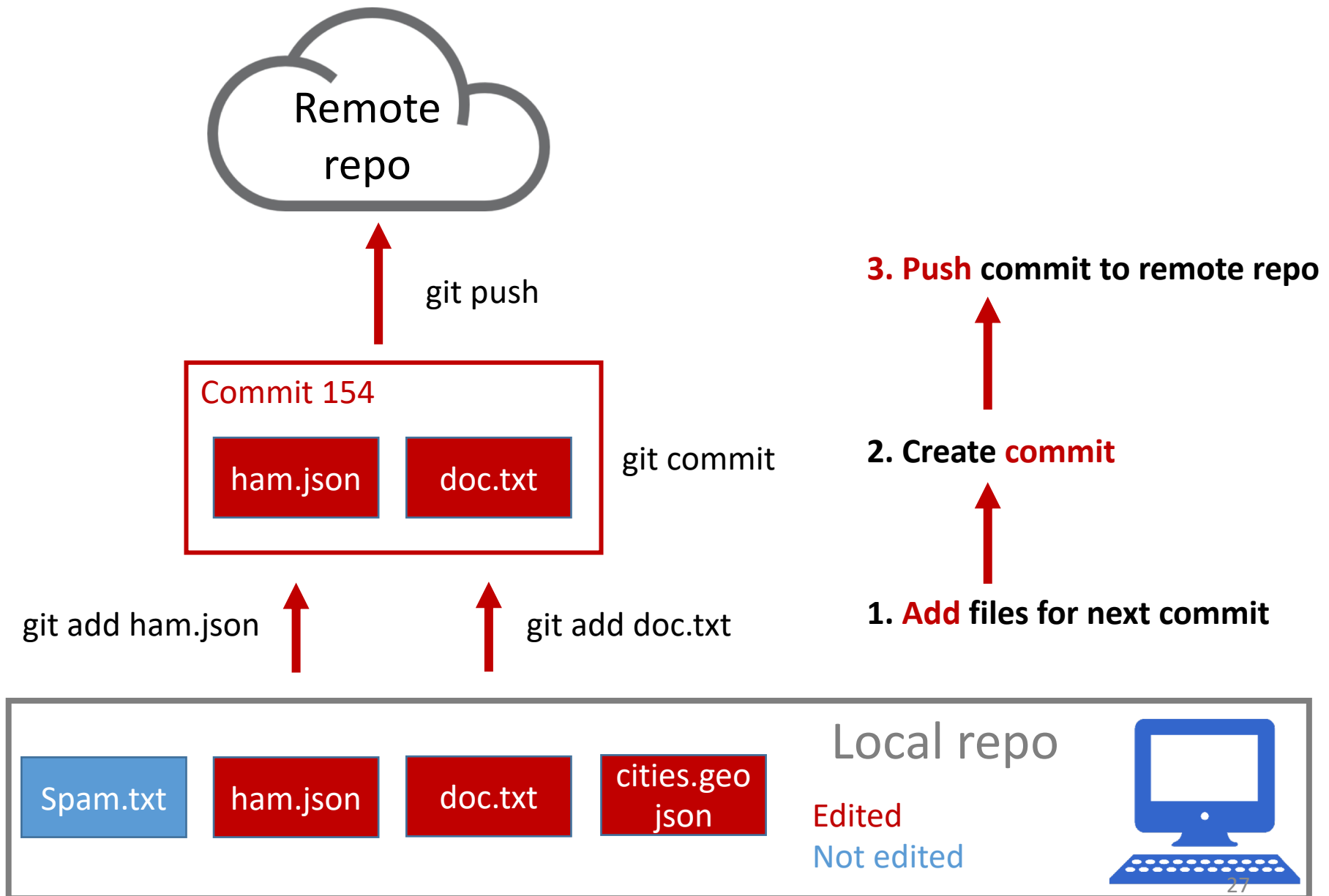
→ <https://github.com/redfrexx/cs4geos19>

I will call this the

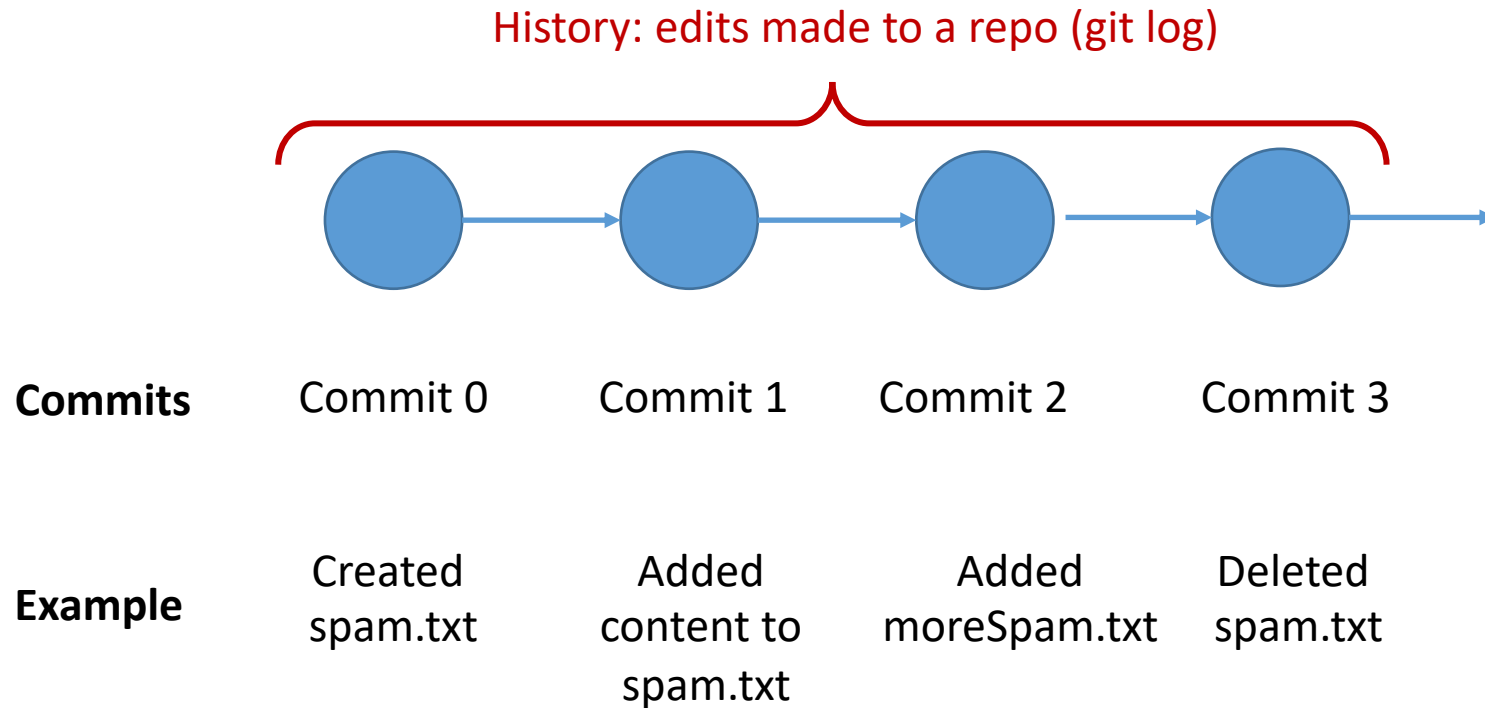
“course repository”

from now on.

How to add changes to the remote repository



Tracking changes of a repository



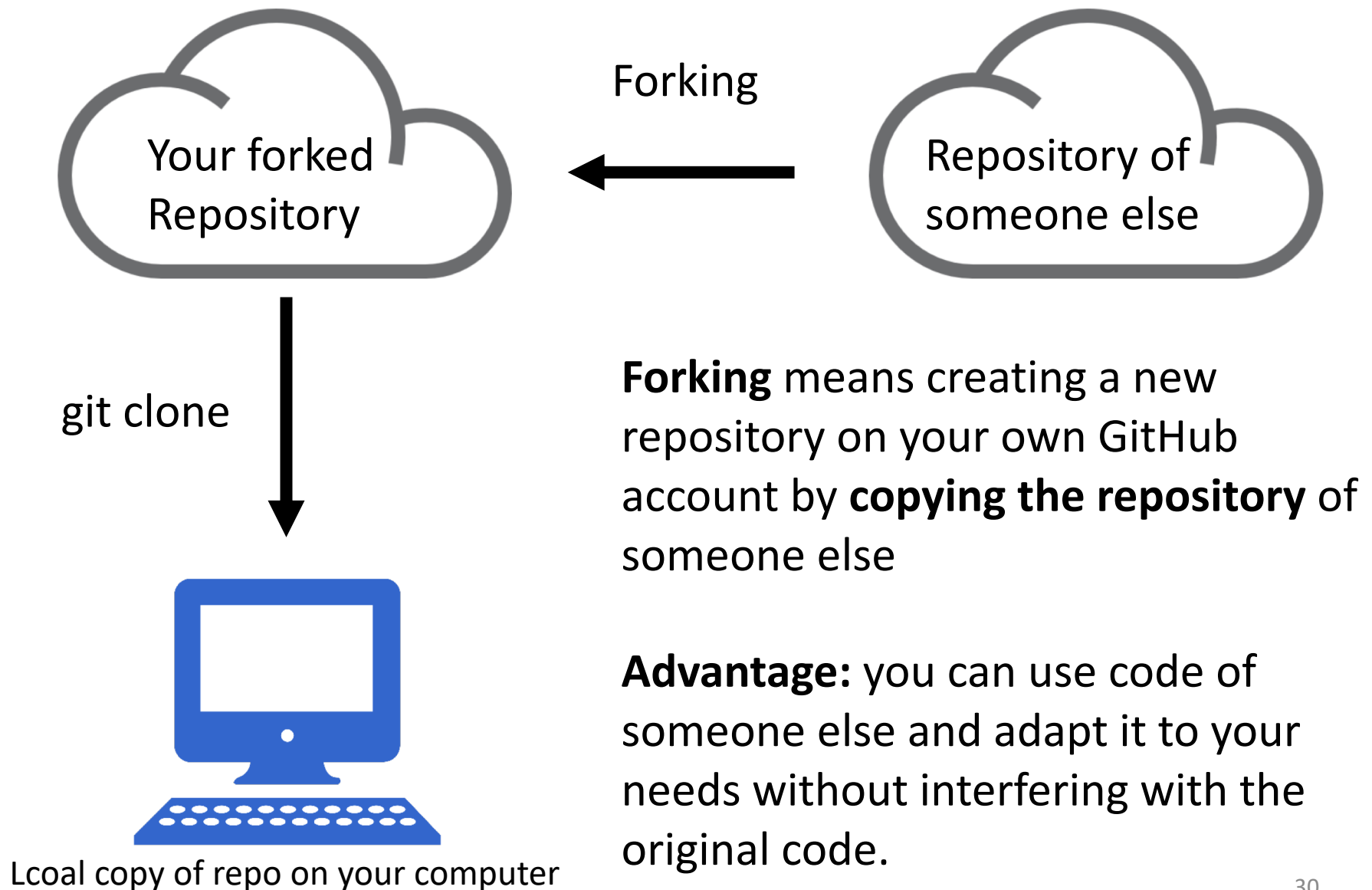
Possible to track all changes:

- Which changes were made?
- Who made the changes?
- When were the changes made?
- Why were changes needed?

Hands-On in Pairs!

1. Create a new repository on GitHub.
2. Clone it to your computer.
3. Create a file in it.
4. Add, commit and push the file to the remote repository.

Forking: Creating a copy of a repository



Important git commands

Get a local copy of a remote repository:

- **git clone *github_url***

e.g. `git clone https://github.com/GIScience/openrouteservice.git`

How to add changes to the remote repository:

- **git add** : add a file that has been modified to be included in the next commit (“stage file for next commit”) e.g. `git add spam.txt`
- **git commit -m “*your description*”**: create a commit with a description of the changes e.g. `git commit -m “added spam.txt”`
- **git push** : upload the commit to the remote repository

Retrieving changes from the remote repo to your local repo

- **git pull**

Important git commands

- Setting up name and e-mail address

- `git config --global user.name "Your Name"`
- `git config --global user.email your_email@whatever.com`

You need to do this only once when you create your first commit.

Show history of commits

- **`git log`**

Check status of repository

- **`git status`**

Hands-on in Pairs!

1. Create a text file called *yournames.txt* within the folder `""` of the course repository. Replace *yourenames* with your names.
2. Push your changes to the remote course repository.

Command line

- [Introduction to the command-line interface](#)
- [Windows command line commands cheat sheet](#)

Git

- [Git How To](#)
- [Git Cheat Sheet](#)
- [Git Branching](#)

Assignment 1:

Submission on moodle until:

12.10.2019 23:55