

# Workshop on Version Control with Git

Edith Scheifele (B7, Z2)

24-01-2020

# Aim for today

- ▶ get to know git
- ▶ download and configure git
- ▶ use basic functions: git init, git add, git commit, git status, git log
- ▶ create an Github account
  - ▶ fork and clone the repository with the slides for today

# Introduction

- ▶ created 2005 by Linus Torvalds (the creator of Linux) for the development of the Linux Kernel
- ▶ system for recording changes and documenting them

# Reasons to use git

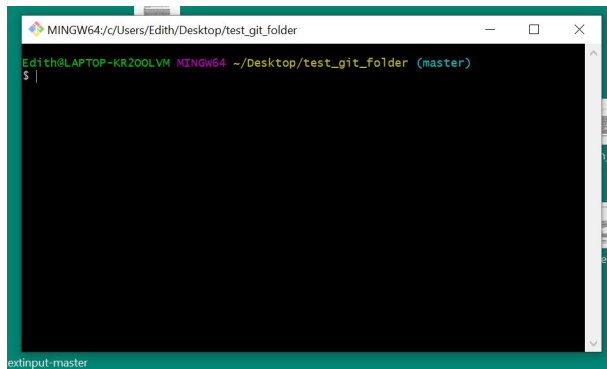
- ▶ better collaboration
- ▶ better tracking and documentation of changes
- ▶ some researchers place their data and code on Github & Co.

# Download and Configuration

- ▶ download git from: `https://git-scm.com/downloads`
- ▶ install it

# Git bash

- ▶ we will use the bash shell that comes with git

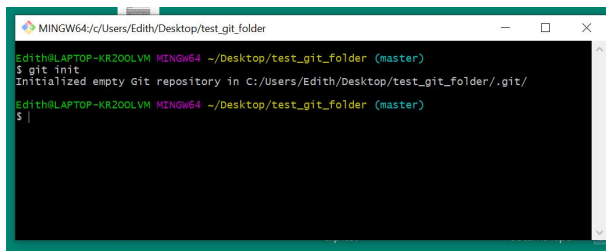
A screenshot of a Git Bash terminal window. The title bar shows the path "MINGW64:/c/Users/Edith/Desktop/test\_git\_folder". The terminal content shows the prompt "Edith@LAPTOP-KR2OOLVM MINGW64 ~/Desktop/test\_git\_folder (master)" followed by a dollar sign "\$" and a cursor. The window has a green border and standard Windows window controls. The text "extinput-master" is visible at the bottom left of the window frame.

```
MINGW64:/c/Users/Edith/Desktop/test_git_folder

Edith@LAPTOP-KR2OOLVM MINGW64 ~/Desktop/test_git_folder (master)
$ |
```

# Repositories

- ▶ create a folder on your desktop called `test_git_folder`
- ▶ this will become your Git **repository (repo)**
- ▶ right click on it > Git Bash Here
- ▶ initialize the repository by typing *git init* into the shell



```
MINGW64:/c:/Users/Edith/Desktop/test_git_folder
Edith@LAPTOP-KR200LVM MINGW64 ~/Desktop/test_git_folder (master)
$ git init
Initialized empty Git repository in C:/Users/Edith/Desktop/test_git_folder/.git/
Edith@LAPTOP-KR200LVM MINGW64 ~/Desktop/test_git_folder (master)
$ |
```

# Basic functions

- ▶ every git function is prepended with *git*
- ▶ the core functions are:
  - ▶ `git init`
  - ▶ `git add`
  - ▶ `git commit -m "some infos"`
  - ▶ `git`
- ▶ also very useful:
  - ▶ `git status`
  - ▶ `git log`



## git status & git log

- ▶ **git status**: shows the status of your project

### Task:

- ▶ type *git status* into the command line to view the status of your project
- ▶ **git log**: shows the commit logs (history of your changes)

### Task:

- ▶ type *git log* into your command line

## git logic

- ▶ three main stages of a file: *untracked* - *tracked* / *staged* - *committed*
- ▶ if you start your project with *git init*, git becomes active (it creates a hidden directory *.git*)
- ▶ if you place a file into your folder, git becomes aware of it as *untracked*

### Task:

- ▶ create a txt file (e.g., myfile.txt) with only one line of text and save it into your folder
- ▶ now type *git status* again. Can you see your file under *Untracked files*?

## git add

- ▶ to add the file to the so called staging area, type *git add myfile.txt*
- ▶ git now deposits it into it so called index

### Task:

- ▶ do this for your txt file, then do *git status*

git commit -m "some infos"

- ▶ to actually record your changes, type: *git commit -m "describe the change you made"*
- ▶ do this for your txt file, then do *git status* and *git log*

# Task

- ▶ write a second line into your txt file
- ▶ re-do the git process above with *git add*, *git commit*, *git status* and *git log*

## See your changes: *git show SHA* and *git diff*

- ▶ write a third line into your file
- ▶ do *git add myfile.txt*
- ▶ then type *git diff myfile.txt* to see the change you made
- ▶ then commit the change
- ▶ every commit has its own unique SHA key (the very long string of numbers and letters)
- ▶ by typing *git show [SHA key]* you can see the committed change
- ▶ be careful: STRG + C / V don't work in the shell, use the menu

# Github

- ▶ <https://github.com/>
- ▶ online platform for your projects
- ▶ free and open source
- ▶ facilitates collaborating with others

## Github: Create an Account, Fork and Clone

- ▶ create an account at Github
- ▶ navigate to: `https://github.com/EdithScheifele/Workshop_2020_git`
- ▶ fork it
- ▶ then go to your Github page and clone the repo by copying the https link
- ▶ go to the desktop, open a bash shell
- ▶ type: `git clone [https link]`



# Wrap-up

- ▶ basic git commands: init, add, commit, status, log
- ▶ we are able to add and commit a change we made to a file
- ▶ we are able to review the change in the staged and committed stage
- ▶ we have a Github account and we are able collaborate by forking and cloning a repo

## Next steps

- ▶ use git for your projects (*practice makes perfect*)
- ▶ read the first chapters in the free book *Pro Git*

# Resources

`https://git-scm.com/`

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

Loelinger, Jon & McCullough, Matthew (2012): Version control with git.