DIS08 – Data Modeling

01 – Introduction – Tutorial Session

Fabian Haak, Technische Hochschule Köln, Cologne, Germany

Version: 2021-10-22

Technology
Arts Sciences
TH Köln

Topics for Today

1. GitHub + Classroom
2. git Setup + Tour
3. Working with git and GitHub

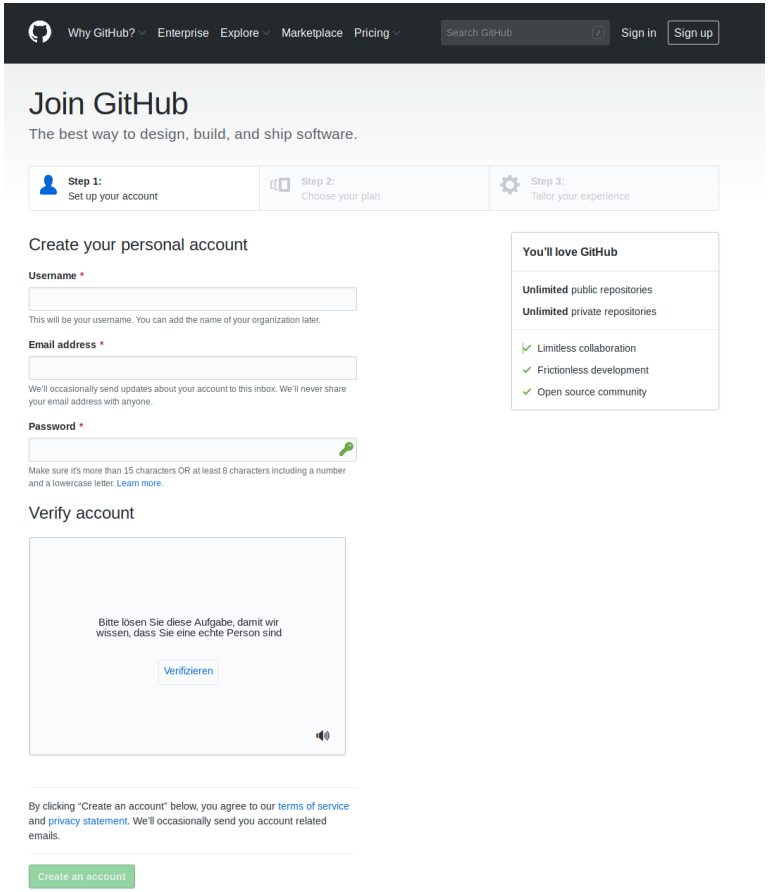
Topics for Today

1. GitHub + Classroom
2. git Setup + Tour
3. Working with git and GitHub

Assignment 1.1.a

Go to https://github.com/ and create a user account if you don‘t already have one.

(optional, but recommended): Apply for the Student Developer Pack at https://education.github.com/students (you should have your TH Köln Smail address in your account to accelerate approval).



Demo: Quick GitHub Tour

Assignment 1.1.b

- Via the invitation link we provide, accept your first assignment on GitHub.
- Make sure to pick your own student ID from the list!
- Follow the instructions and visit your individual repository page („Your assignment has been created here: […]”).

Topics for Today

1. GitHub + Classroom
2. git Setup + Tour
3. Working with git and GitHub

Install git

- Find download and installation instructions at:

https://git-scm.com/downloads

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

- Check your version to see if it‘s correctly installed:

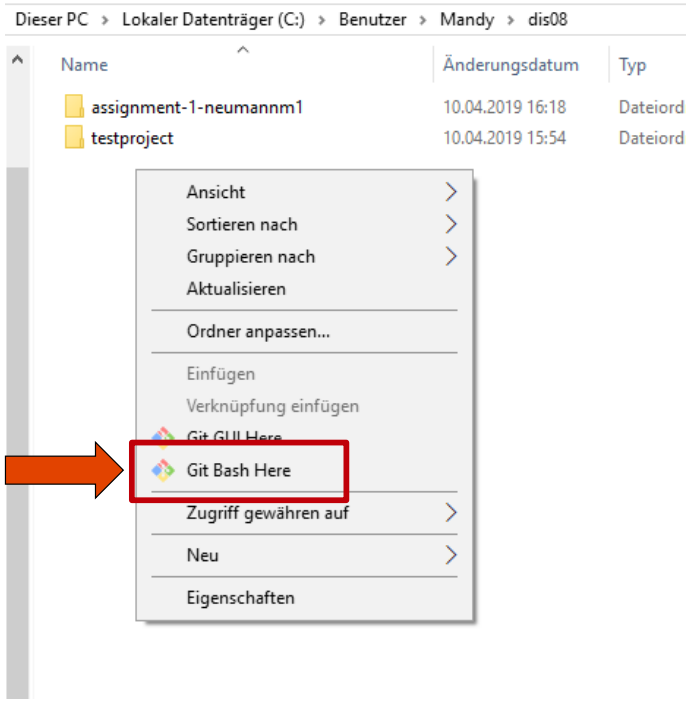
```
git --version
```

should output something like „git version 2.x.y“

- Our lab laptops have git for Windows installed.
- We will work on the git bash and ignore any GUI tools for now.

git Bash

- On Windows, you can right-click in any directory and select „Git bash here“ to open the git bash at this location.



- Otherwise, you need to navigate inside the bash to the directory you want to work in (use bash command `cd`).

Configure git

- Set your name and email (to be used in commit messages):

```
git config --global user.name "Mona Lisa"
git config --global user.email email@example.com
```

- On Windows (optional, but recommended): Change the default editor to e.g. Notepad++ like this:

```
git config --global core.editor "'C:\Program Files
(x86)\Notepad++\notepad++.exe' -multiInst –nosession"
```

- If you want to use another editor, you might need additional parameters – check Google for guides!

Demo: Quick git Tour

Summary of the git Tour

git command	Action
`git init <name>`	Initialize new repository with a name
`git add <file>`	Add file(s) to staging area
`git status`	Check status
`git commit`	Create a snapshot of staging area (opens configured editor for commit message)
`git commit -m`	Same as before, but enter commit message directly
`git log`	Show commit history

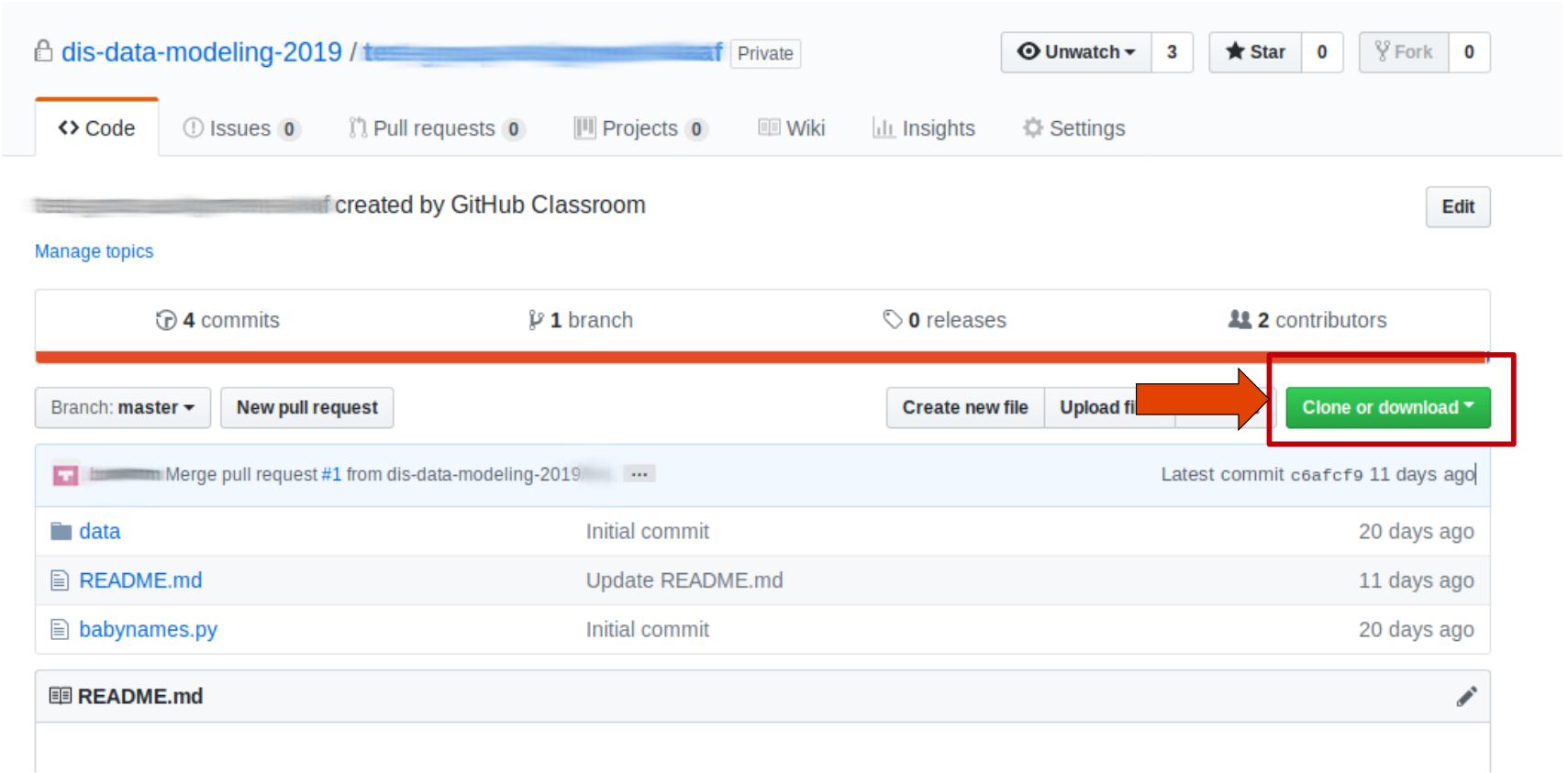
- Remember to navigate into the new directory after git init before working with it. Usually the workflow is:

```
git init newRepository
cd newRepository
```

Topics for Today

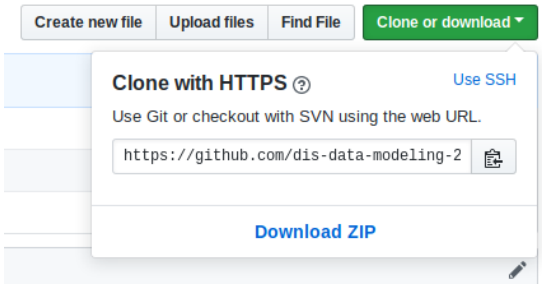
1. GitHub + Classroom
2. git Setup + Tour
3. Working with git and GitHub

Get Repositories Down to Your Machine



Get Repositories Down to Your Machine

- Copy the URL (use HTTPS method)…



- … and clone to your local machine:

```
$ git clone
https://github.com/irgroup-classrooms/dis08-2021/<repository>.git
Cloning into '<repository>'...
remote: Enumerating objects: 22, done.
remote: Counting objects: 100% (22/22), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 22 (delta 8), reused 15 (delta 8), pack-reused 0
Unpacking objects: 100% (22/22), done
```

- (you can give your local directory an alternative name by using `git clone <url> <name>`)
- Don‘t forget to change into the new directory! (-> `cd`)

Working with Cloned Repositories

- You now have a copy of the repository from GitHub on your local machine.
- You can work with it in the same way as with a repository you created on your machine as seen before.
- You can see the remote endpoint it is connected to via `git remote`

```
$ git remote -v
origin https://github.com/dis-data-modeling-2019/assignment-1-neumannm1.git (fetch)
origin https://github.com/dis-data-modeling-2019/assignment-1-neumannm1.git (push)
```

- Receiving updates (from the Server, i.e. GitHub) is called **pull**, sending updates (to the Server) is called **push**.

```
$ git pull origin
# receives new data and integrates into local repository
$ git push origin
# sends local commits to the remote
```

Add Another Remote Repository

- In order to receive updates on the assignment, you need to add the original assignment repository as an additional remote:

```
$ git remote add upstream https://github.com/dis-data-modeli
2019/assignment-1.git
```

„upstream“ is just a named reference

- Then, you can receive updates via **pull**:

```
$ git pull upstream master
[…]
From https://github.com/dis-data-modeling-2019/assignment-
 * [new branch]      master     -> upstream/master
Merge made by the 'recursive' strategy.
 README.md | 13 ++++++++++---
 1 file changed, 10 insertions(+), 3 deletions(-)
```

You „**pull**“ the latest updates from the „**upstream**“ endpoint into your local „**master**“ branch.

- As long as you do not change your local README.md yourself (you shouldn‘t), there will be no conflicts.

Summary: Working with a Remote

git command	Action
`git remote -v`	List all configured remote endpoints
`git remote add <name> <url>`	Add a new remote endpoint identified by a name and with a given URL
`git pull <name> <branch>`	Pull changes from a named remote endpoint into a line of work („branch“)
`git push <name> <branch>`	Push local changes from the branch to a named remote endpoint

- Use **`git pull upstream master`** to integrate changes from the template repository into your work.
- Use **`git push origin master`** to push your changes to your own repository on GitHub!

References

- Git Download, manual and more:
  https://git-scm.com
- Setup:
  https://git-scm.com/book/en/v2/Getting-Started-First-Time-Git-Setup
- Students‘ corner on GitHub:
  https://education.github.com/students