Useful git commands

Lau Møller Andersen

15/9/2021

Cloning the forked version of my repository

Make sure that you are running from a bash terminal (sh)
First, fork my directory, using the GUI, click "Fork" in the upper-right-hand corner (i.e. create a remote copy of your own), then clone it

```
## go to the path where you want to clone YOUR fork
git clone <url_to_your_repository>
```

når du cloner, gemmer du en local file

Pushing your solution to your GitHub repository

Then do the following to push changes to the origin (your fork of my repository)

```
cd <your_local_directory> # the folder you cloned your repository into
git status ## check which files differ
git add <name_of_your_solution_file.Rmd> # add files to what you will later commit
git commit -m "<your message>" # commit all the files you have added above
git push origin main ## push the commited changes to your origin (your remote location)
```

github_methods_3 local folder

tjek hvilke filer der er i mappen

tilføj den fil/filer du vil pushe til dit remote repository (github in cloud/online). tryl "git add ." for at vælge ALLE filer, mega nemt

Push den lokale fil til dit online repository på github (remote location) :))

Adding upstream (my repository) to pull changes from there

You can do this to more easily keep up to date - we'll also go through on Sep 22
See also Figure 1

```
## add another remote spot (my repository)
git remote add upstream https://github.com/ualsbombe/github_methods_3.git
# see which remote channels you have (you should have origin (your repository) and ...
## upstream (my repository)
git remote -v
git pull upstream main ## call this from your local copy to pull my latest changes ...
git push origin main ## ... you can then push these changes to your origin,
## so they match with the upstream
```

pull laus nye ændringer

push laus nye ændringerne til dit origin (github_methods_3)

A schematic of the workflow

Other useful commands

```
git pull origin main ## pull what is on origin (your remote) into your local folder
```
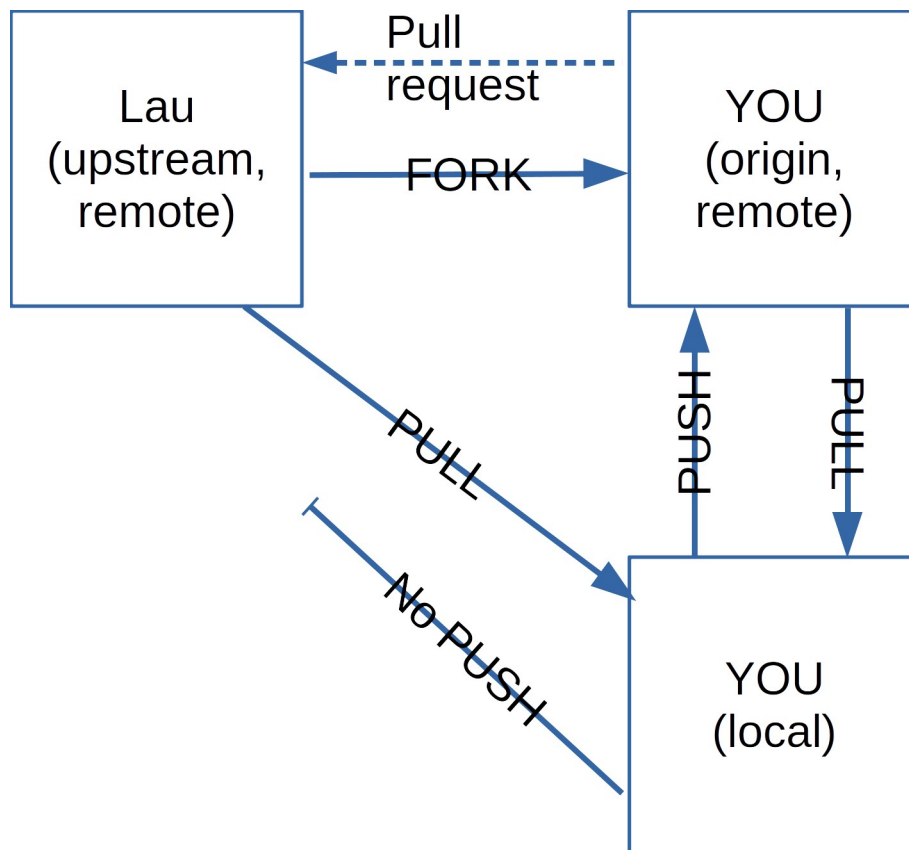
Figure 1: GitHub workflow

Some bash (sh) commands

```
cd <a_folder_name> # change directory to a given folder
mkdir <a_folder_name> # make a directory
pwd # print working directory
ls # list files in directory
cd # go to your home directory
cd .. # go one directory up
cd -- # go back to the directory you were in before
```