Exercise 6

Goals

- Access code from a Git web interface
- Push code changes to a Git web interface
- Examine the code repository on a Git web interface

Initialization

In [ ]:
```
# check current directory with "pwd"
pwd
# go to folder of this exercise using "cd"
```

In [ ]:
```
# execute this code at the very beginning to get access to the helper funct
source ../helpers.sh
init_exercise
```

Optional: clear notebook and restart

In case you mess up your notebook completely,
execute *reset* in the following cell. This will restore a clean environment!

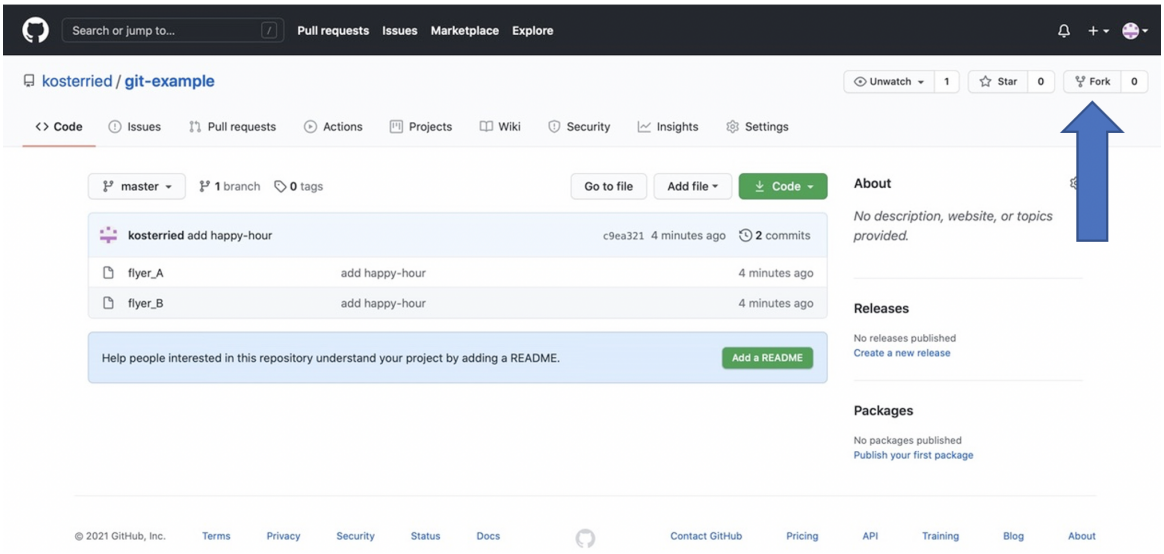
In [ ]:
```
## only execute in case of (serious) trouble ##
## it will delete your entire work-directory ##
reset
```

Exercise

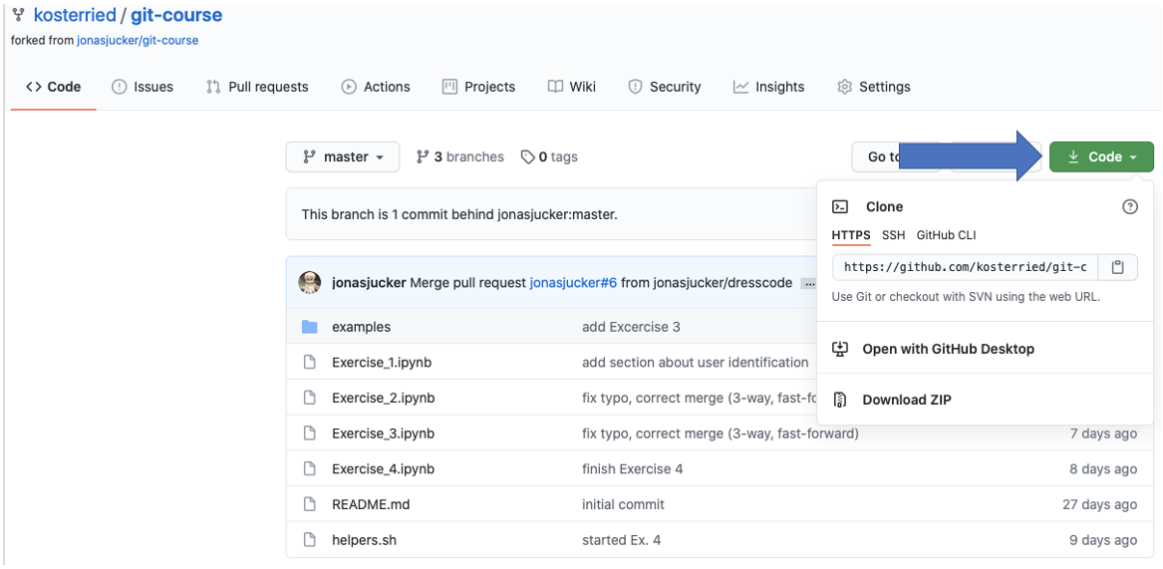
In this exercise we are going to work with a repository we are hosting on the GitHub website.

First, let's make a fork of the repository so that you can have your own copy of a C2SM repository to work with. A fork is a complete copy of a repository into your own account, where you have full permission to make whatever changes you like to your forked repository. To make your fork, open another tab in your browser and navigate here: https://github.com/kosterried/git-example

Use the web interface to make a fork:



You can find the path to your fork on the website:



In [ ]:
```
# use "git clone <path_to_repository>" to download your forked repository
```

In [ ]:
```
# use "cd" to enter the repository
```

Examine the repository

Let's examine the repository. Does it have any remotes? What branches are in it?

In [ ]:
```
# use "git remote -v", "git branch -a", and "git status" to examine the rep
```

You should have seen that your local repository has a remote called "origin", which points to your fork on GitHub. This is the default behavior when you use git clone to copy a repository.

Add to local repository

Next, let's make a new branch and add a commit to it.

In [ ]:
```
# use "git checkout -b <branch_name>" to make a new branch
```

Make a change in your local repository. Remember to do all modifications of the flyers directly via Jupyter Notebooks.

- Go to folder *work* and enter *git-example*
- Open *flyer_A*
- Add more information to your flyer, i.e. music, dresscode, etc.

Don't forget to save your modifications before coming back!

In [ ]:
```
# add and commit your changes
```

Send local information to Github

Now, let's send our new branch to our GitHub fork.

Unfortunately we cannot perform *git push* via jupyter notebooks due to the interactive way of entering username and password.

Please open a terminal and go to the directory you get executing the cell below.

Then run *git push origin <"branch_name">* there.

In [ ]:
```
# go to this folder in the terminal and perform "git push origin <"branch_n
pwd
```

Examine the Github repository

Head back to Github and have a look at your forked repository.

Let's use the web interface to examine the repository. Try the following tasks there:

1. Find the list of commits and examine the files for a specific commit.
2. Modify a file and use the web interface to make a new commit.
3. Make a comparison of your new branch to the master branch.

Update local repository using git fetch

Now, let's get the commit we made on Github into our local repository.

In [ ]:
```
# use "git fetch origin" to download the new commit from your fork
```

In [ ]:
```
# use "git status" to examine your repository
```

Our new commit has been downloaded into a remote branch, but is not available yet in our local branch. Let's use git merge to update our local branch.

In [ ]:
```
# use "git merge <remote_name>/<branch_name>" to sync up your local branch
```

Update local repository using git pull

Let's examine the difference between git fetch and git pull. We just used git fetch to get a commit from our remote repository, and then we used git merge to include it in our local branch.

First, go back to the web interface and use it to make a new commit.

Next, let's get that commit into our repository.

In [ ]:
```
# use "git pull origin" to download the new commit from your fork
```

Have a look at your local branch. You should see that the commit you made has already been put into your local branch, because git pull does both a git fetch AND a git merge automatically.