

Exercise 2

Goals

- Learn how to work with branches and switch between them using `git checkout`

Structure

In this exercise we will work on flyers (flyer_A and flyer_B) for two parties (Party A and Party B). The flyers should contain information about the location, music, door opening and so on. For adding the information to the flyers we work in separate branches to not mix things up.

Again, the exercise consists of short descriptions about a specific Git command, followed by a practical part where you can execute appropriate Git commands.

In order to allow a smooth exercise, there are some functions written by C2SM in the file *helpers.sh* that are **NOT** part of Git. For this exercise we use the following functions from that file:

- **init_exercise:** It will create the *work* directory and navigate into it
- **reset:** It will delete the *work* folder and enable you a clean restart of the exercise in case you completely mess it up
- **init_simple_repo:** setup a Git repository containing a first version of the flyer_A and flyer_B in branch *master*.

Reminder: all text enclosed with `<>` denotes a placeholder to be replaced by a specific string appropriate in your context.

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 initialize the exercise properly
        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

Learn how to work with branches and switch between them using git checkout

In the output above we see two flyers:

- flyer_A
- flyer_B

Let's first have a look at them using the *cat* command:

```
In [ ]: # this line will setup a simple Git repository for you  
init_simple_repo
```

```
In [ ]: # display content with cat: "cat <your_flyer>"
```

As you see, there is not much information there yet!

More information for our guests is needed. To keep everything in order we do this in two different Git branches: one each for flyer_A and flyer_B.

So let's start!

```
In [ ]: # create a new branch for flyer_A  
# use 'git checkout -b "<meaningful_branch_name A>"' to create a new branch
```

From now on, we do all modifications of the flyers directly via Jupyter Notebooks.

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

Do not forget to save your modifications before coming back!

After saving, we run `git status` to see if Git tracked our changes.

```
In [ ]: # see if git tracked our changes
```

The output should look as follows:

```
On branch party_A  
Changes not staged for commit:  
  (use "git add <file>..." to update what will be committed)
```

(use "git restore <file>..." to discard changes in working directory)

modified: flyer_A

no changes added to commit (use "git add" and/or "git commit -a")

Do a commit with these changes...

```
In [ ]: # add flyer_A and commit it
```

For the planning of the other party we want to use another Git branch. To not lose track of all the different branches, Git provides the

`git branch` to see all branches of a repository. The * indicates our current branch.

```
In [ ]: # see all branches of our Git repository
```

We can switch easily between these branches using the `git checkout` function. Don't worry -> Git will keep all your work done in this branch.

```
In [ ]: # go back to branch master using "git checkout master"
```

Create a new branch for flyer_B and extend the flyer_B in that branch similar as already done for flyer_A

```
In [ ]: # create a new branch and edit flyer_B
```

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

```
In [ ]: # see again all branches of our Git repository
```

The output should look similar to:

```
master
party_A
* party_B
```

Said different, our git repository now contains:

- branch `<meaningful_branch_name A>` with modifications of flyer_A
- branch `<meaningful_branch_name B>` with modifications of flyer_B
- branch `master` with the initial version of flyer_A and flyer_B

With `git checkout` it is so easy to jump between these branches and modify our flyers further. To show its capabilities we quickly switch between the branches and see how our flyers change.

To output the content of a text file to the terminal you can use `cat <your_flyer>`.

```
In [ ]: # display the content of flyer_A and flyer_B using "cat"
```

```
In [ ]: # switch to branch master and do the same again
```

```
In [ ]: # switch to the branch not visited yet and display the content of the flyer
```