

The exercises in this course will teach you how to implement important RL algorithms and how every part of the RL pipeline works. The goal of this first exercise is to set up teams and learn about git and the workflow for future exercises.

1. Form teams of up to 3 students

Most exercises will require you to implement some of the techniques you learn during the course (in python 3.6)¹. *Git* is one of the most widely used **version control systems** and allows you to easily collaborate with others on code from the same repository.

Exercises have to be handed in *teams of up to 3 students*. When you have found your partner, open the following link <https://classroom.github.com/a/-2sa0inH>, create a group (you will have to name the group yourself) and both join that group. This will allow you to clone the template repository in which you can add your solutions to this exercise sheet.

Note 1: If you have never worked with *git* before we suggest you take a look at this simple guide <http://rogerdudler.github.io/git-guide/>.

Note 2: Make sure you and your team-mate are happy with each other. GitHub Classroom does not allow to change your groups mid semester.

2. Get familiar with git and GitHub Classroom

To show that you are familiar with the standard git *add*, *commit*, *push* steps add a file called `members.txt` to your repository. The file should contain the names of all members in the following way:

```
member 1: name1
member 2: name2
member 3: name3
```

Afterwards you can push to submit.

We make use of GitHub Classrooms autograde functionality. Essentially, for most exercise sheets we will require you to pass unit tests which are automatically evaluated whenever you push to GitHub. To demonstrate this process, for this exercise we run a test that expects the above file to be present and to contain three lines as above (make sure to replace name1, name2 and name3. If your group has less than 3 students, just add any name you like).

You will be informed if the tests executed successfully or not. (To run tests locally you can always use the provided Makefile via `make all`)

¹We recommend you use anaconda to set up a virtual environment for the lecture, see <https://conda.io/projects/conda/en/latest/user-guide/getting-started.html#managing-environments>