This is a pdf version of https://git.tu-berlin.de/lis-public/ai-student-workspace/-/blob/main/00/README.md

0th Assignment: Test your setup

This assignment only contains a simple test task. You should use it to set up your workflow. Make sure everything works before the graded assignments will start next week. The current assignment will not be graded.

Ex. 0.1: Install git

`git` is a software used in virtually every software project to track changes and versions of code across large collaborations. We will use it in this course to provide coding assignments to you, and to get the solutions back from you.

To install `git`, please follow the instructions here: https://git-scm.com/book/en/v2/Getting-Started-Installing-Git.

Ex. 0.2: Fork this repo and clone it from your own git repository

1. Click the 'fork' button on top right of the gitlab webpage of the course repo at https://git.tu-berlin.de/lis-public/ai-student-workspace/ to create your own fork on this gitlab.
2. Open a bash shell and navigate to a folder of your choice.
3. Clone the repo using `git clone YOURFORK`. Here, `YOURFORK` is the URL of your fork of this repo. For example, if you forked directly on this gitlab, it will look like this: `git@git.tu-berlin.de:YOUR_USER_NAME/PATH/TO/REPO.git`

```
git clone YOURFORK
```

4. `cd` into the repo folder that appeared: `cd ai-student-workspace`

Ex. 0.3 Give the course tutors reading access to your repo

If you are using the TU gitlab, please add the following accounts:

- ischubert
- driessdy
- levit

The role should be at least `Reporter` for these accounts.

Ex. 0.4 Tell us where your repo is located

Fill out the questionnaire on ISIS. The questionnaire will ask you for the following:

1. Your student ID number (Matrikelnummer)
2. Your E-Mail address
3. The SSH-URL of your repo (that's the one that starts with `git@...`), i.e. `YOURFORK`.

You can form groups of up to 3 students. If you are in the same group, you can use the same URL. However, each of you should still fill out the questionnaire individually. Furthermore, please ensure that you are able to present your group's solution during the tutorials.

Ex. 0.5: Open assignment

Throughout this course, each assignment will be given as a subfolder of `ai-student-workspace`. For example, the assignment subfolder for the present assignment is `00`. Inside each assignment subfolder, you will find

1. Files containing code that you **should not edit**
2. One `README.md` that explains the task
3. One single file that you should edit according to the task. For the present assignment, this file is called `solution_00.py`. For later assignments, these files will be called `solution_01.py`, `solution_02.py`, and so on. **This is the only file that should be edited**.

Ex. 0.6: Complete assignment

Inside `solution_00.py`, there will be functions / code that you should change so that they generate the desired output.

As an example, please modify the function body of the function `y = is_even_and_positive(x)` so that

- `y` is True if `x` is an even and positive integer
- `y` is `False` in all other cases

Do not change the function name!

Of course there are many ways to solve this, but one would be to use the functions provided in `module_1.py` like this:

```python
from module_1 import is_positive, is_even
```

Ex. 0.7: Stage, commit, and push your solution

Once you are finished, *stage* your changes using

```
cd ai-student-workspace/00
git add solution_00.py
```

Then *commit* your changes using

```
git commit
```

This will save your changes to `solution_00.py`. You will be asked to provide a commit message.

Push your new commit to your forked repository using

```
git push
```

After the assignment deadline, we will automatically test whether `is_even` returns correct values.

Ex. 0.8: Merge your fork of this repo after the next assignment has been published

The next assignment will be added to this repo, as a new subfolder. In order to bring your own fork up-to-date with these changes, you need to merge.

For this, first add the URL of this repository as upstream to your fork (you only need to do this once):

```
git remote add upstream git@git.tu-berlin.de:lis-public/ai-student-workspace.git
```

By typing `git remote -v`, you can verify that there are 2 remote urls added to your fork now: The url for "origin" contains the branches of your fork, "upstream" contains the branches of this repo.

Every time you want to merge your repo with new updates on the upstream, run

```
git fetch upstream
git merge upstream/main origin/main
```

`fetch` checks for new commits found at the upstream url, and `merge` merges the upstream branch (called `upstream/main`) with your branch (called `origin/main`).

If you only committed changes in `solution_00.py`, you will be able to merge automatically (without conflicts).