



F. Wermelinger
Office: Pierce 211

Pair-Programming 1

Setup your Git repository, tmate, docker, edit text

Issued: September 2, 2022 Due: September 16, 2022 11:59pm
--

In this pair-programming session you setup your class Git repository and discover a set of tools used for the pair-programming sessions.

Due to the ongoing sectioning process, there will be 6 zoom sessions (one around noon and one in the evening) on

- Tuesday, September 6th
- Wednesday, September 7th
- Thursday, September 8th

The zoom sessions are located at

- **Noon (12:30-1:45pm):**
<https://harvard.zoom.us/j/92821477585?pwd=V3NQbXpXYkt2bEVWMS9oL0FNZyZBVUT09>
- **Evening (5:00-6:15pm):**
<https://harvard.zoom.us/j/93701361785?pwd=bnNLUEJKc0ZueURzOVFPVXM4V3NiUT09>

These sessions are intended to provide help for you if something is unclear or does not work. You may also work through this exercise independently without consulting these zoom sessions.

Exercise 1: Setup private Git repository

Deliverables: none

The purpose of this exercise is to help you ease into the course flow for the semester. You will learn where to find the homework assignments and how to correctly setup your Git repo for this course.

Solve Problem 2 in the HW1 exercise sheet <https://code.harvard.edu/CS107/main/tree/master/homework/hw1/hw1.pdf>.

Exercise 2: tmate and docker

Deliverables: none

By the end of this exercise, you will be ready to use tmate for pair-programming.

1. Please read through the protocol section in the pair-programming tutorial on the class website.¹
2. Work through the steps in the tmate section of that tutorial.²

Windows users may need to install the Windows subsystem for Linux. Please see this document for additional instructions: https://harvard-iacs.github.io/2022-CS107/pages/media/linux_subsystem.pdf. Please ask your TF for further assistance if you encounter problems.

Exercise 3: Edit text in the command line

Deliverables:

1. [lab/pp1/exercise_3.md](#)

All work on pair-programming exercises should be committed to your main or master branch. If you have started to work on HW1 and you are on the hw1 branch, please *commit* the work on this branch before checking out main or master to continue with this exercise. If you are not ready to commit, you can also use the `git stash` command to temporarily stash your changes on the hw1 branch and apply them again later when you return to work on HW1. The “Example Homework Workflow”³ section in the tutorials on the course webpage provides further help. Please ask your TF for help if you have any issues.

Get together with your left and right neighbors and work through the following items in a tmate session.⁴ By the end of this exercise, you should be familiar with the pair-programming workflow and have gained some first experience with text editing in the command line.

- a) Start a tmate session inside your pp1 directory⁵ and invite your team members by sharing the link to your session, either ssh or the web link.
- b) Create the file `exercise_3.md` using either of these editors:
 - nano (easy)
 - pico (easy)

¹<https://harvard-iacs.github.io/2022-CS107/pages/tutorials.html#tutorial-pp-protocol>

²<https://harvard-iacs.github.io/2022-CS107/pages/tutorials.html#tutorial-pp-tmte>

³<https://harvard-iacs.github.io/2022-CS107/pages/tutorials.html#tutorial-hw-example>

⁴You can choose to share your code with your partners or not for this session. In the future, you will be required to share code. Make sure you get familiar with tmate and the text editing exercises below.

⁵<https://harvard-iacs.github.io/2022-CS107/pages/tutorials.html#tutorial-pp-tmte>

- vim (powerful editor, advanced)
- nvim (neovim: same as vim but code has been rewritten from scratch)
- emacs (powerful editor, advanced)
- ne (easy, has a TUI, <https://ne.di.unimi.it>)
- tilde (easy, has a TUI, <https://os.ghalkes.nl/tilde>)

All these editors are available in the docker image. You can put anything you want inside your file. Play around with some markdown, see <https://www.markdownguide.org/basic-syntax/> for the basic syntax. For example:

```
# Pair-programming 1
```

```
Hello [`tmate`](https://tmate.io/)
```

- c) Inside the terminal, from the command line, play around with some of the Linux commands you started to learn in the lecture. At the very least, try to use `ls` and some of its options.

Now commit the file you have just created. Note that the docker image contains the `git` command but does not have valid ssh keys. If you commit inside the container you have to always use a token which is not convenient. It is recommended that you use Git on your laptop directly, not from within the container. When you are done, you can push your contribution with `git push` and verify that everything is fine by checking your repository online at <https://code.harvard.edu/CS107>.

Comments

- The tasks above are not necessarily connected. The main goal is to help you get familiar with the correct workflow.
- You can open the editor by passing a filename as an argument. For example, `nano file.md` would open nano with an empty `file.md`. You can then edit the file to your liking and save the contents by pressing `Ctrl-O` and hitting enter to confirm that you want to save. You can then exit nano by pressing `Ctrl-X`.
- vim is a modal editor which means you can not start typing text right away. If you play with vim, these are the steps:
 1. Open a new file: `vim file.md`.
 2. Press `i`, this will enter INSERT mode.
 3. You can now enter text as you like>
 4. To exit INSERT mode and enter NORMAL mode, press the ESC key.
 5. To save your work and exit vim at the same time, press `:wq` followed by enter. This will write (w) and then quit (q). You must be in NORMAL mode for this command.