

# Python

## Special topic: Git

Prof. Dr. Thomas Kopinski

May 31, 2023

### Abstract

In this special exercise you will get an introduction to Git as well as solve some exercises to gather some first-hand experience.

## Task 1: Read the manual

- Please read the following chapters from the "Pro Git" book available [here](https://git-scm.com/book/en/v2) (<https://git-scm.com/book/en/v2>) in order to get a better understanding of Git. This task is more about getting the general principles behind Git and what it offers than about memorizing the exact commands needed to fulfill each of the tasks. The more hands-on part of this exercise will start from Task 2 onwards.

The book is also available in multiple other languages (including German), so feel free to switch to another language if you have problems understanding the content. You should still try English first as most of the information you will encounter later on in your career will be in English anyway, so it makes sense to get comfortable with the language early on.

- [1.1 What is version control](#)
- [1.3 What is Git?](#)
- [2.1 Getting a Git repository](#)
- [2.2 Recording changes](#) (up to section "Viewing Your Staged and Unstaged Changes")
- [2.5 Working with remotes](#) (most relevant: "Adding", "Fetching and Pulling", "Pushing")
- [3.1 Git-branching](#)
- [3.2 Branching and merging](#) (up to section "Basic Merge Conflicts")

## Task 2: Interactive tutorial

- Now that you have read the most important basic topics regarding Git it is time to test out some of the knowledge you acquired. Go to [this interactive tutorial](https://learngitbranching.js.org/) (<https://learngitbranching.js.org/>) and work on the first 8 exercises given under the "Main" tab. You can always opt to do more, but these should teach you the basic concepts you need to know in order to use Git for your own local repositories.
- Try to solve the first 6 tasks under the "Remote" tab in the level selector. Again, feel free to do more if you want to dive in deeper, but these should suffice for most of the basic remote workflows you will encounter.

## Task 3: Further tutorial

- Work through [this tutorial](https://www.freecodecamp.org/news/what-is-git-and-how-to-use-it-c341b049ae61/) (<https://www.freecodecamp.org/news/what-is-git-and-how-to-use-it-c341b049ae61/>) and try to follow along with your previously gathered knowledge. As this is the tutorial given in "Exercise 1" most of you might have already gone through it, but it should be useful to do it again now with your newly gained insights cause it will make it easier to follow along and exactly know what and why things are happening.

## Task 4: Use Git in your own projects

- If you have finished the tasks above you should now be familiar with most of the basic concepts and commands needed to use Git. You should really use Git from now on in your own projects in order to further deepen your understanding and reap the benefits of version control tools. There are no drawbacks when using it, so do yourself a favor and make use of Git from the beginning to get into the workflow.
- [Here](https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet) (<https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>) you can find a cheat sheet with multiple useful Git commands (do not worry about all the commands you do not know or have not encountered (yet)).
- While it is useful (or even necessary) to know your way around Git in the command shell, you can make your life a lot more comfortable by working with an IDE that integrates Git support or allows for the installation of extensions (e.g. Visual Studio Code (VSC)). As I personally work with VSC I can recommend the "GitLens" extension which allows you to easily do all the basic Git operations while offering a lot more powerful features and insights when and if needed as can be seen in fig. 1.

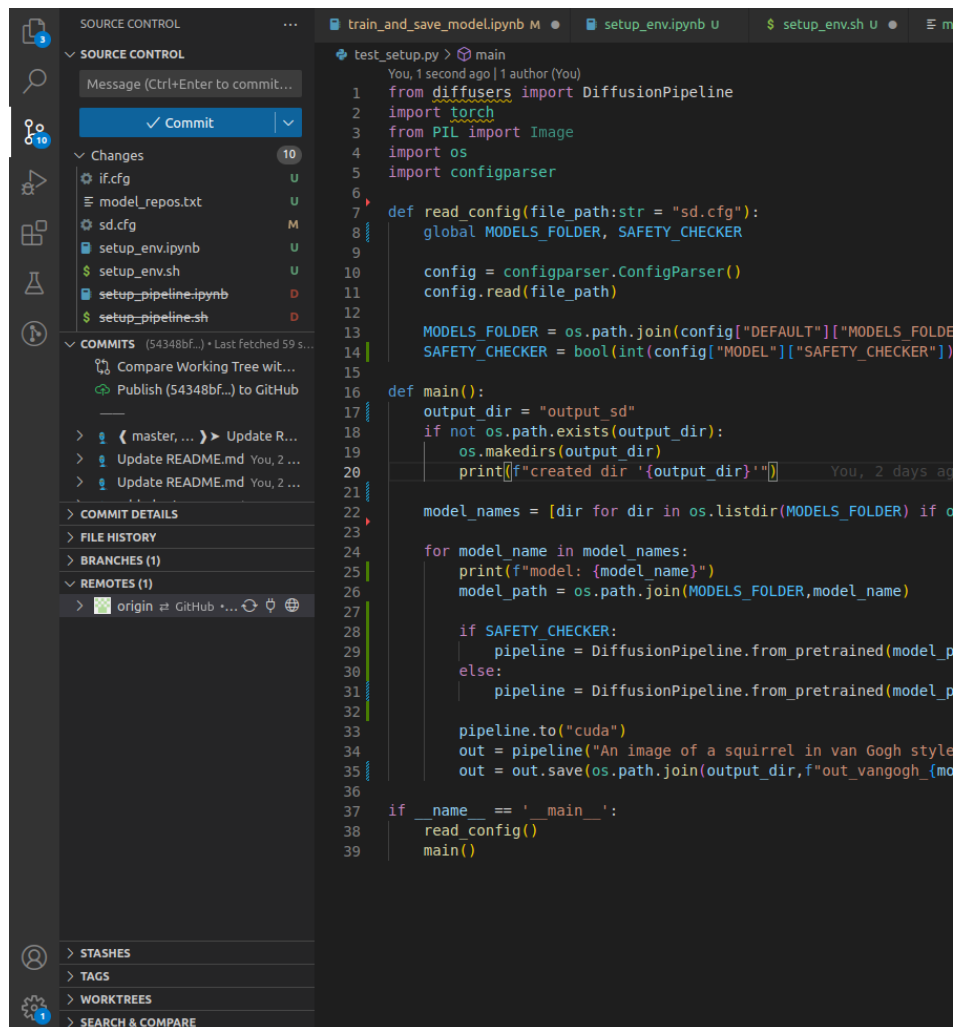


Figure 1: Example of GitLens extension in VS Code. You can easily keep track of your local (and remote) commits on the left side, get color-coded information about the file status in the current stage ("Source control", top-left) and commit directly from the IDE. Additionally it shows you meta-information for every line of code (when, by whom it got written/updated, commit message attached to it).