

# Dive in to Git and GitLab

## Installation of Git

Course for “CompiCampus”

2022-08-17

Roman Plessl



---

# Install Git locally

# If necessary: Install Git locally

(including check if you have already git installed)

macOS (mostly preinstalled, check if `git status` works in your terminal, otherwise):

- a. <https://git-scm.com/download/mac>
- b. I recommend to choose the Homebrew way: Install [homebrew](#) and then `brew install git`

Windows (sometimes preinstalled, check if `git status` works when you open “cmd.exe”):

- Install <https://git-scm.com/download/win>
- or use WSL 2 <https://docs.microsoft.com/en-us/windows/wsl/install-win10>
- or use Windows Shell Integration <https://tortoisegit.org/>
- For SSH and SSH Agent Help follow: <https://cutt.ly/dhEnIH5> and <https://cutt.ly/KhEnQDL>
- I recommend to use the “Git Bash” to open a bash compatible command line shell

Linux:

- Debian / Ubuntu: `sudo apt install git-all`
- Fedora / RHEL : `sudo dnf install git-all`

# Depending on your wished / needed Editor

Here: Install Visual Studio Code and Change Editor for Git

Visual Studio Code has Git commands built-in.

- Usable on all platforms (mac, win, linux)
- Install Visual Studio Code from <https://code.visualstudio.com/>

Git needs an Editor to submit Changes, normally - if not set differently - this is `vi`!

By default, Git uses whatever you've set as your default text editor via one of the shell environment variables `VISUAL` or `EDITOR`, or else falls back to the `vi` editor to create and edit your commit and tag messages. To change that default to something else, you can use the `core.editor` setting:

```
$ git config --global core.editor "code --wait"
```

Now, no matter what is set as your default shell editor, Git will fire up Code to edit messages.

<https://git-scm.com/book/en/v2/Appendix-C%3A-Git-Commands-Setup-and-Config>