

exercise-1 Introduction to `git`

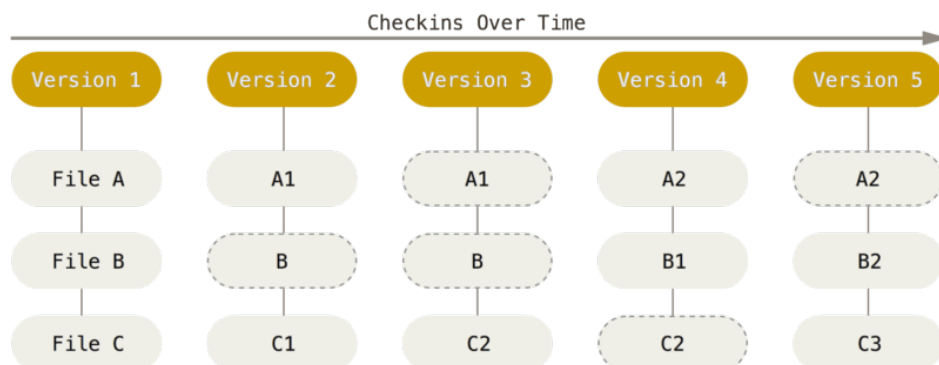
- What is `git`? How to use it in general?
- What is `.md`?
- How to use it specifically for this lecture? How to ask questions via issues?
- Exercise: Create a `.md` file and make a `pull request`

Class

- Language
- OS

What is `git`?

- `git` ist eine Versionierungs-Technologie, welche sich als Standard in der Software-Entwicklung, der open source-Welt und insbesondere der Forschung etabliert hat. Damit Sie diese Technologie kennenlernen, werden wir Sie für den Übungsbetrieb einsetzen.
- Die Grundlage von `git` ist das *repository*, welches alle Daten und die Geschichte Ihres Projekts speichert.
- `git` gibt Ihnen die Möglichkeit, zurück in der Geschichte Ihres Codes zu gehen, in dem Sie an beliebigen Punkten einen Schnappschuss Ihres Projekts speichern (= `commit`).



- `git` macht Ihnen die Zusammenarbeit innerhalb einer Organisation einfach, in dem Sie Ihr repository zentral speichern,

und jeder eine lokale Kopie des repositorys hat, welches sie oder er mittels `push` und `pull` aktualisieren kann. Gibt es Konflikte, versucht `git` diese selber zu lösen, und fragt Sie sonst explizit um Hilfe.

- `git` macht aber auch die Zusammenarbeit über Organisationsgrenzen hinweg einfach mittels `fork` und `pull request`. Dies werden wir in der Übung benutzen.
- `git` ist in diesem Sinne ein Konzept, mit Daten umzugehen, und ein Programm, das Ihnen dabei Hilft. Es gibt Ihnen aber keinen Speicher, und keine Werkzeuge, um Ihre Daten zu verändern. Es gibt aber Plattformen, welche dies mittels `git` tun, die bekannteste davon ist `github.com`. Dort können Sie mittels `git` Ihre Daten speichern, und verteilt an Ihrem Projekt arbeiten.
- `github.com` wartet mit zahlreichen Zusatzdiensten auf, die die Arbeit deutlich erleichtern, mitunter einem Browser-Editor und rendering von `.md`-Dateien.

`.md`

- `.md` steht für Markdown, und ist ein kleines Set von sowohl maschinen- wie auch menschenlesbaren Zeichenketten, mit denen Sie angeben können, wie Sie Ihren Text dargestellt haben möchten.
- Es erfreut sich, auch dank `github.com`, aber vorallem aufgrund seiner Einfachheit, großer Beliebtheit bei Software-Projekten für die Dokumentation.
- Wenn Sie die [Basic Syntax](#) können, sind Sie bereits dabei.

exercise-1

1. Create an account on github.com if you don't have one already. You can use your personal, your university or a [burner](#) email.
2. Find the [course repository](#), and `fork` it. It might also make sense to `watch` it, if you want to receive notification if something changes. However, this is not strictly necessary, since the individual exercises will not change while you work on them.
3. In your forked repository, which should be under `yourname/uibk-python`, locate the file `exercise-1/exercise-1-correctme.md` and edit it in the browser. There are some `.md` errors in this file, try to find them and save your changes in a first `commit`.
4. Now, suggest your changes be added to the course repository by

creating a `pull request`. This will trigger an action, in this case, a `.md` linter. Most likely, your first `pull request` will fail the linting, because the linter is quite strict. In this case, cancel the `pull request`. Check out the error messages, and fix the errors, `commit` and create a new `pull request` until you get a green checkmark.

⚠️ The title of your `pull request` must begin with your last name, so I can give you credit for the exercise. For example, you could use **Colic: Fixed error MD077** as a title. In a real-world `pull request` you wouldn't include your name, but here in the course setting have no other means of mapping your github account name to your real name.

💡 As an optional exercise, you can create a new file in `students/yourname.md`, in which you share as much or as little about yourself as you feel comfortable with sharing, and maybe include a picture of your favorite food or drink for practise. The information doesn't even have to be true; but if you also include some information about your programming experience, I can better adapt the course. If you chose to do this, create a new `pull request` that you prefix with **Bonus:** in the title, so that I can differentiate it from the other `pull requests`.

Further Reading

- Read Chapters 1 and 2 of the [Git Pro Book](#). You don't have to set up `git` on your machine yet if you don't feel comfortable with the command line, but it's a good preparation for the next exercise.