

Software Versionskontrolle mit GIT

Geschichte von GIT

Git wurde ab Sommer 2005 von Linus Thorwalds (ebenso Urheber des Linux Kernels) zu entwickeln begonnen. Dabei wurde versucht, Schwachstellen bekannter, vorhandener Versionskontrollwerkzeuge zu vermeiden.

In einer Umfrage von Stack Overflow aus 2022 ging hervor, daß git für etwa 94% aller befragten Entwickler die am Liebsten verwendete Versionsverwaltung darstellt.

Es gibt zahllose Anbieter online: https://en.wikipedia.org/wiki/Comparison_of_source-code-hosting_facilities. Einer von diesen ist github.com, hier werden die Unterrichts-Repos gehostet.

Was ist ein Repository

Ein GIT **Repository** (kurz: Repo) ist ein Speicherort für ein Software-Projekt. Es enthält alle Dateien und den Verlauf aller Änderungen, die an diesen Dateien vorgenommen wurden. Git-Repositories ermöglichen es, zu früheren Versionen des Projekts zurückzukehren, Änderungen zu verfolgen und zu überprüfen, wer welche Änderungen vorgenommen hat. Sie sind ein wesentliches Werkzeug für die Zusammenarbeit und Versionskontrolle in der modernen Softwareentwicklung.

Gebrauch von GIT

Offizielles Manual

<https://git-scm.com/docs/user-manual.html> Unter dem Dropdown **Topics** findet man die offizielle Dokumentation zu den einzelnen Kommandos.

Kurzanleitung (Cheat-Sheet von github)

<https://education.github.com/git-cheat-sheet-education.pdf>

FAQ

Ich kann nicht committen (user.email, user.name)

Hier ist in einem Terminal Fenster im Hauptordner des Projektes das Folgende einzugeben:

- `git config user.name "Sieglinde Musterfrau"`
- `git config user.email "smf@spengergasse.at"`

Man kann auch den Schalter `--global` dazugeben, dann wird dies auf Ihrem Rechner zur default Config Ihrer Identität für git commits.

Wie bekomme ich Hilfe zu den einzelnen Git Subkommdos?

```
$ git help <command>
```

Bsp: `git help branch`.

Die gleichen Inhalte sind auch (s.o.) unter [Topics](#) der offiziellen Dokumentation zu finden.

Wie mache ich alle lokalen Änderungen rückgängig

Achtung dies löscht alle Änderungen welche nicht committed wurden!. Der Befehl hierzu lautet `git reset`. Folgende Abfolge entfernt alle lokalen Änderungen und setzt Ihr Verzeichnis auf den Zustand des letzten Commits:

```
$ git reset --hard
$ git clean -df
```

Alternativ kann man natürlich auch den ganzen geklonten Ordner löschen und danach neu klonen.

Was machen wenn alles klemmt

Es gibt vielleicht Änderungen, die es jetzt nur auf Ihrer Festplatte gibt, aber git macht nur kryptische Fehlermeldungen, dann wie folgt vorgehen:

1. Die "Schätze" aufheben an einem Ort (A).
2. Den Ordner mit der kaputten Working-Copy *löschen*.
3. Mit einem frischen Clone aus dem zentralen Repository neu erstellen.
4. Auf den gewünschten Branch wechseln.
5. Die gesicherten Dateien von (A) auf den entsprechenden Branch mergen (händisch, ohne git Befehle).
6. Einen neuen Commit und Push durchführen.
7. Alles Gut.

Dokumentversion

Aktuelle Version wird [hier](#) gepflegt.