

L^AT_EX-Kurs: 05 1 GitHub

Philipp Arras

11. Oktober 2014

Git ist ein Versionsverwaltungsprogramm.

1. `sudo apt-get install git`
2. Account bei github erstellen
3. `mkdir repos` (am besten im Homeverzeichnis) In diesen Ordner kommen alle deine Git-Projekte rein
4. `cd repos`
5. Erstelle auf github ein neues Repository und lade das (leere) Repository via `git clone https://github.com/<Benutzername>/<Name des Repositories>` herunter.
6. Setze Benutzernamen und Emailadresse mit `git config --global user.name "<name>"` und `git config --global user.email "<email address>"`
7. Falls du ein Repository eines anderen GitHub Useres geclost hast, musst du ihm eine Mail schreiben, damit er euch die Berechtigung gibt auf dem Projekt Änderungen zu machen.
8. `git status` Das ist ein sehr nützlicher Befehl. Den solltest du zwischen den nächsten Schritten immer ausführen um zu sehen, was passiert
9. Editiere eine Datei, z.B. diese, und speichere sie
10. `git add <Datei>`
Mit diesem Befehl wird die neue Version der Datei gestaged. Das ist ein Zwischenschritt, der für Anfänger überflüssig erscheint, später aber sehr hilfreich sein kann. Alternativ kann man ins Hauptverzeichnis wechseln (`cd /repos/learnLatex`) und `git add -A` ausführen. Dann werden alle neuen und veränderten Dateien gestaged
11. `git commit -m "Hier muss eine Nachricht hin, die beschreibt, was geändert wurde."`
Mit diesem Befehl werden alle gestageten Dateien commitet, d.h. der „Geschichtsschreibung“ hinzugefügt. Ab jetzt hast du die Möglichkeit später genau diesen Zustand deines ganzen Repositories wiederherzustellen
12. `git push origin master`
Hiermit lädst du alle Änderungen zu GitHub hoch. Es ist nicht notwendig nach jedem Commit ein Push auszuführen. Wenn du mehrere Commits hintereinander machst, weil du z.B. gerade unterwegs keine Internetverbindung hast, lädt der push-Befehl alle Commits automatisch auf einmal hoch.
13. `git pull origin master` Hiermit lädst du alle Änderungen von GitHub herunter. Das sollte man also immer ausführen, bevor man anfängt zu arbeiten.

Außerdem wichtig ist:

- Dateien kopieren, verschieben, löschen immer mit `git cp`, `git mv`, `git rm`. Sonst bekommt git das ja nicht mit.
- Es könnte ja theoretisch sein (wird bei uns wahrscheinlich nicht passieren), dass wir beide die gleiche Datei bearbeiten. Angenommen ich pushe zuerst und dann du. Dann wirst du eine Fehlermeldung bekommen und müsstest einen merge ausführen. Das ist nicht so einfach. Damit das Ganze keine Probleme macht, sollte man zuerst `git fetch` und danach `git merge` ausführen. `git pull` macht das beides zusammen und ist deshalb evtl. unübersichtlich. Das mergen ist eine Wissenschaft für sich und kann im Internet gelernt werden.
- Es gibt viele GUIs für git. Diese eignen sich sehr gut um sich die History anzeigen zu lassen (`git log` ist sehr unübersichtlich aber nicht gut um die oben genannten Schritte auszuführen. Dann weiß man nicht, was unter der Oberfläche passiert. Also lieber aus der Shell heraus bedienen.

Git ist zuerst ein bisschen kompliziert zu verstehen, bringt dann aber riesige Vorteile mit sich.