

# Git Cheat Sheet

Maximilian Mayer, Daniel Hartl, Sabine Pfeiffer

November 8, 2025

## Häufige Befehle

- **git fetch:** Holt neue Änderungen vom Remote-Repository (in unserem Fall GitHub) und aktualisiert lokale Referenzen
- **git stash:** "Legt" ungespeicherte Änderungen zur Seite, damit man ohne zu committen den Branch wechseln kann.
- **git stash pop:** Mit diesem Befehl holt man die Änderungen, die im vorherigen Befehl zur Seite gelegt wurden, wieder zurück
- **git clone:** Ein gesamtes Remote-Repository wird auf den Rechner kopiert.
- **git pull:** Holt Änderungen vom Remote und merged sie in den aktuellen Branch. Es entspricht den Befehlen `git fetch` und `git merge`
- **git push:** Die lokalen Commits werden ins Remote-Repository geschickt.
- **git commit:** Die Änderungen werden dauerhaft in der Historie gespeichert. Mit `git commit -m "Nachricht"` kann eine kurze Beschreibung der jeweiligen Änderungen hinzugefügt werden.
- **git status:** Zeigt den aktuellen Zustand des Repos an und gibt an, welche Dateien geändert wurden oder welche gestaged bzw untracked sind.
- **git switch:** Wechselt zwischen verschiedenen Branches und kann neue Branches erstellen. Mit dem Befehl `git switch main` wechselt man in den Branch `main`, während man mit dem Befehl `git switch -c new-branch` den neuen Branch `new-branch` erstellt.
- **git branch:** Listet alle lokalen Branches auf und markiert den aktuell genutzten Branch mit einem Sternchen. Mit `git branch new-branch` erstellt man den neuen Branch `new-branch`, `git branch -d branchname` löscht den Branch `branchname`, `git branch -m alter-branchname neuer-branchname` benennt den Branch `alter-branchname` um in `neuer-branchname`.
- **git add:** Fügt Änderungen zur Staging-Area hinzu. `git add datei.txt` fügt die Datei `datei.txt` hinzu, `git add .` hingegen fügt alle Änderungen hinzu.