

Commit & Branch Manipulation

- `git commit --amend --no-edit`
Commit an den letzten commit "anhängen" ohne neue message
- `git reset --soft HEAD~1`
Macht den letzten Commit rückgängig, behält die Änderungen *gestaged*.
- `git reset --mixed HEAD~1`
Macht den letzten Commit rückgängig, behält Änderungen *unstaged*.
- `git reset --hard HEAD~1`
Letzten Commit rückgängig machen und Änderungen verwerfen.
Vorsicht: destruktiv, letzter commit und working dir sind futsch!!!
- `git reset --hard HEAD`
Alle Änderungen im working directory verwerfen und branch auf den Stand von HEAD setzen.
Vorsicht: man sollte wissen wo HEAD steht (normalerweise am letzten commit).
- `git cherry-pick <commit>`
Einzelnen Commit aus anderem Branch übernehmen.
- `git switch -`
Wechselt zum vorherigen Branch zurück (super für schnelles Hin- und Herspringen).

Historie & Diff

- `git log --oneline --graph --all`
Kompakter, visualisierter Verlauf aller Branches.
- `git diff --staged`
Zeigt Änderungen, die für den nächsten Commit vorgemerkt sind.
- `git diff <branchA>..<branchB>`
Unterschiede zwischen zwei Branches zeigen.
- `git blame <file>`
Zeigt, wer welche Zeile zuletzt geändert hat. Wichtigster Git Befehl um Unruhe im Team zu stiften
=D

Cleanup & Fehlerkorrektur

- `git stash`
Änderungen vorübergehend speichern (z. B. vor `git pull`).
- `git stash pop`
Gestachte Änderungen wieder anwenden.

Remote-Handling

- `git fetch --all --prune`
Alle Änderungen holen & gelöschte Remote-Branches lokal entfernen.