

Git Befehle

Git-Befehle für Alltag und Notfall – durchsuchbar und mit einem Klick kopierbar. Mit Warnhinweisen bei Befehlen, die Arbeit unwiederbringlich verwerfen.

daniele.michelin //
cheat sheet

Alltag: Status & Commit

Kurzstatus mit Branch und Tracking-Info

```
git status -sb
```

Ungestagte Änderungen anzeigen

```
git diff
```

Gestagte Änderungen anzeigen

```
git diff --staged
```

Änderungen stückweise zum Commit vormerken

```
git add -p
```

Commit mit Nachricht erstellen

```
git commit -m "Fix login redirect"
```

! Letzten Commit um gestagte Änderungen ergänzen

```
git commit --amend --no-edit
```

Neuen Branch anlegen und wechseln

```
git switch -c feature/login
```

Log & Suche

Letzte 20 Commits kompakt als Graph

```
git log --oneline --graph --decorate -20
```

Commits einer Person in den letzten 2 Wochen

```
git log --since='2 weeks ago' --author='Daniele'
```

Commits finden, die einen Text hinzufügten oder entfernten

```
git log -S 'password' --oneline
```

Wer hat Zeilen 10–30 zuletzt geändert?

```
git blame -L 10,30 file.py
```

Stash

Änderungen mit Beschreibung zwischenlagern

```
git stash push -m "wip login"
```

Auch unversionierte Dateien zwischenlagern

```
git stash push -u
```

Stash anwenden und behalten

```
git stash apply stash@{0}
```

Neuesten Stash anwenden und entfernen

```
git stash pop
```

! Einen Stash verwerfen

```
git stash drop stash@{0}
```

Rebase & Verlauf aufräumen

Aktuellen Branch auf main aufsetzen

```
git rebase main
```

! Letzte 4 Commits interaktiv ordnen, squashen, umformulieren

```
git rebase -i HEAD~4
```

Rebase abbrechen, Zustand wie vorher

```
git rebase --abort
```

Branch mit Merge-Commit zusammenführen

```
git merge --no-ff feature/login
```

Rückgängig machen

! Ungestagte Änderungen einer Datei verwerfen

```
git restore file.txt
```

Datei aus dem Staging nehmen, Änderungen bleiben

```
git restore --staged file.txt
```

Commit durch einen Gegen-Commit rückgängig machen (sicher für geteilte Branches)

```
git revert abc1234
```

Letzten Commit lösen, Änderungen bleiben gestagt

```
git reset --soft HEAD~1
```

! Letzten Commit samt Änderungen verwerfen

```
git reset --hard HEAD~1
```

! Unversionierte Dateien und Ordner endgültig löschen

```
git clean -fd
```

Reflog & Rettung

Verlauf aller HEAD-Bewegungen anzeigen

```
git reflog
```

! Auf einen früheren Reflog-Stand zurücksetzen

```
git reset --hard HEAD@{2}
```

Verlorenen Commit per Hash als Branch retten

```
git switch -c rescue abc1234
```

Verwaiste Objekte aufspüren

```
git fsck --lost-found
```

! Letzten Reset, Merge oder Rebase rückgängig machen

```
git reset --hard ORIG_HEAD
```

Remotes & Bisect

Alle Remotes holen, gelöschte Branches aufräumen

```
git fetch --all --prune
```

Holen und lokale Commits oben aufsetzen

```
git pull --rebase
```

! Force-Push, der fremde neue Commits nicht überschreibt

```
git push --force-with-lease
```

Binärsuche nach dem fehlerhaften Commit starten

```
git bisect start && git bisect bad && git bisect good v1.2
```

Bisect automatisch mit einem Testskript durchlaufen

```
git bisect run ./test.sh
```

Bisect beenden, zurück zum Ausgangs-Branch

```
git bisect reset
```