

# Spickzettel: Git Reflog – Änderungen rückgängig machen

## 1. Was ist `git reflog`?

- `git reflog` speichert alle lokalen Änderungen an Branches und HEAD-Positionen.
- Hilft beim Wiederherstellen gelöschter Commits oder Branches.
- Funktioniert nur **lokal** – Remote-Commits sind nicht enthalten.

## 2. Reflog-Einträge anzeigen

```
git reflog
```

- Listet alle Änderungen mit Index (HEAD@{0}, HEAD@{1}, ...).
- Zeigt die letzten Commits, Branch-Wechsel und Resets an.

## 3. Gelöschten Commit wiederherstellen

Falls ein Commit versehentlich gelöscht wurde:

```
git checkout -b wiederhergestellt HEAD@{2}
```

- Erstellt einen neuen Branch ab dem gewünschten Reflog-Commit.

## 4. Gelöschten Branch wiederherstellen

Falls ein Branch gelöscht wurde, aber noch im Reflog existiert:

```
git checkout -b <branchname> <commit-hash>
```

- `<commit-hash>` aus `git reflog` entnehmen.

## 5. Zu einem früheren Zustand zurückkehren

Falls ein fehlerhafter Merge oder Reset durchgeführt wurde:

```
git reset --hard HEAD@{3}
```

- Bringt das Repository exakt auf den vorherigen Zustand zurück.

## 6. Spezifische Datei aus einer früheren Version wiederherstellen

Falls eine Datei aus einem alten Commit benötigt wird:

```
git checkout HEAD@{5} -- datei.txt
```

## Best Practices

- `git reflog` regelmäßig prüfen, um unerwartete Änderungen nachzuvollziehen.
- **Nie voreilig `git reset --hard` verwenden** – immer zuerst mit `git reflog` absichern.
- Falls ein Merge oder Rebase fehlschlägt, hilft `git reflog`, um den vorherigen Zustand wiederherzustellen.
- **Nicht zu lange warten!** Reflog-Einträge können nach einiger Zeit aus der Historie entfernt werden.