

# Spickzettel: Commits, Rücknahme & Reset

## 1. Commit erstellen

```
git add <datei>          # Datei für Commit vormerken
git commit -m "Message"  # Commit mit Beschreibung erstellen
```

## 2. Letzten Commit bearbeiten (z. B. Nachricht ändern)

```
git commit --amend -m "Neue Nachricht" # Commit-Nachricht ändern
```

## 3. Datei nachträglich zu letztem Commit hinzufügen

```
git add <datei>
git commit --amend --no-edit # Letzten Commit mit neuer Datei
aktualisieren
```

## 4. Letzten Commit zurücknehmen (Änderungen behalten)

```
git reset --soft HEAD~1 # Letzten Commit entfernen, Änderungen bleiben
```

## 5. Letzten Commit verwerfen (Änderungen gehen verloren!)

```
git reset --hard HEAD~1 # Letzten Commit löschen und Änderungen verwerfen
```

## 6. Änderungen an einer Datei zurücksetzen

```
git checkout -- <datei> # Datei auf letzte Commit-Version zurücksetzen
```

## 7. Änderungen aus Staging-Bereich entfernen

```
git reset HEAD <datei> # Datei aus Staging entfernen, bleibt aber erhalten
```

## 8. Commit in der Historie überschreiben (Vorsicht!)

```
git rebase -i HEAD~3 # Interaktive Rebase der letzten 3 Commits
```

## Häufige Fragen

- **Was tun, wenn ich eine Datei versehentlich gelöscht habe?**
  - o `git checkout -- <datei>` stellt sie aus dem letzten Commit wieder her.
- **Kann ich einen alten Commit nachträglich bearbeiten?**
  - o Ja, mit `git rebase -i` können Commits bearbeitet oder zusammengeführt werden.

- **Was ist der Unterschied zwischen `git reset --soft` und `git reset --hard`?**
  - o `--soft` behält Änderungen, `--hard` verwirft sie komplett.