

Spickzettel: Git Branching (Grundlagen)

1. Was sind Branches?

- Branches ermöglichen parallele Entwicklungen.
- Der main-Branch ist der Standard-Entwicklungszweig.
- Feature-Branches für neue Funktionen oder Bugfixes nutzen.

2. Branch erstellen & wechseln

```
git branch neuer-branch    # Neuen Branch erstellen
git checkout neuer-branch  # Zu neuem Branch wechseln
# Kurzform:
git checkout -b neuer-branch
```

3. Bestehende Branches anzeigen

```
git branch                # Lokale Branches anzeigen
git branch -r             # Remote-Branches anzeigen
git branch -a             # Alle Branches (lokal & remote)
```

4. Änderungen mergen

```
git checkout main         # Zurück zum Hauptbranch
git merge neuer-branch    # Feature-Branch in main übernehmen
```

Merge-Typen & Unterschiede

- **Fast-Forward Merge:**
 - Direktes Einfügen des Branches, wenn keine neuen Commits auf main existieren.
- `git merge --ff neuer-branch`
- **Merge mit Commit:**
 - Erstellt expliziten Merge-Commit, selbst wenn Fast-Forward möglich wäre.
- `git merge --no-ff neuer-branch`
- **Squash Merge:**
 - Vereinigt alle Commits aus dem Branch zu einem einzigen Commit.
- `git merge --squash neuer-branch`
- **Rebase statt Merge:**
 - Ändert die Commit-Historie, indem die Commits des Feature-Branches neu auf main aufgesetzt werden.
- `git rebase main`

5. Branch löschen

```
git branch -d neuer-branch # Gelöschter Branch muss gemergt sein
git branch -D neuer-branch # Erzwingen, auch wenn nicht gemergt
```

6. Letzte Branch-Aktionen mit `reflog` anzeigen

`git reflog`

Zeigt letzte Aktionen im Repository

- Falls ein Branch gelöscht wurde, kann man ihn oft über `git reflog` wiederfinden.
- Wiederherstellung eines versehentlich gelöschten Branches:

`git checkout -b alter-branch <commit-hash>`

Häufige Fragen

- **Was passiert, wenn ich einen Branch lösche?**
 - o Falls der Branch nicht gemergt wurde, gehen die Änderungen verloren!
- **Kann ich auf einen gelöschten Branch zurückgreifen?**
 - o Falls der Branch nicht gepusht wurde, kannst du mit `git reflog` nachsehen, ob die Commits noch vorhanden sind.
- **Warum sollte ich mit Feature-Branche arbeiten?**
 - o Saubere, isolierte Entwicklung, weniger Konflikte, einfacheres Testen.