

Spickzettel: Git Tagging & Workflow

1. Was sind Tags?

- Tags markieren spezielle Commits (z. B. Versionen, Meilensteine).
- Zwei Arten: **annotierte Tags** (mit Metadaten) & **leichte Tags** (nur Marker).

2. Tag erstellen

```
git tag v1.0 # Leichtes Tag erstellen
git tag -a v1.0 -m "Version 1.0" # Annotiertes Tag mit Nachricht erstellen
```

3. Tags anzeigen

```
git tag # Alle Tags auflisten
git show v1.0 # Details zu einem Tag anzeigen
```

4. Tag zu Remote pushen

```
git push origin v1.0 # Einzelnes Tag pushen
git push --tags # Alle lokalen Tags pushen
```

5. Tag löschen

```
git tag -d v1.0 # Tag lokal löschen
git push origin --delete v1.0 # Tag im Remote-Repo löschen
```

6. Vorhandenen Commit nachträglich taggen

```
git tag -a v1.0 <commit-hash> -m "Version 1.0"
```

7. Auf ein Tag wechseln (Checkout)

```
git checkout v1.0 # Wechselt zu einem Tag (Detached HEAD!)
```

8. Tag in Branch umwandeln

```
git checkout -b hotfix-branch v1.0 # Neues Branch basierend auf Tag
```

9. Nutzung im Workflow

- **Versionierung:**
 - Jede stabile Release-Version mit einem Tag versehen (v1.0, v1.1, ...).
- **Hotfixes:**
 - Von einem Tag aus einen Hotfix-Branch erstellen (git checkout -b hotfix v1.0).
- **Deployment:**

- o CI/CD-Pipelines können auf spezifische Tags reagieren (`git push --tags`).
- **Backup von Ständen:**
 - o Ein wichtiger Commit kann jederzeit mit einem Tag versehen werden.

Best Practices

- Immer annotierte Tags (`-a`) nutzen, um Metadaten zu speichern.
- Tags für jede stabile Version setzen.
- Tags für automatisierte Releases in CI/CD nutzen.
- Tags nicht leichtfertig löschen, da sie oft als Referenz für Builds dienen.