

## Tags als Lesezeichen

Man kann sich den Zustand es Repos merken als „Tag“, das ist wie ein Lesezeichen - man kann sich gewissermaßen vorstellen, dass man mit `git tag <tag-name>` das Repo „zwischenspeichert“. (Ein Tag ist eine benannte Referenz auf einen bestimmten Commit).

```
git tag -l // zeigt tags
git tag v1 // legt das tag v1 an
git tag -a v1.4 -m "my version 1.4" // legt einen Tag mit Kommentar an
```

## Zwischen Commits wechseln

Mit dem Befehl `git checkout <commitid>` kann man einen beliebigen Snapshot ins Arbeitsverzeichnis „auschecken“, z.B. so:



Wenn man jetzt allerdings das log betrachtet, sieht man ein Problem:



Man sieht im log nur die Commits, die relativ zum aktuellen Snapshot in der Vergangenheit liegen! Wie soll man nun zurück zum letzten Commit gelangen?

Mit `git rev-list --all --remotes` kann man sich alle Commits anzeigen lassen, mit ein paar Optionen siehts noch netter aus:

```
git rev-list --all --remotes --pretty=format:'%Cred%h%Creset -
%C(yellow)%d%Creset %s %Cgreen(%cr) %C(bold blue)<%an>%Creset' --abbrev-
commit
```



### (A1)

Lege dir einen Alias `ac` an, der alle Commits hübsch ausgibt. Springe dann zwischen deinen Commits hin und her und verfolge die Änderungen nach. Finde heraus, was bei solchen Sprüngen mit nicht versionierten Dateien im Arbeitsverzeichnis geschieht.

## Verwenden von Tags zum Springen

Aktuellen Zustand merken:

```
git tag now
```

Zielcommit raussuchen:

```
git log
```

Zielcommit auschecken:

```
git checkout <commit-id>
```

Umsehen - ist es das was man sich merken will? wenn ja: Tag anlegen mit `git tag v0`.

Nun kann man zwischen den Tags hin und her wechseln, wie man möchte. Am besten ist es natürlich, bereits beim Programmieren zu entsprechenden Zeitpunkten Tags anzulegen, die mit entsprechenden Kommentaren versehen sind.

Wenn man einen Tag nicht mehr benötigt, kann man ihn mit dem Befehl `git tag -d <tagname>` löschen. Der Commit bleibt dabei erhalten, es wird nur die Referenz und ggf. der Kommentar entfernt.



**(A1)**

Gehe in deiner Webseite in der Zeit „hin und her“, indem du Tags verwendest.

From:  
<https://wiki.qg-moessingen.de/> - QG Wiki

Permanent link:  
<https://wiki.qg-moessingen.de/faecher:informatik:oberstufe:git:workflow:start?rev=1619639187>

Last update: **28.04.2021 21:46**

