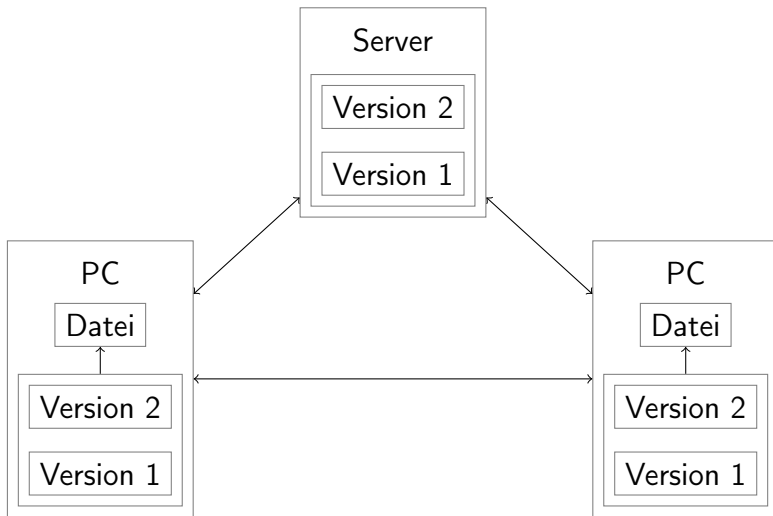


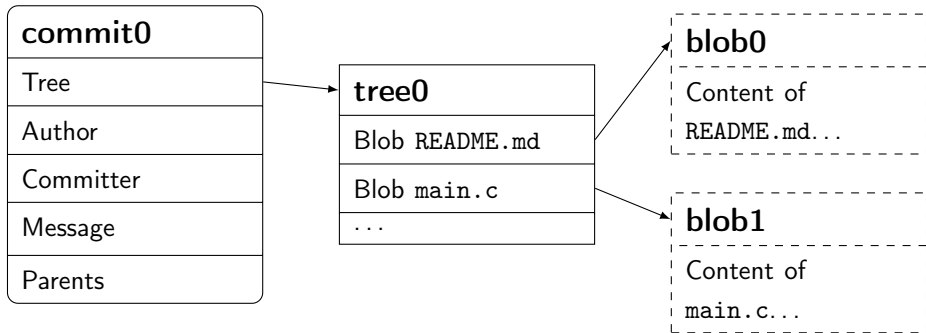
Versionsverwaltung mit Git

- ▶ Git ist ein verteiltes Versionsverwaltungssystem
- ▶ Git ist kein Backupsystem
- ▶ Konflikte werden nicht automatisch gelöst

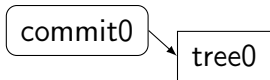
Versionsverwaltung mit Git



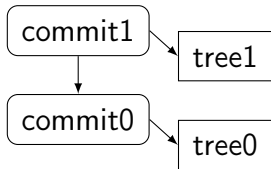
Was ist ein commit?



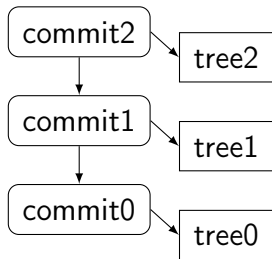
Was ist ein branch?



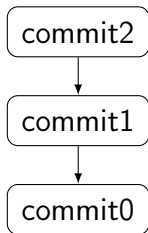
Was ist ein branch?



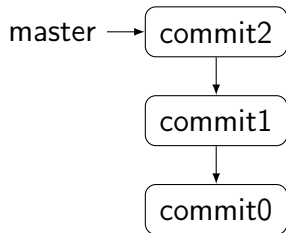
Was ist ein branch?



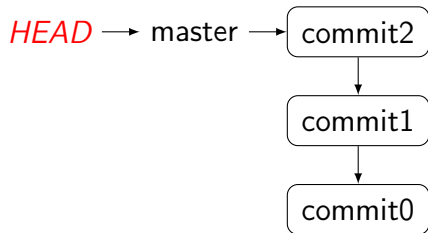
Was ist ein branch?



Was ist ein branch?

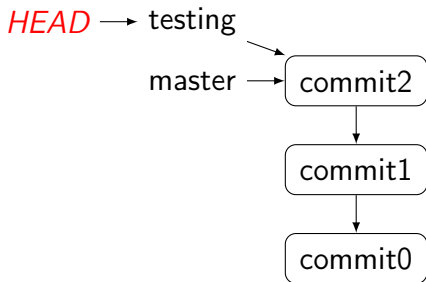


Was ist ein branch?



On branch *master*

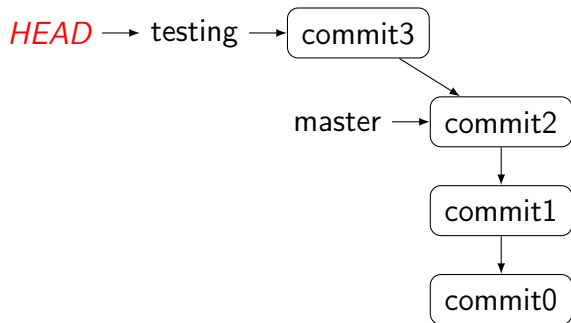
Was ist ein branch?



`git checkout -b testing`

On branch *testing*

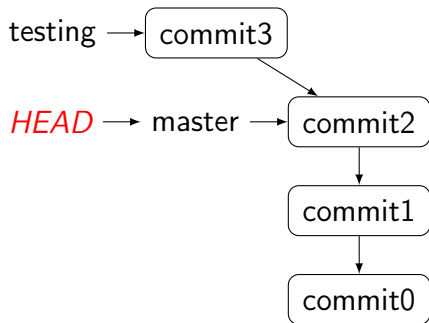
Was ist ein branch?



`git commit`

On branch *testing*

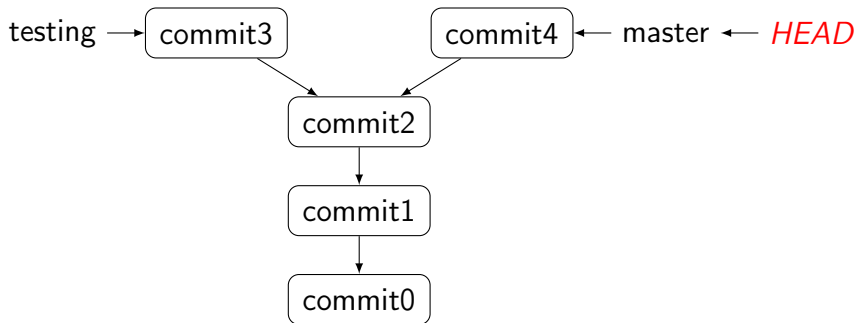
Was ist ein branch?



`git checkout master`

On branch *master*

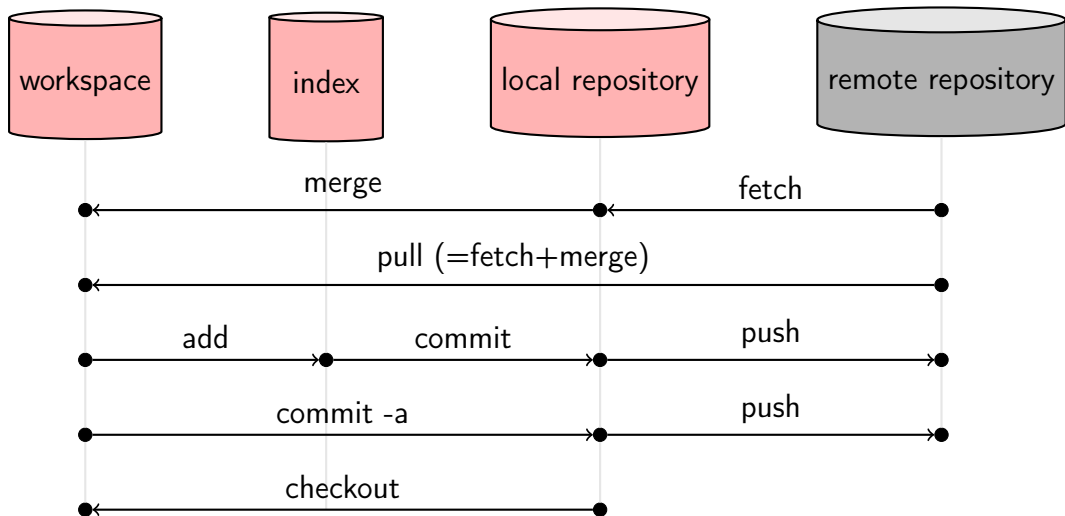
Was ist ein branch?



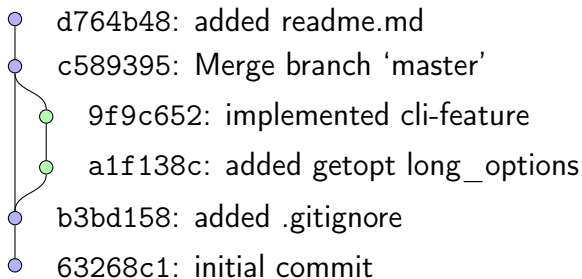
git commit

On branch *master*

Git (Remote) Repositories



Branching



- ▶ Arbeiten auf mehreren Zweigen
- ▶ Nützlich bei Implementierung von Features
- ▶ Masterbranch sollte immer eine funktionierende Version enthalten
- ▶ Konflikte möglich!

.gitignore

- ▶ Ignorieren von bestimmten Dateien
- ▶ Beispielsweise build-Dateien, temporäre Dateien, ...
- ▶ Nützlich falls man das ganze Projekt-Verzeichnis zum Index zufügen will
- ▶ auch mehrere .gitignore-Dateien möglich

```
# executable  
my_program
```

```
# Object files  
*.o
```


best practice

- ▶ Vor Beginn der Arbeit `git pull` ausführen
- ▶ Branches regelmäßig mergen / nicht zu lange offen lassen
- ▶ Branch überprüfen: `git branch -l`
- ▶ In kurzen Abständen committen und pushen
- ▶ Immer eine funktionierende Version im master-branch
- ▶ Nur funktionierenden Code pushen
- ▶ Aussagekräftige commit-messages schreiben

Hilfe zu Git + Übersicht

- ▶ `git add`: Zufügen von Dateien und Verzeichnissen zum Index (staging area)
- ▶ `git commit`: Erzeugt eine neue Version im lokalen Repository
- ▶ `git push`: Änderungen im lokalen Repository werden auf den Server geladen
- ▶ `git pull`: Änderungen vom Server holen + merge des aktuellen branches
 - ▶ (= `git fetch` + `git merge`)
- ▶ `git checkout`: zwischen Branches wechseln
- ▶ `git branch`: u.A. Erzeugen, Auflisten, Löschen von Branches

Quiz: Welche Aussagen treffen zu?



`merge` führt zwei branches auf dem remote-Repository zusammen



`commit` erzeugt eine neue Version im remote-Repository



`fetch` bringt das lokale Repository auf den Stand des remote-Repositorys