

# Software Engineering I (IB)

## Git

Prof. Dr. Oliver Braun

Fakultät für Informatik und Mathematik  
Hochschule München

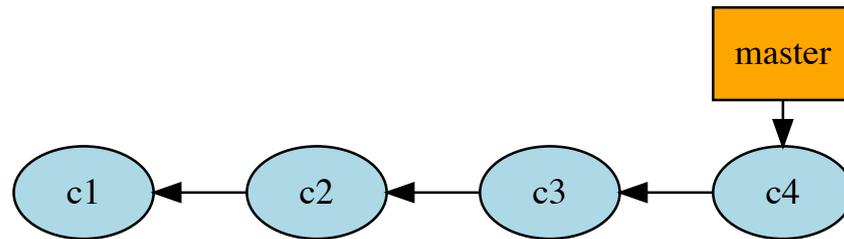
Letzte Änderung: 11.10.2018 08:02

### Inhaltsverzeichnis

Was wir schon wissen...	1
Ein zusätzlicher Branch	2
Zusammenführen (mergen)	3
Merge-Konflikte	5
GitHub-Flow und Git Flow	5
und der ganze Rest...	5

### Was wir schon wissen...

- Git verwaltet Versionen
  - dabei basiert eine neue Version immer auf der Version davor
  - d.h. jeder Commit hat einen Parent Commit
- Beispiel:
  - Commits c1, c2, c3, c4

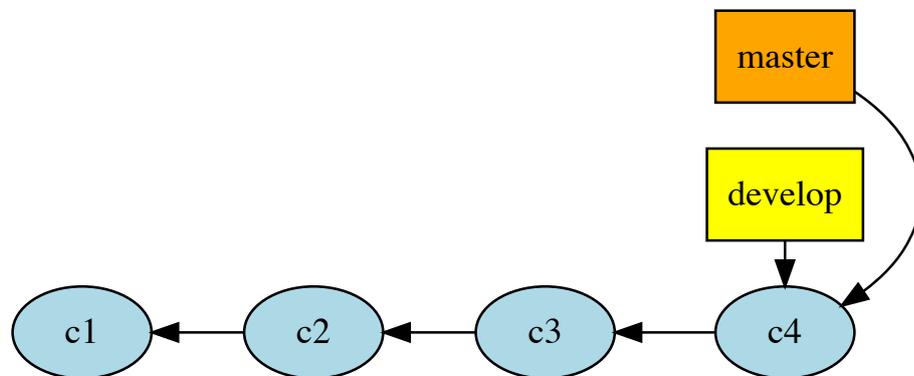


\* der master-Branch ist ein Zeiger auf den letzten Commit

## Ein zusätzlicher Branch

- durch Erzeugen eines Branches, wird ein weiterer Zeiger auf den aktuell ausgecheckten Commit erzeugt
- Beispiel

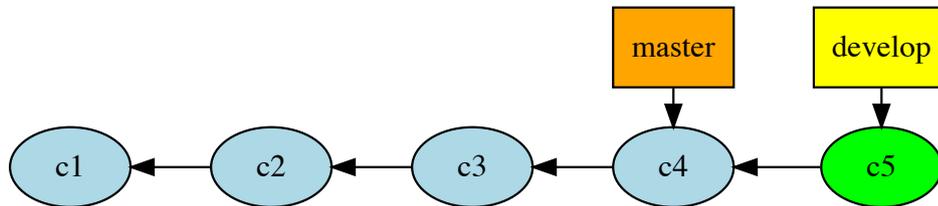
```
git branch develop
```



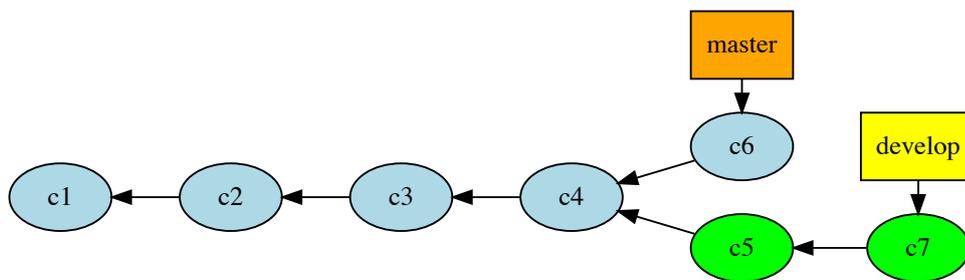
- mit `git checkout` kann dann in den anderen Branch gewechselt werden
- HEAD zeigt an, was aktuell ausgecheckt ist
- Beispiel

```
git checkout develop
```

- nun kann man in dem `develop`-Branch arbeiten und committen
- dadurch verzweigen sich die Branches
- Beispiel: Commit `c5` im `develop`-Branch

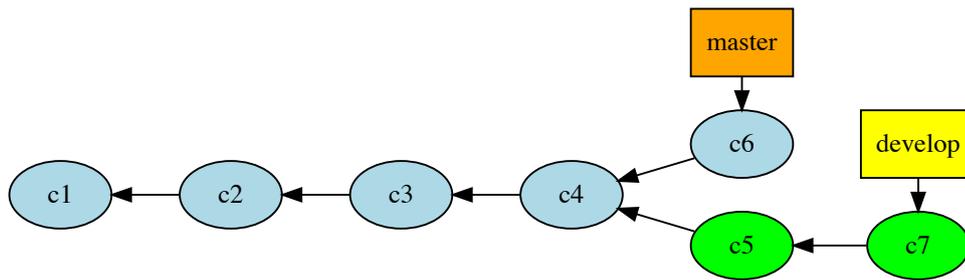


- Commit `c6` im `master`-Branch und `c7` im `develop`-Branch



## Zusammenführen (mergen)

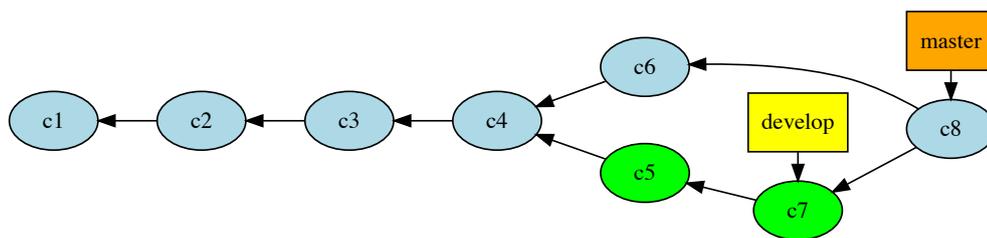
- um die Änderungen aus dem `develop`-Branch in den `master`-Branch zu bringen, müssen die beiden gemerged werden



- c4 ist die gemeinsame Basis
- c6 ist der Stand **in** den gemerged wird
- c7 ist der Stand der gemerged wird

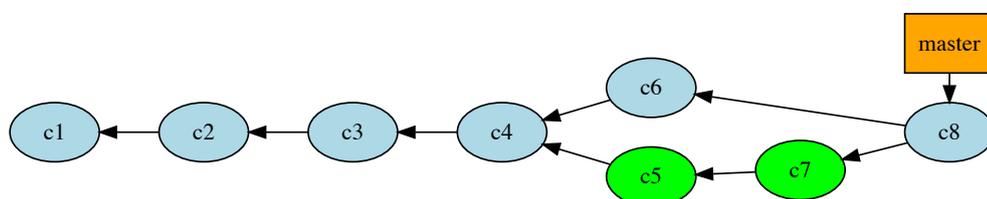
- dazu wechseln in den **master** (zu dem Commit **in** den gemerged wird) und mergen

```
git checkout master
git merge develop
```



- anschließend wird der **develop**-Branch nicht mehr benötigt und kann gelöscht werden

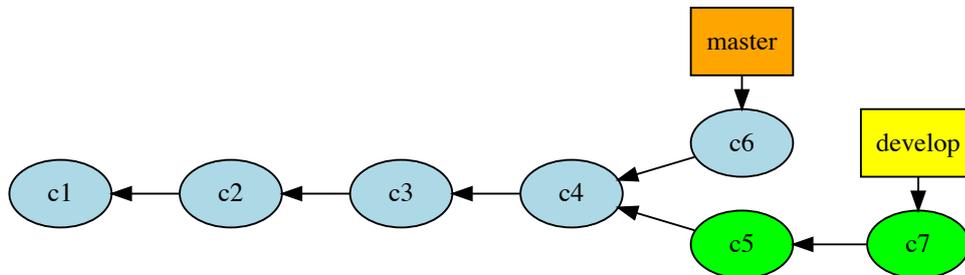
```
git branch -d develop
```



- damit verschwindet nur der *Zeiger* **develop**, nicht aber die Commits

## Merge-Konflikte

- wenn in zwei Branches in der selben Zeile der selben Datei etwas verändert worden ist, kann (will) Git nicht entscheiden, was übernommen wird ⇒ **Merge-Konflikt**
- Git erzeugt keinen Mergecommit, sondern bleibt im Konflikt stehen



- Beispiel

Auto-merging README.md

CONFLICT (content): Merge conflict in README.md

Automatic merge failed; fix conflicts and then commit the result.

- Inhalt der Datei README.md

```
«««« HEAD
Hello World!
=====
Hallo Welt
»»»» develop
```

- Konflikt selbst auflösen (durch Zeilenlöschen) und committen

## GitHub-Flow und Git Flow

- arbeiten mit Branches und Pull-Requests zur Abgabe bzw. für Feedback
- beschrieben unter <https://ob.cs.hm.edu/exercises.html>
- auf Blatt 0 können Sie das einüben

### und der ganze Rest...

- git kann noch viel mehr
- z.B. ist ein Tag eine Art Label für einen bestimmten Commit

- beschäftigen Sie sich damit, wenn Ihnen etwas fehlt oder wenn etwas nicht funktioniert