

Don't be a Git



#hackundsoehne

www.hackundsoehne.de

Terminologie um git

Commits

Branches

Remotes

Repositories



#hackundsoehne

www.hackundsoehne.de



Commits

- Besteht aus
 - Name (z.B. 716cc6be5ba4c7c5112750427281e448457cfe1a)
 - Beschreibung
 - Autor
 - Datum
 - Vorhergehender Commit
 - ...



Branches

- Bestehend aus:
 - Verkettung von Commits
 - Namen



Aufbau eines Repositories



Remotes

- Online Repository
- Austauschplattform von Zuständen
- *origin* Standard Remote name
- *git remote add* <name> <URL>
- *git pull* <name>



Historie auf einem Remote

- Definiert durch eine feste geordnete Reihenfolge von Commits
- Diese sollte nicht gebrochen werden
- Kann erweitert werden durch Anfügen von neuen Commits



Praxis time!



#hackundsoehne

www.hackundsoehne.de



Der Start

- *git init* - Initialisierung des Repositories
- *git clone <url>*- Download eines Repositories von einem Remote



Unser Repository

<https://github.com/hackundsoehne/object-service.git>



#hackundsoehne

www.hackundsoehne.de

git branch <name>

Erstellt einen neuen Branch



#hackundsoehne

www.hackundsoehne.de



git checkout <name>

Wechsel in Zustand des Branches <name>

Remote Branches werden automatisch beim checkout
lokal erstellt
(z.B. *fix-docker-config*)



Commit it!

- Eine Änderung machen (z.B. in der README.md)
- Möglicherweise mit *git add* neue Dateien hinzufügen
- `git commit -a`
 - Öffnet einen Editor, dort die Commit Nachricht eingeben



Nun wieder zusammenführen

Merge oder Rebase



#hackundsoehne

www.hackundsoehne.de

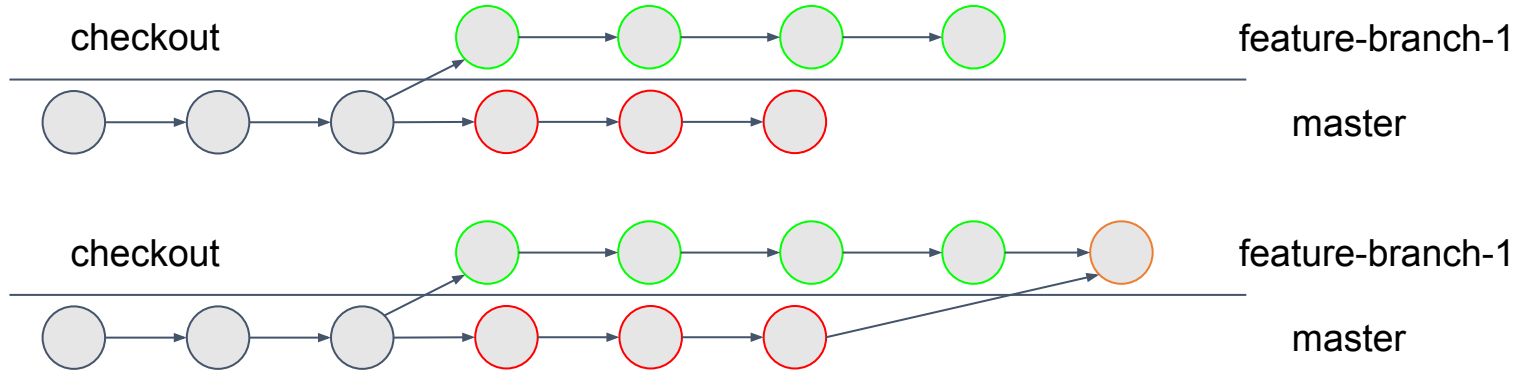


git merge

- Zueinander Führen von Branches
- *git merge B* (während wir auf dem master Branch sind)
 - Ermittlung aller Commits welche nur in B enthalten sind
 - Erneute Anwendung der Commits auf den master
 - Fast forward
 - *Squash* all der Commits in den zum merge Commit
 - No fast forward
- “merge commit” erstellt
 - enthält alle Änderungen von B



git merge master

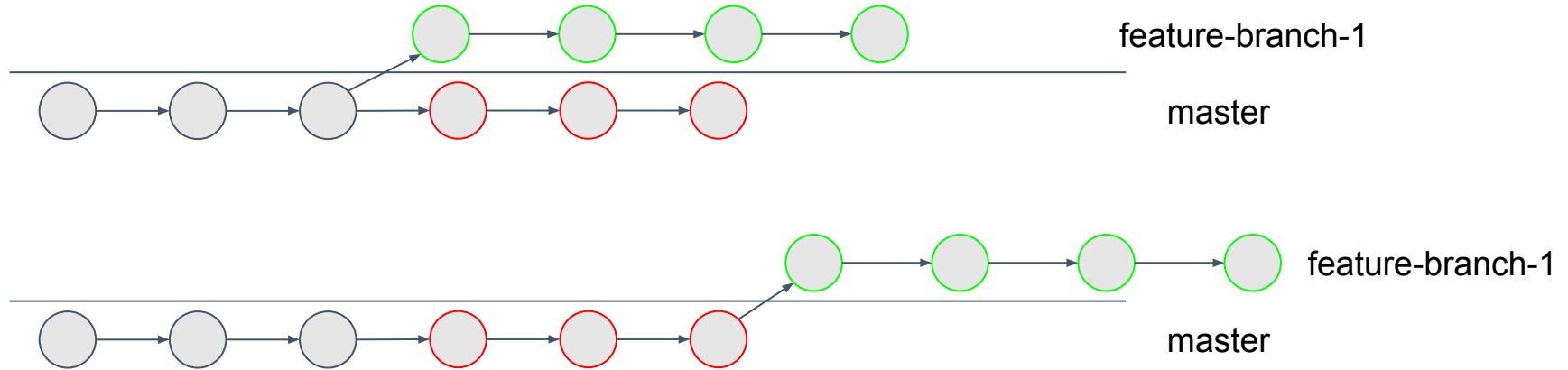


git rebase <name>

- Erkenne alle Commits die auf dem aktuellen Branch sind aber nicht im checkout Branch
- Wende diese Commits erneut auf den <name> Branch an
- Vorteile
 - Lineare Historie
- Nachteil
 - Historie muss u.U. neu geschrieben werden



git rebase <name> (2)



Ausprobieren!

Branch *advanced-documentation* in master mergen!

```
git checkout master
```

```
git merge <der-name>
```



#hackundsoehne

www.hackundsoehne.de



Give me updates!

- *git pull* - Neue Commits werden vom Remote auf den lokalen Branch gelegt
- Verwendete Strategien
 - rebase
 - merge



Who's the one to blame

- *git blame* - Zeigt den letzten Commit jeder Zeile einer Datei an
- Wer hat einen Bug verursacht ?
- Warum wurde dieser Code hinzugefügt



git log

- Welche Änderungen wurde wo zuletzt gemacht
- Was sind die letzten Änderungen eines Autors



Push it to the lim... origin

- *git push (remote)*
- Schickt die Differenz von lokalen Branch zu Remote an den Remote
- Alles was auf dem Remote ist kann nicht verändert werden
 - Es sei denn man pushed mit -f (NEIN NICHT TUN)



Dokumentation

git help <command>

Schnelle Dokumentation zu weiteren Optionen und Kommandos



#hackundsoehne

www.hackundsoehne.de



Weitere Kommandos

...

z.B.:

- `git squash`
- `git cherry-pick`
- `git diff`
- `git revert`
- `git reset`
- `git status`

