

# GIT

Ветвление (Branching)

Часть 1

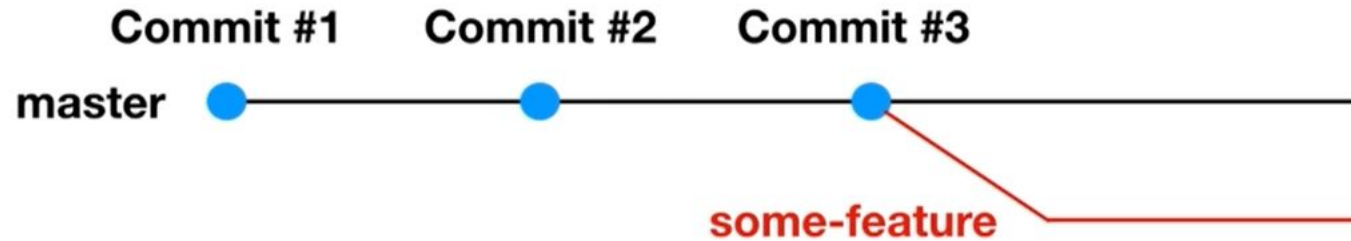
# Зачем использовать ветвление?



- Новые функции разрабатываются в отдельных ветках
- Ветка master содержит стабильную версию проекта. Можем вернуться на master в любой момент.
- Сразу несколько разработчиков могут работать в своих ветках над своими задачами. После завершения работы над задачами, эти ветки "сливаются" в master ветку.

# git branch название\_ветки

**Команда для создания новой ветки**



**git branch some-feature**

\*вызвали команду, находясь на  
Commit #3  
новая ветка "отпочковывается"  
именно с этого коммита

# git branch

**Команда для просмотра, на какой ветке мы сейчас находимся**

```
[test_project $ git branch  
* master
```

**\*На какой ветке  
находится указатель  
HEAD**

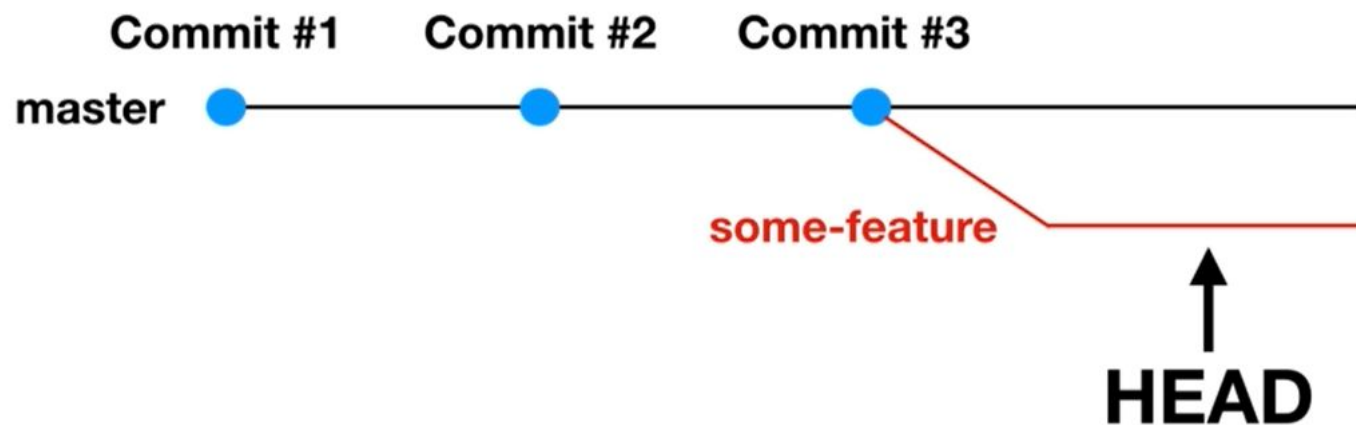
# git branch -d название\_ветки

Команда для удаления ветки

```
test_project $ git branch -d some-feature  
Deleted branch some-feature (was a0f92b5).
```

# Переключение между ветками

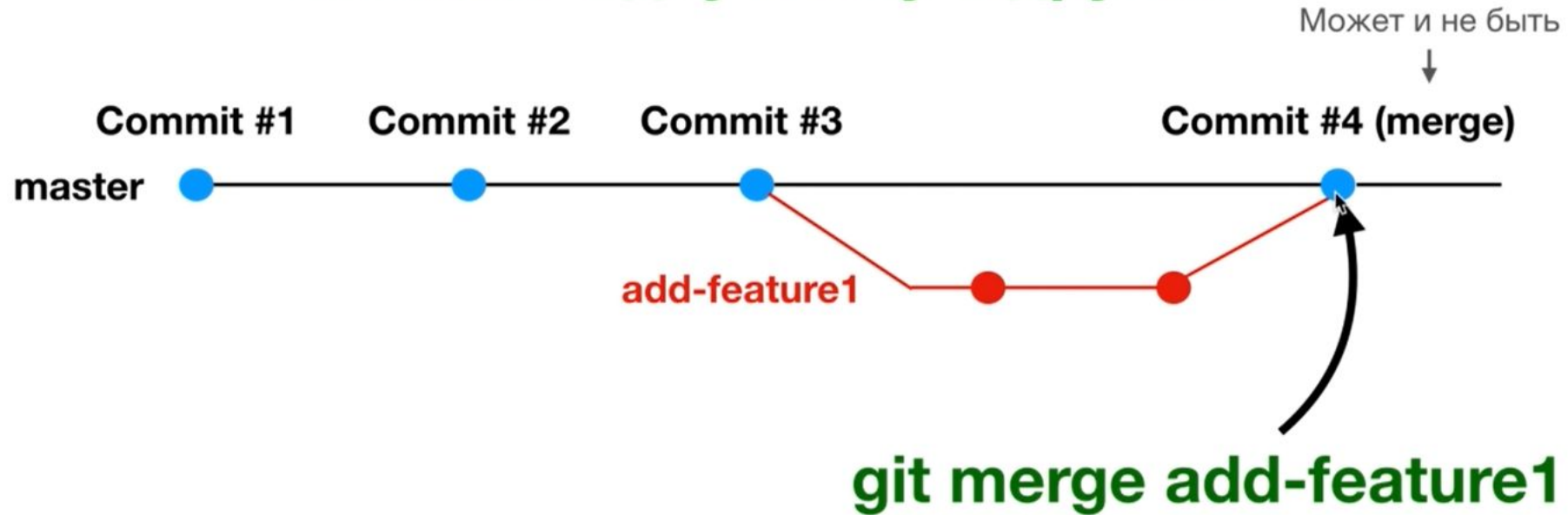
`git checkout` название\_ветки



`git checkout some-feature`

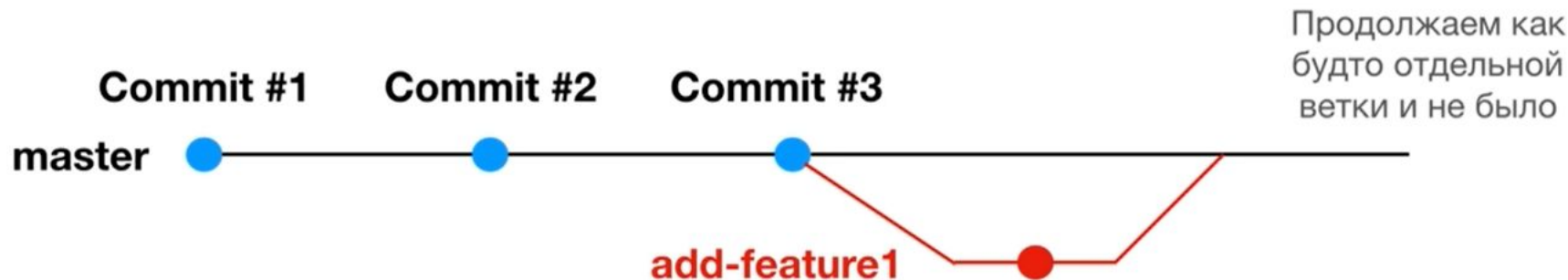
# git merge

Сливает одну ветку с другой



Слили ветку **add-feature1** в ветку master

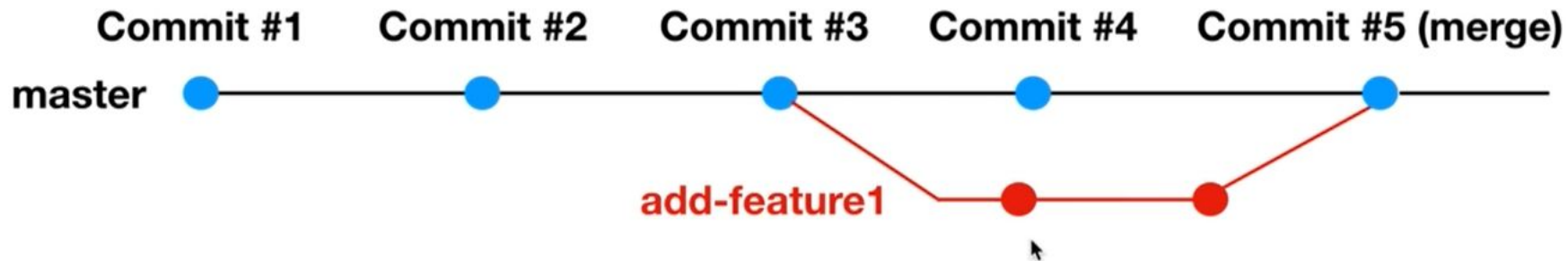
# Fast - Forward merge



- Пока мы работали в **своей** ветке, в ветке master ничего не произошло (не было новых коммитов)
- GIT'у очень легко слить ветку **add-feature1** в master (не может возникнуть конфликтов)
- Не создается отдельный commit для слияния (merge commit)



# ~~Fast - Forward merge~~



- Пока вы работали в своей ветке, кто-то добавил коммиты в ветку master
- Или вы сами добавили новые коммиты в ветку master (пример - вас попросили исправить какой-нибудь критический баг и запустить на GitHub)
- Могут возникнуть конфликты. Гит попытается решить их самостоятельно. Если у него не получится, придется решать их вручную.
- Merge commit создается.

















