Git. Eine Einführung.

Was ist das und was macht man damit?
Aber vor Allem: Wie funktioniert das?

(pro)

- Verteilt
- Nicht-lineare Entwicklung
- Kryptographisch sicher
- Verzeichnisbasiert
- Blitzschnell & effizient

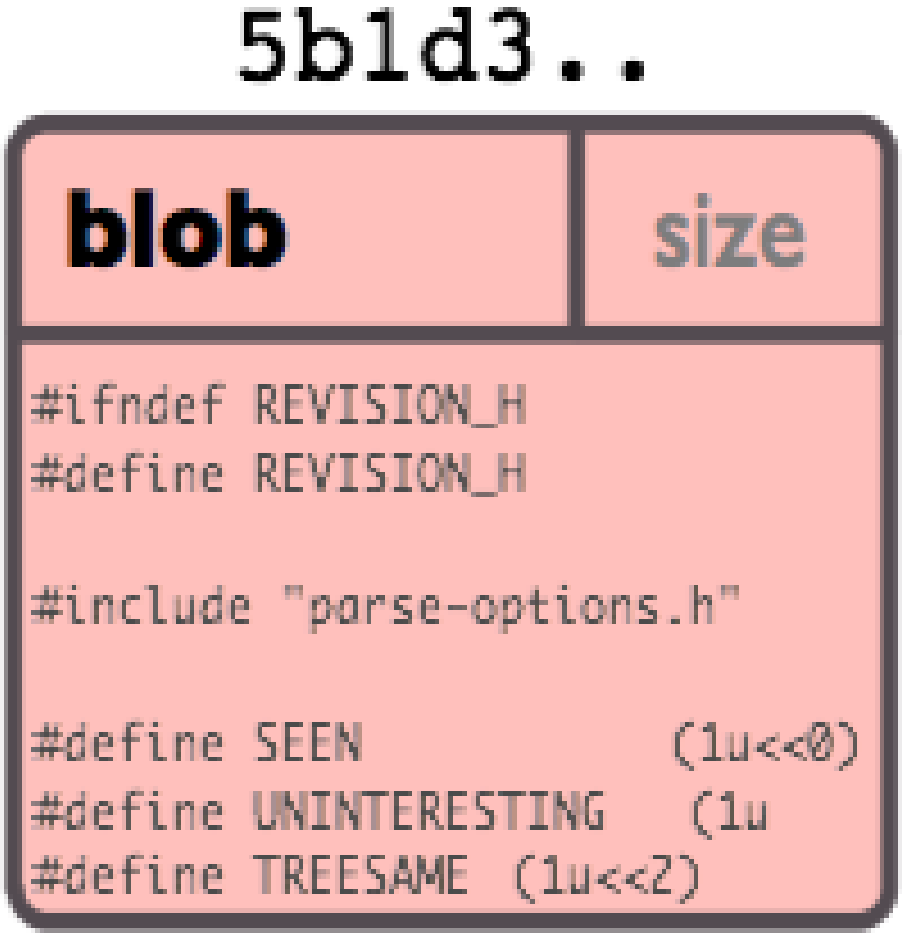
(con)

- Nicht unbedingt benutzerfreundliches user-interface
- Windows-portabilität wackelig...aber geht.

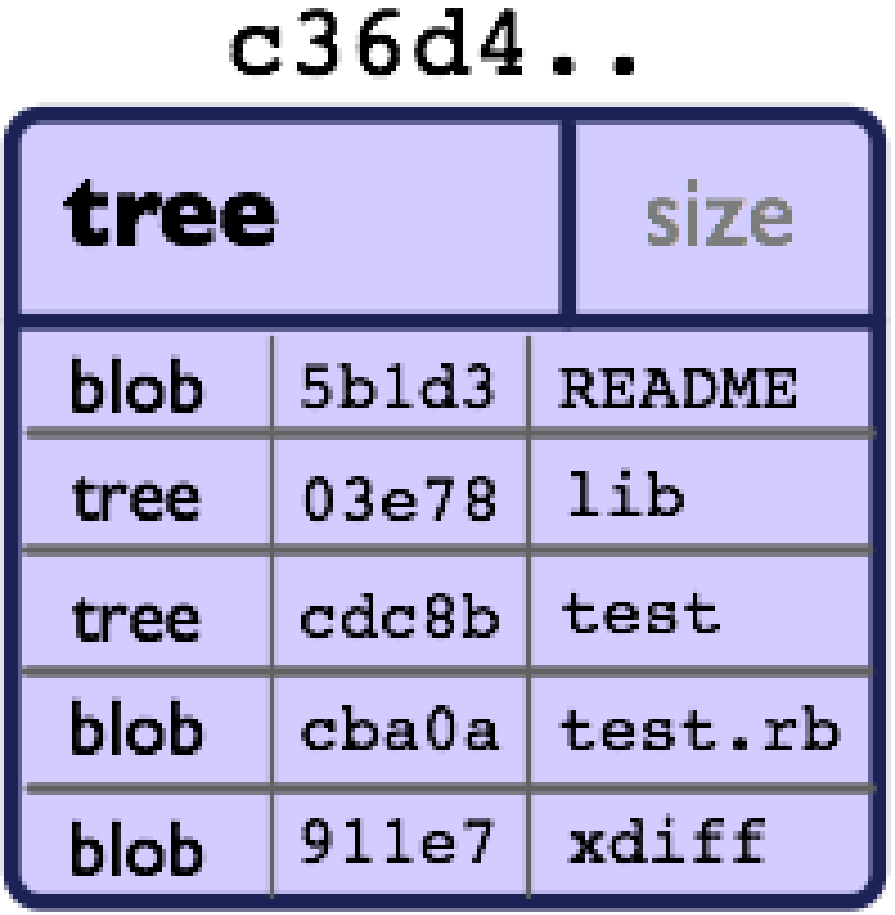
Innereien

- Trackt keine einzelnen Dateien
- sondern Ordnerstrukturen = trees
- Einzelne Dateien = blobs
- Weitere objekte sind commits und tags

blob



tree



commit

Grundlegende Konfiguration

```
$ git config - -global user.name “Foo Nerd”
$ git config - -global user.email “23@42.de”
$ cat ~/.gitconfig
[user]
        name = Foo Nerd
        email = 23@42.de
```

Genug Theorie?

```
$ mkdir foo
$ cd foo
$ git init
Initialized empty Git repository in /home/nerd/foo/
```

Was haben wir denn da?

z.B.

- .git Verzeichnis
- .git/config
- .git/objects
- .git/HEAD
- u.v.m.

Rumspielen...

```
$ cat .git/HEAD
ref: refs/heads/master

$ cat .git/refs/heads/master
cat: No such file or directory
```

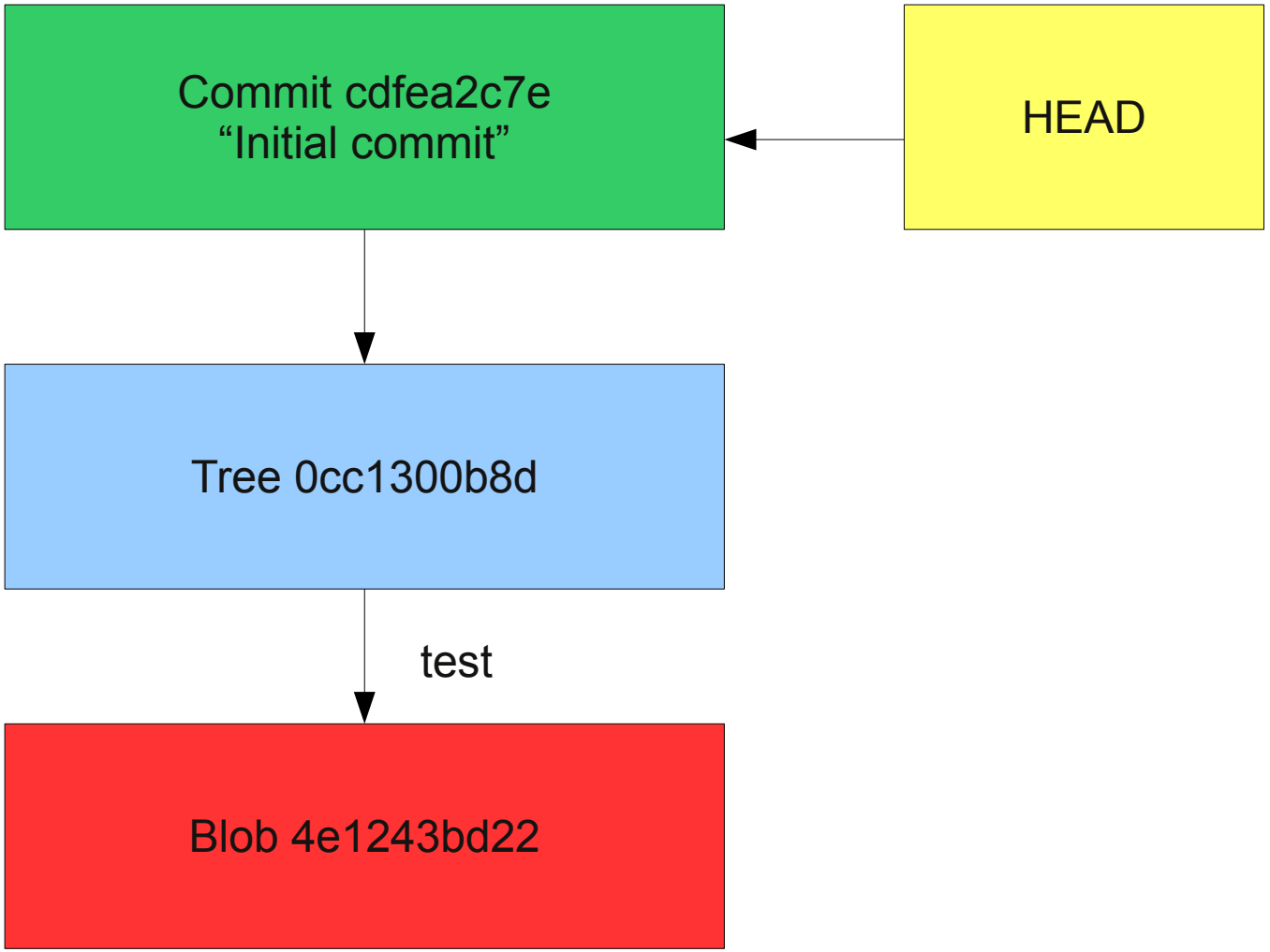
Erster commit

```
$ echo “test” > test
$ git status
# Untracked files:
#       test
$ git add test
$ git status
# Changes to be committed:
#       new file:   test
$ git commit -m “Initial commit”
[master (root-commit) cdfea2c] Initial commit
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 test
```

Rumspielen...

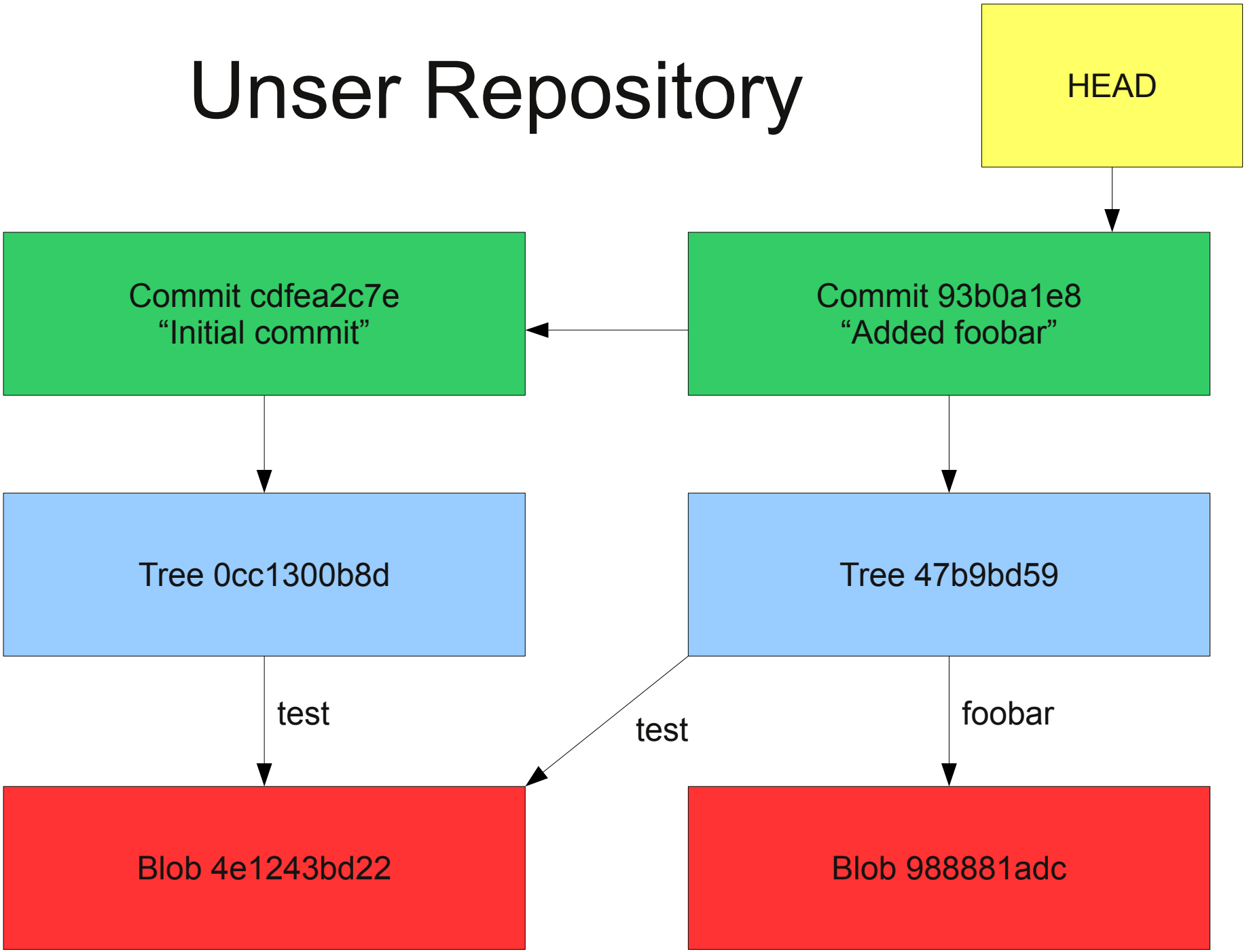
```
$ cat .git/refs/heads/master
cdfea2c7e073cd0de421550cb0b8e7aa479d576d
```

Unser Repository



Zweiter commit

```
$ echo “foobar” > foobar
$ git add foobar
$ git commit -m “added foobar”
[master 93b0a1e] added foobar
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 foobar
```


Unser Repository
HEAD
Commit cdfea2c7e
“Initial commit”
Commit 93b0a1e8
“Added foobar”
Tree 0cc1300b8d
Tree 47b9bd59
test
test
foobar
Blob 4e1243bd22
Blob 988881adc

git log

$ git log

```
commit 93b0a1e86697638ecf7f07beaeb5a694f218f1f0
Author: Foo Nerd <23@42.de>
Date:   Sun Aug 8 21:04:28 2010 +0200

    added foobar

commit cdfea2c7e073cd0de421550cb0b8e7aa479d576d
Author: Foo Nerd <23@42.de>
Date:   Sun Aug 8 20:50:26 2010 +0200

    Initial commit
```

branches

```
$ git branch
* master
$ git branch mybranch
$ git branch
* master
  mybranch
$ git checkout mybranch
Switched to branch 'mybranch'
$ git branch
  master
* mybranch
```

Rumspielen...

```
$ cat .git/HEAD
ref: refs/heads/mybranch
$ ls .git/refs/heads/
master  mybranch
$ diff .git/refs/heads/*
$
```

Dritter commit

```
$ echo “barfoo” > foobar
$ git add foobar
$ git commit -m “changed content of foobar”
[mybranch f60ae9f] changed content of foobar
 1 files changed, 1 insertions(+), 1 deletions(-)
```

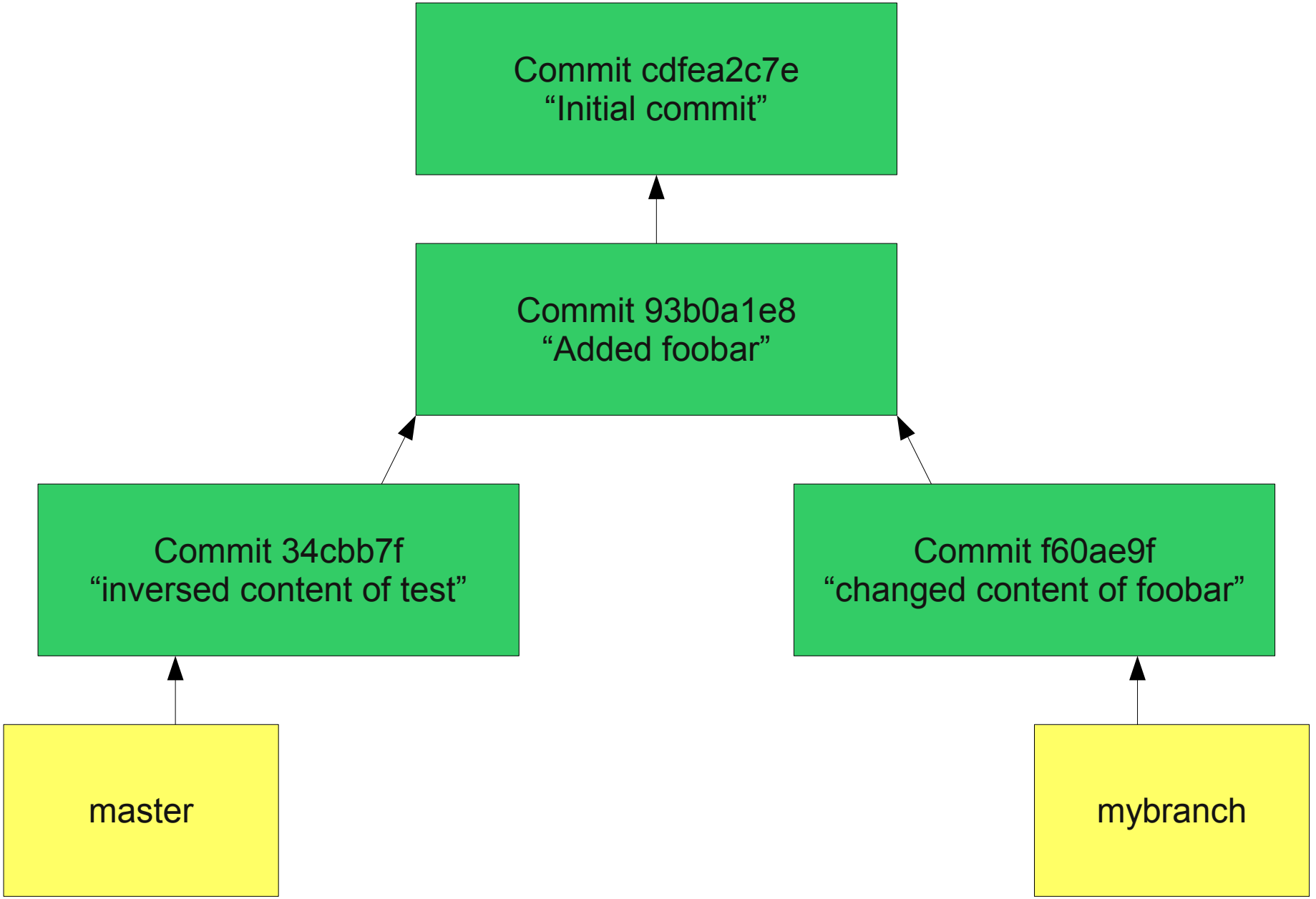
Rumspielen...

```
$ git diff .git/refs/heads/*
1c1
< 93b0a1e86697638ecf7f07beaeb5a694f218f1f0
---
> f60ae9f02c1a758b278fc0d373cd1b8b45913205
```

Vierter commit

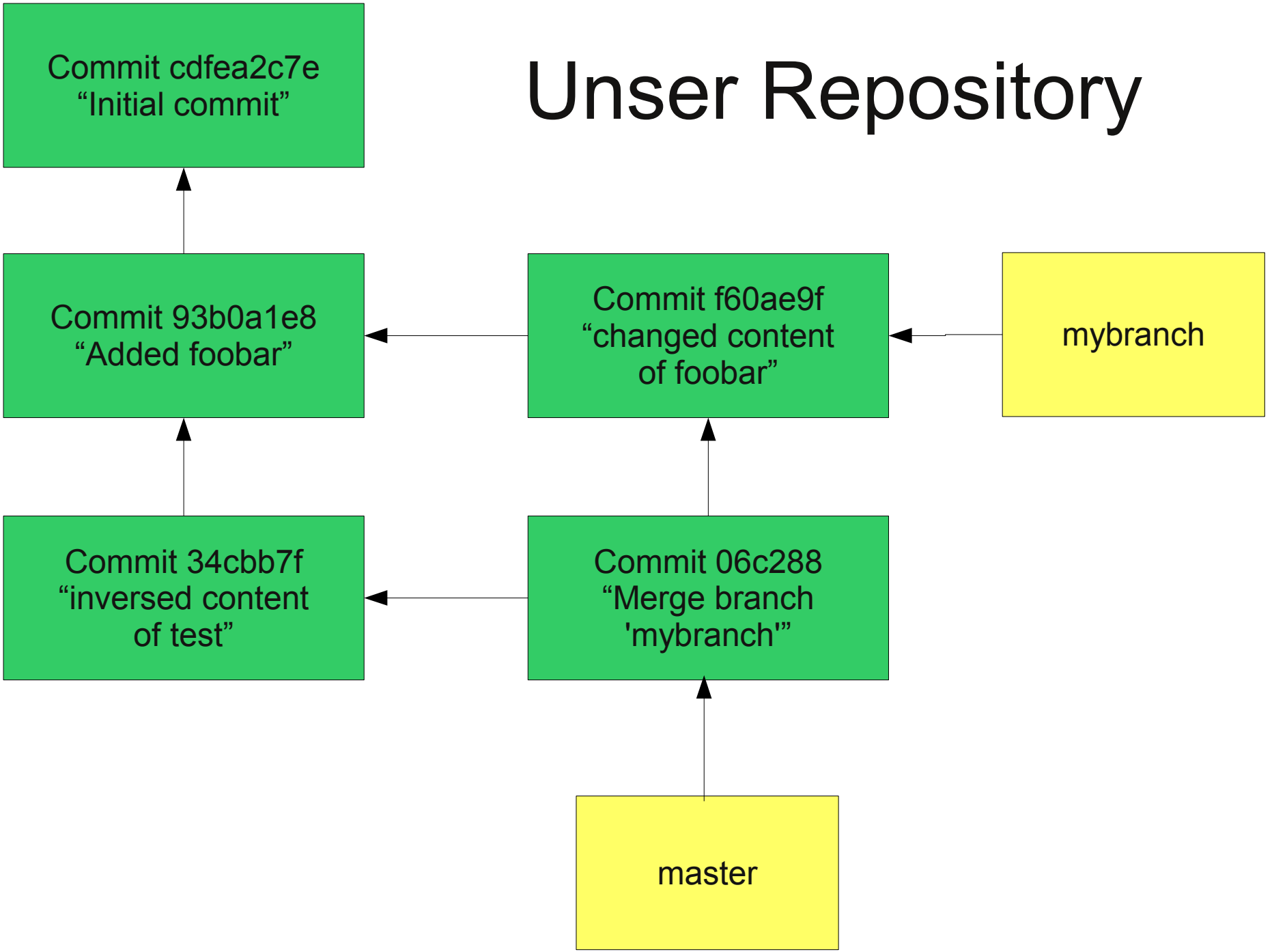
```
$ git checkout master
Switched to branch 'master'
$ git echo “tset” > test
$ git add test
$ git commit -m “inversed content of test”
[master 34cbb7f] inversed content of test
 1 files changed, 1 insertions(+), 1 deletions(-)
```

Unser Repository



git merge

```
$ git branch
* master
  mybranch
$ git merge mybranch
Merge made by recursive.
 foobar |    2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)
```

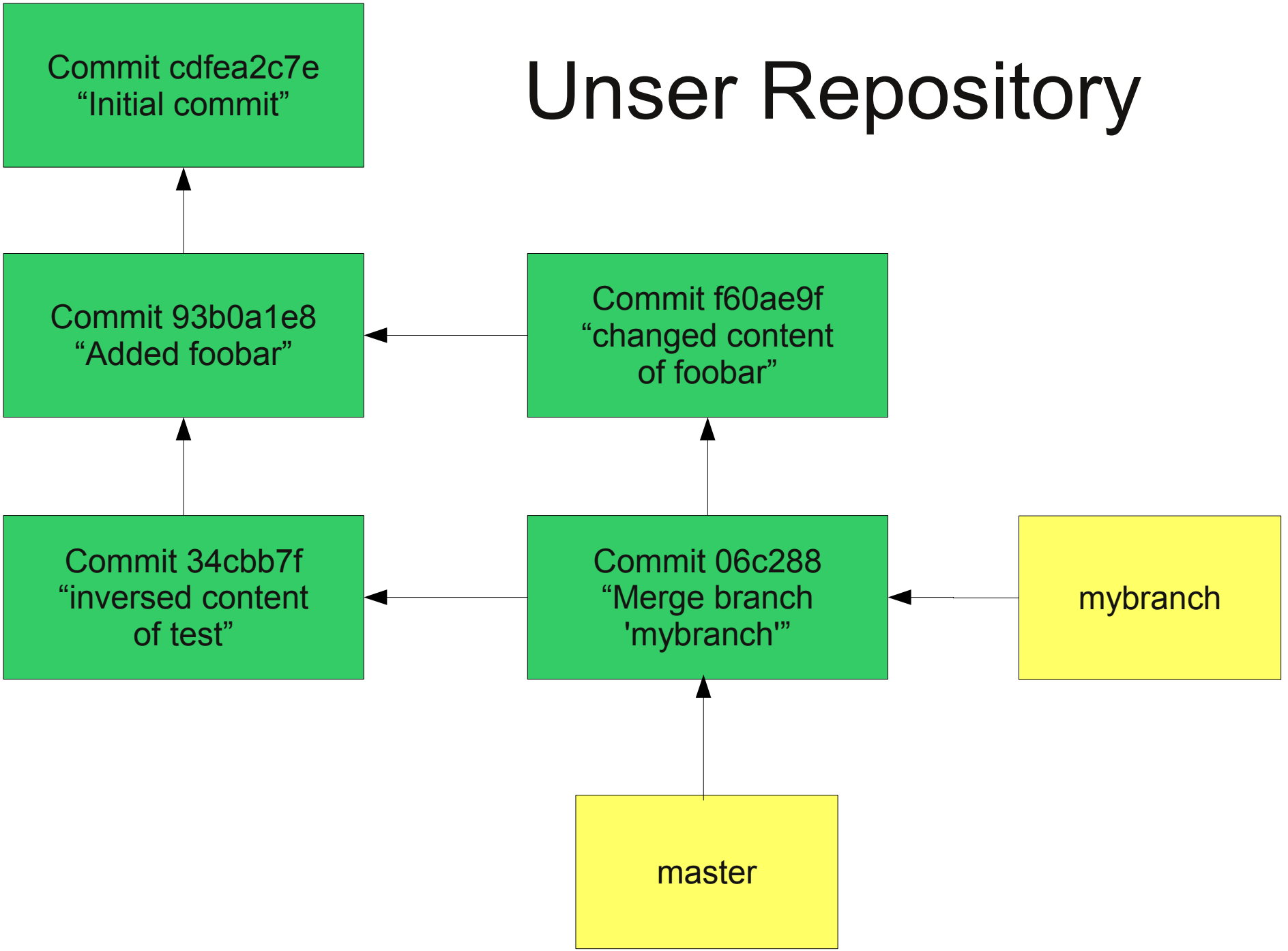

Unser Repository
Commit cdfea2c7e
“Initial commit”
Commit 93b0a1e8
“Added foobar”
Commit f60ae9f
“changed content
of foobar”
mybranch
Commit 34cbb7f
“inversed content
of test”
Commit 06c288
“Merge branch
'mybranch'”
master

git log

```
$ git log --oneline --graph
*   06c2882 Merge branch 'mybranch'
|\
| * f60ae9f changed content of foobar
* | 34cbb7f inversed content of test
|/
* 93b0a1e added foobar
* cdfea2c Initial commit
```

Zurückmergen

```
$ git checkout mybranch
Switched to branch 'mybranch'
$ git merge master
Updating f60ae9f..06c2882
Fast-forward
 test |    2 +-
 1 files changed, 1 insertions(+), 1 deletions(-)
```


Unser Repository
Commit cdfea2c7e
“Initial commit”
Commit 93b0a1e8
“Added foobar”
Commit f60ae9f
“changed content
of foobar”
Commit 34cbb7f
“inversed content
of test”
Commit 06c288
“Merge branch
'mybranch'”
mybranch
master

Konflikte erzeugen

```
$ echo “test1” > foobar
$ git commit -am “foobar → test1”
[mybranch 180d8e6] foobar -> test1
 1 files changed, 1 insertions(+), 1 deletions(-)
$ git checkout master
Switched to branch 'master'
$ echo “test2” > foobar
$ git commit -am “foobar → test2”
$ git merge mybranch
Auto-merging foobar
CONFLICT (content): Merge conflict in foobar
Automatic merge failed; fix conflicts and then
commit the result.
```

Konflikte lösen

```
$ cat foobar
<<<<<<< HEAD
test2
=======
test1
>>>>>>> mybranch
$ git mergetool
[..]
$ cat foobar
lösung
$ git commit
```

git remote

```
$ cat .git/config
[core]
        repositoryformatversion = 0
        filemode = true
        bare = false
        logallrefupdates = true
$ git remote add origin user@host:repo.git
$ cat .git/config
[..]
[remote "origin"]
        url = user@host:repo.git
        fetch = +refs/heads/*:refs/remotes/origin/*
```

Push & Pull

Git merge, branch etc. nur remote...

git push

```
$ git push origin master
Enter passphrase for key '/home/nerd/.ssh/id_rsa':
Counting objects: 21, done.
Compressing objects: 100% (14/14), done.
Writing objects: 100% (21/21), 1.78 KiB, done.
Total 21 (delta 2), reused 0 (delta 0)
To user@host:repo.git
 * [new branch]      master -> master
$ git push
Enter passphrase for key '/home/nerd/.ssh/id_rsa':
Everything up-to-date
```

remote branches

```
$ git push origin mybranch
 * [new branch]      mybranch -> mybranch
$ git branch -r
  origin/master
  origin/mybranch
$ git push origin master:anotherbranch
 * [new branch]      master -> anotherbranch
$ git push origin :anotherbranch
 - [deleted]         anotherbranch
```

Es gibt viel viel mehr..

- Cherry-pick
- Rebase
- Reset
- Stash