



git

Marvin Killing

1 . basics

2 . collabo ration

3 . work flow

1.

basics

git commit

Blob

He l l o W o r l d

a3768903ca03f 7cccb3c6b6
f 474f 88f 8e9a3bb5b

Tree

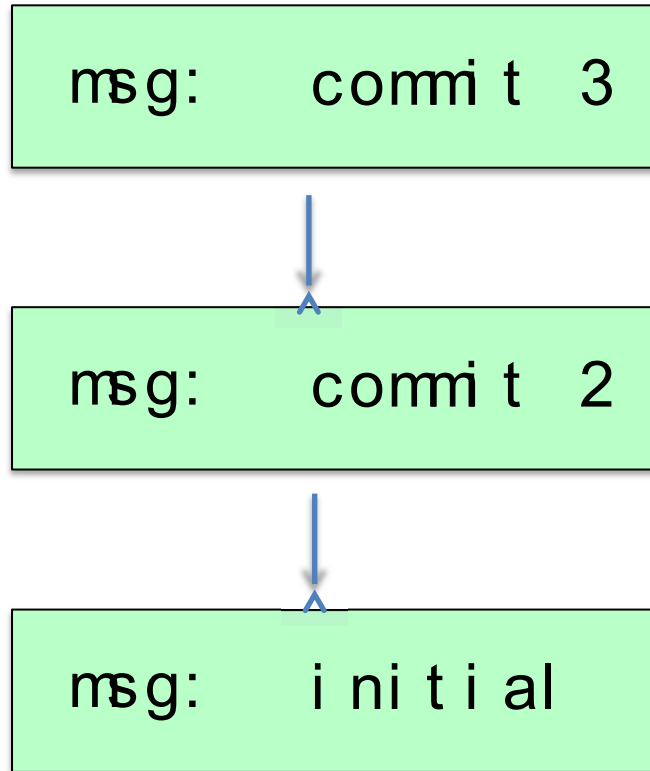
he l l o . t x t A3768903ca03f 7cc...

97b49d4c943e3715f e30f 141cc6f 27a8548cee0e

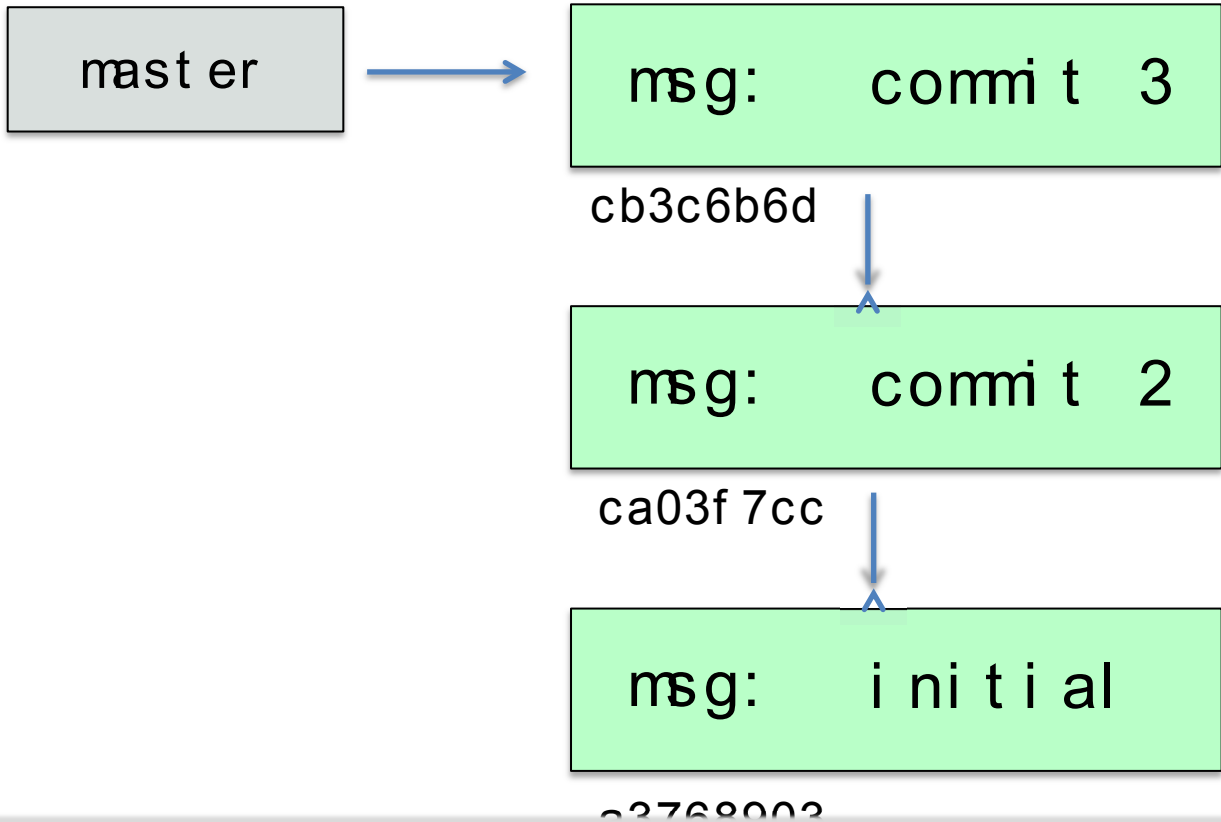
msg: i n i t i a l
tree 97b49d4c943e3715f . . .

a3768903ca03f 7cccb3c6b6f 474f 88f 8e9a3bb5b

Commit

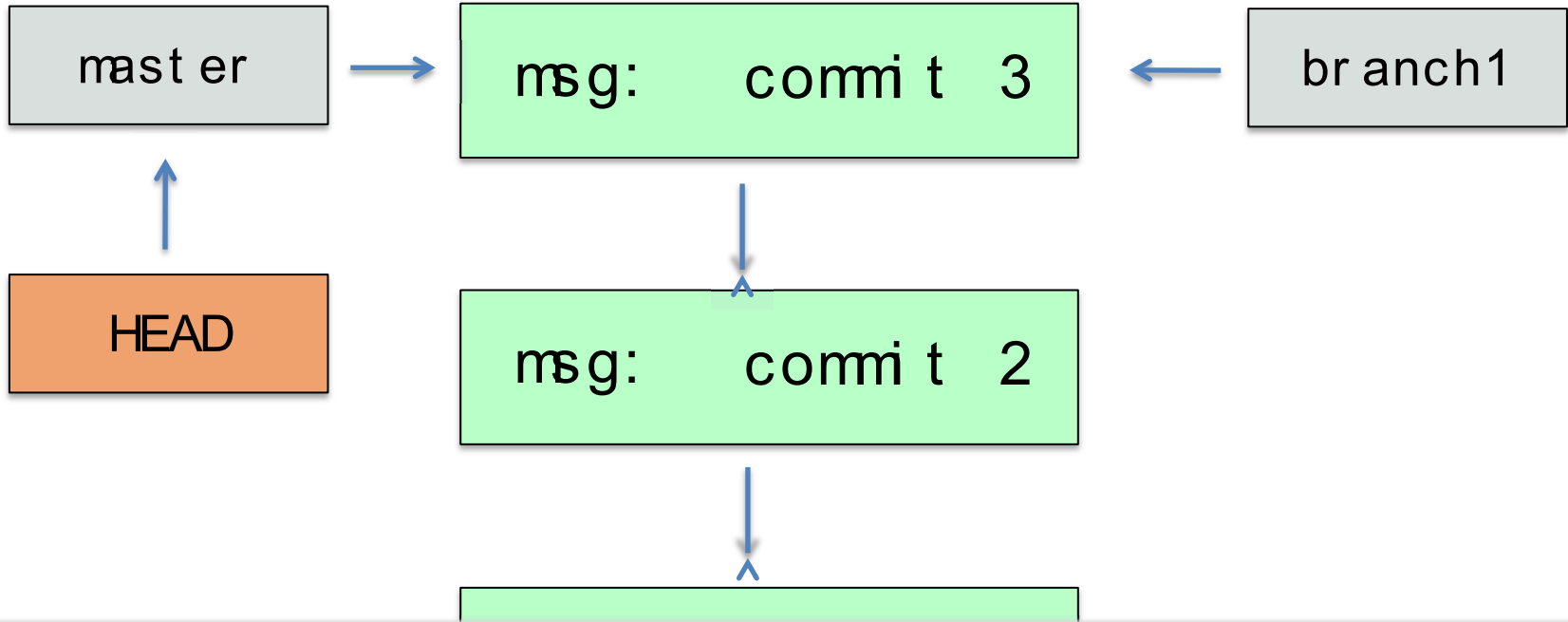


parent



Branch

```
$ git branch branch1
```



HEAD


```
$ git checkout branch1
```

```
$ git checkout a5d743
```

master



HEAD

branch1

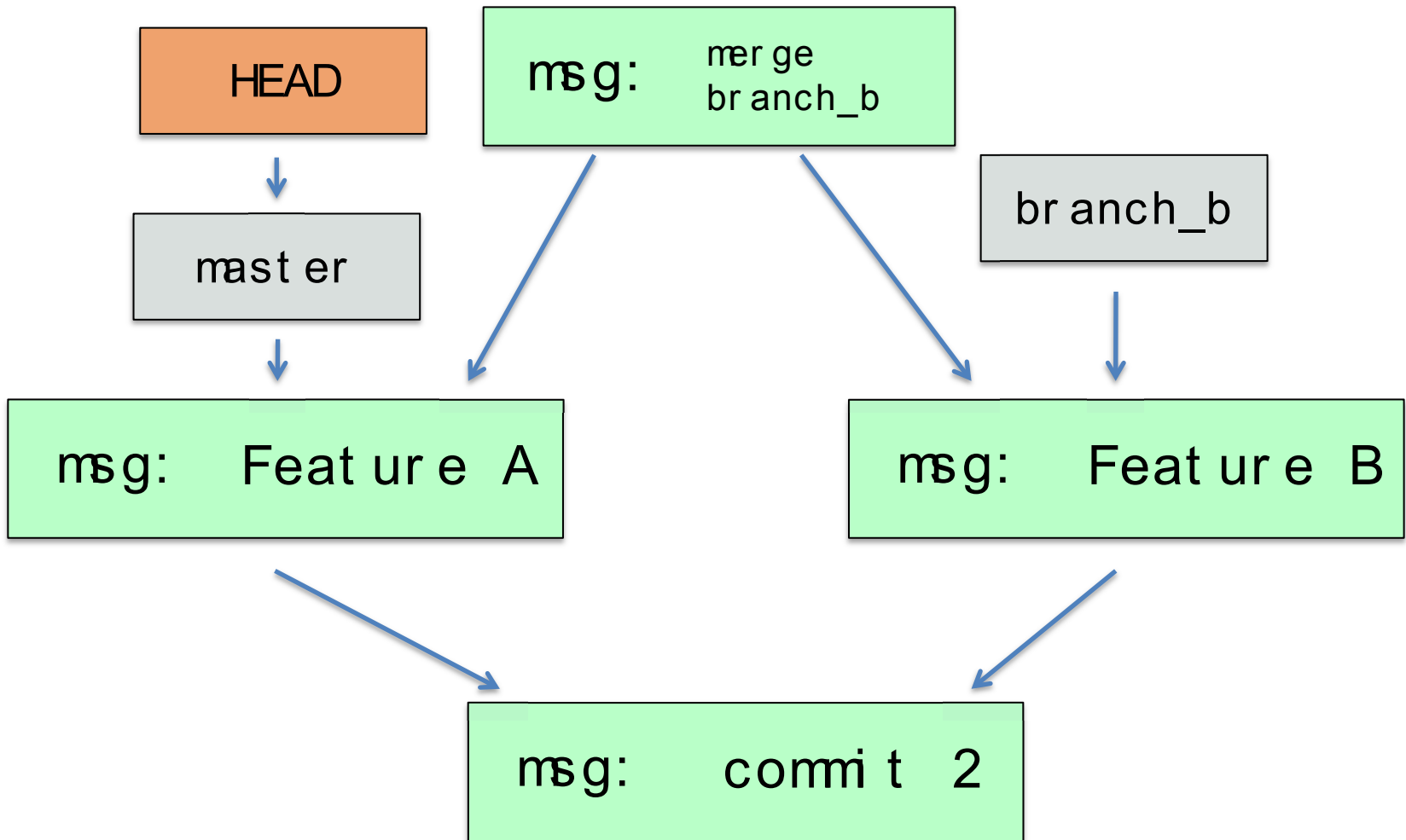
You are in 'detached HEAD' state. You can look around, make experimental changes and commit them, and you can discard any commits you make in this state without impacting any branches by performing another checkout.

msg: commit 2

a5d743

msg: initial

```
$ git merge branch_b
```



<<<<<<< HEAD

Hel l o Wr l d

=====

Hal l o Wel t

>>>>>>> newbr anch

conflict

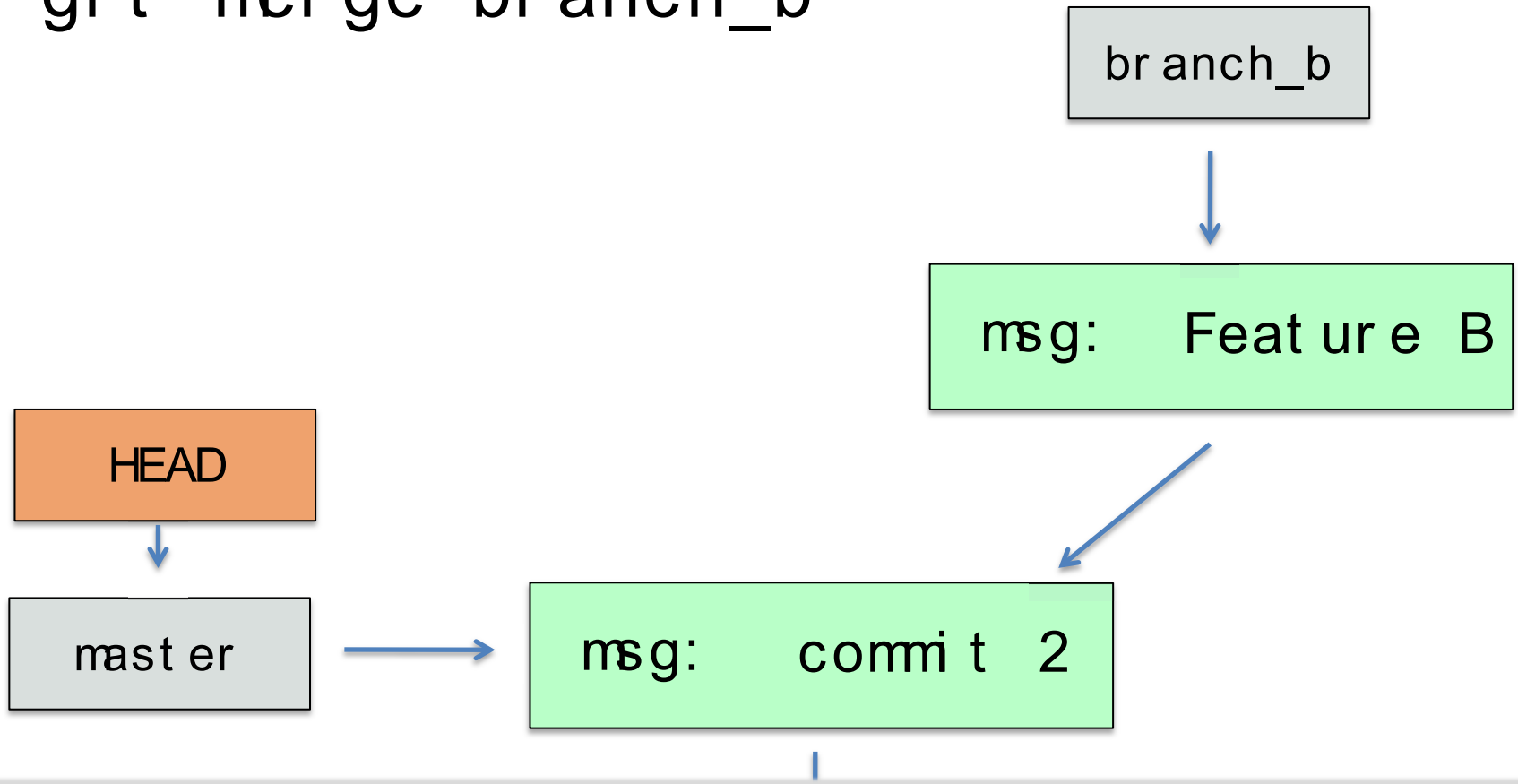
Hel l o Wr l d

resolved!

gi t mer get ool

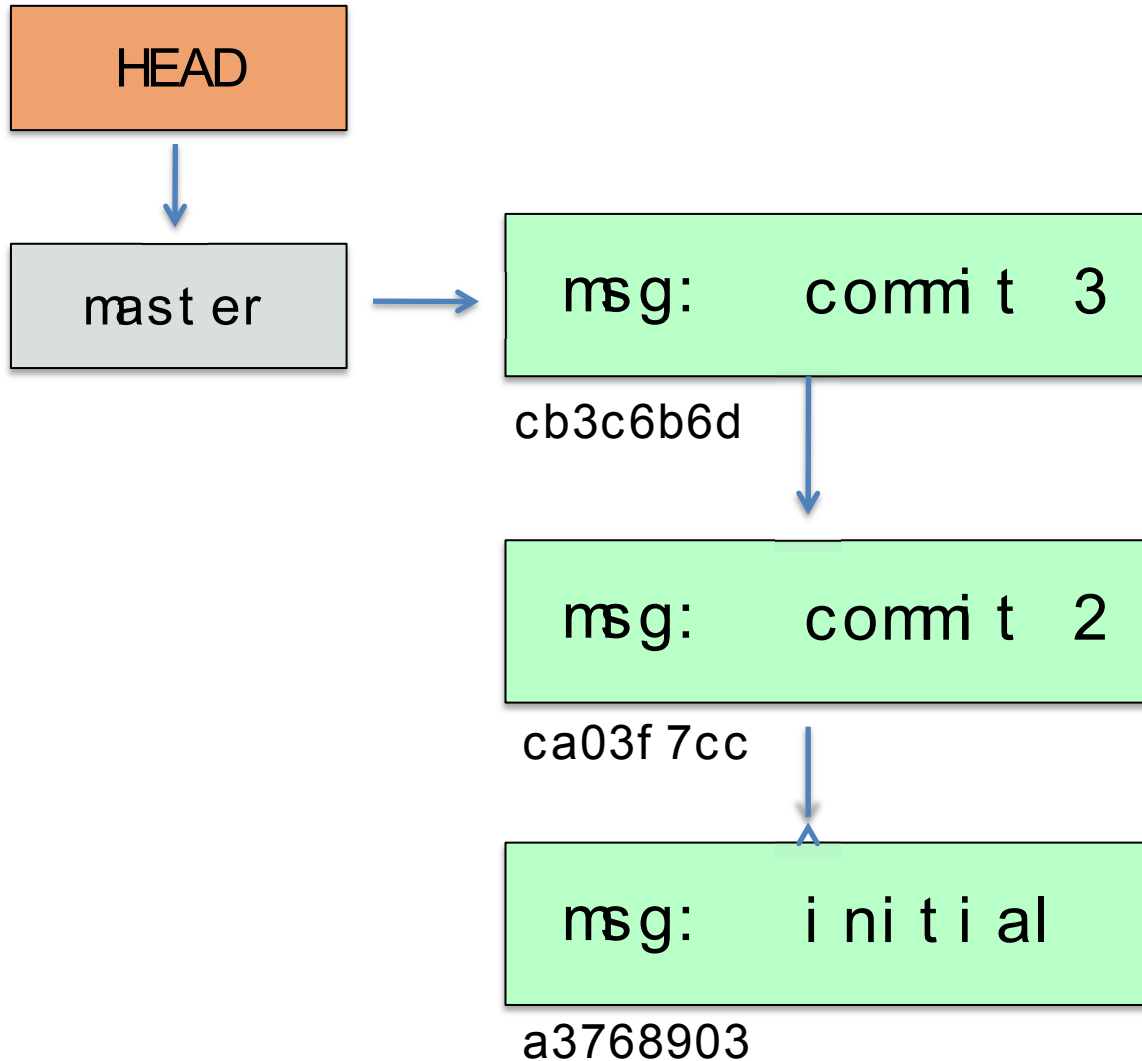
protip

```
$ git merge branch_b
```



fast-forward

```
$ git reset --hard HEAD^
```



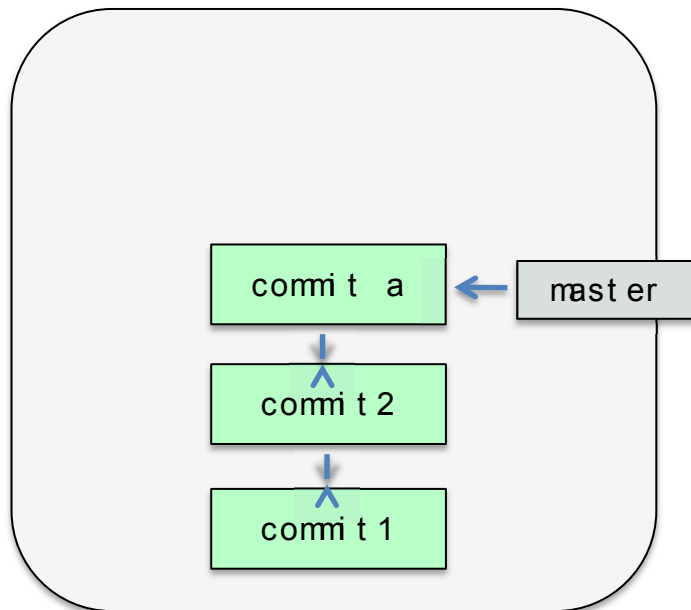
git reflog

protip #2

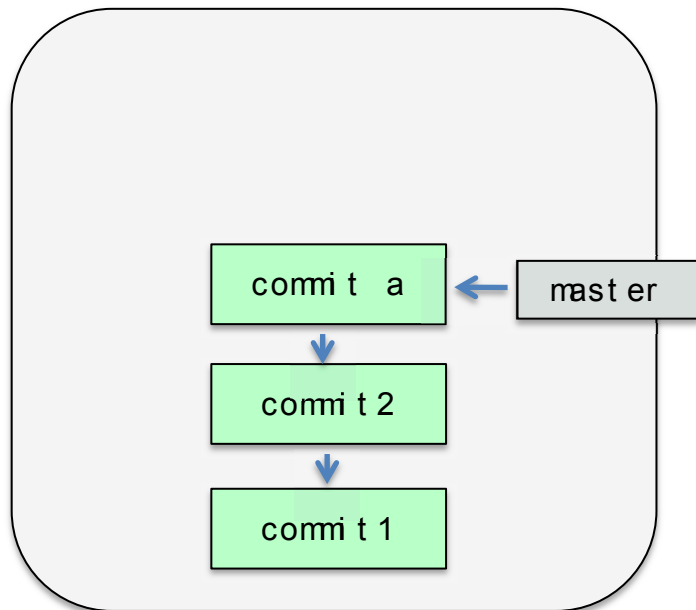
2.

**collabo
ration**

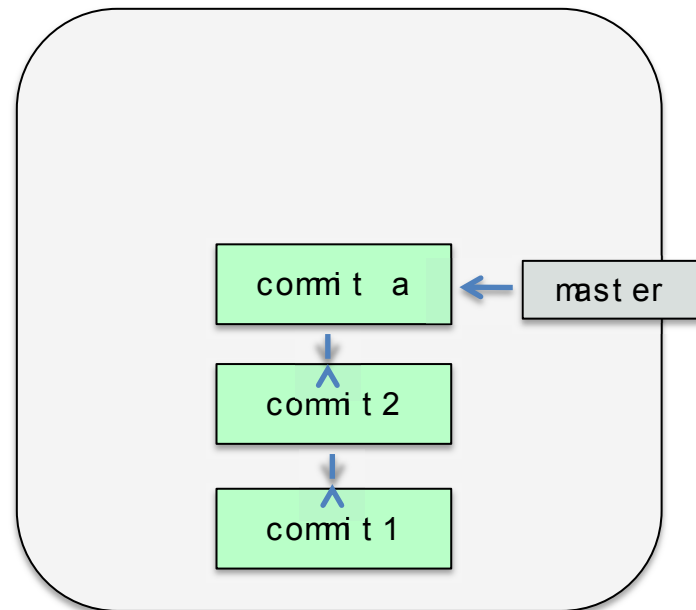
o r i g i n



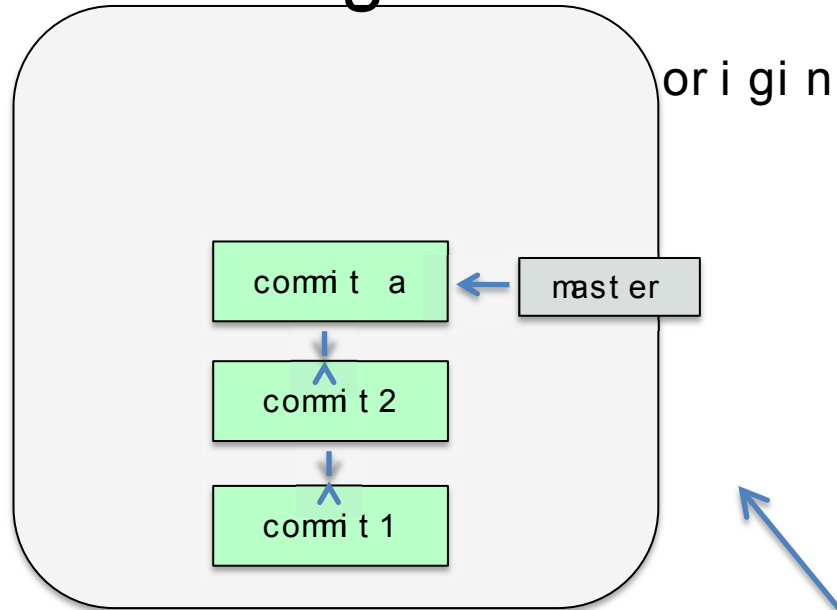
al i ce



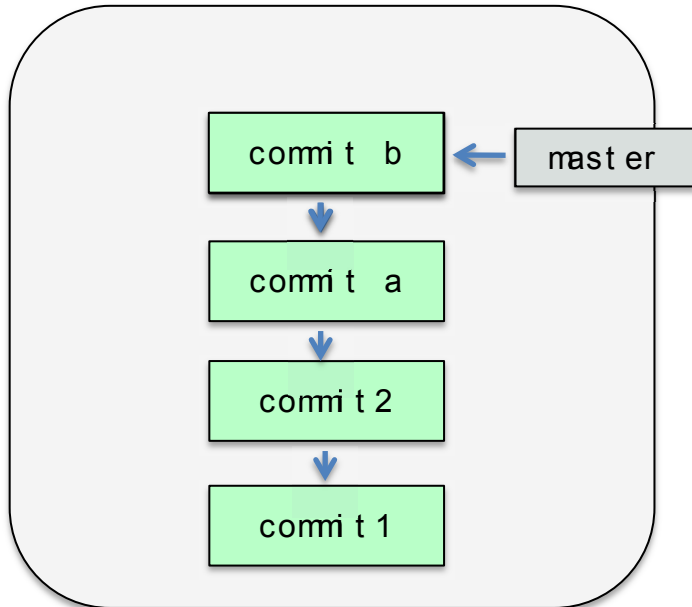
bob



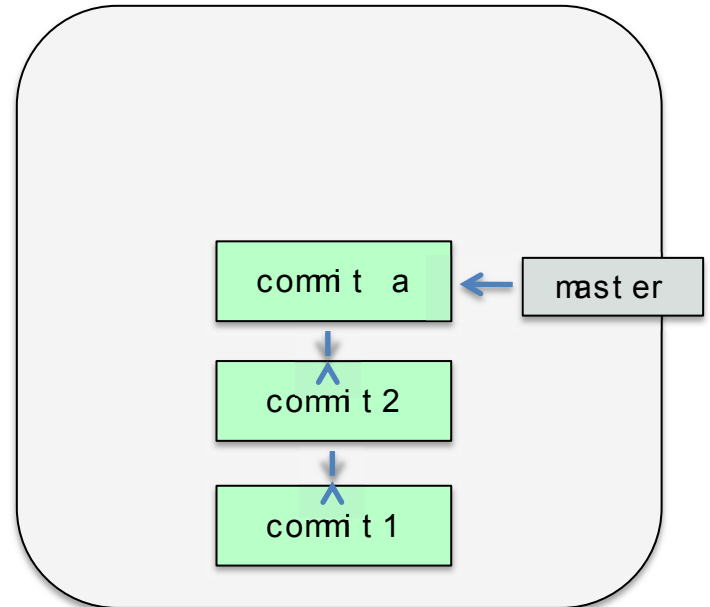
al i ce\$ gi t push ori gi n mast er



al i ce

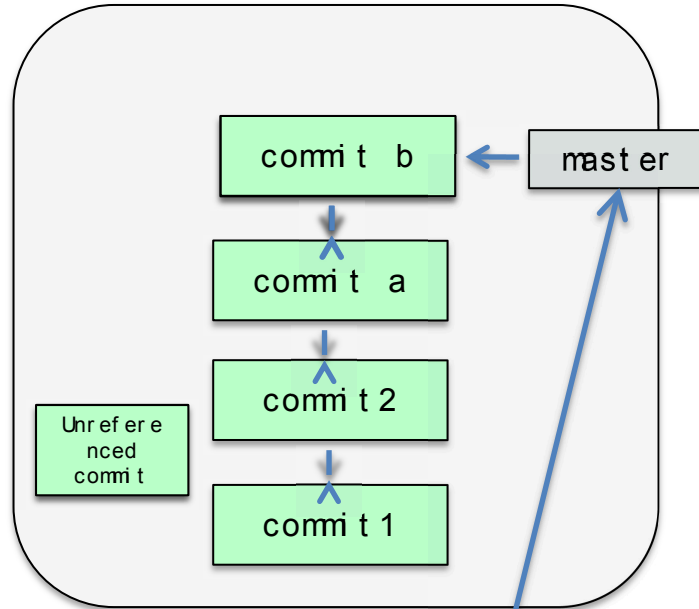


bob

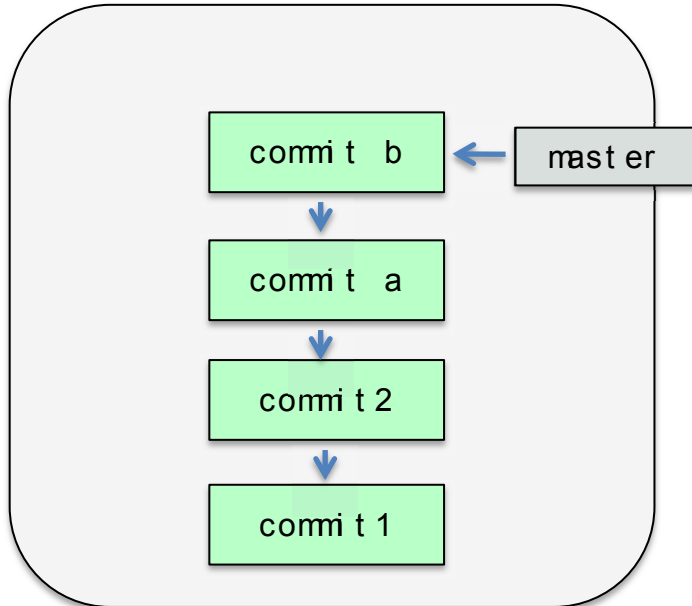


bob\$ git fetch

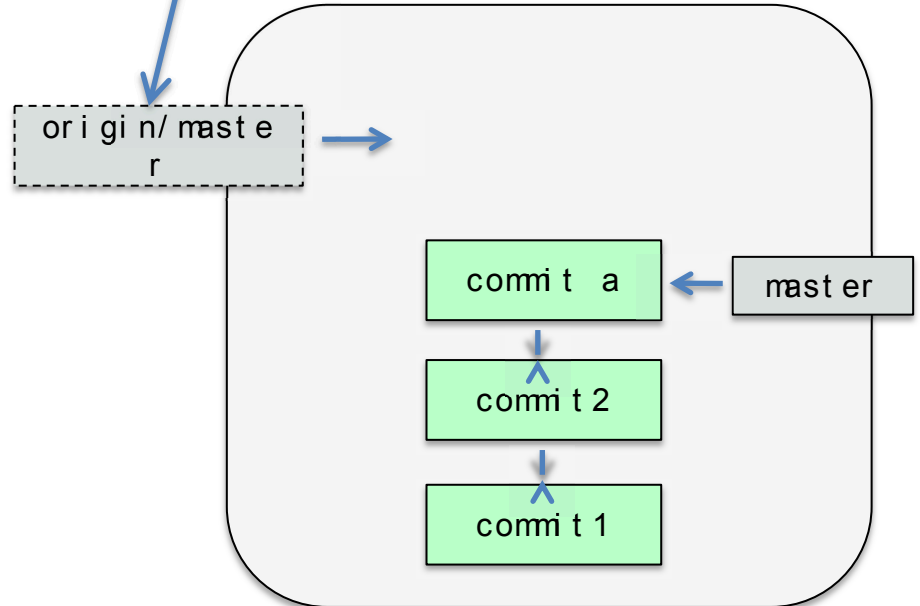
origin



alice

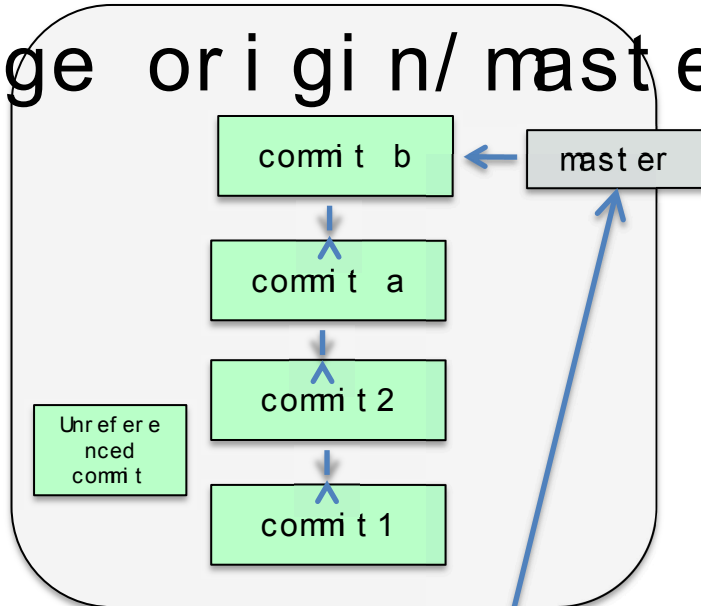


bob

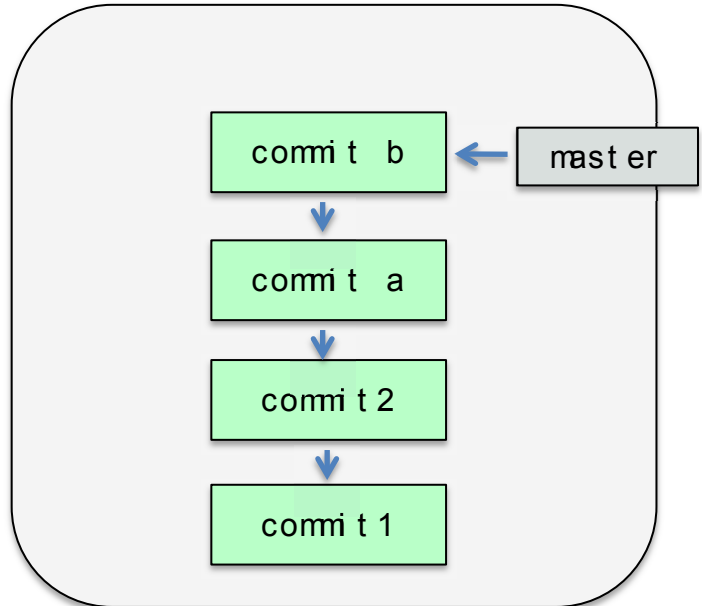


bob\$ git fetch

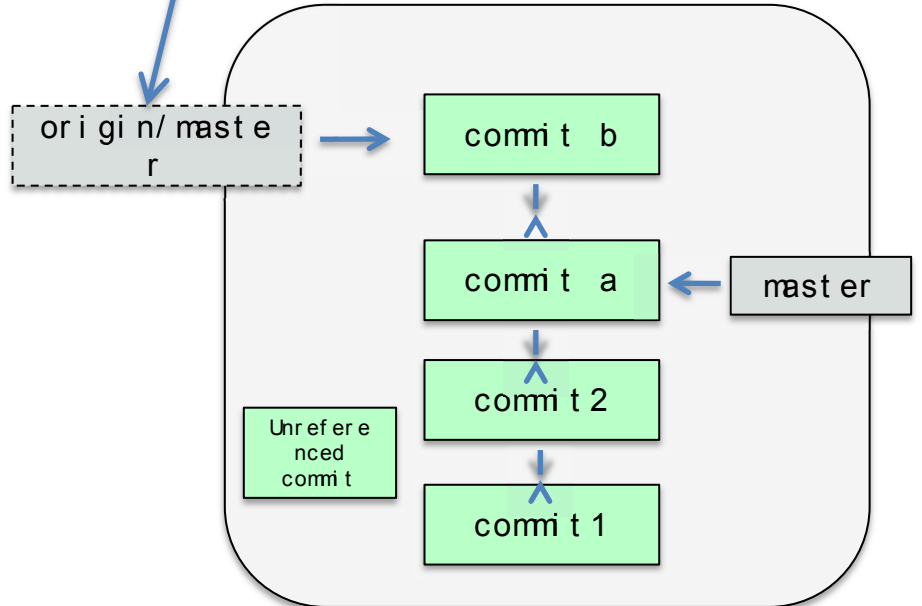
bob\$ git merge origin/master



alice



bob



git pull

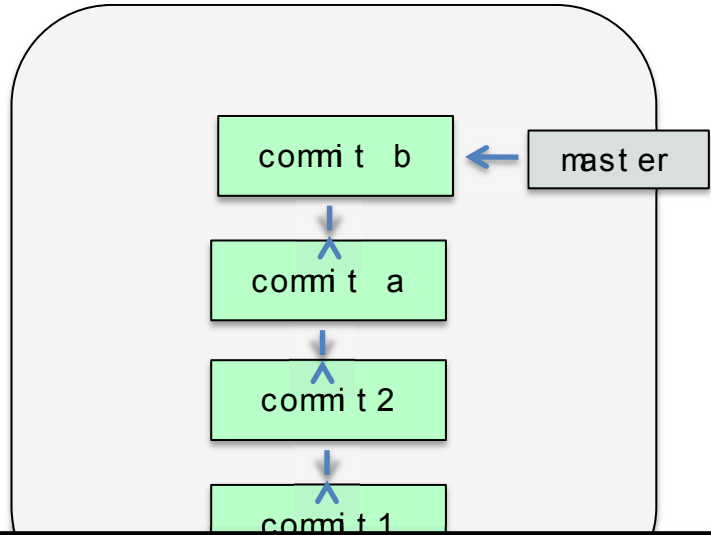
= git fetch +
git merge

protip #3

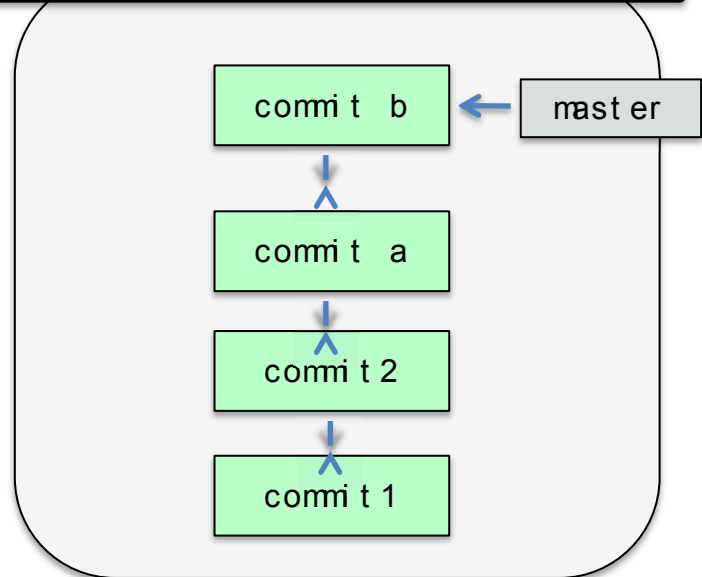
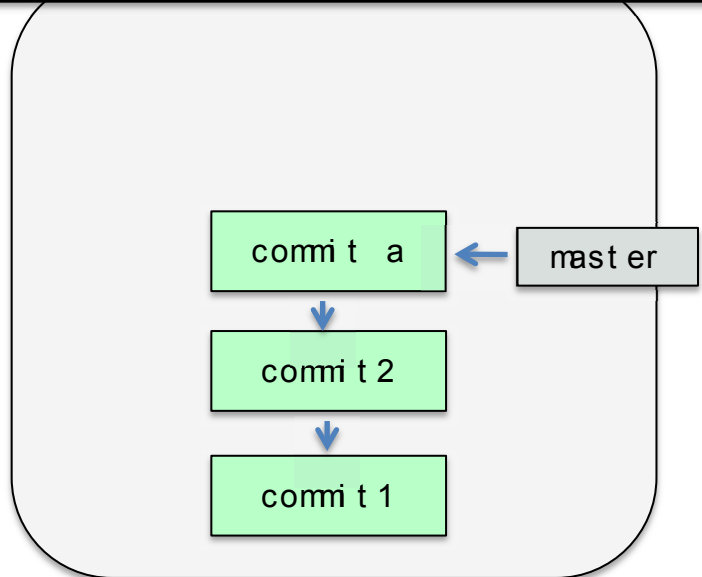
```
git push -f
```

just don't

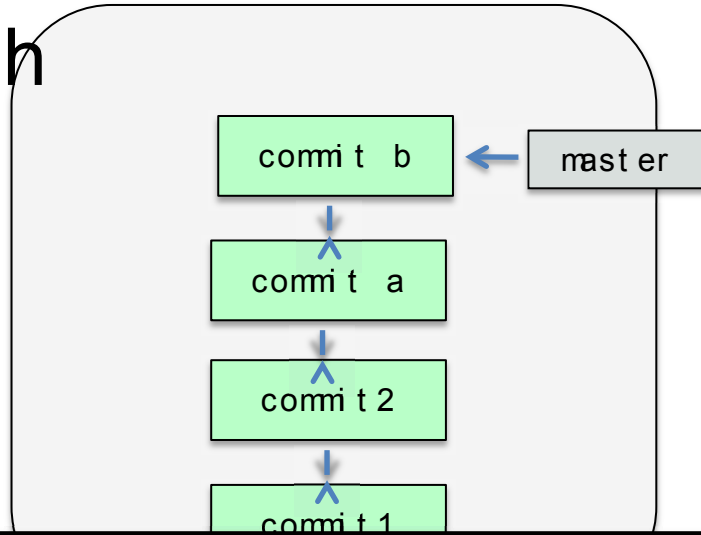
al i ce\$ git push origin



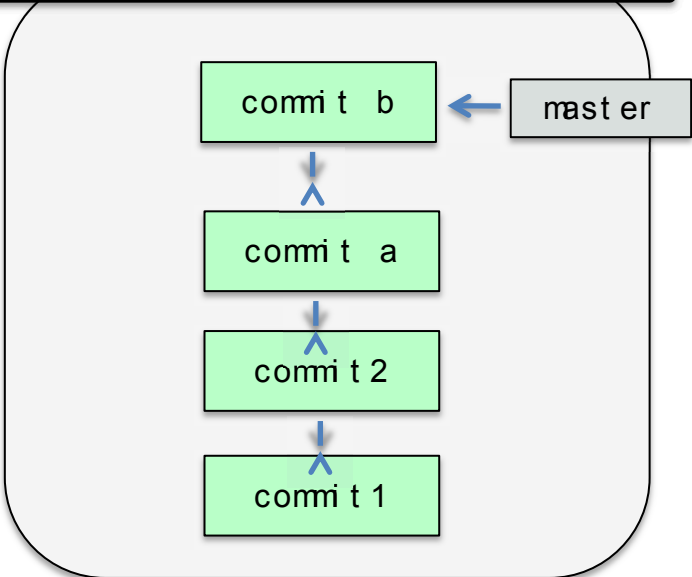
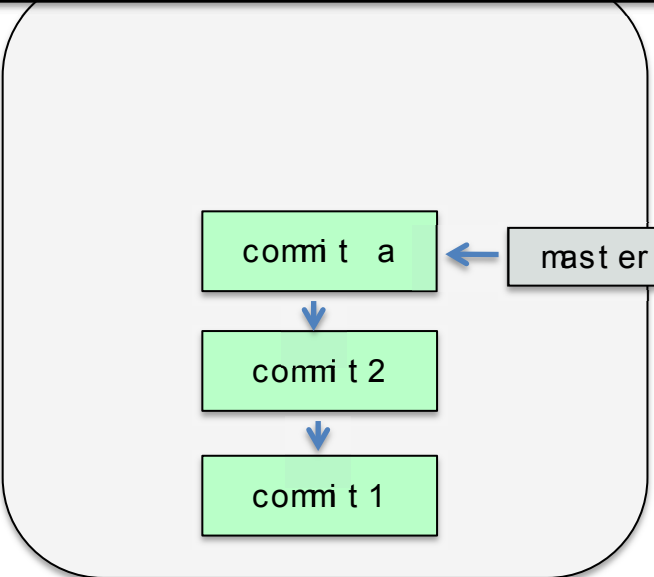
! [r e j e c t e d] mast er - > mast er (n o n - f a s t - f o r w a r d)



alice\$ git push -f origin
bob\$ git push

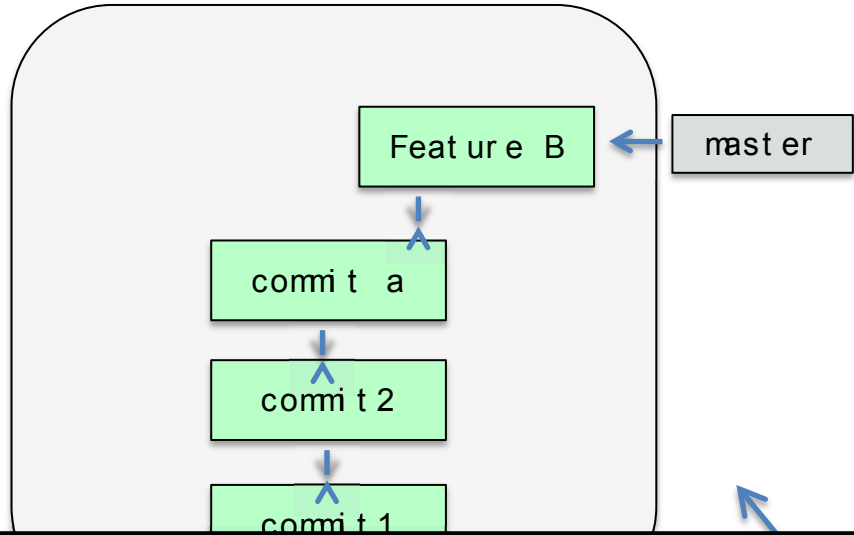


! [rejected] master -> master (non-fast-forward)

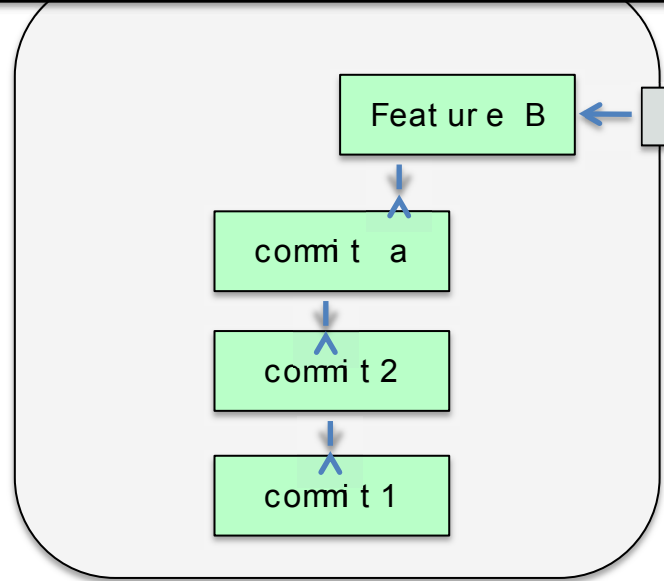
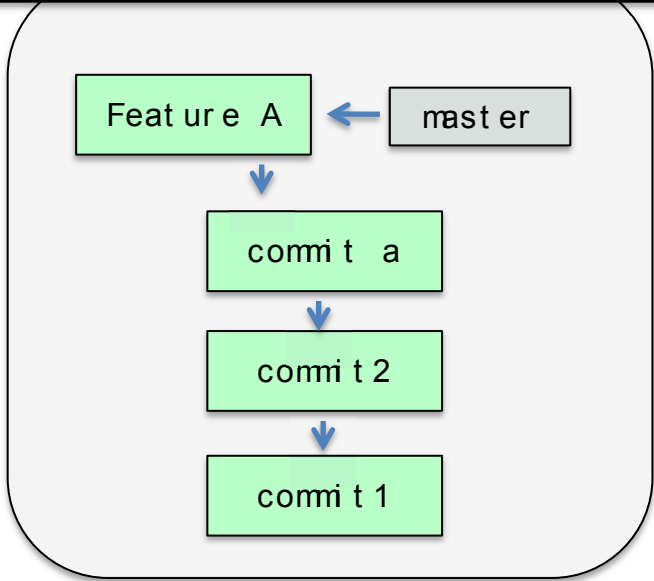


alice\$ git push

origin



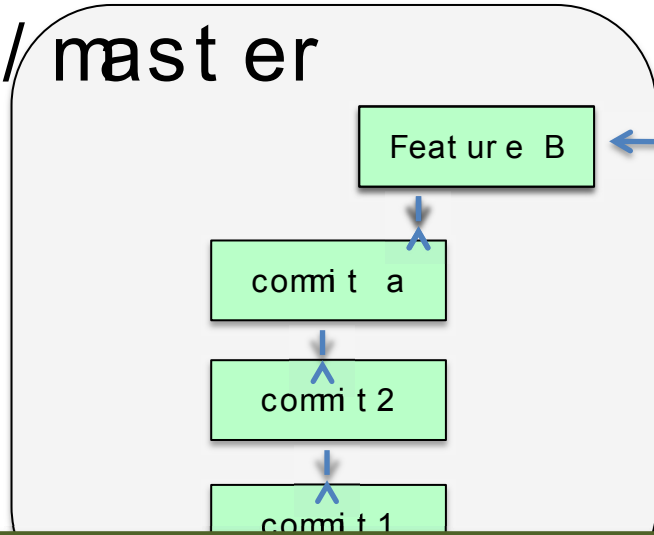
! [rejected] master -> master (non-fast-forward)



fetch
merge
push

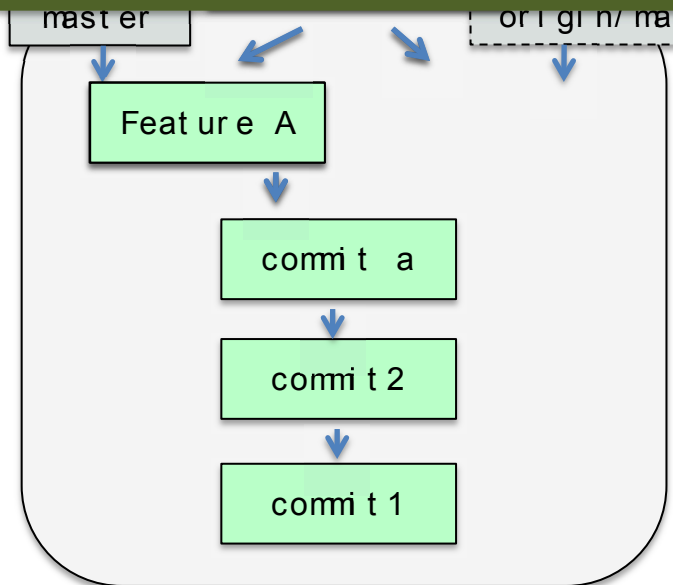
origin

origin/master



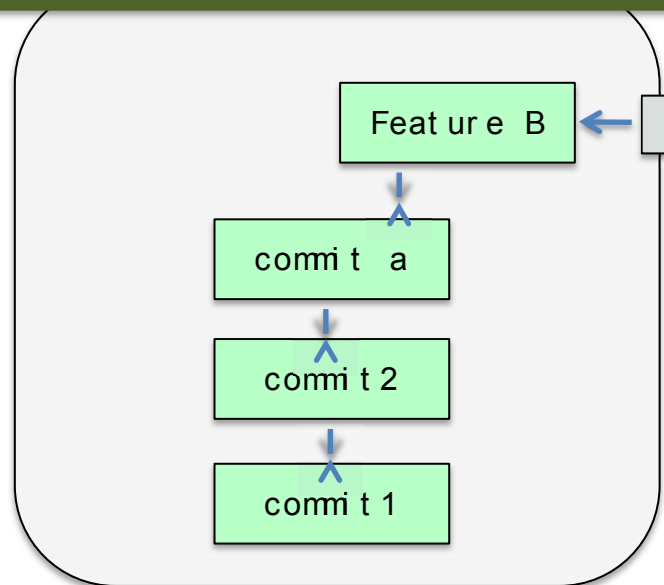
master

6c699cb..04872e1 master -> master



master

origin/master



master

git log

git diff

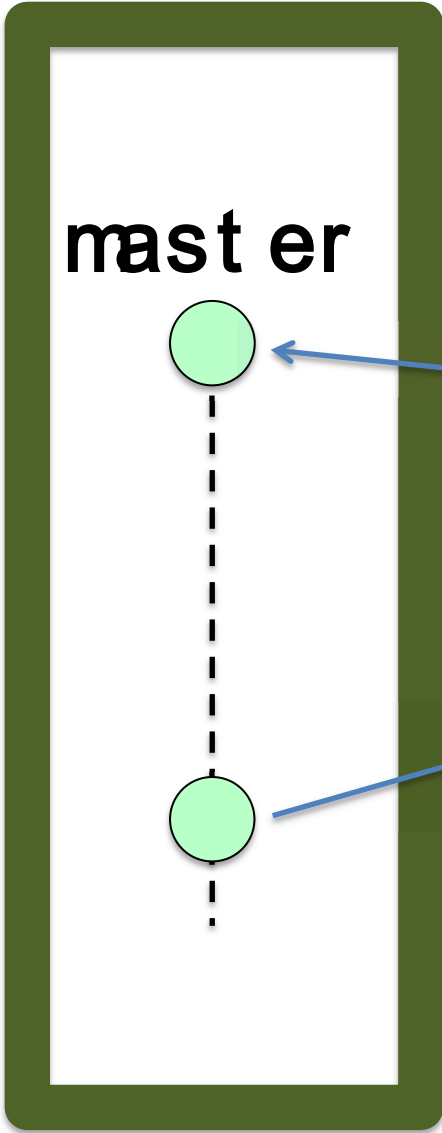
git blame

what

happened?

3.

workflow

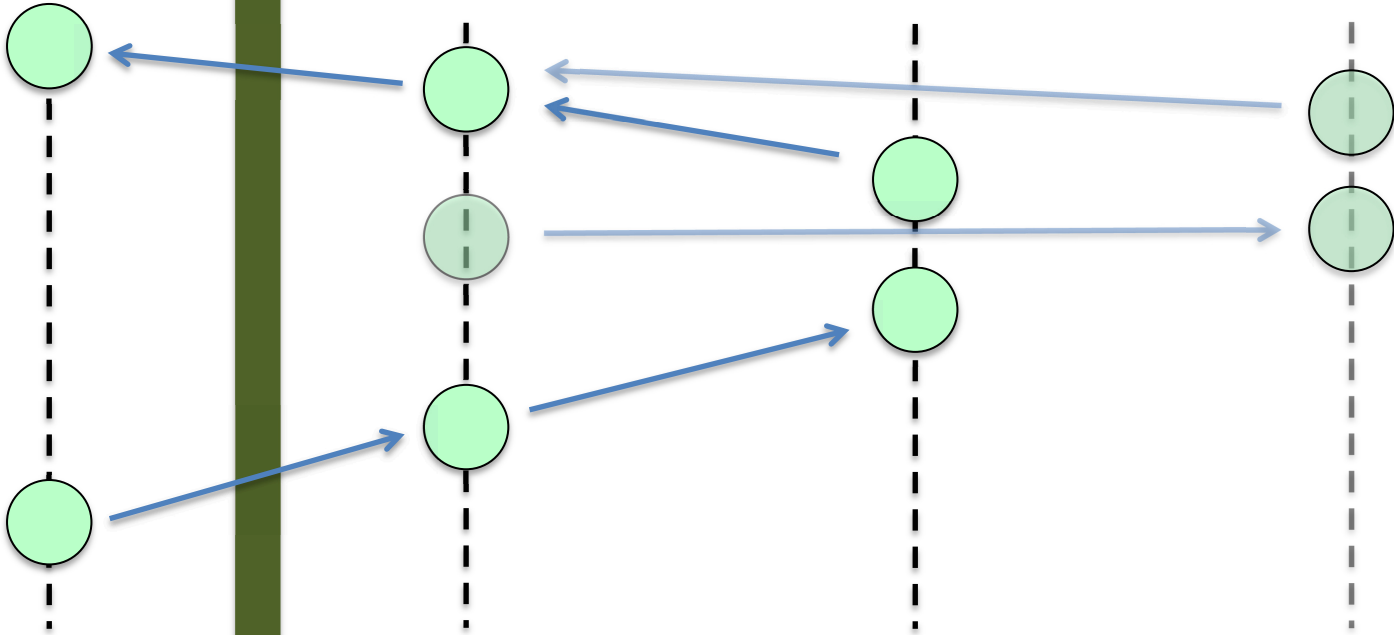


master

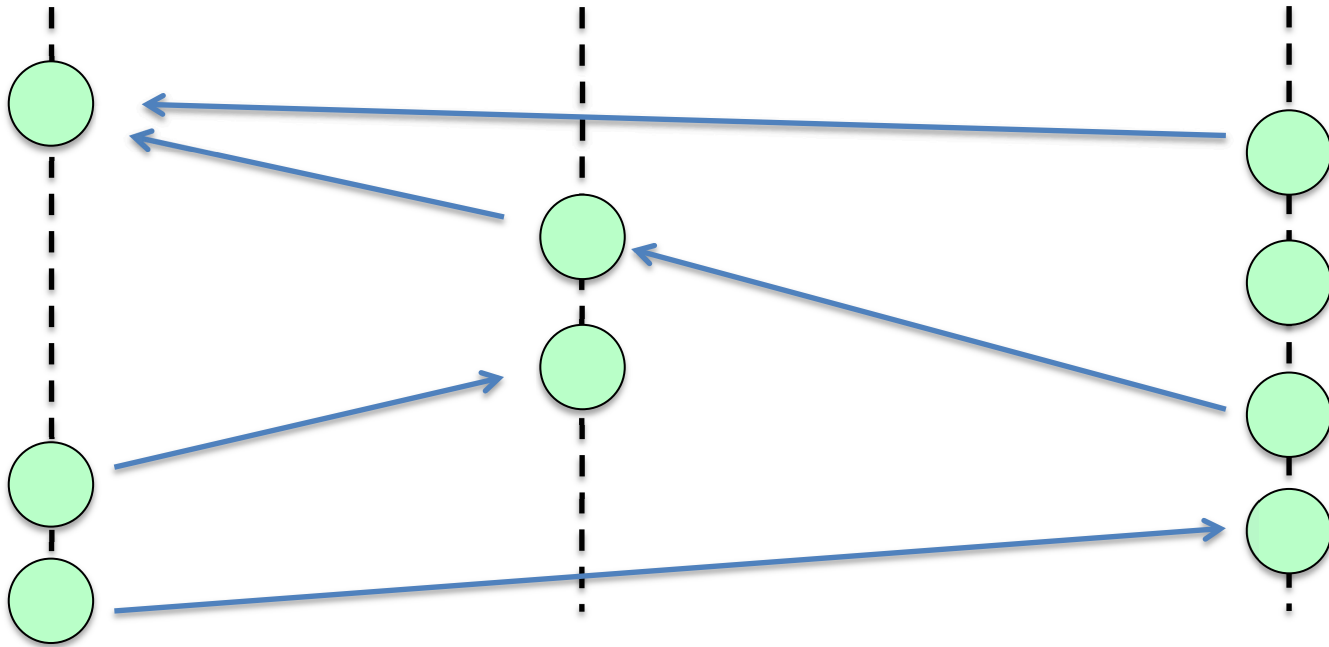
develop

team/w1

team/g1



team w/ | 1- | lead- api | 5- convert - | lead



story branches

Git Workflow Cheat Sheet

0. Create a local branch that tracks the team branch

You only need to do this once. In git there are remote branches and local branches. You can only commit to a local branch, so you will have to create a local team branch that tracks the remote team branch. Like this:

```
git checkout -b team/yy --track origin/team/yy
```

1a. Create a new story branch yourself

```
git checkout -b story/xx team/yy           # create the branch  
git push origin story/xx                  # make the branch available to oth  
git branch --set-upstream story/xx origin/story/xx # make local branch track the remo
```

1b. Someone has already created the story branch

cheat sheet

