

git

Marcos Marado

February 2020

```
$ git init
```

```
# this is a talk for git users, not for those who don't know git
```

no central repository, no “master of the truth”

- this includes your own repositories, gerrit, etc.
- what is a repository? (master, origin/master)

```
git push --force # AKA, I'm doing something wrong
```

commits vs. patches

- commits have commit messages, let's USE them!
 - git log
- let's talk about git format-patch, git am, git amend, git cherry-pick...
- pick between projects
- maintain commit's history!

commits vs. tags

- change vs. code
- branches vs. tags
 - tip: do you want it to be mutable?

commits have a parent...

- Changes have not: Change-Id
- perfect for rebases (gerrit uses it for that all the time!)

merge vs. rebase

Merge:

```
git log --pretty=oneline  
git log --graph --pretty=oneline
```

https://hackernoon.com/hn-images/0*NFscoCQwTuLB8mQu.png

Rebase:

https://miro.medium.com/max/855/1*pzT4KMiZD0FsMOKH-cJjfQ.png

https://miro.medium.com/max/1204/1*bXetvZfHi78_SIWKYriZJw.png

- Use merge in cases where you want a set of commits to be clearly grouped together in history
- Use rebase when you want to keep a linear commit history

HEAD, HEAD~1

- <http://marklodato.github.io/visual-git-guide/checkout-branch.svg>
- <http://marklodato.github.io/visual-git-guide/checkout-detached.svg>

squash!

```
git rebase -i HEAD~4
```

other git commands

git init

git clone (-b)

git add

git rm

git commit (-a, --amend, --date, --author...)

git pull

git push

git fetch # instead of a full git pull

git checkout (-b)

git branch (-a, -D)

git show

git diff (local changes, paths, between commits, branches...)

git grep

git archive

git log (commits, branches, tags, formats...)

git tag

git reset (--hard)

git blame

git bisect

other useful things

- .gitignore
- pathogen

Questions?