Practical intro to git

github.com/david26694/practical-git-intro

David Masip and Carlos Mougan

2020/04/24

Contents

Git and Github

> The key difference between Git and Github is that Git is an open-source tool developers install locally to manage source code, while Github is an online service to which developers who use Git can connect and upload or download resources.

Github is a platform that has Git running on its backend.

You can use Git without using Github.

What is git?

Wikipedia:

> Git is a distributed version-control system for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files.

My view:

> Git is a system to track changes in any set of files.

Why git?

My life before git:

Nombre	Fecha de modificación	Tamaño	Clase
analysis.R	13 mar 2020 11:30	0 bytes	R Source File
analysis_final.R	13 mar 2020 11:30	0 bytes	R Source File
analysis_final_final.R	13 mar 2020 11:30	0 bytes	R Source File
analysis_final_20200303.R	13 mar 2020 11:31	0 bytes	R Source File
analysis_final_20200301.R	13 mar 2020 11:31	0 bytes	R Source File
analysis_final_3.R	13 mar 2020 11:30	0 bytes	R Source File
analysis_final_2.R	13 mar 2020 11:30	0 bytes	R Source File
analysis_final_1.R	13 mar 2020 11:30	0 bytes	R Source File
analysis_4.R	13 mar 2020 11:30	0 bytes	R Source File
analysis_3.R	13 mar 2020 11:30	0 bytes	R Source File
analysis_2.R	13 mar 2020 11:30	0 bytes	R Source File

- Collaboration? 😱
- Git simplifies handling of different versions of a file.
- Nothing is (almost) ever lost, so you can be much more fearless.

Git magic

Track the differences!!

- Change from first to second version
- Change from second to third version

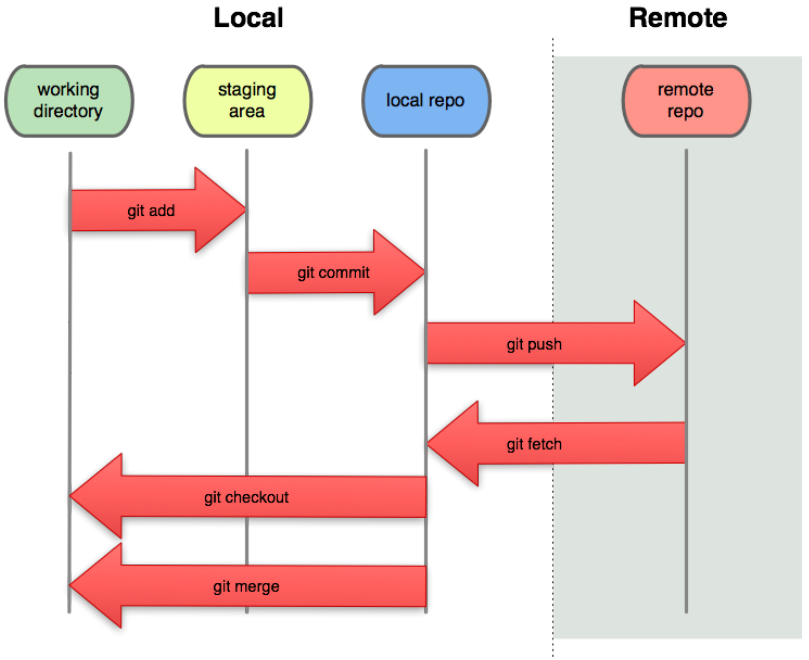
Repositories

A repository is a set of files that is versioned using git.

To create a repo, run `git init`. This creates file called `.git`.

You should also create `.gitignore` file, with specific files to ignore.

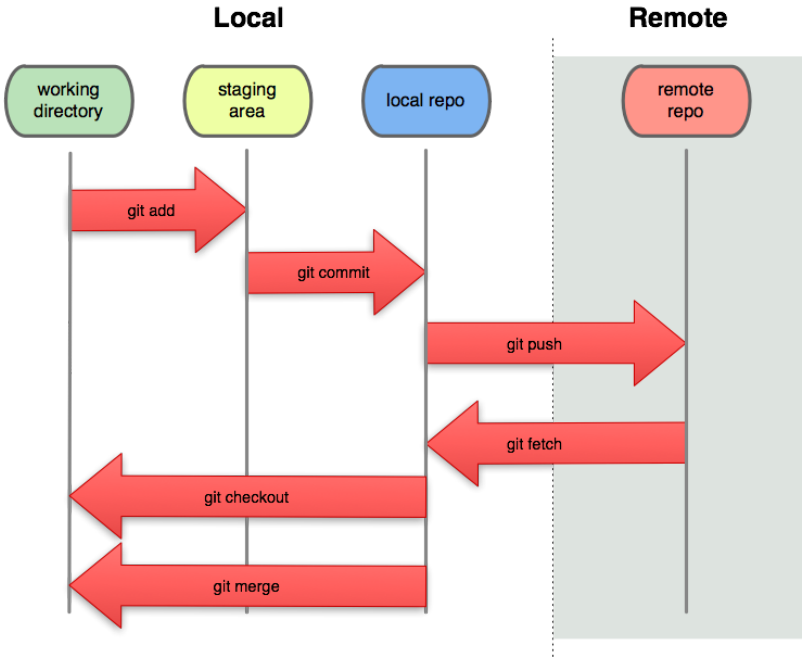
Basic git commands



- `git diff` : Differences between files
- `git status` : Status summary
- `git log` : Shows history of commits

(Demo: add, commit, reset, log, go back to previous commit)

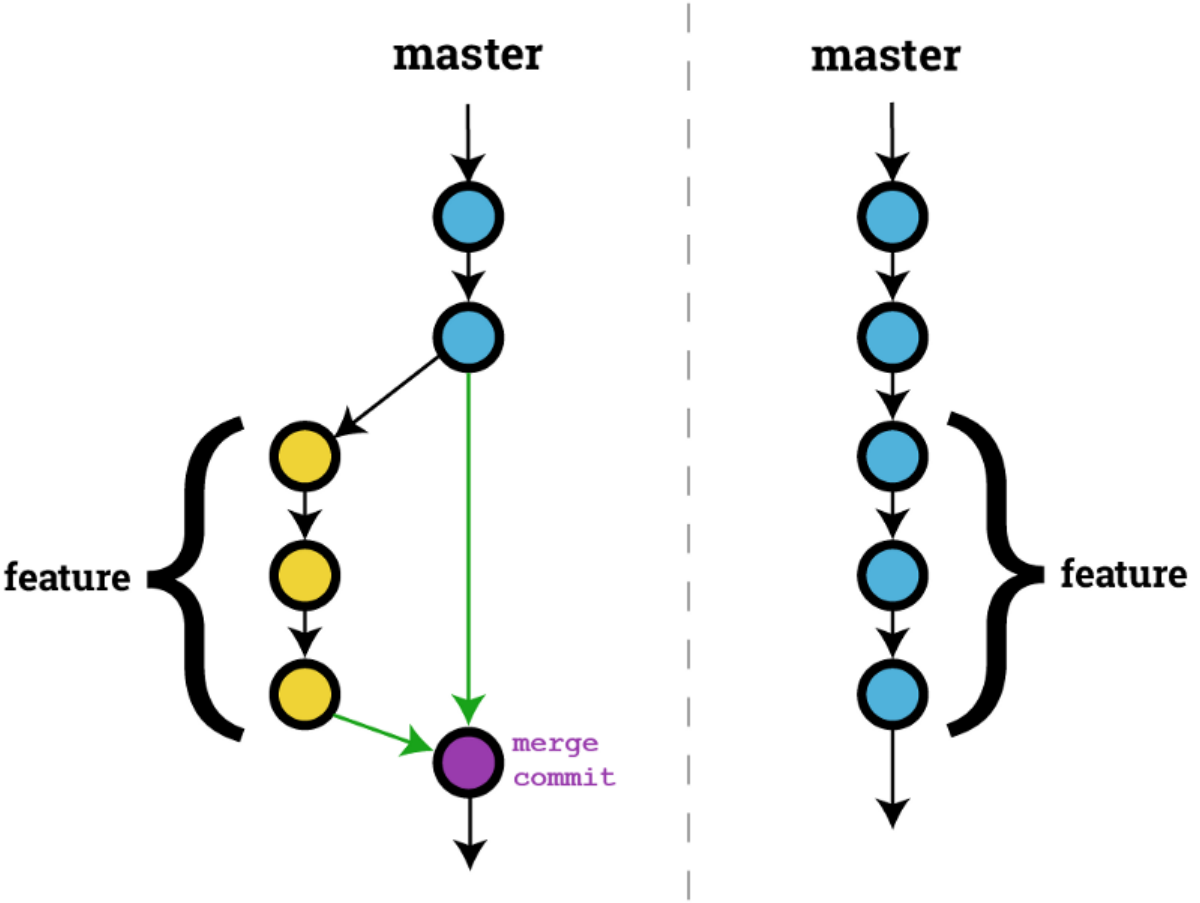
Collaborating (Github)



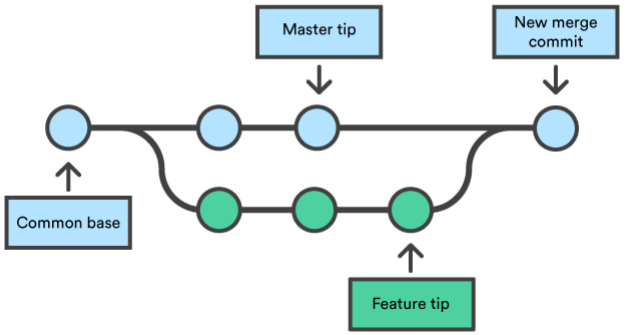
- `git clone el-meu-link`: Copy a github repository in your pc
- `git pull` : Get the version that is in github
- `git push` : Send your version to github

(Demo: push results, change something in repo, pull results, Leo)

Branching



Branching



- `git branch la-meva-branca` : Creates branch
- `git checkout la-meva-branca` : Goes to branch
- `(master) $ git merge la-meva-branca` : Merges branch into master

(Demo: create branch, do some changes, merge it, do more changes, resolve conflicts)

Summary

- Basic workflow: `pull - add - commit - push`
- Intermediate workflow: `branch - checkout - change - checkout - merge`
- Pro workflow: `branch - checkout - change - pull request - merge`

Conclusions

- Git has some barrier to entry
- Git will cause you issues in the beginning
- Git will simplify your life in the long run
- When collaborating, git is almost necessary
- Read about dvc (data version control)

References

Not many, there are 1000 resources to learn git

- https://try.github.io/
- https://stackoverflow.com/questions/4114095/how-do-i-revert-a-git-repository-to-a-previous-commit
- https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging
- https://dvc.org/