Git in 15 minutes

> Luca Barbato - Luminem

What's all about

- Git is a Distributed Version Control System
- It is widely used
- It is surprisingly simple once you get few concepts.
- It has lots of documentation.
  - But you may just rely on cheat sheets
- Commonly you interact with it with CLI tools, but GUI tools are available.
- Integrated platforms such as GitHub and Gitlab are based on Git.

Getting started

- Install git.
  - MacOS: `brew install git`
  - Windows: installer, tortoise
  - Linux: use your distro command.

Key concepts

Git is a Version Control System

- You can store and retrieve snapshots of the documents
  - The snapshots in Git are called `Commits`
- The repository of commits contains all the past history
  - Remote and local repositories are the same
- A `Branch` is set of `Commits`, multiple branches can live in the same repository.
  - It is possible to take commits from a branch to another.
    - It is called `rebasing`.

Key concepts

Git is a Version Control System

- You can `pull` changes from a remote repository or `push` changes to it.
- You can `checkout` a specific point in the history.
- If the files you are working on are textual you can reconcile your local changes with changes present in other branches or repositories.

Key concepts

Git is a set of CLI tools

- There are 5 essential commands
  - `git clone` : to make a local copy of a remote repository
  - `git pull` : to syncronize a local copy from a remote repository
  - `git add` : to add files to a commit
  - `git commit` : to author a commit
  - `git push` : to push changes to the remote repository.
- There are many more
  - `git init` / `git checkout` / `git branch`
  - `git log` , `git rebase` , `git mergetool` , ...
  - ... but we won't need them for now.

Key concepts

Workflows

- Distributed Version Control Systems let you have various workflows
    - Centralized:
        - Everybody pushes change to the same branch in the same remote.
    - Feature Branch:
        - Everybody pushes change to the same remote.
        - New changes are pushed as specific branches
        - A coordinator then rebases the changes in the `master` branch
    - Forking:
        - Everybody has an own public remote repository
        - The new changes are pushed on the personal public remote
        - Merge/Pull Requests are issued

Key concepts

Workflows - commands

- First we `clone` the repository

```
$ git clone https://code.sifis-home.org/example/repo
```

We have now a local repository in `repo`.

- We make our changes

```
$ cd repo
$ vim some_file.md
```

Key concepts

Workflows - commands

- We prepare the commit

```
$ git add some_file.md
$ git commit -v
```

- We push the changes to the remote

```
$ git push
```

- Or we push the changes to a specific branch

```
$ git push origin HEAD:name-of-the-feature-branch
```

Key concepts

Workflows - commands

- Get the new changes

```
$ git pull
```

> **NOTE**: Git will notify if your changes collide, on textual files reconciling is possible, on binary files **NOT**.

Questions?

More

- Cheatsheet
- More Workflows
- Visual Git Reference