

Project Git & GitHubs

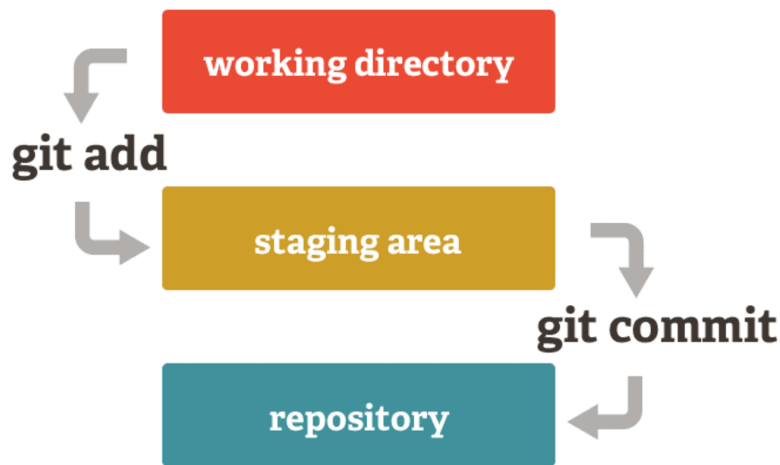
Dictinory:

- Git: the magic timeline, each timeline is reffering to one project.
- GitHub: the backup of my timeline.
- Ungit: visual perception of the local repository
- 4 conceptual area: 3 in my computer and remore repository (Github)

Where is the timeline:

- My computer:
 1. Developing area
 2. Git repository (a local repository): where you save snapshots
 3. Staging area (untracked files): the files that are going to be a part of the next commit, which lets git know what changes in the file are going to occur for the next commit. A channel where you organize your changes and send it to the repository by using "add" comments.
- Developing area: file
- Staging area: `$ git add file`
- Repository: `git commit -m "meaningful message"`

What if we do not use -m? you can do it in the text editor which will be open



How to initialize the git timeline

cd to the developing area, where the .md file exist. Then type "git init", we will see "Initialized empty Git repository"

Saving a point in time

Every commit need to come with a meaningful message

A meaningful messages needs to contain:

- Why was it changed
- How this addresses the issue
- Effects due to the change
- Limitations of the changes

Checking what is where

\$ git status

Why staging before saveing

Working on several files

View the commit history

\$ git log --oneline -> shows the IDs of the commits // **Head is the most uptodate one**

\$ git log -p

It also shows all the differences between the commits

Compare differences between the current and previous one

\$ git log --patch

Traveling in the timeline

\$ git show

\$ git diff

Must-include when creating a repository

1. ReadMe.md: Detaols description of the project and tool usage
2. .gitignore: List of files that should not be added to the repository
 - Data files
 - Backup files
 - Intermedite files

How you can make the .gitignore file:

- Can list one by one
- Can use template (Github)
-