

# Git and Github Dictionary

---

git: a versioning system that keeps track of changes - allows you to go back, allows for collaboration, is locally stored

Github: Where we can access other's files, online repository for your codes (not only)

IMPORTANT: don't do double git init

IMPORTANT

## Commands

---

- initialize timeline: git init
  - add a file: \$ git add File A
  - git commit: final command --> git makes you write a message/ is this meaningful enough
    - \$ git commit -m "meaningful message"
    - meaningful message
      - Why was it changed
      - How this addresses the issue
      - Effects due the change
      - Limitations of the change
    - BE DESCRIPTIVE as possible --> better to have enough information than less
- > SAVES the updates version (snapshot/point in time) in the git repo

## Conceptual Areas

---

1. Developing Area: The working directory
2. Staging Area: \$git add. Dump what you want to commit. Using to prepare snapshot. Gives you some control
3. Local Repository: it is where your snapshots are saved. Locally in your computer --> ls -la command --> .git --> git commit sent things to local repo

## Why command message

This message will later on help you or anyone else that has access to it, to understand the context of changes and updates.

GIT status allows me to check what files are

1. to be staged: you have committed this file or folder before, you have made new changes and git recognises the new changes are not yet added nor committed
2. to be committed; you have committed file before, you have made new changes and git recognises you have added but not committed
3. untracked files

git add regex --> \* : means everything

## Routine

---

1. Git init
2. git add
3. git status
4. git -m "meaningful message"

Create a new repository

1. choose a name for the project --> easier to use same as folder
2. keep it public but private also possible
3. create

## README.txt

---

Github recognises it and shows it at a main page. Introduce project

## .gitignore

---

list of files that should not be added to repo

- Data files
- Backup files
- Intermediate files

limit on what you upload on github! But you want everything to be in the same folder and this can be done by .gitignore. Git and github will recognise this and know this can be ignored --> never added or committed. Everything in ignored file is not backed-up! take care of that yourself even if it is .gitignore --> IMPORTANT

- \*.csv --> ignore all csv files
- !dataset.csv --> ignore only this file