Git

→ Deleting is easy, Creating is hard

→ Snapshots of work at different time stamps

→ Distributed VCS (GIT)

```
git config --global user.name "<your-name>"

  "       "      "     user.email "<your_email>"
  "       "      "     --list
```

```
git add -A                  git commit -m " "        git tag -a v1.0.0
git add <your-file-name>    git log                  git tag -d v1.0.0
                            git log --oneline
git reset                   git log --stat           git tag -a v1.0.0 <SHA>
  git revt <your-file-name>
                            git show <SHA>           git shortlog -n -s
                                                     git log --author=" "
                                                            --grep = ____
```

```
git help <verb>

                e.g.  git help config
git <verb> --help     git config --help
```

```
git clone <url>  <where to clone>

              ↑ . ←——— current
                        working
                        directory
```

mac   win
```
ls -la      dir
```

```
git init
```

```
git remote -v     git pull origin main
git branch -a     git push origin main
git diff
                  git remote remove origin
                  git remote add origin  <remote_repo-url>
```

mac   win
```
rm -rt      rmdir
```

```
git checkout filename
git commit --amend -m " "
```

```
git status
```

```
.gitignore    → add venv/
```

→ to create branches on previous commits.

```
git branch <your-branch-name> <SHA>
git checkout <your-branch-name>
```

→ HEAD is the reference to the most recent commit in the current
  branch. i.e. HEAD is just like a pointer that keeps track of the
  latest commit in your current branch.

```
git push -u origin <your-branch-name>

git checkout main         | git branch --merged |      → git log --oneline --all --graph
git pull origin main
git merge <your-branch_name>                           * Can't delete a branch you are currently
git push origin main                                     checking out, need to delete from other
git branch -d <your-branch-name>                         branch.
git branch -a
git push origin --delete <your-branch_name>
```
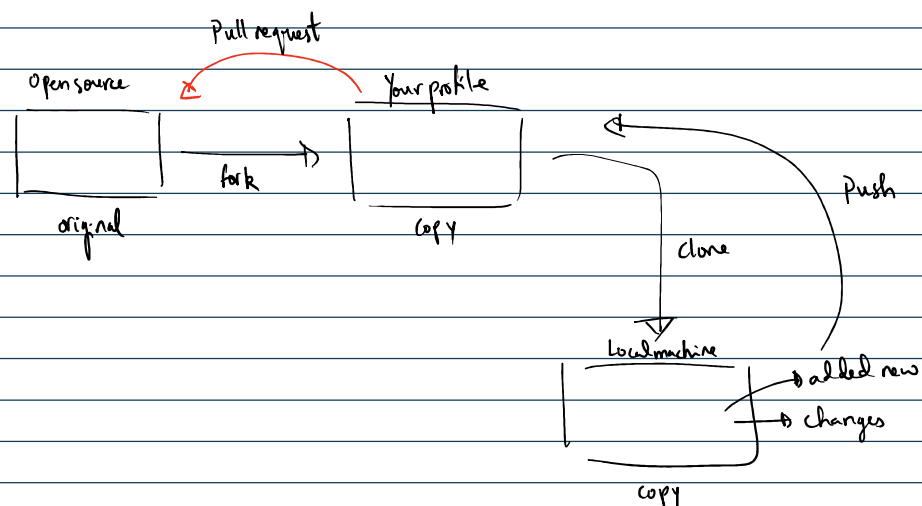
→ Merge conflicts : Different code written on same lines in different
                    branches.

Open Source Contribution

Author ——→ Owner ——→ Maintainers
      (Roles)   ↓
Community Members / Users ←—— Contributors

Pull request

Open source   Your profile

original   fork   copy

Push

Licence
        Contributing.md
(Anatomy)

Documentation   Readme

    code of Conduct.md

clone

Local machine
        → added new
        → changes

copy

Next, it's a good idea to look at the GitHub issues for the project
- look at the existing issues to see if one is similar to the change you want to contribute
- if necessary create a new issue
- communicate the changes you'd like to make to the project maintainer in the issue

When you start developing, commit all of your work on a topic branch:
- do not work on the master branch
- make sure to give the topic branch clear, descriptive name

As a general best practice for writing commits:
- make frequent, smaller commits
- use clear and descriptive commit messages
- update the README file, if necessary