

Full request

- Fork the repository.
 - `git clone forked-your-repo-url`
 - make changes
 - Create a new branch (`git checkout -b name`)
 - add and Commit the changes.
 - `git push origin branch-name`
- Some
- directly make changes to main branch
- `git add.`
 - `git commit -m`
 - `git push`
- local - your system
remote - github

Pulling from remote repo

- `git remote add upstream 'url'`
 - `git pull upstream master.`
- `git pull origin master.`

* => If it shows

fatal: not a git repository (or any parent directory)

- then change the path

Issues:

- Raise an issue by clicking on it.
- You assign to specific members
- milestone in the issue section is used to ~~have~~ create a deadline.
- The page is [here] (.....)

- git log --since = 2 days → filters the
- git log --pretty = format: "%h -- %ae" → It shows the information in the format of hash and author Email.
- git commit --amend → to change the Commit
- git remote add origin "Someurl" → It will create a remote named origin
- git remote → to see the remote in the local repository.
- git remote -v → fetch and push
- git push origin master
- git ~~ex~~ checkout -f → this is used to change the files to the previous Commit
- git push -u origin master → It is push.
- git switch -c new branch-name (Creating a new branch)
- git push origin -u branch-name

[here] (url)

→ Before (git add) it is untracked file.

→ Before (git commit) it is unstaged.

⇒ restore the deleted file by (git rm filename) by

- git reset -- filename (reset the status of filename)
- git checkout -- filename (restore file ~~from~~ in working tree)

Git branches

Shift

→ `git branch branchname` → to Create branch.

→ `git branch` → to see the branches

→ `git checkout branchname` → to change into that branch.

→ `git add .` → to add all the files at a time

→ `git Config --global User-Email email name` → to set the Email

→ `git Config --global user.name Email` → to set the Email

→ `git diff` → it compares the working directory with the staging area.

→ `git diff --staged` → It compares the present staging with the previous Commit stage.

→ `git Commit -a -m "Committed"` → It will Commit all the files in staging area. skipping untracked files.

→ `git rm filename` → it will remove the file and add to staging automatically

→ `git rm --cached filename` → It is used to remove the file from tracking.

→ `git mv old_filename new_filename` → It ~~removes~~ ^{moves} the old file to new file.

→ `git log -p` → It will show all the details of previous Commits with there difference.

→ `git log -p 3` → It will show all the previous 3 Commits with there difference

→ `git log --stat` → It will show all the Summary of the ~~any~~ changes

→ `git log --pretty = oneline` → It will show all Commits in one line

→ `git log --pretty = short` → It will show the short information.

→ `git log --pretty = long` → It will show the long information.