# Starting

\*\*\*\*\*Make sure you have installed Git in your Machine and VS code will help a lot\*\*\*\*\*\*

Let’s configure git in local machine:

* git config --global user.name “User Name”
* git config --global user.email “email@gmail.com”
* git config --list (for checking if everything okay or not)

Some basic operating system commands:

* “cd ..” for navigating previous folder
* “cd nameoftheFolder” for moving forward
* “ls” for seeing normal files in current folder
* “ls -a:” for seeing both hidden files and normal files in current folder

# GitHub to Local repo

\*\*\*Git commands for Cloning a Repo from remote and working on local machine:

* clone: for making a copy of existing repo from GitHub  
  example: “**git clone link**”
* status: for checking do you have changed anything of the repo in our local machine   
  example: “**git status**”
* add (When our task is ready to commit): for adding new file or files in our local repo  
  example: “**git add filename**” (Here we can choose which file we want to add)  
   “**git add .**” (for adding every file we have in our repo)
* commit (when we want to change our repo): after this command our local repo will be saved in our Git with a commit message and now this will be the final repo. Make sure commit message containing a meaningful message.  
  example: “**git commit -m “commit message”**”
* push: when want to change our remote repo which is in our GitHub according to our local repo which is in our machine.  
  example: “**git push origin main**”   
   OR “**git push -u origin main**” if we want to push again in this repo also in main branch then we need to just write “**git push**” only.
* Pull: if we want to update our local repo according to remote repo  
  “git pull origin main” which will make our local repo up to date by downloading remote files.

# Local repo to GitHub

\*\*\*Git commands when we want to make our normal file into Git repo with push in GitHub:

* step 1: check is this folder already git folder or not   
  “ls -a” if it shows .git then it is already a git folder
* step 2: otherwise make the folder a git folder  
  “git init” and check again “ls -a”
* step 3: add files in your Git  
  “git add .” or “git add filename”
* step 4: commit your changes  
  “git commit -m “commit message””
* step 5: Go to GitHub and then make a new repository as your project name is. Keep in mind that for this time you have to ignore readme.md file because you don’t have this file in your local repo.
* Step 6: copy the repo link and run  
  “git remote add origin linkoftherepository” we can verify by “git remote -v”
* Step 7: check the branch. If it is already in main branch then keep it otherwise  
  “git branch” for checking the branch  
  “git branch -M main” for renaming our branch to master or others to main
* Step 8: Now we are ready to push our local repo to GitHub  
  “git push origin main”

# Branch concept

**Branch in Git:** Here we make copy of main branch and work on it after that we merge it with main.

* “git branch” for checking current branch
* “git branch -M main” for changing branch name as main
* “git checkout -b <branch Name>” for making new branch
* “git checkout <branch Name>” for navigating
* “git branch -d <branch Name>” for deleting a branch
* “git push origin <branch Name>” for pushing changes into the branch

**Merge branches:**

Way 1(using terminal):

“git diff <branch name>” which will show us what are differences in them

“git merge <branch name>" to merge changes

Way 2(using PR or Pull Request):

Go to github and then Compare and Pull then made a request

# Undo changes

**Case 1:** when we have added changes but not committed

“git reset <file-name>” for resetting only a particular file

“git reset” for resetting every file and it will take your files before add condition

**Case 2:** when we have committed the change in our local machine

* For going one commit back:  
  “git reset Head~1” Head basically refers to latest commit and we are going one commit back from head.
* If we want to go in specific commit:  
  “git reset <commit hash>” we can get commit hash by running “git log” where we will see our all commit with hash.
* If we want to change our editors code by new repo condition:  
  “git reset --hard <commit hash->”

# Fork

Basically, by fork we make copy of original repo in our GitHub. If we make anything change and want to merge them with original repo then we can make a pull request and that’s it.