Section 01 → Git, Github and Version Control.

① Version control → Control the project, which update to use in it which not to, whether to edit (or) not to i.e simply controlling version of your project.

② To initilize type git init. → ls, ls-a } Seeing current files.

before this touch chap1.txt
open chap1.txt.

③ → git status → Gives files in the current directory.

→ To make changes in file, first have to add it to staging area.

→ Files in the dir, but not added to staging area are called untracked files.

④ → git add chap1.txt. (to add to staging area).

After this type git.commit -m"message of changes made"

↳ commits everything in staging area.

⑤ Now after commiting (one (or) several times)

You can use git log to see all changes made.

⑥ If you want to add everything to staging area, it's tedious to do one by one.

∴ git add . } Adds everything in current dir to staging area.

7) git diff chap1.txt → Show you all commits & diff made to that file.

8) git checkout chap1.txt → Checkouts the latest commit & makes file as it was when 2nd last commit happened.

9). Now look at GUI Github.

» Can contribute to opensource here. (to other people).

** » All that happened till now was using git to make changes to your file locally.

» To put all the files in github, we use » push.

» git remote add origin (URL of our repo) → No brackets.

git push -u origin master

→ Name of branch.

→ Name of remote (default origin) in above line.

10) Gitignore.

» Some files you don't want to put on github.

» ~~touch~~ .gitignore → Used to ignore file (P.T.O).

→ makes file invisible in gui. can be visible by → ls -a

» ** To remove from staging area. (all files).

git rm --cached -r . → everything in current dir

↓ rm → remove. ; -r → recursive

Github has a gitignore repository. Checkout if needed

(11.) Open .gitignore

Type name of files you want to hide/ignore.

» Eg:- chap1.txt.

(12) Then when you commit whole dir

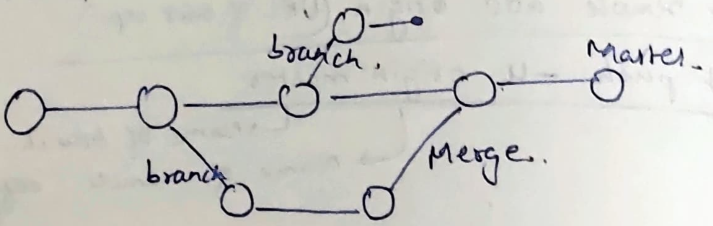
→ These files won't be staged (or) committed.

(13) <u>Cloning, Branching & Merging.</u>

<u>Cloning</u> → Copying a repo from Github to your computer to work on it.

→ git clone (URL of repo) [better than downloading zip file.]

<u>Branching & Merging</u>. →



→ git branch Name } to create new branch

→ git branch → to see all branches.

** → git checkout alien-plot } Switches from master branch to other branch (specified).

• <u>To merge</u>

→ git merge (name of br. to merge) [be in master branch while doing this.]

Showed GUI way too.

(14) Fork and Pull Requests

* Pull requests → Requests to make changes to the main/ master branch.
  → Master decides to accept it or not.

$\boxed{\text{git fetch.}}$ extra.

* Fork → Cloning in Github to comp.
  Fork in Github to your Github profile.

Shown how to do in GUI.
