Sahil Deshpande

Mo | Tu | We | Th | Fr | Sa | Su

Memo No. ________

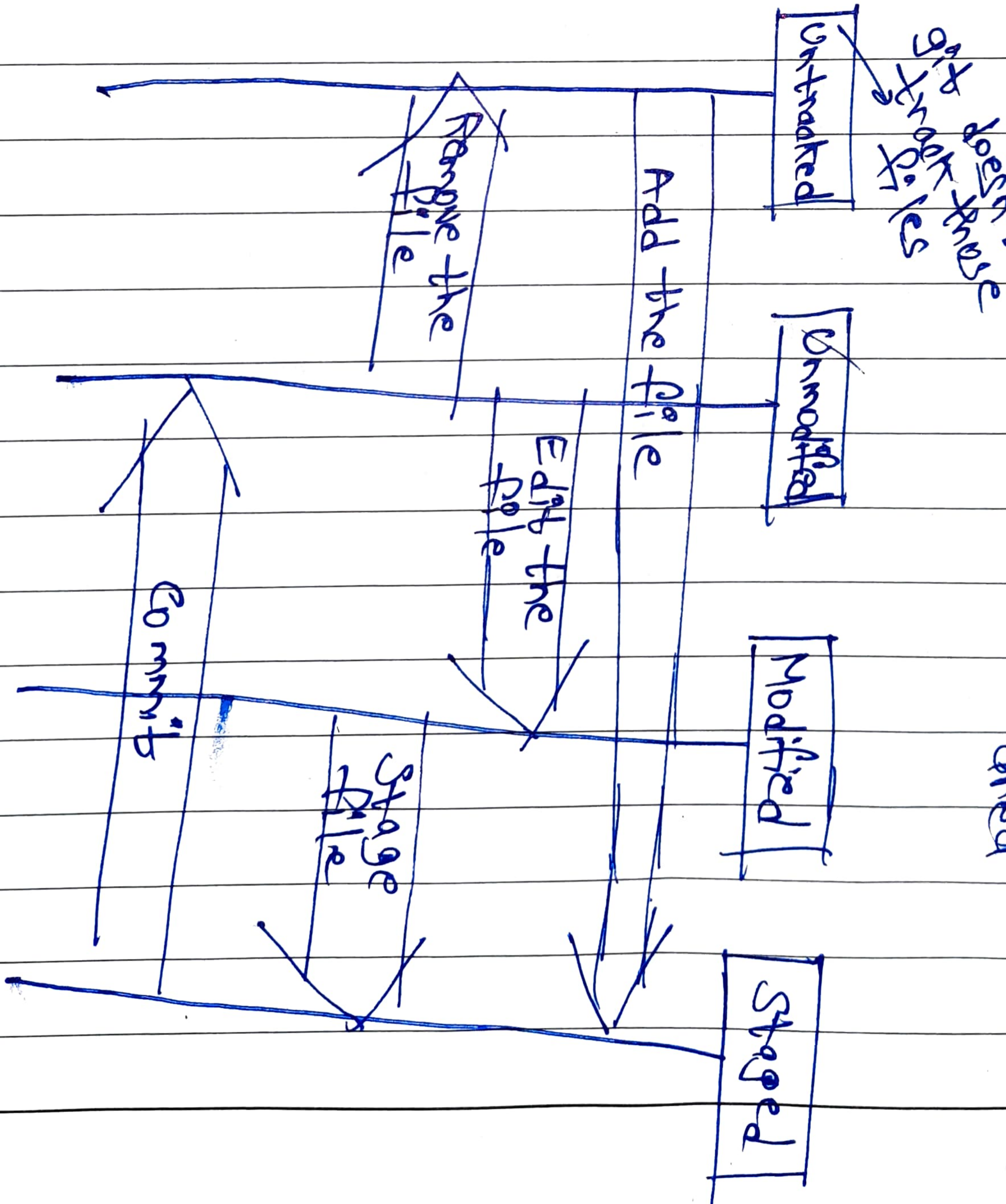
Date   / /

Github (Harry)

1) Git is a distributed version control system means everyone has a copy of that repo

2) git init → empty repo

3) ls -lart → shows hidden folders → .git folder spotted



Commit → Snapshot of file system

git add file ⇒ file enters staging area

→ To add all files to staging area, use git add -A

→ This command stages all changes (new files, modifications, deletions) in ENTIRE repo

→ git add . stages only new & modified files but NOT deletions, only in CURRENT directory & its subdirectories.

<u>Eg</u> /project → Curr Dir
⊢ index.js
⊢ styles) → /project/styles is sub Dir

→ If a file is modified, after being staged, staging area contains its old value before modifying

→ Say file → contact.html, is modified by ur friends, saved ⇒ To recover prev content use git checkout contact.html) → matches working dir content with prev commit

→ git checkout -f matches all files with prev commit

→ git log mentions all commits

→ git log -p -1 mentions all changes
1 → no of commits u want to see

→ git diff compares working dir with staging area

→ To compare staging area with last commit,
git diff --staged

→ If u are working on just specific files & separate commit msg not req. for them & use git commit -a -m "...." → use git log to verify

→ To remove a file from staging area,
git rm --cached file, so that it becomes untracked, but is still present on harddisk

→ git rm file → file is removed from harddisk as well

→ git status -s
1st box → staging area
2nd box → working dir

→ used for summarized status

1) 3 files were already comitted

2) All 3 modified & saved in editor now

3) git status -s → 2nd M ✓

4) git add contact.html → M — contact.html
Curr copy added to staging area
→ Now working dir unmodified, but since staging area & commit dont match, Modified
— M index —
— M monuments

5) Again change contact.html

6) Now it is modified in working dir + staging area

7) git add contact.html

8) Now only staging area me modified.

9) .gitignore → make it using terminal only or else windows → confuse X

Github Notes (Continue)

→ To ignore mylogs.log, simply add it in .gitignore

→ ———— all .log, add *.log

→ ———— only mylogs.log present in root ⇒ /mylogs.log

→ ———— a folder say ignore ⇒ ignore/

→ Create a branch w/o moving to it → git branch <name>

→ To shift, git checkout <name>

→ To merge changes in a branch with main, git merge <b-name>

→ git checkout -b <b-name> to create & shift

→ remote is a url where we host our repo, git remote add origin means short name of that url is origin for fetch & push

→ git remote ⇒ o/p → origin & git remote -v mentions url

→ git push origin main ⇒ push main branch on origin

→ If repo is private, u can't directly push, since github shows that this repo DNE, hence to obtain its read-write access ⇒ Settings → SSH keys

→ SSH key is used to give access of github a/c to a local machine

→ git remote set-url origin <link> → To change url

→ git push -u origin main sends your local main branch's changes to main branch on remote repo named origin OR if it DNE it'll create branch & upload contents

→ Pull request is made to a repo owner to change code in that repo

→ git push, pushes change to that branch, where -u flag was used to push contents.

→ To clone contents in harry folder, git clone <url> harry