Git And Github

Commands to follow by order (after creating a file and locating to that folder)

1. git init
2. git add .
3. vi ‘file name’
4. git commit -m “message”
5. git remote add origin ‘link to repository’
6. git push origin master(master is the branch name by default)

Commands for forking and cloning

** git branch ** to check the current branch you’re working on

Always create a new branch for a new commit i.e a new branch for a new feature

One branch = One pull request

1. git clone ‘cloned url’
2. git remote add upstream ‘url of main file’ (to the main file; not the forked one)
3. IN CASE, we’ve to change this upstream url after setting it(i.e by mistake kuch aur ho gya krna kuch aur tha), then use git remote set-url upstream ‘url’
4. git branch ‘name of branch’ (always do this for creating cloning)
5. git checkout ‘name of branch’ (all the commits that are made now will go to this new branch)

▼ Removing a commit:

1. git log: to get the log of commits
2. git reset ‘commit no’: to reset
3. git status: to check the status

4. git add .
5. git stash
6. git push origin ‘branch name’ -f : to push the branch

▼ Making forked project even with main project:

1. Can be done through site by clicking on fetch upstream also

▼ Through codes it goes like

1. git checkout ‘main branch’
2. git status: to check the status
3. git log: to check the log
4. git fetch —all —prune (all: to fetch all branches, prune: to fetch even the deleted ones)
5. git reset —hard upstream/main
6. git push origin main

1. Or it could be done in one command : git pull upstream main

▼ Squashing commits (combining many commits into one)

1. git log: to check the log of commits
2. git rebase -i ‘commit id’: to enter the rebase kinda section
3. You are given many options there, basically we’ve to do ‘pick’ and ‘squash(s)’
4. Pick(pick) the one in which we’ve to add, Squash(s) the rest
5. The squashed commits will go into the one that we picked above it
6.

▼ Merge Conflicts and How to resolve them

Commands

1. ls for listing
2. ls -a for listing hidden files
3. ls .git for showing what’s inside the git file

4. git status for previous commands
5. git add for adding a file(for commiting purpose) - Summon the guests on stage
6. git commit -m “message” - click their photo
7. cat ‘file_name’ to display the contents of file
8. git remote add origin ‘ link to the repository’
9. git remote -v: to check the commands available(not sure rn)
10. git push origin master :
11. git restore —stage ‘file_name’ to remove the file without editing - Summon the guests and remove them from stage without clicking their photo
12. rm -rf ‘file_name’ to remove file from git status
13. git add .
14. git stash for saving the current changes but also bringing the old file back - Telling the guests to go backstage and only come when called
15. git stash pop to bring back the latest changes i.e the ones removed by stash command - Telling the guests to come front stage i.e they are called now
16. git stash clear to remove the changes made to the backstage ones
17.

Vim Mode(editing mode)

1. vi ‘file_name’ to enter editing mode

Exit Vim Using a Shortcut Key

In addition to command mode, Vim also has the option for shortcut keys:

- To save a file in Vim and exit, press **Esc** > **Shift + ZZ**
- To exit Vim without saving, press **Esc > Shift + ZX**

More Command Options to Quit Vim

Here’s a list of commands for quitting Vim:

- `Esc` – switch to command mode
- `:w` – write out changes that were made
- `:q` – exit Vim
- `:q!` – exit Vim and discard any changes
- `:wq` – saves the changes, and exits Vim
- :`x` – save the changes made, and exits Vim