

## BrSOC

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

- 1) Organization announcement → Late February
- 2) Student discuss idea = (Feb - Mar)
- 3) Submit proposals (Late Mar - Early April).
- 3) Contributors announced (Early May).

### Skills

1) Learn a language.

→ Javascript (beginner to intermediate)

→ TypeScript (Intermediate to advanced)

→ Java (most used at

and most difficult to learn)

→ C/C++ (beginner to intermediate)

→ Python (beginner to intermediate)

2) Git and GitHub, Terra (Version control system)

3) Open Source knowledge.

(Remote ← local) ⇒ push

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

## Git and Github

Git is a version control system.

- `git config --global user.name "Basu Kallapur"`
- `git config --global user.email "basukallapur.0@gmail.com"`
- `git config --list`

→ local ⇒ laptop / PC

Remote → Github

to clone from Github to laptop

- `git clone link`. link → https when use rare
- ~~cd~~ `cd` wr tab → change directory
- ~~cd~~ → `ls` → show file list.
- `ls --a` → show hidden files.

→ `git status` ⇒ shows state of git

### Type

1) Untracked

→ new files that git doesn't yet track

2) modified

→ changed

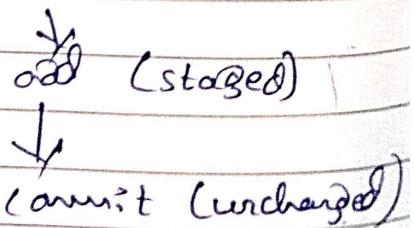
3) Staged

→ file is ready to be committed

4) Unmodified

→ Unchanged.

Change / new file  
(modified)      (untracked)



(②) `cd ..` → to come out of directory.)

~~mkdr~~ `mkdir LocalRepo` → to create new folder

classmate

Date \_\_\_\_\_

Page \_\_\_\_\_

### Add and commit

→ `git add filename` → to add file to staging area

→ `git add .` → to add all files at once.

→ `git commit -m "some message"` → to commit

Push command: upload local repo content to remote repo.

→ `git push origin main`

### init command

→ `init` → used to create a new git repo

→ `git remote add origin <link>`

→ `git remote -v` → (to verify remote)

→ `git branch`

(master ⇌ main)

### Branch

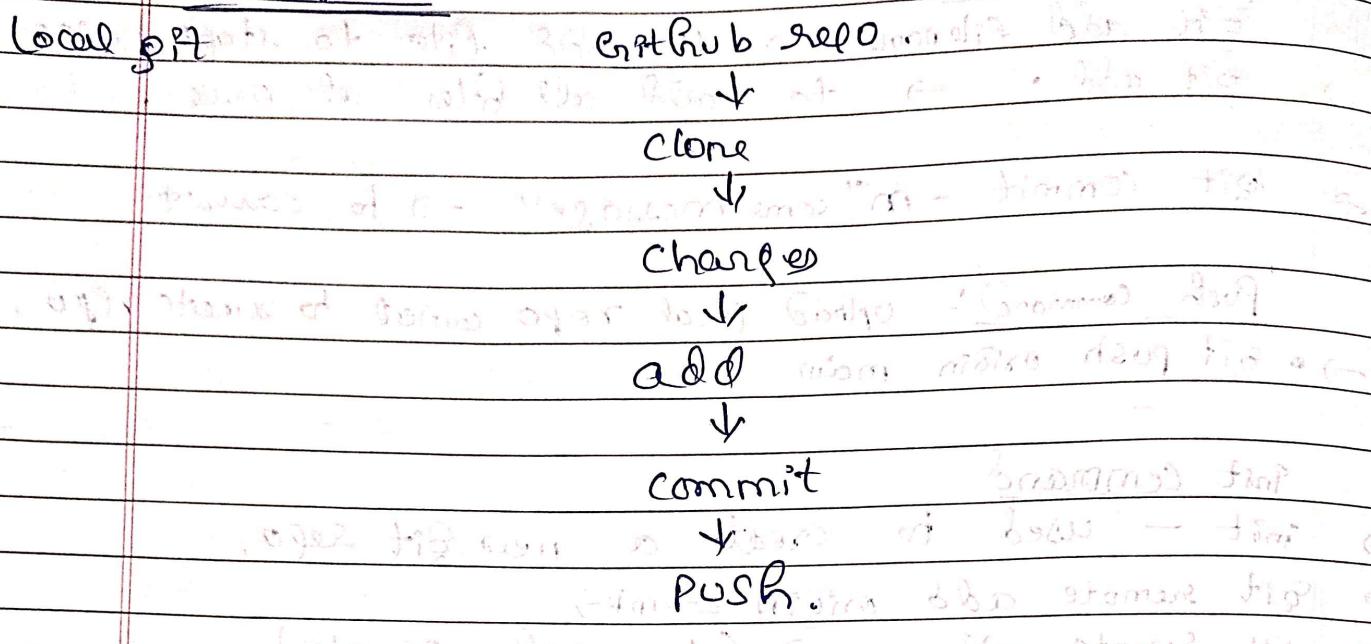
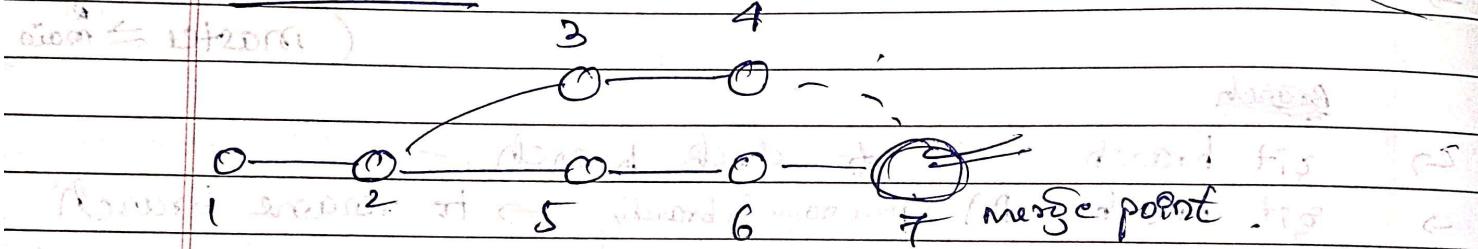
→ `git branch` → to check branch

→ `git branch -M <newname for branch>` → to rename branch

→ `git push origin main`

① → `git push -u origin main` → { if we want work on these  
for long time,  
(Instead of writing origin main  
again and again),  
(set upstream)

hint: Whenever we are doing project, repos, it's better first  
mostly try to do in GitHub (start in GitHub), then clone in  
local system. (which is easier).

work flowCat Branchesbranch commands

- `git branch` → to check branch.
- `git branch -M main` → (to rename branch)
- `git checkout <branch name>` → to navigate.
- `git checkout -b <new branch name>` → to create new branch.
- `git branch -d <branch name>` → to delete branch.

Note! - if you are in feature2 you can't delete feature2.

main → button  
feature → dropdown  
CLASSMATE  
Date \_\_\_\_\_  
Page \_\_\_\_\_

## merging branch

way 1

- `git diff <branch name>` → (to compare commits, branches, files and more).
- `git merge <branch name>` → to merge 2 branches.

way 2 is by using `github`.

→ Create a PR (pull request)  
→ pull request: it tells you what you tell others about changes you have pushed to a branch in a repository on `github`.

## Pull command

→ `git pull origin main` → used to fetch and download content from a remote repo and immediately update the local repo to match that content.

Resolving merge conflicts: an event that takes place when `git` is unable to automatically resolve differences in code between two commits.

current change → feature 1 (new branch)  
incoming change → from ~~main~~ main.

→ `git log` → to check all commits.  
and press q to quit.

### Undoing changes,

case 1 → staged changes (at .) → command is `git reset`.

→ `git reset <filename>`

→ `git reset -` (at .) → unstage after `git add`.

case 2 → Committed changes (for one commit)

→ `git reset HEAD~1`

case 3 → Committed changes (for many commits)

→ `git reset --commit hash` → at first it's temporary.

→ `git reset --hard --commit hash` → is already

### fork

- A fork is a new repository that shares code and visibility setting with the original "upstream" repository.

fork is roughly copy + change without base.

host?

iva accept

blush da. blous of its needs aqbek. with now no  
it's not asserted also in snapshot writer who

Forward and backward → same busin

direction used on system domain