Git

1. Git ki?
   Ans: version control system.
2. Github ki?
   Ans: remote repository rakhar jayga.

Basic Commands:

1. Directory;te ki ki folder ba file ache ta dekhar jonno command= **ls**
2. Nirdisto akti directory'te jawar jonno: **cd folderName**
3. Jodi directory'te kono hidden folder thake, ta dekhar jonno = **ls -a**
4. Folder create korar jonno command= **mkdir folderName**
5. Bortoman Directory theke ber houar jonno= **cd ..**
6. Kono folder delete korar jonno= **rmdir /folderName.**
7. File create korar jonno = **echo > file.extension**
8. File create kora hoye kina = **ls text.txt**
9. File delete korar jonno = **del file.extension**
10. Fill view korar jonno = **type text.txt**
11. Kono file open korar jonno command = **start text.txt**

Git Setup and Configuration:

Step-1: Download git ➔ https/git_scm.com/

Step-2: Install

Step-3: To check git version ➔ git –version

Step-4: username setup ➔ git config --global user.name "aisajid"

Step-5: email setup ➔ git config –global user.email aisajid13@gmail.com

Step -6: sob kichu setup hoyeche kina dekhar jonno: git config --list

[Note: username and email change korar dorkar hole, abar username and email change korlei hobe]

➔**Commands:**

1. Git a set kora username dekhar jonno = **git config user.name**
2. Git a set kora email dekhar jonno = **git config user.email**
3. Local configuration dekhar jonno= **git config --local --list**
4. Global configuration dekhar jonno= **git config --global --list**
5. System configuration dekhte chaile = **git config --system –list**
6. Username unset korar jonno = **git config --global --unset user.name**
7. Email unset korar jonno = **git config --global --unset user.email**

Now, Global configuration kon jaygay ache ta dekhar jonno:

➔ls –a

➔Type .gitconfig

Globally ja config kora hoy, ta ai .gitconfig file er vitorei thake.

SSH Key;

Amar pc'te public ba private ssh key toiri kora ache kina seta dekhar jonno:

- ⇨ mkdir .ssh
- ⇨ ls -a
- ⇨ cd .ssh
- ⇨ ls
- ⇨ ssh-keygen -o -t rsa -C aisajid13@gmail.com
- ⇨ enter key and press enter
- ⇨ enter
- ⇨ enter
- ⇨ ls
- ⇨ code id_rsa.pub
  [open .sshkey in vscode]

Ssh key generate korar jonno .ssh fil a dukte hobe, tai check kore nilam .ssh file er vitore kichu ache kina

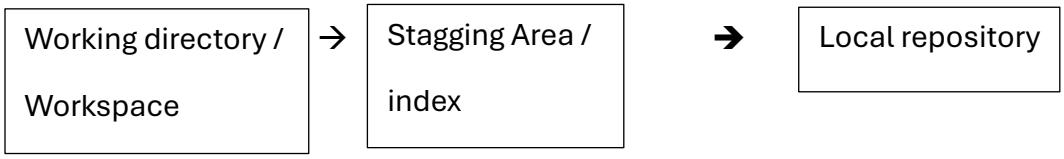
To check keys are generate or not

1. Er por vscode theke code'ta copy kore github a rakhbo, then setting theke SSH and GPG keys te jabo, tarpor new ssh keys- title "My desktop", then key option a key'ta paste korbo. ➔ Add ssh key ➔user sequrity key.
2. Ekhon sei repository'ta copy korte chacchi oitar ssh'ta copy kore nibo.

Now, cmd'te

- ⇨ git clone copy_kora_repository_url'ta
- ⇨ yes
- ⇨ yes

Create git folder:



Working directory:

Now in CMD ():

- ⇨ mkdir notes
- ⇨ cd notes
- ⇨ ls -a
- ⇨ git init
- ⇨ ls -a
- ⇨ echo > day1.txt
- ⇨ start day1.txt
  [write some text on day1.txt file, and save]
- ⇨ ls -a
- ⇨ git status

for practice:

- ⇨ dir
- ⇨ ls
- ⇨ cd Desktop
- ⇨ ls
- ⇨ mkdir notes
- ⇨ ls
- ⇨ cd notes
- ⇨ ls -a
- ⇨ git init
- ⇨ echo > day1.txt
- ⇨ start day1.txt

[write some text on day1.txt file, and save]

- ⇨ git status
- ⇨ echo > day2.txt
- ⇨ start day2.txt

[write some text on day2.txt file, and save]

- ⇨ git status

Git Stagging and unstagging:

1. **git add filename** = stagging area'te newar jonno command
2. **git add -A** = directory and subdirectory er sob file stagging area'te newar jonno .
3. **git add .** = directory er sokol file staggin area'te chole asbe, but subdirectory jabe na.
4. **git add *js** = directory er joto .js extension er file ache sob gulu stagging area'te chole jabe.
5. **git add **/**.js** = directory and subdirectory er sob .js extension file gulu stagging area'te chole asbe.
6. **git diff** = file a ki ki poriborton aseche seta dekhar jonno command.

Now, CMD-

- ⇨ ls
- ⇨ cd notes
- ⇨ ls -a
- ⇨ git status
- ⇨ git add .
- ⇨ git status
- ⇨ start day1.txt

[add some text on day1.txt file, and save]

- ⇨ git status
- ⇨ git add day1.txt
- ⇨ start day2.txt

[add some text on day2.txt file, and save]

- ⇨ git status
- ⇨ git restore day.txt

Amra jei change ba modify kortechi seita jodi ager jaygay niye aste chai

Unstage korar jonno:
In cmd:

- ⇨ git rm –cached day2.txt
- ⇨ git status

amra jodi day1.txt file e aro kichu text add kori..

- ⇨ git diff

For practice:

*Create git folder

* add files

* Add data

* stage the files

* Modify the files and check the differences

* restore the files

* track the Modified data

Solve (in cmd):

- ⇨ mkdir my-notes
- ⇨ cd my-notes
- ⇨ git init
- ⇨ ls -a
- ⇨ echo > day1.txt
- ⇨ start day1.txt
  [writing something on day1.txt file]
- ⇨ git status
- ⇨ git add day1.txt
- ⇨ git status
  [again, add some text on day1.txt file]
- ⇨ git status
- ⇨ git diff
- ⇨ git status

- ⇨ git restore day1.txt
- ⇨ start day1.txt
  [again, add some text on day1.txt]
- ⇨ git add day1.txt
- ⇨ git status

Local Repository (git commit and uncommit):

Ai massage ta obosshoi clear and understandable hote hobe

1. git commit -m “massage here”
2. git commint -am “massage here” = er maddhome amra akoi sathe staggin and commit korte pari.
3. git log = commit er history dekhar jonno.

Now , In cmd:

- ⇨ git status
- ⇨ git commit -m “day1 data is added”
- ⇨ echo > day2.txt
- ⇨ open day2.txt
  [writing some text on day1.txt file]

- ⇨ git status
- ⇨ git add day2.txt
- ⇨ git commit -m “day2 data is added”
- ⇨ git status
- ⇨ git log
- ⇨ echo > day3.txt
- ⇨ start day3.txt
  [writing some text on day3.txt file]
- ⇨ git add . && git commit -m “day2 data is added”.

Jodi akoi sathe stagging and commit korte chai

Uncommit:

1. git rest --soft HEAD^ = recent commit’take undo korar jonno
2. git reset HEAD = aita korle recent commit’ta uncommit hoye jabe and setake stagging area theke remove kore dibe.
3. Git reset -- hard HEAD^ = amader commit ta remove hoye akebare Prothom stage e chole jabe
4. git reset --soft HEAD~2 = ekebare duita commit undo korte chaile.

5. git reset --soft HEAD~3 = ekebare 3ta commit undo korte chaile.

CMD:

- ⇨ git log
  (clone diye 1 dilei aita thek quit hoye jabe (:q))

- ⇨ git reset --hard HEAD^
- ⇨ git log
- ⇨ git reset --soft HEAD^
- ⇨ git status
- ⇨ git log
- ⇨ git commit -m “day2 data is added”
- ⇨ git status
- ⇨ git log
- ⇨ git reset HEAD^
- ⇨ git status
- ⇨ git add . && commit -m “day2 data is added”
- ⇨ git status
- ⇨ git log.

Git commit best practice:

Note: commit massage always 50 character er vitore likhte hobe.

In vscode terminal:

- ⇨ git add index. html
- ⇨ git commit -m “initial commit”
- ⇨ git add.
- ⇨ Git commit -m “Add navbar feature” -m “-Implement navbar feature” -m “-Add home, about and contact option for navigation.”
- ⇨ git log --online
- ⇨ git log

rules:

1. commit massage clear and concise (max 50 character)
2. Commit often, but not too often
3. Imperative mode
4. Add description for the commit

5. Test before commiting

⇨ git add index. html

⇨ git commit – m “Add footer feature” -m “closes #1”

HEAD, COMMIT:

1. **git log** = commit er history dekhar jonno
2. **git log --online** = history er moddhe shudu important masses gulu dekhar jonno
3. **git show** = commit er ki ki kaj kora holo ta dekhar jonno.
4. **Git show commit_id** othoba **git show HEAD-Number** = nidisto commit a ki ki kaj hoyche ta dekhar jonno.

Git checkout:

1. **git checkout** = “git reset” use korata nirapod noy, er poriborte amra git checkout use korte pari.
2. **git checkout commitID/HEAD-Number**

Jei commit e amra fire jete chacchi.

3. **git checkout master** = aita diye amra sorbosesh commit er chole jabo, recent commit a chole asbe.

Cmd:

⇨ mkdir notes-2

⇨ cd notes-2

⇨ git init

⇨ ls -a

⇨ echo > test1.txt

⇨ start test1.txt

(write some text on test1.txt)

⇨ git status

⇨ git add .

⇨ git status

⇨ git commit -m “first commit”

- ⇨ git status
- ⇨ git log
- ⇨ git log --online
- ⇨ git show commit_id
- ⇨ echo > test2.txt
- ⇨ start test2.txt
  (Write some random text on test2.txt)
- ⇨ git status
- ⇨ git add test2.txt
- ⇨ start test2.txt
  (add some extra text)
- ⇨ git checkout test2.txt
- ⇨ git status
- ⇨ git commit -m “Second massage”
- ⇨ git log

akta commit theke kivabe arekta commit a move korte hoy:

- ⇨ echo > test3.txt
- ⇨ start test3.txt
  (write some text on test.txt)
- ⇨ git add . && git commit – m “Third commit”
- ⇨ git log --online
  ekhane theke ber houar jono “:+Q+enter”
- ⇨ git show commitIdD
- ⇨ git checkout commitID
- ⇨ git log --online
- ⇨ git checkout master

Jei commit e amra jete chacchi tar id

Amra ager commit a fire jacchi

.git ignore gile:

➔ekhane amra jesob jinis ignore korte chai sei sob jinis rakhbo.

➔ekhane amra secret file gulu rakhte pari- orthat je gulu amra git a rakhte chachhi na.

Cmd:

- ⇨ git init
- ⇨ git status

- ⇨ echo > .gitignore
- ⇨ ls -a
- ⇨ start .gitignore

  (write the name of files with extension which want to ignore

  Eg.

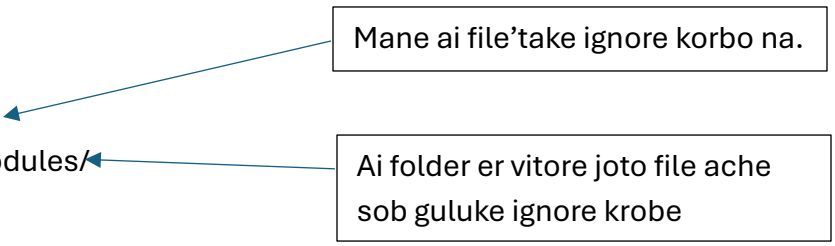
  .env

  *.text

  !story.txt

  node_modules/

  temp/

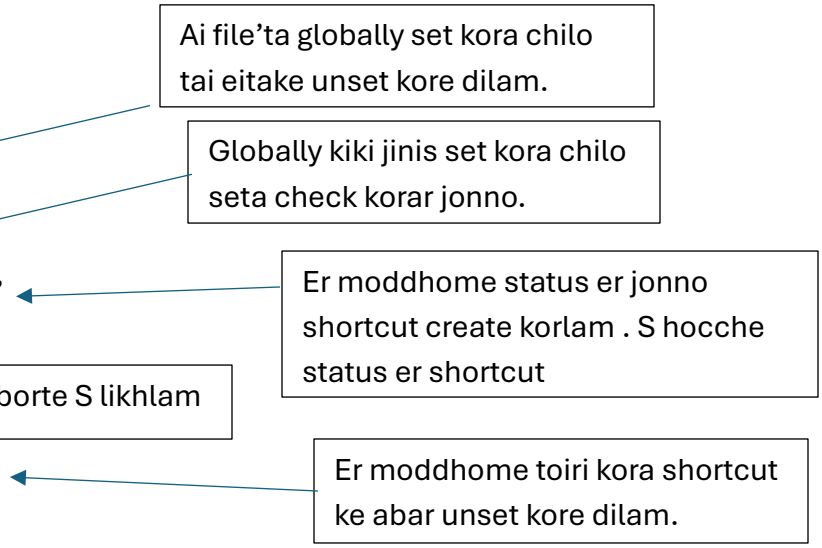


Git alias:

➔**git alias** = er moddhome commands'ke shortcut kore use kroa jay

In vs code terminal:

- ⇨ git config --list
- ⇨ git config --global --unset alias.st
- ⇨ git config --global --list
- ⇨ git config --global alias.S "status"
- ⇨ git s
- ⇨ git config --global --unset alias.S



github repository and commit:

Summary:

➔git initialize

➔adding some file

➔git add

➔git commit – commit history

➔.gitignore

[note: repository name e kono space thakbe na]

MarkDown:

1. Mark down ki?
   == amra jei plane text gulu sei gulu formate kore web a prodhorshoner er amr markdown syntax use kore thaki. Markdown hocche akta light weight language. (.md) hocche markdown er extention.
   README.md – ai file kichu summary dewa ache.
2. **Mark down Comment :** amra jevabe himl er comment kori seivabei akhone comment kora hoy.
3. **New line create:** text er pore duiti space dilei new line create hoy.
4. Create Horizontal Line: (---) ei rokom tinti hyphen dilei Horizantal Line create hoye jay.
5. Create Headding: Heading er jonno '#' use korte hoy.
   Eg:
   #SAJID ← H1 heading hisebe kaj kore
   ##SAJID ← H2 heading hisebe kaj kore
   ###SaJID ← H3 heading hisebe kaj kore

6. Create Paragraph: Paragraph create korar jonno html er p tag use korte hoy.
7. Creat Italic text: italic text er jonno , amr jei text'ke italic korte chacchi oitar shuru abong sheshe underscore use korbo.
   Eg: -this is an italic text-

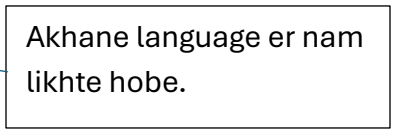
8. Create Bold text: Strong ba bolt text korar, jonno amr jei text'ke bolt korte chacchi oitar shuru abong sheshe duiti kore underscore use korbo.
   Eg. --this is a bolt text--
9. Strikethrough text: amr jei text'ke strikethrough korte chacchi oitar shuru abong sheshe duit kore "~" ai symbol use korbo.
   Eg. ~~ this is a strikethrough text ~~

10. Inline code block and multiple code block

    `this is inline`

    ```
    this is multiple code block
    ```

Jodi multiple line er moddhome je kono language er code likha gulu dekhte chaile:




```html
<html>
<head></head>
<body></body>
</html>
```

```javascript
Console.log("hello")
```

```CSS
head {
background-color: red;
}
```

11. Create list:

    oder list er jonno:

    1.Item1
    2.Item2
        1.Item1.1
        2. Item1.2
    3.item3

Unorder list er jonno:

- item1
  - item1.1
  - item.1.2
- item2
- item3

12. Task list
    - [x] task1
    - [x] task2
    - [x] task3
    - [ ] task4
13. Create Link

Automatic link:

https: //www.youtube.com

disable link

`https:// www.youtube.cm

Valid syntax of link (markdown):

[title](link)

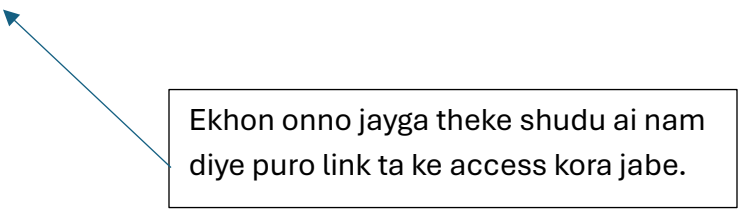
Eg:

[youtube](https;//www.youtube.com)

Link er poriborte kono nirdisto nam set korte chiale:

[linkname]: (website)

[youtube]: https://www.youtube.com



14. Image add korar jonno:

Syntax:

```
![alt text] (image source)
```

Eg:

```
![Profile](./image/me.JPG)
```

15. Table:

    Syntax:

```
| NAME | EMAIL |
|-----|----- |
|Anisul Islam | texts here |
|anisul islam  | anisul |
|Anisull Islam | text is here |
```

Vs code terminal:

- ⇨ git init
- ⇨ git add .
- ⇨ git commit -m “reame templete”

now, singin github ➔ then goto repository ➔ new ➔ Repository name: readme-template ➔Descripton:” a basic template for readme file” ➔ create repository.

Vs code terminal:

- ⇨ copy and paste (gitremote add orgine https: //)
- ⇨ copy and paste (git push -u orgin master)

Connecting local and remote repository:

Push: amra ja ja changes korichi setar jonno push korbo.

Pull : onnora ki ki update korlo, seta pawar jonno Pull korbo.

1. **git remote -v** =Local repository er sath remote repository er connection ache kina check korar jonno.
2. **git remote add name <REMOTE_URL>** = Remote url ba repository add korte chaile syntax.
   eg:

⇨ git remote add orgin https://github.com/aisajid/life-story

Jekono akta name dilei hobe, but sadharonoto orgin use kora hoy

in cmd:

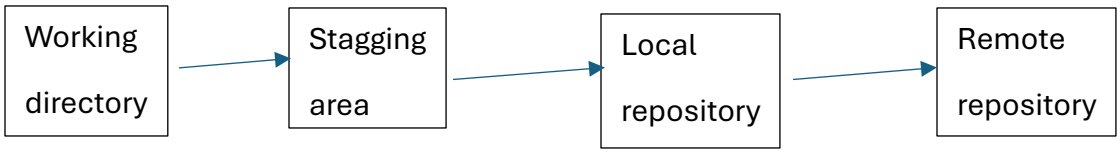
⇨ cd desktop
⇨ mkdir life-story
⇨ cd life-story
⇨ git status
⇨ git remote
⇨ git remote add orgin https://github.com/aisajid/life-story
⇨ git remote
⇨ git remote -v

remote repository copy kore ba clone kore kivabe amder pc'te ana jay:

⇨ cd destop
⇨ git clone (HTTP url paste here)
⇨ rm -rf life-story
⇨ git clon (paste here HTTP url of remote repository)

Git push & pull:

Push: jokhoni amra files guluke local repository'te niye jete parbo tokhoni amra remote repository'te push korte parbo.



➔git pull = er maddhome pull korte pari.

In cmd:

- ⇨ git status
- ⇨ start test1.txt
  (adding some text on test1.txt)
- ⇨ git status
- ⇨ git add test1.txt
- ⇨ git status
- ⇨ git commit -m “added info about my HSC”
- ⇨ git startus
- ⇨ git log
- ⇨ git log --online
- ⇨ git push -u origin main

For Password generation: singin github ➔setteing ➔ developer setting ➔ Personal access ➔ tokers➔

Cmd:

- ⇨ start test.txt
  (again, add some text on test1.txt)
- ⇨ git status
- ⇨ git add test1.txt
- ⇨ git commit -m “added BSC info”
- ⇨ git push

aibar remote repository’te change kore se gulu local e niye jabo, I mean notun kichu text add korlam and notun kichu commit korlam.

- ⇨ git log --online
- ⇨ git pull

Branch and merging:

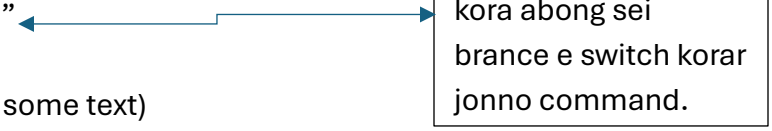
Branch: main / master je repository’ta ache tar akta copy toiri kore onno akta nam diye Bracnchong korte pari.

1. git branch branch_name = er maddhore arma branch create korte pari.

2. git checkout branch_name: amra jodi master / main branch theke sei branch e move krote chai.
3. git checkout master: master e move korar jonno.
4. git merge brach_name: duita branch'k marge kora jonno.

Cmd:

- ⇨ git branch
- ⇨ git branch feature1
- ⇨ git checkout feature1
- ⇨ git branch
- ⇨ git branch -d feature1
- ⇨ git checkout main
- ⇨ git branch -d feature1
- ⇨ git branch
- ⇨ git cheackout -b "feature-1"
- ⇨ git branch
  (open test1.txt file and add some text)
- ⇨ git status
- ⇨ git add tex1.txt
- ⇨ git commit -m "add employment history"
- ⇨ git log --online
- ⇨ git push -u orgin feature1



Locally Branch:

Locally Branch er sathe kivabe local branch er merge krte hoy:

Cmd:

- ⇨ git branch
- ⇨ git checkout -b "feature2"
- ⇨ git branch
- ⇨ ls
- ⇨ start test1.txt
  (add some new txt)
- ⇨ git status
- ⇨ git add .

- ⇨ git commit -m “Job starting date is added”
- ⇨ git checkout main
- ⇨ ls
- ⇨ start test1.txt
  (no need to add text)
- ⇨ git branch
- ⇨ git merge feature2
- ⇨ ls
- ⇨ git branch
- ⇨ start test1.txt
- ⇨ git putt
  press: w+g+enter
- ⇨ git push -u orgin main

Practice:

Step-1: Create a git repository

Cmd:

- ⇨ mkdir test.txt
- ⇨ cd test.txt
- ⇨ git init
- ⇨ ls -a

Step-2: Add files(Such as .gitignore and README.md)

Cmd:

- ⇨ echo > README.md
- ⇨ start README.md

now, README.md file er vitore:

#Practiceing git

Cmd:

- ⇨ git status
- ⇨ echo > .gitignore
- ⇨ ls -a
- ⇨ stat .gitignore

now, in .gitingnore file:

node_modules /

step-3,4: Stage and commit change

cmd:

- ⇨ git status
- ⇨ git add .
- ⇨ git commit -m “initial commit”
- ⇨ git status
- ⇨ git log --online

Step-5: Create branch and switch to branch

Cmd:

- ⇨ git branch
- ⇨ git checkout -b “feature”
- ⇨ git branch
- ⇨ echo > test.txt
- ⇨ start test1.txt
  (write:”This is test 1”)
- ⇨ git add .
- ⇨ git commit -m “added text1 file”
- ⇨ git status
- ⇨ git log
- ⇨ git checkout master
- ⇨ git branch
- ⇨ git merge feature

Step-6: Connect local repo and remote repo

Now, sing in github➔Create a new repository: “test” ➔ disciption: “This is test repository for Practicing git and github knowledge”

Cmd:

- ⇨ gfit remote
- ⇨ paster the whole code (repository url)
- ⇨ git remote
- ⇨ git push -u origin master

➔now add some text in text.1txt from github and commit massag: “added secondline”.

Cmd:

- ⇨ git pull
- ⇨ start text1. Txt

resolve merge conflict:

cmd:

- ⇨ cd desktop
- ⇨ mkdir test-git
- ⇨ cd test-git
- ⇨ git init
- ⇨ ls -a
- ⇨ echo > story.txt
- ⇨ ls
- ⇨ start story.txt
  (write some text on story.txt)
- ⇨ git status
- ⇨ git add story.txt
- ⇨ git status
- ⇨ git commit -m “created story.txt”
- ⇨ git log --online
- ⇨ git branch feature
- ⇨ git checkout feature
- ⇨ start story.txt
  (add some text on story)
- ⇨ git commit -am “modify by text from feature branch in story.txt”
- ⇨ git checkout mater
- ⇨ start story.txt

(add some text on story.txt)

- ⇨ git commit -am “added by text from master branch in story.text”
- ⇨ git log --online
- ⇨ git merge feature
- ⇨ git status
- ⇨ start story.txt
  (amra jei branch er text rakhche chacchi, oita rekhe baki text gulu delete kore dibo.)
- ⇨ git commit -m “resolve conflict from master branch”
- ⇨ git status
- ⇨ git log --onine

fork & clone:

jodi apni onno karo project a contribute chan tahole tar github repository’ke fork korte hobe. – tahole tar repostiroy’ta apnar github e repository hosebe chole asbe.tarpor sei repostory’ta clone kore amra locally niye asi.

➔prothome fork

➔tarpor clone

➔tarpor locally kaj korbo, change korbo.

➔tarpor seta push korbo,,, tarpor seta amder repository’te chole asbe.

➔tarpor pull request pathabo take.