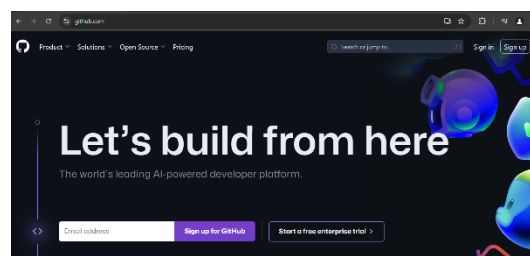


We can show practically create **github Account :**

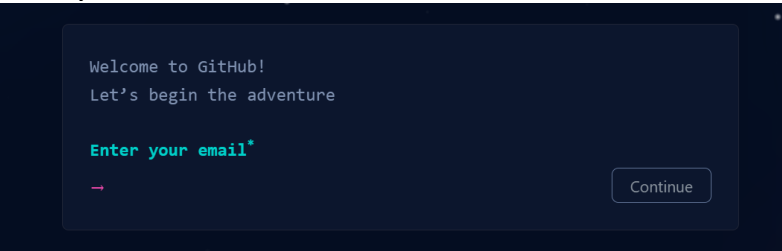


<https://github.com>

Goto google—github—click on that page— click on sign up option

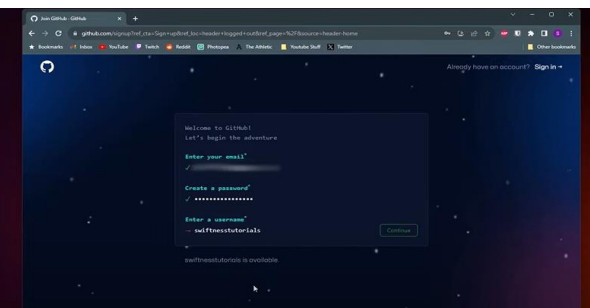
U have already github account u can sign in

Give your email id .....

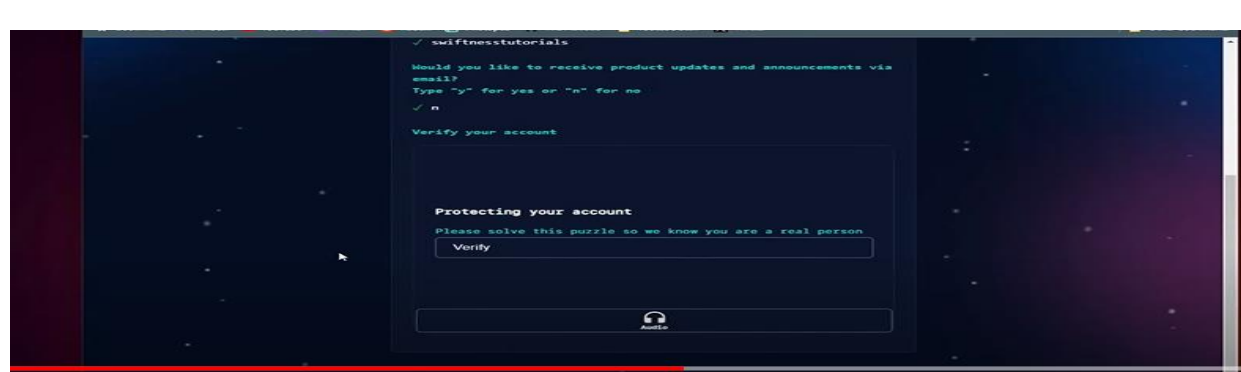


After u can set the password :

Set the user name



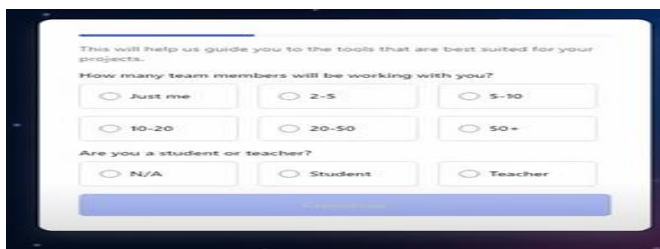
After set the username u can solve the one puzzle ..is the security purpose



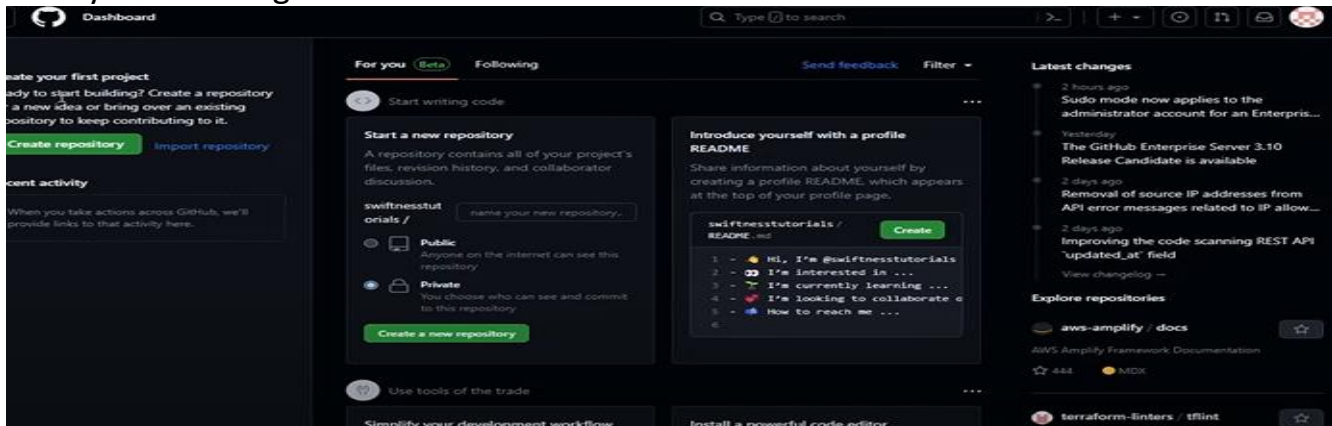
Then u got it one otp in your mail enter that otp here



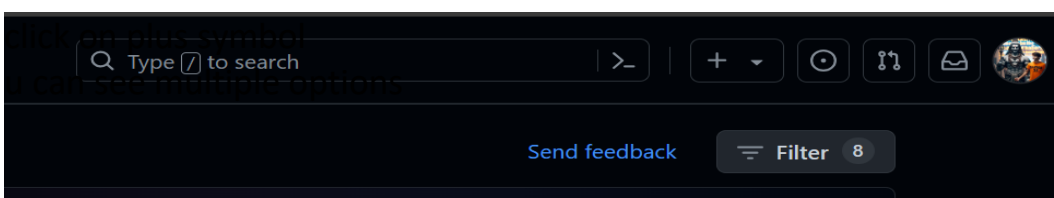
After enter otp set the date of birth , and also enter are u teacher or student u can enter

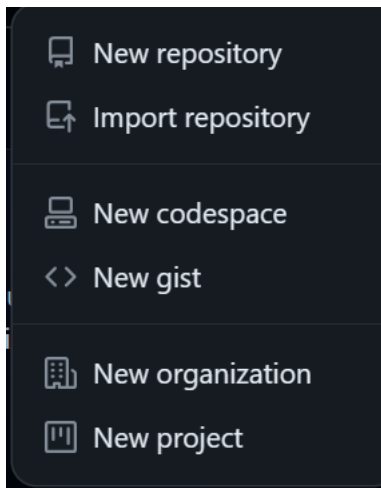


Finally u created github account



Create a repository :

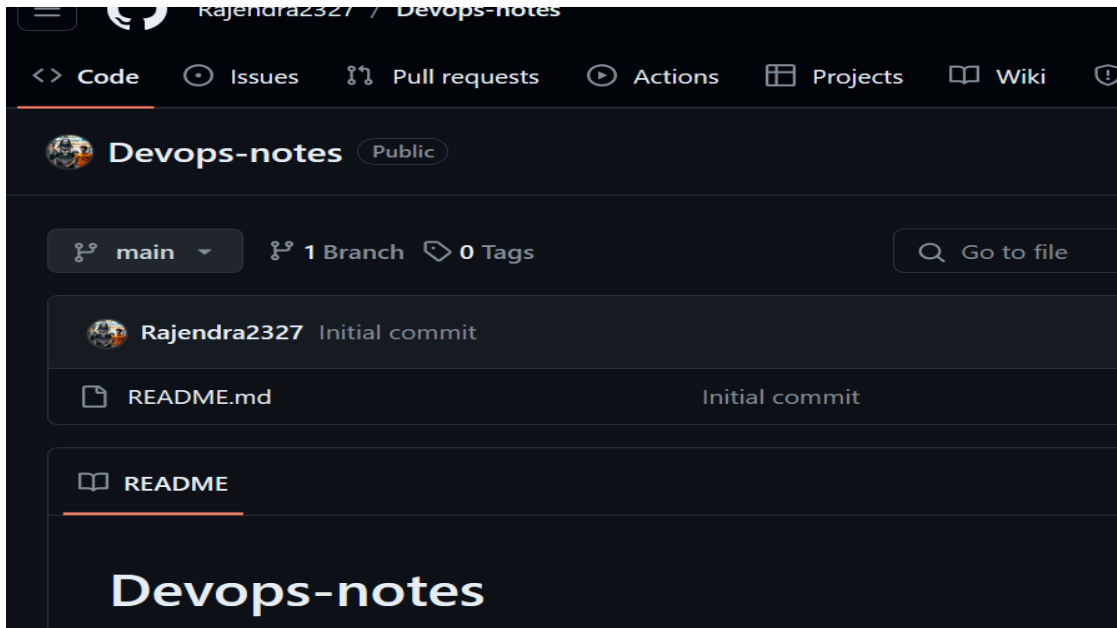




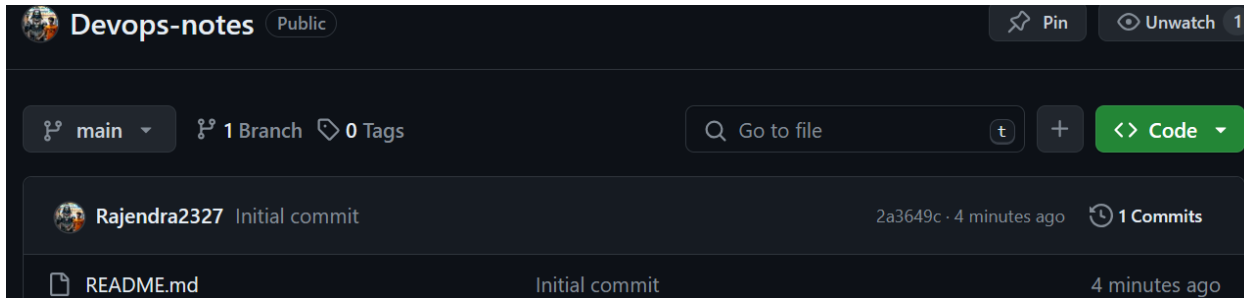
u can click on new repository

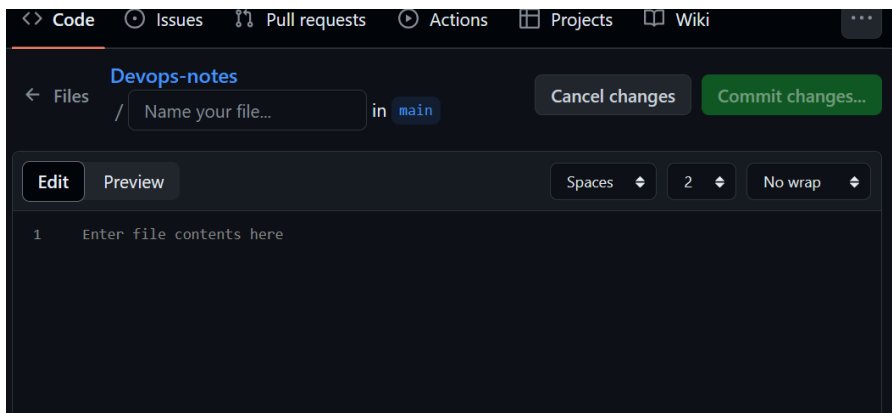
A dark-themed screenshot of the 'Create a new repository' form on GitHub. The form has a title 'Create a new repository' and a subtitle 'A repository contains all project files, including the revision history. Already have a project repository? [Import a repository.](#)'. Below this is a note: 'Required fields are marked with an asterisk (\*)'. The form contains two main sections: 'Owner \*' with a dropdown menu showing 'Rajendra2327' and 'Repository name \*' with an empty text input field. Below these is a hint: 'Great repository names are short and memorable. Need inspiration? How about [solid-octo-meme](#) ?'. There is a 'Description (optional)' text area. At the bottom, there are two radio buttons: 'Public' (selected) with the description 'Anyone on the internet can see this repository. You choose who can commit.' and 'Private' with the description 'You choose who can see and commit to this repository.'. At the very bottom, it says 'Initialize this repository with:'.

➡ U can give repo name and description also then click on repo is public or private anything ur wish . click on readme option -- scroll down click on create repository



Create file : click on plus symbol ➡ click on new file



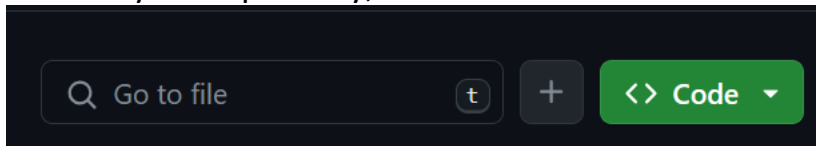


Give your repo name ex: Raju java1 file

Here u can write code or anything after writing click commit it can save .

## How to clone the central repo to local:

Click on your repository, click on code



Here u can see multiple clone paths ....now u can copy Http path.

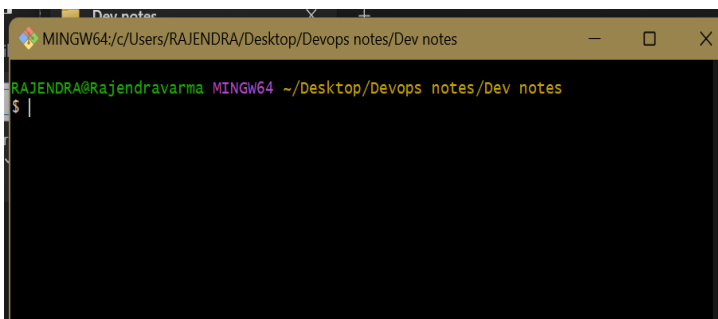


Now create one folder in desktop or any (drive c or D) desktop is easy to access .

First u install git or git bash :

<https://github.com/git-for-windows/git/releases/download/v2.30.1.windows.1/Git-2.30.1-64-bit.exe>

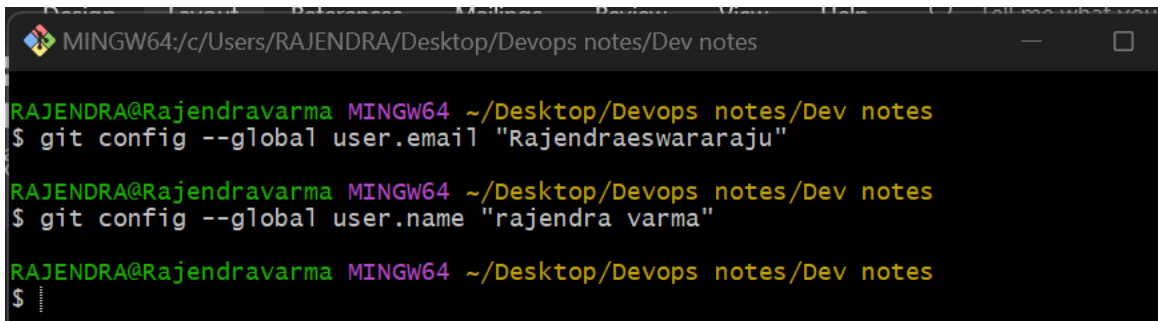
1. install the git bash in your computer.
2. Go to (GITHUB) online repo copy the **CLONE** URL link.
3. go to desktop create folder.
4. go to folder give right click and open git-bash.



```
RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes
$ |
```

# Configure Username & email

5. git config --global user.email "emilid"
6. git config --global user.name "name"



```
RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes
$ git config --global user.email "Rajendraeswararaju"

RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes
$ git config --global user.name "rajendra varma"

RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes
$ |
```

U can see user details



```
RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes
$ git config user.name
rajendra varma

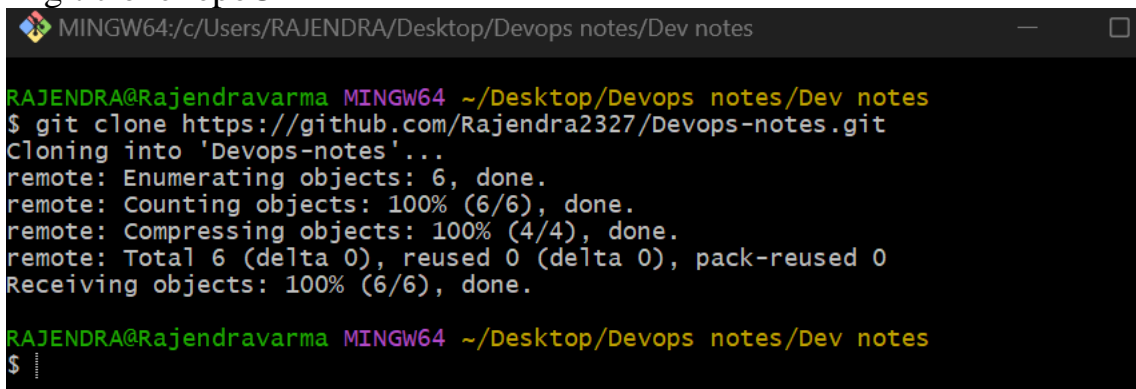
RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes
$ |
```

Use commands ....i can give full commands notes above file .

Now clone the central repo to local :

#Clone the code from github

git clone repoURL-link



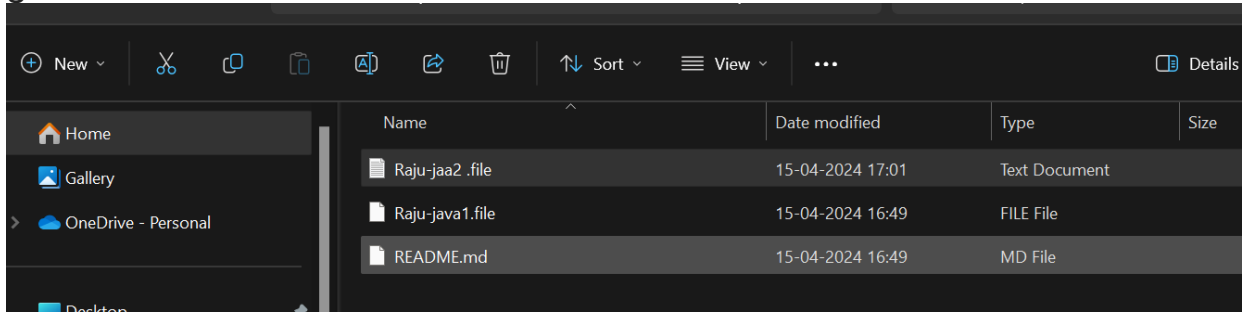
```
RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes
$ git clone https://github.com/Rajendra2327/Devops-notes.git
Cloning into 'Devops-notes'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.

RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes
$ |
```

Successfully clone that repo to local . any code file any other data have central repo now it come your local folder .u read it and edit , modify that files not effect on central repo data...

***How to push the file local to central :***

You can create files and modify the existing data in that folder after save files open gitbash .



Here I am create one new file and also modify one file .

```
RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes
$ ls
Devops-notes/  'New Section 1.one'  'Open Notebook.onetoc2'

RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes
$ cd devops-notes

RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes/devops-notes (main)
$ ls
README.md  'Raju-jaa2 .file.txt'  Raju-java1.file

RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes/devops-notes (main)
$ git add .

RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes/devops-notes (main)
$ git status
on branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Raju-jaa2 .file.txt
        modified:   Raju-java1.file

RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes/devops-notes (main)
$ |
```

You can see how many file in my folder use ls command .

Enter **Git add** . files go to staging area -git status command use see staging area files in **green colour** ,file not add it ill show **red colour**.

After git add . u can commit these files use command **git commit -m** “ modify file”  
Now give **Git push** command .

```
RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes/devops-note
in)
$ git commit -m "modify files"
[main b4e68d1] modify files
2 files changed, 4 insertions(+)
create mode 100644 Raju-jaa2 .file.txt
```

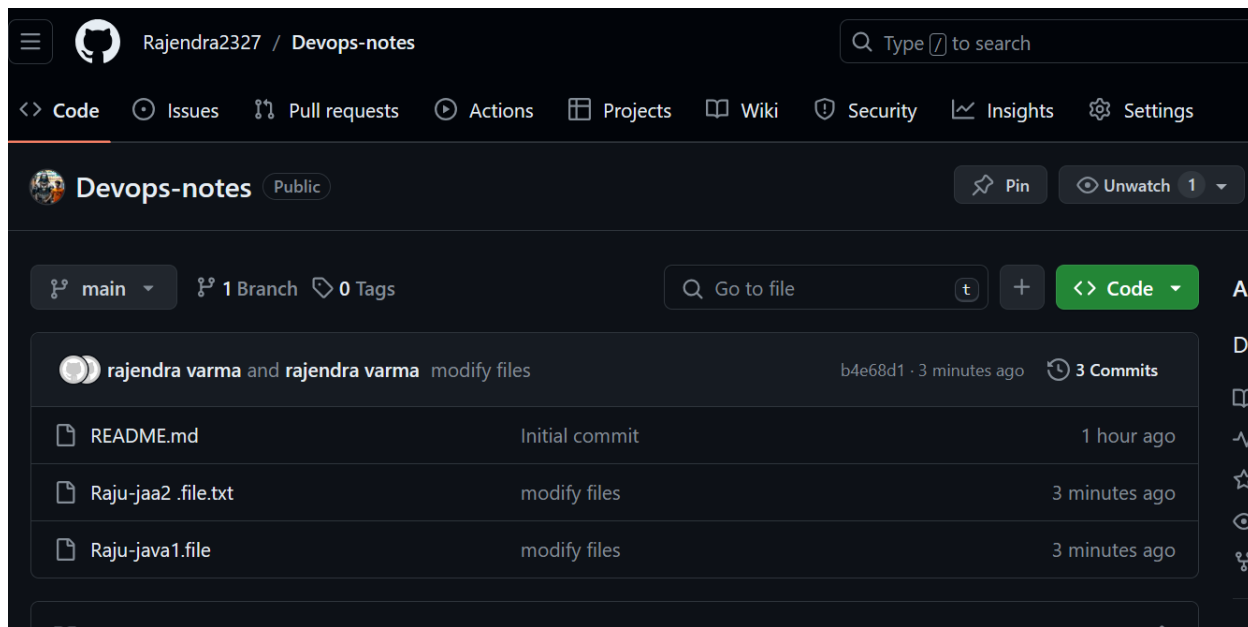
You first time to push the code u can enter your github details user name and password once u give details next not ask it ill direct push the file github .

```

RAJENDRA@Rajendravarma MINGW64 ~/Desktop/Devops notes/Dev notes/devops-notes (main)
$ git push
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 8 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 395 bytes | 395.00 KiB/s, done.
Total 4 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Rajendra2327/Devops-notes.git
c710a9b..b4e68d1 main -> main

```

Now u see after push in central repo u have add 1 file and 1 modify file .  
Whenever u push the files refresh the browser page .



Suppose another person push some file in github repo....but that files not come to your local that time u use `Git pull` or `git – rebase` commands it ill come to your local .

Ex commands : above file give total command notes u can learn it.

- 1.git add .
2. git commit -m
- 3.git push
4. git pull
- 5.git status
- 6.git diff
- 7.git log
- 8 git merge
- 9.git touch
- 10 .cat
- 11.mkdir