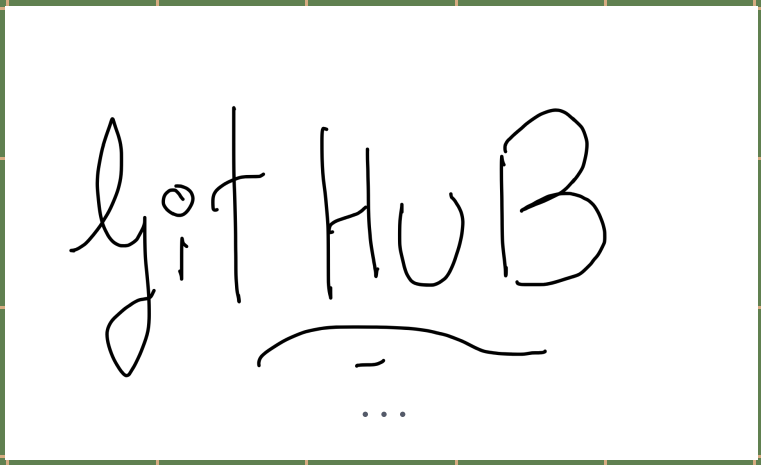

git HuB

github

what is GIT? → version control
- → Time Machine
- → Check points (commit)
- Synchronize Branches.

→ Git Config

git config --global user.name "Your Name"

git config --global user.email "your Email"

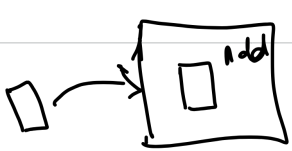
→ Open git bash

→ Prepare a folder to push directories & open the folder in vs code.
In terminal, command

→ # git init // (intialize git towards files)

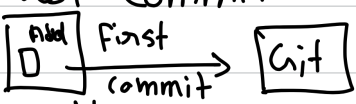
→ Staging files.

git add .



→ Commit files with a message

git commit -m "first commit"



→ To see last log of commit -

git log

⇒ Git Environments

git status // (To see at what position the files are)

→ To Restore files (last commit)

git restore --staged README.md

(To discard changes in working directory)

→ # git restore . // (it will take back the file)

→ Why Ignore files?

- Sensitive info
- Personal notes
- System files.

→ create a file name .gitignore in that

paste this →

```
.DS_Store
.vscode/
authentication.js
node_modules
notes/
**/*-todo.md
```

& add & commit it.

So whenever you will make file it will hided, when you check git status.

Global Ignore File

git config --global core.excludesfile [file]

// will automatically exclude file if added.

To clear the cache:

git rm -r --cached

→ To delete a file in git

git rm index.html

// automatically deleting from git.

→ To restore the file

git restore -S index.html // (or .)

git restore .

⇒ To rename a file in git.

git mv index.html home.html

(saves up staging a file)

→ Difference between the files

git diff // (if you modify anything it will show you what you have done)

---

gitlog.

git log --oneline (shows logs in one line)

Branches

git branch // (display branches)

New Branches

git switch -c fix-classes
// (& then you can modify changes, resting previous work in the orginal state)

git switch main
// (just to change branches)

Merging Branches

git merge fix-classes
→ (Additional Branch)

// (fix-classes added to main branch)

→ Deleting Branches

git branch -d fix-classes (To delete a branch)
git branch --delete } (Alternative commands)
git branch -D

→ Merge Conflicts

1) First install Live Server in VS Code.
2) Open live server through Ctrl + shift + p
3) Start live server.

→ Create another brach, → fix it → comit it & another merge it again.

git Stash

grey

Methods → Modify to Blue → # git stash (return to grey)

Blue Background (stored the condition)

git stash list

(list the stashes)

Blue

> can also → Modify

git restore . to Red background

(dark grey) Red

(dark grey)

git stash

git stash ← # git stash pop

git stash list

(Getting Red Colour: Red, Blue)

(Dark grey)

> git restore . → git stash pop (Blue) → Blue

git commit -m "Blue - added" ← git add. ←

Now you can add it if you want

A: you can also use # git stash apply without removing the item from the stash. (we used "pop")

git clean

git clean -n → files (if you added new folder or files)

|| (display files)

git clean -dn → files
display directories (folders)

git clean -df → directories & files

git clean -dn → only files

⇒ More on git stash (Again)

1) we modify code (Blue) → # git stash (return to orginal state) (grey)
→ For ex:

2) #git stash list (List of stacks of modifications)

stores in index form (0, 1, 2 ---)

3) #git stash apply 0 → return the modified code.

4) # git restore . (To go back to orginal state)

5) Let take one more modification (Red)

6) # git stash (Red → list & orginal code)
(so blue → index-1 & red → index-0)

7) # git stash list

8) # git stash pop (will get last item on list & delete it)

9) # git stash list

10) # git restore .

11) # git stash pop.

12) & if you want add it the
git add .
git commit -m " "

To Rename master to main :
git branch -m master main

GitHub

- Cloud Repository
- Collaborative Development
- Project Management

→ Open your github profile.

1) give repository name - "github project" & you can also add description (optional) if you want

2) redirect to setup page.

→ Pushing :

1) Adding Remotes → Rename master to main:

```
$ git branch
-m master main
```

git remote add origin https://.....

2) # git push --all (All the branches & main)

(optional for 1 file) 3) # git push -u origin main (if you want to push a particular file)

4) For Authentication use token from github # git remote set-url origin https://(token)@github.com/ayush----.

or you can clone the repository & paste files in that fary.

→ Pull Request (in github)

For ex: open → index.html & make a change like title "Hello, welcome"

↓

Preview the change & commit to main or can create a new branch.

Opens pull request. ← Can Ask for Comments ← if it's ok. ← Merge the pull request.

(can also hold labels) & can

close the issue by connecting it.

→ Organize Projects

go to manage access → invite a friend

→ My friend can review & make issues & comments about the project.

↓

Can answer by comment & can add milestone.

→ You can Create a Project:

→ go to project & create one with predefined template.

& can add the comment in to-dos, progress & etc.

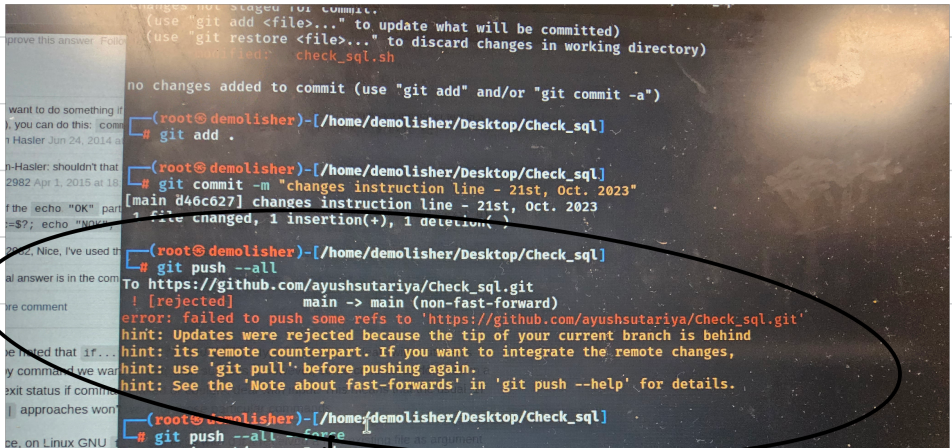
→ Syncing github.

git clone https://....
(To copy on local device)

(in visual code) git fetch
(To get all new data but will not modify it)

git pull (integrate the remote data with the local files)

→ we can create a releases to store different version.



if error comes like this.

```
# git rebase main
# gish push origin main
```

(your main branch name)

Sometimes merge would clusters the history, so to make it reverse & to make it do again we use rebase.