

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 // (initialize git towards files)

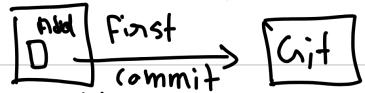
→ Staging files.

```
# 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 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  
in that

paste this →

- DS\_Store
- VSCode / authentication .js
- node\_modules
- notes /
- \*\* / - tools .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.excludes  
file [file]
```

It 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 original  
state)

# git switch main  
|| (just to change branches)

---

## Merging Branches

# git merge fix-classes

→ (Additional branch)

1) (fix-classes added to main branch)

## → Deleting Branches

branch

# git ^ -d fix-classes (To delete a branch)

# git branch --delete } (Alternative  
# git branch -D } Commands

## → Merge Conflicts

install

1) First ^ live Server in VS Code .

2) Open live server through

ctrl + shift + p

3) Start live server.

→ Create another branch , → fix it → commit it  
& another merge it again ←

# git Stash

grey

Methods →

Modify to → # git stash  
Blue background (stored the condition)

# git stash list  
(list the stashes)  
Blue

> can also → Modify  
git restore . to Red background  
(dark grey)

git  
stash list

# git stash

Red  
(dark grey)

pop  
(Getting Red)  
Colours  


# git stash

(Dark grey)

git restore .

→ git stash pop

Blue

(Blue)

git commit -m  
"Blue-added"

A You can also use # git stash apply  
without removing the item from the stash.

Now you can add it if you want

We used "pop"

## git Clean

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

(I) (display files)

# git clean -d n → display directories (folders)

# git clean -d f → directories & files

+ git clean -d n → only files

⇒ More on git stash (Again)

i) we modify code → # git stash

(Blue)

(return to original state)  
→ For ex:

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

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

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

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

5) Let take one more modification (Red)

6) # git stash (Red → list & original 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

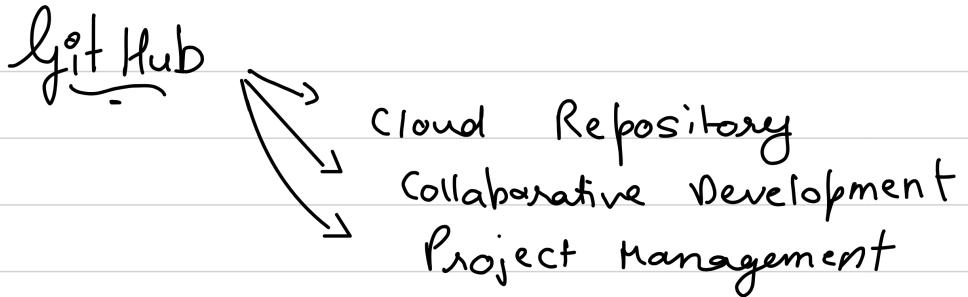
# git add .

# git commit -m ""

---

To Rename Master to main :

# git branch -m master main



- 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  
# git remote add origin https://.....
  - 2) # git push --all  
(All the branches & main)
  - (optional 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/username`  
---.

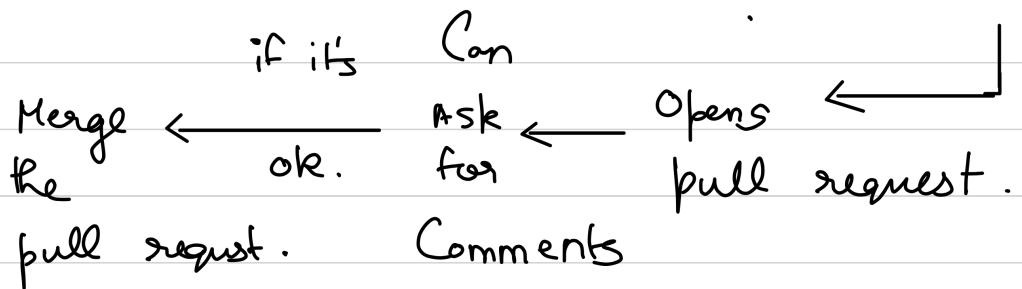
or you can clone the repository &  
paste files in that way.

## → Pull Request (in github)

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



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



(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.

---