

Git & Github

✓ Git = version control system = is a tool that helps to track changes in a code.

✓ Github = website that allow developers to store & manage their code using Git.

✓ any edit inside README.md file enhances via HTML code.

✓ Commands:-

(in gitbook)

✓ `git --version` (to get version of your git)

✓ `clear` (clear screen)

also in
vs code terminal ✓ `pwd` (working directory)

at Start only → Configuring git :-

`git config --global user.name "My Name"`

`git config --global user.email " - email "`

`git config --list` (to check setup in git)

→ (left bottom)

✓ In vs code terminal after making a file :-

`git --version` (can check here also)

✓ Clone & status

cloning a
repo on our local
machine

→ displays the state of code

↓
`git status`

(to check status)

cd = change
directory

↓
copy code (green box → HTTP) of your repo

↓
inside vs code file terminal

`git clone` paste the link

↓
our repo will get clone in vs code

↓
To use this folder → `cd demo-git`

Date : 11
Page : 1

error:-

- git fetch origin
- git rebase origin/main



3. Add something on Readme → `git add.` → `git commit -m "add new"`

↓ `git push origin main`

4. (Better approach) add something/edit → `git commit -am "added dot"`

↓ `git push`

For this you have to do

first → `git push -u origin main` (likh do ek baar bas)

Workflow :-

Github

↓
Code changes

↓
Commit

Local Git

code changes

↓
"add" (stage a change)

↓
"commit" (commit changes)

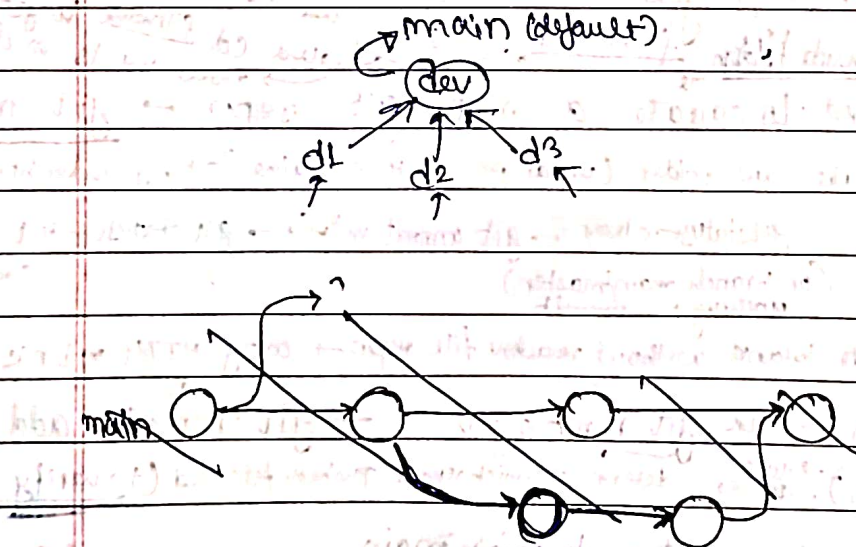
↓
"push"

further we'll study pull so it

will continue...

Git branches

It is a pointer to commit that allows for isolated work on features or bug fixes, enabling parallel development & collaboration without affecting main code. (with multiple users)



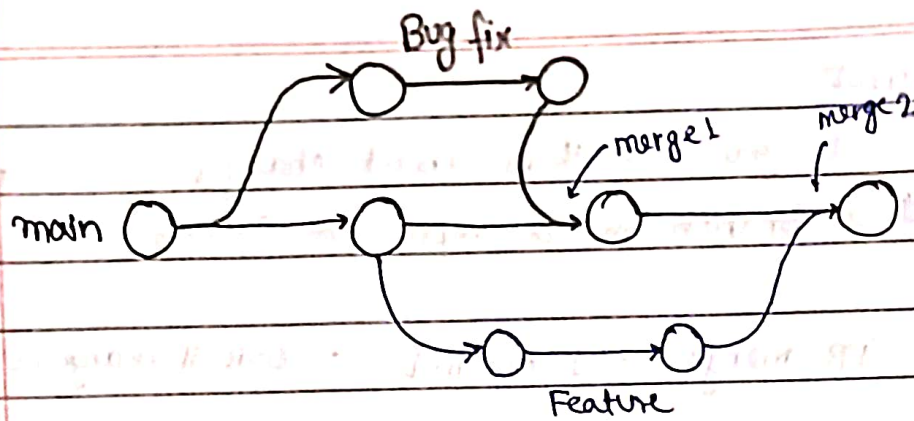


fig: How branch works

• Branch commands:-

- `ux → git branch` → to check branch

`git branch -M main` → to rename branch

`git checkout branchname` → to go from one branch to other
 Since at start we have only 1 branch i.e. main it kaha navigate karne jaye

`git checkout -b new branchname` → to create new branch

`git checkout main` → to switch back to main branch

`git checkout branchname` → to go back to other branch name

`git branch -d branchname` → to delete branch

Note:- You cannot delete a branch you are actually in.

- ∴ Branch new banao usme changes karo → push kardo

using → `git push --set-upstream origin branchname`

nhi yaad ho rha ho sirf `git push` likho wo suggest karega
 + this is coming becoz -u nhi hua tha pehle

- Merging code you can write main also here if in diff. branch (just compare)

`git diff branchname` → to compare commits, branches, files, etc.

`git merge branchname` → to merge 2 branches

OR → Create a PR → then merge pull request → confirm merge

preferred
 Pull request
 (directly on GitHub)

push: local \rightarrow github
pull: github \rightarrow local

• Pull request

\rightarrow it lets you tell others about changes you've pushed to a branch in a repo on Github.

Note:- PR merge \rightarrow 1 commit \therefore code ki change ho rha

\rightarrow After merging you can safely delete your other branch
 \therefore (merge ho gya toh uska koi kaam nhi hai)

\rightarrow when changes occurred on Github we have to bring back on local vs code we use:-

type git checkout main \rightarrow you will see file added via other branch is cut.

git pull origin main

used to fetch and download content from a remote repo (Github) + immediately update the local repo (vs code) to match the content.

• Merge Conflicts

\rightarrow rises when eic file mein changes hua in a certain branch + other branch ki bhi wohi same changes hua to.

\therefore it is an event that take place when Git is unable to automatically resolve differences in code b/w two commits.

\therefore after dono branch ki change to ~~push~~ commit

karne ke baad when we merged via \rightarrow git merge branch

(stop diff)
accept current changes / accept incoming change / accept both ^{compare} \rightarrow in vs code we have \rightarrow conflict arised
means take changes of main | \downarrow our change of | \downarrow direct editons
branch only remove change | other branch except | (without vs code editor)
of other branch | main | manually erase extra \therefore accepted \rightarrow git add
parts both \downarrow commit
 \downarrow git push origin main

• Forking

→ search any repo on Github e.g. Node → click on fork → uncheck only main code → create your own repo (copied) → now you can change it by yourself → earlier not possible → we get edit option now → after changing you can click on pull req. Now it's their duty that they will merge it or not.

← click on create pull request ← new pull req. branch (auto) → can change khud se

Not

1. A fork is a new repository that shares code & visibility settings with original "upstream" repository.

→ Fork is a rough copy.

Check code