

~~THE GIT~~

GIT

Git is a distributed version control system.

Ques What is a version control system?

A. version control system is a system that maintains different versions of your project when we work in a team or as an individual.

As the project progresses, new features get added to it. So, a version control system maintains all the different versions of the project for us. and we can roll back to any version we

want without causing any trouble to us for maintaining different versions by giving names.

Distributed version control system means every collector has a local repository of the project in his/her local machine.

Unlike in ~~Central~~ Central where Team members should have an internet connection to every time update their work to the main central repository.

Ques What is a repository?

A repository is an area that keeps all your project files, images, etc. In terms of Github: different versions of projects corresponds to commit.

GIT Repository Structure

It consists of four parts

i) Working Directory

This is your local directory where you make the project and make changes to it.

(ii) Staging Area (or Index)

This is an area where you first need to put your project before committing. This is used for code review by other team members.

(iii) Local Repository

This is your local repository where you commit changes to the project before pushing them to the central repository on Github. This is what is provided by the distributed version control system. This corresponds to the .git folder in our directory.

(iv) Central Repository

This is the main project on the central server, a copy of which is with every team member as a local repository.

#

GITHUB

It is code hosting platform for version control & collaboration.
~~Github basically is a~~
It helps users share their git repository online, with other users, or access it remotely.

Users share their repository online for various reasons including but not limited to project deployment, project sharing, open source contribution, helping out the community and many such.

#

GIT Commands

i)

`git --version``git config --global user.name "your name here"``git config --global user.email "your email here"`

Git user configuration.

ii)

`git init`

Initialize your directory to work with git and makes a local repository.
· git folder is made.

A git branch → gives all the info/list of all the branches in git repo

Date: / /

Page No.

(iii) git status

The git command is used to show the status of our git repository. This command displays the state of the local repository and the staging area.

Ques What is Branch?

Branching means diverging from our mainline and continue to work separately without messing with the mainline.

(iv) git branch branchname

It creates a new branch with the name mentioned.

e.g. git branch mybranch

(V) git checkout -b branchname

It creates and switch to the new branch created.

e.g. git checkout -b mybranch

(vi) git checkout branchname

It switches to the mentioned branch
but it only works for existing branches

(vii) git branch -m branchname

It will rename the branch we are
in to the name mentioned in the
command

(viii) git branch -c branchname

It will copy the branch we are
in and create a new branch
with the same copied data.

(ix) git branch -d branchname

It will delete the mentioned branch

(x) git merge branch name (with in master
branch)

This will merge the contents with
the branch

This merger makes a new commit.

Ques What is commit?

A commit is a point in git which is treated as a "save point" which keeps records of all save points, so if we want to go back to the previous save point we can switch back.

* first we need to add the files to our git repository/staging area so that they can be tracked by git.

(xi) git add filename.

or to add the file in git tracked files used to stage the untracked files

(xii) git add -A or -all (very risky)
(never use it)

It will add all the files present in the repository.

(xiii) git add .

To add all files only in the current repository/directory

(XIV)

git commit -m "commit message here"

to commit the changes with message/comment

★

git add and commit both commands
are required to convert untracked
files to tracked files.

(XV)

git restore filename

to restore a deleted file back

(XVI)

git restore --staged filename

to remove a file from staging area
& put back to untracked files

(XVII)

git ~~diff~~ log

(XVIII)

git log

will give all the commits

(XIX)

git revert ~~hash number~~ log

It will revert the commit

(xix) git clone http://url

to copy repository content from github
to new place or on local.

(xx) git push origin main

pushes the files and changes to my
github master/main branch.

(xxi) git push anyotherbranchname

pushes to the branch ~~anywhere~~ on github
mentioned in command

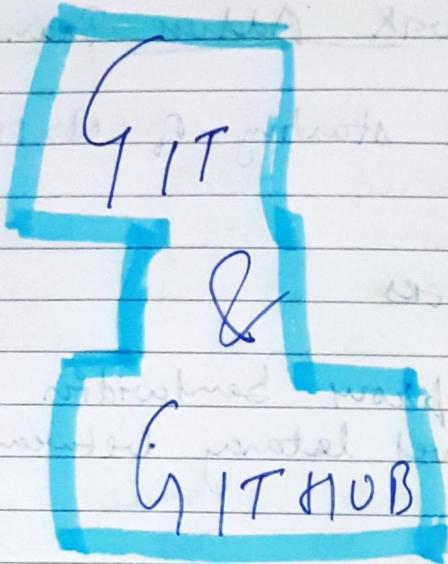
(xxii) git pull origin main/ or any repo

The git pull command is used to
fetch and download content from a
remote repository and immediately update
the local repository to match the content.

(xxiii) git ~~copy~~ myfilename Newfilename

To rename the file.

continued again



Continued

- (xiv) git remote add origin remote URL
- ↓ ↓
remote name remote repository
 URL

This command is used to add a remote repository to the local git repository.

- (xv) git remote -v

to check/verify if new remote is added or not.

(xvi) `git remote rename` origin destination

↑ ↓
existing new
remote remote
name name

This command is used to rename a remote.

(xvii) `git remote rm` origin

*
remote
name

This command is used to remove/unlink a remote repo from local.

(xviii) `git fetch` origin

↑
remote
name

This command is used to fetch changes from a remote repository.

(xix) `git pull --rebase` origin main

↑ ↓
remote branch
name name

This will pull changes from remote repo to local and redo local commits on top of it.