Some helpful recalls for git commands:

● git init: initialize your repository

with git. git starts tracking

changes in your files

● git add [file]: add a file/file

change from your working

directory to staging area

● git commit-m: Commit a file to

commit history

● git remote add [remotename]

[remoteURLAddress]: Connect

your local repository to a remote

repository (example GitHub).

● git push: Push the current local

repository to remote

● git clone [URL]: Clone the remote

repository into a local repository

●Commit is adding the work to the

HISTORY of the works you are doing.

You can move back and forward in your

commits.

Open source projects are projects which are initiated by a developer or a team of developers and made open so that anyone can contribute to its development and maintenance. Such projects are made by collaborative efforts of several people. There are a lot software’s we use today which started as open source projects!

Some of the popular open source projects which you might have heard of:

● Android Operating System

● VLC media player

● Firefox Web Browser

Git divides our work area into 3 stages:

1. Working directory: Where we

are creating and making edits to

our files.

2. Staging area: When the files are

ready to be committed, we have

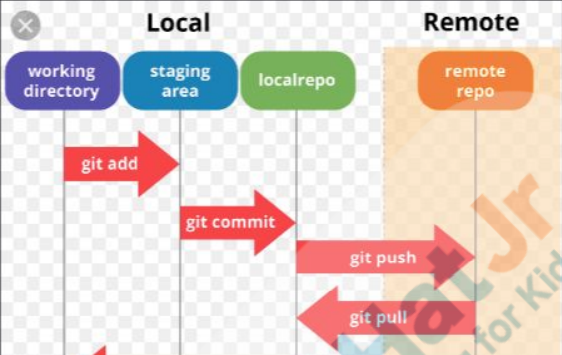
to stage them first.

3. Local Git Repository: After the

files are committed, they become

a part of commit history in a local

repository.



Step 1: mkdir mystory and git init mystory

Step 2: Go to the mystory folder manually and create two files “Characters” and “Story” using vi <file name>

Step 3: To quit editing press “esc :wq”

Step 4: Check git status

Step 5: git add Characters Story

Step 6: git commit-m "a message"

Step 7: again make changes in any one of the files

Step 8: git diff –staged (This will show you the difference between the last committed file and the currently staged file)

Step 9: again commit the file

Step 10: to check the number of commits make you will write git log

Step 11: There is also a commit id given to each commit. You can use the id or even the first 5 characters of the id to move back to that commit using git checkout <commit id>

Step 12: You can create a new branch using git branch <branch name> (branch name cannot have a space in between)

Step 13: git checkout <branch name>

Step 14: git log

Step 15: At any point of time, you can abandon this branch and move to the master branch using 'git checkout master'. It will take you to the last commit you had made in the master branch.

Step 16: You can check the difference between the lines in the HEAD of the two branches using git diff branchA branchB HEAD here refers to the latest commit of the two branches.

Step 17: create empty repository on GitHub

Step 18: git remote add [remoteName] [remoteURL]

Step 19: git push [remote name] master

Step 20: Manually change in GitHub and commit changes

Step 21: Now, we will not be able to make any new pushes to the remote repo since our local repo is out of sync. We can use git pull to update our repo with the upstream repo. Teacher shows how to use git pull [remoteName] [currentBranchName].

Forking a repository means Forking creates a duplicate of the repository in your own account where you can work and modify the contents of the project.