Centralized system

Decentralized system

**-m <message>**

**Sets the commit's message.** Make sure to provide a concise description that helps your teammates (and yourself) understand what happened

#### -a

**Includes all currently changed files in this commit.** Keep in mind, however, that untracked (new) files are not includedIf you have lots of changed files in your working copy - and want all of them included in the next commit - you can make use of the "-a" parameter and thereby omit the "git add" step

The "--amend" option comes in handy, for example, when you mistyped the last commit's message or forgot to add a change. The following example will correct the very last commit by overwriting its message and adding another change:

git add forgotten-change.js

git commit --amend -m "New commit message"

**In Git how do you revert a commit that has already been pushed and made public?**

Why Git:-

1. Open source
2. Massive scale
3. Most operations are local so its fast
4. Active community

Three state of Git= working dir, stage area, commit (repository)

Git staging area = git index

Master branch

Cloning a repository:- git clone <http URL>

Direct commit= git commit –am “commit message” == this will add the files to staging area and will do the commit also

Suppose you are adding files recursively then to add that to stage area command in got add .(dot)

Here .(dot) will help in adding recursively.

If you want to unstage a file which you added to staging area command is = git reset HEAD <file name>

Suppose you want to old version of the file then command is = git checkout -- <file name>

**Git Log:-**

To get the details of a specific file command is = **git log - - <file name>**

To get the details of a file which was renamed command is = git log -- follow -- <path of the file>/<file name >

**Git Alias:-**

Git config -- global alias.hist “log – graph --decorate --oneline”

Here if we type git hist it will show us the details history.