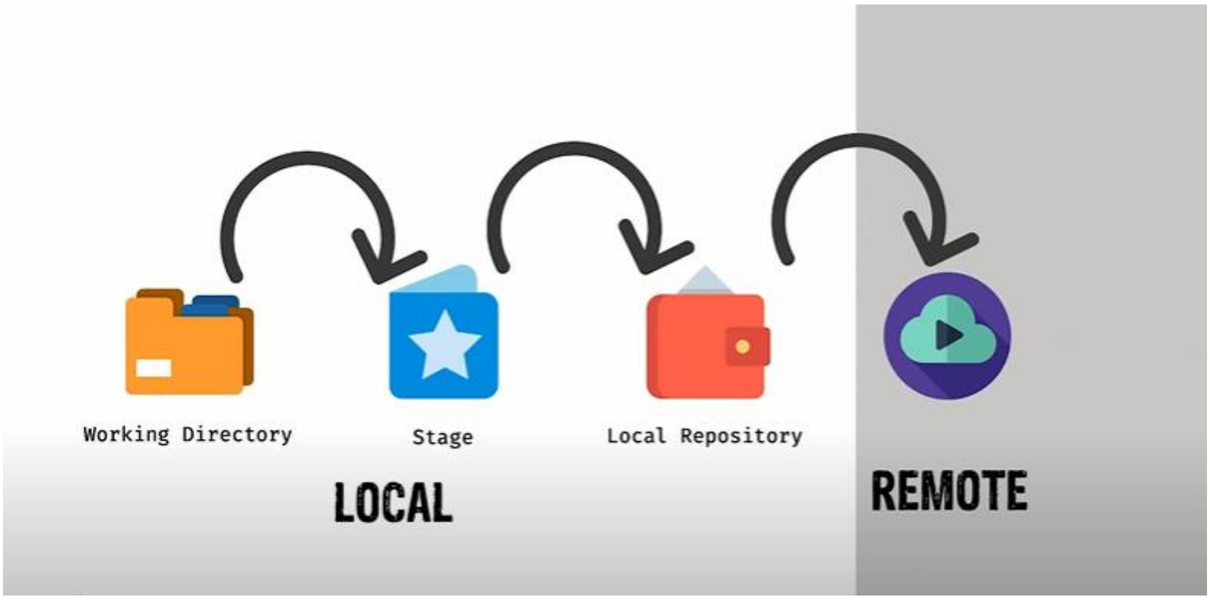
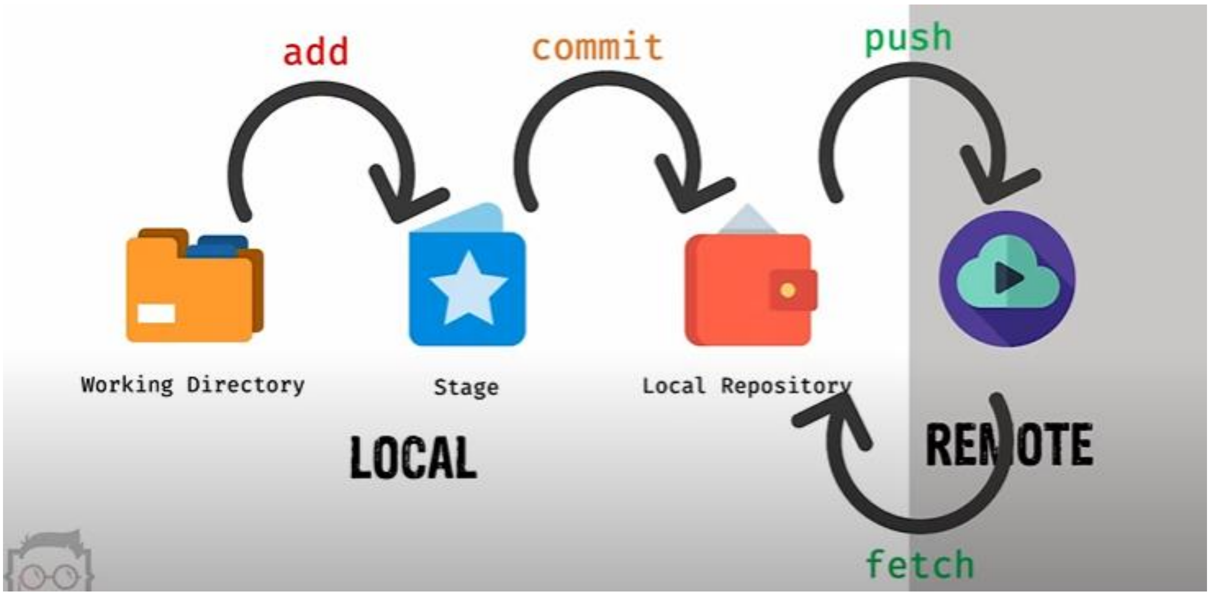
## Creator of git - Linus Torvalds

## Basic Concept of Git

**Working Directory -> Stage -> Local Repository -> Remote**





**Add ->** Send the changes to the stage
**Commit ->** Finally determine to commit the changes from the stage
**Push ->** Send the changes of local repository to the remote repository in (GitHub)
**Fetch ->** Fetch the changes of remote repository (GitHub) to the local repository but file not reflect to the working directory
**Pull -> (Fetch + Merge) ->** Update the changes and reflect the file

# Git Basic Command

1. **git init** - initialized an empty git repository and a local repository
2. **git clone URL (**https://github.com/jabed15578/gitone.git**)** – clone a repository from remote / cloud (GitHub) by using URL
3. **git status –** show the what changes in git folder
4. **git add –all/-A –** add all file to the stage from all folder
5. **git reset –** restore to Unstage / reset the stage / Unstaged changes after reset
6. **git add . (Recommended to use all time) –** add all file to the stage from only the current working folder or present working folder
7. **git add * -** add all file to the stage without a deleted file (if one file was deleted than it will not to the stage by this command but all other file will be staged)
8. **git add filename.extension –** add an individual file to the stage
9. **git add *.extension –** add all file to the stage with this extension in current working directory
10. **git commit -m "message" –** commit file finally with a message from the stage
11. **git reset HEAD~ (work after commit) -** Reset all track and return to working directory (this command work after committed else not)
12. **git reset --hard (work if remove a file manually and want to return back) –** return the changes also return the remove file hardly
13. **git rm filename –** Remove a file and send the changes to the stage
14. **git rm filename -f –** Remove a file with forcedly if it not wants to be deleted (Occur if there have any change in file)
15. **git rm --cached filename –** send the change to the stage but not remove the file on the other hand **git rm filename -f** send the change to the stage also remove the file from the repository
16. **git rm folder-name –** remove a folder from a repository
17. **git rm -r folder name –** remove all folder recursively that means remove folder/folder/folder……

## Branching With Git

18. **git branch –** to know how many branches we have in our repository / application
19. **git branch branch-name (Ex- git branch development) –** create a new branch in our repository
20. **git checkout branch-name (Ex- git checkout development) –** switch into branch (must be committed before switch to another branch)
21. **git merge branch-name -m "message" (Ex1- git merge master -m "merging on development branch with master branch") (Ex2- git merge development -m "merging on master branch with development branch")-** merge with one branch to another branch in a project

## Git Push, Fetch and Pull

1. **git push origin branch-name (Ex1- git push origin master) (Ex2- git push origin development) (Ex3- git push origin staging)–** push a branch with all changes to the remote repository (GitHub)
2. **git fetch –** get the changes of remote repository to the local repository (but not update or impact file in working directory)
3. **git merge –** update or impact file of local directory after **git fetch**
4. **git pull – (git fetch + get merge) –** get changes of remote repository and also update the changes of file in working directory