**Git Tutorial**

**GIT:**

* Git is a version control system.
* It is software.
* It is installed locally on the system.
* Git is used to save our project code files and other related documents in Remote Repository which is hosted on the web.
* Git allows Developers to collaborate and work on different parts of the project like adding features or bug fixes and finally merge into the main code base.

**GIT HUB:**

* Git hub is service that is hosted on the web.
* It is used to save project files remotely from local system.
* Developers or users can download these files from anywhere through the internet incase if our local system has been failed.

\***Git init\***

**Git init**

>this command is used to create local empty repository.

\***Git status\***

**Git status**

>this command is used to check the status of the files whether they are in working directory, modified, or in staged area.

**Tracked and untracked**

**Tracked:** Files which are staged, modified and unmodified are tracked by git.

**Modified:** when the files are staged and make changes in working directory then those files are modified.

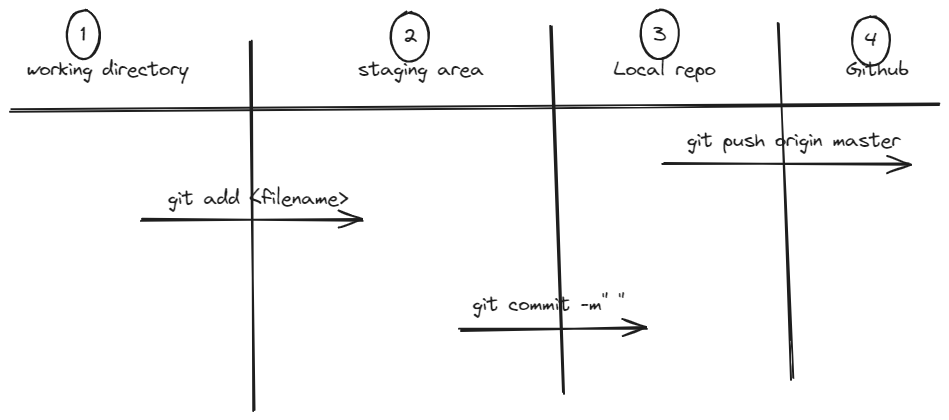
**Unmodified:** files which are not modified is known as unmodified.

**Changes to be commited:** this notification tells that the files which are present in staging area are ready to commit.

**Changes not staged for commit:** this notification tells that the files in working directory has to be staged for commit.

**Untracked:** Files which are present in working directory but git can’t track those files is known as untracked files.

**Work flow between Git and Github**

****

* If we want to add the files from working directory to staging area the you can use command.

To Add single file:

**Git add <filename>**

To Add all files:

**Git add .**

(or)

**Git add -all**

* If we want to commit files from staging area to local repository this commit will save all files which are present in staging area will commit into local repo.

**Git commit -m” this is description for understand”**

**-m:** for commit message.

* If we want to move from local repository to Remote repository branch this command is used to send files from local master to remote master.

**Git push origin master**

Note: git remote add origin https://github.com/prudhvinath-buddhiraju/practice.git

this command is used to establish the connection between Local Repository and Remote Repository.

\***Git Diff\***

* Git diff command is used to compare the content of the files which are present in working directory and staging area and staged area file.

**Git diff:**

* Git diff command is used to compare the content between files which are present in working directory and staging area.

**Git diff**

* Git diff –staged command is used to show the content of the file which is in staged.

**Git diff –-staged**

(or)

**Git diff –cached**

\***Git log\***

> this is used to check the history of the commits.

>Git log is used to check the history of the commit like author name, commit id and the

Commit date and time.

**Git log**

>to display the commit log history in single line.

**Git log –oneline**

>to display the commit log history in graph.

**Git log –graph**

>to display the last 3 commits.

**Git log -3**

**\*Git stash\***

Git stash is used to save our untracked and tracked and modified files in temporary place for later use.

>Git stash push and Git stash commands are used to send those files to the stashing area.

**Git stash**

(or)

**Git stash push**

>To list stash

**Git stash list**

>To revert back the top most file in stash.

**Git stash apply**

**>**To delete the stash

**Git stash drop**

**>**To revert back specified file based on stash id

**Git stash apply stash@{3}**

>To retrieve and drop the stash file which is on the top

**Git stash pop**

**>**To retrieve and drop the specified stash file

**Git stash pop stash@{2}**

\***Git cloning\***

**>**git cloning is used to copy the files from the existing remote repository to Local Repositroy.

>to copy the files from existing remote repository to local repository.

**Git clone <url>**

>to copy the files from existing remote repository to local repository with different directory name.

**Git clone <url> <directory-name>**

**\*Git cleaning\***

Git cleaning is used to remove the working directory files permanently

>to delete the files in working directory forcefully

**Git clean -f**

**>**to delete the folders in working directory

**Git clean -d**

>to delete the files which are present in the .ignore file

**Git clean -x**

**>**if we want to delete the files, folders and files present in .gitignore file then we have to use

**Git clean -f -d -x**

**\*Git tagging\***

if we want to assign the tag for commit history then we will use this git tagging.

>to create a tag.

**Git tag <tagname>**

**>**this command is used to create a tag and provide information.

**Git tag -a<tag-name> -m “description”**

**>**to list all the tags

**Git tag**

>to delete a tag

**Git tag -d**

>this command is used to assign tag for specific commit.

**Git tag <tagname> commitid**

**\*Git Branch\***

Branches that are used to work on different parts of the project and they will merge into the main branch which consist of code base when they work on bugs or adding features.

>To create a branch

**Git branch <branch-name>**

(or)

**Git checkout -b <branch-name>**

>To switch into the branch

**Git checkout <Branch-name>**

>To delete a branch

**Git branch -d <branch-name>**

>To rename a current branch

**Git branch -m <rename branch>**

**>**To rename a branch currently not in

**Git branch -m <old-branch-name> <new-branch-name>**

>To list all the files that are present in branch firstly we have to switch to that particular branch and later use the command.

**Git ls-tree –name-only**

(or)

**Git ls-files**

**\*Git push\***

Git push is used to push the files from local repository to git hub.

**Git push origin master**

**-push:** push the file from local repo

**-origin:** Default remote name

**-master:** it is the branch name in git hub

**\*Git pull\***

Git pull is used to pull the updated commit files from remote repository to local repository.

It directly download files from remote to working directory

It is the combination of fetch and merge

**Git pull origin master**

**-pull:** pull the files from remote repo

**-origin:** Default remote name

**-master:** it is the branch name in git hub

**\*Git fetch\***

Git fetch is used to fetch the files from global to local but it does not download the files directly

**Git fetch**

**\*Git merge\***

Git merge is used after fetch the files from remote to local later we have to use this command to merge the files into working directory.

**Git merge**

\***.gitignore\***

**.**gitignore is a file that can be used to tell git to untack the specified files inside the .gitignore.

Inside the .gitignore we will specify

\*.txt -> it ignores all the current future .txt files

Sample.txt -> it will ignore sample.txt file.

**html/**: Ignores any directory named html anywhere in the project.

**/html**: Ignores only the html directory at the root level of the project.

**Handling conflicts:**

When a single file has been edited by multiple users or developers then conflicts will be occurred.

>so we have to pull the updated changes from remote to local repo and do the modifications or changes in the file and push into the remote repository.