# Git and GitHub

**Git:** It is a version control system that helps developers to keep track of changes in source code during software development. It records snapshots of users' files. It supports distributed and non-linear workflows.

## Main Commands:

- **git - -version**: checks which version of the tool installed in the system
- *Configuration:* configures the user, this lets other users know who made the changes in the file.
    - **git config --global user.name <"*username*">:** configures user name.
    - **git config --global user.email <"*email*">:** configures user email id.
    - **git config --global --edit:** changes username and email id.
- **git init:** initializes an empty repository.
- **git add <*filename*>:** adds file to a staging/ holding area. It is an intermediate stage before a user pushes the file to their cloud repository.
- **git status:** shows the current status of the files present in git folder i:e, files that have been modified, files currently in staging area and files that are not present in the remote repository (untracked files).
- **git commit –m <"*message to be displayed*">:** commits the changes that would be made in the remote repository.
- **git log:** allows the user to track all the changes that have been made by the collaborative authors.
- *Branches:* These form a tree like structure where each node represents a particular feature and top node, known as master, culminates all the features into a program.
    - **git branch <*branchname*>:** creates a new branch.
    - **git branch:** to see which branch the user is currently accessing.
    - **git checkout <*hashcode*/*branchname*>:** allows the user to access a particular branch or return files to a state before particular commits had been made.
    - **git checkout –b <*branchname*>:** creates a branch and allows the access to the same.
    - **git merge <*branchname*>:** allows the user to merge files of another branch with the files of the current branch.

- **touch .gitignore:** To create an ignore file that holds the names of all the sensitive files and folders. It prevents the user from unintentionally committing the sensitive files to the remote repository.
- *Push Files:*
    - **git remote add origin <*link to remote repository.git*> :**creates a link to the remote repository where the committed files will be stored.
    - **git branch –M <*branchname*>**: specifies the branch of the remote repository.
    - **git push –u origin <*branchname*>:** moves committed files to the selected remote repository branch.
- **git clone:** allows the user to create local file of the forked repository.
- **git pull:** merges changes made in the remote repository with that of the local file.

## Additional Commands:

- **git reset <*filename*>:** unstages the file from the staging area.
- Comparison:
    - **git diff:** shows files that have been modified but not staged.
    - **git diff --staged:** files that are staged but not committed.
    - **git log branch1…branch2:** commits on branch 2 that are not in branch 1.
    - **git log --follow <*filename*>:** gives history of changes made to the file.
    - **git diff branch1…branch2:**shows files in branch 2 that are not in branch 1.
- **git fetch <*alias*>:** fetches all branches from remote repository.
- **git rm <*filename*>:** removes file from the staging area.
- **git mv <*current path*> <*new path*>:**changes path of the file.
- **git log --stat –M:** shows log of committed files along with the path changes.
- **git rebase:** applies all commits of current branch on top of the specified branch.
- **git reset --hard <*commit*>:** clears staging area and rewrites tree from a specific commit.
- Temporary commits:
    - **git stash:** saves modified and staged changes.
    - **git stash list:** lists stack order of stashed file changes.
    - **git stash pop:** returns the top of the stack list.
    - **git stash drop:** discards changes from top of the stack.

**GitHub:** It is web service that hosts git software. It incorporates functionality of git, as well as, adds its own functionality.

**GitHub Functions:**

- **Invite collaborators:** adds collaborators which allow the users to make changes in the existing files.
- **Fork:** creates a copy of other user's repository in the user's github.
- **Open pull request:** allows the user to create a pull request.
- **Create pull request: s**ends a pull request from the secondary user that let's the primary user know of the changes made in the forked repository.
- **Merge pull request:** merges the secondary file with the primary file.