**Git And GitHub**

Git

Download and install Git

<https://git-scm.com/downloads>

GitHub

Signup and Create Account

<https://github.com/>

**Version Controlling System (Tool)**

1. Is use to manage the code version (File version management).
2. Using VCS tool you can integrate the code from the multiple developer.
3. The after every commit the code version will be manage.
4. You can easily switch back to an older version.
5. Along with the file version it also maintains the metadata of the file such as the owner, version, date time, messages actual changes.
6. There are two typed of VCS tool
   1. Centralized VCS
      1. The File and metadata will be store into a single central system from where every user will be use the files.
   2. Distributed VCS
      1. The file and metadata will be store into a single location from where every user can get the file also every user will be a server as well there the file and its metadata will be store.



**Git Bash**

1. It is command line tool to execute git command
2. Linux based command tool
3. You can also use a git UI toll to achieve the same thing.

**Git Command**

**Git Configurations**

**git config --list**

You can get the all-configuration list.

Git User name and email

**git config --global user.name <Username>**

**git config --global user.email <UserEmail>**

**Commands to setup the user and email globally.**

**Create a Empty Local Repository**

1. **git init**

You can create a git empty local repository. After execution of this command you can see the .git folder which is used by git for the further execution.

**Git file/folder Operations**

Git Manages the version for the file. Git has 2 file status

Untrack file

It is file for which git do not have any prior details. These files are normally a new file. You use a git command to change the file status to track.

Track File

It is for the files whose prior details (version) already available with git. If you make any modification into existing files then those files come under track status. There are different track file status

New file

Edited file

Rename file

Removed file

**git status**

you can check the status of the files using this command

**git add**

Using this command you can add the files into git working area.

Git add command will be use to add single file or all the files at a time.

Syntax:

git add ‘filename’ : This is to add single file at a time

git add **.** : this is use to add all the files at a time.

**Git commit**

This command is use to create aversion for the changes which is added inside the git working area. After this the file changes, and also file metadata will be generated by the git.

**Syntax:**

**git commit -m “message”**

**Git Checkout**

This command is use to switch between the different version of the code or in the different branches.

Syntax:

git checkout <commitId> : To Switch into another commit

git checkout <branch name> : To Switch between another branch

**Branch**

1. It is use to differentiate you original code from the ongoing changes.
2. Using branches you can create logical copy of the original code and use the copy for further modification.
3. You can also merge the changes of the one branch another.
4. By Default On Git you will get a “master” branch and on GitHub you will get an “main” branch.
5. To Get the list of all branches you can use following command

**git branch** : Using this command you can get the list of branch.

1. To Create a new branch

**git branch <branch name>** : This command is use to create new branch.

**Merge**

1. In this process you can merge the changes of one branch into another.
2. To do this you have to follow steps
   1. Switch into a branch where you wanted to merge the changes
   2. Use Git merge Command to merge the changes

**git merge <branch-Name>**

**GitHub**

1. **To Connect Local repository to remote/cloud repository**

git remote add origin <Remote\_Repo\_URL>

1. To Push the local repository changes into remote/cloud repository.

git push

1. To get the changes done one remote repository to a local repository

git pull