Git & GitHub

**Version Control**

* Centralized Version Control
* Decentralized Version Control – Cloning, Working Copy, Push/Pull

**What is Git?**

**Documenting the history of your project**

* Version Control
* Time Machine
* Checkpoints (commits – with messages)
* Multiverse (branches)
* Synchronize (merging)

**Git Environments**

* Working
* Staging
* Commit

**File States**

* Tracked
* Unmodified
* Modified
* Staged
* Untracked – New files added since the last commit

**Git Flow**

* Create a copy of the main branch
* Make changes
* Merge to main
* Delete the newly created branch

Notes

Git tracks all the changes w.r.t. the last commit.

Moving files in git means you are deleting a file and adding new files.

# **Git Commands**

1. Configuring Git

* git config --global user.name “Name”
* git config --global user.email “Email”

1. Initializing a directory with git

* **git init**

**NOTE:** ls -la to view the hidden .git directory

1. Staging any changes made

* **git add filename.xyz** // Staging a single file
* **git add filename.abc filename.xyz** // Staging multiple files
* **git add .** // Staging all the files/changes
* **git add mydirectory/** // To stage all the changes in a specific directory

1. De-stage any changes

* **Git -S restore . (or file name instead of period (.))**
* **Git restore . (restore previous version of file in the file system)**

1. Commit the staged changes

* **git commit -m “Your Message”**

1. Show all the commits

* git log
* git log --oneline

1. Stage of our working directory

* git status

1. Preventing git from tracking a file/directory

* Create a file name “.gitignore” and add the file or directory name in it

1. Seeing the difference/changes in file w.r.t last commit

* git diff

CHANGING HISTORY

1. Changing a commit message

* git commit –amend (ecs -> `:wq` + `Enter`)

1. Rebasing

* git rebase -i –root

1. Going back to some previous commit

* git reset *commitCode*
* git reset --hard commitCode

BRANCHES

1. Viewing all the branches

* **git branch**
* **git branch -r** // List all remote branches
* **git branch -a** // List all the local and remote branches

1. Switching a branch

* **git switch branchName OR git checkout branch-name**
* **git switch -c new-branch-name OR git checkout -b new-branch-name** // Create a copy and switch to a New Branch

1. Merging 2 branches

* git merge branchToMergeFrom-Name

1. Renaming a branch

* **git branch -m new-branch-name** // Renaming current branch
* **git branch -m old-branch-name new-branch-name** // Renaming any existing branch

1. Delete a branch

* **git branch -d branch-name**
* **git branch -D branch-name** // Force deleted a branch