Git Basic commands cheat sheet

- Git help [Command]
- git config --global user.name <user_Name> : set git user name globally at system level.
- git config --global user.email <user_Email> : set git user email globally at system level.
- git config user.name <user_Name> : set git user name locally at project directory level.
- git config --global user.email <user_Email > : set user email locally at project directory level.
- git config --global --unset user.name : reset git user name.
- git config --global --unset user.email : reset git user email.
- git Init : This will setup git related configuration on folder [.git folder will be created].
- git Clone <repo_Link> : If you want to get a copy of an existing Git repository. (example : git clone https://github.com/Abc-Test /exampleRepo -Repo.git)
- git remote -v : You will get current repo git url.
- git remote show origin : This command will give more information about remote repo and its branches.
- git add <file_Name> : Adding files to staging area.
- git add . : Add all files In current directory to staging area.
- git add <file1> <file2> : Adding multiple files.
- git restore <file_Name> : remove file from staging area or retrieve back.

- git status : displays current working directory, branch and staging area.
- git commit -m “short_message”: commit changes to local directory.
- git log : This will get u HEAD information that will tell which branch you are in currently.
- git log –oneline : This will list all your commits.
- git log –oneline –graph –decorate –all : Here you can see all commits in visual design using ASCI art.
- git log –stat : detail level expiation about committed files.
- git log -p : list all project commit history.
- git rm <file_Name> : remove file from staging and commit both at a time.
- git rm –cached <files> : stop tracking this files but keep in working area. [after this you can delete file directly].
- git fetch : fetch the latest changes from remote repo.
- git pull : pull the new changes from git repo.
- git push : Push the committed changes to remote repo.
- git branch <new_Branch_name> : Create new branch.
- git checkout <branch_name> : change/switch to another branch.
- git checkout -b <branch_Name> : create new branch and switch to that branch.
- git switch <branch_Name> : switch to branch.
- [merge] -> git checkout <source_Branch> : go to the branch where you want to merge the feature branch.

- [merge] -> git merge <destination_Branch> : the branch which you have to merge.

- git branch -d <branch_Name> : Delete the branch.
- git reset --hard <branch_Name> : reset the commits.[be careful with this you may loss the changes]

- git reset –-soft HEAD~1 : Undo the file from committed area to stage area. [HEAD~1 – commit order or give number which you want to undo the changes.]

- git revert <commit_ID> : revert this commit.