GIT

* Git init //initialize local git repo
* Git add <filename> //add the files to staging area
* Git status //it will show difference staging area and working area
* Git commit //it will taking to index from staging area
* Git push // it will push to remote repo
* Git pull //it will take changes from remote repository and update
* Git clone //cloning to new dir
* Touch <filename.txt> //it will create local index files
* Git config --global user.name “\*\*” //giving name
* Git config –global user.email “\*\*” //giving email ID
* Git rm –cached \*\*\*\*.txt //removing from staging
* Git rm –cached \* //removing all staged files
* Git add . //adding all the file to staging area
* Git commit -m “message” //committing all the file with delta message
* Touch .gitignore //excluding file ,if we do not want to commit
* Git branch <branchname > //creating slave
* Git checkout <branchname > // logging to branch
* Git branch -d <branchname> //deletes the branchname ,gives an error if uncommitted files available
* Git branch -D <branchname> //deletes the branchname ,will not gives an error if uncommitted files available
* Git checkout -b <branchname>
* Git remote //shows the connected remote repo
* Git merge branchname //merging branch to master
* Git remote add <NAME> <url> //adding the remote repo URL
* Git push -u <remote **NAME>** master //
* Git push -f <remote Name> master //force all the files in local to remote
* Git restore –staged dir3 //removing all staging files from directory

Advanced commands:

1. Git stash \\ to save changes made when they are not in a stage to commit them to a repository.
2. Git stash -u
3. Git stash list
4. Git stash show
5. Git stash apply
6. Git log
7. Git rebase
8. Git revert

**06-feb-2020**

Git diff \\ changes applied between the previous and current file

Git diff <hash1><hash2> \\ difference between commits

Git commit –amend -m “message” // change last commit message

Git commit –amend \\ commit with the existing \ previous commit

Git log –stat

Git cherry-pick

Git reset

Git reset –soft <hash> [\\ revert](file:///\\revert) to any log

Git reset <hash>

Git reset –hard <hash> \\ deleted all tracked files

Git clean -df [\\ untracked](file:///\\untracked) -d directories and -f files

Git reflog

test