GIT COMMANDS

* GIT USER DETAILS INTIALISATION
* To create user details (name) : git config --global user.name “name”
* To create user details (email) : git config --global user.email “email”
* To display user detail name : git config --global user.name
* To display user detail email : git config --global user.email
* To see all details : git config –list
* GIT REPO INTIALISATION
* To intialise git repo : git init .
* To list all files : ls
* To list all hidden files : ls –a
* To exit from git bash command line : press ESC and type wq
* To insert text into git bash command line : press INSERT
* GIT COMMIT TO LOCAL REPO
* To add files to staging area : git add --all
* GIT STATUS
* To see status : git status
* To see details of commit : git log
* COMPARE FILES IN GIT
* To compare files in working directory with staging area : git diff
* To compare files in working directory with REPO : git diff HEAD
* To compare files in staging area with local repo :

git diff --staged /git diff --staged HEAD(HEAD=>LOCAL REPO)

* To compare latest commit with previous commit :

git diff < latest commit id> <previous commit id>

* To view changes pushed in a specified commit : git show
* To view who has worked on a particular file :

git annotate <filename>

* WORKING WITH REMOTE REPO
* To push commited files to remote repo :

git push origin <branchname>

* To pull commited files from remote repo :

git pull origin <branchname>

git pull = git fetch + git merge

git fetch = sync the remote repo to local repo

git merge = sync the local repo to working directory

* To clone a remote repo : git clone <URL>
* When pushing a local repo to github using terminal must specify:

git remote add origin <URL>

* ADDING KEYS TO GIT
* Please see the link below : <https://docs.github.com/en/free-pro-team@latest/github/authenticating-to-github/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>
* WORKING WITH BRANCHES
* To create a branch : git branch <branchname>
* To list branches : git branch
* To switch between branches : git checkout <branchname>
* To checkout and create between branches :

git checkout –b <branchname>

* To create tag : git tag
* IT IS BEST PRACTICE TO PERFORM PULL BEFORE MERGING TWO BRANCHES
* WORKING WITH CHANGES OF GIT COMMITS
* To revert changes from working directory :

git restore <filename>(will not work for newly added files)

* To revert changes from staging area :

git restore --staged <filename>

git restore <filename>

* To revert changes from local repo :

git reset HEAD~

git restore <filename(s)>

* To squeeze multiple commits into single :

git rebase –i HEAD~4(commit no)