**USEFUL SHORTSCUTS AND CMD**

**Untracked > Unmodified > Modified > Staged > Commit**

**use '...'** if file has spaces in it

**git config --global user.name "Waleed"**

**git config --global user.email "waleedahsan101@gmail.com"**

**git --list** (check complete ID details)

**git config user.name / email** (to check id)

**git config –global alias.st status** > to config any shortcut

git config --global alias.unstage "restore --staged" > git unstage *(shortcut to unstage)*

**pwd**-(check)present working directory

**ls** (to list files in folder u r also)

**cd /...** (to open a specific directory)

**git init** (make .git) / **rm -rf .git** (Del .git)

**git status**

**git add** --a OR **git add "...."**

**git restore –staged xyz.abc** > to unstage

**git restore file.abc** > restore changes in a file

**git checkout -- xyz.abc >** unmodify a file if done by mistake (Ctrl Z of file but will only work on files staged before)

**git checkout –f** > To get files in previous commit state

**git commit -m** "third commit"

**git commit -a -m ""** > used to skip staging area However untracked/ new files that are not staged before won’t be added

**git commit –amend** > amend last commit using wim editor (I=insert, esc=ok, :wq=exit editor)

**git log**

**git log –p** **-1**> To Check one last commit (**Add 2/3 to check 2/3 commits)**

**git log --stat** > Check overall summary of all commit

**git log –pretty=short/full** > Overall short summary commits (Author/Author+commit)

**git log –pretty=oneline** > summary commits one liner

**git log –pretty=format:”%h -- %an”** > Show commit in req format (hash—auth name) shortcuts https://git-scm.com/docs/git-log

**git log –since=2.days** > Commit details of 2 days (days/moths/years)

**git rm .....** > remove and stage

**git mv first.txt first\_1.txt** > rename and stage

**git rm --cached xyz.xyz** > To remove a file from being tracked (in case u want to add in .gitignore)

**touch waleed.log** > used to add a folder

**touch .gitignore** > file will be created in which u can add files that u dont want to be tracked and ignore (if same extn files use \*.extn)

**git diff** (to check differnece btw staging area and working dir)

**git diff** **--staged** > diff btw staged data and previus commit data

**git clone https://xzy.com/abc.git "name of folder"** (To clone complete repository)

**Starting Remote/Generate SSH key/ Add SSH to Git hub/ Push to GitHub**

**git remote** > create a remote

**git remote –v** > To check remote push and pull link

**git push –u origin master** > to Push in repository and keep track (replace ***master*** with***master:waleed***to make waleed github repository)

git push origin develop > create and Push branch develop

**git push –d origin develop** > Delete branch ‘’develop’’ from github

**Generate SSH Key and add to SSH Agent** > <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

**Add new SSH Key to github acct** > <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account> (RUN: “tail < ~/.ssh/id\_ed25519.pub” to show in git bash or else use clip to copy to clipbord)

**Create and switch branch**

**git checkout –b develop >** Create new branch namely ‘’develop” (-b is added to develop new branch)

**git checkout master** > back to master branch

**git checkout develop** > if already created

**git branch** > check total and current branches

**git branch –v** > Branch + commit hash + msg

**git merge develop** > Merges changes committed in develop with master

**git branch –merged** > Shows Merged br

**git branch – nomerged** > br not merged

**git branch –d develop** > del a branch *(If not merged will give error to del, hence add –D instead of -d)*

***NOTE:*** *merge conflict (If changes are made on both branches of same nature, option to select the change is given)*