**Git Commands.**

Use GitBash Tool for windows [http://gitforwindows.org/]

1. **git init** 🡪 initialized new git repository/[create new local git repository]
2. **git clone urladdress**🡪copies remote git repository to local repository
3. **git branch [newbranch]** 🡺 This is for creating newBranch to work with
4. **git branch** 🡪 To List All the local branches
5. **git checkout [branch\_name]** 🡪 To checkout and work with local branch
6. **git status 🡪** to check the status of the current branch
7. **git add <file>** 🡪 To stage the specific file for preparation for commit.
8. **git reset HEAD <file>** 🡪 To ignore local changes for the specific file
9. **git commit -m "Commit message"** 🡪 To commit the changes
10. **git push origin master/branchname** 🡪To push the branchname to the remote repository
11. **git pull origin v-sesiga/CP** 🡪 to pull the remote branch
12. **git remote: 🡪 T**o check the remote branches
13. **git stash** 🡪 (to save your current work)
14. **git diff🡪** To check the difference on the committed changes
15. **git pull** 🡪 To pull from remote the current branch/synch up.
16. **git rebase origin/master-🡪** fast forward the current branch to origin/master skipping history of commits.
17. **git stash –help 🡪** To get help and check various options for stashing
18. **git stash clear**🡪 To clear current stash.
19. **git stash save "message" 🡪 To save the current changes in the stash message.**
20. **git stash apply stash@{3} 🡪 To apply the local stash changes on the current branch**
21. **git stash pop** 🡪 getting the latest stash.
22. **git tag** --🡪 to create tag readonly branch for deployment
23. **git diff** 🡪 show changes between commit
24. **git branch -D testbranch 🡪 To remove the testBranch**
25. **git merge –abort** 🡪 to abort the merging
26. **git checkout feature1** -🡪To checkout New Features.
27. **git merge develop** 🡪merges develop branch to the current/working branch
28. **git commit -a -m "Modify changes"** 🡪 stage and commit at the same time.
29. **git reset --hard/soft HEAD^ 🡪** To reset the staged changes.
30. **git commit --amend -m "New Mess"** 🡪 to amend the commited message.

For more refer, the documentation. <https://git-scm.com/documentation>