**Git Basic Commands For Beginners**

1. **git init:** initialize git on a folder
2. **git status:** track of recent changes on file
3. **git add *filename.extension*:** moves specific file to staging area
4. **git add. / --all:** moves all new files to staging area
5. **git log:** all recent commits
6. **git log --oneline:** all recent commits in short form
7. setting up username and email for current project:
8. **git config user.name “*username*”**
9. **git config user.email “email”**
10. setting up username and email for all project:
11. **git config --global user.name “*username*”**
12. **git config --global user.email “email”**
13. **git commit -m “*commit description*”:** commit to local repository
14. **git checkout “*commit id from log*”:** go to different commit stage or branch
15. **git checkout master:** back to master branch
16. **git show “*commit id*”:** changes after that commit
17. **git diff:** latest changes on file that have not been committed
18. **git diff “*commit id*” “*commit id*”:** difference between two commits
19. **git diff --staged:** changes after staged(i.e. added)
20. **git rm “*filename*”:** delete a file from all commit
21. **git clone “*link*” *name*(optional):** clone a project on your device. If you put a name project will be saved as given name.
22. **git fetch:** bring all commit info to your local repository(not added yet)
23. **git pull:** bring all commit to your local repository(commits may be done by other people)
24. **git branch “*name*”:** create a branch with given name
25. **git branch:** show all branches
26. **git checkout “*branch name*”:** switch to specific branch to work
27. **git checkout -b “*branch name*”:** create and switch to new branch
28. **git merge “*branch name*”:** go to specific branch, write the branch name which you want to merge with current branch
29. **git branch -D “*branch name*”:** delete branch
30. **git stash:** put current works into a temporary repository and start project from beginning
31. **git stash apply/pop:** bring back last stashed codes/files
32. **git stash list:** isn’t this obvious?
33. **git stash pop “*stash name*”:** go to specific stash/temporary repository
34. **git clean -f -n(optional):** deletes untracked files, “-n” gives warning about the files
35. **touch .gitignore:** this will create a file named “.gitignore.txt”. Open the file and write “\*.extension”(extension of which type of files you want to hide or ignore from commit

**Useful Link**

<https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud>