1. To initialize local Git repository: **git init**
2. To check status: **git status**
3. To add a file into Git repo: **git add <filename.extn>**
4. To add everything in the directory into Git repo: **git add .**
5. To save or commit: **git commit –m “commit message”**
6. To add all ‘.txt’ files: **git add ‘\*.txt’**
7. To check the log: **git log**
8. To push our local *repo* to the remote server (GitHub, Bitbucket etc.), we'll need to add a remote repository, and to do so: **git remote add origin <link of the repo/RepoName.git>**

***[\*\*\*Note: Here ‘origin’ is the reference of the remote Git repository, it can be anything.]***

1. The push command tells Git where to put our commits when we're ready, and now we're ready. The name of our remote repo is ***origin*** and the default local branch name is ***master***. The ***–u*** tells Git to remember the parameters, so that next time we can simply run ***git push*** and Git will know what to do. The git command is: **git push –u origin master**
2. We can check for changes on our remote (GitHub/Bitbucket) repository and pull down any new changes by running: **git pull origin master**
3. We can take a look at what is **different** from our last commit by using the **git diff** command. In this case we want the diff of our most recent commit, which we can refer to using the **HEAD** pointer. The command is: **git diff HEAD** [\*\*\* Another great use for ‘**diff’** is looking at changes within files that have already been staged. Need to remember, staged files are files we have told git that are ready to be committed.]
4. To see the changes we just staged, run **git diff** with the ‘**–staged’** option: **git diff –staged**
5. To unstage file(s) in a location: **git reset <location/fileName.extn>**
6. Files can be changed back to how they were at the last commit by using the command: **git checkout -- <target>**
7. When developers are working on a feature or bug they'll often create a copy (aka. **branch**) of their code they can make separate commits to. Then when they're done they can merge this branch back into their main **master** branch. To create a branch: **git branch <branch\_name>**
8. To see the existing branches: **git branch**
9. To switch to a branch: git checkout **<branch\_name>**
10. To remove all ‘.txt’ files: **git rm ‘\*.txt’**
11. To ***merge*** the changes from a branch *(****<branch\_name>****)* into the ***master*** branch, checkout to the ***master*** branch first, and then run the command: **git merge** **<branch\_name>**
12. To delete a branch: **git branch –d <branch\_name>**
13. To push finally everything on to the remote repository after it is done: **git push**
14. To clone (or download) the remote Git repo into local: **git clone** **<link of the repo/RepoName.git>**
15. To update remote repo files with local, need to fetch or pull that remote repo first, so if the remote repo reference is ‘origin,’ then from local ‘master’ branch: **git fetch origin master**
16. To know the remote Git repo URL, if the ref is origin, one way is: **git remote get-url origin**
17. To know more about remote repo: **git remote show origin**
18. To push in a remote branch: **git push <link or ref of repo> <branch-name>**
19. To stage and commit only changed files: **git commit –am “commit message”**
20. The number (23) – action can also be done by running: **git pull origin master**
21. To set up username: **git config user.name “username”**
22. To set up email: **git config user.email “email@id”**
23. To verify username or email: **git config user.name** or **git config user.email**
24. To set up username for every repo on computer: **git config --global user.name**
25. To change: **git config user.name “New username”**
26. To cancel a commit and revert to previous point: **git revert <commit\_hash>**
27. To revert a range of commits: **git revert <oldest\_commit\_hash>..<latest\_commit\_hash>**
28. In the server, move the cursor back to the last known good commit: **git push -f origin <last\_known\_good\_commit>:<branch\_name>**
29. To know/show all the remote git repos and their refs: **git remote –v**