|  |  |  |
| --- | --- | --- |
| **GIT COMMANDS** | | |
| 1 | ls | list the files and folder (Linux command) |
| 2 | ls -a | it shows hidden files and folder also |
| 3 | cd foldername | entering into specific folder |
| 4 | cd .. | getting back to previous folder |
| 5 | git log | it displays all commits |
| 6 | git config --list | git configuration list |
| 7 | git config --global user.email "xxxxxxxxxx@gmail.com" | adding Email |
| 8 | git config --global user.name "xxxxx" | adding user name |
| 9 | git config --global --list | to check email and username and push status |
| 10 | git status | status of the commits |
| 11 | git add file1.ext file2.ext | adding file separately by naming it |
| 12 | git add . | adding entire path |
| 13 | git commit -m "commit title" | committing the file after adding |
| 14 | git push | pushing to central repository |
| 15 | git pull | pulling to local repository from central |
| 16 | git rm README.md | removing file |
| 17 | git log --oneline | it will show all commits in one line in current branch |
| 18 | git log --online -3 | it shows last 3 commits in one line |
| 19 | git checkout file.ext | we can bring back deleted file |
| 20 | touch .gitignore | it is Linux cmd which it will create an empty file (in it we should type the files name which we should ignore) |
| 21 | git branch | it displays the list of branches |
| 22 | git branch branchname | it will create branch |
| 23 | git branch -D branchname | it will delete the branch |
| 24 | git checkout branchname | it will switch to specific branch |
| 25 | git log --oneline --all | it will display all the commits including other branches |
| 26 | git merge branch2name branch1name | merging other branch commits to current branch bit it will create extra commit for merging |
| 27 | git rebase branch2name branch1name | merging other branch commits to current branch it will not create any extra commit |
| 28 | git show commitid | to show the changes we have done in that commit |
| 29 | git reset 58be407 --soft | it will delete till which commit u need only in branch (reset will delete the file in local not in central) |
| 30 | git reset 58be407 --mixed | it will delete till which commit u need in branch + repository |
| 31 | git reset 58be407 --hard | it will delete till which commit u need in branch +repository +working directory, we can’t get that file back |
| 32 | git revert 4f253b4 | it will ignore commits until which u need and create a revert commit (if the reverted commit is deleted by reset the ignored commit will come back) |
| 33 | git config --global --list | we can check push default is simple or current |
| 34 | git config --global push.default "simple" | to push the new branch, it should not be simple, if it is simple git push will not work |
| 35 | git config --global push.default "current" | this will change branch to current so git push will work |
| 36 | git tag | it displays all the tags |
| 37 | git tag v1.0 | creating a tag with version 1.0 |
| 38 | git show v1.0 | it will show the tags v1.0 commit information |
| 39 | git tag -a v1.1 -m "text" | to add tags version and annotation text parallelly |
| 40 | git tag -a v1.2 715c5e5 -m "text" | to add tags version and annotation text parallelly to existing commit by adding commit id |
| 41 | git branch nameofbranch v1.2 | it will create a specific named branch up to the tag v1.2 commit |
| 42 | git stash list |  |
| 43 | git stash pop |  |
| 44 | git stash |  |
| 45 | git stash apply stash@{0} |  |
| 46 | git stash drop stash@{0} |  |