



TOP GIT COMMANDS

Prepared By
Oranium Tech Team



For More Update
www.orianumtech.com
+91-80723 63772

1. `git init`: Initializes a new Git repository.
2. `git config --global user.name <user-name>`: Sets a name that will be associated with your Git commits.
3. `git config --global user.email <email-address>`: Sets an email address that will be associated with your Git commits.
4. `git clone <repo>`: Creates a copy of a remote repository on your local machine.
5. `git status`: Shows the current status of your working directory, including any changes or untracked files.
6. `git add <file>`: Adds a file to the staging area in preparation for a commit.
7. `git add .`: Adds all new and changed files to the staging area.
8. `git commit -m "<message>"`: Commits changes to the repository with a brief message describing the changes.
9. `git push`: Pushes committed changes to a remote repository.
10. `git pull`: Updates your local repository with the latest changes from the remote repository.
11. `git branch`: Lists all local branches in the current repository.
12. `git branch <branch-name>`: Creates a new branch.
13. `git checkout <branch-name>`: Switches to the specified branch and updates the working directory.
14. `git merge <branch-name>`: Merges the specified branch into the current branch.
15. `git remote -v`: Lists all remote repositories connected to your local repository.
16. `git remote add <name> <url>`: Adds a new remote repository to your local repository.
17. `git log`: Shows the commit history in reverse chronological order.
18. `git log --oneline`: Shows the commit history, one commit per line.
19. `git revert <commit-id>`: Reverts the changes made in the specified commit.
20. `git reset --hard <commit-id>`: Resets your working directory and index to the state of the specified commit.
 Be careful with this command: it discards all changes since then!
21. `git fetch`: Fetches updates from the remote repository without merging them.
22. `git stash`: Temporarily stashes changes in the working directory for later use.
23. `git stash pop`: Restores the most recently stashed changes.
24. `git diff`: Shows differences between the working directory and the index (staging area).
25. `git diff <commit1> <commit2>`: Shows differences between two commits.
26. `git rm <file>`: Removes a file from the working directory and the index.
27. `git mv <file-original> <file-renamed>`: Renames or moves a file both in the working directory and the index.
28. `git tag`: Lists all tags in the repository.
29. `git tag <tagname>`: Creates a new tag.
30. `git rebase <branch-name>`: Reapplies commits on top of another base tip.
31. `git cherry-pick <commit-id>`: Applies the changes introduced by the named commit on the current branch.