

Git-Commands List

By @xemishra

1. Configuration Commands

- `git config --global user.name "Your Name"`
 - `git config --global user.email "your@email.com"`
 - `git config --list`
 - `git config --global core.editor nano`
 - `git config --global init.defaultBranch main`
-

2. Repository Setup

- `git init`
- `git init --bare`
- `git clone <url>`
- `git clone --branch <branch> <url>`

3. File Tracking & Staging

- `git status`
 - `git add <file>`
 - `git add .`
 - `git add -A`
 - `git add -p`
-

4. Commit Commands

- `git commit -m "message"`
 - `git commit -a -m "message"`
 - `git commit --amend`
 - `git commit --amend -m "new message"`
-

5. Branching

- `git branch`
- `git branch <name>`

- `git branch -d <name>`
 - `git branch -D <name>`
 - `git branch -m <old> <new>`
 - `git branch -a`
 - `git branch -r`
-

6. Checkout / Switch

- `git checkout <branch>`
 - `git checkout -b <branch>`
 - `git switch <branch>`
 - `git switch -c <branch>`
 - `git checkout -- <file>`
-

7. Merging

- `git merge <branch>`
- `git merge --no-ff <branch>`

- `git merge --abort`
-

8. Rebase

- `git rebase <branch>`
 - `git rebase -i <branch>`
 - `git rebase --continue`
 - `git rebase --abort`
 - `git rebase --skip`
-

9. Remote Commands

- `git remote`
 - `git remote -v`
 - `git remote add origin <url>`
 - `git remote remove <name>`
 - `git remote rename <old> <new>`
-

10. Push Commands

- `git push`
 - `git push -u origin main`
 - `git push --force`
 - `git push --force-with-lease`
 - `git push origin --delete <branch>`
-

11. Pull & Fetch

- `git pull`
 - `git pull origin <branch>`
 - `git pull --rebase`
 - `git fetch`
 - `git fetch --all`
-

12. Stash Commands

- `git stash`

- `git stash save "message"`
 - `git stash list`
 - `git stash apply`
 - `git stash pop`
 - `git stash drop`
 - `git stash clear`
-

13. Log & History

- `git log`
 - `git log --oneline`
 - `git log --graph`
 - `git log --stat`
 - `git log -p`
-

14. Diff Commands

- `git diff`

- `git diff --staged`
 - `git diff <branch1> <branch2>`
 - `git diff <commit1> <commit2>`
-

15. Tagging

- `git tag`
 - `git tag <tagname>`
 - `git tag -a <tagname> -m "message"`
 - `git push origin <tagname>`
 - `git push --tags`
 - `git tag -d <tagname>`
-

16. Reset Commands

- `git reset <file>`
- `git reset --soft <commit>`
- `git reset --mixed <commit>`

- `git reset --hard <commit>`
-

17. Restore Commands

- `git restore <file>`
 - `git restore --staged <file>`
 - `git restore --source <commit> <file>`
-

18. Clean Workspace

- `git clean -n`
 - `git clean -f`
 - `git clean -fd`
 - `git clean -fx`
-

19. Cherry-Pick

- `git cherry-pick <commit>`

- `git cherry-pick --continue`
 - `git cherry-pick --abort`
-

20. Submodule Commands

- `git submodule add <url>`
 - `git submodule init`
 - `git submodule update`
 - `git submodule update --remote`
 - `git submodule deinit -f .`
-

21. Bisect (Debugging)

- `git bisect start`
 - `git bisect good <commit>`
 - `git bisect bad <commit>`
 - `git bisect reset`
-

22. Archive & Export

- `git archive --format zip --output file.zip HEAD`
 - `git archive --format tar HEAD`
-

23. Grep (Search)

- `git grep <text>`
 - `git grep -n <text>`
 - `git grep -p <pattern>`
-

24. Blame & Annotate

- `git blame <file>`
 - `git blame -L <start>,<end> <file>`
-

25. Reflog

- `git reflog`

- `git reflog expire`
 - `git reflog delete`
-

26. Worktree

- `git worktree add <path> <branch>`
 - `git worktree list`
 - `git worktree remove <path>`
-

27. Apply Patch

- `git apply <patch>`
 - `git apply --check <patch>`
-

28. Format Patch

- `git format-patch HEAD~1`
 - `git format-patch --cover-letter`
-

29. Sparse Checkout

- `git sparse-checkout init`
 - `git sparse-checkout set <folder>`
-

30. Maintenance Commands

- `git gc`
- `git fsck`
- `git prune`