

Git Command Documentation

1. Git Configuration

Command	Usage
git --version	Check installed Git version.
git config --global user.name "Your Name"	Set your username for commits.
git config --global user.email "you@example.com"	Set your email for commits.
git config --list	View all Git configuration settings.

2. Repository Setup

Command	Usage
git init	Initialize a new local Git repository.
git clone <repo-url>	Clone an existing remote repository.
git clone -b <branch> <repo-url>	Clone a specific branch.

3. Staging and Committing

Command	Usage
git status	Check status of files (staged, modified, untracked).
git add <file>	Stage a specific file for commit.
git add .	Stage all modified files.
git commit -m "message"	Commit changes with a message.
git commit -am "message"	Add and commit tracked files together.

4. Branching and Merging

Command	Usage
git branch	List all branches.
git branch <branch>	Create a new branch.
git checkout <branch>	Switch to another branch.
git checkout -b <branch>	Create and switch to a new branch.
git merge <branch>	Merge another branch into the current one.
git branch -d <branch>	Delete a branch.

5. Remote Operations

Command	Usage
---------	-------

git remote -v	View configured remotes.
git remote add origin <url>	Link local repository to a remote one.
git push	Push commits to the remote repository.
git push -u origin <branch>	Push a new branch and set upstream tracking.
git pull	Fetch and merge changes from remote repository.

6. Undo and Restore

Command	Usage
git restore <file>	Undo changes in the working directory.
git restore --staged <file>	Unstage a staged file.
git reset --soft <commit>	Undo commit but keep staged changes.
git reset --hard <commit>	Completely reset to a specific commit.
git revert <commit>	Create a new commit that undoes an earlier one.

7. Viewing History

Command	Usage
git log	View detailed commit history.
git log --oneline	Compact one-line commit view.
git show <commit>	View details for a specific commit.
git blame <file>	Show who changed each line of a file.

8. Stashing

Command	Usage
git stash	Temporarily save uncommitted changes.
git stash list	View list of stashed changes.
git stash apply	Reapply last stash.
git stash pop	Apply and remove last stash.

9. Tagging

Command	Usage
git tag	List all tags.
git tag <name>	Create a lightweight tag.
git tag -a <name> -m "msg"	Create annotated tag with message.
git push origin --tags	Push all tags to remote repository.