

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

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

6. Undo and Restore

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

7. Viewing History

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

8. Stashing

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

9. Tagging

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