

Git Commands Reference

Category	Command	Description
Git Configuration	<code>git config --global user.name \"Name\"</code>	Set your global username.
	<code>git config --global user.email \"email@example.com\"</code>	Set your global email.
	<code>git config --global color.ui auto</code>	Enable colored output in the terminal.
	<code>git help</code>	Display Git help documentation.
Repository Initialization	<code>git init</code>	Initialize a new Git repository.
	<code>git init <directory></code>	Initialize a Git repository in a specified directory.
	<code>git clone <repository_url></code>	Clone a repository from a remote to your local machine.
	<code>git clone --branch <branch_name> <repository_url></code>	Clone a specific branch from a repository.
Staging and Changes	<code>git add <file></code>	Stage a specific file.
	<code>git add .</code> Or <code>git add --all</code>	Stage all modified and new files.
	<code>git status</code>	Show the repository's status, including untracked and staged files.
	<code>git status --ignored</code>	Include ignored files in the status output.
	<code>git diff</code>	Show unstaged changes between working directory and staging area.
	<code>git diff --staged</code> Or <code>git diff --cached</code>	Show staged changes relative to the last commit.
	<code>git diff <commit1> <commit2></code>	Compare differences between two commits.
Committing Changes	<code>git commit -m \"<message>\"</code>	Commit staged changes with a message.
	<code>git commit -a -m \"<message>\"</code>	Commit all tracked file changes directly, bypassing staging.
	<code>git commit -m \"feat: message\"</code>	Commit indicating a new feature.
	<code>git commit -m \"fix: message\"</code>	Commit indicating a bug fix.
	<code>git commit -m \"docs: message\"</code>	Commit documentation changes.
Branching and Merging	<code>git branch</code>	List all local branches.
	<code>git branch <branch-name></code>	Create a new branch.
	<code>git branch -d <branch-name></code>	Delete a local branch.
	<code>git branch -a</code>	List all local and remote branches.
	<code>git checkout <branch-name></code>	Switch to a branch.
	<code>git checkout -b <branch-name></code>	Create and switch to a new branch.
	<code>git merge <branch></code>	Merge the specified branch into the current branch.

Category	<div>git stash Command</div>	<div>Save changes in a stash to work on another task without committing. Description</div>
	<code>git stash pop</code>	Apply and remove the most recent stash.
Remote Operations	<code>git fetch</code>	Fetch changes from the remote repository without merging.
	<code>git fetch --prune</code>	Remove stale references to remote-tracking branches.
	<code>git pull</code>	Fetch and merge changes from the remote repository.
	<code>git pull --rebase</code>	Fetch changes and rebase your branch on top of the updated branch.
	<code>git push</code>	Push local commits to the remote repository.
	<code>git push --all</code>	Push all branches to the remote repository.
	<code>git remote</code>	List all configured remote repositories.
	<code>git remote add <name> <url></code>	Add a new remote repository.
History and Comparison	<code>git log</code>	Show commit history for the current branch.
	<code>git log --all</code>	Show commit history across all branches.
	<code>git show <commit></code>	Show details of a specific commit, including changes.
	<code>git diff HEAD</code>	Show changes in the working directory relative to the last commit.
	<code>git revert <commit></code>	Create a new commit that reverses changes introduced by a specific commit.
Tagging	<code>git tag</code>	List all tags in the repository.
	<code>git tag <tag-name></code>	Create a lightweight tag at the current commit.
	<code>git tag -a <tag-name> -m "<message>"</code>	Create an annotated tag with a custom message.