

# Git-Commands

---

## Basic Git Commands

### Setting up Your Git Identity

Command	Description	Example(s)
<code>git config --global user.name</code>	Setup your name in Git	<code>git config --global user.name Your-Name</code>
<code>git config --global user.email</code>	Setup your mail in Git	<code>git config --global user.email Your-mail@example.com</code>

### Getting started with Git

Command	Description	Example(s)
<code>git init</code>	Initialize empty Git repository	<code>git init new-project</code>
<code>git clone</code>	Clone a repository from remote server	<code>git clone https://github.com/user/repo.git</code>

### Committing Changes

Command	Description	Example(s)
<code>git status</code>	Show the status of Working Directory	-
<code>git add</code>	Add files to staging Area	<code>git add file1.txt file2.txt</code>
<code>git add -A</code>	Add all new and changed files to staging area	-
<code>git commit</code>	Commit staged changes	<code>git commit -m "latest bug fix"</code>

### Branching

Command	Description	Example(s)
<code>git branch</code>	List branches	-
<code>git branch &lt;branch-name&gt;</code>	Create a new branch	<code>git branch feature-branch</code>
<code>git branch -d &lt;branch-name&gt;</code>	Delete a merged branch	<code>git branch -d feature-branch</code>

Command	Description	Example(s)
<code>git branch -D &lt;branch-name&gt;</code>	Delete a branch (Merged or not)	<code>git branch -D test-branch</code>
<code>git branch -m &lt;old-branch-name&gt; &lt;new-branch-name&gt;</code>	Rename a branch	<code>git branch -m feature-branch-v1.0 feature-branch-v2.0</code>
<code>git push origin --delete &lt;branch-name&gt;</code>	Delete a remote branch	<code>git push origin --delete feature-branch</code>
<code>git checkout &lt;branch-name&gt;</code>	switch to a branch	<code>git checkout feature-branch</code>
<code>git checkout -b &lt;branch-name&gt;</code>	create a new branch and switch to it	<code>git checkout -b feature-branch</code>
<code>git checkout -- &lt;file-name&gt;</code>	Discard changes to a file	<code>git checkout -- test.txt</code>
<code>git checkout .</code>	Discard all changes in your working directory and revert the files to the last committed state.	-

## Merging

Command	Description	Example(s)
<code>git merge &lt;branch-name&gt;</code>	Merge a branch	<code>git merge feature-branch</code>
<code>git merge &lt;source-branch&gt; &lt;target-branch&gt;</code>	Merge a branch into target branch	<code>git merge feature-V1 feature-V0</code>
<code>git merge --squash &lt;branch-name&gt;</code>	Merge a branch with all commits squashed into single commit	<code>git merge --squash feature-V0</code>

## Reviewing

Command	Description	Example(s)
<code>git status</code>	Displays the current status of your working directory and staged changes	-
<code>git log</code>	Displays a detailed commit history of the current branch	-
<code>git log --oneline</code>	Displays a simplified commit history of the current branch in a compact format	-
<code>git diff</code>	Shows the differences between your working directory and the most recent commit	-

Command	Description	Example(s)
<code>git diff &lt;commit1&gt; &lt;commit2&gt;</code>	Compares two commits and displays the differences between them	<code>git diff abc123 xyz987</code>

## Stashing

Command	Description	Example(s)
<code>git stash</code>	store modified and staged changes	-
<code>git stash save "comment"</code>	stash the modified changes with description	<code>git stash save "Work in Progress - Feature 1"</code>
<code>git stash list</code>	List all stashes	-
<code>git stash apply</code>	Applies the most recent stash entry to your working directory, without deleting the stash entry	-
<code>git stash pop</code>	Applies the most recent stash entry to your working directory and deletes it	-
<code>git stash show</code>	Show the difference summary of recent stash entry	-
<code>git stash drop</code>	Delete the recent stash entry without applying it	-
<code>git stash clear</code>	Delete all stash entries	-

## Git commands related to remote repositories

Command	Description	Example(s)
<code>git push</code>	Push changes to remote repository	-
<code>git pull</code>	Fetch the latest update from remote repository and merge with local one	-
<code>git push origin &lt;branch-name&gt;</code>	Push a branch to the remote repository	<code>git push origin feature-branch</code>
<code>git push origin -- delete &lt;branch-name&gt;</code>	Delete a remote branch	<code>git push origin -- delete feature-branch</code>
<code>git fetch origin</code>	Fetch the changes from remote repository without merging	-

Command	Description	Example(s)
<code>git fetch origin &lt;branch-name&gt;</code>	fetch changes from a remote branch without merging	<code>git fetch origin main</code>
<code>git remote</code>	view all remote connections	-
<code>git remote remove &lt;connection-name&gt;</code>	Remove a connection	-
<code>git remote rename &lt;old-name&gt; &lt;new-name&gt;</code>	Rename a connection	-
<code>git pull --rebase origin</code>	Rebase local changes on top of new changes from remote repository	-

## Reverting and Resetting Commands

Command	Description	Example(s)
<code>git revert &lt;commit&gt;</code>	Creates a new commit that undoes the changes introduced by the specified commit	<code>git revert abc123</code>
<code>git revert &lt;commit1&gt; &lt;commit2&gt; ... &lt;commit n&gt;</code>	Reverts a range of commits, creating new commits that undo the changes introduced by each of them.	<code>git revert abc123 xyz987</code>
<code>git reset --soft &lt;commit&gt;</code>	Moves the branch pointer to the specified commit, preserving the changes in your working directory and staging area.	<code>git reset --soft abc123</code>
<code>git reset --hard &lt;commit&gt;</code>	Moves the branch pointer to the specified commit and discards all changes, both in the working directory and staging area (Deletes all commits ahead of it).	<code>git reset --hard abc123</code>
<code>git revert &lt;commit&gt;</code>	Creates a new commit that undoes the changes introduced by the specified commit	<code>git revert abc123</code>
<code>git rm &lt;file&gt;</code>	Remove file from Working Directory and staging area	<code>git rm test.txt</code>
<code>git rm --cached &lt;file&gt;</code>	Remove file from staging area only	<code>git rm -- cached test.txt</code>

## Common Configuration

Command	Description	Example(s)
---------	-------------	------------

Command	Description	Example(s)
<code>git config --global alias.&lt;alias-name&gt; &lt;git command&gt;</code>	Configure Git aliases	<code>git config --global alias.co checkout</code>
<code>git config --global core.editor</code>	Configure default text editor	<code>git config --global core.editor "code"</code>
<code>git config --global --edit</code>	Edit Git configuration in an editor	-

## Miscellaneous

Command	Description	Example(s)
<code>git log -S "string"</code>	Search git commits with particular string	<code>git log -S "feature"</code>
<code>git clean -f</code>	clean all untracked files	-
<code>git commit --amend</code>	Amends the most recent commit with the new changes	-
<code>git show &lt;commit&gt; -stat</code>	display a histogram showing inserts,deletion and modifications per file for a specific commtit along with general commit information	-
<code>git bisect</code>	A powerful Git command used to find the commit that introduced a bug or regression in your codebase. It performs a binary search through the commit history to identify the specific commit where the issue was introduced.	<code>git bisect start , git bisect good , git bisect bad</code>
<code>git commit --amend</code>	Amends the most recent commit with the new changes	-
<code>git cherry-pick &lt;commit from another branch&gt;</code>	Merges a particular commit from another branch	<code>git cherry-pick xyz123</code>
<code>git tag</code>	List all tags in the repository	-
<code>git tag &lt;tag-name&gt;</code>	Create a new tag	<code>git tag v1.0.0</code>
<code>git tag -a &lt;tag-name&gt; -m "message"</code>	Create an annotated tag with a message	<code>git tag -a v1.0.0 -m "Release version 1.0.0"</code>
<code>git show &lt;tag-name&gt;</code>	Show details of a specific tag	<code>git show v1.0.0</code>
<code>git push --tags</code>	Push tags to a remote repository	<code>git push --tags</code>

Command	Description	Example(s)
<code>git blame &lt;file&gt;</code>	Show who last modified each line of a file	<code>git blame myfile.txt</code>
<code>git reflog</code>	Show a log of all Git references	-
<code>git gc</code>	Perform Git garbage collection	-
<code>git archive -- format=zip -- output=filename.zip &lt;branch-name&gt;</code>	Create a zip archive of a specific branch	<code>git archive -- format=zip -- output=mybranch.zip main</code>
<code>git grep &lt;text&gt;</code>	Search for text in your Git repository	<code>git grep "search text"</code>