

# Git Cheatsheet

By Dejan Panovski • Updated on Feb 9, 2026 • [Download PDF](#)

## Quick reference for common Git commands and workflows

Git is a distributed version control system used to track changes in source code. This cheatsheet covers everyday commands for working with repositories, branches, commits, and remote collaboration.

### Getting Started

Initialize and configure Git.

<code>git init</code>	Initialize a new repository
<code>git clone url</code>	Clone a remote repository
<code>git status</code>	Show working tree status
<a href="#">git config --global user.name "Name"</a>	Set global name
<code>git config --global user.email "email"</code>	Set global email

### Staging and Committing

Add changes and create commits.

<code>git add file</code>	Stage a file
<code>git add .</code>	Stage all changes
<code>git commit -m "message"</code>	Commit staged changes
<a href="#">git commit --amend</a>	Amend last commit
<a href="#">git revert commit</a>	Revert a commit

## Branches

Create, list, and switch branches.

<a href="#">git branch</a>	List branches
git branch name	Create a branch
git switch name	Switch branch
git switch -c name	Create and switch
<a href="#">git branch -d name</a>	Delete local branch
git branch -D name	Force delete branch
<a href="#">git branch -m new-name</a>	Rename current branch

## Merge and Rebase

Combine branches and rewrite history.

git merge branch	Merge branch into current
git merge --no-ff branch	Merge with merge commit
git merge --abort	Abort merge in progress
git rebase branch	Rebase onto branch
git rebase -i HEAD~3	Interactive rebase last 3
git rebase --abort	Abort rebase in progress
git cherry-pick commit	Apply a single commit

## Remote Repositories

Work with remotes and synchronization.

git remote -v	List remotes
<a href="#">git remote add origin url</a>	Add remote
<a href="#">git remote set-url origin url</a>	Change remote URL
<a href="#">git remote remove origin</a>	Remove remote
git fetch	Fetch from remote
git pull	Fetch and merge
git push	Push to remote
git push -u origin branch	Push and set upstream
git push origin --delete branch	Delete remote branch

## Stash

Temporarily save uncommitted changes.

<code>git stash</code>	Stash changes
<code>git stash push -m "message"</code>	Stash with message
<code>git stash list</code>	List stashes
<code>git stash pop</code>	Apply and remove latest
<code>git stash apply stash@{0}</code>	Apply without removing
<code>git stash drop stash@{0}</code>	Delete a stash
<code>git stash clear</code>	Delete all stashes

## History and Diff

Inspect history and changes.

<code>git log</code>	View commit history
<code>git log --oneline --graph</code>	Compact graph view
<code>git log -p file</code>	History of a file
<code>git show commit</code>	Show a commit
<code>git blame file</code>	Show who changed each line
<code>git reflog</code>	Show reference log (recovery)
<code>git diff</code>	Unstaged diff
<code>git diff --staged</code>	Staged diff
<code>git diff branch1..branch2</code>	Compare branches

## Undo and Cleanup

Discard or reset changes safely.

<code>git restore file</code>	Discard local changes
<code>git restore --staged file</code>	Unstage a file
<a href="#"><code>git reset --soft HEAD~1</code></a>	Undo commit, keep changes
<code>git reset --hard HEAD</code>	Reset to last commit
<a href="#"><code>git clean -fd</code></a>	Remove untracked files/dirs

## Tags

Create and manage tags.

<code>git tag</code>	List tags
<code>git tag v1.0.0</code>	Create tag
<code>git tag -a v1.0.0 -m "msg"</code>	Annotated tag
<code>git push --tags</code>	Push all tags
<code>git tag -d v1.0.0</code>	Delete local tag
<code>git push origin --delete v1.0.0</code>	Delete remote tag

## .gitignore

Exclude files from version control.

<code>*.log</code>	Ignore all .log files
<code>node_modules/</code>	Ignore directory
<code>!important.log</code>	Negate ignore rule
<code>**/build</code>	Ignore in any directory
<code>git rm --cached file</code>	Untrack a tracked file