

# Neovim Keymaps Quick Reference

Leader Key: , (comma) | Updated: 2025-11-08 | Config: ~/.config/nvim

---

## Essential Commands

Category	Keymap	Action	Category	Keymap	Action
<b>Editor</b>	,l	Save file	<b>LSP Nav</b>	gd	Go to definition
	,f	Format code		gD	Go to declaration
<b>File Tree</b>	,n	Toggle Neo-tree	<b>LSP Edit</b>	gi	Go to implementation
	,F	Reveal file		gr	Find references
	,e	Focus tree		K	Hover docs
<b>Telescope</b>	-	Oil (parent dir)	<b>Diagnostics</b>	<C-k>	Signature help
	,t	Find files		1gD	Type definition
	,g	Live grep		,rn	Rename symbol
<b>Git</b>	,o	Search buffer	<b>AI</b>	,ac	Code actions
	;	Buffers		[d	Previous diagnostic
	,j	Find (word)		]d	Next diagnostic
	,k	Grep (word)	<b>Motion</b>	,dd	Diagnostic list
	,vg	LazyGit UI		<C-,>	Claude Code
	,vd	Diff view		Tab	Copilot (insert)
	,vh	File history	<b>Comment</b>	gc	Toggle comment
	,gy	Copy permalink		gcc	Toggle comment line
	,hs	Stage hunk		s	Flash jump
	,hp	Preview hunk		S	Flash treesitter
	]c / [c	Next/Prev hunk			

---

## Harpoon - Quick File Navigation

Mark frequently accessed files for instant navigation

	Primary	Navigation		
	,a	Add current file	<C-h>	Jump to file 1
	,hm	Toggle menu	<C-j>	Jump to file 2
			<C-k>	Jump to file 3
<b>Alternative</b>			<C-l>	Jump to file 4
	,1	Jump to file 1	,3	Jump to file 3
	,2	Jump to file 2	,4	Jump to file 4

---

## Debugging (nvim-dap)

Breakpoints		Execution		UI	
,db	Toggle breakpoint	,dc	Continue	,du	Toggle UI
		,ds	Step over	,dt	Terminate
		,di	Step into	,dr	Open REPL
		,do	Step out		

**Debug UI:** Auto-opens on debug start · Variables, stack, breakpoints · Console output

---

## Testing (neotest)

Run Tests		UI		Watch	
,tt	Run nearest test	,ts		Toggle summary	,tw Toggle watch
,tf	Run file tests	,to		Toggle output	,td Debug test
		,tS		Stop tests	

**Test Summary:** Tree view · Inline status · Click to run · Press **a** to run all · **o** for output

---

## Neo-tree Commands (when focused)

Navigation		File Ops		Window		Search/Git	
<CR>	Open a file	Add s		V-split	/		Fuzzy find
<Space>	Toggle A node	Add S		H-split	H		Toggle hidden
<BS>	Parent d dir	Delete t		New tab	[g		Prev git change
.	Set r root	RenameP			Preview]g		Next git change
R	Refresh y	Copy q		Close </>			Switch source
?	Help p	Paste z/Z		Close/Expand all			

## Oil.nvim - Edit Directories as Buffers

Use Vim motions to manage files - edit directory like a buffer

Key	Action	Key	Action
-	Open parent directory	<CR>	Select file/directory

Key	Action	Key	Action
g.	Toggle hidden files	<C-s>	Open in vsplit
g?	Show help	<C-h>	Open in hsplit
g\\	Toggle trash	<C-c>	Close

**Edit Mode:** Use normal Vim commands (dd, cw, o, etc.) · Save with :w to apply changes

---

## Git Workflow (LazyGit + Gitsigns + Diffview)

Primary (,v)	Hunks (,h)	Navigation		
,vg	LazyGit ,hs UI	Stage	]c	Next hunk
,vs	Git ,hr status	Reset	[c	Prev hunk
,vf	LazyGit ,hs (file)	Stage	<b>Conflicts</b>	
,vd	Diff ,hR view	buffer		Choose ours
,vc	Close ,hu diff	Reset	co	Choose theirs
,vh	File ,hp history	Undo	ct	Choose both
,vr	Refresh ,hb	Preview	cb	
,vb	Toggle ,gy blame	Blame	c0	Choose none
		line		
		Copy	]x / [x	Next/Prev conflict
		perma-link		

**Diff View:** <Tab>/<S-Tab> cycle files · - stage/unstage · gf goto file · [x/]x conflicts · ,co/,ct choose ours/theirs

**Inside LazyGit:** Space stage/unstage · c commit · P push · p pull · ? help · q quit

---

## Flash Motion - Enhanced Navigation

Jump to any visible location with labels

Key	Mode	Action
s	Normal/Visual/Operator	Flash jump (label-based)
S	Normal/Visual/Operator	Flash treesitter (syntax-aware)
r	Operator-pending	Remote flash
R	Visual/Operator	Treesitter search
<C-s>	Command mode	Toggle flash search

---

## Surround Operations (mini.surround)

Add, delete, replace surrounding characters (quotes, brackets, tags)

Key	Action	Example
sa	Add surrounding	saiw" → add quotes around word
sd	Delete surrounding	sd" → delete surrounding quotes
sr	Replace surrounding	sr'" → replace " with '
sf	Find surrounding (right)	sf" → find next "
sF	Find surrounding (left)	sF" → find previous "
sh	Highlight surrounding	sh" → highlight quotes

---

## Tree-sitter Text Objects

Usage: <operator><text-object> (e.g., **daf** = delete entire function, **vif** = select function body)

Around	Inside	Description	Around	Inside	Description
af	if	Function	ac	ic	Class
al	il	Loop	ai	ii	Conditional
aa	ia	Parameter/Argument			

**Navigation:** ]f/[f (next/prev function start) · ]F/[F (function end) · ]c/[c (class start) · ]C/[C (class end)

**Incremental Selection:** gnn (init) → grn (expand node) → grc (expand scope) → grm (shrink)

**Context:** Shows current function/class name at top of screen (automatic)

**Auto-tag:** Auto-closes and renames HTML/JSX tags (automatic)

---

## Completion (nvim-cmp) - Insert Mode

Key	Action	Key	Action	Key	Action
<C-b>	Scroll docs up	<C-f>	Scroll docs down	<C-o>	Manual trigger
<C-e>	Abort	<CR>	Confirm	<Tab> / <S-Tab>	Next/Prev item

**Sources:** LSP · Buffer · Path · Snippets · **Auto-pairs:** (), [], {}, "", ' ', "

---

## Telescope Picker Controls

Key	Action	Key	Action
<C-n> / <C-p>	History next/prev	<C-d>	Delete buffer

---

## Essential Vim Motions

Movement	Editing		Visual/Search		
h/j/k/l	←↓↑→	i/a	Insert be- fore/after	v/V	Visual/Line
w/b/e	Word fwd/back/end	I/A	Line start/end	<C-v>	Visual block
0/^/\$	Line start/first/end	o/O	New line	/ / ?	Search fwd/back
gg/G	File start/end	dd/yy/p	Delete/Yank/Paste		Next/Prev match
{/}	Para prev/next	u/<C-r>	line	Undo/Redo/#	Search word
%	Match bracket	.	Repeat change	>> / <<	fwd/back
<C-d>/<C-u>	Scroll half page	<C-o>/<C-i>	Jump back/fwd		Indent/Unindent

## Window & Tab Management

Windows		Tabs	
<C-w>s	Split horizontal	:tabnew	New tab
<C-w>v	Split vertical	gt/gT	Next/Prev tab
<C-w>c	Close window	<num>gt	Go to tab #
<C-w>h/j/k/l	Navigate windows	:tabclose	Close tab
<C-w>=	Equalize sizes		

## Marks

Key	Action	Key	Action
ma	Set mark 'a'	`a	Jump to mark 'a'
'a	Jump to line of mark 'a'	:marks	List marks

## Quick Command Reference

WORKFLOW	NAVIGATION	CODE EDITING
,l → ,f	Save & format	gd Goto def
,t	Find files	K Docs
,g	Search code	gr References
,n	Toggle tree	[d / ]d Diagnostics
,vg	LazyGit UI	]c / [c Next/Prev hunk
,a → <C-h>	Mark + Jump	s Flash jump
		,rn Rename
		,ac Actions
		,f Format
		gc Comment
		,hs Stage hunk
		sa/sd/sr Surround
TEXT OBJECTS	SELECTION	DEBUG & TEST
vaf	Select function gnn → grn	Smart select
		,db Breakpoint

dif	Delete fn body	grm	Shrink	,dc	Continue
vac	Select class	grc	Scope	,ds	Step over
-	Oil (edit dir)	saw"	Add quotes	,tt	Run test

---

## New Features Summary

### Productivity Enhancements

- **Harpoon:** Mark 4 frequently accessed files for instant navigation
- **Flash:** Label-based motion for jumping to any visible location
- **Oil:** Edit directories as buffers using normal Vim motions
- **Which-key:** Press `,` and wait to see all available commands

### Development Tools

- **Debugging:** Full DAP support for TypeScript, JavaScript, React Native, PHP
- **Testing:** Integrated test runner with inline results (Jest, PHPUnit)
- **Linting:** Automatic linting with eslint (JS/TS) and phpcs (PHP)

### Editor Enhancements

- **Mini.surround:** Quick surrounding operations (quotes, brackets, tags)
- **Mini.comment:** Smart commenting with `gc`
- **Mini.pairs:** Auto-close brackets and quotes
- **Treesitter Context:** Shows current function/class at top
- **Auto-tag:** Auto-closes HTML/JSX tags

### LSP & Completion

- **JSON/YAML Schemas:** 400+ schemas with automatic validation
- **Intelephense:** Superior PHP language server for Symfony
- **Prettierd:** 10-20x faster formatting with daemon
- **Modern LSP:** Single LspAttach pattern for all servers

### Git Enhancements

- **git-conflict:** Simple conflict resolution with `co`, `ct`, `cb`
- **gitlinker:** Generate GitHub/GitLab permalinks with `,gy`

---

**Config Files:** `lua/config/keymaps.lua` · `lua/config/lsp/keymaps.lua` · `lua/plugins/*.lua`

**Plugin Count:** ~55 plugins | **Startup Time:** <100ms | **Architecture:** Modular with lazy loading