Neovim Commands Cheat Sheet - Complete Edition

Understanding Neovim Modes

Neovim operates in different modes, each serving specific purposes:

- **Normal Mode**: Default mode for navigation and text manipulation
- **Insert Mode**: For typing and inserting text
- **Visual Mode**: For selecting text
- **Command Mode**: For executing commands and operations
- **Replace Mode**: For overwriting existing text

Mode Switching Commands

Command	Mode Change	Why Use	When to Use
i	Normal → Insert (before cursor)	Start typing at current position	Beginning to write or edit text
I	Normal → Insert (beginning of line)	Start typing at line start	Adding content at line beginning
a	Normal → Insert (after cursor)	Start typing after current character	Continuing text after cursor
A	Normal → Insert (end of line)	Start typing at line end	Adding content at line end
o	Normal → Insert (new line below)	Create and edit new line below	Adding new content below current line
O	Normal → Insert (new line above)	Create and edit new line above	Adding new content above current line
v	Normal → Visual (character)	Select individual characters	Selecting specific text portions
V	Normal → Visual (line)	Select entire lines	Selecting complete lines for operations
Ctrl+v	Normal → Visual (block)	Select rectangular blocks	Editing columns or blocks of text
R	Normal → Replace	Overwrite existing text	Replacing text without inserting
Esc	Any → Normal	Return to navigation mode	Finishing edits, canceling operations
:	Normal → Command	Execute commands	Running operations, saving, searching

---

Navigation Commands (Normal Mode)

Basic Movement

Command	Action	Why Use	When to Use
h	Move left	Single character navigation	Fine positioning
j	Move down	Single line navigation	Moving through text line by line
k	Move up	Single line navigation	Moving through text line by line
l	Move right	Single character navigation	Fine positioning
w	Next word start	Jump between words efficiently	Navigating through text quickly
W	Next WORD start (space-separated)	Jump over punctuation	Moving through code or complex text
e	Next word end	Position at word endings	Precise word-end positioning
E	Next WORD end (space-separated)	Position at WORD endings	Working with punctuated text
b	Previous word start	Navigate backwards by words	Backtracking through text efficiently
B	Previous WORD start	Navigate backwards by WORDs	Backtracking over punctuation
0	Beginning of line	Jump to line start	Quick line beginning access
^	First non-blank character	Jump to first text character	Skipping indentation
$	End of line	Jump to line end	Quick line end access
gg	First line of file	Jump to document start	Navigating to file beginning
G	Last line of file	Jump to document end	Navigating to file end
{number}G	Go to line number	Jump to specific line	Direct line access
Ctrl+f	Page down	Scroll full screen down	Fast document navigation
Ctrl+b	Page up	Scroll full screen up	Fast document navigation
Ctrl+d	Half page down	Scroll half screen down	Moderate scrolling speed
Ctrl+u	Half page up	Scroll half screen up	Moderate scrolling speed

Advanced Navigation

Command	Action	Why Use	When to Use
f{char}	Find character forward	Jump to specific character	Quick character location
F{char}	Find character backward	Jump to specific character backwards	Reverse character search
t{char}	Till character forward	Stop before character	Precise positioning before target
T{char}	Till character backward	Stop after character backwards	Precise reverse positioning
;	Repeat last f/F/t/T	Repeat character search	Finding multiple occurrences
,	Reverse last f/F/t/T	Reverse character search direction	Changing search direction
%	Match bracket/parenthesis	Navigate between matching pairs	Code structure navigation
*	Search word under cursor forward	Find current word occurrences	Quick word searching
#	Search word under cursor backward	Find current word backwards	Reverse word searching
Ctrl+o	Jump to previous location	Navigate jump history backwards	Returning to previous positions
Ctrl+i	Jump to next location	Navigate jump history forwards	Moving through jump history

---

Text Editing Commands (Normal Mode)

Deletion

Command	Action	Why Use	When to Use
x	Delete character under cursor	Remove single character	Quick character deletion
X	Delete character before cursor	Remove character to the left	Backspace-like deletion
dw	Delete word	Remove entire word	Word-level deletion
dW	Delete WORD	Remove space-separated word	Deleting complex words
dd	Delete line	Remove entire line	Line-level deletion
D	Delete to end of line	Remove from cursor to line end	Partial line deletion
d0	Delete to beginning of line	Remove from line start to cursor	Beginning line deletion
d{motion}	Delete with motion	Combine deletion with movement	Flexible deletion operations
{number}dd	Delete multiple lines	Remove specified number of lines	Bulk line deletion

Copying (Yanking)

Command	Action	Why Use	When to Use
yw	Yank word	Copy word to clipboard	Word-level copying
yy	Yank line	Copy entire line	Line-level copying
Y	Yank to end of line	Copy from cursor to line end	Partial line copying
y{motion}	Yank with motion	Copy with movement command	Flexible copying operations
{number}yy	Yank multiple lines	Copy specified number of lines	Bulk line copying

Pasting

Command	Action	Why Use	When to Use
p	Paste after cursor/below line	Insert copied text after position	Standard pasting operation
P	Paste before cursor/above line	Insert copied text before position	Reverse pasting operation
{number}p	Paste multiple times	Repeat paste operation	Multiple insertions

Change Operations

Command	Action	Why Use	When to Use
`cw`	Change word	Replace word with new text	Word replacement
`cW`	Change WORD	Replace WORD with new text	Complex word replacement
`cc`	Change line	Replace entire line	Line replacement
`C`	Change to end of line	Replace from cursor to line end	Partial line replacement
`c{motion}`	Change with motion	Replace text with movement	Flexible replacement operations
`r{char}`	Replace character	Replace single character	Quick character replacement
`R`	Enter replace mode	Overwrite existing text	Extended character replacement
`s`	Substitute character	Delete character and enter insert	Character substitution
`S`	Substitute line	Delete line and enter insert	Line substitution

---

Search and Replace Commands

Searching (Normal Mode)

Command	Action	Why Use	When to Use
`/{pattern}`	Search forward	Find text patterns ahead	Forward text location
`?{pattern}`	Search backward	Find text patterns behind	Backward text location
`n`	Next search result	Continue search in same direction	Finding multiple matches
`N`	Previous search result	Continue search in opposite direction	Reverse search direction
`/\c{pattern}`	Case-insensitive search	Ignore case in search	Flexible text matching
`/\C{pattern}`	Case-sensitive search	Enforce case matching	Precise text matching

Replace (Command Mode)

Command	Action	Why Use	When to Use
`:s/old/new/`	Replace first occurrence in line	Single replacement per line	Targeted line replacement
`:s/old/new/g`	Replace all occurrences in line	Multiple replacements per line	Complete line replacement
`:%s/old/new/g`	Replace all in file	Global file replacement	Document-wide changes
`:%s/old/new/gc`	Replace all with confirmation	Controlled global replacement	Verified replacements
`:s/old/new/i`	Case-insensitive replace	Ignore case in replacement	Flexible replacement matching

---

Visual Mode Commands

Selection

Command	Action	Why Use	When to Use
v	Character visual mode	Select individual characters	Precise text selection
V	Line visual mode	Select entire lines	Line-based operations
Ctrl+v	Block visual mode	Select rectangular blocks	Column editing
gv	Reselect last visual selection	Restore previous selection	Repeating visual operations
o	Move to other end of selection	Change selection direction	Adjusting selection boundaries

Visual Operations

Command	Action	Why Use	When to Use
d	Delete selection	Remove selected text	Visual deletion
y	Yank selection	Copy selected text	Visual copying
c	Change selection	Replace selected text	Visual replacement
>	Indent selection	Increase indentation	Code formatting
<	Unindent selection	Decrease indentation	Code formatting
=	Auto-indent selection	Format code automatically	Code beautification
u	Lowercase selection	Convert to lowercase	Text case conversion
U	Uppercase selection	Convert to uppercase	Text case conversion
~	Toggle case of selection	Switch character cases	Case manipulation

---

File Operations (Command Mode)

Basic File Operations

Command	Action	Why Use	When to Use
:w	Save file	Preserve changes	Regular saving
:w {filename}	Save as filename	Save with new name	Creating file copies
:q	Quit	Exit Neovim	Normal exit
:q!	Quit without saving	Force exit	Discarding changes
:wq	Save and quit	Save then exit	Standard completion
:x	Save and quit (if changed)	Conditional save and exit	Efficient completion
ZZ	Save and quit	Quick save and exit	Fast completion
ZQ	Quit without saving	Quick force exit	Fast abandonment

File Navigation

Command	Action	Why Use	When to Use
:e {filename}	Edit file	Open new file	File switching
:e!	Reload current file	Refresh file contents	Undoing all changes
:enew	Create new buffer	Start new document	New file creation
:pwd	Show current directory	Display working directory	Directory awareness
:cd {path}	Change directory	Set working directory	Directory navigation

---

Window and Buffer Management

Window Operations

Command	Action	Why Use	When to Use
:split or :sp	Horizontal split	Divide window horizontally	Multi-file viewing
:vsplit or :vs	Vertical split	Divide window vertically	Side-by-side editing
Ctrl+w h	Move to left window	Navigate between windows	Window switching
Ctrl+w j	Move to bottom window	Navigate between windows	Window switching
Ctrl+w k	Move to top window	Navigate between windows	Window switching
Ctrl+w l	Move to right window	Navigate between windows	Window switching
Ctrl+w w	Cycle through windows	Move to next window	Window cycling
Ctrl+w q	Close current window	Remove window	Window management
Ctrl+w o	Close other windows	Keep only current window	Focus single window
Ctrl+w =	Equal window sizes	Balance window dimensions	Window layout
Ctrl+w +	Increase window height	Expand current window	Window sizing
Ctrl+w -	Decrease window height	Shrink current window	Window sizing

Buffer Management

Command	Action	Why Use	When to Use
:ls or :buffers	List buffers	Show open files	Buffer awareness
:b {number}	Switch to buffer number	Navigate to specific buffer	Direct buffer access
:b {name}	Switch to buffer by name	Navigate by filename	Name-based navigation
:bn or :bnext	Next buffer	Move to next open file	Buffer cycling
:bp or :bprev	Previous buffer	Move to previous open file	Buffer cycling
:bd	Delete buffer	Close file from memory	Buffer cleanup
:bd!	Force delete buffer	Force close unsaved buffer	Forced buffer cleanup

---

Advanced Commands

Marks and Jumps

Command	Action	Why Use	When to Use
m{letter}	Set mark	Save position with letter	Creating position bookmarks
'{letter}	Jump to mark line	Go to marked line	Returning to saved positions
\`{letter}`	Jump to mark position	Go to exact marked position	Precise position return
''	Jump to previous line	Return to last line position	Quick line return
\``	Jump to previous position	Return to exact last position	Precise position return
:marks	List marks	Show all saved marks	Mark management

Macros

Command	Action	Why Use	When to Use
q{letter}	Start recording macro	Begin command recording	Automating repetitive tasks
q	Stop recording macro	End command recording	Finishing macro creation
@{letter}	Execute macro	Run recorded commands	Executing automated tasks
@@	Repeat last macro	Re-run previous macro	Quick macro repetition
{number}@{letter}	Execute macro multiple times	Run macro repeatedly	Bulk automated operations

Folding

Command	Action	Why Use	When to Use
zf{motion}	Create fold	Hide code sections	Code organization
zo	Open fold	Expand folded section	Viewing hidden code
zc	Close fold	Collapse code section	Hiding code details
za	Toggle fold	Switch fold state	Quick fold management
zR	Open all folds	Expand all sections	Full code visibility
zM	Close all folds	Collapse all sections	Overview mode
zd	Delete fold	Remove fold definition	Fold cleanup

Registers

Command	Action	Why Use	When to Use
"{letter}y	Yank to register	Save to specific clipboard	Multiple clipboard management
"{letter}p	Paste from register	Retrieve from specific clipboard	Accessing saved content
":p	Paste last command	Repeat previous command	Command repetition
"/p	Paste last search	Insert search pattern	Search pattern reuse
:registers	Show registers	Display clipboard contents	Register management

Text Objects

Command	Action	Why Use	When to Use
`ciw`	Change inner word	Replace word content	Word editing
`caw`	Change a word	Replace word including spaces	Complete word replacement
`ci"`	Change inside quotes	Replace quoted content	String editing
`ca"`	Change around quotes	Replace including quotes	Complete string replacement
`ci(`	Change inside parentheses	Replace parenthetical content	Function parameter editing
`ca(`	Change around parentheses	Replace including parentheses	Complete expression replacement
`cit`	Change inside tag	Replace HTML/XML tag content	Markup editing
`cat`	Change around tag	Replace including tags	Complete element replacement

---

Configuration and Settings

Common Settings (Command Mode)

Command	Action	Why Use	When to Use
`:set number`	Show line numbers	Display line references	Code navigation
`:set nonumber`	Hide line numbers	Clean display	Distraction-free editing
`:set relativenumber`	Show relative line numbers	Efficient motion commands	Advanced navigation
`:set hlsearch`	Highlight search results	Visual search feedback	Search visibility
`:set nohlsearch`	Remove search highlighting	Clean display after search	Removing search clutter
`:set ignorecase`	Ignore case in search	Flexible searching	Case-insensitive search
`:set smartcase`	Smart case searching	Context-aware case matching	Intelligent searching
`:set autoindent`	Auto-indent new lines	Maintain indentation	Code formatting
`:set expandtab`	Use spaces for tabs	Consistent spacing	Code standardization
`:set tabstop=4`	Set tab width	Control indentation size	Formatting preference

Help and Information

Command	Action	Why Use	When to Use
`:help`	Open help system	Access documentation	Learning and reference
`:help {topic}`	Help on specific topic	Targeted information	Specific feature help
`:version`	Show Neovim version	Display version information	Compatibility checking
`:checkhealth`	Run health check	Diagnose configuration issues	Troubleshooting

---

Practical Usage Tips

When to Use Different Modes

- **Normal Mode**: Default state for navigation, quick edits, and command execution
- **Insert Mode**: When actively typing or adding new content
- **Visual Mode**: When selecting text for operations like deletion, copying, or formatting
- **Command Mode**: For file operations, search/replace, and configuration changes

Efficiency Tips

- Use word motions (w, b, e) instead of character movement for faster navigation
- Combine commands with numbers for bulk operations (5dd deletes 5 lines)
- Use text objects for precise editing (ciw to change a word)
- Master visual mode for complex selections and operations
- Use marks for quickly jumping between frequently accessed locations
- Record macros for repetitive editing tasks

Common Workflows

1. **Code Editing**: Normal mode for navigation → Insert mode for writing → Normal mode for commands
2. **Text Selection**: Normal mode → Visual mode for selection → Operation → Normal mode
3. **Search and Replace**: Normal mode → / for search → n/N for navigation → :s for replace
4. **File Management**: Command mode for opening, saving, and switching between files

This cheat sheet covers the essential Neovim commands with explanations of why and when to use each one. Practice these commands regularly to build muscle memory and increase your editing efficiency.