Vim Commands

Complete command reference — Learning Vim course

N Normal mode I Insert mode C Command mode V Visual mode S Shell / terminal RC .vimrc setting
ANY Any mode

Learning Vim

Appendix B — Vim Commands

A complete reference of every command introduced across all eight chapters of this course, grouped by topic. Use this alongside the lessons as a quick lookup or print it as a desk reference.

N Normal mode | I Insert mode | C Command mode | V Visual mode | S Shell / terminal | RC .vimrc setting | ANY Any mode

Opening & Quitting

OPENING FILES

`vim <file>`	Open or create a file	S	`vim +N <file>`	Open file at line N	S
`vim +/pattern <file>`	Open and jump to first match	S	`vim +"cmd" <file>`	Run command after opening	S
`vim -R <file>`	Open in read-only mode	S	`vim -u NONE <file>`	Open without loading vimrc	S
`vim --version`	Show Vim version	S	`:e <file>`	Open/edit a file (Tab completes)	C
`:e!`	Reload file from disk, discard changes	C	`:e .`	Open directory browser (netrw)	C

SAVING

`:w`	Write (save) the file	C	`:w <file>`	Save to a new filename (Save As)	C
`:saveas <file>`	Save to new filename and switch to it	C			

QUITTING

`:q`	Quit (fails if unsaved changes)	C	`:q!`	Force quit, discard changes	C
`:wq`	Write and quit	C	`:x`	Write (only if changed) and quit	C
`ZZ`	Write (if changed) and quit	N	`ZQ`	Force quit without saving	N
`:qa`	Quit all open buffers	C	`:qa!`	Force quit all buffers, discard all changes	C

Modes

`i`	Enter Insert mode at cursor	N	`Esc`	Return to Normal mode	ANY
`Ctrl+C`	Alternative to Esc — return to Normal	I	`v`	Enter character-wise Visual mode	N
`V`	Enter line-wise Visual mode	N	`Ctrl+V`	Enter block-wise (column) Visual mode	N

`Ctrl+G`	Show filename, line count, cursor position	N			

Navigation — Moving the Cursor

BASIC MOVEMENT

`h / j / k / l`	Left / down / up / right	N	`{n}h/j/k/l`	Move n times	N

WORD MOVEMENT

`w / W`	Forward to start of next word / WORD	N	`e / E`	Forward to end of word / WORD	N
`b / B`	Backward to start of word / WORD	N	`ge`	Backward to end of previous word	N

LINE MOVEMENT

`0`	Start of line (column 1)	N	`^`	First non-whitespace character	N
`$`	End of line	N	`g_`	Last non-whitespace character	N
`f{char} / F{char}`	Jump to char on line (forward/back)	N	`t{char} / T{char}`	Jump before/after char on line	N
`; / ,`	Repeat last f/t in same / opposite direction	N			

SENTENCE & PARAGRAPH

`) / (`	Next / previous sentence	N	`} / {`	Next / previous paragraph	N

FILE-LEVEL JUMPS

`gg`	Jump to first line	N	`G`	Jump to last line	N
`{n}G / :{n}`	Jump to line n	N	`{n}%`	Jump to n% through the file	N

SCROLLING

`Ctrl+F / Ctrl+B`	Full page forward / backward	N	`Ctrl+D / Ctrl+U`	Half page down / up	N
`Ctrl+E / Ctrl+Y`	Scroll screen down / up one line	N	`zz / zt / zb`	Centre / top / bottom current line on screen	N

Search

`/{pattern}`	Search forward for pattern	N	`?{pattern}`	Search backward for pattern	N
`n / N`	Next / previous search match	N	`* / #`	Search exact word under cursor (fwd/back)	N

`g* / g#`	Search partial word under cursor (fwd/back)	N	`:noh`	Clear search highlights	C
`:set hlsearch`	Highlight all search matches	C	`:set incsearch`	Incremental search as you type	C
`:set ignorecase`	Case-insensitive search	C	`:set smartcase`	Case-sensitive when pattern has uppercase	C

Editing Text

DELETING

`d{motion}`	Delete text covered by motion	N	`dd / {n}dd`	Delete current line / n lines	N
`D`	Delete from cursor to end of line	N	`x / X`	Delete character under / before cursor	N
`xp`	Swap current character with next	N			

CHANGING

`c{motion}`	Change text covered by motion	N	`cc / C`	Change line / from cursor to end of line	N
`s / S`	Substitute character / line + Insert mode	N	`ciw / caw`	Change inner / around word	N
`ci" / ca"`	Change inside / around double quotes	N	`ci( / ca(`	Change inside / around parentheses	N

UNDO & REDO

`u`	Undo last change	N	`U`	Undo all changes to current line	N
`Ctrl+R`	Redo	N			

YANK (COPY) & PUT (PASTE)

`y{motion}`	Yank text covered by motion	N	`yy / {n}yy`	Yank current line / n lines	N
`p / P`	Put after / before cursor	N	`gp / gP`	Put and move cursor to end of paste	N

INSERT MODE EDITING

`Ctrl+W`	Delete word before cursor	I	`Ctrl+U`	Delete from cursor to start of line	I

Registers

`"x{op}`	Use named register x (e.g. "ayy, "ap)	N	`"X{op}`	Append to named register x (uppercase)	N
`"0p`	Paste from yank register (safe after deletes)	N	`"+y / "+p`	Yank to / paste from system clipboard	N

`"_d{motion}`	Delete to black hole register (truly removes)	N	`:registers`	Show contents of all registers	C

Substitution & Spell Checking

SUBSTITUTION

`:s/old/new`	Replace first occurrence on current line	C	`:s/old/new/g`	Replace all on current line	C
`:s/old/new/gc`	Replace all on line with confirmation	C	`:%s/old/new/g`	Replace all in entire file	C
`:5,15s/old/new/g`	Replace in line range	C	`:'<,'>s/old/new/g`	Replace in Visual selection	C

SPELL CHECKING

`:set spell`	Enable spell checking	C	`:set spelllang=en_gb`	Set spell check language	C
`:set nospell`	Disable spell checking	C	`]s / [s`	Jump to next / previous misspelled word	N
`z=`	Show spelling suggestions	N	`zg / zw`	Add to personal dictionary / bad-word list	N

Marks & the Jump List

SETTING & JUMPING TO MARKS

`m{a-z}`	Set a local mark (current buffer)	N	`m{A-Z}`	Set a global mark (across files, persists)	N
`'{mark}`	Jump to start of line containing mark	N	`` `{mark} ``	Jump to exact position (line + column) of mark	N
`'' / ```	Toggle back to line / exact position before last jump	N			

AUTOMATIC MARKS

`` `. / '. ``	Exact position / line of last change	N	`` `[ and `] ``	Start and end of last changed or yanked text	N
`` `< and `> ``	Start and end of last Visual selection	N	`` `^ ``	Position where Insert mode was last exited	N

MANAGING MARKS

`:marks`	List all current marks	C	`:delmarks {letter}`	Delete a named mark	C
`:delmarks!`	Delete all lowercase marks in buffer	C			

MARKS AS OPERATOR TARGETS

`y`{mark}`	Yank from cursor to exact position of mark	N	`d'{mark}`	Delete from cursor to line of mark	N

`c`{mark}`	Change from cursor to exact position of mark	N			

JUMP LIST

`Ctrl+O`	Go to older (previous) position in jump list	N	`Ctrl+I`	Go to newer (next) position in jump list	N
`:jumps`	Display the full jump history	C			

Buffers, Windows & Tabs

BUFFERS

`:ls`	List all open buffers	C	`:bn / :bp`	Next / previous buffer	C
`:bf / :bl`	First / last buffer	C	`:b {n} / :b {name}`	Switch to buffer by number / name	C
`Ctrl+^`	Toggle between current and alternate buffer	N	`:bd / :bd!`	Delete / force-delete current buffer	C
`:bw`	Wipe buffer (remove from list entirely)	C			

SPLITS

`:split / :sp`	Horizontal split	C	`:vsplit / :vsp`	Vertical split	C
`Ctrl+W s / v`	Horizontal / vertical split (shortcut)	N	`Ctrl+W h/j/k/l`	Move to window left/down/up/right	N
`Ctrl+W w / W`	Cycle to next / previous window	N	`Ctrl+W =`	Equalise all window sizes	N
`Ctrl+W _ / |`	Maximise window height / width	N	`Ctrl+W c`	Close current window	N
`Ctrl+W o / :only`	Close all other windows	N	`Ctrl+W H/J/K/L`	Move window to edge of screen	N
`:resize {n}`	Set window height to n lines	C	`:vertical resize {n}`	Set window width to n columns	C

TABS

`:tabnew`	Open new tab	C	`:tabclose`	Close current tab	C
`gt / gT`	Next / previous tab	N	`{n}gt`	Go to tab n	N
`:tabs`	List all tabs	C			

READING FILES INTO BUFFER

`:r {file}`	Insert file contents below current line	C	`:0r {file}`	Insert file at top of buffer	C
`:r! {cmd}`	Insert shell command output into buffer	C			

Diff Mode

`vim -d file1 file2`	Open two files in diff mode	S	`vimdiff file1 file2`	Alias for vim -d	S
`:diffsplit {file}`	Open file in diff mode (horizontal split)	C	`:vert diffsplit {file}`	Open file in diff mode (vertical split)	C
`]c / [c`	Jump to next / previous difference	N	`do`	Diff Obtain — pull change from other file	N
`dp`	Diff Put — push change to other file	N	`:diffupdate`	Recalculate and refresh diff highlights	C
`:set scrollbind`	Link scrolling of multiple windows	C	`set diffopt+=vertical`	Always use vertical diff layout	RC

File Navigation

`gf`	Open file whose name is under the cursor	N	`gF`	Open file under cursor at the line number shown	N
`Ctrl+W f`	Open file under cursor in a new split	N	`Ctrl+W gf`	Open file under cursor in a new tab	N
`vim {file}.zip`	Browse & edit files inside a ZIP archive	S	`vim {file}.tar.gz`	Browse & edit files inside a TAR archive	S
`:set path+=src/**`	Add directories to search path for gf	C			

External Commands & Shell Filters

`:! {cmd}`	Run shell command; show output in overlay	C	`:r ! {cmd}`	Insert shell output below cursor	C
`:0r ! {cmd}`	Insert shell output at top of file	C	`:%! {cmd}`	Filter entire buffer through external command	C
`:{range}! {cmd}`	Filter a line range through external command	C	`'<,'>! {cmd}`	Filter Visual selection through external command	V
`:%!jq .`	Format JSON with jq	C	`:%!sort`	Sort all lines alphabetically	C
`:%!xxd`	Convert buffer to hex dump	C	`:%!xxd -r`	Convert hex dump back to binary	C
`% (in :! commands)`	Expands to the current filename	C			

Configuration (.vimrc)

RELOADING

`:source ~/.vimrc`	Reload vimrc without restarting Vim	C	`:so %`	Source the current file	C

QUERYING SETTINGS

`:set`	Show all non-default settings active	C	`:set all`	Show every setting with its current value	C
`:set {option}?`	Query the current value of an option	C	`:set no{option}`	Disable a boolean option	C
`:set {option}!`	Toggle a boolean option	C			

COMMON SETTINGS

`set nocompatible`	Disable Vi compatibility (always first)	RC	`set number / nonu`	Show / hide line numbers	RC
`set relativenumber`	Relative line numbers	RC	`set cursorline`	Highlight the current line	RC
`set hidden`	Allow switching buffers without saving	RC	`set wildmenu`	Enhanced Tab-completion menu	RC
`set tabstop=4`	Tab display width	RC	`set shiftwidth=4`	Indent width for >> and <<	RC
`set expandtab`	Insert spaces instead of tabs	RC	`set clipboard=unnamedplus`	Use system clipboard by default	RC
`set history=1000`	Increase command history size	RC	`set undolevels=1000`	Increase undo history depth	RC
`set backspace=indent,eol,start`	Sensible Backspace in Insert mode	RC			

COLOUR SCHEMES

`:syntax on / off`	Enable / disable syntax highlighting	C	`:colorscheme {name}`	Switch to a colour scheme	C
`:colorscheme <Tab>`	Browse all available colour schemes	C			

KEY MAPPINGS

`nnoremap {key} {action}`	Map key in Normal mode (non-recursive)	RC	`inoremap {key} {action}`	Map key in Insert mode	RC
`vnoremap {key} {action}`	Map key in Visual mode	RC	`let mapleader = ","`	Set the leader key	RC

ABBREVIATIONS

`:ab {trigger} {expansion}`	Define an abbreviation interactively	C	`iabbrev {trigger} {expansion}`	Define an abbreviation in vimrc	RC
`Ctrl+V (in Insert)`	Prevent abbreviation from triggering	I			

CUSTOM COMMANDS & HELP

`:command! {Name} {action}`	Define a custom Command-mode command	C	`:help`	Open the built-in help index	C
`:help {topic}`	Open help for a specific topic	C	`:help option-summary`	Browse all available settings	C
`Ctrl+] (in help)`	Follow a help hyperlink	N			