



Vim Cheat Sheet

🕒 Created time	@November 29, 2025 8:26 PM
🕒 Last Edited	@November 30, 2025 7:08 AM
🏷️ Tags	

1. Navigation (Normal Mode)

Command	Action	Example / Mnemonic
h	Move Left	Left finger
j	Move Down	J ump down
k	Move Up	K limb up
l	Move Right	Right finger
G	Go to last line	G round floor (bottom)
gg	Go to first line	gg (Go Go to top)
[#]G	Go to specific line number	42G goes to line 42
CTRL-G	Show file status	Shows current line number & file path
CTRL-O	Jump back	O ld position (where you just were)
CTRL-I	Jump forward	I nto the future (redo jump)

2. Insert & Editing

Command	Action	Example / Mnemonic
i	Insert before cursor	I nsert text here
a	Append after cursor	A ppend after character
A	Append at end of line	A fter all text on line
o	Open new line below	O pen below

Command	Action	Example / Mnemonic
O	Open new line above	O ver the current line
x	Delete character	Like crossing out a letter
r	Replace single character	rx replaces current char with 'x'
R	Replace Mode	Overwrites text as you type
u	Undo	U ndo mistake
CTRL-R	Redo	R edo the undo

3. The Grammar: **[Operator]** + **[Count]** + **[Motion]**

Think of this as speaking a sentence: "Verb" (Operator) + "Adjective" (Count) + "Noun" (Motion).

Operators (The Verbs)

- **d**: Delete
- **c**: Change (Delete + Enter Insert Mode)
- **y**: Yank (Copy)

Motions (The Nouns)

- **w**: Start of next word
- **e**: End of current word
- **\$**: End of line
- **o**: Start of line
- **b**: previous word or start of current words

Common Combinations

Command	Action	Logic
dw	Delete word	d delete w ord
cw	Change word	c hange w ord (rewrites it)
d\$	Delete to end of line	d delete to \$ (end)

Command	Action	Logic
<code>yy</code>	Yank (Copy) line	Doubles act on whole line
<code>dd</code>	Delete (Cut) line	Doubles act on whole line
<code>p</code>	Paste (Put)	P ut text after cursor

Using Counts

Command	Action	Logic
<code>2w</code>	Move 2 words forward	2 + Word
<code>d2w</code>	Delete 2 words	d delete + 2 + w ords
<code>3dd</code>	Delete 3 lines	3 + d delete d lines

4. Search & Replace

Command	Action	Example
<code>/text</code>	Search forward	<code>/error</code> finds "error"
<code>?text</code>	Search backward	<code>?config</code> looks up for "config"
<code>n</code>	Next match	Go to next result
<code>N</code>	Previous match	Go to previous result
<code>%</code>	Find matching pair	Jumps between <code>{</code> and <code>}</code>
<code>:s/old/new</code>	Replace first match on line	<code>:s/cat/dog</code>
<code>:%s/old/new/g</code>	Replace globally (whole file)	<code>:%s/foo/bar/g</code>
<code>:%s/old/new/gc</code>	Replace globally with check	Asks for confirmation each time

5. Tabs & File Management

Command	Action	Example
<code>:tabnew [file]</code>	Open new tab	<code>:tabnew script.py</code>
<code>gt</code>	Go to next tab	G o Tab
<code>gT</code>	Go to previous tab	G o Tab (back)
<code>:w [file]</code>	Write (Save) to file	<code>:w backup.txt</code>

Command	Action	Example
<code>:r [file]</code>	Read file into current	<code>:r data.txt</code> inserts data here

6. Visual Mode (Selecting Text)

Command	Action	Workflow Example
<code>v</code>	Visual Character Mode	<code>v</code> → move arrows → <code>y</code> (copy specific chars)
<code>V</code>	Visual Line Mode	<code>V</code> → move down → <code>d</code> (delete block of lines)
Export Selection	Save highlighted text to file	Select text (<code>v</code>), then type <code>:w filename</code>

Visual Block

Command	Action	Logic
<code>CTRL + q</code>	Enter Visual Block Mode	Selects rectangular columns.
<code>Shift + I</code>	Insert at start of block	Insert before selection add <code>#</code> to block to comment
<code>Shift + A</code>	Append at end of block	Append after selection.
<code>c</code>	Change block	Deletes block and lets you type over it.
<code>CTRL + q</code> → <code>j</code> → <code>Shift + I</code> → <code>#</code> → <code>ESC</code>	Comment out a block	

7. Power command

Command	Action	Why it's useful
<code>.</code>	Repeat last change	The biggest time saver in Vim.
<code>CTRL + q</code>	Visual Block mode	Edit columns (great for YAML/bulk edits).

<code>*</code>	Search word under cursor (Forward)	Instant search without typing.
<code>#</code>	Search word under cursor (Backward)	Same as <code>*</code> , but goes up.
<code>^</code>	Go to start of text on line	Better than <code>0</code> for indented code (Python).
<code>ciw</code>	C hange I n W ord	Delete current word & switch to Insert mode (super precise).
<code>dd</code> vs <code>D</code>	<code>D</code> deletes from cursor to end of line	Faster than <code>d\$</code> .
<code>:! [cmd]</code>	Execute shell command	<code>:!ls -la</code> (Lists files)