

Vim

Cursor Movement

h	- Move cursor left
j	- Move cursor down
k	- Move cursor up
l	- Move cursor right
gj	- Move cursor down (multi-line text)
gk	- Move cursor up (multi-line text)
H	- Move to top of screen
M	- Move to middle of screen
L	- Move to bottom of screen
w	- Jump forwards to the start of a word
W	- Jump forwards to the start of a word (words can contain punctuation)
e	- Jump forwards to the end of a word
E	- Jump forwards to the start of a word (words can contain punctuation)
b	- Jump backwards to the start of a word
B	- Jump backwards to the start of a word (words can contain punctuation)
ge	- Jump backwards to the end of a word
gE	- Jump backwards to the end of a word (words can contain punctuation)
%	- Move cursor to matching character (default supported pairs: <code>()</code> , <code>{}</code> , <code>[]</code> - use <code>:h matchpairs</code> in Vim for more info)
0	- Jump to the start of the line
^	- Jump to the first non-blank character of the line
\$	- Jump to the end of the line
g_	- Jump to the last non-blank character of the line
gg	- Go to the first line of the document
G	- Go to the last line of the document
5gg or 5G	- Go to line 5
gd	- Move to local declaration
gD	- Move to global declaration
Fx	- Jump to next occurrence of character x
tx	- Jump to before next occurrence of character x
Fx	- Jump to the previous occurrence of character x
Tx	- Jump to after previous occurrence of character x
;	- Repeat previous f, t, F or T movement
,	- Repeat previous f, t, F or T movement, backwards
}	- Jump to next paragraph (or function/block, when editing code)
{	- Jump to previous paragraph (or function/block, when editing code)
zz	- Center cursor on screen
zt	- Position cursor on top of the screen
zb	- Position cursor on bottom of the screen
Ctrl + e	- Move screen down one line (without moving cursor)
Ctrl + y	- Move screen up one line (without moving cursor)
Ctrl + b	- Move screen up one page (cursor to last line)
Ctrl + f	- Move screen down one page (cursor to first line)
Ctrl + u	- Move cursor and screen up 1/2 page
Ctrl + d	- Move cursor and screen down 1/2 page

Tip: Prefix a cursor movement command with a number to repeat it. For example, `4j` moves down 4 lines.

Insert Mode - Inserting / Appending Text

i	- Insert before the cursor
I	- Insert at the beginning of the line
a	- Insert (append) after the cursor
A	- Insert (append) at the end of the line
o	- Append (open) a new line below the current line
O	- Append (open) a new line above the current line
ea	- Insert (append) at the end of the word
Ctrl + h	- Delete the character before the cursor during insert mode
Ctrl + w	- Delete word before the cursor during insert mode
Ctrl + j	- Add a line break at the cursor position during insert mode
Ctrl + t	- Indent (move right) line one shiftwidth during insert mode
Ctrl + d	- De-indent (move left) line one shiftwidth during insert mode
Ctrl + n	- Insert (auto-complete) next match before the cursor during insert mode
Ctrl + p	- Insert (auto-complete) previous match before the cursor during insert mode
Ctrl + rx	- Insert the contents of register x
Ctrl + ox	- Temporarily enter normal mode to issue one normal-mode command x
Esc or Ctrl + c	- Exit insert mode

Editing

r	- Replace a single character
R	- Replace more than one character, until ESC is pressed
J	- Join line below to the current one with one space in between
gj	- Join line below to the current one without space in between
gwl	- Reflow paragraph
g~	- Switch case up to motion
gu	- Change to lowercase up to motion
gu	- Change to uppercase up to motion
cc	- Change (replace) entire line
c\$ or C	- Change (replace) to the end of the line
cW	- Change (replace) entire word
ci" or ci'	- Change (replace) string in double quotes or quotes
cw or ce	- Change (replace) to the end of the word
s	- Delete character and substitute text
S	- Delete line and substitute text (same as cc)
xp	- Transpose two letters (delete and paste)
u	- Undo
U	- Restore (undo) last changed line
Ctrl + r	- Redo
.	- Repeat last command

Marking Text (Visual Mode)

v	- Start visual mode, mark lines, then do a command (like y-yank)
V	- Start linewise visual mode
o	- Move to other end of marked area
Ctrl + v	- Start visual block mode
O	- Move to other corner of block
aw	- Mark a word
ab	- A block with <code>()</code>
aB	- A block with <code>{}</code>
at	- A block with <code><></code> tags
ib	- Inner block with <code>()</code>
iB	- Inner block with <code>{}</code>
it	- Inner block with <code><></code> tags
i"	- Inner block with double quotes
i'	- Inner block with quotes
Esc or Ctrl + c	- Exit visual mode
Tip: Instead of <code>b</code> or <code>B</code> one can also use <code>(</code> or <code>{</code> respectively.	

Visual Commands

>	- Shift text right
<	- Shift text left
y	- Yank (copy) marked text
d	- Delete marked text
~	- Switch case
u	- Change marked text to lowercase
U	- Change marked text to uppercase

Registers

:register	- Show registers content
"xy	- Paste contents of register x
"xp	- Yank into register x
"+y	- Yank into the system clipboard register
"+p	- Paste from the system clipboard register
Tip: Registers are being stored in <code>~/.viminfo</code> , and will be loaded again on next restart of vim. Special registers:	
0	- Last yank
%	- Unnamed register, last delete or yank
#	- Current file name
#	- Alternate file name
*	- Clipboard contents (X11 primary)
+	- Clipboard contents (X11 clipboard)
/	- Last search pattern
:	- Last command-line
.	- Last inserted text
.	- Last small (less than a line) delete
=	- Expression register
=	- Black hole register

Marks and Positions

:marks	- List of marks
ma	- Set current position for mark a
'a	- Jump to position of mark a
y'a	- Yank text to position of mark a
'0	- Go to the position where Vim was previously exited
''	- Go to the position when last editing this file
.'	- Go to the position of the last change in this file
''	- Go to the position before the last jump
:jumps	- List of jumps
Ctrl + i	- Go to newer position in jump list
Ctrl + o	- Go to older position in jump list
:changes	- List of changes
g,	- Go to newer position in change list
g;	- Go to older position in change list
Ctrl +]	- Jump to the tag under cursor

Tip: To jump to a mark you can either use a backtick (```) or an apostrophe (`'`). Using an apostrophe jumps to the beginning (first non-blank) of the line holding the mark.

A local mark is specific to a particular file and is indicated by a lowercase letter, while global marks are universal across all of your files and use uppercase letters. That is, many files can have an ``a` mark, while `^A` points to a particular location in one particular file.

Macros

qa	- Record macro a
q	- Stop recording macro
@a	- Run macro a
@@	- Rerun last run macro

Comment

gc	- Toggle comment for selection
gcc	- Toggle comment for the current line

Surround

gs(or gs)	- Surround selection with <code>()</code>
gs{ or gs}	- Surround selection with <code>{}</code>
gs[or gs]	- Surround selection with <code>[]</code>
gs< or gs>	- Surround selection with <code><></code>
gs"	- Surround selection with <code>"</code>
gs'	- Surround selection with <code>'</code>
gs`	- Surround selection with <code>`</code>

Cut and Paste

yy	- Yank (copy) a line
2yy	- Yank (copy) 2 lines
yw	- Yank (copy) the characters of the word from the cursor position to the start of the next word
y\$w	- Yank (copy) word under the cursor
yaw	- Yank (copy) word under the cursor and the space after or before it
y\$	- Y - Yank (copy) to end of line
p	- Put (paste) the clipboard after cursor
P	- Put (paste) before cursor
gp	- Put (paste) the clipboard after cursor and leave cursor after the new text
gP	- Put (paste) before cursor and leave cursor after the new text
dd	- Delete (cut) a line
2dd	- Delete (cut) 2 lines
dw	- Delete (cut) the characters of the word from the cursor position to the start of the next word
d\$w	- Delete (cut) word under the cursor
daw	- Delete (cut) word under the cursor and the space after or before it
d\$	- D - Delete (cut) to the end of the line
x	- Delete (cut) character
:3,5d	- Delete lines starting from 3 to 5
:g/{pattern}/d	- Delete all lines containing pattern
:g!/{pattern}/d	- Delete all lines not containing pattern

Tip: You can also use the following characters to specify the range:

- `1, $d` - From the current line to the end of the file
- `1, 1d` - From the current line to the beginning of the file
- `:10, $d` - From the 10th line to the beginning of the file

Indent Text

>> or >	- Indent (move right) line one shiftwidth
<< or <	- De-indent (move left) line one shiftwidth
>%	- Indent a block with <code>()</code> or <code>{}</code> (cursor on brace)
<%	- De-indent a block with <code>()</code> or <code>{}</code> (cursor on brace)
>1b	- Indent inner block with <code>()</code>
>at	- Indent a block with <code><></code> tags
3==	- Re-indent 3 lines
=%	- Re-indent a block with <code>()</code> or <code>{}</code> (cursor on brace)
=1B	- Re-indent inner block with <code>{}</code>
gg=G	- Re-indent entire buffer
]p	- Paste and adjust indent to current line

Move Text

Alt + j	- Move line down
Alt + k	- Move line up

Exiting

:w	- Write (save) the file, but don't exit
:w !sudo tee %	- Write out the current file using sudo
:wq	- Write (save) and quit
:x	- Write (save) and quit
:q	- Quit (fails if there are unsaved changes)
:q! or ZQ	- Quit and throw away unsaved changes
:wqa	- Write (save) and quit all

Search and Replace

/pattern	- Search for pattern
?pattern	- Search backward for pattern
\vpattern	- Non-alphanumeric characters are interpreted as special regex symbols (no escaping needed)
n	- Repeat search in same direction
N	- Repeat search in opposite direction
:%s/old/new/g	- Replace all old with new throughout file
:%s/old/new/gc	- Replace all old with new throughout file with confirmations
:%s/old/new/gc1	- Replace all old with new throughout file with confirmations (case insensitive)
:%s/old/new/gcI	- Replace all old with new throughout file with confirmations (case sensitive)
:noh or ESC	- Remove highlighting of search matches
Space + rr	- Replace in buffer
Space + rw	- Replace in buffer (word)
Space + rR	- Replace in quickfix list
Space + rW	- Replace in quickfix list (word)

Tip: To replace a pattern for all files in the quickfix list the following command can be used: `ccdo %s/old/new/gc1 | update`. Items can be added to the quickfix list with `Ctrl + q` from the picker. To remove items from the quickfix list `dd` can be used.

Working with Multiple Files

:e[dit] file	- Edit a file in a new buffer
:%!xxd	- Switch to binary format for existing file
:bn[ext] or Tab	- Go to the next buffer
:bp[revious] or Shift + Tab	- Go to the previous buffer
:ls or :buffers	- List all open buffers
:bd[etete]	- Delete a buffer (close a file)
:b[uffer]#	- Go to a buffer by index #
:b[uffer] file	- Go to a buffer by file
:sp[lit] file	- Open a file in a new buffer and split window
:vs[plit] file	- Open a file in a new buffer and vertically split window
:vert[ical] ba[ll]	- Edit all buffers as vertical windows
Ctrl + ws	- Split window
Ctrl + vw	- Split window vertically
Ctrl + ww	- Switch windows
Ctrl + wq	- Quit a window
Ctrl + wx	- Exchange current window with next one
Ctrl + w=	- Make all windows equal height & width
Ctrl + wh	- Move cursor to the left window (vertical split)
Ctrl + wl	- Move cursor to the right window (vertical split)
Ctrl + wj	- Move cursor to the window below (horizontal split)
Ctrl + wk	- Move cursor to the window above (horizontal split)
Ctrl + wh	- Make current window full height at far left (leftmost vertical window)
Ctrl + wL	- Make current window full height at far right (rightmost vertical window)
Ctrl + wJ	- Make current window full width at the very bottom (bottommost horizontal window)
Ctrl + wK	- Make current window full width at the very top (topmost horizontal window)
gt	- Go to the next tab
gT	- Go to the previous tab

Diff

zf	- Manually define a fold up to motion
za	- Delete fold under the cursor
za	- Toggle fold under the cursor
zo	- Open fold under the cursor
zc	- Close fold under the cursor
zr	- Reduce (open) all folds by one level
zm	- Fold more (close) all folds by one level
zi	- Toggle folding functionality
]c	- Jump to start of next change
[c	- Jump to start of previous change
do or :diffg[et]	- Obtain (get) difference (from other buffer)
dp or :diffpu[t]	- Put difference (to other buffer)
:diffthis	- Make current window part of diff
:diff[update]	- Update differences
:diff[ff]	- Switch off diff mode for current window

Tip: The commands for folding (e.g. `za`) operate on one level. To operate on all levels, use uppercase letters (e.g. `zA`). To view the differences of files, one can directly start Vim in diff mode by running `vimdiff` in a terminal. One can even set this as `git difftool`.

Plugins

Gitsigns

]c	- Jump to next hunk
[c	- Jump to previous hunk
Space + gss	- Stage hunk
Space + gsr	- Reset hunk
Space + gsS	- Stage all hunks in current buffer
Space + gsR	- Reset all hunk in current buffer
Space + gsu	- Undo last hunk
Space + gsp	- Preview hunk
Space + gsb	- Git blame (short)
Space + gsB	- Git blame (full)
Space + gsd	- Toggle deleted
Space + gsq	- Show hunks via quickfix list
Space + gsl	- Show hunks via location list

Diffview

Space + gdo	- Open
Space + gdO	- Open (HEAD)
Space + gdc	- Close
Space + gdt	- Toggle files
Space + gdf	- Focus files
Space + gdhb	- Branch history
Space + gdhf	- File history
Tab	- Open diff for the next file
Shift + Tab	- Open diff for the previous file
Space + gdx	- Cycle layout through available layouts
[x	- Jump to previous conflict
]x	- Jump to next conflict
Space + gdcO	- Choose the OURS version of a conflict
Space + gdcT	- Choose the THEIRS version of a conflict
Space + gdcB	- Choose the BASE version of a conflict
Space + gdca	- Choose all the versions of a conflict
Space + gdcx	- Delete the conflict region

Snacks - Picker

Space + p	- Command palette
Space + e	- Explorer
Space + ff	- Files
Space + fs	- Search
Space + fb	- Buffers
Space + f/	- Search buffers
Space + fr	- Recent
Space + fR	- Resume
Space + fu	- Undo
Space + fk	- Keymaps
Space + fm	- Marks
Space + ft	- Todo comments
Space + fq	- Quickfix list
Space + fl	- Location list
Space + fd	- Buffer diagnostics
Space + fD	- Workspace diagnostics
Space + fls	- Buffer symbols
Space + flS	- Workspace symbols
Space + flr	- References
Space + flD	- Definitions
Space + fLD	- Declarations
Space + fly	- Type definitions
Space + fli	- Implementations
Space + fgf	- Git files
Space + fgb	- Git branches
Space + fgl	- Git log for (buffer)
Space + fgL	- Git log for (workspace)
Space + fgs	- Git status
Space + fgS	- Git stash
/	- Toggle focus
Enter	- Confirm
Down or Ctrl + j or Ctrl + n or j	- List down
Up or Ctrl + k or Ctrl + p or k	- List up
Ctrl + d	- List scroll down
Ctrl + u	- List scroll up
G	- List bottom
gg	- List top
Ctrl + Down	- History forward
Ctrl + Up	- History back
Tab	- Select and next
Shift + Tab	- Select and prev
Ctrl + a	- Select all
Ctrl + b	- Preview scroll up
Ctrl + f	- Preview scroll down
Ctrl + g	- Toggle live
Ctrl + q	- Quickfix list
Ctrl + s	- Edit split
Ctrl + v	- Edit vsplit
Esc or Ctrl + c or q	- Close

Snacks - Explorer

l	- Confirm
h	- Close directory
a	- Add
d	- Delete
r	- Rename
c	- Copy
m	- Move
o	- Open with system application
P	- Toggle preview
u	- Update
I	- Toggle ignored
H	- Toggle hidden
Z	- Close all
y	- Yank
s	- Search in directory
S	- Search in directory (case sensitive)
D	- Diff selected files

Todo Comments

]t	- Next todo comment
[t	- Previous todo comment

LSP

Space + lr	- Rename
Space + lca	- Code actions
Space + lf	- Format
Space + ll	- Lint
Space + lq	- Quickfix list
Space + lL	- Location list
gd	- Go to definition
gr	- Go to references
gI	- Go to implementation
gy	- Go to type definition
gD	- Go to declaration
K	- Hover documentation
J	- Hover diagnostics

Treesitter

aa	- Select around parameter
ia	- Select inside parameter
af	- Select around function
if	- Select inside function
ac	- Select around class
ic	- Select inside class
al	- Select around loop
il	- Select inside loop
ab	- Select around block
ib	- Select inside block
]f	- Go to Next Start Function
]c	- Go to Next Start Class
]l	- Go to Next Start Loop
]b	- Go to Next Start Block
]F	- Go to Next End Function
]C	- Go to Next End Class
]L	- Go to Next End Loop
]B	- Go to Next End Block
[f	- Go to Previous Start Function
[c	- Go to Previous Start Class
[l	- Go to Previous Start Loop
[b	- Go to Previous Start Block
[F	- Go to Previous End Function
[C	- Go to Previous End Class
[L	- Go to Previous End Loop
[B	- Go to Previous End Block

Completions

Ctrl + Space	- Show completions, show / hide documentation
Ctrl + e	- Hide completions
Enter	- Accept completion
Down	- Select next completion
Up	- Select previous completion
Ctrl + n	- Select next completion
Ctrl + p	- Select previous completion
Ctrl + f	- Scroll down in documentation
Ctrl + b	- Scroll up in documentation

Multicursor

Space + mm or Ctrl + n	- Add cursor for next word
Space + mN	- Add cursor for all words
Space + ms	- Skip next word
Space + m + Down	- Add cursor below
Space + m + Up	- Add cursor above
Space + m + Left	- Rotate cursor (next)
Space + m + Right	- Rotate cursor (prev)
Space + mx	- Delete cursor
Space + ma	- Align cursors
Space + mS	- Split cursors
Space + mM	- Match cursors

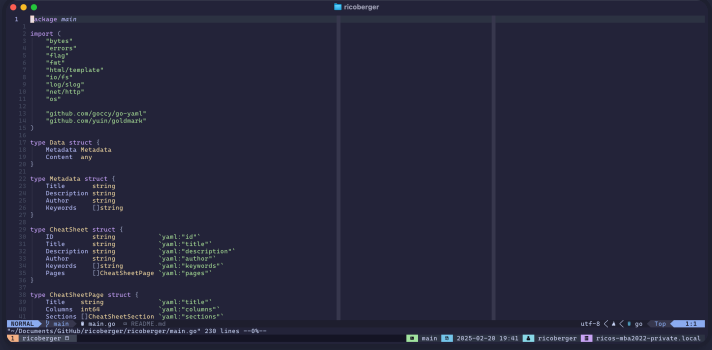
Copilot Chat

Space + ca	- Actions
Space + co	- Open
Space + cc	- Close
Space + ct	- Toggle
Space + cr	- Reset

Links

The Vim Cheat Sheet is inspired by the Vim Cheat Sheet from rtorr.

The configuration for the plugins with the mentioned keymaps can be found in my personal dofiles.



More Neovim related links:

- Website
- GitHub Repository
- Collections of Awesome Neovim Plugins
- Download Cheat Sheet as PDF | PNG