

Kakoune

A cheat sheet with the most important commands. For more, see kakoune.org

NORMAL MODE

Going to insert mode

- i a enter insert mode before selection / after selection
- I A ...at beginning of line / end of line
- o O ...on a new line below / above cursor
- \i prefix with \ to disable hooks, use to paste text into kakoune

Movement

- h j k l left / down / up / right
- <c-f> <c-b> scroll one page down / up
- <c-d> <c-u> scroll 1/2 page down / up

Select text and move around

A selection is a range of characters in the text. You can have multiple selections and change all of them simultaneously.

- w select next word (with following space)
- e to end of word (with preceding space)
- b back to previous word
- with alt: "word" is any non-space text, e.g. x+y-z is one word

- f<a-f> move right / left to the next
- t<a-t> move right / left to just before the next

- m select next sequence enclosed by brackets
- x % select whole line / whole buffer

- ; reduce selection to cursor
- <space> clear selections to keep only the main one
- C<a-C> copy selection to next / previous line
- <a-s> split selection on line boundaries
-) rotate selection (main selection moves to next one)
- (rotate selections backward

- Z z save / restore selection

- H J K L W E B M X like lower-case, but extend selection

Goto

- gh gi gl line begin / first character / line end
- gk gj ge first line / last line / last character of last line
- gt gc gb first / middle / last displayed line
- g. last modification position
- ga previous (alternate) buffer
- gf open file whose name is selected

use G instead of g to extend selection to target

View

- vt vv vb scroll to put selection at top / center / bottom

Searching

- /<pat> search for pattern
- n<a-n> select next / previous match
- N<a-N> add selection with next / previous match
- * search for selection
- s search within selection

Search and replace in the whole buffer (<ret> is the enter key):

%s<pattern><ret>c<replacement><esc>

o	a key	On the Mac, use the option key for alt .
<a-o>	alt and o	If that doesn't work, set "Use Option as
<c-o>	ctrl and o	Meta key" in Terminal preferences.

Changes

- y yank (copy) selection
- d<a-d> yank & delete selection / just delete selection
- c<a-c> change text (yank, delete, insert) / just delete, insert
- p P paste after the end / before the start of selection
- with alt: select pasted text afterwards
- . repeat last insert mode change
- u U undo / redo

- R replace selection with yanked text
- r<a-r> replace each character with o
- <a-j> join selected lines
- > < indent / unindent selected lines
- <a-o> <a-O> add empty line below / above cursor
- `~<a-`> lower / upper / swap case
- <a-)> <a-<> rotate selections (with multiple selections)
- & align selections by inserting spaces
- <a-&> copy indentation of main selection to all others

Object selection

Select an "object" such as a word or paragraph with two keystrokes: the first determines *how*, the second *what* to select.

- <a-.) repeat previous object selection

How:

- <a-a> whole object, e.g. a quoted string including quotes
- <a-i> inner object, e.g. a quoted string without the quotes
- [] from cursor to whole object start / end
- { } ...same but extend selection
- with alt: same but inner object start/end

What:

- | | | | |
|---|-------------------|-------|---------------------------|
| (| (parenthesis) | " | "double-quoted strings" |
| { | {curly braces} | ' | 'single-quoted strings' |
| [| [square brackets] | ` | `backtick-quoted strings` |
| < | <angle brackets> | w | word |
| s | sentence | <a-w> | any non-space text |
| p | paragraph | i | indentation block |

Macros

- Q q start or end macro recording / play macro

Registers

- "oy yank to register o (same for d c)
- "op paste from register o (same for P)
- also works for storing selection (Z z) and macros (Q q)

Jump list

Goto, search, buffer switch commands go on the jump list.

- <c-o> <c-i> jump back / forward in jump list
- <c-s> save current selection as jump step

External commands

- | send selection to external cmd, replace with output
- <a-|> send to external cmd, ignore output
- !<a-!> insert cmd output before / after selection

INSERT MODE

Insert mode is mostly about typing text directly.

<esc>	back to normal mode
<a-;>	back to normal mode for a single command
<c-r>○	insert register ○ (<c-r>" for default register)
<c-n> <c-p>	next / previous autocomplete candidate

COMMANDS

Pressing : in normal mode opens the command prompt.

:doc <topic>	open documentation for a topic
:w	write buffer
:w <file>	write to file
:e <file>	open file in new buffer
:q :wq :q!	quit / write and quite / quit without writing
:bn :bp	switch to next / previous buffer
:b <name>	switch to buffer <name>
:db <name>	delete buffer <name>
: <cmd>	<cmd> but don't add it to the prompt history

STARTING KAKOUNE

kak <file>	open file
cmd kak	edit command output
kak -s <name>	start named session
kak -d -s <name>	headless session (just server)
kak -c <name>	connect to session

With sessions, multiple editor instances can share open buffers and registers so you can e.g. copy in one and paste in another.

REGULAR EXPRESSIONS

For searches with / or s, these are special: \^\$. *+?[]{}|() \

\○	literal ○ (for special characters)
\n \t	newline character / tab character
.	any character
[abc]	a, b, or c
[a-z]	any character from a to z
[^a-z]	any character except a-z
	include a literal ^ or] with \^ or \]
\d	digits 0-9
\w	word characters A-Za-z0-9
\s \h	whitespace / whitespace except line breaks
	uppercase negates, e.g. \D is everything except digits
○?	zero or one ○
○*	zero or more ○
○{4}	four ○
○{2,4}	two to four ○
○+	one or more ○
○{4,}	four or more ○
○{,4}	up to four ○
^ \$	start / end of a line
\b	start or end of a word
(?i)	start case-insensitive matching
(?I)	back to case-sensitive matching
x y	match x or y (typically used with grouping)
(regex)	group part of a regex

The text matching the first, second, etc group is captured into register 1, 2, etc (entire match: register 0). You can combine s with "○p or <c-r>○ to rearrange text on multiple lines at once.

CONFIGURATION

Permanent config goes in **kakrc**, usually ~/.config/kak/kakrc

Options

echo %opt{<name>}	show current value
:set-option <scope> <name> <values>	set option
	<scope> can be global, buffer, window, or current
	current means the scope where the option is currently set
tabstop <n>	width of a tab character, default 8
indentwidth <n>	width for indentation, default 4, use 0 for tab
scrolloff <l>,<c>	lines, columns to keep visible around cursor, default 0,0
autocomplete insert prompt	
	insert: show completions automatically (default)
	prompt: show completions only on <c-n> <c-p>
extra_word_chars <list>	
	characters considered part of a word for w b e, default _
autoreload yes no ask	
	reload externally-modified file automatically
ui_options terminal_enable_mouse=true false	
	enable/disable mouse

Highlighters

Highlighters change how text is displayed.

show-matching	highlight matching bracket
show-whitespaces	show symbols for whitespace
number-lines	show line numbers
wrap	wrap text at window width
-word	wrap at word boundaries
-indent	preserve line indent

Set global highlighter (note space after / for add but not remove):

add-highlighter global/ <name> [<options>]
remove-highlighter global/<name> [<options>]

Mapping

Create keyboard mappings with

map global <mode> <key> <keys>

Example:

map global insert <a-space> <esc>

The mode determines where the key is available:

insert	insert mode	normal	normal mode
goto	after goto key (g)	view	after view key (v)
prompt	prompts, e.g. after : or /		
user	when user prefix key (default ,) is hit		

Keys mostly stand for themselves (a means a). Special cases:

<a-○> <c-○> <s-○>	alt / ctrl / shift and ○
<ret> <ins>	return or enter / insert key

These all do what you'd expect:

<space>	<esc>	<tab>	<backspace>		<F1> ... <F12>
<up>	<down>	<left>	<right>	<pageup>	<pagedown>
<plus>	<minus>	<home>	<end>	<semicolon>	<percent>

Map a key to trigger a command like this:

map global user w ': add-highlighter global/ wrap<ret>'