

Helix

Enjoy this cheat sheet at its fullest within Dash, the macOS documentation browser.

Movement

`h`
Left

Move left

`j`
Down

Move down

`k`
Up

Move up

`l`
Right

Move right

`w`

Move next word start

`b`

Move previous word start

`W`

Move next WORD start

`B`

Move previous WORD start

`E`

Move next WORD end

`t`

Find 'till next char

`f`

Find next char

`T`

Find 'till previous char

`F`

Find previous char

`G`

Go to line number `<n>`

`G`

Go to line number `<n>`

`Alt-.`

Repeat last motion (f, t or m)

`Home`

Move to the start of the line

`End`

Move to the end of the line

`Ctrl-b`
`PageUp`

Move page up

`Ctrl-f`

Move page down

PageDown	
Ctrl-u	Move half page up
Ctrl-d	Move half page down
Ctrl-i	Jump forward on the jumplist
Ctrl-o	Jump backward on the jumplist
Ctrl-s	Save the current selection to the jumplist

Changes

r	Replace with a character
R	Replace with yanked text
~	Switch case of the selected text
`	Set the selected text to lower case
Alt-`	Set the selected text to upper case
i	Insert before selection
a	Insert after selection (append)
I	Insert at the start of the line
A	Insert at the end of the line
o	Open new line below selection
0	Open new line above selection
.	Repeat last insert
u	Undo change
U	Redo change
Alt-u	Move backward in history
Alt-U	Move forward in history
y	Yank selection
p	Paste after selection
P	Paste before selection
" <reg>	Select a register to yank to or paste from
>	Indent selection
<	Unindent selection
=	Format selection (currently nonfunctional/disabled) (LSP)

d	Delete selection
Alt-d	Delete selection, without yanking
c	Change selection (delete and enter insert mode)
Alt-c	Change selection (delete and enter insert mode, without yanking)
Ctrl-a	Increment object (number) under cursor
Ctrl-x	Decrement object (number) under cursor
Q	Start/stop macro recording to the selected register (experimental)
q	Play back a recorded macro from the selected register (experimental)

Shell

I	Pipe each selection through shell command, replacing with output
Alt-	Pipe each selection into shell command, ignoring output
!	Run shell command, inserting output before each selection
Alt-!	Run shell command, appending output after each selection
\$	Pipe each selection into shell command, keep selections where command returned 0

Selection manipulation

`s`

Select all regex matches inside selections

`S`

Split selection into sub selections on regex matches

`Alt-s`

Split selection on newlines

`Alt_-`

Merge consecutive selections

`&`

Align selection in columns

`_`

Trim whitespace from the selection

`;`

Collapse selection onto a single cursor

`Alt-;`

Flip selection cursor and anchor

`Alt-:`

Ensures the selection is in forward direction

`,`

Keep only the primary selection

`Alt-,`

Remove the primary selection

`c`

Copy selection onto the next line (Add cursor below)

`Alt-c`

Copy selection onto the previous line (Add cursor above)

`(`

Rotate main selection backward

`)`

Rotate main selection forward

`Alt-(`

Rotate selection contents backward

`Alt-)`

Rotate selection contents forward

`%`

Select entire file

`x`

Select current line, if already selected, extend to next line

`X`

Extend selection to line bounds (line-wise selection)

`Alt-x`

Shrink selection to line bounds (line-wise selection)

`J`

Join lines inside selection

`Alt-J`

Join lines inside selection and select the inserted space

K	Keep selections matching the regex
Alt-K	Remove selections matching the regex
Ctrl-c	Comment/uncomment the selections
Alt-o Alt-up	Expand selection to parent syntax node (TS)
Alt-i Alt-down	Shrink syntax tree object selection (TS)
Alt-p Alt-left	Select previous sibling node in syntax tree (TS)
Alt-n Alt-right	Select next sibling node in syntax tree (TS)

Search	
/	Search for regex pattern
?	Search for previous pattern
n	Select next search match
N	Select previous search match
*	Use current selection as the search pattern

Minor modes

v	Enter select (extend) mode
g	Enter goto mode
m	Enter match mode
:	Enter command mode
z	Enter view mode
Z	Enter sticky view mode
Ctrl-w	Enter window mode
Space	Enter space mode

View Mode

z	Vertically center the line
c	
t	Align the line to the top of the screen
b	Align the line to the bottom of the screen
m	Align the line to the middle of the screen (horizontally)
j down	Scroll the view downwards
k up	Scroll the view upwards
Ctrl-f PageDown	Move page down
Ctrl-b PageUp	Move page up
Ctrl-d	Move half page down
Ctrl-u	Move half page up

Goto mode

- g** Go to line number else start of file
- e** Go to the end of the file
- f** Go to files in the selection
- h** Go to the start of the line
- l** Go to the end of the line
- s** Go to first non-whitespace character of the line
- t** Go to the top of the screen
- c** Go to the middle of the screen
- b** Go to the bottom of the screen
- d** Go to definition (LSP)
- y** Go to type definition (LSP)
- r** Go to references (LSP)
- i** Go to implementation (LSP)
- a** Go to the last accessed/alternate file
- m** Go to the last modified/alternate file
- n** Go to next buffer
- p** Go to previous buffer
- .** Go to last modification in current file
- j** Move down textual (instead of visual) line
- k** Move up textual (instead of visual) line

Match mode

`m`

Goto matching bracket (TS)

`s <char>`

Surround current selection with

`r <from><to>`

Replace surround character with

`d <char>`

Delete surround character

`a <object>`

Select around textobject

`i <object>`

Select inside textobject

Window mode

W	Switch to next window
Ctrl-w	
V	Vertical right split
Ctrl-v	
S	Horizontal bottom split
Ctrl-s	
F	Go to files in the selection in horizontal splits
Ctrl-f	
H	Go to files in the selection in vertical splits
Ctrl-h	
Left	Move to left split
J	
Ctrl-j	Move to split below
Down	
K	
Ctrl-k	Move to split above
Up	
L	
Ctrl-l	Move to right split
Right	
Q	
Ctrl-q	Close current window
O	
Ctrl-o	Only keep the current window, closing all the others
H	Swap window to the left
J	Swap window downwards
K	Swap window upwards

 L

Swap window to the right

Space mode

- f** Open file picker
- F** Open file picker at current working directory
- b** Open buffer picker
- j** Open jumplist picker
- g** Debug (experimental)
- k** Show documentation for item under cursor in a popup (LSP)
- s** Open document symbol picker (LSP)
- S** Open workspace symbol picker (LSP)
- d** Open document diagnostics picker (LSP)
- D** Open workspace diagnostics picker (LSP)
- r** Rename symbol (LSP)
- a** Apply code action (LSP)
- h** Select symbol references (LSP)
- '** Open last fuzzy picker
- w** Enter window mode
- p** Paste system clipboard after selections
- P** Paste system clipboard before selections
- y** Join and yank selections to clipboard
- Y** Yank main selection to clipboard
- R** Replace selections by clipboard contents
- /** Global search in workspace folder
- ?** Open command palette

Popup

Ctrl-u

Scroll up

Ctrl-d

Scroll down

Unimpaired

[d]	Go to next diagnostic (LSP)
[d]	Go to previous diagnostic (LSP)
]D	Go to last diagnostic in document (LSP)
[D]	Go to first diagnostic in document (LSP)
]f	Go to next function (TS)
[f]	Go to previous function (TS)
]t	Go to next type definition (TS)
[t]	Go to previous type definition (TS)
]a	Go to next argument/parameter (TS)
[a]	Go to previous argument/parameter (TS)
]c	Go to next comment (TS)
[c]	Go to previous comment (TS)
]T	Go to next test (TS)
[T]	Go to previous test (TS)
]p	Go to next paragraph
[p]	Go to previous paragraph
]g	Go to next change
[g]	Go to previous change
]G	Go to last change
[G]	Go to first change
]Space	Add newline below
[Space]	Add newline above

Insert mode

Escape	Switch to normal mode
Ctrl-s	Commit undo checkpoint
Ctrl-x	Autocomplete
Ctrl-r	Insert a register content
Ctrl-w	Delete previous word
Alt-Backspace	
Alt-d	Delete next word
Alt-Delete	
Ctrl-u	Delete to start of line
Ctrl-k	Delete to end of line
Ctrl-h	
Backspace	Delete previous char
Shift-Backspace	
Ctrl-d	Delete next char
Delete	
Ctrl-j	Insert new line
Enter	
Up	Move to previous line
Down	Move to next line
Left	Backward a char
Right	Forward a char
PageUp	Move one page up
PageDown	Move one page down
Home	Move to line start
End	Move to line end

Picker

Shift-Tab

Up

Ctrl-p

Tab

Down

Ctrl-n

PageUp

Ctrl-u

PageDown

Ctrl-d

Home

Go to first entry

End

Go to last entry

Enter

Open selected

Ctrl-s

Open horizontally

Ctrl-v

Open vertically

Ctrl-t

Toggle preview

Escape

Ctrl-c

Close picker

Prompt

Escape

Ctrl-c

Alt-b

Ctrl-Left

Ctrl-b

Left

Alt-f

Ctrl-Right

Ctrl-f

Right

Ctrl-e

End

Ctrl-a

Home

Ctrl-w

Alt-Backspace

Ctrl-Backspace

Alt-d

Alt-Delete

Ctrl-Delete

Ctrl-u

Delete to start of line

Ctrl-k

Delete to end of line

Backspace

Ctrl-h

Delete previous char

Shift-Backspace

Delete

Ctrl-d

Delete next char

`Ctrl-s`

Insert a word under doc cursor, may be changed to
Ctrl-r Ctrl-w later

`Ctrl-p``Up`

Select previous history

`Ctrl-n``Down`

Select next history

`Ctrl-r`

Insert the content of the register selected by following
input char

`Tab`

Select next completion item

`BackTab`

Select previous completion item

`Enter`

Open selected

Commands

`:quit``:q`

Close the current view.

`:quit!``:q!`

Force close the current view,
ignoring unsaved changes.

`:open``:o`

Open a file from disk into the
current view.

`:buffer-close``:bc``:bclose`

Close the current buffer.

`:buffer-close!``:bc!``:bclose!`

Close the current buffer
forcefully, ignoring unsaved
changes.

`:buffer-close-others``:bco``:bcloseother`

Close all buffers but the currently
focused one.

`:buffer-close-others!``:bco!``:bcloseother!`

Force close all buffers but the
currently focused one.

`:buffer-close-all``:bca``:bcloseall`

Close all buffers without quitting.

`:buffer-close-all!``:bca!``:bcloseall!`

Force close all buffers ignoring
unsaved changes without
quitting.

`:buffer-next``:bn``:bnext`

Goto next buffer.

<code>:buffer-previous</code>	<code>:bp</code> <code>:bprev</code>	Goto previous buffer.
	<code>:write</code> <code>:w</code>	Write changes to disk. Accepts an optional path (<code>:write some/path.txt</code>)
	<code>:write!</code> <code>:w!</code>	Force write changes to disk creating necessary subdirectories. Accepts an optional path (<code>:write! some/path.txt</code>)
	<code>:write-buffer-close</code> <code>:wbc</code>	Write changes to disk and closes the buffer. Accepts an optional path (<code>:write-buffer-close some/path.txt</code>)
	<code>:write-buffer-close!</code> <code>:wbc!</code>	Force write changes to disk creating necessary subdirectories and closes the buffer. Accepts an optional path (<code>:write-buffer-close! some/path.txt</code>)
	<code>:new</code> <code>:n</code>	Create a new scratch buffer.
	<code>:format</code> <code>:fmt</code>	Format the file using the LSP formatter.
	<code>:indent-style</code>	Set the indentation style for editing. ('t' for tabs or 1-8 for number of spaces.)
	<code>:line-ending</code>	Set the document's default line ending. Options: crlf, lf.
	<code>:earlier</code> <code>:ear</code>	Jump back to an earlier point in edit history. Accepts a number of steps or a time span.

<code>:later</code>	Jump to a later point in edit history. Accepts a number of steps or a time span.
<code>:lat</code>	
<code>:write-quit</code>	Write changes to disk and close the current view. Accepts an optional path (:wq some/path.txt)
<code>:wq</code>	
<code>:x</code>	
<code>:write-quit!</code>	Write changes to disk and close the current view forcefully. Accepts an optional path (:wq! some/path.txt)
<code>:wq!</code>	
<code>:x!</code>	
<code>:write-all</code>	Write changes from all buffers to disk.
<code>:wa</code>	
<code>:write-quit-all</code>	Write changes from all buffers to disk and close all views.
<code>:wqa</code>	
<code>:xa</code>	
<code>:write-quit-all!</code>	Write changes from all buffers to disk and close all views forcefully (ignoring unsaved changes).
<code>:wqa!</code>	
<code>:xa!</code>	
<code>:quit-all</code>	Close all views.
<code>:qa</code>	
<code>:quit-all!</code>	Force close all views ignoring unsaved changes.
<code>:qa!</code>	
<code>:cquit</code>	Quit with exit code (default 1). Accepts an optional integer exit code (:cq 2).
<code>:cq</code>	
<code>:cquit!</code>	Force quit with exit code (default 1) ignoring unsaved changes. Accepts an optional integer exit code (:cq! 2).
<code>:cq!</code>	
<code>:theme</code>	Change the editor theme (show current theme if no name)

	<p>specified).</p>
<code>:clipboard-yank</code>	<p>Yank main selection into system clipboard.</p>
<code>:clipboard-yank-join</code>	<p>Yank joined selections into system clipboard. A separator can be provided as first argument. Default value is newline.</p>
<code>:primary-clipboard-yank</code>	<p>Yank main selection into system primary clipboard.</p>
<code>:primary-clipboard-yank-join</code>	<p>Yank joined selections into system primary clipboard. A separator can be provided as first argument. Default value is newline.</p>
<code>:clipboard-paste-after</code>	<p>Paste system clipboard after selections.</p>
<code>:clipboard-paste-before</code>	<p>Paste system clipboard before selections.</p>
<code>:clipboard-paste-replace</code>	<p>Replace selections with content of system clipboard.</p>
<code>:primary-clipboard-paste-after</code>	<p>Paste primary clipboard after selections.</p>
<code>:primary-clipboard-paste-before</code>	<p>Paste primary clipboard before selections.</p>
<code>:primary-clipboard-paste-replace</code>	<p>Replace selections with content of system primary clipboard.</p>
<code>:show-clipboard-provider</code>	<p>Show clipboard provider name in status bar.</p>
<code>:change-current-directory</code> <code>:cd</code>	<p>Change the current working directory.</p>

<code>:show-directory</code>	Show the current working directory.
<code>:encoding</code>	Set encoding. Based on https://encoding.spec.whatwg.org .
<code>:character-info</code>	Get info about the character under the primary cursor.
<code>:char</code>	
<code>:reload</code>	Discard changes and reload from the source file.
<code>:reload-all</code>	Discard changes and reload all documents from the source files.
<code>:update</code>	Write changes only if the file has been modified.
<code>:u</code>	
<code>:lsp-workspace-command</code>	Open workspace command picker
<code>:lsp-restart</code>	Restarts the Language Server that is in use by the current doc
<code>:lsp-stop</code>	Stops the Language Server that is in use by the current doc
<code>:tree-sitter-scopes</code>	Display tree sitter scopes, primarily for theming and development.
<code>:debug-start</code>	Start a debug session from a given template with given parameters.
<code>:dbg</code>	
<code>:debug-remote</code>	Connect to a debug adapter by TCP address and start a debugging session from a given template with given parameters.
<code>:dbg-tcp</code>	

	<code>:debug-eval</code>	Evaluate expression in current debug context.
	<code>:vsplit</code> <code>:vs</code>	Open the file in a vertical split.
	<code>:vsplit-new</code> <code>:vnew</code>	Open a scratch buffer in a vertical split.
	<code>:hsplit</code> <code>:hs</code> <code>:sp</code>	Open the file in a horizontal split.
	<code>:hsplit-new</code> <code>:hnew</code>	Open a scratch buffer in a horizontal split.
	<code>:tutor</code>	Open the tutorial.
	<code>:goto</code> <code>:g</code>	Goto line number.
	<code>:set-language</code> <code>:lang</code>	Set the language of current buffer (show current language if no value specified).
	<code>:set-option</code> <code>:set</code>	Set a config option at runtime. For example to disable smart case search, use <code>:set search.smart-case false</code> .
	<code>:toggle-option</code> <code>:toggle</code>	Toggle a boolean config option at runtime. For example to toggle smart case search, use <code>:toggle search.smart-case</code> .
	<code>:get-option</code> <code>:get</code>	Get the current value of a config option.
	<code>:sort</code>	Sort ranges in selection.
	<code>:rsort</code>	Sort ranges in selection in reverse order.

	<code>:reflow</code>	Hard-wrap the current selection of lines to a given width.
	<code>:tree-sitter-subtree</code> <code>:ts-subtree</code>	Display tree sitter subtree under cursor, primarily for debugging queries.
	<code>:config-reload</code>	Refresh user config.
	<code>:config-open</code>	Open the user config.toml file.
	<code>:config-open-workspace</code>	Open the workspace config.toml file.
	<code>:log-open</code>	Open the helix log file.
	<code>:insert-output</code>	Run shell command, inserting output before each selection.
	<code>:append-output</code>	Run shell command, appending output after each selection.
	<code>:pipe</code>	Pipe each selection to the shell command.
	<code>:pipe-to</code>	Pipe each selection to the shell command, ignoring output.
	<code>:run-shell-command</code> <code>:sh</code>	Run a shell command
	<code>:reset-diff-change</code> , <code>:diffget</code> , <code>:diffg</code>	Reset the diff change at the cursor position.
	<code>:clear-register</code>	Clear given register. If no argument is provided, clear all registers.

Notes

References

- <https://docs.helix-editor.com/keymap.html>

- <https://docs.helix-editor.com/commands.html>

You can modify and improve this cheat sheet [here](#)