

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

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

	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

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

View Mode

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

<code>m</code>	Goto matching bracket (TS)
<code>s <char></code>	Surround current selection with
<code>r <from><to></code>	Replace surround character with
<code>d <char></code>	Delete surround character
<code>a <object></code>	Select around textobject
<code>i <object></code>	Select inside textobject

Window mode

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

<code>]d</code>	Go to next diagnostic (LSP)
<code>[d</code>	Go to previous diagnostic (LSP)
<code>]D</code>	Go to last diagnostic in document (LSP)
<code>[D</code>	Go to first diagnostic in document (LSP)
<code>]f</code>	Go to next function (TS)
<code>[f</code>	Go to previous function (TS)
<code>]t</code>	Go to next type definition (TS)
<code>[t</code>	Go to previous type definition (TS)
<code>]a</code>	Go to next argument/parameter (TS)
<code>[a</code>	Go to previous argument/parameter (TS)
<code>]c</code>	Go to next comment (TS)
<code>[c</code>	Go to previous comment (TS)
<code>]T</code>	Go to next test (TS)
<code>[T</code>	Go to previous test (TS)
<code>]p</code>	Go to next paragraph
<code>[p</code>	Go to previous paragraph
<code>]g</code>	Go to next change
<code>[g</code>	Go to previous change
<code>]G</code>	Go to last change
<code>[G</code>	Go to first change
<code>]Space</code>	Add newline below
<code>[Space</code>	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 Alt-Backspace	Delete previous word
Alt-d Alt-Delete	Delete next word
Ctrl-u	Delete to start of line
Ctrl-k	Delete to end of line
Ctrl-h Backspace Shift-Backspace	Delete previous char
Ctrl-d Delete	Delete next char
Ctrl-j Enter	Insert new line
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	
<div>Shift-Tab</div> <div>Up</div> <div>Ctrl-p</div>	Previous entry
<div>Tab</div> <div>Down</div> <div>Ctrl-n</div>	Next entry
<div>PageUp</div> <div>Ctrl-u</div>	Page up
<div>PageDown</div> <div>Ctrl-d</div>	Page down
<div>Home</div>	Go to first entry
<div>End</div>	Go to last entry
<div>Enter</div>	Open selected
<div>Ctrl-s</div>	Open horizontally
<div>Ctrl-v</div>	Open vertically
<div>Ctrl-t</div>	Toggle preview
<div>Escape</div> <div>Ctrl-c</div>	Close picker

Prompt

<div>Escape</div> <div>Ctrl-c</div>	Close prompt
<div>Alt-b</div> <div>Ctrl-Left</div>	Backward a word
<div>Ctrl-b</div> <div>Left</div>	Backward a char
<div>Alt-f</div> <div>Ctrl-Right</div>	Forward a word
<div>Ctrl-f</div> <div>Right</div>	Forward a char
<div>Ctrl-e</div> <div>End</div>	Move prompt end
<div>Ctrl-a</div> <div>Home</div>	Move prompt start
<div>Ctrl-w</div> <div>Alt-Backspace</div> <div>Ctrl-Backspace</div>	Delete previous word
<div>Alt-d</div> <div>Alt-Delete</div> <div>Ctrl-Delete</div>	Delete next word
<div>Ctrl-u</div>	Delete to start of line
<div>Ctrl-k</div>	Delete to end of line
<div>Backspace</div> <div>Ctrl-h</div> <div>Shift-Backspace</div>	Delete previous char
<div>Delete</div> <div>Ctrl-d</div>	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

<div>:quit</div> <div>:q</div>	Close the current view.
<div>:quit!</div> <div>:q!</div>	Force close the current view, ignoring unsaved changes.
<div>:open</div> <div>:o</div>	Open a file from disk into the current view.
<div>:buffer-close</div> <div>:bc</div> <div>:bclose</div>	Close the current buffer.
<div>:buffer-close!</div> <div>:bc!</div> <div>:bclose!</div>	Close the current buffer forcefully, ignoring unsaved changes.
<div>:buffer-close-others</div> <div>:bco</div> <div>:bcloseother</div>	Close all buffers but the currently focused one.
<div>:buffer-close-others!</div> <div>:bco!</div> <div>:bcloseother!</div>	Force close all buffers but the currently focused one.
<div>:buffer-close-all</div> <div>:bca</div> <div>:bcloseall</div>	Close all buffers without quitting.
<div>:buffer-close-all!</div> <div>:bca!</div> <div>:bcloseall!</div>	Force close all buffers ignoring unsaved changes without quitting.
<div>:buffer-next</div> <div>:bn</div> <div>:bnext</div>	Goto next buffer.

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

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

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

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

<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, :diffget, :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](#)