Helix Modal selection editor

START HERE

- **w** — Select next word
- **g** › **d** — Go to definition
- **d** — Delete the selection
- **:w** — Save
- **x** — Select the line
- **i** — Insert before selection
- **Esc** — Back to Normal mode
- **Space** › **f** — Open file picker

Select w, x, s, % | **Refine** C, K, ;, , narrow → | **Act** d, c, y, > then →

CONCEPTS

Selection first, then action: w selects a word, d deletes it. There is no operator-pending state like in Vim.

Every selection has a cursor; several selections edit in parallel. ; collapses to one and , keeps only the primary.

Keys marked LSP need a language server, TS needs a tree-sitter grammar. Run hx --health to check a language.

MODES & FILES

Esc Return to Normal mode

i / **a**
Insert before / after selection

I / **A** Insert at line start / end

o / **O**
Open line below / above

v Select (extend) mode

Motions extend the selection instead of replacing it; Esc or v leaves.

:w Save file

:q Quit window

:wq / **:x** Save and quit

MOVEMENT

h / **j** / **k** / **l**
Left / down / up / right

MOVEMENT (cont.)

w / **b** / **e**
Next word start / previous start / end

W / **B** / **E**
Same, for WORDs

f / **t** Find / till next char

F / **T** Find / till previous char

Ctrl + **u** / **Ctrl** + **d**
Half page up / down

PageUp / **PageDown**
Page up / down

G Go to line number

Ctrl + **o** / **Ctrl** + **i**
Jumplist back / forward

GOTO & MATCH

g › **g**
File start (or line number)

g › **e** End of file

g › **d** Go to definition (LSP)

g › **r** Go to references (LSP)

g › **y**
Go to type definition (LSP)

g › **i**
Go to implementation (LSP)

g › **f**
Open file under selection

g › **n** / **g** › **p**
Next / previous buffer

GOTO & MATCH (cont.)

m › **s** › *<char>*
Surround selection with char

m › **d** › *<char>*
Delete surrounding char

m › **i** › *<obj>*
Select inside text object

Objects: w word, ( ) parens, " quotes, f function, t type, a argument, c comment.

m › **a** › *<obj>*
Select around text object

EDITING

d Delete selection

c
Change selection (delete + insert)

y Yank (copy)

p / **P** Paste after / before

u / **U** Undo / redo

r › *<char>*
Replace selection with char

R Replace with yanked text

> / **<** Indent / unindent

~ Switch case

Alt + **d** / **Alt** + **c**
Delete / change without yanking

Keeps the register intact, handy before a paste.

EDITING (cont.)

J Join lines

Ctrl + **c** Toggle comment

. Repeat last insert

= Format selection (LSP)

SELECTIONS

x
Select line, again extends down

% Select whole file

s
Select regex matches inside selection

The core multi-cursor move: select a block, press s, type a pattern, Enter.

S Split selection on regex

C Add cursor on next line

; Collapse to a single cursor

,
Keep only primary selection

K
Keep selections matching regex

Alt + **K**
Remove selections matching regex

Alt + **s**
Split selection on newlines

Turns a multi-line selection into one cursor per line.

SELECTIONS (cont.)

Alt + **o** / **Alt** + **i**
Expand / shrink syntax node (TS)

SEARCH

/ Search forward (regex)

n / **N** Next / previous match

*****
Search for selection as a word

Space › **/**
Search the whole workspace

v › **n**
Add next match to the selection

Press v first so n extends instead of replacing.

SPACE MENU & PICKERS

Space › **f**
File picker (workspace root)

Space › **b** Buffer picker

Space › **s**
Document symbols (LSP)

Space › **a**
Code actions (LSP)

Space › **r**
Rename symbol (LSP)

Space › **k** Hover docs (LSP)

Space › **g**
Changed files picker

SPACE MENU & PICKERS (cont.)

Space › **?** Command palette

Space › **y** / **Space** › **p**
Yank / paste via system clipboard

] › **d** / **[** › **d**
Next / previous diagnostic (LSP)

PICKER & COMPLETION

Ctrl + **n** / **Ctrl** + **p**
Next / previous entry

Tab and Shift+Tab or the arrows work too.

Enter Open selected

Ctrl + **t** Toggle preview

WINDOWS & SHELL

Ctrl + **w** › **v** / **Ctrl** + **w** › **s**
Vertical / horizontal split

Ctrl + **w** › **h/j/k/l**
Move between splits

|
Pipe selections through a command

Replaces each selection with the command output, e.g. | sort.

HELP

:tutor Interactive tutorial