

LEKTRA

High-performance Document and Image viewer that prioritizes screen space and control.

Important Links

Homepage <https://dheerajshenoy.github.io/lektra>

Configuration <https://dheerajshenoy.github.io/lektra/configuration.html>

Commands <https://dheerajshenoy.github.io/lektra/commands.html>

Code <https://codeberg.org/lektra/lektra>

Note: This guide assumes you have installed **LEKTRA** and have not changed the default keybindings.

Contents

1	Getting Started	3
1.1	Opening a file	3
1.2	Tabs and Splits	3
1.3	Keyboard Navigation and Keybindings	3
1.3.1	Scrolling and Movement	3
1.3.2	Page Navigation	3
1.3.3	Search	4
1.3.4	Zoom and View Control	4
1.3.5	Outline and History	4
1.3.6	Annotations and Interaction Modes	4
1.3.7	Links and Actions	5
1.3.8	Editing and UI	5
1.3.9	Rotation	5
2	Configuration	5
2.1	Key sections	5
2.2	Example snippet	6

3	Commands	6
4	Lua API	6
4.1	Configuration file	6
4.2	Namespaces	6
4.3	Quick examples	7
4.4	Type stubs	7

1 Getting Started

1.1 Opening a file

You can open a document in multiple ways:

- **Command line:** `lektra <path-to-file>`
- **File explorer:** Right-click a file and choose Open with lektra
- **From within lektra:** Press `o` to open the file picker

Supported formats include PDF, EPUB, XPS, CBZ, and common image formats. DjVu support is detected at runtime if `djvulibre` is installed.

1.2 Tabs and Splits

LEKTRA supports tabs and splits for viewing multiple documents simultaneously.

- `Ctrl+t` opens a new tab
- `Ctrl+w` closes the current tab
- `gt` / `gT` cycles to the next / previous tab
- `Ctrl+\` splits the view vertically
- `Ctrl+-` splits the view horizontally

1.3 Keyboard Navigation and Keybindings

The core philosophy of **LEKTRA** is keyboard-driven navigation (mouse support is available too). Default keybindings follow VIM conventions.

1.3.1 Scrolling and Movement

- **Vertical:** `j` scrolls down, `k` scrolls up
- **Horizontal:** `h` scrolls left, `l` scrolls right

1.3.2 Page Navigation

- `Shift+j` moves to the next page
- `Shift+k` moves to the previous page

- `g,g` jumps to the first page
- `Shift+g` jumps to the last page
- `Ctrl+g` go to a specific page (prompts for page number)

1.3.3 Search

- `/` opens or focuses the search bar
- `n` jumps to the next search result
- `Shift+n` jumps to the previous search result

1.3.4 Zoom and View Control

- `=` zooms in, `-` zooms out, `0` resets zoom

Fit modes:

- `Ctrl+Shift+W` fit to width
- `Ctrl+Shift+H` fit to height
- `Ctrl+Shift+=` fit to window
- `Ctrl+Shift+R` toggle auto-resize

1.3.5 Outline and History

- `t` opens the document outline (table of contents) overlay
- `Alt+Shift+H` opens the highlight annotation search overlay
- `Ctrl+o` goes back to the previous location in history

1.3.6 Annotations and Interaction Modes

Note: These keys toggle modes. Some modes require clicking or dragging on the page to take effect.

- `1` toggle text selection mode
- `2` toggle text highlight mode
- `3` toggle rectangle annotation mode
- `4` toggle region selection mode
- `5` toggle annotation popups

1.3.7 Links and Actions

- `f` link-hint visit (keyboard link navigation)
- `o` open file
- `Ctrl+s` save current document

1.3.8 Editing and UI

- `u` undo, `Ctrl+r` redo
- `i` invert colors
- `Ctrl+Shift+m` toggle menubar
- `:` open command palette

1.3.9 Rotation

- `<` rotate counter-clockwise, `>` rotate clockwise

2 Configuration

LEKTRA can be configured using **TOML** or **Lua**. Let us look at how to configure using TOML first. The configuration file should be at `$HOME/.config/lektra/config.toml`.

2.1 Key sections

`[window]` Window size, fullscreen, menubar visibility, title format, accent color.

`[page]` Page foreground and background colors.

`[statusbar]` Visibility and which fields to display (page number, progress, file info, etc.).

`[split]` Split behavior — focus follows mouse, dim inactive panes, etc.

`[portal]` Picture-in-picture portal overlay settings.

`[annotations.*]` Highlight, rectangle, and popup annotation appearance.

`[thumbnail_panel]` Thumbnail strip width, page numbers, and scrollbar visibility.

`[keymap]` Override any keybinding by specifying `action = "Key"`.

`[mousemap]` Mouse button bindings, similar to keymap.

2.2 Example snippet


```
[window]
menubar = false
accent = "#3daee9FF"

[statusbar]
visible = true
show_page_number = true
show_progress = true

[page]
bg = "#1e1e2e"
fg = "#cdd6f4"
```

For the full list of options see the [configuration reference](#) on the homepage.

3 Commands

Press  to open the command palette and run any command by name. Commands cover opening files, navigation, zoom, annotations, sessions, and more.

The full command reference is on the [commands page](#) of the homepage.

4 Lua API

LEKTRA embeds a Lua 5.x interpreter (build with `-DWITH_LUA=ON`). All APIs live under the global `lektra` table.

4.1 Configuration file

Place your Lua init file at `$HOME/.config/lektra/init.lua`. It is sourced at startup after the TOML config is applied.

4.2 Namespaces

`lektra.view` Document view helpers — navigation, zoom, rotation, layout, search, text extraction, outline, and per-view event listeners.

`lektra.opt` Read/write runtime options (fit mode, layout mode, DPR, statusbar fields, ...).

`lektra.cmd` Execute any lektra command by name from Lua.

`lektra.keymap` Bind keys to Lua functions at runtime.

`lektra.mousemap` Bind mouse buttons to Lua functions.

`lektra.event` Register global event listeners (document open/close, page change, ...).

`lektra.ui` Show message bars, input prompts, and pickers from Lua.

`lektra.bookmark` Programmatic bookmark management.

`lektra.tabs` Tab management — open, close, focus, list tabs.

`lektra.utils` Utility helpers (shell execute, clipboard, path helpers).

4.3 Quick examples

Bind a key to a Lua function:

```
lektra.keymap.map("n", "<C-e>", function()
  local v = lektra.view.current()
  lektra.ui.message("Page: " .. v:pageno())
end)
```

React to a page change event:

```
local v = lektra.view.current()
v:register("page_changed", function()
  print("Now on page", v:pageno())
end)
```

Open a file in a new tab:

```
lektra.tabs.open("/path/to/document.pdf")
```

4.4 Type stubs

LSP-friendly type stubs for the Lua API are installed at `$prefix/share/lektra/lua/` (Linux/macOS). Point your Lua language server at this directory for autocompletion.

```
// .luarc.json example
{
  "workspace.library": ["/usr/share/lektra/lua"]
}
```