

Configuring your lab

ar025 · 11 July 2026

A lab is configured by one file: `demolab.yaml` at the repository root. It does two jobs. It sets the **branding** (the site’s name, tagline, and byline) and it registers the **collections** that group your writings on the homepage. Every key inside it is optional: the engine merges what you write over its own defaults, so a missing key just means the default. The file itself, though, is not optional. It is the **lab marker**: the `demolab` command finds the lab root by walking up from wherever you are until it hits a `demolab.yaml`, the way git finds `.git`. Edit it freely, but don’t delete it.

Updates never touch this file. Like `HOUSESTYLE.local.md`, it is yours; the engine reads it and fills in the rest.

Branding

Six keys, each with an engine default:

- **name**: the site’s title. Renders as the homepage heading and in every page’s browser title. Default: `Demolab`.
- **description**: a one-line tagline shown under the homepage title. Default: none (no tagline line at all).
- **author**: the lab’s owner. Renders as a byline (“by Your Name”) under the homepage title, and as the `<meta name="author">` tag on every web page. Default: none.
- **contact**: an email address or URL. If set, the byline links to it: an address containing `@` becomes a `mailto:` link, anything else a plain link. Ignored unless **author** is also set. Default: none.

So a personal lab’s byline is two lines:

```
author: Ada Lovelace
contact: ada@example.com
```

- **book-title**: the document title of the bound book, `pdfs/book.pdf`. Default: `Demolab – the book`.
- **contents-title**: the heading over the book’s table of contents. Default: `Demolab – contents`.

Collections

Every writing declares a **collection**: in its meta, and the homepage lists one row per collection, linking to a page of its entries. A collection needs no registration to work: an unregistered slug is simply title-cased for display (`neuron-models` becomes “Neuron Models”). The **collections:** map in `demolab.yaml` upgrades that: per slug, a **label** (the display name) and a **description** (shown under the label on the homepage and at the top of the collection’s own page).

collection-order: is a list of slugs setting the homepage order. Collections you leave out of the list still appear; they trail after the listed ones. Two special cases:

- A writing with **no collection**: in its meta lands in a collection called `uncategorized` (displayed “Uncategorized”).
- A slide deck with **no collection**: lands in `slides`.

Curated ordering

Within a collection page, entries normally list newest first (settled work leads: final entries before drafts, then newest id first). For a collection that reads in a deliberate sequence, like this

documentation, you can curate the order instead. This is not a `demolab.yaml` key; it lives in each writing's meta:

```
#let meta = (  
  title: "Getting started",  
  // ...  
  order: 1,  
)
```

`order`: is an integer rank. The moment **any** entry in a collection carries one, that whole collection becomes curated: its page lists entries by ascending `order`, and unranked entries trail in id order. Rank all of a curated collection's entries, not a few; a half-ranked collection reads as an accident.

A worked example

The `demolab.yaml` behind this documentation site is:

```
name: Demolab  
book-title: Demolab – the book  
contents-title: Demolab – contents  
description: A lab notebook for computational science – reproducible results,  
published and citable.  
  
collection-order: [documentation, slides]  
collections:  
  documentation:  
    label: Documentation  
    description: "How demolab works, in reading order: what it is and why, setting  
up, the contract, the mechanics of experiments and writeups, and running the lab day  
to day."  
  slides:  
    label: Talks & slides  
    description: Deck PDFs from talks – paged-only, linked as PDF.
```

Delete a key and its default takes over; add a collection when a new theme emerges. The config is small on purpose: branding, grouping, order, and nothing that could contradict what the runs actually produced.