Table of Contents

1. Installation

1.1 Requirements

Python 3.10 or later. Tested on 3.10, 3.11, 3.12 and 3.13; `pip` will refuse to install on anything older rather than failing later at import.

Everything below is pulled in automatically by `pip install prodockit`, except where noted:

Requirement	Needed for
`Markdown` (>= 3.10.3)	every extension
`zensical` (>= 0.0.55)	Zensical integration and `prodockit.zensical_macros`
PyMdown Extensions (>= 11.0.1)	`prodockit.steps` and `prodockit.tree` are built directly on the PyMdown Blocks API; `prodockit.pdf` also preserves the output of PyMdown features
`beautifulsoup4` (>= 4.12)	`prodockit.pdf`
`click` (>= 8.0)	the `prodockit` command-line tool
`pypdf` (>= 4.0)	`prodockit.pdf`
`tomli` (>= 2.0)	reading a template manifest on Python 3.10, where `tomllib` does not exist yet
`pymupdf` (>= 1.24)	only the back-of-book index - `pip install prodockit[index]`

The floors above are the ones declared in `pyproject.toml`, and a test keeps this table in step with them - the two had drifted apart, with `Markdown` recorded here as >= 3.4 long after the real floor moved to 3.10.3 (prodockit-extensions#372).

1.1.1 Not installed by pip

These are the ones `pip install prodockit` does **not** bring, and they differ in kind:

Requirement	Needed for
`weasyprint` (>= 69)	`prodockit.pdf`. A Python package, but not a dependency of prodockit - install it yourself. `prodockit.pdf` runs its command-line rather than importing it
`pandoc` (>= 3, builds pin 3.10.1)	`prodockit.pdf`, and `prodockit.bibliography` even without a PDF build. Genuinely not a Python package - there is nothing for `pip` to install
`mermaid-cli`, `mathjax-full` (Node >= 22)	only Mermaid diagrams and TeX maths in the PDF
Chrome or Chromium	only Mermaid diagrams - `mermaid-cli` renders them through a headless browser
A citation style (`.csl`)	only `prodockit.bibliography`. Fetched per build, not vendored - see below

The citation style is a download rather than an install. Pandoc resolves `harvard-cite-them-right.csl` from the directory it runs in, and every CI script here fetches it immediately before building:

```
curl -fsSL -o harvard-cite-them-right.csl "https://www.zotero.org/styles/harvard-cite-them-right"
```

It is deliberately **not** committed: it is third-party content with its own licence and its own release cadence, and a vendored copy would go stale silently while every build kept succeeding. `prodockit bootstrap` fetches it for you, and `.gitignore` keeps a local copy out of commits.

`weasyprint` is worth separating from `pandoc` rather than filing both as "external binaries": one is a `pip install` away and the other is not, and a reader who treats them alike goes looking for a package that does not exist, or misses one that does.

Pandoc is version-sensitive in a way that changes output rather than breaking the build: a major version below 3 renders code blocks as justified prose, and the builds pin an exact release because two 3.x versions have already disagreed about the same source. `prodockit bootstrap` installs the pinned version where a package manager allows it and tells you when your local pandoc differs - see Pinning build inputs.

See PDF generation for how `prodockit.pdf` locates these, and Known limitations for why the Node ones are needed at all. A build with neither Mermaid diagrams nor maths needs neither of them, and no browser.

1.2 From PyPI

```
pip install prodockit
```

For a minimal project that needs no PDF toolchain, continue with Build your first site. The external tools above can be added later when the document needs their features.

1.3 Enabling an extension

Each prodockit extension is registered as a standard Python-Markdown extension under the `markdown.extensions` entry point group, so it can be enabled by name, the same way you'd enable a built-in extension like `toc` or a `pymdownx` one:

```
import markdown

html = markdown.markdown(
    text,
    extensions=["prodockit.headings", "prodockit.refs",
"prodockit.tables"],
)
```

Or, for a Zensical project, in `zensical.toml` alongside the built-in and `pymdownx` extensions. Unlike `pymdownx`'s and Zensical's own namespaces, Zensical doesn't hoist a nested `prodockit.headings` table into that dotted extension name, so each one needs a quoted key instead:

```
[project.markdown_extensions."prodockit.headings"]
[project.markdown_extensions."prodockit.refs"]
[project.markdown_extensions."prodockit.citations"]
[project.markdown_extensions."prodockit.glossary"]
[project.markdown_extensions."prodockit.tables"]
[project.markdown_extensions."prodockit.tree"]
[project.markdown_extensions."prodockit.steps"]
[project.markdown_extensions."prodockit.bibliography"]
[project.markdown_extensions."prodockit.index"]
```

Enable only the ones you use - each is independent, and none of them requires another.

1.3.1 The nine extensions

See each extension's own page for its syntax, examples, and configuration:

Extension	What it adds
`prodockit.headings`	Numbered headings, and a number a cross-reference can point at
`prodockit.refs`	Cross-references that resolve to a number *and* a name
`prodockit.citations`	Citation handling
`prodockit.glossary`	Acronyms and a glossary
`prodockit.tables`	Column widths, dense tables, multi-row headers, merged cells, rotated headings
`prodockit.tree`	A directory listing that looks like one
`prodockit.steps`	Numbered steps a reader works through in order
`prodockit.bibliography`	A bibliography built from your `.bib` files
`prodockit.index`	A back-of-book index (PDF only)

1.3.2 What is *not* an extension

Several parts of prodockit have no `markdown.extensions` entry point and nothing to add to `zensical.toml`, because they are not Markdown syntax:

`prodockit pdf`	A separate PDF-generation build step
`prodockit source-bundle`	Packages documentation source as a separate PDF
`prodockit.zensical_macros`	A `define_env()` module for Zensical's macros plugin, named under its `modules` config rather than as an extension
`prodockit.testing`	pytest fixtures and checks for an already-built site and PDF
`prodockit bootstrap`	Sets up a machine to build the docs
`prodockit sync-repo`	Keeps repository metadata and README badges matching the git remote
`prodockit pins`	Moves build-input version pins together
`prodockit template-sync`	Brings a project back into step with the template it came from
`prodockit init-tools` / `init-mathjax`	Sets up optional Mermaid and maths rendering tools

Contributors changing the package itself should use the editable installation and repository checks in Development and code map.

Index