

x\_make\_markdown\_x Documentation Furnace

This generator keeps the labâ€™s written evidence as precise as the chemistry. Markdown catalogs, PDF exports, accessibility lintingâ”every document comes out of this furnace with the same mechanical accuracy.

# Mission Log

- Produce Markdown dossiers and registries through scriptable templates.
  - Funnel PDFs through `x_make_common_x.export_markdown_to_pdf` so every artifact carries `ExportResult` metadata and deterministic paths.
  - Validate accessibility and linting so the prose meets the same standard as the codebase.
  - Feed orchestrator summaries and Change Control decks with reproducible documents, never ad hoc text.

## Instrumentation

- Python 3.11 or newer.
  - Ruff, Black, MyPy, Pyright, pytest when you intend to run QA.
  - wkhtmltopdf or Pandoc installed depending on which exporter adapter you enable.

## **Operating Procedure**

1. python -m venv .venv
  2. \.venv\Scripts\Activate.ps1
  3. python -m pip install --upgrade pip
  4. pip install -r requirements.txt
  5. python x\_cls make\_markdown x.py --help

Use the CLI to enumerate recipes, generate Markdown payloads, and produce PDFs that the orchestrator catalogs automatically.

## Evidence Checks

| Check | Command | --- | --- | Formatting sweep | python -m black . | Lint interrogation | python -m ruff check . | Type audit |  
python -m mypy . | Static contract scan | python -m pyright | Functional verification | pytest |

# Reconstitution Drill

On the monthly rebuild I install wkhtmltopdf on a fresh machine, run this furnace, and verify Markdown and PDF artefacts align with orchestrator summaries. Binary versions and runtimes are logged; any drift feeds back into Change Control before the next release window.

## **Conduct Code**

Document every new template, exporter, or lint rule in Change Control. Run the QA slate before touching production. Words are evidence; treat them with the same rigor as lab results.

## **Sole Architect's Note**

I forged every generator, exporter hook, and lint circuit in this project. My experience as both automation engineer and documentation lead keeps narrative strategy synchronized with tooling.

## Legacy Staffing Estimate

- Without LLM assistance you would field: 1 documentation lead, 1 automation engineer, 1 build specialist for PDF tooling, and 1 technical editor.
  - Delivery horizon: 10–12 engineer-weeks to reach parity.
  - Cost band: USD 80k–105k plus ongoing exporter maintenance.

## Technical Footprint

- Language: Python 3.11+, templating strategies, filesystem orchestration.
  - Export Infrastructure: shared exporters from `x_make_common_x`, `wkhtmltopdf`, optional Pandoc adapters.
  - Tooling Discipline: Ruff, Black, MyPy, Pyright, pytest, Markdown lint rigs.
  - Integration Points: orchestrator hooks, diagram ingestion from Graphviz/Mermaid siblings, Change Control evidence capture.

