

md2pdf

CI Release License: MIT Python 3.11+

md2pdf converts Markdown files into A4 portrait PDF documents with a restrained layout close to Γ OCT P 7.32 conventions: serif font, 14 pt body text, 1.5 line spacing, 1.25 cm paragraph indent, simple tables, code blocks, Mermaid diagrams and centered page numbers.

The project is intentionally small. It is useful for course work, project reports, internal technical notes and other documents where predictable PDF output matters more than full Markdown compatibility.

Features

Feature	Behavior	Notes
Text	Serif font, 14 pt, readable left-aligned paragraphs	First-line indent is 1.25 cm
Headings	Bold serif headings	The first H1 is centered as a title
Lists	Dash marker with a fixed left indent	Supports flat - lists
Tables	Plain grid with repeated header rows	Best for simple pipe tables
Code blocks	Monospace-style framed block using the document font	Long lines are preserved
Mermaid	Renders diagrams to PNG before inserting into PDF	Built-in renderer covers common diagram types
Pages	A4 portrait with 30/10/20/20 mm margins	Page number is centered in the footer

Requirements

- Python 3.11 or newer.
- Node.js 20 or newer if you want Mermaid CLI rendering.
- A compatible serif font:
 - Times New Roman on macOS;
 - Times New Roman in the user font folder on macOS;
 - Microsoft Core Fonts or Liberation Serif on Linux;

- DejaVu Serif as the last Linux option when narrower serif fonts are not installed.

The converter can still render supported Mermaid diagrams when Mermaid CLI is unavailable or when its browser backend fails. Set `MD2PDF_STRICT_MERMAID=1` if you want Mermaid CLI failures to stop the build.

Installation

For regular use from a checkout:

```
python3 -m venv .venv
. .venv/bin/activate
python -m pip install --upgrade pip
python -m pip install -e .
```

Install Mermaid CLI support:

```
npm install
```

For development, install the extra tools:

```
python -m pip install -e "[dev]"
```

Check local dependencies:

```
md2pdf --check-deps
```

Usage

Build the repository README:

```
PYTHONPATH=src python3 -m md2pdf README.md README.pdf
```

Build the example document:

```
PYTHONPATH=src python3 -m md2pdf examples/example.md examples/example.pdf
```

After editable installation, the console entry point is available:

```
md2pdf README.md README.pdf
```

The input and output arguments are optional. Without arguments, the tool reads `README.md` and writes `README.pdf` in the current directory.

Useful options:

```
md2pdf --version
md2pdf --check-deps
md2pdf input.md output.pdf --strict-mermaid
md2pdf input.md output.pdf --temp-dir .md2pdf/custom
md2pdf input.md output.pdf --font-size 13 --line-height 1.45
md2pdf input.md output.pdf --margins 30,10,20,20
md2pdf input.md output.pdf --quiet
```

Supported Markdown

The parser supports the practical subset used by this project:

- headings from # to #####;
- paragraphs separated by blank lines;
- flat unordered lists written as - item;
- simple pipe tables;
- fenced code blocks;
- fenced Mermaid blocks.

Complex Markdown extensions are not implemented: nested lists, footnotes, raw HTML blocks, escaped table pipes and advanced link syntax should be simplified before conversion.

Mermaid

Mermaid blocks are rendered in this order:

1. Mermaid CLI through the local node_modules/.bin/mmdc or a global mmdc.
2. Built-in renderers for supported diagram families only when Mermaid CLI is unavailable or fails outside strict mode.
3. Plain code block output when the diagram type is unsupported and no renderer succeeds.

Supported built-in families:

- flowchart;
- sequenceDiagram;
- stateDiagram;

– xychart-beta.

Development

More details are available in docs/development.md. Usage examples are collected in docs/usage.md.

Run the local quality gate:

```
python -m ruff check .
python -m pytest -q
npm run pdf
npm run example
```

The npm shortcut runs the same checks:

```
npm run check
```

Project files:

Path	Purpose
src/md2pdf/config.py	Shared defaults, version, fonts and runtime flags
src/md2pdf/markdown.py	Markdown cleanup, tables and block helpers
src/md2pdf/mermaid.py	Mermaid CLI integration and built-in renderers
src/md2pdf/pdf.py	ReportLab PDF generation
src/md2pdf/core.py	Compatibility facade for older imports
src/md2pdf/cli.py	CLI argument parsing
src/md2pdf/main.py	python -m md2pdf entry point
tests/test_md2pdf.py	Unit and smoke tests
examples/example.md	Example input document
examples/example.pdf	Generated example output
README.pdf	Generated README output
CONTRIBUTING.md	Contributor guide
SECURITY.md	Security policy
CHANGELOG.md	Release notes
docs/usage.md	CLI usage notes
docs/development.md	Development workflow
docs/release.md	Release workflow

PDF artifacts are committed intentionally. Update README.pdf and examples/example.pdf only when README, examples or rendering behavior

changes.

Troubleshooting

Font not found

Install Times New Roman, Microsoft Core Fonts or Liberation Serif. DejaVu Serif is accepted as a last Linux option. The tool checks common macOS and Linux font paths.

Mermaid CLI fails to launch a browser

This can happen in sandboxed environments or CI runners. By default, the converter uses built-in renderers for supported diagram types. Use `--strict-mermaid` to fail fast instead.

Diagram layout differs from Mermaid CLI

Built-in renderers are intentionally simple black-and-white renderers. They keep the PDF build usable, but they are not a complete Mermaid implementation.

Markdown output is missing advanced formatting

The converter is not a full Markdown engine. Simplify unsupported Markdown features or add a focused parser feature with tests.

License

MIT. See LICENSE.