MARDAS MD2PDF
MARKDOWN TO PDF ENGINE

راهنمای کامل

راهنمای Mardas MD2PDF

راهنمای کامل استفاده و مرجع امکانات

این فایل راهنمای کامل نصب، پیکربندی و استفاده از Mardas MD2PDF است.
همین سند به عنوان نمونه زنده رندر نیز استفاده می‌شود و جلد، فهرست مطالب، متن ترکیبی فارسی/English، فرمول، کد، نمودار Mermaid، تصویر، جدول، پانویس، شکست صفحه و HTML امن را نمایش می‌دهد.

نویسندگان
تیم Mardas MD2PDF (مستندسازی)
Meraj Rastegar (mragetsars@gmail.com)

تاریخ
۱۴۰۵-۰۲-۳۰

مؤسسه
Mardas Lab

درس / دوره
انتشار حرفه‌ای Markdown

نسخه
1.6.2

وضعیت
Stable

کلیدواژه‌ها
Markdown
PDF
فارسی/English
RTL/LTR
MathJax

فهرست مطالب

1 معرفی
- 1-1 چک‌لیست نمونه‌های رندر
- 1-2 قابلیت‌های اصلی

2 نصب
- 2-1 پیش‌نیازها
- 2-2 نصب از سورس
- 2-3 نصب برای توسعه

3 اولین خروجی PDF

4 روند پیشنهادی کار

5 درباره Front Matter
- 5-1 فیلدهای رایج
- 5-2 رفتار جلد

6 زبان و جهت سند
- 6-1 اولویت تعیین جهت
- 6-2 نمونه متن ترکیبی

7 فهرست مطالب

8 مرجع امکانات Markdown
- 8-1 پاراگراف و تاکید
- 8-2 لیست‌ها
- 8-3 فهرست وظایف
- 8-4 جدول
- 8-5 نقل‌قول
- 8-6 انواع Callout

9 فرمول‌های MathJax
- 9-1 رفع اشکال فرمول‌ها

10 بلوک‌های کد
- 10-1 بلوک‌های کد پیشرفته

11 نمودارهای Mermaid

12 تصویر و HTML امن

12-1 قواعد استفاده از تصویر
12-2 کد HTML امن
13 امنیت و ورودی‌های قابل اعتماد
13-1 ناوبری PDF و metadata
14 صفحه‌بندی و Layout
14-1 شکست صفحه دستی
14-2 اندازه صفحه
14-3 انواع Margin
15 اعمال Watermark
16 سیستم Appearance
17 روند کار با GUI
18 مرجع CLI
19 اتوماسیون و CI
20 رفع اشکال
20-1 پیدا نشدن Chromium
20-2 متن فارسی ظاهر مناسبی ندارد
20-3 تصویرها دیده نمی‌شوند
20-4 فرمول‌ها به شکل TeX خام دیده می‌شوند
20-5 نیاز به بررسی Layout
21 پانویس
22 چک‌لیست نهایی انتشار

معرفی

برنامه Mardas MD2PDF ابزاری برای تبدیل Markdown به PDF است که مخصوص سندهای فارسی، انگلیسی و ترکیبی طراحی شده است. هدف پروژه این است که نویسنده بتواند متن را در قالب ساده Markdown بنویسد، اما خروجی نهایی شبیه یک سند PDF مرتب، حرفه‌ای و قابل انتشار باشد.

این پروژه برای گزارش‌های دانشگاهی، مستندات فنی، جزوه‌های آموزشی، راهنماهای نرم‌افزاری، پیش‌نویس‌های پژوهشی، گزارش پروژه و هر سند Markdown که نیاز به خروجی PDF تمیز دارد مناسب است.

```text
Markdown -> HTML ساختاریافته -> PDF با Chromium
```

در این پروژه متن‌ها مستقیماً روی canvas فایل PDF رسم نمی‌شوند. ابتدا Markdown به HTML ساختاریافته تبدیل می‌شود، سپس CSS چاپی و تنظیمات appearance اعمال می‌شود، فرمول‌ها با MathJax رندر می‌شوند و در مرحله آخر Chromium خروجی PDF را تولید می‌کند. این روش باعث پشتیبانی بهتر از layout چاپی، متن‌های ترکیبی RTL/LTR، فرمول‌های SVG، کدهای هایلایت‌شده، تصویرهای محلی و جدول‌های پیچیده می‌شود.

نکته

این راهنما هم مستند استفاده است و هم نمونه رندر. نسخه PDF همین فایل در پوشه examples/ قرار دارد تا کاربر بتواند خروجی واقعی هر قابلیت مهم را بررسی کند.

چک‌لیست نمونه‌های رندر

این راهنما عمداً چند نمونه کوچک اما مهم دارد تا کنار راهنمای برنامه، مثل test case تصویری هم عمل کند. وقتی PDF خروجی را بررسی می‌کنید، این موارد را ببینید:

چیزی که باید در PDF بررسی شود	بخش نمونه
عنوان، زیرعنوان، نویسنده‌ها، خلاصه، نسخه، وضعیت، کلیدواژه و labelهای وابسته به زبان.	جلد و metadata
شماره‌گذاری تودرتو، لینک‌های داخلی و bookmarkهای PDF viewer که از headingهای Markdown ساخته می‌شوند.	فهرست مطالب و outline
متن فارسی/English، inline code و شناسه‌ها در یک پاراگراف خوانا بمانند.	متن ترکیبی
فرمول درون‌خطی با متن هماهنگ باشد و فرمول نمایشی وسط‌چین و خوش‌اندازه دیده شود.	فرمول MathJax
بدون زبان بدون خراب شدن محتوا رندر شوند code block و fenced، indented.	بلوک کد
بلوک کد از نوع `flowchart` به جای نمایش کد خام، به نمودار SVG تبدیل شود.	نمودار Mermaid
تصویر Markdown و تگ امن HTML در PDF دیده شوند.	تصویر و HTML
پانویس چندخطی، شکست صفحه دستی، marginها و شماره صفحه پایدار باشند.	پانویس و صفحه‌بندی

قابلیت‌های اصلی

توضیح	قابلیت
پشتیبانی از `lang: fa`، `lang: en`، جهت RTL/LTR و متن ترکیبی.	سندهای فارسی و انگلیسی
عنوان، زیرعنوان، نویسنده‌ها، خلاصه، لوگو، تاریخ، نسخه، وضعیت، کلیدواژه و metadata آموزشی.	جلد حرفه‌ای
ساخت فهرست چندسطحی و bookmarkهای PDF viewer از headingهای Markdown.	فهرست مطالب و outline
رندر فرمول‌های درون‌خطی و نمایشی.	فرمول MathJax
هایلایت کدهای fenced و indented با Pygments.	بلوک کد
رندر آفلاین نمودارهای کاربردی `flowchart` / `graph` به صورت SVG.	نمودار Mermaid
جاسازی تصویرهای Markdown و HTML امن به صورت data URI.	تصویر محلی
پاک‌سازی HTML خام به صورت پیش‌فرض.	کد HTML امن
پشتیبانی از پانویس‌های چندخطی با محتوای Markdown.	پانویس
انتخاب جداگانه `style`، `palette` و `mode` برای شکل سند، رنگ‌بندی و خروجی روشن/تاریک.	سیستم Appearance
رابط CLI مناسب برای اسکریپت‌ها و CI.	اتوماسیون
رابط گرافیکی محلی برای ویرایش، پیش‌نمایش تقریبی، تنظیم گزینه‌ها و خروجی گرفتن.	رابط گرافیکی GUI

نصب

پیش‌نیازها

برای استفاده از پروژه بهتر است این موارد آماده باشند:

- نصب بودن Python نسخه 3.10 یا جدیدتر؛
- محیط مجازی Python؛
- نصب بودن Chromium مربوط به Playwright؛
- فونت مناسب فارسی، ترجیحاً Vazirmatn؛

- نصب بودن Git برای clone کردن پروژه.

نصب از سورس

```bash
git clone https://github.com/mragetsars/Mardas-MD2PDF.git
cd Mardas-MD2PDF
python -m venv .venv
source .venv/bin/activate
pip install -e .
python -m playwright install chromium
```

در Windows PowerShell:

```powershell
python -m venv .venv
.venv\Scripts\Activate.ps1
pip install -e .
python -m playwright install chromium
```

نصب برای توسعه

برای توسعه و اجرای تست‌ها:

```bash
pip install -e .[dev]
pytest
```

بعد از نصب، دو دستور اصلی در دسترس است:

کاربرد	دستور
تبدیل Markdown به PDF از خط فرمان.	`mrs-md2pdf`
اجرای رابط گرافیکی محلی.	`mrs-md2pdf-gui`

اولین خروجی PDF

تبدیل ساده:

```bash
mrs-md2pdf input.md -o output.pdf
```

تبدیل همراه با فهرست مطالب و ظاهر GitHub-style:

```bash
mrs-md2pdf input.md -o output.pdf --toc --style github --palette blue --mode light
```

تولید خروجی طولانی با رفتار شبیه کتاب:

```bash
mrs-md2pdf input.md -o output.pdf \
  --toc \
  --toc-depth 4 \
  --toc-page-break \
  --h1-page-break \
  --style textbook \
  --palette slate \
  --mode light
```

ذخیره HTML میانی برای بررسی و رفع اشکال:

```bash
mrs-md2pdf input.md -o output.pdf --debug-html output.html
```

نمایش صریح progress bar در ترمینال:

```bash
mrs-md2pdf input.md -o output.pdf --progress on
```

حالت پیش‌فرض `--progress auto` فقط وقتی progress bar را نشان می‌دهد که دستور در ترمینال تعاملی اجرا شود. برای اسکریپت‌های ساکت‌تر می‌توانید از `--progress off` استفاده کنید.

روند پیشنهادی کار

برای رسیدن به خروجی تمیز، این روند پیشنهاد می‌شود:

1. محتوای سند را در Markdown بنویسید.
2. در ابتدای فایل، front matter شامل عنوان، نویسنده، زبان، جهت و metadata جلد را اضافه کنید.
3. یک خروجی اولیه با `--toc` و appearance مناسب بگیرید.
4. جلد، فهرست مطالب، صفحه‌های دارای فرمول، صفحه‌های دارای کد، تصویرها و شماره صفحه را بررسی کنید.
5. اگر layout نیاز به بررسی داشت، خروجی `--debug-html` بگیرید و HTML تولیدشده را در مرورگر ببینید.
6. اصلاحات نهایی را در Markdown انجام دهید و PDF را دوباره بسازید.

برای سندهای مهم، بررسی انسانی خروجی نهایی ضروری است. تست خودکار بسیاری از خطاها را می‌گیرد، اما تایپوگرافی، شکست صفحه و فاصله‌ها باید دیده شوند.

درباره Front Matter

بخش Front matter یک بخش YAML اختیاری در ابتدای فایل Markdown است. این بخش جلد، metadata فایل PDF، زبان سند، جهت سند و چند فیلد مخصوص گزارش‌های رسمی یا آموزشی را کنترل می‌کند.

```yaml
---
title: "گزارش فنی من"
subtitle: "ساخته‌شده با Markdown یک سند PDF"
authors:
  - name: "Mardas"
    email: "mragetsars@gmail.com"
    affiliation: "Mardas Lab"
    role: "Author"
summary: |
  این متن روی جلد نمایش داده می‌شود.
  متن چندخطی پشتیبانی می‌شود.
institution: "نام دانشگاه یا سازمان"
department: "نام دانشکده یا دپارتمان"
course: "نام درس یا پروژه"
supervisor: "نام استاد یا راهنما"
date: "۱۴۰۵-۰۲-۳۰"
version: "1.6.2"
status: "Draft"
keywords: [Markdown, PDF, RTL, MathJax]
cover_label: "گزارش فنی"
cover_logo: "src/assets/Mardas.png"
lang: fa
dir: rtl
---
```

فیلدهای رایج

کاربرد	فیلد
عنوان جلد و title در metadata فایل PDF.	`title`
متن اختیاری زیر عنوان اصلی.	`subtitle`
نویسنده ساده و تک‌مقداری.	`author`
فهرست نویسنده‌ها با `name`، `email`، `affiliation` و `role`.	`authors`
خلاصه روی جلد و subject در metadata.	`summary` / `description`
اطلاعات دانشگاهی یا سازمانی.	`institution`, `department`, `course`
فیلدهای اختیاری برای گزارش‌های آموزشی.	`supervisor`, `group`, `student_id`

کاربرد	فیلد
کارت‌های وضعیت سند روی جلد.	`date`, `version`, `status`
کلیدواژه‌های جلد و metadata فایل PDF.	`keywords` / `tags`
برچسب کوچک بالای عنوان جلد.	`cover_label`
مسیر لوگوی سفارشی نسبت به فایل Markdown.	`cover_logo` / `logo`
زبان داخلی رابط سند، معمولاً `fa` یا `en`.	`lang`
جهت پوسته سند: `auto`، `ltr` یا `rtl`.	`dir`

رفتار جلد

جلد جدا از محتوای اصلی رندر می‌شود. نتیجه این تصمیم چنین است:

- جلد جزو شماره صفحات محتوایی حساب نمی‌شود؛
- شماره‌گذاری footer از صفحه بعد از جلد شروع می‌شود؛
- طرح watermark فقط روی صفحات محتوایی اعمال می‌شود؛

خروجی می‌تواند برای جلد پس‌زمینه تمام‌صفحه داشته باشد؛ style

- در صورت نیاز، جلد کاملاً قابل حذف است.

حذف جلد:

```bash
mrs-md2pdf input.md -o output.pdf --no-cover
```

استفاده از لوگوی سفارشی:

```bash
mrs-md2pdf input.md -o output.pdf --cover-logo ./src/assets/Mardas.png
```

حفظ جلد بدون نمایش لوگو:

```bash
mrs-md2pdf input.md -o output.pdf --no-cover-logo
```

زبان و جهت سند

کنترل جهت یکی از مهم‌ترین بخش‌های تولید PDF فارسی/انگلیسی است. در Mardas MD2PDF زبان سند و جهت سند از هم جدا در نظر گرفته شده‌اند.

`lang: fa` باعث ایجاد پوسته فارسی/RTL، لیبل‌های فارسی جلد، عنوان فارسی calloutها و عنوان `فهرست مطالب` می‌شود.

`lang: en` باعث ایجاد پوسته انگلیسی/LTR، لیبل‌های انگلیسی جلد، عنوان انگلیسی calloutها و عنوان `Table of Contents` می‌شود.

اولویت تعیین جهت

جهت سند با این ترتیب مشخص می‌شود:

1. گزینه خط فرمان `--dir rtl|ltr|auto`
2. فیلدهای front matter مثل `dir`، `direction`، `text_direction` یا `document_direction`
3. پیش‌فرض به‌دست‌آمده از `lang`
4. تشخیص خودکار از متن Markdown

نمونه متن ترکیبی

در متن انگلیسی می‌توان واژه‌های فارسی مثل راست به چپ، فونت فارسی و گزارش فنی را آورد بدون اینکه جمله انگلیسی به هم بریزد.

در متن فارسی نیز می‌توان شناسه‌های English مثل `Playwright`، `MathJax`، `GitHub Actions`، `PDF` و `RTL/LTR` را داخل همان پاراگراف استفاده کرد.

همچنین Inline code هم پایدار می‌ماند: `mrs-md2pdf input.md -o output.pdf --toc`.

فهرست مطالب

فعال کردن فهرست مطالب:

```bash
mrs-md2pdf input.md -o output.pdf --toc
```

تعیین عمق فهرست:

```bash
mrs-md2pdf input.md -o output.pdf --toc --toc-depth 3
```

شروع متن اصلی از صفحه جدید بعد از فهرست:

```bash
mrs-md2pdf input.md -o output.pdf --toc --toc-page-break
```

شروع هر heading سطح اول از صفحه جدید:

```bash
mrs-md2pdf input.md -o output.pdf --h1-page-break
```

فهرست مطالب از headingهای Markdown ساخته می‌شود و فرمول‌های درون‌خطی داخل headingها مثل $E = mc^2$ و ϵ را خوانا نگه می‌دارد.

مرجع امکانات Markdown

پاراگراف و تاکید

پاراگراف‌های Markdown، **متن bold**، *متن italic*، `inline code`، لینک، لیست مرتب، لیست نامرتب، task list و نقل‌قول پشتیبانی می‌شوند.

لیست‌ها

- نوشتن محتوای اصلی با Markdown.
- افزودن front matter برای metadata.
- انتخاب appearance مناسب.
- بررسی خروجی نهایی PDF.

1. نصب پکیج.
2. نصب Chromium.
3. اجرای CLI.
4. بررسی خروجی.

فهرست وظایف

- [x] ورودی Markdown
- [x] تایپوگرافی فارسی/English
- [x] رندر MathJax
- [x] هایلایت کد
- [ ] بازبینی انسانی نهایی

جدول

توضیح	وضعیت	قابلیت
پاراگراف، heading، لیست و سلول جدول با direction-aware styling رندر می‌شوند.	بله	متن ترکیبی RTL/LTR
تصویرهای Markdown و HTML امن در صورت امکان embed می‌شوند.	بله	تصویر محلی
فرمول درون‌خطی و نمایشی اندازه‌گذاری جدا دارند.	بله	فرمول MathJax
برای fenced و indented code block از Pygments استفاده می‌شود.	بله	هایلایت کد
تبدیل می‌شوند SVG های کاربردی `flowchart` و `graph` به نمودارfence.	بله	نمودار Mermaid
پانویس چندخطی با Markdown داخلی پشتیبانی می‌شود.	بله	پانویس
های خطرناک حذف می‌شوندevent handler ها وtag.	بله	کد HTML خام امن

نقل‌قول

کیفیت خروجی PDF فقط تبدیل متن نیست. تایپوگرافی، line height، کنتراست، صفحه‌بندی، تصویرها، فرمول‌ها و قابل پیش‌بینی بودن رندر اهمیت دارند.

انواع Callout

انواع Calloutها از markerهای مشابه GitHub استفاده می‌کنند و عنوان آن‌ها با توجه به زبان سند ترجمه می‌شود.

نکته

برای نکته‌های توضیحی که باید از متن اصلی جدا دیده شوند، از callout استفاده کنید.

پیشنهاد

وقتی لازم است HTML دقیق ارسال‌شده به Chromium را ببینید، از --debug-html output.html استفاده کنید.

هشدار

گزینه --unsafe-html فقط برای فایل‌های محلی قابل اعتماد مناسب است، چون sanitizer داخلی را غیرفعال می‌کند.

فرمول‌های MathJax

فرمول‌های درون‌خطی باید با ارتفاع خط اطراف هماهنگ باشند: $E = mc^2$ ، $T = 500$ و $\Sigma = I \cdot \epsilon$ باید طبیعی وسط جمله قرار بگیرند.

فرمول نمایشی فضای بیشتری می‌گیرد:

$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}$$

نمونه ماتریسی:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \qquad \det(A) = -2$$

نمونه aligned:

$$\begin{aligned} \text{precision} &= \frac{TP}{TP + FP} \\ \text{recall} &= \frac{TP}{TP + FN} \end{aligned}$$

رفع اشکال فرمول‌ها

اگر فرمول‌ها به شکل TeX خام دیده شدند:

- مطمئن شوید `--no-mathjax` استفاده نشده باشد؛
- هشدارهای ترمینال را بررسی کنید؛
- برای سندهای خیلی بزرگ مقدار timeout مرورگر را افزایش دهید؛
- یک فایل کوچک‌تر بسازید تا فرمول مشکل‌دار را جدا کنید.

بلوک‌های کد

بلوک‌های کد fenced از برجسته‌سازی نحوی و برچسب زبان پشتیبانی می‌کنند.

```python
from dataclasses import dataclass

@dataclass
class Document:
    title: str
    lang: str = "fa"


def render_message(doc: Document) -> str:
    return f"Rendering {doc.title} as PDF"

print(render_message(Document("راهنمای Mardas")))
```

```javascript
const items = ["Markdown", "Persian", "English", "MathJax", "PDF"];
const message = items.map((item, index) => `${index + 1}. ${item}`).join("\n");
console.log(message);
```

بلوک کد fenced بدون زبان هم معتبر است:

```text
این بلوک زبان مشخصی ندارد.
با این حال باید بدون خطا رندر شود.
```

همچنین indented code block نیز پشتیبانی می‌شود:

```text
SELECT title, lang, version
FROM documents
WHERE renderer = 'mardas-md2pdf';
```

توجه کنید که inline code در برابر پردازش فرمول و پانویس محافظت می‌شود. یعنی `$x$` و `[^note]` وقتی داخل backtick باشند، به شکل literal باقی می‌مانند.

بلوک‌های کد پیشرفته

برای آموزش، بازبینی کد یا مستندات فنی، بلوک‌های کد می‌توانند عنوان فایل، شماره خط و خطوط برجسته‌شده داشته باشند.

```md
```python title="renderer.py" {2,5-6} linenos
def convert(markdown: str) -> bytes:
    html = render_markdown(markdown)
    pdf = render_pdf(html)
    return pdf
```
```

نمونه بالا در PDF به شکل یک بلوک کد با caption سفارشی، شماره خط و highlight رندر می‌شود:

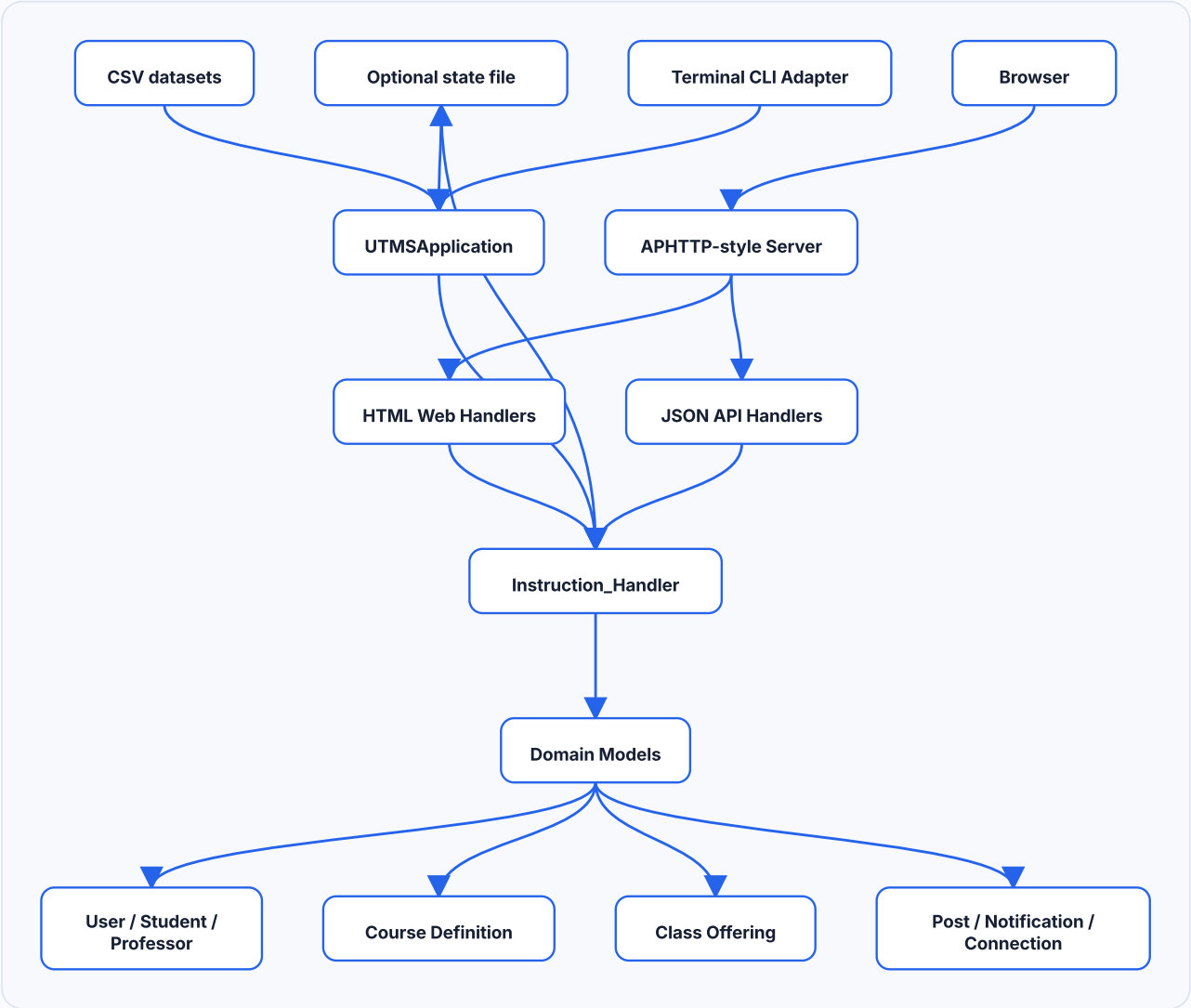
```python
def convert(markdown: str) -> bytes:
    html = render_markdown(markdown)
    pdf = render_pdf(html)
    return pdf
```
renderer.py

نمودارهای Mermaid

بلوک‌های Mermaid برای flowchart به صورت SVG در HTML میانی رندر می‌شوند و بعد داخل PDF قرار می‌گیرند. تمرکز renderer فعلاً روی زیرمجموعه رایج در مستندات پروژه است: `flowchart` / `graph`، جهت‌های `TD`، `TB`، `BT`، `LR`، `RL`، nodeهای مستطیلی، گرد، دایره‌ای و لوزی، edgeهای ساده، dotted، ضخیم و edgeهای دارای label.

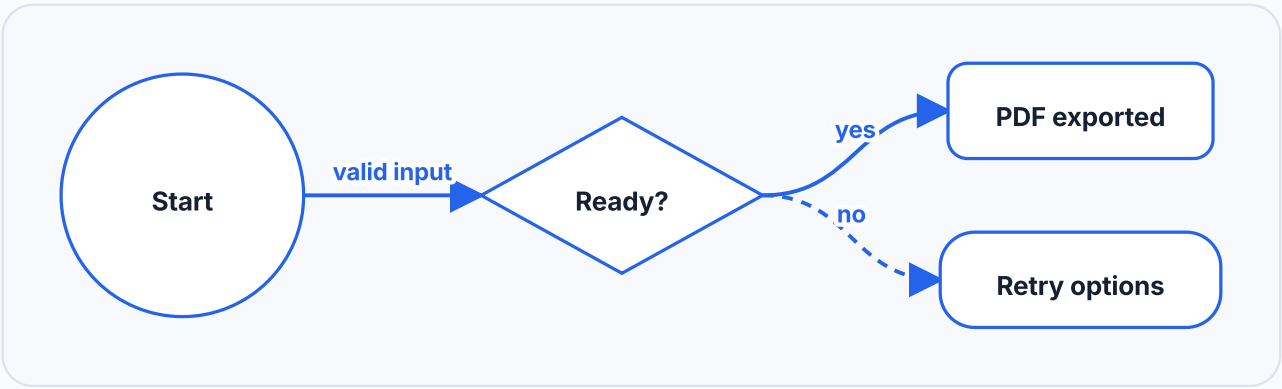
نکته مهم در خروجی PDF این است که بلوک زیر باید به شکل نمودار دیده شود، نه کد خام:

MERMAID



این نمونه کوچک‌تر برای بررسی label روی edgeها و شکل nodeها مفید است:

MERMAID



اگر یک بلوک Mermaid از syntaxهای پیشرفته خارج از این زیرمجموعه استفاده کند، بهتر است تا زمان اضافه‌شدن آن قابلیت، یک تصویر fallback کوچک از همان نمودار کنار Markdown نگه داشته شود. برای معماری‌های معمول، data flow و نمودارهای آموزشی، renderer داخلی کافی است و به اینترنت نیاز ندارد.

تصویر و HTML امن

مسیر تصویرها نسبت به فایل Markdown resolve می‌شود. تصویرهای محلی اگر حجم مناسبی داشته باشند داخل HTML/PDF نهایی embed می‌شوند.

وقتی اندازه‌دهی دقیق‌تر لازم است، می‌توان از HTML امن استفاده کرد:

قواعد استفاده از تصویر

- برای خروجی پایدار PDF، تصویرهای محلی بهتر از تصویرهای remote هستند.
- قبل از embed کردن، حجم تصویرها را منطقی نگه دارید.
- مسیر تصویر را نسبت به محل فایل Markdown بنویسید.
- مسیر تصویر را داخل پوشه همان سند نگه دارید. مسیرهای URL، absolute‌های `file:`، خروج از پوشه با الگوهایی مثل `../secret.png` و fallback به current working directory به صورت پیش‌فرض block می‌شوند.

- اگر یک تصویر محلی به شکل امن embed نشود، با یک placeholder شفاف جایگزین می‌شود تا Chromium آن را از طریق `<base>` سند load نکند.
- در GUI، فایل‌ها یا پوشه تصویر را قبل از export attach کنید.
- تصویرهای خیلی بزرگ block می‌شوند و renderer برای جلوگیری از مصرف زیاد حافظه هشدار می‌دهد.

کد HTML امن

کد HTML خام به صورت پیش‌فرض sanitize می‌شود. sanitizer عناصر مناسب سند مثل `<div>`، `<span>`، `<table>`، `<figure>` و `<img>` را نگه می‌دارد، اما محتوای فعال یا خطرناک مثل iframe، event handler، script، stylesheet، form خارجی، URLهای `file:`، URL scheme ناامن و `data:` imageهای غیر-raster را حذف می‌کند.

مجاز است AVIF و BMP، WebP، GIF، JPEG، PNG رایج مثل raster هایformat امن فقط برای image `data:` می‌شود؛ برای نمودارها بهتر است از فایل محلی قابل اعتماد یا بلوک block امن HTML در SVG از نوع data URL استفاده کنید Mermaid.

غیرفعال کردن sanitizer فقط برای فایل‌های قابل اعتماد:

```bash
mrs-md2pdf input.md -o output.pdf --unsafe-html
```

امنیت و ورودی‌های قابل اعتماد

Mardas MD2PDF یک ابزار انتشار محلی است. فایل Markdown، assetهای attach شده در GUI، لوگوی جلد، watermark و HTML خام را محتوای نویسنده و قابل اعتماد در نظر بگیرید؛ مگر اینکه renderer جداشده را داخل محیط اجرا کنید.

رندر پیش‌فرض یک مرز محافظه‌کارانه برای فایل‌ها نگه می‌دارد: تصویرهای محلی نسبت به محل فایل Markdown resolve می‌شوند، اگر امن باشند به `data:` URL تبدیل می‌شوند و اگر به بیرون از پوشه سند اشاره کنند یا قابل embed نباشند block می‌شوند. HTML خام هم به صورت پیش‌فرض sanitize می‌شود، مگر اینکه `--unsafe-html` را فعال کنید.

حالت sandbox کرومیوم با `--chromium-sandbox` کنترل می‌شود:

رفتار	حالت
برای کاربر عادی sandbox را روشن نگه می‌دارد و فقط هنگام اجرای root آن را خاموش می‌کند.	`auto`
همیشه sandbox کرومیوم را درخواست می‌کند.	`on`
گزینه `--no-sandbox` را به Chromium می‌دهد؛ فقط در containerهای قابل اعتماد یا CI جداشده استفاده شود.	`off`

برای سندهای نامطمئن، خروجی را داخل container یا محیط disposable بسازید و `--unsafe-html` را خاموش نگه دارید. جزئیات کامل در `SECURITY.md` آمده است.

ناوبری PDF و metadata

outline یک Markdown هایheading می‌گیرند و از front matter استاندارد سند را از metadata های تولیدشدهPDF به PDF کنار outline می‌سازند. فهرست مطالب چاپی همچنان داخل متن سند دیده می‌شود، اما PDF viewer برای خواننده کمک می‌کند سریع بین بخش‌ها جابه‌جا شود.

وقتی فهرست مطالب چاپی هم لازم دارید از `--toc` استفاده کنید. outline از همان مجموعه heading ساخته می‌شود تا فهرست چاپی و bookmarkهای PDF با هم هماهنگ بمانند.

صفحه‌بندی و Layout

شکست صفحه دستی

برای شکست صفحه دستی می‌توانید HTML امن زیر را قرار دهید:

```html
<div class="md2pdf-page-break"></div>
```

اندازه صفحه

استفاده از اندازه نام‌دار:

```bash
mrs-md2pdf input.md -o output.pdf --page-size A4
```

استفاده از حالت landscape:

```bash
mrs-md2pdf input.md -o output.pdf --page-size "A4 landscape"
```

استفاده از ابعاد دقیق:

```bash
mrs-md2pdf input.md -o output.pdf --page-size "210mm 297mm"
```

انواع Margin

```bash
mrs-md2pdf input.md -o output.pdf \
  --margin-top 18mm \
  --margin-bottom 18mm \
  --margin-x 16mm
```

اعمال Watermark

اعمال Watermark متنی:

```bash
mrs-md2pdf input.md -o output.pdf --watermark "DRAFT"
```

اعمال Watermark تصویری:

```bash
mrs-md2pdf input.md -o output.pdf \
  --watermark-image ./src/assets/Mardas.png \
  --watermark-opacity 0.05 \
  --watermark-width 95mm
```

طرح Watermark فقط روی صفحات محتوایی اعمال می‌شود و روی جلد قرار نمی‌گیرد.

سیستم Appearance

برنامه Mardas MD2PDF به جای چند سیستم موازی، یک مدل ظاهر دارد: `style` شکل و layout سند را کنترل می‌کند، `palette` رنگ‌های accent را تعیین می‌کند و `mode` خروجی روشن یا تاریک را انتخاب می‌کند.

Style	کاربرد پیشنهادی
`modern`	مستندات عمومی، proposal، گزارش نرم‌افزاری و راهنمای محصول.
`github`	مستندات پروژه‌ای شبیه README و خروجی نزدیک به GitHub-style Markdown.
`textbook`	جزوه‌های آموزشی طولانی و محتوای آموزشی فارسی/English.
`academic`	گزارش رسمی، سند دانشگاهی، پیش‌نویس پایان‌نامه و مقاله ساختاریافته.

Palette	حس رنگی
`blue`	آبی حرفه‌ای پیش‌فرض.
`emerald`	سبز آرام برای گزارش‌ها و داشبوردها.
`violet`	بنفش خلاقانه برای سندهای محصولی.
`amber`	رنگ گرم برای محتوای آموزشی و review.
`rose`	رنگ برجسته برای گزارش‌های editorial.
`slate`	خنثی و رسمی برای مستندات فنی.
`neutral`	خاکستری مینیمال برای خروجی رسمی.

انتخاب appearance از CLI:

```bash
mrs-md2pdf input.md -o output.pdf --style github --palette blue --mode light
mrs-md2pdf input.md -o output.pdf --style academic --palette emerald --mode dark
```

یا ذخیره در front matter:

```yaml
appearance:
  style: modern
  palette: blue
  mode: light
```

برای دیدن گزینه‌ها از `--list-styles`، `--list-palettes` و `--list-modes` استفاده کنید.

حالت تاریک برای هر style سطح رنگی مخصوص خودش را دارد. `modern` از پس‌زمینه سرمه‌ای عمیق استفاده می‌کند، `github` به سطح تاریک شبیه GitHub نزدیک است، `textbook` ظاهر تقریباً سیاه و هماهنگ با جلد قدیمی را نگه می‌دارد و `academic` پس‌زمینه ذغالی گرم دارد. Palette همچنان رنگ accent را در حالت روشن و تاریک، از جمله تزئینات جلد و calloutها، کنترل می‌کند.

روند کار با GUI

اجرای GUI:

```bash
mrs-md2pdf-gui
```

رابط کاربری GUI برای کاربرانی مناسب است که روند بصری را ترجیح می‌دهند:

1. نوشتن یا paste کردن Markdown.
2. انتخاب mode، palette، style، زبان، جهت، اندازه صفحه و گزینه‌های خروجی.
3. امکان attach کردن فایل‌ها یا پوشه‌های تصویر محلی.
4. دیدن پیش‌نمایش تقریبی.
5. در آخر export گرفتن از PDF نهایی.
6. استفاده از **Ctrl/Cmd+S** برای ذخیره Markdown و **Ctrl/Cmd+Enter** برای export سریع PDF.

Studio فعلی پیش‌نویس، layout، جهت، حالت روشن/تاریک، preview، عرض editor و گزینه‌های export را در local storage ذخیره می‌کند. این کار باعث می‌شود ناخواسته صفحه در جلسه‌های ویرایش طولانی کمتر refresh مرورگر نگه می‌دارد. برای پاک کردن پیش‌نویس محلی و برگشتن به حالت تمیز از **Reset State** استفاده کنید.

اگر export با خطا روبه‌رو شود، Studio وضعیت HTTP و کد پایدار backend مثل `invalid_json`، `invalid_page_size`، `invalid_toc_depth`، `invalid_watermark_opacity`، `markdown_too_large` یا `render_failed` را نشان می‌دهد. اگر GUI را روی host غیرلوکال bind کنید، backend هشدار می‌دهد؛ چون کاربران قابل دسترس در شبکه می‌توانند Markdown و asset بفرستند.

مهم

پیش‌نمایش داخل GUI تقریبی است. PDF نهایی توسط backend renderer ساخته می‌شود و پردازش کامل MathJax، CSS appearance، Markdown، جلد و layout چاپی Chromium روی آن اعمال می‌شود.

مرجع CLI

توضیح	گزینه
فایل Markdown ورودی.	`input`
مسیر PDF خروجی.	`-o`, `--output`
front matter. موجود در metadata کردن override	`--title`, `--author`, `--description`
فعال‌سازی و تنظیم فهرست مطالب.	`--toc`, `--toc-depth`
کنترل صفحه‌بندی چاپی.	`--toc-page-break`, `--h1-page-break`
انتخاب `modern`، `github`، `textbook` یا `academic`.	`--style`
انتخاب `blue`، `emerald`، `violet`، `amber`، `rose`، `slate` یا `neutral`.	`--palette`
انتخاب `light` یا `dark`.	`--mode`
نمایش گزینه‌های appearance و خروج.	`--list-styles`, `--list-palettes`, `--list-modes`
اندازه صفحه مثل `A4`، `Letter`، `Legal`، `A4 landscape` یا ابعادی مثل `210mm 297mm`.	`--page-size`
اجبار جهت به `auto`، `ltr` یا `rtl`.	`--dir`
کنترل margin صفحه.	`--margin-top`, `--margin-bottom`, `--margin-x`
مسیر فونت‌های محلی Vazirmatn.	`--font-dir`

توضیح	گزینه
مسیر سفارشی Chromium یا Chrome.	`--chromium-path`
حالت sandbox مرورگر: `auto` ، `on` یا `off` . مقدار پیش‌فرض: `auto` .	`--chromium-sandbox`
ذخیره HTML میانی.	`--debug-html`
تنظیم جلد.	`--no-cover` , `--cover-logo` , `--no-cover-logo`
افزودن watermark متنی یا تصویری با لایه‌بندی هماهنگ با mode.	`--watermark` , `--watermark-image`
اجازه به سندهای قابل اعتماد برای بارگذاری تصویرهای remote `http(s)` . پیش‌فرض غیرفعال است.	`--allow-remote-assets`
حذف footer چاپی.	`--no-header-footer`
غیرفعال کردن MathJax.	`--no-mathjax`
غیرفعال کردن sanitization برای فایل‌های قابل اعتماد.	`--unsafe-html`
مرورگر بر حسب میلی‌ثانیه timeout.	`--timeout-ms`
حالت progress bar ترمینال: `auto` ، `on` یا `off` . مقدار پیش‌فرض: `auto` .	`--progress`

برای دیدن همه گزینه‌ها:

```bash
mrs-md2pdf --help
```

اتوماسیون و CI

یک دستور ساده برای ساخت PDF در اسکریپت:

```bash
mrs-md2pdf docs/report.md -o build/report.pdf --toc --style modern --palette blue --mode light
```

مخزن شامل workflow گیت‌هاب Actions است که Ruff، pytest و یک smoke test واقعی رندر با Chromium را روی نسخه‌های پشتیبانی‌شده Python اجرا می‌کند. برای CI بهتر است گزینه‌ها صریح باشند:

```bash
mrs-md2pdf docs/report.md -o build/report.pdf \
  --toc \
  --toc-depth 4 \
  --style modern \
  --palette blue \
  --mode light \
  --page-size A4 \
  --dir auto \
  --timeout-ms 60000 \
  --progress off
```

برای اشکال‌زدایی در CI، HTML میانی را به عنوان artifact نگه دارید:

```bash
mrs-md2pdf docs/report.md -o build/report.pdf --debug-html build/report.html
```

رفع اشکال

پیدا نشدن Chromium

این دستور را اجرا کنید:

```bash
python -m playwright install chromium
```

یا مسیر مرورگر را صریح بدهید:

```bash
mrs-md2pdf input.md -o output.pdf --chromium-path /path/to/chrome
```

متن فارسی ظاهر مناسبی ندارد

یک فونت فارسی مناسب مثل Vazirmatn نصب کنید یا مسیر فونت را بدهید:

```bash
mrs-md2pdf input.md -o output.pdf --font-dir ./fonts
```

اگر مسیر فونت وجود نداشته باشد یا فایل شناخته‌شده‌ای داخل آن پیدا نشود، renderer هشدار می‌دهد و از فونت‌های سیستم استفاده می‌کند.

تصویرها دیده نمی‌شوند

مطمئن شوید مسیر تصویرها نسبت به فایل Markdown درست است و داخل پوشه همان سند باقی می‌ماند. مسیرهای absolute، URLهای `file:`، خروج از پوشه با `..`، فایل‌های گم‌شده و تصویرهای خیلی بزرگ با placeholder امن جایگزین می‌شوند تا Chromium آن‌ها را از فایل‌سیستم load نکند. اگر از GUI استفاده می‌کنید، فایل‌ها یا پوشه‌های تصویر را قبل از خروجی گرفتن attach کنید و هشدارهای renderer را بررسی کنید.

فرمول‌ها به شکل TeX خام دیده می‌شوند

مطمئن شوید MathJax فعال است و هشدارهای renderer را بررسی کنید. برای سندهای بزرگ مقدار timeout را افزایش دهید:

```bash
mrs-md2pdf input.md -o output.pdf --timeout-ms 90000
```

نیاز به بررسی Layout

فایل HTML میانی را ذخیره کنید:

```bash
mrs-md2pdf input.md -o output.pdf --debug-html output.html
```

سپس `output.html` را در مرورگر باز کنید و ساختار و CSS تولیدشده را بررسی کنید.

پانویس

پانویس برای ارجاع، یادداشت فنی و توضیح تکمیلی مناسب است.[pipeline]

چک‌لیست نهایی انتشار

قبل از انتشار PDF مهم، این موارد را بررسی کنید:

- [x] عنوان جلد، زیرعنوان، نویسنده، تاریخ و metadata.
- [x] زبان و جهت سند.
- [x] عمق و لیبل‌های فهرست مطالب.
- [x] صفحه‌های دارای فرمول.
- [x] صفحه‌های دارای code block و inline code.
- [x] صفحه‌های دارای تصویر محلی.
- [x] جدول‌هایی که محتوای پهن دارند.
- [x] پانویس‌ها و لینک‌ها.
- [x] شماره‌گذاری footer بعد از جلد.
- [ ] بازبینی بصری نهایی در PDF viewer.
- [ ] هنگام انتشار نسخه tag شده، چک‌لیست release در `docs/RELEASE.md` کامل شده باشد.

---

1. برنامه Mardas MD2PDF به‌جای رسم مستقیم هر پاراگراف روی canvas فایل PDF، از Chromium برای layout استفاده می‌کند. این انتخاب باعث پشتیبانی بهتر از CSS print، متن ترکیبی، خروجی SVG MathJax، جدول‌ها، تصویرهای محلی و کدهای هایلایت‌شده می‌شود. ↩
   - پانویس چندخطی پشتیبانی می‌شود.
   - متن Markdown داخل پانویس حفظ می‌شود.

- توجه کنید که inline code مثل `@page`، `$x$` و `[^id]` وقتی داخل backtick نوشته شود خوانا و literal باقی می‌ماند.