

CSS 組版で Markdown を提出物に

Markdown を、そのまま お客様に出せる PDF へ

— 組版ツール「Vivliostyle」のすすめ —

Markdown で書きたい。でも提出は PDF。その悩み、今日で終わりです。

AI 駆動開発ドキュメント PDF化検証

Markdown で作成した開発ドキュメントを Vivliostyle で提出用 PDF に変換するための検証サンプル。

項目	内容
作成日	2026-06-30
作成者	開発チーム
版数	0.1
文書種別	検証用サンプル

▲ Vivliostyle で生成した実際の PDF

なぜドキュメントを Markdown で書くのか



理由① Git で管理できる

- Word / Excel はバージョン管理が破綻しがち
- GitHub 等で差分レビューができない
- テキストの Markdown なら差分・履歴・レビューが全部使える



理由② 生成 AI が読み書きできる

- Word / Excel は図形・結合セルで認識精度が落ちる
- 読めてもトークン（≒お金）を無駄に消費
- バイナリの「編集」は読むよりさらに困難



Markdown はただのテキスト。だから Git でも生成 AI でも自在に扱える。

具体例：追加仕様が来て、修正が必要になったら



Excel で管理していた場合

- ① 該当箇所を手で全部探して修正
- ② 修正箇所を目印 → レビュー → 目印を外す
- ③ そのあと実装・テスト・実装レビュー…



修正漏れや「実装だけして仕様書は放置」が頻発



Markdown + AI の場合

- ① 人が AI に修正を指示
- ② AI がドキュメント・コード・テストを修正し、自動テスト後にプルリクエスト
- ③ 人は Git 上で差分をレビューしてマージ



人間の仕事は「方向性の決定」と「受け入れ」だけ

ただし、Markdown には弱点がある



エンジニア同士なら

エディタや GitHub でそのまま読める。何の問題もない。



お客様への提出物としては

Markdown を読める標準ソフトがない。求められるのは、体裁の整った PDF。



Markdown から“きれいな PDF”を自動生成できればいい
それをしてくれるのが Vivliostyle

Vivliostyle とは? — CSS 組版ツール



CSS 組版

Markdown / HTML の原稿を、CSS のルールで「本の見た目」にレイアウトする技術



書籍品質の PDF 出力

表紙・目次・ページ番号・脚注まで自動で組んで PDF に書き出す



オープンソース & CLI

無料で使える OSS。コマンド 1 発でビルド、ライブプレビューも可能

Markdown / HTML



+ CSS (テーマ)



書籍レイアウトの PDF

出力例① 表紙も目次も、自動で「本」になる

AI 駆動開発ドキュメント PDF化検証

Markdown で作成した開発ドキュメントを Vivliostyle で提出用 PDF に変換するための検証サンプル。

項目	内容
作成日	2026-06-30
作成者	開発チーム
版数	0.1
文書種別	検証用サンプル

表紙

目次

1	はじめに	4
1.1	基本表現	4
2	AI 駆動開発の作業計画	5
2.1	作業フェーズ	5
2.2	長い表の確認	5
3	設計方針	6
3.1	文書テンプレートの考え方	6
3.1.1	注意書き	6
3.1.2	補足	6
4	サンプルコード	7
4.1	TypeScript	7
4.2	横に長いコード	7
4.3	JSON	7
5	Mermaid 図表サンプル	8
5.1	Markdown 内に直接書く Mermaid	8
5.2	Mermaid から変換した SVG	8
6	テスト計画	9
7	リストクヒ対策	10
8	補足資料	11
8.1	検証メモ	11

目次（見出しから自動生成）

原稿はぜんぶ Markdown。目次もページ番号も手作業ゼロ。

出力例② 表・コード・図も、そのままきれいに

2 AI 駆動開発の作業計画

2.1 作業フェーズ

フェーズ	目的	成果物	確認観点
企画	解くべき問題を明確にする	課題メモ、成功条件	目的が曖昧でないか
要件整理	必要な機能と制約を整理する	要件定義書	スコープが適切か
設計	実装前に構成を固める	設計書、API仕様	レビュー可能か
実装	AIと協働してコードを作る	Pull Request	差分が追えるか
検証	品質を確認する	テスト結果、PDF文書	再現性があるか

2.2 長い表の確認

ID	タスク	担当	優先度	ステータス	備考
T-001	MarkdownからPDFを生成する最小構成を作る	Dev	High	Doing	Vivliostyle CLIの設定、CSS、出力先を確認する
T-002	Mermaid図を選択記述した場 合の見え方を確認する	Dev	High	Todo	直帰レンダリングできない場合は、事前変換したSVGまたはPNGを利用する
T-003	横に長いコードブロックと URLの折り返しを確認する	Dev	Medium	Todo	縦方向PDFでは読みやすさを優先し、必要に応じてフォントサイズを調整する

表組み

4 サンプルコード

4.1 TypeScript

```
type TaskStatus = 'todo' | 'doing' | 'done';

type DevelopmentTask = {
  id: string;
  title: string;
  status: TaskStatus;
  owner: string;
};

const tasks: DevelopmentTask[] = [
  { id: 'T-001', title: 'Create Vivliostyle config', status: 'doing', owner: 'Dev' },
  { id: 'T-002', title: 'Render Mermaid diagrams', status: 'todo', owner: 'Dev' },
];

const activeTasks = tasks.filter(task => task.status !== 'done');
console.log(activeTasks);
```

4.2 横に長いコード

```
npm run build -- --output dist/vivliostyle-sample.pdf --theme styles/document.css --size A1 --timeout 120000
```

4.3 JSON

```
{
  "documentType": "development-plan",
  "output": "pdf",
  "checks": ["markdown", "tables", "code", "images", "mermaid", "footnotes", "page-media"]
}
```

コードハイライト

5 Mermaid 図表サンプル

5.1 Markdown内に直接書く Mermaid

```
flowchart TD
  A[要件を整理] --> B[AIに実装案を生成させる]
  B --> C[レビュー依頼]
  C --> D[問題なし] 何実装を採用
  C --> E[修正あり] プロンプトと設計を調整
  E --> B
  D --> F[テストとPDF化]
```

```
sequenceDiagram
    participant User as 利用者
    participant AI as AIエージェント
    participant Repo as リポジトリ
    User->>AI: 要件と制約を送る
    AI->>Repo: 実装と文書を更新する
    Repo->>AI: テスト結果を送る
    AI->>User: 変更内容とPDFを共有する
```

5.2 Mermaid から変換した SVG

次の図は samples/diagrams/ai-driven-workflow.mmd を Mermaid CLI で SVG に変換して貼り込む予定です。



AI 駆動開発ワークフロー

Mermaid 図

開発ドキュメントに必要な表現は、ひと通りこのまま通る。

使い方は、実質 2 コマンド

1

書く

いつも通り Markdown
で書くだけ



2

ビルド

CLI で PDF を生成



3

できあがり

そのまま提出できる PDF



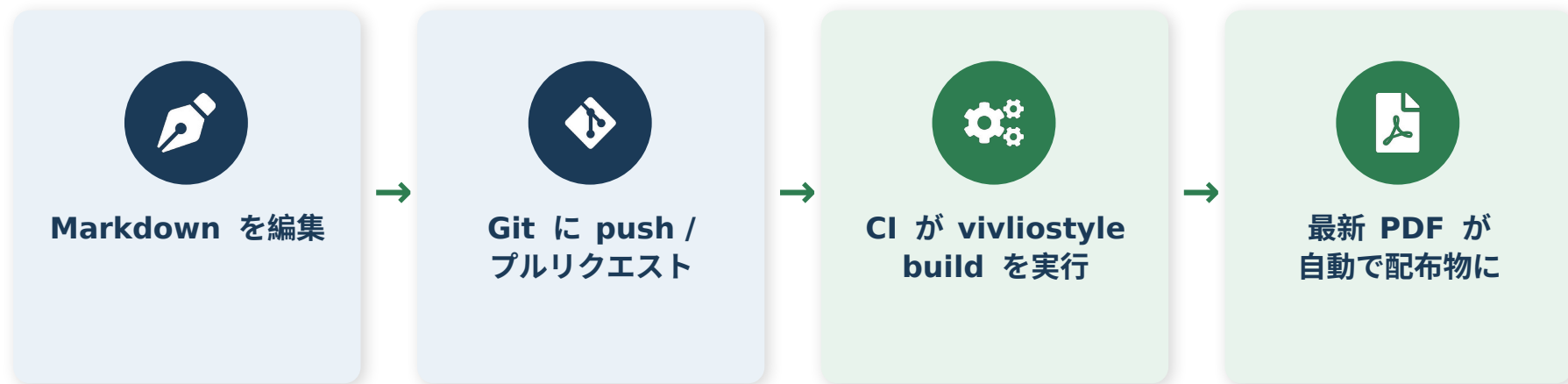
```
$ npm install -g @vivliostyle/cli
```

```
$ vivliostyle build spec.md -o spec.pdf
```

```
# 書きながら確認するなら : vivliostyle preview spec.md
```

見た目を変えたいときはテーマ（CSS）を差し替えるだけ。公式テーマも多数。

チーム運用: CI/CD で PDF は勝手に最新になる



人が触るのは **Markdown** だけ。提出用 **PDF** は常に最新が置いてある状態に。

補足： CSS 組版だから、「書籍」までつくれる



紙面設計は CSS で自由自在

判型・余白・柱・ノンブル（ページ番号）まで CSS で指定できる



日本語組版に強い

縦書き・ルビなど、日本語の本に必要な組版に対応



本づくりの実績

技術同人誌から商業出版まで、実際に本の制作に使われている

同じ Markdown 原稿から

提出用 PDF

印刷入稿用 PDF

技術同人誌・書籍

テーマ CSS を差し替えるだけ

まとめ: Markdown で書いて、Vivliostyle で出す

- 1 Markdown で書けば、Git と生成 AI をフル活用できる
- 2 Vivliostyle が、そのまま提出できる品質の PDF に変えてくれる
- 3 CSS 組版だから、その先の「書籍化」まで視野に入る

今日から試すなら → `npm install -g @vivliostyle/cli`