

Pandoc のトレンド 2024 : Djot と Typst

Pandoc ONLINE #1

藤原 惟 (sky_y)

2024-08-18

Pandoc のトレンド

Pandoc のトレンド

- Djot ¹
 - John MacFarlane (jgm) が開発した新しいマークアップ言語
 - CommonMark を一から作り直したような感じ
- Typst
 - 次世代の組版システム
 - タイプセットが速く、リアルタイムプレビューできる
 - マークアップ言語がフレンドリー

¹/d3at/ と読む (<https://djot.net/>)

なぜこの2つがトレンドなのか

※主観的なチョイスです

- Djot
 - jgm による意欲的な提案（面白そう）
- Typst
 - Pandoc の最近のリリースノートで頻繁に出てくる
 - 組版界隈でもよく聞く気がする

Djot

参考文献

- 日本における Djot エヴァンジェリスト：@daisuke (daisuke osada) さん
 - Djot はマークダウン愛から産まれてる～次代のマークダウン「Djot」をウォッチしよう #Markdown - Qiita
 - 日本語版 README
 - Markdown ユーザー向けのクイックスタート
 - Djot 構文リファレンス
 - Djot Japanese Docs
- 次世代の Markdown みたいな Djot の話

あらためて Djot とは

- Djot $\hat{=}$ CommonMark - CommonMark の複雑で解析が難しい部分 + 便利な記法
- 発端は jgm のエッセイ「Beyond Markdown」
 - <https://johnmacfarlane.net/beyond-markdown.html>

Beyond Markdown

- CommonMark
 - 可能な限り John Gruber の Markdown 仕様に忠実であるように努めた
 - 曖昧さを排除、統一性を高める
 - フェンスコードブロックなどいくつかの記法を追加
- CommonMark 仕様は複雑になってしまった
 - 強調を制御する原則は 17 個もあり、未解決なケースもある
- CommonMark 仕様の肥大化と複雑さを修正し、Markdown のよいところを維持したパラダイスみてえな軽量マークアップ言語を作りてえ

Beyond Markdown：例 強調のルール

強調の 17 個の非常に複雑なルールは、人間が最も自然だと感じる種類の読み方を可能にするために設計されている

- Q: `**this* text**` はどう解釈されうるか？

Beyond Markdown：例 強調のルール

強調の 17 個の非常に複雑なルールは、人間が最も自然だと感じる種類の読み方を可能にするために設計されている

- Q: `**this* text**` はどう解釈されうるか？
- `this* text`

Beyond Markdown：例 強調のルール

強調の 17 個の非常に複雑なルールは、人間が最も自然だと感じる種類の読み方を可能にするために設計されている

- Q: `**this* text**` はどう解釈されうるか？
- `this* text`
- `this text*`

Beyond Markdown：例 強調のルール

強調の 17 個の非常に複雑なルールは、人間が最も自然だと感じる種類の読み方を可能にするために設計されている

- Q: `**this* text**` はどう解釈されうるか？
- `this* text`
- `this text*`
- Pandoc の CommonMark の場合:

Beyond Markdown：例 強調のルール

強調の 17 個の非常に複雑なルールは、人間が最も自然だと感じる種類の読み方を可能にするために設計されている

- Q: `**this* text**` はどう解釈されうるか？
- `this* text`
- `this text*`
- Pandoc の CommonMark の場合:
 - `echo '**this* text**' | pandoc -f commonmark -t html`

Beyond Markdown：例 強調のルール

強調の 17 個の非常に複雑なルールは、人間が最も自然だと感じる種類の読み方を可能にするために設計されている

- Q: `**this* text**` はどう解釈されうるか？
- `this* text`
- `this text*`
- Pandoc の CommonMark の場合:
 - `echo '**this* text**' | pandoc -f commonmark -t html`
 - `→ <p>this text*</p>`

Djot ならどう書くか

強調 (emphasis) と強い強調 (strong emphasis) の書き方

- `_強調_`
- `*強い強調*`
- ルール
 - 強調の開き：直後に空白が続かない場合にのみ
 - 強調の閉じ：直前に空白がない場合、および開始文字と終了文字の間に区切り文字以外の文字がある場合のみ
 - `_ 強調されない例（余計な空白が入ってる） _`
 - `___ 2番目のアンダースコアは強調されない`
 - 入れ子にできる
 - 必要であれば `{_}` と `_}` で囲むことで開き・閉じを強制できる
 - [HTML] `this text*` → [Djot] `__{_this_} text_*`

Djot のうれしい記法

- ハイライト (mark)
 - `{=ここがハイライトされます=}`
- 挿入と取り消し線 (insert/delete)
 - `うちの上司は{-いじわる-}{+ナイス+}です`
- span
 - `[span]{.some-class #some-id some-key="some val"}.`
- div

```
{.warning}  
:::  
Watch out!  
:::
```

※上記と同等の div

```
::: warning  
Watch out!  
:::
```


Markdown ユーザー向けのクイックスタート

<https://github.com/dai/djot/blob/main/doc/ja-quick-start-for-markdown-users.md>

- Djot Playground (<https://djot.net/playground/>) で試すことができる
- Pandoc の最新版の場合 (現時点で 3.3)
 - `-f djot` または `-t djot` で変換可能

Djot : まとめ

- CommonMark
 - 人間にとって自然な読み方を可能にするために複雑なルールを持つ
- Djot
 - CommonMark の複雑さを排除し、Markdown のよいところを維持した軽量マークアップ言語

Typst

参考文献

- Typst 公式
 - GitHub - typst/typst
- Typst：いい感じの LaTeX の代替

Typst のインストール

```
# Linux (cargo)
$ cargo install typst-cli

# macOS
$ brew install typst
```

- VS Code: 次の拡張機能をインストール
 - Typst LSP (Nathan Varner)
 - vscode-pdf (tomoki1207)

素の Typst を試してみる

VS Code で `hello.typ` ファイルを作成

こんにちは

→保存すると即座に `hello.pdf` が生成される

もうちょっとレポートっぽくする

- GitHub - satshi/typst-jp-template

```
#import "template.typ": *  
#show: doc => jarticle(  
  fontsize: 11pt,  
  title: [夏に食べたいアイスについて],  
  authors: ([藤原 惟]),  
  date: [2024年8月18日],  
  doc,  
)
```

= 序論：夏とアイスの関係

夏は、気温の上昇により冷たい食べ物が特に求められる季節である。

Pandoc で変換する：Pandoc' s Markdown で書く

example-pandoc.md

```
---  
title: 夏に食べたいアイスについて  
author: 藤原 惟  
date: 2024年8月19日  
---
```

序論：夏とアイスの関係

夏は、気温の上昇により冷たい食べ物が特に求められる季節である。

Pandoc で変換する：Pandoc 側の設定

defaults.yaml

```
variables:  
  mainfont: "Harano Aji Mincho"  
  section-numbering: true  
  papersize: a4  
  lang: ja
```

※あえてテンプレートは使わず、Pandoc のデフォルト設定を使う

おまけ：Pandoc がデフォルトで使うテンプレートファイルを見る

- テンプレートに埋め込まれている変数 or メタデータを確認する

```
$ pandoc -D typst | less
```

または <https://github.com/jgm/pandoc-templates>

Pandoc で変換する：ビルド

```
$ pandoc example-pandoc.md -s -d defaults.yaml -t typst -o example-pandoc.typ
```

- `-s` は `--standalone` の省略形（ヘッダ・フッタ付き文書）
- `-d` は `--defaults` の省略形（デフォルトファイルを読み込む）

```
$ typst compile example-pandoc.typ
```

Typst : 所感とまとめ

- Typst はいいぞ
 - ビルドが爆速なのが嬉しい
 - 日本語テンプレートは点在するが、LaTeX ほど標準化されていない印象
 - マークアップの学習コスト低いし、もう素の Typst でいいんじゃない？
- Pandoc + Typst はがんばる必要あり
 - デフォルトのテンプレートが使いにくいので自作が必要
 - 既存の LaTeX や Markdown 文書の資産を移行するには役立ちそう

サンプルコード

- [GitHub - sky-y/pandoc-online-20240818](#)

おわり