

薄い本を作ってみたという話

名古屋高校 中西渉 watayan@meigaku.ac.jp

2023年の夏休みに、Vivliostyleという組版環境の勉強会があったので行ってきました。そこで作ったのがお配りした本です。これはHTMLとCSSの技術を使って組版を行うものです。準備が案外簡単だったので、ここで紹介する次第です。

その1 Vivliostyleとは

Vivliostyleはオープンソースで開発されている、CSS組版用のソフトウェアです。CSSで組版を行うというアイデアは古くからありますが、Webブラウザでの表示がそのまま印刷物に使える状態にはなっていません。そこでCSS組版を行うソフトウェアがいくつか作られており、Vivliostyleもその1つです。特徴としては、日本のコミュニティによる開発なので日本語組版用の機能（たとえば^{ふりがな}ルビ）があることや、レンダリングエンジンにWebブラウザであるChromiumを使用していることなどがあげられます。 $\int_0^{\infty} e^{-x^2} dx = \frac{\sqrt{\pi}}{2}$ のような数式もLaTeXと同じ記法で書くことができます。

CSSで細かい指定をするためにはHTMLも複雑なものにならざるを得ません。そこでVivliostyleではMarkdown形式で原稿を書き、それをHTMLに変換するようになっています。

その2 ソフトウェアの準備

その2の1 Node.js + Vivliostyle 本体

普通にNode.jsをインストールします（バージョン16以上）。その上でnpmを使ってVivliostyle本体をインストールします。私は-gをつけずにローカルインストールしていますが、それは好みの問題です。

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

その2の2 VS Code + 拡張機能

私が普段使うテキストエディタはMicrosoft Visual Studio Codeです（以下、VS Code）。VS CodeにはVivliostyle用の拡張機能vivliostyle-cli-helperがあるので、インストールします。これを入れると、右クリックで出るメニューでビルドしたりプレビューしたりすることができるので、いちいちコマンドを打つ必要がありません。

その3 原稿ファイルの準備

その3の1 フォルダ

私はCSS用にcss、画像用にimgというフォルダを掘っています。このあたりは各人の美意識次第でいろいろあると思います。

その3の2 CSSファイル

cssフォルダに適当な名前のCSSファイルを作ります。最低限次のことは書いておきます（サイズはA5以外でもいいです）。あとはデザインやフォントをいじるのにあわせていろいろ書き足します。そのあたりは書籍とか参考にしてください。私は『Web技術で「本」が作れる CSS組版Vivliostyle入門』（リブロワークス著）を傍らに置いています。

```
@charset "utf-8";  
@page{  
  size: A5;  
}
```

その3の3 原稿本体

原稿はMarkdown形式で書きますが、冒頭に文書に関する指定をYAML形式で書きます。ハイフンの有無など気をつけてください。

```
---
link:
  - rel: 'stylesheet'
    href: 'css/index.css'
lang: 'ja'
---
```

Markdown形式といっても、いろいろな方言があります。それらを統一しようという考えでCommonMarkが考えられていますが、Vivliostyleで用いるのはVFMという形式です。その違いのために、VS Code上では少し文句を言われたりすることもあります。あまり気にしないことにしています。CSSについても同じようなことがあります。気にしないでおきましょう。

その4 ビルド・プレビュー

拡張機能が入っていればVS Codeの画面で右クリックのメニューから選んだり、あるいは次のコマンドを入力したりすることでビルドやプレビューができます（index.mdは原稿ファイルの名前で置き換えてください。previewをbuildに変えるとビルドになり、output.pdfが生成されます。出力ファイル名は--outputで変更できます）。プレビューでChromiumが開かれますが、原稿ファイルやCSSをセーブするたびに表示内容も更新されます。

```
npx vivliostyle preview index.md
```

その5 印刷・製本

Adobe Readerには小冊子用のページ順で印刷するオプションがあります。おかげで印刷→紙折り機→ステプラで簡単に製本することができました。本校の印刷室にはなぜかそれ用の中綴じステプラ（しかも普通の10号針が使える）があったので助かりました。



おわりに

インストールなど少し手順が面倒なところがありますが、原稿がMarkdownで書けることや、CSSを共通にすることで統一したフォーマットの仕上がりになることなど、いろいろなメリットもあると思います。LaTeXだと体裁の変更はそれなりに面倒ですが、CSSならハードルはそんなに高くありません。

私自身は普段ワープロを使いませんが、それはWYSIWYG(What you see is what you get.)であることが煩わしいからです。生徒にワープロを使わせると、細かい体裁いじりに気持ちが行ってしまったり肝心の中身が疎かになってしまうこともあります（生徒に限らないかも）。しかしMarkdownで書いている間は整形結果がわかりませんから、文章の中身に集中しやすくなります。

情報の教科の話に戻りますが、私は教科書でテキストファイルの扱いが軽いことがずっと気になっています。文字コードや文字コード体系、エンコーディングなどの話がありますが、それを羅列して作られる最もシンプルなデータ形式であるテキストファイルのことが取り上げられていません。仕事のほとんどをテキストエディタ上でやっている身としては寂しいものがあります。