i 補足: この記事は Typst Advent Calender 2025 の TODO 日目の記事です。

執筆環境

ソフト名	バージョン	補足
Typst	0.14	TYPST_FEATURES=html
Tinymist Typst	0.14.4	—

1. はじめに

Typst、とてもいいですよね。 Markdown のような書き心地で figure や footnote や参考文献への参照等が書けるためとても気に入っています。 その Typst ですがバージョン 0.13 で HTML エクスポート機能が実験的に実装され、バージョン 0.14 では（依然として発展途上の機能ではあるものの）その機能が大幅に強化され、変換対象の関数も増えた上に、`html.elem` を使うことで直接 HTML タグを記述できるようになりました。

そこで今なら Typst でブログ作れるんじゃねぇかと考えたのです。 作ったのがこのブログです。

Typst で HTML を生成し CSS で色付けるだけなので技術的に難しいことはないのですがいくつか落とし穴があるのと、HTML エクスポートを実際に使用してみた内容の日本語の記事がみあたらなかったので記事を残すことにしました。

一応色々 WTFPL で公開しているものの、個人用に作ったものであるため他人が使うことを想定していません。 現時点でもここまでできるんだよということを紹介するためのものとして受け取ってもらいたいです。

Typst の HTML エクスポート機能はデフォルトだと有効化されていないため Tinymist で `html.hogehoge` を使うと `html` なんてねぇよというエラーが出てしまいます。 `TYPST_FEATURES` 変数に html を設定して VSCode を再起動することによって拡張機能に html 機能を認識させることができます。

2. 実装

2.1. Typst のテンプレートを作成する

文書を HTML エクスポートするだけなら `typst compile --features html --format html input.typ` すればよいのですが、この方法でコンパイルすると、style や meta であっても body 内に記述されてしまいます。 head に記述したい内容がある場合は `html.html` を使用してエクスポートする HTML の構造を一から記述する必要があるみたいです。

また、0.14 時点では align 等のレイアウト系命令がエクスポート時に無視されるため、そのあたりは CSS を使用して実装する必要があります。

2.2. Zotero のエクスポート形式に Hayagriva を追加、キーを Webサイトメインに修正

まず Typst の `bibliography` は BibLaTeX と Hayagriva の二つの形式をロード可能です。 Hayagriva というのは Typst が開発している文献管理用の形式で YAML を使用しています。

単にブログを書くだけなら文献リストは YAML の方が取り回しがよいでしょう。 しかし、Zotero のエクスポートは Hayagriva（Typst が開発している bibliography の管理用形式）に対応していません。 というわけで作成したのがこちらです。

論文を書くことはないので Web 用にカスタマイズしてあります。

2.3. サイトのアクセス解析を行う

アクセス解析に使えるのは色々ありますが 簡単に使えて同意を求めるプロンプトとか必要ない Cloudflare Web Analytics を採用します。

2.4. 検索をサポートする

PageFind を使用して検索をサポートします。

2.5. GitHub Actions を作成し GitHub Pages で公開する

`typst compile --features html --format html --root . (ディレクトリ)/index.typ public/(ディレクトリ)/index.html`

でお k！

3. 詰まった点・落とし穴

3.1. 数式のエクスポートが未実装

現時点では数式のエクスポートが未実装です。 しかし、数式を SVG 画像に変換して HTML に埋め込むことができます。

```
show math.equation.where(block: false): it => {
  html.elem("span", attrs: (role: "math"), html.frame(it))
}
show math.equation.where(block: true): it => {
  html.elem("figure", attrs: (role: "math"), html.frame(it))
}
```

3.2. html を一から構築する場合 footnote が使用できない

`html.html` でファイルを生成する場合、footnote を使用するとエラーが出ます。 幸い counter は使えるので自分で実装することで対処できます。

```
let note-counter = counter("my-footnote")
show footnote: it => {
  note-counter.step()
  let num = note-counter.get().first()
  html.span(class: "footnote-wrapper", {
    html.span(class: "footnote-marker", "[" + str(num) + "]")
    html.span(class: "footnote-content", it.body)
```

```
  })
}
```

参考文献