

- tenteroring project -

tenteroring project Flyer

202405-001号 / 発行者 : sierra

フリーペーパーの管理構成について

裏面 (PDF では次頁) に管理構成について書きましたので、御覧ください。

ゴールデンウィーク

とある事情により実家に帰れなくなったので、長く眠らせていた3Dプリンタを動かすことにしました。

手始めに作ったのが、M5StickCホルダーの改良版です。サイドの囲いを付け足して落ちないようにしました。当初は横にも置けるようにあえて付けていなかったのですが、横にすることも無いと思い付けました。あと、フックに引っ掛ける事が出来るようにしたパーツも作ってみました。技書博10に持っていかようと思います。



次に作ったのが amiibo の台。Xenoblade のホームラとヒカリ、ノアとミオを持っているのですが、2×2のひな壇のような台が欲しくて作りました。1段目の台を出力したとき、横が約115mmと長かったためか反ってシーソーのようになってしまいました。2段目のときは、1段目で出力したデータ(モデル)を半分に分割して最後に瞬間接着剤でくっつけました。とりあえず反ってはいませんが、真ん中に接合部があるのもなんだか見栄えが悪いような気がしました。

技術系情報ラジオ Recalog

Recalog は1週間にあったニュースや記事から kokorokagami と touden がピックアップして話す番組です。



<https://lit.link/recalog>

サークルについて tetro とゆかいな仲間たちのサークルです。各メンバー、自由気まま・好き勝手に活動しています。

サークルのホームページ <https://tenteroring.org/>



フリーペーパーの管理構成について

はじめに

202312-001号より、フリーペーパーをVivliostyleで作成することにしました。初回は、急ごしらえなどもあり、レイアウトや記事の(パージョン)管理の面でイマイチな部分がありました。今回は、そこら辺の改善を図った次第です。

構成

大きく分けると「content」「build」「theme」の3つの構成から成り立っています。Gitでの管理も、この構成単位で行っています。

content

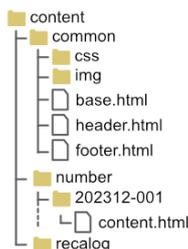


図1: "contentのフォルダ構成"

commonはヘッダー(header.html)やフッター(footer.html)などの共通的なパーツのHTMLファイルやCSS、画像を格納しています。

numberには記事を「YYYYMMDD-NNN」でフォルダ分けをして格納しています。

Recalogはtoudenさんとkokorokagamiさんのポッドキャストです。Recalogの紹介記事をフリーペーパーに載せており、それら関連するファイルを格納しています。

build

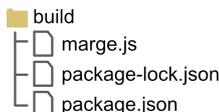


図2: "buildのフォルダ構成"

ビルドするためのプロジェクトです。

共通部分と記事のHTMLファイルを分けている関係で、ひとつのファイルにする処理を実装しました。処理としては、header.htmlやfooter.htmlなどを順番に読み込んで文字列結合をし、最終的にはbase.html(の文字列)に埋め込みます。(実際には置換)

また、どの記事でフリーペーパーを作成するのかコンソールで入力できるようにしています。

marge.jsの実行とvivliostyle buildを個別に叩くのが面倒なので、npmスクリプトで直列実行ができるようにnpm-run-allをインストールしています。

```
root@raspberrypi:/home/sierra/books/tpf/build# npm run tpf
> tpf-build@1.0.0 tpf
> run-s marge build

> tpf-build@1.0.0 marge
> node marge.js
Year (YYYY) > 2023
Month (MM) > 12
Number (NNN) > 001
> 202312-001
Read File : base.html
Read File : header.html
Read File : content.html
Read File : recalog.html
Read File : footer.html
Write File : output.html

> tpf-build@1.0.0 build
> vivliostyle build ../content/output.html --style ../theme/tpf-theme/theme.css
-o ../output.pdf
Processing PDF
✔ /output.pdf has been created.
✔ Built successful! (v)
root@raspberrypi:/home/sierra/books/tpf/build#
```

図3: "実行例"

theme

vivliostyle create themeで作成したテーマを格納しています。とは言っても、フォントサイズの変更と上下左右の余白ページを狭めているほか、プロパティの追加はしていません。なので、上部にファイル名や下部にページ番号などの表示もありません。

theme.css

```
@import url(./node_modules/@vivliostyle/theme-base/theme-all.css);
@import url(./node_modules/@vivliostyle/theme-base/css/lib/prism/base.css);
@import url(./node_modules/@vivliostyle/theme-base/css/lib/prism/theme-okaidia.css);

:root{
  --vs-font-size-on-print: 62.5%;
  --vs-page--margin-top: 10mm;
  --vs-page--margin-bottom: 10mm;
  --vs-page--margin-inner: 10mm;
  --vs-page--margin-outer: 10mm;
}
```

所感

1ページのフリーペーパーではvivliostyleを使うまでもないような気がしましたが、デザインまわり(theme、CSSのこと)までGitでの管理ができるのは良いかなと思います。まだ仕様について詳しくない部分もあり、theme.cssに組込むべきか別出し(今のCSSの管理形態)にすべきか、線引きを迷っているところはありますが。

本とかを書くともう少し理解が進むのかな?と思いました。