



\*\*\*\* 合同LT大会 / 小林 栄太

# LismUI-Vueという、 Vue.js向けのUIライブラリを作った

～ 新世代のCSSフレームワーク「Lism CSS」を土台にした開発記 ～



## 小林 栄太 (こばやし えいた)

S高等学校 2年生



Vue.js が好きで、  
UIライブラリ「LismUI-Vue」を個人で開発中

- CSSが得意・強い関心がある
- 趣味: 神道学(宗教思想史)・ネクタイ集め



## 前 提

UIの元となるCSSフレームワーク  
「Lism CSS」とは？



## 【レイアウト構築に特化し、CSS設計が厳密な「後発・日本製」FW】

- レイアウトはProps／クラスで素早く組む
- 作り込みはネイティブCSSで書く
- React, Astro向けに専用のコンポーネント！

### ■ 厳密なCSS設計

- @layerとコンテナクエリで破綻を防ぐ

開発者は、著名なWordPressテーマ「SWELL」などの開発で知られる[「了」氏](#)



---

2

---

なぜ、数あるFWの中から  
「Lism CSS」を選んだのか？

TailwindCSS v4、Bootstrapなどもあるのになぜ？

# 同じCSSフレームワークでも役割が違う



「CSSを書く余地を残しながら、設計とレイアウトを任せたい」  
→ 私はLism CSSが最も合っていた

※優劣ではなく、主な設計思想・担当範囲の比較



### ■ その中でも、Lism CSSの「ネイティブCSSと共存する」 思想が刺さった

- FWの責務： レイアウト構造とCSS設計（@layer等）をカバー
- 開発者によるCSS本来の表現力や「書く楽しさ」をしっかりと残してくれる

しかし、私が使うには一つ問題があった...

React, Astro向けのコンポーネントは提供されているが、  
一方、Vue向けの実装はなかった

なら私が作るしかない！



# 本題

## 無いなら作ろう 「LismUI-Vue」

Vue.jsへの移植プロジェクト「LismUI-Vue」始動



## ■ 作ったもの

- Vue向けに最適化したコンポーネント群
  - Vue向けアダプターを作り、Propsの型補完と本家の挙動を再現
- Nuxt 統合モジュール

## ■ 開発する上での考慮した点

1. 本家ドキュメントを読めばVue版でも使えるようにした
2. 一部React向けの仕様があるので、Vue.js 向けに改良。
  - a. 例えば、コンポーネントの命名規則、ケバブケースなど



- m="20" のようなPropsの型補完をVueでも再現
  - TypeScriptの型定義を工夫し、React/Astro版と同じ快適な体験を提供
- 「m」と入力すると、型補完が効いていることがわかる。
- 「mt」と入力すると、中身の型補完も効いている

```
<LismBox m bgc="base-2" bxsh="10"  
  | <p>ダウン m  
</LismBox> max-bsz  
  max-h  
<Lism bre max-sz  
<Lism util max-w  
  | <p>Lorem min-bsz  
  | facere min-h  
</Lism> min-sz  
<Lism atom min-w
```

```
b="50" mt="10">  
  10  
  15  
  20  
  30  
  40  
sectetur <a 5  
>doloribus 50  
  60  
  70  
  80
```

# アダプターの原理 (型補完はどう再現した?)



Lism CSSでは、Propsの型が用途ごとに分かれている。

それらを共通Props型 (LismCoreBaseProps) にまとめ、VueのdefinePropsへ渡す。さらに、ケバブケースや属性の扱いなど、Vue固有の差をアダプター側で調整する。

```
/**
 * LismCSS本家からインポートされた純粋なスタイリング・レイアウト用の基底プロパティ型
 * HTMLレンダリングに関連する属性 ('as', 'tag' など) は含まれません。
 */
export type LismCoreBaseProps = Partial<PropValueTypes & TraitProps & LayoutProps>
```

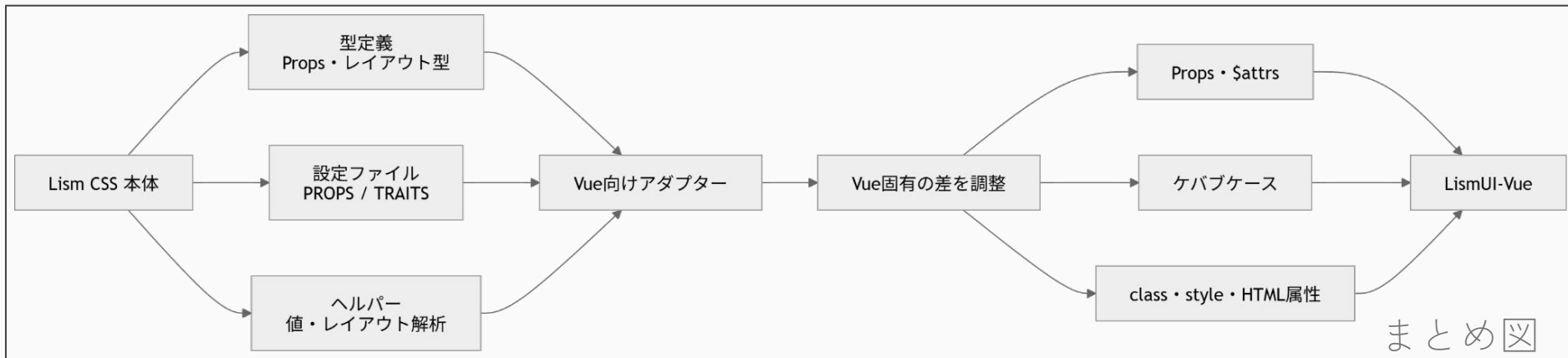
```
<script setup lang="ts">
import { Lism } from '../Lism'
import type { LismBaseProps } from '../../core/types'

type Props = /* @vue-ignore */ LismBaseProps

defineOptions({ inheritAttrs: false })
const props = defineProps<Props>()
</script>

<template>
  <Lism v-bind="{ ...props, ...$attrs }" layout="flex">
    <slot />
  </Lism>
</template>
```

# すべて作り直さず、本家の仕組みを再利用



- 本家の型定義・設定ファイル・ヘルパーを直接利用し、Vue固有の差だけをアダプターで調整。
  - 同じ定義の二重管理を減らし、本家の変更にも追従しやすい構成

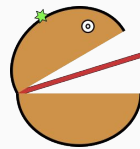


## ■ 今後の課題

- 現在、alpha版なのでbeta版に引き上げられるぐらい完成度を高める
- Lism CSS 本体の最新版へ追従
- LismUI-Vueの公式サイトを作り、ドキュメントを充実させる

## ■ 開発を通じて得たこと

- [unjs/changelogen](https://unjs.org/changelogen) を使用した変更履歴の生成とリリース作業の効率化
- 既存資産を活かし、保守しやすく設計する考え方



Vue版がないなら、自分で作ればいい！  
欲しいものがなければ、設計して実装する！  
これがエンジニアの探究心だ！

[npm](#), [GitHub](#) にも公開済なのでぜひ見てね！

ご清聴ありがとうございました！

