

引数のあるmixin  
のような仕組みを  
プラグインとして実装する

安田 祐平 (2024年3月12日)

Webアプリケーション開発のためのプログラミング技術情報誌  
FOR ALL WEB APPLICATION DEVELOPERS ウェブDBプレス

# WEB+DB PRESS

リント バンドラ ランタイム エッジ

識者がたどり着いた最適解 vol. 133 2023

# TypeScript 最新活用

コミッター直伝! Wasm対応、ReDoS対策、性能改善

# 速習 Ruby 3.2

まず作ってから、あとで共通化する

# Tailwind CSS

ツールを自由につないでWeb開発を効率化!  
今こそシェルスクリプト  
Go 1.20 / テストピラミッド / Radix UI



特集 3

# Tailwind CSS 実践入門

まず作ってから、あとで共通化する

安田祐平 YASUDA Yuhei  
著ブレイド

URL <https://yuheiy.com/>  
mail yuhei.yasuda1003@gmail.com  
GitHub yuheiy  
Twitter @\_yuheiy

CSSフレームワークのTailwind CSSが注目を集めています。その特徴は、flex、pt-4、text-centerといったユーティリティクラスの組み合わせだけで、ほとんどすべてのスタイルをてしまおうという大胆なアプローチです。これまでのベストプラクティスと真っ向から対立するようなやり方ですが、だからこそ享受できるいくつもの強みがあります。本特集では、そうしたTailwind CSSの考え方や、具体的な使い方について紹介します。

1章 ユーティリティファーストとは何か  
従来の課題を解決する大胆なアプローチ..... 62

2章 Tailwind CSS の基本  
開発環境へのインストール、設定ファイルの記述、エディタの設定..... 68

3章 Tailwind CSS の活用  
ユーティリティクラス、修飾子、カスタムスタイル..... 78

4章 Tailwind CSS とデザイン  
より良いデザインのためにCSSはどうあるべきか..... 84



```
// https://qiita.com/degudegu2510/items/09f34d4b218c9df6bb57
@mixin triangle($size) {
  width: $size;
  height: calc(#{$size} / 2 * tan(60deg));
  clip-path: polygon(50% 0, 100% 100%, 0 100%);
}

.triangle {
  @include triangle(100px);
  background-color: rebeccapurple;
}
```

```
const plugin = require('tailwindcss/plugin');

module.exports = {
  theme: {
    tabSize: { 1: '1', 2: '2', 4: '4', 8: '8' },
  },
  plugins: [
    plugin(({ matchUtilities, theme }) => {
      matchUtilities(
        { tab: (value) => ({ 'tab-size': value }) },
        { values: theme('tabSize') },
      );
    }),
  ],
};
```

```
matchUtilities(  
  { tab: (value) => ({ 'tab-size': value }) },  
  { values: theme('tabSize') },  
);
```

テーマを使う例:

```
<div class="tab-4">  
  ←— ... →  
</div>
```

```
.tab-4 {  
  tab-size: 4  
}
```

arbitrary valuesを使う例:

```
<div class="tab-[13]">  
  ←— ... →  
</div>
```

```
.tab-\[13\] {  
  tab-size: 13  
}
```

```
matchComponents(  
{  
  'auto-grid': (value) => ({  
    display: 'grid',  
    'grid-template-columns':  
      `repeat(auto-fill, minmax(min(${value}, 100%), 1fr))`,  
  }),  
},  
);
```

```
<p class="text-base/6 ... ">So I started to walk into the water ... </p>
<p class="text-base/7 ... ">So I started to walk into the water ... </p>
<p class="text-base/loose ... ">So I started to walk into the water ... </p>
```

---

tailwindcss/src/corePlugins.js :

```
text: (value, { modifier }) => {
  let [fontSize, options] = Array.isArray(value) ? value : [value]

  if (modifier) {
    return {
      'font-size': fontSize,
      'line-height': modifier,
    }
  }

  let { lineHeight, letterSpacing, fontWeight } = isPlainObject(options)
```

```
<table class="border-separate border-spacing-2×3 border border-slate-400 ... ">
  <thead>
    <tr>
      <th class="border border-slate-300 ... ">State</th>
```

---

```
.border-spacing-2×3 {
  border-spacing: 0.5rem 0.75rem;
}
```

```
<table class="border-separate border-spacing-x-2 border-spacing-y-3 border borde
  <thead>
    <tr>
      <th class="border border-slate-300 ... ">State</th>
```

```
.border-spacing-x-2 {
  --tw-border-spacing-x: 0.5rem;
  border-spacing: var(--tw-border-spacing-x) var(--tw-border-spacing-y)
}

.border-spacing-y-3 {
  --tw-border-spacing-y: 0.75rem;
  border-spacing: var(--tw-border-spacing-x) var(--tw-border-spacing-y)
}
```

Transforms:

```
<div class="scale-75 translate-x-4 skew-y-3">  
  ←— ... →  
</div>
```

Gradient Color Stops:

```
<div class="bg-gradient-to-r from-indigo-500 via-purple-500 to-pink-500 ..." >  
  ←— ... →  
</div>
```