



?



CSS Architecture

About Me

- yuki ohnaka
- 株式会社ナンバーフォー 🍀
 - **Front-end** / Server-side / Native-App / AWS engineer
- **Web Components** / React / **TypeScript** / Node.js / Docker / AWS / Vue / Java / PHP / Kotlin / Swift / Python ...
- [Qiita](#) / [Zenn](#)



良い CSS Architecture とは？

- 予測しやすい
- 再利用しやすい
- 保守しやすい
- 拡張しやすい

出典: [CSS Architecture](#)

(c) 2017 niewals



CSS 設計をするときの7つのポイント

1. コンテンツとスタイリングは疎結合にする

```
<!-- no good 🙄 -->  
<article class="article">  
  <h2 class="article__title">title</h2>  
</article>
```

```
/** no good 🙄 */  
.article h2 {  
  /** */  
}
```

1. コンテンツとスタイリングは疎結合にする

```
<!-- good 👍 -->  
<article class="article">  
  <h2 class="article__title">title</h2>  
</article>
```

```
/** good 👍 */  
.article__title {  
  /** */  
}
```


2. 影響範囲を無闇に広げない

```
/** no good 👎 */
```

```
#main-nav > div {
```

```
  /** */
```

```
}
```

```
/** good 👍 */
```

```
.main-nav__section {
```

```
  /** */
```

```
}
```

3. 親に関わらず動くようにする

```
/** no good 👎 */  
#main-nav .article {  
  /** */  
}
```

```
/** good 👍 */  
.article {  
  /** */  
}
```

4. 詳細度を無闇にあげない

```
/** no good 👎 */  
#main-nav ul li ul li div {  
  /** */  
}
```

```
/** good 👍 */  
.main-nav-sub-list__inner {  
  /** */  
}
```

5. 影響範囲が想像できるクラス名

```
<!-- no good 🙅 -->  
<article class="article">  
  <h2 class="title">title</h2>  
</article>
```

```
/** no good 🙅 */  
.article .title {  
  /** */  
}
```

5. 影響範囲が想像できるクラス名

```
<!-- good 👍 -->  
<article class="article">  
  <h2 class="article__title">title</h2>  
</article>
```

```
/** good 👍 */  
.article__title {  
  /** */  
}
```

6. 見た目・機能・役割が想像できるクラス名

```
/** no good 👎 */
```

```
.title1 {
```

```
  /** */
```

```
}
```

```
/** good 👍 */
```

```
.page-title {
```

```
  /** */
```

```
}
```

7. コンポーネントにマージンを持たせない

```
/** no good 👎 */  
.article {  
  margin-left: 20px;  
}
```

```
/** good 👍 */  
.article-container {  
  display: grid;  
  column-gap: 20px;  
}  
.article {  
  /** */  
}
```

さあ、やってみよう

チームで開発するためには、具体的なルールが
必要

既存の CSS を設計を取り入れる

- **BEM (MindBEMding)**
- OOCSS
- SMACSS
- FLOCSS
- Enduring CSS (ECSS)

必要に応じてカスタムする場合はある。

ご清聴ありがとうございました 🙌🙌🙌

@おまけ

CSS Architecture が必要か？

最近 CSS の周辺環境 は便利になってきている

- scoped CSS (Vue)
- CSS Modules (with React)
- CSS in JS (with React)
- styled-components (with React)
- Utility first CSS FW (tailwind)
- with Web Components

もう CSS Architecture なんてどうでもいいのでは？ 🤔

引き続き CSS Architecture は必要

- 解決できていない課題もある
- モダンな開発以外の開発案件もまだまだたくさんある
- CSS Architecture で考慮していた要素が、別（コンポーネント設計など）に移っただけなので、考えなければいけないことは一緒。

シングルクラス VS マルチクラス

抽象的過ぎないマルチクラスがおすすめ