

## コーディング規約 / 03\_ID・CLASS

---

- ID・CLASS
  - ID・CLASSの使用方法
    - - ID
    - - CLASS
  - IDの命名規則
  - CLASSの命名規則
    - - 基本構成
    - - 接頭辞
    - - BEM (Block、Element、Modifier)

## ID・CLASS

---

以下に記述する通りに使用します。

### ID・CLASSの使用方法

---

#### - ID

- 一つのHTMLファイル内に、同じIDは使用できない
- メンテナンス性を考慮し、IDに対してCSSのスタイルを当てることは禁止する
- Javascriptで個別に要素を指定する場合の使用は可とする

#### - CLASS

- 要素にCSSのスタイルを当てる場合は、原則的にCLASSを使用する
- 原則として「!important」は使用しない（utilityでの使用は可）

### IDの命名規則

---

- 必ず小文字で記述する
- 2単語以上使う場合は、2単語目の頭文字を大文字にする（キャメルケース）
- 記述をといつするために英単語の省略は禁止する（あまりに長い（3音節以上）場合は、コメントに残す前提で許可する）
- セクション、ブロックを簡潔に表す

# CLASSの命名規則

- 必ず小文字で記述する
- 2単語以上使う場合は、2単語目の頭文字を大文字にする（キャメルケース）
- 記述をといつするために英単語の省略は禁止する（あまりに長い（3音節以上）場合は、コメントに残す前提で許可する）
- FLOCSSとBEM（Block、Element、Modifier）を複合したものを使用する
- 接頭辞がつけられた親要素以下にある子要素、孫要素には、別の接頭辞を使用する場合以外でも接頭辞は省略しない
- クラス名が長くなりすぎることを避けるため、1つ上の親要素のクラス名を継承して、命名する

## - 基本構成

根本的な思想はFLOCSS を採用しています。  
クラス名の基本構成は、[接頭辞]-[Block]\_\_[Element]--[Modifier]です。

```
//例
[接頭辞]-[Block]__[Element]--[Modifier]
→ l-header__logo--white
```

## - 接頭辞

セクション、ブロックにあたる要素にクラスを付ける際に、接頭辞を付けることで、こういった役割をもっているのか、どのCSSファイルに記述されているのか、どれくらいの詳細度を持っているのかをクラス名だけで判別できるようになります。

	接頭辞	役割	例	CSSの格納場所
レイアウト (layouts)	l-	ページを構成する大枠の要素（全ページ共通部分など）に使用するクラス	l-header、l-footer	layouts/
コンポーネント (components)	c-	複数のページで再利用できる要素に使用するクラス	c-button、c-table、c-title	objects/components/
プロジェクト (projects)	p-	いくつかのComponentと、他の要素によって構成される大きな単位のモジュールを管理するクラス	p-card、p-article	objects/projects/
ユーティリティ (utilities)	u-	強制的にスタイルを当てる際に使用するクラス（「!important」の使用可）	u-clearfix、u-margin	objects/utilities/
Javascript	js-	Javascriptを当てる際に使用するクラス（スタイルをあててはいけない）	js-modal、js-accordion	objects/utilities/
ステート	is-	Javascriptなどによって、動的に追加・変更されるクラス	is-active、is-open、is-hidden	使用するCSSに記載

## - BEM（Block、Element、Modifier）

BEMを使用したクラス名を付けることで、クラス名だけでその要素がページの構成上、何を表しているのかを判別できるようになります。

	説明	接続詞
Block	ページ構成のルートとなる要素	先頭の為、無し
Element	Blockの構成要素	「_（アンダースコア）」2つ
Modifier	変化した状態を表す要素	「-（ハイフン）」2つ

## Block

ページ構成のルートとなる要素を表します。

## Element

「Element」は必ずしも、親要素となる「Block」の子要素に使用するわけではなく、あくまで「Block」の構成要素として使用します。「Element」に子要素がある場合は、その「Element」が「Block」になります。

## Modifier

「Block」、「Element」の変化した状態を表します。

```
//例
<header class="l-header">
  <div class="l-header__inner">
    
    
    <nav>
      <ul class="c-headerList">
        <li class="c-headerList__item"></li>
        <li class="c-headerList__item"></li>
        <li class="c-headerList__item"></li>
      </ul>
    </nav>
  </div>
</header>
<main class="l-main">
  <div class="l-main__inner">
    <div class="c-heading">
      <h1 class="c-heading__ttl">見出し①</h1>
      <p class="c-heading__txt">フォローテキスト</p>
    </div>
    <section class="l-section">
      <div class="p-contents">
        <h2 class="p-contents__ttl">見出し②</h2>
        <dl class="p-○○○○List">
          <dt class="p-○○○○List__ttl">データリストタイトル</dt>
          <dd class="p-○○○○List__detail">データ</dd>
          <dd class="p-○○○○List__detail">データ</dd>
          <dd class="p-○○○○List__detail">データ</dd>
        </dl>
      </div>
    </section>
  </div>
</main>
```

## 添付ファイル (0)

---



白藤 卓也によって 2020/10/26 19:15:11 に作成されました。



白藤 卓也によって 2020/12/22 11:11:39 に最終更新されました。