

Le code et les commentaires seront uniquement écrits en ANGLAIS pour **éviter le manque de cohérence et faciliter la visibilité**.

Dans le but de rendre les choses plus faciles à lire, séparez les classes dans votre HTML avec **deux (2) espaces** :

```
<div class="foo--bar bar__baz">
```

L'augmentation des espaces blancs devrait faciliter le repérage et la lecture de plusieurs classes.

Commentez un maximum pour rendre le tout **lisible** !

Pour le Dev Front BRYAN :

Notation BEM (Block, Élément, Modifieur).

BEM est une méthode pour nommer et classifier vos sélecteurs CSS de façon à les rendre beaucoup plus **strict, transparent et informatif**.

La convention de nommage suit ce modèle :

```
.block {}
```

```
.block__element {}
```

```
.block--modifieur {}
```

.block représente le niveau supérieur d'une abstraction ou d'un composant.

.block__element représente un descendant de .bloc puisqu'il contribue à former .bloc dans son ensemble.

.block--modifieur représente un état ou une version différente de .block.

Une analogie du fonctionnement de la méthode BEM :

```
.personne {}  
.personne--femme {}  
.personne__main {}  
.personne__main--gauche {}  
.personne__main--droite {}
```

Ici, nous pouvons voir que l'objet de base que nous décrivons est une personne, et qu'un autre type de personne pourrait être une femme. Nous pouvons également voir que les gens ont des mains, qui sont des sous-parties des personnes, et il y a différentes variantes, comme main gauche et main droite.

Nous pouvons maintenant nommer nos sélectionneurs en fonction de leurs objets de base et nous pouvons également savoir ce que le sélecteur fait, est-il un sous-composant (__) ou une variation (--)?

Ainsi, .page-wrapper est un sélecteur autonome, il ne fait pas partie d'une abstraction ou d'un composant et comme tel il est nommé correctement. Toutefois, .widget-heading est lié à un composant, c'est un enfant du constructeur .widget. Nous devrions renommer cette classe .widget__heading.

La notation BEM est un peu laide, plus verbeuse, mais elle nous donne beaucoup plus de pouvoirs dans le glanage d'informations sur les fonctions et sur les relations entre les éléments, uniquement avec leur nom de classes. En outre, la syntaxe BEM sera généralement très bien compressée (gzip) puisqu'elle favorise / fonctionne bien avec la répétition.

Pour la Dev Back Laury QUARTO :

Les fonctions et méthodes - camelCase

Considérations spéciales: alphanumériques uniquement, à l'exception de '_' comme décrit ici. Le nom doit décrire le comportement de la fonction ou de la méthode. Les méthodes qui sont déclarées avec modificateur privé ou protégé de visibilité doit commencer par un _.

Les variables - camelCase

Considérations spéciales : Les noms doivent décrire les données dans la variable.

Les constantes - ALL_CAPS

Considérations spéciales : alphanumériques et '_' sont autorisés cependant '_' doit être utilisé pour séparer les mots dans des constantes.