

Simplifiez votre code

- Lorsque vous codez, pensez à adopter le DRY : ne vous répétez pas !
- Lorsque vous créez des sélecteurs, faites attention à ne pas y entasser trop de règles. Séparer les sélecteurs par leur fonction vous aidera à garder votre code compréhensible et maintenable sur le long terme.
- Nommez vos sélecteurs en fonction de leur utilité.

Définissez des priorités

- La **spécificité** permet de mesurer le poids d'un sélecteur au sein du HTML, parmi les autres.
- Il existe **quatre niveaux** de spécificité :
 - les feuilles de style locales (*inline styles*) ;
 - les id ;
 - les classes, pseudoclasses et attributs ;
 - les éléments.
- Votre navigateur applique les éléments **les plus spécifiques** de façon prioritaire.
- En **créant et en appliquant votre propre structure**, vous pouvez mettre de l'ordre dans votre CSS.

Clarifiez votre HTML avec BEM

- Quand vous construisez vos pages, imaginez que vous empilez des boîtes dans des boîtes en les dessinant, ça vous aidera à mieux visualiser votre structure HTML et prévoir votre structure avec BEM.
- BEM signifie *bloc, élément, modificateur* :
 - les blocs sont des bouts de code autonomes ;
 - les éléments sont les parties qui forment le bloc ;
 - les modificateurs changent l'apparence ou le comportement d'un bloc ou d'un élément.
- Les blocs sont nommés en fonction de leur rôle : `navbar` .
- Les éléments indiquent le nom de leur bloc parent, suivi d'un double underscore/dunder (__) puis du rôle de l'élément : `navbar__link` .
- Les modificateurs utilisent le nom du bloc ou de l'élément qu'ils modifient, suivi de deux tirets (--) et de ce que le sélecteur modifie : `navbar__link--purple` .
- Les sélecteurs BEM sont toujours implémentés sous forme de classe.

Passez au niveau supérieur avec Sass

- Les préprocesseurs CSS vous permettent d'imbriquer votre code (on appelle ça le *nesting*) pour créer une hiérarchie plus facile à lire, et regrouper des morceaux de code entre eux.
- Il existe plein de préprocesseurs, mais le plus courant est Sass.
- La syntaxe `.scss` est très similaire à la syntaxe CSS.

- La syntaxe `.sass` est plus concise, mais `.scss` reste plus couramment utilisée.
 - On peut utiliser du code CSS préexistant dans Sass en l'intégrant dans un fichier `.scss`.
 - Le code Sass doit être compilé en CSS pour que les navigateurs puissent le lire et l'utiliser.
-

Optimisez la lecture du code

- Dans Sass, les sélecteurs peuvent être imbriqués, ou nestés, dans d'autres sélecteurs.
- Le nesting permet d'imbriquer les sélecteurs comme en HTML.
- Sass peut créer tous les combineurs CSS.
- Utilisez l'esperluette pour lier un sélecteur imbriqué à un sélecteur parent sans recourir à un combineur.
- Attention à ne pas créer de sélecteurs trop spécifiques en abusant du nesting !

Utilisez les variables et mixins

- Les variables stockent des valeurs que vous pouvez réutiliser dans tout votre codebase.
- Pour définir une variable, tapez le signe dollar `$` puis le nom de la variable : `$nom-de-votre-variable`.
- Quand vous modifiez la valeur de la variable, cela modifiera cette valeur partout où la variable a été utilisée.
- Les mixins sont très similaires aux variables, mais au lieu de ne stocker que des valeurs, elles stockent des blocs de code entiers.
- On utilise le mot clé `@mixin` pour déclarer une mixin.
- On utilise le mot clé `@include` pour utiliser une mixin dans son code.
- Lorsque Sass compile les instances d'une mixin, il remplace la mixin par les règles qu'elle contient.
- Pour rendre les mixins plus adaptables et réutilisables, vous pouvez inclure des arguments lorsque vous les déclarez.

Améliorez vos mixins grâce aux fonctions

- Les fonctions sont des bouts de code réutilisables qui effectuent des tâches prédéfinies.
- Une fonction doit être appelée pour pouvoir être utilisée.
- Sass fournit tout un tas de fonctions prêtes à l'emploi que vous pouvez utiliser pour manipuler des valeurs et rendre votre code mieux organisé, et du coup plus facile à maintenir.
- Pour comprendre comment utiliser une fonction, lisez sa documentation, elle regorge d'explications.

Adaptez votre code sur tous supports

- Les media queries vous permettent d'adapter votre contenu à différents écrans, en disant au navigateur d'utiliser un ensemble de règles alternatives dans certaines circonstances.
- Pour exécuter une media query, on utilise la règle CSS `@media`, suivie de la liste des queries et d'une paire d'accolades contenant les ensembles de règles alternatives requises.
- Avec Sass, vous pouvez nester les media queries, ce qui vous permet de les imbriquer dans le bon bloc BEM.

Déployez votre code Sass en ligne