ESN 81 - Bachelor CDWFS
Développement d'interfaces front-end
2025-2026

SCSS

Le concept de base : écrire du code scss facile à maintenir.
Le principe de fonctionnement : On écrit du code scss. Ce code est compilé en css, utilisé par les navigateurs.

Installation

En ligne de commande : https://sass-lang.com/install/
Extension VsCode : « Live Sass Compiler »

Imbrication

La syntaxe SCSS permet de faire de l'imbrication. Cette imbrication permet de regrouper les règles affectées par un même parent et de garder le code organisé. En revanche, il faudra garder en tête les bonnes pratiques CSS et éviter de trop imbriquer les règles au risque de créer des règles trop complexes.

L'élément parent &

Exemple :

```
// & : représente l'élément parent (optionnel)

a {
     color: #ce649b;
     &:hover {
         color: #9b3168;
     }
     &:focus {
         color: #da8bb4;
     }
}
```

Media-queries

Il est possible d'utiliser des media queries au sein même des sélecteurs

```
.container {
  width: 100%;
  padding: 20px;

  @media (max-width: 768px) {
    padding: 10px;
```

```
  }

  @media (max-width: 480px) {
    padding: 5px;
  }
}
```

Pour aller plus loin

Combinaison avec la méthodologie BEM

Exemple d'un composant « Carte de produit » en BEM

syntaxe html

```
<div class="product-card">
  <h2 class="product-card__title">Nom du produit</h2>
  <p class="product-card__description">Description du produit</p>
  <button class="product-card__button--primary">Acheter</button>
</div>
```

Code Scss

```
.product-card {
  border: 1px solid #ddd;
  padding: 16px;

  &__title {
    font-size: 20px;
  }

  &__button {
    background-color: blue;
    color: white;

    &--primary {
      background-color: red;
    }
  }
}
```

Les variables

Il est possible de définir des variables sass.

Exemple :

```
$link_color: #ce649b;
$color_hover: #9b3168
$color_focus: #da8bb4

a {
     color: $link_color;
     &:hover {
         color: $color_hover;
     }
     &:focus {
         color: $color_focus;
     }
}
```

Différence entre variables sass et scss ?
https://walkerspider.com/blog/differences-entre-les-variables-css-et-sass/

Les fonctions

Sass comportent de nombreuses fonctions, bien utiles.

darken() et lighten()

```
$link_color: #ce649b;
$color_hover: darken($link_color, 20%); // fonction SASS qui assombrit la couleur
$color_focus: lighten($link_color, 10%); // fonction SASS qui éclaircit la couleur

a {
     color: $link_color;
     &:hover {
         color: $color_hover;
     }
     &:focus {
         color: $color_focus;
     }
}
```

Autres exemples :
https://www.pierre-giraud.com/sass-apprendre-cours-complet/sass-fonction/

Importation de feuilles de styles

En css, il est déjà possible d'organiser son css en plusieurs feuilles de style.

En html avec la balises <link>

On peut utiliser plusieurs balises <link> L'inconvénient : c'est le navigateur qui a la charge de lancer une requête http, qui nuit au temps de chargement et à la performance.

En css avec la fonction @import

La fonction @import url() permet de les appeler à partir de la feuille de style principale. Mais c'est pire encore car les chargements ne sont pas parallélisés.

En scss

En scss, c'est plus performant : l'importation se fait au moment de la compilation.

Syntaxe de base

```
// main.scss
@use 'lists.scss';
// lists.scss
ul, ol {
  text-align: left;
}
```

Syntaxe courante

Nom de fichier

le nom de fichier peut être précédé d'un symbole « _ », afin que la compilation ne produise pas 2 fichiers, mais un seul, le fichier main.css.

```
// _lists.scss
```

Omission de l'extension

On peut omettre l'extension, sass saura trouver le bon fichier

```
@use 'lists';
```

Le modèle 7-1

Le modèle 7-1 est une pratique typique pour organiser les fichiers Sass dans les grands projets. Il divise votre code de style en sept dossiers thématiques plus un fichier principal (main.scss) qui importe tout :

```
|– base/
|   |– _reset.scss       # Reset/normalize
|   |– _typography.scss  # Typography rules
|   ...                  # Etc…
|
|– components/
|   |– _buttons.scss     # Buttons
|   |– _carousel.scss    # Carousel
|   ...                  # Etc…
```

```
|
|– layout/
|   |– _navigation.scss  # Navigation
|   |– _grid.scss        # Grid system
|   |– _header.scss      # Header
|   |– _footer.scss      # Footer
|   |– _sidebar.scss     # Sidebar
|   |– _forms.scss       # Forms
|   ...                  # Etc…
|
|– pages/
|   |– _home.scss        # Home specific styles
|   |– _contact.scss     # Contact specific styles
|   ...                  # Etc…
|
|– themes/
|   |– _theme.scss       # Default theme
|   |– _admin.scss       # Admin theme
|   ...                  # Etc…
|
|– utils/
|   |– _variables.scss   # Sass Variables
|   |– _functions.scss   # Sass Functions
|
|– vendors/
|   |– _bootstrap.scss   # Bootstrap
|   |– _jquery-ui.scss   # jQuery UI
```

https://gist.github.com/rveitch/84cea9650092119527bc

La compilation

Une fois Sass installé, vous pouvez compiler votre code Sass en CSS à l'aide de la commande `sass`. Vous devrez indiquer à Sass le fichier source à compiler et le répertoire de destination du CSS. Par exemple, la commande suivante exécutée dans votre terminal compilera un fichier Sass unique, `input.scss`, en un fichier `output.css`.

```
sass input.scss output.css
```

Vous pouvez également surveiller des fichiers ou des répertoires spécifiques avec l'option `--watch`. Cette option indique à Sass de surveiller vos fichiers sources et de recompiler le CSS à chaque sauvegarde. Si vous souhaitez surveiller (au lieu de compiler manuellement) votre fichier `input.scss`, ajoutez simplement l'option `--watch` à votre commande, comme ceci :

```
sass --watch input.scss output.css
```

Vous pouvez surveiller et exporter vers des répertoires en utilisant leurs chemins respectifs comme entrées et sorties, séparés par un deux-points. Dans cet exemple :

```
sass --watch app/sass:public/stylesheets
```

Documentation

https://sass-lang.com/documentation/
https://sass-guidelin.es/fr/
https://devhints.io/sass
https://www.pierre-giraud.com/sass-apprendre-cours-complet/sass-heritage/