

Développement front

MMI 2 – CM#1 S3



Danielo **JEAN-LOUIS**

Limites du CSS

- Très verbeux
- Beaucoup de répétitivité

CSS Nesting (Imbrication)

- Nouveauté CSS (apparu fin 2023)
 - Non géré par les anciens navigateurs
- Syntaxe inspirée des préprocesseurs CSS comme SCSS

Source(s) :

- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_nesting/Using_CSS_nesting - anglais
- <https://genesis-technology.fr/pourquoi-le-css-nesting-module-est-une-revolution-pour-le-frontend/>

CSS Nesting (Imbrication)

- Permet d'imbriquer des sélecteurs CSS
 - Réduit la duplication de code
 - Améliore sa maintenance
 - Fonctionne avec les règles-at / pseudo-classes et pseudo-éléments

Source(s) :

- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_nesting/Using_CSS_nesting - anglais
- <https://genesis-technology.fr/pourquoi-le-css-nesting-module-est-une-revolution-pour-le-frontend/>

CSS Nesting - Imbrication

```
.navigation {  
  display: flex;  
  .navigation-el {  
    font-size: 1.25rem;  
    color: blue;  
  }  
}
```

L'imbrication est interprétée de la façon suivante par le navigateur

```
.navigation {  
  display: flex;  
}  
.navigation .navigation-el {  
  font-size: 1.25rem;  
  color: blue;  
}
```

Source(s) :

- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_nesting/Using_CSS_nesting - anglais
- <https://genesis-technology.fr/pourquoi-le-css-nesting-module-est-une-revolution-pour-le-frontend/>

CSS Nesting – Imbrication deux sélecteurs

```
...  
.header,  
.footer {  
  a {  
    color: #007566;  
  }  
}
```

L'imbrication est interprétée de la façon suivante par le navigateur

```
...  
.header a {  
  color: #007566;  
}  
.footer a {  
  color: #007566;  
}
```

Source(s) :

- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_nesting/Using_CSS_nesting - anglais
- <https://genesis-technology.fr/pourquoi-le-css-nesting-module-est-une-revolution-pour-le-frontend/>

CSS Nesting - Pseudo-classes

```
● ● ●  
  
.paragraphe {  
  color: red;  
  
  &:hover {  
    color: blue;  
  }  
}
```

Dans le cas d'une pseudo-classe, on la préfixe d'une esperluette "&".
Fonctionne également avec les pseudo-éléments et les classes

```
● ● ●  
  
.paragraphe {  
  color: red;  
}  
  
.paragraphe:hover {  
  color: blue;  
}
```

Source(s) :

- https://developer.mozilla.org/en-US/docs/Web/CSS/Nesting_selector#syntax - anglais

CSS Nesting – Conseils et limites

- Éviter des imbrications trop profondes
- Impossibilité de faire de la programmation
- Ne fonctionne pas avec les vieux navigateurs
 - Dans une moindre mesure

Source(s) :

- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_nesting/Using_CSS_nesting - anglais
- <https://genesis-technology.fr/pourquoi-le-css-nesting-module-est-une-revolution-pour-le-frontend/>

CSS Nesting – Syntaxe alternative

- Nécessite l'utilisation d'une esperluette (&) devant chaque élément imbriqué

```
...  
.parent {  
  /* style du parent */  
  & .child {  
    /* style de l'enfant */  
  }  
}
```

```
...  
.parent {  
  /* style du parent */  
  .child {  
    /* style de l'enfant */  
  }  
}
```

Les deux codes sont équivalents, à gauche syntaxe avec le sélecteur d'imbrication à droite sans le sélecteur.
Les deux codes seront interprétés de la même façon par le navigateur

Source(s) :

- https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_nesting/Using_CSS_nesting - anglais
- <https://genesis-technology.fr/pourquoi-le-css-nesting-module-est-une-revolution-pour-le-frontend/>

CSS Nesting – Syntaxe alternative

```
...  
.header {  
  font-size: 2rem;  
  @media (orientation: landscape) {  
    & {  
      font-size: 1.75rem;  
    }  
  }  
}
```

```
...  
.header {  
  font-size: 2rem;  
  @media (orientation: landscape) {  
    font-size: 1.75rem;  
  }  
}
```

```
...  
.paragraphe {  
  & > a {  
    color: #000000;  
  }  
}
```

```
...  
.paragraphe {  
  > a {  
    color: #000000;  
  }  
}
```

Les deux codes sont équivalents, à gauche syntaxe avec le sélecteur d'imbrication à droite sans le sélecteur.
Les deux codes seront interprétés de la même façon par le navigateur

Pratiquons ! - Découvrons CSS Nesting

Pré-requis :

- Avoir la ressource `ressources/css-nesting`

A télécharger ici :

https://github.com/DanYellow/cours/raw/refs/heads/main/integration-web-s3/cours-magistraux/numero-1/integration-web-s3_cours-magistraux_numero-1.ressources.zip

Préprocesseurs CSS

- Méta-langages CSS
- Ne sont pas lus par les navigateurs
 - **Doivent être compilés en CSS**
- Simplifient l'écriture du CSS
- Utilisent une syntaxe proche du CSS
 - Tout ce que vous avez appris en CSS est réutilisable

Source(s) :

- <https://www.alsacreations.com/article/lire/1717-les-preprocesseurs-css-c-est-sensass.html>
- https://developer.mozilla.org/fr/docs/Glossary/CSS_preprocessor

Préprocesseurs CSS

- Apportent de nouvelles fonctionnalités :
 - Conditions / boucles
 - **Variables compilées** – Elles ne sont pas modifiables dans le CSS
 - ...

Source(s) :

- <https://www.alsacreations.com/article/lire/1717-les-preprocesseurs-css-c-est-sensass.html>
- https://developer.mozilla.org/fr/docs/Glossary/CSS_preprocessor

Préprocesseurs CSS

- Compatibles avec tous les navigateurs
- **Ne jamais éditer le fichier CSS compilé**
 - Les modifications seront écrasées à la modification des fichiers sources
- **Ne pas commiter le fichier CSS compilé**
 - Car vous avez déjà les sources

Source(s) :

- <https://www.alsacreations.com/article/lire/1717-les-preprocesseurs-css-c-est-sensass.html>
- https://developer.mozilla.org/fr/docs/Glossary/CSS_preprocessor

SASS / SCSS

- Préprocesseurs CSS les plus populaires
- .scss et .sass comme extension de fichiers
- SASS : Utilise une syntaxe proche du ruby
- SCSS : Utilise une syntaxe proche du CSS
 - Plus populaire
- Possibilité de faire des commentaires avec //

Source(s) :

- <https://grafikart.fr/tutoriels/differences-sass-scss-329>
- <https://la-cascade.io/se-lancer-dans-sass/>
- <https://sass-lang.com/>
- <https://codebuds.com/blog/sass-scss-introduction>

SASS / SCSS - Imbrication

- Permet de grouper des sélecteurs CSS
 - Réduit la duplication de code
 - Améliore sa maintenance
 - Fonctionne avec les règles-at / pseudo-classes
- S'inspire de l'imbrication des balises HTML

Source(s) :

- <https://sass-lang.com/guide/#nesting>

SASS / SCSS - Imbrication

```
.navigation {  
  display: flex;  
  .navigation-el {  
    font-size: 1.25rem;  
    color: blue;  
  }  
}
```

Le SCSS est compilé en CSS pour être lu par le navigateur

```
.navigation {  
  display: flex;  
}  
.navigation .navigation-el {  
  font-size: 1.25rem;  
  color: blue;  
}
```

Source(s) :

- <https://sass-lang.com/guide/#nesting>

SASS / SCSS - Imbrication

- Fonctionne également avec :
 - Les règles-at (par exemple, medias queries)
 - Les pseudo-classes / elements

Source(s) :

- <https://sass-lang.com/guide/#nesting>

SASS / SCSS - Imbrication

```
...  
.lien {  
  color: blue;  
  transition: color 350ms;  
  
  &:hover {  
    color: black;  
  }  
}
```

L'esperluette (&) permet d'indiquer que ce qui suit doit être "collé" au sélecteur

```
...  
.lien {  
  color: blue;  
  transition: color 350ms;  
}  
.lien:hover {  
  color: black;  
}
```

Source(s) :

- <https://sass-lang.com/guide/#nesting>

SASS / SCSS - Imbrication

```
...  
.list-images {  
  display: flex;  
  flex-wrap: wrap;  
  flex-wrap: wrap;  
  justify-content: space-between;  
  
  @media (max-width: 767px) {  
    overflow: auto;  
    flex-wrap: nowrap;  
    column-gap: 10px;  
  }  
}
```

Dans le cas d'une règle-at, on ne remet dans le sélecteur dedans, ainsi elle s'applique sur le sélecteur racine

```
...  
.list-images {  
  display: flex;  
  flex-wrap: wrap;  
  flex-wrap: wrap;  
  justify-content: space-between;  
}  
@media (max-width: 767px) {  
  .list-images {  
    overflow: auto;  
    flex-wrap: nowrap;  
    column-gap: 10px;  
  }  
}
```

Source(s) :

- <https://sass-lang.com/guide/#nesting>

SASS / SCSS - Imbrication

- Conseil : évitez de faire des imbrications trop profondes
 - Création de sélecteurs CSS trop précis

Source(s) :

- <https://sass-lang.com/guide/#nesting>

SASS / SCSS – Import / Module

- Permet la séparation du code CSS en plusieurs fichiers (appelés “partials”)
 - Fusion à la compilation
 - Nécessite d'importer les fichiers
- Fichiers importés sont préfixés par un underscore (_)
 - Évite à SASS/SCSS de créer un fichier dédié

Source(s) :

- <https://sass-lang.com/guide/#modules>
- <https://sass-lang.com/guide/#partials>

SASS / SCSS – Import / Module



```
// _base.scss
body {
  max-width: 1280px;
  margin: 0 auto;
  background-color: aliceblue;
}
```



```
@import '_base';

.lien {
  font-family: 'Courier New', Courier, monospace;
  color: #333;
}
```



```
// style.css
body {
  max-width: 1280px;
  margin: 0 auto;
  background-color: aliceblue;
}

.lien {
  font-family: 'Courier New', Courier, monospace;
  color: #333;
}
```

A la compilation les deux fichiers sont fusionnés

Source(s) :

- <https://sass-lang.com/guide/#modules>
- <https://sass-lang.com/guide/#partials>

SASS / SCSS – Import / Module

- La méthode “use” permet de faire la même chose mais de façon moderne
 - A préconiser
 - Ne fonctionne pas partout en revanche

Source(s) :

- <https://sass-lang.com/guide/#modules>
- <https://sass-lang.com/guide/#partials>

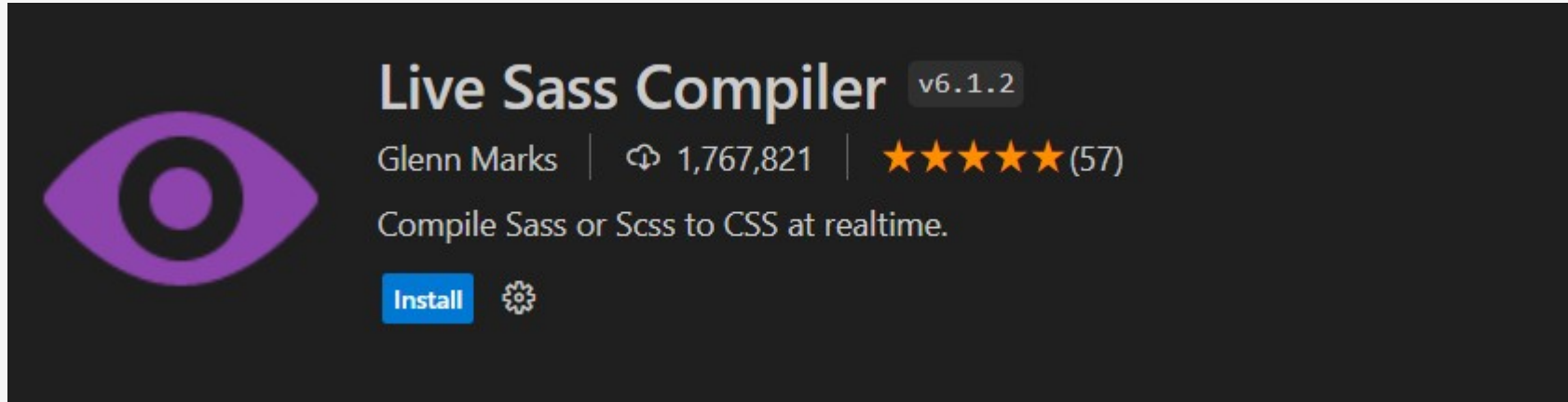
SASS/SCSS - Compilation

- Action de transformer les fichiers sources en CSS
- Plusieurs méthodes :
 - En ligne (utilisation d'un playground – voir sources)
 - Node js. Solution la plus répandue. Abordé en S5
 - Extension VS Code

Source(s) :

- <https://sass-lang.com/playground/>

SASS/SCSS - Compilation

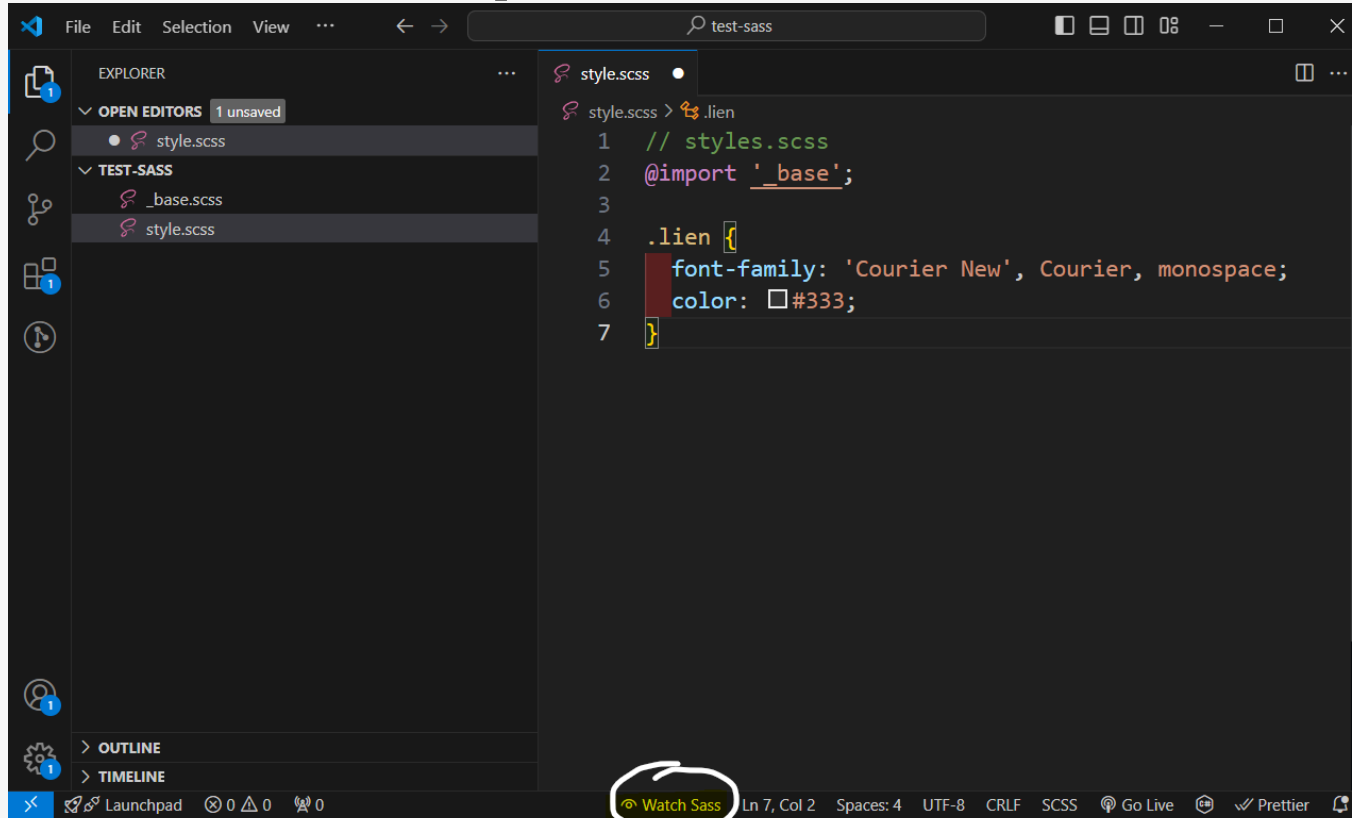


Live Sass Compiler est une extension VS Code gratuite permettant de compiler le SASS et le SCSS en CSS

Source(s) :

- <https://marketplace.visualstudio.com/items?itemName=glenn2223.live-sass>

SASS/SCSS - Compilation

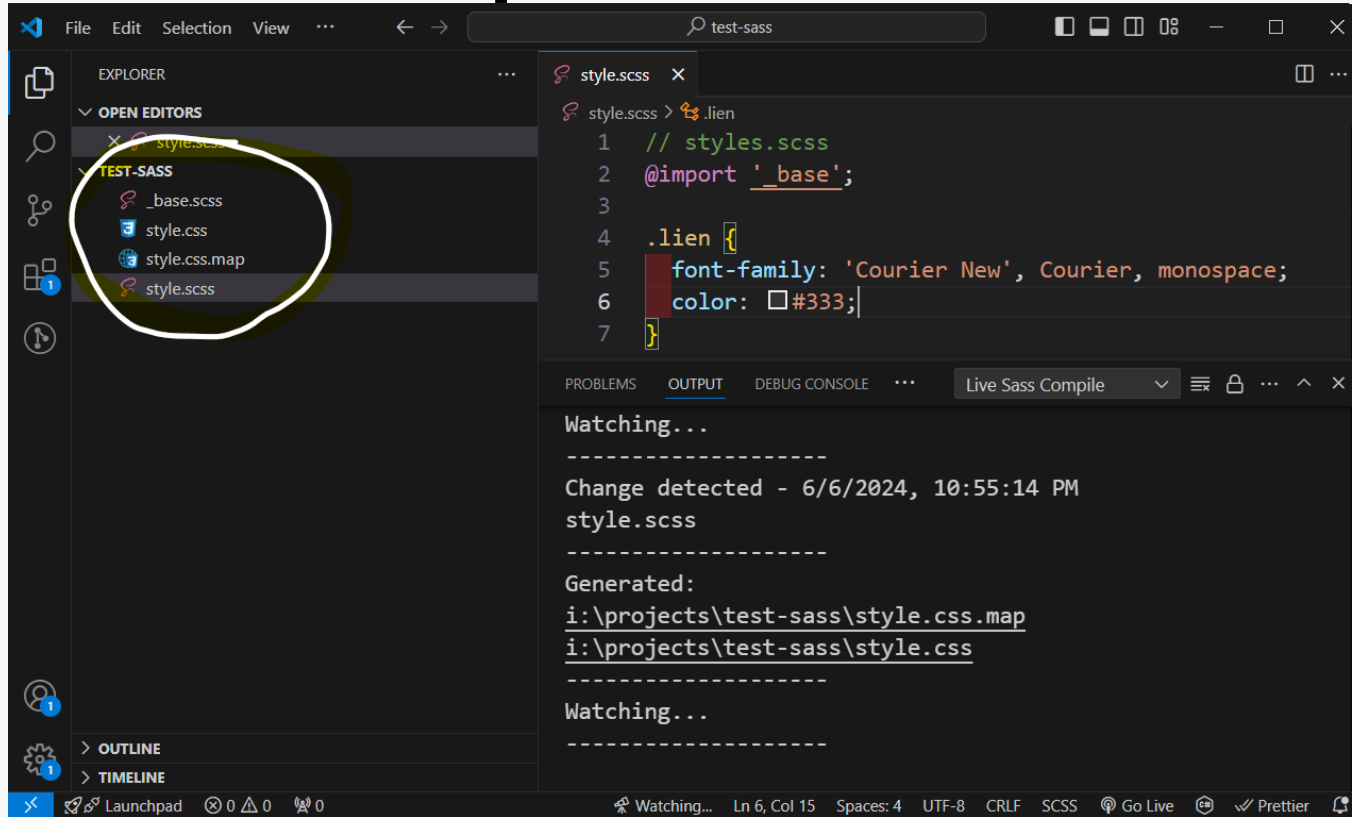


L'extension rajoute l'option "watch Sass" et après chaque sauvegarde de fichiers .scss ou .sass...

Source(s) :

- <https://marketplace.visualstudio.com/items?itemName=glenn2223.live-sass>

SASS/SCSS - Compilation



...nos fichiers sont compilés en CSS

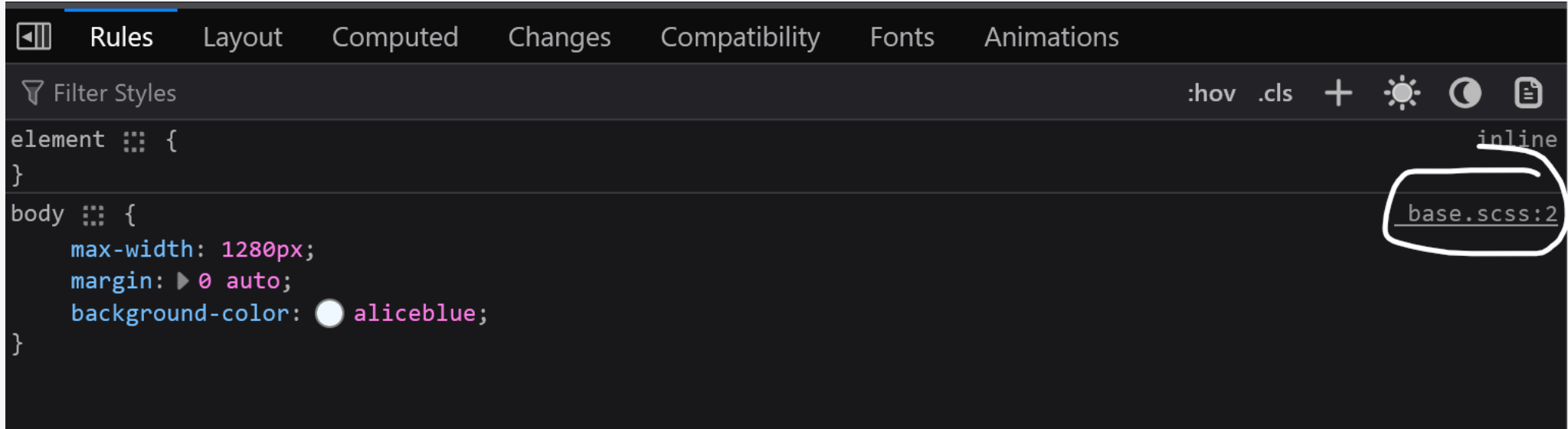
Source(s) :

- <https://marketplace.visualstudio.com/items?itemName=glenn2223.live-sass>

***.css.map**

- Permet d'indiquer où se trouve le code CSS compilé dans les fichiers SCSS
- N'a pas besoin d'être commité via git
 - Tout comme le fichier CSS compilé

SASS/SCSS - Compilation



Dans le navigateur, le source map permet de savoir dans quel fichier SCSS se trouvent nos sélecteurs CSS

Pratiquons ! - Découvrons SASS/SCSS

Pré-requis :

- Avoir la ressource ressources/scss

A télécharger ici :_

https://github.com/DanYellow/cours/raw/refs/heads/main/integration-web-s3/cours-magistraux/numero-1/integration-web-s3_cours-magistraux_numero-1.ressources.zip

Questions ?

