

# Intégration Web

MMI 1 – TP#3 S2





Danielo **JEAN-LOUIS**  
Développeur front-end

# Pratiquons !

## Pré-requis :

- Avoir la ressource "preprocesseur/"

## Consignes :

- Ouvrir le fichier index.html dans votre navigateur
- Ouvrir le projet dans VS Code

## Sources :

- <https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s2/travaux-pratiques/numero-3/ressources/preprocesseur>



**Ça fonctionne mais...**

```
.liste-emissions .lien {  
  padding: 10px;  
  display: inline-block;  
  margin: 10px;  
  color: rebeccapurple;  
  font-weight: bold;  
  font-size: 1.3rem;  
}
```

```
.liste-emissions .description {  
  padding: 10px;  
}
```

```
.liste-emissions .hl {  
  font-size: 1.6rem;  
  position: relative;  
  padding: 5px 8px;  
  top: -20px;  
  left: 5px;  
  color: white;  
  background-color: rebeccapurple;  
  display: inline-block;  
  margin-bottom: -5px;  
}
```

```
.liste-emissions li {  
  flex: 0 0 49%;  
  margin-top: 25px;  
}
```

```
.liste-emissions li:last-child {  
  margin-right: auto;  
}
```



**Ça fait beaucoup de ".liste-emissions", non ?**

## Préprocesseurs CSS

- Méta-langages CSS
- Ne sont pas lus par les navigateurs
- **Doivent être compilés en CSS**
- Facultatifs mais indispensables
  - 99,99 % des projets utilisent un préprocesseur CSS
- Existe une multitude de préprocesseurs CSS
  - Le plus populaire : sass/scss – On l'utilisera

### Sources :

- <https://www.alsacreations.com/article/lire/1717-les-preprocesseurs-css-c-est-sensass.html>
- [https://fr.wikipedia.org/wiki/Pr%C3%A9processeur\\_CSS](https://fr.wikipedia.org/wiki/Pr%C3%A9processeur_CSS)
- <https://grafikart.fr/tutoriels/differences-sass-scss-329>
- <https://la-cascade.io/se-lancer-dans-sass/>
- <https://sass-lang.com/>

## Préprocesseurs CSS

- "Même" syntaxe que le CSS – ça va dépendre du préprocesseur utilisé
- Apportent de nouvelles fonctionnalités
  - Imbrication de sélecteurs
  - Conditions / boucles
  - Import de fichiers\*
  - Fonctions
  - **Variables compilées** – Elles ne sont pas modifiables après coup
- Equivalent HTML : PHP
- Les fonctionnalités peuvent également couler un projet : **Attention**



## Préprocesseurs CSS - Compilation

- Utilisation d'un serveur node (méthode la plus utilisée dans le milieu pro)
  - Voir le projet d'architecture posté sur Teams
- Compilation en ligne
- Extension d'éditeur de code
  - Nous allons utiliser cette méthode

### Sources :

- Projet archi front-end : <https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s2/outils/archi-fe>

## Préprocesseurs CSS - Compilation

- Ne jamais, jamais, jamais modifier le fichier CSS généré à la compilation
- Overkill pour un petit projet (sauf ce TP)

### Sources :

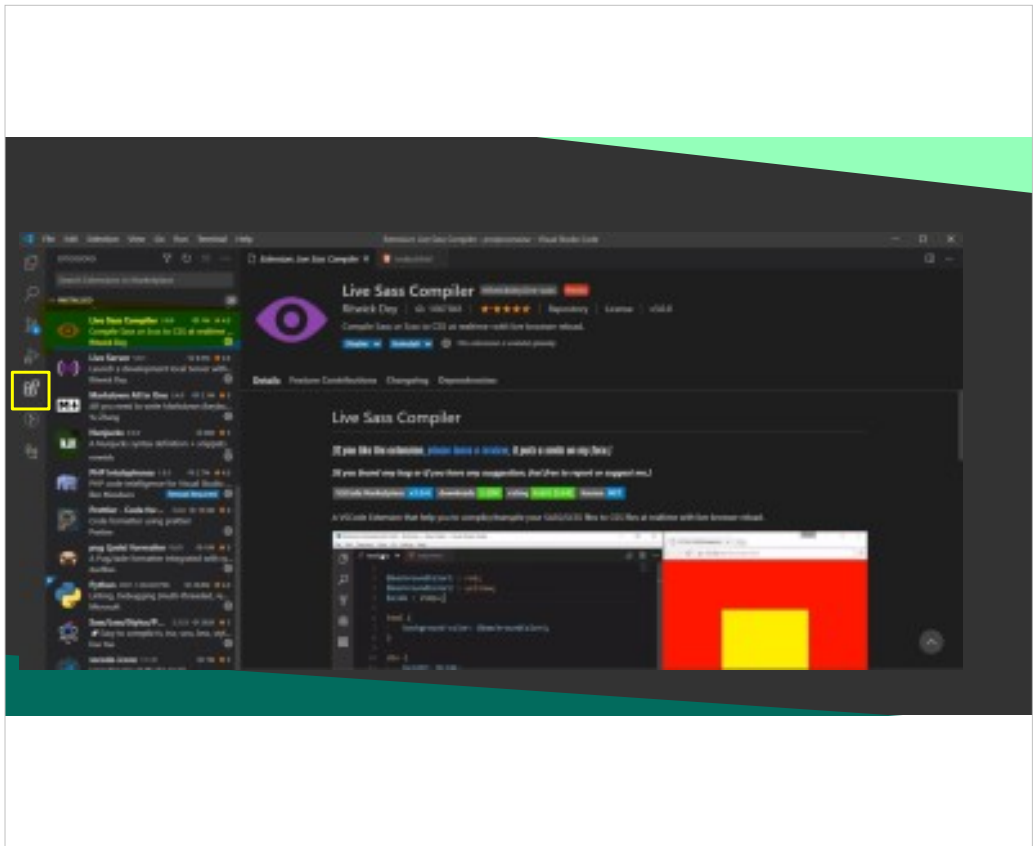
- <https://iamvdo.me/blog/pourquoi-je-nutilise-pas-les-preprocesseurs-css>

## **VS Code – Compilation SCSS/SASS Extension Live Sass Compiler**

- Permet de compiler le fichier .scss/.sass en .css
- Installe par la même occasion l'extension "Live Server"
  - Permet de recharger votre navigateur après chaque modification de fichiers



## **Installons l'extension Live Sass Compiler**



## Pratiquons ! - Écrivons le projet en SASS

Pré-requis :

- Avoir la ressource "preprocesseur/"

Consignes :

- Copier le dossier de la ressource pour avoir un "master"
- Réécrire le fichier "style.css" en SCSS ou SASS
- Rajouter une pseudo-classe ":hover" + attr css pour la classe ".lien"

Sources :

- <https://downgit.github.io/#/home?url=https://github.com/DanYellow/cours/tree/main/integration-web-s2/travaux-pratiques/numero-3/ressources/preprocesseur>

**Questions ?**

