

# LES PRÉPROCESSEURS SASS & SCSS

---

**P/2** Introduction

**P/3** Le processus

**P/4** SASS & SCSS

- Installation :
- Configuration avec l'IDE
- Variables
- Nesting
- Opérateurs
- Mixins
- Fonctions
- Extends
- Conditions

**P/11** Documentation

**P/12** Crédits

---

## Introduction

---

**SASS** est un préprocesseur **CSS**.

C'est un langage de description compilé en **CSS**

Le **CSS** est limité contrairement à des langages de programmation où l'on peut créer des variables, des instructions...

Le **CSS** impose aussi d'écrire les choses de façon redondante et il engendre beaucoup de répétitions.

Or comme vous le savez, les développeurs n'aiment pas réécrire plusieurs fois les mêmes choses (Don't repeat yourself).

Les préprocesseurs **CSS** sont donc là pour pallier à ces manques.

## Le processus

---

Les navigateurs ne comprennent pas le code **SASS**, il sera compilé par la suite en **CSS** pour le rendre compréhensible.



## SASS & SCSS

---



**SCSS** est identique au **SASS** mais il dispose d'accolade et de point virgule.

L'écriture du **SCSS** est privilégiée par les développeurs par rapport au **SASS**

SASS :

```
$font-stack: Helvetica, sans-serif
$primary-color: #333

body
  font: 100% $font-stack
  color: $primary-color
```

SCSS :

```
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

## Installation :

---

Télécharger les packages via Github, ajouter également à la variable **PATH** (variable d'environnement)

**Dépôt Github.**

Ou

Utilisez Node Package Manager (**NPM**).

**NPM SASS**

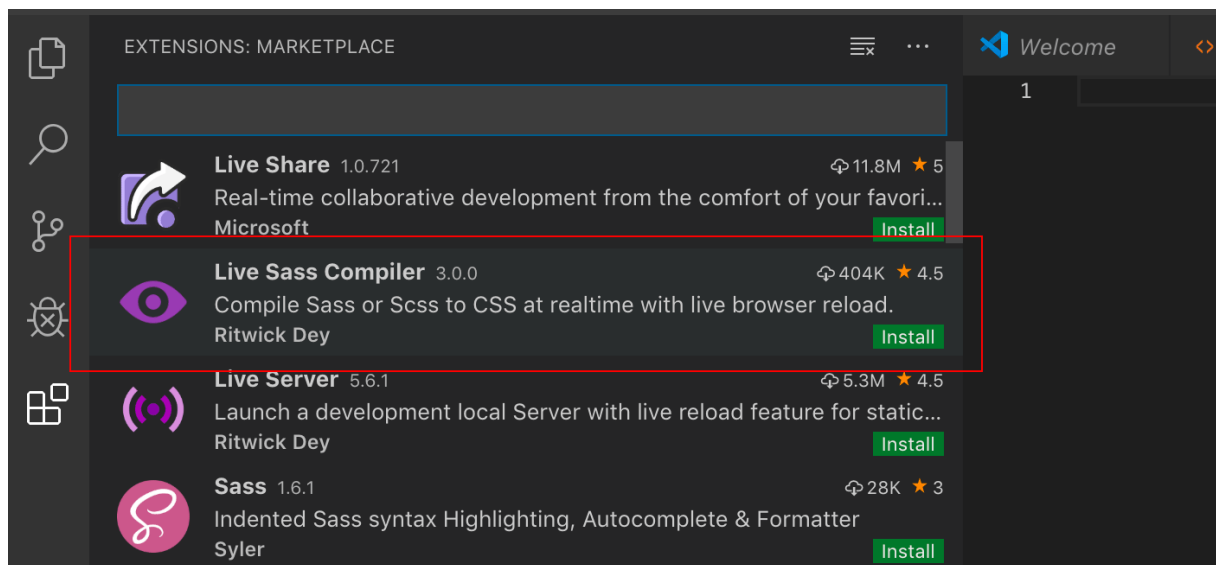
**npm install -g sass**

## Configuration avec l'IDE

---

Avec **Visual Studio Code**, installez un compilateur **Sass** dans l'onglet "**extensions**".

Installez le plugin "**sass live compiler**".



## Variables

Elles permettent de stocker des couleurs, tailles, etc...

**CSS :**

```
.alert {  
  border: 1px solid rgba(198, 83, 140, 0.88);  
}
```

**SCSS :**

```
$border-dark: rgba($base-color, 0.88);  
  
.alert {  
  border: 1px solid $border-dark;  
}
```

## Nesting

Permet d'imbriquer plus intuitivement des éléments les uns dans les autres.

Exemple : avec les éléments **ul**, **li** & **a** d'une balise **<nav>**

**CSS :**

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav li {  
  display: inline-block;  
}  
nav a {  
  display: block;  
  padding: 6px 12px;  
  text-decoration: none;  
}
```

SCSS :

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```

Le code est clairement plus lisible en SCSS.

## Opérateurs

---

Les opérateurs "+, -, \*, /, %" permettent de faire des calculs en CSS.

CSS :

```
.container {  
  width: 100%;  
}  
  
article[role="main"] {  
  float: left;  
  width: 62.5%;  
}  
  
aside[role="complementary"] {  
  float: right;  
  width: 31.25%;  
}
```

SCSS :

```
.container {  
  width: 100%;  
}  
  
article[role="main"] {  
  float: left;  
  width: 600px / 960px * 100%;  
}  
  
aside[role="complementary"] {  
  float: right;  
  width: 300px / 960px * 100%;  
}
```

## Mixins

---

Les mixins permettent d'écrire des morceaux de codes réutilisables.

CSS :

```
.box {  
  -webkit-transform: rotate(30deg);  
  -ms-transform: rotate(30deg);  
  transform: rotate(30deg);  
}
```

SCSS :

```
@mixin transform($property) {  
  -webkit-transform: $property;  
  -ms-transform: $property;  
  transform: $property;  
}  
  
.box { @include transform(rotate(30deg)); }
```

## Fonctions

---

Permet de réaliser des fonctions comme dans les autres langages de programmation.

SCSS :

```
@function pow($base, $exponent) {  
  $result: 1;  
  @for $_ from 1 through $exponent {  
    $result: $result * $base;  
  }  
  @return $result;  
}  
  
.sidebar {  
  float: left;  
  margin-left: pow(4, 3) * 1px;  
}
```

CSS :

```
.sidebar {  
  float: left;  
  margin-left: 64px;  
}
```



## Extends

---

Permet d'étendre les propriétés d'un élément.

**SCSS :**

```
/* This CSS will print because %message-shared is extended. */
%message-shared {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

// This CSS won't print because %equal-heights is never extended.
%equal-heights {
  display: flex;
  flex-wrap: wrap;
}

.message {
  @extend %message-shared;
}

.success {
  @extend %message-shared;
  border-color: green;
}

.error {
  @extend %message-shared;
  border-color: red;
}

.warning {
  @extend %message-shared;
  border-color: yellow;
}
```

**CSS :**

```
/* This CSS will print because %message-shared is extended. */
.message, .success, .error, .warning {
  border: 1px solid #ccc;
  padding: 10px;
  color: #333;
}

.success {
  border-color: green;
}

.error {
  border-color: red;
}

.warning {
  border-color: yellow;
}
```

## Conditions

---

**Sass** permet d'utiliser des conditions tel que "if", "for", "while", etc...

```
1  $light: #eee;
2  $dark: #333;
3
4  @mixin theme($light-theme: true){
5      @if($light-theme){
6          color: $dark;
7          background-color: $light;
8      }@else{
9          color: $light;
10         background-color: $dark;
11     }
12 }
13
14 body{
15     @include theme($light-theme: true);
16
17     span{
18         @include theme($light-theme: false);
19     }
20 }
```

## Documentation

---

Parcourez la documentation officielle pour approfondir vos connaissances :

[SASS DOCUMENTATION](#)

## Crédits

---

### OEUVRE COLLECTIVE DE L'AFPA

Sous le pilotage de la Direction de l'ingénierie

### DATE DE MISE À JOUR

17/09/2020

© AFPA

#### **Reproduction interdite**

*Article L 122-4 du code de la propriété intellectuelle.*

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconques. »