



Développement Web Côté client

Module 11 - Préprocesseurs SASS



Objectifs

Développement Web côté client
PRÉPROCESSEUR SASS



Découvrir SASS



Introduction

SASS



Définition :

SASS : Syntactically Awesome Style Sheets

Rôles :

Langage de script préprocesseur compilé ou interprété

Définition :

Un préprocesseur est un programme qui procède à des transformations sur un code source avant la compilation ou l'interprétation

Installation

SASS



Sass

Installation

Développement Web côté client
PRÉPROCESSEUR SASS

1. Installation de node.js
2. Installation de SASS via npm

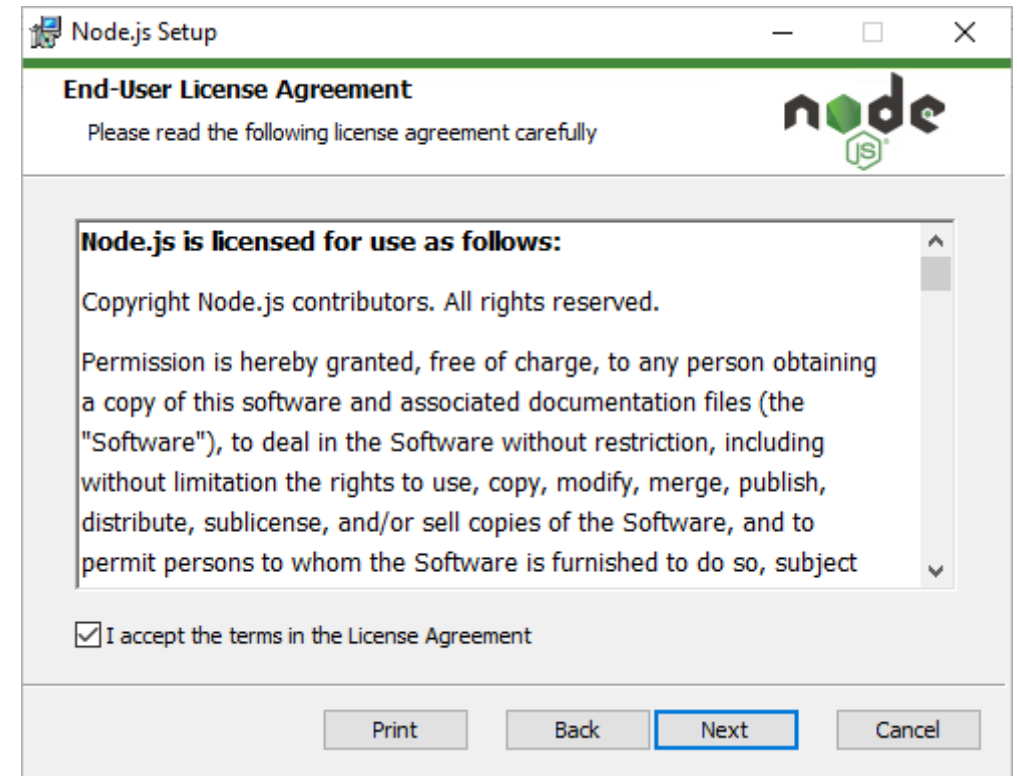
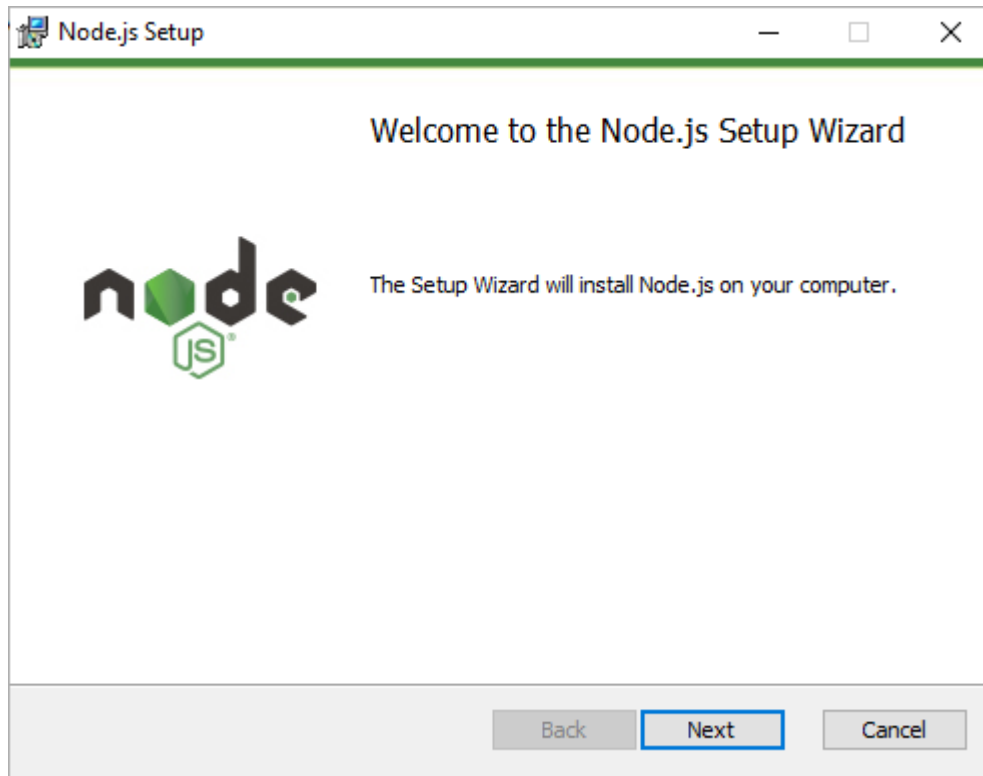


Sass

Installation

Node.js

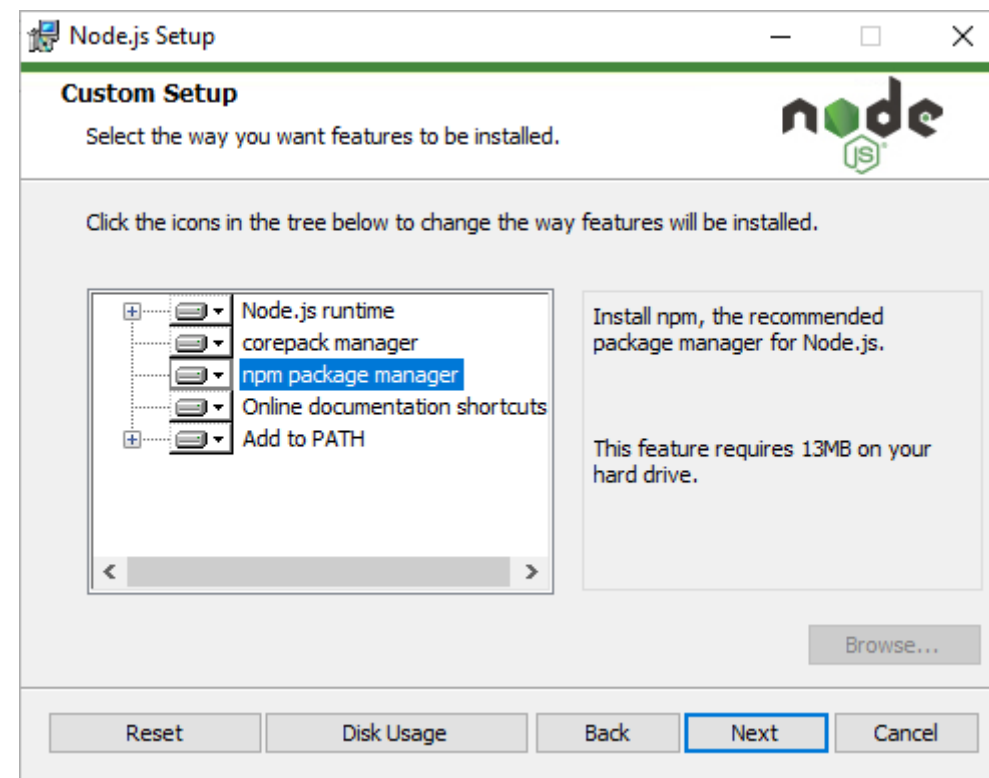
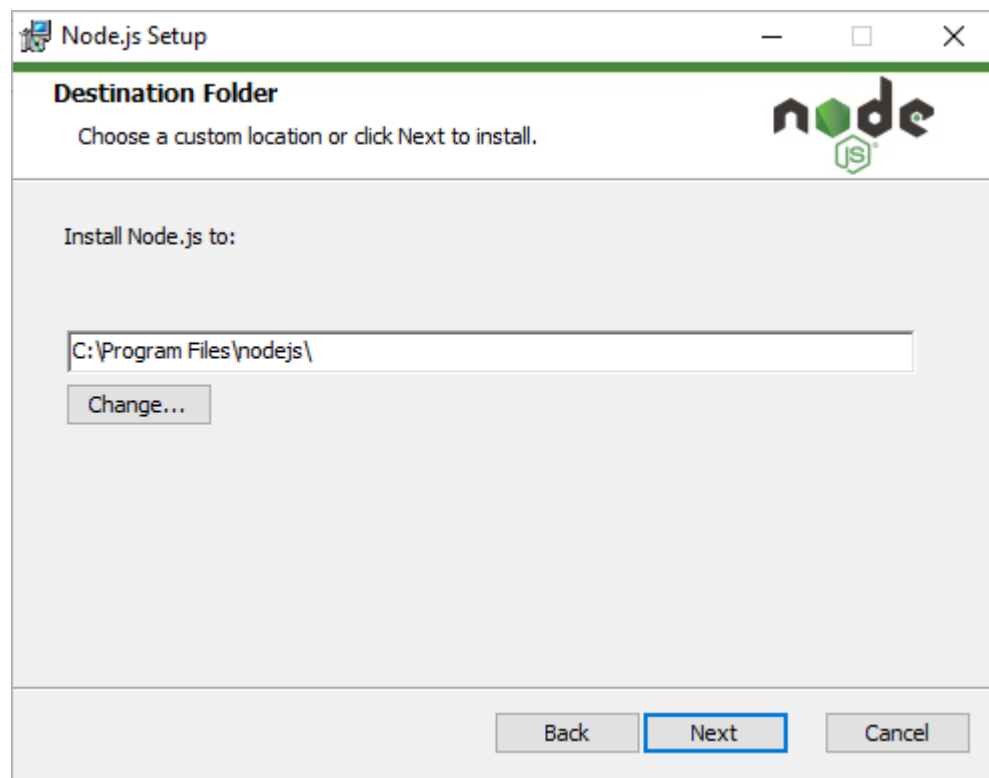
- Télécharger et installer la dernière version **LTS** de Node.js depuis le site <https://nodejs.org/en/>



Sass

Installation

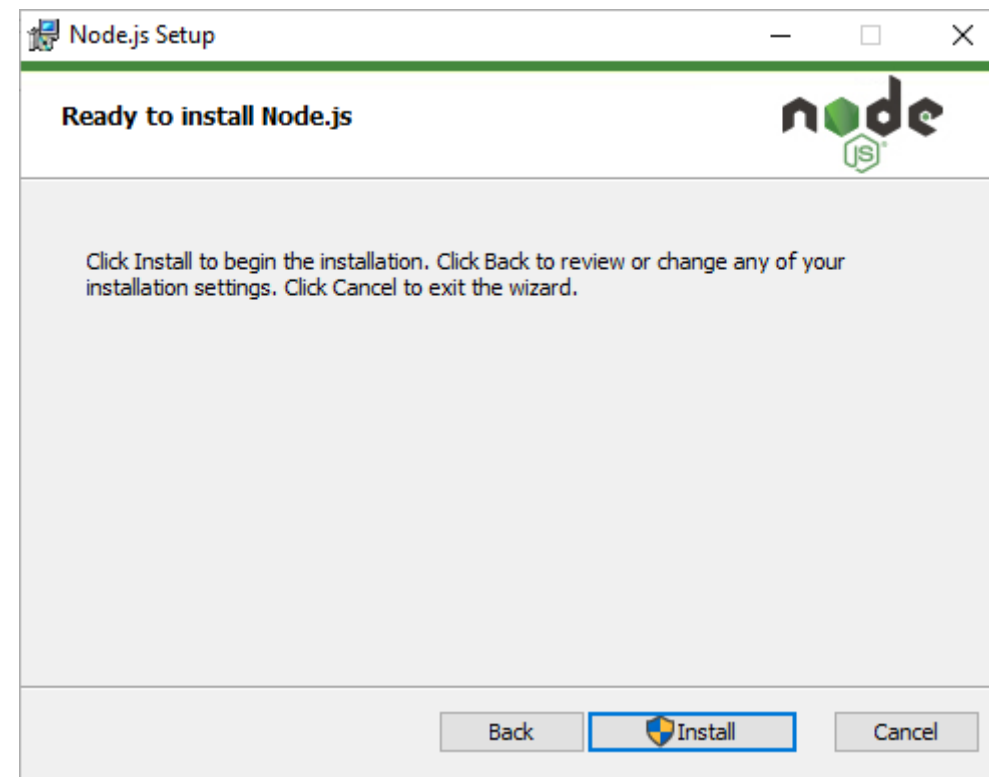
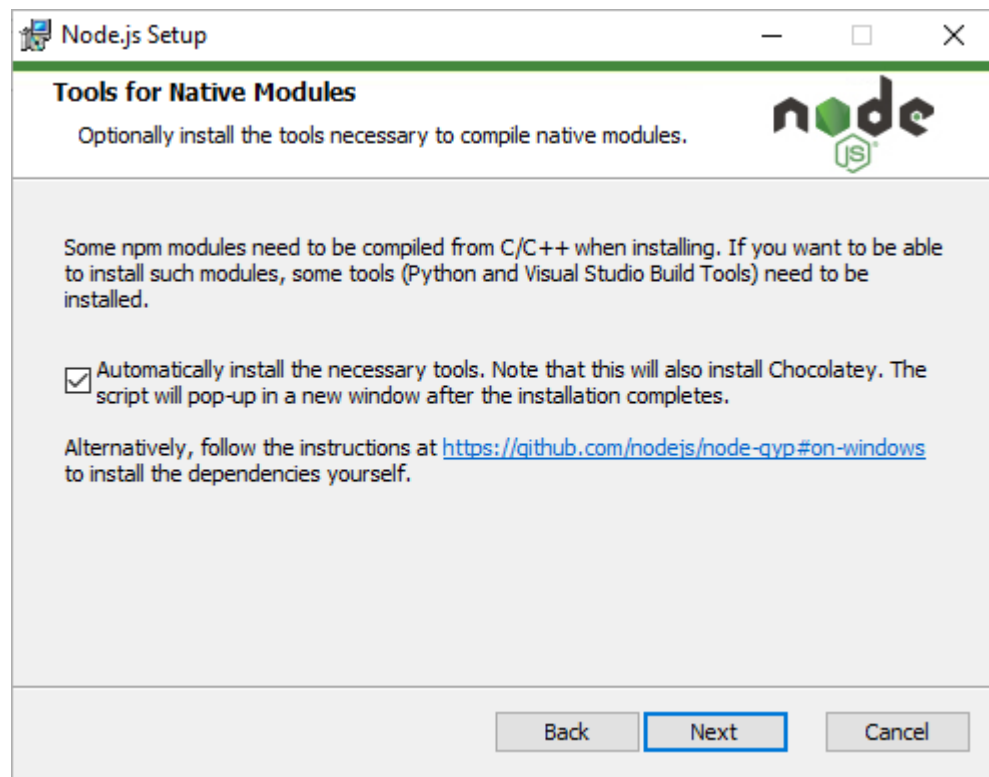
Développement Web côté client
PRÉPROCESSEUR SASS



Sass

Installation

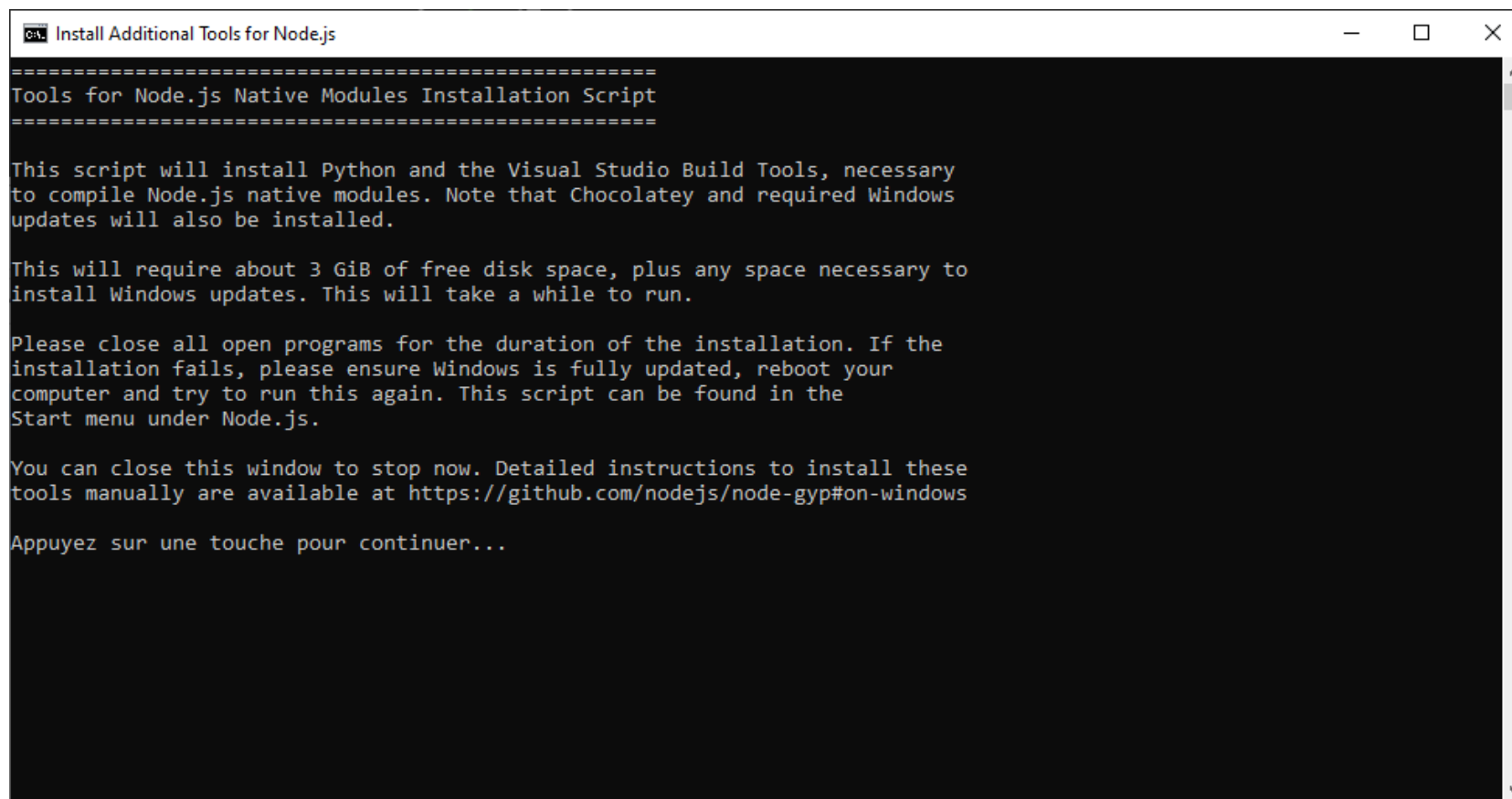
Développement Web côté client
PRÉPROCESSEUR SASS



Sass

Installation

Développement Web côté client
PRÉPROCESSEUR SASS



```
=====  
Tools for Node.js Native Modules Installation Script  
=====
```

This script will install Python and the Visual Studio Build Tools, necessary to compile Node.js native modules. Note that Chocolatey and required Windows updates will also be installed.

This will require about 3 GiB of free disk space, plus any space necessary to install Windows updates. This will take a while to run.

Please close all open programs for the duration of the installation. If the installation fails, please ensure Windows is fully updated, reboot your computer and try to run this again. This script can be found in the Start menu under Node.js.

You can close this window to stop now. Detailed instructions to install these tools manually are available at <https://github.com/nodejs/node-gyp#on-windows>

Appuyez sur une touche pour continuer...

Sass

- L'installation se fait en ligne de commande
- Ouvrir le terminal et saisir :

```
npm install -g sass
```

- La vérification de l'installation se fait en ligne de commande
- Toujours dans le terminal, saisir :

```
sass --version
```

Utilisation Sass

La compilation

- Il est possible de travailler avec deux types de fichiers dont la syntaxe reste assez proche :
 - .scss (plus répandu)
 - .sass
- Les navigateurs ne peuvent pas interpréter ces fichiers. Ils doivent être compilés en format .css
- La compilation (*Preprocessing*) se fait en ligne de commande :

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

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

Le flag **--watch** permet de recompiler automatiquement le fichier (si modification du code) à chaque sauvegarde

Sass

Variables

- Il est possible de déclarer des variables
- Elles se déclarent généralement en début de fichier et sont préfixées de **\$**
- L'appel des variables se fait par leur nom
- Exemple :

```
$maPolice: sans-serif;  
$maCouleur1: #132;  
$maCouleur2: rgb(223, 155, 88);  
  
body {  
    background-color: $maCouleur1;  
    color: $maCouleur2;  
    font-family: $maPolice;  
}
```

Imbrications

- En css, il faut souvent multiplier les sélecteurs afin d'être précis.
- Par exemple, pour un **article** qui comporte un titre **h1** et un paragraphe **p**, nous aurons :

```
article { //some code }  
article h1{ //some code }  
article p { //some code }
```

- Avec **sass**, il est possible d'écrire :

```
article {  
  background-color: red;  
  
  h1 {  
    color: white;  
  }  
  
  p {  
    color: black;  
  }  
}
```


DÉMONSTRATION



- Afin de ne pas centraliser tout le code dans un seul fichier, il est possible de créer des **partials**, c'est-à-dire des sous-fichiers
- Ils seront ensuite appelés avec l'annotation **@use**
- Leur nom doit être précédé d'un **underscore**. Ex. `_comp.scss`

```
//_comp.css
$color-perso: white;
$background-perso: rgba(0, 0, 0, 0.4);

body {
  background-color: $background-perso;
  color: $color-perso;
}
```

```
//principal.scss
//Appel de _comp.scss
@use 'comp';

div {
  //Appel d'une variable de comp
  background-color: comp.$color-perso;
  color: green;
}
```



```
//principal.css
body {
  background-color: rgba(0, 0, 0, 0.4);
  color: white;
}

div {
  background-color: white;
  color: green;
}
```

DÉMONSTRATION

Modules et partials



- Le **mixin** est une notion avancée qui combine plusieurs avantages que l'on retrouve dans les fonctions et les héritages
- Ils se mettent en place par la directive **@mixin**
- L'appel se fait ensuite avec **@include**
- Ils peuvent être contenu dans un partial

```
@mixin theme($couleur: grey) {  
  background-color: $couleur;  
  border: 1px solid black;  
  box-shadow: 0 0 3px rgba($couleur, 0.3);  
}  
  
.success {  
  @include theme($couleur: green);  
}  
  
.normal {  
  @include theme;  
}
```

DÉMONSTRATION

UPDATE
Mixin



- L'héritage permet de transmettre des informations
- La syntaxe pour mettre en place un héritage est **%nomHéritage**
- L'appel s'effectue avec **@extend**

```
%alert-config {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
  
.alert {  
  @extend %alert-config;  
}  
  
.success {  
  @extend %alert-config;  
  border-color: green;  
}
```


DÉMONSTRATION

Héritage

Sass permet la réalisation de calcul

- Les opérateurs `+`, `-`, `*`, `math.div()` et `%` sont à disposition
- Pour la division, il faut importer la librairie math : `@use 'sass:math'`

Sass permet la mise en place de conditions

- Opérateurs de comparaison : `==` `!=` `>` `>=` `<` `<=`
- Opérateurs logiques : `And` `Or` `Not`
- Directive `@if(condition) {} @elseif {} @else {}`

Sass permet la mise en place des boucles

- **@for**
- **@while**
- **@each**

DÉMONSTRATION

Boucles et conditions



Sass permet l'écriture de fonctions

- La mise en place se fait avec la directive **@fonction**
- Si l'exécution prévoit un retour : **@return**
- L'appel se fait par le nom simplement
- Il existe aussi un certain nombre de packages SASS qui fournissent diverses fonctions prêtes à l'emploi :
 - <https://sass-lang.com/documentation/modules>
 - L'import de ces modules se fait avec la directive **@use**

DÉMONSTRATION

Fonctions

**TRAVAUX
PRATIQUES**

Préprocesseur CSS - Sass

CONCLUSION

Vous avez découvert SASS