

Formation SASS

Syntactically **A**wesome **S**tyle **S**heets

Objectif : réalisation d'un site statique en mobile first



Intro

SASS est à la fois un préprocesseur et un langage de feuille de style.

SASS est une extension de CSS qui ajoute puissance et élégance au langage de base.

Avant que SASS puisse être lu, le code source doit être compilé en CSS ordinaire.

SCSS (*Sassy CSS*) est une version plus récente de SASS, qui elle est plus proche de l'écriture traditionnelle des CSS.

Installer SASS

Pour installer SASS sur votre système, ouvrez le terminal ou l'invite de commande et entrez la commande suivante:

sudo gem install sass

Configurer la compilation

Ici on va travailler avec le task runner **gulp**

<https://css-tricks.com/gulp-for-beginners/>

Vérifier si gulp est installé sur votre ordinateur :

gulp -v

Installer gulp (globalement -g) dans le terminal Linux

```
npm install -g gulp
```

```
npm install gulp gulp-sass
```

Terminal > Configure Tasks and click Create tasks.json file from template

Copier dans task.json

```
// Sass configuration
var gulp = require('gulp');
var sass = require('gulp-sass')(require('sass'));
var autoprefixer = require('gulp-autoprefixer');

gulp.task('sass', function(cb) {
  gulp
    .src('*.scss')
    .pipe(sass())
    .pipe(autoprefixer())

    .pipe(
      gulp.dest(function(f) {
        return f.base;
      })
    );
  cb();
});

gulp.task(
  'default',
  gulp.series('sass', function(cb) {
    gulp.watch('*.scss', gulp.series('sass'));
    cb();
  })
);
```

Modifier src destination

`liveSassCompile.settings.formats`

Settings --> Select "Live Sass Compiler Config --> add

Compilation

SASS est écrit en Ruby et distribué via son gestionnaire de paquets [RubyGems](#).

Pour le compiler, on peut se servir de kits préinstallés comme [Koala](#) et [CodeKit](#),

ou **passer via la ligne de commande de la manière suivante :**

`sass --watch example.sass:example.css` *pour surveiller le fichier*

`sass --watch sass:css` *pour surveiller tout le dossier*

Ou

Dans VS Code utiliser le bouton **watching** dans la barre des tâches de votre IDE

La syntaxe

Il existe deux syntaxe, SASS et la plus récente SCSS

SASS est sans accolade, dites imbriquée

SASS

```
$bg-color: #df0174
$size: 1em
body
background-color: $bg-color
margin: $size * 2
```

SCSS avec accolade est plus proche du CSS

SCSS

```
$bg-color: #df0174;
$size: 1em;
body {
background-color: $bg-color;
margin: $size * 2;
}
```

Les variables

Une variable peut être utilisée pour stocker la valeur hexadécimale d'une couleur ou des fonctions mathématiques.

Les variables sont introduites dans SASS avec le signe dollar (\$):

Elles sont déclarées de manière globale au début du fichier.

```
$bg-color: #df0174;  
$size: 1em;
```

Ensuite on les utilise dans le code comme ceci :

```
body {  
    background-color: $bg-color;  
    margin: $size * 2;  
}
```

Une fois compilé le fichier CSS sortira :

```
body {  
    background-color: #df0174;  
    margin: 2em;  
}
```

Importer des partiels

Les partiels sont des fragments de code dans des fichiers que l'on importe dans le fichier principal.

Le nom du fichier(partial) est précédé d'un **underscore**.

```
_variables.scss  
_font.scss  
_mixins.scss
```

Dans le fichier principal on importe les partiels comme suit :

```
@import 'variables';  
@import 'font';  
@import 'mixin';
```

Notez que l'on omet l'underscore et l'extension lors de l'importation

Les mixins

Ce sont des règles fixes que vous pouvez rappeler encore et encore dans la feuille de style sans avoir à réinsérer le code complet.

Cela aide à **travailler plus rapidement et à garder le code plus simple**.

```
@mixin big-blue-text {
    font-family: Arial;
    font-size: 25px;
    font-weight: bold;
    color:#0000ff;
}

@mixin flexRow ($display, $direction) {
    display: $display;
    -webkit-flex-direction: $direction;
    -ms-flex-direction: $direction;
    flex-direction: $direction;
}

ul {
    @include flexRow(flex, row);
}
```

Dans le code CSS final :

```
ul {
    display: flex;
    -webkit-flex-direction: row;
    -ms-flex-direction: row;
    flex-direction: row;
}
```

Étendre (@extend)

La règle d'extension vous épargne beaucoup de travail.

Cette directive garantit que toutes les propriétés d'une classe sont transmises à une autre.

Pour éviter de tout redéfinir, utilisez « @extend ». La directive fonctionne également comme une chaîne.

Une classe définie par « @extend » peut faire partie d'une troisième classe :

```
.button-scope {  
  margin: 5px;  
  border-radius: 2px;  
}  
  
.home-button {  
  @extend .button-scope;  
  background-color: $black;  
}  
  
.back-button {  
  @extend .home-button;  
}
```

Tester dans sassmeister.com

Les effets de mixins et de « @extend » sont très similaires.
Un article détaillé sur les différences peut être trouvé sur csswizardry.com.

Dans la plupart des cas, il est préférable d'utiliser les @mixin

Le nesting

En HTML les balises sont imbriquées.

CSS ignore cela alors que SASS l'utilise pour une meilleure lisibilité de votre code

```
a {
  color: $blue;
  &:visited {
    color: $red;
  }
  &:hover {
    color: $purple;
  }
}

.header-center {
  text-align: center;
  padding: 0;
  figure {
    min-width: 130px;
    &:before {
      content: "";
      position: absolute;
      top: -6px;
      left: 50%;
      transform: translateX(-50%);
      height: 120px;
      width: 180px;
      background-color: white;
      z-index: -1;
    }
    &::after {
      content: "";
      position: absolute;
      top: 0;
      left: 50%;
      transform: translateX(-50%);
      min-height: 140px;
      min-width: 140px;
      border-bottom: 1px solid #b1b1b1;
      border-bottom-left-radius: 50%;
      background-color: white;
      z-index: -2;
    }
  }
}
```

Il ne faut pas surexploiter cette possibilité d'imbrication,
3 imbrications maximum pour garder votre code maintenable.

Les commentaires

Il existe deux méthodes.

```
/*
```

Ceci est un commentaire pour un bloc de code

```
*/
```

```
// Ceci est une ligne commentée
```

<http://sass.js.org/>

<https://www.sassmeister.com/>

VS Code sass

<https://code.visualstudio.com/Docs/languages/CSS>

Grid

<https://mozilladevelopers.github.io/playground/css-grid/>

Flexbox

<https://progressived.com/fla/?d=3&v=5&h=0&s=0&i=010&a=000>

Sprite

<https://www.alsacreations.com/tuto/lire/1068-sprites-css-background-position.html>

Flag en CSS

<https://pixelastic.github.io/css-flags/>

Z-index

<https://la-cascade.io/comment-fonctionne-z-index/>