SASS

Syntactically Awesome StyleSheets

Langage préprocesseur

Un préprocesseur est un programme qui intervient avant la compilation.
Le sass sera compilé puis générer en fichiers css.

- ➔ Simplifier le code
- ➔ écrire plus rapidement et des fonctions plus avancé
- ➔ rendre les feuilles de style plus maintenables

Installations

Ruby guide installation

La syntaxe

Il existe deux syntaxes différentes pour Sass :

- la **syntaxe SCSS** se veut proche de celle du CSS, rajoutant principalement l'imbrication des blocs. Les fichiers portent l'extension ".scss". (introduit à la version 3,0)
- la syntaxe SASS se veut plus épurée, en retirant les accolades et les point-virgules. Les fichiers portent l'extension ".sass".

SASS SYNTAX Sass SCSS CSS

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

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

SCSS SYNTAX Sass SCSS CSS

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

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

Les variables

➔ **stocker des valeurs**

➔ **réutiliser dans plusieurs feuilles de style**

Créer un fichier _vars.scss séparé contenant toutes vos variables et l'inclure dans les autres fichiers grâce à la ligne de code :

@import "vars";

```
$blue: #087CC9;
$blueception: $blue;
$path: './images/background.png';
$fonts-default: Helvetica, serif;
$margin: 13px;
$debug: true;
$inspiration: null;
$idList: home-container, logo, tuto;
$articlesMap: (angularjs: tomato, sass: rgb(204,102,153), browserify: #3c6991);
$wtf: 1.7em, false, logo, #CD4F39, Helvetica, null, $blue;
```

Les mixins

➔ **Les « mixins » sont des morceaux de codes paramétrables et réutilisables. Elles se déclarent avec le mot clé « @mixin » et s'utilisent avec le mot clé « @include ».**

➔ **media-queries**

```
@mixin transform($property) {
  -webkit-transform: $property;
  -ms-transform: $property;
  transform: $property;
}

.box { @include transform(rotate(30deg)); }
```

Le Nesting

Une hiérarchie plus intuitive

CSS OUTPUT Sass SCSS 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 SYNTAX

Sass SCSS CSS

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }

  li { display: inline-block; }

  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

Les opérations

possibilité d'additionner, soustraire, multiplier, divisé et%

SCSS SYNTAX Sass SCSS CSS

```scss
.container {
  width: 100%;
}

article[role="main"] {
  float: left;
  width: 600px / 960px * 100%;
}

aside[role="complementary"] {
  float: right;
  width: 300px / 960px * 100%;
}
```

We've created a very simple fluid grid, based on 960px. Operations in Sass let us do something like take pixel values and convert them to percentages without much hassle.

CSS OUTPUT

Sass SCSS CSS

```
.container {
  width: 100%;
}

article[role="main"] {
  float: left;
  width: 62.5%;
}

aside[role="complementary"] {
  float: right;
  width: 31.25%;
}
```

We've created a very simple fluid grid, based on 960px. Operations in Sass let us do something like take pixel values and convert them to percentages without much hassle.

http://compass-style.org

https://www.supinfo.com/articles/single/6187-bien-commencer-avec-sass

https://medium.com/accessmentorat/sass-le-css-avec-de-supers-pouvoirs-f6d5ef67bd40

https://sass-lang.com/documentation/file.SASS_REFERENCE.html#syntax

https://medium.com/swlh/advantages-of-using-a-preprocessor-sass-in-css-eb7310179944

https://medium.freecodecamp.org/how-to-get-better-at-writing-css-a1732c32a72f