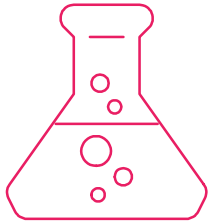
Syntactically Awesome Style Sheets

POURQUOI Sass ?

Le CSS est trop simple.

- Imbrication impossible
- Obligation de tout réécrire
- Modification des couleurs avec CTRL+F
- ...



UTILISATION DE SASS

SASS n’est pas reconnu automatiquement par le navigateur.

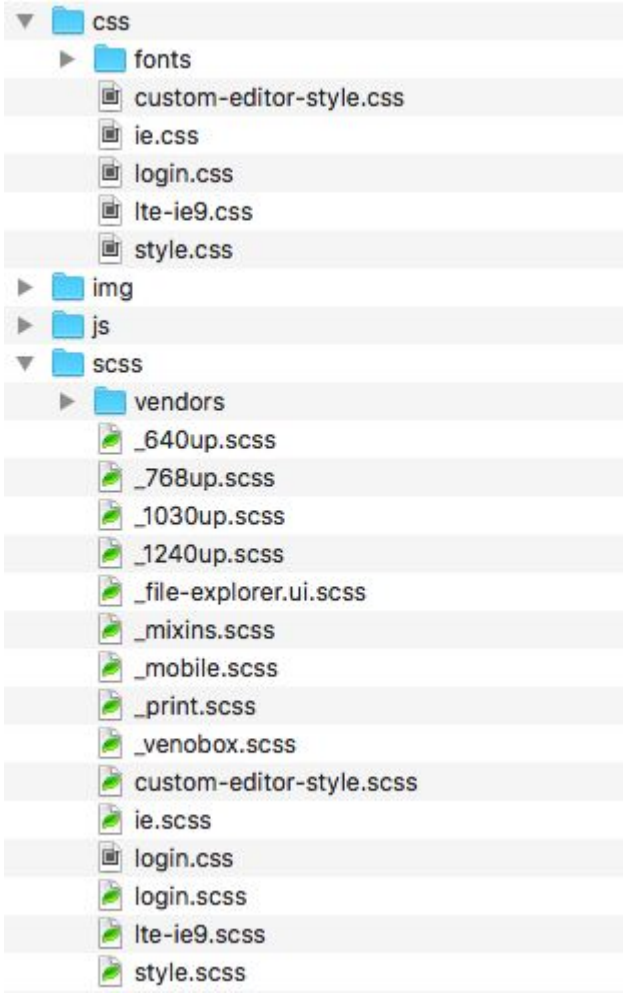
Il faut utiliser un compilateur SASS qui va nous générer notre CSS lisible et interprétable par le navigateur.

PARTIELS / IMPORT

Fichiers CSS ou SASS séparés qu'on importe avec **`@import`**

```
@import "custom.css";
@import "file.scss";
@import "file"
@import "_partial.scss";
@import "_partial";
```

Pour distinguer les partiels des autres fichiers, on préfixe leur nom avec un underscore. Par exemple _partial.scss.



Structure des fichiers d'un projet utilisant SASS

Identifie les "partials" et les fichiers qui seront réellement compilés.

Combien y a-t-il de stylesheets dans ce projet ?

Peux-tu déduire des noms des fichiers partiels ce que contient chacun ?

Avantage des partiels => code css modulaire et plus facilement réutilisable d'un projet à l'autre.
Et donc, une chambre bien rangée.

Exemple de ce que cela permet ...

Contenu de style.scss

```
@import 'mobile';

@media only screen and (min-width: 640px) {
    @import "640up";
}
@media only screen and (min-width: 768px) {
    @import "768up";

}
@media only screen and (min-width: 1024px) {
    @import "1030up";
}
@media only screen and (min-width: 1240px) {
    @import "1240up";
}
@media print {
    @import "print";
}
```

... Donc du mobile-first , *easy peasy !*

VARIABLES

SASS

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


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

INDENTATION

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;
}
```

SASS

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

  li { display: inline-block;
}

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

MIXINS

CSS

```
.box {
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
  -ms-border-radius: 10px;
  border-radius: 10px;
}
```

SASS

```
@mixin border-radius( $radius) {
  -webkit-border-radius:  $radius;
     -moz-border-radius:  $radius;
      -ms-border-radius:  $radius;
          border-radius:  $radius;
}

.box { @include border-radius(10px); }
```

HÉRITAGE

```
.sassExParent {
background: #fdd ;
width: 97% ;
}
.sassExEnfant{
@extend .sassExParent ;
color: white ;
}
```

SELECTEUR PLACEHOLDER

% permet de créer un sélecteur SASS qui n'est pas pris en compte pendant la compilation

SASS

```
%btn {
    Padding: 10px;
    Margin: 0;
}
.btn-blue {
    @extend %btn;
    background-color: blue;
}
.btn-red {
    @extend %btn;
    background-color: red;
}
```

CSS

```
.btn-blue,
.btn-red {
    Padding: 10px;
    Margin: 0;
}

.btn-blue {
    background-color:
blue;
}
.btn-red {
    background-color: red;
}
```

APPEL À CLASSE IMBRIQUANTE

& permet d'appeler la classe englobante du bloc

```
#sassEx {
    color: black ;
    a {
        font-weight: bold ;
        &:hover{
        color: red ;
        }
    }
}
```

Quelques **ressources**...

- SASS Cheat sheet
- Sass Meister (bac à sable)
- Tuto via Blog du Webdesign
- Initiation va La Cascade
- That Sass Way
- Sass Documentation