Cours HTML

3WAcademy

Cours d'intégration Web

8. SASS

8.1. Présentation

SASS *(**S**yntactically **A**wesome **S**tylesheets)* est un préprocesseur CSS, c'est à dire un langage de génération de feuille de style CSS.

C'est à la fois

- un langage (initialement avec la syntaxe indentée, désormais avec la syntaxe nommée SCSS) proche syntaxiquement du CSS mais qui apporte des fonctionnalités importantes comme les variables, l'imbrication du code, les tests et les boucle...
- un logiciel qui permet de traduire le SCSS en CSS pour être interprété côté client par le navigateur.

8.1.1. Concurrence

Il existe d'autres préprocesseurs CSS, notamment: LESS et Stylus.

8.1.2. Installation

Suivre l'une des procédures d'installation sur le site de SASS.

8.1.3. Quelques ressources

- http://sass-lang.com/
- https://sass-cheatsheet.brunoscopelliti.com/
- https://www.cheatography.com/mist-graphx/cheat-sheets/sass-script/pdf/

8.2. Utilisation

8.2.1. Utilisation directe via le terminal

Ouvrir le terminal à l'aide du raccourci Ctrl+Alt+T
Se placer dans le dossier de travail ou se trouve le fichier SCSS.
Pour compiler le fichier SCSS en fichier CSS:

Dans le Terminal

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

… où input.scss est le nom de votre fichier SCSS et output.css le nom de votre feuille de style CSS utilisée dans vos documents HTML

Pour compiler automatiquement:

Dans le Terminal

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

L'écran du terminal doit indiquer que le fichier est en cours compilation toutes les 2 secondes. Utiliser la commande Ctrl+C pour sortir.

Pour un projet complet avec compilation automatique lors de l'enregistrement:

Dans le Terminal

```
sass --watch path/to/css/folder
```

8.2.2. Utilisation via un plugin de votre éditeur

Les différents éditeurs ont des plugins permettant de lancer automatiquement la compilation des fichiers SCSS lors de leur enregistrement. Certains proposent la coloration syntaxique associée (celle du CSS peut aussi être utilisée).

8.2.3. Dans le HTML

Attention, ne pas lier directement le fichier SCSS mais bien le fichier CSS compilé:

Dans le HTML

```
<link rel=”stylesheet” href =”output.css”>
```

8.3. Syntaxe

8.3.1. Variables

SASS autorise l'utilisation de variables. Les variables sont préfixées avec un signe dollar ($). Les variables sont affectées par un deux-points (:). Lors de la compilation, les valeurs des variables sont substituées dans le document CSS.

Dans le SCSS

```
$color: #4D926F;

.header {
    color: $color;
}
h2 {
    color: $color;
}
```

Compilé en

Dans le CSS

```
.header {
    color: #4D926F;
}
h2 {
    color: #4D926F;
}
```

8.3.2. Imbrication

CSS supporte l'imbrication, mais les blocs eux-mêmes ne peuvent pas être imbriqués. SASS permet l'imbrication des sélecteurs à l'intérieur d'autres sélecteurs. Ceci rend l'héritage plus clair et les feuilles de style plus courtes.

Dans le SCSS

```
.header {
    h1 {
        font-size: 26px;
        font-weight: bold;
    }
    p {
        font-size: 12px;
        a {
            text-decoration: none;
            &:hover {
                text-decoration: underline;
            }
        }
    }
}
```

Compilé en

Dans le CSS

```
.header h1 {
    font-size: 26px;
    font-weight: bold;
}
.header p {
    font-size: 12px;
}
.header p a {
    text-decoration: none;
}
.header p a:hover {
    text-decoration: underline;
}
```

8.3.3. Import

Il est possible de découper sous code SCSS en plusieurs fichiers pour améliorer la maintenance et la lisibilité. Le nom des fichiers importés doivent être préfixé par un trait de soulignement "_".

Dans le SCSS

```
// _reset.scss
html,
body,
ul,
ol {
    margin: 0;
    padding: 0;
}
```

et

Dans le SCSS

```
// input.scss
@import 'reset';

body {
    font-family: Arial, sans-serif;
    color: #333333;
}
```

Compilé en

Dans le CSS

```
html, body, ul, ol {
    margin: 0;
    padding: 0;
}
body {
    font-family: Arial, sans-serif;
    color: #333333;
}
```

8.3.4. Mixins

Les mixins permettent d'inclure des propriétés d'un groupe de propriétés (nommé mixin) au sein d'un autre groupe de propriétés en incluant le nom de cette mixin dans les propriétés.

En supprimant les répétitions, les mixins permettent d'avoir un code plus court (pendant le développement uniquement,une fois compilé, il redevient aussi long que lors d'un développement "normal") et plus facile à modifier, au contraire de CSS qui ne supporte pas les mixins et où tout code doit être répété autant de fois qu'il est utilisé.

Dans le SCSS

```
@mixin bordered {
    border-top: dotted 1px black;
    border-bottom: solid 2px black;
}

.post a {
    color: red;
    @include bordered;
}
```

Compilé en

Dans le CSS

```
.post a {
    color: red;
    border-top: dotted 1px black;
    border-bottom: solid 2px black;
}
```

Il est possible d'ajouter des paramètres à la mixin:

Dans le SCSS

```
@mixin bordered($color: black) {
    border-top: dotted 1px $color;
    border-bottom: solid 2px $color;
}
```

```
.post a {
    color: red;
    @include bordered(red);
}
```

Compilé en

Dans le CSS

```
.post a {
    color: red;
    border-top: dotted 1px red;
    border-bottom: solid 2px red;
}
```

8.3.5. Héritage

Il est possible d'étendre les propriétés d'une classe à une autre:

Dans le SCSS

```
.message {
  padding: 1em;
  color: #333;
}

.success {
  @extend .message;
  color: green;
}

.error {
  @extend .message;
  color: red;
}
```

Compilé en

Dans le CSS

```
.message, .success, .error {
  padding: 1em;
  color: #333;
}

.success {
  color: green;
}

.error {
  color: red;
}
```

8.3.6. Opérations

Il est possible d'utiliser des opérations:

Dans le SCSS

```
$size: 2px;

.bordered {
    border: $size * 2 solid black;
}
```

Compilé en

Dans le CSS

```
.bordered {
    border: 4px solid black;
}
```

8.3.7. Fonctions

SASS est inclus avec certaines fonctions très utiles, par exemple pour la gestion des couleurs.

La liste complète est disponible ici:
http://sass-lang.com/documentation/Sass/Script/Functions.html

Par exemple:

Dans le SCSS

```
$color: #842210;

.footer {
    color: $color;
    border-color: desaturate($color, 10%);
}
```

Compilé en

Dans le CSS

```
.footer {
    color: #842210;
    border-color: #7d2717;
}
```

Il est également possible de déclarer ses propres fonctions avec la directive @function.

8.3.8. Directives

SASS inclus certaines directives propres aux langages de programmation comme les tests conditionnels et les boucles:

8.3.9. Exemple de test conditionnel

Dans le SCSS

```
$type: success;
```

```
.msg {
    @if $type == success {
        color: green;
    } @else {
        color: black;
    }
}
```

Compilé en

Dans le CSS

```
.msg {
    color: green;
}
```

8.3.10. Exemple de boucle

Dans le SCSS

```
@for $i from 1 through 3 {
  .item-#{$i} { width: 2em * $i; }
}
```

Compilé en

Dans le CSS

```
.item-1 {
  width: 2em; }
.item-2 {
  width: 4em; }
.item-3 {
  width: 6em; }
```