αds-COM
SSII WEBAGENCY

CER ALIANCE CENTRE

Référence n° : 201509RES-003-ADS

INTRODUCTION A BOOTSTRAP, LESS & SASS

Dossier suivi par :
Pierre-Antoine VIALLON

Version du document

N°	Date	Auteur	Commentaires
3.0	09/03/2016	Pierre-Antoine VIALLON	Ajustement procédure

Relecture du document

N°	Date	Auteur	Résultat	Commentaires

Références

Référence	Commentaire
www.getbootstrap.com	Site officiel Bootstrap
www.lesscss.org	Site officiel Less

1 Préambule 3

2 Bootstrap less et Sass, c'est quoi ? 3

2.1 Bootstrap en quelques mots 3

2.2 Tour d'horizon 3

2.3 Less et Sass en quelques mots 4

3 L'outillage 8

3.1 Le workflow 8

3.2 Pour commencer avec Less et Sass 9

3.3 Focus sur l'architectures Joomla! 10

3.4 Focus sur l'architecture Prestashop 11

1 Préambule

Depuis quelques années, on voit émerger tout un tas d'outils qui visent à faciliter et à fiabiliser l'intégration web HTML/CSS. Parmi ces outils, nous trouvons des framework et des préprocesseurs css.

Les framework tout d'abord proposent des modules prêts à l'emploi, menus, carrousels, formulaires. Les préprocesseurs vont ajouter de l'intelligence pour faciliter l'écriture des CSS et leur maintenance.

Nous allons présenter ici Bootstrap, un des framework les plus complets et présent sur le marché et le langage Less qui avec son cousin germain Sass est également un des préprocesseurs star.

2 Bootstrap less et Sass, c'est quoi ?

2.1 Bootstrap en quelques mots



Développé par les équipes de Twitter puis diffuser en Open Source, ce framework :

- Intègre nativement le responsive design de manière « mobile first »
- Garantit la compatibilité sur les principaux navigateurs web du marché
- Augmente la productivité lors des phases d'intégration
- Facilite les interventions de maintenance

Techniquement, Bootstrap c'est également :

- De nombreux widget HTML/JS : slideshow, menus, accordéon, base d'icônes, alerte, infobulles…
- Une intégration native de HTML5
- Une approche modulaire qui permet une optimisation du temps de chargement des pages (les portions CSS/JS de modules non utilisés ne sont pas appelées)
- Une intégration parfaite dans des environnements comme Grunt : contrôle et tests automatisés, minification des fichiers CSS et JS
- Les technologies LESS et SaSS pour une maintenance simplifiée et une meilleure productivité
- Une communauté grandissante, le Framework CSS/JS le plus utilisé bien avant d'autres outils comme Fondation, Skeleton ou Blueprint
- Un nombre considérable de template (gratuits ou payants) téléchargeables
- Une base de modules complémentaires conséquente

2.2 Tour d'horizon

www.getbootstrap.com :

- Pour commencer : Getting started
- L'approche mobile, les grilles et autres joyeusetés CSS
- Les modules : Components
- Les widgets Javascript : JavaScript

2.3 Less et Sass en quelques mots

Les langages Less et Sass rendent les feuilles de style intelligentes grâce à l'intégration de variables, de formules mathématiques et de fonctions.

Le principe, coder les feuilles de styles au format Less ou Sass et convertir ces fichiers en feuille de styles CSS traditionnelles.

Par ailleurs, Less et Sass permet de générer des feuilles de style minifiée à savoir sans les commentaires ou les espaces donc plus rapides à charger.

2.3.1 Variables

Les variables vont permettre une gestion bien plus simple des valeurs utilisées à plusieurs endroits dans les CSS, par exemple les couleurs, les tailles de police et aussi les dimensions d'écran pour gérer les variations d'affichage d'une résolution à une autre (média Queries).

Voici quelques exemples d'utilisation de ces variables :

Définition des variables

```
//
// Custom Colors (ads-COM)
// ------------------------------------------------

@minerales-1 : # f0dfc5;
@minerales-2 : # dbc3a0;
@minerales-3 : # c0ae8a;

@vegetales-1 : #999065;
@vegetales-2 : #739956;
@vegetales-3 : #516c5b;
@vegetales-4 : #272d30;

@liquides-1 : #34697d;
@liquides-2 : #6b979f;
@liquides-3 : #9ac2d6;
@liquides-4 : #9ccbe1;
@liquides-5 : #d4e3eb;

// Complementary
@minerales-comp-1 : #e2af6c;
@minerales-comp-2 : #f29400;
@minerales-comp-3 : #e75112;
@minerales-comp-4 : #ac311c;

@vegetales-comp-1 : #c2d35c;
@vegetales-comp-2 : #76b860;
@vegetales-comp-3 : #4c8b44;
@vegetales-comp-4 : #004a24;

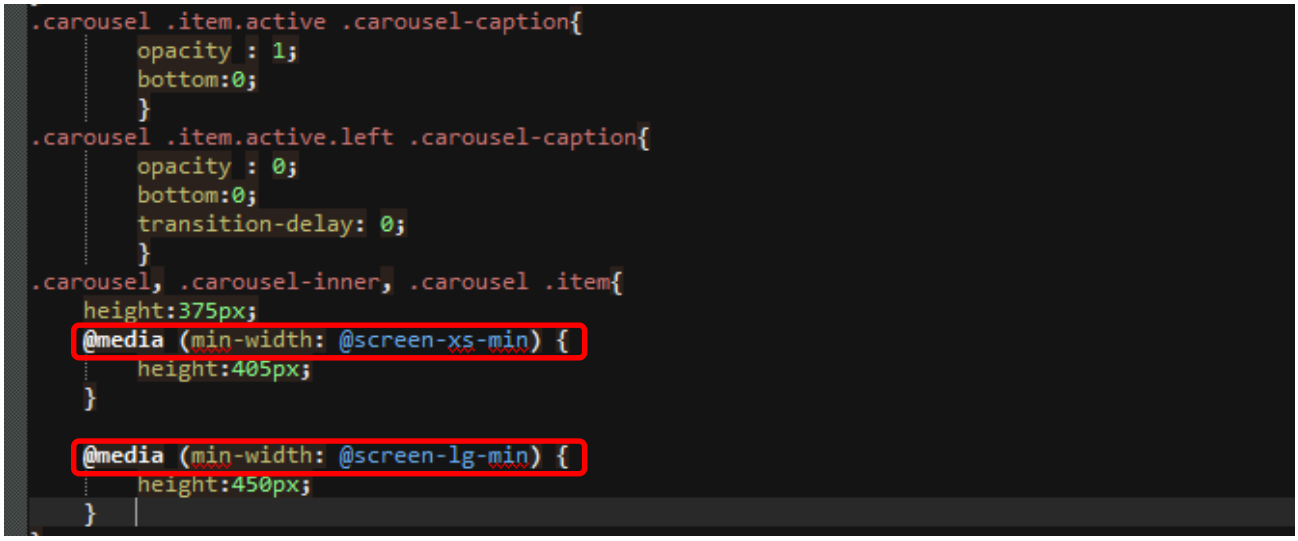
@liquides-comp-1 : #55afd0;
@liquides-comp-2 : #007cb2;
@liquides-comp-3 : #004393;
```

Utilisation de ces variables

```
// ------------------------------------------------
// Carousel
// ------------------------------------------------
.carousel{
    background-color:@mineral-1;
    margin:@grid-gutter-width 0;
    .carousel-caption{
        margin:0;
        width:100%;
        left:0;
        bottom:-20px;
        color:darken(@liquid-1,10%);
        text-shadow:none;
        padding: 20px 50px 5px;
        opacity : 0;
        transition:1s;
        transition-delay: 0.125s;
        position:absolute;
        @media (min-width: @screen-md-min) {
            z-index:100;
        }
        font-size:@font-size-base*1.25;
        @media (max-width: @screen-sm-max) {
            background-color:fade(@mineral-1,90%);
            padding-top:0;
            }
        h2{
            font-weight:800;
            text-transform:uppercase;
            }
    }
```

Côté syntaxe pour la définition des variables, Sass et Less sont très proches. Pour Less, nous prefixons les variables par @ pour Sass par $.

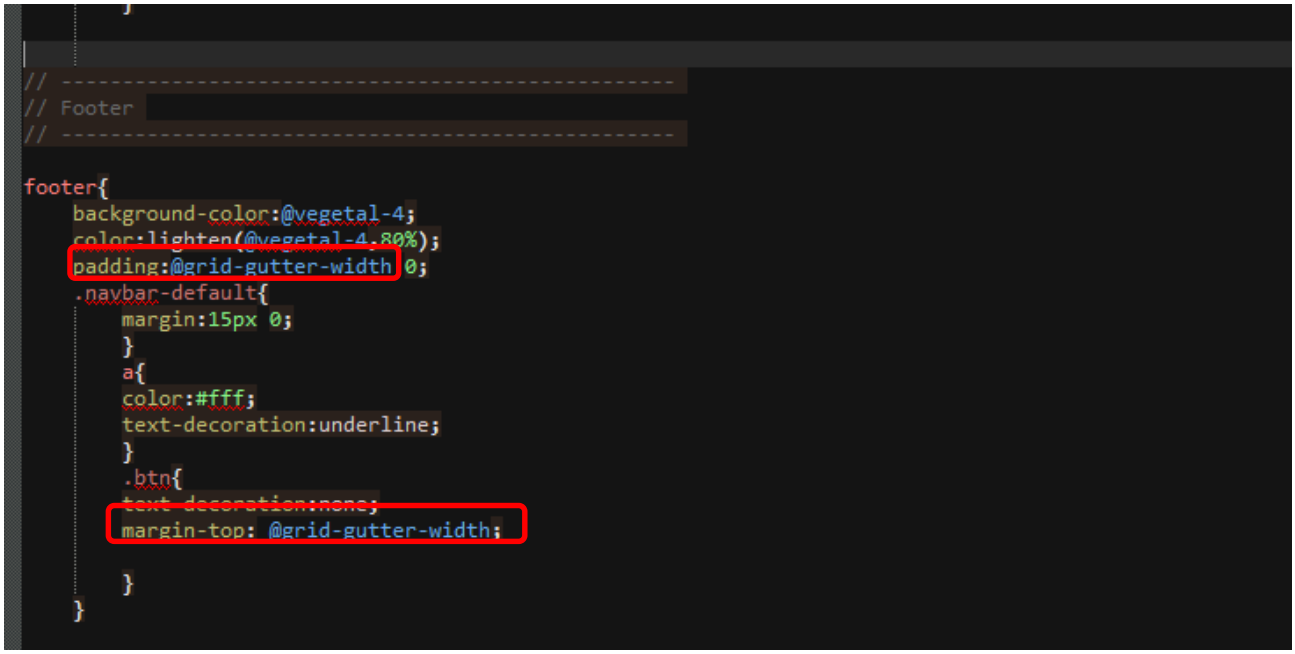
Gestion des breackpoints pour le responsive design



```
.carousel .item.active .carousel-caption{
        opacity : 1;
        bottom:0;
        }
.carousel .item.active.left .carousel-caption{
        opacity : 0;
        bottom:0;
        transition-delay: 0;
        }
.carousel, .carousel-inner, .carousel .item{
    height:375px;
    @media (min-width: @screen-xs-min) {
        height:405px;
    }

    @media (min-width: @screen-lg-min) {
        height:450px;
    }
```

Gestion de marges homogènes entre les blocs



```
// ------------------------------------------------
// Footer
// ------------------------------------------------

footer{
    background-color:@vegetal-4;
    color:lighten(@vegetal-4,80%);
    padding:@grid-gutter-width 0;
    .navbar-default{
        margin:15px 0;
        }
        a{
        color:#fff;
        text-decoration:underline;
        }
        .btn{
        text-decoration:none;
        margin-top: @grid-gutter-width;

        }
    }
```

2.3.2 Formules mathématiques

```
// ------------------------------------------------
// NEWS & Diary
// ------------------------------------------------

#actus, #agenda {
    margin-top:@grid-gutter-width;
    @media (max-width: @screen-xs-max) {
        text-align:center;
        img{
            display:inline;
            }

    }
    article{
        div:first-child{
            margin-bottom:0;
            }
        h3{
            margin-top:@grid-gutter-width/2;
            font-size:@font-size-base*1.55;

        }
        .caption{
            *{
                margin:0 0 2px;
                }
        }
        .badge {
            margin-top:5px;
                }
        .date{
            display:block;
            font-size:@font-size-small;
            color:lighten(@text-color,20%);
        }
    }
}


// ------------------------------------------------
// Jumbotron
// ------------------------------------------------

#jumbotron h3, .carousel h3 {
        font-weight:800;
        text-transform:uppercase;
        font-size:@font-size-h2;
```

2.3.3 Fonctions

A titre d'exemple, voici des fonctions natives de less. Celles-ci sont bien pratiques comme darken ou saturate qui permettent d'assombrir ou de saturer une couleur.

```
.quicklink  {
    @media (min-width: @screen-lg-min) {
    padding:@grid-gutter-width 0 ;
    }
    .well {
        padding: 5px 2px;
        text-align:center;
        background-color:@liquid-1;
        background-image:none;
            background-color:saturate(@minerales-comp-3,10%);
        transition:0.5s;
            box-shadow: 0 7px 30px rgba(0, 0, 0, 0);
        &:hover{
            box-shadow: 0 7px 30px rgba(0, 0, 0, 0.25);
                background-color:saturate(@minerales-comp-4,20%);
            }

        }
```

3 L'outillage

3.1 Le workflow



Téléchargement du Package Bootstrap

Personnalisation des modules

Surcharge des modules de base et des modules métiers du site

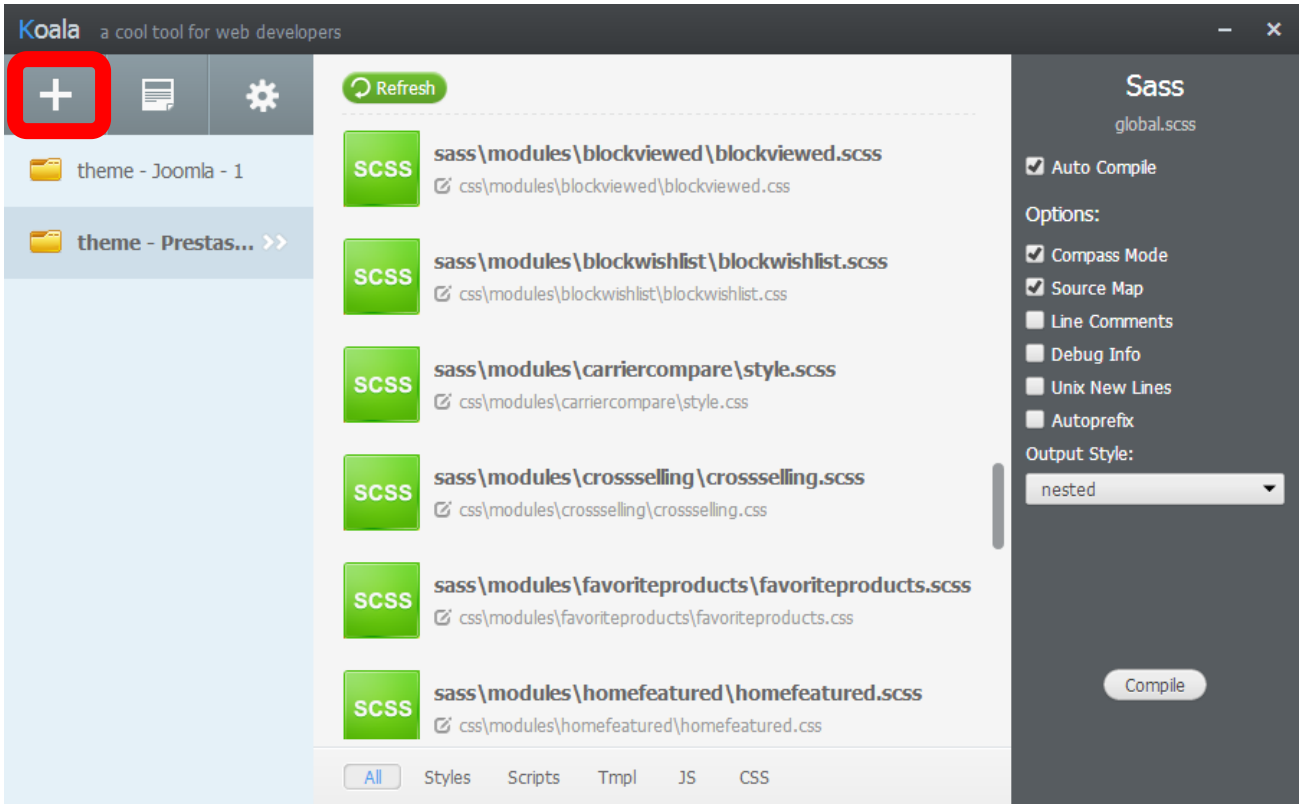
disctibution CSS, JS, Fonts

3.2 Pour commencer avec Less et Sass

Lors du développement, nous pouvons utiliser de petits utilitaires tels que koala ou winless qui permettent de compiler à la volée (lors de l'enregistrement les fichiers Less).

Pour utiliser ces outils, il suffit de glisser/déposer le répertoire Less dans Koala et de sélectionner les fichiers Less que l'on souhaite convertir en fichier css. Dans bootstrap, les fichiers bootstrap.less et theme.less sont généralement les fichiers compilés en CSS sur prestashop ou joomla, l'architecture est un peu différente.

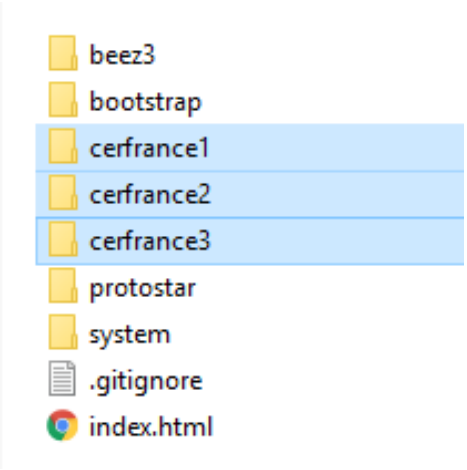
Lors de l'enregistrement le préprocesseur convertit ce fichier en fichier CSS (minifié ou non).



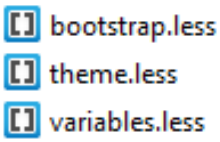
Clic droit sur le fichier bootstrap.less puis « select output File » et l'on sélectionne le fichier de destination « \dist\css\bootstrap.min.css ».

3.3 Focus sur l'architectures Joomla!

Composition du répertoire template : \templates



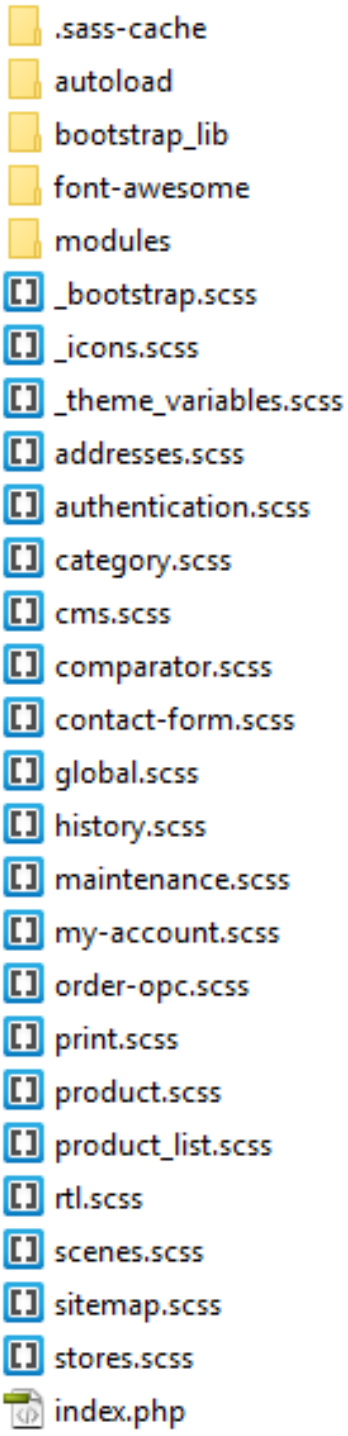
Composition d'un répertoire de thème :



La philosophie pour Joomla :

- Définir les variables
- Surcharger les bibliothèques Bootstrap grâce au fichier thème.less

3.4 Focus sur l'architecture Prestashop



La philosophie pour Prestashop :

- Définir les variables « _theme_variables.scss » ou nootstrap_lib/_variables.scss
- Cibler les fichiers Sass correspondants aux grandes fonctions et aux modules
- Modifier ces fichiers
- Surcharger le thème