



Kit CSS Oxatis

Prise en main et mise en application des ressources

Le Kit CSS Oxatis vous est proposé sous la forme d'un thème graphique complet et modulaire.

Grâce au langage SASS ainsi que l'architecture du kit, vous pouvez déployer rapidement et de manière cohérente vos designs sur la solution Oxatis.

04 86 26 26 26

www.oxatis.com

Table des matières

Table des matières	1
Prérequis.....	2
Environnement technique.....	2
Mise en place du CSS sur la solution Oxatis	2
Compréhension de la structure html et des class CSS sur la solution Oxatis	2
Les class spécifiques aux pages de la solution Oxatis	3
Les zones de contenu.....	3
Architecture du Kit CSS	4
Notions de base	4
Structure du projet	4
Focus sur le dossier 00-global.....	5
Les valeurs dynamiques du kit CSS.....	6
Focus sur les variables de couleur	6
Largeur du site et gestion du layout.....	7
Focus sur le sous-dossier _layout	7
Utilisation du layout et des variables associées.....	7
Largeur des contenus personnalisés.....	8
Templates de pages	9
Focus sur la page userdashboard	9
Entête et pied de page	10
Entête HTML	10
Pied de page	10

Prérequis

Environnement technique

Pour utiliser le kit CSS propulsé par le langage SASS, vous aurez besoin d'un IDE configuré pour compiler du scss en css. Vous pouvez par exemple utiliser NetBeans (gratuit).

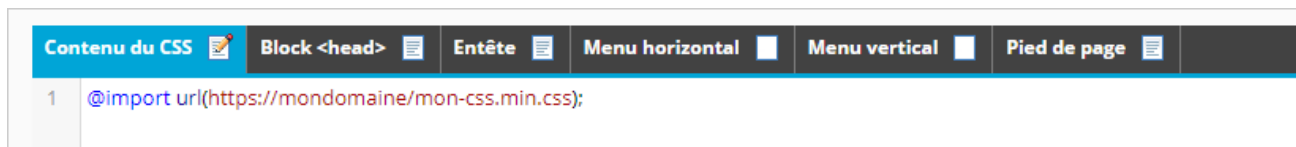
Vous devrez déployer le contenu fournit dans votre environnement et compiler le fichier map.scss présent à la racine (voir page 4).

Nous vous conseillons de configurer le sourcemap CSS à la compilation SASS, afin de retrouver rapidement la localisation des règles CSS dans la console de développement des navigateurs.

Mise en place du CSS sur la solution Oxatis

Vous devrez créer un Design dans le Back Office de votre site (Configuration > Design).

Nous vous conseillons d'appeler le fichier CSS dans le « contenu du CSS » via la règle `@import url(votre-fichier-css.css)`. De cette manière, votre fichier sera chargé sur les mises en page sans en-tête, contrairement à un import html.



Compréhension de la structure html et des class CSS sur la solution Oxatis

Vous trouverez les informations essentielles sur l'aide en ligne à destination des partenaires : <https://partners.oxatis.com/s/article/structure-generale-site?language=fr>

Les class spécifiques aux pages de la solution Oxatis

Chaque page est « flaguée » grâce à une ou plusieurs class CSS.

Celles-ci se retrouvent sur le conteneur principal portant l'ID #maincontainer.

Certaines de ces class sont « génériques ». Par exemple, vous retrouverez la class `.useraccount` sur la majorité des pages/composants du compte utilisateur. Elle sera accompagnée d'une class spécifique en fonction de la page correspondante. La page « Votre Compte » comporte les class `.useraccount.userdashboard` par exemple.

```

<script type="text/javascript" language="javascript">...</script>
<div class="l-site__before-header-container js-before-header-container">...</div>
.. <div id="maincontainer" class="useraccount userdashboard">...</div> == $0
  <!-- maincontainer -->
  <script type="text/javascript" id="oxStatRecorder" defer>...</script>

```

Les zones de contenu

Le contenu des pages est majoritairement généré dans deux div imbriquées.

Le design proposé dans le kit CSS est organisé autour de ces « sections » :

`.sectiondata > .sectiondataarea`.

Le nom d'usage employé dans le kit CSS pour ces zones est « content ». Par exemple, la variable permettant de contrôler le blanc tournant (padding) se nomme `$content__white-edge`.



Architecture du Kit CSS

Le kit CSS s'organise autour des composants/pages que vous retrouverez sur la solution Oxatis.

Vous retrouverez par ailleurs les éléments dits « globaux » dans un même dossier. Nous avons par exemple rassemblé les variables SASS principales, les règles générales pour les éléments de formulaires, tableaux, textes, titres, ou encore les boutons ; mais aussi le layout général, le header, le footer ou encore les sidebars.

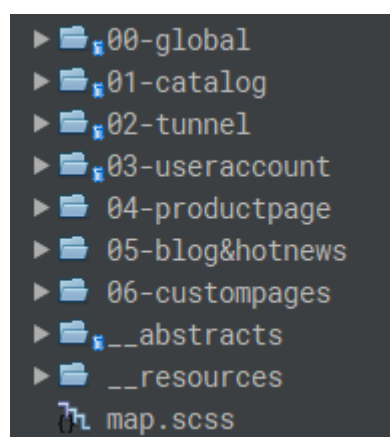
Notions de base

Vous trouverez dans chaque dossier un fichier « base » (ex : `base-submenu.scss`, `base-layout.scss`, etc) qui rassemblent les règles CSS communes et l'essentiel pour assurer l'affichage des pages concernées.

Chaque dossier possède son propre index nommé `__index.scss` dans lequel les fichiers de style sont appelés. Ces index ne comportent pas directement de CSS.

Le fichier `map.scss` à la racine du projet est le chef d'orchestre du kit CSS. C'est ce fichier qui doit être compilé en css.

Structure du projet



00-global contient l'essentiel pour personnaliser le kit CSS.

01-catalog contient les fichiers de style catalogue.

02-tunnel rassemble les pages du tunnel d'achat.

03-useraccount rassemble les pages du compte utilisateur.

04-productpage contient les styles génériques et spécifiques de la fiche produit et du template de base.

05-blog&hotnews rassemble les style pour ces composants.

06-custompages accueille vos pages sur mesure.

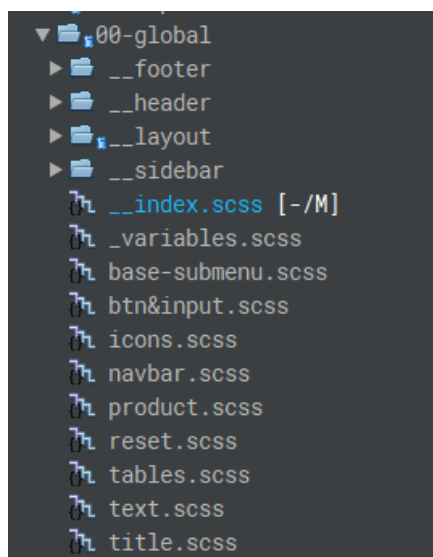
__abstracts rassemble les fonctions et mixins du kit CSS.

__resources rassemble les ressources fournies avec le kit.

map.scss est le fichier maître du projet.

Focus sur le dossier 00-global

Ce dossier rassemble les éléments essentiels vous permettant de personnaliser le design.



Vous retrouverez dans le dossier global les éléments de design généraux tels que les boutons, les textes ou les titres ; mais aussi le layout du site, l'entête, le pied de page ou encore les sidebar.

_variables.scss contient toutes les variables SASS permettant de piloter le design.

product.scss sert à modifier le design général des produits dans les listings (catalogue, recherche rapide et page de recherche).

Les informations remontées des produits affichés dans le listing catalogue et leur organisation reste dépendant des paramètres configurés dans le back office.



Les valeurs dynamiques du kit CSS

Vous retrouverez des variables commentées permettant de piloter les grandes lignes de votre design dans le fichier `00-global/_variables.scss`.

Des placeholders/extend locaux sont utilisés. Les boutons sont par exemple stylisés de cette manière, voir fichier `00-global/btn&input.scss`.

Focus sur les variables de couleur

`$color--primary100` | Couleur primaire de votre design

`$color--primary100-contrast` | Couleur de contraste en superposition

La couleur primaire est utilisée pour les boutons principaux du design ou encore les titres des composants.

`$color--secondary100` | Couleur secondaire de votre design

`$color--secondary100-contrast: #262D54` | Couleur de contraste en superposition

La couleur secondaire est utilisée pour les boutons généraux du design, elle est aussi utilisée au survol des boutons principaux dans le kit CSS.

`$color--tonic100` | Couleur d'accent de votre design

`$color--tonic100-contrast` | Couleur de contraste en superposition

`$color--tonic50` | Couleur d'accent secondaire de votre design

`$color--tonic50-contrast` | Couleur de contraste en superposition

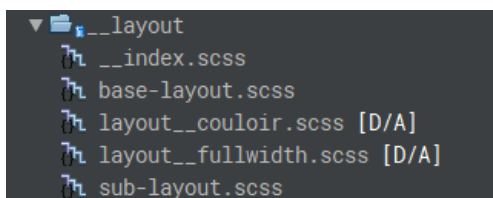
La couleur d'accent est utilisée sur les prix ou encore les boutons liés l'achat. La couleur secondaire sert principalement au survol des boutons d'accent.

Largeur du site et gestion du layout

Nous avons mis à disposition deux layout gérant la largeur de votre site. Vous pouvez de cette manière changer à la volée votre affichage en pleine page ou couloir.

L'affichage en pleine page/fullwidth permet de « coller » les sidebars de chaque côté de la fenêtre tout en contraignant la « zone centrale » de contenu à la largeur souhaitée.

Focus sur le sous-dossier `__layout`

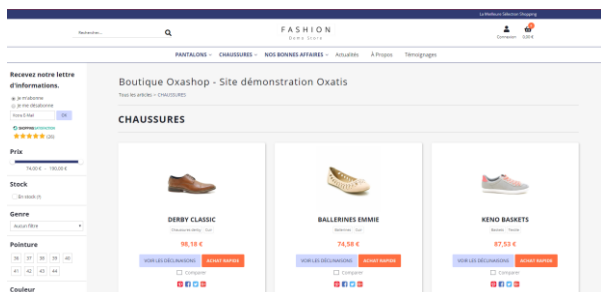


Vous retrouverez dans le dossier `__layout` les fichiers pour passer le « main content » en fullwidth ou couloir.

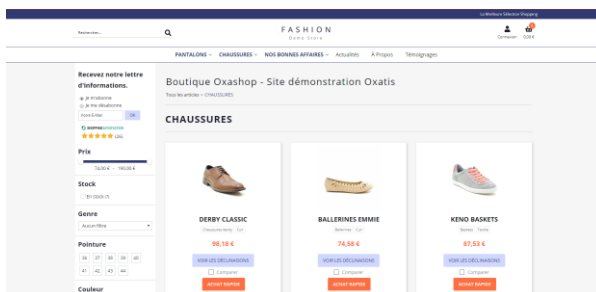
Il faudra remplacer le fichier appelé dans l'index pour jongler entre les deux.

Utilisation du layout et des variables associées

Exemple fullwidth



Exemple couloir



Vous pourrez piloter la largeur du site avec la variable `$layout__site-width` (100% de base). La largeur du « main content » sera pilotée avec `$layout__content-width` (1440px de base).

Avec le layout couloir, tout le « main content » est contraint par `$layout__content-width`, la valeur renseignée dans `$layout__site-width` n'est pas utilisée.

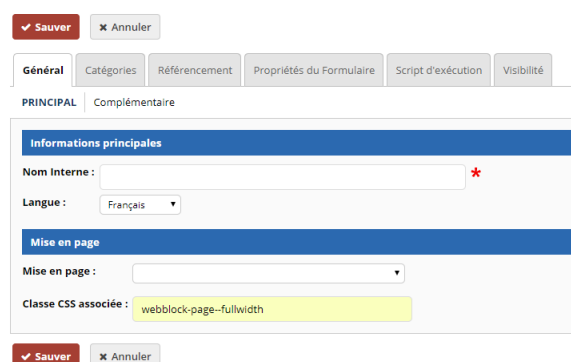
Largeur des contenus personnalisés

Il est possible de dissocier la largeur du header ainsi que vos pages sur mesure.

Vous trouverez dans 06 - custompages, un fichier nommé `base-custompage.scss`.

Nous y avons créé deux class CSS pour vous faciliter de vos designs et contenus personnalisés.

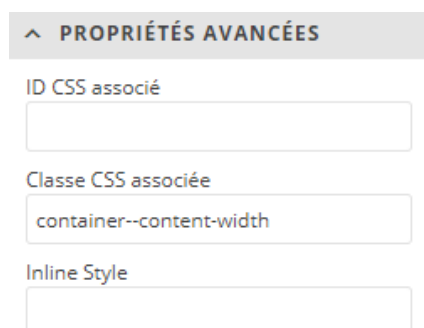
- La class `.custom-page--fullwidth` est à renseigner dans les paramètres avancés de votre page sur mesure. Comme son nom l'indique, la page sur mesure sera alors à 100%.



- La class CSS `.container--content-width` est à renseigner sur les conteneurs principaux/sections de vos pages sur mesure glisser/déposer. Nous l'utilisons aussi dans le modèle d'entête HTML fournit.

Cette class est pilotée par la variable `$layout__content--width`

widget page sur mesure



Utilisation dans l'entête du design

```

1 <div class="header-banner__container">
2 <div class="container--content-width header-banner__inner">
3 <div class="l-ox-grid l-ox-grid--bleed l-ox-grid--justify-space-between l-ox-grid--align-center">
4 <div class="l-ox-grid__col-auto header-img__container">
5
6 <span class="header-img__inner">#oxtagbegin#hding#oxtagend#</span>
7
8 </div>
9 <div class="l-ox-grid__col-auto header-text__container">
10
11 <span class="header-text__inner">#oxtagbegin#hdtxt#oxtagend#</span>
12
13 </div>
14 </div>
15 </div>
16 </div>
17 <div class="header-main__container">
18 <div class="container--content-width header-main__inner">
19 <div class="l-ox-grid l-ox-grid--bleed l-ox-grid--justify-space-between l-ox-grid--align-center">
20 <div class="l-ox-grid__col-auto header-shop-search__container">

```

Templates de pages

Grâce à l'architecture du kit CSS, vous pourrez créer de multiples Templates de pages afin de personnaliser le design de vos sites.

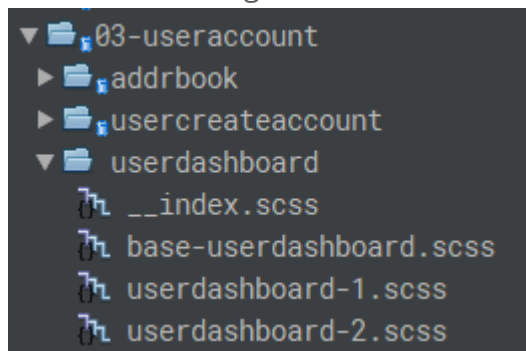
Nous vous proposons un premier exemple sur la page « userdashboard » du compte utilisateur. Vous pourrez reproduire et/ou adapter cette organisation de fichier pour créer votre propre Templates de pages.

Focus sur la page userdashboard

Vous retrouverez deux thèmes disponibles dans le dossier correspondant.
Vous devrez importer le fichier correspondant dans l'index du dossier userdashboard.

Attention, quand un thème est disponible dans les fichiers, il est obligatoire de l'appeler pour assurer un affichage optimal de de la page concernée.

Localisation et organisation



__index.scss

```
1 @import "base-userdashboard.scss";  
2 @import "userdashboard-2.scss";
```

userdashboard-1



userdashboard-2



Entête et pied de page

L'entête et le pied de page sont à renseigner dans le Design du Back-office.

Ce sont des zones d'insertion HTML. Le contenu du header est à construire autour de tag que vous trouverez dans l'aide Partenaires.

Aide en ligne : <https://partners.oxatis.com/s/article/zones-edition-points-insertion-html?language=fr>

Entête HTML

Nous vous proposons un entête accompagné de son CSS, vous retrouverez le html dans le fichier `__resources/header-1.html`. Vous devrez coller le code dans l'entête de votre design.

Le panier comporte quelques particularités, nous vous conseillons de conserver le html fourni. Les ID CSS `#headerInfoCart` et `#cartItems` ainsi que la class CSS `.cartPrice` doivent être présents tel que proposés. Vous pouvez modifier les autres class si vous le souhaitez.

`#headerInfoCart` | sert d'ancre pour l'APP du panier

`#cartItems` | permet de mettre à jour dynamiquement le nombre d'articles dans le panier

`.cartPrice` | permet de mettre à jour dynamiquement le prix du panier

Pied de page

Le contenu ajouté dans le champ « Pied de page » du design est englobé dans un conteneur portant l'ID `#footerblock` et se trouvant au même niveau que le menu de pied de page.

Pour ajouter du contenu nous vous conseillons d'intégrer une « page sur mesure » à l'aide du tag `#oxincbegin#wb|XXXXXX#oxincend#` (XXXXX étant l'ID de la « page sur mesure »).

Astuce : vous pouvez déplacer des éléments « génériques » du footer (menu, mentions) dans votre « page sur mesure » en utilisant `<div cs cs-grab target=".class or #id"></div>` dans un widget HTML.