

Customisation

Prestashop

Table des matières

| | |
|--|---|
| Les prérequis..... | 2 |
| Le thème par défaut..... | 2 |
| Le principe du thème enfant..... | 3 |
| Création d'un thème-enfant..... | 4 |
| Activation du thème enfant..... | 5 |
| Mise en pratique..... | 6 |
| Modifier l'habillage CSS du site..... | 6 |
| Modifier la structure d'un module..... | 7 |
| Exercices :..... | 9 |

Les prérequis

Avoir installer un site Prestashop 1.7.

Posséder un compte Admin de l'espace d'administration de Prestashop

Avoir les droits en écriture via un terminal sur les dossiers et fichiers du serveur web.

Le thème par défaut

Qu'est-ce c'est ?

Pour faire simple c'est l'habillage de votre site E-Commerce. Ou mise en forme graphique de votre boutique.

Présenté sous la forme d'une suite de fichiers contenus dans un dossier, un theme PrestaShop permet de changer l'apparence d'une boutique PrestaShop pour l'adapter à la thématique de l'offre produit qui y est commercialisée.

Le theme par défaut de Prestashop s'appel « classic » et est le seul disponible à la fin de l'installation de la plate-forme.

Dans le répertoire `/themes/classic/` vous avez 5 dossiers dans lesquels se trouve des fichiers template (`.tpl`), `css`, ou `js`.

Dans **/assets** se trouve les fichiers `Css` et `JS`

Dans **/config** se trouve le fichier de configuration du thème : `theme.yml`

Dans **/modules** se trouve les fichiers des modules installé, mais qui peuvent être désinstallés.

Dans **/plugin**, il n'y a rien.

Dans **/template** se trouve les fichiers plus généraliste et obligatoire.

Enfin une image « **preview.png** » qui servira d'aperçu dans le Back-end.

Le thème peut être modifié, customisé afin d'avoir sa propre identité visuel.

L'emplacement des bloc ou module ou addon, peut-être modifié ainsi que la couche `CSS` (couleur, fond, police, marges, etc...)

Pour cela, nous n'allons pas modifier notre thème directement.

Car les modifications risquerai de disparaître en cas mise à jours de la plate-forme.

Nous allons créer ce que nous appelons un thème-enfant.

Le principe du thème enfant

Le thème-enfant est une copie des fichiers du thème par défaut ou thème-parent. On peut modifier le thème-enfant sans crainte, le thème-parent ne sera pas impacté.

Cependant, les modifications des fichiers du thème-enfant seront pris en compte en priorité par Prestashop. Tel une surcouche du code.

Intérêt 1 :

Permet de modifier un thème tout en conservant de manière fiable les mises à jours de la plateforme.

Raison, si vous n'utilisez pas de thème enfant (c'est possible) , si vous décidez de faire des mise a jour de la plateforme (ce qui est conseillé par mesure de sécurité), vous risquez de perdre les modifications que vous aviez effectué sur votre thème.

Intérêt 2 :

Vous disposerez d'un moyen afin d'augmenter les fonctionnalités natives de votre thème parent.

Il est donc important de créer un thème enfant dès la conception de votre site E-Commerce.

Création d'un thème-enfant

A l'aide d'un terminal, positionnez-vous dans le dossier **/themes** de votre plate-forme.

Créer un nouveau dossier **/classic-child** (Il est très important de nommer ce répertoire par le nom du thème-parent et d'y ajouter « -child ». Sans quoi le thème-enfant ne fonctionnera pas.

A l'intérieur du nouveau dossier **/classic-child**, créer un nouveau dossier **/config**

Créez un fichier nommé **theme.yml** et enregistrez-le dans le répertoire **/config**

((Si vous regarder dans le thème-parent, vous remarquerez qu'il y a les mêmes dossiers et fichier, que nous venons de créer.))

Maintenant ouvrez le fichier **theme.yml** de notre theme-enfant

écrire ce qui suit :

```
parent: classic
name: classic-child
display_name: Mon theme enfant
version: 1.0.0
assets:
  use_parent_assets: true
```

Explication :

parent : c'est le nom du dossier du thème parent, dans notre cas le thème classic

name c'est le nom de votre thème, j'ai fait simple...

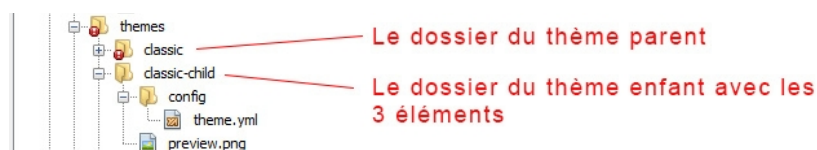
display_name : le nom qui s'affichera.

laissez asset vide

et mettez use_parent_assets sur true

Enregistrer et fermer.

Pour finir, créer puis déposer une image « **preview.png** » à la racine du dossier **/classic-child** qui représentera le design final de votre thème.



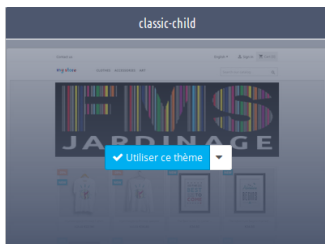
Et Voilà ! Le plus difficile est fait.

Activation du thème enfant

Maintenant rendez-vous dans votre back-end, dans le Menu **PERSONNALISER / Apparence** puis « **Thème et logo** »

Dans la section « **Sélectionner un thème pour la boutique "PrestaFMS"** » vous découvrez une étiquette portant le nom classic-child.

Pour activer le thème-enfant, Cliquer sur le bouton « ✓ **Utiliser ce thème** » qui apparaît au survole de cette étiquette.



Maintenant que le thème-enfant est actif, nous allons pouvoir commencer ce que l'on appelle l'Override

Que nous dit Google Traduction ?

« **Override** » peut être traduit par : « outrepasser », « ne pas tenir compte »... et c'est exactement ça, le principe.

En gros, on duplique un fichier que l'on veut modifier dans un répertoire précis, et PrestaShop l'utilise, sans tenir compte des fichiers d'origine.

Mise en pratique

Modifier l'habillage CSS du site.

Changer la couleur de fond du menu Nav-top (Contactez-nous, Connexion, Panier.)

Prestashop mets à disposition un fichier [custom.css](#) qui me permet de rajouter mon propre CSS et d'**override** les éléments du thème par défaut.

Ce fichier est situé dans :

<http://localhost/prestaFMS/themes/classic/assets/css/custom.css>

Nous allons copier ce fichier « [custom.css](#) » dans notre thème-enfant.

Mais pour que le fichier soit pris en compte, il faudra aussi créer les même dossiers dans l'arborescence.

> **<http://localhost/prestaFMS/themes/classic-child/assets/css/custom.css>**

Changer la couleur de la barre haute de couleur de blanc à gris.

Avec l'inspecteur de code :

1/ noté le nom de la class qui gère l'affichage du menu Nav-top (ici : `#header .header-nav`).

2/ récupérer le code css (Vous pouvez remarquer que la class se trouve dans le thème-parent.)

```
#header .header-nav {  
    border-bottom: 2px solid #f1f1f1;  
    max-height: 50px;  
    margin-bottom: 30px;  
}
```

Dans cette exemple nous voulons juste modifier la couleur de fond de la barre.

Ouvrir le fichier [custom.css](#) coller la class css et supprimer les propriétés et enfin ajouter la propriété **background**.

```
#header .header-nav {  
    background: #ddd;  
}
```

Enregistrer, Actualiser votre page. Et voilà !

La barre tout en haut de l'interface est devenu grise. Et nous n'avons pas modifier le code d'origine du thème par défaut.

Si vous ne voyez rien, allez vider le cache de prestashop dans « **Paramètres avancés / Performances** »

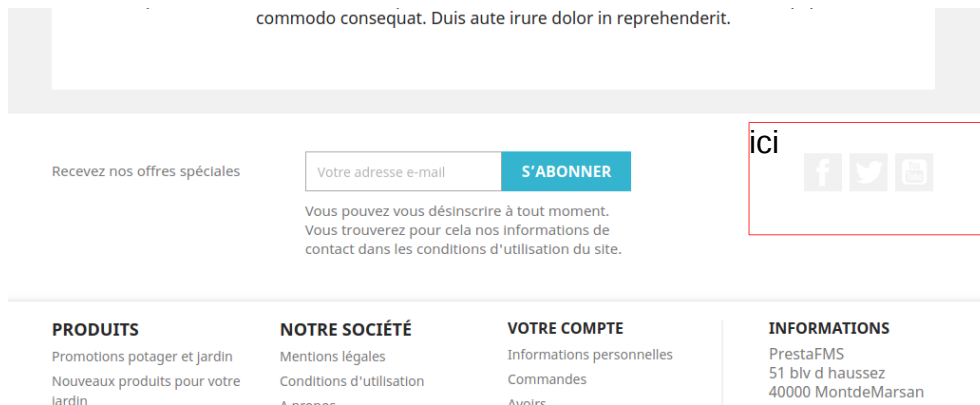
La Customisation à commencé ! (Vous pouvez modifier d'autre éléments pour vous entraîner)

Modifier la structure d'un module

Exemple : Module réseau sociaux

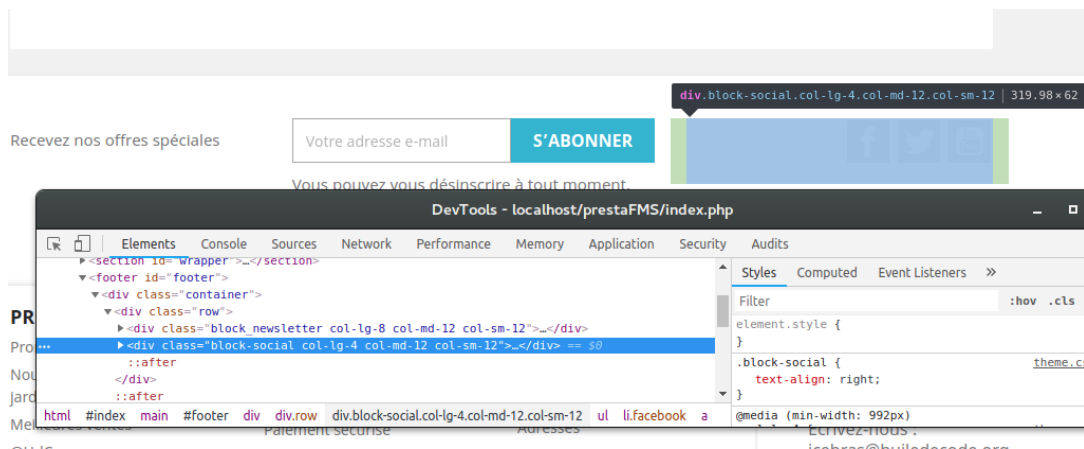
Nous allons modifier l'emplacement, la couleur, la forme, la taille de ce module.

Par défaut son affichage ressemble à ça :



Avec l'inspecteur de code :

1/ repérez le nom de la class css utilisé pour intégrer le module.



On remarque :

`<div class="block-social`

Cherchons maintenant un module qui portera un nom similaire dans les dossiers du thème-parent. Et en cherchant bien, on trouve !!!

http://localhost/prestaFMS/themes/classic/modules/ps_socialfollow/ps_socialfollow.tpl

Construisons notre Override en copiant ce fichier dans notre thème-enfant. Nous allons devoir créer aussi les dossiers.

> http://localhost/prestaFMS/themes/classic-child/modules/ps_socialfollow/ps_socialfollow.tpl

Le Template du module est prêt à être modifié.

Éditez le fichier [ps_socialfollow.tpl](#)

Modifier la ligne où se trouve la largeur des colonnes Bootstrap. En taille écran ordi (lg), passer la valeur de 4 à 12.

avant : `<div class="block-social col-lg-4 col-md-12 col-sm-12">`

après : `<div class="block-social col-lg-12 col-md-12 col-sm-12">`

Enregistrer, Actualiser votre page.

Ce qui aura pour effet pour le module de prendre 12 colonnes sur 12 disponibles. En d'autre terme le module prendra la largeur de la page, en dessous du module de Newsletter.

Le CSS maintenant.

Ouvrez le fichier [custom.css](#) de notre thème-enfant. Et ajouter le code suivant à la fin du fichier.

Nous allons centrer le contenu de la `<div>`

```
.block-social {
    text-align: center;
}
```

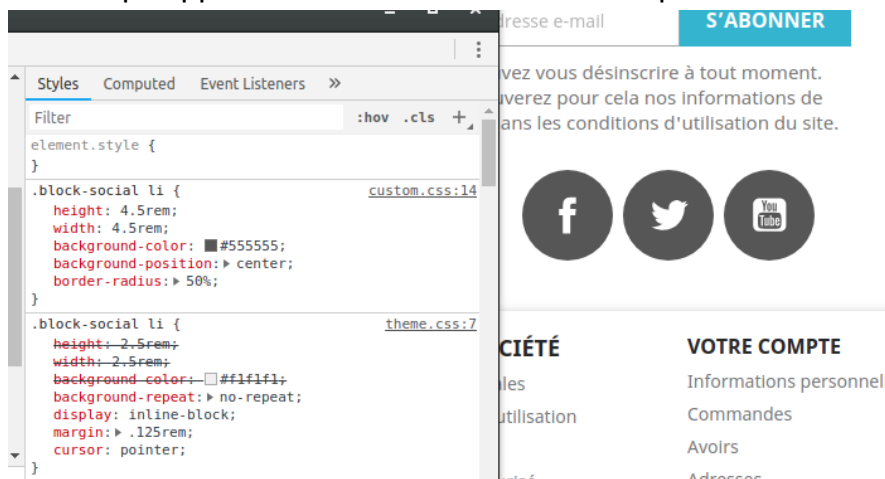
Enregistrer, Actualiser votre page.

Maintenant on modifie les « li » qui se trouvent dans notre div, afin d'obtenir des ronds à la place des carrés, une taille plus grande pour nos boutons, une couleur différente et un centrage de l'icone se trouvant dans le « li »

```
.block-social li {
    height: 4.5rem;
    width: 4.5rem;
    background-color: #555555;
    background-position: center;
    border-radius: 50%;
}
```

Enregistrer, Actualiser votre page. C'est fini.

Voici ce qui apparaît maintenant dans notre Inspecteur de code.



Nous remarquons que le code se trouvant dans custom.css surcouche, le code par défaut du module.

Exercices :

Pour vous entraîner , je vous invite à continuer la customisation avec d'autres modifications.
En adaptant les tutoriels suivants à votre plate-forme :

<https://www.team-ever.com/prestashop-1-7-creer-un-theme-enfant-child-theme/>

>> Ne réalisez que la partie **2/ Boost d'un outil natif : les favicons**

<http://eric.delord.fr/modifier-le-footer-du-theme-par-defaut-de-prestashop-1-7/>

>> Réaliser la personnalisation du footer du site pour adapter avec un lien vers le site d'FMS la mention

© 2017 – Ecommerce software by PrestaShop™