

Trusty, BROUILLON

# Pluxml, personnalisation : thèmes et templates

Le **thème** est le look général du site.

il comporte des **templates** qui définissent le look d'une ou plusieurs pages.



Les thèmes et templates sont physiquement indépendants du système de **Pluxml** → ils ne sont pas affectés par les mises à jour.

Il existe de nombreux thèmes et templates, en particulier ici : <http://ressources.pluxml.org/>.

Pour un site donné, le nom du thème apparaît dans **Admin/Paramètres/Options d'affichage**, sous "Choix du thème :" :

Un template définit le look d'une ou plusieurs pages et est sélectionné lors de la rédaction d'une page statique ou d'un article de blog :

Elle peut être "activée" ou non, je peux y mettre du PHP et y produire tout ce que je désire.

Je ne la confonds pas avec les "articles" qui apparaissent dans le "blog" selon mon bon vouloir ou sont utilisées pour plein d'autres choses.

Visualiser la page Test sur le site

Template : static-full-width.php

Contenu balise title (option) :

Test de Pluxml

Contenu de la balise meta "description" pour cette page statique (option) :

Contenu de la balise meta "keywords" pour cette page statique (option) :

Enregistrer cette page statique

En ajoutant quelques fichiers dans le thème choisi, il est possible de changer facilement le look des pages ; par exemple, vous pouvez :

- mettre le menu (la sidebar) à droite, en mettre un de chaque côté de la page d'accueil,
- avoir une sidebar différente selon les pages ou rubriques du site,
- ou un Header différent selon ce dont on parle.
- etc.

Changeons le look d'une page statique pour qu'elle n'ait plus de sidebar (= menu "automatique").

Ouvrez votre page de test en mode édition (dans la liste des pages statiques, cliquez sur le lien [Éditer](#))

La liste déroulante "Template :" affiche "static.php" → cette page utilise donc le template **static.php** dans le thème courant (**theme-zen**).

Cliquez sur la flèche pour ouvrir le menu déroulant , choisissez le template **static-full-width.php**<sup>1)</sup> et enregistrez.

C'est tout : le nouveau template s'applique. Voici le résultat sur la page test :



Voyons maintenant comment créer un template.

## Créer un template

Nous allons créer un template pour personnaliser le look de certaines pages. Il suffira d'attribuer ce template aux pages voulues pour changer leur look.

Un template se compose d'un fichier selon le type de pages auxquelles il s'appliquera :

- **home-xxx.php** pour une page d'accueil
- ou **article-xxx.php** pour des articles de blog
- ou **categorie-xxx.php** pour les pages d'une catégorie
- ou **static-xxx.php** pour des pages statiques



Le nom d'un template doit commencer par **home-**, **article-**, **categorie-** ou **static-** et se terminer par **.php**. Entre les deux, le nom choisi pour le template **xxx**.

Il est accompagné d'un ou plusieurs fichiers définissant le look des diverses parties de la page :

- **header-xxx.php** pour l'en-tête
- **footer-xxx.php** pour le pied de page
- **commentaires-xxx.php** pour les commentaires
- **sidebar-xxx.php** pour la sidebar
- **tags-xxx.php** pour les tags



Les fichiers de zone doivent commencer par **header-**, **footer-**, **commentaires-**, **sidebar-** ou **tags-** et se terminer par **php**. Entre les deux, le nom du template.

**Pour une plus grande facilité, nous remplacerons ces fichiers par des pages statiques faciles à personnaliser.**

Un fichier de style est associé, nommé selon le type de modèle :

- **home-xxx.css**
- ou **article-xxx.css**
- ou **categorie-xxx.css**
- ou **static-xxx.css**



Les fichiers de style reprennent le nom du fichier modèle en remplaçant **.php** par **.css**

## Exemple

Faisons un template **xxx** pour des pages statiques avec un header personnalisé (nous ferons ce dernier dans une page statique).

Ouvrez **FTP** et :

rapatriez dans un dossier de travail les fichiers :

- **static.php**
- **style.css**

2. renommez-les :

- **static.php** → **static-xxx.php**
- **style.css** → **static-xxx.css**

3. renvoyez-les sur le serveur à côté des autres



Si votre template s'appelle **static-xxx.php**, votre fichier **.css** devra s'appeler **static-xxx.css**.

Il est alors chargé automatiquement par votre template lorsque celui-ci est utilisé.

## Définir le look pour le template



Nous prenons l'exemple de pages statiques mais la méthode est la même pour les autres. Remplacez **static-xxx...** par :

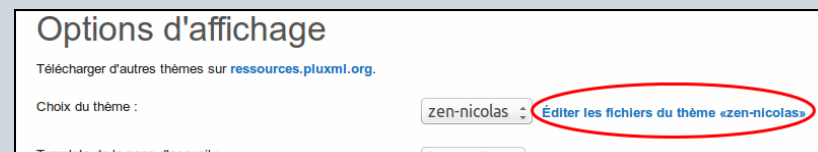
- **home-xxx...** pour une page d'accueil
- **article-xxx...** pour des articles de blog
- **categorie-xxx...** pour les pages d'une catégorie

Nous avons donc les fichiers :

- **static-xxx.php**
- **static-xxx.css**

Pour éditer les fichiers du template,

- ouvrez les paramètres d'affichage de l'administration
- ouvrez le lien Editer les fichiers du thème zen :



- dans le menu déroulant, choisissez le fichier à éditer
- cliquez sur le bouton Charger
- éditez le fichier dans la zone d'édition
- et cliquez sur le bouton Sauvegarder 'le' 'fichier'

## L'en-tête (header)

### Modifier l'image de fond du header

Créez l'image de fond que vous voulez :

- récupérez par FTP l'image **images/header.jpg** du header

code

- renommez-la **header-xxx.jpg**
- éditez-la selon vos désirs dans un éditeur d'image comme GIMP (garder les mêmes dimensions).
- renvoyez par FTP l'image modifiée sur le serveur à coté de l'original.

Éditez (voir plus haut comment) la feuille de style **static-xxx.css** pour modifier le nom de l'image à la ligne

```
#title {  
    background: url("images/header-xxx.jpg") no-repeat;
```

## Créer une page statique pour le nouveau header



La méthode est la même pour les autres zones de la page

Créez une nouvelle page statique et placez-y tout ce que vous voulez (sauf l'image de fond qui sera apportée par la feuille de style).

Supposons qu'elle ait le numéro **006** (**Identifiant** dans la liste des pages statiques)

## Appel de la page du nouveau Header

Éditez le fichier **static-xxx.php** pour remplacer la première ligne

```
<?php include(dirname(__FILE__).' /header.php'); # On insere le header ?>
```

par les lignes suivantes :

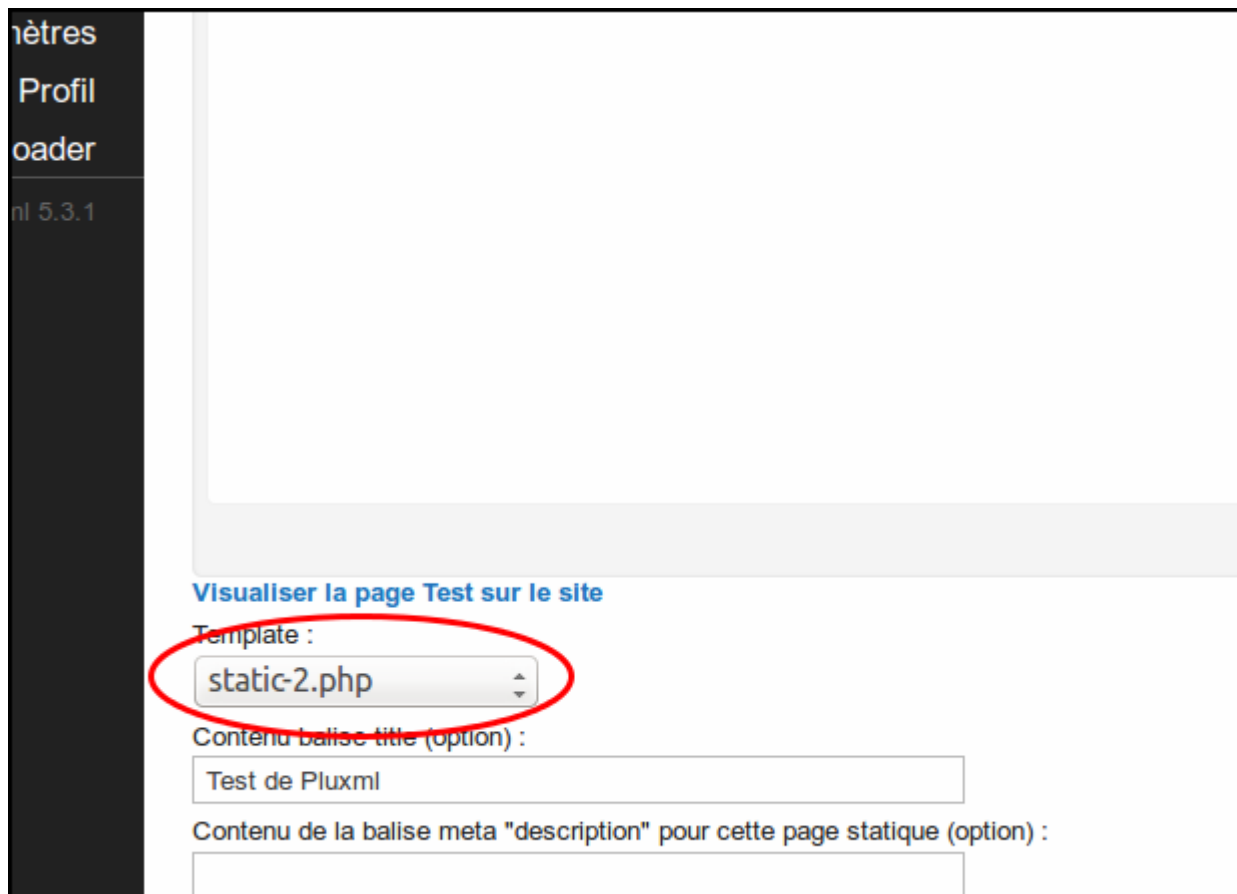
[static-xxx.php](#)

```
<?php $plxShow->templateCss() ?>  
<div id="title">  
    <?php $plxShow->staticInclude(6) ?>  
</div>  
<ul id="menu">  
    <?php $plxShow->staticList($plxShow->getLang('HOME'), '<li  
id="#static_id"><a href="#static_url" class="#static_status"  
title="#static_name">#static_name</a></li>', '', 0); ?>  
</ul>  
...
```



Mettre le numéro de la page statique du header dans  
→staticInclude(**x**)

Pour vérifier, dans l'administration, éditez une page et choisissez comme template **static-xxx.php** :



C'est tout. La page statique a changé de look...

En créant des templates différents, on ne touche pas aux pages déjà en ligne. Il suffit de modifier le template d'un clic pour le mettre en ligne.



Évidemment, si vous changez de thème, il faudra refaire les templates additionnels pour que cela fonctionne...

## La sidebar

### Les tags de la sidebar

Ouvrez votre page d'accueil.

La **sidebar** (menu dans l'espace de droite) est livrée d'origine avec des tags (ou instructions) qui affichent des informations que nous trouvons en général sur les sites modernes et surtout viennent apparaître automatiquement.

Il y a au départ 5 catégories :

<b>Fil RSS</b>	ressource (fichiers XML) dont le contenu est produit automatiquement selon les mises à jour d'un site Web. souvent utilisés par les sites d'actualité et les blogs pour présenter les titres des dernières informations consultables en ligne.
<b>Catégories</b>	pour ranger vos articles ou pages statiques dans des catégories pour mieux les retrouver et personnaliser leur affichage.
<b>Mots clefs</b>	peuvent apparaître sous forme de nuage de mots ou être trouvés par un moteur de recherche interne (plugin sous Pluxml) En cliquant sur ces mots clef, le visiteur retrouvera tous les articles qui parlent de ce mot là.
<b>Archives</b>	Pluxml répertorie toutes vos publications par mois et années, ce qui permet une recherche par date.
<b>Derniers articles</b>	Derniers articles publiés
<b>Derniers commentaires</b>	Derniers commentaires des utilisateurs

Ces informations se modulent et on peut en rajouter où en enlever.

Par exemple, on peut enlever l'affichage des derniers commentaires, limiter (ou augmenter) le nombre de mots clefs, faire disparaître les archives, et placer les derniers articles ailleurs.

En fait, ce système est prévu pour les sites dédiés blog et les internautes ont l'habitude de ce genre d'affichage dans le menu vertical.

Pour ce qui nous concerne, comme on fait surtout un site (équipé d'un blog si nous le voulons), nous pouvons changer complètement cet affichage pour l'adapter à nos besoins (un menu perso par exemple) quitte à réserver la sidebar originale à la partie blog.

Nous modulerons cette ou ces sidebars comme nous l'entendons parce qu'avec Pluxml, nous pouvons avoir un menu (et des couleurs) différent pour chaque page à n'importe quel niveau du site.

## Modifier la sidebar

Pour examiner le code de la sidebar, ouvrez, dans **admin/Paramètres/Options d'affichage/Éditer** les fichiers du thème «theme-votre-theme», en choisissant le fichier **sidebar.php** dans la liste déroulante. Il s'affiche ceci :

```
<?php if(!defined('PLX_ROOT')) exit; ?>
<div id="sidebar">
  <div class="section">
    <h3><?php $plxShow->lang('RSS_FEEDS') ?></h3>
    <ul>
      <li><a class="feed noactive" href="<?php $plxShow->urlRewrite('feed.php') ?>" title="<?php $plxShow->lang('ARTICLES_RSS_FEED_TITLE') ?>"><?php $plxShow->lang('ARTICLES_RSS_FEED') ?></a></li>
      <li><a class="feed noactive" href="<?php $plxShow->urlRewrite('feed.php?commentaires') ?>" title="<?php $plxShow->lang('COMMENTS_RSS_FEED_TITLE') ?>"><?php $plxShow->lang('COMMENTS_RSS_FEED') ?></a></li>
```



```

        </ul>
    </div>

    <div class="section">
        <h3><?php $plxShow->lang('CATEGORIES')?></h3>
        <ul>
            <?php $plxShow->catList('', '<li><a href="#cat_url"
class="#cat_status">#cat_name (#art_nb)</a></li>'); ?>
        </ul>
    </div>

    <div class="section">
        <h3><?php $plxShow->lang('TAGS')?></h3>
        <p>
            <?php $plxShow->tagList('<a href="#tag_url" class="tag
#tag_status">#tag_name</a> ', 20); ?>
        </p>
    </div>

    <div class="section">
        <h3><?php $plxShow->lang('ARCHIVES')?></h3>
        <ul>
            <?php $plxShow->archList('<li><a href="#archives_url"
class="#archives_status">#archives_name (#archives_nbart)</a></li>'); ?>
        </ul>
    </div>

    <div class="section">
        <h3><?php $plxShow->lang('LATEST_ARTICLES')?></h3>
        <ul>
            <?php $plxShow->lastArtList('<li><a href="#art_url" title="#art_title"
class="#art_status">#art_title</a></li>'); ?>
        </ul>
    </div>

    <div class="section">
        <h3><?php $plxShow->lang('LATEST_COMMENTS')?></h3>
        <ul>
            <?php $plxShow->lastComList('<li><a
href="#com_url">#com_content(33)</a></li>'); ?>
        </ul>
    </div>
</div>

```

Il s'agit d'une succession de <div> qui contiennent chacun une instruction. Par exemple :

```

<div class="section">
    <h3><?php $plxShow->lang('ARCHIVES')?></h3>
    <ul>
        <?php $plxShow->archList('<li><a href="#archives_url"
class="#archives_status">#archives_name (#archives_nbart)</a></li>'); ?>
    </ul>
</div>

```

```
</ul>  
</div>
```

affiche le mois concerné, la date et le nombre d'articles sur le site pour ce mois-là.

Pour supprimer l'affichage des archives dans la sidebar, il suffirait de supprimer ce `<div></div>` ou de le commenter pour ne plus l'afficher (et libérer de l'espace).



Gardez toujours une sauvegarde de vos fichiers avant vos essais. En cas de malheur, vous pourriez toujours récupérer un fichier original dans l'archive avec laquelle vous avez installé Pluxml.

Pour remplir la sidebar avec des informations qu'il sait mettre en avant avec un éditeur de texte comme au bon vieux temps, nous allons appeler dans cette sidebar une page préalablement conçue avec les informations à faire apparaître.

La manip est décrite dans cet article :

<http://tuto-pluxml.reseaux.info/article15/modifier-la-sidebar-pour-y-faire-apparaître-autre-chose>.

## Ajouter un favicon

Un favicon est une icône que les navigateurs web affichent dans la barre d'adresse, la barre de titre, les favoris, les onglets et les autres raccourcis.

Pour ajouter un favicon à PluXml :

éditer le fichier **header.php**

au-dessus de la ligne

```
</head>
```

ajouter le code suivant :

```
<link rel="shortcut icon" href="<?php $plxShow->template() ?>/favicon.png"  
>
```

copier le fichier **favicon.png** dans le dossier du thème

## Voir aussi

—

Basé sur [Ajouter un favicon](#) par Pluxml.org.

<sup>1)</sup>

template fourni

From:

<https://nfrappe.fr/doc-0/> - **Documentation du Dr Nicolas Frappé**

Permanent link:

<https://nfrappe.fr/doc-0/doku.php?id=logiciel:internet:pluxml:themes:start>



Last update: **2022/08/13 22:14**