

## Chapitre 3 : La structure des thèmes

A. L'objectif .....	72
B. Les fichiers des thèmes .....	72

## A. L'objectif

Nous allons voir dans ce chapitre comment sont structurés les thèmes WordPress et quels sont les fichiers qui peuvent intervenir pour créer un thème.

Pour ce faire, nous allons nous baser sur une structure « classique », voire « historique », des thèmes pour WordPress. Les termes « classique » et « historique » représentent l'affichage des sites WordPress sous forme de blog. Dans ce type d'affichage, nous avons usuellement un en-tête en haut, contenant une barre de menus pour la navigation. En dessous l'écran est divisé en deux, avec à gauche l'affichage des articles du blog sous forme chronologique et à droite, la colonne latérale, la *sidebar* en anglais, affichant les widgets. Pour terminer, en bas, nous avons le pied de page.

Cette architecture tend à disparaître au profit d'un affichage sous la forme de site web plus orienté entreprise, plus *corporate*, l'affichage en blog étant plutôt réservé aux particuliers et indépendants publiant régulièrement des articles.

Ce qu'il faut bien comprendre c'est que chaque concepteur de thème peut structurer comme il veut ses thèmes, avec les fichiers qu'il souhaite. Donc chaque structure de thème pourra être unique, même s'il existe des éléments constants et obligatoires.

## B. Les fichiers des thèmes

### 1. Les fichiers obligatoires

Les fichiers strictement obligatoires sont au nombre de deux :

- **index.php** : c'est le fichier qui est utilisé quand aucun autre fichier de mise en page (template) n'est présent dans le dossier de votre thème. Il est aussi utilisé pour afficher la page d'accueil de votre site.
- **style.css** : c'est le fichier des styles CSS qui sont utilisés pour la mise en forme et la mise en page du thème. Ce fichier contient aussi des paramètres indispensables pour la gestion des thèmes dans l'interface d'administration de WordPress.

### 2. Le fichier de personnalisation

Le fichier **functions.php** va permettre, entre autre et si besoin est, d'ajouter des fonctionnalités de paramétrage, des options spécifiques de personnalisation du thème.

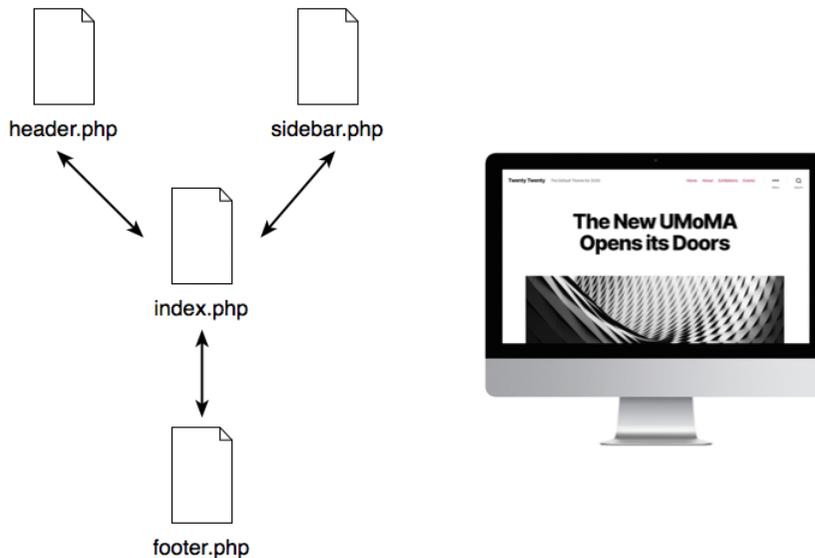
### 3. Les fichiers de structure

Pour créer un thème, vous pouvez utiliser plusieurs fichiers. Chaque fichier va s'occuper de l'affichage d'une partie des pages présentes à l'écran. Cela va permettre d'éviter de répéter plusieurs fois la même structure.

Avec une structure « classique », vous avez :

- le fichier **index.php** qui fait office de "metteur en scène". C'est dans ce fichier que sont appelés les autres fichiers spécialisés dans tel ou tel type d'affichage.
- le fichier **header.php** s'occupe exclusivement de l'affichage de l'en-tête.
- le fichier **footer.php** s'occupe exclusivement de l'affichage du pied de page.
- le fichier **sidebar.php** s'occupe exclusivement de l'affichage de la colonne latérale lorsqu'elle existe.

Ces trois derniers fichiers sont appelés par le fichier **index.php** pour afficher toutes les pages du site.



## 4. La structure du thème TwentyTwenty

La structure du thème par défaut de WordPress 5.3, **Twenty Twenty**, n'est pas strictement identique. En effet, dans ce thème il n'y a pas de fichier pour la colonne latérale de gauche car cette dernière n'est pas présente. Il n'y a donc pas de fichier **sidebar.php**.

Voici comment sont utilisés les fichiers pour gérer l'affichage dans le thème **Twenty Twenty** de WordPress 5.3 :



## 5. L'appel des fichiers et des fonctions

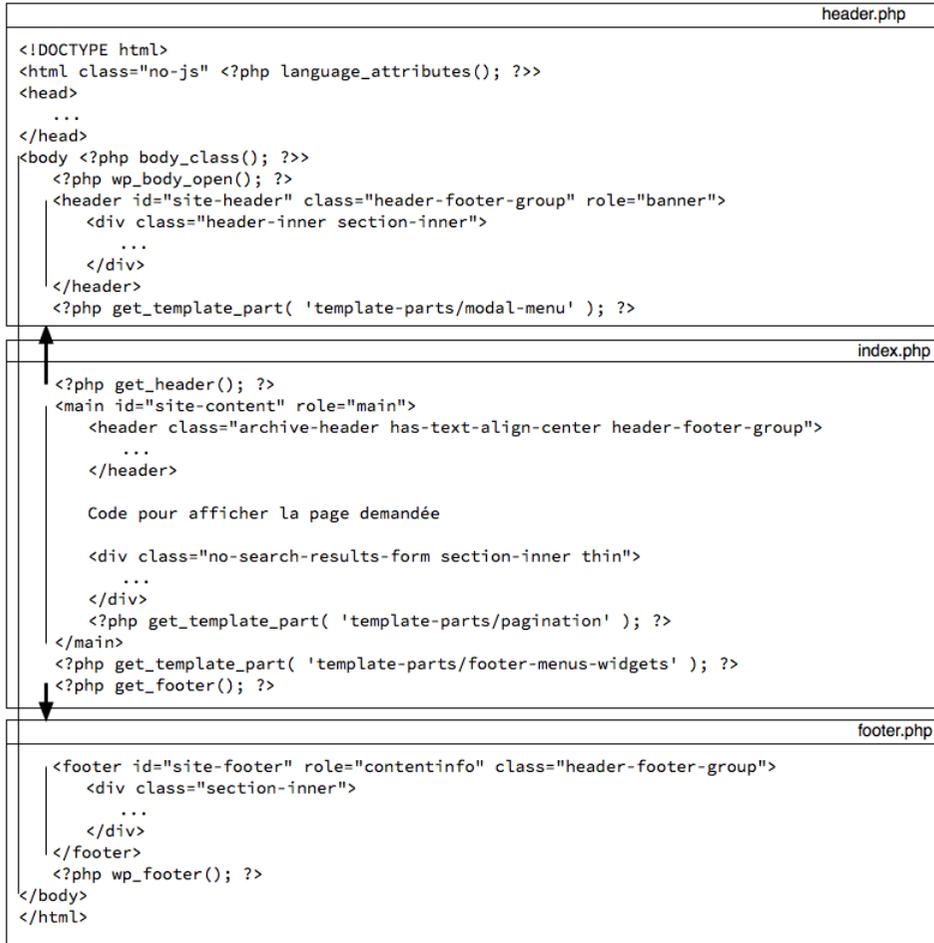
Comment les fichiers constitutifs sont-ils appelés et inclus dans la page **index.php** ? Grâce à des fonctions propres à WordPress :

`<? get_header(); ?>` : permet l'insertion du fichier **header.php** dans la page qui utilise cette fonction. Voici l'URL de cette fonction sur le site des développeurs de WordPress : [https://developer.wordpress.org/reference/functions/get\\_header/](https://developer.wordpress.org/reference/functions/get_header/)

`<? get_footer(); ?>` : permet l'insertion du fichier **footer.php** dans la page qui utilise cette fonction. Voici l'URL de cette fonction sur le site des développeurs de WordPress : [https://developer.wordpress.org/reference/functions/get\\_footer/](https://developer.wordpress.org/reference/functions/get_footer/)

Dans cette structure, vous notez aussi l'appel à deux fonctions WordPress. La première est placée dans le fichier **header.php**, c'est `wp_body_open()`. Elle permet d'insérer du code juste après l'ouverture de l'élément `<body>`. Ce qui est très utile par exemple pour gérer les fonctionnalités de **Google Tag Manager**, ajouter des scripts ou des métadonnées supplémentaires. Voici sa référence sur le site des développeurs WordPress : [https://developer.wordpress.org/reference/functions/wp\\_body\\_open/](https://developer.wordpress.org/reference/functions/wp_body_open/). La deuxième est `wp_footer()` qui est insérée dans le fichier **footer.php**, juste avant la balise de fermeture de l'élément `<body>`. Elle permet aussi d'insérer du code avant la fermeture du corps de la page.

Voici la structure simplifiée des fichiers avec le thème **Twenty Twenty**.



## 6. Le code et l'affichage généré

Voici le code généré et très simplifié par le thème **Twenty Twenty** de WordPress 5.3 pour l'affichage de la page d'accueil en blog :

```

<!DOCTYPE html>
<html class="no-js" lang="fr-FR">
  <head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" >
    <title>...</title>
    <script>...</script>
    <style>...</style>
  </head>
  <body class="home blog logged-in admin-bar no-customize-support
enable-search-modal has-no-pagination showing-comments show-avatars
footer-top-visible">
    <a class="skip-link screen-reader-text" href="#site-content">Skip to
the content</a>
    <header id="site-header" class="header-footer-group" role="banner">
      <div class="header-inner section-inner">
        ...
      </div>
      <div class="search-modal cover-modal header-footer-group"
data-modal-target-string=".search-modal" aria-expanded="false">
        ...
      </div>
    </header>
    <div class="menu-modal cover-modal header-footer-group"
data-modal-target-string=".menu-modal" aria-expanded="false">
      ...
    </div>
    <main id="site-content" role="main">
      <article class="post-1 post type-post status-publish
format-standard hentry category-non-classe" id="post-1">
        ...
      </article>
    </main>
    <div class="footer-nav-widgets-wrapper header-footer-group">
      ...
    </div>
    <footer id="site-footer" role="contentinfo"
class="header-footer-group">
      ...
    </footer><!-- #site-footer -->
    <script>...</script>
  </body>
</html>>

```