

WordPress Créer un thème simple

Issus de wpPourLesNuls.com

I. Débuter la création de votre thème WordPress

Dans le processus de développement d'un thème ou de modification d'un site WordPress, sachez que vos modifications, votre programmation et vos ajouts de fichiers se feront à l'intérieur du dossier « wp-content » situé à la racine de votre dossier d'installation.

Je vous montre ici comment créer la base de votre thème. Si par la suite vous souhaitez approfondir vos connaissances, le [Codex de WordPress](#) et [StackOverflow](#) sont d'excellentes ressources en ligne.

Commençons par nous diriger dans le dossier des thèmes situés dans « wp-content -> themes ». Vous y trouverez les thèmes par défaut **Twentyfifteen**, **Twentysixteen**, **Twentyseventeen**. Créez-y un nouveau dossier du nom de votre thème. Seulement 2 fichiers sont nécessaires pour que WordPress reconnaisse votre nouveau thème : **style.css** et **index.php**.

1. CSS – STYLE.CSS

Dans votre nouveau dossier, créez un fichier appelé « style.css ». Ajoutez-y les commentaires suivants :

```
/*
Theme Name: ThemeSimple
Author: Louis-Philippe Dea
Description: Theme simple pour Wordpress
*/
```

Modifiez bien sûr le nom du thème, le nom de l'auteur et la description. Vous pourrez ensuite y ajouter tous les styles, classes et IDs désirés pour personnaliser les visuels de votre site.

2. INDEX – INDEX.PHP

Voici un fichier de base pour vous aider à démarrer votre thème. Vous verrez que j'y ai ajouté la feuille de style de Bootstrap avec son adresse en CDN, son fichier JS (min) en bas de la page ainsi que le fichier jQuery.

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="">
<meta name="author" content="">
<title>Exemple de thème WordPress</title>
<!-- CSS de Bootstrap -->
<link
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css"
rel="stylesheet">
<!-- Ajout d'une nouvelle feuille de style qui sera spécifique à notre thème -->
```

```
<link href= "<?php echo get_stylesheet_directory_uri() . "/style.css" ?>"
rel="stylesheet">
<!-- Ajout d'une nouvelle feuille de style qui sera spécifique à nos blogs -->
<link href="<?php bloginfo('template_directory');?>/blog.css" rel="stylesheet">
<!-- HTML5 shim et Respond.js pour supporter les éléments HTML5 pour les versions
plus anciennes que Internet Explorer 9 -->
<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
<![endif]-->
</head>
<body>
<div class="header">
<div class="container">
<nav id="navigation-principale" role="navigation">
<a class="active" href="#">Accueil</a>
<a href="#">Services</a>
<a href="#">À Propos</a>
<a href="#">Contact</a>
</nav>
</div>
</div>
<div class="container">
<div class="blog-header">
<h1 class="blog-title">Exemple de thème WordPress</h1>
<p class="lead blog-description">Création d'un thème WordPress à titre de
tutoriel sur WP Pour Les Nuls.</p>
</div>
<div class="row">
<div class="col-sm-8 blog-main">
<div class="blog-post">
<h2 class="blog-post-title">Un article de blog</h2>
<p class="blog-post-meta">20 décembre 2017 par <a href="#">Louis-Philippe</a></p>
<p>Cum sociis natoque penatibus et magnis <a href="#">dis parturient montes</a>,
nascetur ridiculus mus. Aenean eu leo quam. Pellentesque ornare sem lacinia quam
venenatis vestibulum. Sed posuere consectetur est at lobortis. Cras mattis
consectetur purus sit amet fermentum.</p>
</div><!-- /.blog-post -->
</div><!-- /.blog-main -->
<div class="col-sm-3 col-sm-offset-1 blog-sidebar">
<div class="sidebar-module sidebar-module-inset">
<h4>À Propos</h4>
<p>Etiam porta <em>sem malesuada magna</em> mollis euismod. Cras mattis
consectetur purus sit amet fermentum. Aenean lacinia bibendum nulla sed
consectetur.</p>
</div>
</div><!-- /.blog-sidebar -->
</div><!-- /.row -->
</div><!-- /.container -->
<footer class="blog-footer">
<p>Par <a href="https://www.wppourlesnuls.com/">WP Pour les Nuls</a>.</p>
</footer>
<!-- JavaScript de Bootstrap et jQuery -->
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></scrip
t>
</body>
</html>
```

Dans votre console de gestion WordPress, cliquez sur « Apparence -> Themes ». Vous y verrez votre nouveau thème et pourrez ainsi l'activer.

II. Séparer votre page selon la structure de pages WordPress

Pour l'instant dans votre thème, tout le code se trouve dans votre fichier **index.php**. Nous voulons que notre en-tête (header), notre pied-de-page (footer) et notre barre latérale (sidebar) se trouvent sur toutes les pages du site. C'est pourquoi nous diviserons notre fichier **index.php** en 4 fichiers distincts : **header.php**, **footer.php**, **sidebar.php** et **content.php**.

1. HEADER – HEADER.PHP

Dans **index.php**, copiez tout le code à partir de `<!DOCTYPE html>` jusqu'au titre du blog dans un nouveau fichier **header.php**. Nous allons aussi y ajouter le code `<?php wp_head(); ?>` juste avant la fin de la balise `</head>`. Cette fonction permet à des plugins et à différentes fonctions d'ajouter des éléments à cet endroit précis. Voici le code à ajouter dans votre **header.php** :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1">
<meta name="description" content="">
<meta name="author" content="">
<title>Exemple de thème WordPress</title>
<!-- CSS de Bootstrap -->
<link href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css"
rel="stylesheet">
<!-- Ajout d'une nouvelle feuille de style qui sera spécifique à notre thème -->
<link href="<?php bloginfo('template_directory');?>/blog.css" rel="stylesheet">
<!-- HTML5 shim et Respond.js pour supporter les éléments HTML5 pour Internet
Explorer 8 -->
<!--[if lt IE 9]>
<script src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
<script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
<![endif]>
<?php wp_head(); ?>
</head>

<body>
<div class="header">
<div class="container">
<nav id="navigation-principale" role="navigation">
<a class="active" href="#">Accueil</a>
<a href="#">Services</a>
<a href="#">À Propos</a>
<a href="#">Contact</a>
</nav>
</div>
</div>
<div class="container">
<div class="blog-header">
```

```
<h1 class="blog-title">Exemple de thème WordPress</h1>
<p class="lead blog-description">Création d'un thème WordPress à titre de tutoriel sur
WP Pour Les Nuls.</p>
</div>
```

* Vous remarquerez que nous ne fermons pas la balise `<div class="container">` puisque nous la fermerons dans notre fichier footer.php.

2. FOOTER – FOOTER.PHP

Pour le fichier footer.php, nous ajouterons la ligne de code `<?php wp_footer() ?>` juste avant la fermeture de la balise `</body>`.

```
</div> <!-- /.container -->
<footer class="blog-footer">
<p>Par <a href="https://www.wppourlesnuls.com/">WP Pour les Nuls</a>.</p>
</footer>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js"></scrip
t>
<?php wp_footer(); ?>
</body>
</html>
```

3. SIDEBAR – SIDEBAR.PHP

La plupart des blogs et des sites Internet possèdent une barre latérale pour y afficher différents types de contenus tels que des archives, des catégories d'articles, un module de recherche, des publicités, etc. Voici quoi ajouter pour le moment dans votre fichier sidebar.php.

```
<div class="col-sm-3 col-sm-offset-1 blog-sidebar">
<div class="sidebar-module sidebar-module-inset">
<h4>À Propos</h4>
<p>Etiam porta <em>sem malesuada magna</em> mollis euismod. Cras mattis
consectetur purus sit amet fermentum. Aenean lacinia bibendum nulla sed
consectetur.</p>
</div>
</div><!-- /.blog-sidebar -->
```

4. ZONE DE CONTENU – CONTENT.PHP

Content.php contiendra tout le contenu principal de vos articles.

```
<div class="blog-post">
<h2 class="blog-post-title">Un article de blog</h2>
<p class="blog-post-meta">20 décembre 2017 par <a href="#">Louis-Philippe</a></p>
<p>Cum sociis natoque penatibus et magnis <a href="#">dis parturient montes</a>,
nascetur ridiculus mus. Aenean eu leo quam. Pellentesque ornare sem lacinia quam
venenatis vestibulum. Sed posuere consectetur est at lobortis. Cras mattis
consectetur purus sit amet fermentum.</p>
</div><!-- /.blog-post -->
```

5. INDEX.PHP

Votre page `index.php` devrait maintenant être plutôt vide! Nous allons y ajouter un peu de code pour que la page puisse aller chercher le contenu de nos différents fichiers **header.php**, **content.php**, **sidebar.php** et **footer.php**.

```
<?php get_header(); ?>
<div class="row">
<div class="col-sm-8 blog-main">
<?php get_template_part( 'content', get_post_format() ); ?>
</div> <!-- /.blog-main -->
<?php get_sidebar(); ?>
</div> <!-- /.row -->
<?php get_footer(); ?>
```

Si vous rechargez votre page d'accueil, votre site devrait s'y afficher au complet avec les contenus des fichiers php que nous venons de créer.

III. Paramètres généraux

Avant de commencer à développer nos pages d'articles et nos pages de contenu, nous devons définir quelques paramètres de notre site WordPress. Dans notre fichier **header.php**, nous pouvons voir que le titre de notre site dans la balise `<title>`, notre titre dans la balise `<h1>` et la description de notre site sont (comme on appelle dans le milieu) « hardcodés » dans le fichier. Donc rien de dynamique. Nous souhaitons que ceux-ci soient modifiables dans notre console de gestion de contenu. Repérez donc ceci dans **header.php** :

```
<div class="blog-header">
<h1 class="blog-title">Exemple de thème WordPress</h1>
<p class="lead blog-description">Création d'un thème WordPress à titre de
tutoriel sur WP Pour Les Nuls.</p>
</div>
```

Et remplacez-le par ceci :

```
<div class="blog-header">
<h1 class="blog-title"><a href="<?php echo get_bloginfo( 'wpurl' );?>"><?php echo
get_bloginfo( 'name' ); ?></a></h1>
<p class="lead blog-description"><?php echo get_bloginfo( 'description' ); ?></p>
</div>
```

La ligne `<a href="<?php echo get_bloginfo('wpurl');?>">` ajouter un lien vers la page d'accueil de votre site web. Vous pourrez modifier votre titre et votre description dans votre console d'administration sous **Réglages -> Général**.

1. La loop WordPress

Nous allons ajouter du contenu dans nos pages et dans nos articles grâce à la loop WordPress. Il s'agit de la fonction la plus importante de WordPress puisque presque tous vos contenus textes de votre console seront appelés par l'entremise de celle-ci.

Dans la console, si vous cliquez sur Articles, vous verrez un article par défaut « Bonjour tout le monde ! » que WordPress a ajouté lors de notre installation. Notre but est maintenant d'afficher le contenu de cet article dans une page.

Pour y parvenir, nous allons modifier le contenu de notre fichier **index.php** par ceci :

```
<?php get_header(); ?>
<div class="row">
<div class="col-sm-8 blog-main">
<?php
if ( have_posts() ) : while ( have_posts() ) : the_post();
get_template_part( 'content', get_post_format() );
endwhile; endif;
?>
</div> <!-- /.blog-main -->
<?php get_sidebar(); ?>
</div> <!-- /.row -->
<?php get_footer(); ?>
```

La loop indique à WordPress que **SI** il y a des articles, **JUSQU'À CE QU'**il y ait des articles, **AFFICHE** les articles. Tout ce qu'il y a dans cette boucle sera alors répété pour tous les articles du blog. Dans notre cas, nous lui avons dit d'aller chercher le contenu du fichier **content.php**. Nous allons donc remplacer le contenu de **content.php** par ceci :

```
<div class="blog-post">
<h2 class="blog-post-title"><?php the_title(); ?></h2>
<p class="blog-post-meta"><?php the_date(); ?> par <a href="#"><?php
the_author(); ?></a></p>
<?php the_content(); ?>
</div><!-- /.blog-post -->
```

C'est super simple, nous lui indiquons d'afficher le titre, la date, l'auteur et le contenu de l'article.

2. RENDRE LA BARRE LATÉRALE DYNAMIQUE

Pour notre sidebar, nous pourrions y ajouter un bloc À Propos qui affichera la description de l'auteur (dans la section « Utilisateurs » de votre WordPress en cliquant sur l'auteur désiré) et une section Archives. Modifiez le code de la sidebar.php par ceci :

```
<div class="col-sm-3 col-sm-offset-1 blog-sidebar">
<div class="sidebar-module sidebar-module-inset">
<h4>À Propos</h4>
<p><?php the_author_meta( 'description' ); ?> </p>
<h4>Archives</h4>
<ol class="list-unstyled">
<?php wp_get_archives( 'type=monthly' ); ?>
</ol>
</div>
</div><!-- /.blog-sidebar -->
```

3. MENU

Nous allons rendre la navigation du thème WordPress dynamique. Nous allons utiliser le module de « Menus » dans votre panneau WordPress, mais par contre, celui-ci n'est pas disponible par défaut dans votre console. Pour le rendre accessible, ajoutez le code ci-dessous dans votre fichier **functions.php** :

```
<?php
register_nav_menus( array('menu-principal' => 'Menu principal') );
?>
```

Rendez-vous ensuite dans votre CMS sous **Apparence -> Menus**. Créez-y votre menu et associez celui-ci au « Menu principal ».

Modifiez ensuite les lignes suivantes dans votre fichier **header.php** pour afficher ce menu à votre site :

```
<div class="header">
<div class="container">
<nav id="navigation-principale" role="navigation">
<?php wp_nav_menu( array( 'theme_location' => 'menu-principal' ) ); ?>
</nav>
</div>
</div>
```

Pour styliser le menu, nous pourrions ajouter quelques propriétés CSS assez simples dans votre fichier **blog.css** :

```
.header {
background: #377bb5;
}
#navigation-principale li {
position: relative;
display: inline-block;
padding: 10px;
font-weight: 500;
}
#navigation-principale li a {
color: #fff;
}
```

4. PAGE – PAGE.PHP

J'aimerais finalement que l'interface de mes pages soit légèrement différente de la mise en page de mes articles de blog. Je me créerai donc un nouveau fichier **page.php**, j'y retirerai la barre latérale et indiquerai à mon contenu de prendre toute la largeur de la page. Pour ce faire, je vais ajouter ma loop dans une balise **<div class="col-sm-12"></div>**. Voici le code à ajouter dans un nouveau fichier **page.php** :

```
<?php get_header(); ?>
<div class="row">
<div class="col-sm-12">
<?php
if ( have_posts() ) : while ( have_posts() ) : the_post();
get_template_part( 'content', get_post_format() );
endwhile; endif;
?>
</div> <!-- /.col -->
</div> <!-- /.row -->
<?php get_footer(); ?>
```

5. Conclusion

Ce tutoriel n'a fait qu'effleurer le sujet de la création d'un thème WordPress. Il y a tant à apprendre avec cette console de gestion de contenu!