

## ----- 3WA - REDWOLF - WORDPRESS -----

L'objectif : que tous les contenus soient accessibles dans le back-office, et non pas dans les fichiers / dossiers.

Pré requis : reprendre les codes de la correction Redwolf

Bases du développement :

- le menu du haut utilisera la fonctionnalité "Menu"
- pour le diaporama, nous n'utiliserons pas flexslider mais nous installerons un plugin
- "Découvrez nos motos" correspond à un blog de 4 articles liés à une catégorie
- "Plus en détail" correspond à un widget texte
- "Lorem ipsum" correspond à un widget texte
- Le menu du bas s'affichera avec un widget de menu
- Le copyright de 3WA correspond à un widget texte

## PERSONNALISER-----

Dans Modifier mon profil : décocher "Afficher la barre d'outils lorsque vous visitez le site", c'est plus confortable pour faire du webdesign (à vous de voir).

Lorsque vous utilisez l'éditeur de Wordpress, n'oubliez pas qu'il y a deux modes : Visuel et Texte (HTML), vous pouvez privilégier ce mode HTML dans votre profil : "Désactiver l'éditeur visuel pour écrire"

Avec le bouton "Option de l'écran", afficher/masquer les différentes boîtes nécessaires pour le back-office.

Les fichiers du site sont modifiables via Apparence / Editeur, mais nous conservons l'habitude Editeur et travaillons directement sur les fichiers du serveur.

Dans Réglages / permaliens, champ "Structure personnalisée", indiquer `/%category%//%postname%/%`. Ainsi les URL reprendront vos titres d'articles/pages et catégories en SEF (alias d'URL simplifiées).

Si vous modifiez un titre, assurer vous de re-générer l'alias via la boîte "Identifiant".

Si vous êtes sur un serveur distant, vous devriez vous assurer dans Réglages généraux que les URL indiquent "www". Un redirect 301 redirigera alors `http://` vers `http://www`

## INTEGRER LES CONTENUS-----

Il convient de calibrer les différents formats d'image afin qu'elles ne soient pas déformées, cela est relatif à la maquette graphique.

Réglages des médias / Tailles des images

- Taille des miniatures : 213x160 (format blog, ('thumbnail'))
- Taille moyenne : 600x450 (format article ('large'))

Insérer alors les images Motos et slides dans Médias / Ajouter

Créer la catégorie "Nos produits" avec la description "Découvrez nos motos"

Créer les articles Motos 1 à 4 avec le texte d'introduction avant `<!--more-->` et le texte complet après.

Dans la boîte "Image mise en avant", sélectionner les images d'introduction (grâce à un code, les images en grand seront insérées automatiquement).

Créer les pages Notre concept, Nos tarifs, Contactez-nous, About, Clients

## CREATION DU THEME-----

>> Commençons par la création des quatre fichiers indispensables : style.css, index.php, header.php, footer.php

Nous découpons donc nos pages en trois parties, nous n'aurons ainsi pas à reproduire les contenus de header et footer sur les différents templates du corpus.

### ---"style.css"-----

C'est la déclaration du thème, et les styles. N'oubliez pas que ce fichier est à la racine du thème.

```
/*
```

```
Theme Name: redwolf
```

```
*/
```

Nous pourrions ajouter d'autres informations : description, version...

### ---"header.php"-----

C'est l'en-tête permanent du site

```
<!DOCTYPE html>
<html <?php language_attributes(); ?>>
<head>
<title><?php bloginfo('name'); ?></title>
<base href="<?php echo get_bloginfo('url') ?>/">
<meta charset="UTF-8">
<link rel="stylesheet" href="<?php bloginfo('stylesheet_url') ?>">
<?php wp_head() ?>
</head>
<body>
<header>
</header>
<main class="container">
```

Base href nous permet de définir les liens absolus ce qui nous évitera des erreurs de liens relatifs  
wp\_head doit toujours être en dernier, il insère des metas et des scripts.

### ---"index.php"-----

C'est le fichier par défaut, il est indispensable, même s'il n'est pas utilisé (ce seront les "templates" de notre thème qui seront appelés : page, category, single...)

```
<?php get_header(); ?>
```

Bienvenue sur Redwolf, ceci est "index", le template par défaut que nous n'utiliserons pas.

```
<?php get_footer(); ?>
```

Fondamentalement, ce fichier devrait indiquer the\_post() pour les templates que nous ne développons pas.

---"footer.php"-----

C'est le pied de page permanent du site

```
</main>
<footer class="container">
</footer>
<?php wp_footer(); ?>
</body>
</html>
```

wp\_footer insère des scripts (Wordpress, plugins...)

Ajouter alors:

- tous les styles CSS dans style.css (afin d'être plus à l'aise, nous passons le container à 1170px, le standard Bootstrap)
- le logo et le background dans /img
- la vidéo dans /video
- Normalize , Awesome et Bodega dans /fonts et /css avec les <link> respectifs.

Notons que les chemins relatifs seront du type : wp-content/themes/redwolf/... et côté CSS : url("img/bg\_body.svg")

>> on peut maintenant dans Apparence / Thèmes, activer Redwolf

Si vous souhaitez afficher l'image de votre thème, vous devez ajouter le fichier screenshot.png à la racine de votre thème.

>> Nous allons ajouter les widgets et les menus.

En effet, contrairement aux thèmes natifs, nous n'avons pas encore les onglets Widgets et Menus (nous ne développerons pas En-tête, Arrière-plan...)

Il va falloir préalablement déclarer les "hooks" dans le fichier "functions"

---"functions.php"-----

Le fichier "centre nerveux" des fonctionnalités

```
<?php
register_nav_menus( array(
'menuhaut' => __( 'Menu haut', 'redwolf' )
) );
function themeredwolf_widgets_init() {
register_sidebar( array(
'name' => __( 'accueil', 'redwolf' ),
'id' => 'accueil',
'description' => __( 'Zone de widget Accueil', 'redwolf' ),
'before_widget' => '<section class="media">',
'after_widget' => '</section>',
) );
register_sidebar( array(
'name' => __( 'footer', 'redwolf' ),
'id' => 'footer',
'description' => __( 'Zone de widget texte pied de page', 'redwolf' ),
'before_widget' => "",
'after_widget' => "",
) );
}
```

```

));
register_sidebar( array(
    'name'      => __( 'menubas', 'redwolf' ),
    'id'        => 'menubas',
    'description' => __( 'Zone de widget Menu bas', 'redwolf' ),
    'before_widget' => "",
    'after_widget' => "",
));
register_sidebar( array(
    'name'      => __( 'copyright', 'redwolf' ),
    'id'        => 'copyright',
    'description' => __( 'Zone de widget Copyright', 'redwolf' ),
    'before_widget' => '<small id="licence">',
    'after_widget' => '</small>',
));
} add_action( 'widgets_init', 'themeredwolf_widgets_init' );

```

>> Nous dynamisons ainsi les contenus complémentaires aux articles et pages avec :

- 'menuhaut' (menu / register\_nav\_menus)
- 'accueil' (bloc "Nous connaître" et vidéo / widget de texte)
- 'footer' (bloc du pied de page / widget de texte)
- 'menubas' (menu / widget de menu / widget de menu)
- 'copyright' (bloc Copyright 3WA / widget de texte)

>> A présent, les liens d'administration Widgets et Menus apparaissent dans Apparence, et nous pouvons créer les items :

Créer deux menus (de liens de pages, catégorie, et lien personnalisé pour l'accueil (/votre-dossier de localhost) ) :

- Menu haut (concept, produits, tarifs, contact) en cochant "Menu haut" dans "Afficher l'emplacement"
- Menu bas (home, about, client, contact)

Créer quatre widgets (texte et menu) :

- Accueil (hook 'accueil') : tout le contenu de <section class="media">
- Footer (hook 'footer') : h2 + p
- Menu bas (hook 'menubas') : widget de menu personnalisé pour Menu bas
- Copyright (hook 'copyright') : tout le contenu de <small id="licence">

>> Tous les éléments sont prêts, commençons par intégrer l'en-tête et le pied de page

Dans <header>, ajouter :

- le premier enfant <nav class="container">
- qui contient le logo avec son ID et son lien vers l'accueil /redwolf (éviter le lien index.php qui génère du duplicate content)
- ainsi que <?php wp\_nav\_menu( array( 'theme\_location' => 'menuhaut' ) ); ?>

Dans <footer>, ajouter :

- <?php dynamic\_sidebar( 'footer' ); ?>

- <?php dynamic\_sidebar( 'menubas' ); ?>
- <?php dynamic\_sidebar( 'copyright' ); ?>

>> Nous allons maintenant décomposer le corpus du site en quatre "templates" Wordpress (catégorie, article, page, accueil) qui se substitueront au fichier index.

Tous ces templates commencent et finissent par :

- <?php get\_header(); ?>
- <?php get\_footer(); ?>

---"page.php"-----

Le fichier des pages

```
<article>
<h1><?php the_title(); ?></h1>
<?php echo get_post_field('post_content', $post->ID); ?>
</article>
```

---"category.php"-----

Le fichier des catégories

```
<h2><?php echo category_description(); ?></h2>
<section class="work">
<?php
if ( have_posts() ) : ?>
<?php
while ( have_posts() ) : the_post(); ?>
<article>
<?php echo the_post_thumbnail('thumbnail'); ?>
<h3><?php the_title(); ?></h3>
<?php the_content(); ?>
</article>
<?php endwhile;
else: ?>Articles prochainement
<?php endif; ?>
</section>
```

Après la description de catégorie, nous appelons la "boucle" (the loop)

Comme nous manipulons l'ordre d'affichage titre / texte / image des introductions d'article, nous insérons dans functions.php : add\_theme\_support( 'post-thumbnails' )

Cela affiche le bloc "Image à la une" dans l'édition les articles que nous pourrions donc utiliser pour insérer les images d'articles en introduction et en entier.

Côté CSS, l'alignement des quatre articles est fait en inline-block. Nous ne pouvons techniquement pas coller les balises.

- supprimer le display et le width de .work article
- passer .work en flex avec space-between

Nous avons bien sorti le h2 de .work (mais nous aurions pu utiliser flex-basis)

Notons que le style responsive .work img est incomplet, il manque height:auto

---"single.php"-----

Le fichier des articles entiers des catégories

```
<article>
<?php if (have_posts()) : ?>
<?php while (have_posts()) : the_post(); ?>
<h1><?php the_title(); ?></h1>
<div class="left"><?php echo the_post_thumbnail('large'); ?></div>
<?php the_content(); ?>
<?php endwhile; ?>
<?php endif; ?>
</article>
```

L'image en grand ('large') est appelée automatiquement grâce à la boîte "Image mise en avant".  
Nous l'ajoutons dans un élément flottant, qu'il faudra bien sûr désactiver en responsive.

---"front-page.php"-----

Le fichier de page d'accueil

```
<h1><?php bloginfo('description') ?></h1>
<h2><?php echo category_description(2); ?></h2>
<section class="work">
<?php query_posts('posts_per_page=4&cat=2'); if ( have_posts() ) : ?>
<?php while ( have_posts() ) : the_post(); ?>
<article>
<?php echo the_post_thumbnail('thumbnail'); ?>
<h3><?php the_title(); ?></h3>
<?php the_content(); ?>
</article>
<?php endwhile; ?>
<?php endif; ?>
</section>
<?php dynamic_sidebar( 'accueil' ); ?>
```

Notons qu'à la différence de la page Category, nous précisons le nombre d'articles à afficher (4), et l'ID de la catégorie (il pourrait y en avoir plusieurs).

Pour connaître l'ID d'une catégorie, éditez la, l'information se trouve dans l'URL.

Nous insérons le titre H1 grâce au slogan du site, qui s'ajoute dans Apparence / Personnaliser / Identité (nous ne pouvons pas insérer de span dans ce titre).

Les deux blocs Détails et Vidéo ne sont pas alignés car ils sont en inline-block et nous ne pouvons pas coller les balises.

Supprimer inline-block de .media article puis styliser .media .textwidget en flex et wrap, puis .media h2 en flex-basis:100%

Les widgets ont le défaut d'ajouter des <br>, coller les </i><i> pour que les icônes Awesome soient alignés.

>> Pour terminer, insérons le diaporama

- Dans Extensions, cliquer sur "Ajouter" puis rechercher, installer et activer "Slider by WD – Responsive Slider for WordPress" (WebDorado)
- Dans le nouveau menu Slider WD, cliquez sur Add new, ajouter un titre puis les trois slides à partir de la bibliothèque de médias.
- Dans Settings, indiquez les dimensions 1500x650
- Dans le fichier front-page.php, insérer alors le code:  

```
<section class="slider">
<?php wd_slider(2); ?>
</section>
```

## >> BONUS

### Optimiser les balises <title>

Dans le fichier header.php, nous avons indiqué <?php bloginfo('name'); ?> pour la balise <title> De la sorte, <title> est indentique sur toutes les pages du site (nom du blog), ce qui n'est pas bon pour le référencement.

Remplacer par <?php bloginfo('name'); ?> <?php is\_front\_page() ? bloginfo('description') : wp\_title(); ?>

Ainsi toutes les balises <title> commencent par Redwolf puis le titre de l'article ou de la page, à l'exception de la page d'accueil qui affiche le slogan.

### Ajouter la fonctionnalité "custom logo"

- insérer dans functions.php : add\_theme\_support( 'custom-logo' );
  - dans <header>, remplacer <img> par <?php the\_custom\_logo(); ?>
  - Dans Apparence / Thème / Personnaliser, Identité du site, insérer le logo (attention SVG refusé pour raisons de sécurité).
- Pour autant, est-il vraiment utile au client qu'il puisse changer le logo à volonté ?

### Insérer des meta-descriptions sur les articles et pages

- Activer via "Options de l'écran" la boîte de dialogue "Champs personnalisés"
- Indiquer le Nom "meta-description" (ou le sélectionner une fois créé), et remplissez la valeur (une phrase en 160 caractères d'après Google).
- Ajouter dans le <head> : <meta name="description" content="<?php echo get\_post\_meta( \$post->ID, 'meta-description', true ); ?>" />

Comprenez bien le potentiel des champs personnalisés : vous pourriez ainsi proposer à vos clients le champs "couleur" avec des valeurs prédéfinies (rouge, bleu...)

Ces valeurs pourraient alors servir de classes dans votre CSS afin que le client puisse lui même choisir les couleurs de blocs, titres et autres sur ses pages.

### Modifier le nom par défaut du lien de blog "(suite...)"

Dans le fichier "category", remplacer the\_content par the\_excerpt, cela supprime le lien.

A la suite, ajoutez <a href="<?php echo get\_permalink( \$post->ID ); ?>">En savoir plus</a>