

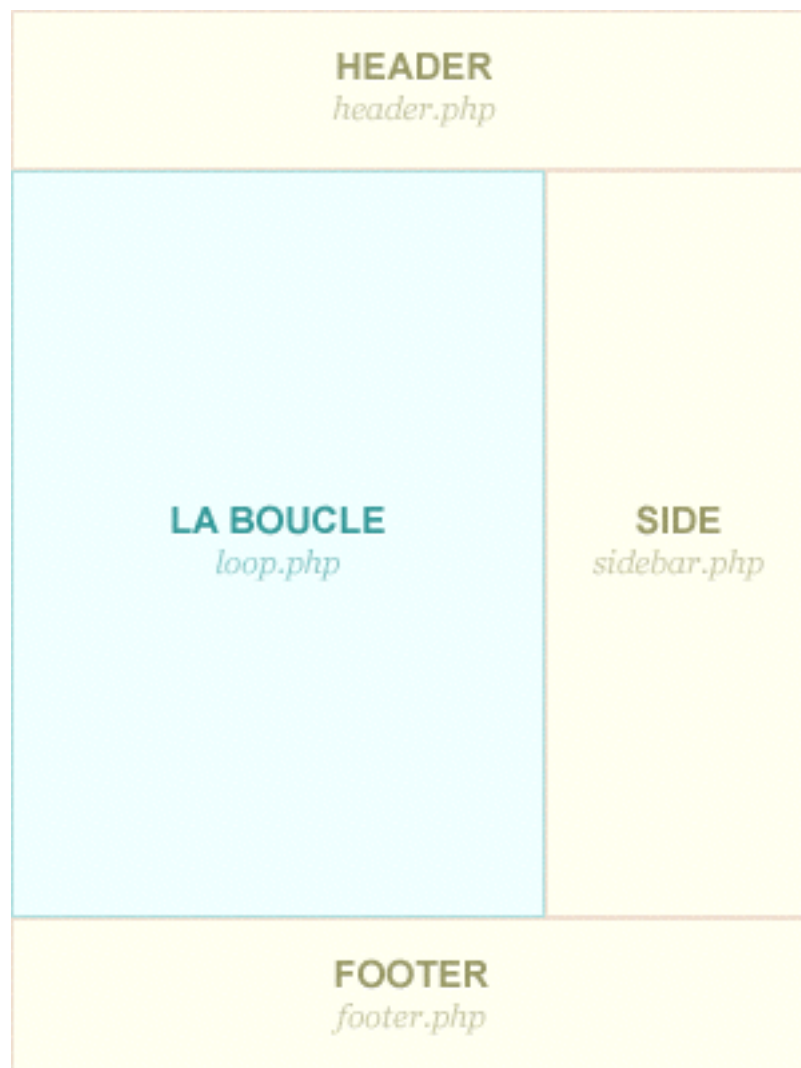


Wordpress pour les développeurs

Les boucles

en mode avancé

Rappel: la boucle



```
<?php if (have_posts()) : ?>  
    <!-- Si j'ai des Posts, j'affiche cette partie -->  
    <?php while (have_posts()) : the_post(); ?>  
        <!-- Pour CHAQUE Post, j'affiche ça -->  
    <?php endwhile; ?>  
<?php else : ?>  
    <!-- Si il n'y a pas de Post... -->  
<?php endif; ?>
```

Boucle personnalisée avec query_posts()

```
<!-- Include appelant le fichier header.php -->  
<?php get_header(); ?>
```

```
<div class="content">
```

```
<?php query_posts('posts_per_page=20'); ?>  
<!-- On modifie la requête par défaut de la boucle Wordpress -->  
<!-- On demande à la boucle par défaut d'afficher 20 articles par page -->
```

```
<?php get_template_part('loop'); ?>  
<!-- On appelle la boucle : loop.php -->
```

```
</div>
```

```
<?php wp_reset_postdata(); ?>  
<!-- On restaure la boucle à son fonctionnement par défaut -->
```

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

https://codex.wordpress.org/Function_Reference/query_posts

Boucles multiples avec query_posts()

```
<!-- Derniers articles -->
<h4 class="section">Derniers articles</h4>
<ul class="list">
    <?php wp_reset_postdata();
    query_posts( 'posts_per_page=5' );
    while (have_posts()) : the_post(); ?>
        <li><a href="<?php the_permalink(); ?> ">
            <?php the_title(); ?></a></li>
    <?php endwhile; ?>
</ul>
<?php wp_reset_postdata(); ?>
```

Boucles multiples avec WP_Query()

```
<?php
    $args = array( 'post_type' => 'post', 'posts_per_page' => 4 );
    $loop = new WP_Query( $args );
    while ( $loop->have_posts() ) : $loop->the_post(); ?>
```

```
<h3><?php the_title(); ?></h3>
```

```
<?php
    endwhile;
    wp_reset_postdata();
?>
```

https://codex.wordpress.org/Function_Reference/query_posts

query_posts() ≠ WP_Query()

- `query_posts` permet de modifier la boucle principale
- `WP_Query` permet de créer une deuxième boucle

`WP_Query` génère un objet PHP à transférer à la boucle
`$loop->the_post();`

`query_posts` permet d'utiliser les fonctions principales intégrées à WordPress.

https://codex.wordpress.org/Function_Reference/query_posts

Functions.php

Le plugin de votre thème

Support de fonctionnalités

add_theme_support

```
add_theme_support('post-thumbnails');
    add_theme_support('html5', ['caption', 'comment-form',
'comment-list', 'gallery', 'search-form']);
    add_theme_support('title-tag');
    add_theme_support( 'post-formats', array( 'video', 'aside',
'gallery' ) );
    add_theme_support( 'custom-background');
```

```
$args = array(
    'width'          => 560,
    'height'         => 120,
    'default-image' => get_template_directory_uri() . '/images/
default-header.jpg',
    'uploads'        => true,
);
add_theme_support( 'custom-header', $args );
```

dans header.php

```
<?php $header_image = get_header_image(); ?>

```

Ajout de formats d'images

add_image_size

```
add_image_size( 'custom-size', 220, 180 );  
// 220 pixels wide by 180 pixels tall, soft proportional crop mode
```

```
add_image_size( 'custom-size', 220, 180, true );  
// 220 pixels wide by 180 pixels tall, hard crop mode
```

```
add_image_size( 'custom-size', 220, 220, array( 'left', 'top' ) );  
// Hard crop left top
```

Les widgets

functions.php

```
if ( function_exists('register_sidebar') ) {  
    register_sidebar([  
        'name'          => __('Sidebar', TEXT_DOMAIN),  
        'id'            => 'sidebar-primary',  
        'before_widget' => '<section class="widget %1$s %2$s">',  
        'after_widget'  => '</section>',  
        'before_title'  => '<h3>',  
        'after_title'   => '</h3>'   
    ]);  
}
```

dans un template:

```
<?php dynamic_sidebar('sidebar-primary'); ?>
```

Les menus

Déclaration du ou des menu(s) (functions.php)

```
register_nav_menus( array(  
    'primary' => 'Primary Menu'  
) );
```

Appel dans le template :

```
<?php  
wp_nav_menu( array(  
    'menu'           => 'primary',  
    'theme_location' => 'primary',  
    'depth'          => 2  
) ); ?>
```

Possibilité de redéfinir le code HTML rendu avec des 'Walkers'

https://codex.wordpress.org/Navigation_Menus

Les shortcodes

functions.php

```
//[foobar]  
function foobar_func( $atts ){  
    return "foo and bar";  
}  
add_shortcode( 'foobar', 'foobar_func' );
```

Dans un template :

```
<?php echo do_shortcode( '[foobar]' ); ?>
```

Dans l'éditeur Wysiwyg de Wordpress :

```
[foobar]
```

https://codex.wordpress.org/Shortcode_API

Les hooks

2 types de hook:

Action

permettent d'ajouter des fonctionnalités à **un moment précis durant le chargement d'une page** (ex : ajouter un widget, charger un script ou une css).

Filter

Les filtres permettent d'intercepter le rendu d'une fonction de wordpress et d'en modifier la sortie (ex : ajouter une classe CSS, modifier certains blocs HTML, changer la longueur de l'excerpt...).

[add_action](#), [do_action](#), [remove_action](#), [add_filter](#), [apply_filters](#)

Action

```
// Ajouter du texte après le header  
add_action( 'hook-name' , 'my_action_function' );
```

```
function my_action_function() {  
    // Vérifier qu'on est sur la page d'accueil  
    if ( is_front_page() ) {  
        echo "<div>Promotion du jour...</div>";  
    }  
}
```

https://codex.wordpress.org/Plugin_API/Action_Reference

http://adambrown.info/p/wp_hooks/hook/actions

Filter

```
//Change WordPress Excerpt length  
function mytheme_excerpt_length( $length ) {  
    return 200;  
}  
add_filter( 'excerpt_length', 'mytheme_excerpt_length' );
```

https://developer.wordpress.org/reference/functions/add_filter/

http://adambrown.info/p/wp_hooks/hook/filters

Créer ses propres types de contenus et taxonomies

Articles, Pages, Medias, Commentaires... **Mon type de contenu**

Catégorie, Étiquettes... **Ma taxonomie**

Custom Post Type

Permet l'ajout d'un nouveau type de contenu au même titre que 'post' et 'page'

Idéal pour administrer un portfolio, des citations, tout autre type de contenu...

C'est la propriété `capability_type` qui détermine si votre nouveau contenu fonctionnera comme un `post` (un *article*) ou bien comme une `page` (une *page*).

Custom Post Type

```
register_post_type(
    'projet',
    array(
        'label' => 'Projets',
        'labels' => array(
            'name' => 'Projets',
            'singular_name' => 'Projet',
            'all_items' => 'Tous les projets',
            'add_new_item' => 'Ajouter un projet',
            'edit_item' => 'Éditer le projet',
            'new_item' => 'Nouveau projet',
            'view_item' => 'Voir le projet',
            'search_items' => 'Rechercher parmi les projets',
            'not_found' => 'Pas de projet trouvé',
            'not_found_in_trash' => 'Pas de projet dans la corbeille'
        ),
        'public' => true,
        'capability_type' => 'post',
        'supports' => array(
            'title',
            'editor',
            'thumbnail'
        ),
        'has_archive' => true
    )
);
```

Custom Taxonomy

Permet l'ajout d'un nouveau type de taxonomie au même titre que 'category' et 'tags'

C'est la propriété **hierarchical** qui permet de déterminer si la nouvelle taxonomie créée est une catégorie (**true**) ou une étiquette (**false**).

Custom Taxonomy

```
register_taxonomy(  
    'type',  
    'projet',  
    array(  
        'label' => 'Types',  
        'labels' => array(  
            'name' => 'Types',  
            'singular_name' => 'Type',  
            'all_items' => 'Tous les types',  
            'edit_item' => 'Éditer le type',  
            'view_item' => 'Voir le type',  
            'update_item' => 'Mettre à jour le type',  
            'add_new_item' => 'Ajouter un type',  
            'new_item_name' => 'Nouveau type',  
            'search_items' => 'Rechercher parmi les types',  
            'popular_items' => 'Types les plus utilisés'  
        ),  
        'hierarchical' => true  
    ),  
);
```

Custom fields

Les champs personnalisés permettent d'ajouter des métadonnées supplémentaires dans vos contenus, qu'il s'agisse d'Articles, de Pages ou de Contenus personnalisés.

Gérées par des paires clés/valeurs:

- la clé représente le **nom** de la méta-donnée
- la valeur contient l'**information** associée

Custom fields

functions.php

```
add_action('wp_insert_post', 'mes_champs_personnalisés');
```

```
function mes_champs_personnalisés ($post_id) {  
    if ( $_GET['post_type'] != 'page' ) {  
        add_post_meta($post_id, 'champ_perso_1',  
'valeur_par_defaut', true);  
    }  
    return true;  
}
```

Dans un template :

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

OU :

```
get_post_meta($post_id, $cle, $unique);
```

https://codex.wordpress.org/Custom_Fields

Gestion avancée des custom fields avec :
(version gratuite et premium)

[https://wordpress.org/plugins/
advanced-custom-fields/](https://wordpress.org/plugins/advanced-custom-fields/)

<http://advancedcustomfields.com/>

Requêtes SQL avec \$wpdb

Requêtes SQL avec \$wpdb

La class globale \$wpdb

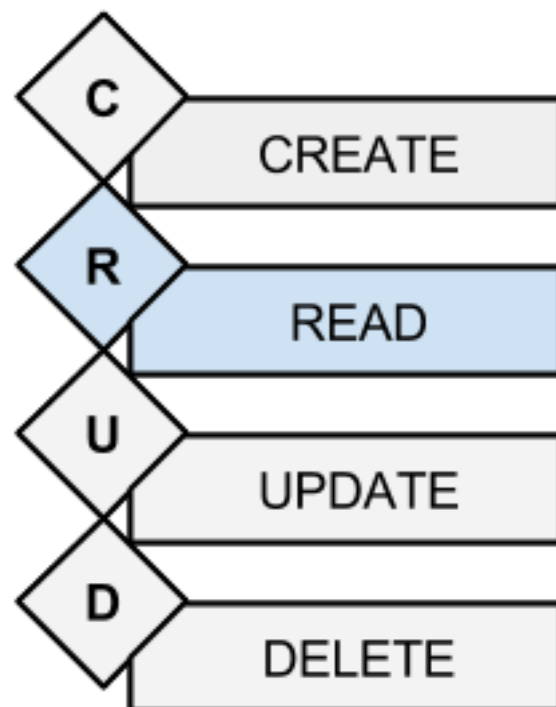
```
<?php global $wpdb; ?>
```

Récupération du préfix des tables

```
$wpdb->prefix
```

au choix :

```
$query = "SELECT * FROM {$wpdb->prefix}comments" ;  
$query = "SELECT * FROM $wpdb->comments" ;
```



```
global $wpdb;
```

```
// Interrogation de la base de données
```

```
$resultats = $wpdb->get_results("SELECT * FROM {$wpdb->prefix}posts") ;
```

```
// Parcours des resultats obtenus
```

```
foreach ($resultats as $post) {
```

```
    echo $post->post_title ;
```

```
    echo '<br/>' ;
```

```
}
```

Requêtes SQL avec \$wpdb

Get_results

```
global $wpdb;
```

```
// Interrogation de la base de données
```

```
$result = $wpdb->get_results("SELECT * FROM {$wpdb->prefix}posts") ;
```

```
// Parcours des resultats obtenus
```

```
foreach ($result as $post) {
```

```
    echo $post->post_title ;
```

```
    echo '<br/>' ;
```

```
}
```

Get_row

```
$result = $wpdb->get_row("SELECT post_title, post_date FROM {$wpdb->prefix}posts WHERE ID = 12") ;
```

```
echo $result->post_title . ' : ' . $result->post_date;
```

Requêtes SQL avec \$wpdb

Sécuriser ses requêtes:

```
$id = $_GET['id'];  
$result = $wpdb->get_results("SELECT * FROM {$wpdb->prefix}ma_table WHERE id = $id" );
```

```
$id = $_GET['id'];  
$result = $wpdb->get_results(  
    $wpdb->prepare(  
        "SELECT * FROM {$wpdb->prefix}ma_table WHERE id = %d",  
        $id  
    )  
);
```

<http://www.responsive-mind.fr/developper-pour-wordpress-wpdb-et-les-requetes-sql/>

Requêtes SQL avec \$wpdb

Manipulation de données:

`$wpdb->insert()`

`$wpdb->update()`

`$wpdb->delete()`

```
$wpdb->insert(  
    $wpdb->prefix.'ma_table',  
    array(  
        'nom' => 'Laetitia',  
        'activite' => 'Ski',  
        'date' => '2014-12-25',  
        'duree' => 4  
    ),  
    array(  
        '%s',  
        '%s',  
        '%s',  
        '%d'  
    )  
);
```

<http://www.responsive-mind.fr/developper-pour-wordpress-wpdb-et-les-requetes-sql/>

Publication du site

<http://www.mon-site.com> -> <http://www.autre-site.fr>

<http://localhost> -> <http://www.monsite.fr>

```
# Changer l'URL du site
UPDATE wp_options
SET option_value = replace(option_value, 'http://
www.ancien-site.com', 'http://www.nouveau-site.com')
WHERE option_name = 'home'
OR option_name = 'siteurl';
```

```
# Changer l'URL des GUID
UPDATE wp_posts
SET guid = REPLACE (guid, 'http://www.ancien-site.fr',
'http://www.nouveau-site.fr');
```

```
# Changer l'URL des médias dans les articles et pages
UPDATE wp_posts
SET post_content = REPLACE (post_content, 'http://
www.ancien-site.fr', 'http://www.nouveau-site.fr');
```

```
# Changer l'URL des données meta
UPDATE wp_postmeta
SET meta_value = REPLACE (meta_value, 'http://www.ancien-
site.com', 'http://www.nouveau-site.com');
```

D'autres solutions

<https://github.com/interconnectit/Search-Replace-DB>

<https://fr.wordpress.org/plugins/duplicator/>

Checklist sécurité WP

Sauvegardez !
(-> duplicator, BackupBuddy,
BackUpWordPress, Vaultpress...)
ou avec votre hébergement

Utiliser des mots de passe sécurisés

Changer le préfix des tables

Bloquez la lecture des répertoires apache

Supprimer le compte admin par défaut

Masquer les données métas de Wordpress
dans le header (plugin)
et supprimez le fichier readme.html
qui renseigne sur la version de votre
Wordpress

Protéger les droits d'accès aux fichier wp-
config.php à l'aide d'un .htaccess

```
<FilesMatch ^wp-config.php$>  
deny from all  
</FilesMatch>
```

et des droits d'utilisateurs unix:

```
chown -R www-data:www-data .  
find . -type f -exec chmod 644 {} +  
find . -type d -exec chmod 755 {} +  
chmod 644 wp-config.php
```

Masquez les erreurs de connexion :

```
add_filter('login_errors',create_function  
n('$a', "return null;"));
```

Vérifiez la sécurité de votre hébergeur

Utiliser des plugins de sécurité