



# Manuel de survie Wordpress

Rush

Staff 42 [piscine@42.fr](mailto:piscine@42.fr)

*Résumé: Ce document sert de kit de survie au Rush Wordpress.*

# Table des matières

<b>I</b>	<b>Wordpress, C'est quoi</b>	<b>2</b>
<b>II</b>	<b>Installation de Wordpress</b>	<b>4</b>
<b>III</b>	<b>Entracte</b>	<b>6</b>
<b>IV</b>	<b>Vue d'ensemble Admin</b>	<b>7</b>
<b>V</b>	<b>Codex</b>	<b>10</b>
<b>VI</b>	<b>Personnalisation/Le code</b>	<b>11</b>
<b>VII</b>	<b>Aller plus loin</b>	<b>19</b>

# Chapitre I

## Wordpress, C'est quoi

### 1. UN CMS

Wordpress est un CMS (Content Management System) permettant à n'importe qui de créer un blog ou site internet.

### 2. POURQUOI ?

Wordpress peut aussi bien permettre la création de blogs comme de sites vitrine ou e-commerce, ainsi tout type d'entité peut être intégré à cette plateforme. (Événements, actualités, services, témoignages, projets, pages... )

Wordpress possède une grande communauté et est facile d'accès, de plus les nombreuses mises à jour permettent une plus grande sécurité des sites propulsés par ce moteur.

### 3. EXEMPLE

Quelques exemples de site sous Wordpress :

- TechCrunch\*
- Frenchweb
- TEDx\*
- CNN\*
- TIME\*
- UPS\*
- LACTUALITE.COM
- 42.FR

\*clients VIP (source <http://vip.wordpress.com>)

#### 4. GESTION INTUITIVE

Wordpress est pensé pour être intuitif; aussi bien pour créer son site que pour l'installer.

Tout comme le fameux fichier install.php du RUSH0, wp-config.php permet cette installation.

# Chapitre II

## Installation de Wordpress

### 1. BESOINS (Serveur, php, mysql)

Wordpress est codé en PHP. Nous avons donc besoin d'un serveur où est installé PHP puis d'une base de données MySQL pour y stocker nos informations. De la même manière que pour les projets 42 nous aurons besoin d'un dossier `mamp/apps/wp/htdocs/[root_website]`

### 2. TELECHARGEMENT

Wordpress est distribué selon les termes de la GNU GPL. Pour bénéficier des bienfaits de Wordpress pas besoin d'y investir des sous seulement de se rendre sur la page (en Français) <http://www.wordpress-fr.net/> et de télécharger le « CMS LE PLUS UTILISE AU MONDE » \*

*\*(source wordpress.com) :)*

### 3. INSTALLATION

Une fois le CMS téléchargé, le décompresser, ajouter le code sur votre serveur (à la racine idéalement).

Accédez à votre site via votre navigateur préféré et laissez vous guidez, c'est intuitif.



(Si on vous propose d'indexer le site sur Google, REFUSEZ)

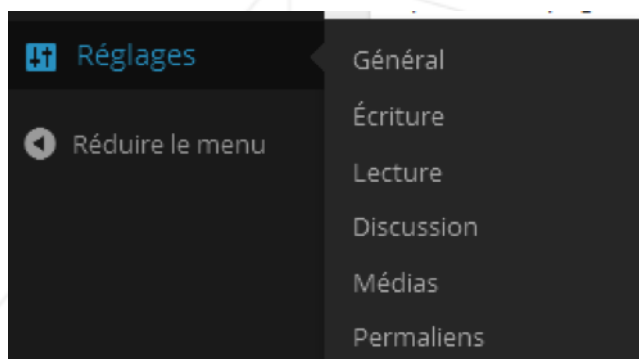
### 4. REGLAGES

Une fois l'installation du fichier `wp-config.php` fini, rendez-vous sur `wp.local.42.fr/wp-admin`.

Connectez-vous pour accéder à la zone d'administration de votre super nouveau site.

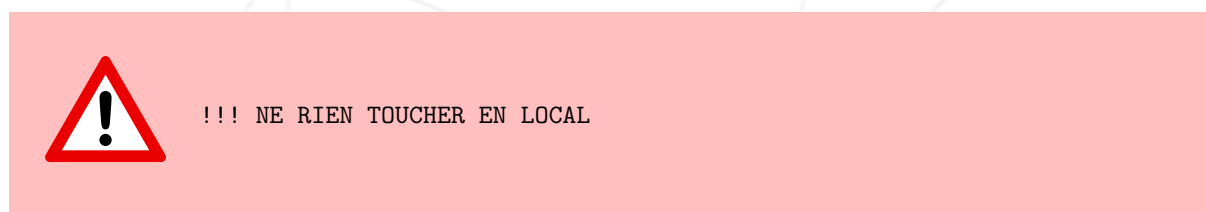
Réglages > Général

Laissez-vous guider



Réglages > Permalien

Cliquez sur 'Nom de l'article'



Ici on réécrit les liens. Exemple pour la catégorie « 123 » qui correspond à la catégorie « voiture ». <http://www.votresite.com/?cat=123> vers <http://www.votresite.com/voiture>

## 5. MISE A JOUR

Wordpress propose des mises à jour (actuellement la V 3.8 le 07/04/2014).

Nous n'aurons pas besoin de mettre à jour le site pour l'instant ; mais ne faites aucune mise à jour sans être sûr de ce que vous faites.

Cf. [http://codex.wordpress.org/fr:Mettre\\_a\\_Jour\\_WordPress](http://codex.wordpress.org/fr:Mettre_a_Jour_WordPress)

# Chapitre III

## Entracte



### Recette du Milkshake Banane

#### Ingrédients :

- Des bananes bananes
- De la glace vanille, caramel ou noix de coco
- Du lait
- Des glaçons
- Un mixeur

#### Préparation de la recette :

Coupez les bananes en morceaux et ajoutez-les dans un mixeur.

Ajoutez la (ou les) boule(s) de glace(s) ainsi que le lait.

Mixez pour obtenir une pâte liquide et homogène.

Pour la présentation, vous pouvez rajouter de la chantilly par dessus, saupoudré de s

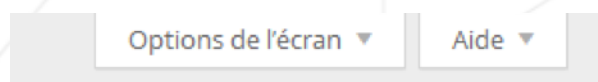


Un flamby ou un yahourt brassé vous permettront d'ajuster la consistance à votre goût

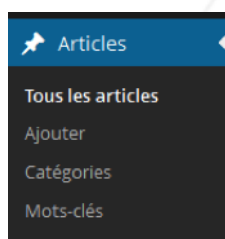
# Chapitre IV

## Vue d'ensemble Admin

A noter les onglets « Options de l'écran » et « Aide » en haut à droite de l'administrateur.



### 1. ARTICLES (Post)



**Post** : dynamique, chronologique

**Tous les articles**

L'onglet « Articles » nous envoie sur un écran d'administration des articles.

Pour chacun des articles on peut faire une « modification rapide » en modifiant Titre, Date, Auteur, Catégorie, Mots-Clés,...

Sinon en cliquant sur **Articles** **Ajouter**, on peut y ajouter : Nom de l'article, Corps de l'article (Format Visuel ou Texte (html)), Catégorie, Mots-clés, Images à la Une, Champs personnalisés...

### **Catégories**

L'onglet « Catégories » nous envoie sur une zone on l'on va pouvoir y ajouter nos catégories d'articles.

Exemple : L'article « J'aime les poneys » dans la catégorie « 42 »

### **Mot-clés**

L'onglet « Mots-clés » ou tags ou taxonomies nous envoie sur une zone on l'on va



pouvoir y ajouter nos mots clés.

Exemple : L'article « J'aime les poneys » dans la catégorie « 42 » ayant pour mots-clés « aimer », « poneys ».

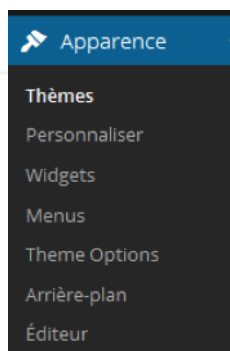
## 2. PAGE

**Page** : statique, intemporel

Lors de l'ajout d'une page on doit informer le modèle de cette page.

On peut personnaliser le modèle de page. *Cf. Créer un thème > page.php*

## 3. APPARENCE



### Thèmes

Ici on sélectionne le thème de notre site c'est-à-dire un fichier source qui va donner son apparence.

L'intérêt est de pouvoir créer son propre thème plutôt que d'en télécharger un payant.

*Cf. Créer un thème*

### Widgets

Très important, la partie « widgets » vous permettra d'ajouter un « widgets » sur votre site au niveau de la sidebar et/ou footer (par défaut selon le thème choisis)

On verra plus tard comment créer un “point d'ancrage” sur votre site pour récupérer ces widgets.

*Cf. Sidebar*

### Menu

Très important également, la partie « Menus » vous permettra d'ajouter un menu ou plusieurs à un ou des endroits de votre site selon votre thème.

On verra plus tard comment créer un “point d’ancrage” sur votre site pour récupérer les différents menus.

*Cf. Ajout de menu(s)*

### **Editeur**

La partie « éditeur » vous permettra d’accéder à tous les fichiers de votre thème (css y compris) pour pouvoir les modifier depuis l’administrateur. (Evitez de modifier le fichier functions.php depuis l’admin)

*Cf. Functions.php*

## **4. EXTENSIONS**

Les extensions ou plugins vont vous sauver la vie, c’est certain. Si vous souhaitez installer une newsletter vous téléchargez un plugin ‘newsletter’ vous l’installez et puis c’est tout.

A vous de trouver les meilleurs plugins en fonction de vos besoins où d’en créer un. Cf. Créer un plugin

De plus, un « éditeur » existe aussi pour cette partie ; vous pouvez simplement modifier le code source des plugins téléchargés ou créés pour encore plus de personnalisation.

## **5. UTILISATEURS**

Je vous laisse regarder cette partie seul. Cependant, jeter un coup d’œil ici :) :

*Cf. [https://codex.wordpress.org/Roles\\_et\\_Permissions](https://codex.wordpress.org/Roles_et_Permissions)*

# Chapitre V

## Codex

Le codex est là où vous devez passer le plus clair de votre temps pour la simple et bonne raison que toutes les fonctions de références de Wordpress y sont expliquées (équivalent à [php.net](http://php.net))

1. FUNCTIONS\_REFERENCE

*Cf. [https://codex.wordpress.org/Function\\_Reference](https://codex.wordpress.org/Function_Reference)*

# Chapitre VI

## Personnalisation/Le code

Le thème Wordpress est le cœur de votre projet web. Il va contenir toutes vos fonctions, vos fichiers CSS, PHP. C'est exactement ici que tout l'aspect structurel de votre site va se faire.

### 1. CREER UN THEME

#### *Structure*

- css/style.css : la feuille de style

```
/* Theme Name: XXXX
```

```
Description: XXXXXXXXXXXX
```

```
Version: X.X
```

```
Author: XXXXXX
```

```
*/
```

```
ICI CODE CSS
```

- Screenshot.png
- index.php //Page d'accueil

Porte d'entrer sur votre site.

```
<?php get_header() ?> //header.php
```

```
/*code*/
```

```
<?php get_sidebar() ?> //sidebar.php
```

```
<?php get_footer() ?> //footer.php
```

- header.php //En tête de votre site

```

<!DOCTYPE html>
<html <?php language_attributes(); ?>>
<head>
<meta charset="<?php bloginfo( 'charset' ); ?>" />
<meta name="viewport" content="width=device-width" />

<title><?php wp_title( '|', true, 'right' ); ?></title>

<link rel="profile" href="http://gmpg.org/xfn/11">
<link rel="pingback" href="<?php bloginfo( 'pingback_url' ); ?>">
<link rel="stylesheet" type="text/css" media="all"
href="<?php bloginfo( 'stylesheet_url' ); ?>" />
<!--[if lt IE 9]>
<script src="<?php echo get_template_directory_uri(); ?>/js/html5.js">
</script>
<![endif]-->

<?php wp_head(); ?>

</head>
<body>

<div id="page">
<div id="main">

/* code recur

```

- sidebar.php //Barre latérale sur votre site par défaut

```

<div id="sidebar">
/* code */

    <ul class="xoxo">
<?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar(1) ) :
// begin first sidebar widgets ?>

<?php if ( function_exists('wp_tag_cloud') ) : ?>
<li id="tag-cloud">
<h3 class="widget-title"><?php _e('Popular Tags'); ?></h3>
<div>
<?php wp_tag_cloud('smallest=8&largest=18&number=15'); ?>
//Va afficher les mots clefs les plus
//populaires par défaut sur la sidebar
//si il n'y a pas de widget installé.
</div>
</li><!-- end Popular Tags section -->

```

```

<?php endif; ?> //sinon on affiche les widgets

<?php wp_list_bookmarks('title_before=<h3 class="widget-title">
&title_after=</h3>&show_images=0') ?>

<?php endif; // end first sidebar widgets ?>

</ul>
</div>

```

On peut remarquer que la structure d'une sidebar est une grande DIV avec un grand UL et des LI

Dans functions.php //fichier où ranger toutes ses fonctions *cf.functions.php*

On initialize notre sidebar

```

<?php
function wpbx_widgets_init () {
if ( !function_exists('register_sidebars') ) {
return;
}
$p = array(
'before_widget' => '<li>',
'after_widget'   => "</li>\n",
'before_title'   => '<h3>',
'after_title'    => "</h3>\n"
);
register_sidebars( 1, $p );
}
add_action( 'init', 'wpbx_widgets_init' );
?>

```

Vous pouvez évidemment attribuer un style.

- footer.php //Pied de page

```

</div>
</div>

/* code recurrent
/ Exemple:
/ menu footer
/ copyright
*/

<?php wp_footer() ?>

```

```
</body>
</html>
```

- age.php //Modèle pour créer une page

Il est possible de créer son propre modèle de page.



copier/coller un modèle qui existe déjà puis le modifier.

```
<?php
/*
Template Name: XXXX
*/

/*
code
/exemple y ajouter le header du site, le footer, le corps, mais
pas la sidebar pour avoir une page full
*/
?>
```

Lors de l'ajout d'une page on doit choisir le modèle de cette page.  
Ici on décide de choisir le modèle XXXX que l'on vient de créer.



- single.php //Modèle pour créer un article  
Il est possible de créer des déclinaisons de thèmes mais vous trouverez seul.
- Loop

Vous permettra d'afficher le corps d'un article

```
<?php
if ( have_posts() ) {
while ( have_posts() ) {
the_post();
```

```
//
// Post Content here
//

} // end while
} // end if
?>
```

Cf. [http://codex.wordpress.org/The\\_Loop](http://codex.wordpress.org/The_Loop)

Cf. [http://codex.wordpress.org/Theme\\_Development#Anatomy\\_of\\_a\\_Theme](http://codex.wordpress.org/Theme_Development#Anatomy_of_a_Theme)

### **Conditionnal Tags**

Les fonctions INCLUDES sont les suivantes :

- is\_home() // si on est sur index.php
- is\_admin() // si on est Admin
- ...

Cf. [https://codex.wordpress.org/Conditional\\_Tags](https://codex.wordpress.org/Conditional_Tags)

## **2. SIDEBAR**

La sidebar Wordpress est un « point d'ancrage » sur le site où l'on va pouvoir y ajouter des widgets.

Sur les blogs par exemple la sidebar peut-être située sur la droite et/ou gauche et est composée de widgets comme un calendrier, un bloc de texte, un menu vertical, une newsletter etc...

### **Ajout d'une sidebar**

Cf. *Structure > sidebar.php*

Il suffit d'y ajouter :

```
register_sidebar( array(

    'name'      => 'my_sidebar',

    'before_widget' => '<div>',

    'after_widget'  => "</div>\n",

    'before_title'  => '<h3>',

    'after_title'   => "</h3>\n"

));
} // ici on ferme la fonction function wpbx_widgets_init()
```



pour ajouter une sidebar avec comme nom « my\_sidebar », sinon incrémenter

```
register_sidebars( 2, $p );
```

Le fichier my\_sidebar.php devra contenir le même code que sidebar.php en remplaçant :

```
<?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar(1) ) :  
// begin first sidebar widgets ?>
```

par

```
<?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar(2) ) : ?>
```

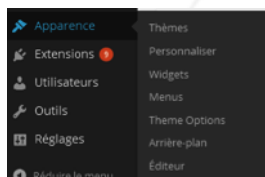
Inclure la sidebar dans votre thème avec

```
<?php include(TEMPLATEPATH."/ my_sidebar.php");?> \\
```

Cf. [https://codex.wordpress.org/Function\\_Reference/dynamic\\_sidebar](https://codex.wordpress.org/Function_Reference/dynamic_sidebar)

## Widgets

Un widget est un mini-plugin qui a son « point d'accroche » sur une sidebar. Lors de l'installation d'un plugin il est possible que celui-ci crée un widget.

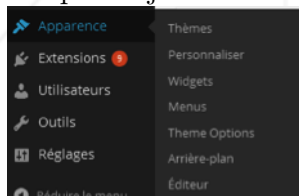


## 3. FUNCTIONS PHP

Le fichier functions.php est le fichier le plus important. C'est ici où sont rangées toutes les fonctions que vous allez créer pour votre thème. Ne modifiez pas ce fichier depuis votre éditeur, il se peut qu'il y ait des soucis.

## 4. AJOUT DE MENU(S)

On peut ajouter autant de menus que l'on veut.



## Point d'ancrage

Avant d'ajouter un menu il nous faut l'initialiser. Dans functions.php :

```
add_action( 'init', 'register_my_menus' );

function register_my_menus() {
    register_nav_menus(
        array(
            'primary-menu' => __( 'Main Menu' ), //ici un premier menu
            'secondary-menu' => __( 'Footer Menu' ) //ici un premier menu
            //Le nom du menu se fait depuis l'admin.
            //etc... ));}

```

Le « point d'ancrage » se fait en ajoutant cette ligne dans votre thème.

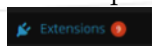
```
<?php wp_nav_menu( array( 'theme_location' => ' primary-menu',
'sort_column' => 'menu_order', 'container_class' => 'nav' ) ); ?>

```

Cf. [http://codex.wordpress.org/Function\\_Reference/wp\\_nav\\_menu](http://codex.wordpress.org/Function_Reference/wp_nav_menu)

## 5. AJOUT DE PLUGIN(S)

Les plugins peuvent être ajoutés depuis l'onglet « extensions ».



Vous pouvez rechercher vos plugins depuis l'admin ou sur <http://www.wordpress-fr.net/plugins/>

### Créer un plugin

On nomme le plugin de la façon suivante :

```
<?php
/*
Plugin Name: MON PREMIER PLUGIN WP
Description: DESCRIPTION DU PLUGIN
License: GPL ...
Author NOM DE L'AUTEUR
Version: 1.0
*/

function fb_add_custom_user_profile_fields( $user ) { // On ajoute les données ici
?>
<h3><?php _e('MORE INFOS', 'your_textdomain'); ?></h3>
<table class="form-table">
<tr>
<th>
<label for="address"><?php _e('Address', 'your_textdomain'); ?></label>
</th>
<td>
<input type="text" name="address" id="address"
value="<?php echo esc_attr( get_the_author_meta( 'address', $user->ID ) ); ?>"
class="regular-text" /><br />
<span class="description">

```

```
<?php _e('DESCRIPTION DU CHAMP', 'your_textdomain'); ?>
</span>
</td>
</tr>
</table>
<?php }

function fb_save_custom_user_profile_fields( $user_id ) {
// On sauvegarde les données ici

if ( !current_user_can( 'edit_user', $user_id ) )
return FALSE;

update_usermeta( $user_id, 'address', $_POST['address'] );
}

// On dit que fb_add_custom... va « agir » avec show_user_profile et
// edit_user_profile/on exécute
add_action( 'show_user_profile', 'fb_add_custom_user_profile_fields' );
add_action( 'edit_user_profile', 'fb_add_custom_user_profile_fields' );

// On dit que fb_save_custom... va « agir » avec personl_option_update et
// edit_user_profile_update /on exécute
add_action( 'personal_options_update', 'fb_save_custom_user_profile_fields' );
add_action( 'edit_user_profile_update', 'fb_save_custom_user_profile_fields' );

?>
```

On peut créer un plugin pour tout. Dans notre exemple on ajoute un champ « Address » pour permettre aux utilisateurs d'entrer leur adresse dans leur espace membre.

# Chapitre VII

## Aller plus loin

### 1. META-BOX

Les METAs-BOX sont des champs supplémentaires de type text, checkbox, etc... qui ont un « point d'ancrage » sur vos ajouts de pages, d'articles \*.

```
<?php

//Ici on ajoute les actions/exécutions
add_action( 'admin_menu', 'my_create_post_meta_box' );
add_action( 'save_post', 'my_save_post_meta_box', 10, 2 );

function my_create_post_meta_box() {

    /*Cf Codex pour voir les différents paramètres de la fonction add_meta_box()
    add_meta_box( 'my-meta-box', 'Second Excerpt',
    'my_post_meta_box', 'post', 'normal', 'high' );
    }

function my_post_meta_box( $object, $box ) { ?>
    <p>
        <label for="second-excerpt">Second Excerpt</label>
        <br />
        <textarea name="second-excerpt" id="second-excerpt" cols="60"
        rows="4" tabindex="30" style="width: 97%;">
        <?php echo wp_specialchars(
        get_post_meta( $object->ID, 'Second Excerpt', true ), 1 ); ?>
        </textarea>
        <input type="hidden" name="my_meta_box_nonce"
        value=
        "<?php echo wp_create_nonce(plugin_basename( __FILE__ ) ); ?>"
        />
    </p>
<?php }
```

```
//Ici on sauve

function my_save_post_meta_box( $post_id, $post ) {

    if ( !wp_verify_nonce( $_POST['my_meta_box_nonce'],
    plugin_basename( __FILE__ ) ) )
        return $post_id;

    if ( !current_user_can( 'edit_post', $post_id ) )
        return $post_id;

    $meta_value = get_post_meta( $post_id, 'Second Excerpt', true );
    $new_meta_value = stripslashes( $_POST['second-excerpt'] );

    if ( $new_meta_value && '' == $meta_value )
        add_post_meta( $post_id, 'Second Excerpt', $new_meta_value, true );

    elseif ( $new_meta_value != $meta_value )
        update_post_meta( $post_id, 'Second Excerpt', $new_meta_value );

    elseif ( '' == $new_meta_value && $meta_value )
        delete_post_meta( $post_id, 'Second Excerpt', $meta_value );
}
?>
```

Récupérer les données envoyées sur votre thème.

```
<?php if ( get_post_meta($post->ID, 'Second Excerpt', true) ) : ?>
//Cf. Codex get_post_meta()
<?php echo get_post_meta($post->ID, 'Second Excerpt', true) ?>
<?php endif; ?>
```

(source [Article de wpchannel.com](http://wpchannel.com))

## 2. CHAMPS PERSONNALISABLES

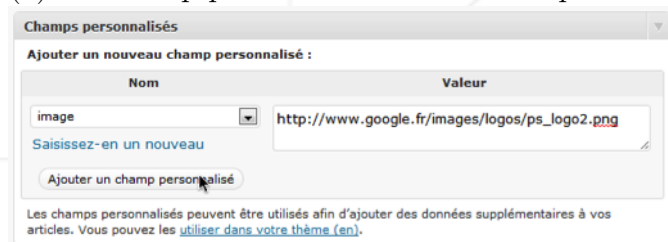
Les champs personnalisables permettent d'ajouter des métadonnées dans vos contenus.

On a deux possibilités d'ajout de champs personnalisés :

- Ajout depuis le type de contenu (1)
- Ajout automatiquement à tous les types de contenus choisis (2)

Hein ? Détaillons un peu...

(1) Un champ personnalisable est défini par une CLE (son nom) et une VALEUR



Wordpress va enregistrer les champs personnalisables dans la table wp\_postmeta.  
->phpmyadmin

Pour récupérer la valeur de votre champ on utilisera `get_post_meta()` qu'on pourra assigner à une variable.

```
$image = get_post_meta($post->ID, 'image', true); Cf. codex get_post_meta()
```

```
< ?php echo $image; ?> //affiche le lien du logo de votre ami Google
```

(2) On peut attribuer par défaut des champs à tous les posts, pages ... donc plus besoin d'ajouter manuellement.

```
add_action('wp_insert_post', 'wpc_champs_personnalises_default');
function wpc_champs_personnalises_default($post_id)
{
    if ( $_GET['post_type'] != 'page' ) {
        //on ajoute les champs par défaut sauf sur les pages on peut modifier

        add_post_meta($post_id, 'custom_field_1', '', true);
        //sans valeur par défaut
        add_post_meta($post_id, 'custom_field_2', 'ma_valeur_defaut', true);
        //avec valeur par défaut

    }
    return true;
}
```

### 3. CUSTOM POST TYPE

*«Pour nous punir de notre paresse, il y a, outre nos insuccès, les succès des autres.»*  
Jules Renard

### 4. TAXONOMY

Cf. [http://codex.wordpress.org/Function\\_Reference/register\\_taxonomy](http://codex.wordpress.org/Function_Reference/register_taxonomy)