

types de contenu personnalisés

(Custom Post Types, CPT)

PRÉPARÉ PAR:

Bernard Imbert

Jean-François Leblanc

Adapté par Mathieu Dupont

Pour le cours 420-964-EM

retour sur « *the loop* »

La boucle sert à afficher le contenu de plusieurs *posts* (articles) mais aussi d'autres types de contenus.

```
<!-- s'il y a au moins un article à afficher -->
<?php if ( have_posts() ) : ?>

    <!-- tant qu'il reste un article à afficher, on y avance --> the_post(); -->
    <?php while ( have_posts() ) : the_post(); ?>

        <!-- C'est ici le cœur de la boucle, où on affiche du contenu! -->

    <?php endwhile; ?> <!-- fin du while -->

<?php else : ?> <!-- s'il n'y a aucun article à afficher -->

    <p>Aucun article disponible!</p>

<?php endif; ?> <!-- fin du if -->
```

La boucle récupère automatiquement les « **posts** » correspondant au gabarit sélectionné :

- **page.php** : page sélectionnée
- **single.php** : article sélectionné
- **category.php** : tous les articles de la catégorie sélectionnée

Les fonctions comme **the_title()** ou **the_content()** doivent être utilisées à l'intérieur de la boucle **while()**.

Il est possible de modifier ou de créer une boucle pour afficher du contenu plus spécifique ou personnalisé.



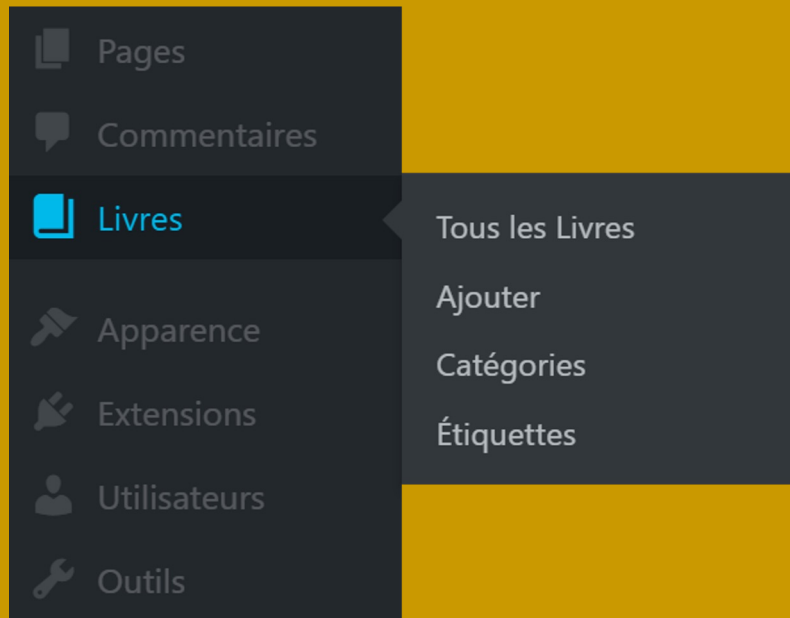
types de contenu par défaut dans WordPress

- **post (article)**

articles dynamiques affichés du plus récent au plus ancien pouvant être catégorisés

- **page**

article statique qui ne fait pas partie du fil de blogue



type de contenu personnalisé (cpt)

- c'est **comme des Articles**, mais personnalisés
- on peut en créer autant qu'on en a besoin (ex : Livres, Albums, Spectacles, etc.)
- c'est **mieux organisé!!**

créer des CPT

```
graph TD; A[créer des CPT] -.- B[01 en codant dans functions.php]; A -.- C[02 en utilisant une extension]; A -.- D[03 en utilisant un outil en ligne];
```

01

en codant dans
functions.php

02

en utilisant une
extension

03

en utilisant un
outil en ligne

1. en codant dans functions.php

```
// Type de contenu personnalisé (CPT)
function creer_mon_posttype() {
    register_post_type(
        'livres',
        array(// CPT Options
            'labels' => array(
                'name' => __( 'Livres' ),
                'singular_name' => __( 'Livre' )
            ),
            'public' => true,
            'has_archive' => true,
        )
    );
}
add_action( 'init', 'creer_mon_posttype' );
```

codex: register_post_type




**dans la réalité..
il y a un peu plus
de paramètres que ça...**



1. en codant dans functions.php...

```
// Register Custom Post Type
```

```
function cw5_partenaires() 
```

```
$labels = array(
    'name' => _x( 'Partenaires', 'Post Type General Name', 'text_domain' ),
    'singular_name' => _x( 'Partenaire', 'Post Type Singular Name', 'text_domain' ),
    'menu_name' => __( 'Partenaires', 'text_domain' ),
    'name_admin_bar' => __( 'Partenaire', 'text_domain' ),
    'archives' => __( 'Item Archives', 'text_domain' ),
    'attributes' => __( 'Item Attributes', 'text_domain' ),
    'parent_item_colon' => __( 'Parent Item:', 'text_domain' ),
    'all_items' => __( 'All Items', 'text_domain' ),
    'add_new_item' => __( 'Add New Item', 'text_domain' ),
    'add_new' => __( 'Add New', 'text_domain' ),
    'new_item' => __( 'New Item', 'text_domain' ),
    'edit_item' => __( 'Edit Item', 'text_domain' ),
    'update_item' => __( 'Update Item', 'text_domain' ),
    'view_item' => __( 'View Item', 'text_domain' ),
    'view_items' => __( 'View Items', 'text_domain' ),
    'search_items' => __( 'Search Item', 'text_domain' ),
    'not_found' => __( 'Not found', 'text_domain' ),
    'not_found_in_trash' => __( 'Not found in Trash', 'text_domain' ),
    'featured_image' => __( 'Featured Image', 'text_domain' ),
    'set_featured_image' => __( 'Set featured image', 'text_domain' ),
    'remove_featured_image' => __( 'Remove featured image', 'text_domain' ),
    'use_featured_image' => __( 'Use as featured image', 'text_domain' ),
    'insert_into_item' => __( 'Insert into item', 'text_domain' ),
    'uploaded_to_this_item' => __( 'Uploaded to this item', 'text_domain' ),
    'items_list' => __( 'Items list', 'text_domain' ),
    'items_list_navigation' => __( 'Items list navigation', 'text_domain' ),
    'filter_items_list' => __( 'Filter items list', 'text_domain' ),
);
```

codex: register_post_type



...en codant dans functions.php (suite)

```
$args = array(
    'label'           => __( 'Partenaire', 'text_domain' ),
    'description'     => __( 'Post Type Description', 'text_domain' ),
    'labels'          => $labels,
    'supports'        => array( 'title', 'editor', 'thumbnail' ),
    'taxonomies'      => array( 'category', 'post_tag' ),
    'hierarchical'    => false,
    'public'          => true,
    'show_ui'         => true,
    'show_in_menu'    => true,
    'menu_position'   => 5,
    'menu_icon'       => 'dashicons-buddicons-replies',
    'show_in_admin_bar' => true,
    'show_in_nav_menus' => true,
    'can_export'      => true,
    'has_archive'     => true,
    'exclude_from_search' => false,
    'publicly_queryable' => true,
    'capability_type' => 'page',
    'show_in_rest'    => true,
);
register_post_type( 'partenaire', $args );
```

« slug » utilisé
dans les gabarits

```
add_action( 'init', 'cw5_partenaires', 0 );
```

codex: register_post_type



2. en utilisant une extension

- Articles
- Médias
- Pages
- Commentaires
- Apparence
- Extensions
- Utilisateurs
- Outils
- Réglages
- CPT UI**
- Ajouter/modifier des types de publication
- Ajouter/modifier des taxonomies
- Types/taxonomies enregistrés

Configuration par défaut ▲

Slug du type de publication *

livres

Le nom/slug du type de publication. Utilisé pour les différentes requêtes pour le contenu du type de publication.

Les slugs doivent uniquement contenir des caractères latins alphanumériques. Le tiret bas doit être employé à la place des espaces. Définissez un « slug personnalisé de réécriture » pour permettre aux slugs d'utiliser des tirets dans les URL.

Libellé pour le pluriel *

Livres

Utilisé pour l'élément de menu d'administration de type de publication.

Libellé pour le singulier *

Livre

Utilisé quand un libellé au singulier est nécessaire.

Auto-remplir les libellés

[Remplissez les libellés supplémentaires en fonction des libellés choisis.](#)

Ajouter un type de publication

extension: custom post type ui



Tableau de bord

Articles

Médias

Pages

Commentaires

Livres

Apparence

Extensions

Utilisateurs

Outils

Réglages

CPT UI

Ajouter/modifier des
types de publication

Ajouter/modifier des
taxonomies

Types/taxonomies
enregistrés

Outils

Aide/Support

Types de publication enregistrés avec Custom Post Type UI.

[Ajouter un nouveau type de publication](#)

[Ajouter une nouvelle taxonomie](#)

Types de publication

Nombre total de types de publication Custom Post Type UI enregistrés : 1

Type de publication	Réglages	Supports	Taxonomies	Libellés	Hiérarchie des fichiers du thème
Modifier Livres (livres) Obtenir le code	<code>name: livres</code> <code>label: Livres</code> <code>singular_label: Livre</code> <code>description: ""</code> <code>public: true</code> <code>publicly_queryable: true</code> <code>show_ui: true</code> <code>show_in_nav_menus: true</code> <code>delete_with_user: false</code> <code>show_in_rest: true</code> <code>rest_base: ""</code> <code>rest_controller_class: ""</code> <code>has_archive: false</code> <code>has_archive_string: ""</code> <code>exclude_from_search: false</code> <code>capability_type: post</code> <code>hierarchical: false</code> <code>rewrite: true</code> <code>rewrite_slug: ""</code> <code>rewrite_withfront: true</code> <code>query_var: true</code> <code>query_var_slug: ""</code> <code>menu_position: ""</code> <code>show_in_menu: true</code> <code>show_in_menu_string: ""</code>	<code>title</code> <code>editor</code> <code>thumbnail</code>	—	—	<p>Exemples de noms de fichiers d'archives. archive-livres.php archive.php index.php</p> <p>Exemples de noms de fichiers de publications uniques. single-livres-post_slug.php * single-livres.php single.php singular.php index.php</p> <p>* Remplacez « post_slug » par le slug de l'article actuel.</p> <p>Manuel de hiérarchie des fichiers modèles de thème</p>

extension: custom post type ui



extension : pour ou contre?

avantages

- plus simple et visuel
- suivra d'un thème à l'autre

désavantages

- maintenance requise (mises à jour)
- un site pourrait ne plus se mettre à jour à cause de CPT UI
- dépendance au soutien du développeur
- ajout de requêtes à la base de données et baisse des performances



3. en utilisant un outil en ligne



Generators Snippets Join Login

Post Type Generator

i General Post Type Labels Options Visibility Query Permalinks Capabilities Rest API

Overview

Use this tool to create custom code for [Post Types](#) with [register_post_type\(\)](#) function.

Usage

- Fill in the user-friendly form.
- Click the "Update Code" button.
- Copy the code to your project.
- Or save it as a snippet and share with the community.

Examples

If you are still learning how to use this tool, check out the following examples:

- [Products](#)
- [Articles](#)
- [Testimonials](#)
- [Property](#)
- [Car](#)

Update Code

Save Snippet

référence: generatewp.com



les avantages d'un outil en ligne

- plus simple et visuel
- on **copie/colle** le code généré dans **functions.php**
- on garde les performances et pas de mise à jour d'extension
- On va utiliser **GenerateWP** !!