

WordPress : assigner un template sur mesure à un CPT et aux taxonomies

Assigner un template sur mesure à un CPT

Remarque : en absence de template spécifique, les **CPT** vont utiliser le **Template Hierarchy** et utiliser **archive.php** et **single.php**.

Créer le template personnalisé

Maintenant que vous avez votre CPT, vous pouvez créer un template spécifique pour les afficher correctement.

[Template-Hierarchy-CPTTélécharger](#)

Pour cela, créez un fichier dans votre thème WordPress avec le nom **single-mon_cpt.php** pour afficher les contenus individuels de votre CPT et **archive-mon_cpt.php** pour afficher les archives.

Dans les fichiers de templates nouvellement créés, vous pouvez utiliser les boucles WordPress pour afficher les contenus et les informations spécifiques à votre CPT et à votre taxonomie.

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

```
<h1 class="site_heading"><?php post_type_archive_title(); ?></h1>
```

```
<?php if( have_posts() ) : while( have_posts() ) : the_post(); ?>
    <div class="mon_cpt">
        <h2 class="mon_cpt_title"><a href="<?php the_permalink(); ?>"><?php
the_title(); ?></a></h2>
        <?php the_post_thumbnail(); ?>
    </div>
<?php endwhile; endif; ?>
```

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

Remarque : la fonction **post_type_archive_title()** est une fonction utilitaire dans WordPress qui sert à afficher le titre (ou l'intitulé) pour les archives d'un **CPT**.

Assigner un template sur mesure aux taxonomies

[Template-Hierarchy-Custom-Taxonomy-ArchiveTélécharger](#)

Pour assigner un template sur mesure aux archives d'une taxonomie personnalisée (custom taxonomy) dans WordPress, vous devrez créer un fichier de modèle spécifique pour cette taxonomie et le placer dans le thème actif de votre site.

Suivez ces étapes pour accomplir cela :

1. **Créer un fichier de modèle personnalisé** : tout d'abord, vous devez créer un fichier PHP pour votre archive de taxonomie personnalisée. Le nom du fichier doit être en fonction du **slug** (identifiant) de votre taxonomie. Par exemple, si votre taxonomie personnalisée a un slug « **genre** », le fichier de modèle doit être nommé **taxonomy-genre.php**.
2. **Trouver le slug de votre taxonomie personnalisée** : assurez-vous de connaître le slug de votre taxonomie personnalisée. Vous pouvez le trouver lorsque vous enregistrez votre taxonomie en utilisant la fonction **register_taxonomy()** dans le fichier **functions.php** de votre thème.
3. **Copier le fichier d'archive** : si vous avez déjà un fichier **archive.php** ou **taxonomy.php** dans votre thème, vous pouvez copier le contenu de ce fichier dans le fichier de modèle personnalisé que vous avez créé à l'étape 1. Sinon, vous pouvez créer le nouveau fichier en vous basant sur les besoins spécifiques de votre mise en page.
4. **Personnaliser le fichier de modèle** : dans le fichier de modèle personnalisé, vous pouvez utiliser les boucles WordPress pour afficher les publications associées à cette taxonomie personnalisée. Vous pouvez utiliser des fonctions comme **have_posts()** et **the_post()** pour parcourir les articles et afficher leurs contenus.
5. **Enregistrer les modifications** : assurez-vous de sauvegarder et de télécharger le fichier de modèle personnalisé dans le bon emplacement de votre thème. Le chemin typique est **/wp-content/themes/votre-theme/**.

Une fois que vous avez suivi ces étapes, WordPress détectera automatiquement le fichier de modèle personnalisé et l'utilisera pour afficher les archives de votre taxonomie personnalisée.

Notez que vous pouvez également utiliser d'autres modificateurs de fichiers de modèle personnalisé pour des configurations plus spécifiques.

Par exemple, pour un terme de taxonomie spécifique, vous pouvez utiliser le fichier **taxonomy-{taxonomy}-{term}.php** où « **{taxonomy}** » est le **slug de la taxonomie** et « **{term}** » est le **slug du terme spécifique**.

Remarque : en pratique, cela n'arrive presque jamais.

Les conditional tags pour CPT

Les **conditional tags** (balises conditionnelles) sont des fonctions intégrées dans WordPress qui permettent de vérifier différents contextes et états de page pour effectuer des actions ou afficher du contenu conditionnellement.

Pour les CPT, vous pouvez utiliser des conditional tags spécifiques pour cibler des pages ou des affichages liés à ces CPT.

is_singular()

Vérifie si vous êtes sur une page unique (single) d'un CPT spécifique ou non.

Vous devez passer le slug du CPT comme argument.

Par exemple :

```
if ( is_singular( 'mon_cpt' ) ) {
```

```
    // code à exécuter pour les pages uniques du CPT 'mon_cpt'
} else {
    // code à exécuter pour les autres pages
}
```

is_archive()

Vérifie si vous êtes sur la page d'archive d'un CPT spécifique.

Cela inclut la page d'archive principale du CPT et les pages d'archive pour les termes de taxonomie associés.

Vous devez passer le slug du CPT comme argument.

Par exemple :

```
if ( is_archive( 'mon_cpt' ) ) {
    // code à exécuter pour les pages d'archive du CPT 'mon_cpt'
} else {
    // code à exécuter pour les autres pages
}
```

is_post_type_archive()

Vérifie si vous êtes sur la page d'archive principale d'un CPT spécifique.

Vous devez passer le slug du CPT comme argument.

Par exemple :

```
if ( is_post_type_archive( 'mon_cpt' ) ) {
    // code à exécuter pour la page d'archive principale du CPT 'mon_cpt'
} else {
    // code à exécuter pour les autres pages
}
```

is_tax()

Vérifie si vous êtes sur une page d'archive de taxonomie spécifique pour un CPT donné.

Vous devez passer le slug de la taxonomie et le slug du CPT comme arguments.

Par exemple :

```
if ( is_tax( 'ma_taxonomie', 'mon_cpt' ) ) {
    // code à exécuter pour les pages d'archive de taxonomie pour le CPT
    'mon_cpt'
} else {
    // code à exécuter pour les autres pages
}
```

Ces conditional tags sont utiles pour personnaliser le contenu ou les fonctionnalités en fonction du type de publication et de la page sur laquelle vous vous trouvez.

Vous pouvez les utiliser dans le fichier **functions.php** de votre thème, dans des templates spécifiques ou dans des fichiers de plugins pour conditionner le comportement en fonction des CPT de votre site WordPress.