

WordPress : gérer la pagination et la navigation entre articles

Pour gérer la **pagination** et la **navigation entre les articles** dans un thème WordPress sur mesure codé en PHP, vous pouvez utiliser les fonctions et les classes fournies par WordPress.

Remarque : la pagination et la navigation vont améliorer le **maillage interne** permettant de plus facilement accéder aux contenus dans l'architecture du site web.

Pagination

Pour paginer les articles dans une boucle personnalisée, vous pouvez utiliser la fonction **posts_nav_link()** de WordPress.

Tout va ici s'effectuer dans les templates **archive.php** et **single.php**.

Voici un exemple de code pour paginer les articles WordPress :

```
<?php
// déterminer le nombre d'articles par page
$posts_per_page = 5;

// récupérer le numéro de page actuel
$paged = (get_query_var('paged')) ? get_query_var('paged') : 1;

// créer un nouvel objet WP_Query pour récupérer les articles paginés
$args = array(
    'post_type'      => 'post', // remplacer 'post' par le type de
    publication que vous souhaitez paginer
    'posts_per_page' => $posts_per_page,
    'paged'          => $paged,
);

$custom_query = new WP_Query($args);

// boucler à travers les articles paginés
if ($custom_query->have_posts()) :
    while ($custom_query->have_posts()) : $custom_query->the_post();
        // afficher les informations de l'article ici
        the_title('<h2>', '</h2>');
        the_content();
    endwhile;

    // afficher la pagination
    echo '<div class="pagination">';
    posts_nav_link();
    echo '</div>';
endif;
```

```
// réinitialiser les données de la requête après la boucle
wp_reset_postdata();
?>
```

Remarque : la fonction **posts_nav_link()** laisse peut de possibilités de personnalisation.

Exemple de HTML affiché :

```
<a href="https://{monsite.fr}/blog/">« Page précédente</a>
-
<a href="https://{monsite.fr}/blog/page/2/">Page suivante »</a>
```

Vous pouvez utiliser les fonctions **previous_posts_link()** et **next_posts_link()** pour plus de contrôle et disposer de la possibilité d'intégrer davantage de **style CSS**.

Ces fonctions affichent des liens vers l'article précédent et suivant, respectivement.

Voici comment les utiliser :

```
<?php
// single.php

// ... Votre code HTML et PHP pour l'affichage de l'article ...

// Affichage du lien vers l'article précédent
previous_post_link('<p>Article précédent : %link</p>', 'Article précédent');

// Affichage du lien vers l'article suivant
next_post_link('<p>Article suivant : %link</p>', 'Article suivant');
?>
```

Vous pouvez leur passer deux arguments : un format pour le lien (ex. ici '**<p>Article précédent : %link</p>**') et un texte alternatif pour le lien (ici '**Article précédent**').

Remarque : ces fonctions doivent être placées en dehors de la boucle WordPress.

Ces fonctions utiliseront automatiquement l'ordre chronologique des articles basé sur leur date de publication.

Remarque : si un article précédent ou suivant n'existe pas, les liens ne seront pas affichés.

Depuis la version WordPress 4.1, la fonction **the_posts_pagination()** permet d'afficher une pagination numérotée.

Syntaxe :

```
<?php the_posts_pagination( $args ); ?>
```

Paramètre :

\$args (optionnel) : un tableau d'arguments permettant de personnaliser l'apparence et le comportement de la pagination. Ce paramètre est facultatif, et si vous ne le spécifiez pas, WordPress utilisera les valeurs par défaut.

Exemple d'utilisation :

```
<?php
// archive.php, category.php, search.php ou autre modèle d'archive

if ( have_posts() ) :
    while ( have_posts() ) :
        the_post();
        // afficher le contenu de chaque article ici
    endwhile;

    // afficher la pagination des articles
    the_posts_pagination( array(
        'mid_size' => 2, // nombre de liens avant et après le lien actif
        'prev_text' => 'Précédent',
        'next_text' => 'Suivant',
    ) );

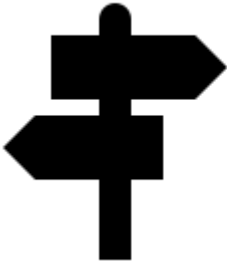
else:
    // message en cas d'absence d'articles
    echo "Aucun article trouvé.";
endif;
?>
```

Quelques arguments couramment utilisés pour personnaliser la pagination :

- **mid_size** : le nombre de liens de pagination affichés avant et après le lien de page active (le numéro de la page actuelle). Par exemple, si **mid_size** est défini à 2, il y aura deux liens avant et deux liens après le lien de la page active.
- **prev_text** : le texte à afficher pour le lien “Page précédente”.
- **next_text** : le texte à afficher pour le lien “Page suivante”.

Remarque : la fonction **the_posts_pagination()** permet en plus des liens “Précédent” / “Suivant” d’afficher les numéros de pages correspondant.

Alternative : pour vous faciliter la tâche, vous pouvez également gérer la pagination grâce à une extension comme **WP-PageNavi**.



WP-PageNavi

Ajoute une interface de pagination plus avancée.

Par [Lester 'GaMerZ' Chan](#)

[Installer maintenant](#)[Plus de détails](#)

★★★★★ (144)

600 000+ installations actives

Dernière mise à jour : il y a 3 mois

✓ Compatible avec votre version de WordPress

Il vous faudra ensuite mettre en place la navigation en modifiant votre template :

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

Navigation entre les articles

Pour gérer la navigation entre les articles avec les fonctions **`previous_post_link()`** et **`next_post_link()`**, vous devez les utiliser après la boucle principale de WordPress, généralement utilisée pour afficher les articles individuels.

Ces fonctions permettent de créer des liens vers les articles précédents et suivants en fonction de l'ordre de publication.

Voici comment les utiliser dans le fichier “single.php” de votre thème WordPress :

```
<?php
// single.php

// vérifier s'il y a des articles dans la boucle
if (have_posts()) {
    // la boucle principale de WordPress
    while (have_posts()) {
        the_post();
        // Afficher le contenu de l'article ici
    }

    // affichage du lien vers l'article précédent
    previous_post_link('<p>Article précédent : %link</p>', 'Article
précédent');

    // affichage du lien vers l'article suivant
    next_post_link('<p>Article suivant : %link</p>', 'Article suivant');
}
?>
```

Remarque : référez-vous à la [documentation officielle WordPress](#) pour connaître toutes les spécificités de ces fonctions.