



La Revue du Web

02 Juin 2022

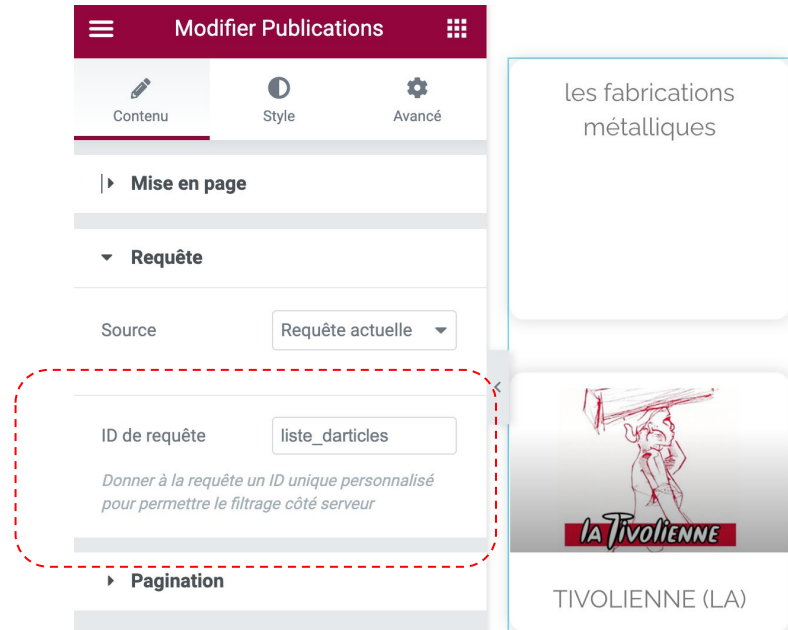




Y voir un peu plus clair avec la loop

- Qu'est ce que c'est
- Comment l'utiliser
- Et avec Elementor ?

Le départ de l'enquête



The screenshot displays a web application interface for managing publications. The main header is 'Modifier Publications'. Below it, there are three tabs: 'Contenu', 'Style', and 'Avancé'. The 'Contenu' tab is selected. The interface is divided into sections: 'Mise en page', 'Requête', and 'Pagination'. The 'Requête' section is expanded, showing a 'Source' dropdown set to 'Requête actuelle' and an 'ID de requête' input field containing 'liste_articles'. A red dashed box highlights the 'ID de requête' field and the explanatory text below it: 'Donner à la requête un ID unique personnalisé pour permettre le filtrage côté serveur'. To the right of the interface, there are two preview cards. The top card shows the text 'Les fabrications métalliques'. The bottom card features a red line-art illustration of a man carrying a long beam on his shoulder, with the text 'LA TIVOLIENNE' in a stylized font, and 'TIVOLIENNE (LA)' below it.

Modifier Publications

Contenu Style Avancé

Mise en page

Requête

Source Requête actuelle

ID de requête liste_articles

Donner à la requête un ID unique personnalisé pour permettre le filtrage côté serveur

Pagination

Les fabrications métalliques

LA TIVOLIENNE

TIVOLIENNE (LA)



Y voir un peu plus clair avec la loop

1. La Loop
2. La Main query
3. WP Query
4. query vars
5. Comment modifier la main query par le code
6. Utilisation dans Elementor



La loop de WordPress

[https://codex.wordpress.org/fr:La Boucle](https://codex.wordpress.org/fr:La_Boucle)

[https://codex.wordpress.org/fr:La Boucle En Action](https://codex.wordpress.org/fr:La_Boucle_En_Action)

- "The Loop" ou "La Boucle" est un terme qui se réfère au processus principal de WordPress.
- La Boucle (The Loop) est utilisée par WordPress pour afficher chacun de vos Articles.
=> catégories, articles, tags, etc

Analyse d'une page simplifiée

```
<?php
get_header();
if (have_posts()) :
    while (have_posts()) :
        the_post();
        the_content();
    endwhile;
endif;
get_sidebar();
get_footer();
?>
```

Analyse d'une page simplifiée

```
<?php
get_header();
if (have_posts()) :
    while (have_posts()) :
        the_post();
        the_content();
    endwhile;
endif;
get_sidebar();
get_footer();
?>
```

D'abord il vérifie si il y a des articles avec la fonction `have_posts()`.

Si des articles existent, une boucle PHP `while` est démarrée. Une instruction `while` s'exécute tant que la condition entre parenthèses est vrai. Donc tant que `have_posts()` retourne une valeur vrai, La Boucle continue.

Analyse d'une page simplifiée

```
<?php
get_header();
if (have_posts()) :
    while (have_posts()) :
        the_post();
        the_content();
    endwhile;
endif;
get_sidebar();
get_footer();
?>
```

La fonction `the_post()` prends l'item courant dans la collection d'articles et rends celui-ci utilisable dans l'itération de La Boucle.

Sans la fonction `the_post()` certains des marqueurs de modèle (template tags) utilisé dans votre thème ne fonctionnerait pas.

Une fois que les données de l'article sont accessibles, le modèle (template) peut commencer l'affichage de ces données au visiteur.

C'est quoi les templates tags (ou marqueur de modèle) ?

Des fonctions pour afficher du contenu

[https://codex.wordpress.org/fr:Marqueurs de Modele](https://codex.wordpress.org/fr:Marqueurs_de_Modele)

Analyse d'une page simplifiée

```
<?php
get_header();
if (have_posts()) :
    while (have_posts()) :
        the_post();
        the_content();
    endwhile;
endif;
get_sidebar();
get_footer();
?>
```

the_post initialise en fait les templates tags propres aux articles et les renseignent dans ce contexte.


Marqueurs des Articles

- the_ID
- the_title
- single_post_title
- the_title_rss
- the_content
- the_content_rss
- the_excerpt
- the_excerpt_rss
- previous_post_link
- previous_post (Obsolète)
- next_post_link
- next_post (Obsolète)
- posts_nav_link
- the_meta

Analyse d'une page simplifiée

```
<?php
get_header();
if (have_posts()) :
    while (have_posts()) :
        the_post();
        the_content();
    endwhile;
endif;
get_sidebar();
get_footer();
?>
```

the_post est initialisé, et donc the_content peut fonctionner et affiche le contenu de chaque article dans la boucle





La loop de WordPress

- Utilisée par défaut pour la page d'accueil, la page catégorie, la page archives, et single post (article)





Le fonctionnement de la loop

1. Vérifie si les fichiers php nécessaires existent
2. Vérifie les paramètres par défaut définis dans WP (nb articles par page, commentaires ou pas ?, etc.)
3. Vérifie ce que le visiteur veut voir : une catégorie ? un article ? une page ?

Ceci lui permet de lancer la bonne requête mysql pour aller extraire les articles

=> La MAIN QUERY : la requête SQL principale utilisée par WP pour afficher du contenu

4. Si le visiteur n'a pas demandé un article, une catégorie, une page, ou une date spécifique, WordPress utilise les paramètres par défaut précédemment collectées pour déterminer quels articles préparer

La Main Query de la loop de WordPress

utilise la classe WP_Query

https://developer.wordpress.org/reference/classes/wp_query/

càd permet de faire simplement des requêtes mysql

Display posts that have this category (not children of that category), using category id:

```
1 | $query = new WP_Query( array( 'category__in' => 4 ) );
```

Display posts that have several categories, using category id:

```
1 | $query = new WP_Query( array( 'cat' => '2,6,17,38' ) );
```

Display posts that have these categories, using category slug:

```
1 | $query = new WP_Query( array( 'category_name' => 'staff,news' ) );
```

Display posts that have "all" of these categories:

```
1 | $query = new WP_Query( array( 'category_name' => 'staff+news' ) );
```

```
1 <?php
2 // 1. On définit les arguments pour définir ce que l'on souhaite récupérer
3 $args = array(
4     'post_type' => 'post',
5     'category_name' => 'films',
6     'posts_per_page' => 3,
7 );
8
9 // 2. On exécute la WP Query
10 $my_query = new WP_Query( $args );
11
12 // 3. On lance la boucle !
13 if( $my_query->have_posts() ) : while( $my_query->have_posts() ) : $my_query->the_post();
14
15     the_title();
16     the_content();
17     the_post_thumbnail();
18
19 endwhile;
20 endif;
21
22 // 4. On réinitialise à la requête principale (important)
23 wp_reset_postdata();
```

<https://capitainewp.io/formations/developper-theme-wordpress/wp-query/>



Comment connaître les query_vars ? et pourquoi ?

https://developer.wordpress.org/reference/classes/wp_query/

- **Tout est dans la doc ! (quasiment)**
- Liste des arguments par type (posts, pages, catégorie, etc) + des tonnes d'exemples

En connaissant les query vars, vous saurez agir sur la main query !



Pourquoi agir sur la main query ?

<https://capitainewp.io/formations/developper-theme-wordpress/modifier-parametres-boucle-wordpress>

- Pleins de cas en fonction de vos besoins
- exemple :
- Toutes les catégories affichent les articles par date inverse, mais pour telle et telle catégorie il faut que ça s'affiche par ordre alphabétique inverse
- Pour la catégorie "Actus" pagination par 5, alors que sur la catégorie portfolio c'est par 12



Comment agir sur la main query

https://developer.wordpress.org/reference/classes/wp_query/

- Par du code php
- généralement dans le function.php de votre thème (enfant)
- On va ajouter ou modifier des paramètres existants

On repart de cet exemple :

Pour la catégorie “Actus” pagination par défaut (réglage wp), alors que sur la catégorie portfolio c’est par 12

Pour la catégorie “Actus” pagination par défaut alors que sur la catégorie portfolio c’est par 12

```
// On définit la fonction qui va agir sur la main query
function capitaine_override_query( $wp_query ) {

    // Comme la fonction sera appelée sur toutes les query, on vérifie qu'on est bien sur la principale
    // On vérifie aussi qu'on est dans la rubrique qu'on souhaite modifier
    if( $wp_query->is_main_query() and ($wp_query->query_vars['category_name']=='portfolio'))
        // On modifie la query var pour cette catégorie
        $wp_query->set( 'posts_per_page', 12 );
    endif;
}

//On se hook sur ce point d'entrée, qui est celui de base pour la main query
add_action( 'pre_get_posts', 'capitaine_override_query' );
```

Certaines catégories en ordre alphabétique

```
//-----  
//  
// LISTE MEMBRES DANS L'ANNUAIRE  
//  
//-----  
// Si on est dans une des catégorie de l'annuaire, on altere la main query  
// pour changer l'ordre par ordre alphabétique croissant  
//  
// Reverse default date descending order for specific categories  
function change_category_order( $query ) {  
    //Sort all posts from category with id=19 (interviews) by date asc order  
    $cat_membres=array(  
        "Auto"=>12,  
        "Agroalimentaire"=>10,  
        "Ameublement"=>13,  
        "Chimie & parachimie"=>14,  
        "Distilleries" =>11,  
        "Énergie" => 15,  
        "Environnement" =>16,  
        "Matériaux de construction" =>18,  
        "Travail des métaux" =>19,  
    );  
    $query_test_id = $query->get_queried_object_id();  
    if($query->is_category() && (in_array($query_test_id, $cat_membres, true)) && $query->is_main_query()){  
        //$query->query_vars['orderby'] = 'name';  
        $query->query_vars['order'] = 'ASC';  
        $query->query_vars['posts_per_page'] = 60;  
    }  
}  
  
add_action( 'pre_get_posts', 'change_category_order' );
```



Comment agir sur la main query dans Elementor

<https://developers.elementor.com/docs/hooks/custom-query-filter/>

1. Donner un id à votre requête
2. code php dans functions.php pour cibler cette requête et agir sur cette query

Comment agir sur la main query dans Elementor

Modifier Publications

Contenu Style Avancé

Mise en page

Requête

Source Requête actuelle

ID de requête

Donner à la requête un ID unique personnel pour permettre le filtrage côté serveur

Pagination

```
* @param \WP_Query $query The WordPress query instance.
*/
function my_query_by_different_order( $query ) {
    $query->set( 'orderby', 'title' );
    $query->set( 'order', 'ASC' );
    $query->set( 'posts_per_page', '60' );
}
add_action( 'elementor/query/liste_darticles', 'my_query_by_different_order' );
```

Pour ne rien rater :



Twitter : @soon716



site : www.soon7.net