Pagination

- Using Pagination to Navigate Post Lists
- Examples
  - Loop with Pagination
  - Methods for displaying pagination links
    - Simple Pagination
    - Numerical Pagination
    - Pagination Between Single Posts
  - Pagination within a post

Pagination allows your user to *page* back and forth through multiple pages of content.

WordPress can use pagination when:

- Viewing lists of posts when more posts exist than can fit on one page, or
- Breaking up longer posts by manually by using the following tag: `<!--nextpage-->`

Using Pagination to Navigate Post Lists

The most common use for pagination in WordPress sites is to break up long lists of posts into separate pages. Whether you're viewing a category, archive, or default index page for a blog or site, WordPress only shows 10 posts per page by default. Users can change the number of posts that appear on each page on the Reading screen: **Admin > Settings > Reading**.

Examples

Loop with Pagination

This simplified example shows where you can add pagination functions for the main loop. Add the functions just before or after the loop.

```
<?php if ( have_posts() ) : ?>

    <!-- Start the pagination functions before the loop. -->
    <div class="nav-previous alignleft"><?php next_posts_link( 'Older posts' ); ?></div>
    <div class="nav-next alignright"><?php previous_posts_link( 'Newer posts' ); ?></div>
    <!-- End the pagination functions before the loop. -->

    <!-- Start of the main loop. -->
    <?php while ( have_posts() ) : the_post();  ?>

    <!-- the rest of your theme's main loop -->

    <?php endwhile; ?>
    <!-- End of the main loop -->

    <!-- Start the pagination functions after the loop. -->
    <div class="nav-previous alignleft"><?php next_posts_link( 'Older posts' ); ?></div>
    <div class="nav-next alignright"><?php previous_posts_link( 'Newer posts' ); ?></div>
    <!-- End the pagination functions after the loop. -->

<?php else : ?>

    <?php _e( 'Sorry, no posts matched your criteria.' ); ?>

<?php endif; ?>
```

Methods for displaying pagination links

When using any of these pagination functions outside the template file with the loop that is being paginated, you must call the global variable $wp_query.

```
function your_themes_pagination() {
    global $wp_query;
    echo paginate_links();
}
```

WordPress has numerous functions for displaying links to other pages in your loop. Some of these functions are only used in very specific contexts. You would use a different function on a single post page then you would on a archive page. The following section covers archive template pagination functions. The section after that cover single post pagination.

Simple Pagination

posts_nav_link

One of the simplest methods is posts_nav_link(). Simply place the function in your template after your loop. This generates both links to the next page of posts and previous page of posts where applicable. This function is ideal for themes that have simple pagination requirements.

```
posts_nav_link();
```

next_posts_link & prev_posts_link

When building a theme, use next_posts_link() and prev_posts_link(). to have control over where the previous and next posts page link appears.

```
next_posts_link();
previous_posts_link();
```

If you need to pass the pagination links to a PHP variable, you can use get_next_posts_link() and get_previous_posts_link().

```
$next_posts = get_next_posts_link();
$prev_posts = get_previous_posts_link();
```

Numerical Pagination

When you have many pages of content it is a better experience to display a list of page numbers so the user can click on any one of the page links rather then having to repeatedly click next or previous posts. WordPress provides several functions for automatically displaying a numerical pagination list.

For WordPress 4.1+

If you want more robust pagination options, you can use the_posts_pagination() for WordPress 4.1 and higher. This will output a set of page numbers with links to previous and next pages of posts.

```
the_posts_pagination();
```

For WordPress prior to 4.1

If you want your pagination to support older versions of WordPress, you must use paginate_links().

```
echo paginate_links();
```

Pagination Between Single Posts

All of the previous functions should be used on index and archive pages. When you are viewing a single blog post, you must use prev_post_link and next_post_link. Place the following functions below the loop on your single.php.

```
previous_post_link();
next_post_link();
```

Pagination within a post

WordPress gives you a tag that can be placed in post content to enable pagination for that post: `<!--nextpage-->`

If you use that tag in the content, you need to ensure that the wp_link_pages function is placed in your single.php template within the loop.

```
<?php if ( have_posts() ) : ?>

    <!-- Start of the main loop. -->
    <?php while ( have_posts() ) : the_post(); ?>

        <?php the_content(); ?>

        <?php wp_link_pages(); ?>

    <?php endwhile; ?>
    <!-- End of the main loop. -->

<?php endif; ?>
```

□□□ #2
□ □□□ □□□ 2026-10-31 18:36:25 UTC
□ □□□ □□□ 2026-10-31 18:36:26 UTC