How To Add Pagination To Your WordPress Theme

If you are looking to add pagination support to your WordPress theme with cool numbers instead of the default next & previous post, you can do so using the famous PageNavi plugin, however, I much prefer adding pagination manually to my themes so people don't have to go searching for a plugin. It also helps keep the site faster without all the external scripts and CSS.

Luckily there is a great function in wordpress called "paginate_links" which was added in WordPress 2.1 and will allow you to create a paginated style navigation for any query on your site. Here is a quick tutorial for adding a simple page navigation to your theme that looks just like the pagination in my "Total WordPressTheme".

Pagination PHP

Simply add the following code at the end of your functions.php file (or whatever file in your theme where you want to keep it).

```
// Numbered Pagination
if ( !function_exists( 'wpex_pagination' ) ) {

    function wpex_pagination() {

        $prev_arrow = is_rtl() ? '→' : '←';
        $next_arrow = is_rtl() ? '←' : '→';

        global $wp_query;
        $total = $wp_query->max_num_pages;
        $big = 999999999; // need an unlikely integer
        if( $total > 1 )  {
             if( !$current_page = get_query_var('paged') )
                 $current_page = 1;
             if( get_option('permalink_structure') ) {
                 $format = 'page/%#%/';
             } else {
                 $format = '&paged=%#%';
             }
            echo paginate_links(array(
                'base'         => str_replace( $big, '%#%', esc_url( get_pagenum_link( $big ) ) ),
                'format'       => $format,
                'current'      => max( 1, get_query_var('paged') ),
                'total'        => $total,
                'mid_size'     => 3,
                'type'             => 'list',
                'prev_text'        => $prev_arrow,
                'next_text'        => $next_arrow,
             ) );
        }
    }

}
```

VIEW ALL PARAMETERS

Pagination CSS

Copy the following CSS and paste into your style.css file.

```
ul.page-numbers {
    list-style: none;
    margin: 0;
}

.page-numbers:after {
    content: ".";
    display: block;
    clear: both;
    visibility: hidden;
    line-height: 0;
    height: 0;
}
```

```css
ul.page-numbers li {
    display: block;
    float: left;
    margin: 0 4px 4px 0;
    text-align: center;
}

.page-numbers a,
.page-numbers span {
    line-height: 1.6em;
    display: block;
    padding: 0 6px;
    height: 18px;
    line-height: 18px;
    font-size: 12px;
    text-decoration: none;
    font-weight: 400;
    cursor: pointer;
    border: 1px solid #ddd;
    color: #888;
}

.page-numbers a span { padding: 0 }

.page-numbers a:hover,
.page-numbers.current,
.page-numbers.current:hover {
    color: #000;
    background: #f7f7f7;
    text-decoration: none;
}

.page-numbers:hover { text-decoration: none }
```

Adding The Pagination Function To Your Theme

To call back the pagination function it's really simple. All you have to do is add the following code to your theme files where you want to show any sort of pagination. The most common are your index.php, home.php, category.php, tags.php, archive.php and search.php. But if you have any custom page templates with pagination support, you'll want to add them here.

Replace default pagination with the following (normally located somewhere after endif):

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

Custom Queries?

If you are creating a custom query using WP_Query this function won't work unless you've defined your query in the $wp_query variable (which is bad, don't do it). To fix it, I generally create new queries like the following:

```php
$wpex_query = new WP_Query( $args );
```

This way I can alter the main pagination function to look for the variable when creating the pagination, example (editing the first snippet):

```php
global $wp_query, $wpex_query;
if ( $wpex_query ) {
    $total = $wpex_query->max_num_pages;
} else {
    $total = $wp_query->max_num_pages;
}
```

Update: In this example I check for the global variable…However you could simply pass the query variable directly to the wpex_pagination function which is probably a better choice ☺