/\*\*

\* WordPress Query Comprehensive Reference

\* Compiled by luetkemj - luetkemj.com

\*

\* CODEX: http://codex.wordpress.org/Class\_Reference/WP\_Query

\* Source: http://core.trac.wordpress.org/browser/tags/3.3.1/wp-includes/query.php

\*/

$args = array(

**//////Author Parameters - Show posts associated with certain author.**

'author' => 1,2,3, //(int) - use author id [use minus (-) to exclude authors by ID ex. 'author' => -1,-2,-3,]

'author\_name' => 'luetkemj', //(string) - use 'user\_nicename' (NOT name)

**//////Category Parameters - Show posts associated with certain categories.**

'cat' => 5,//(int) - use category id.

'category\_name' => 'staff', 'news', //(string) - use category slug (NOT name).

'category\_\_and' => array( 2, 6 ), //(array) - use category id.

'category\_\_in' => array( 2, 6 ), //(array) - use category id.

'category\_\_not\_in' => array( 2, 6 ), //(array) - use category id.

**//////Tag Parameters - Show posts associated with certain tags.**

'tag' => 'cooking', //(string) - use tag slug.

'tag\_id' => 5, //(int) - use tag id.

'tag\_\_and' => array( 2, 6), //(array) - use tag ids.

'tag\_\_in' => array( 2, 6), //(array) - use tag ids.

'tag\_\_not\_in' => array( 2, 6), //(array) - use tag ids.

'tag\_slug\_\_and' => array( 'red', 'blue'), //(array) - use tag slugs.

'tag\_slug\_\_in' => array( 'red', 'blue'), //(array) - use tag slugs.

**//////Taxonomy Parameters - Show posts associated with certain taxonomy.**

**//Important Note: tax\_query takes an array of tax query arguments arrays (it takes an array of arrays)**

**//This construct allows you to query multiple taxonomies by using the relation parameter in the first (outer) array to describe the boolean relationship between the taxonomy queries.**

'tax\_query' => array( //(array) - use taxonomy parameters (available with Version 3.1).

'relation' => 'AND', //(string) - Possible values are 'AND' or 'OR' and is the equivalent of ruuning a JOIN for each taxonomy

array(

'taxonomy' => 'color', //(string) - Taxonomy.

'field' => 'slug', //(string) - Select taxonomy term by ('id' or 'slug')

'terms' => array( 'red', 'blue' ), //(int/string/array) - Taxonomy term(s).

'include\_children' => true, //(bool) - Whether or not to include children for hierarchical taxonomies. Defaults to true.

'operator' => 'IN' //(string) - Operator to test. Possible values are 'IN', 'NOT IN', 'AND'.

),

array(

'taxonomy' => 'actor',

'field' => 'id',

'terms' => array( 103, 115, 206 ),

'include\_children' => false,

'operator' => 'NOT IN'

)

),

**//////Post & Page Parameters - Display content based on post and page parameters.**

'p' => 1, //(int) - use post id.

'name' => 'hello-world', //(string) - use post slug.

'page\_id' => 1, //(int) - use page id.

'pagename' => 'sample-page', //(string) - use page slug.

'pagename' => 'contact\_us/canada', //(string) - Display child page using the

slug of the parent and the child page,

separated ba slash

'post\_parent' => 1, //(int) - use page id. Return just the child

Pages.

'post\_\_in' => array(1,2,3), //(array) - use post ids. Specify posts to

retrieve.

'post\_\_not\_in' => array(1,2,3), //(array) - use post ids. Specify post NOT to

retrieve.

**//NOTE: you cannot combine 'post\_\_in' and 'post\_\_not\_in' in the same query**

**//////Type & Status Parameters - Show posts associated with certain type or status.**

'post\_type' => array( //(string / array) - use post types.

Retrieves posts by Post Types, default value

is 'post';

'post', // - a post.

'page', // - a page.

'revision', // - a revision.

'attachment', // - an attachment. The default WP\_Query sets

'post\_status'=>'published', but atchments default to 'post\_status'=>'inherit' so you'll need to set the status to

'inherit' or 'any'.

'my-post-type', // - Custom Post Types (e.g. movies)

),

'post\_status' => array( //(string / array) - use post status.

Retrieves posts by Post Status, default

value i'publish'.

'publish', // - a published post or page.

'pending', // - post is pending review.

'draft', // - a post in draft status.

'auto-draft', // - a newly created post, with no content.

'future', // - a post to publish in the future.

'private', // - not visible to users who are not logged

in.

'inherit', // - a revision. see get\_children.

'trash' // - post is in trashbin (available with

Version 2.9).

),

**//NOTE: The 'any' keyword available to both post\_type and post\_status queries cannot be used within an array.**

'post\_type' => 'any', // - retrieves any type except revisions and types with 'exclude\_from\_search' set to true.

'post\_status' => 'any', // - retrieves any status except those from post types with 'exclude\_from\_search' set to true.

**//////Pagination Parameters**

'posts\_per\_page' => 10, //(int) - number of post to show per page

(available with Version 2.1). Use

'posts\_per\_page'=-1 to show all posts. Note

if the query is in a feed, wordpress

overwrites this parameter with the stored 'posts\_per\_rss' option. Treimpose the limit, try using the 'post\_limits' filter, or

filter 'pre\_option\_posts\_per\_rss' and return

-1

'posts\_per\_archive\_page' => 10, //(int) - number of posts to show per page –

on archive pages only. Over-rides showposts

anposts\_per\_page on pages where is\_archive()

or is\_search() would be true

'nopaging' => false, //(bool) - show all posts or use pagination.

Default value is 'false', use paging.

'paged' => get\_query\_var('page'), //(int) - number of page. Show the posts that

would normally show up just on page X when

usinthe "Older Entries" link.

**//NOTE: You should set get\_query\_var( 'page' ); if you want your query to work with pagination. Since Wordpress 3.0.2, you dget\_query\_var( 'page' ) instead of get\_query\_var( 'paged' ). The pagination parameter 'paged' for WP\_Query() remains the same.**

**//////Offset Parameter**

'offset' => 3, //(int) - number of post to displace or pass

over.

**//////Order & Orderby Parameters - Sort retrieved posts.**

'order' => 'DESC', //(string) - Designates the ascending or

descending order of the 'orderby' parameter.

Defaultto 'DESC'.

//Possible Values:

//'ASC' - ascending order from lowest to

highest values (1, 2, 3; a, b, c).

//'DESC' - descending order from highest to

lowest values (3, 2, 1; c, b, a).

'orderby' => 'date', //(string) - Sort retrieved posts by

parameter. Defaults to 'date'.

//Possible Values://

//'none' - No order (available with Version

2.8).

//'ID' - Order by post id. Note the

captialization.

//'author' - Order by author.

//'title' - Order by title.

//'date' - Order by date.

//'modified' - Order by last modified date.

//'parent' - Order by post/page parent id.

//'rand' - Random order.

//'comment\_count' - Order by number of

comments (available with Version

2.9).

//'menu\_order' - Order by Page Order. Used most often for Pages (Order field in the EdiPage Attributes box) and for Attachments (the integer fields in the Insert / Upload MediGallery dialog), but could be used for any post type with distinct 'menu\_order' values (theall default to 0).

//'meta\_value' - Note that a 'meta\_key=keyname' must also be present in the query. Note alsthat the sorting will be alphabetical which is fine for strings (i.e. words), but can bunexpected for numbers (e.g. 1, 3, 34, 4, 56, 6, etc, rather than 1, 3, 4, 6, 34, 56 as yomight naturally expect).

//'meta\_value\_num' - Order by numeric meta value (available with Version 2.8). Also notthat a 'meta\_key=keyname' must also be present in the query. This value allows for numericasorting as noted above in 'meta\_value'.

//'post\_\_in' - Preserve post ID order given in the post\_\_in array (available with Version 3.5).

**//////Sticky Post Parameters - Show Sticky Posts or ignore them.**

'ignore\_sticky\_posts' => false, //(bool) - ignore sticky posts or not.

Default value is false, don't ignore.

Ignore/excludsticky posts being included at

the beginning of posts returned, but the

sticky post will still be returned in the

natural order othat list of posts returned.

//NOTE: For more info on sticky post queries see: http://codex.wordpress.org/Class\_Reference/WP\_Query#Sticky\_Post\_Parameters

**//////Time Parameters - Show posts associated with a certain time period.**

'year' => 2012, //(int) - 4 digit year (e.g. 2011).

'monthnum' => 3, //(int) - Month number (from 1 to 12).

'w' => 25, //(int) - Week of the year (from 0 to 53).

Uses the MySQL WEEK command. The mode is

dependenon the "start\_of\_week" option.

'day' => 17, //(int) - Day of the month (from 1 to 31).

'hour' => 13, //(int) - Hour (from 0 to 23).

'minute' => 19, //(int) - Minute (from 0 to 60).

'second' => 30, //(int) - Second (0 to 60).

**//////Custom Field Parameters - Show posts associated with a certain custom field.**

'meta\_key' => 'key', //(string) - Custom field key.

'meta\_value' => 'value', //(string) - Custom field value.

'meta\_value\_num' => 10, //(number) - Custom field value.

'meta\_compare' => '=', //(string) - Operator to test the

'meta\_value'. Possible values are '!=', '>',

'>=', '<', or ='. Default value is '='.

'meta\_query' => array( //(array) - Custom field parameters

(available with Version 3.1).

array(

'key' => 'color', //(string) - Custom field key.

'value' => 'blue', //(string/array) - Custom field value (Note:

Array support is limited to a compare value

of 'IN', 'NOT IN', 'BETWEEN', or 'NOT

BETWEEN')

'type' => 'CHAR', //(string) - Custom field type. Possible

values are 'NUMERIC', 'BINARY', 'CHAR',

'DATE', 'DATETIME', 'DECIMAL', 'SIGNED',

'TIME', 'UNSIGNED'. Default value is 'CHAR'.

'compare' => '=', //(string) - Operator to test. Possible

values are '=', '!=', '>', '>=', '<', '<=',

'LIKE', 'NOT LIKE', 'IN', 'NOT IN',

'BETWEEN', 'NOT BETWEEN'. Default value is

'='.

),

array(

'key' => 'price',

'value' => array( 1,200 ),

'compare' => 'NOT LIKE'

)

**//////Permission Parameters - Display published posts, as well as private posts, if the user has the appropriate capability:**

'perm' => 'readable' //(string) Possible values are 'readable',

'editable' (possible more ie al

capabilitiealthough I have not tested)

**//////Parameters relating to caching**

'no\_found\_rows' => false, //(bool) Default is false. WordPress uses

SQL\_CALC\_FOUND\_ROWS in most queries in

order timplement pagination. Even when you

don’t need pagination at all. By Setting

this parameter to true you are telling

wordPress not tcount the total rows and

reducing load on the DB. Pagination will NOT

WORK when this parameter is set to true. For

more informatiosee:

<http://flavio.tordini.org/speed-up->

wordpress-get\_posts-and-query\_posts-

functions

'cache\_results' => true, //(bool) Default is true

'update\_post\_term\_cache' => true, //(bool) Default is true

'update\_post\_meta\_cache' => true, //(bool) Default is true

**//NOTE Caching is a good thing. Setting these to false is generally not advised. For more info on usage see: http://codex.wordpresorg/Class\_Reference/WP\_Query#Permission\_Parameters**

**//////Search Parameter**

's' => $s, //(string) - Passes along the query string

variable from a search. For example usage

see: <http://www.wprecipes.com/how-to->

display-the-number-of-results-in-wordpress-

search

'exact' => true //(bool) - flag to make it only match whole

titles/posts - Default value is false. For

more information see:

<https://gist.github.com/2023628#gistcomment->

285118

'sentence' => true //(bool) - flag to make it do a phrase search

- Default value is false. For more

information see:

<https://gist.github.com/2023628#gistcomment->

285118

**//////Post Field Parameters**

//Not sure what these do. For more info see: http://codex.wordpress.org/Class\_Reference/WP\_Query#Post\_Field\_Parameters

//////Filters

//For more information on available Filters see: http://codex.wordpress.org/Class\_Reference/WP\_Query#Filters

);

$the\_query = new WP\_Query( $args );

// The Loop

if ( $the\_query->have\_posts() ) :

while ( $the\_query->have\_posts() ) : $the\_query->the\_post();

// Do Stuff

endwhile;

endif;

// Reset Post Data

wp\_reset\_postdata();