# Самостоятельное получение записей из БД от Дмитрия Полоса

#### get\_post()

<https://wp-kama.ru/function/get_post> - документация для get\_post()

Пользуясь **get\_post()**, можно вывести на страницу, н-р, данные с другой страницы. Но нужно иметь ввиду, что при выводе контента другой страницы этот контент, возможно, не будет,н-р, обрабатываться плагинами, н-р, будет не доступен для перевода на другой язык.

$post = get\_post( 2 ); // 2- id главной страницы

echo '<p>'. $post->post\_content; '</p>';

Тогда нужно использовать:

<?php echo apply\_filters( 'the\_content' , $post->post\_content ); ?>

Посмотреть весь массив:

$post = get\_post( 2 );

echo '<p>'. var\_dump( $post ). '</p>';

#### get\_posts()

<https://wp-kama.ru/function/get_posts> - документация для get\_posts()

$results = get\_posts([

'include' => [61],

'post\_type'=> 'capabilities'

]);

var\_dump($results);

Пример с сортировкой:

$results = get\_posts([

'post\_type'=> 'capabilities',

'numberposts' => 3,

'orderby' => 'title',

'order' => 'desc'

]);

Если используются **таксономии**, использовать **meta\_key** и **meta\_value.** Можно сделать по ним сортировку.

Для вывода в верстке использовать цикл:

**foreach**( $results as $post ){ ?>

<h2>

<?php echo $post->post\_title; ?>

</h2>

<p>

<?php echo apply\_filters( 'the\_content' , $post->post\_content ); ?>

</p>

<?php }

Использование в верстке:

<?php

global $post;

$results = get\_posts([

'post\_type'=> 'products',

'numberposts' => -1,

]);

foreach( $results as $post ){

setup\_postdata( $post ); ?>

...

<?php

}

wp\_reset\_postdata();

?>

#### Функции шаблона вне стандартного цикла

Функции шаблона можно использовать и вне стандартного цикла. Тогда делаем так:

foreach( $results as $post ){

setup\_postdata( $post ); ?>

<h2> <?php **the\_title();** ?> </h2>

<p> <?php **the\_content();** ?> </p>

<p> <?php **the\_field('subtitle');** ?> </p>

<?php }

wp\_reset\_postdata(); // Не забыть!

#### WP\_Query{}

Эти функции работают на основе WP\_Query{}

<https://wp-kama.ru/function/wp_query> - документация

Пример использования:

$query = new WP\_Query( [

'posts\_per\_page' => 2,

'post\_type' => 'capabilities',

'meta\_key' => 'capabilities-title',

'orderby' => 'meta\_value',

'order' => 'DESC',

// если нужно получить таксономию

'тип таксономии' => 'ярлык таксономии'

// или более сложный запрос

'tax\_query' => [

'relation' => 'AND',

[

'taxonomy' => 'movie\_janner',

'field' => 'slug',

'terms' => array( 'action', 'comedy' ),

],

[

'taxonomy' => 'actor',

'field' => 'id',

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

'operator' => 'NOT IN',

]

]

// сложный запрос для мета полей

'meta\_query' => [

'relation' => 'OR',

[

'key' => 'color',

'value' => 'blue'

],

[

'key' => 'price',

'value' => 20,

'compare' => '<='

]

]

Для таксономий id можно посмотреть, если навести мышкой.

Заметим, что **сортировка** делается по строкам. Если нам нужна сортировка по числу, использовать **meta\_value\_num**

Вывод полей в **верстке**

if ( $query->have\_posts() ) {

while ( $query->have\_posts() ) {

$query->the\_post();

?>

<h2> <?php the\_title(); ?> </h2>

<p> <?php the\_content(); ?> </p>

<p> <?php the\_field('capabilities-title'); ?> </p>

<p> <?php the\_field('capabilities-content'); ?> </p>

<?php

}

} else {

// Постов не найдено

}

#### query\_posts()

<https://wp-kama.ru/function/query_posts> - документация

Используется самим WordPress при получении записей для текущей страницы (архив авторов, отдельная запись, рубрика, метка). Определяет какие посты будут показаны в базовом Цикле WordPress. Создает базовый Цикл WordPress. Возвращает список записей (постов).

query\_posts() предназначена для самого WordPress и должна использоваться только для изменения основного запроса WordPress на базе которого потом строится Цикл WordPress.

Поэтому, когда нужно создать еще один запрос (цикл), нужно использовать get\_posts() или WP\_Query. Эти функции могут принимать все те же параметры что и query\_posts().

Сейчас, фактически, нет необходимости использовать query\_posts(), все есть в WP\_Query, но если использовать, то нужно быть осторожными.

Проще всего использовать get\_posts()