# Поля записей от Дмитрия Полоса

#### Создание кастомного мета поля

В Wordpress к любому типу записей можно добавить свою информацию и сохранять ее в специальном хранилище post meta по принципу ключ - значение. Для этого есть функции add\_post\_meta и update\_post\_meta.

add\_post\_meta( $post\_id, $meta\_key, $meta\_value, $unique );

*$post\_id(число*) (обязательный) - ID поста, произвольное поле к которому нужно добавить.

*$meta\_key(строка)* (обязательный) - Название (ключ) произвольного поля.

*$meta\_value(строка)* (обязательный) - Значение произвольного поля.

*$unique(логический) -* true - уникальный (только один), false - не уникальный - полей с одним ключом может быть много. Если поставить параметр в true, то при добавлении поля, сначала будет проверяться существует ли уже поле с таким же ключом, если существует, то поле не будет добавлено. По умолчанию: false.

В **functions.php:**

add\_filter( 'add\_meta\_boxes', 'robot\_lending\_meta\_boxes');

function robot\_lending\_meta\_boxes() {

add\_meta\_box(

'rl-digit',

'Цифра 1: ',

'rl\_meta\_digit\_cb',

'page',

'normal'

);

function rl\_meta\_digit\_cb( $post\_obj ){

add\_post\_meta( $post\_obj->ID, 'rl-digit', true);

echo "<input type=\"text\" name=\"rl-digit\" value=\"555\">";

}

}

В **верстке**

<?php echo get\_post\_meta($id, 'rl-digit', true); ?>

Важно: $id можем использовать, если определена переменная $post (организован цикл).

Чтобы, значение поля, измененное в админке, сохранялось, в **functions.php:**

add\_filter( 'save\_post', 'robot\_lending\_save\_meta');

function robot\_lending\_save\_meta( $post\_id ) {

if( isset( **$\_POST['rl-digit']** ) ) {

update\_post\_meta( $post\_id, 'rl-digit', $\_POST['rl-digit'] );

}

}

#### Создание кастомных полей с помощью плагина ACF

Установим плагин.

<https://acfwp.ru/> - документация

Пробуем создать поле с типом **Текст**. **Значение по умолчанию** - не пропадает, когда пользователь начинает правки, **Текст заглушки** = placeholder, пропадает при вводе текста пользователем.

Иногда при создании полей заголовки лучше сделать невидимыми для пользователя, чтобы они были уникальными.

При задании формата для поля типа **Время**, am - 12-часовой формат.

**Чтобы выбрать поле из списка**, в котором находятся наши пользовательские типы записей (н-р, тренеры, тип поля выбираем **Отношение - объект записи**

Для **таксономий** предварительно заполнить данные в полях записей, с которыми они связаны.

Для **рубрик** введем кастомное изображение. Создадим поле с типом Изображение. Условия отображения - **Таксономия - Рубрика**

Если хотим задать поле для **цвета**, тип поля выбрать **JQUERY - цвет**.

Если поле выбирается для **таксономии**, то Условия отображения - **Таксономия - название таксономии**.

Несмотря на то, что поля созданы плагином, их также можно получить через get\_post\_meta

Но мы будем использовать функции плагина, н-р, get\_field() и the\_field();. Потому, что ACF формирует значения в зависимости от типов полей и делает разработку проще!

В **верстке** вставляем для постов стандартный цикл Loop, внутри поста используем **the\_title();** **the\_field();** и др. Для картинок, если использовался Массив изображений, для вывода, н-р, ссылки использовать

echo get\_field('ярлык')['url'];

Сам массив можно посмотреть так:

var\_dump(get\_field('ярлык'));

Для **категорий** нет стандартного цикла вывода постов. Тогда нужен 2-ой аргумент для get\_field() и the\_field();.

Как быть с категориями? См. документацию <https://acfwp.ru/get_field/>

Для категорий

**$post\_id = "category\_3";** // рубрика с ID = 3

Тогда в верстке для категорий вывод поля:

echo get\_field('ярлык', 'category\_' . $cat->cat\_ID);

где $cat - элемент икла при выводе категорий

Для **таксономий**:

**$post\_id = "event\_4**"; // event (произвольная таксономия) с ID = 4