カスタムフィールドを自作する方法【プラグインなし】

ワードプレス開発 2023.01.21

目次

カスタムフィールドとは

カスタムフィールドとは、「タイトル」や「本文」以外に、任意で自作フォームを作成できる機能のことをいいます。

カスタムフィールドを投稿タイプに自作する方法

カスタム投稿タイプについての記事は、下記記事で解説していますので、気になる方はご覧ください。当記事では、カスタムフィールドについてのみ解説します。

✔ あわせて読みたい

カスタム投稿を自作する方法【プラグインなし】

下記コードを「functions.php」に、まるっとコピペして実装してみましょう。

functions.php

```php
//////////////////まるっとコピペOK//////////////////
<?php
add_action('init', 'create_post_type');
function create_post_type()
{
    //投稿時に使用できる投稿用のパーツを指定
    $supports = array(
        'title', //タイトルフォーム
        'editor', //エディター(内容の編集)
        'thumbnail', //アイキャッチ画像
        'author', //投稿者
        'excerpt', //抜粋
        'revisions', //リビジョンを保存
    );
    register_post_type(
        'sample', // 投稿タイプ名の定義
        [
        'labels' => [
            'name' => 'サンプル投稿', // 管理画面上で表示する投稿タイプ名
        ],
        'public'        => true,  // カスタム投稿タイプの表示(trueにする)
        'has_archive'   => true, // カスタム投稿一覧(true:表示/false:非表示)
        'menu_position' => 5,     // 管理画面上での表示位置
        'show_in_rest'  => false,  // true:「Gutenberg」/ false:「ClassicEditor」
        'supports' => $supports
        ]
    );
}


add_action('admin_menu', 'create_custom_fields');
function create_custom_fields()
{
    add_meta_box(
        'sample_setting', //編集画面セクションID
        'サンプルカスタムフィールド', //編集画面セクションのタイトル
        'insert_custom_fields', //編集画面セクションにHTML出力する関数
        'sample', //投稿タイプ名
        'normal' //編集画面セクションが表示される部分
    );
}

function insert_custom_fields()
{
    global $post;
    $sample = get_post_meta($post->ID, 'sample', true); ?>

<form method="post" action="admin.php?page=site_settings">
    <label for="sample">サンプルフォーム：</label>
    <input id="sample" type="text" name="sample" value="<?php echo $sample ?>">
</form>

<?php
}
add_action('save_post', 'save_custom_fields');
function save_custom_fields($post_id)
{
    if (isset($_POST['sample'])) {
        update_post_meta($post_id, 'sample', $_POST['sample']);
    }
}
```

下記画像のようなフォームが、サンプル投稿の編集画面で表示されます。
入力して、公開ボタンを押し登録される事を確認しましょう。

カスタムフィールドを自作する編集画面セクションを作成

add_action(‘admin_menu’, ‘呼び出す関数’)で、カスタムフィールドを表示する編集画面セクションを作成することができます。

呼び出す関数に、add_meta_boxで項目の設定をします。

functions.php

```php
<?php
add_action('admin_menu', 'create_custom_fields');
function create_custom_fields()
{
    add_meta_box(
        'sample_setting', //編集画面セクションID
        'サンプルカスタムフィールド', //編集画面セクションのタイトル
        'insert_custom_fields', //編集画面セクションにHTML出力する関数
        'sample', //投稿タイプ名
        'normal' //編集画面セクションが表示される部分
    );
}
```

これだけの記述だけだと、編集画面セクションの作成はできますが、「insert_custom_fields」という関数を読み込めていないので、エラーが表示されます。

編集画面セクションに出力するフォームを作成

それでは、「insert_custom_fields」というコールバック関数について解説します。

functions.php

```php
<?php
function insert_custom_fields()
{
    global $post;
    $sample = get_post_meta($post->ID, 'sample', true); ?>

<form method="post" action="admin.php?page=site_settings">
    <label for="sample">サンプルフォーム：</label>
    <input id="sample" type="text" name="sample" value="<?php echo $sample ?>">
</form>

<?php
}
add_action('save_post', 'save_custom_fields');
function save_custom_fields($post_id)
{
    if (isset($_POST['sample'])) {
        update_post_meta($post_id, 'sample', $_POST['sample']);
    }
}
```

登録用の変数を作成します。今回は、$sampleという変数を記述しました。

その後、フォームタグで、作成したい項目を記述します。

functions.php

```php
<?php
global $post;
$sample = get_post_meta($post->ID, 'sample', true); ?>

<form method="post" action="admin.php?page=site_settings">
    <label for="sample">サンプルフォーム：</label>
    <input id="sample" type="text" name="sample" value="<?php echo $sample ?>">
</form>
?>
}
```

この状態だけでは、フォームは作成されますが、入力したデータが保存されません。そのため、保存するためのコードを記述します。

add_action(‘save_post’,’更新処理をする関数’)を記述することで、データの保存をすることができます。

functions.php

```php
add_action('save_post', 'save_custom_fields');
function save_custom_fields($post_id)
{
    if (isset($_POST['sample'])) {
        update_post_meta($post_id, 'sample', $_POST['sample']);
    }
}
```

これで、テキトフォームのカスタムフィールの作成が完成です。

編集画面セクションには、テキストフォーム以外にも、テキストエリアタグ、セレクトタグなども配置することも可能です。

様々なフォームの作成方法については、下記記事で解説してますので、ご覧ください。

✔ あわせて読みたい

カスタムフィールドで様々なフォームを自作する方法【WordPress】

カスタムフィールドに登録した項目をページに表示する

まずは、カスタム投稿タイプを表示するページを作成します。

デフォルトである「投稿」のページは、single.phpで表示されます。

カスタム投稿タイプのページは、single-{‘カスタム投稿タイプ名’}.phpで表示されます。今回の場合は、single-sample.phpというファイルを作成し、記述します。

single-sample.php

```php
<?php echo get_post_meta($post->ID, "sample", true); ?>
```

get_post_meta(‘投稿ID’,’カスタムフィールドの変数名’,true)で、カスタムフィールドから保存しているデータを取得し、echoで表示させることができます。

(adsbygoogle = window.adsbygoogle || []).push({});