

## 21 オリジナルテーマの作成 5

## 14. カスタム投稿タイプ

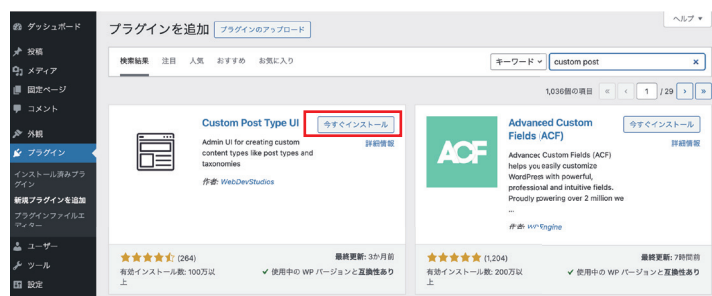
## 14-1. カスタム投稿タイプとは

WordPress では、コンテンツに関する情報を「投稿タイプ」という型に分けて管理しています。標準の投稿タイプには「投稿 (post)」「固定ページ (page)」「添付ファイル (attachment)」「リビジョン (revision)」「ナビゲーションメニュー (nav\_menu\_item)」などがあり、コンテンツの作成では主に投稿と固定ページが使われます。

事業用のウェブサイトなどで、デフォルトの「投稿 (post)」(今回のサイトではお知らせ) 以外に、例えば「製品紹介」や「お客様の声」など、投稿とは別の区分で記事を管理したい場合もあります。このような場合に「カスタム投稿タイプ」として新しい投稿タイプを追加し、記事を管理できます。今回は「Custom Post Type UI」というプラグインを使って、新作スイーツを「カスタム投稿タイプ」として管理できるように設定していきます。

## 14-2. Custom Post Type U プラグインのインストール

[ プラグイン ] > [ 新規プラグインを追加 ] から、「Custom Post Type UI」を検索し、プラグインをインストールして有効化します。



### 14-3. カスタム投稿タイプの追加

[CPT UI]>[ 投稿タイプの追加と編集 ] から、カスタム投稿作成に必要な設定をしていきます。「投稿タイプスラッグ」は、ページ名と同じスラッグを指定します。(今回は「sweets」)



※管理画面のメニューの表示位置を指定する場合



#### 14-4. カスタム投稿用のテンプレートファイルを作成する

デフォルトの投稿同様に、投稿詳細用の single.php と archive.php を作成します。

カスタム投稿用のテンプレートファイルの場合はそれぞれに「- 投稿タイプのスラッグ名」をつけたファイル名を指定します。  
今回の場合、single-sweets.php、archive-sweets.php となります。

#### 14-5. カスタム投稿記事詳細ページ single-sweets.php

single.php を複製し、single-sweets.php にリネームします。

静的コーディングデータ (/sweets/single.html) の内容に置き換え、基本的なテンプレートタグを置き換えていきます。

single-sweets.php

```
<?php get_header(); ?>
<main class="main">

<div class="pkz">
  <ul class="pkz__list">
    <li class="pkz__item">
      <a href="<?php echo esc_url( home_url()); ?>"/>Home</a>
    </li>
    <li class="pkz__item">
      <a href="<?php echo esc_url( home_url()); ?>/sweets/">新作スイーツ </a>
    </li>
    <li class="pkz__item"><?php the_title(); ?></li>
  </ul>
</div>

<section class="section__page sweets__section01">
  <div class="content content--small">
    <div class="singleCont__head">
      <h2 class="singleCont__ttl"> アイテム 1</h2>
    </div>
    <div class="sweetsSingle__img">
      
    </div>
    <div class="singleCont__cont">
      <p> テキストが入ります。テキストが入ります。～省略～テキストが入ります。 </p>
    </div>
    <div class="sweetsSingle__note">
      <p> アレルギー物質：小麦粉 , 卵 </p>
    </div>
  </div>
</section>

</main>
<?php get_footer(); ?>
```

single-sweets.php

～省略～

&lt;section class="section\_\_page sweets\_\_section01"&gt;

&lt;div class="content content--small"&gt;

&lt;?php if ( have\_posts() ) : ?&gt;

&lt;?php

while ( have\_posts() ) :

the\_post();

?&gt;

&lt;div class="singleCont\_\_head"&gt;

&lt;h2 class="singleCont\_\_ttl"&gt;&lt;?php the\_title(); ?&gt;&lt;/h2&gt;

&lt;/div&gt;

&lt;div class="sweetsSingle\_\_img"&gt;

&lt;?php if ( has\_post\_thumbnail() ) : ?&gt;

&lt;?php the\_post\_thumbnail(); ?&gt;

&lt;?php else : ?&gt;

&lt;img src="&lt;?php echo esc\_url( get\_template\_directory\_uri() ); ?&gt;/assets/img/sweets/thumb\_sweets.jpg" alt="Sweets"&gt;

&lt;?php endif; ?&gt;

&lt;/div&gt;

&lt;div class="singleCont\_\_cont"&gt;

&lt;?php the\_content(); ?&gt;

&lt;/div&gt;

&lt;div class="sweetsSingle\_\_note"&gt;

&lt;p&gt; アレルギー物質：小麦粉 , 卵 &lt;/p&gt;

&lt;/div&gt;

&lt;?php endwhile; ?&gt;

&lt;?php endif; ?&gt;

&lt;/div&gt;

&lt;/section&gt;

～省略～

投稿のタイトルを表示

投稿があるかどうかを判定し、投稿がある場合表示

サムネイルがある場合は表示  
ない場合は固定画像を表示

投稿の本文を表示

#### 14-6. カスタム投稿記事一覧ページ archive-sweets.php

archive.php を複製し、archive-sweets.php にリネームします。

静的コーディングデータ (/sweets/index.html) の内容に置き換え、基本的なテンプレートタグを置き換えていきます。

archive-sweets.php

```
<?php get_header(); ?>
<main class="main">

    <div class="pkz">
        <ul class="pkz__list">
            <li class="pkz__item">
                <a href="<?php echo esc_url( home_url()); ?>/">Home</a>
            </li>
            <li class="pkz__item">新作スイーツ </li>
        </ul>
    </div>

    <section class="section__page sweets__section01">
        <div class="content content--large">
            <div class="ttl__base">
                <p class="ttl__baseSub">スイーツ </p>
                <h2 class="ttl__base--jp">新作スイーツ </h2>
            </div>
            <div class="sweetsList">
                <a href=" ../sweets/single.html" class="sweetsList__item">
                    <div class="sweetsList__img">
                        
                    </div>
                    <h4 class="sweetsList__ttl">アイテム 01</h4>
                    <div class="sweetsList__txt">
                        <p>テキストが入ります。テキストが入ります。テキストが入ります。テキストが入ります。テキストが入ります。</p>
                    </div>
                </a>
                ~省略~
            </div>
        </div>
    </section>

</main>
<?php get_footer(); ?>
```

ループ部分を記載していきます

archive-sweets.php

～省略～

```
<section class="section__page sweets__section01">
```

```
<div class="content content--large">
```

```
<div class="ttl__base">
```

```
<p class="ttl__baseSub"> スイーツ </p>
```

```
<h2 class="ttl__base--jp"> 新作スイーツ </h2>
```

```
</div>
```

```
<?php if ( have_posts() ) : ?>
```

```
<div class="sweetsList">
```

```
<?php
```

```
while ( have_posts() ) :
```

```
the_post();
```

```
?>
```

```
<a href="<?php the_permalink(); ?>" class="sweetsList__item">
```

```
<div class="sweetsList__img">
```

```
<?php if ( has_post_thumbnail() ) : ?>
```

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

```
<?php else : ?>
```

```

```

```
<?php endif; ?>
```

```
</div>
```

```
<h4 class="sweetsList__ttl"><?php the_title(); ?></h4>
```

```
<div class="sweetsList__txt">
```

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

```
</div>
```

```
</a>
```

```
<?php endwhile; ?>
```

```
</div>
```

```
<?php endif; ?>
```

```
</div>
```

```
</section>
```

～省略～

パーマリンク

サムネイルがある場合は表示  
ない場合は固定画像を表示

投稿のタイトルを表示

投稿の本文を表示

投稿があるかどうかを判定し、投稿がある場合表示

#### 14-7. 汎用パーツを作成する

投稿ループ部分はトップページでも同様のソースで設計しています。お知らせ一覧と違い、今回トップページではサムネイルのみの一覧となっていますので、その部分のみを共通パーツ化します。

テーマフォルダの template-parts フォルダ内に、archive-sweets.php を複製し「loop-sweets.php」とリネームします。今回は while の中身のサムネイル呼び出しの部分のみを残しそれ以外の記述を削除します。

/template-parts/loop-sweets.php

```
<div class="sweetsList__img">
  <?php if ( has_post_thumbnail() ) : ?>
    <?php the_post_thumbnail(); ?>
  <?php else : ?>
    
  <?php endif; ?>
</div>>
```

archive-sweets.php 内のサムネイル読み込み部分を、この loop-sweets.php を読み込むためのテンプレートタグに置き換えます。

archive-sweets.php

```
～省略～
<section class="section__page sweets__section01">
  <div class="content content--large">
    <div class="ttl__base">
      <p class="ttl__baseSub">スイーツ </p>
      <h2 class="ttl__base--jp">新作スイーツ </h2>
    </div>
    <?php if ( have_posts() ) : ?>
      <div class="sweetsList">
        <?php
          while ( have_posts() ) :
            the_post();
          ?>
          <a href="<?php the_permalink(); ?>" class="sweetsList__item">
            <?php get_template_part( 'template-parts/loop', 'sweets' ); ?>
            <h4 class="sweetsList__ttl"><?php the_title(); ?></h4>
            <div class="sweetsList__txt">
              <?php the_excerpt(); ?>
            </div>
          </a>
        <?php endwhile; ?>
      </div>
    <?php endif; ?>
  </div>
</section>
～省略～
```

## 15. フロントページに新作スイーツ一覧を読み込む

### 15-1. フロントページの一覧部分を読み込む

お知らせ一覧同様にサブグループで、スイーツ一覧 4 件分をフロントページに読み込みます。

post\_type はカスタム投稿スラッグを指定します。(今回は” sweets”)

front-page.php 変更前

```

~省略~
<div class="ttl__base txt--center fadein">
  <h3 class="ttl__base--jp"> 新作スイーツ </h3>
  <span class="ttl__base--en font--en">Sweets</span>
</div>
<div class="sweetsList fadein">
  <a href="<?php echo esc_url( home_url()); ?>/sweets/#sweets01" class="sweetsList__item">
    
  </a>
  ~省略~
</div>
~省略~

```

front-page.php 変更後

```

~省略~
<div class="ttl__base txt--center fadein">
  <h3 class="ttl__base--jp"> 新作スイーツ </h3>
  <span class="ttl__base--en font--en">Sweets</span>
</div>
<?php
  $sweets_args = array(
    'post_type' => 'sweets',
    'posts_per_page' => 4,
  );
  $asabenSweets_query = new WP_Query( $sweets_args );
  if ( $asabenSweets_query->have_posts() ) :
    ?>
    <div class="sweetsList fadein">
      <?php
        while ( $asabenSweets_query->have_posts() ) :
          $asabenSweets_query->the_post();
        ?>
      <a href="<?php the_permalink(); ?>" class="sweetsList__item">
        <?php get_template_part( 'template-parts/loop', 'sweets' ); ?>
      </a>
      <?php
        endwhile;
        wp_reset_postdata();
      ?>
    </div>
  <?php endif; ?>
~省略~

```

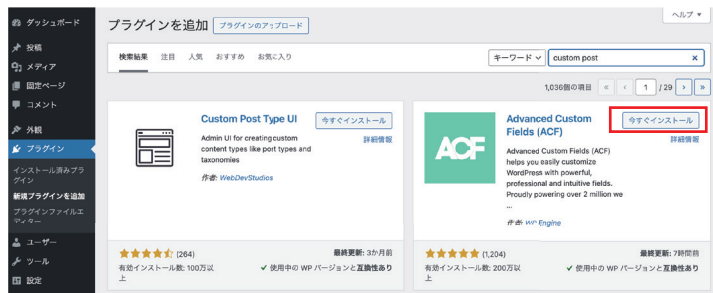
## 16. カスタムフィールドを作成する

## 16-1. カスタムフィールドとは

カスタムフィールドとは、固定ページや投稿などにある標準のタイトルや本文以外にも、独自に入力項目を追加できる機能です。

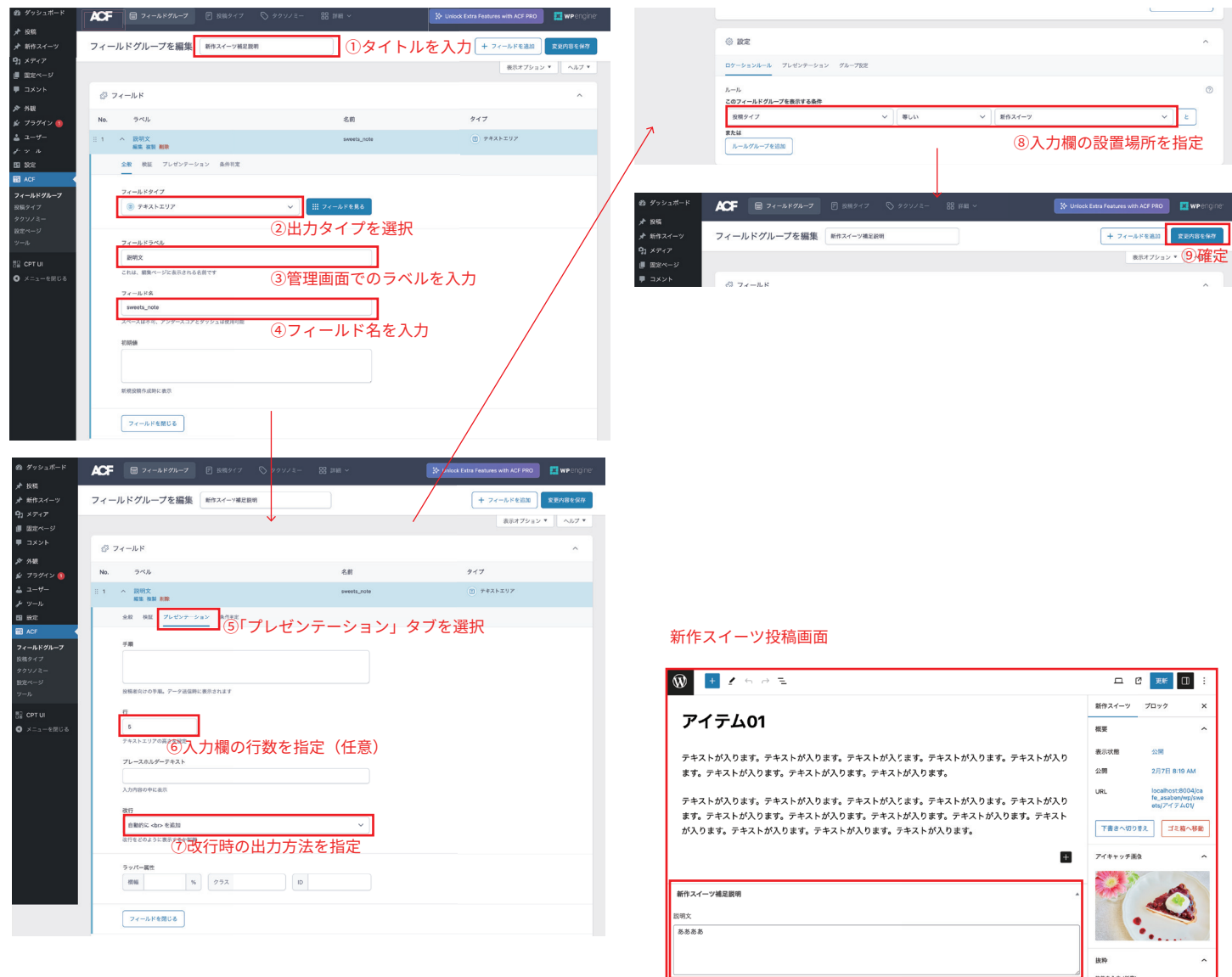
## 16-2. Advanced Custom Fields プラグインのインストール

[ プラグイン ] > [ 新規プラグインを追加 ] から、「Advanced Custom Fields」を検索し、プラグインをインストールして有効化します。



### 16-3. カスタム投稿タイプの追加

[ACF]> フィールドグループ [新規追加] から、カスタムフィールド作成に必要な設定をしていきます。





#### 16-4. カスタムフィールドを出力するためのコードを記述する

基本的に、テキストやテキストエリアは `the_field('フィールド名');` で出力することができます。  
 今回は、カスタムフィールドに記載がなければそのエリアごと表示させないように条件分岐を行います。  
 フィールド値を取得する `get_field()` を使用して、変数に代入し `echo` で出力する方法をとります。

single-sweets.php 変更前

```

～省略～
<section class="section__page sweets__section01">
  <div class="content content--small">
    <div class="singleCont__head">
      <h2 class="singleCont__ttl"> アイテム 1</h2>
    </div>
    <div class="sweetsSingle__img">
      
    </div>
    <div class="singleCont__cont">
      <p> テキストが入ります。テキストが入ります。～省略～テキストが入ります。</p>
    </div>
    <div class="sweetsSingle__note">
      <p> アレルギー物質：小麦粉, 卵 </p>
    </div>
  </div>
</section>
～省略～

```

single-sweets.php 変更後

```

～省略～
<section class="section__page sweets__section01">
  <div class="content content--small">
    <div class="singleCont__head">
      <h2 class="singleCont__ttl"> アイテム 1</h2>
    </div>
    <div class="sweetsSingle__img">
      
    </div>
    <div class="singleCont__cont">
      <p> テキストが入ります。テキストが入ります。～省略～テキストが入ります。</p>
    </div>
    <?php _____ 任意の変数名
      $note = get_field('sweets_note');
      if($note): _____ 16-3 で指定したフィールド名
      ?>
      <div class="sweetsSingle__note">
        <p><?php echo $note; ?></p>
      </div>
    <?php endif; ?>
  </div>
</section>
～省略～

```