

21 オリジナルテーマの作成 2

7. 固定ページの作成

7-1. page.php を作成する

固定ページのテンプレートファイルを作成していきます。

「index.php」を複製し、複製したファイルの名前を「page.php」に変更し、以下のように固定ページの読み込み用のタグに書き換えます。

変更前

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

<main class="main">
  ~省略~
</main>

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

変更後

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

<?php
if ( have_posts() ) :
while ( have_posts() ) : the_post();
?>
<?php the_content(); ?>
<?php
endwhile;
endif;
?>

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

7-2. 管理画面から固定ページを追加

管理画面から「固定ページ」→「新規固定ページを追加」を選択します。

- ①タイトルは作成する固定ページのタイトルを入力します。
- ②③「/concept/index.html」をテキストエディタで開き、共通部分以外のコンテンツ部分のソースをコピーし、「カスタム HTML」の入力ブロックを選択し、そこにペーストします。
- ④右メニューの「固定ページ」のタブを選択し、「URL」をクリックし、パーマリンクを任意の文字列に指定します。(コンセプトページは”concept”と指定)
- ⑤ここまでの入力が完了したら、右上の青い「公開」ボタンで公開します。

①タイトルを入力

②カスタム HTML を選択

③<main class="main"> ~</main> の内容をペースト

④パーマリンクを変更

⑤公開

②カスタム HTML を選択
選択肢がない場合は「全て表示」
で全選択肢を表示するか、検索バーで検索して選択。

8. functions.php を作成する

8-1.functions.php でパスの読み込みを変更する

src="../../ や href="../../ のままでは画像やリンクを読み込みません。functions.php に以下の記述を追加し、パスを自動で変換します。

```
// 画像パス変更
function replaceImagePath($arg) {
    $content = str_replace("../../assets/img/", "" . get_template_directory_uri() . '/assets/img/', $arg);
    return $content;
}
add_action('the_content', 'replaceImagePath');

// URL パス変更
function replaceUrlPath($arg) {
    $content = str_replace('href="../../', 'href="" . home_url() . '/', $arg);
    return $content;
}
add_action('the_content', 'replaceUrlPath');
```

../../assets/img/ という文字列を、get_template_directory_uri() . '/assets/img/' (テンプレート階層 /assets/img/ の URL) に変換
href="../../ という文字列を、home_url() . '/' (サイト URL/ の URL) に変換
を行っています。

※置換前後のパスはコーディングデータに合わせて適宜変更しましょう。

8-2.body のクラス名に固定ページのスラッグ名を出力する

header.php をテキストエディタで開き、body タグに個別のクラス名を出力するためのタグを設置します。

```
<body <?php body_class(); ?>>
```

このままでもサイトの HOME では body タグに「class=" home"」home クラスを、カテゴリーページでは「class=" category-slugname"」(slugname はカテゴリのスラッグ名)、投稿のページでは「class=" single"」というクラスを設定してくれます。

しかし、固定ページではページの ID 番号を記すだけでページのスラッグ名は表示してくれません。

そこで、body タグのクラス名に固定ページのスラッグ名を出力するための命令を functions.php に追記します。

```
// body のクラスにスラッグ名を出力
function my_body_class($classes)
{
    if (is_page()) {
        $page = get_post();
        $classes[] = $page->post_name;
    }
    return $classes;
}
add_filter('body_class', 'my_body_class');
```

8-2. アイキャッチを有効にする

投稿のアイキャッチ画像の設定を有効にする指定も追加します。

```
// アイキャッチ画像を有効化
add_theme_support('post-thumbnails');
```

9. 投稿用のテンプレートファイルの設定

9-1. 投稿を作成する

管理画面の「投稿」→「カテゴリ」から今回はダミーのカテゴリ（カテゴリ名「カテゴリ 1」/カテゴリースラッグ「cat01」）を、「投稿」→「タグ」からダミーのタグ（カテゴリ名「タグ 1」/カテゴリースラッグ「tag01」）を追加します。

次に、「新規投稿を追加」から、投稿を追加します。

- ①タイトルと本文を入力
- ②右メニューの「投稿」のタブを選択し、先ほど追加したカテゴリとタグを選択します。
- ③アイキャッチ画像を設定します。
- ④右上の青い「公開」ボタンで公開します。

9-2. 投稿詳細ページテンプレート single.php

コーディングデータ /news/single.html をテーマフォルダ内にコピーし、ファイル名を single.php にリネームします。
header, footer の共通部分やパスを、テンプレートタグに置き換えます。

次に、タイトルや本文の読み込みをテンプレート化していきます。

single.php 変更後

```
<?php get_header(); ?>
<main class="main">
  <div class="pkz">
    <ul class="pkz__list">
      ~省略~
      <li class="pkz__item"><?php the_title(); ?></li>
    </ul>
  </div>
  <section class="section__page news__section01">
    <div class="content content--large">
      <div class="content__row">
        <div class="content__row--left">
          <?php if ( have_posts() ) : ?>
          <?php
            while ( have_posts() ) :
              the_post();
            ?>
          <div class="singleCont__head">
            <h2 class="singleCont__ttl"><?php the_title(); ?></h2>
            <div class="newsList__data">
              <a class="newsList__cat" href=""> カテゴリー </a>
              <p class="newsList__time"><?php echo get_the_date(); ?></p>
            </div>
          </div>
          <div class="singleCont__cont">
            <?php the_content(); ?>
          </div>
          <?php endwhile; ?>
          <?php endif; ?>
        </div>
        ~省略~
```

投稿のタイトルを表示

投稿のタイトルを表示

投稿の日付を表示

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

カテゴリ名とカテゴリ一覧用のリンクを出力するための記述をします。

single.php 変更後

```

～省略～
<div class="singleCont__head">
  <h2 class="singleCont__ttl"><?php the_title(); ?></h2>
  <div class="newsList__data">
    <?php
      $category = get_the_category();
      if (!empty( $category )) :
        ?>
        <?php
          foreach($category as $cat):
            echo '<a class="newsList__cat" href="' . get_category_link( $cat->cat_ID ) . '">' . $cat->cat_name . '</a>';
          endforeach;
        ?>
        <?php endif; ?>
      <p class="newsList__time"><?php echo get_the_date(); ?></p>
    </div>
  </div>
～省略～

```

●カテゴリ出力方法 その他

テンプレートタグ（リンクつきリスト、区切りで出力の例）

```
<?php the_category(','); ?>
```

↓

出力例

```
<a href="http://example.com/category/cat01"> カテゴリー 1</a>,<a href="http://example.com/category/cat02"> カテゴリー 2</a>
```

テンプレートタグ（リンクつき ul リストで出力）

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

↓

出力例 ※ul に自動的に WordPress で用意されているクラス post-categories が付与される

```
<ul class="post-categories">
```

```
  <li><a href="http://example.com/category/cat01" rel="category tag"> カテゴリー 1</a></li>
```

```
  <li><a href="http://example.com/category/cat01" rel="category tag"> カテゴリー 2</a></li>
```

```
</ul>
```

9-3. サイドバーをテンプレート化する

サイドバー部分を header や footer と同じようにテンプレートファイル化し、読み込みます。

sidebar.php ファイルをテーマフォルダ内に作成します。

single.php 内の以下のサイドバーのソース部分をコピーして、sidebar.php 内にペーストし保存します。

次に、先ほどコピーした single.php のサイドバーのソース部分をテンプレート読み込みタグに変更します。

single.php 変更前

～省略～

```
<div class="content__row--right">
```

```
  <aside class="sidebar">
```

```
    <div class="sidebar__cont">
```

```
      <h2 class="sidebar__ttl"> 最新の投稿 </h2>
```

```
      <ul class="archive__list">
```

```
        <li class="archive__item">
```

```
          <a href=""> タイトルが入ります </a>
```

```
        </li>
```

```
      ～省略～
```

```
    </ul>
```

```
  </div>
```

```
  <div class="sidebar__cont">
```

```
    <h2 class="sidebar__ttl"> タグ </h2>
```

```
    <div class="tag__list">
```

```
      <a class="tag__item"> タグ </a>
```

```
    ～省略～
```

```
    </div>
```

```
  </div>
```

```
  <div class="sidebar__cont">
```

```
    <h2 class="sidebar__ttl"> アーカイブ </h2>
```

```
    <ul class="archive__list">
```

```
      <li class="archive__item">
```

```
        <a href="">2021 年 3 月 </a>
```

```
      </li>
```

```
    ～省略～
```

```
    </ul>
```

```
  </div>
```

```
</aside>
```

```
</div>
```

～省略～

sidebar.php にコピペ

single.php 変更後

～省略～

```
<div class="content__row--right">
```

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

```
</div>
```

～省略～

サイドバーの内容の動的な取得については後述します。

9-4. 投稿一覧ページテンプレート archive.php

コーディングデータ /news/index.html をテーマフォルダ内にコピーし、ファイル名を archive.php にリネームします。
header,footer,sidebar の共通部分やパスを、テンプレートタグに置き換えます。

次に、h2 タイトルの読み込み部分を変更していきます。カテゴリーのアーカイブや年月アーカイブページ用に条件分岐をしていきます。

```
<?php get_header(); ?>
<main class="main">
  ~省略~
  <section class="section__page news__section01">
    <div class="content content--large">
      <div class="ttl__base">
        <h2 class="ttl__base--jp">
          <?php if ( is_month() ) : ?>
            <?php echo get_the_date( 'Y 年 n 月 ' ); ?>
          <?php else : ?>
            <?php single_term_title(); ?>
          <?php endif; ?>
        </h2>
      </div>
    </div>
  ~省略~
```

月別アーカイブの場合：例 2024 年 1 月

ターム名を表示：例 カテゴリー名

次にループ部分をテンプレート化していきます。まず投稿一件ぶんを残してそれ以外を削除します。
そこにループの処理を施していきます。

変更前

```
<ul class="newsList">
  <li class="newsList__item">
    <a href=" ../news/singlel.html" class="newsList__link"></a>
    <div class="newsList__itemInner">
      <div class="newsList__img">
        
      </div>
      <div class="newsList__cont">
        <h3 class="newsList__ttl"> タイトルが入ります </h3>
        <p class="newsList__txt"> いつも CAFE ASABEN をご利用いただき、
          ありがとうございます。誠に勝手ながら、防火設備点検のため下記
          期間を臨時休業とさせていただきます。休業期間：2020 年 11 月 3
          日（火）ご不便をおかけいたしますが、何卒 [...]
        </p>
        <div class="newsList__data">
          <a href=" ../news/category.html" class="newsList__cat"> カテゴリー </a>
          <p class="newsList__time">2023 年 0 月 00 日 </p>
        </div>
      </div>
    </li>
  </ul>
```

投稿 1 件を残して削除

変更後

```

<?php if ( have_posts() ) : ?>
<ul class="newsList">
  <?php
  while ( have_posts() ) :
    the_post();
    ?>
    <li class="newsList__item">
      <a href="<?php the_permalink(); ?>" class="newsList__link"></a>
      <div class="newsList__itemInner">
        <div class="newsList__img">
          <?php if ( has_post_thumbnail() ) : ?>
            <?php the_post_thumbnail(); ?>
          <?php else : ?>
            
          <?php endif; ?>
        </div>
        <div class="newsList__cont">
          <h3 class="newsList__ttl"><?php the_title(); ?></h3>
          <?php the_excerpt(); ?>
          <div class="newsList__data">
            <?php
            $category = get_the_category();
            if ( !empty( $category ) ) :
              ?>
            <?php
            foreach( $category as $cat ):
              echo '<a class="newsList__cat" href="' . get_category_link( $cat->cat_ID ) . '">' . $cat->cat_name . '</a>';
            endforeach;
            ?>
          <?php endif; ?>
          <p class="newsList__time"><?php echo get_the_date(); ?></p>
        </div>
      </div>
    </li>
  <?php endwhile; ?>
</ul>
<?php endif; ?>

```

投稿がある場合表示

投稿をループ

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

カテゴリを表示

投稿の日付

「the_content)」タグは本文を出力しますが、一覧で本文すべてを出力するのは長すぎるので、代わりに投稿の抜粋を出力します。投稿の抜粋の出力は、テンプレートタグ「the_excerpt()」を使います。

「抜粋」欄に入力された文字を出力しますが、「抜粋」欄が未入力の場合は投稿本文のテキストを取り出し、自動で作成されます。

9-5. 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。そのため、使い回せるように汎用パーツ化します。

テーマフォルダ内に「template-parts」というフォルダを作成します。

template-parts フォルダ内に、archive.php を複製し「loop-post.php」とリネームします。while の中身を残しそれ以外の記述を削除します。

/template-parts/loop-post.php

```
<li class="newsList__item">
  <a href="<?php the_permalink(); ?>" class="newsList__link"></a>
  <div class="newsList__itemInner">
    <div class="newsList__img">
      <?php if ( has_post_thumbnail() ) : ?>
        <?php the_post_thumbnail(); ?>
      <?php else : ?>
        
      <?php endif; ?>
    </div>
    <div class="newsList__cont">
      <h3 class="newsList__ttl"><?php the_title(); ?></h3>
      <?php the_excerpt(); ?>
      <div class="newsList__data">
        <?php
          $category = get_the_category();
          if ( !empty( $category ) ) :
            ?>
          <?php
            foreach( $category as $cat):
              echo '<a class="newsList__cat" href="' . get_category_link( $cat->cat_ID ) . '"' . $cat->cat_name . '</a>';
            endforeach;
            ?>
          <?php endif; ?>
          <p class="newsList__time"><?php echo get_the_date(); ?></p>
        </div>
      </div>
    </div>
  </li>
```

archive.php 内の while の中身を、この loop-post.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.php

```
～省略～
<div class="content__row">
  <div class="content__row--left">
    <?php if ( have_posts() ) : ?>
      <ul class="newsList">
        <?php
          while ( have_posts() ) :
            the_post();
            ?>
            <?php get_template_part( 'template-parts/loop', 'post' ); ?>
          <?php endwhile; ?>
        </ul>
      <?php endif; ?>
    </div>
  </div>
  ～省略～
```

9-6. ページネーションを実装する

template-parts フォルダ内に、loop-post.php を複製し「parts-pagination.php」とリネームします。記述を削除し、ページネーション用のテンプレートタグを読み込みます。

the_posts_pagination() でページネーションを実装できますが、さらに「次」と「前」のページ送りも追加実装します。

/template-parts/parts-pagination.php

```
<?php
the_posts_pagination(
    array(
        'prev_text' => '&lt;<span class="pc--on"> 前 </span>',
        'next_text' => '<span class="pc--on"> 次 </span>&gt;',
    )
);?>
```

archive.php に、この parts-pagination.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.php

```
～省略～
<div class="content__row">
    <div class="content__row--left">
        <?php if ( have_posts() ) : ?>
            <ul class="newsList">
                <?php
                while ( have_posts() ) :
                    the_post();
                ?>
                <?php get_template_part( 'template-parts/loop', 'post' ); ?>
            <?php endwhile; ?>
            </ul>
        <?php endif; ?>
        <?php get_template_part( 'template-parts/parts', 'pagination' ); ?>
    </div>
～省略～
```

今回は生成される html にあらかじめ css でスタイルを当てています。