

21 オリジナルテーマの作成

1. WordPress のテーマ作成について

1-1. WordPress の必要最小の構成

Wordpress テーマファイルを作るには、「index.php」と「style.css」の2ファイルが必要最低限です。実際に Wordpress にテーマを追加してみましょう。

1-2. テーマを入れる場所

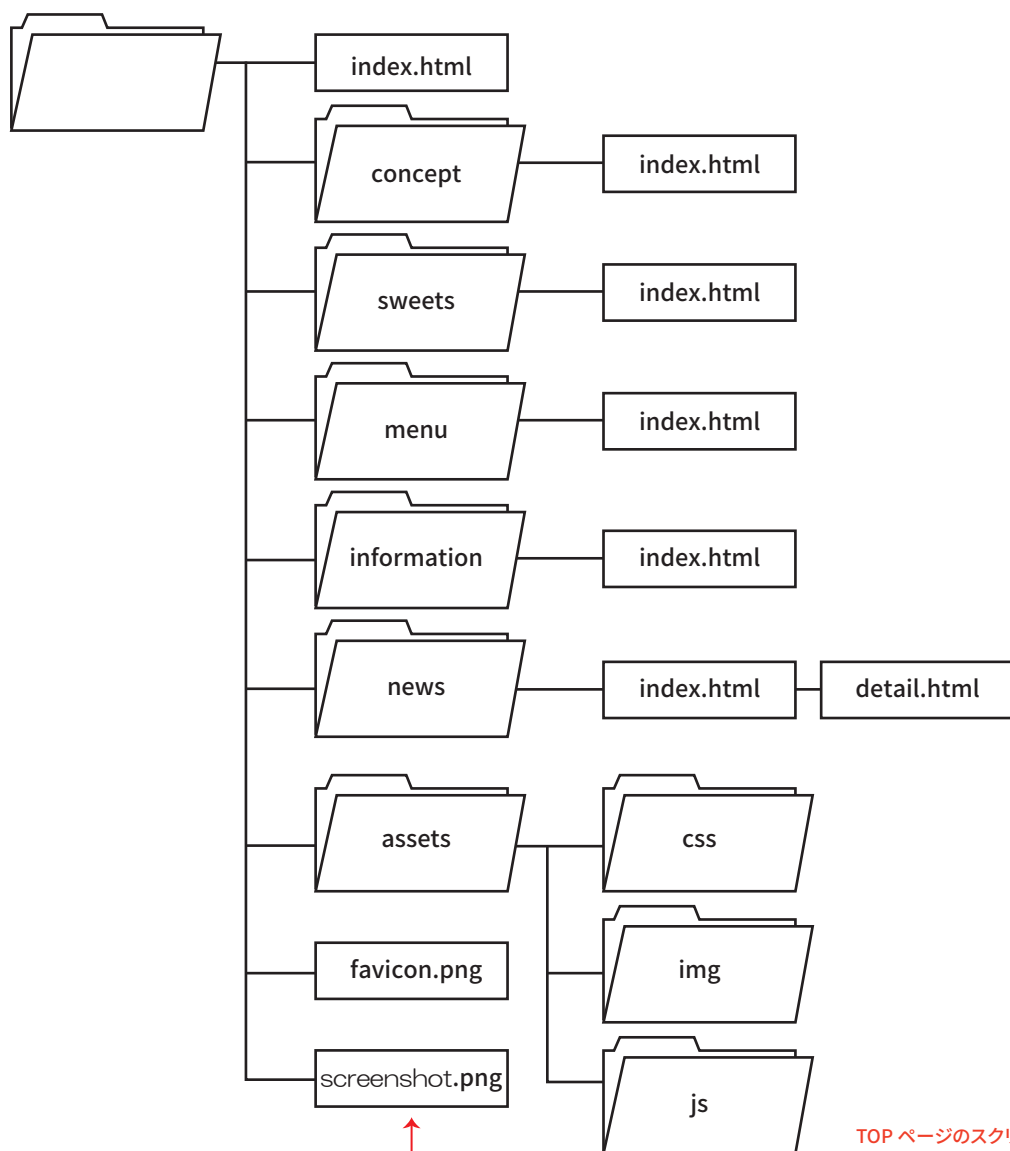
今回はローカル環境で作成するため、テーマを直接追加します。

xampp/htdocs/ 任意のフォルダ名 /wp-content/themes/ (MAMP の場合は MAMP/htdocs/ 任意のフォルダ名 /wp-content/themes/) を開いてください。フォルダ内には、初期テーマである「twentytwentytwo」「twentytwentythree」などのフォルダと「index.php」のファイルがあります。オリジナルテーマはここに追加していきます。

2. 静的 HTML ページの用意 (/original-theme01/cafe_asaben/)

2-1. サンプルサイトの概要

オリジナルテーマの作成の手順は様々ありますが、今回はすでに完成している静的 HTML ページを用意し、それをオリジナルテーマにする方法で作っていきます。



TOP ページのスクリーンショットです。
なくても問題はありませんが、テーマを視覚的に
分かりやすくすることが出来ます。
画像の名前を「screenshot」にすると適応されます。
880px×660px のサイズで、拡張子は .png に設定
することを推奨されています。

3. WordPress にテーマフォルダを追加する

3-1. theme フォルダの中にテーマフォルダをコピーする

「original-theme01/cafe_asaben/」のフォルダ内のファイルを、xampp/htdocs/ 任意のフォルダ名 /wp-content/themes/ （MAMP の場合は MAMP/htdocs/ 任意のフォルダ名 /wp-content/themes/）の中にコピーします。

次に、コピーしたフォルダを任意のテーマ名に変更します。

ここでは、「cafe_asaben-theme」に変更します。

3-2. テーマフォルダの中に style.css を作成する

つづいて、「cafe_asaben-theme」フォルダのなかに、必須ファイルである style.css を新規作成します。

この style.css には、テーマの名前やテーマの作成者の名前など、テーマに必要な情報を入力します。今回は以下のように入力して、保存してください。

```
/*
theme Name: オリジナルテーマ作成
Author: 山田太郎
Description: このテーマは、オリジナルテーマ作成の練習です。
*/
```

style.css の中に記述する内容は、上記以外にもいくつかあります。 基本は以下の通りです。

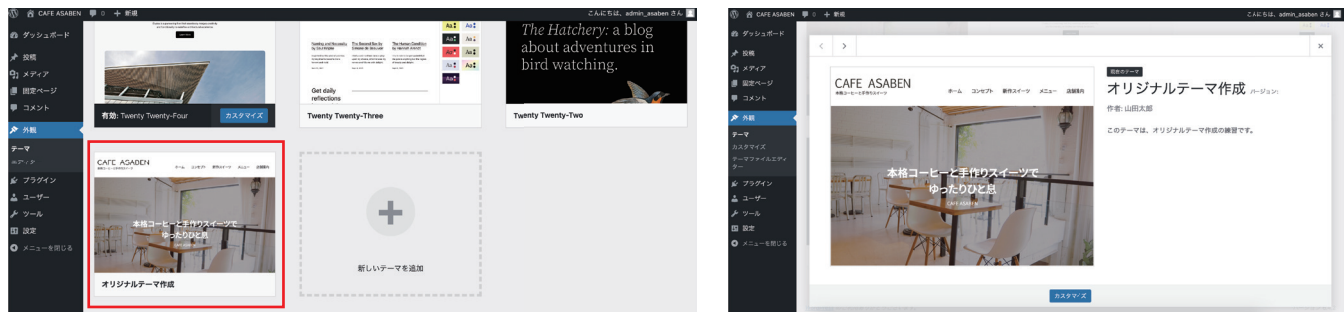
```
/*
Theme Name: テーマの名前（必須）
Description: テーマの説明
Theme URI: テーマの URL
Author: 作った人の名前
Author URI: 作った人の URL
Version: バージョン
License: ライセンス
License URI: ライセンスの URL
Tags: テーマの特徴を表すタグ（カンマ区切り / オプション）
Template: twentyseventeen // 子テーマ制作時限定 // 親テーマの「ディレクトリ名」を入力（Twenty Sixteen を例とする
*/
```

3-3. テーマフォルダの中に index.php を用意する

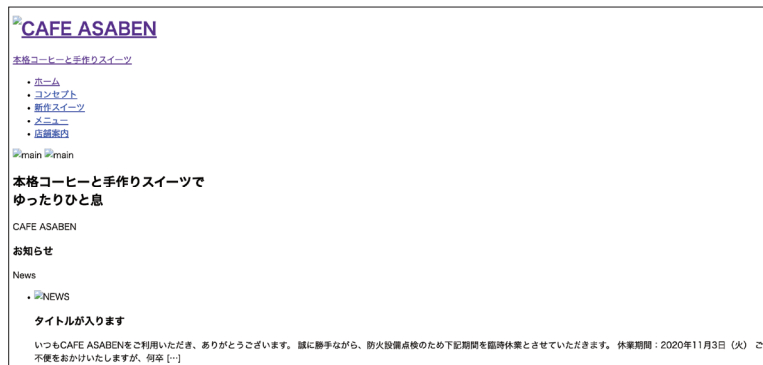
次に、必須ファイルである「index.php」を用意します。既にある「index.html」の拡張子を .php に変更し「index.php」にします。

これで、必要最低限のテーマファイルの作成が完了しました。実際に WordPress で「外観＞テーマ」を確認してみましょう。

新しく追加したオリジナルテーマが表示されました。テーマの名前とスクリーンショットが表示されています。
左クリックして詳細を確認すると、先程スタイルシートで設定した「テーマの名前」、「作成者の名前」、
「テーマの説明」が表示されます。



次に、有効化のボタンを押して、サイトを確認してみましょう。



サイトを確認すると、CSS ファイルが適用されていなかったり、画像が表示されなくなっていました。これは WordPress 内部でリダイレクトが行われているため、指定した相対パスの場所とは違う場所からファイルを読み込もうとするためです。
正しく表示させるには、現在相対パスで記述している箇所に、ワードプレス独自の関数を入れて修正する必要があります。

4. 相対パスの修正

4-1. CSS ファイルや JavaScript ファイルの相対パスの修正

index.php をテキストエディタで開きます。

<link> タグや <script> タグで記述された相対パスを以下のように修正します。

変更前

```
<link href="./assets/css/style.css" rel="stylesheet">
<script type="text/javascript" src="./assets/js/common.js"></script>
```

変更後

```
<link href="php echo get_template_directory_uri(); ?/assets/css/style.css" rel="stylesheet">
<script type="text/javascript" src="php echo get_template_directory_uri(); ?/assets/js/common.js"></script>
```

つづいて、ロゴ画像の相対パスも同じように修正しましょう。

変更前

```

```

変更前

```

```

5. テンプレートタグ

5-1. 「get_template_directory_uri();」について

先程、CSS や画像のパスに設定した「get_template_directory_uri();」は、現在利用しているテーマ (親テーマ) のパスを戻り値として返すWordPress独自の関数です。この指定をすることにより、相対パスで指定していたパスを、絶対パスに変更することができます。

このような WordPress 独自の関数のことを、テンプレートタグといいます。

5-2. タイトルを WordPress 管理画面から変更できるようにする

つづいて、ブラウザのタブなどに適応されるタイトルを WordPress 管理画面から変更できるようにしましょう。

「bloginfo();」というテンプレートタグを使います。このテンプレートタグは引数 ((); の中身) によって、WordPress 管理画面で設定した様々な情報を表示することができます。ここでは title タグに既に設定しているタイトルを、WordPress 管理画面の「設定＞一般」で設定したタイトルを表示させる「<?php bloginfo('name'); ?>」に置き換えます。

変更前

```
<title> サイト名が入ります </title>
```

変更前

```
<title><?php bloginfo('name'); ?></title>
```

ブラウザに表示させるのタイトルに、WordPress 管理画面で設定したサイトのタイトル名が表示されます。

5-3. リンク先をテンプレートタグに置き換える

つづいて、ナビゲーションメニューのリンクの URL を修正します。

ここでは、「home_url();」を使い、現在のブログのホーム URL を返します。

デフォルトの使い方として「<?php echo esc_url(home_url('/')); ?>」を推奨されているので、この書き方でリンクを修正していきます。

変更前

```
<a href=".." class="header__logo">～  
<a href="./concept/"> コンセプト </a>～
```

変更後

```
<a href="<?php echo esc_url( home_url( '/' ) ); ?>" class="header__logo">～  
<a href="<?php echo esc_url( home_url( '/concept/' ) ); ?>"> コンセプト </a>～
```

リンクの URL を修正することができました。

ナビゲーションメニューのリンクを押しても TOP ページが表示されるのは、まだ固定ページのテンプレートファイルを追加していないからです。

固定ページのテンプレートファイルを作成していない場合、index.php のテンプレートが表示されます。

5. テンプレートファイルについて

5-1. テンプレートファイルとは

テンプレートファイルとは、WordPress サイトの構成要素です。サイト上のどの Web ページを生成するときにも、パズルのピースのように組み合わせることができます。(WORDPRESS Codex 日本語版から引用)

現在、作成したオリジナルテーマには「index.php」と「style.css」のテーマを作成する上で必要最低限のテンプレートファイルがあるだけの状態です。この状態では固定ページや投稿ページは構成されず、表示されません。それぞれのページのテンプレートファイルを作成することによって、構成をすることが可能になります。

以下に、WordPress で必要になるテンプレートファイルを説明していきます。

index.php

テーマに必要なメインテンプレート。 基本は TOP ページのテンプレートファイルに使われます。

該当するテンプレートファイルがないときはすべて index.php が表示されます。

TOP ページのテンプレートにはほかに、「front-page.php」や「home.php」が使われる場合があります。

それぞれが存在する場合、TOP ページに適用される優先順位は以下の通りです。

front-page.php > home.php > index.php

header.php

ヘッダー出力用のテンプレート。 `<?php get_header(); ?>` を記述することで header.php を読み込むことができます。

footer.php

フッター出力用のテンプレート。 `<?php get_footer(); ?>` を記述することで footer.php を読み込むことができます。

page.php

固定ページのテンプレートファイルになります。WordPress の方で、自動的に固定ページのテンプレートファイルと認識されます。

single.php

投稿ページのテンプレートファイルになります。WordPress の方で、自動的に固定ページのテンプレートファイルと認識されます。

archive.php

投稿記事や、カテゴリーなどの一覧を表示させるページのテンプレートファイルです。

404.php

404 エラー（存在しない URL）が起きた際に表示されるページのテンプレートファイルです。

sidebar.php

サイドバー出力用のテンプレート。 `<?php get_sidebar(); ?>` を記述することで sidebar.php を読み込むことができます。

functions.php

いろいろな機能を加えるためのコードを梱包するファイルです。

テンプレートファイルは上記以外にもたくさん存在します。

一般的によくつかわれるテンプレートファイルは以上の通りです。

今回は既に作成した「index.php」のほかに、「header.php」と「footer.php」、「page.php」を追加していきます。

6. テンプレートファイルの分割、作成

6-1. 共通部分を分割する

TOP ページと下層ページ（固定ページ）には、同じコード内容が記述されている部分があります。これらの部分を共通部分といいます。

基本、header や footer は、TOP ページと下層ページの共通部分であることがほとんどです。この共通部分を分割し読み込むことによって、効率よくメンテナンス・更新をすることが可能になります。

「index.php」から「header.php」と「footer.php」を分割させていきます。

6-2. header.php を作成する

「index.php」を複製し、複製したファイルの名前を「header.php」に変更します。「header.php」をテキストエディタで開き、

<head> ～ </head> と <header> ～ </header> から下の記述を全て削除してください。

さらに、</head> 直前に wp_head() 関数を記述します。wp_head() 関数は head 内にコードを出力するための関数です。正しく記述されていないと、WordPress 本体や、テーマ、プラグインが正しく動作しない原因となります。

完了したら、上書き保存をします。

```
<!DOCTYPE html>
<html>
<head>
  ～省略～
  <?php wp_head(); ?>
</head>
<body id="home">
<header> ～省略～ </header>
これより下の記述をすべて削除
```

6-3. footer.php を作成する

先程と同じように「index.php」を複製し、複製したファイルの名前を次は「footer.php」に変更します。

「footer.php」をテキストエディタで開き、<footer> ～ </footer> より上の記述を全て削除してください。

さらに、</body> 直前に wp_footer() 関数を記述します。wp_footer() 関数は wp_head() 関数同様にコードを出力するための関数です。こちらも正しく記述されていないと、WordPress 本体や、テーマ、プラグインが正しく動作しない原因となります。

完了したら、上書き保存をします。

```
これより上の記述を全て削除
<footer class="footer">
  ～省略～
</footer>
<a class="pagetop">
  <span></span>
</a>
<!-- js -->
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script type="text/javascript" src="<?php echo get_template_directory_uri(); ?>/assets/js/common.js"></script>
<?php wp_footer(); ?>
</body>
</html>
```

6-4. 共通部分を読み込む

共通部分を分割することができました。次に、「index.php」に先程作成した「header.php」と「footer.php」を読み込みます。

既に「header.php」と「footer.php」で記述した内容は不要になるので削除し、代わりに「header.php」と「footer.php」を読み込むためのテンプレートタグを記述しましょう。

変更前

```
<!DOCTYPE html>
<html>
<head>
  ~省略~
</head>
<body id="home">
  <header> ~省略~ </header>

  <main class="main">
    ~省略~
  </main>

  <footer class="footer">
    ~省略~
  </footer>
  <a class="pagetop">
    <span></span>
  </a>
  <!-- js -->
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script type="text/javascript" src="<?php echo get_template_directory_uri(); ?>/assets/js/common.js"></script>
</body>
</html>
```

変更後

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

<main class="main">
  ~省略~
</main>

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

共通部分を読み込むことができました。

以降、header や footer を変更する際は、「header.php」や「footer.php」を変更します。