

WordPressで独自テーマを作る

大まかな手順

1. HTML/CSS/JSのページコーディング
(各代表ページ。例：ホーム、ブロッグ一覧ページ、ブロッグ記事ページ、お問い合わせ)
 2. WordPressのインストール
 3. テーマフォルダの作成
 4. ホームと、共通部分（ヘッダー、フッター、サイドバーなど）の実装
 5. アーカイブページ（記事一覧ページ）の実装、ページング（プラグインの導入）
 6. 記事ページの実装
 7. お問い合わせフォームの実装（プラグインの導入）
- 解説）テーマのテンプレートファイルが選ばれる仕組み＝テンプレート階層
 - 参考）バックアップの取得（プラグインの導入）
 - 参考）検索エンジンブロック設定（表示設定またはrobots.txt）

1. HTML/CSS/JSのページをコーディングしておく

WordPressは、テーマフォルダ直下にstyle.cssを置くルールがあるので、以下のような構成にしておくとWPへ移行し易い。

```
/ (HTML ではルート。テーマフォルダに対応させる)
-- style.css
-- index.html (ホーム)
-- blog-home.html (ブロッグ一覧)
-- single.html (ブロッグ記事)
-- contactUs.html (お問い合わせ)
-- /img
-- /css
-- /js
```

2. WordPressのインストール、パーマリンク設定、HTTPS利用時のさくら特有の設定

レンタルサーバーでは、簡単インストール機能が用意されている事が多い。
WordPressをダウンロードしてZIPを解凍後、サーバーへUPしてinstall.phpを実行してもOK。
(予め新しいDBを作り、wp-config.phpへ記述してからinstall.phpを実行する)

パーマリンク設定を変更する。（投稿名 : <https://frog-eight.sakura.ne.jp/rehe/sample-post/>）

?page_id=17の形以外で運用するほうが検索エンジン対策として有効。

さくらインターネットはhttpsにくせがあり、HTTPSの自動判別が一部動作しないので、以下をwp-config.phpへ追加する。

（https/http判別がデフォルトではうまくいかない為）

```
if( isset($_SERVER['HTTP_X_SAKURA_FORWARDED_FOR']) ) {  
    $_SERVER['HTTPS'] = 'on';  
    $_ENV['HTTPS'] = 'on';  
}
```

3. テーマフォルダの作成

テーマフォルダを作成する。**style.css**と**index.php**は必須。

今回は「oak」という無料のHTMLテンプレートをカスタムして、「oak」というWordPressのテーマを作る。

例) /home/frog-eight/www/rehe/wp-content/themes/oak

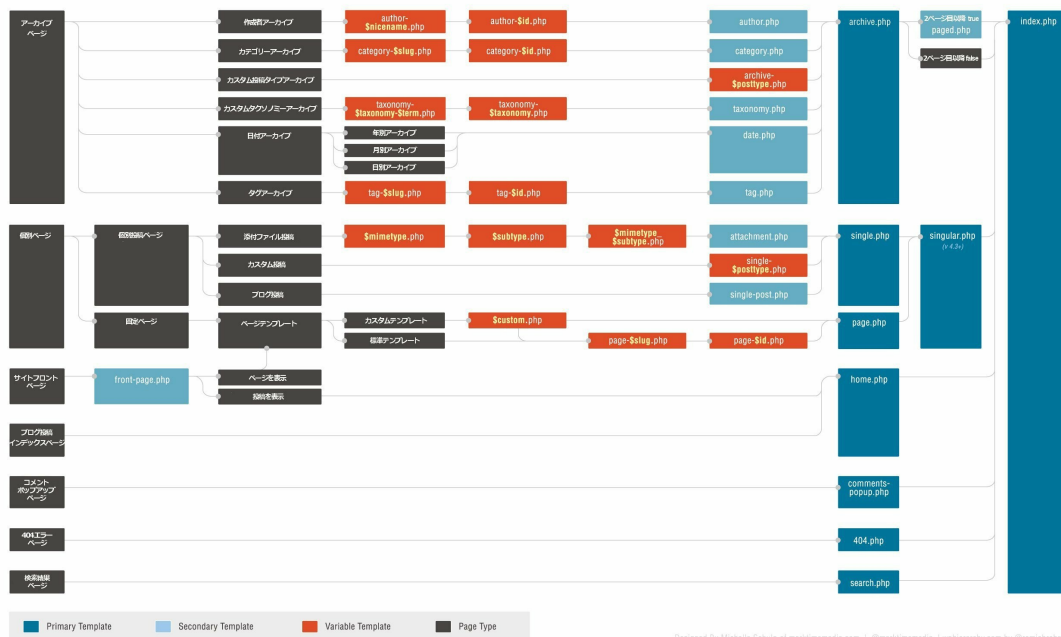
```
/ (HTML ではルート。テーマフォルダに対応させる)  
-- screenshot.png (外観のテーマ一覧に表示されるサムネイル用。推奨する画像サイズは 880x660)  
-- style.css (テーマのデフォルトCSS) *必須  
-- index.php (デフォルト汎用テンプレート) *必須  
-- front-page.php (ホーム用テンプレート)  
-- custom-blog.php (ブロッガー一覧: カスタム固定ページテンプレート)  
-- home.php (ブロッガー一覧: ブログホーム用テンプレート)  
-- single.php (ブログ記事: 投稿テンプレート)  
-- /img  
-- /css  
-- /js
```

参考) [WordPress Codex日本語版: テーマの作成](#)

解説) テーマのテンプレートファイルが選ばれる仕組み=テンプレート階層

WordPressは、テーマフォルダに存在するファイル名を優先度順に探して、ページ表示に使用する。

左から順に探し、最終的にはindex.phpが使用される。



4. ホームと、共通部分（ヘッダー、フッター、サイドバーなど）の実装

4.1 ヘッダー部分をheader.phpへ保存

ヘッダー部分を切り出し、header.phpへ保存。

header.phpの中でcssや画像など外部ファイルの相対パスは、テーマのディレクトリからのパスへ書き換える。

ヘッダーテンプレート（header.php）の呼び出し

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

'wp_head' アクションをスタートさせる。（＝プラグイン等がjsやcssを吐き出す位置）
テーマテンプレートファイル内の `</head>` タグ直前で使う。今回はheader.phpに記載。

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

有効化したテーマのディレクトリを表示

```
<?php echo get_template_directory_uri(); ?>
```

サイトのホームURLを表示

```
<?php echo esc_url( home_url( '/' ) ); ?>
```

header.phpにデフォルトスタイルの読み込みを追加。

```
<link rel="stylesheet" href="<?php echo get_template_directory_uri(); ?>/style.css">
```

4.2 フッター部分をfooter.phpへ保存

フッター部分を切り出し、header.phpへ保存。

footer.phpの中でcssや画像など外部ファイルの相対パスは、テーマのディレクトリからのパスへ書き換える。

フッターテンプレート（footer.php）の呼び出し

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

'wp_footer' アクションフックをスタートさせる。（＝プラグイン等がjsやcssを吐き出す位置）
テーマテンプレートファイル内の `</body>` タグ直前で使う。今回はfooter.phpに記載。

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

有効化したテーマのディレクトリを表示

```
<?php echo get_template_directory_uri(); ?>
```

サイトのホームURLを表示

```
<?php echo esc_url( home_url( '/' ) ); ?>
```

4.3 ホームの記事編集用に固定ページ「ホーム（スラッグ：home）」を作成し、表示設定にて、「ホームページ」を「ホーム」に設定する

ヘッダー・フッター以外の内容を切り取って、固定ページ「ホーム」の内容へテキストエディタからペーストする。

The screenshot shows the WordPress admin interface for editing a fixed page titled 'ホーム'. The left sidebar contains navigation menus like 'ダッシュボード', '投稿', 'メディア', '固定ページ', 'コメント', '外観', 'プラグイン', 'ユーザー', 'ツール', '設定', and 'メニューを閉じる'. The main content area is titled '固定ページを編集' and shows the page title 'ホーム', a permalink 'https://frog-eight.sakura.ne.jp/rehe/', and a rich text editor with HTML code. The right sidebar shows the '公開' (Public) status, a '変更をプレビュー' button, and the '固定ページの属性' (Fixed Page Attributes) section, which includes a '親' (Parent) dropdown set to '(親なし)' and a '順序' (Order) field set to '0'. A note at the bottom of the sidebar asks if help is needed and directs the user to the 'ヘルプ' (Help) tab.

functions.phpへ、テンプレートURLを埋め込むためのショートコード用関数を作成する。

「ショートコード」とは、管理画面のテキストエディタ上にPHPの処理を埋め込むための機能。

管理画面の中では、原則PHPコードは埋め込みできない。

```
//short code -----  
function shortcode_templateurl() {  
    return get_template_directory_uri();  
}  
add_shortcode('template_url', 'shortcode_templateurl');
```

エディタ内の相対パスは、[template_url]というショートコードを頭につけるよう、修正する。以下のように、編集画面上で編集する。20か所弱あるので、全て修正する。

```
<div id="slider-ef" class="slider-images-wrapper">  
      
      
      
</div>
```

ホームが一部崩れる場合、内容の改行が自動でpタグへ変換されていることが影響する場合がある。

その場合、自動変換を抑止することができる。

functions.phpへ記述する。

```
// 改行の時に自動的にPタグが挿入されるのを防ぐ  
remove_filter('the_content', 'wpautop');
```

index.phpでは、WordPressループからコンテンツの内容を呼び出すよう、変更する。

ループ内で the_content() と書くと、記事の内容を呼び出す。

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

解説) WordPressループ

WordPressは、記事の内容を呼び出す場合に「WordPressループ」という仕組みを記述する。投稿記事のURLを呼ぶと、該当する記事のデータを1件、ループの中で呼び出す。アーカイブページの場合は、URLに対応した複数の記事データを、ループの中で呼び出す。(繰り返し処理)
書き方は、以下。

```
<?php if ( have_posts() ) : ?>
    <?php while ( have_posts() ) : the_post(); ?>

    <!-- なにかの処理 ... -->

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

5. アーカイブページ（記事一覧ページ）の実装、ページング

5.1 カスタム固定ページテンプレートを作る

カスタム固定ページテンプレートとは、固定ページ用に任意に選択できるよう定義されたテンプレートのこと。

以下のコメントをテンプレートの先頭に入れると、認識される。

index.phpを別名保存して、テンプレートファイル「custom-blog.php」を作ったあと、コンテンツ部分をコピーする。

```
<?php
/*
Template Name: ブログ一覧用
*/
?>
```

テンプレートファイル「custom-blog.php」を、ブログ記事の一覧を表示するよう、カスタマイズする。

```
<?php
/*
Template Name: ブログ一覧用
```

```

*/
?>
<?php get_header(); ?>

<div class="container">
    <div class="header-page ef-parallax-bg" style="background-image:url(<?php echo get_
        <div class="col-md-6 col-md-offset-6">
            <div class="row">
                <div class="inner-content">
                    <div class="header-content">
                        <h1>Blog Posts</h1>
                        <hr>
                        <p>Everything created in simple way</p>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <div class="container margin-top">
        <div class="blog-wrapper">

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

        <div class="blog-post">
            <div class="blog-front-image">
                <div class="row">
                    <div class="col-md-6 wow animated fadeInUp" data-wow-delay="0.10s">
                        
                    <div class="col-md-5 col-md-offset-1">
                        <div class="blog-front-content wow animated fadeInUp" data-wow-
                            <div class="blog-front-content-inner">
                                <span class="post-date"><?php the_time('d M - Y'); ?></
                                <h1><?php the_title(); ?></h1>
                                <?php the_excerpt(); ?>
                                <a href="<?php the_permalink(); ?>"><i class="read-more
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>

    <?php endwhile; ?>
<?php endif; ?>

        </div>
    </div>

    <div class="container">
        <ul class="pagination-ef wow animated fadeInUp" data-wow-delay="0.20s">
            <li>
                <a href="#">

```

```

        <i class="pe-7s-angle-left"></i>
    </a>
</li>
<li class="current"><a href="#">1</a></li>
<li><a href="#">2</a></li>
<li><a href="#">...</a></li>
<li><a href="#">8</a></li>
<li>
    <a href="#">
        <i class="pe-7s-angle-right"></i>
    </a>
</li>
</ul>
</div>

```

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

custom-blog.phpがブローグ一覧で使われないので、home.phpのファイル名で保存したものをUPする。

サムネイルを投稿からUpできるように、functions.phpへ追加する。

```

// This theme uses post thumbnails
add_theme_support( 'post-thumbnails' );

```

5.2 固定ページ「ブログ（スラッグ：blog）」を作り、カスタム固定用テンプレートを指定する。



5.3 表示設定にて、「投稿ページ」を「ブログ」に設定する

(このとき、カスタム固定ページテンプレートは無効となるので注意する。home.phpを用意すると使用される。)

別名保存でhome.phpを作成してテーマフォルダへUpする。



5.4 ナビゲーションメニューを設定する

カスタムメニューを設定すると、管理画面からメニューを編集できる。
functions.phpへ以下を設定。

```
// This theme uses wp_nav_menu() in one location.  
register_nav_menus( array(  
    'gnav' => 'グローバルメニュー'  
) );
```

header.phpのナビゲーションのliタグを差し替える。

liタグを削除し、以下に差し替える。

```
<?php wp_nav_menu( array(  
    'theme_location' => 'gnav',  
    'items_wrap' => '%3$s'  
) ); ?>
```

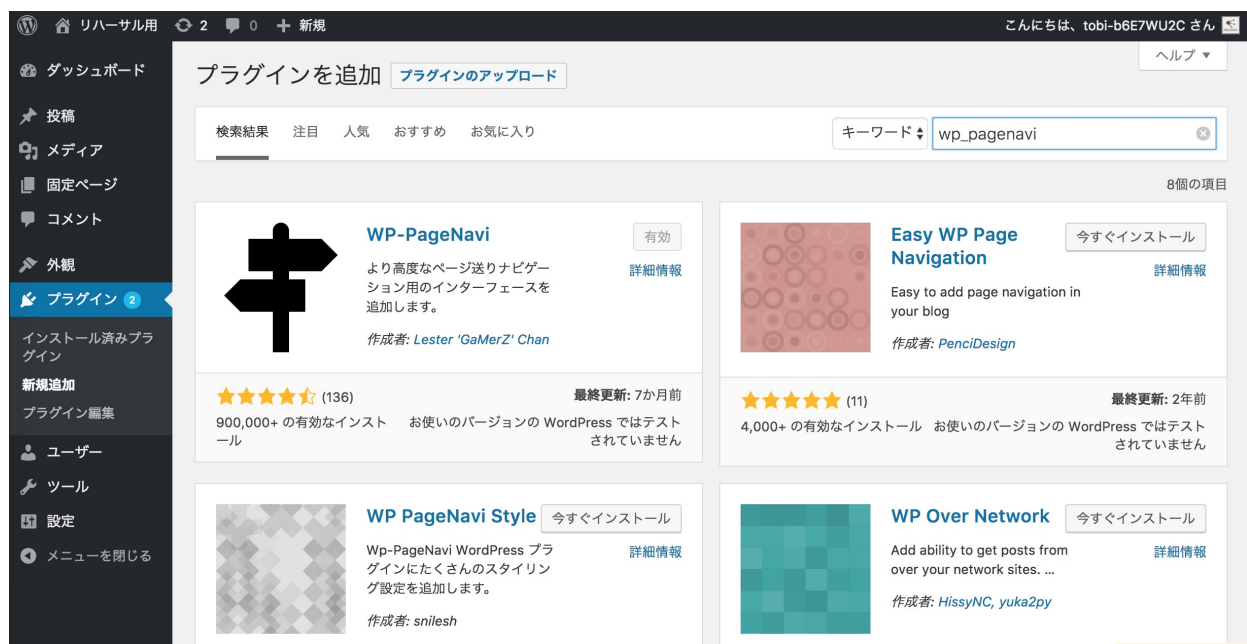
具体的には、header.phpのナビを以下のように変更。

```
<nav class="menu">  
    <div class="menu-list">  
        <ul>  
            <?php wp_nav_menu( array(  
                'theme_location' => 'gnav',  
                'items_wrap' => '%3$s'  
            ) ); ?>  
        </ul>  
    </div>  
</nav>
```

以下のように、外観＞メニュー から、メニュー内容を編集する。



5.5 ページングのプラグインを導入する



home.phpのページネーションの部分は、以下の通り修正。

```
<div class="container">
  <div class="pagination-ef wow animated fadeInUp" data-wow-delay="0.20s">
    <?php if(function_exists('wp_pagenavi')) {
      wp_pagenavi();
    } ?>
  </div>
</div>
```

6. 記事ページの実装

記事ページ用テンプレートをsingle.phpとして保存

home.phpをsingle.phpとして保存し、カスタマイズする。

```
<div class="blog-post">
  <div class="blog-front-image">
    <div class="row">
      <div class="col-md-6 wow animated fadeInUp" data-wow-delay="0.10s">
        
      </div>
      <div class="col-md-5 col-md-offset-1">
        <div class="blog-front-content wow animated fadeInUp" data-wow-delay="0.20s">
          <div class="blog-front-content-inner">
            <span class="post-date"><?php the_time( 'd M - Y' ); ?></span>
            <h1><?php the_title(); ?></h1>
            <?php the_content(); ?>
            <a href="<?php echo esc_url( home_url( '/' ) ); ?>blog/"><i class="
```

style.cssへ追加。

```
.read-more-blog-text{
  position: absolute;
  right: 36px;
  bottom: 6px;
  font-size: 14px;
  color: #fff;
}
```

7. お問い合わせフォームの実装（プラグインの導入）

固定ページで「お問い合わせページ（スラッグ：contact-us）」を作成する。

外観＞メニューにて、固定ページ「お問い合わせ」を追加する。

ソースは、contactUs.htmlを参考にする。（以下、ソース参照）

プラグイン>新規追加 で「contact form 7」をインストールして有効化する。

サイドメニュー「お問い合わせ」から、フォームの設定を行う。
デフォルトで、「コンタクトフォーム 1」が出来上がっているのので、そのまま使う場合はショートコードをコピーして記事に貼る。

編集のポイントは、html内にあったformタグは不要。

```
<div class="container">
  <div class="contact-map" id="map"></div>
  <div class="col-md-6 col-md-offset-6 col-sm-6 col-sm-offset-6 hidden-xs">
    <div class="row">
      <div class="inner-map">
        <div class="inner-map-content">
          <h1>Contact</h1>
          <hr>
          <p>Everything created in simple way</p>
        </div>
      </div>
    </div>
  </div>
</div>

<div class="container margin-top">
  <div class="contact-wrapper">
    <div class="row">
      <div class="col-md-3">
        <div class="contact-info">
          <span class="icon-info">
            <i class="pe-7s-phone"></i>
          </span>
          <span class="title-info">CALL US:</span>
          <span class="description-info">+001 123 222</span>
        </div>

        <div class="contact-info">
          <span class="icon-info">
            <i class="pe-7s-map-marker"></i>
          </span>
          <span class="title-info">ADDRESS:</span>
          <span class="description-info">Street 23, CITY p.n 300 Canada</span>
        </div>

        <div class="contact-info">
          <span class="icon-info">
            <i class="pe-7s-mail"></i>
          </span>
          <span class="title-info">EMAIL:</span>
          <span class="description-info">info@gmail.com</span>
        </div>
      </div>
    </div>
  </div>
</div>
```

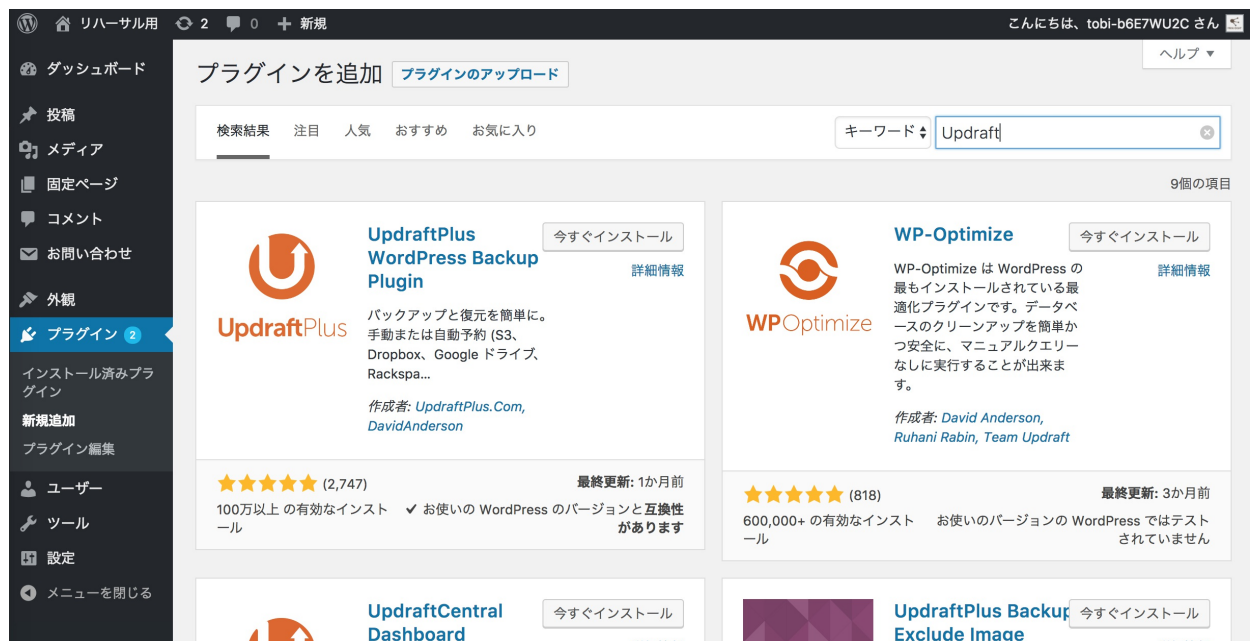
```
<div class="contact-form">
  <div class="col-md-5">
    [contact-form-7 id="33" title="コンタクトフォーム 1"]
  </div>
</div>
</div>
</div>
</div>
```

フォームの詳しいカスタマイズは、以下のマニュアルを参照。

[Contact form 7 使い方（日本語）](#)

参考）バックアップの取得（プラグインの導入）

「UpdraftPlus WordPress Backup Plugin」はバックアップとリストアが簡単にできる。
定期的なバックアップ自動実行も可能。



参考）検索エンジンブロック設定（表示設定またはrobots.txt）

設定＞表示設定 の「検索エンジンでの表示」の「検索エンジンがサイトをインデックスしないようにする」にチェックを入れる。

robots.txtを置く場合は、サイトのルートディレクトリへ。

User-agent: *

Disallow: /