

WordPressのためのPHP入門

WordPressのインストール

さくらインターネットのサーバーパネルから、データベースを作成

- データベース名は半角英数字小文字かハイフン・アンダースコア
- パスワードは半角英数字大文字小文字かハイフン・アンダースコア

生成サービス： [パスワード生成ツール](#)

サーバーパネルから、簡単インストール機能でWordPressをインストール

インストール先は、一旦httpで進める。任意のディレクトリへインストール。（例：rehe）
さくら簡単インストールでは、ZIPとwp-config.phpを用意してくれる。

「アプリケーションの設定へ進む」 から、 インストーラーへ進む。

例）frog-eightアカウントで、ディレクトリreheへインストールした場合
<http://frog-eight.sakura.ne.jp/rehe/wp-admin/install.php>

インストール後、サーバーパネルからドメイン設定でhttpsを有効にする

*ドメイン 一覧 *

新しいドメインの追加

ドメイン名	ウェブ		SSL			メール	
	アクション	パス	種別	利用中	証明書		
frog-eight.sakura.ne.jp ★	マルチドメイン	/	共有	表示	—	受信	変更 削除

<注意事項>

- ❗ 初期ドメイン(★)は削除できません。
- ❗ 共有SSL、SNI SSL、[Webアプリケーションファイアウォール](#)を利用したドメインは削除できません。ドメインを削除する前に解除してください。
- ❗ 追加したドメインがインターネット全体へ反映されるまでに、数時間～2日程度かかることがあります。
- ❗ ドメインを追加する事により「www.」が付与されたサブドメインが同時に使用可能となります。

[>> トップへ戻る <<](#)

WordPress管理画面へログインし、設定＞基本設定のURLを2か所、httpsへ変更する

[メディア](#)
[固定ページ](#)
[コメント](#)
[外観](#)
[プラグイン 2](#)
[ユーザー](#)
[ツール](#)
[設定](#)
[一般](#)
[投稿設定](#)

一般設定

サイトのタイトル

キャッチフレーズ
このサイトの簡単な説明。

WordPress アドレス (URL)

サイトアドレス (URL)
[サイトのホームページとして WordPress のインストールディレクトリ](#)にしてください。

テーマ「Twenty Seventeen」の子テーマを作成

子テーマとは、親のテーマを一部だけ修正したい場合に使う手段。

子テーマのフォルダにテンプレートファイルが存在する場合、それが優先される。ない場合は親のテーマフォルダのテンプレートが使われる。

フォルダ「rehe-child」を作成。

子テーマ

style.css にコメントを入れる。（以下）

Template: twentyseventeen という感じで、親テーマのフォルダ名を入れると認識される。

```
/*
Theme Name:      Rehe
Description:     Twenty Seventeen Child Theme
Author:          Maki Tobisawa
Author URI:      http://frog-right.com
Template:        twentyseventeen
Version:         1.0.0
License:         GNU General Public License v2 or later
License URI:     http://www.gnu.org/licenses/gpl-2.0.html
Text Domain:     rehe
*/
```

子テーマを適用してみると、CSSが崩れる。

子テーマのCSSだけでなく親テーマのCSSも読み込むよう、functions.phpにてenqueue_styleを設定する。

```
<?php
add_action( 'wp_enqueue_scripts', 'theme_enqueue_styles' );
function theme_enqueue_styles() {
    wp_enqueue_style( 'parent-style', get_template_directory_uri() . '/style.css' );
    wp_enqueue_style( 'child-style',
        get_stylesheet_directory_uri() . '/style.css',
        array( 'parent-style' )
    );
}
```

子テーマフォルダには、最低限「style.css」「functions.php」が必要。

トップページの表示設定

トップに最新の投稿（デフォルト10件）を表示したい場合

Twenty Seventeenの場合、index.phpが使われる。



トップに固定ページの内容を表示したい場合

Twenty Seventeenの場合、front-page.phpが使われる。



この2つのパターンを使って、PHPの記述のいろいろな方法を勉強していく。
親テーマから、index.php（投稿用）、front-page.php（固定ページ用）をコピーして修正し、reheへアップ。

front-page.phpにて試してみよう

PHPとして動かすには

拡張子.phpで保存し、WebサーバーにUpする。PHPが実行可能である必要がある。
通常のレンタルサーバーではほとんどの場合、実行可能となっている。
.htaccess などのWebサーバー設定により、拡張子.htmlでもPHPを実行可能とすることもできる。

PHPの書き方

phpファイルには、HTMLタグも混在できる。

```
//<?php で始まり、PHPコードの終わりはセミコロン。?>で終了。  
<?php PHPのコードをここに記述; ?>
```

コメント

1行コメントは//。#も使える。複数行は/* と */ で囲む。

```
<p><?php
# 1行コメント
//1行コメント
/*
複数行コメント
複数行コメント
*/
echo 'PHPからテキストを出力'; ?></p>
<?php echo '<p>PHPのechoで出力してみる</p>'; //HTML タグはPHP タグに含めないほうが煩雑にならない
```

変数

先頭に\$マークをつけると変数となる。

```
<p><?php //変数をつかってみよう。echoする前はエスケープ処理を入れて無害化する。
$myPostId = esc_html($post->ID);
$myPostTitle = esc_html($post->post_title);
$myPostSlug = esc_html($post->post_name);
echo '固定ページの投稿IDは「' . $myPostId . '」<br>';
echo '固定ページのタイトルは「' . $myPostTitle . '」<br>';
echo '固定ページのスラッグは「' . $myPostSlug . '」<br>';
?>
</p>
```

文字列

文字列はシングルクォーテーション「'」または、ダブルクォーテーション「"」で括る。
シングルクォーテーションの中では変数が有効。
ドット「.」で文字列や変数を結合することが可能。

```
<p><?php //変数をつかってみよう。echoする前はエスケープ処理を入れて無害化する。
$myPostId = esc_html($post->ID);
$myPostTitle = esc_html($post->post_title);
$myPostSlug = esc_html($post->post_name);
echo '固定ページの投稿IDは「' . $myPostId . '」<br>';
echo '固定ページのタイトルは「' . $myPostTitle . '」<br>';
echo '固定ページのスラッグは「' . $myPostSlug . '」<br>';
?>
```

</p>

アロー演算子

\$postはWordpressの記事データが格納されている変数。

\$postの属性「ID」の内容を参照するには、アロー演算子「->」を使う。

```
<p><?php //変数をつかってみよう。echoする前はエスケープ処理を入れて無害化する。
    $myPostId = esc_html($post->ID);
    echo '固定ページの投稿IDは「' . $myPostId . '」<br>';
?>
</p>
```

GET方式でページにパラメタと値を渡す

URLに?をつけ、パラメタ=値で指定可能。複数のパラメタを指定する場合は&で区切る。

PHP側では、スーパーグローバル変数\$_GET['パラメタ']で取得可能。

自由に値を設定できることから、不正なコードを混入されないよう、必ずエスケープ処理を行うこと。

(使用例：

[からころ配布施設・地図](#))

URLにparamというパラメタをつけて、何か値をページに渡して表示してみよう。

例) <https://frog-eight.sakura.ne.jp/rehe/?param=tobisawa>

```
<section class="first wrap">
    <h3>スーパーグローバル変数「GETパラメタ」を表示してみる</h3>
    <p>スーパーグローバル変数とは、スクリプト全体を通してすべてのスコープで使用可能な変数。<br>URLに/
    <?php $param = esc_attr($_GET['param']);
    echo 'GETパラメタparamの内容は「' . $param . '」です。'; ?>
</section>
```



エスケープ処理（無害化）

外部から取得したもの、関数の処理結果を使う場合などをページに表示したり、DBへ格納したりする場合は、必ずエスケープ処理を行うこと。

esc_html()は、HTMLコードとして使用するため無害化する。

esc_attr()は、属性値として使用するため無害化する。

esc_url()は、URLとして使用するため無害化する。

```
<?php $param = esc_attr($_GET['param']);  
<?php $param = esc_html($_GET['param']);  
<?php $param = esc_url($_GET['param']);
```

ダンプ

echoでは、文字列や数値などは表示できるが、配列やオブジェクトは表示できない。
\$postなどのオブジェクトの中身を確認したい場合は、以下の関数のいずれかでダンプする。

```
print_r(変数)  
var_dump(変数)  
var_export(変数)
```

使用例は、以下。ダンプ結果は、ソース表示から見たほうが見やすい。

```
<section class="first wrap">  
  <h3>変数の中身をダンプしてみる</h3>  
  <?php print_r($post); ?>  
</section>
```

index.phpにて、投稿やカテゴリを表示してみよう

トップページの表示を「最新投稿」に切り替えて、index.phpから投稿を表示してみよう。
設定＞表示設定 から設定可能。

まずは投稿を計3件に増やす。次にカテゴリを作成しよう。

カテゴリの例： 「日常」「技術ネタ」「お仕事」「ブログ（未分類を変更）」。

スラッグは英字に変更しておく。

それぞれの投稿に、カテゴリをつけてみよう。（複数OK）

元々あったWPループをフロントページ以外で表示、追加した記述をフロントページで表示するよう、IF文で制御。

```
<?php if ( is_home() || is_front_page() ) : ?>  
<!-- ここにトップ用のコードを書く-->  
<?php else : ?>
```



```
<!--ここに元々記載されていたコードを書く-->
<?php endif; ?>
```

配列の練習

配列は、arrayで定義する。配列にデータを追加するには、3パターンの書き方がある。（コメント参照）

```
<section class="first">
  <h3>配列の練習</h3>
  <p><?php
    $week = array('月','火','水','木','金'); //パターン1
    foreach($week as $day){
      echo '「'.$day.'」';
    }
    $week[] = '土'; //パターン2
    echo '「'.$week[5].'」';
    $week[6] = '日'; //パターン3
    echo '「'.$week[6].'」';
  ?></p>
</section>
```

連想配列の練習

連想配列は、パラメタ名と値のセットで格納される。

WordPressにおいては、各種関数への条件パラメタの設定に使用されることが多い。

```
<!--WPループで表示したい記事の条件 ここから-->
<!--連想配列で指定する-->
<?php
  $args = array(
    'post_type' => 'post', //投稿タイプ「投稿 (post)」
    'category__in' => array( 1, 2, 4 ), //カテゴリーID
  );
  $the_query = new WP_Query( $args );
?>
<!--WPループで表示したい記事の条件 ここまで-->
```

カテゴリーIDを確認するには、カテゴリー一覧からカテゴリー名のリンクへカーソルを当てると、ブラウザの下あたりにURLが表示され、tag_IDというパラメタから番号が分かる。（投稿IDも同様、postというパラメタ）



WordPressループを条件付きで実行する

WordPressで記事の内容を表示するには、「WordPressループ」という記述を使用する。デフォルトループと呼ばれるものには、条件式はない。

WP_Queryで実行する場合は、任意のパラメタを指定して実行することが可能。ページ内で複数のWP_Queryを実行することも可能。

```
<!--WPループの開始 ここから-->
<?php if ( $the_query->have_posts() ) : ?>
<?php while ( $the_query->have_posts() ) : $the_query->the_post(); ?>
<!--WPループの開始 ここまで-->
<!--記事毎の表示処理を書く ここから-->

<!--記事毎の表示処理を書く ここまで-->
<!--WPループの終了 ここから-->
<?php endwhile; ?>
<?php wp_reset_postdata(); ?>
<?php endif; ?>
<!--WPループの終了 ここまで-->
```

記事ごとの処理を書いてみる

articleタグで1件分の記事出力の処理を書いてみよう。
the_が先頭につくものは、echoしなくても結果がそのまま出力されます。
the_が先頭につくものは、「その記事の」という意味。

```
<!--記事毎の表示処理を書く ここから-->
<article>
```

```
<a href="<?php the_permalink(); ?>">
<h4><?php the_title(); ?></h4>
</a>
<p class="post_cat">カテゴリ: <?php the_category( ' ', ' '); ?></p>
<p class="post_date">更新日: <?php the_time( 'Y年n月j日' ); ?></p>
<div class="post_excerpt"><?php the_excerpt(); ?></div>
<hr>
</article>
<!-- 記事毎の表示処理を書く ここまで-->
```

CSSの表示調整をしてみよう

子テーマのstyle.cssに記述を書くと、親テーマのCSSに追加して定義されます。
(functions.phpにて、親テーマのCSSを読むよう、enqueue_styleを定義したため)

```
.first{
    padding: 3rem 0;
}
.post_cat{
    font-size: 0.875rem;
    float: left;
    width: 70%;
    color: #999;
}
.post_cat a{
    color: #3d81bd;
    transition: 0.3s;
}
.post_cat a:hover,
.post_cat a:active{
    color: #666;
}
.post_date{
    font-size: 0.875rem;
    float: right;
    width: 30%;
    color: #999;
}
```