forever 朝活

php 講座　WordPress の巻

株式会社フォーエバー

【目次】

WordPress を使ってみる 2
まずはインストール 2
最初の設定で使ってみる 5
少し設定をさわってみる 7
テーマを改造してみる 11
テーマごとに改造できる 11
オリジナルテーマを作ってみる 14
asakatsu テーマのガラを作成 14
CSS フレームワーク –foundation を使ってみる- 15
front-page.php を部品に分割します 21
固定ページを作る –page.php- 25
４０４ページの作成　-404.php- 26
お知らせページをアーカイブページにする 27

WordPress を使ってみる

php を使えば、掲示板が作れることは教科書でわかりました。しかし、ブログはどうでしょう。たしかに作れるのですが、デザインを変えたり、連携機能のトラックバックを付けたいと思ってもなかなか手が出ません。

そこで php で作られた CMS である WordPress を使ってみましょう。これを使えば簡単にブログを公開でき、さらに企業の固定ページにも対応でき、受注してから納品までの期間圧縮をすることができます。

php で作られているので、読んで内容がわかること、改造が可能なことが利点です。そしてなんといっても無料というところが、すばらしい。

まずはインストール

http://jp.wordpress.org/

このページの右下のダウンロードリンクから最新版をダウンロードします。

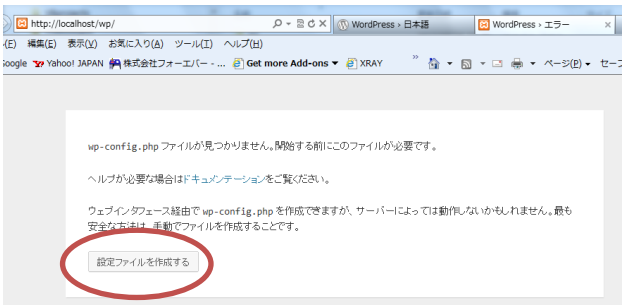
ダウンロードしたら、すべて展開します。そして、wordpress フォルダーを適当な名前に変更して(wp 等)htdocs フォルダー配下に移動します。

つづいて、xampp コントローラーから MySQL のａｄｍｉｎを開いて、wordpress データベーススペースを作成します。

続いて、

http://localhost/wp/

にアクセスします。すると設定がはじまります。



1）設定ファイルを作成する

2）さあ、始めましょう！

この画面で、必要事項を入力します。

データベース名は先ほど作ったデータベーススペース名。ユーザーは MySQL の ID。xampp ならばで admin。パスワードも MySQL のパスワード。xampp ならば「なし」。
データベースのホストは xampp ならば localhost。テーブル接頭語は MySQL のテーブル名称の頭文字。複数作るときに変えます。
送信をクリックします。

続いて、サイトの設定を行います。

サイトのタイトルは後からも変えられます。まずは設定。
ユーザー名はサイトの ID です。MySQL とは別です。同様にパスワードを入れます。
メールアドレスとプライバシーのチェックを外します。後から変えられます。
そしてインストールします。

そして成功です。

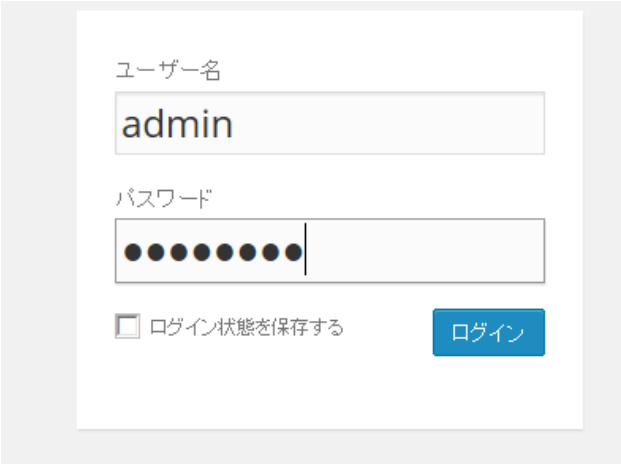
がっかりさせてゴメンナサイのいつものメッセージが出て成功です。

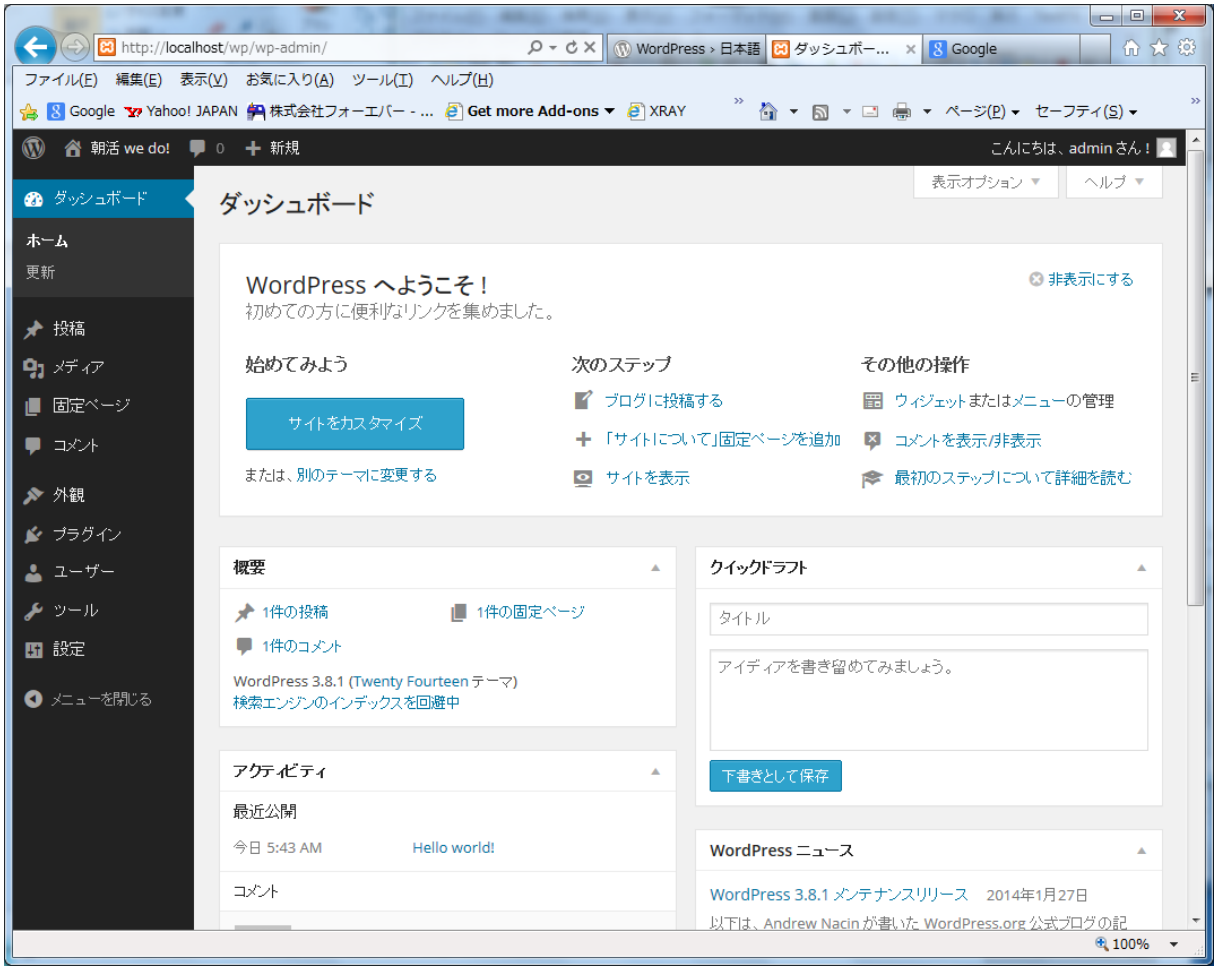
これが終わると、MySQL にテーブルができています。
wordpress の設定のほとんどは、このテーブルのなかにできます。

最初の設定で使ってみる

では、最初の設定でログインしてみましょう。

さきほど登録したサイトの ID とパスワードでログインします。





これが、最初のページのダッシュボード、管理画面です。

そして、ブログサイトページがこちらです。

これで、ブログがスタートしています。

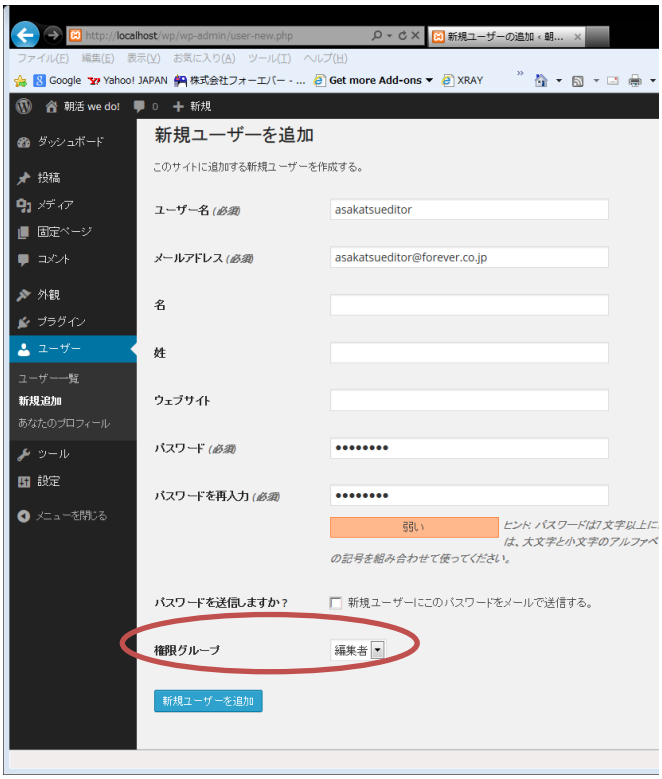
少し設定をさわってみる

まずダッシュボードの右上の表示オプションをクリックして、表示するものを変更できます。また、ようこそ以外のブロックは、タイトルバーをドラッグすれば移動できますので、自分の好きな形に変更できます。

まずは、ユーザーを追加しましょう。左のナビゲーションのユーザー＞＞新規追加をクリックします。

ユーザー名、メールアドレス、パスワードを入力し、権限グループを編集者にします。

これで、管理者以外のユーザーができました。必要な時以外は、admin は使わないようにしましょう。



続いて、プラグインを入れます。

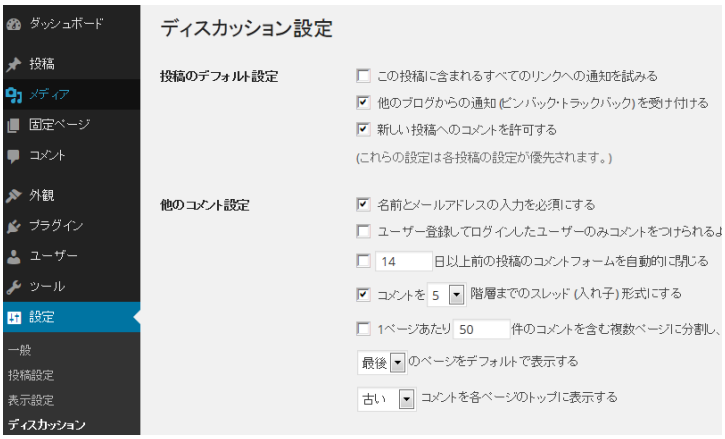
左のナビゲーションでプラグインをクリックすると、３つのプラグインが表示されますが、これはインストールされているだけで、有効化されていません。

ここで３つ目の WP Multibyte Patch を有効化します。

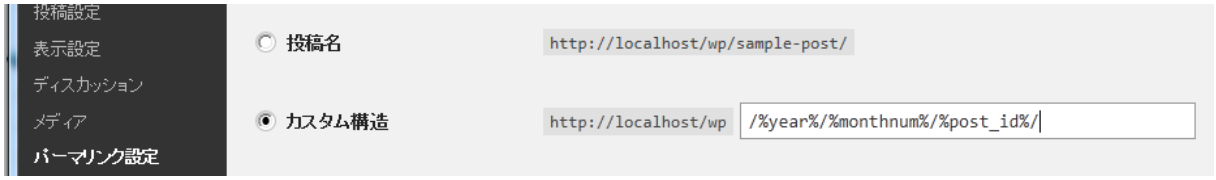
つづいて、キャッチフレーズを設定します。

ディスカッション設定でコメントやトラックバックの設定を調整します。

デフォルトはコメント等できますが、ビジネスサイトでは危険な場合もあります。



続いて、各投稿の保存場所の命名方法を設定します。デフォルトでは、日付で小分けにされないので、少し改造します。



／％ｙｅａｒ％／％ｍｏｎｔｈｎｕｍ％／％ｐｏｓｔ＿ｉｄ％／
にすると、／年／月／ID 番号になります。
投稿名にすると、日本語のとき困ったことになります。

続いて、カテゴリーを設定します。wordpress では、投稿するときにかならずカテゴリーを必要とします。

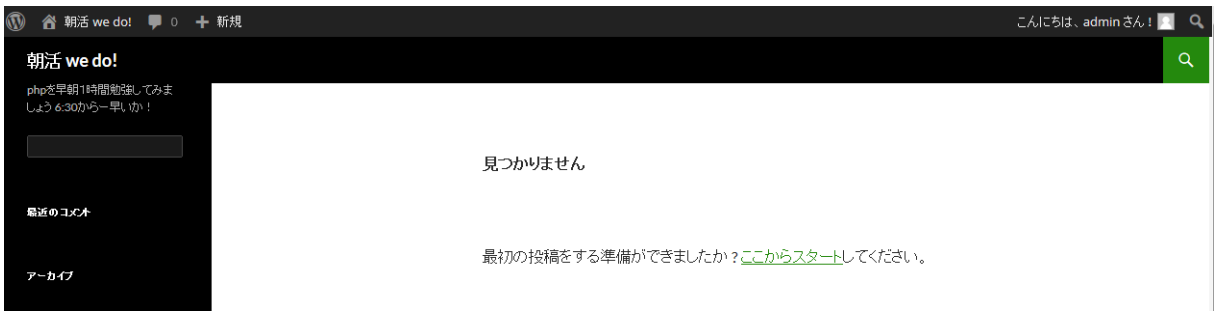
そして、最後にサンプルを消します。

投稿一覧で消したい投稿のゴミ箱をクリックします。

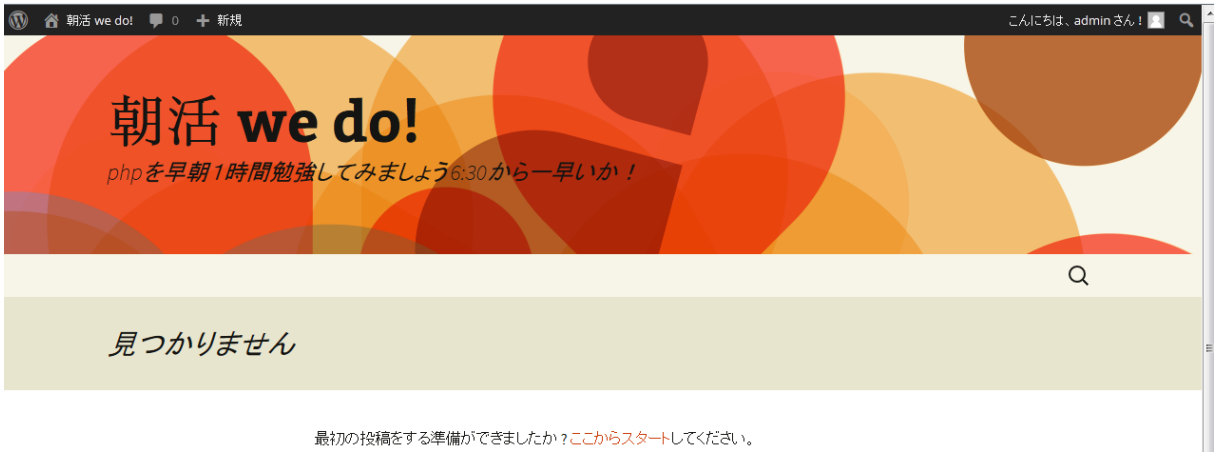
また、サイトの右上の緑で表示されているサンプルの固定ページを消します。

固定ページ一覧で、消したいページの下のゴミ箱をクリックします。

これで 2014 年版、Twenty Fourteen の出来上がりです。さっぱりした画面です。



同じデータでも、違うテーマでは違って見えます。
2013 年版、Twenty thirteen に切り替えてみましょう。外観メニューで別のテーマをだして、切り替えたいテーマを選んで、有効化します。



このようにテーマを変えて、見た目を変えることも、ソースを自分で修正して変えることも出来ます。

テーマを改造してみる

では、せっかく php が分かっているのですから、少し改造してみましょう。

テーマごとに改造できる

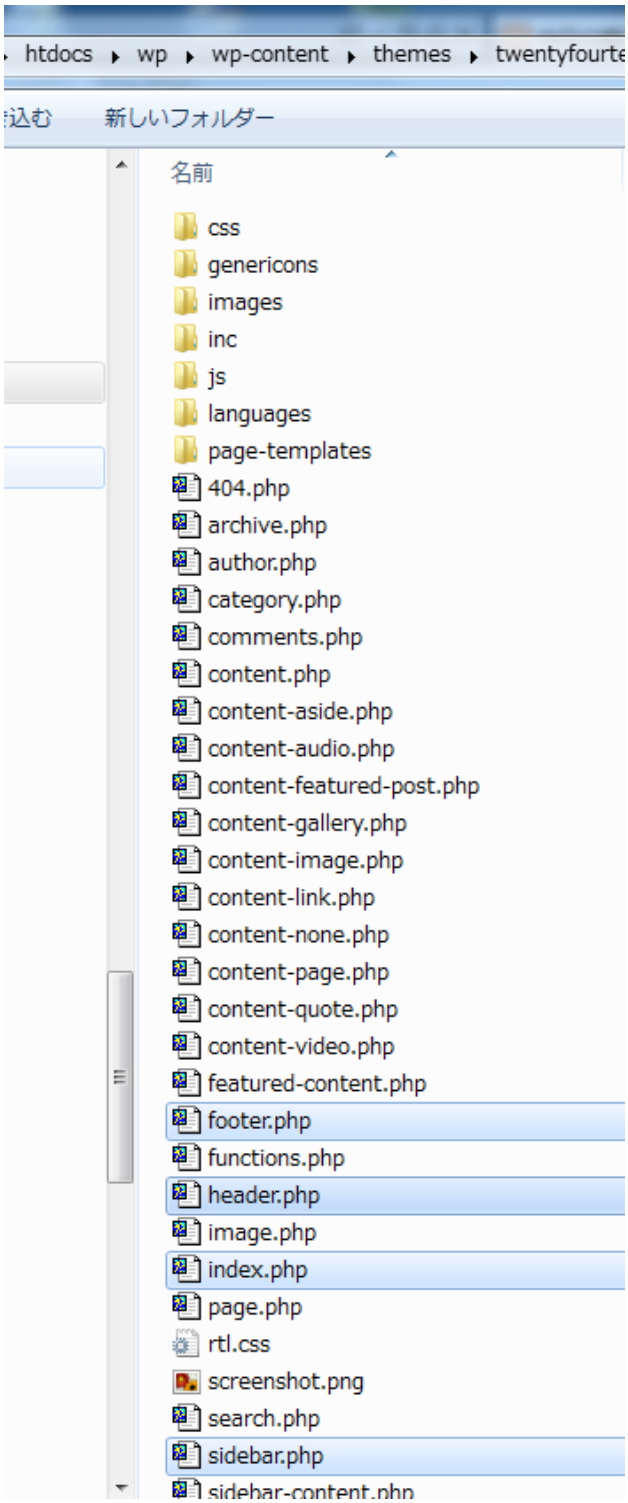
wordpress の php のソースは、テーマごとに分けられています。ですから、改造はテーマごとに行うことになります。

Twenty Fourteen のソースを見てみましょう。

wp/wp-content/themes/twentyfourteen のフォルダーに、右のようなファイルが入っています。

footer.php,header.php,index.php,sidebar.php が主なソースファイルで、これらで HTML を構成しています。

たとえば、sidebar.php の最後にこのような echo 文を入れると、

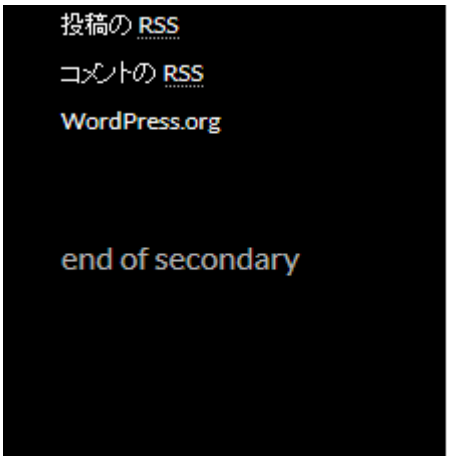


```
            </div><!-- #primary-sidebar -->
            <?php endif; ?>
            <?php echo 'end of secondary'; ?>
</div><!-- #secondary -->
```

最後から２行目に echo 文を入れると、

左のサイドバーの最後に、end of secondary と表示されました。

このように、各ソースで各 div を担当して、それが組み合わさって、全体が表示されています。



続いて、ブログのテンプレートをちょっと触ってみましょう。

現在、日付は、2014 年 2 月 25 日と日本語表記になっていますが、これを変更してみましょう。

```
function twentyfourteen_posted_on() {
		// Set up and print post meta information.
		printf( '<span class="entry-date"><a href="%1$s" rel="bookmark"><time class="entry-date" datetime="%2$s">%3$s</time></a></span> <span class="byline"><span class="author vcard"><a class="url fn n" href="%4$s" rel="author">%5$s</a></span></span>',
				esc_url( get_permalink() ),
				esc_attr( get_the_date( 'c' ) ),
				esc_html( the_time('Y/n/j') ),
				esc_url( get_author_posts_url( get_the_author_meta( 'ID' ) ) ),
				get_the_author()
```

wp/wp-content/themes/twentyfourteen/inc/templete-tags.php ファイルの twentyfourteen_posted_on()の中の printf 中の esc_html()の中身を get_the_date()から the_time('Y/n/j')に書き換えてみましょう。

すると、右の様に、日付の書式が 4 桁年／月／日と変わりました。

どこを変えるとなにが変わるかは、表示領域の div の id 名などを調べて、そのソースを検索するなどするとわかります。

今日2/25の朝活はphp日付関数でした

ただし、標準のテーマは、内部が非常に複雑に絡んでいますので、改造するのは難しいです。
以下のページでテンプレート構造を確認してください。
http://wpdocs.sourceforge.jp/テンプレート階層

ページの種類	テンプレートファイル名
404 エラーページ	404.php
検索結果ページ	search.php
アーカイブページ	archive.php
個別投稿ページ	single.php
固定ページ	page.php

サイトフロントページ	front-page.php
ブログ投稿インデックスページ	home.php
コメントポップアップページ	commnet-popup.php

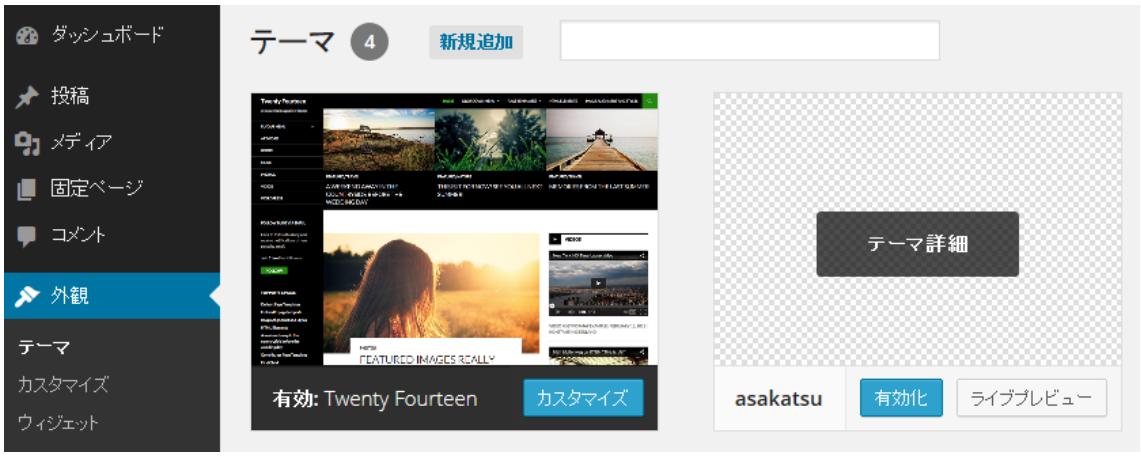
オリジナルテーマを作ってみる

標準テーマはさわりづらいので、単純なオリジナルテーマを作ってみましょう。

asakatsu テーマのガラを作成

themes の中に asakatsu フォルダーを作り、空の index.php と空の style.css を入れておきます。

そして、admin で入ってから、外観のテーマを見てみましょう。



有効化すると、サイトを見た時真っ白になってしまいます。

０から作っていきましょう。ｓｔｙｌｅ．ｃｓｓを改造します。

```
/*
Theme Name: 朝活
Author: eternal laboratory
Author URL: http://eternalkagosima.chobi.net/
Description: 朝活用テーマ
Version: 1.0
*/
```

みなさんは、自分なりに作り変えてください。

すると、再読み込みするとテーマが変わります。

front-page.php

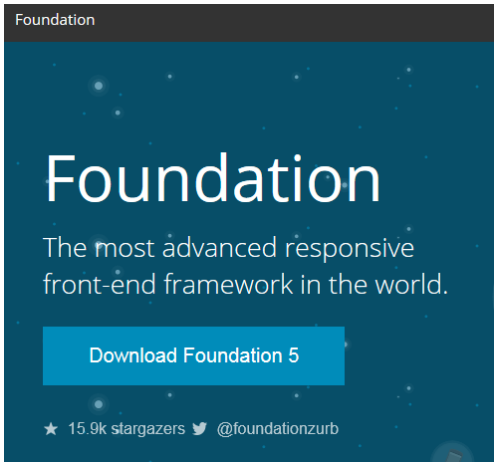
```
<!DOCTYPE html>
<html lang="ja">
<head>
        <meta charset="UTF-8">
        <title></title>
</head>
<body>
        <h1><?php bloginfo('name'); ?></h1>
</body>
</html>
```

このようなファイルを同じフォルダーに追加して、サイトを表示してみましょう。

今作ったファイルは、サイトのページでした。ですから、サイトの画面が表示されました。このようにして、それぞれの画面はテンプレートという形で、別々のソースになっています。

wordpress は、今どの画面を出すのか判断して、テンプレートを切り替えて表示していきます。

切替は、URL によって行い、そのテンプレートを読み込み、テンプレートが必要なデータをデータベースから取得して、画面に出力するという流れになっています。



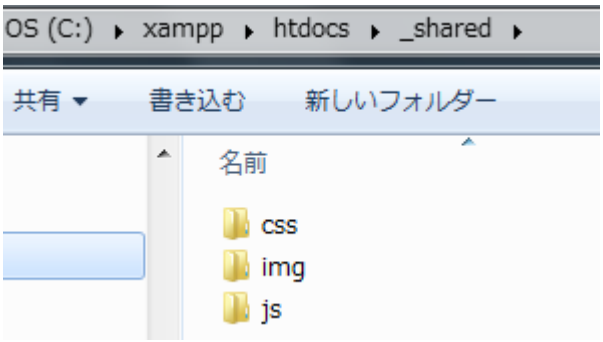
CSS フレームワーク -foundation を使ってみる-

最近の流れとして、パソコンのブラウザーだけでなく、スマートホンで wordrdpress を見ることが多くなっています。

そこで、折角ですので、CSS のフレームワークの一つ Foundation を使って、レスポンシブなテーマを無料で手に入れてしまいましょう。

まずは、http://foundation.zurb.com にアクセスして、Foundation5 をダウンロードします。

ZIP ファイルを解凍したら、xampp の共通として使いたいので、htdocs の配下に _shared フォルダーを作って、そこに CSS,IMG,JS などのフォルダーを移動します。



そして、foundation を使った front-page.php のサンプル版を作ってみましょう。

```
<!DOCTYPE html>
<!-- [if IE 8]>	<html class="no-js lt-ie9" lang="ja">><![endif]-->
<!-- [if gt IE 8]><!-->	<html class="no-js" lang="ja">	<!--<![endif]-->

<head>
	<meta charset="utf-8">
	<meta name="viewport" content="width-device-width" />
	<title>朝活 ｜ php 復習講座</title>
	<link rel="stylesheet" href="/_shared/css/foundation.css" />
	<script src="/_shared/js/vendor/modernizr.js"></script>

	<style type="text/css">
	header #news p{
		margin: 0;
	}
	header h1 {
		margin: 50px auto 80px;
		text-align: center;
	}
	</style>
</head>
```

```
<nav class="top-bar" data-topbar>
	<ul class="title-area">
		<!-- Title Area -->
		<li class="name">
			<h1><a href="/">朝活 we do!</a></h1>
		</li>
		<li class="toggle-topbar menu-icon"><a
href="#"><span>menu</span></a></li>
	</ul>

	<section class="top-bar-section">
		<!-- Right Nav Section -->
		<ul class="right">
			<li><a href="/">トップ</a></li>
			<li><a href="/news">お知らせ</a></li>
			<li><a href="/contact">お問い合わせ</a></li>
		</ul>
	</section>
</nav>

<div id="front-page" class="row">
	<div class="large-12 columns">

		<header>
			<h1><img src="/_shared/img/logo.png" alt="asakatsu"
/></h1>

			<div id="catchcopy" class="row">
				<div class="columns large-4">
					<h2>朝活</h2>
				</div>
				<div class="columns large-8">
					<p>朝 6:30。朝早いですが、この時間に勉強
すると得した気分になります。まさに「早起きは三文の得です」</p>
				</div>
```

```
                    </div>
                    <section id="news">
                        <a href="http://localhost/2014/2/1/">
                            <div class="alert-box">
                                <p>お知らせ - 2014.02.25</p>
                            </div>
                        </a>
                    </section>
                </header>

                <div id="content">
                    ...
                </div>
            </div>
        </div>
        <footer class="row">
            <div class="large-12 columns">
                <hr />
                <div class="row">
                    <div class="large-6 columns">
                        <p>&copy; Eternal lab</p>
                    </div>
                </div>
            </div>
        </footer>
        <script src="/_shared/js/vendor/jquery.js"></script>
        <script src="/_shared/js/foundation.min.js"></script>
        <script>
        $(document).foundation();
        </script>
        </body>
        </html>
```

wordpress の関数はまだ使っていませんが、だいたいの見栄えがわかります。

また幅を狭くすると

こんな感じになります。

ナビゲーションはこんなふうにまとまりました。
レスポンシブになっています。

今回作った front-page.php に WordPress のテンプレートタグを入れていきましょう。

まず、/style と/head の間に wp_head();を追加してヘッダーを出します。

```
        </style>
        <?php wp_head(); ?>
</head>
```

つづいて、/body の直前に wp_footer()を追加して、wordpress 管理から見てみましょう。

```
</script>
<?php wp_footer(); ?>
</body>
```

みごと、ドッキングしました。

さて、次は、本当のメッセージを表示するように改修しましょう。あらかじめ、未設定のカテゴリーを「お知らせ」スラッグを news に変更して、１メッセージ入れておいてください。

front-page.php の section od="news"の中身を次のように書き換えます。posts_per_page=1 なので１メッセージだけ表示されます。

```
<section id="news">
        <?php $news = get_posts('posts_per_page=1'); ?>
        <?php foreach($news as $post): ?>
        <?php setup_postdata($post); ?>
        <a href="<?php the_permalink(); ?>">
                <div class="alert-box">
                        <p><?php the_title(); ?> - <?php the_date(); ?></p>
                </div>
        </a>
        <?php endforeach; ?>
        <?php wp_reset_postdata(); ?>
</section>
```

これで再表示してみましょう。

まだリンクは変ですが、これで wordpress と連動しました。

front-page.php を部品に分割します

front-page.php をヘッダー、フッター、サイドバーに分割します。

先頭から/nav までを header.php に分割。その場所に get_header();を入れます。
続いて、footer 以下全部を footer.php に分割。その場所に get_footer();を入れます。

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

<div id="front-page" class="row">
        <div class="large-12 columns">
```

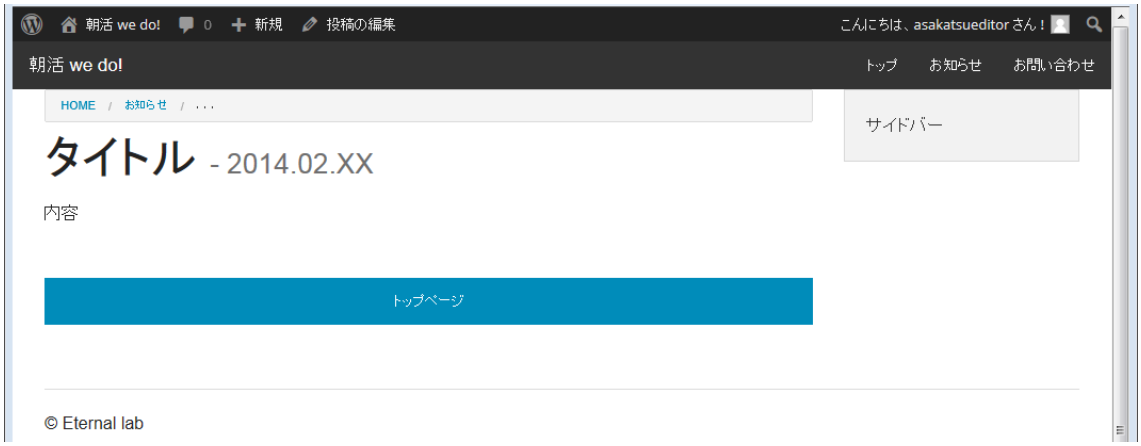
```
        </div>
</div>

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

これで、見た目が変わらなければ OK です。

これで、分割したテンプレートを使って、個別投稿ページを作りましょう。
個別投稿画面は、single.php になります。

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

<div class="row">
	<div class="large-9 columns">
		<nav class="breadcrumbs">
			<a href="/wp/">Home</a>
			<a href="/wp/news">お知らせ</a>
			<a class="current">...  </a>
		</nav>

		<article id="content" style="margin-bottom: 3em">
			<h1>タイトル  <small>- 2014.02.XX</small></h1>

			<p>内容</p>
		</article>

		<p><a href="/wp" class="button expand small">トップページ</a></p>
	</div>

	<div id="sidebar" class="large-3 columns">
		<div class="panel">
			<p>サイドバー</p>
		</div>
	</div>
</div>

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

これで個別を画面を見ると、



これで、個別画面の仮ページができました。サイドバーも出ています。

細かいところで言うと、タイトルが固定になっています。
そこで title を php で切り分けます。

ｈｅａｄｅｒ．ｐｈｐ

```
<?php if (is_front_page()): ?>
        <title><?php bloginfo('name'); ?> - <?php bloginfo('description'); ?></title>
<?php else: ?>
        <title><?php wp_title(''); ?>  |  <?php bloginfo('name'); ?></title>
<?php endif; ?>
```

これで、それぞれのページでタイトルが正しく表示されます。

続いて、個別ページを正しく変更します。
ｓｉｎｇｌｅ．ｐｈｐの先頭から/article までを改造して、データベースの内容を表示するようにします。

```
<?php get_header(); ?>
<?php the_post(); ?>
<div class="row">
        <div class="large-9 columns">
                <nav class="breadcrumbs">
                        <a href="/wp/">Home</a>
                        <a href="/wp/news">お知らせ</a>
                        <a class="current"><?php the_title(); ?></a>
                </nav>
                <article id="content" style="margin-bottom: 3em">
                        <h1><?php the_title(); ?> <small>- <?php the_date(); ?></small></h1>
                        <?php the_content(); ?>
                </article>
```

これで、データベースを読み込むようになります。
サイドバーも header と同じように分解しましょう。
ｓｉｄｅｂａｒ．ｐｈｐ

```
        <div id="sidebar" class="large-3 columns">
                <div class="panel">
                        <p>サイドバー</p>
                </div>
        </div>
```

データベースからデータを読み込んで表示できるようになりました。

固定ページを作る -page.php-

固定ページを作りましょう。

固定ページに「お知らせ」を作成しパスを/wp/news にします。

同様に「お問い合わせ」 /wp/contact

同様に「トップページ」 /wp/top とします。

single.php から page.php を作ります。

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

<div class="row">
	<div class="large-9 columns">
		<nav class="breadcrumbs">
			<a href="/wp/">Home</a>
			<a class="current"><?php the_title(); ?></a>
		</nav>

		<article id="content" style="margin-bottom: 3em">
			<h1><?php the_title(); ?></h1>
			<?php the_content(); ?>
		</article>

		<p><a href="/wp" class="button expand small">トップページ</a></p>
	</div>
	<?php get_sidebar(); ?>
</div>

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

404ページの作成　-404.php-

不正なURLが指定された時や固定ページが削除された時に表示するメッセージを404エラーといいますが、そのページを用意します。
page.php をコピーして作成します。

４０４．ｐｈｐ

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

<div class="row">
	<div class="large-9 columns">
		<nav class="breadcrumbs">
			<a href="/wp/">Home</a>
			<a class="current">ページが見つかりません</a>
		</nav>

		<article id="content" style="margin-bottom: 3em">
			<h1>ページが見つかりません</h1>
			<p>お探しのページは見つかりませんでした。</p>
		</article>

		<p><a href="/wp" class="button expand small">トップページ</a></p>
	</div>
	<?php get_sidebar(); ?>
</div>

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

不正な url を入れたら出てきます。　ｈｔｔｐ：／／ｌｏｃａｌｈｏｓｔ／ｗｐ／６６６

お知らせページをアーカイブページにする

右上の「お知らせ」をブログの一覧ページにしてみましょう。

左のメニューから設定＞＞表示設定を選択し、固定ページを変更します。

フロントページを固定ページにして、投稿ページをお知らせにします。

そして、いままで空だった index.php を改造します。

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

<div class="row">
	<div class="large-9 columns">
		<nav class="breadcrumbs">
			<a href="/wp/">Home</a>
			<a class="current">お知らせ</a>
		</nav>
		<h1>お知らせ</h1>
		<section class="list">
			<?php while( have_posts()): ?>
			<?php the_post(); ?>
			<a class="article" href="<?php the_permalink(); ?>">
				<article>
					<h2><?php the_title(); ?> - <small><?php
the_time('Y.m.d'); ?></small></h2>
					<?php the_excerpt(); ?>
				</article>
			</a>

			<hr />
			<?php endwhile; ?>
		</section>
		<p><a href="/wp" class="button expand small">トップページ</a></p>
	</div>
	<?php get_sidebar(); ?>
</div>

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

これで、ブログに書いた一覧が出てきます。

こんな風に、自分のコーディングでブログが出来上がって行きます。

これですべてではありませんが、このように自分なりのサイトを作っていくことができる wordpress は自由度が高いと言えます。

あなたもチャレンジしてみましょう。

参考文献

「WordPress をちゃんと使うための教科書」作者：たにぐち　まこと
出版社：マイナビ　定価：¥2,800+税

forever 朝活　php 講座

2014 年2月 26日　初版発行

著作/制作：株式会社フォーエバー

〒890-0053　鹿児島市中央町 22-16 アエールプラザ 4F

TEL:099-296-9118　FAX：099-250-2333　http://www.forever.co.jp

●本書は、構成・文書・プログラム・画像・データなどのすべてにおいて、著作権上の保護を受けています。

本書の一部あるいは全部について、いかなる方法においても複写・複製など、著作権法上で規定された権利を侵害する行為を行うことは禁じられています。