

# WordPress

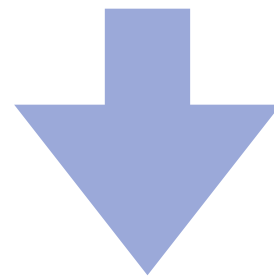
## Bootstrapサイト2

### Ver1

# この章での問題点

テンプレートの共通部分はパーツとして分割して使用することには成功した。

けれども投稿した記事が全く見れない状態。



テンプレートタグを使用して表示させる

# WordPressループの基本構造

```
<?php  
  
if(have_posts()):  
  
    while(have_posts()):the_post();  
  
        //ここに繰り返し表示のコード  
  
    endwhile;  
  
else:  
  
    //表示する記事がない場合のコード  
  
endif;  
  
?>
```

# テンプレートタグ

記事のループ処理内で使用するのがポイント

タグ	意味
<code>&lt;?php the_content(); ?&gt;</code>	投稿内容を表示させる
<code>&lt;?php the_excerpt(); ?&gt;</code>	記事の抜粋110文字
<code>&lt;?php the_title(); ?&gt;</code>	記事のタイトル
<code>&lt;?php the_permalink(); ?&gt;</code>	記事のURL
<code>&lt;?php the_date(); ?&gt;</code>	記事の投稿日時（投稿日が重複してる場合は最初の1記事のみ）
<code>&lt;?php the_time(); ?&gt;</code>	記事の投稿日時
<code>&lt;?php the_author(); ?&gt;</code>	記事の投稿者名
<code>&lt;?php the_author_posts_link(); ?&gt;</code>	記事の投稿者名とリンクを出力
<code>&lt;?php the_category(); ?&gt;</code>	記事のカテゴリー名とリンクを出力
<code>&lt;?php the_tags(); ?&gt;</code>	記事に付けたタグ

# ページ編集内容の表示

1. page.phpに次の内容を記述しましょう。
2. コンテンツ部分に以下ソースを記述します。

```
<?php if (have_posts()):  
    while (have_posts()) : the_post(); ?>  
    <h2><?php the_title(); ?></h2>  
    <?php the_content();  
    endwhile;  
endif; ?>
```

# the\_title()について

the\_title()にはhtmlタグが付かないのでh1またはh2などを付ける

```
<?php if (have_posts()):  
    while (have_posts()) : the_post(); ?>  
        <h1><?php the_title(); ?></h1>  
        <?php the_content(); ?>  
    <?php endwhile; endif; ?>
```

# テンプレートの役割

HOMEページ→front-page.php

投稿ページ→single.php

固定ページ→page.php

カテゴリページ→archive.php

# 各テンプレートの作成

front-page.php

index.phpとは別に、index.phpをコピーして  
front-page.phpにリネーム

single.php

single.htmlをコピーしてsingle.phpにリネーム



# フロントページについて

TOPページはWordPressのダッシュボードの「設定」>「表示設定」のフロントページの表示で設定します。

- デフォルトの状態では「最新の投稿」がフロントページ
- 投稿ページをフロントページにした場合のテンプレートが  
home.php
- フロントページを固定ページに指定した場合はその固定ページに  
紐付けしたテンプレート 例business.php
- フロントページの表示設定にかかわらずフロントページとして表示するにはfront-page.php

# 投稿ページ用テンプレート

投稿用single.phpは固定ページ用のテンプレート  
page.phpの内容をコピーして使用。

\* contents部分のモジュール化（別ファイル化）で汎  
用性を高めることもできますが、今回は行いません。  
余裕のある人はトライしてください。

# single.phpとpage.phpを確認

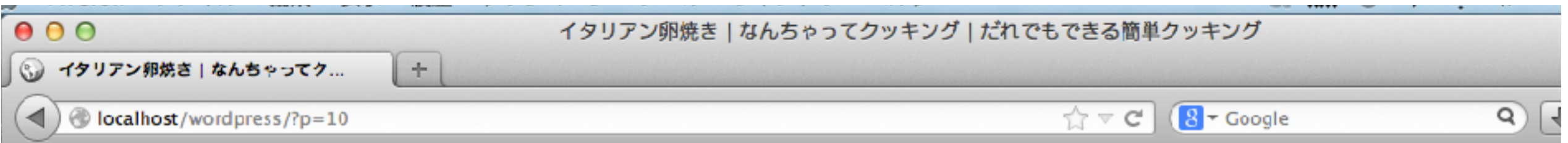
投稿ページの表示と固定ページの表示を確認しましょう。

# bodyのclass名を確認

bodyにWordPressが付与したclass名が新たに作成されるので影響のあるCSSを変更する。

HTMLはBootstrapを使用して簡単にデザインしています。さらにデザインを作りこみたい場合に活用できます。

# titleを複雑に設定



head内のtitleの内容をページの遷移と同時に変更したい。

```
<title><?php
    wp_title( ' | ', true, 'right' );
    bloginfo( 'name' );
    echo ' | ';
    bloginfo( 'description' );
    global $page, $paged;
    if ( $paged >= 2 || $page >= 2 ) {
        echo ' | ' . max( $paged, $page ) . 'ページ';
    }
?></title>
```

以下余力のある人はトライ！

# 前後記事へのリンク

前後の記事をリンクしたい場合は以下の  
コードをループ内に追加します。  
右のサンプルはCSSの例

```
<div class="pagenav">
  <span class="old">
    <?php previous_post_link('%link' ); ?>
  </span>
  <span class="new">
    <?php next_post_link('%link' ); ?>
  </span>
</div>
```

## CSS

```
/* 前後の記事へのリンク */
.pagenav a {
    padding: 5px 10px;
    border: solid 1px #cccccc;
    border-radius: 10px;
    color: #666666;
    font-size: 12px;
    text-decoration: none
}
.pagenav .old a {
    float: left
}
.pagenav .new a {
    float: right
}
.pagenav {
    overflow: hidden;
    margin-top: 40px;
    margin-bottom: 40px;
    clear: both
}
```

# content.phpの作成

articleで囲まれたコンテンツそのものを別ファイルのパーツ化にすることもできます。コンテンツの種類に応じて細かくデザインを変更したい場合などに有効です。

## content.php

```
<article>  
    <h2><?php the_title(); ?></h2>  
    <?php the_content(); ?>  
</article>
```



# content.php読み込み

別ファイルcontent.phpを読み込む方法

```
<?php
    if (have_posts()):
        while(have_posts()):the_post();
            get_template_part('content');
        endwhile;
    endif; ?>
```

```
get template part($slug, $name);
```

```
get template part($slug, $name);
```

具体例

```
<?php get_template_part( 'content','aside' ); ?>
```

↑ 読み込み

content-aside.php

```
<?php get_template_part( 'content','main' ); ?>
```

↑ 読み込み

content-main.php

# 投稿フォーマット使用

functions.phpに記述

```
add_theme_support( 'post-formats', array( 'aside', 'chat',  
'gallery', 'image', 'link', 'quote', 'status', 'video', 'audio' ) );
```

# get\_template\_part(\$slug, \$name); 詳細説明

get\_template\_part(\$slug, \$name);

\$slug : テンプレート名のスラッグ名

\$name : 特定テンプレートの名前

```
<?php get_template_part( 'content', get_post_format() ); ?>
```

get\_post\_format() で投稿記事の投稿フォーマットを取得する。

aside, chat ,gallery, link ,image, quote, status, video ,audioの  
いずれかの投稿フォーマットが第2引数として入る例えば投稿フォーマットが「aside」だった場合には、

```
<?php get_template_part( 'content','aside' ); ?>
```

を指定していることと同義なので、content-aside.phpが読み込まれる。