

- ✓ SEO и реклама
- ✓ CMS и фреймворки
- ✓ Сайтостроение

ЗАЯВИ О СЕБЕ В ИНТЕРНЕТЕ



ZYUBIN.RU

С ХОРОШИМ САЙТОМ - КЛИЕНТОВ БОЛЬШЕ



Сайтостроение

Фреймворки

CMS

Интернет-маркетинг

Карта сайта

Главная > CMS > WordPress

Создание собственной темы WordPress

Здравствуйте, друзья! Сегодня я расскажу о принципах и технических моментах создания темы для WordPress с нуля. Эта статья написана для тех, кто имеет какой-то опыт работы с движком WordPress и, хотя бы на базовом уровне, разбирается в HTML и CSS.

Темы разрабатываются под разные задачи: интернет-магазин, фото или видео сайт и т.д., мы же рассмотрим обычную тему блога и алгоритм действий по ее созданию.

Если вы решили создать собственную уникальную тему оформления, то перед созданием темы нужно продумать дизайн и сверстать HTML шаблон сайта или заказать у фрилансеров, решать вам. В общем, у вас должен получиться сверстаный шаблон, например, вот с таким содержимым: файлы `index.html` и `style.css`, папка картинок `images` и папка `js` со скриптами. Из этого шаблона мы и будем делать тему оформления.

Если же у вас пока нет собственного HTML шаблона для темы, то можете потренироваться, взять решение из данной статьи и набросать стили под себя. Для примера я возьму вот такой простенький HTML код главной страницы с разметкой шапки сайта, одного анонса поста, правой колонки и футера:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Мой блог</title>
  <link href="style.css" rel="stylesheet" type="text/css" />
</head>
<body>
  <div id="header">
    <div id="logo">
      <a href="index.html"></a>
    </div>
    <ul id="menu">
      <li><a href="#">Главная</a></li>
      <li><a href="#">Контакты</a></li>
      <li><a href="#">Рубрика 1</a></li>
      <li><a href="#">Рубрика 2</a></li>
    </ul>
  </div>
```

Для создания темы оформления необходимо знать функции или как их еще называют — теги WordPress. Мы разберем основные функции темы, со всеми тегами вы можете ознакомиться на официальном сайте codex.wordpress.org.

Структура темы WordPress

В папке `themes` создаем папку и называем ее, например, `mytheme`. Затем в нее переносим наш `style.css` и создаем необходимые файлы.

Создаем шаблоны блоков, которые отображаются на всех страницах:

- ✓ `header.php` — шапка вашего сайта;
- ✓ `sidebar.php` — боковая колонка с различными виджетами;
- ✓ `footer.php` — подвал сайта.

Шаблоны контента:

Поиск ...

Профиль

[Авторизация](#) | [Регистрация](#)

Подписка на новости



✓ Подписаться

Все рубрики блога

Сайтостроение

- WEB-разработка
- WEB-инфо

Фреймворки

- Bootstrap
- Yii

CMS

- WordPress
- OpenCart

Интернет-маркетинг

- Арбитраж трафика
- SEO-оптимизация



Свежие записи



Реализация тонких контроллеров в Yii2
09.08.2018 нет



Axios: Ajax-запросы

- `index.php` – шаблон главной страницы, выводятся короткие анонсы статей;
- `single.php` – шаблон постов (статей);
- `page.php` – шаблон статических страниц;
- `category.php` – шаблон категорий;
- `search.php` – шаблон выдачи результатов поиска;
- `404.php` – шаблон сообщения о несуществующей странице;
- `comments.php` – шаблон комментариев.

Дополнительные файлы:

- `functions.php` – файл дополнительного функционала и изменения стандартного функционала движка WordPress;
- `searchform.php` – код формы поиска;
- `screenshot.png` – Изображение 880 x 660, превью отображается в админке во вкладке «ТЕМЫ».

Не обязательные шаблоны (по моему мнению):

- `author.php` – шаблон страницы автора;
- `archive.php` – шаблон архивов;
- `date.php` – шаблон вывода дат;
- `tag.php` – шаблон меток;
- `attachment.php` – шаблон страниц вложения.

Лично я редко использую пять последних шаблонов, в основном делаю перенаправление на «главную» или на «404 ошибку» при запросе данных шаблонов. Информацию об авторе можно оформить на основе `page.php`, а остальные четыре только плодят дублированный контент на вашем сайте, организацию архивов можно реализовать намного продуктивнее.

Начинаем с файла `style.css`, делаем отступ и в самом верху вставляем закомментированный текст с названием и описанием темы.

```

/*
Theme Name: Название темы
Theme URI: URL адрес, если есть отдельный сайт о теме
Description: Описание темы оформления
Version: Версия, например (1.0)
Author: Ваше имя
Author URI: URL адрес автора темы
*/

```

Файл стилей придется редактировать при подключении функционала WordPress, который добавляет дополнительные участки кода.

Формируем header.php

Из HTML файла переносим сюда все, что находится в `<head>` и элементы в `<body>`, которые относятся к шапке сайта. У меня в шапке большое изображение с информацией о сайте и меню.

Код был таким.

```

<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Мой блог</title>
  <link href="style.css" rel="stylesheet" type="text/css" />
</head>

```

без jQuery
AXIOS 17.07.2018 1

Расширение корзина для Yii2
Shopping Cart 07.06.2018 нет

Всплывающая форма обратной связи
07.06.2018 28

Форма обратной связи для сайта
07.06.2018 130

Макхост
ПРЕМИУМ ХОСТИНГ
ДЛЯ ЛЮБЫХ САЙТОВ
НОВЕЙШЕЕ SSD ОБОРУДОВАНИЕ
ПРЕМИАЛЬНАЯ ПОДДЕРЖКА 24x7
НЕ ЭКОНОМЬ на хостинге
ЗАКАЗАТЬ

Метки записей

- #PHP
- #Yii2
- #Ajax
- #Bootstrap 4
- #Admin
- #Frontend
- #JavaScript
- #SEO
- #Advert
- #WP
- #Bootstrap 3
- #Backend

Мои услуги

Разработка CMS



● ○ ○ ○

Подробнее

```
<body>
<div id="header">
<div id="logo">
  <a href="index.html"></a>
</div>
<ul id="menu">
<li><a href="#">Главная</a></li>
<li><a href="#">Контакты</a></li>
<li><a href="#">Рубрика 1</a></li>
<li><a href="#">Рубрика 2</a></li>
</ul>
</div>
```

После переноса и редактирования стал таким.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="<?php bloginfo( 'charset' ); ?>" />
  <title><?php echo wp_get_document_title(); ?></title>
  <link href="<?php bloginfo( 'stylesheet_url' ); ?>" rel="stylesheet" type="text/css" />
  <?php wp_head(); ?>
</head>
<body>
<div id="logo">
  <a href="<?php echo home_url(); ?>">
<div class="content">
<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
<div class="post-content">
<div class="post-img"><a href="<?php the_permalink(); ?>"><?php the_post_thumbnail('full')
<h2 class="post-title"><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h2>
<div class="post-text"><?php the_excerpt(); ?></div>
<div class="readmore"><a href="<?php the_permalink(); ?>">Читать далее</a></div>
</div>
<?php endwhile; ?>
<?php else: ?>
<?php endif; ?>
</div>

```

- `the_title();` – выводит название поста;
- `the_permalink();` – выводит ссылку на отображаемый пост;
- `the_excerpt();` – выводит установленное количество символов поста со вставкой [...];
- `the_post_thumbnail('full');` – выводит миниатюру поста, а 'full' задает полный размер;
- `get_header();` – подключение файла `header.php`;
- `get_sidebar();` – подключение файла `sidebar.php`;
- `get_footer();` – подключение файла `footer.php`.



Важно. Весь наш код и функции входят в так называемый цикл WordPress, который и позволяет нам выводить несколько таких анонсов статей на одной странице, написав при этом код всего для одного анонса. Некоторые функции корректно работают только в цикле.

Вот как выглядит чистый цикл без нашего кода.

```

<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?><!-- Начало цикла -->
<!-- Весь код анонса поста -->
<?php endwhile; ?><!-- Окончание цикла -->
<!-- Постраничная навигация -->
<?php else: ?>
<!-- Альтернативный код если нечего не найдено -->
<?php endif; ?>

```

Все расписано, думаю с этим понятно.

Прописываем функции в `functions.php`

Открываем файл и первым делом пишем конструкцию PHP, а затем в нее добавляем следующие функции.

```

<?php
/**
 * Поддержка миниатюр
 */
add_theme_support( 'post-thumbnails' );

/**
 * Подключение меню
 */
register_nav_menus( array(
    'main-menu' => __( 'main-menu' )
));

/**
 * Подключение сайдбара
 */
register_sidebar( array(
    'name' => __( 'Sidebar', 'mytheme' ),
    'id' => 'sidebar',

```

```
'description' => 'Правая колонка сайта.'
```

Заполняем файлы sidebar.php и footer.php

В файл `sidebar.php` прописываем:

```
<div class="sidebar-right">
  <?php if ( is_active_sidebar( 'sidebar' ) ) : ?>
    <ul id="sidebar">
      <?php dynamic_sidebar( 'sidebar' ); ?>
    </ul>
  <?php endif; ?>
</div>
```

- `is_active_sidebar();` – проверяет, установлен хотя бы один виджет в нашей панели `'sidebar'`;
- `dynamic_sidebar();` – выводит активную панель виджетов.

В файл `footer.php` вставляем нужный вам код, например, параграф с копирайтом, а в конце перед закрывающим тегом `</body>` вставляем `wp_footer();`. Как и `wp_head();` это обязательный хук-действие темы.

```
<!-- Нужный вам код --->
<?php wp_footer(); ?>
</body>
</html>
```

После этих настроек главная страница уже готова. Дальнейшие действия будут гораздо проще, просто будем брать уже готовый код, вставлять его в другие шаблоны и редактировать.

Формируем page.php и single.php

Копируем код из индексного файла и вставляем в `page.php`, убираем лишнее и в блоке с классом `.post-text` изменяем вывод анонса на вывод всего контента страницы функцией `the_content();`.

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

<div class="content-main">

  <div class="content">
    <div id="page">

      <?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
        <h1 class="post-title"><?php the_title(); ?></h1>
        <div class="post-text"><?php the_content(); ?></div>
      <?php endwhile; ?>

      <?php else: ?>

      <?php endif; ?>
    </div>
  </div>

  <?php get_sidebar(); ?>

</div>
```

В `single.php` получается тот же код, что и в шаблоне страниц, добавляем только подключение комментариев `comments_template();` из файла `comments.php`.

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

<div class="content-main">

  <div class="content">
    <div id="page">

      <?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
        <h1 class="post-title"><?php the_title(); ?></h1>
        <div class="post-text"><?php the_content(); ?></div>
        <?php comments_template(); ?>
      <?php endwhile; ?>

      <?php else: ?>

      <?php endif; ?>
    </div>
  </div>

  <?php get_sidebar(); ?>

</div>
```

```
<?php else: ?>
    <?php endif; ?>
</div>
</div>

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

При желании можете добавить:

- `the_time('j.m.Y');` – выводит дату публикации поста (число, месяц, год);
- `the_author();` – выводит автора статьи;
- `the_category();` – выводит рубрику, в которую входит данный пост.

Шаблон комментариев довольно сложная задача для начинающих, поэтому я рекомендую взять готовое решение из стандартной темы WordPress в файлах `comments.php` и `functions.php` и стилизовать внешний вид на ваш вкус.

Заполняем оставшиеся шаблоны `404.php`, `category.php` и `search.php`

В `404.php` копируем содержимое `page.php`, удаляем лишнее, прописываем уведомление о том, что страницы не существует, можно подкрепить текст какой-нибудь картинкой.

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

<div class="content-main">

    <div class="content">
        <div id="page">

            <?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
                <h1 class="page-title">Такой страницы не существует!</h1>
                
                <?php endwhile; ?>

            <?php else: ?>

            <?php endif; ?>
        </div>
    </div>

    <?php get_sidebar(); ?>

</div>
```

В `category.php` просто копируем содержимое файла `index.php` и добавляем вывод заголовка категорий `single_cat_title('Рубрика: ')`.

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

<div class="content-main">

    <div class="content">
        <div id="category">
            <h1><?php single_cat_title('Рубрика: ');?></h1>
        </div>

        <?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
            <div class="post-content">
                <div class="post-img"><a href="<?php the_permalink(); ?>"><?php the_post_thumbnail('full')
                <h2 class="post-title"><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h2>
                <div class="post-text"><?php the_excerpt(); ?></div>
                <div class="readmore"><a href="<?php the_permalink(); ?>">Читать далее</a></div>
            </div>
            <?php endwhile; ?>

        <?php else: ?>


```

В `search.php` аналогично копируем из индексного файла код и после цикла WordPress, там, где должен быть альтернативный код, о котором я говорил выше, вписываем сообщение о результатах поиска.

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

```
<div class="content-main">

<div class="content">

<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
<div class="post-content">
  <div class="post-img"><a href="<?php the_permalink(); ?>"><?php the_post_thumbnail('full')
  <h2 class="post-title"><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h2>
  <div class="post-text"><?php the_excerpt(); ?></div>
  <div class="readmore"><a href="<?php the_permalink(); ?>">Читать далее</a></div>
</div>
<?php endwhile; ?>

<?php else: // Если ничего не найдено ?>
<div id="page">
  <h1 class="post-title">По запросу ни чего не найдено!</h1>
  <h4 class="post-text">Попробуйте еще раз с другими ключевыми словами.</h4>
</div>
</div>
```

Код формы поиска размещаем в файле `searchform.php`. Берем стандартную форму и редактируем, в `input` добавляем параметр «s» для `name` и `id`, используем метод «GET» для отправки запроса на главную страницу `home_url('/')` и очищаем URL функцией `esc_url()`.

```
<form role="search" method="get" id="searchform" class="search-form" action="<?php echo esc_url(
<div class="widget-search">
  <input type="search" class="search-field" value="" placeholder="Искать..." name="s" id="s" cl
  <button type="submit" class="search-submit" >Поиск</button>
</div>
</form>
```

Важно. После всех проделанных действий, нужно будет лезть в код и стилизовать все новые элементы, которые выдаст вам WordPress! Для этих целей я использую расширение «Firebug» для Mozilla Firefox, это расширение облегчает и ускоряет решение таких задач.

На этом создание простенькой блоговой темы оформления WordPress закончено, постарался объяснить все по пунктам и как можно короче, но получилась внушительная статья. Если остались вопросы, задавайте в комментариях. Рекомендую статью: «[Основные функции движка WordPress](#)».

© 07.05.2016 WordPress Backend, Frontend, WP

Похожие записи:



Добавлено комментариев - 9

Илья 01.03.2018 21:49
А по организации архивов новостей можно где то почтить все же `is_koprino@mail.ru`?

Ответить ↩

Евгений 01.03.2018 22:41
Погуглите, думаю должно быть в интернете. Организация в принципе такая же, как и у статей, просто используете функции для работы с архивами: `the_archive_title()`, `the_archive_description()`, `the_archive_description()` и т.д. Читайте описание функций в документации и делайте как вам нужно.

Ответить ↩



Паша 14.11.2018 14:09

В файле header нужно вместо

```
<title><?php wp_get_document_title(); ?></title>
```

написать:

```
<title><?php echo wp_get_document_title(); ?></title>
```

Иначе отобразится просто url страницы

Ответить ↩



Евгений 14.11.2018 21:50

Спасибо, исправил.

Ответить ↩



Александр 29.11.2018 16:59

Ух ты, а как вы сделали, чтобы комментариев можно было писать с использованием мини-редактора?

Ответить ↩



Евгений 01.12.2018 02:40

Итегрировал редактор))

Ответить ↩



Serega! 05.01.2019 15:32

Полезная статья. Круто!!!

Ответить ↩



Евгений 20.01.2019 22:34

Привет!

Подскажите пожалуйста, как решить эту проблему?

Я переносу сверстанный шаблон в wordpress. Но не получается сделать динамический контент в статичных блоках.

В общем я у меня такой код

```
<div class="bg-one align-items-center d-flex" style="">
  <div class="container mt-5 mb-5">
    <div class="row mt-5">
      <div class="col-md-12" style="">
        <div class="row">
          <div class="col-md-5">
            <div class="row">
              <div class="col-md-12"></div>
            </div>
          </div>
        </div>
      <div class="post-box col-md-7 p-4 pb-5" style="">
        <a class="post-title" href="<?php the_permalink(); ?>">
          <h2 class="post-title display-3 text-primary mx-4 mt-4 ml">
        </a>
        <div class="post-info lead text-info m-4 mb-4"><?php the_exce
        <a class="link-read mx-4" href="<?php the_permalink(); ?>">
          ДАЛЕЕ <img src="<?php echo get_stylesheet_directory_uni()
        </a>
      </div>
    </div>
  </div>
</div>
```

Один блок справа, другой слева отображается. Но проблема в том, что я получаю один и тот же заголовок, описание и ссылку на статью в обоих блоках. Как вывести разные, например 2 последних поста с блога? Решить циклом не получается, так как суть данных блоков теряется, всего их будет 4 и каждый будет иметь уникальный дизайн.



Евгений 21.01.2019 10:01

Ответить ↩

Здравствуйте. Если вы не используете цикл WP, то не используйте функции работающие с ним: `the_permalink()`, `the_title()` и т.д. Делайте запрос в базу получайте сколько вам нужно записей и просто выводите их в той верстке в какой вам нужно, делайте это просто средствами PHP без использования WP плюшек и все!

Ответить ↩

Добавить комментарий

Ваше имя:

Комментарий:

B I U |

Нажимая на кнопку «Добавить», я даю согласие на обработку своих персональных данных в соответствии с [политикой конфиденциальности](#)

Добавить

Я в социальных сетях



Главная страница сайта



Идеи и предложения

Есть интересные идеи или предложения, напишите мне!

Написать