

Как сверстать тему для WordPress

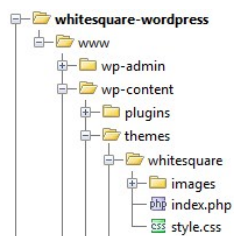
WordPress
Разработка веб-сайтов
Войти Регистрация
Tutorial

Введение

В предыдущих статьях мы рассмотрели принципы верстки на чистом CSS и с помощью Bootstrap. Сегодняшней статьей мы начинаем рассматривать особенности верстки под популярные CMS. И начнем с WordPress, как самой популярной из них. Будем считать, что WordPress у вас уже установлен и перейдем непосредственно к созданию шаблона, в качестве которого у нас по-прежнему будет выступать Corporate Blue от студии Pcklaboratory. Если вы не знаете как установить WordPress, то инструкцию можно найти [здесь](#). В данном руководстве мы не будем подробно описывать CSS стили и HTML код – это было сделано уже в предыдущих статьях. Вместо этого рассмотрим детально особенности создания темы именно под WordPress.

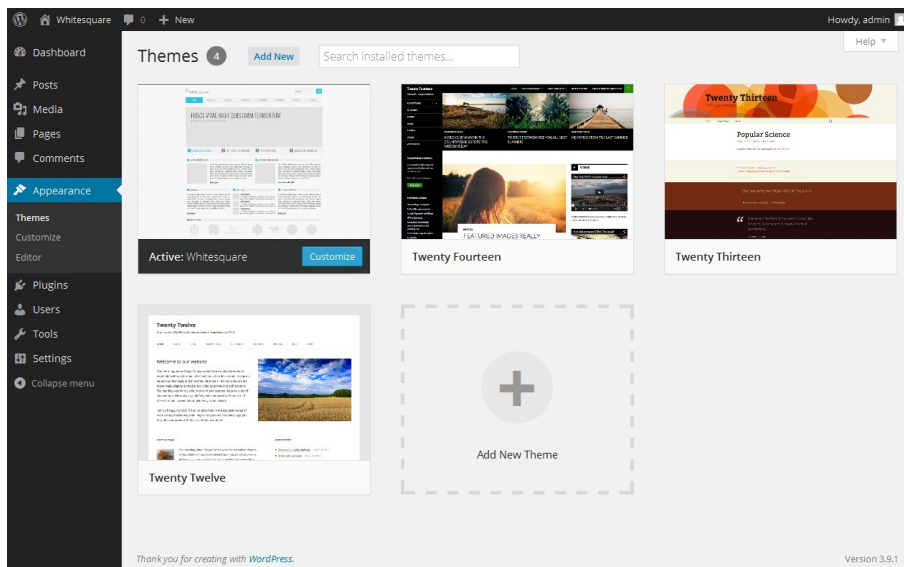
Добавление темы

Для начала в папке themes создадим папку нашего шаблона «whitesquare». В ней будет находиться папка images и два необходимых пустых файла index.php и style.css.



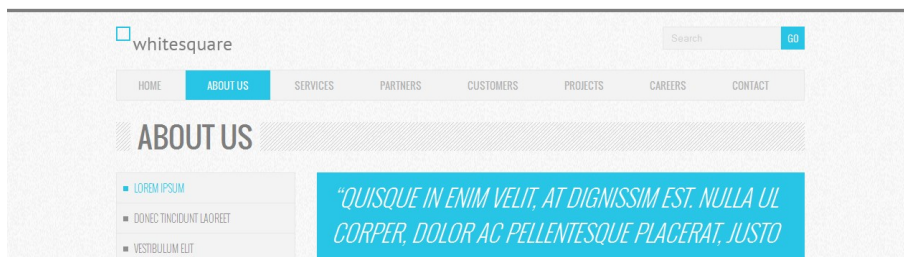
Следующим шагом нужно добавить скриншот нашей темы. Сохраните изображение главной страницы из psd макета размером 880x660 в папку темы whitesquare с именем screenshot.png.

После этого мы уже можем использовать нашу тему. Зайдите в панель администратора по адресу <http://{site-name}/wp-admin>, а затем в управление темами (Appearance -> Themes). В списке уже должна появиться наша тема whitesquare. Наведитесь на тему и нажмите «Activate».



После того, как тема активируется, вы увидите сообщение со ссылкой на сайт. Уже сейчас можно зайти на него и увидеть пустую страницу.

Предварительный осмотр



ЧИТАЮТ СЕЙЧАС

Что не так с Raspberry Pi

50,6k 104

Безумие dotфайлов

16k 48

Корпоративный туалет

10,2k 37

Сколько доменных имён .com не используется?

16,2k 22

Электромобили утонул на дно автокорпорации?

15,4k 138

InterNyet — как в Советском Союзе изобрели интернет и почему он не заработал

54,6k 184



Если посмотреть на макет, то можно увидеть, что наша страница состоит из шапки и футера, которые повторяются на всех страницах. Также на всех страницах, кроме главной, есть сайдбар слева. Для того чтобы руководство было более универсальным, давайте страницы главного меню оформим как «страницы WordPress» (page), а страницы подмену как «посты блога WordPress» (post) с комментариями. Главную же страницу сделаем как отдельную страницу (front-page) с собственной разметкой. Здесь надо отметить, что, несмотря на то, что страницы WordPress могут быть реализованы двумя способами (как страницы или как посты) в базе данных они различаются только типом, однако при создании шаблонов они обрабатываются немного по-разному. В этом вы сможете убедиться чуть ниже.

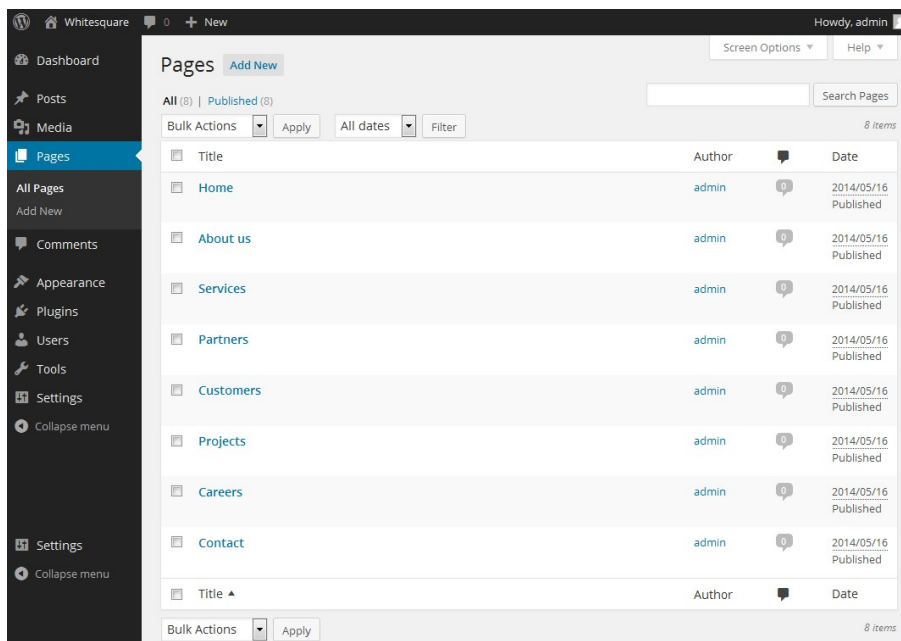
Структура страниц

Большинство руководств по созданию тем для WordPress ограничивается созданием макета и стилей для него. Мы же опишем весь порядок действий по созданию полноценного сайта на WordPress в рамках предоставленных макетов.

Давайте начнем со структуры страниц. Прежде чем добавлять страницы, нужно сначала зайти в настройки и указать стиль ссылок: Settings -> Permalink settings -> Post name. Это позволит сделать ссылки в виде «http://site_name/page_name».

Добавление страниц осуществляется через панель администратора: Pages -> Add new. Для каждой страницы в области Page Attributes в поле Order укажите цифрой порядок страницы в главном меню.

После добавления, список страниц должен выглядеть вот так:



Когда все страницы будут добавлены, нужно указать, что страница Home будет главной. Для этого зайдите

Header.php и Footer.php

Шапка в терминологии WordPress, это не только визуальная шапка на макете сайта. По сути, она содержит весь общий код, который встречается в начале всех страниц сайта. Давайте создадим файл header.php в папке нашего шаблона и наполним его содержимым.

```
<!doctype html>
<html>
<head>
    <meta http-equiv="Content-type" content="text/html; charset=<?php bloginfo('charset'
); ?>">
    <meta http-equiv="X-UA-Compatible" content="IE=Edge">
    <title><?php wp_title('«', true, 'right'); ?> <?php bloginfo('name'); ?></title>
    <link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />
    <?php wp_head(); ?>
</head>
<body <?php body_class(); ?>>
    <div class="wrapper">
```

Внутри тега head мы установили кодировку, указанную в WordPress, заголовок страницы и pingback (для связи с другими сайтами). В последней строке вызываем команду wp_head(), которая добавляет заголовки WordPress. Также открываем блок «wrapper».

Кроме этого, нам нужно подключить css и js файлы. В текущих версиях wordpress это делается не прямым текстом в header.php, а через подключение в специальных функциях. Для этого откройте файл functions.php внутри нашей темы и добавьте в него следующий код:

```
function enqueue_styles() {
    wp_enqueue_style( 'whitesquare-style', get_stylesheet_uri() );
    wp_register_style( 'font-style', 'http://fonts.googleapis.com/css?family=Oswald:400,300' );
    wp_enqueue_style( 'font-style' );
}
add_action( 'wp_enqueue_scripts', 'enqueue_styles' );

function enqueue_scripts () {
    wp_register_script( 'html5-shim', 'http://html5shim.googlecode.com/svn/trunk/html5.js' );
    wp_enqueue_script( 'html5-shim' );
}
add_action( 'wp_enqueue_scripts', 'enqueue_scripts' );
```

В функции enqueue_styles мы зарегистрировали и подключили нужные стили, а затем указали вордпрессу, что эта функция является подключением стилей. Аналогично и для js файла, который требуется для отображения html5 тегов в старых браузерах.

Футер аналогично шапке — содержит общий код, который встречается в конце всех страниц сайта. Давайте запишем его содержимое в файл footer.php.

```
</div>
<footer></footer>
<?php wp_footer(); ?>
</body>
</html>
```

Здесь мы закрываем открытые блоки и вызываем wp_footer(), чтобы добавить скрипты футера WordPress.

Шаблон страницы

Следующим шагом, нам нужно сделать шаблон обычной страницы WordPress. Создайте в папке темы файл page.php и добавьте в него следующий код:

```
<?php get_header(); ?>
<div class="main-heading">
    <h1><?php the_title(); ?></h1>
</div>
<section>
    <?php if (have_posts()): while (have_posts()): the_post(); ?>
        <?php the_content(); ?>
    <?php endwhile; endif; ?>
</section>
<?php get_footer(); ?>
```

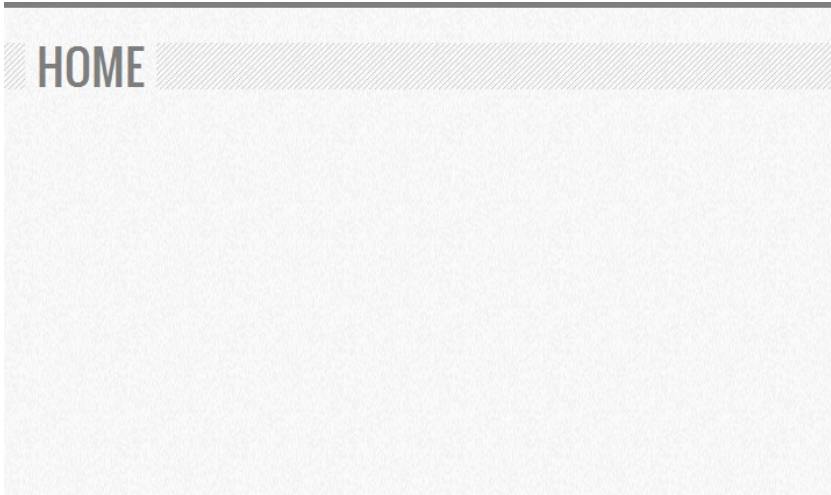
Здесь мы подключили наши файлы шапки и футера, создали блок названия страницы и в теге section вставили стандартный блок вывода контента страниц и постов.

Теперь нужно стилизовать получившуюся страницу. Сохраните фоны в файлы images/bg.png и images /h1-

bg.png. Далее добавьте немного базовых стилей в файл style.css:

[Показать код](#)

В результате должна получиться вот такая картина:



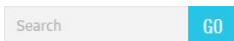
Логотип



После того, как мы закончили с каркасом основной страницы, приступим к ее наполнению. Начнем с логотипа. Сохраните изображение логотипа в images/logo.png. В шаблон шапки, в файле header.php вставьте разметку для логотипа:

```
<div class="wrapper">
  <header>
    <a href="/"></a>
  </header>
```

Форма поиска



WordPress позволяет достаточно гибко сохранять блоки кода в отдельные файлы, а затем использовать их в нескольких разных местах. Рассмотрим пример на нашей форме поиска.

В папке темы создайте новый файл searchform.php и сохраните в него код формы поиска:

```
<form name="search" action="<?php echo home_url( '/' ) ?>" method="get" class="search-form">
  <input type="text" value="<?php echo get_search_query() ?>" name="s" placeholder="<?php echo __('Search', 'whitesquare'); ?>" class="input">
  <button type="submit" class="button"><?php echo __('GO', 'whitesquare'); ?></button>
</form>
```

А в файл стилей запишите стили для формы:

[Показать код](#)

Всё, что осталось сделать – это подключить searchform.php внутри header.php.

```
<header>
  ...
  <?php get_search_form(); ?>
</header>
```

Навигация



Добавить навигацию на страницу можно разными способами, например через функцию wp_list_pages, но более современным способом будет добавление через админку. По умолчанию функционал добавления меню не активирован, для того, чтобы это сделать, добавьте в файл темы functions.php следующий код:

```
if (function_exists('add_theme_support')) {
```

```
add_theme_support('menus');
}
```

После этого, в панели администрирования в пункте меню Appearance появится подпункт Menus, в котором нужно нажать на ссылку Create new menu, ввести имя меню «top-menu», выделить страницы из левой колонки и добавить их в меню кнопкой Add to menu.

Теперь, когда меню создано, нужно его показать на страницах сайта. Для этого добавьте в конец header.php следующий код:

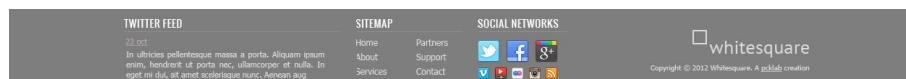
```
<nav class="main-navigation">
    <? wp_nav_menu(array('menu' => 'top-menu', 'menu_class' => 'top-menu')); ?>
</nav>
```

Функция wp_nav_menu отобразит меню с именем «top-menu» и css классом «top-menu».

После этого меню уже появится на страницах но без стилей. Стилизуем его:

► [Показать код](#)

Футер



Теперь приступаем к самой сложной части нашего шаблона. Давайте нарежем изображения и сохраним их в в папку images

images/footer-logo.png – логотип футера

images/social.png – спрайты больших иконок

images/social-small.png – спрайты маленьких иконок

Далее делаем вёрстку в файле footer.php:

```
<footer>
    <div class="footer-content">
        <div class="twitter">
            <h3 class="footer-heading"><?php echo __('Twitter feed', 'whitesquar
e'); ?></h3>
            <time datetime="2012-10-23"><a href="#" class="twitter-time">23 oct<
/a></time>
            <p><?php echo __('In ultricies pellentesque massa a porta. Aliquam i
psum enim, hendrerit ut porta nec, ullamcorper et nulla. In eget mi dui, sit amet scelerisque
e nunc. Aenean aug', 'whitesquare'); ?></p>
        </div>
        <div class="sitemap">
            <h3 class="footer-heading"><?php echo __('Sitemap', 'whitesquare');
?></h3>
            <div class="column first">
                <a href="/home/"><?php echo __('Home', 'whitesquare'); ?></a
>
                <a href="/about/"><?php echo __('About', 'whitesquare'); ?><
/a>
                <a href="/services/"><?php echo __('Services', 'whitesquare'
); ?></a>
            </div>
            <div class="column">
                <a href="/partners/"><?php echo __('Partners', 'whitesquare'
); ?></a>
                <a href="/customers/"><?php echo __('Support', 'whitesquare'
); ?></a>
                <a href="/contact/"><?php echo __('Contact', 'whitesquare');
?></a>
            </div>
        </div>
        <div class="social">
            <h3 class="footer-heading"><?php echo __('Social networks', 'whitesq
uare'); ?></h3>
            <a href="http://twitter.com/" class="social-icon twitter-icon"></a>
            <a href="http://facebook.com/" class="social-icon facebook-icon"></a
>
            <a href="http://plus.google.com/" class="social-icon google-plus-ico
n"></a>
            <a href="http://vimeo.com/" class="social-icon-small vimeo-icon"></a
>
            <a href="http://youtube.com/" class="social-icon-small youtube-icon"
></a>
            <a href="http://flickr.com/" class="social-icon-small flickr-icon"><
/a>
            <a href="http://instagram.com/" class="social-icon-small instagram-i
con"></a>
            <a href="/rss/" class="social-icon-small rss-icon"></a>
        </div>
```

```

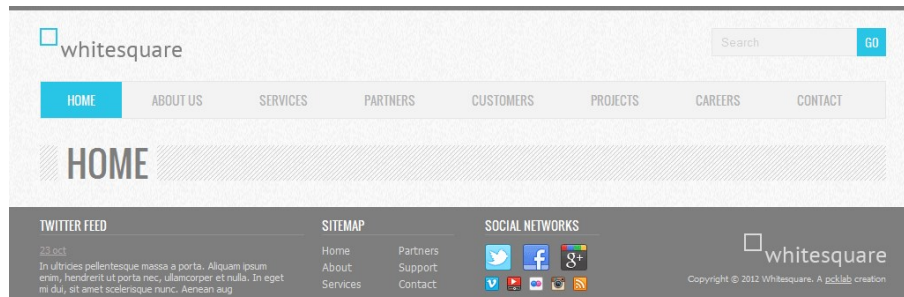
<div class="footer-logo">
  <a href="/"></a>
  <p><?php echo __('Copyright © 2012 Whitesquare. A <a href="http://pcklab.com">pcklab</a> creation', 'whitesquare'); ?></p>
</div>
</footer>

```

И прописываем стили в style.css:

► [Показать код](#)

В итоге главная страница сайта должна выглядеть вот так:



Главная страница

Если вы посмотрите на psd макеты, то увидите, что разметка главной страницы отличается от внутренних. В частности, на главной странице нет сайдбара и заголовка страницы.

WordPress позволяет задавать разные шаблоны для разных страниц. Такой шаблон должен храниться в файле `page-{название страницы}.php`. Если шаблон не найден, то будет подключаться шаблон по умолчанию `page.php`.

Для главной страницы создайте новый файл `front-page.php` в папке темы. Добавьте в него следующий код:

```

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

```

Он отличается от кода `page.php` только тем, что в нем нет заголовка страницы.

То, что мы будем делать дальше, уже выходит за рамки создания темы WordPress, однако мы попытаемся воссоздать полную картину из psd макета и наполним контентом главную страницу. Содержимым страница наполняется из панели администратора для того, чтобы владелец сайта мог зайти в эту панель и что-то изменить на странице, не изменяя код самой темы.

Прежде, чем добавить контент главной страницы – необходимо нарезать все изображения этой страницы и добавить их в библиотеку WordPress через меню Media -> Library. Назовём эти изображения так:

- home-1.png
- home-2.png
- home-3.png
- home-4.png
- home-5.png
- clients-1.png
- clients-2.png
- clients-3.png
- clients-4.png
- clients-5.png
- clients-6.png
- clients-7.png

Далее, перейдите в панели администратора к форме редактирования главной страницы Pages -> Home -> Edit Page и в поле text введите содержимое страницы:

► [Показать код](#)

Не забудьте поправить пути до картинок. Они зависят от даты добавления.

Теперь осталось стилизовать данный код.

► [Показать код](#)

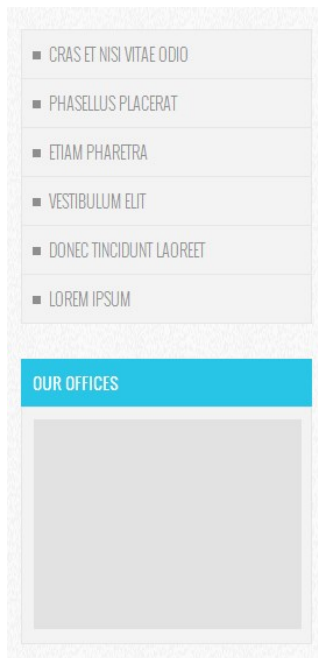
Если вы сейчас посмотрите на страницу, то увидите, что заголовки блоков разъехались. Это связано с тем, что редактор WordPress добавил пустые параграфы в наш код в местах перевода строк. Чтобы решить эту проблему, создайте в папке темы файл `functions.php` и поместите в него код:

```
<?php
remove_filter('the_content', 'wpautop');
```

После этого, главная страница должна отобразиться правильно.

Сайдбар

Теперь, когда главная страница готова, давайте вернемся к шаблону внутренних страниц и добавим в него сайдбар.



Для этого создайте файл `sidebar.php` и поместите в него следующий код:

```
<aside>
  <nav class="aside-navigation">
    <? wp_nav_menu(array('menu' => 'aside-menu', 'menu_class' => 'aside-menu'));
  <?
  </nav>
  <h2 class="sidebar-heading"><?php echo __('Our offices', 'whitesquare'); ?></h2>
  <div class="map">
    ">
  </div>
</aside>
```

Здесь верстка состоит из двух частей: отображение навигации подменю и блок карты офисов. Для полноты картины в качестве подменю предлагаем показать список всех постов на сайте. Подменю создается аналогично тому, как мы создавали меню. При создании подменю в разделе админки Appearance -> Menus создадим новое меню с именем `aside-menu`. Чтобы добавить в него посты, кликните наверху в Screen options и отметьте галочкой Show on screen: Posts. Блок карты особых вопросов не вызывает.

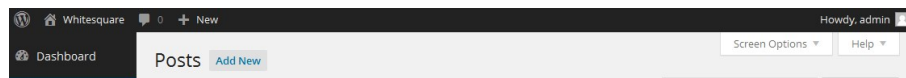
После того, как блок сайдбара готов, нужно его подключить для всех подстраниц в файле `page.php`:

```
<div class="main-heading">
  ...
</div>
<?php get_sidebar(); ?>
<section>
  ...
```

Далее нам нужно добавить стили для верстки:

► [Показать код](#)

Ну и в заключение, добавьте несколько постов в наш блог. Делается это в панели администратора на вкладке Posts -> Add New. Мы добавили точно такие же названия, как были в psd макете.



All (6) Published (6) Trash (3)						Search Posts
Bulk Actions	Apply	All dates	View all categories	Filter	6 items	
<input type="checkbox"/>	Title	Author	Categories	Tags	<input type="checkbox"/>	Date
<input type="checkbox"/>	Cras et nisi vitae odio	admin	Uncategorized	—	<input type="checkbox"/>	2014/06/08 Published
<input type="checkbox"/>	Phasellus placerat	admin	Uncategorized	—	<input type="checkbox"/>	2014/06/08 Published
<input type="checkbox"/>	Etiam pharetra	admin	Uncategorized	—	<input type="checkbox"/>	2014/06/08 Published
<input type="checkbox"/>	Vestibulum elit	admin	Uncategorized	—	<input type="checkbox"/>	2014/06/08 Published
<input type="checkbox"/>	Donec tincidunt laoreet	admin	Uncategorized	—	<input type="checkbox"/>	2014/06/08 Published
<input type="checkbox"/>	Lorem ipsum	admin	Uncategorized	—	<input type="checkbox"/>	2014/06/08 Published
<input type="checkbox"/>	Title	Author	Categories	Tags	<input type="checkbox"/>	Date
Bulk Actions		Apply	6 items			

Страница About us



Теперь, когда у нас готов шаблон для внутренних страниц, давайте наполним содержимым страницу About us. По той же схеме, как мы наполняли главную страницу – подготовьте изображения и добавьте их в медиатеку. Назовите изображения вот так:

- about-1.png
- about-2.png
- team-Nobriga.jpg
- team-Pittsley.jpg
- team-Rousselle.jpg
- team-Shoff.jpg
- team-Simser.jpg
- team-Tondrea.jpg
- team-Venuti.jpg
- team-Wollman.jpg

Далее перейдите в редактирование страницы в панели администратора и добавьте код:

► [Показать код](#)

И стили в style.css:

Шаблон поста

После предыдущих действий, на нашем сайте уже доступны все страницы, кроме страниц с постами. За отображение одиночных постов отвечает шаблон `single.php`. Создайте его со следующим содержимым:

```

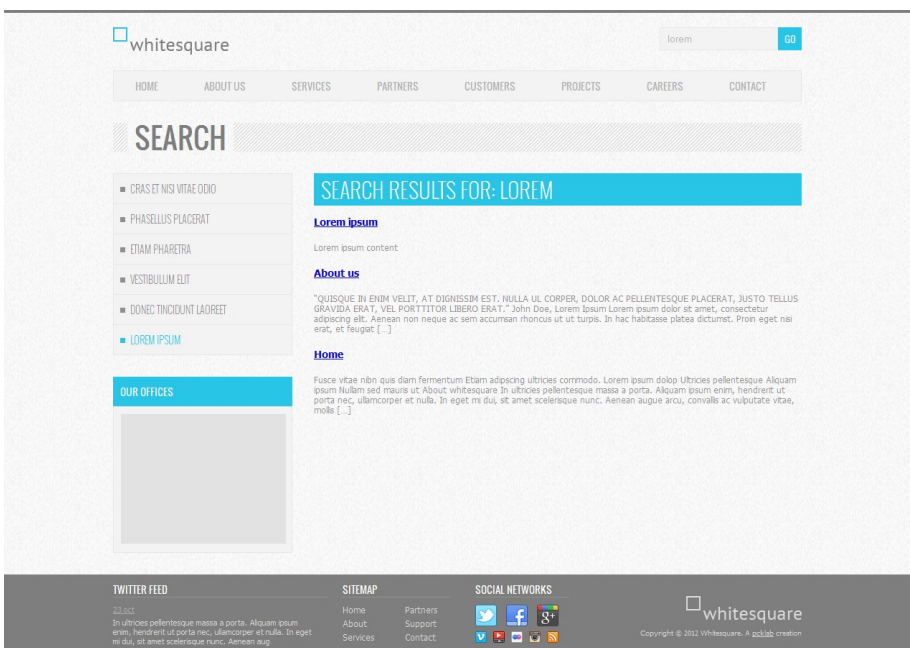
<?php get_header();?>
<div class="main-heading">
    <h1><?php the_title(); ?></h1>
</div>
<?php get_sidebar();?>
<section>
    <?php while (have_posts()): the_post();?>
        <?php the_content();?>
        <?php
            if ( comments_open() || get_comments_number() ) {
                comments_template();
            }

            ?>
        <?php endwhile; ?>
</section>
<?php get_footer(); ?>

```

Этот шаблон аналогичен предыдущим шаблонам с той лишь разницей, что здесь мы сначала отображаем контент поста, а затем комментарии к нему.

Страница поиска



В самом начале, когда мы создавали шаблон шапки мы добавили в него поиск, однако страницы для отображения результатов поиска у нас еще нет. Создайте в папке темы файл `search.php` и добавьте в него следующее:

```

<?php get_header(); ?>
<div class="main-heading">
    <h1>Search</h1>
</div>
<?php get_sidebar(); ?>
<section>
    <h2 class="content-heading"><?php printf( __('Search Results for: %s', 'default'), get_search_query() ); ?></h2>
    <?php if (have_posts()): while (have_posts()): the_post(); ?>
        <h3><a href="<?php the_permalink(); ?>"><?php the_title(); ?></a></h3>
        <p><?php the_excerpt(); ?></p>
    <?php endwhile; else:?>
        <p><?php echo __('Sorry, no results found', 'whitesquare'); ?></p>
    <?php endif; ?>
</section>
<?php get_footer(); ?>

```

От других шаблонов этот отличается тем, что мы выводим заголовки с текстом поиска, а затем список результатов поиска.

Страница архива

В WordPress помимо одиночных постов существуют страницы, на которых отображаются списки постов. Это могут быть категории, сортировка по дате, автору или по ключевым словам. В нашем макете такого

функционала нет, но добавить его обязательно нужно, так как мы создаем шаблон, который может использоваться с разным контентом. Для каждого из указанных списков, существуют свои шаблоны, однако если они не найдены, WordPress пытается найти общий файл archive.php. Мы этим воспользуемся и создадим его. Поскольку его содержимое ничем не будет отличаться от содержимого обычной страницы – просто скопируйте page.php в archive.php.

404

Последний шаблон, который осталось добавить — это шаблон для 404 страницы, когда WordPress не смог найти запрошенную страницу. Он будет точно такой же, как page.php только вместо вывода постов будет написано сообщение об ошибке:

```
<?php get_header(); ?>
<div class="main-heading">
  <h1><?php the_title(); ?></h1>
</div>
<?php get_sidebar(); ?>
<section>
  <p><?php echo __('It looks like nothing was found at this location.', 'whitesquare')
; ?></p>
</section>
<?php get_footer(); ?>
```

Заключение

На этом создание шаблона WordPress закончено. Готовый проект можно скачать [здесь](#).

Ссылки на предыдущие статьи:

[Как сверстать веб-страницу. Часть 1](#)

[Как сверстать веб-страницу. Часть 2 — Bootstrap](#)

[Верстка для самых маленьких. Верстаем страницу по БЭМу \(@xnim\)](#)

[Создаём шаблон 1С-Битрикс на базе Bootstrap вёрстки \(@lexnekr\)](#)

Update:

Статья была существенно обновлена в соответствии с пожеланиями комментаторов.

- Создание меню теперь описано не на основе wp_list_pages, а с помощью wp_nav_menu.
- Все фразы, используемые в шаблоне вставляются через функцию локализации __("")
- Стили и скрипты подключаются не в файле header.php, а через специальные функции в functions.php
- Код главной страницы в шаблоне перенесен из page-home.php во front-page.php

Теги:

wordpress,

тема,

вёрстка,

html,

css

+42

1220



Михаил @Mirantus
Пользователь

Поделиться публикацией



ПОХОЖИЕ ПУБЛИКАЦИИ

14 октября 2012 в 08:39

Sciter — встраиваемый HTML/CSS/scripting engine

+27 26k 168 30

20 сентября 2012 в 18:54

+21 30,7k 97 28

3 апреля 2009 в 19:39

Notepad++ для верстки HTML/CSS

+36 26,9k 41 64

ВАКАНСИИ **Мой круг**

PHP разработчик WordPress, Woocommerce

Diffco · Возможна удаленная работа

от 50000 до 130000 Р



PHP-Разработчик (WordPress)

AMarkets · Возможна удаленная работа

от 90000 Р



Специалист техподдержки - со знанием программирования

Вебразработка · Нижний Новгород

от 25000 до 50000 Р



WordPress Техподдержка (неполный день, удаленно)

MapSVG · Возможна удаленная работа

от 25000 до 35000 Р



Web-программист / Веб-разработчик / продвинутый - Junior/ Middle

VipDeposits · Москва

от 40000 до 70000 Р

[Все вакансии](#)Комментарии **28**

hermit931 3 июля 2014 в 09:50

+5

Добавлю несколько замечаний к статье:

1. На самом деле верстка темы и настройка темы это несколько разные вещи. Если подали материал в таком виде, то лучше статье дать более подходящее название, к примеру «Создание сайта на WordPress» или что то в этом духе.
2. Функции `wp_herd()` и `wp_footer()` являются нечто большим, чем «добавляет заголовки WordPress» и «добавить скрипты футера WordPress». Я бы назвал их «техническими» функциями, которые позволяют легко (грамотно?) управлять добавлением скриптов, стилей и мета соответственно в шапку или подвал сайта.
3. Навигация. Почему бы ее не вывести встроенной диалог настройки меню (Arreagance -> Menu)? Так мне кажется наглядней и более удобно управлять меню.
4. Главная страница. Есть также шаблоны `front-page.php` и `home.php` для этих целей. Использовать `page-home.php` я бы не рекомендовал в силу того, что при переименовывании страницы (я имею ввиду не намо название, а именно редактирую перmalink) с `home` в `main` страница отобразится дефолтная. Ну или переименовать главную страницу. Вообще неплохо было бы указать на [иерархию шаблонов](#) в статье.



kovalevsky 3 июля 2014 в 11:03

+3

Так тема-то заезженная до дыр, даже в рунете. Причём за 1 минуту можно найти намного лучше и подробнее, а автор ерунду здесь написал, которую можно найти в кодексе.

У вас статичное меню, то, что должно быть виджетами Вы тоже запихали в шаблон. Вашу тему невозможно будет перевести на другой язык без правки файлов, а некоторые куски куда тупо выдраны из дефолтной

ЧТО ОБСУЖДАЮТ

Сейчас

Вчера

Неделя

Ускоряем неускоряемое или знакомимся с SIMD

12,2k 71

1С и Яндекс.Облако Compute Cloud. Вдоль и поперек

1,5k 13

Сознание без внутреннего «Я»

19,8k 178

Не кодом единым: тест на знание серверной инфраструктуры

8,2k 20

Важные изменения в работе CTE в PostgreSQL 12

2,9k 5

Mirantus 3 июля 2014 в 11:18

0

Поправил на __('Search Results for: %s', 'default')



kovalevsky 3 июля 2014 в 15:30

0

Да суть не в том, что Вы поменяли. Суть в том, что Вы пишете статью о том, в чём Вы сами очень слабо разбираетесь.

Mirantus 3 июля 2014 в 15:51

-1

Статичное меню — это меню в таком виде:

```
<ul class="top-menu">
  <li><a href="/home/">Home</a></li>
  <li class="active">About us</li>
  <li><a href="/services/">Services</a></li>
  <li><a href="/partners/">Partners</a></li>
  <li><a href="/customers/">Customers</a></li>
  <li><a href="/projects/">Projects</a></li>
  <li><a href="/careers/">Careers</a></li>
  <li><a href="/contact/">Contact</a></li>
</ul>
```

В данном случае, мы видим совсем другой код. Меню в этой статье меняется динамически при добавлении или удалении страниц. Если вы считаете, что в статье допущена ошибка — напишите, как должно быть правильно — это будет интересно всем читателям.

mgremlin 4 июля 2014 в 09:07

0

Даже в стандартных темах WP меню != списку статических страниц. Страницы могут автоматом появляться в меню, но это не обязательно, и любые другие пункты могут быть добавлены. Так что, в вашем случае явный downgrade даже по сравнению с тупым стандартом.



soif 3 июля 2014 в 16:44

0

Уважаемый @kovalevsky (=) Не затруднит ли накидать ссылок более компетентно описывающих верстку тем для WP ??



quantum 4 июля 2014 в 02:56

0

codex.wordpress.org/Theme_Development



q_styler 3 июля 2014 в 10:27

-5

Спасибо! Буду кидать эту статью своим верстальщикам.

2fast4rabbit 3 июля 2014 в 11:22

+12

не надо... :]

-1

прочитал заголовок так: «Как сверстать Тему»



danaki 3 июля 2014 в 11:26

+5

Рекомендую взглянуть на Roots



utkorose 3 июля 2014 в 12:48

+2

даёшь максимальную информативность комментариев Хабра! roots.io/



danaki 3 июля 2014 в 15:18

0

У меня не хватает кармы ставить ссылки

maxxannik 5 июля 2014 в 13:11

0

roots сложная тема. Мы обожглись. Там ребята в угоду функциональности и идеализму пожертвовали гибкостью и сделали ее хрупкой. К примеру дочку уже не сделать. Ни у стартера, ни у производных тем. А это сразу минус по всем фронтам. Подойдет разве что для сайтов, которые пишут тему под себя и не планируют как либо менять ее в ближайшее время. Тем кому нужна стабильность и отсутствие переменчивости. Но практика показывает что таких не бывает.



messersveet 3 июля 2014 в 13:48

0

Так и не нашел в статье стилей для заголовков (Которые с косым фоном). Потерялись?

Mirantus 3 июля 2014 в 14:23

0

Да, вы правы. Спасибо за внимательность. Добавил в блок «Шаблон страницы».



franzose 3 июля 2014 в 15:03

+1

В заключении вы приводите ссылку на статью «Верстаем по БЭМу», однако в данной статье БЭМа как-то и не видно. К тому же было бы гораздо интереснее почитать про модификацию стандартных страниц с настройками (типа Custom Header) добавлением собственных параметров. Например о том, что без относительно грязного хака вставить свои настройки на главный экран «Внешний вид -> Меню» не получится.



darkrain 3 июля 2014 в 17:00

+1

У меня есть 5 летний опыт разработки под wordpress. Но я вот не понимаю о чем бы мне написать, для меня почти все очевидно. Напишите пожалуйста о чем вы бы хотели узнать и я напишу статью.



foxkeys 4 июля 2014 в 00:06

+1

Да. Данную статью трудно признать образцом...

`wp_list_pages`, статические (да еще и относительные — без домена) ссылки в футере, **не**использование `wp_nav_menu`, игнорирование `__()`, жестко заданный протокол (`http`) в ссылках...

Даже не знаю, с одной стороны — процентов 70 сделано более-менее правильно. И для новичков полезно. И для уважения к «кодексу» — он (кодекс) содержит только короткие примеры, но не полные мануалы. Да и на английском, что не для всех хорошо.

С другой стороны — если уж публиковать статью — то как образец.

Думаю, автору стоит изучить «свежие» темы из поставки самого вордпресса и доработать статью, с использованием всех рекомендованных технологий и функций.

Mirantus 4 июля 2014 в 05:08

0

Большое спасибо за конструктивный комментарий. Я обязательно воспользуюсь вашим советом и обновлю статью по мере возможности.



jandosul 4 июля 2014 в 22:21

0

Как уже упоминали выше есть root.io и cherry framework

maxxannik 5 июля 2014 в 13:15

+1

Начинать с нуля это опасно. Постоянно обжигаемся об такие сайты. Чтобы писать тему с нуля под WP, нужно иметь довольно высокую квалификацию и подготовку, иначе неминуемы ошибки.

Лучше использовать стартер-темы, которые с ходу содержат в себе нужную совместимость и соблюдение кодекса:

1. Супер чистая стартер тема understand github.com/automatic/_s от Automatic и лидеров сообщества WordPress
2. п.1 + Bootstrap 3 github.com/mindctrl/alienship
3. Еще хвалят Genesis, но он как бы платный.

Roots.io — мне не нравится тем что он хрупок. Там нарушение кодекса по всем фронтам. Не возможность создать дочернюю тему типовыми методами.

Он подойдет только в том случае, если нужно создать сайт, который не будет меняться долгое время и где архитектор готов мериться с нарушением кодекса и производными от этого затратами и проблемами.



itdef 6 июля 2014 в 11:58

0

автор, как давно ты знаком с wp ;)?



Zippovich 9 июля 2014 в 16:00

+3

эмм... мне это напомнило самоучитель Попова.

1. не раскрыт самый главный файл темы functions.php
2. ничего нет о хуках и пр. вкуснячках
3. статическое меню в html? вы шутите? а если я захочу урл или титл поменять?
4. подключение css и js непосредственно в footer.php и header.php? вы шутите, а зачем тогда функции wp_head() и wp_footer() — неужели только для плагинов, или что бы потом 2 jquery подключались на сайте?
5. Еще неплохо было бы рассказать про кастомные типы, настройку админки, работу с данными и пр.

P.S.: это вредная статья!

Casufi 10 июля 2014 в 20:25

0

Подскажите пожалуйста полезную статью. В идеале статью типа Quick Start и книгу, чтобы влезть глубже.



Zippovich 10 июля 2014 в 21:37

0

Мне хватило документации на сайте wordpress — там документация такая, что больше ничего не нужно. И статьи там хорошие, для начала вообще нужно структуру шаблона изучить, потом просмотреть темы стандартные, там все есть что нужно использовать и как, что не понятно — опять же в справке смотреть. Книгу когда-то встречал хорошую для девелоперов на Wordpress — помоему вот эта:

www.sitepoint.com/store/the-wordpress-anthology/



0

Есть еще замечательный ресурс <http://wordpress.stackexchange.com>. Ответы на всё подряд найти можно.

Только [полноправные пользователи](#) могут оставлять комментарии. [Войдите](#), пожалуйста.

САМОЕ ЧИТАЕМОЕ

Сутки

Неделя
Месяц

Что не так с Raspberry Pi

+94 50,6k 208 104

InterNyet — как в Советском Союзе избрали интернет и почему он не заработал

+75 54,6k 161 184

Сколько доменных имён .com не используется?

+29 16,2k 37 22

Безумие дотфайлов

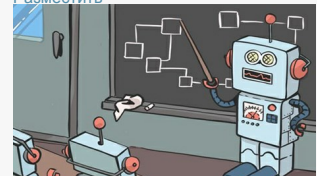
+73 16k 91 48

Электромобили утянут на дно автокорпорации?

+4 15,4k 21 138

ХАБР РЕКОМЕНДУЕТ

Разместить



Несколько интересных кейсов с SAP HANA: потенциал big data и машинного обучения



Проверь свои IT-скиллы на skillotron.com

Аккаунт

Войти

Регистрация

Разделы

Публикации

Хабы

Компании

Пользователи

Песочница

Информация

Правила

Помощь

Документация

Соглашение

Конфиденциальность

Услуги

Реклама

Тарифы

Контент

Семинары

