# Настройка Wordpress от Дмитрия Полос

**Админка** – Настройки – Общие

Настройки - Написание – Основная рубрика – рубрика по умолчанию установить (предварительно ее создать)

Создать главную страницу и страницу блога, для блога подправить url.

Настройки – Чтение – На главной странице отображать – Статическую страницу – и выбрать главную страницу и страницу записей.

Настройки – Постоянные ссылки – Произвольно – категории, postname.

Добавим плагины: ACF. Contact Form 7, HC Custom WP-Admin URL – чтобы скрыть админку, rus to lat advanced,

Устанавливать плагины и темы будем через composer. Нужно настроить наш composer так, чтобы он мог работать с репозиторием <https://wpackagist.org> В **composer.json** добавляем репозиторий wpackagist, правим пути к плагинам и темам

{

"require": {

"johnpbloch/wordpress": "^5.8",

},

**"repositories":[**

**{**

**"type":"composer",**

**"url":"https://wpackagist.org"**

**}**

**],**

**"extra": {**

**"installer-paths": {**

**"wordpress/wp-content/plugins/{$name}/": [**

**"type:wordpress-plugin"**

**],**

**"wordpress/wp-content/themes/{$name}/": [**

**"type:wordpress-theme"**

**]**

**}**

**}**

}

В терминале набираем команду:

composer require wpackagist-plugin/advanced-custom-fields

composer require wpackagist-plugin/rus-to-lat-advanced

composer require wpackagist-plugin/contact-form-7

composer require wpackagist-plugin/hc-custom-wp-admin-url

Название плагина взять со страницы плагина **Страница на Wordpress.org**, ссылка.

После активации HC Custom WP-Admin URL в Настройки – Постоянные ссылки появился новый пункт WP-Admin slug. Можем его настроить, н-р, rl-admin

# Темы и шаблоны Wordpress

Обязательный файл – **style.css.** Если тема не наследована, то обязателен также **index.php**

### Создадим унаследованную тему:

Создадим папку в themes, создадим там файл style.css, в нем обязательны для заполнения:

/\*\*

\* Theme Name: Название темы (обязательно)

\* Template: twentytwenty

\*/

где template – название род.темы.

Можем добавить

@include ‘../twentytwenty/style.css’;

Для формирования страниц в первую очередь поиск **шаблонов** идет в дочерней теме, если шаблонов нет, то в родительской. Для **functions.php** по-другому – отрабатывают оба файла: сначала в дочерней, затем в род. темах

Создадим тему с нуля:

Создадим папку в wordpress\wp-content\themes, создадим файл **style.css**, в нем обязательны для заполнения:

/\*\*

\* Theme Name: Название темы (обязательно)

\*/

Создадим **index.php.** Создадим **screenshot.png** для картинки в админке.

Все страницы можно разделить на два типа – содержащие одну сущность (н-р, запись, Singular.php) или несколько (Archive.php).

Копируем все **файлы верстки** в папку с темой. Можно также создать папку **assets**

Создаем **страницы** по след. алгоритму: для отдельных статических страниц **page-произвольно.php**. Заполнить в начале страницы комментарий с названием шаблона и в админке для соответствующих страниц выберем эти шаблоны.

Для страниц со своими типами записей **archive-название типа записей.php**. Шаблон в этом случае создавать не нужно.

Создадим для категорий **category.php**. **single.php** – для отдельных записей, **404.php**, **header.php, footer.php, functions.php**.

Нарежем **header.php** и **footer.php.** Для подключения вставим:

<?php get\_header(); ?>

<?php get\_footer(); ?>

Модальные окна отнести к футеру. Заполнить **404.php.**

Из страницы с блогами **blog.html** скопируем главную часть **в index.php**. Мы не можем для блога сделать свой шаблон, так как в настройках сайта мы указали, что страница Блог является страницей для записей, а Wordpress для таких страниц не даст сделать свой шаблон.

Также **index.php** должен использоваться тогда, когда не нашлось другого шаблона для чего-либо. Добавим после вывода шапки проверку:

if ( is\_home()) { ) и вставим соответствующий код для каждого случая.

is\_home() для страницы блога вернет true. Проверяет показывается ли страница с последними постами, обычно это главная страница сайта. Работа этого тега зависит от настроек параметра Параметры > Чтение > На главной странице отображать (Settings > Reading > Front page displays). По умолчанию, на главной странице выводятся последние записи сайта. По умолчанию, этот тег будет работать для главной страницы сайта, но если изменить настройки и указать для главной странице сайта статическую страницу, а для последних постов указать другую статическую страницу, то этот тег будет работать для страницы где выводятся последние посты и перестанет работать для главной страницы сайта.

Далее заполняем главную страницу **page-main.php,** заполняем и др. страницы

***подключим поддержку html5***

**В functions.php** подключим поддержку html5:

// подключаем html5

add\_theme\_support( 'html5', array(

'comment-list',

'comment-form',

'search-form',

'gallery',

'caption',

'script',

'style',

) );

#### Работаем с header.php

Вместо lang=”ru” вставляем

<?php language\_attributes(); ?>

Правим также кодировку:

charset="<?php bloginfo('charset');?>"

Удаляем title, description (вставить потом в админке), стили. Вставляем

<?php wp\_head(); ?>

Заметим, что для не главных страниц у body может быть добавлен определенный класс (н-р, для подчеркивания ссылок). Решение:

$body\_class = ‘’;

if (!is\_front\_page()) {

$body\_class = ‘название класса’;

}

и в теге body:

<body class=’<?php echo $body\_class;?>’>

Можно к этому тегу добавить:

<body <?php body\_class(); ?>>

Тогда в классах выведется доп. инф-ция, н-р, об авторизации

#### Работаем с footer.php

Удаляем скрипты. Вставляем (перед </body>)

<?php wp\_footer(); ?>

Появилась после этого плашка сверху.

#### Для правки путей (н-р, для картинок) использовать

echo get\_template\_directory\_uri();

Если таких путей много, можем создать отдельную функцию в functions.php:

function \_si\_assets\_path( $path ) {

return get\_template\_directory\_uri(). ‘/assets/’. $path;

}

#### Ссылка на главную страницу

<?= get\_home\_url() ?>

#### Подключаем стили и скрипты:

См. др. word файл

add\_action( 'wp\_enqueue\_scripts', 'robot\_lending\_scripts' );

function robot\_lending\_scripts() {

// styles

wp\_enqueue\_style( 'main', get\_stylesheet\_uri() );

wp\_enqueue\_style( 'style', get\_template\_directory\_uri() . '/css/style.css' , array('main'));

//scripts

wp\_enqueue\_script( 'app', get\_template\_directory\_uri() . '/js/app.js', false, null, true );

}

#### Логотип, title, favicon, миниатюры

В **functions.php**:

add\_action(‘after\_setup\_theme’, si\_setup());

function si\_setup() {

// custom logo

add\_theme\_support( 'custom-logo', [

'width' => 246.7,

'flex-width' => true, //true, если хотим, чтобы менялась

'flex-height' => true, //true, если хотим, чтобы менялась

'header-text' => '',

'unlink-homepage-logo' => false,

] );

где **unlink-homepage-logo** - При его включении, будет удалена ссылка на главную страницу сайта с картинки лого, если мы находимся на главной странице сайта. Также будет обнулен alt атрибут картинки на главной странице.

**Фильтр для класса и атрибута:**

// custom logo

add\_filter('get\_custom\_logo\_image\_attributes', 'filter\_logo\_class');

function filter\_logo\_class($custom\_logo\_attr) {

$custom\_logo\_attr['alt'] = 'logo';

return $custom\_logo\_attr;

}

add\_filter('get\_custom\_logo', 'filter\_function\_custom\_logo');

function filter\_function\_custom\_logo($html) {

$html\_class\_link = preg\_replace('/class="/i', 'class="header\_\_logo logo ', $html, 1);

return $html\_class\_link;

}

**В header.php:**

<?php

if (has\_custom\_logo()) {

echo get\_custom\_logo();

}

?>

Чтобы сохранить классы для логотипа, можем также сделать обертку для логотипа.

//title

add\_theme\_support( 'title-tag');

Заглавие сайта, описание вставить в админке

Миниатюры

add\_theme\_support(‘**post-thumbnails’**);

favicon

**Админка** – свойства сайта – выбрать логотип, нарезать из логотипа иконку.

#### Убираем плашку сверху:

В **functions.php**:

// убираем плашку сверху

add\_filter('show\_admin\_bar', '\_\_return\_false');

Этот фильтр подключен на wp\_footer()