**Подготовка**

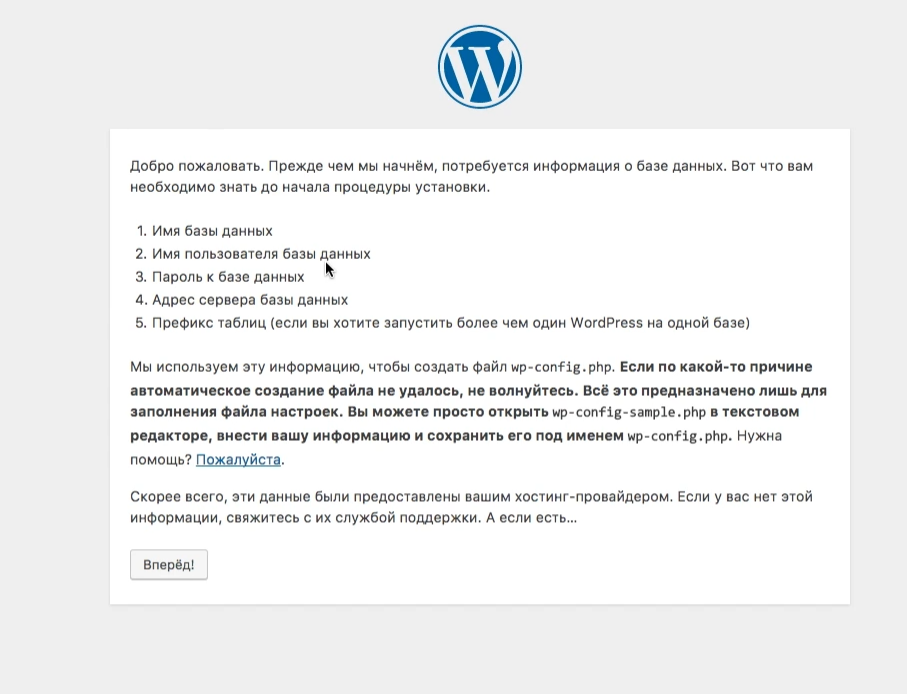
Скачать WordPress

В папке с установленным локальным сервером, напр. Open Server , находим папку с проектами. В Open Server это domains. Там создаём папку с названием проекта:

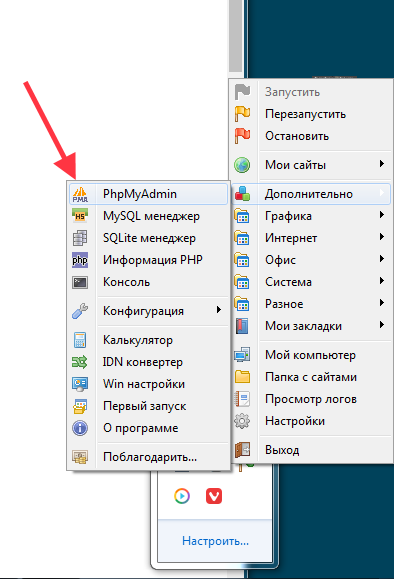
myProject.loc

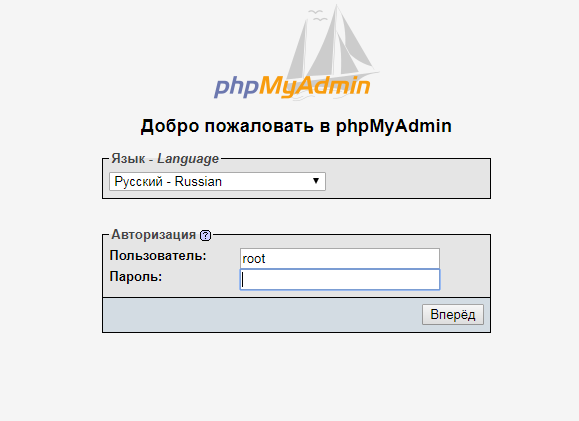
Сюда перенести содержимое скачанного архива WordPress

Запустить сайт на локальном сервере. Должно появиться такое окошко



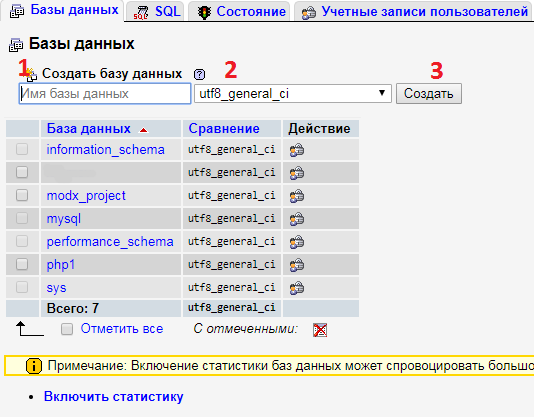
**Создание базы данных**



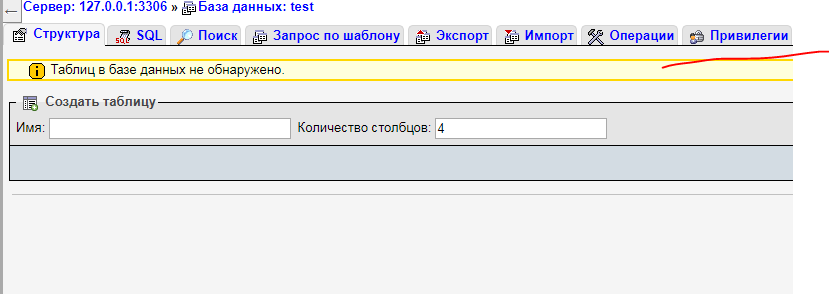


Заходим в phpMyAdmin, создаём новую базу данных, нажав на соответствующую ссылку в левом верхнем углу панели.

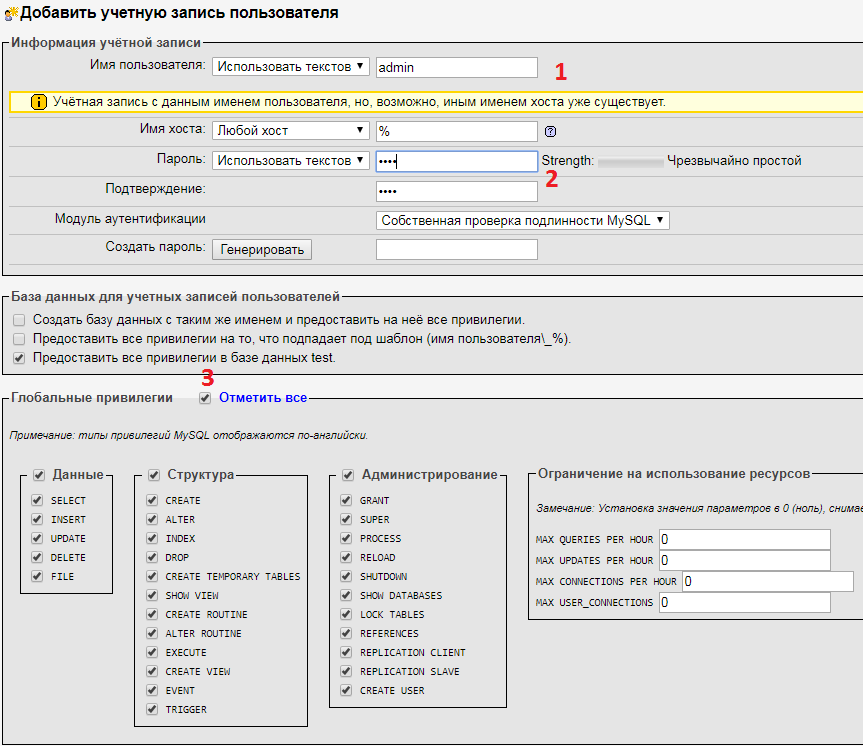
Даём название базе (1), выбираем кодировку (2), и жмём на кнопочку (3)



Создаём пользователя, нажав на «привелегии»



Далее [Добавить учетную запись пользователя](http://127.0.0.1/openserver/phpmyadmin/server_privileges.php?adduser=1&dbname=test&token=bc0e6fb805eff9fa357e4f8f5b7c11db)



Устанавливаем имя пользователя(1). Если всплыло предупреждение, как в примере, добавьте к admin название базы данных\сайта, что придаст уникальности имени.

Далее пароль(2) и привилегии(3).

Жмём «вперёд»!

Всё. Переходим к сайту на локальном сервере. Вводим имя пользователя, имя базы и пароль. «Сервер» и «префикс» оставляем в покое). Жмём кнопоку…

Это ещё не конец. Заполняем поля. В поле email вводим или свой или заказчика, если сайт под заказ.

Жмём «устанавливаем WordPress».

Вводим логин, пароль. Теперь всё -вошли в админку WP.

**Шаг 2. Установка вёрстки**

Установка будет происходить в папку с темами WP.

Минимальная тема WordPress состоит из двух файлов:

* style.css
* index.php

Оба этих файла помещают в папку темы.

Темы WordPress находятся в поддиректориях wp-content/themes/.

Создаём новую папку под нашу будущую тему. Папку назовём, напр., **TestTheme**.

Тема WordPress состоит из трёх основных типов файлов, в дополнение к изображениям.

Первый - это таблица стилей под именем style.css, которая контролирует внешний вид страниц сайта.

Второй предоставляет собой файл дополнительного функционала (functions.php). Остальные файлы - это [файлы шаблонов](https://codex.wordpress.org/index.php?title=%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B2_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D1%8B&action=edit&redlink=1), которые определяют, каким образом выводится информация из базы данных на веб-страницу.

Помимо собственно таблицы стилей вашей темы, файл style.css обязательно должен содержать информацию о вашей теме в виде комментариев. Каждая тема должна иметь свою собственную уникальную информацию в комментариях заголовка, в обратном случае возникнут проблемы в [диалоге выбора темы](https://codex.wordpress.org/%D0%90%D0%B4%D0%BC%D0%B8%D0%BD%D0%B8%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D1%8B%D0%B5_%D0%BF%D0%B0%D0%BD%D0%B5%D0%BB%D0%B8). Создавая новую тему на основе уже существующей, в первую очередь измените комментарии заголовка.

Ниже приведен шаблон заголовка таблицы стилей (style sheet header) вашей темы Rose. Он должен находится в начале файла стилей style.css, в первых строках:

/\*

Theme Name: Rose

Theme URI: домашняя-страница-темы

Description: краткое-описание-темы

Author: ваше-имя

Author URI: ваш-URI

Template: напишите-здесь-наименование-родительской-темы--необязательное-поле

Version: номер-версии--необязательное-поле

.

Развёрнутое описание темы/Лицензия-если-нужно.

.

\*/

### По идее, достаточно одной строки /\* Theme Name: сюда имя темы \*/

### в файл style.css в папке TestTheme. После комментария вставить скопированные стили из основного файла стилей.

### Создадим в папке TestTheme файл index.php.

### Перенесём вёрстку из index.html в index.php.

### Активируем тему в панели администратора. Внешний вид сайта будет не стилизованный.

### Создать папку assets в директории TestTheme. Туда скопировать папки img, js, css, fonts…

### Подключение стилей

### Перед </head> (важно именно это место!) пишем функцию

### <?php wp\_head(); ?>

### Файл дополнительного функционала

Тема может дополнительно использовать файл с необходимыми функциями для работы темы, он должен располагаться **в каталоге темы** и называться functions.php.

Этот файл будет нужен для подключения стилей.

https://wp-kama.ru/function-cat/podklyuchenie-faylov

# В файле подключаем функцию, которая правильно добавляет файл CSS стилей и регистрирует файл стилей, если он еще не был зарегистрирован - **wp\_enqueue\_style()**:

Ссылка на код …. https://wp-kama.ru/function/wp\_enqueue\_style

<?php

// правильный способ подключить стили и скрипты

add\_action( 'wp\_enqueue\_scripts', 'theme\_name\_scripts' );

// add\_action('wp\_print\_styles', 'theme\_name\_scripts'); // можно использовать этот хук он более поздний

function theme\_name\_scripts() {

wp\_enqueue\_style( 'style-name', get\_stylesheet\_uri() );

wp\_enqueue\_script( 'script-name', get\_template\_directory\_uri() . '/js/example.js', array(), '1.0.0', true );

}

'theme\_name\_scripts' можно поменять на 'theme\_name\_styles' т.к сейчас стили подключаем.

Удаляем вторую строку,   'style-name' можно заменить на main-style'

Теперь нужно указать путь к этому файлу со стилями. Нам поможет функция get\_stylesheet\_uri() . Она получает готовую ссылку (URL) на файл стилей style.css текущей темы.

Теперь запись в файле выглядит так:

<?php

add\_action( 'wp\_enqueue\_scripts', 'theme\_name\_styles' );

function theme\_name\_styles() {

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

}

Для подключения стилей из не корневой папки понадобится другая функция.

# **get\_template\_directory\_uri()**Получает URL текущей темы, т.е функция выдаст путь до корневой папки. В нашем случае это папка TestTheme.

# Получаем путь к стилевому файлу из папки assets:

# wp\_enqueue\_style( 'animate-style', get\_template\_directory\_uri() . '/assets/css/animate.css' );

В результате имеем что-то вроде:

<?php

add\_action( 'wp\_enqueue\_scripts', 'theme\_name\_styles' );

function theme\_name\_styles() {

wp\_enqueue\_style( 'animate-style', get\_template\_directory\_uri() . '/assets/css/animate.css' );

wp\_enqueue\_style( 'bootstrap-style', get\_template\_directory\_uri() . '/assets/css/bootstrap.css' );

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

}

**Картинки**

1. В style.css

Ищем в style.css url с путями к картинкам. Для этого CTRL + F пишем часть пути и ALT + Enter - можем менять путь с учётом появившейся папки assets.

1. В index.php

# Делаем это с помощью функции **bloginfo()** с аргументомtemplate\_url**,** который вернёт URL директории текущей темы

Функция выводит на экран различную информацию о блоге, которая, в основном, указывается в настройках сайта.

Пишем в <img src=’ <?php bloginfo( ‘template\_url‘) ?>/assets/….

**Скрипты**

# Для подключения скриптов используем функцию **wp\_enqueue\_script()**

Она срабатывает в момент подключения в очередь на вывод файлов стилей и скриптов.

https://wp-kama.ru/function/wp\_enqueue\_script

### Использование

wp\_enqueue\_script( $handle, $src, $deps, $ver, $in\_footer );

$deps*(массив)*

Массив названий скриптов от которых зависит этот скрипт; скрипты которые должны быть загружены перед этим скриптом. Другими словами, указываем после каких файлов будет подключен наш файл.

$in\_footer*(логический)*

Подключить скрипт в подвал?

Обычно скрипт подключается в <head> документа, если указать true, то скрипт будет подключен перед тегом </body>, точнее там где вызывается тег шаблона wp\_footer().

Итак, в файле index.php в конце вёрстки перед закрывающим тегом body пишем <?php wp\_footer(); ?>

Далее по примеру

#### **#1 Загружаем базовый скрипт WordPress с нестандартного адреса.**

Допустим, вы хотите вместо библиотеки jQuery из состава WordPress подключить её CDN-копию. **Добавьте этот код в файл functions.php** активной темы:

add\_action( 'wp\_enqueue\_scripts', 'my\_scripts\_method', 11 );

function my\_scripts\_method() {

wp\_deregister\_script( 'jquery' );

wp\_register\_script( 'jquery', 'http://ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js');

wp\_enqueue\_script( 'jquery' );

}

Итоговый вид:

