# Структура WordPress

Корневой каталог WordPress содержит три папки: ****wp-content****, ****wp-includes**** и ****wp-admin**** вместе с кучей разных PHP файлов, которые требуются для основных операций WP. Наиболее значимым из этих файлов является «wp-config.php». Изменяя этот файл, можно добавить к настройки WordPress, которые не доступны из консоли администратора. Рассмотрим вкратце анатомию движка ВордПресс и за что отвечают те, или иные файлы и папки.

****wp-admin****
Эта папка содержит различные файлы, такие как CSS, JavaScript, и PHP, которые обеспечивают функциональность консоли и административной части сайта.

****wp-content****
Папка wp-content содержит все загруженные пользовательские данные и разделяется на другие вложенные папки:

* languages
* plugins
* themes
* uploads

Папка **«languages»** содержит файлы переводов и локализации движка в формате .mo и .po.

Каталог **«themes»** содержит все загруженные темы (шаблоны).
Вы можете загрузить в эту папку много тем, но активировать сможете только одну тему (не считаем некоторые плагины, которые позволяют активировать больше). Кроме того, каталог «themes» **не может быть пустым**, так как для WordPress необходима по крайней мере [одна тема для корректной работы](https://sebweo.com/ru/kak-ustanovit-temu-na-wordpress/%22%20%5Co%20%22%D0%9A%D0%B0%D0%BA%20%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%B8%D1%82%D1%8C%20%D1%82%D0%B5%D0%BC%D1%83%20%D0%BD%D0%B0%20WordPress)!
По умолчанию папка «themes» уже содержит внутри стандартные темы. После выбора темы, рекомендуется удалить остальные.

Аналогичным образом, **«plugins»** используется для хранения установленных плагинов на вашем сайте. В отличие от каталога «themes», этот каталог **может быть пуст** и WordPress будет прекрасно работать без использования каких-либо сторонних плагинов. Вы также можете активировать столько плагинов, сколько необходимо (правда, хорошая практика заключается в установке только необходимых плагинов, поскольку количество установленных плагинов влияет на скорость работы сайта).

Все картинки (и другие медиа-файлы) хранятся в каталоге **«uploads»**, с разбивкой по годам, месяцам и/или дням. Эта папка представляет собой базу данных для всего не-текстового контента: изображения, видео, MP3, PDF-файлы, и т.д.
Сразу после установки WordPress папки «uploads» не будет, она будет создана автоматически после того, как вы начнете загружать медиа-файлы через консоль.

****wp-includes****
Папка «wp-includes» содержит в себе все основные и необходимые файлы для запуска WordPress через фронтэнд (пользовательский интерфейс). Папка содержит файлы PHP, CSS, JavaScript, и файлы изображений WordPress, которые обеспечивают основные функции программного обеспечения. Другими словами – это ядро движка ВордПресс.

Папки ****wp-admin**** и ****wp-includes**** являются основными папками WP, поэтому в них желательно ничего **не** изменять. По причине того, что при каждом обновлении ВордПресс происходит замена файлов в этих папках, и ваши правки будут утеряны.

## Структура темы



Путь к файлам темы **/wp-content/themes/название-темы/.**
Основная структура папок темы может включать в себя такие папки, как:

* /css/ – файлы стилей CSS, которые использует тема; ****ВАЖНО****: файл style.css должен размещаться в ****корневой**** папке темы
* /js/ – файлы JavaScript
* /images/ – хранит встроенные в тему изображения
* /languages/ – каталог хранит файлы переводов темы

Есть только два основных файла, которые необходимы [для активации темы](https://sebweo.com/ru/kak-ustanovit-temu-na-wordpress/%22%20%5Co%20%22%D0%9A%D0%B0%D0%BA%20%D1%83%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%B8%D1%82%D1%8C%20%D1%82%D0%B5%D0%BC%D1%83%20%D0%BD%D0%B0%20WordPress):

* index.php – файл, который создает блог
* style.css – стилевое оформление блога

Тем не менее, большинство тем включают в себя и другие файлы.
Вкратце разберем основные из них:

* ****header.php**** — глобальный файл, который отображает мета-данные страницы в разделе <head> и меню верхнего уровня.
* ****sidebar.php**** – этот файл отвечает за генерацию боковой колонки сайта. В основном здесь выводятся: список рубрик (категорий), последние записи, теги, различные баннеры.
* ****footer.php**** – этот файл отвечает за вывод футера, нижнего меню, копирайта и закрывает HTML-теги.
* ****index.php**** – это шаблон главной страницы сайта. По умолчанию этот файл отображает список последних постов и подгружает информацию с sidebar.php (сайдбара).
* ****single.php**** – отвечает за отображение отдельных постов. Файл содержит цикл, который запрашивает только один пост и формирует его вывод.
* ****page.php**** – формирует отдельные (статические) страницы (например, «Контакты», «О нас» и т.п.)
* ****archive.php**** – этот файл отвечает за вывод страницы архива записей
* ****category.php**** – формирует шаблон страницы, которая выводит посты по категориями
* ****tag.php**** – шаблон страницы, которая выводит список постов по тегам
* ****comments.php**** – этот файл управляет выводом комментариев
* ****functions.php**** – позволяет добавлять пользовательский код PHP и может влиять на основные элементы темы. Он добавляет функции и расширяет возможности вашего сайта. Работает как плагин WordPress.
* ****style.css**** – основной файл CSS-стилей темы.

Это основные файлы, которые могут присутствовать в теме. [Некоторые темы](https://sebweo.com/ru/gotovaya-tema-dlya-wordpress-znews/%22%20%5Co%20%22%D0%93%D0%BE%D1%82%D0%BE%D0%B2%D0%B0%D1%8F%20%D1%82%D0%B5%D0%BC%D0%B0%20%D0%B4%D0%BB%D1%8F%20WordPress%20%E2%80%93%20ZNews) могут содержать и другие файлы, а некоторые и пару-тройку. Все зависит от разработчика темы, и его желания более тонкой настройки сайта.





Тема (Theme, [https://codex.wordpress.org/Using\_Themes](https://codex.wordpress.org/%D0%98%D1%81%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D1%82%D0%B5%D0%BC)) – составляет основу внешнего вида вашего сайта на WordPress. По сути, вся разработка сайта на WordPress у большинства разработчиков сводится к тому, чтобы сделать тему, подключить её и несколько плагинов к WP и настроить сайт.

**Неправильно**, когда вся дополнительная функциональность типа Custom Post Types, реализуется в теме – она должна быть реализована в плагинах, темы отвечают только за внешний вид.

На [http://wphierarchy.com](https://wphierarchy.com/) приведена **полная иерархия** файлов в теме, на самом деле все намного проще – по минимуму в теме должны быть файлы style.css и index.php. Все остальное – лишь расширяет функциональность и внешний вид темы.







[Основы шаблонов](https://codex.wordpress.org/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8B_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%BE%D0%B2)

[Иерархия шаблонов](https://codex.wordpress.org/%D0%98%D0%B5%D1%80%D0%B0%D1%80%D1%85%D0%B8%D1%8F_%D1%88%D0%B0%D0%B1%D0%BB%D0%BE%D0%BD%D0%BE%D0%B2)

[Архитектура сайта 1.5](https://codex.wordpress.org/%D0%90%D1%80%D1%85%D0%B8%D1%82%D0%B5%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D1%81%D0%B0%D0%B9%D1%82%D0%B0_1.5)

## Как это работает

За всю логику: какой файл подключать, отвечает файл ядра [wp-includes/template-loader.php](https://github.com/WordPress/WordPress/blob/master/wp-includes/template-loader.php). Если разобраться, в нем все описано. Но, это занятие не особо интересное, поэтому я его распишу.

****Прежде всего.**** template-loader.php подключается после того как загрузится вся среда WordPress. После того, как отработает файл wp-load.php и обработается основной запрос - функция [wp()](https://developer.wordpress.org/reference/functions/wp/). Т.е. template-loader.php подключается в самом-самом конце PHP скрипта...

Что делает wp()?

создает запрос в базу данных, на основе переданных аргументов (параметр $query\_vars), результатами запроса заполняет глобальные переменные;



Array ( [page] => 0 [pagename] => s [error] => [m] => [p] => 0 [post\_parent] => [subpost] => [subpost\_id] => [attachment] => [attachment\_id] => 0 [name] => s [static] => [page\_id] => 0 [second] => [minute] => [hour] => [day] => 0 [monthnum] => 0 [year] => 0 [w] => 0 [category\_name] => [tag] => [cat] => [tag\_id] => [author] => [author\_name] => [feed] => [tb] => [paged] => 0 [comments\_popup] => [meta\_key] => [meta\_value] => [preview] => [s] => [sentence] => [fields] => [menu\_order] => [category\_\_in] => Array ( ) [category\_\_not\_in] => Array ( ) [category\_\_and] => Array ( ) [post\_\_in] => Array ( ) [post\_\_not\_in] => Array ( ) [tag\_\_in] => Array ( ) [tag\_\_not\_in] => Array ( ) [tag\_\_and] => Array ( ) [tag\_slug\_\_in] => Array ( ) [tag\_slug\_\_and] => Array ( ) [post\_parent\_\_in] => Array ( ) [post\_parent\_\_not\_in] => Array ( ) [author\_\_in] => Array ( ) [author\_\_not\_in] => Array ( ) [ignore\_sticky\_posts] => [suppress\_filters] => [cache\_results] => 1 [update\_post\_term\_cache] => 1 [update\_post\_meta\_cache] => 1 [post\_type] => [posts\_per\_page] => 10 [nopaging] => [comments\_per\_page] => 10 [no\_found\_rows] => [order] => DESC )

****Сначала**** срабатывает хук [template\_redirect](https://wp-kama.ru/hook/template_redirect).

//Хуками в WordPress называются фильтры (filter) и события (action). В программной части это абсолютно одно и тоже, т.е. обрабатывается и то и другое одинаково, можно например заменять функции add\_filter() и add\_action() — все будет работать! Разделение нужно, потому что по смыслу это разные вещи...

Фильтры нужны, чтобы отфильтровать передаваемое значение, т.е. фильтр получает значение и обязательно должен его вернуть (изменённое или нет). А события — это события, они просто срабатывают в определенный момент и ничего не возвращают.

К хукам (фильтрам или событиям) прикрепляются PHP функции, затем, в момент срабатывания хука, эти функции будут срабатывать.

В этом хуке можно произвести какие-то проверки и если надо перенаправить на другой URL. В нем нужно обрывать работу скрипта через [die()](https://developer.wordpress.org/reference/functions/wp_die/). Т.е. если этот хук что-то меняет, то на этом работа файла template-loader.php заканчивается и мы попадаем на какую-то другую страницу.

****Дальше**** идут проверки по всем условным тегам где выясняется какой файл шаблона подходит под текущий запрос. Проверяется условный тег (это вспомогательные PHP функции WordPress, а называют их тегами, чтобы как-то выделить из числа прочих PHP функций), затем еще один и еще... Как только один из условных тегов сработал, вызывается соответствующая функция, которая находит подходящий файл шаблона и возвращает путь до него. Все такие функции описаны в get\_query\_template(). Получает путь до указанного файла шаблона. Указывается тип запроса: index, category, 404 и т.д.

****Дальше**** - путь до файла темы определен! Теперь он прогоняется через фильтр template\_include. Который позволяет изменить файл шаблона для текущего запроса. Например, мы зашли на постоянную страницу, WordPress подбирает какой файл шаблона должен быть показан - это файл page.php: home/site.ru/wp-content/themes/mytheme/page.php. С помощью этого фильтра можно изменить путь такого файла.

****Дальше****, файл подключается в PHP и запускается визуальная часть генерации страницы.

Генераторы тем:

<http://www.yvoschaap.com/wpthemegen/> создание простых тем

<http://templatr.cc/> создание простых тем

[www.lubith.com](http://www.lubith.com) создание тем средней сложности (Очень популярный сервис. Жаль, закрылся в январе 2018г.)

<http://cooltemplate.com/> создание тем средней сложности (автоматически)

<http://www.wpthemegenerator.com/> генерация сложных (премиум) шаблонов (быстро, качественно и платно)

<http://www.artisteer.com/> конструктор тем для ПК (для многих CMS. Платно)