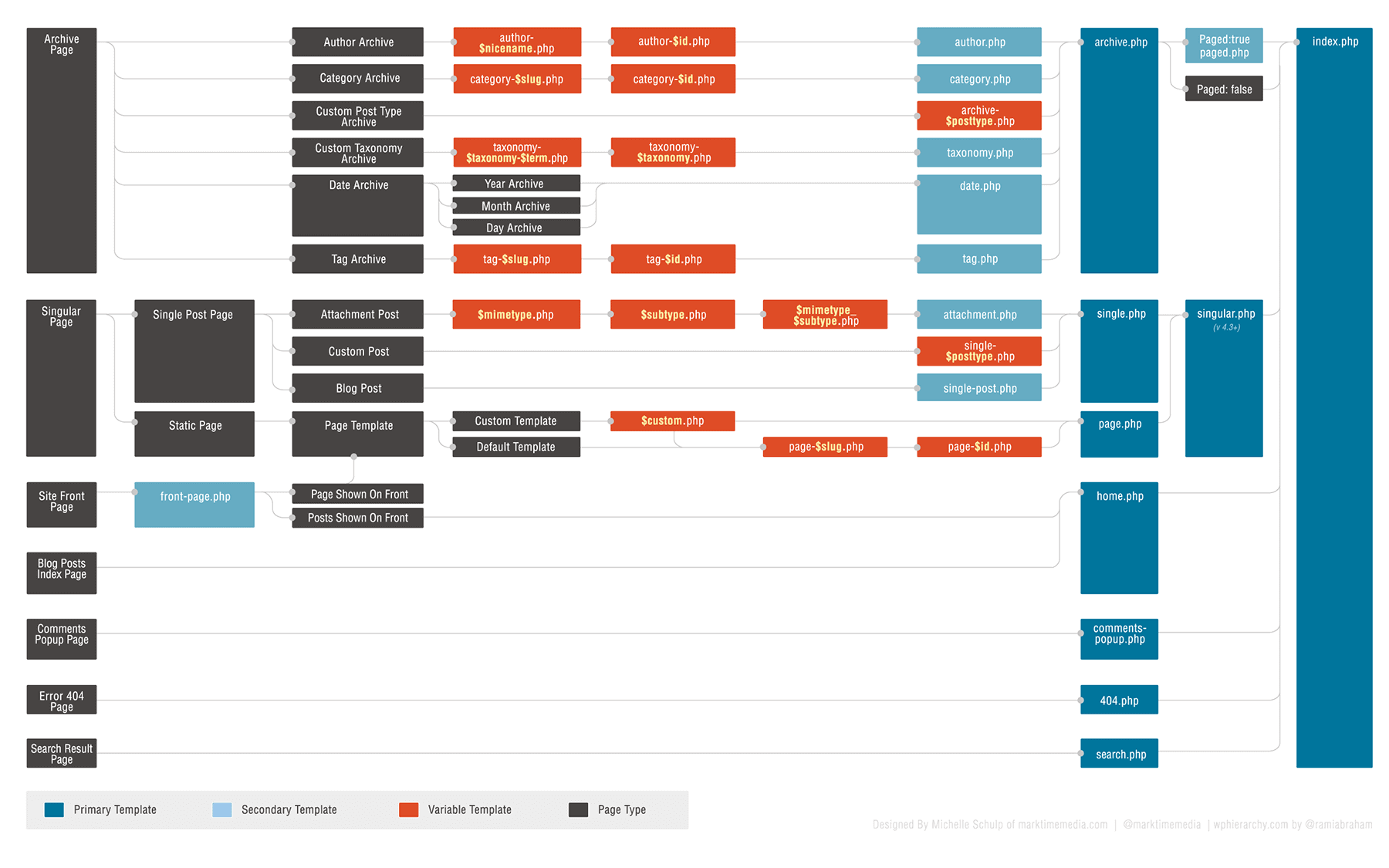
Иерархия файлов темы (шаблона)



файлы должны находится в корневой папке темы.

Записи

Страница (запись page)

* {любое\_название}.php (когда используется шаблон страницы)
* page-{ярлык\_записи}.php
* page-{ID\_записи}.php
* page.php
* singular.php
* index.php

Запись (запись post)

* single-post-{ярлык\_записи}.php
* single-post.php
* single.php
* singular.php
* index.php

Произвольный тип записи

{любое\_название}.php (для древовидного типа с поддержкой шаблонов. С WP 4.7)

single-{post\_type}-{ярлык\_записи}.php

single-{post\_type}.php

single.php

singular.php

index.php

Вложение

{начало\_MIME\_типа}.php

{конец\_MIME\_типа}.php

{начало\_MIME\_типа}-{конец\_MIME\_типа}.php

attachment.php

single-attachment-{ярлык\_вложения}.php (позволяет указать шаблон для отдельной картинки)

single-attachment.php (тоже что и attachment.php)

single.php

singular.php

index.php

Под началом и концом MIME типа, имеется виду первая и последняя часть MIME типа, разделенная /. Например, MIME-тип текстового файла: 'text/plain' и значит будет проверяться наличие файл text.php, затем plain.php, затем text-plain.php.

Полный список MIME типов смотрите в описании get\_allowed\_mime\_types().

меню

Архивы

Рубрика

* category-{ярлык}.php
* category-{id}.php
* category.php
* archive.php
* paged.php (если страница пагинации)
* index.php

Метка

* tag-{ярлык}.php
* tag-{id}.php
* tag.php
* archive.php
* paged.php (если страница пагинации)
* index.php

Таксономия

taxonomy-{ярлык\_таксономии}-{ярлык\_элемента}.php

taxonomy-{ярлык\_таксономии}.php

taxonomy.php

archive.php

paged.php (если страница пагинации)

index.php

Страница архива типа записи

archive-{ярлык\_типа\_записи}.php

archive.php

paged.php (если страница пагинации)

index.php

Страница автора

author-{nickname}.php

author-{id}.php

author.php

archive.php

paged.php (если страница пагинации)

index.php

Страница архива по дате (день, месяц, год)

date.php

archive.php

paged.php (если страница пагинации)

index.php

404 страница

404.php

index.php

Страница поиска

search.php

index.php

Главная страница

front-page.php

(логика постоянных страниц, если для главной выбрана страница)

home.php

index.php

Страница блога

Страница блога появляется, когда для главной выбрана постоянная страница

* home.php
* index.php

Встраивания (embeds)

Шаблоны встраивания используются когда запрашивается запись через REST API. Встраивания появились в версии 4.5 и позволяют встраивать ваши записи в чужие сайты. См. get\_post\_embed\_url()

* embed-{post-type}-{post\_format}.php
* embed-{post-type}.php
* embed.php

Чтобы изменить только контент встраивания, можно в теме создать файл embed-content.php и описать там HTML. Оригинальный HTML находится в файле движка /wp-includes/theme-compat/embed-content.php

Как это работает

За всю логику: какой файл подключать, отвечает файл ядра wp-includes/template-loader.php.

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(). Т.е. template-loader.php подключается в самом-самом конце PHP скрипта...

Сначала срабатывает хук template\_redirect. В этом хуке можно произвести какие-то проверки и если надо перенаправить на другой URL. В нем нужно обрывать работу скрипта через die(). Т.е. если этот хук что-то меняет, то на этом работа файла template-loader.php должна закончиться и мы «улетаем» на какую-то другую страницу.

Дальше срабатывает никому не интересный хук exit\_on\_http\_head. Он позволяет сделать так, чтобы при HTTP HEAD запросе что-либо выводилось на экран...

Дальше идут проверки по всем условным тегам, где выясняется какой файл шаблона подходит под текущий запрос. Проверяется условный тег, 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().

На деле, условные теги — это вспомогательные PHP функции WordPress, а называют их тегами, чтобы как-то выделить из числа прочих PHP функций. В большинстве своем, условные теги созданы для удобного выполнения проверок различных условий, например, находимся ли мы на главной странице сайта, авторизован ли пользователь. Самый полный список условных тегов: <https://wp-kama.ru/function-tag/uslovnyie-tegi>.

Дальше - путь до файла темы определен! Теперь он прогоняется через фильтр template\_include. Который позволяет нам изменить файл шаблона для текущего запроса.

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

Другие файлы темы

Файлы ниже также используются в теме, но они просто подключаемые и не участвуют в иерархии. Список таких «подключаемых» файлов.

* functions.php - всегда подключается автоматически при загрузке темы.
* header.php - get\_header()
* sidebar.php - get\_sidebar()
* footer.php - get\_footer()
* searchform.php - get\_search\_form()

Хуки

Генерация страницы в WordPress – это целый набор функций и запросов к базе данных. Чтобы иметь возможность вклинится в процесс формирования страницы в процессе вывода, разработчики включили так называемые “крючки”, чтобы разработчики могли “повесить” на них свой собственный код. Эти крючки и есть хуки.

Хуки в WordPress бывают двух видов: 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 эти функции сработают, в момент срабатывания хука. Такое поведение в программировании называется перехват.

В функцию прикрепленную к хуку могут передаваться дополнительные данные (параметры). На основе этих данных в функции можно создавать какую-либо логику - что-то проверять перед тем, как что-то сделать.

Все хуки ядра WordPress 5.2.2 на <https://wp-kama.ru/hooks/hooks-db>. Всего в WordPress их около 2000.

Нужно записать хуки в файл functions.php. Этот файл находится в директории wp-content/themes/yourtheme (где /yourtheme – директория, в которой находится текущая тема).

Вот как примерно выглядит конструкция с использованием хука:

function add\_point() {

//Тут функция

}

add\_action('publish\_post', 'add\_point');

Все что внутри функции, будет срабатывать при публикации поста. Вы можете использовать любой другой хук, который есть на сайте WordPress.

События

События позволяют добавлять дополнительные пользовательские функции в определенный момент обработки страницы. Например, Вы можете добавить дополнительные виджеты и меню или добавить рекламные сообщения на свою страницу.

Фильтры

Фильтры позволяют перехватывать и изменять данные по мере их обработки. Например, Вы можете захотеть вставить другой класс CSS в HTML-элемент WordPress или изменить некоторые блоки Ваших страниц.