### Часть 3

## Создание темы Law

**Урок 1 - Стартовая тема и шаблон**

Подготовливаем директорию и создаем стартовую тему.

preview: <https://www.free-css.com/assets/files/free-css-templates/preview/page232/law/>

download template: <https://www.free-css.com/free-css-templates/page232/law>

Создаем домен сайта (директорию для сайта) на сервере -> Устанваливаем WP -> в папку themeы помещаем стартовую тему с underscores -> перезапускаем сервер.

Создаем БД для сайта.

Открываем law.loc в браузере -> устанивливаем WP и настравиваем config для подключения к БД.

В папке с темой можно создать временную папку template/ для хранения исходного макета сайта.

В корень темы кидаем screenshot.png чтобы в админке у темы была картинка preview.

**Урок 2 - Подключение стилей и скриптов**

Создаем assets/ в корне темы -> копируем в assets/ папки из исходного шаблона (css/, js/, fonts/ и т.д.).

В этой части автор решил пойти другим путем по переносу исходного шаблона в WP.

Поэтому берем содержимое index.html из исходного шаблона, и под чистую заменяем им содержимое файла index.php, в разрабатываемой теме.

Далее необходимо подключить скрипты и стили в functions.php:

Создаем свой файл стилей law/assets/css/law.css (чтобы наши стили не перемешивались с исходними стилями \_underscores).

В index.php смотрим необходимые скрипты для подключения и подключаем их через functions.php в law\_scripts()

*/\*\*  
 \* Enqueue scripts and styles.  
 \*/***function** law\_scripts() {  
. . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . .

/\*My scripts and styles  
\* header part  
\*/  
 wp\_enqueue\_style( 'law-googlefonts','https://fonts.googleapis.com/css?family=Work+Sans:300,400,500,700,800' );  
 wp\_enqueue\_style( 'law-animate', get\_template\_directory\_uri() . '/assets/css/animate.css' );  
 wp\_enqueue\_style( 'law-icomoon', get\_template\_directory\_uri() . '/assets/css/icomoon.css' );  
 wp\_enqueue\_style( 'law-bootstrap', get\_template\_directory\_uri() . '/assets/css/bootstrap.css' );  
 wp\_enqueue\_style( 'law-magnific-popup',get\_template\_directory\_uri() . '/assets/css/magnific-popup.css' );  
 wp\_enqueue\_style( 'law-owl-carousel',get\_template\_directory\_uri() . '/assets/css/owl.carousel.min.css' );  
 wp\_enqueue\_style( 'law-owl-theme-default',get\_template\_directory\_uri() . '/assets/css/owl.theme.default.min.css' );  
 wp\_enqueue\_style( 'law-flexslider',get\_template\_directory\_uri() . '/assets/css/flexslider.css' );  
 wp\_enqueue\_style( 'law-theme',get\_template\_directory\_uri() . '/assets/css/style.css' );  
 /\* my styles law.css\*/  
 wp\_enqueue\_style( 'law-custom',get\_template\_directory\_uri() . '/assets/css/law.css' );  
  
 wp\_enqueue\_script( 'law-modernizr-js', get\_template\_directory\_uri() . '/assets/js/modernizr-2.6.2.min.js' );  
 wp\_enqueue\_script( 'law-respond-js', get\_template\_directory\_uri() . '/assets/js/respond.min.js' );  
  
 /\* footer part \*/  
 /\* Подключаем Last.ver - JQuery  
 \* Подключаем именно в header так как эту библ. будет использ. слайдер из unyson  
 \*/  
 wp\_deregister\_script( 'jquery' );  
 wp\_register\_script(  
 'jquery',  
 get\_template\_directory\_uri() . '/assets/js/jquery-3.4.1.min.js'  
 );  
 wp\_enqueue\_script('jquery');  
 wp\_enqueue\_script(  
 'law-jquery-easing-js',  
 get\_template\_directory\_uri() . '/assets/js/jquery.easing.1.3.js', **array**(), '', **true** );  
 wp\_enqueue\_script(  
 'law-bootstrap-js',  
 get\_template\_directory\_uri() . '/assets/js/bootstrap.min.js', **array**(), '', **true** );  
 wp\_enqueue\_script(  
 'law-jquery-waypoints-js',  
 get\_template\_directory\_uri() . '/assets/js/jquery.waypoints.min.js', **array**(), '', **true** );  
 wp\_enqueue\_script( //этот атвор пропустил  
 'law-jquery-stellar-js',  
 get\_template\_directory\_uri() . '/assets/js/jquery.stellar.min.js', **array**(), '', **true** );  
 wp\_enqueue\_script(  
 'law-owl-carousel-js',  
 get\_template\_directory\_uri() . '/assets/js/owl.carousel.min.js', **array**(), '', **true** );  
 wp\_enqueue\_script(  
 'law-jquery-flexslider-js',  
 get\_template\_directory\_uri() . '/assets/js/jquery.flexslider-min.js', **array**(), '', **true** );  
 wp\_enqueue\_script(  
 'law-jquery-countTo-js',  
 get\_template\_directory\_uri() . '/assets/js/jquery.countTo.js', **array**(), '', **true** );  
 wp\_enqueue\_script(  
 'law-jquery-magnific-popup-js',  
 get\_template\_directory\_uri() . '/assets/js/jquery.magnific-popup.min.js', **array**(), '', **true** );  
 wp\_enqueue\_script(  
 'law-magnific-popup-options-js',  
 get\_template\_directory\_uri() . '/assets/js/magnific-popup-options.js', **array**(), '', **true** );  
 wp\_enqueue\_script(  
 'law-main',  
 get\_template\_directory\_uri() . '/assets/js/main.js', **array**(), '', **true** );

}

так как в wp\_enqueue\_style() и wp\_enqueue\_script() есть heandler(ы) с одинаковыми именами, автор присвоил heandler(ам) из wp\_enqueue\_script() постфикс -js, чтобы не было пересечения со style.

Далее необходимо полностью очистить <header></header> -> вызвать wp\_head() в <header>

Затем чистим футер, т.е. убираем явное подключение скриптов и вызываем

<?php wp\_footer(); ?> перед закрывающим тегом </body>

Теперь необходимо настроить вывод картинок:

сначала в index.php находим подстроку 'url(images', выделяем ее и наж-ем ctrl+r, затем указываем чем заменять: url(<?php echo get\_template\_directory\_uri(); ?>/assets/images  
и наж-ем replace all чтобы заменить все подобные вхождения, получится вот так:

<div class="video fh5co-video" style="background-image: url(**<?php** get\_template\_directory\_uri(); **?>**/assets/images/video.jpg);">

Дальше ищем остальные вхождения для подключения картинок с источником src, также при помощи replace all меняем все вхожения src="images на src="<?php echo get\_template\_directory\_uri(); ?>/assets/images

Автор начал создавать тему с полного переноса шаблона index.html в index.php (что не обязательно) во избежании ошибок, так как страница большая и содержить много элементов верстки.

И последнее, файл со скриптом respond.min.js подключается по условию(по условию могут также подключатсья и стили), чтобы этого добиться в WP есть 2 функции:

**wp\_script\_add\_data()** - Добавляет данные к подключенным на странице скриптам с помощью wp\_enqueue\_script(). Используйте когда, например нужно выводить скрипт только для «IE 6» или «lt IE 9» (меньше IE 9)... - <https://wp-kama.ru/function/wp_script_add_data>

**wp\_script\_add\_style()** - аналогичная функция - <https://wp-kama.ru/function/wp_style_add_data>

сразу после подключения

wp\_enqueue\_script( 'law-respond-js', get\_template\_directory\_uri() . '/assets/js/respond.min.js' );

вызываем:

wp\_script\_add\_data('law-respond-js', 'conditional', 'lt IE 9');

если посмотреть код страницы мы увидим:

<!--[if lt IE 9]>

<script type='text/javascript' src='http://law.loc/wp-content/themes/law/assets/js/respond.min.js?ver=5.2.2'></script>

<![endif]-->

**Урок 3 - Плагин TGM**

В данном уроке подключим Unyson.   
Ранее во второй части данного курса автор упоминал, что при установки темы пользователем, теме могут быть необходимы различные плагины для корректной работы ( + это требование Forest Team для premium тем ) и чтобы не писать мануал для польз. какие плагины необходимы для корректн. работы, как эти плагины установить, так как это далеко не самая лучшая практика, нам будет необходим плагин:

TGM Plugin Activation: <http://tgmpluginactivation.com/download/>

Инструкция по работе с TGM: <http://oddstyle.ru/wordpress-2/wordpress-plaginy/komanda-stoyashhaya-za-tgm-plugin-activation-vypustila-custom-generator.html>

Данный плагин проверяет установлен ли плагин из списка, который мы указали в настройках. Если какой-либо плагин, из списка, не найдено, то TGM выдаст предупреждение, либо требование об установке данного плагина, в зависимости от того, что указано в настройках.

Скачиваем TGM -> 

скачиваем и распаковываем плагин из архива в папку   
/law/inc  
и переименовываем /inc в /tgm

На что стоит обратить внимание, это:  
class-tgm-plugin-activation.php - класс который необходимо подключить.  
example.php - пример подключения

создаем файл tgm.php(Имя любое) и копируем в него содержимое из файла example.php из корневой папки плагина tgm

Подключаем tgm.php в functions.php в самом конце:

/\* TGM plugin \*/  
**require\_once** get\_template\_directory() . '/inc/tgm/tgm.php';

Затем в файле tgm.php указываем путь к классу class-tgm-plugin-activation.php:

**require\_once** get\_template\_directory() . '/inc/tgm/class-tgm-plugin-activation.php';

Далее настроим Array of plugin arrays в файле tgm.php,  
Array of plugin arrays - это массив который содержит массивы плагинов, в каждом подмассиве описывается отдельный плагин и способ его подключения.

tgm может устанавливать плагин как из оф.репозитрия ( это требование TF для premium тем ) так и из локального хранилища:  
law\inc\tgm\plugins  
в которое плагины помещаются в виде архива .zip

Удалим из law\_register\_required\_plugins() все не нужные плагины и оставим вот так:

/\*  
 \* Array of plugin arrays. Required keys are name and slug.  
 \* If the source is NOT from the .org repo, then source is also required.  
 \*/  
$plugins = **array**(  
 // This is an example of how to include a plugin from the WordPress Plugin Repository.  
 **array**(  
 'name' => 'Unyson', // Название плагина, можно посмотреть на https://ru.wordpress.org/plugins/  
 'slug' => 'unyson', // slug плагина (название папки в которой лежит сам плагин)  
 'required' => **false**, // false - предупреждение, true - требование об установке плагина  
 ),  
);

Согласно требованиям TM: 'required' => true, не допускается, разрешено только 'required' => false

Так же, ниже, в разделе

/\*  
 \* Array of configuration settings. Amend each line as needed.

В массиве $config = **array**( . . . .

Возможно нас заинтересует опция:

'is\_automatic' => **false**, // Automatically activate plugins after installation or not.

*Согласно требованиям TM: 'is\_automatic' => true, не допускается, но это надо проверить.*

Теперь если обновить страницу админки мы увидим предупреждение о том, что необходимо установить плагин Unyson:



так как мы его еще не установили.

Чтобы перейти к списку плагинов которые необходимо установить кликаем по 'Begin installing plugin' затем Install

Далее необходимо активировать плагин кликнув по 'Return to Required Plugins Installer' -> active плагин

Затем ставим расширения:  
Unyson -> Конструктор страниц - загрузить  
Unyson -> Слайдер - загрузить

По итогу: таким образом можно предлагать пользователю темы к установке, без которых тема будет работать не корректно.