Оглавление

[**Общая информация о сборке:** 1](#_Toc179448567)

[Для работы со сборкой нужно: 1](#_Toc179448568)

[Сборка состоит из: 2](#_Toc179448569)

[Папка dev состоит из следующих папок/файлов: 2](#_Toc179448570)

[Папка components состоит из: 3](#_Toc179448571)

[Особенности использования сборки gulp: 3](#_Toc179448572)

# **Общая информация о сборке:**

Все стили и скрипты, присутствующие в сборке, можно изменять/удалять/заменять в соответствии с потребностями.   
Они присутствуют для упрощения и ускорения работы с проектом, но если на данный момент, вам неудобно/непривычно/сложно и т.д. разобраться в стилях/скриптах – лучше писать своё.

Все скрипты написаны на jQuery.

Все команды для работы со сборкой написаны в файле README.txt

Первый уровень папок, которые находятся в папке components изменять/удалять нельзя (**исключение**: если вы сможете разобраться в настройках сборки и изменить там названия), кроме папки modals.   
На этот уровень вложенности можно добавлять другие папки. (если будет такая потребность)

Названия основных файлов/папок на первом уровне вложенности проекта (dev, gulpfile и т.д.) нельзя изменять или удалять.

## Для работы со сборкой нужно:

1. Установить node версии 18.14.0
2. Установить Visual Studio Code
3. Установить npm версии 9.3.1. (если версия будет выше/ниже могут быть проблемы со сборкой)
4. В папку с проектом расположить папку dev и файлы: gulpfile.js, package.json
5. Так же в папке с проектом выполнить команду npm i (писать не нужно, если в папке с проектом есть папка node\_modules и сборка запускается без проблем)

1, 2, 3 пункты выполнять не нужно, если на ПК это установлено.

Проверить наличие и версию node: node -v

Проверить наличие и версию npm: npm -v

(Команды писать в консоли)

**После выполнения всех пунктов, можно повторно их не выполнять**.   
Для работы с новым проектом можно **заменить папку dev**.   
(**папку dev и, если нужно, папку release** предыдущего проекта где-нибудь сохранить)

## Сборка состоит из:

* Папка **dev** - нужна для всех видов работ с вёрсткой
* Папка **release** - создаётся при запуске команд сборки. Нужна для отображения сайта в браузере и демонстрации дизайнеру.
* Файл **gulpfile.js** - нужен для настройки сборки
* Файл **package.json** - тут указаны все используемые плагины в сборке
* **README.txt** - команды для работы со сборкой.

## Папка dev состоит из следующих папок/файлов:

* **сomponents** - хранит все созданные компоненты.
* **fonts** - для добавления файлов шрифтов. Папку можно удалить если файлов не будет
* **images** - для всех картинок/иконок/фавикона
* **js** - общие скрипты, которые будут использоваться на всех страницах (или большинство из них и вынести в отдельный компонент не выйдет)
* **libraries** - для хранения всех используемых библиотек
* **style** - для всех общих шрифтов, которые будут использоваться на всех страницах (или большинство из них и вынести в отдельный компонент не выйдет)

**В папку style можно добавлять файлы только формата css.**

**Файлы формата scss и sass нужно добавлять в папку scss.**

Все файлы в папке scss будут помещены на 1 уровень вложенности папки style (в папке release)

Файлы **common.scss** и **common.js** удалять/изменять название нельзя. (тут лежат общие скрипты/стили)

* Файлы .**html** (добавлять новые страницы только на этом уровне)

Наличие файлов **index.html (карта сайта)** и **main.html (главная страница)** обязательны.

## Папка components состоит из:

* **blocks** - сюда добавлять все новые section со своими стилями и скриптами.
* **elements** - сюда добавлять дополнительные блоки по типу: фильтр, хлебные крошки, пагинация, карта и т.д.
* **footer** - Подвал сайта (можно добавлять стили и скрипты)
* **header** - Шапка сайта (можно добавлять стили и скрипты)
* **modals** - Все модальные окна.

Всё, что лежит в папках **blocks** и **elements** можно изменять/удалять.

Все названия файлов в папке **components** должны называться следующим образом:

* .scss/.css - style.scss(.css)
* .js - script.js
* .html - template.html

Названия папок, в свою очередь, можно называть как угодно. Желательно стараться делать такие же имена, какие будут использованы в качестве названия класса.

## Особенности использования сборки gulp:

В сборке есть [плагин](https://www.npmjs.com/package/gulp-file-include), который позволяет использовать в html несколько функций. Основные из них:

1. @@include - добавление блока html в другой файл html
2. @@name - создание переменной, для передачи параметров в @@include.
3. @@if – условие, при котором блок html выведется.

Если нужно написать переменную в условие, то ёё (переменную) писать без @@

1. @@foreach – Повторить блок html n-е количество раз
2. Ещё несколько функций. Можно почитать в описании плагина.

**Примеры**:

|  |  |
| --- | --- |
| @@include | **--** пути до компонентов **--** значения для переменных компонентов |
| @@name  +  @@if | **--** переменные |
| @@foreach |  |

1. Функции (в общем файле common.js), которые будут нужны в section, нужно выносить за пределы $(document).ready(function () {}) (document.addEventListener("DOMContentLoaded", () => {}))
2. Все скрипты section, в свою очередь, нужно оборачивать в $(document).ready(function () {}) (document.addEventListener("DOMContentLoaded", () => {}))
3. Нужно учитывать, что стили section не будут главнее общего файла стилей (common.css). Если нужно перезаписать стили общего файла (для section), тогда нужно использовать вложенность.