



помогаем бизнесу
продавать онлайн

ООО «Редлайн»
г. Томск, ул. Карла Маркса 17а, 4 этаж
Тел.: 8 (3822) 99-41-30
info@redlg.ru
<https://redlg.ru>

Требования к frontend-разработке

Дата **28.05.2018**

Git

Вёрстка должна храниться в гит-репозитории, доступ к которому предоставит менеджер (репозиторий должен быть на гит-сервере Редлайн).

Подробнее о правилах работы с Git:

- Каждая задача - отдельный коммит
- Коммит не должен приводить к неработоспособности проекта
- Проект должен содержать dev и master ветки

Структура проекта

Структура верстки должна придерживаться данного шаблона:

- assets - папка для ресурсов
 - styles.min.css - минифицированный, скомпилированный файл для продакшена
 - bundle.min.js - минифицированный, скомпилированный файл для продакшена
 - fonts - шрифты
 - img - картинки распределены по блокам(папкам) в которых они используются
 - header
 - logo.png
 - footer
 - visa.png
- src - исходники
 - js
 - class - классы, например класс для работы с попапами
 - function - отдельные функции, например для парсинга строк
 - module - модули, например модуль для страницы корзины, или модуль для отправки заявок на сервер
 - scss
 - [global.scss](#) - входная точка сборки стилей, сюда импортируются остальные стили
 - base - базовые файлы
 - [base.scss](#)
 - [fonts.scss](#)
 - lib - файлы подключаемых библиотек, например слайдеров
 - [normalize.scss](#)
 - mixin
 - [media.scss](#)
 - modifier - модификаторы для быстрого пользования(не злоупотреблять)

- [margin.scss](#)
- [padding.scss](#)
- module - структурные блоки
 - footer.scss
 - header.scss
- util
 - [grid.scss](#)
- svg - иконки которые используются в проекте, иконки собираются в шрифт с помощью сервиса <https://icomoon.io/app/>
 - vkontakte.svg
 - facebook.svg

Разметка

- Использовать табы, 1 таб = 4 пробела
- Использовать семантические теги и элементы HTML5.
- Необходимо указывать доктайп - <!DOCTYPE html>.
- Код должен быть валидным и проверенным на наличие критических ошибок.
 - <https://validator.w3.org/>
- Семантичность верстки(например правильное использование тегов section или label).
 - Плохой вариант
 - Имя
 - <input name="name" value="Иван"/>
 - Хороший вариант
 - <label>Имя</label>
 - <input name="name" value="Иван"/>
- Именованние классов по БЭМ (Two dashes style - block__element--modifier).
 - [BEM](#)
- Избегать миксования классов
 - Плохой вариант
 - class="news-item__photo left"
 - class="news-item__photo right"
 - Хороший вариант
 - class="news-item__photo--left".
- Если имеется список элементов(например новостей) и каждый элемент имеет дочерние элементы, то элемент должен быть блоком.
 - Плохой вариант
 - news
 - news__item
 - news__title
 - news__photo
 - news__link

- Хороший вариант
 - news
 - news-item
 - news-item__title
 - news-item__photo
 - news-item__link

Стили

- Использовать табы, 1 таб = 4 пробела
- Использовать SCSS.
- Каждый блок должен быть в отдельном файле, в папке module.
- Использовать переменные для повторяющихся значений например: цвета, медиа поинты и тд.
- Запрещается использовать float для позиционирования блоков.
- Не злоупотреблять флексом, например: если блоки идут в колонку сверху вниз, то не нужно использовать **display: flex, flex-direction: column**, когда можно обойтись обычным **display: block**;

Скрипты

- Использовать табы, 1 таб = 4 пробела
- Модульный подход - разделять логические части на разные файлы.
- Если есть готовое решение(библиотека), рекомендуется использовать её.
- С целью отделения представления от логики, классы-селекторы для запуска любых скриптов должны иметь постфикс --js. Эти классы не должны использоваться в стилях. Например:
 - news-slider news-slider--js
- Комментируйте код
- Соглашение об именовании
 - Классы / Объекты с функциями
 - UpperCamelCase (*PascalCase*)
 - Функции
 - lowerCamelCase
 - Переменные
 - lowerCamelCase
- Для установки библиотек использовать npm или yarn
- Для подключения библиотек использовать import
 - [gulpfile](#)
- Библиотеки рекомендуемые к использованию
 - <https://www.npmjs.com/package/jquery>
 - <https://www.npmjs.com/package/swiper> - слайдер
 - <https://www.npmjs.com/package/superagent> - взаимодействие с сервером

- <https://www.npmjs.com/package/hotkeys-js> - горячие клавиши на клавиатуре
- <https://www.npmjs.com/package/dropzone> - для загрузки файлов на сервер
- <https://www.npmjs.com/package/moment> - работа с датами
- <https://www.npmjs.com/package/lodash>
- <https://www.npmjs.com/package/imask> - маска для инпутов
- <https://www.npmjs.com/package/air-datepicker> - календарь(input)

Картинки/иконки

- JPG должен использоваться для больших иллюстраций и фотографий благодаря эффективному сжатию и оптимизации их размера (tinypng). Для текста, логотипов, графиков и других небольших картинок - используйте PNG.
- по возможности использовать SVG шрифты(иконки из figma легко интегрируются в шрифт с помощью <https://icomoon.io/app/#/select>)

Адаптивность

- [media.scss](#)
- Медиапоинты работают до следующего медиапоинта вверх. Например:
 - определены следующие разрешения: 320, 768, 1024, 1280, 1920
 - дизайн для 320 работает на промежутке 0 - 767
 - дизайн для 768 работает в промежутке 768 - 1023
 - дизайн для 1024 работает в промежутке 1024 - 1280
 - и т.д.

Поддержка браузеров

По браузерам обязательно уточняйте у менеджера, но стандартная поддержка браузеров:

- Chrome, Firefox, Safari, Opera, Microsoft Edge
- Мобильные браузеры
- Поддержка IE осуществляется только по требованию клиента