Оглавление

[Модуль 1. Основные составляющие фронтенда. 3](#_Toc195710370)

[Материалы 3](#_Toc195710371)

[Раздел 1.1. Изучение основ работы с редактором исходного кода, например, VS Code. 3](#_Toc195710372)

[Раздел 1.2. Изучение основ работы с хранилищем исходного кода Git. 4](#_Toc195710373)

[Раздел 1.3.Изучение основ HTML (содержание и разметка). 4](#_Toc195710374)

[Раздел 1.4. Изучение основ CSS (внешний вид, позиционирование). 4](#_Toc195710375)

[Раздел 1.5. Изучение основ JavaScript (логика). 5](#_Toc195710376)

[Итоговое задание 6](#_Toc195710377)

[Модуль 2. Использование библиотек в Javascript, на основе JQuery. 7](#_Toc195710378)

[Материалы 7](#_Toc195710379)

[Раздел 2.1. Изучение основ использования JQuery. 7](#_Toc195710380)

[Раздел 2.2. Изучение основ использования JQuery. События и анимации. 8](#_Toc195710381)

[Раздел 2.3. Изучение основ использования JQuery. Работа с Ajax. 8](#_Toc195710382)

[Раздел 2.3. Изучение основ использования JQuery. Плагины. 8](#_Toc195710383)

[Итоговое задание 9](#_Toc195710384)

[Модуль 3. Работа с NodeJS. Пакеты. Основы сборки и создания пакетов. 10](#_Toc195710385)

[Материалы 10](#_Toc195710386)

[Раздел 3.1. Основы NodeJS. Установка NVM и установка нескольких версий NodeJS. 10](#_Toc195710387)

[Раздел 3.2. Создание пакета. Подключение сторонних пакетов. Основы сборки пакета и выполнение команд. 11](#_Toc195710388)

[Итоговое задание 12](#_Toc195710389)

[Модуль 4. Компонентная архитектура. Основы React JS. 13](#_Toc195710390)

[Материалы 13](#_Toc195710391)

[Раздел 4.1. Введение в React. Основы React. 13](#_Toc195710392)

[Раздел 4.2. Классовый подход в ReactJS. 14](#_Toc195710393)

[Раздел 4.3. Функциональный подход в ReactJS. 14](#_Toc195710394)

[Раздел 4.4. Основы Webpack. 15](#_Toc195710395)

[Раздел 4.5. Основы Sass. 15](#_Toc195710396)

[Раздел 4.5. Основы TypeScript. 16](#_Toc195710397)

[Раздел 4.6. HOC (Higher-Order Components) в ReactJS. Сравнение и общие выводы об использовании разных подходов. 16](#_Toc195710398)

[Раздел 4.7. Расширенные возможности ReactJS. Оптимизация. Пользовательские «хуки». 17](#_Toc195710399)

[Итоговое задание 18](#_Toc195710400)

[Модуль 5. Подготовка окружения React-приложения. 19](#_Toc195710401)

[Материалы 19](#_Toc195710402)

[Раздел 5.1. Основные настройки VS Code. 19](#_Toc195710403)

[Раздел 5.2. Настройка Prettier. 19](#_Toc195710404)

[Раздел 5.2. Настройка ESLint. 20](#_Toc195710405)

[Итоговое задание 21](#_Toc195710406)

[Модуль 6. Основы по работе с API. 22](#_Toc195710407)

[Материалы 22](#_Toc195710408)

[Раздел 6.1. Основы работы с API. 22](#_Toc195710409)

[Раздел 6.2. Инструменты тестирования и проверки стороннего API. 23](#_Toc195710410)

[Итоговое задание 24](#_Toc195710411)

[Модуль 7. Планы на развитие обучения и дальнейшего самостоятельного обучения. 25](#_Toc195710412)

[Материалы 25](#_Toc195710413)

[Раздел 7.1. Дополнительные инструменты и библиотеки. 26](#_Toc195710414)

[Раздел 7.2. Основы принципов в программировании. 26](#_Toc195710415)

[Раздел 7.3. Паттерны проектирования. Структурирование кода. 27](#_Toc195710416)

[Раздел 7.4. Основы архитектуру приложения. Принципы построения библиотеки компонентов 27](#_Toc195710417)

[Итоговое задание 29](#_Toc195710418)

# Модуль 1. Основные составляющие фронтенда.

В первом модуле необходимо пройти базовые элементы работы с фронтенд-разработкой. Этот модуль является «нулевым» и является не обязательным, но рассчитан на отсутствие знаний и может быть частично, или полностью пропущен ментором, в зависимости от уровня стажируемого.

## Материалы

* <https://learn.javascript.ru/>
* <https://www.schoolsw3.com/css/index.php>
* <https://www.schoolsw3.com/html/index.php>
* <https://www.schoolsw3.com/js/index.php>
* Изучаем HTML, XHTML и CSS 2-е изд.
* Новая большая книга CSS Макфарланд Дэвид | Макфарланд Дэвид
* <https://learn.microsoft.com/ru-ru/shows/visual-studio-code/learn-visual-studio-code-in-7min-official-beginner-tutorial>
* <https://code.visualstudio.com/docs/getstarted/getting-started>
* <https://habr.com/ru/articles/490754/>
* <https://metanit.com/web/html5/>
* <https://metanit.com/web/javascript/>
* <https://habr.com/ru/articles/891050/>
* <https://smartiqa.ru/courses/git/lesson-1>
* <https://www.geeksforgeeks.org/git-tutorial/>
* <https://www.w3schools.com/git/>
* Изучаем программирование на JavaScript | Фримен Эрик, Робсон Элизабет

## Раздел 1.1. Изучение основ работы с редактором исходного кода, например, VS Code.

* Структура файлов и способы хранения проекта в папке.
* Загрузка VS Code и базовая настройка.
* Работа с плагинами и настройкой среды.

Задание

установить у себя VS Code.

## Раздел 1.2. Изучение основ работы с хранилищем исходного кода Git.

* Основы работы с Git.
* Основы работы с GitHub.
* Регистрация и создание своего репозитория.
* Основы работы с репозиторием через клиент, например, SourceTree и через консоль.

Задание

Зарегистрироваться в GitHub. Создать свой первый репозиторий, добавить свой первый коммит.

## Раздел 1.3.Изучение основ HTML (содержание и разметка).

* История HTML. Синтаксис HTML. Основные правила верстки.
* Основы построения страницы, структура HTML-документа.
* Комментирование блоков;
* Теги заголовочного раздела документа;
* Стандарты HTML.

Задание

Создать репозиторий, добавить HTML страницу в репозиторий по условию ментора.

## Раздел 1.4. Изучение основ CSS (внешний вид, позиционирование).

* Настройка файла css.
* синтаксис таблицы стилей;
* способы внедрения в документ;
* селекторы, псевдоклассы и псевдоэлементы;
* основные свойства;
* наследование и группировка свойств;
* Основы написания файла, визуальное воздействие на оформление элементов. CSS3 и современные возможности стилей.

Задание

Добавить оформление сайту из Раздела 1.3. Обновить свой репозиторий.

## Раздел 1.5. Изучение основ JavaScript (логика).

* Написание скриптов js.
* синтаксис языка;
* управляющие конструкции;
* функции и функциональные выражения;
* контекст вызова this;
* области видимости;
* «строгий режим»;
* наследование в функциональном стиле;
* наследование в прототипном стиле;
* классы;
* стандартные объекты JS;
* объекты надстройки браузера;
* DOM;
* механизмы коллбеков;
* отложенное исполнение кода и исполнение через «равные» промежутки времени.
* Основы написания файла, добавление поведения на сайте.
* Основные конструкции. Работа с основами языка JavaScript.

Задание

Добавить поведение сайту из Раздела 1.4. Обновить свой репозиторий.

Итоговое задание

Предлагается выполнить разработку сложного калькулятора с возможность ввода разных операций и типов данных.  
Основные операции: +, -, \*, /;  
Дополнительные операции: сложение массивов, отрицание значения, вычисление среднего арифметического(количество полей от 2 до 10, с соответствующими настройками в интерфейсе);

\*Сложные операции: Сложение изображений, обрезание изображение по маске, замена в тексте символов.   
Используя только «чистый» JS, HTML, CSS.

# Модуль 2. Использование библиотек в Javascript, на основе JQuery.

Во втором модуле необходимо пройти базовые элементы работы с фронтенд-разработкой с использованием библиотек на примере работы с JQuery. Этот модуль важен, так как сталируемый может работать с кодом Программного комплекса.

## Материалы

* <https://tproger.ru/articles/jquery-cheatsheet>
* <https://jquery.com/>
* <https://yourtodo.life/ru/posts/osnovyi-jquery-polnyij-gid-dlya-nachinayuschih/>
* Изучаем работу с jQuery: Бенедетти, Крэнли

## Раздел 2.1. Изучение основ использования JQuery.

* Что такое jQuery
* Основные принципы
* Установка и подключение
* Подключение к HTML-документу
* Основы синтаксиса
* Селекторы
* Псевдоклассы и псевдоэлементы
* Атрибутные селекторы
* Манипуляция элементами DOM
* Изменение элементов
* Обработка тайм-аутов
* Обработка кэширования

Задание

Переписать задание из первого модуля через JQuery. Обновить свой репозиторий.

## Раздел 2.2. Изучение основ использования JQuery. События и анимации.

* События
* Дополнительные методы для работы с событиями
* Эффекты и анимация
* Пользовательские анимации

Задание

Переписать задание из первого модуля через JQuery. Обновить свой репозиторий.

## Раздел 2.3. Изучение основ использования JQuery. Работа с Ajax.

* Работа с AJAX
* Обработка ответов сервера
* Ошибки и отладка AJAX-запросов

Задание

Переписать задание из первого модуля через JQuery. Обновить свой репозиторий.

## Раздел 2.3. Изучение основ использования JQuery. Плагины.

* Плагины
* Что такое плагины
* Поиск готовых плагинов jQuery

Задание

Переписать задание из первого модуля через JQuery. Обновить свой репозиторий.

Итоговое задание

Предлагается выполнить переработку задания из первого модуля или разработку при пропуске первого модуля. Предлагается выполнить разработку сложного калькулятора с возможность ввода разных операций и типов данных.  
Основные операции: +, -, \*, /;  
Дополнительные операции: сложение массивов, отрицание значения, вычисление среднего арифметического(количество полей от 2 до 10, с соответствующими настройками в интерфейсе);

\*Сложные операции: Сложение изображений, обрезание изображение по маске, замена в тексте символов.   
Используя только «чистый» JS, JQuery, HTML, CSS. Обеспечив разделение файлов.

# Модуль 3. Работа с NodeJS. Пакеты. Основы сборки и создания пакетов.

В третьем модуле необходимо пройти базовые элементы по вопросам работы с NodeJS и создании пакетов, загрузка сторонних пакетов. Компоновка пакетов и сборки. Отладка скриптов в VS Code.

## Материалы

* <https://habr.com/ru/companies/ruvds/articles/422893/>
* <https://practicum.yandex.ru/blog/rukovodstvo-po-node-js-dlya-nachinayushchih/>
* <https://nodejsdev.ru/guides/freecodecamp/>
* <https://metanit.com/web/nodejs/>
* <https://github.com/nvm-sh/nvm>
* <https://betterstack.com/community/guides/scaling-nodejs/nvm-explained/>
* <https://www.geeksforgeeks.org/how-to-install-and-use-nvm-on-windows/>
* <https://306.antroot.ru/nvm>
* <https://nodejs.org/en>
* <https://skillbox.ru/media/code/chto-takoe-npm-gayd-po-menedzheru-paketov-nodejs/>

## Раздел 3.1. Основы NodeJS. Установка NVM и установка нескольких версий NodeJS.

* Знакомство с NVM. Установка
* Знакомство с NPM
* Запуск скриптов
* Файл package.json
* Файл package-lock.json

Задание

Разворачивание nvm, установка nodejs определенной версии. Создание своего первого пакета. Добавление скриптов.

## Раздел 3.2. Создание пакета. Подключение сторонних пакетов. Основы сборки пакета и выполнение команд.

* Загрузка и установка пакетов
* Установка зависимостей
* Автоматическое добавление зависимостей
* Обновление пакетов
* Загрузка пакетов определённых версий
* Запуск скриптов и групповое выполнение скриптов
* Настройка отладки выполнения в VS Code

Задание

Установка сторонних пакетов, группировка пакетов для решения задача.

Итоговое задание

Предлагается выполнить разработать пользовательского пакет, решающего какую-нибудь задачу в пространстве nodeJS.

# Модуль 4. Компонентная архитектура. Основы React JS.

В четвертом модуле необходимо пройти базовые элементы по вопросам перехода от Реального DOM к виртуальному DOM и основы компонентного подхода. Этот модуль важен, так как стажируемый может работать с кодом подсистем СУНТД.

## Материалы

* <https://habr.com/ru/companies/ruvds/articles/437990/>
* <https://www.geeksforgeeks.org/reactjs-class-components/>
* <https://reactdev.ru/types/050/>
* <https://habr.com/ru/companies/ruvds/articles/428077/>
* Основы TypeScript | Фримен Адам
* Эффективный TypeScript: 83 способа улучшить код. 2-е изд. | Вандеркам Дэн
* <https://metanit.com/web/typescript/1.1.php>
* Стефанов Стоян. React. Быстрый старт, 2-е изд.
* <https://habr.com/ru/articles/514838/>
* <https://tproger.ru/translations/webpack-basics>
* <https://vc.ru/dev/418917-krupnyi-gaid-po-webpack>
* <https://sass-scss.ru/>
* <https://gitverse.ru/blog/articles/development/516-osnovy-sassscss-nachinaem-rabotu-s-preprocessorami>
* <https://sass-lang.su/guide>
* <https://tproger.ru/translations/complete-sass-guide>
* <https://metanit.com/web/react/>
* <https://tproger.ru/articles/podgotovka-okruzhenija-react-prilozhenija-vscode-prettier-eslint-stylelint-husky>

## Раздел 4.1. Введение в React. Основы React.

* Основы JSX
* Babel-cli. Процесс транспиляции.
* React Developer Tools
* Рендеринг элементов
* Компоненты
* Свойства Props
* События
* Составные компоненты

Задание

Переписать задание из первого модуля через React, NodeJS, Webpack, JS, SCSS. Обновить свой репозиторий. Задача ставится последовательно на несколько разделов.

## Раздел 4.2. Классовый подход в ReactJS.

* Синтаксис и основы написания в классовых компонентах
* Передача пропсов в классовых компонентах
* Работа с состоянием в классовых компонентах
* События и цикл жизни скрипта в классовых компонентах
* Рендеринг компонента в классовых компонентах

Задание

Переписать задание из первого модуля через React, NodeJS, Webpack, JS, SCSS. Обновить свой репозиторий. Задача ставится последовательно на несколько разделов. Компонент на классовом подходе.

## Раздел 4.3. Функциональный подход в ReactJS.

* Синтаксис и основы написания в функциональных компонентах
* Передача пропсов в функциональных компонентах
* Работа с состоянием в функциональных компонентах
* События и цикл жизни скрипта в функциональных компонентах
* Рендеринг компонента в функциональных компонентах
* Понятие «хуков». Основы работы с «хуками».

Задание

Переписать задание из первого модуля через React, NodeJS, Webpack, JS, SCSS. Обновить свой репозиторий. Задача ставится последовательно на несколько разделов. Компонент на функциональном подходе.

## Раздел 4.4. Основы Webpack.

* Что такое Webpack
* Зачем он нужен
* Установка и первоначальная настройка
* Как работает Webpack
* Режимы работы
* Загрузчики
* Плагины

Задание

Переписать задание из первого модуля через React, NodeJS, Webpack, JS, SCSS. Обновить свой репозиторий. Задача ставится последовательно на несколько разделов. Настройка Webpack.

## Раздел 4.5. Основы Sass.

* Что такое Sass
* Зачем он нужен
* Установка и первоначальная настройка
* Переменные
* Вложенность
* Составляющие
* Примеси (Миксины)
* Расширение/Наследование
* Операторы

Задание

Переписать задание из первого модуля через React, NodeJS, Webpack, JS, SCSS. Обновить свой репозиторий. Задача ставится последовательно на несколько разделов. Настройка Sass. Настройка Webpack.

## Раздел 4.5. Основы TypeScript.

* Установка и разворачивание. Настройка компиляции.
* Переменные и константы
* Типы данных
* Функции
* Тип функции и стрелочные функции
* Объекты
* Массивы
* Кортежи
* Неопределенный набор и наполнение параметров функции
* Перечисление enum
* Классы
* Наследование. Абстрактные классы, методы и поля
* Модификаторы доступа. Методы доступа get и set
* Статические поля и методы
* Интерфейсы
* Преобразование типов
* Обобщения
* Миксины
* Создание и подключение модулей
* Работа с модулями

Задание

Переписать задание из текущего модуля через React, NodeJS, Webpack, TS, SCSS. Обновить свой репозиторий. Задача ставится последовательно на несколько разделов. Настройка TypeScript. Настройка Webpack.

## Раздел 4.6. HOC (Higher-Order Components) в ReactJS. Сравнение и общие выводы об использовании разных подходов.

* Синтаксис и основы написания HOC компонентов
* Сравнительный анализ разных подходов
* Типизация компонентов
* Перевод проекта на работу через типы и интерфейсы
* Процесс транспиляции и использование разных стандартов языка

Задание

Переписать задание из текущего модуля через React, NodeJS, Webpack, TS, SCSS. Обновить свой репозиторий. Задача ставится последовательно на несколько разделов. Добавить HOC-компонент.

## Раздел 4.7. Расширенные возможности ReactJS. Оптимизация. Пользовательские «хуки».

* Использование сложных «хуков» useRef, useMemo, useContext и др.
* Рекомендации по их применению
* Основные возможности по сокращению «лишних» ре-рендеров, исключение лишнего пересчета.
* События в React.
* «Небезопасные» части компонента, модификация элемента через JS.
* Пользовательские «хуки». Создаем свои «хуки»
* Анимации в ReactJS
* Отложенная загрузка компонентов

Задание

Переписать задание из текущего модуля через React, NodeJS, Webpack, TS, SCSS. Обновить свой репозиторий. Задача ставится последовательно на несколько разделов. Добавление функционала по оптимизации компонентов.

Итоговое задание

Предлагается выполнить разработать SPA-приложения с несколькими компонентами, содержащими список, кнопки, диалоговое окно и вкладки. Например, ToDo лист с особенностями, где можно что-то добавлять, удалять, помечать, раскрывать, скрывать и редактировать часть чего-то общего.

# Модуль 5. Подготовка окружения React-приложения.

В пятом модуле необходимо пройти базовые элементы по вопросам настройки окружения, удобств разработки и обеспечению чистоты кода, а также первичного тестирования. Этот модуль важен, так как стажируемый должен проводить разработку качественно и следить за «чистотой» кода, который он пишет.

## Материалы

* <https://tproger.ru/translations/setting-up-eslint-and-prettier>
* <https://gitverse.ru/blog/articles/development/344-eslint-kak-uluchshit-kachestvo-koda-javascript-i-typescript>
* <https://dev.to/ivannicksimeonov/vs-code-setup-for-frontend-devs-577f>
* <https://habr.com/ru/companies/kts/articles/779670/>
* <https://learnjs.ru/journal/nastroyka-vs-code/>
* <https://ru.hexlet.io/blog/posts/vscode-js-setup>

## Раздел 5.1. Основные настройки VS Code.

* Использование settings.json — настройки VSCode для проекта.
* Использование extensions.json— настройки VSCode для проекта.
* Полезные расширения в VS Code
* Поиск и применение расширений
* Настройка отладки launch.json— настройки VSCode для проекта.

Задание

Настроить расширение для проекта.

## Раздел 5.2. Настройка Prettier.

* Форматирование кода с помощью Prettier в Visual Studio Code
* Настройка рабочей среды
* Автоформатирование
* Настройка собственных правил форматирования

Задание

Настроить Prettier в своем проекте.

## Раздел 5.2. Настройка ESLint.

* Что такое Lint?
* ESLint и расширение для VS Code
* Настройка ESLint
* Установка и подключение дополнительных плагинов ESLint
* eslint-plugin-react
* eslint-plugin-react-hooks

Задание

Настроить ESLint в своем проекте.

Итоговое задание

Предлагается выполнить разработать SPA-приложения с несколькими компонентами, содержащими список, кнопки, диалоговое окно и вкладки. Например, реализовать Инстаграмм для анекдотов, с возможностью работы со списками анекдотов, добавлять свои, помечать избранные и выполнять группировку по жанрам, с возможностью лайкнуть\дизлайкнуть.

# Модуль 6. Основы по работе с API.

В шестом модуле необходимо пройти базовые элементы по вопросам работы с API и инструментами по взаимодействию с ним.

## Материалы

* <https://habr.com/ru/articles/768752/>
* <https://htmlacademy.ru/blog/js/javascript-api>
* <https://sky.pro/wiki/javascript/rabota-s-api-osnovy-i-primery/>
* <https://skillbox.ru/media/code/chto_takoe_api/>
* <https://skillbox.ru/media/code/postman-chto-eto-takoe-i-kak-im-polzovatsya/>
* <https://skillbox.ru/media/code/kak-rabotat-s-curl-sintaksis-i-osnovnye-komandy/>
* <https://curl.se/docs/>
* <https://losst.pro/kak-polzovatsya-curl>
* <https://gitverse.ru/blog/articles/development/430-chto-takoe-curl-i-kak-im-polzovatsya>
* <https://sky.pro/media/chto-takoe-curl-i-kak-polzovatsya-utilitoj/>
* <https://coursme.com/article/api-na-javascript-poshagovoe-rukovodstvo-dlya-nachinayushchih-s-primerami>
* <https://result.school/roadmap/frontend/article/work-with-api>

## Раздел 6.1. Основы работы с API.

* Что такое API и как с ним работать?
* Основы работы с API
* Основные виды API
* Принципы работы API
* Как устроен запрос и ответ
* Форматы данных
* Интеграция API в коде
* Чтение документации API
* Обработка ошибок и исключений

Задание

Найти публичное API и проанализировать его с помощью изученных знаний.

## Раздел 6.2. Инструменты тестирования и проверки стороннего API.

* Что такое Postman и зачем он нужен
* Как устроен Postman
* Как пользоваться Postman
* Как отправить запрос
* Что такое Curl и зачем он нужен
* Как установить и запустить Curl
* Как пользоваться Curl

Задание

Найти публичное API и проанализировать его с помощью изученных знаний. Реализовать отображение данных в соответствии с типом данных и специфики области.

Итоговое задание

Предлагается выполнить разработать SPA-приложения с несколькими компонентами, содержащими взаимодействие с разными API и группирует представление из нескольких API в интерфейсе.

# Модуль 7. Планы на развитие обучения и дальнейшего самостоятельного обучения.

В седьмом модуле необходимо пройти базовые элементы по вопросам дальнейшего развития специалиста и тем для самостоятельного обучения.

## Материалы

* <https://habr.com/ru/articles/768752/>
* <https://kedu.ru/press-center/articles/info-printsipy-programmirovaniya-osnovnye-pravila-i-podkhody-v-razrabotke-programmnogo-obespecheniya/>
* <https://www.it-courses.by/basic-programming-principles/>
* <https://www.geeksforgeeks.org/7-common-programming-principles-that-every-developer-must-follow/>
* <https://habr.com/ru/companies/vk/articles/325492/>
* <https://blog.skillfactory.ru/glossary/pattern/>
* <https://habr.com/ru/companies/simbirsoft/articles/729066/>
* <https://habr.com/ru/articles/340384/>
* <https://storybook.js.org/docs>
* <https://thecode.media/arhitektura-programmnogo-obespecheniya/>
* <https://practicum.yandex.ru/blog/arhitektura-programmnogo-obespecheniya/>
* <https://www.ramotion.com/blog/single-page-application-architecture/>
* <https://blog.skillfactory.ru/glossary/single-page-application/>
* <https://thecode.media/spa/>
* <https://atwinta.ru/material/blog/single-page-application-spa/>
* <https://codelab.pro/osnovnye-patterny-programmirovaniya-dlya-novichkov/>
* <https://metanit.com/sharp/patterns/1.1.php>
* <https://redux.js.org/tutorials/essentials/part-1-overview-concepts>
* <https://habr.com/ru/companies/otus/articles/863002/>
* <https://htmlacademy.ru/blog/js/redux>
* <https://proglib.io/p/redux-basis>
* <https://metanit.com/web/nodejs/2.6.php>
* <https://www.geeksforgeeks.org/node-js-nodemon-module/>
* <https://blog.skillfactory.ru/biblioteki-i-freymvorki-javascript/>
* <https://jestjs.netlify.app/ru/docs/getting-started>
* <https://habr.com/ru/companies/ruvds/articles/427293/>
* <https://proglib.io/p/js-design-patterns>
* <https://bool.dev/blog/detail/gof-design-patterns>
* <https://habr.com/ru/companies/rshb/articles/680770/>
* <https://www.freecodecamp.org/news/javascript-design-patterns-explained/>
* <https://wiki.merionet.ru/articles/polnoe-rukovodstvo-po-sablonam-proektirovaniia-javascript>
* Рефакторинг кода на JavaScript. Улучшение проекта существующего кода | Фаулер Мартин
* JavaScript и Node.js для веб-разработчиков | Дронов Владимир Александрович, Прохоренок Николай Анатольевич
* Рецепты TypeScript | Баумгартнер Стефан
* JavaScript в примерах и задачах | Васильев Алексей Николаевич
* Эффективный TypeScript: 83 способа улучшить код. 2-е изд. | Вандеркам Дэн
* Профессиональный TypeScript. Разработка масштабируемых JavaScript-приложений | Черный Борис
* Тестирование JavaScript | да Коста Лукас

## Раздел 7.1. Дополнительные инструменты и библиотеки.

* Основы фреймворка для тестирования JestJS и особенности
* Основы библиотеки ReduxJS и особенности
* Основы Axios и особенности
* Основы Lodash и особенности
* Основы Uuid и особенности
* Основы инструмента сборки Vite и особенности
* Основы инструмента Nodemon и особенности
* Основы Storybook и особенности
* Сторонние тулзы для примера

Задание

Исследовать и установить в проекте дополнительные инструменты и библиотеки, упрощающие разработку кода во фронтенде.

## Раздел 7.2. Основы принципов в программировании.

* Принцип «простоты» KISS
* Принцип «отсутствия повторов» DRY
* Принципы SOLID
* Разделение интересов или основы model-view-controller (MVC)
* Принцип возможно это может пригодиться YAGNI
* Принцип избежания преждевременной оптимизации
* Рефакторинг, как лучшее оружие, хотя ...
* Чистый код > Умный код

Задание

Изучить принципы и проанализировать свой предыдущий код на тему возможного использования принципов или поиск мест, нарушающих принципы.

## Раздел 7.3. Паттерны проектирования. Структурирование кода.

* Шаблоны проектирования, а что это?
* Виды шаблонов проектирования
* Порождающие
* Структурные
* Поведенческие
* Для чего нужны паттерны
* Кто пользуется паттернами проектирования
* Что такое ООП
* Понятие паттерна и связь с ООП

Задание

Самостоятельно изучить основные аспекты паттернов и попробовать применить в некоторых задачах паттерны. Провести анализ возможного переиспользования паттернов в своих проектах.

## Раздел 7.4. Основы архитектуру приложения. Принципы построения библиотеки компонентов

* Заметки об основах программной архитектуры
* Общая концепция архитектуры программного обеспечения
* Архитектурные стили
* Софт скиллз
* Зачем нужна архитектура в проекте
* Модульность
* Инкапсуляция
* Разделение ответственности
* Повторное использование
* Масштабируемость
* Типы архитектурных стилей
* Монолитная архитектура
* Микросервисная архитектура
* Сервис-ориентированная архитектура
* Архитектура вертикального среза

Задание

Проанализировать свой код, попробовать решить задачи модульности, масштабируемости, повторного использования.

Итоговое задание

Предлагается выполнить разработать библиотеку собственных компонентов, а также SPA-приложение, в котором эти компоненты будут использоваться.