# React.js

Сперва разберем основы. React — это не очередной MVC-фреймворк или какой-то другой фреймворк. Это просто библиотека для рендеринга ваших представлений (проще говоря - библиотека создания интерфейсов страниц, которые меняются во времени). Если вы пришли из мира MVC, то стоит понять, что React — это только «V», а «M» и «C» придётся поискать где-то ещё. Для реакт нужны React.js и React DOM.

## JSX

Компоненты Реакта типично пишутся в JSX (JavaScript exntension). Синтах JSX позволяет брать в кавычки HTML и использовать синтаксис HTML тегов для создания подкомпонентов. HTML синтаксис обрабатывается в JS вызовах Реакт библиотеки.

## NPM

NPM - это очень большое общественное хранилище, где умные люди постят код и зависимости в виде модулей.

Browserify - Это инструмент, который позволяет связать CommonJS описания зависимостей для файлов, которые могут быть запущены в браузере. Он был создан, потому что большинство людей публикуют эти зависимости в NPM.

# Функциональное программирование в JS

Flow — инструмент для проверки статической типизации, сделанный парнями из Facebook. Они написали его на OCaml, так как функциональное программирование является удивительно крутым.

Ramda - библиотека, которая позволяет нам использовать функциональное программирование на простом JavaScript.

# AJAX

AJAX - это набор приемов веб-разработки использующий много веб технологий на клиентской стороне чтобы создавать асинхронные веб приложения.

С помощью Аджакса веб-приложения могут отправлять и получать данные от сервера асинхронно (в бекграунде) без вмешательства в дисплей и поведения на текущей странице.

AJAX базируется на двух основных принципах:

1. использование технологии динамического обращения к серверу «на лету», без перезагрузки всей страницы полностью, например с использованием XMLHttpRequest (основной объект);
   1. через динамическое создание дочерних фреймов;
   2. через динамическое создание тега <script>.
   3. через динамическое создание тега <img>, как это реализовано в Google Analytics.
2. использование DHTML для динамического изменения содержания страницы ; (DHTML = CSS, DOM, JS)

Стандартный подход:

1. Пользователь заходит на веб-страницу и нажимает на кнопку;
2. Браузер формирует и отправляет запрос серверу;
3. В ответ сервер генерирует новую веб-страницу и отправляет браузеру. После чего браузер перезагружает страницу

Подход с AJAX:

1. Пользователь нажимает кнопку;
2. Скрипте на JS определяет какая инфа небходима для обновления страницы
3. Браузер отправляет соответствующий запрос на сервер.
4. Сервер возвращает только часть документа, которая нужна.
5. Скрипт вносит изменения с учетом полученной информации без польной перезагрузки страницы.

# XMLHttpRequests

XMLHttpRequest (XMLHTTP, XHR) — API, доступный в скриптовых языках браузеров, таких как JavaScript. Использует запросы HTTP или HTTPS напрямую к веб-серверу и загружает данные ответа сервера напрямую в вызывающий скрипт. Информация может передаваться в любом текстовом формате, например, в XML, HTML или JSON. Позволяет осуществлять HTTP-запросы к серверу без перезагрузки страницы.  
  
XMLHTTP является важной составляющей технологии AJAX (Asynchronous JavaScript And XML), используется многими сайтами для создания динамичных, быстро реагирующих на запросы пользователя приложений.

Fetch это имя нативной реализации для выполнения XMLHttpRequests.

AJAX это просто запросы XMLHttpRequest. А Fetch позволяет делать AJAX на основе промисов, которые затем можно резолвить, чтобы избежать callback hell.

Bluebird, Request и Axios - это библиотеки для выполнения XMLHttpRequests, которые возвращают промисы.

Мы больше не используем «J» в 2016. Просто используйте Fetch polyfill или Bluebird, Request или Axios. Затем управляй промисами с async, await и Бац!, у тебя правильный поток управления.

# Разное

## Актуальные вещи

React, JSX, Babel, AMD или CommonJS, NPM, Browserify,

## Неактуальные вещи

JQuery