# Урок 1-2

Домашнее задание:

1. Написать [функцию](https://learn.javascript.ru/function-basics) pow, аналогичную [Math.pow](http://javascript.ru/Math-pow), которая должна возводить указанное число в указанную степень. Указать число и степень пользователь должен через команду [prompt](https://learn.javascript.ru/uibasic). Результат выполнения функции вывести в консоль. Работать с целыми числами, большими, меньшими, и равными нулю. Бесконечности можно не обрабатывать
2. Создать программу, которая будет выполнять следующие действия:

* [Циклом](https://learn.javascript.ru/array#%D0%BF%D0%B5%D1%80%D0%B5%D0%B1%D0%BE%D1%80-%D1%8D%D0%BB%D0%B5%D0%BC%D0%B5%D0%BD%D1%82%D0%BE%D0%B2) заполнить [массив](https://learn.javascript.ru/array) с помощью команды prompt в котором будет список из 5-ти любых имен
* Потом вывести с помощью prompt сообщение с просьбой ввести имя пользователя
* Введенное имя, циклом [сравнивать](https://learn.javascript.ru/ifelse) с именами в массиве
* Если нет совпадения, то есть введенное имя пользователя не существует в массиве - выдавать с помощьюalert сообщение об ошибке
* Если есть совпадение - выводить сообщение "Андрей, вы успешно вошли". Вместо "Андрей" должно быть имя текущего пользователя

Дома нужно прочесть следующие разделы из учебника learn.javascript.ru:

1. [Советы по стилю кода](https://learn.javascript.ru/coding-style) - обязательно к прочтению
2. [Основы JavaScript](https://learn.javascript.ru/first-steps)
3. [Структуры данных](https://learn.javascript.ru/data-structures)
4. [Очень полезный ресурс по тонкостям JavaScript](http://bonsaiden.github.io/JavaScript-Garden/ru/), периодически сюда нужно заглядывать в темы, о которых говорилось в видео

# Урок 3-4

Домашнее задание:

1. Создать объект с [методами](https://learn.javascript.ru/object-methods), которые будут динамически [генерировать](https://learn.javascript.ru/modifying-document) DOM вот [такой страницы](https://github.com/lialuckianchenko/markup_fe2o/blob/master/js_03-04/hw_js_03-04.jpg). Это будет тест, который мы будем разрабатывать в следующих заданиях. Сейчас вам нужно только динамически создать html, методы должны храниться в одном объекте. Для того чтоб страница выглядела красиво можете использовать [Bootstrap](http://getbootstrap.com/) и создавать DOM-элементы с готовыми бутстраповскими классами. Изначально на странице должен быть только <body>, вызывая методы объекта нужно создать dom-элементы

Дома посмотрите следующие материалы:

1. [Преобразование типов для примитивов](https://learn.javascript.ru/types-conversion)
2. [Преобразование объектов: toString и valueOf](https://learn.javascript.ru/object-conversion)
3. [Методы объектов, this](https://learn.javascript.ru/object-methods)
4. [Документ и объекты страницы](https://learn.javascript.ru/document)
5. [Видео о типах в javascript](https://www.destroyallsoftware.com/talks/wat)

В конце каждой статьи на learn.javascript.ru есть задачи, рекомендовано над ними поработать

# Урок 5-6

Домашнее задание:

1. Нужно сделать вот такой [таймер](http://www.online-stopwatch.com/full-screen-stopwatch/) (выбираете Stop Watch).
   1. Кнопка Start запускает таймер, так же нужно выводить милисекунды. После запуска она меняется на кнопку Pause, которая может приостанавливать таймер.
   2. Кнопка Clear останавливает и обнуляет таймер.
   3. Дизайн сделайте свой, используя Pure или Bootstrap.
2. Это более сложное задание, для тех кому покажется мало :) Не обязательно, но рекомендовано к выполнению. Сделать точно такой же таймер как [здесь](http://stopwatch.onlineclock.net/new/). Основная функциональность:
   1. Start - начинается отсчет с нуля. При работе таймера выводим миллисекунды. Кнопка Start заменяется на Stop.
   2. Split - засекаем текущий отсчет и выводим его под таймером с меткой Split, при этом продолжаем работу таймера.
   3. Stop - останавливаем текущий отсчет. Время, которое засекли, выводим под таймером с пометкой Stop.
   4. Reset - обнуляем таймер, удаляем все сохраненные отметки Split и Stop.
   5. Если будет не понятно как должны работать кнопки - просто посмотрите как они реализованы на текущем таймере по ссылке.

Так же нужно прочесть следующие материалы:

1. [Основы работы с событиями](https://learn.javascript.ru/events-and-interfaces)
2. [Events bubbling и events capturing](http://habrahabr.ru/post/126471/)
3. [setTimeout и setInterval](https://learn.javascript.ru/settimeout-setinterval)
4. [О том, как работают JavaScript таймеры](http://habrahabr.ru/post/138062/)
5. [Создание объектов через “new”](https://learn.javascript.ru/constructor-new)
6. [Явное указание this: “call”, “apply”](https://learn.javascript.ru/call-apply)
7. [JavaScript для котов](http://vova.io/blog/javascript-for-cats/)

# Урок 7-8

Домашнее задание:

1. Создать переключаемые табы, точно такие же как [здесь](https://jqueryui.com/tabs/). Визуально табы могут отличаться от примера, но постарайтесь выполнить частое пожелание заказчиков: "просто сделайте мне красиво!"
2. Сверстать форму, как на [примере](https://jqueryui.com/tooltip/#forms), и сделать для неё аналогичные всплывающие подсказки
3. Желательно сделать все это с интересной анимацией. Но не обязательно, с ней будем разбираться позже
4. Все реализовываем с помощью jQuery, стили через css-классы
5. Точно такую же страницу как в ссылках делать не надо, только функциональные элементы (табы и форма с подсказками)
6. jQueryUI и сторонние плагины не используем

Так же нужно прочесть следующие материалы:

1. [Функциональные выражения](https://learn.javascript.ru/function-declaration-expression)
2. [Именованные функциональные выражения](https://learn.javascript.ru/named-function-expression)
3. [Immediately-Invoked Function Expression (IIFE)](http://benalman.com/news/2010/11/immediately-invoked-function-expression/) - очень хорошая статья, нужно осилить
4. [Замыкания, область видимости](https://learn.javascript.ru/functions-closures) - читать весь раздел, кроме последних двух. Для общего развития можете и их прочесть
5. [Цикл статей "jQuery для начинающих"](http://anton.shevchuk.name/javascript/jquery-for-beginners/)
6. [Загрузка документа: DOMContentLoaded, load, beforeunload, unload](https://learn.javascript.ru/onload-ondomcontentloaded)
7. [Учебник "jQuery для начинающих"](http://anton.shevchuk.name/jquery-book/) - рекомендовано читать на досуге

# Урок 9-10

Домашнее задание:

1. Создать страницу, к ней подключить [карусель](http://sorgalla.com/jcarousel/)
2. К той же странице подключить любой, понравившийся вам нестандартный select из [списка](http://habrahabr.ru/company/aiken/blog/114927/). Если лень выбирать - подключайте [cuSel](http://www.xiper.net/collect/html-and-css-tricks/verstka-form/nice-select-jquery.html)
3. Подключить к странице нестандартные чекбоксы двумя способами: с помощью [jquery](http://xiper.net/collect/html-and-css-tricks/verstka-form/nice-checkbox) (сделать такой же вариант как [здесь](http://xiper.net/examples/html-and-css-tricks/forms/nice-checkbox-3.html)); с помощью [css](http://xiper.net/collect/html-and-css-tricks/verstka-form/kastomnie-checkbox-i-radio-na-css)
4. Дизайн карусели, селекта и чекбокса любой
5. Создать анимированную выпадающую двухуровневую меню, как на [примере](http://callmenick.com/_development/slide-down-menu/). Плагины для меню не используем, логика появления и анимирования должна быть написана вами. Так же, во время анимации желательно менять цвет фона появившейся менюшки. Для этого вам может понадобится [плагин](http://xiper.net/collect/js-plugins/effects/jquery-color)
6. Бонусное задание, для тру-разрабов: написать такую же [выпадающую меню](http://callmenick.com/_development/slide-down-menu/) на чистом js. Анимировать цвет фона не обязательно. Всем выполнять не обязательно. Только тем, кто хочет постичь больше

Так же нужно прочесть следующие материалы:

1. [25 советов по улучшению вашего кода jQuery](http://habrahabr.ru/post/63797/)
2. [JS-Анимация](https://learn.javascript.ru/js-animation)
3. [Программная анимация на xiper](http://xiper.net/collect/js-plugins/overclock-site/programm-animation)
4. [Оптимизация DOM анимации](http://xiper.net/collect/js-plugins/overclock-site/optimizacia-dom-animacii)
5. [jQuery.animate()](http://api.jquery.com/animate/)
6. [Функция плавности (easing)](http://easings.net/ru)
7. [cubic-bezier](http://cubic-bezier.com/#.17,.67,.83,.67)
8. [jQuery Easing Plugin](https://github.com/gdsmith/jquery.easing)
9. [jQuery.color — учим animate работать с цветом](http://xiper.net/collect/js-plugins/effects/jquery-color)

# Урок 11-12

Домашнее задание:

1. Создать карусель по [туториалу](http://ru.jakeroid.com/sozdaem-prostuyu-karusel-slajder-s-pomoshhyu-jquery.html). Оформить её как jQuery-плагин
2. Подключить [шаблонизатор](http://ejohn.org/blog/javascript-micro-templating/). Отрендерить с его помощью несложную html-структуру. Можно вывести простой профиль пользователя, как в [первом домашнем задании](https://github.com/goit-fe/markup_fe2o/blob/master/html_01/homework1.png)
3. Бонусное задание: подключить [шаблонизатор LoDash](https://learn.javascript.ru/template-lodash), с его помощью сделать то же что и в предыдущем пункте

Так же нужно прочесть следующие материалы:

1. [jQuery для начинающих. Пишем плагины](http://anton.shevchuk.name/javascript/jquery-for-beginners-write-plugins/)
2. [How to Create a Basic Plugin](http://learn.jquery.com/plugins/basic-plugin-creation/)
3. [Пишем плагин для jQuery](http://habrahabr.ru/post/158235/)
4. [jQuery.extend()](http://jquerybook.ru/api/jQuery.extend()-fn86.html)
5. [Шаблонизация, шаблонизатор LoDash](https://learn.javascript.ru/template-lodash)
6. [Шаблонизатор Джона Резига](http://ejohn.org/blog/javascript-micro-templating/)

# Урок 13-14

Домашнее задание:

1. За основу взять [домашку 3-4](https://github.com/lialuckianchenko/markup_fe2o/blob/master/js_03-04). Перед стартом приложения записать в localStorage объект с вопросами и ответами теста. Потом прочесть данные из localStorage, отрендерить их используя любой javascript-шаблонизатор
2. Так же в приложении должна быть прописана директива 'use strict;', в консоли не должно быть никаких javascript-ошибок
3. На странице должна быть возможность пройти тест. При клике на кнопку "Проверить мои результаты" нужно выполнить проверку правильных ответов и вывести сообщение об успешном или не успешном прохождении теста. После вывода сообщения, обнулить отметки чтоб тест можно было пройти опять
4. В каждом вопросе может быть только один правильный вариант ответа
5. Само сообщение о результатах теста нужно сделать в стиле простого [модального окна](http://getbootstrap.com/javascript/#modals-examples). Не обязательно соблюдать такой же дизайн как в Bootstrap, можете сделать по своему, но верстка и код появления модальки должны быть написаны вами. Допускается использование jQuery. Кому хочется больше веселья, делайте без неё
6. Бонусная часть: в вопросах должна быть возможность выбрать несколько правильных вариантов ответа, не только один

Так же нужно прочесть следующие материалы:

1. [Формат JSON, метод toJSON](https://learn.javascript.ru/json)
2. [Introducing JSON](http://json.org/)
3. [XML wiki](https://ru.wikipedia.org/wiki/XML)
4. [JSON wiki](https://ru.wikipedia.org/wiki/JSON)
5. [localStorage](http://htmlbook.ru/html5/storage)
6. [Dive into HTML5 - Storage](http://diveinto.html5doctor.com/storage.html)
7. [Cookies](https://learn.javascript.ru/cookie)
8. [Современный стандарт, "use strict"](https://learn.javascript.ru/strict-mode)
9. [Перехват ошибок, "try..catch"](https://learn.javascript.ru/exception)

# Урок 15-16

Домашнее задание:

1. Сделать форму поиска как на google.com - должно быть поле для ввода запроса, и кнопка отправить. Так же отправка поискового запроса должна осуществляться по нажатию кнопки Enter
2. Сделать запрос на Google API по [примеру](http://anton.shevchuk.name/javascript/jquery-and-google-search-api-or-yahoo-search-api-yahoo-search-api/)
3. Вывести результаты поиска под форму. Дизайн постараться сделать близким к дизайну google.com
4. Тестовый API-ключ для поиска Google -ABQIAAAACKQaiZJrS0bhr9YARgDqUxQBCBLUIYB7IF2WaNrkYqF0tBovNBQFDtM\_KNtb3xQxWff2mI5hipc3lg
5. Создать класс Human, у которого будут свойства обычного человека: имя, возраст, пол, рост, вес. Используя прототипное наследование создать дочерние классы Worker (дописать в них поля места работы, зарплатой, методом "работать") и Student (дописать поля места учебы, стипендией, методом "смотреть сериалы")
6. Создать несколько экземпляров классов Worker и Student, вывести их в консоль. Убедиться что они имеют поля родительского класса Human

Так же нужно прочесть следующие материалы:

1. [GET или POST: что выбрать?](http://xiper.net/learn/also-need-to-know/get-or-post)
2. [Как работает браузер: HTTP запрос](http://xiper.net/learn/also-need-to-know/how-does-a-browser-HTTP-request)
3. [Как работает браузер: обмен данными с сервером](http://xiper.net/learn/also-need-to-know/how-does-a-browser-communicate-with-the-server)
4. [XMLHTTPRequest: описание, применение, частые проблемы](http://xmlhttprequest.ru/)
5. [Основы XMLHttpRequest](https://learn.javascript.ru/ajax-xmlhttprequest)
6. [jQuery для начинающих. Часть 3. AJAX](http://anton.shevchuk.name/javascript/jquery-for-beginners-ajax/)
7. [jQuery + AJAX + (Google Search API || Yahoo Search API)](http://anton.shevchuk.name/javascript/jquery-and-google-search-api-or-yahoo-search-api-yahoo-search-api/)
8. [AJAX и COMET](https://learn.javascript.ru/ajax) - раздел учебника learn.javascript.ru, рекомендуемо на досуге прочесть все
9. [ООП в прототипном стиле](https://learn.javascript.ru/prototypes)
10. [ООП в функциональном стиле](https://learn.javascript.ru/oop)

# Урок 17-18

Домашнее задание:

1. Установить [node.js](https://nodejs.org/en/)
2. Установить [grunt.js](http://gruntjs.com/)
3. Настроить сборку проекта простого. Должны выполняться задачи по конкатенации и минификации вашихjs и css файлов
4. Обычно системную папку node\_modules принято добавлять в [.gitignore](https://git-scm.com/book/ru/v1/%D0%9E%D1%81%D0%BD%D0%BE%D0%B2%D1%8B-Git-%D0%97%D0%B0%D0%BF%D0%B8%D1%81%D1%8C-%D0%B8%D0%B7%D0%BC%D0%B5%D0%BD%D0%B5%D0%BD%D0%B8%D0%B9-%D0%B2-%D1%80%D0%B5%D0%BF%D0%BE%D0%B7%D0%B8%D1%82%D0%BE%D1%80%D0%B8%D0%B9#%D0%98%D0%B3%D0%BD%D0%BE%D1%80%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5-%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2), поскольку там хранится очень много файлов, и держать их в репозитории плохая практика
5. Бонусное задание: настроить аналогичную сборку, но на [gulp.js](http://gulpjs.com/). Он немного сложнее для первого понимания, но более современный и быстрый. Настоятельно рекомендую вернуться к gulp.js после того как разберетесь с grunt.js
6. Для работы с командной строкой в Windows, вам может быть полезным использовать git bash

Полезные материалы:

1. [Стандартные команды ОС UNIX](https://ru.wikipedia.org/wiki/UNIX#.D0.A1.D1.82.D0.B0.D0.BD.D0.B4.D0.B0.D1.80.D1.82.D0.BD.D1.8B.D0.B5_.D0.BA.D0.BE.D0.BC.D0.B0.D0.BD.D0.B4.D1.8B_.D0.9E.D0.A1_UNIX)
2. [КОМАНДНАЯ СТРОКА WINDOWS](http://www.allmbs.ru/cmd-01.html)
3. [Несколько полезностей по работе с NPM](https://habrahabr.ru/post/206678/)
4. [Grunt для тех, кто считает штуки вроде него странными и сложными](http://frontender.info/grunt-is-not-weird-and-hard/)
5. [Grunt, инструмент для сборки javascript проектов](https://habrahabr.ru/post/148274/)
6. [Самые нужные плагины для Grunt](https://habrahabr.ru/post/251157/)
7. [Сборка проектов с помощью Gulp.js. Семинар в Яндексе](https://habrahabr.ru/company/yandex/blog/239993/)
8. [Приятная сборка frontend проекта](https://habrahabr.ru/post/250569/)
9. [Скринкаст по Gulp](http://learn.javascript.ru/screencast/gulp)
10. [Скринкаст Node.JS. Основные возможности и средства создания веб-сервисов, включая внутренние особенности самого сервера Node.JS](https://learn.javascript.ru/screencast/nodejs) - для тех кто хочет глубже разобраться в Node.JS

# Урок 19-20

Домашнее задание:

1. Сверстать [макет](https://github.com/lialuckianchenko/markup_fe2o/blob/master/js_19-20/homework19_20.psd) используя SCSS, BEM. Так же прикрутить слайдер с помощью jQuery. Кроссбраузерность и пиксельная точность не обязательны
2. Скопировать себе в script.js [data.json](https://github.com/lialuckianchenko/markup_fe2o/blob/master/js_19-20/data.json), сделать из него обычный javascript-массив (распарсить). С помощьюlodash получить несколько массивов: 1. Массив скиллов (поле skills) всех людей, не должно быть повторяющихся скиллов, так же они должны быть отсортированы по алфавиту; 2. Массив имен (поле name) людей, отсортированных в зависимости от количества их друзей (friends); 3. Массив всех друзей всех пользователей, не должно быть повторяющихся людей

Полезные материалы:

1. [Документация SASS на русском](http://sass-scss.ru/)
2. [SCSS — немного практики](https://habrahabr.ru/post/140612/)
3. [Как писать более чистый CSS и рациональный SASS](https://habrahabr.ru/company/paysto/blog/255981/)
4. [SCSS: пара полезных техник](https://habrahabr.ru/post/151679/)
5. [Начинаем писать CSS](http://frontender.info/starting-css/) - немного вспомним прошлое
6. [Sass-архитектура вашего проекта](https://habrahabr.ru/post/249047/)
7. [Organizing Your CSS Code for Preprocessors](http://modernweb.com/2014/04/14/organizing-your-css-code-for-preprocessors/)
8. [SASS против LESS](https://habrahabr.ru/post/144309/)
9. [CSS-препроцессоры и преимущество их использования](http://zencoder.ru/css/css-sass-less-stylus/)
10. [LESS](http://lesscss.org/), [Stylus](http://stylus-lang.com/) - аналоги SASS
11. [Документация lodash](https://lodash.com/)
12. [lodash (underscore) — знай свою стандартную библиотеку](https://habrahabr.ru/post/217515/)
13. [Хватит писать циклы for. Пришло время использовать Underscore](http://frontender.info/stop-writing-for-loops-start-using-underscorejs/)
14. [Lodash: 10 Javascript Utility Functions That You Should Probably Stop Rewriting](http://colintoh.com/blog/lodash-10-javascript-utility-functions-stop-rewriting)
15. [Введение в Source Maps](https://habrahabr.ru/post/178743/)

# Урок 21-22

Домашнее задание:

1. Взять код [домашки 13-14](https://github.com/lialuckianchenko/markup_fe2o/blob/master/js_13-14), покрыть тестами хоть какие-то методы. 3-4 теста будет достаточно. Если не получится покрыть код 13-14, можете взять метод pow из [домашки 01-02](https://github.com/lialuckianchenko/markup_fe2o/blob/master/js_01-02), покрыть его тестами
2. Переписать всю [домашку 13-14](https://github.com/lialuckianchenko/markup_fe2o/blob/master/js_13-14) на ES6

Полезные материалы:

1. [Jasmine и юнит тесты](http://stepansuvorov.com/blog/2012/10/jasmine-%D0%B8-%D1%8E%D0%BD%D0%B8%D1%82-%D1%82%D0%B5%D1%81%D1%82%D1%8B/)
2. [Jasmine introduction](http://jasmine.github.io/2.0/introduction.html)
3. [Автоматические тесты при помощи chai и mocha](https://learn.javascript.ru/testing) - как аналог Jasmine
4. [Современные возможности ES-2015](https://learn.javascript.ru/es-modern) - прочесть первые 9 разделов
5. [Babel](https://babeljs.io/)
6. [ES6 в деталях: введение](https://habrahabr.ru/post/257005/)
7. [Таблица поддержки ES6 браузерами](https://kangax.github.io/compat-table/es6/)

# Урок 23-24

Домашнее задание:

1. Работа над ToDo-list: сделать то же, что было показано в видео, и добавить возможность редактирования элемента списка. Cделать все красиво, без использования Bootstrap. Применение БЭМ приветствуется.
2. Вынести классы Model, View, Controller в отдельные файлы. Подключить их с помощью [require.js](http://requirejs.org/)
3. Бонусное задание: собрать файлы с помощью [r.js](http://requirejs.org/docs/optimization.html)

Полезные материалы:

1. [Загрузчик модулей RequireJS](http://stepansuvorov.com/blog/2012/10/%D0%B7%D0%B0%D0%B3%D1%80%D1%83%D0%B7%D1%87%D0%B8%D0%BA-%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D0%B5%D0%B9-requirejs/)
2. [Модульный подход к разработке web-приложений с использованием JavaScript: AMD и RequireJS](https://habrahabr.ru/post/152833/)
3. [Оптимизация кода с помощью RequireJS: как это делается и для чего это нужно](https://habrahabr.ru/post/254675/)
4. [MVC в JavaScript](http://designformasters.info/posts/mvc-javascript/)
5. [Javascript MVC. Hello world](http://sandbox.thewikies.com/javascript-mvc-hello-world/)
6. [И опять про MVC](https://habrahabr.ru/post/119369/)
7. [Basic JavaScript MVC Implementation](https://gist.github.com/g6scheme/4157554)
8. [JavaScript MVC on "A List Apart"](http://alistapart.com/article/javascript-mvc)
9. [TodoMVC](http://todomvc.com/)
10. [Используем паттерн Наблюдатель(Observer) для создания индикатора выполнения процесса на Javascript](https://habrahabr.ru/post/70793/)
11. [Свои события или observer на Javascript](http://php-zametki.ru/javascript-laboratoriya/79-svoi-sobytiya-javascript-observer.html)