**Кто есть кто в семейке Джавы Скрипт**

**Глава семейства : Джава Скрипт.**

Это нативный язык програмирования, используемый с 1995 года. Это язык Интернета, язык сценариев взаимодействия с пользователем и работы с сервером. Похоже на работу официанта в ресторане- взял заказ, отдал на кухню, взял другой, подал, рассчитал клиентов.

**Лучшая половина Джавы Скрипт : Библиотека Джава Скрипт .**

Библиотека является многоразовой частью кода, которая предлагает определенные функции. Это набор функций, объектов и классов, которые мы можем использовать в своем приложении. Библиотека абстрагирует различные слои, так что не нужно беспокоиться об их деталях реализации. Мы можем вызвать библиотечную функцию и передать ей некоторые параметры, и библиотека выполнит ее и вернет нам элемент управления. Как в жизни. Мама знает и может всё. Или почти всё.

**Дети:**

[React](https://facebook.github.io/react/) - это библиотека JavaScript, созданная разработчиками Facebook и Instagram, с его помощью можно создать интерактивный интерфейс приложения.  Он использует компонентную модель, в которой компоненты являются повторно используемыми элементами пользовательского интерфейса, и каждый компонент имеет свое собственное состояние.

### [JQuery](https://jquery.com/)

### jQuery - это библиотека, которая сделала DOM-манипуляции проще.Несколько лет назад jQuery считался прочным решением для создания надежных веб-сайтов с поддержкой кросс-браузерности. Основные функции jQuery, такие как манипулирование DOM на основе селекторов CSS, обработка событий и создание вызовов AJAX.

### [D3: Документы, управляемые данными](https://d3js.org/)

D3 (или D3.js) - мощная библиотека JavaScript для создания интерактивных визуализаций с использованием веб-стандартов, таких как SVG, HTML и CSS. В отличие от других библиотек визуализации, D3 предлагает лучший контроль над окончательным визуальным результатом.D3 работает, связывая данные с DOM и затем преобразуя их в документ. Она также имеет собственную экосистему, которая состоит из плагинов и библиотек, которые расширяют ее базовые функциональные возможности. Библиотека существует с 2011 года, и в ней есть множество документации и учебников.

### [Moments](https://d3js.org/)

библиотека JavaScript , благодаря которой можно очень удобно производить различные манипуляции над датами: сложение, вычитание различных временных единиц.

### js-datepicker

### библиотека JavaScript *для ввода дат*

### ChartJS

### библиотека JavaScript *для работы с графиками*

### Datatable

### библиотека JavaScript *для простых и удобных таблиц с пейджингом и сортировкой.*

[**gm**](https://github.com/aheckmann/gm)

библиотека JavaScript, которая позволяет создавать, править, комбинировать изображения (менять их размеры, кадрировать, выполнять преобразование

[validate.js](https://github.com/chriso/validator.js)

библиотека JavaScript для проверки введённых данных на стороне клиента и на сервере — для того, чтобы отфильтровать опасные данные, отправленные злоумышленником.Это модуль , [validate.js](https://github.com/chriso/validator.js), имеет методы isEmail(), isURL(), isMobilePhone(), isCreditCard

[Nodemailer](https://github.com/nodemailer/nodemailer)

библиотека JavaScript для отправки пользователям подтверждений регистрации, для рассылки уведомлений о важных событиях, новостей. Он поддерживает отправку писем, содержащих в себе обычный текст, HTML, изображения, и, что самое важное, поддерживает безопасный протокол обмена информацией SSL/STARTTLS.

**Лучшие друзья: Фреймворки**

Фреймворк имеет архитектуру, которая диктует поток управления в приложении. Фреймворк описывает скелет и рассказывает вам, как все должно быть организовано. Вам также предоставляются основные функции, необходимые для запуска вашего приложения. Кроме того, вы обязаны следовать принципам и шаблонам фреймворка. Разница между фреймворком и библиотекой заключается в том, что вы вызываете библиотеку, тогда как фреймворк вызывает вас.

Фреймворк часто включает в себя множество библиотек и имеет более высокий уровень абстракции. Функциональность, такая как обработка событий, создание вызовов AJAX, привязка шаблонов и данных и тестирование, уже встроены в фреймворк/

### [Angular](https://angular.io/)

AngularJS когда-то была самой популярной технологией JavaScript среди разработчиков интерфейсов. Он был поддержан Google и сообществом частных лиц и корпораций.

### [Vue.js](https://vuejs.org/)

Vue.js - это легкий JavaScript фреймворк, который активно развивается.

### [Ember.js](https://www.emberjs.com/)

Ember.js является фронт-енд фреймворком на основе шаблона Model-View-ViewModel (MVVM). Фреймворк использует стандартный подход к конфигурации, который популярен среди серверных фреймворков таких как Ruby on Rails и Laravel.

## Прислуга: выполнение общих задач и инструменты

Инструменты, выполняющие общие задачки - это инструменты, используемые для автоматизации определенных повторяющихся задач. В число самых популярных  входят:

### [Gulp](https://gulpjs.com/)

Gulp - это набор инструментов JavaScript, который используется как менеджер задач и как система сборки в веб-разработке. Компиляция, минимизация кода, оптимизация изображений, модульное тестирование, линтинг и т.д. - это повторяющиеся задачи, которые должны быть автоматизированы. Gulp упрощает процесс написания задач даже для людей, которые менее знакомы с JavaScript.

Gulp использует конвейеры для потоковой передачи данных из одного плагина в другой, и конечный результат выводится в папку назначения. Gulp работает лучше по сравнению с Grunt, потому что он не создает временные файлы для хранения промежуточных результатов, что приводит к меньшему количеству вызовов ввода-вывода.

### [Grunt](https://gruntjs.com/)

Grunt - это менеджер задач и инструмент автоматизации JavaScript. Grunt имеет интерфейс командной строки, который позволяет запускать настраиваемые задачи, определенные в файле, называемом Gruntfile. У Grunt есть тысячи плагинов на выбор, которые должны охватывать большинство обычных повторяющихся задач, с которыми вы столкнулись. С помощью Grunt вы можете запускать все задачи под одной командой, что упрощает вашу жизнь.

### [npm](https://www.npmjs.com/)

Gulp и Grunt требуют, чтобы вы потратили время на изучение и освоение нового инструмента, что требует времени. Ввод дополнительных зависимостей в ваш проект можно избежать, выбирая альтернативу, которая уже связана с Node.js. Хотя npm более известен как менеджер пакетов, сценарии npm можно использовать для выполнения основной части вышеупомянутых задач.

### [Jest](https://facebook.github.io/jest/)

### Jest - это относительно новый фреймворк для тестирования, созданный в Facebook и хорошо принятый сообществом React. Существует распространенное заблуждение, что Jest специально разработан для работы с React; однако,его можно использовать для тестирования любого кода JavaScript.

### [Mocha](https://mochajs.org/)

Mocha - это фреймворк для тестирования JavaScript, который предоставляет поддержку браузера, асинхронные вызовы, включая обещания, отчеты о тестовом покрытии и JavaScript API для запуска тестов. Mocha часто используется с библиотекой утверждений, такой как Chai, should.js, expect.js  или better-assert, потому что у нее нет собственной библиотеки утверждений.

### [Jasmine](https://jasmine.github.io/)

Jasmine - это фреймворк для тестирования JavaScript, ориентированный на поведение. Jasmine стремится стать браузером, платформой и независимым от платформы набором тестов. У Jasmine есть своя собственная библиотека утверждений, называемая matchers, которая дает ему чистый и легко читаемый синтаксис. Jasmine не имеет встроенной команды для выполнения тестов, и вам, возможно, придется использовать какую-нибудь общую, например, Karma.