Проект

Проект называется KFund School - происходит от фонда KFund.

Мы будем с вами изучать WEB Front-End.

Занятия проходят дважды в неделю. Каждый урок длится 2 часа.

Нужно создать группу в телеграмме для быстрой коммуникации ( объяснить правила поведения в группе )

У нас будет 4 модуля - спринта ( рассказать каких ):  
от простого к сложному и в финале мы реализуем большой проект, который пойдет вам в портфолио.

Что такое WEB?

Многие начинающие пользователи сети интернет, часто заблуждаются думая, что веб (web) или всемирная паутина это и есть сеть Internet. На самом деле, все далеко не так, как кажется на первый взгляд. Интернет является гораздо более широким понятием, включающем в себя, в том числе и веб или WWW. Давайте разберемся более подробно, что такое веб на самом деле.

Веб, web, всемирная паутина, WWW (World Wide Web) это разные названия одного и того же. Представляет собой систему взаимосвязанных гипертекстовых документов (веб-сайтов) расположенных на компьютерах по всему миру, подключенных к глобальной сети [интернет](https://beginpc.ru/internet/chto-takoe-internet). Другими словами то, что большинство людей в обиходе называют интернетом, на самом деле является всемирной паутиной или World Wide Web.

Как устроен веб

Отцом основателем World Wide Web считается Тимоти Джон Бернерс-Ли работавший в Европейском совете по ядерным исследованиям (CERN) и придумавший в 1989 году базовые основы всемирной паутины: протокол HTTP, язык HTML на основе гипертекстовой разметки документа и идентификаторы URI. Применение этих технологий сделало информацию общедоступной в не зависимости от аппаратного и программного обеспечения.

ВЕБ сейчас

Необходимость в создании сайтов для различных компаний и организаций прямо пропорциональна уровню развитию веб-технологий. Мы живем в цифровую эру, а некоторые из нас проводят почти все свое время в виртуальной реальности. В интернете мы работаем, учимся, находим покупателей, партнеров, заказчиков.

Современные тенденции развития бизнеса таковы:

некоторые компании разрастаются в интернете в огромные порталы, приносящие своим владельцам миллиардные прибыли;

многие магазины обзаводятся онлайн-площадками для продаж или даже полностью «переезжают» из офлайна в онлайн;

с каждым годом все больше рекламных кампаний проводятся в сети, а не на привычных радио и телевидении;

все больше людей предпочитают «ходить» по интернет-магазинам, экономя при этом свое время и деньги.

Вот почему потребность в специалистах по разработке сайтов так резко возросла (и продолжает расти). Вот почему в вас родилось закономерное желание «влиться» в этот процесс и стать web-программистом с нуля.

Хороший специалист по веб-программированию должен уметь правильно создавать архитектуру сайта, автоматизировать работу по добавлению, удалению и обновлению контента, сбору информации от посетителей и т. д.

Что такое веб-программирование?

Одни говорят, что веб-программирование — это талант, заложенный на генетическом уровне, другие — что это искусство. А Википедия говорит, что это раздел веб-разработки, ориентированный на создание веб-приложений (программ, обеспечивающих работу динамических сайтов в Интернете). Сейчас поясню.

Сайт — это не набор статичных веб-страниц. Да, они могут быть изумительно красивыми, иметь интересное, идеально организованное содержание, но это далеко не все. Сайт должен быть интерактивным, то есть, способным реагировать на действия посетителя, собирать и обрабатывать определенные данные, полученные от пользователей.

На современном сайте невозможно обойтись без специальных программ — скриптов. С их помощью на сайт можно добавить форму обратной связи, личный кабинет, голосование, поиск, фильтр и т. д. Таким образом, задача веб-программиста — «вдохнуть» жизнь в веб-страницы, чтобы пользователь мог не только ими любоваться, но и в интерактивном режиме взаимодействовать с ними.

Составляющие веб-программирования

Веб-дизайн.

Этот подраздел отвечает за разработку графического и функционального содержимого пользовательского веб-приложения, которое создается с помощью стандартов разметки гипертекста HTML и XHTML, а также CSS (каскадных таблиц стилей). Для тех, кто не знает, что это такое, советую ликвидировать безграмотность в этом вопросе и пройти [бесплатный курс по HTML и CSS](https://webformyself.com/minikurs/html5/?utm_medium=systema&utm_source=statya&utm_campaign=html5_new). XHTML представляет собой новое поколение языка гипертекстовой разметки.

Программирование на стороне клиента (фронтенд).

Клиентское программирование — это создание логики на стороне браузера (сохранение и обработка данных пользователей, динамическое изменение веб-страницы без перезагрузки, отправка серверу Ajax-запросов). Оно поддерживается следующими технологиями:

JavaScript. Это самый популярный язык программирования, который поддерживается всеми устройствами и интернет-браузерами. Для изучения основ JavaScript можете воспользоваться [этим видеокурсом](https://webformyself.com/category/premium/javascript-premium/javascript-javascript-premium/), в котором подробно объясняются основные моменты языка. Если вы решили всерьез заняться web-программированием на фронтенде, рекомендую пройти курс JavaScript & jQuery для начинающих с нуля до профи;

[jQuery](https://webformyself.com/category/premium/javascript-premium/jquerypremium/). С помощью данной библиотеки, написанной на JavaScript, вы сможете выполнять множество операций при работе с объектной моделью (редактирование содержимого веб-страниц, создание различных визуальных эффектов и т. д.);

[AJAX](https://webformyself.com/category/premium/javascript-premium/listajaxpremium/). Это технология создания интерактивных пользовательских интерфейсов на основе асинхронной загрузки гипертекста.

Программирование на стороне сервера (бэкенд).

Это создание логики по части сервера, которая включает взаимодействие с БД, обработка клиентских запросов, проверка корректности переданных данных, динамическое формирование HTML-страниц и т. д.). Серверное программирование поддерживается следующими технологиями:

PHP. Этот язык веб-программирования является самым простым и широко распространенным. Он поддерживается большинством хостинг-провайдеров; на нем написано множество движков для сайтов. Для изучения основ PHP могу порекомендовать [крутой видеокурс](https://webformyself.com/category/premium/php-premium/phppremium/);

ASP.NET. Это технология от компании Microsoft. Ее окружение позволяет легко и быстро создавать различные веб-приложения. Инфраструктура ASP.NET берет на себя асинхронное обновление страницы средствами AJAX, упаковку удаленных вызовов к прочим веб-сервисам в сообщения SOAP, генерацию Proxy-классов по описанию WSDL, преобразование элементов управления в код JavaScript и HTML;

Python, Ruby, Perl и другие языки программирования для бэкенда.

Далее поговорим о том, как стать веб-программистом с нуля: самостоятельно или все же с помощью курсов и наставников.

Чем отличается верстальщик от front-end developer?

**Верстальщик** преобразует графический макет (Photoshop или иной) в набор HTML + CSS + картинки. Иногда к свёрстанному макету может подключить типовые библиотеки Javascript, например, slider для картинок, или всплывающие подсказки (tooltip), или диалоговые окна (dialog/popup).

Знания и навыки:

* работа с графическими программами, чтобы понять, как собран макет
* знание HTML, HTML5, CSS, CSS3, понятие про веб-шрифты, спрайты и другие технологии
* пригодятся знания по HTML-фреймворкам, например, Twitter Bootstrap или Semantic UI
* навыки кроссбраузерной вёрстки, чтобы в разных браузерах выглядело и работало одинаково
* навыки отзывчивой вёрстки, чтобы можно было использовать на устройствах с разными возможностями и разрешениями
* знание типовых решений javascript, чтобы реализовать простейшие вещи, заложенные в макете

**Фронтенд-разработчик** делает так, чтобы макеты, полученные от верстальщика, были наполнены реальными данными. Если приложение построено как client-side (то есть вся основная логика загружается в виде огромного javascript в браузер, а данные запрашиваются с сервера по AJAX; это называется "толстый клиент"), то фронтенд-разработчику потребуется следующее:

* знание HTML, HTML5, CSS, CSS3, понятие про веб-шрифты, спрайты, Comet и другие технологии
* глубокое знание Javascript, включая использование готовых фреймворков, библиотек и написание расширений для них, что подразумевает объектно-ориентированное и событийное программирование
* знание AJAX, CORS и навык создания тестовых затычек на стороне сервера, чтобы можно было разрабатывать приложение пока бакенд не готов

Если фронтенд строится на стороне сервера, то дополнительно потребуется знать используемый серверный язык программирования (например, Python, Ruby или PHP) и используемый фреймворк (Django, Ruby-on-Rails, Yii). На практике бывало такое, что фронтендер просил в нужной части проекта сделать var\_dump от структуры данных, которую надо показать и перечислить серверные методы, которые надо вызвать по нажатию предполагаемых кнопок.

Зачастую фронтенд-разработчик может и сам закодировать эти серверные методы, если не требуется углубляться в серверную логику (отношения в данных, конкретная бизнес-логика, хранение данных, кэширование, очереди, крон-задачи). Я лично таких очень ценю.

И моё личное мнение - фронтенд разработчику не помешают базовые знания про UML. Иногда с ними так тяжело обсуждать обмен данными по AJAX. У них это какой-то непрерывный поток магической энергии, волшебным образом преобразующийся в буковки на экране пользователя, а вот для бакенда это набор отдельных операций, иногда ещё и асинхронный. Диаграммы последовательностей ни читать, ни писать многие не умеют. Таймлайны составлять не умеют.

Популярные виды браузеров

На просторах глобальной сети можно найти множество программ, с помощью которых можно посещать страницы сайтов. Пользователю следует знать, какие есть браузеры для выхода в [интернет в частном доме](https://altclick.ru/internet-v-chastnyj-dom/), какие из них пользуются популярностью среди людей.

**На нынешний момент можно составить рейтинг интернет-браузеров, которые получили наибольшее распространение:**

1. Google Chrome. Созданный специалистами компании Google, этот браузер работает на собственном движке. Оснащен поисковой системой Гугл, и идеально подходит для работы с одноименными сервисами.
2. Mozilla Firefox. Популярный среди пользователей браузер, который называют основным конкурентом Гугл Хром и Яндекс. Недостатком этой программы считают создание большой нагрузки на систему ПК или ноутбука.
3. Opera. Отличается массой возможностей и дополнительных опций. Начинающему пользователю будет сложно работать с этим браузером. Требователен к системе ПК.
4. Microsoft Edge. Новый браузер от Майкрософт интернет позволит использовать на 100%. Эта программа сменила стандартный Internet Explorer, который устанавливался по умолчанию с ОС Windows.
5. Safari. Продукт корпорации Apple, который устанавливается вместе с OS X и iOS. Отличительная особенность — используется преимущественно с операционными системами Apple.

Такое разнообразие популярных браузеров для интернета позволяет каждому пользователю выбрать оптимальный для себя вариант, опираясь на личные требования.

HTML/CSS/JS

HTML и CSS на самом деле всего лишь структура страницы и информация о стиле. Прежде чем перейти к JavaScript и другим языкам программирования, необходимо знать основы HTML и CSS, поскольку они находятся на передней части каждой веб-страницы и приложения.

В самом начале 1990-х годов HTML был единственным языком, доступным в Интернете. С тех пор многое изменилось и теперь одним из самых распространенных языков программирования является JavaScript.

HTML

В первую очередь, необходимо понять, что HTML - основа каждой веб-страницы, независимо от сложности сайта или количества задействованных технологий. Это важный навык для любого веб-профессионала и отправная точка для всех, кто имеет отношение к созданию или редактированию контента в Интернете. И, к счастью для нас, ему удивительно легко учиться.

Как работает HTML?

HTML отображает язык разметки гипертекста. «Язык разметки» означает, что HTML использует теги для идентификации различных типов контента и целей, которые каждый преследует на веб-странице.

Для разметки разметки используются HTML-теги, также известные как «элементы». Они имеют довольно интуитивные типы: заголовки, теги абзацев, теги изображений и т. д.

*Каждая веб-страница состоит из нескольких тегов HTML, обозначающих определенный тип контента на странице. Каждый тип содержимого на странице «обернут», т. е. окружен тегами.*

Используя HTML, вы можете добавлять заголовки, форматировать абзацы, разрывы строк, создавать списки, выделять текст, создавать специальные символы, вставлять изображения, создавать ссылки, создавать таблицы, управлять некоторым стилем и многое другое.

CSS

CSS - это каскадные таблицы стилей. Этот язык разметки определяет, как HTML-элементы веб-сайта должны отображаться на интерфейсе страницы.

*Если HTML - это гипсокартон, CSS - это краска.*

В то время как HTML является основной структурой вашего сайта, CSS - это то, что дает всему вашему сайту стиль. Цвета, интересные шрифты и фоновые изображения – все это заслуга CSS. Этот язык влияет на все настроение веб-страницы, что делает его невероятно мощным инструментом и важным навыком для веб-разработчиков. Он также позволяет веб-сайтам адаптироваться к различным размерам экрана и типам устройств.

Итак, что же такое CSS? Он обозначает каскадные таблицы стилей - и «таблицу стилей» самого документа. Любой браузер имеет таблицу стилей по умолчанию, поэтому на каждой веб-странице отражается хотя бы одна таблица стилей, в зависимости от того, какой браузер посетитель использует, независимо от дизайна страницы. К примеру, допустим стиль шрифта нашего браузера по умолчанию - Times New Roman, размер 12, если разработчик не применил собственную таблицу стилей, то посетитель увидит веб-страницу в Times New Roman размером 12.

JavaScript

JavaScript является более сложным языком, чем HTML или CSS, и он не был выпущен в бета-версии до 1995 года. В настоящее время JavaScript поддерживается всеми современными веб-браузерами и используется практически на каждом сайте в Интернете для более мощных и сложных функций.

Для чего используется JavaScript?

JavaScript - это логический язык программирования, который можно использовать для изменения содержимого веб-сайта и заставить его вести себя по-разному в ответ на действия пользователя. Общее использование JavaScript включает в себя окна подтверждения, призывы к действию и добавление новых идентификаторов к существующей информации.

Короче говоря, JavaScript - это язык программирования, который позволяет веб-разработчикам создавать интерактивные сайты с динамичными элементами.

Всплывающие окна

Одним из примеров JavaScript в действии являются окна, которые появляются на вашем экране. Подумайте, как в последний раз, когда вы ввели свою информацию в онлайн-форму, и появилось окно с подтверждением, попросив вас нажать «ОК» или «Отменить», чтобы продолжить. Это стало возможным благодаря JavaScript - в коде вы найдете инструкцию if else, которая говорит компьютеру делать что-то одно, если пользователь нажимает «ОК», и другое дело, если пользователь нажимает «Отмена».

Установка текстового редактора VS Code

<https://code.visualstudio.com/>

Устройство структуры документа

<https://habr.com/ru/company/macloud/blog/555082/>

Верстка задания

Lorem Ipsum