# **ЧТО ДОЛЖЕН ЗНАТЬ FRONTEND РАЗРАБОТЧИК В 2021 ГОДУ**

Верстка сайтов и веб-программирование привлекают большое количество новичков в мир IT. Это связано с достаточно низким порогом вхождения. Количество желающих стать фронтендщиком с каждым годом увеличивается, вследствии чего растут и требования к кандидатам.

Какие технологии необходимо изучить, чтобы стать FrontEnd разработчиком в 2021 году? Давайте разберемся.

**HTML5 & CSS3**

HTML5 и CSS3 — это фундаментальные технологии, без знания которых не обойтись ни одному веб-разработчику. С помощью языка гипертекстовой разметки HTML создается разметка (каркас) каждой интернет-страницы. Затем язык стилей CSS преображает сайт и придает ему привлекательный и эффектный внешний вид. Помимо этого, необходимо владеть:

* **кроссбраузерной адаптивной версткой**, чтобы уметь создавать сайты под мобильные устройства, планшеты и широкоформатные экраны и для различных браузеров;
* **семантической версткой** для повышения качества разметки и улучшения поисковой индексации сайта;
* **валидной версткой**, которая предполагает полное соответствие кода разработчика всем стандартам W3C — организации, которая создает и внедряет технологические стандарты для World Wide Web.

Уделите особое внимание верстке под мобильные устройства, так как современный веб-потребитель очень много времени проводит в смартфоне. Существует даже специальный подход, при котором сначала верстается веб-сайт под мобильные устройства, а только потом под десктопы — называется Mobile First.

Это общие требования по верстке. Теперь же рассмотрим более детально требования конкретно по языку стилей. Здесь вы должны знать основные свойства CSS3: фоны, градиенты, тени, анимации, трансформации, переходы, а также технологии Flex и Grid, о которых мы еще поговорим.

Хорошее владение HTML и CSS уже позволяет заниматься версткой сайтов и зарабатывать деньги. Именно с этих двух базовых технологий начинается путь к профессии FrontEnd разработчика.

Качественно изучить HTML и CSS вы сможете при помощи следующих наших курсов:

* [HTML5 & CSS3 Стартовый](https://itvdn.com/ru/video/html5-css3-starter);
* [HTML5 & CSS3 Базовый](https://itvdn.com/ru/video/html5-and-css3);
* [HTML5 & CSS3 Углубленный](https://itvdn.com/ru/video/html-css-advanced).

**Flex и Grid CSS**

Технологии верстки надежных адаптивных веб-страниц, которые позволяют легче создавать динамические сайты и удобнее структурировать их содержимое. Лучше всего Flex-верстку в действии показывает интерактивный сайт [flexboxfroggy.com](https://flexboxfroggy.com/#ru), а Grid-верстку - [cssgridgarden.com](https://cssgridgarden.com/#ru).

На нашем ресурсе также есть курсы, которые хорошо объясняют темы Flex и Grid верстки: “[Верстка сайта на CSS Grid](https://itvdn.com/ru/video/grid-css)”, “[Верстка сайта на FlexBox CSS](https://itvdn.com/ru/video/flexbox-css)”.

**Bootstrap 4**

Популярная HTML/CSS платформа для разработки адаптивных веб-приложений, которую применяют при создании сайтов и интерфейсов администраторских панелей. Основные преимущества Bootstrap:

* высокая скорость верстки;
* кроссбраузерность и кроссплатформенность;
* наличие хорошей документации, большого сообщества и огромного количества разнообразных обучающих материалов;
* низкий порог вхождения (необходимо знать лишь основы HTML, CSS, JavaScript и jQuery).

Также, будет полезно знать еще 2 CSS-фреймворка: Foundation и Materialize. Первый является комплексным и профессиональным инструментом, который следует принципу Mobile First. Имеет продвинутый интерфейс и позволяет создавать действительно уникальные, элегантные и легко настраиваемые веб-сайты. Под эгидой Foundation были созданы такие веб-сервисы, как: Facebook, Mozilla, Adobe, Amazon, eBay и многие другие.

Materialize исповедует принципы Material Design от компании Google, которые преследуют идеи цифровой унификации для более легкого и комфортного восприятия интерфейса. На сайте [material.io](https://material.io/) можно более детально ознакомиться с идеологией материального дизайна. Фреймворк Materialize незаменим при разработке несложного многостраничного веб-сайта или лендинга за счет богатых возможностей уже готовых и встроенных компонентов и стилей. Также, Materialize предоставляет разные компоненты и варианты поведения, обычно не встречающиеся в других фронтенд-фреймворках (поведение волн, scrollFire, scrollSpy, или специфичные для мобильных устройств выдвижные меню и всплывающие подсказки и т. д.).

Foundation подходит разработчикам с опытом, в то время, как Materialize — отличный выбор для начинающих девелоперов, которые хотят как можно плотнее познакомиться с материальным дизайном.

**CSS препроцессоры**

CSS препроцессор — это программа, которая имеет свой собственный синтаксис, но может сгенерировать из него CSS код. Самыми популярными считаются **SASS, Stylus, LESS и PostCSS**, однако, наибольшее комьюнити имеет именно SASS. Препроцессоры предназначены для:

* ускорения процесса написания кода;
* упрощения чтения кода и дальнейшей его поддержки;
* минимизации рутинной работы при написании кода.

Для повышения эффективности написания CSS кода вполне достаточным будет изучение лишь одного препроцессора.

### **Git & GitHub**

Git — наиболее популярная система контроля версий, которая позволяет вести историю разработки проекта с возможностью доступа к каждой сохраненной версии.

Помимо этого, стоит уметь работать с сервисом онлайн-хостинга проектов, использующих систему контроля версий. В данном случае это GitHub. В тандеме с Git он позволяет разработчикам сохранять свой код онлайн, а затем взаимодействовать с другими разработчиками в разных проектах.

Данные системы позволяют команде программистов работать над одним проектом одновременно, сохраняя внесенные изменения, а также отслеживать выполнение задач каждым членом группы.

Git & GitHub — очень важные инструменты для любого IT-разработчика, изучением которых следует заняться как можно раньше.

Мы рекомендуем изучать Git на ITVDN при помощи курса “[Основы работы с Git](https://itvdn.com/ru/video/git-basics)”.

**Знания о веб-технологиях и сети интернет**

Чтобы качественно выполнять свою работу, фронтенд разработчик должен разбираться в вебе и понимать принципы его функционирования. Таким образом, необходимо знать:

* как работает интернет;
* протоколы HTTP/HTTPS, веб-сокеты;
* как работают браузеры;
* что такое DNS и как он работает;
* что такое доменное имя;
* что такое хостинг.

Также, будет неплохо, если вы попробуете настроить сайт на каком-нибудь бесплатном хостинге, привяжете домен к этому хостингу.

**JavaScript**

Язык программирования, который используется как при разработке клиентской стороны веб-приложения, так и серверной. При помощи JavaScript (сокращенно — JS) можно писать даже настольные и мобильные приложения, используя определенные программные платформы и библиотеки. Этот язык позволяет:

* динамически изменять разметку;
* осуществлять интерактивное взаимодействие с пользователем;
* анимировать изображения;
* совершать валидацию форм;
* управлять мультимедиа и т. д.

Другими словами, JavaScript “оживляет” страницу и добавляет ей функциональности. Хорошее владение данным языком программирования является обязательным для каждого FrontEnd разработчика.

Сергей Росоха, Software Architect с 11-летним опыта во FrontEnd/JS, отмечает важность изучения алгоритмов и структур данных на JavaScript:

*“JavaScript давно уже используется не только для разработки динамических интерфейсов пользователя, но и для написания достаточно сложной бизнес-логики. Поэтому знание алгоритмов и структур данных становится критичным для JS-разработчиков. ” (*[*источник*](https://dou.ua/lenta/articles/senior-frontend-tips/)*)*

JavaScript использует официальный стандарт ECMAScript (сокращенно - ES), который подразумевает определенное формальное описание синтаксиса, базовых объектов и алгоритмов. На данный момент существует множество различных версий ES. Работодатели чаще всего требуют знание **ES6+**.

Однако, вначале необходимо изучить чистый JavaScript и лишь потом вникать в новые стандарты. Как ни крути, а классику надо знать. Благодаря хорошему владению JS можно быстро разобраться в любой версии ES и затем освоить любой фреймворк или библиотеку.

ITVDN предоставляет возможность изучить JavaScript с нуля до продвинутого уровня при помощи курсов:

* [JavaScript Стартовый](https://itvdn.com/ru/video/javascript-starter);
* [JavaScript Углубленный](https://itvdn.com/ru/video/javascript-advanced);
* [JavaScript: Расширенные возможности](https://itvdn.com/ru/video/javascript-extended-capabilities).

**Линтеры**

Инструменты, которые позволяют анализировать качество JavaScript-кода согласно определенному стандарту ES. Они встраиваются в среду разработки и указывают на наличие несоответствий стандарту в коде, если таковые имеются. Данный механизм полезен как для одиночной разработки (для самоконтроля), так и для командной, когда каждый разработчик должен следовать одним и тем же языковым конструкциям, дабы на выходе получить единый, целостный проект. Среди известных линтеров можно выделить ESLint и Prettier.

**Тестирование**

Jest, Cypress и Enzyme — главные инструменты модульного тестирования, которое возлагается на плечи разработчика. Но что это такое?

Unit тестирование (оно же — модульное тестирование) — процесс, который заключается в создании тестов для проверки работоспособности отдельных участков написанного программистом кода. Выполняется непосредственно автором кода.

Какую пользу приносит Unit тестирование:

* существенное сокращение багов в коде;
* упрощение рефакторинга кода;
* обеспечение качественного отделения интерфейса от реализации;
* лучшее понимание написанного кода;
* возможность протестировать мельчайшие участки кода.

Все три инструмента учить не надо — достаточно научиться управляться лишь с одним из них.

**Паттерны проектирования JavaScript**

Это лучшие практики, которые описывают типичные способы решения распространенных задач, возникающих в ходе проектирования программного обеспечения. Знание шаблонов проектирования позволяет писать более чистый, понятный и читаемый код, а также избегать “изобретения велосипеда”. Более того, владение паттернами показывает вашу грамотность, как разработчика, и повышает вашу ценность в глазах работодателя, что поможет выделиться на фоне конкурентов.

Чтобы вы могли хорошенько разобраться с темой шаблонов в JavaScript, рекомендуем курс “[JavaScript Шаблоны](https://itvdn.com/ru/video/JavaScript-patterns)”.

**Алгоритмы и структуры данных**

Понимание алгоритмов и структур данных — обязательные знания для любого программиста.

К структурам данных относятся: стеки, очереди, связанные списки, графы и т. д. Изучив их, вы сможете управлять сложностью своих программ, делая их более доступными для понимания, а также разрабатывать высокопроизводительные программы, которые будут эффективно работать с памятью.

Знание алгоритмов позволит вам создавать сложные конструкции для эффективного решения широкого спектра задач. Когда говорят об алгоритмах, обычно имеют ввиду алгоритмы сортировки и поиска: сортировки прямым включением, прямым выбором, слиянием, пирамидальная, прямой поиск, бинарный, индексно-последовательный и другие.

Важно разбираться и в нотации Big O, которая описывает сложность каждого алгоритма. Данный механизм помогает определить, при каких условиях выгоднее использовать тот или иной алгоритм.

***Отметим, что как новичку, вам не надо нырять вглубь этих тем***. Будет достаточно теоретического владения, а также умения написать несколько алгоритмов. В разработке сайтов это не сильно нужно, но практически ни одно собеседование не обходит стороной тему шаблонов, алгоритмов и структур данных.

**Фреймворки JavaScript**

Это инструменты, с помощью которых создаются динамические веб/мобильные/десктопные приложения на языке JavaScript. Они ускоряют разработку веб-приложений и предусматривают четко структурированную организацию кода, повышая его качество и чистоту.

Самыми популярными фреймворками для фронтенд-разработки можно назвать **Vue.js, React и Angular**. Каждый из них предназначен для решения своего спектра задач и имеет различную степень сложности: Vue.js - самый легкий (но и с наименьшим сообществом), React - средней сложности, Angular - высокой сложности. Стоит сконцентрироваться на глубоком изучении одного фреймворка, но в то же время очень рекомендуется знать особенности и сферу применения всех вышеперечисленных технологий.

Какой фреймворк все же выбрать? Мнения на этот счет расходятся. Инструментарий выбирается индивидуально под проект и трудно предугадать, какие задачи вам нужно будет решать. *На данный момент наибольшей популярностью пользуется React*, поскольку он сбалансирован как с точки зрения сложности, так и с точки зрения богатства функционала. К тому же, наибольшее количество вакансий припадает именно на этот фреймворк, а значит, что он интересен не только разработчикам, но и бизнесу.

Каждый из фреймворков доступен для изучения на платформе ITVDN. По направлению Angular:

* [Angular 11.0 Базовый](https://itvdn.com/ru/video/angular-essential);
* [Angular Углубленный](https://itvdn.com/ru/video/angular-advanced).

По направлению React:

* [React Базовый](https://itvdn.com/ru/video/react-essential);
* [React Углубленный](https://itvdn.com/ru/video/react_adv).

По направлению Vue.js:

* [Vue.js Стартовый](https://itvdn.com/ru/video/vuejs);
* [Vue.js Базовый](https://itvdn.com/ru/video/vuejs-essential).

**Инструменты управления состоянием приложения (State Management)**

Большие веб-приложения постоянно растут в сложности, а потому хранить данные, которые могут использоваться в разных местах становится все сложнее. Для решения этой проблемы были придуманы отдельные библиотеки, идущие параллельно с фреймворками — они служат для управления глобальным хранилищем данных всего веб-приложения и называются менеджерами состояний. Их применение уже стало стандартом во FrontEnd разработке. Для девелоперов, которые используют Vue.js, это VueX, для React-разработчиков — Redux и MobX, для Angular-девов — RxJS, NgRx.

**JavaScript Core (DOM, AJAX, JSON)**

**DOM** (**D**ocument **O**bject **M**odel) - объектное представление исходного HTML-документа. Ключевым является понятие DOM-дерева, которое описывает структуру страницы. С помощью объектной модели JavaScript получает полную власть над HTML-документом: возможность редактировать, удалять и добавлять элементы и атрибуты HTML, менять CSS код и т. д.

**AJAX** (**A**synchronous **J**avaScript **A**nd **X**ML) - это синтез технологий JavaScript и XML, который фактически представляет собой комбинацию:

* встроенного в браузер XMLHttpRequest-объекта (чтоб запрашивать данные с веб-сервера);
* JavaScript и HTML DOM (чтобы отображать или использовать данные).

AJAX позволяет веб-страницам совершать асинхронное обновление, обмениваясь данными с веб-сервером. Благодаря этой технологии страница не нуждается в перезагрузке - обновляется лишь конкретная ее часть (вспомните ленту новостей в социальных сетях).

**JSON** (**J**ava**S**cript **O**bject **N**otation) - это общий формат обмена данными. Позволяет совершать обмен информацией между программными продуктами, написанными на разных языках. Таким образом, клиент, использующий JavaScript, может легко передавать данные на сервер, который реализован с помощью Ruby/Java/PHP.

Все три технологии являют особую ценность для каждого веб-разработчика и раскрывают организацию работы интернет-приложения.

**БЭМ**

“**Б**лок, **Э**лемент, **М**одификатор” — методология, предусматривающая компонентный подход к разработке веб-страниц, в основе которого лежит принцип разделения интерфейса на независимые блоки. Подход БЭМ позволяет повторно использовать существующий код в создании других страниц с сохранением всех его свойств (размеры, шрифт, цвет и т. д.).

**Webpack**

Мощный сборщик модулей, который позволяет скомпилировать в один файл несколько разных модулей. Используется во время работы над объемными проектами. Успешно применяется как во фронтенд-разработке, так и при создании бэкенд-приложений.

**Gulp / Grunt**

Системы сборки, которые автоматизируют рутинные задачи разработчиков: минификацию кода, оптимизацию изображений, тестирование, анализ качества кода и прочее. Подходят при разработке небольших проектов.

**TypeScript**

Кроссплатформенный строго типизированный язык, который является расширением JavaScript. Строгая типизация позволяет уменьшить количество потенциальных ошибок в исходном коде, написанном на TypeScript. Также, этот язык реализует концепции, которые близки объектно-ориентированным языкам, таким как C#, Java и подобным. TypeScript повышает скорость и удобство написания сложных комплексных программ, вследствии чего их становится легче поддерживать, масштабировать и тестировать. *В большинстве случаев его необходимо изучать лишь тем, кто ориентирован на работу с фреймворком Angular.*

**SVG**

Язык разметки масштабируемой векторной графики. Изображения на странице, сделанные с помощью SVG, корректно отображаются на экранах с различным разрешением, не теряя при этом своего качества, в отличии от традиционных растровых .jpeg, .png и других.

**Английский язык**

Знание английского языка является одним из основных требований к фронтенд-разработчику, поскольку большое количество полезной информации находится именно на англоязычных сайтах. Уровень чтения технической документации будет достаточным для комфортного пользования иностранными ресурсами.

**Итоги**

FrontEnd разработчик — достаточно универсальный боец в мире веб-разработки. Он должен уметь и верстать, и создавать логику работы клиентской части, и понимать работу серверной части веб-приложения. Для освоения такого обширного инструментария стоит запастись временем, терпением и упорством. Перечисленные в статье средства разработки сайтов также имеют аналоги, поскольку для решения разных задач подходят разные веб-инструменты. Однако мы выбрали самые популярные и эффективные из них.

Если у вас остались вопросы о последовательности и необходимости изучения тех или иных технологий, ответы вы можете найти в видео [”Как стать FrontEnd разработчиком?”](https://itvdn.com/ru/webinars/description/how-to-frontend2020), в котором подробно рассматриваются основные технологии создания клиентских веб-приложений.

Для тех, кто хочет стать FrontEnd разработчиком, на ITVDN создана [комплексная программа обучения](https://itvdn.com/ru/specialities/frontend-developer), которая включает в себя 42 видео курса.

Желаем вам успехов в достижении ваших целей!

Оставайтесь с ITVDN!