**Тема 3. «Как работает интернет. Frontend и Backend.»**

**Работа браузера, HTTP. Установка программного обеспечения. Создание простого сайта.**

Тем, кто впервые решил заняться созданием, раскруткой или администрированием сайтов, часто бывает сложно понять основные термины и взаимосвязи между ними. На этом занятии поговорим о том, что такое домен и сайт, для чего нужен хостинг и домен, в чем разница между ними, чем отличается FrontEnd от BackEnd и какое программное обеспечение нам потребуется для работы.

**В чем разница между доменом и хостингом**

**Сайт, домен и хостинг – что это такое?**

Сайт – это, по сути, набор различных файлов: шаблоны страниц, скрипты, картинки, видео и т. п. А также база данных, в которой хранятся тексты страниц, характеристики товаров, данные зарегистрированных пользователей и тому подобное.

Доменом часто называют имя, наименование сайта в интернете, хотя это не совсем верно. Правильнее использовать термин "доменное имя". К примеру, ru.wikipedia.org, hoster.ru – это домены (доменные имена).

Хостингом называют место или компанию, где (у которой) расположены файлы сайта и его база данных. То есть, если кто-то говорит, "мой хостинг – hoster.ru", это означает, что владелец сайта арендовал место на наших серверах для размещения файлов своего сайта.

**Какая разница между доменом и хостингом?**

Примерно такая же, как между адресом вашего дома и участком, на котором он стоит. Это совершенно разные, хотя и связанные, понятия. Адрес необходим дому для того, чтобы его могли найти ваши гости (посетители сайта). Участок – чтобы дом (сайт) в принципе существовал и в него можно было попасть.

**Домен (доменное имя)**

Первое, что нужно сделать для размещения вашего сайта, – это зарегистрировать домен. Это должно быть простое, запоминающееся имя, отражающее ваш бренд или вид деятельности.

Доменное имя придумали для того, чтобы не приходилось запоминать сложные комбинации цифр IP-адреса, по которому идет обращение к сайту в сети. Вы пишете в браузере имя сайта, а система DNS-записей преобразовывает его в IP-адрес, по которому сейчас находится сайт – так происходит соединение.

**Для чего нужен хостинг сайта?**

Веб-хостинг – это платная услуга. Поэтому первое, что спрашивают многие начинающие владельцы сайтов, – можно ли обойтись без него? Для чего нужен хостинг для сайта, если файлы и базу данных можно разместить на домашнем компьютере?

Можно, но в домашних условиях очень сложно и дорого обеспечить стабильную работу сайта. Вам понадобятся:

* круглосуточно работающий мощный компьютер, способный обработать весь поток посетителей сайта;
* хороший скоростной канал связи, а также резервный от другого провайдера – на случай проблем на первом канале;
* мощный ИБП или даже генератор – на случай отключения электроэнергии;
* много знаний по сетевым и веб-технологиям, чтобы поднять на домашнем сервере необходимое программное обеспечение, настроить сеть, обеспечить защиту от хакерских атак и т. п.

У компаний-хостеров все это предусмотрено по умолчанию. Серверы хостинга располагаются в сертифицированных ЦОД, с круглосуточным обслуживанием, резервными каналами связи и питания, передовым ПО. Гораздо проще оплатить готовые качественные услуги веб-хостинга, чем пытаться создать такое у себя дома с нуля.

# **Frontend vs backend: различия и особенности разработки**

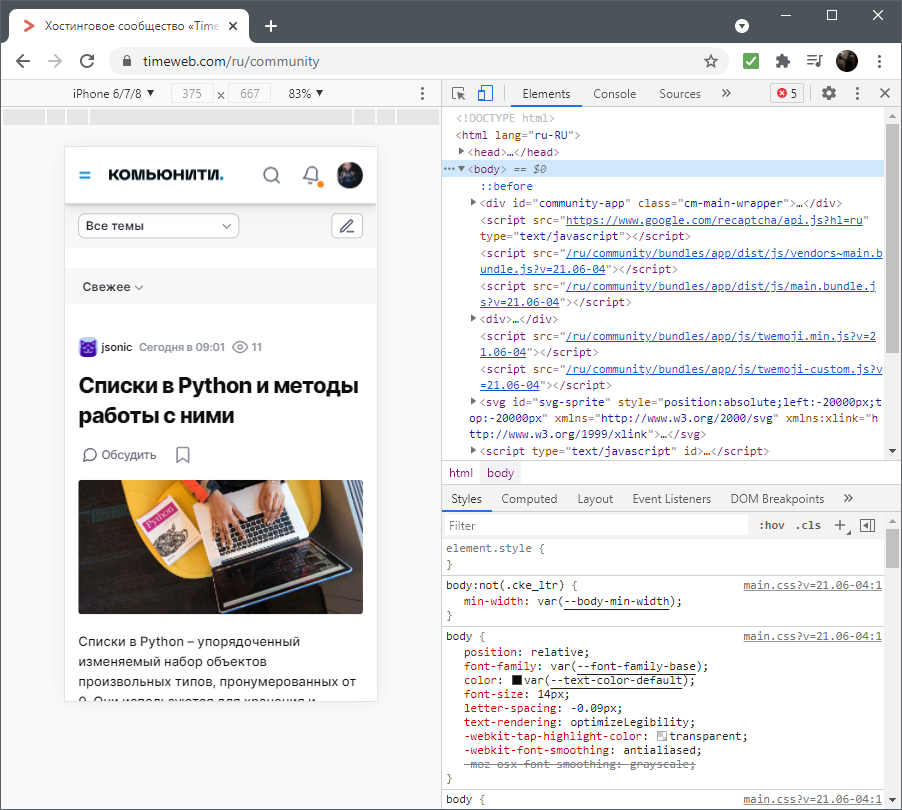
Frontend- и backend-разработка тесно связаны между собой и не могут существовать по отдельности. Однако это совершенно разные направления программирования как по типу выполняемых задач, так и по общему предназначению.

В этом материале речь пойдет о различиях этих направлений, их особенностях и требованиям к разработчикам.

## Понятие frontend-разработки

Под понятием frontend подразумевается разработка видимого для пользователя интерфейса и всех функций, с которыми он может взаимодействовать. По сути, когда вы переходите на любой сайт, то видите там кнопки, текст, различную анимацию и другие составляющие – все это реализовано при помощи фронтенда. Для создания этих элементов используются три разных языка – HTML, CSS и JavaScript.

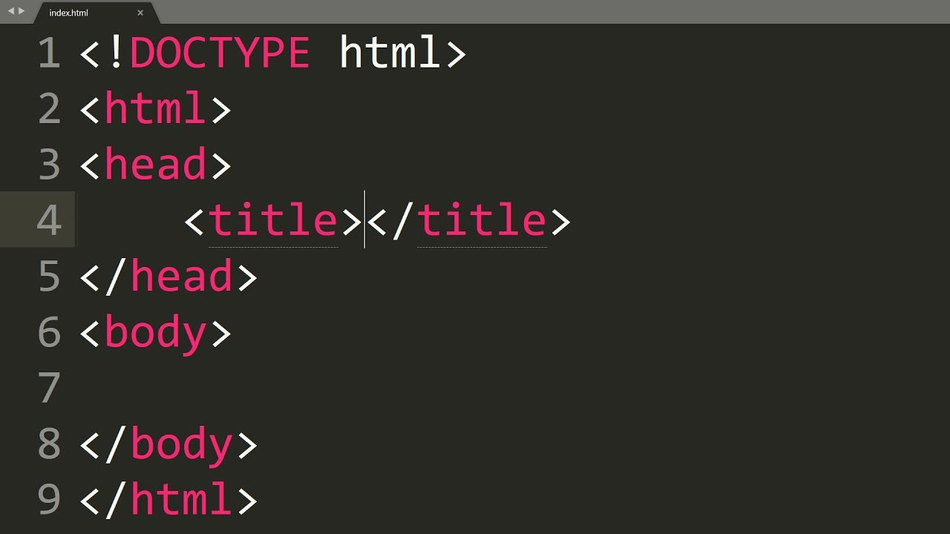
Посмотрите на структуру страницы прямо сейчас – для этого на любом сайте вызовите консоль разработчика, нажав клавишу F12. Вы увидите набор строчек кода, которые и формируют внешний вид данной страницы.



Код страницы используется для описания всего видимого: шрифтов, картинок, размеров элементов, анимаций, всплывающих окон, звуков при нажатиях и всего другого, с чем может взаимодействовать пользователь. За каждую из частей интерфейса отвечают разные языки. Как раз те, что и должен выучить frontend-разработчик.

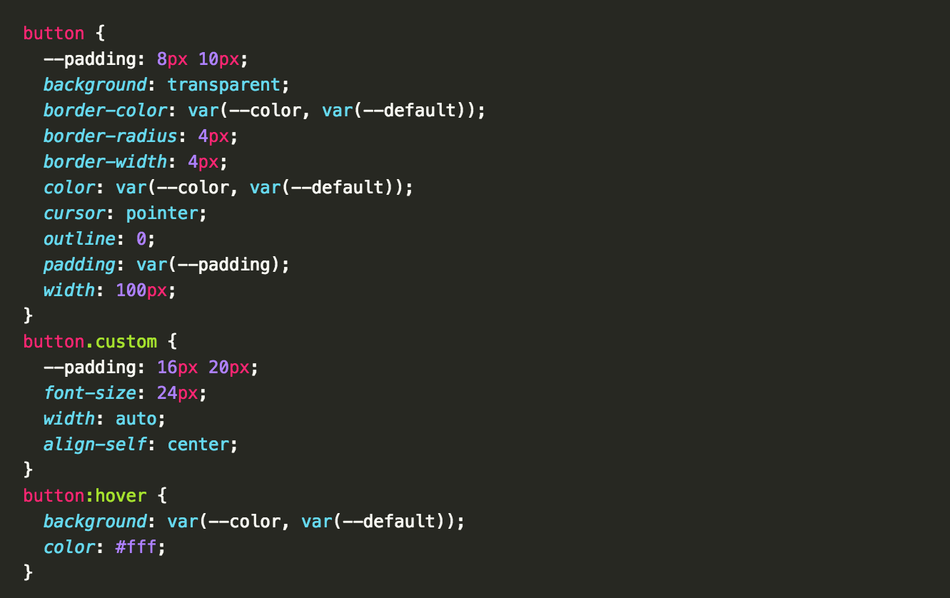
### HTML

Основной инструмент в этой сфере – язык гипертекстовой разметки HTML. Он нужен в основном для разметки документа, то есть страницы в браузере. С помощью него разработчик создает структуру, добавляет заголовки, списки и осуществляет другое форматирование контента.



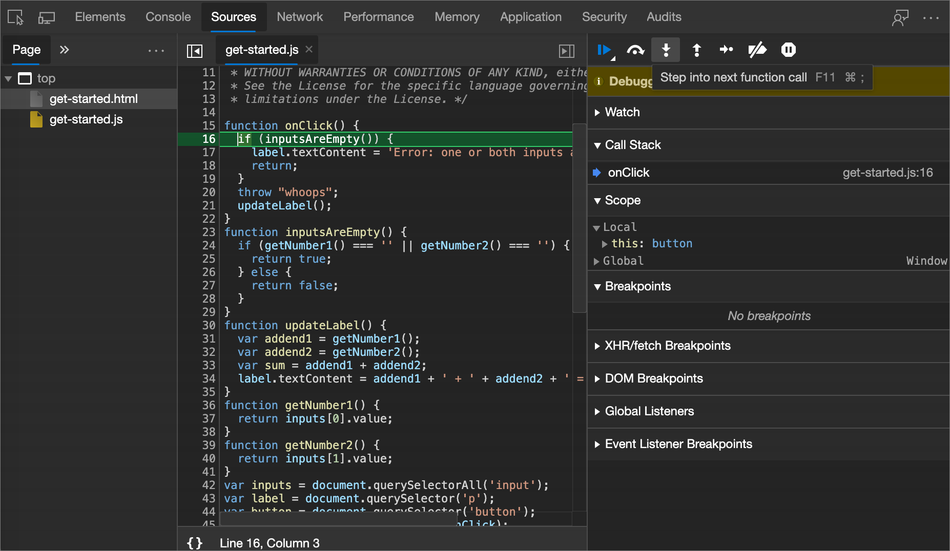
### CSS

Если разработчик учит HTML, он обязательно осваивает и CSS (Cascading Style Sheets). Этот язык отвечает за внешний вид страницы. С его помощью вы работаете с цветами, шрифтами и расположением различных блоков. Если простыми словами, то CSS используется для красивого оформления страницы и настройки ее внешнего вида уже после того, как основная структура была написана при помощи HTML.



### JavaScript

С помощью JavaScript реализуется выполнение различных действий на странице, то есть добавляется анимация и отклик на запросы пользователя. Например, страница реагирует на перемещение курсора и клики мышкой, изменяя поведение элементов в соответствии с действиями пользователя. Благодаря JS осуществляется отправка и получение данных с сервера без необходимости перезагрузки страницы, а значит, некоторые задачи реализуются более просто, например, когда речь идет об отправке и получении сообщений.



Фронтенд – это сфера, в которой разработчик создает видимые для пользователя элементы интерфейса и все функции для взаимодействия с сайтами и приложениями. Все это тесно связывается с бэкендом, о котором пойдет речь далее.

## Backend-разработка

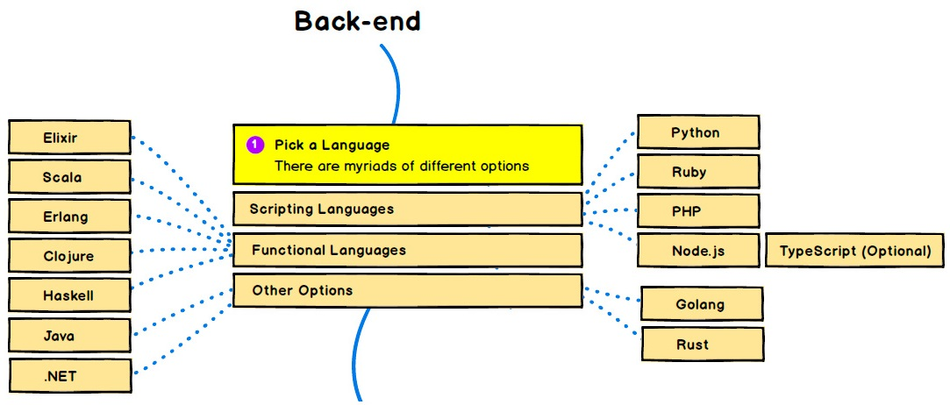
Считается, что backend-разработка сложнее фронтенда, поскольку программист обходится без видимых элементов интерфейса, ведь он работает над логикой сайта. Пользователь не видит эту сферу, потому что все действия осуществляются вне его браузера и даже компьютера. Как только поступает запрос (например, когда вы нажали Enter при запросе в поисковике), сигнал сразу же отправляется на сервер, где и обрабатывается для дальнейшего вывода информации на экран. Это и есть логика сайта, заключающаяся в трех простых шагах:

1. отправка информации от пользователя,
2. ее обработка на сервере,
3. получение информации и форматирование кода в читаемый вид.

### Используемые языки программирования в бэкенде

Если в случае с фронтендом нет никакой вариативности, так как разработчики всегда используют описанные языки программирования для достижения желаемого результата, то вот с backend все обстоит немного иначе. Выбор языка и инструментов зависит от сервера. Зачастую ограничений нет, и разработчик выбирает практически любой универсальный ЯП. К таким относятся:

* Java;
* PHP;
* Python;
* Ruby и другие.



В большинстве случаев бэкенд-разработчику понадобятся знания по управлению базами данных, которых тоже существует немало. Популярнее всего MySQL, но есть еще и PostgreSQL, SQLite, MongoDB. От выбранной системы управления базами данных меняются и обязанности разработчика, поскольку действия выполняются разными путями.

Приведенное выше описание дает ответ на вопрос «что такое backend‎».‎

## Варианты взаимодействия frontend и backend

В завершение рассмотрим несколько примеров того, как взаимодействуют между собой две эти сферы разработки. О принципах работы уже шла речь выше, поэтому остается только разобраться во взаимосвязи этих сфер. Разделим принципы работы на четыре популярных варианта:

1. Первый тип подразумевает отправку HTTP-запроса на сервер, где она ищется, встраивается в шаблон и возвращается пользователю в читаемом виде HTML-страницы.
2. Есть еще инструментарий под названием AJAX. В таком случае запрос отправляется при помощи JavaScript, который подключен в веб-обозревателе. Ответ возвращается в XML или JSON, а с чтением этих форматов отлично справляется JS.
3. Существуют одностраничные сайты, загружающие данные без предварительного обновления страницы. Осуществляется данная операция при помощи AJAX или фреймворков Angular и Ember.
4. Подключаемые библиотеки Ember или React предназначены для использования приложения одновременно на сервере и в клиентской части. Две рассматриваемые сферы связываются через AJAX и HTML-код с обработкой на сервере.

Подводя итоги, хочется отметить, что требования к специалистам в backend и frontend разные, обычно каждый занимается своим делом, но есть и fullstack-разработчики, одновременно программирующие как логику сайта/приложения, так и его внешний вид. Вам понадобится ознакомиться с приведенными схемами и языками программирования, чтобы понять, что вам ближе.

Программное обеспечение (устанавливаем бесплатную версию Community)

[PyCharm: the Python IDE for Professional Developers by JetBrains](https://www.jetbrains.com/pycharm/?ysclid=l4e130nhtl57266103)

Практика:

Установить среду разработки, создать HTML файл, сохранить его и запустить в своем браузере. Проверить корректный вывод текста.