

# Web технологии

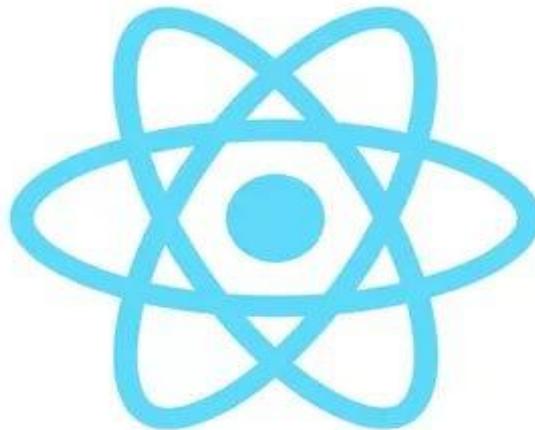


# React.js

С помощью React.js разработчики могут создавать повторно используемые компоненты (reusable components), каждый из которых будет со своим собственным “state”, то есть “внутренним состоянием”. Если на странице необходимо обновить только один компонент (например, пользователь нажимает кнопку “Мне нравится”), React изменяет только этот компонент (к примеру, появляется большой палец вверх), не обновляя всю страницу целиком.

Для этого React использует Virtual DOM — способ представления UI (интерфейс пользователя), который хранится в памяти и постоянно синхронизируется с “реальным” DOM. Неважно какой браузер вы используете, Virtual DOM всегда будет работать быстрее, так как он не включает в себя все тяжеловесные части реального DOM.

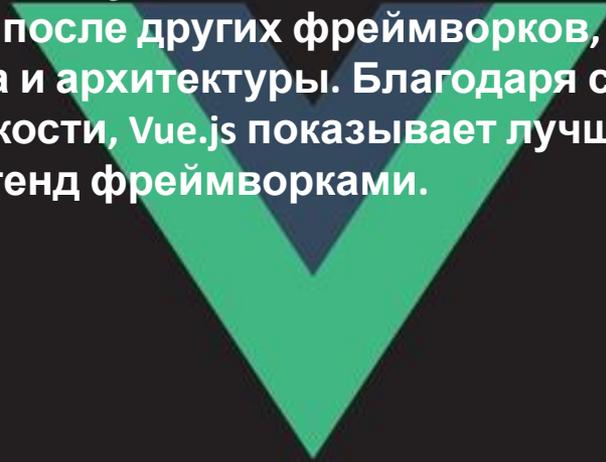
Такие крупные компании, как Airbnb, Instagram и Netflix уже несколько лет успешно используют React.js, и мы видим, как увеличивается число компаний, заинтересованных в данной технологии. Интерес к React, среди разработчиков, стремительно набирает темпы, именно поэтому в ближайшие годы данный фреймворк нисколько не потеряет в популярности.



# Vue.js

Еще одним важным игроком на арене фронтенд-разработки является Vue.js — JavaScript-фреймворк с самым быстрорастущим и полным энтузиазма сообществом разработчиков на данный момент. Vue.js — подходящий выбор для тех, кто хочет поэкспериментировать с новым фреймворком, поскольку у него очень подробная документация и низкая кривая обучаемости.

К Vue.js быстро привыкаешь после других фреймворков, в виду его сходства с Angular и React, с точки зрения дизайна и архитектуры. Благодаря своему маленькому размеру (всего 20 КБ), скорости и гибкости, Vue.js показывает лучшую производительность, по сравнению с другими фронтенд фреймворками.



Vue.js

# jQuery

Судя по всему, сообщество разработчиков постепенно отказывается от использования самой известной JavaScript-библиотеки. Одной из причин может быть то, что jQuery плохо уживается с более новыми библиотеками, такими как React. Также, в настоящее время, современные браузеры поддерживают множество jQuery-like функций, что во многих случаях делает jQuery ненужным. По этой причине мы видим, что сейчас гораздо чаще используется чистый JavaScript. Например, в GitHub недавно объявили, что они заменили весь jQuery на чистый JavaScript.



Да, jQuery теряет свою былую популярность, но 300.000 загрузок в день демонстрируют хороший рост на 300%, по сравнению с началом 2017 года. Еще один удивительный факт: 90% из 500000 самых популярных веб-сайтов, как и прежде используют jQuery! Это наглядно иллюстрирует тот случай, когда библиотека не обязательно должна быть новой и крутой, чтобы оставаться актуальной. Выбор какой-либо конкретной библиотеки полностью зависит от типа проекта, для которого вы хотите ее использовать.

# Kotlin

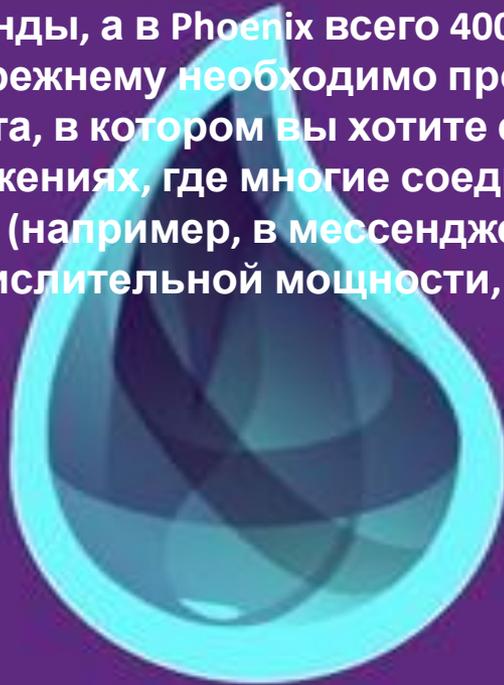
В последние годы огромную популярность приобрел язык программирования Kotlin. Он является современной и более простой альтернативой Java. Kotlin, преимущественно, начал набирать аудиторию в прошлом году, когда компания Google объявила, что Kotlin получил официальную поддержку для разработки Android-приложений. После этого такие компании, как Slack и Netflix начали разрабатывать свои приложения для Android с помощью Kotlin. Роб Флетчер, старший разработчик программного обеспечения в Netflix, сказал:



Согласно исследованию на сайте Pusher, в первые годы существования (с 2011 года) Kotlin использовали, в основном, опытные разработчики, но с момента конференции Google в 2017 году, именно молодые разработчики, особенно студенты, стали причиной бурного роста.

# Elixir и Phoenix

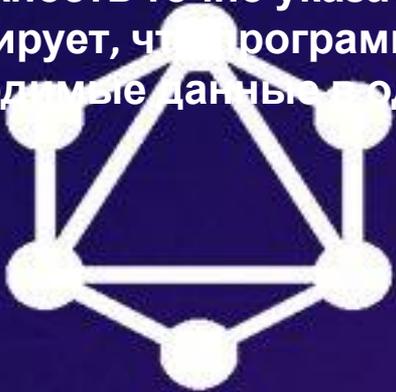
Phoenix, встроенный в функциональный язык программирования Elixir, является относительно новым фреймворком, о котором все чаще и чаще говорят в сообществе разработчиков. Одна из причин такой заинтересованности со стороны разработчиков — высокая производительность. Например, обработка большого JSON-запроса занимает в Rails 1.5 секунды, а в Phoenix всего 400 миллисекунд. Но не спешите, для начала по-прежнему необходимо проверить, подходит ли данная технология для того типа проекта, в котором вы хотите ее применить. Например, Elixir и Phoenix сильны в real-time приложениях, где многие соединения должны обрабатываться одновременно (например, в мессенджерах). Но для приложений, которые требуют большой вычислительной мощности, Elixir не совсем подходит.



# GraphQL

GraphQL — это язык запроса данных с открытым исходным кодом и один из самых современных способов создания API, разработанный Facebook. Популярность GraphQL стремительно растет, и такие крупные компании, как Twitter, Shopify и Amazon уже начали работать с этой технологией.

GraphQL помогает разработчикам справиться с часто встречающимися проблемами, поскольку значительно упрощает загрузку и обработку данных. Самым важным различием между REST API и GraphQL является то, что GraphQL предоставляет возможность точно указать, какие данные вы хотите получить обратно из API. Это гарантирует, что программное обеспечение, вызывающее API, получит все необходимые данные в одном запросе.



# GraphQL

Это удобно в том случае, когда приложение содержит много вложенных моделей и ассоциаций. GraphQL помогает, например, при разработке мобильных приложений, загружать только соответствующие данные, и тем самым снижает нагрузку на мобильную сеть. GraphQL берет на себя большую часть накладных расходов (overhead), которые возникают в REST API, и таким образом обеспечивает преимущества как для фронтенда, так и для бэкенда.