React.js (также называют React.js или ReactJS) — это сторонняя библиотека языка JavaScript, созданная для разработки интерактивных пользовательских интерфейсов. Благодаря этой библиотеке развертывание веб-приложений и интерактивных UI-интерфейсов значительно ускоряется (по сравнению с программированием на «ванильном» JS). Кроме того, разработчику на React необходимо писать гораздо меньшее количество кода (чем если бы он пытался запрограммировать функциональность интерфейса только на стандартном JavaScript). Чтобы задать функциональность компоненту веб-приложения в «Реакт» — достаточно описать то, как будут выглядеть определённые части интерфейса в различных состояниях. Изменять существующий код, при этом, не обязательно. Кроме того, логика описывается на стандартном JS, а не в шаблонах, поэтому можно передавать любые данные по всему веб-приложению. Если проводить аналогию с реальным миром, то разработку на React.js можно представить как процесс сбора Lego-конструктора, другими словами, вы создаёте компоненты многоразового использования. Такие блоки представляются отдельными, независимыми частями одного большого интерфейса, ну а в собранном состоянии эти блоки образуют весь пользовательский интерфейс веб-приложения целиком.

**Особенности React.js**

Это библиотека для frontend-разработки с открытым исходным кодом и компонентной основой. Ее Ключевыми особенностями можно назвать следующие характеристики:

* Высокий уровень производительности. Достигается за счет использования виртуального DOM. Подробнее о нем мы скажем чуть ниже.
* Наличие *компонентов*. Библиотека позволяет разделить страницу на составные части: такие компоненты, как строительные плиты, являются частью одного большого пользовательского интерфейса. При этом, логика таких компонентов написана на JS, что оптимизирует работу программиста, упрощая логику разработки и позволяя использовать компоненты повторно.
* Наличие условных операторов. Это особенность библиотеки сводится к тому, что мы можем писать условные выражения. При этом данные в браузере выводятся в соответствии с условиями, которые указаны в JSX.
* Присутствие виртуального DOM. Document Object Model — это независимый от языка программный интерфейс, который используется для доступа к содержимому HTML- или XML-документа.

## Зачем нужен React.js, если есть JavaScript

React — не фреймворк, а сторонняя библиотека. И в этом различии проявляется сила его использования, например, в соединении с другими библиотеками. Фреймворк, в основном, имеет все функции и должен быть написан только одним способом, в то время как библиотеки — могут быть использованы в соответствии с потребностями разработчика.

**Плюсы и минусы**

Приложения React создаются с использованием компонентов; эти компоненты являются многоразовыми, то есть нет необходимости переписывать код для каких-то конкретных вещей. Раньше многие вещи, такие как навигационная панель, нижний колонтитул или статусная строка, нужно было добавлять на разные страницы. Используя компоненты, мы можем просто написать их один раз и использовать в любом месте.

**Плюсы использования React**

Но компонентность — не единственное преимущество. Вот другие плюсы «Реакт»:

* Огромное сообщество. React имеет огромную поддержку среди разработчиков и широчайшее сообщество. Многие разработчики-любители и профессионалы — пытаются решить проблемы новичков.  Вы можете решить любую проблему в своем приложении, просто погуглив ее в интернете. С нестандартными случаями — поможет сообщество.
* Легкость в освоении: поначалу React, конечно, кажется сложным. Но после того, как вы поймете основные компоненты и их работу — кодинг пойдет как по маслу. С помощью простых функций можно разрабатывать очень сложные приложения. Вы должны помнить: как технология «Реакт» прост в изучении, но сложен в освоении.
* React Native. Большинство веб-приложений также нуждается в версии для мобильных устройств. И если вы сталкивались с мобильной разработкой, то наверняка знаете как непросто сделать мобильное приложение для двух разных архитектур: Android и iOS. Но эту проблему можно решить, изучив React, ведь в нем существует так называемый React Native. RT оптимизирует развертывание мобильных приложений для Android и iOS. При этом подход использует аналогичные методологии, что и сам React. Поэтому — изучение React также поможет стать мобильным разработчиком.
* Широкая масштабируемость и гибкость. React можно использовать со многими другими библиотеками, которые обеспечивают широкий спектр функциональности, инструментов и возможностей.

**Недостатки использования React**

Без минусов — никуда. Вот основные недостатки библиотеки:

* Сложность обучения. Можно сказать, что у этой библиотеки кривая обучения. Как уже говорилось ранее, изучить React может быть легко, но вот освоить его разносторонне — очень сложно. На обучение может потребоваться много времени и практики. Особенно, если вы никогда не работали с JavaScript.
* Библиотека фронтенда. «Реакт» — только фронтенд-технология и для работы в бэкенде вам нужно изучить другие инструменты (например, Express или Next.js). «Реакт» можно использовать, если нужно развернуть одностраничное или статичное приложение.
* Не SEO-friendly из коробки. Это одна из основных проблем библиотеки. Веб-приложения на React.js имеют ограничения для SEO. Это можно понять на простом примере: вы отправляете кому-то ссылку на сайт, когда вы отправляете ссылку — с ней передается небольшое изображение и описание. Но у сайтов на React (или других веб-приложений) ситуация иная: весь JavaScript должен быть обработан, прежде чем вышеуказанные данные смогут быть переданы. Таким образом, краулеры ПС также не смогут прочитать нужные им данные о сайте (и это может снизить рейтинг домена в поиске). Эту проблему можно решить с помощью Next.js, который задействует серверный рендер.
* Частые изменения. Над ошибками и проблемами React постоянно работают его создатели. Это также означает, что за небольшой промежуток времени может быть выпущено значительное количество обновлений. Например, недавно было опубликовано обновление ES6. И фактически оно поменяло привычные паттерны разработки «Реакт». Так что — нужно постоянно оставаться в курсе и менять приложение в соответствии с обновлениями библиотеки.

Пример использования React.js:







