

# 1 Добро пожаловать в React

React — популярная библиотека, используемая для создания пользовательских интерфейсов. Она была создана в Facebook с целью решить ряд проблем, связанных с крупномасштабными сайтами, управляемыми данными. В момент выпуска библиотеки в 2013 году этот проект воспринимался с некоторой долей скептицизма, поскольку соглашения по React довольно необычны.

В попытке не отпугнуть новых пользователей основная команда разработчиков React написала статью под названием *Why React?* (Почему именно React?), в которой рекомендовалось Give It (React) Five Minutes (Уделить ей (React) всего пять минут). Им хотелось сначала воодушевить людей на работу с библиотекой, прежде чем те подумают, что подход разработчиков слишком безумен.

React действительно является небольшой библиотекой, поставляемой без того набора инструментов, который может понадобиться для создания ваших приложений. Но уделите ей пять минут.

Да, благодаря React прямо в вашем JavaScript создается код, похожий на HTML. И разумеется, все эти теги требуют предварительной обработки перед запуском в браузере. А для этого, скорее всего, понадобится встроенный инструмент, такой как Webpack.

Если вы, как и мы, читали эту статью, то, возможно, были впечатлены обещаниями, связанными с новой библиотекой JavaScript, способной решить все проблемы с DOM (объектной моделью документа); с библиотекой, которая всегда будет проста в применении и никогда не причинит неудобств.

А потом начали возникать вопросы: как можно будет преобразовать этот JSX? Как загрузить данные? Куда девать CSS? Что такое декларативное программирование? В каждом направлении возникало еще больше вопросов о том, как внедрить эту библиотеку в повседневную работу. С каждым разговором появлялась новая терминология, возникали новые технические приемы и все больше вопросов.

## Препятствия и трудности

Уделив несколько минут изучению компонентов React, вы откроете для себя новые горизонты осмысления подходов к веб-разработке. Разумеется, не обойдется и без трудностей, которые придется преодолеть, прежде чем приступить к созданию готового к практическому применению кода на основе React.

### React как библиотека

Во-первых, стоит заметить, что библиотека невелика по размеру и используется только для одной части всей работы. Она не содержит инструментария, ожидаемого от традиционного фреймворка JavaScript. Основные решения о том, какими средствами экосистемы воспользоваться, принимают разработчики. Кроме того, постоянно появляются новые наборы инструментов, а старые отходят на второй план. При таком количестве библиотек совершенно не удивляет ощущение, что за ними всеми невозможно угнаться.

### Новый синтаксис ECMAScript

React вступила в эпоху зрелости в весьма важный, но хаотичный период в истории JavaScript. Раньше Европейская ассоциация производителей компьютеров (ЕСМА) довольно редко выпускала новые спецификации. Иногда на выпуск одной из них уходило до десяти лет. Это значило, что разработчикам не нужно было очень часто осваивать новый синтаксис.

А с 2015 года новые функциональные свойства языка и синтаксические дополнения станут выпускаться каждый год. Номерные выпуски системы (ECMAScript 3, ECMAScript 5) будут заменены ежегодными (ECMAScript 2016, ECMAScript 2017). По мере развития языка первопроходцы сообщества React стремятся использовать новый синтаксис. Зачастую это говорит о том, что документация предполагает знание синтаксических новинок ECMAScript. Если вы не знакомы с самой последней спецификацией, то просмотр кода React может вызвать неуверенность в своих силах.

### Популярность функционального JavaScript

Помимо изменений, появляющихся на уровне языка, существует также множество подвижек для функционального программирования на JavaScript. Заведомо функциональным языком JavaScript не является, но в его коде могут использоваться функциональные методы. В React придается особое значение приоритету функционального программирования над объектно-ориентированным. Этот сдвиг в мышлении может привести к существенным преимуществам в таких областях,

как пригодность к тестированию и повышение производительности. Но при таком количестве материалов по React, предполагающих осмысление парадигмы, усвоить сразу весь объем информации будет нелегко.

## Утомительный JavaScript

Разговоры об утомительности JavaScript (<https://medium.com/@ericclemmons/javascript-fatigue-48d4011b6fc4>) превратились уже в некое клише, но такое мнение сложилось только из-за сложности сборки программного продукта. В прошлом файлы JavaScript просто добавлялись к вашей странице. Теперь же они должны пройти процесс сборки, обычно в виде автоматизированного непрерывного процесса поставки программного продукта. Имеется формируемый синтаксис, который нужно транспилировать для работы во всех браузерах. Имеется JSX, подлежащий преобразованию в JavaScript. Имеется SCSS, возможно требующий предварительной обработки. Эти компоненты нуждаются в тестировании, которое они должны успешно пройти. Вам могла бы понравиться React, но теперь еще нужно освоить Webpack, справиться с разбиением кода, его сжатием, тестированием и т. д. и т. п.

## Почему React будет не трудно изучить

Цель данной книги — избежать путаницы в процессе обучения; мы даем материал в четкой последовательности, создавая прочную обучающую основу. Начнем с обновления синтаксиса, чтобы познакомить вас с последними свойствами JavaScript, особенно с теми, которые часто используются с React. Затем представим функциональный JavaScript, так что вы сможете сразу же применить эти методы и понять парадигму, породившую React.

Затем мы рассмотрим основополагающие сведения о React, вы сможете создать свои первые компоненты и получить ответы на вопросы, зачем и как нужно транспилировать код. После этого заложим площадку под новое приложение, позволяющее пользователям сохранять и упорядочивать цвета. Оно будет создано с использованием библиотеки React, затем код усовершенствуется благодаря ее новейшим методам. Мы расскажем о библиотеке Redux как о контейнере для данных клиента и завершим разработку приложения, внедрив Jest-тестирование и маршрутизацию с помощью средства React Router. В заключительной главе представим универсальный и структурно однородный код и усовершенствуем органайзер цветов, переведя его на сервер.

Надеемся на то, что ваша работа существенно ускорится благодаря применению экосистемы React наряду с подходом, предусматривающим не только поверхностное изучение, но и приобретение инструментов и навыков, необходимых для создания настоящих React-приложений.

## Будущее библиотеки React

React пока не утратила своей новизны. Она дошла до стадии стабильности основных функциональных свойств, но даже они еще могут измениться. В следующие версии библиотеки будет включено средство Fiber — новая реализация основного алгоритма React, целью которого является увеличение скорости выдачи изображения на экран. Выстраивать предположения о том, насколько оно повлияет на React-разработчиков, пока рано, но оно, несомненно, отразится на скорости, с которой приложения выводят на экран изображение и обновляют его.

Многие из этих изменений связаны с целевыми устройствами. В данной книге рассматриваются способы разработки одностраничных веб-приложений с помощью React, но не нужно предполагать, что браузеры являются единственным местом, где могут работать такие приложения. Средство React Native, выпущенное в 2015 году, позволяет пользоваться преимуществами React-приложений в программах, предназначенных для работы под iOS и Android. Хотя об этом еще рано говорить, но уже появился фреймворк React VR для создания интерактивных приложений с поддержкой виртуальной реальности как способ разработки 360-градусных интерфейсов с использованием React и JavaScript. Команда библиотеки React настроит вас на быструю разработку восприятий для широкого диапазона размеров и типов экрана.

Мы надеемся предоставить вам достаточно прочную базу, позволяющую приспособиться к изменениям экосистемы и к созданию приложений, способных работать на платформах за пределами браузера.

## Нужно идти в ногу со временем

По мере внесения изменений в React и сопутствующие инструменты иногда что-то перестает работать. Фактически некоторые из будущих версий данных утилит могут привести к неработоспособности ряда примеров, приводимых в данной книге. Но вы все равно можете придерживаться этих примеров. Точная информация о версии будет предоставлена в файле `package.json`, чтобы вы могли установить упомянутые там пакеты с нужной версией.

Кроме этой книги, обо всех последних изменениях можно узнать из новых публикаций в официальном блоге React (<https://facebook.github.io/react/blog/>). При выпуске новых версий React основная команда разработчиков опубликует там подробную информацию с описанием изменений и нововведений.

Существует также множество популярных React-конференций, которые можно посетить, чтобы получить самую свежую информацию о библиотеке. Если не получается принять в них личное участие, то можно изучить материалы сразу же после

проведения конференций, поскольку зачастую они публикуются на YouTube. К их числу относятся:

- ❑ *React Conf* (<http://conf.reactjs.org/>) — конференция, спонсируемая Facebook, в Санта-Кларе, Калифорния;
- ❑ *React Rally* (<http://www.reactrally.com/>) — конференция сообщества в Солт-Лейк-Сити;
- ❑ *ReactiveConf* (<https://reactiveconf.com/>) — конференция сообщества в Братиславе, Словакия;
- ❑ *React Amsterdam* (<https://react.amsterdam/>) — конференция сообщества в Амстердаме.

## Работа с файлами

В данном разделе будут рассмотрены работа с файлами для этой книги и порядок установки некоторых полезных инструментов React.

### Файловый репозиторий

GitHub-репозиторий, связанный с настоящей книгой (<https://github.com/moonhighway/learning-react>), предоставляет все файлы с программным кодом, систематизированные по главам. Репозиторий представлен сочетанием файлов с кодом и JSBin-образцов. JSBin — это интерактивный редактор кода, похожий на CodePen и JSFiddle.

Одно из основных преимуществ использования JSBin заключается в том, что после щелчка на ссылке можно сразу же экспериментировать с файлом. При создании JSBin-файла или в момент его исправления редактор сгенерирует уникальный URL для вашего примера кода, как показано на рис. 1.1.



Рис. 1.1. URL, созданный JSBin

Буквы после `jsbin.com` представляют собой уникальный ключ URL. После следующего слеша указан номер версии. В последней части URL будет указано одно из двух слов: `edit` для режима редактирования или `quiet` для режима предварительного просмотра.

## Инструменты React-разработчика

Существует несколько инструментов, которые могут быть установлены в качестве расширений или дополнительных модулей браузера и представлять для вас определенный интерес:

- ❑ *react-detector* (<https://chrome.google.com/webstore/detail/react-detector/jaaklebbonndhkanegppscanebkdljh?hl=en-US>) — расширение браузера Chrome, позволяющее узнать, какие сайты используют React, а какие нет;
- ❑ *show-me-the-react* — еще одно средство, доступное для Firefox и Chrome, обнаруживающее React в ходе исследования ресурсов Интернета;
- ❑ *React Developer Tools* (рис. 1.2) (<https://github.com/facebook/react-devtools>) — дополнительный модуль, который может расширить функциональность инструментов разработчика браузера. Он создает новую вкладку в инструментах разработчика, где можно будет просматривать элементы React.

Если вы предпочитаете браузер Chrome, то этот модуль можно установить как расширение (<https://chrome.google.com/webstore/detail/react-developer-tools/fmkadmapgofadopljbjfkapdkoienihi>); его можно также установить в качестве дополнения для Firefox (<https://addons.mozilla.org/en-US/firefox/addon/react-devtools/>).

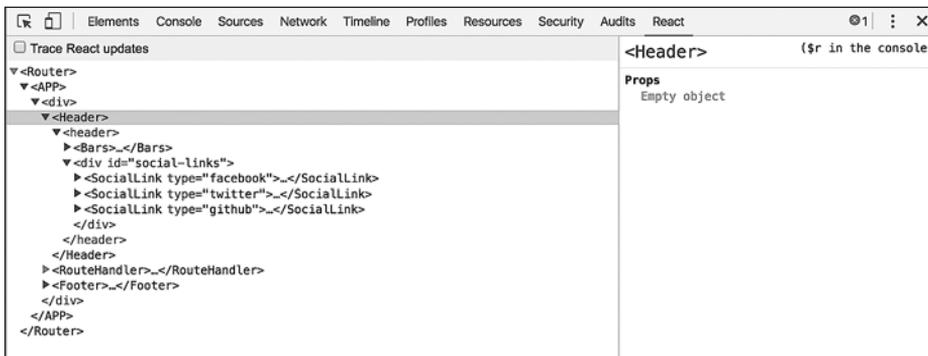


Рис. 1.2. Просмотр React Developer Tools

Всякий раз, когда вы увидите, что *react-detector* или *show-me-the-react* активны, можете открыть инструменты разработчика и получить представление о том, как React используется на сайте.

## Установка Node.js

Node.js — JavaScript-язык без браузера. Это среда исполнения, применяемая для создания JavaScript-приложений на стороне как клиента, так и сервера. Node является средством с открытым кодом и может быть установлен на Windows, macOS, Linux и другие платформы. Мы прибегнем к Node в главе 12 при создании сервера Express.

Чтобы пользоваться React, Node не нужен. Но при работе с React для установки зависимостей необходимо будет задействовать диспетчер пакетов Node под названием `npm`. Он устанавливается автоматически вместе с Node.

Если вы не уверены в том, что на вашей машине установлен Node.js, то можно открыть терминал или окно командной строки и набрать команду:

```
$ node -v
Output: v7.3.0
```

В идеале у вас должен быть Node версии 4 или выше. Появившееся после набора команды сообщение об ошибке `Command not found` (Команда не найдена) означает, что среда не установлена. Установить ее можно непосредственно с сайта Node.js (<https://nodejs.org/en/>). Просто пройдите по автоматически предлагаемым этапам установки и, когда снова наберете команду `node -v`, увидите номер версии.

### Управление зависимостями с помощью Yarn

Дополнительной альтернативой `npm` является Yarn. Этот инструмент был выпущен в 2016 году компанией Facebook в сотрудничестве с Exponent, Google и Tilde. Он помогает Facebook и другим компаниям надежнее управлять своими зависимостями; если использовать Yarn для установки пакетов, то можно заметить, что он работает намного быстрее. Производительность работы `npm` и Yarn можно сравнить на сайте Yarn (<https://yarnpkg.com/en/compare>).

Если вы знакомы с рабочим циклом `npm`, то воспользоваться ускорением, предлагаемым Yarn, будет очень просто. Сначала глобально установите Yarn с помощью `npm`.

```
npm install -g yarn
```

И после этого все будет готово к установке пакетов. При установке зависимостей из `package.json` вместо `npm install` можно запустить команду `yarn`.

При установке отдельных пакетов вместо запуска команды `npm install -save [имя_пакета]` запустите:

```
yarn add [имя_пакета]
```

Для удаления зависимости служит уже известная команда:

```
yarn remove [имя_пакета]
```

Yarn используется на практике компанией Facebook и включается в такие проекты, как React, ReactNative и create-react-app. Если попадется проект с файлом `yarn.lock`, значит, в нем задействован Yarn. По аналогии с командой `npm install`, все зависимости проекта можно установить, набрав команду `yarn install` или просто `yarn`.

Мы настроили среду React-разработки. Теперь можно начинать преодолевать учебные препятствия. В главе 2 мы рассмотрим ESMA и разберемся с самым последним синтаксисом JavaScript, наиболее часто встречающимся в коде React.