

جاده‌ی یادگیری

React

Robin Wieruch

ویرایش نخست

ترجمه: سید منصور عمرانی

انتشارات پندار پارس

سرشناسه	:	ویروخ، روبین Wieruch, Robin
عنوان و نام پدیدآور	:	جاده یادگیری / [روبین ویروخ] ؛ [مترجم] سیدمنصور عمرانی.
مشخصات نشر	:	تهران : پندار پارس ، ۱۳۹۸.
مشخصات ظاهری	:	۱۸۸ ص. : مصور.
شابک	:	978-600-8201-71-7
وضعیت فهرست نویسی	:	فیبا
یادداشت	:	عنوان اصلی: [The road to learn React, 2018].
موضوع	:	برنامه‌های کاربردی وب -- طراحی و توسعه
موضوع	:	-- Development Web applications
موضوع	:	نرم‌افزار کاربردی -- طراحی و توسعه
موضوع	:	Application software -- Development
موضوع	:	رابطه‌های نرم‌افزاری بر پایه وب
موضوع	:	Web-based user interfaces
موضوع	:	جاوا اسکریپت (زبان برنامه‌نویسی کامپیوتر)
موضوع	:	JavaScript (Computer program language)
شناسه افزوده	:	عمرانی، سیدمنصور، ۱۳۵۶ -- مترجم
رده بندی کنگره	:	۷۴/۷۶QA
رده بندی دیویی	:	۰۰۶/۷۸
شماره کتابشناسی ملی	:	۵۸۱۲۵۳۲



انتشارات پندارپارس

دفتر فروش: انقلاب، ابتدای کارگر جنوبی، کوی رشتچی، شماره ۱۴، واحد ۱۶ www.pendarepars.com

تلفن: ۶۶۵۷۲۳۳۵ - تلفکس: ۶۶۹۲۶۵۷۸ همراه: ۰۹۲۱۴۳۷۱۹۶۴ info@pendarepars.com



نام کتاب	:	جاده یادگیری React
ناشر	:	انتشارات پندار پارس
تالیف	:	روبین وایروچ
ترجمه	:	سید منصور عمرانی
چاپ نخست	:	مهر ماه ۹۸
شمارگان	:	۵۰۰ نسخه
طرح جلد	:	رامین شکرالهی
چاپ، صحافی	:	فرشیوه

قیمت : ۴۹۰۰۰ تومان شابک : ۹۷۸-۶۰۰-۸۲۰۱-۷۱-۷



* هرگونه کپی برداری، تکثیر و چاپ کاغذی یا الکترونیکی از این کتاب بدون اجازه ناشر تخلف بوده و پیگرد قانونی دارد *

فهرست

۷	کتاب را توثیق کنید!
۸	پیشگفتار
۹	درباره‌ی مولف
۱۰	نظر دیگران درباره‌ی این کتاب
۱۲	آموزش به بچه‌ها
۱۳	پرسش‌های متداول
۱۵	سابقه‌ی تغییرات کتاب
۱۷	نحوه‌ی خواندن کتاب چگونه است؟
۱۸	چالش
۲۱	فصل ۱. مقدمات React
۲۱	سلام، من React هستم!
۲۲	تفاوت React با JQuery چیست؟
۲۳	تمرین
۲۳	نیازمندی‌ها
۲۳	ادیتور و ترمینال
۲۴	Node و NPM
۲۵	مروری بر npm
۲۵	نصب پکیج‌های node
۲۶	انواع نصب پکیج
۲۷	فایل package.json
۲۸	تفاوت فایل package.json با package-lock.json چیست؟
۲۹	ایجاد سریع فایل package.json
۲۹	بازیابی لیست وابستگی‌های برنامه
۲۹	سوئیچ --save-dev
۳۰	Yarn
۳۰	تمرین
۳۰	نصب
۳۱	تمرین
۳۲	بیکربندی از نقطه‌ی صفر
۳۴	ساختار یک برنامه‌ی create-react-app
۳۴	فولدرها
۳۶	فایل‌ها
۳۸	فولدر src
۳۹	دستورها
۳۹	تمرین
۴۰	معرفی JSX
۴۲	تمرین
۴۳	ES6 و دستور const و let

۴۳const
۴۳let
۴۳let یا const
۴۴ تفاوت var با let
۴۶تمرین
۴۷ReactDOM
۴۸تمرین
۴۸ Hot Module Replacement
۴۹تمرین
۴۹کدنویسی پیچیده‌ی جاوااسکریپت در JSX
۵۲تمرین
۵۳ توابع پیکان یا Arrow function در ES6
۵۴تمرین
۵۴تعریف کلاس در ES6
۵۵متد super()
۵۶تمرین
۵۹فصل ۲. مبانی React
۵۹وضعیت محلی کمپوننت
۶۰شیء props
۶۱تمرین
۶۱راه‌انداز شیء یا object initializer در ES6
۶۲تمرین
۶۲جریان داده‌ی یک طرفه
۶۵تمرین
۶۵Binding
۶۸تمرین
۶۹اداره‌گر رویداد
۷۲تمرین
۷۲تعامل با فرم‌ها و رویدادها
۷۲تعریف اداره‌گر رویداد
۷۴نمایش آیتم‌ها بر اساس عبارت جستجوی وارد شده
۷۴نوشتن شرط تابع filter()
۷۶پیش‌گیری از باگ ناشی از undefined بودن خصوصیت‌های وضعیت محلی
۷۸جستجو با استفاده از indexOf()
۷۸ساده‌سازی تعریف توابع higher-order
۷۹تمرین
۷۹Destructure کردن یا ساختارزدایی در ES6
۸۱تمرین
۸۱کمپوننت‌های کنترل شده

۸۳	تمرین
۸۳	شکستن یک کمپوننت به چندین کمپوننت
۸۵	تمرین
۸۶	کمپوننت‌های مرکب یا composable
۸۷	تمرین
۸۷	کمپوننت‌های قابل استفاده‌ی مجدد یا reusable
۸۹	تمرین
۸۹	دیگر روش‌های تعریف کمپوننت‌های React
۹۲	تمرین
۹۲	استایل دادن به کمپوننت‌ها
۹۷	تمرین
۹۹	فصل ۳. کار با داده‌های واقعی با استفاده از API
۹۹	متمدهای چرخه‌ی حیات کمپوننت‌های React
۱۰۲	تمرین
۱۰۲	بازایی اطلاعات از طریق API
۱۰۵	چک کردن truthfulness یا درستی شرط
۱۰۶	تمرین
۱۰۶	عملگر بسط یا spread در ES6
۱۰۸	تمرین
۱۰۹	رندر کردن شرطی
۱۰۹	مشکل خروج زود هنگام
۱۱۱	تمرین
۱۱۱	جستجو سمت کلاینت یا سمت سرور
۱۱۴	تمرین
۱۱۵	صفحه‌بندی نتیجه‌ی جستجو
۱۱۶	نحوه‌ی صفحه‌بندی
۱۱۷	تعداد خبر در صفحه
۱۱۸	تمرین
۱۱۸	کش کردن اطلاعات در سمت کلاینت
۱۲۳	دیکشنری در جاوااسکریپت
۱۲۴	اداره کردن خطا
۱۲۶	تمرین
۱۲۶	استفاده از Axios بجای fetch()
۱۲۸	تمرین
۱۳۱	فصل ۴. سازمان‌دهی کدها و تست
۱۳۱	ماجول‌ها در ES6، دستور export و import
۱۳۳	تمرین
۱۳۴	سازمان‌دهی سورس پروژه با استفاده از ماجول‌های ES6

۱۳۴	روش‌های مختلف سازمان‌دهی اجزای پروژه.....
۱۳۷	تمرین.....
۱۳۸	اجرای تست Snapshot با استفاده از Jest.....
۱۳۸	چه نوع و چه تعداد تست باید بنویسیم؟.....
۱۳۹	Jest.....
۱۴۳	تمرین.....
۱۴۴	انجام unit test با استفاده از Enzyme.....
۱۴۵	تمرین.....
۱۴۶	تعریف واسط کمپوننت با استفاده از PropTypes.....
۱۴۹	تمرین.....
۱۴۹	دیباک کردن برنامه‌های React با استفاده از React Developer Tools.....
۱۵۰	تمرین.....
۱۵۳	فصل ۵. کمپوننت‌های پیشرفته.....
۱۵۳	ارجاع مستقیم به node های یک کمپوننت React در DOM.....
۱۵۵	تمرین.....
۱۵۶	نمایش پیام Loading ... Loading.....
۱۵۸	تمرین.....
۱۵۸	کمپوننت‌های رتبه بالا یا higher-order.....
۱۶۲	تمرین.....
۱۶۲	مرتب‌سازی پیشرفته.....
۱۷۱	تمرین.....
۱۷۳	فصل ۶. مدیریت وضعیت در React.....
۱۷۳	جابجا کردن وضعیت.....
۱۷۸	تمرین.....
۱۷۸	مروری دوباره بر setState().....
۱۸۱	تمرین.....
۱۸۱	رام کردن وضعیت.....
۱۸۲	تمرین.....
۱۸۳	آخرین گام‌ها پیش از انتشار برنامه.....
۱۸۳	Eject.....
۱۸۳	تمرین.....
۱۸۴	Deploy کردن برنامه.....
۱۸۵	خاتمه.....

کتاب را توئیت کنید!

لطفا با انتشار و تبلیغ این کتاب در تویتر، نویسنده‌ی کتاب، رابین ویراک را یاری کنید.

متن پیشنهادی توئیت چیزی شبیه زیر است:

I just bought The Road to Learn React by @rwieruch #ReactJs <https://roadtoreact.com>

برای توئیت خود از هشت‌تگ #ReactJs استفاده کنید.

برای مشاهده‌ی نظرات دیگران درباره‌ی این کتاب در تویتر لینک زیر را کلیک کنید.

<https://twitter.com/search?q=%23ReactJs>

پیشگفتار

کتاب «جاده‌ی یادگیری React» مبانی و اصول React را به شما یاد می‌دهد. در این کتاب نحوه‌ی نوشتن یک برنامه‌ی واقعی که به طور کامل با React و بدون استفاده از هیچ ابزار پیچیده‌ی دیگری نوشته شده به شما نشان داده می‌شود. همه‌ی مراحل برنامه از ایجاد پروژه تا انتشار و deploy کردن آن روی یک سرور واقعی توضیح داده می‌شود. هر فصل کتاب تعدادی تمرین و مقاله‌ی جانبی هم دارد. محتوای کتاب جامع بوده و هر آنچه را برای ایجاد یک برنامه‌ی React نیاز دارید به شما یاد می‌دهد. از این رو پس از خواندن کتاب می‌توانید به طور عملی برنامه‌ی React درست کنید. این کتاب به طور پیوسته توسط من و جامعه‌ی کاربران آن به‌روز نگه داشته می‌شود.

در این کتاب به جای آن که شما را یک راست وارد اکوسیستم شلوغ و گسترده‌ی React کنم نخست زیر بنای React را توضیح می‌دهم. از نظر محتوا کمتر به ابزارهای مختلف و مدیریت وضعیت خارجی می‌پردازم اما تا دل‌تان بخواهد درباره‌ی React صحبت می‌کنم. همچنین مفاهیم کلی React، الگوها و بهترین تکنیک‌های کدنویسی یک برنامه‌ی واقعی React را بیان می‌کنم.

به طور کلی در این کتاب یاد می‌گیرید چطور از نقطه‌ی صفر یک برنامه‌ی React را با قابلیت‌هایی مانند صفحه‌بندی، کش کردن محتوا در سمت کلاینت، روش‌های مختلف تعامل کاربر مانند جستجو و مرتب‌سازی بنویسید. همچنین در طول کتاب از جاوااسکریپت ES5 به ES6 مهاجرت خواهیم کرد. این کتاب تمام شور و اشتیاق من نسبت به React و جاوااسکریپت است. امیدوارم به شما هم کمک کند بتوانید React را با آن شروع کنید.

درباره‌ی مولف

من یک مهندس کامپیوتر و برنامه‌نویس آلمانی هستم که تخصص‌ام آموزش برنامه‌نویسی جاوااسکریپت است. پس از فارغ‌التحصیلی از دانشگاه با درجه‌ی کارشناسی ارشد در رشته‌ی Computer Science دست از یادگیری برداشتم. تجربیاتم را از دنیای استارت‌آپ که در آن با جدیت در زندگی حرفه‌ای و همچنین اوقات فراغت خود از جاوااسکریپت استفاده می‌کردم به دست آوردم و آن تجربیات در نهایت به این منجر شد که برنامه‌نویسی جاوااسکریپت را به دیگران هم آموزش بدهم.

برای چندین سال در شرکتی به نام Small Improvements از نزدیک با تیم خوبی از مهندسين روی یک برنامه‌ی مقیاس بزرگ کار کردم. این شرکت یک محصول SaaS دارد که به مشتری‌ها امکان می‌دهد به شرکت‌های تجاری بازخورد بدهند. بخش front-end این برنامه با جاوااسکریپت و بخش back-end آن با جاوا (و فریم‌ورک Wicket) نوشته شده بود. در ابتدا بخش front-end برنامه با JQuery پیاده‌سازی شده بود. هنگامی که نسل نخست برنامه‌های SPA محبوبیت پیدا کرد، شرکت Small Improvements بخش front-end برنامه را به Angular 1.x مهاجرت داد. پس از ۲ سال کار، تیم توسعه به این نتیجه رسید Angular برای برنامه‌هایی که زیاد از وضعیت یا state استفاده می‌کنند بهترین گزینه نیست. به همین دلیل شرکت تصمیم خود را گرفت و بخش front-end را به React و Redux مهاجرت داد و با این کار برنامه توانست با موفقیت در مقیاس بزرگ کار کند.

یکی از علاقمندی‌های من نوشتن مقاله در زمینه‌ی توسعه و برنامه‌نویسی وب است. آنها را در سایت شخصی‌ام منتشر می‌کنم. در مقاله‌هایم بازخورد بسیار خوبی از سوی کاربران دریافت کردم. این باعث شد نگارش و مهارتم در آموزش بهبود پیدا کرده و توانایی آموزشی‌ام با نوشتن هر مقاله رشد پیدا کند. نخستین مقاله‌های من طولانی و پر از جزئیات بود و می‌توانست خواننده را گیج کند، اما بازخورد کاربران باعث شد رویه‌ام را عوض کنم و مقاله‌هایم را بهبود بدهم. از این رو مقاله‌های بعدی من کوتاه‌تر و مختصرتر شد. اکنون در هر مقاله تنها روی یک موضوع تمرکز می‌کنم.

در حال حاضر یک برنامه‌نویس جاوااسکریپت شغل آزاد (self-employed) هستم. برای خودم کار می‌کنم و یکی از کارهایم آموزش جاوااسکریپت به دیگران است. مشاهده‌ی رشد و ترقی دیگران با شفاف کردن مفاهیم برایشان و دریافت بازخورد از آنها کار لذتبخش و جذابی است. برای کسب اطلاع بیشتر درباره‌ی من و پشتیبانی یا همکاری با من به سایت شخصی‌ام به آدرس زیر مراجعه کنید.

<https://www.robinwieruch.de>

نظر دیگران درباره‌ی این کتاب

افراد زیادی به کتابم نظر یا امتیاز داده‌اند که تاییدی بر ارزش و کیفیت کتاب است. به خودم افتخار می‌کنم، زیرا حقیقتاً انتظار چنین بازخوردی را نداشتم. اگر شما هم از خواندن کتاب لذت بردید، خوشحال می‌شوم از نظر یا امتیاز شما هم آگاه شوم. این کار به من کمک می‌کند بیشتر بتوانم کتابم را نشر داده و سخن آن را به گوش دیگران برسانم. در اینجا تعدادی از نظرات مثبت دیگران را در خصوص کتاب ذکر می‌کنم.

محمد کاشف: The Road to Learn React کتاب منحصر به فردی است و آن را به همه‌ی برنامه‌نویسان مبتدی یا حرفه‌ای که دوست دارند React را از پایه تا پیشرفته یاد بگیرند توصیه می‌کنم. این کتاب پر از نکته‌ها و تکنیک‌های آموزنده است که به سختی می‌توانید آن را جای دیگری پیدا کنید. همچنین این کتاب پر از مثال است و مشکلات رایج برنامه‌های React را بیان می‌کند. من ۱۷ سال در برنامه‌نویسی وب و برنامه‌های رومیزی تجربه دارم. پیش از خواندن این کتاب در یادگیری React مشکل داشتم اما این کتاب معجزه می‌کند!

آندره وارگاس: The Road to Learn React نوشته‌ی رابین ویراک واقعا کتاب فوق‌العاده‌ای است. بیشتر چیزی که درباره‌ی React و حتی ES6 یاد گرفتم از طریق این کتاب بوده!

نیکلاس هانت-واکر^۱: این کتاب از نظر نگارش و محتوا یکی از بهترین کتاب‌های برنامه‌نویسی‌ای است که تا به حال خوانده‌ام. یک آشنایی بسیار محکم و اصولی با React و ES6.

آستین گرین: خیلی ممنون. واقعا از کتاب لذت بردم. کتاب شما آمیزه‌ای عالی از React، ES6 و مفاهیم سطح بالای برنامه‌نویسی است.

نیکول فرگوسن: من آخر هفته دارم یک دوره‌ی آموزش React طبق کتاب The Road to Learn React ارائه می‌دهم. خیلی خوش گذشت.

کارن: کتاب Road to Learn React شما را همین الان تمام کردم. بهترین کتاب یادگیری مقدماتی در دنیای React و Js است. به طور کامل و عالی ES را توضیح می‌دهد. احسنت! 😊

اریک پریو: کتاب The Road to Learn React نوشته‌ی رابین ویراک را حتما باید بخوانید. کتابی شسته‌رفته، تروتمیز و مختصر برای یادگیری React و Javascript.

یک برنامه‌نویس تازه‌کار: الان کتاب را تمام کردم. از این که چنین کتابی نوشته‌ی خیلی ممنونم. محتوای کتاب کاملا سرراست بود و به راحتی می‌شد آن را دنبال کرد. اکنون آنقدر اعتماد به نفس دارم که خودم بتوانم از نقطه‌ی صفر یک برنامه‌ی React بنویسم. این کتاب از کتاب رسمی آموزش React.js که قبلا خوانده بودم (و البته به دلیل نپرداختن به جزئیات نتوانسته بودم آن را تمام کنم) بسیار بهتر است. تمرین‌های انتهایی بخش‌ها بسیار ارزنده بود.

^۱ مدرس Python در یک موسسه‌ی آموزشی در سیاتل

یک کاربر: این کتاب بهترین کتاب برای شروع یادگیری ReactJS است. برنامه‌ای که در طول کتاب یاد داده می‌شود همگام با محتوا و مفاهیم مورد آموزش تکمیل می‌شود، در نتیجه به خوبی می‌توانید مفاهیم را یاد بگیرید. به نظر من روش «یادگیری از طریق کدنویسی» یکی از بهترین روش‌های یادگیری برنامه‌نویسی است. این کتاب نیز دقیقاً همین کار را می‌کند.

توماس لاکنی: این کتاب پیش‌درآمدی اساسی برای React است که هدفش این نیست به تمام معنی جامع باشد. قصد شخصی من این بود بینم React اصلاً چیست. این کتاب هم دقیقاً همین را به من یاد داد. من همه‌ی پایین‌نویس‌های پایین صفحات را در خصوص قابلیت‌های جدید ES6 پیگیری نکردم (البته نمی‌گویم دلم برایشان تنگ می‌شده). اما مطمئنم کسانی که هنوز کتاب را تمام نکرده‌اند و آن پایین‌نویس‌ها را هم با پشتکار می‌خوانند بسیار بیشتر از آنچه که کتاب آموزش می‌دهد یاد می‌گیرند.

آموزش به بچه‌ها

این کتاب می‌تواند برای یادگیری React به همه کمک کند. اما همه نمی‌توانند از آن استفاده کنند، زیرا زبان مادری‌شان انگلیسی نیست. با این کتاب از پروژه‌های دیگری که با آموزش برنامه‌نویسی به بچه‌ها انگلیسی یاد می‌دهند حمایت می‌کنم.

1 April to 18 April, 2017, [Giving Back, By Learning React](#).

پرسش‌های متداول

چطور به روزرسانی‌ها و تغییرات جدید کتاب را دریافت کنم؟

برای کسب اطلاع از به‌روز رسانی محتوای کتاب دو راه دارید: مشترک خبرنامه‌ی ایمیلی من بشوید^۱ یا مرا در توئیتر دنبال کنید.^۲

آیا React استفاده شده در کتاب آخرین نسخه‌ی React است؟

هر موقع نسخه‌ی جدید React منتشر می‌شود کتاب را به‌روز می‌کنم. معمولاً کتاب‌های دیگر مدتی پس از انتشارشان منسوخ می‌شوند. از آنجایی که این کتاب را خودم منتشر می‌کنم، هر زمان که بخواهم می‌توانم آن را به‌روز کنم.

آیا Redux هم در کتاب آموزش داده می‌شود؟

خیر. Redux را در کتاب دیگری آموزش داده‌ام. این کتاب برای شما دیدگاهی بنیادی فراهم می‌کند که پیش از پریدن به اعماق موضوع‌های پیشرفته‌تری مانند Redux به آن نیاز دارید. برنامه‌ای که در طول کتاب توضیح داده‌ام نشان می‌دهد برای ایجاد یک برنامه‌ی React به Redux نیاز ندارید. به همین دلیل پس از خواندن کتاب می‌توانید بدون Redux یک برنامه‌ی اصولی و محکم React بنویسید. اما پس از محکم شدن زیر بنای دانش خود می‌توانید برای یادگیری Redux سراغ کتاب بعدی من بروید.

آیا در کتاب از Javascript ES6 استفاده می‌شود؟

بله. اما اگر با ES6 آشنا نیستید و بجایش ES5 را بلدید نگران نباشید. مشکلی نخواهید داشت. همه‌ی قابلیت‌های ES6 را که در طول کتاب حین آموزش React توضیح می‌دهم با تبدیل کد ES5 به ES6 بیان می‌کنم. همچنین در طول کتاب همه‌ی قابلیت‌های ES6 را توضیح می‌دهم. به همین دلیل این کتاب نه فقط آموزش React بلکه آموزش قابلیت‌های مفید ES6 برای استفاده در React هم هست.

آیا در آینده ممکن است فصل(های) جدیدی به کتاب افزوده شود؟

اگر نگاهی به بخش «سابقه‌ی تغییرات کتاب» ببینید می‌توانید تغییراتی را که از ابتدا تاکنون به کتاب اعمال شده ببینید. البته ممکن است پیش از انتشار نسخه‌ی جدیدی از کتاب بهبودهایی را هم بدون اعلام عمومی به کتاب بیافزایم. اما در حالت کلی ادامه‌ی کار من روی این کتاب به پشتیبانی خوانندگان بستگی دارد. اگر کتاب نزد مخاطبان مقبول بیافتد، در آینده فصل‌های جدیدی به آن افزوده و محتوای قدیمی‌تر را به‌روز می‌کنم. همچنین تلاش می‌کنم محتوای کتاب را با بهترین نکات، تکنیک‌ها، الگوها و مفاهیم React به‌روز نگه دارم.

اگر حین خواندن کتاب جایی به کمک نیاز داشتیم چه باید بکنیم؟

برای مخاطبین کتاب یک گروه Slack درست کرده‌ام. می‌توانید برای دریافت کمک یا کمک به خوانندگان دیگر به کانال مزبور بپیوندید. به هر حال کمک کردن به دیگران باعث می‌شود آموزه‌های خودتان را هم بهتر یاد بگیرید.

¹ <https://www.getrevue.co/profile/rwieruch>

² <https://twitter.com/rwieruch>

اگر در استفاده از کتاب یا گدهای آن با خطا مواجه شدم جایی برای رفع اشکال وجود دارد؟

می‌توانید مشکل خود را در کانال Slack ای که در بالا بیان کردم مطرح کنید. همچنین می‌توانید مشکل خود را در قسمت Issues مخزن این کتاب در Github هم مطرح کنید. شاید مشکل شما توسط یک نفر دیگر مطرح شده باشد. در این صورت می‌توانید راه رفع مشکل خود را همانجا ببینید. اگر مشکل شما از پیش مطرح نشده بود حتماً آن را ثبت کرده و با اسکرین‌شات و جزئیات (مانند شماره صفحه، نسخه‌ی node خودتان) توضیح بدهید. سعی می‌کنم در ویرایش‌های بعدی کتاب همه‌ی خطاها را بر طرف کنم.

آیا می‌توانم در بهبود کتاب مشارکت داشته باشم؟

حتماً. می‌توانید با ثبت نظرات خود در بخش Contributions مخزن کتاب در Github به طور مستقیم در بهبود کتاب مشارکت داشته باشید. من ادعا نمی‌کنم فرد بسیار حرفه‌ای و متخصصی هستم. زبان اصلی من نیز انگلیسی نیست. پس می‌توان محتوا و نحوه‌ی نگارش کتاب را بهبود داد و من در این زمینه از شنیدن نظرات و کمک شما واقعا خوشحال می‌شوم.

آیا می‌توانم کتاب را پشتیبانی کنم؟

قطعا. من بیشتر وقتم را صرف آموزش این سورس و منابع آموزشی می‌کنم. می‌توانید نگاهی به بخش about me در سایت‌م ببینید. قطعاً خوشحال می‌شوم در سایت‌م نام شما را در صفحه‌ی Patron on Patreon به عنوان یکی از پشتیبان‌ها داشته باشم.

در خاتمه آیا توصیه‌ی کوتاه دیگری در خصوص کتاب هم هست؟

بلی. از شما می‌خواهم کمی تامل کنید ببینید فردی را سراغ دارید که بتوان React را به او یاد داد. ممکن است خودش هم دنبال یاد گرفتن React بوده یا شاید در حال یاد گرفتن React باشد. شاید هم اصلاً نداند React به چه درد او می‌خورد! سراغ چنین فردی بروید و کتاب را به او بدهید. این کار خیلی برای من ارزش دارد. من از ابتدا کتاب را با این نیت نوشتم که خوانندگان بتوانند آن را به دیگران هم بدهند.

سابقه‌ی تغییرات کتاب

۱۰ ژانویه‌ی ۲۰۱۷

- [v2 Pull Request](#)
- ساده‌تر کردن بخش آغازین کتاب
- ۳۷٪ محتوای جدید
- ۳۰٪ بهبود محتوا
- ۱۳ فصل بهبود یافته و جدید
- افزایش تعداد صفحات کتاب به ۱۴۰ صفحه
- ایجاد یک دوره‌ی آموزشی برای کتاب در سایت [educative.io](https://www.educative.io)^۲

۸ مارس ۲۰۱۷

- [v3 Pull Request](#)^۳
- ۲۰٪ محتوای جدید
- ۲۵٪ بهبود محتوا
- ۹ فصل جدید
- افزایش تعداد صفحات کتاب به ۱۷۰ صفحه

۱۵ آوریل ۲۰۱۷

- به روز کردن React استفاده شده در کتاب به React 15.5

۵ جولای ۲۰۱۷

- به روزرسانی نسخه‌ی node استفاده شده در کتاب به node 8.1.3
- به روزرسانی نسخه‌ی npm استفاده شده در کتاب به npm 5.0.4
- به روزرسانی نسخه‌ی create-react-app استفاده شده در کتاب به ورژن 1.3.3

۱۷ اکتبر ۲۰۱۷

- به روزرسانی نسخه‌ی node استفاده شده در کتاب به node 8.3.0
- به روزرسانی نسخه‌ی npm استفاده شده در کتاب به npm 5.5.1
- به روزرسانی نسخه‌ی create-react-app استفاده شده در کتاب به ورژن 1.4.1
- به روز کردن React استفاده شده در کتاب به React 16

- [v4 Pull Request](#)

¹ <https://github.com/the-road-to-learn-react/the-road-to-learn-react/pull/18>

² <https://www.educative.io/collection/5740745361195008/5676830073815040>

³ <https://github.com/the-road-to-learn-react/the-road-to-learn-react/pull/34>

- ۱۵٪ محتوای جدید
- ۱۵٪ بهبود محتوا
- ۳ فصل جدید (Bindings، اداره‌گرهای رویداد، اداره کردن خطا)
- افزایش تعداد صفحات کتاب به ۱۹۰ صفحه
- ۹ پروژه همراه با سورس کد^۲

۱۷ فوریه‌ی ۲۰۱۸

- به روزرسانی نسخه‌ی node استفاده شده در کتاب به node 8.9.4
- به روزرسانی نسخه‌ی npm استفاده شده در کتاب به npm 5.6.0
- به روزرسانی نسخه‌ی create-react-app استفاده شده در کتاب به ورژن 1.5.1
- [v5 Pull Request](#)^۳
- توسعه‌ی سیر یادگیری
- محتوای خواندنی جدید
- یک فصل جدید (Axios بجای Fetch)

۳۱ آگوست ۲۰۱۸

- بازخوانی و ویرایش کل محتوا به طور تخصصی توسط امانوئل استالینگ
- افزودن ۱۶ پروژه با سورس کد^۴
- [v6 Pull Request](#)^۵

۳ اکتبر ۲۰۱۸

- ارتقا به node 10.11.0
- ارتقا به npm 6.4.1
- ارتقا به create-react-app 2.0.2

¹ <https://github.com/rwieruch/the-road-to-learn-react/pull/72>

² <https://roadtoreact.com>

³ <https://github.com/the-road-to-learn-react/the-road-to-learn-react/pull/105>

⁴ <https://roadtoreact.com>

⁵ <https://github.com/the-road-to-learn-react/the-road-to-learn-react/pull/172>

نحوه‌ی خواندن کتاب چگونه است؟

برنامه‌ی من در این کتاب آموزش عملی همراه با نوشتن یک برنامه‌ی React بوده است. به همین دلیل کتاب فعلی یک راهنمای کلی است نه مرجع React. در این کتاب یک برنامه‌ی خبری ساده درست می‌کنیم که از یک api واقعی استفاده می‌کند. هنگام نوشتن کتاب آموزش React موضوع‌های جالبی وجود دارد که می‌توان در کتاب آورد. من در این کتاب از میان موضوع‌های مختلف، مدیریت وضعیت، کش کردن دیتا و تعامل کاربر با صفحه (مرتب‌سازی و جستجو) را پوشش داده‌ام. همچنین بهترین تکنیک‌ها و الگوهای برنامه‌نویسی React را هم بیان نموده‌ام.

از سوی دیگر این کتاب امکان مهاجرت شما را از جاوااسکریپت ES5 به ES6 هم فراهم می‌کند. React بسیاری از قابلیت‌های ES6 را استفاده می‌کند. من هم به شما نشان می‌دهم چگونه می‌توانید از این قابلیت‌ها استفاده کنید.

از نظر محتوا هر فصل بر پایه‌ی فصل‌های پیش نوشته شده، از این رو باید محتوای فصل‌های پیش را بلد باشید. با توجه به این مساله و این که کتاب بر پایه‌ی تکمیل یک برنامه‌ی عملی نوشته شده شاید نتوانید از فصل‌ها پرش کرده و مثلاً یک راست سراغ یک فصل به خصوص بروید. بلکه باید کتاب را از ابتدا شروع کرده و فصل به فصل تا انتها بخوانید. در هر فصل یک چیز جدید آموزش می‌دهم. اما مراقب باشید در خواندن فصل‌ها عجله نکنید و سریع از روی آنها رد نشوید. باید مطالب را به طور عمقی یاد بگیرید. بعداً می‌توانید کدها را مطابق میل خودتان تغییر داده و بیشتر درباره‌ی آنچه آموزش داده شده مطالعه کنید. در انتهای هر فصل تعدادی تمرین و لینک تعدادی مقاله قرار داده‌ام. اگر می‌خواهید React را خوب یاد بگیرید توصیه می‌کنم حتماً محتوای اضافی انتهای فصل‌ها را هم بخوانید و تمرین‌ها را انجام بدهید. پس از خواندن هر فصل پیش از رفتن سراغ فصل بعد کمی به خودتان استراحت بدهید.

در پایان خواندن کتاب یک برنامه‌ی واقعی React خواهید داشت. خیلی مشتاقم برنامه‌ای را که می‌نویسید ببینم. حتماً پس از تمام کردن کتاب به من پیغام بدهید. در فصل پایانی تعدادی موضوع برای یادگیری بیشتر ذکر کرده‌ام که برای ادامه‌ی سفر سیاحتی‌تان در دنیای React هر کدام را دوست داشتید می‌توانید دنبال کنید. اگر به سایت شخصی من هم سر بزنید کلی مطلب مرتبط با React پیدا خواهید کرد.

از آنجایی که این کتاب را برداشته‌اید، فرضم این است که یادگیری React را شروع کرده‌اید. عالی است! امیدوارم در پایان کتاب بازخورد شما را هم دریافت کنم تا بتوانم کتاب را بهبود داده و کاری کنم همه React را یاد بگیرند. می‌توانید نظراتان را در مخزن Github کتاب ثبت کنید یا در توئیتر مستقیماً به خودم پیغام بدهید.

چالش

من شخصا درباره‌ی چیزهایی که یاد می‌گیرم خیلی مطلب می‌نویسم. این گونه است که اکنون اینجا در خدمت شما هستم. واقعیت این است که وقتی خودتان چیزی را یاد بگیرید می‌توانید آن را خوب هم یاد بدهید. از آنجایی که آموزش برنامه‌نویسی خیلی به من کمک کرده، دوست دارم شما هم آن را تجربه کنید. اما نخست باید انگیزه‌اش را ایجاد کنید: یعنی خودتان یاد بگیرید. صرفنظر از یادگیری React من یک چالش هم برایتان دارم: حین خواندن کتاب چیزهایی را که یاد می‌گیرید به دیگران هم یاد بدهید!

چند کار در این زمینه می‌توانید انجام بدهید:

- درباره‌ی یک موضوع به خصوصی که در کتاب یاد گرفتید بلاگ بنویسید. البته نه این که محتوای کتاب را copy-paste کنید، بلکه موضوع را به شیوه‌ی خودتان آموزش بدهید. مطلب را به زبان و بیان خودتان توضیح بدهید، روی مشکل تمرکز کرده و آن را حل کنید و با درک همه‌ی جزئیات موضوع در آن عمیق شوید. سپس آن را به شکل مقاله به دیگران یاد بدهید. خودتان خواهید دید چطور این روش فضاهای خالی دانسته‌های شما در خصوص موضوع مورد آموزش را پر می‌کند، زیرا برای آموزش دادن باید در خصوص موضوع مورد آموزش عمیق‌تر شوید. همچنین به چشم خودتان خواهید دید که این روش می‌تواند در دراز مدت، مهارت فنی شما را افزایش داده و درهای جدیدی در زمینه‌ی کارتان برای شما باز کند.
- اگر در شبکه‌های اجتماعی فعال هستید، چند تا از مطالبی را که با خواندن کتاب یاد گرفتید انتخاب کرده و share کنید. برای نمونه می‌توانید یک نکته‌ی داغ برنامه‌نویسی را که تازه یاد گرفتید و بنظرتان می‌تواند برای دیگران هم جالب باشد در تویتر توثیت کنید. می‌توانید یک اسکرین‌شات از صفحه‌ای از کتاب که مطلب مورد نظر در آن است هم بگیرید. اگر بتوانید آن را به زبان خودتان هم بیان کنید دیگر عالی است. به این ترتیب می‌توانید وارد حرفه‌ی آموزش شوید بدون آن که مجبور باشید وقت زیادی برای این کار صرف کنید.
- اگر اعتماد به نفس کافی دارید که حین یادگیری صدای خودتان را ضبط کنید می‌توانید سیر یادگیری‌تان را از طریق Facebook Live، Youtube Live یا Twitch به اشتراک بگذارید. این کار باعث می‌شود بهتر بتوانید روی محتوا تمرکز کرده و بهتر با کتاب پیش بروید. حتی اگر افراد زیادی هم جلسات آموزشی زنده‌ی شما را دنبال نکنند می‌توانید ویدئوی آموزشی خود را بعدا هم در Youtube بگذارید. مزیت این کار این است که به طور شفاهی و به شکل بسیار خوبی می‌توانید مشکلات خود را بیان کرده و نحوه‌ی حل کردن آنها را بیان کنید.

خیلی دوست دارم افرادی را که سومین مورد را انجام می‌دهند ببینم. صدا و تصویر خود را حین خواندن کتاب، پیاده‌سازی برنامه‌ای که در کتاب توضیح داده شده و انجام دادن تمرین‌ها ضبط کرده و نسخه‌ی نهایی را در Youtube بگذارید. اگر فاصله‌ی خالی بین ویدئوهای شما زیاد است آنها را cut کنید یا یک افکت تغییر زمان برایش بگذارید. اگر حین تمرین جایی با خطا مواجه شده و گیر افتادید اما بعدا رفعش کردید آن تکه را از فیلم خود حذف نکنید. فیلم مشکل و باگ را هم در Youtube بگذارید، زیرا برای دیگران که ممکن است با همین خطا مواجه شوند خیلی مفید خواهد بود. اتفاقا بنظرم خیلی مهم است که این چیزها هم در ویدئوهای شما وجود داشته باشد.

چند نکته در مورد تهیهی فیلم به ذهنم می‌رسد که بد نیست به شما هم بگویم:

- آموزش خود را به زبان مادری خودتان و اگر با انگلیسی راحت هستید به زبان انگلیسی بگویید.
 - هر چیزی را که به ذهن‌تان می‌رسد به زبان بیاورید، چه موقع کُدنویسی، چه موقع مواجه شدن با خطاها. تهیهی فیلم ویدئویی تنها یک بخش از چالش است، بخش دیگر چالش این کار روایت و نقل شما از ماجرا است. نیازی نیست روایت بی‌عیب و نقصی باشد. فقط کافی است طبیعی باشد. نیازی هم نیست مثل ویدئوهای آموزشی‌ای که در اینترنت وجود دارد کاملاً تر و تمیز و بدون خطا و باگ باشد.
 - اگر با خطایی مواجه شدید نترسید و آن را با روی باز بپذیرید. سعی کنید مشکل را رفع کنید یا در اینترنت دنبال راه‌حل‌اش بگردید. ناامید نشوید و مشکل را شفاها توضیح بدهید و بگویید برای رفع‌اش چه کار دارید می‌کنید. این کار باعث می‌شود دیگران بتوانند سیر فکری شما را دنبال کنند. همان گونه که گفتم تماشای ویدئوهایی که در آن فرد آموزش‌دهنده با هیچ خطایی مواجه نمی‌شود آنقدرها با ارزش نیست. اتفاقاً مشاهدهی همین خطاها و رفع نمودن آنها است که ارزش دارد.
 - چند نکته در خصوص بخش فنی ضبط فیلم بگویم: پیش از ضبط یک ویدئوی طولانی، میکروفون خود را چک کنید. باید ولوم و کیفیت مناسبی داشته باشد. در ویرایشگر سورس کُد خود و پنجره‌ی ترمینال خط فرمان اندازه‌ی فونت را زیاد کنید تا محتوای آنها بهتر قابل مشاهده باشد. اگر برایتان ممکن بود پنجره‌ی مرورگر و ویرایشگر خود را همزمان کنار هم قرار بدهید. اگر چنین کاری برایتان میسر نبود پنجره‌های آنها را fullscreen کنید و هر بار (با استفاده از کلید CMD + Tab در MacOS یا ALT + Tab در ویندوز) بین آنها سوئیچ کنید.
 - ویدئویی را که ضبط کردید پیش از آپلود به Youtube ویرایش کنید. نیازی نیست کیفیت خیلی بالایی داشته باشد. اما بهتر است مختصر بوده و روی اصل محتوا تمرکز داشته باشد (مثلاً قسمت‌های خواندن جملات کتاب را حذف کنید و فقط کلماتی را که بعداً به زبان می‌آورید باقی بگذارید).
- در انتها اگر دوست داشتید مجموعه کلیپ‌هایی را که منتشر کردید به من هم اطلاع بدهید. اگر ویدئوهای شما خوب از آب در بیاید خوشحال می‌شوم آنها را به طور رسمی به عنوان محتوای تکمیلی به کتاب اضافه کنم. تنها کافی است پس از خاتمه‌ی کارتان مرا خبر کنید. در کل امیدوارم این چالش را که واقعا یادگیری‌تان را بهبود می‌دهد و می‌تواند برای دیگران هم مفید باشد بپذیرید. برای همه‌ی شما بهترین‌ها را آرزو می‌کنم.

فصل ۱. مقدمات React

در این فصل به طور مختصر React را معرفی می‌کنم. ممکن است از خودتان بپرسید چرا باید React را یاد بگیرم و React به چه درد من می‌خورد؟ این فصل پاسخ همین پرسش را به شما می‌دهد. سپس با ایجاد یک برنامه‌ی React از نقطه‌ی صفر و بدون هیچ پیکربندی از پیش آماده مستقیماً وارد دنیای React می‌شویم. همچنین در طول فصل مروری بر JSX (زبان نحوی React) و ReactDOM خواهیم داشت تا دیدی کلی نسبت به استفاده‌ی عملی React پیدا کنیم.

سلام، من React هستم!

در طول سال‌های اخیر برنامه‌های تک صفحه‌ای SPA^۱ محبوبیت زیادی پیدا کرده‌اند. برنامه‌نویس‌ها به کمک فریم‌ورک‌هایی مانند Angular، Ember و Backbone توانسته‌اند برنامه‌های پیشرفته‌ای و رای جاوااسکریپت محض و jQuery بنویسند. لیست جامعی برای فریم‌ورک‌های SPA وجود ندارد و در این زمینه می‌توانید فریم‌ورک‌های زیادی در اینترنت پیدا کنید. اما از نظر تاریخ انتشار، سه فریم‌ورکی که نام بردیم جزو نخستین فریم‌ورک‌های SPA محسوب می‌شوند که بین سال‌های ۲۰۱۰ و ۲۰۱۱ منتشر شدند (Angular 2010، Backbone 2010 و Ember 2011).

اولین نسخه‌ی React سال ۲۰۱۳ توسط فیسبوک منتشر شد. البته React یک فریم‌ورک SPA محسوب نمی‌شود، بلکه کتابخانه‌ای برای مدیریت view یا نمای صفحه است. در واقع تمرکز React به حرف V در الگوی MVC (مدل-ویو-کنترلر^۲) است (اگر اصطلاح MVC را قبلاً نشنیده‌اید نگران نشوید. این کلمه را محض این بیان کردم تا افرادی که از پیش با الگوی MVC آشنا هستید حوزه‌ی کار React را بهتر درک کنند). با React می‌توانید در صفحه‌ی وب کمپوننت درست کرده و با آنها کار کنید. با این حال با استفاده از اکوسیستمی که حول React شکل گرفته می‌توانید برنامه‌های بزرگ تک صفحه‌ای SPA هم درست کنید.

اما چرا بجای فریم‌ورک‌های ویژه‌ی SPA باید از React استفاده کنیم؟ علتش این است که فریم‌ورک‌های اختصاصی SPA به دغدغه‌های متعددی می‌پردازند اما React تنها به لایه‌ی view تمرکز دارد. به همین دلیل شسته‌رفته‌تر است، سریع‌تر می‌توانید آن را یاد بگیرید و در نتیجه سریع‌تر می‌توانید با آن برنامه‌ی خود را درست کنید. ایده‌ی کلی React این است که view یا نمای صفحه از ساختاری سلسله‌مراتبی از کمپوننت‌ها تشکیل می‌شود.

¹ Single Page Application

² Model-View-Controller

در React می‌توانید پیش از افزودن جنبه‌های مختلف به برنامه از ابتدا روی لایه‌ی view تمرکز کنید. هر جنبه‌ی دیگری که بعداً می‌خواهید به برنامه اضافه کنید بلاک دیگری در ساختمان برنامه‌ی SPA شما است.^۱ این بلاک‌بندی برای ایجاد یک برنامه‌ی بالغ SPA ضروری است. چنین مدلی دو مزیت دارد:

- نخست این که بلاک‌ها را یکی یکی یاد می‌گیرید و نیازی نیست همه را از ابتدا بلد باشید. تفاوت React با فریم‌ورک‌های SPA که با کل صفحه درگیر هستند همین است. در این کتاب نیز به React به عنوان نخستین بلاک ساختمان برنامه می‌پردازیم، اما سراغ بلاک‌های دیگر هم می‌رویم.
- دومین مزیت بلاک‌بندی ساختار برنامه این است که بلاک‌ها مستقل از هم هستند. می‌توانید هر کدام را که خواستید بردارید و یک چیز دیگر جایش بگذارید. به همین دلیل اکوسیستمی که حول React شکل گرفته پر از نوآوری و ابتکار است. در هر زمینه راه‌حل‌ها و گزینه‌های مختلف با هم رقابت می‌کنند و می‌توانید بسته به نیاز برنامه و میل خودتان هر چیزی را با یک چیز دیگر جایگزین کرده و استفاده کنید.

نخستین نسل فریم‌ورک‌های SPA کارشان را در سطح انترپرایز شروع کردند. به همین دلیل ساختار سفت و سختی داشتند. اما React کتابخانه‌ی ساده و مختصری است که با وجود ساده بودنش توانسته نظر بسیاری از سایت‌های مشهور و رهبران فناوری وب مانند Airbnb، Netflix و Facebook را به سوی خود جلب کند. همه‌ی این شرکت‌ها روی React سرمایه‌گذاری کرده‌اند و از مدل React و اکوسیستمی که دورش شکل گرفته راضی هستند.

بدون تردید امروزه React یکی از بهترین گزینه‌ها برای ایجاد برنامه‌های پیشرفته‌ی وبی است. اگرچه این کتابخانه تنها با لایه‌ی view کار می‌کند اما اکوسیستم‌اش بسیار انعطاف‌پذیر و تعویض‌پذیر است.^۲ API این کتابخانه سبک است، اکوسیستم شگفت‌انگیزی دارد و جامعه‌ی برنامه‌نویسان‌اش هم عالی است. اگر دوست داشتید مقاله‌ی مرا تحت عنوان [why I moved from Angular to React](#) «چرا از Angular به React مهاجرت کردم»^۳ بخوانید.

بسیار توصیه می‌کنم دلیل این که چرا باید بجای فلان فریم‌ورک یک چیز دیگر استفاده کنیم را خوب متوجه بشوید. به هر حال همه مایلند بدانند React در سال‌های بعد کجا می‌خواهد آنها را ببرد.

تفاوت React با jQuery چیست؟

سطح React از jQuery بالاتر است. در جایی که در کتابخانه‌ی سطح پایین jQuery مستقیماً با DOM و گره‌های آن کار می‌کنید در React بجای DOM با کمپوننت‌ها کار می‌کنید.

^۱ مقصود نویسنده از «جنبه» یا «بلاک» را نمی‌شود در یک کلمه توضیح داد. مقصودش این است که وقتی دارید صفحه را درست می‌کنید با مسائل و جنبه‌های مختلفی درگیر هستید و هر جنبه نیز از نظر فنی با چیز مجزایی سر و کار دارد. مثلاً در یک جنبه‌ی برنامه ممکن است درگیر بحث مدل باشید و در یک جنبه‌ی دیگر درگیر فراخوانی REST API باشید. این جنبه‌ها حتی می‌تواند مسائلی مانند حفظ state، تصدیق هویت، تست برنامه، شیوه‌نامه‌های CSS و همانند آن هم باشد. React به طور مستقیم با اینها کاری ندارد و در هر زمینه می‌توانید از هر چیزی که دوست دارید استفاده کنید. React تنها به کمپوننت‌های UI صفحه تمرکز دارد.

^۲ <https://www.robinwieruch.de/essential-react-libraries-framework>

^۳ <https://www.robinwieruch.de/reasons-why-i-moved-from-angular-to-react>

تمرین

اگر دوست دارید بدانید چرا React را انتخاب کردم و از کجا سراغ React آدمم یا می‌خواهید دید عمیق‌تری نسبت به چیزهایی که در بالا ذکر کردم پیدا کنید مقاله‌های زیر را مطالعه کنید:

- چرا از Angular به React مهاجرت کردم^۱
- اکوسیستم انعطاف‌پذیر React^۲
- چگونه یک فریم‌ورک را یاد بگیریم^۳

نیازمندی‌ها

برای استفاده از این کتاب باید از پیش با مفاهیم اولیه‌ی برنامه‌نویسی وب یعنی HTML، CSS و Javascript آشنا باشید. اگر تا حدی مفهوم API^۴ را بدانید خیلی خوب خواهد بود. توصیه می‌کنم عضو گروه Slack این کتاب^۵ هم بشوید تا بتوانید پرسش‌هایتان را مطرح کنید و هم از دیگران چیز یاد گرفته و به دیگران کمک کنید.

ادیتور و ترمینال

برای نوشتن برنامه‌های React به یک ادیتور و همچنین یک ترمینال (پنجره‌ی خط فرمان) نیاز دارید. در این زمینه می‌توانید راهنمای قدم به قدم^۶ مطالعه کنید. مقاله را برای کاربران MacOS نوشته‌ام، اما می‌توانید بیشتر ابزارهای معرفی شده را در سیستم عامل‌های دیگر هم پیدا کنید. در اینترنت هم هزاران مقاله وجود دارد که توضیح می‌دهد چطور می‌توانید محیط برنامه‌نویسی خود را در سیستم عامل مد نظرتان به شکل حرفه‌ای راه‌اندازی کنید.

اگر دوست داشتید می‌توانید برای برنامه‌ای که در این کتاب درست می‌کنیم یک مخزن Github هم درست کنید تا در حین تکمیل برنامه بتوانید تغییرات کدها را در مخزن خودتان ثبت کنید. در این زمینه نیز یک راهنما^۷ درست کرده‌ام تا به سرعت بتوانید مخزن خود را درست کرده و از git و Github استفاده کنید.

اگر اصطلاحات Git و Github برایتان جدید است نگران نشوید. Git یک سیستم ورژن کنترل فوق‌العاده است که سابقه‌ی تغییرات کدها را ثبت می‌کند. مزیت استفاده از Git این است که اگر جایی اشتباه کردید می‌توانید به سابقه‌ی کدها برگشته و کد قبلی را احیا کنید. Github هم سایتی است که می‌توانید مخزن پروژه را در آن بگذارید تا دیگران هم بتوانند از پروژه‌ی شما استفاده کنند.

¹ <https://www.robinwieruch.de/reasons-why-i-moved-from-angular-to-react>

² <https://www.robinwieruch.de/essential-react-libraries-framework>

³ <https://www.robinwieruch.de/how-to-learn-framework>

⁴ <https://www.robinwieruch.de/what-is-an-api-javascript/>

⁵ <https://slack-the-road-to-learn-react.wieruch.com>

⁶ <https://www.robinwieruch.de/developer-setup>

⁷ <https://www.robinwieruch.de/git-essential-commands>

با این حال دوباره یادآوری می‌کنم. درست کردن مخزن Git برای پروژه و قرار دادن آن در Github الزامی نیست. به خصوص اگر تازه دارید برنامه‌نویسی را شروع می‌کنید این که علاوه بر React بخواهید با git و Github هم درگیر شوید می‌تواند شما را گیج کند. از این رو پیشنهاد ایجاد مخزن را فراموش کنید تا بتوانید تنها روی مطالب کتاب تمرکز کرده و آنها را بهتر یاد بگیرید.

NPM و Node

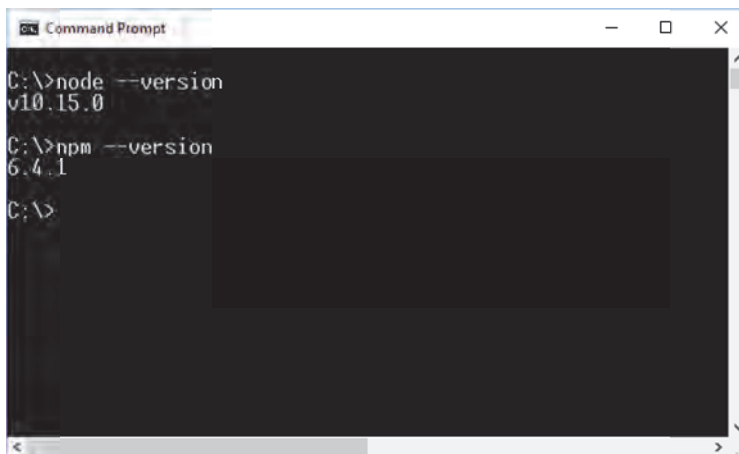
و در نهایت باید نسخه‌ای از `node` و `npm` را هم در کامپیوترتان داشته باشید. Node فریم‌ورکی است که زبانش جاوااسکریپت است و عموماً برای تولید برنامه‌های شبکه و سرور به کار می‌رود. همچنین می‌توان با آن ابزارهای خط فرمان هم درست کرد. `npm` یا `node package manager` هم ابزاری است که برای نصب کتابخانه‌ها یا پکیج‌های `node` به کار می‌رود. هر پکیج خارجی یا اکسترنال می‌تواند از تعدادی تابع ساده تشکیل شود، یک کتابخانه‌ی کوچک بوده یا حتی کتابخانه‌ای به بزرگی یک فریم‌ورک کامل باشد. از `npm` برای مدیریت و نصب پکیج‌هایی که در طول کتاب به آنها نیاز داریم استفاده می‌کنیم.

محض گرم شدن بگذارید نسخه‌ی `node` و `npm` کامپیوتر را چک کنیم ببینیم چه نسخه‌ای از آنها نصب است. برای این کار باید از دستورهای زیر در پنجره‌ی خط فرمان استفاده کنید. شماره نسخه‌هایی که اینجا می‌بینید مربوط به `node` و `npm` کامپیوتر من در زمان تالیف کتاب است. به احتمال زیاد شماره نسخه‌های شما چیز دیگری خواهد بود.

```
> node --version
*v10.15.0

> npm --version
*v6.4.1
```

تصویر زیر اجرای این دستورات را در ویندوز نشان می‌دهد.



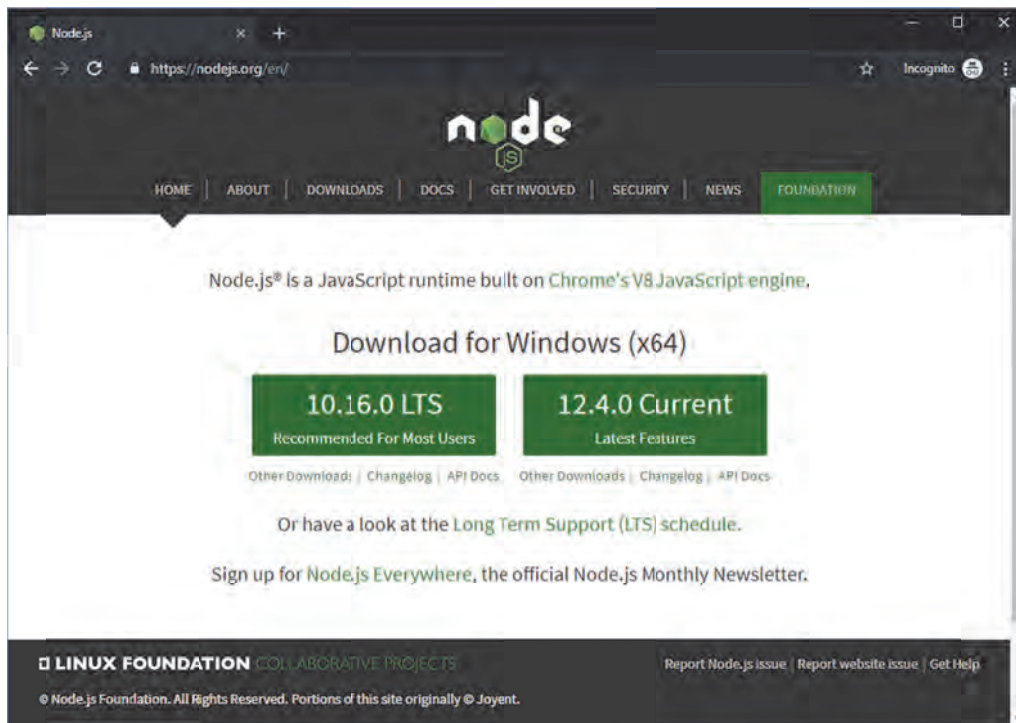
```
Command Prompt
C:\>node --version
v10.15.0

C:\>npm --version
6.4.1

C:\>
```

¹ <https://nodejs.org/en>

اگر هنگام اجرای این دستورها پیام خطا دریافت کردید node و npm در سیستم شما وجود ندارد و باید آنها را نصب کنید. برای این کار به سایت nodejs.org مراجعه کنید.



مروری بر npm

در این قسمت مرور سریعی بر npm خواهیم داشت. زیرا برای کار با React به آن نیاز داریم. البته مطالب این قسمت بسیار خلاصه و کوتاه بوده و آموزش جامعی محسوب نمی‌شود، اما چیزی را که نیاز داریم پوشش می‌دهد. اگر از پیش با node و npm آشنا هستید می‌توانید از خواندن این قسمت صرفنظر کنید.

نصب پکیج‌های node

همان گونه که گفتیم با استفاده از node package manager یا به اختصار npm می‌توانید از طریق خط فرمان پکیج‌های جاوااسکریپتی نصب کنید. به طور ساده هر پکیج مجموعه‌ای از یک یا چند فایل است که می‌تواند حاوی تعدادی تابع، یک کتابخانه یا فریم‌ورک بزرگی مانند Angular باشد. به پکیج‌های استفاده شده در برنامه وابستگی‌های برنامه گفته می‌شود. برای نصب پکیج توسط npm از فرمان install استفاده می‌شود.

```
> npm install <package>
```

می‌توانید بجای کلمه‌ی install از حرف i هم استفاده کنید.

```
> npm i <package>
```

همچنین می‌توانید با مشخص کردن نام چندین پکیج پشت سر هم همه‌ی آنها را همراه هم نصب کنید.

```
> npm install <package1> <package2> <package3> ...
```

انواع نصب پکیج

نصب پکیج دو حالت دارد. می‌توانید پکیج را به صورت سراسری یا global نصب کنید. در این حالت پکیج مزبور در فولدیری که خود برنامه‌ی node در آن قرار دارد نصب می‌شود و توسط همه‌ی برنامه‌های کامپیوتر نیز قابل استفاده خواهد بود. حالت دیگر نیز نصب محلی است، یعنی پکیج را تنها برای برنامه‌ی خودتان نصب می‌کنید.

برای نصب پکیج به صورت سراسری باید از سوئیچ -g یا global- استفاده کنید.

```
> npm install -g <package >
```

در حالت عادی وقتی سوئیچی برای دستور npm install مشخص نمی‌کنید پکیج به صورت محلی در فولدر جاری نصب می‌شود. محض تفریح بگذارید React را نصب کنیم. برای این کار ابتدا یک فولدر درست کنید. سپس از طریق خط فرمان وارد آن شده و دستور زیر را اجرا کنید.

```
> npm install react
```

```

C:\>md my-react-app
C:\>cd my-react-app
C:\my-react-app>npm install react
npm WARN saveError ENOENT: no such file or directory, open 'C:\my-react-app\package.json'
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN enoent ENOENT: no such file or directory, open 'C:\my-react-app\package.json'
npm WARN my-react-app No description
npm WARN my-react-app No repository field.
npm WARN my-react-app No README data
npm WARN my-react-app No license field.

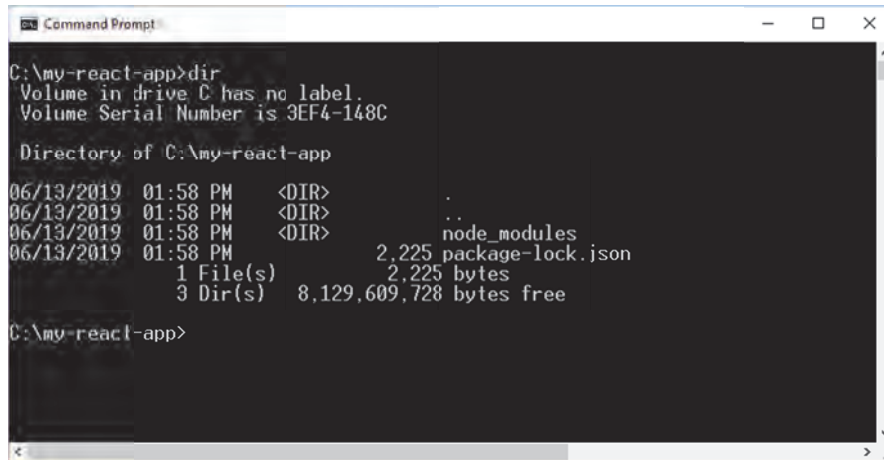
+ react@16.8.6
added 7 packages from 3 contributors and audited 13 packages in 6.541s
found 0 vulnerabilities

C:\my-react-app>tree
Folder PATH listing
Volume serial number is 3EF4-148C
C:..
|-- node_modules
    |-- .bin
    |-- js-tokens
    |-- loose-envify
    |-- object-assign
    |-- prop-types
    |-- lib
    |-- react
        |-- cjs
        |-- umd
    |-- react-is
        |-- cjs
        |-- umd
    |-- scheduler
        |-- cjs
        |-- umd
C:\my-react-app>

```

پس از اجرای دستور `npm install react` در فولدر شما زیر فولدری به نام `node_modules` ایجاد شده و React داخل آن قرار می‌گیرد. همان گونه که در تصویر می‌بینید داخل فولدر `node_modules` علاوه بر فولدرهای دیگری هم ایجاد شده است. این فولدرها پکیج‌هایی هستند که React به آنها وابستگی دارد. هنگامی که توسط `npm` پکیجی را نصب می‌کنید آن پکیج و همچنین وابستگی‌های آن نصب می‌شود.

اگر فولدر محل نصب شما خالی بوده باشد پس از نصب نخستین پکیج، فایلی به نام `package-lock.json` هم در آن ایجاد خواهد شد.



```

C:\my-react-app>dir
Volume in drive C has no label.
Volume Serial Number is 3EF4-148C

Directory of C:\my-react-app

06/13/2019  01:58 PM  <DIR>          ..
06/13/2019  01:58 PM  <DIR>          .
06/13/2019  01:58 PM  <DIR>          node_modules
06/13/2019  01:58 PM                2,225 package-lock.json
               1 File(s)                2,225 bytes
               3 Dir(s)      8,129,609,728 bytes free

C:\my-react-app>

```

فایل `package-lock.json` برای ثبت نسخه‌ی دقیق پکیج‌ها و وابستگی‌های آنها به کار می‌رود. کمی جلوتر درباره‌ی این فایل توضیح می‌دهیم.

فایل package.json

هر برنامه می‌تواند دارای مشخصات و تنظیمات باشد. یکی از این مشخصه‌ها لیست وابستگی‌ها یا پکیج‌های برنامه است. طبیعا بهتر است لیست وابستگی‌ها جایی ثبت شود تا بعدا از روی آن بتوانیم بفهمیم برنامه به چه چیزهایی وابسته است. برای ثبت مشخصات و وابستگی‌های برنامه از فایلی به نام `package.json` با قالب JSON استفاده می‌شود. در زیر نمونه‌ای از یک فایل `package.json` نشان داده شده است.

```

{
  "name": "my-react-app",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "author": "",
  "license": "ISC",
  "dependencies": {
    "react": "*"
  }
}

```

لیست وابستگی‌های برنامه در قسمت “dependencies” ذکر می‌شود. ابتدا نام پکیج و سپس شماره‌ی نسخه‌ی آن ذکر می‌شود. اگر شماره‌ی نسخه را به طور صریح مشخص کنیم یعنی برنامه دقیقاً از این نسخه استفاده کند. می‌توانیم مانند بالا از کاراکتر ستاره * هم استفاده کنیم. این یعنی برنامه با هر نسخه از این پکیج کار می‌کند. همچنین می‌توانیم با کاراکتر ^ مشخص کنیم برنامه با هر نسخه‌ای بزرگ‌تر از نسخه‌ی مشخص شده کار می‌کند.

“react”: “^16.*”

اگر npm ببیند در فولدر شما فایل package.json وجود دارد هنگام نصب یک پکیج به طور خودکار نام آن را به بخش dependencies فایل package.json اضافه می‌کند. به عبارت دیگر نام پکیج را به لیست وابستگی‌های پروژه افزوده یا لیست وابستگی‌ها را به‌روز می‌کند.

تفاوت فایل package.json با package-lock.json چیست؟

ممکن است بپرسید چه تفاوتی بین فایل package.json با فایل package-lock.json وجود دارد. اگر لیست وابستگی‌های برنامه در فایل package.json ذکر شده دیگر چه نیازی به فایل package-lock.json هست؟

پاسخ این است که در package.json لیست وابستگی‌های خود برنامه ذکر می‌شود اما در package-lock.json لیست وابستگی‌ها و همچنین وابستگی‌های آن وابستگی‌ها ثبت می‌شود. همچنین در package.json می‌توانید شماره‌ی دقیق پکیج‌ها را مشخص نکرده و اعلام کنید برنامه با هر نسخه‌ای از آن پکیج (یا هر نسخه‌ای بیشتر از یک شماره نسخه‌ی به خصوص) کار می‌کند. اما در فایل package-lock.json شماره‌ی دقیق پکیج‌ها (در زمان نصب‌شان) ثبت می‌شود.

اهمیت فایل package-lock.json این است که با مراجعه به آن می‌توانید متوجه شوید دقیقاً چه نسخه‌ای از پکیج‌ها نصب شده است (به هر حال باید مشخص شود زمانی که فلان پکیج برای برنامه نصب شده کدام نسخه از آن نصب شده است). با فایل package-lock.json می‌توانیم پروژه را بدون نیاز به کپی کردن فولدر node_modules در اختیار دیگران قرار بدهیم بدون آن که مجبور باشیم فولدر node_modules و تغییرات آن را در مخزن سورس کنترل برنامه قرار بدهیم. برای ردگیری تغییرات فولدر node_modules و نسخه‌ی پکیج‌های برنامه نیز تنها کافی است فایل package-lock.json را در سورس کنترل نگهداری کنیم.

اگر کسی پروژه‌ای دریافت کند که فولدر node_modules نداشته باشد با استفاده از package-lock.json می‌تواند فولدر node_modules و محتویات آن را و در واقع درختواره‌ی وابستگی‌های برنامه را دقیقاً به همان شکلی که صاحب برنامه در کامپیوتر خود داشته ایجاد کند. این کار به تنهایی با در دست داشتن فایل package.json میسر نیست. زیرا همان‌گونه که دیدید در فایل package.json می‌توانید برای پکیج‌ها شماره‌ی نسخه ذکر نکنید.

ایجاد سریع فایل package.json

با توجه به آنچه بیان شد این که پروژه‌ی ما فایل package.json داشته باشد ایده‌ی خوبی است. با وجودی که می‌توانیم این فایل را به طور دستی درست کنیم یک راه سریع برای ایجاد این فایل استفاده از دستور npm init است.

```
> npm init
```

پس از اجرای این دستور، npm چند پرسش در مورد مشخصات برنامه‌ای که قصد ایجادش را دارید از شما می‌پرسد، مانند نام برنامه، نام شما به عنوان سازنده‌ی برنامه، شماره نسخه‌ی برنامه، توضیح برنامه و پروانه یا مجوز استفاده از برنامه توسط دیگران. سپس بر اساس پاسخ‌های شما فایل package.json را ایجاد می‌کند. اگر این پرسش‌ها حوصله‌تان را سر می‌برد می‌توانید با استفاده از سوئیچ -y از نمایش آنها صرف‌نظر کنید تا فایل package.json بی‌معطلی برای شما ایجاد شود.

```
> npm init -y
```

در این حالت npm برای پرسش‌هایی که بیان کردم از مقدار پیش فرض استفاده می‌کند. مثلاً برای نام برنامه از نام فولدر جاری و برای مجوز یا پروانه‌ی استفاده از MIT استفاده می‌کند.

بازیابی لیست وابستگی‌های برنامه

اما ببینیم اگر پروژه‌ای دریافت کنیم که فولدر node_modules نداشته باشد چطور می‌توانیم پکیج‌هایش را بازیابی کنیم. کار بسیار ساده‌ای است. می‌توانید با اجرای دستور npm install در فولدر برنامه همه‌ی پکیج‌ها و وابستگی‌های برنامه را یکجا نصب کنید (البته به شرطی که پروژه‌ی مزبور فایل package.json داشته باشد). در این حالت npm به فایل package.json (و package-lock.json در صورت موجود بودن آن) مراجعه کرده و به طور خودکار همه‌ی وابستگی‌های پروژه را که در package.json ذکر شده نصب می‌کند.

```
> npm install
```

سوئیچ --save-dev

بگذارید یک سوئیچ مفید دیگر را هم ذکر کنم. با استفاده از سوئیچ --save-dev در دستور npm install می‌توانید مشخص کنید پکیجی که نصب می‌کنید تنها در زمان development مورد نیاز برنامه بوده و در زمان انتشار و در محیط تولید یا production نیازی به آن نیست.

```
> npm install --save-dev react
```

اما چه نوع پکیجی تنها در زمان توسعه استفاده می‌شود؟ یک نمونه از این مساله ابزارهای تست است که برای تست کردن برنامه به کار می‌رود. تست برنامه عملی است که تنها در زمان توسعه‌ی برنامه انجام شده و پس از تولید برنامه و در زمان اجرا نیازی به این ابزارها نیست.

Yarn^۱

می‌توانید برای مدیریت پکیج‌ها و وابستگی‌های برنامه‌ی خود بجای npm از ابزاری به نام Yarn هم استفاده کنید. دستورات Yarn کمی با npm فرق دارد اما نام پکیج‌های آن با npm یکسان است. Yarn برای رفع مشکلاتی که npm توان رفعشان را نداشت ابداع شد. امروزه هر دوی این ابزارها به رشد و بلوغ کاملی رسیده‌اند. از نظر کدنویسی هیچ فرقی نمی‌کند کدام یک را استفاده کنید. هر کدام را دوست داشتید می‌توانید استفاده کنید. اما از نظر سرعت، Yarn سریع‌تر از npm پکیج‌ها را نصب می‌کند.

تمرین

- با استفاده از npm یک پروژه‌ی جدید ایجاد کنید.
 - یک پنجره‌ی خط فرمان باز کنید.
 - با دستور mkdir یا md یک دایرکتوری ایجاد کنید.
 - با دستور cd یا cdوارد آن شوید.
 - با استفاده از npm یک فایل package.json در آن ایجاد کنید.
 - پکیج React را به طور محلی در آن نصب کنید.
 - پس از ایجاد پروژه محتوای فایل package.json، package-lock.json و فولدر node_modules را بررسی کرده و ببینید چه نسخه‌ای از React نصب شده است.
 - در پایان ببینید چگونه می‌توانید وابستگی برنامه به React را حذف کنید.
- با مراجعه به سایت مستندات npm به آدرس <https://docs.npmjs.com> دانش خود را گسترش دهید.
- درباره‌ی yarn تحقیق کنید.

نصب

خوب. پس از مرور سریع npm برویم سراغ React. به طور کلی برای استفاده از React چند راه وجود دارد. یک راه این است که React را از طریق CDN استفاده کنیم. اگر نمی‌دانید CDN چیست باید بگویم CDN یا **Content Delivery Network**^۲ یک شبکه‌ی تحویل محتوا است که فریم‌ورک‌ها و کتابخانه‌های معروف در آن نگهداری شده و به طور رایگان با سرعت دسترسی بسیار بالا جهت استفاده به طور عمومی فراهم می‌شود. بدین ترتیب بجای این که فایل کتابخانه‌ای مانند React را دانلود کنید در پروژه‌ی خود قرار بدهید و در صفحه‌ی وب خود به آن ارجاع کنید از آدرس CDN آن استفاده می‌کنید.

مزیت شبکه‌های CDN این است که بسیار سریع هستند، سرورهای متعددی در نقاط مختلف دنیا دارند و درخواست‌ها را از نزدیک‌ترین سرور (بر حسب موقعیت جغرافیایی) پاسخ می‌دهند. در نتیجه پهنای باند سرور شما هم برای سرویس‌دهی فایل مورد نظر هدر نمی‌رود.

^۱ <https://yarnpkg.com/en/docs/>

^۲ https://en.wikipedia.org/wiki/Content_delivery_network