How to learn React.js in 2020

JANUARY 01, 2020 BY ROBIN WIERUCH - EDIT THIS POST

Follow on Twitter 17k Follow on Facebook

How to learn React is high on the agenda for plenty of JavaScript developers for this year. The recent State of JS survey has shown that many developers are content with React for creating modern web applications. However, due to its popularity and job demand, there are still many JavaScript developers who want to learn React. In this guide, I want to give beginners a comprehensive overview of how to approach learning React without getting distracted or overwhelmed by other topics along the way.

LEARN REACT [QUICKLY || FAST || EASY]

There is no way to learn something the easy way. You have to be patient. That applies for developers too. Personally, that's why I like being a developer, because there will be always something new to learn. If you stay curious, you will not be disappointed while becoming a developer. Learning is always a challenge, a challenge to hone your skills, if challenge and developer. Learning is always a challenge, a challenge to hone your skills, if challenge and skill at hand are in balance. That's why it takes time to learn React as well.

However, React does not have a steep learning curve when following one simple rule: Learn React, **only React step by step**, and don't let yourself distract from other tech on this journey. JavaScript and React are evolving constantly, because both stay innovative and want to stay relevant in the future too; so learn and keep up with them first before learning anything else on top of it. I have heard success stories from developers who went from not knowing React to getting offered a React job at a company from a few days to a few months. On the job they got the chance to learn more about all the other fancy technologies that come along with React.

> *"I had a job interview coming up where I was asked to complete a project in React. I sat down on a Saturday and worked through The Road to Learn React and on Sunday I completed my project with plenty of time left before my interview Wednesday. In one weekend I went from 0% to 80% comfortable working independently in React."* (Source)

There are a couple of tech stacks that people want to learn with React straightaway. I want to pick up a few of them and give a brief explanation why it isn't a good idea to marry them with React while learning it:

- **Learn React with TypeScript?** No way, learn React first before converting JavaScript to a typed language. Not only the vanilla JavaScript code will be typed with TypeScript, but also your React components and everything that comes with them. It will become a mess and overwhelming when you haven't learned React itself from scratch. Also 99% of the React tutorials and courses out there are not written in TypeScript, so it makes things more complicated than easier.

- **Learn React with Redux?** Redux is a popular state management library for JavaScript. Again, learn React and its built-in state management first before reaching out to a sophisticated external state management library which is meant to be used in large scale and complex applications. React already comes with plenty of tools to handle state in your application. The vast majority of React applications out there doesn't even need Redux or any other state management library. Learn how to deal with the state in React first before throwing another library on the problem. Redux makes React more complicated for beginners.

- **Learn React with Gatsby?** Gatsby.js got quite popular in the recent time. It's the go-to solution for creating static websites such as personal websites, blogs and landing pages. However, Gatsby comes with its own learnings such as GraphQL, its large plugin system, and the problems coming with server-side rendered React applications. So don't mistake the problems coming with server-side rendered React applications. So don't mistake

Gatsby.js for having an easier time learning React. It makes writing static websites with React easier, but not learning React itself.

These were only a few things React beginners tend to associate with React when starting out with it. There are definitely more of them. However, they will not make the learning experience easier. In contrast, you will have to learn two things instead of one. So start out with React first before learning anything on top of it. Learn one thing at a time.

REACT PREREQUISITES

Since React is a JS library, there is no way around learning JavaScript. You cannot learn React without JavaScript or without knowing JavaScript in the first place. In comparison to other frontend solutions, React makes heavily use of JavaScript.

Learn React or JavaScript first?

You should understand at least the basics of JavaScript, often referred to as JavaScript ES5, and have decent knowledge about HTML and CSS. React also utilizes JavaScript syntax, most often referred to JavaScript ES6 and beyond (ES7, ES8, ES9). That's why it makes sense to get comfortable with at least JavaScript ES6 to learn React.

There are many people who recommend to learn web development and *all* JavaScript fundamentals as prerequisite before learning React. The skills they recommend to acquire go far beyond what's needed to build basic applications with React.js. A couple of them may be:

- Functions are first-class citizens in JavaScript
- bind(), apply(), and call()
- scopes and closures
- this
- Object Methods
- Array Methods
- Prototypical Inheritance -> ES6 Class
- Callbacks and Promises
- Event Loop
- Event Bubbling
- Regular Expressions
- Error Handling

- JavaScript ES6 and beyond

- Hoisting, Memoization
- Declarative vs imperative programming
- Functional vs object-oriented programming

However, I believe it's not true to have all this knowledge. In contrast, it can be very disillusioning for beginners being new to JavaScript, React and Web Development. You don't want to hang around in the "Scopes and Closures"-land while you could build real projects instead that motivate you to keep going. The "Aha"-moment for grasping closures will come along the way. Personally I didn't know about half of the things in the list when I started to learn React and I still have to look up regular expressions online because I don't really bother learning them; and I still help companies with their JavaScript projects.

I believe **most of JavaScript can be learned along the way when learning React**, because then you experience it in context while building web applications. That's exciting and that's what drives us as human beings. So keep yourself excited while learning and don't hang around too long with the fundamentals where you cannot see the purpose without ever applying them in a real world scenario. Start to build real projects. Stay curious! Move forth and back between JavaScript and React while you learn React.

Learn React or jQuery?

Historically jQuery filled the gap between vanilla JavaScript and modern libraries/frameworks such as React, Angular, and Vue. Early JavaScript wasn't as efficient in the browser and other subjects as it is nowadays. That's why there was a real need for jQuery. However, modern JavaScript is capable of dealing with all the things itself. Hence my recommendation: Don't bother about jQuery.

If you are working on a larger application that utilizes React at your new job, it is likely to be confronted with jQuery in the application's legacy areas. But it's okay to not being proficient with jQuery then and just look up online what you need to get the job done. As alternative, try to migrate the code from jQuery to React instead as a learning opportunity. That's how I did it in my last job, because I became a JavaScript developer in the early days of Angular, but never really had to learn jQuery to a degree of being proficient with it. Instead I focused on the journey ahead with Angular which then became React.

Why should I learn React?

Why do you want to learn React in the first place? Beginners know about all the shiny modern libraries in JavaScript but often don't ask themselves why they need them. They may only know that they are in huge demand by companies and jobs. However, sometimes it makes sense to backpedal to the question: *Why do I need this?*

It doesn't only apply to React, but to all the other libraries and frameworks you will learn in your life as a developer. What problem does it solve for me? Too often people throw libraries on top of their tech stack without experiencing the problem in the first place. That's why I believe it is a great learning experience to implement the identical application with vanilla JavaScript and React. It demonstrates you which problem the new library is solving for you. The same technique can be applied when deciding whether you want to learn React or another solution such as Vue or Angular. Build a basic application with the solutions of your choice and compare your the development experience. Which of the solutions felt just right for you?

LEARN REACT.JS STEP BY STEP

After we got clarity about learning *only* React and React's prerequisites, let's see how to learn React from zero from there. I strongly believe the best start is going through the official React documentation. The documentation keeps up with the recent changes, is not biased, teaches you everything the React way, and is put together by the React core team and the React community in your interest.

Learn with React Docs

Being referred to the documentation of a library as a beginner in web development can be intimidating. But I strongly believe it's the best way to learn something new. On your career path to becoming a developer, you will get used to learn new things with the documentation of the new thing, so why not get used to it now? It comes with certain benefits to learn documentation driven:

- beginner tutorial
- no commercial interest
- opinionated way of doing things
- thoughtfully put together by core contributors and community
- latest material and updates for new versions
- API references

Every time I pick up a new tech, I don't check any courses or books, I go straight to the tech's documentation and check whether there is a beginner tutorial. Often there is one and that's my entry point into a new world. From there, I can always check for other learning material, but my starting point will always be the official documentation of the new thing I intend to but my starting point will always be the official documentation of the new thing I intend to learn. After all, that's why I write my tutorials and books in close relation to the

documentation by always refereing it in my guides. That's how beginners learn to use the documentation and that's what will make them more efficient and effecrtive when learning a new tech in the future.

Learn React with Projects

After you went through the React beginner tutorial from the React documentation and learned the basics of React from scratch, continue learning by building a project yourself before learning any other tech. It's not always simple to come up with project ideas, but there is plenty of inspiration out there.

For instance, there are also people who want to learn React by building games. I guess you will find plenty of React Tic Tac Toe implementations out there. Implement the game in React yourself and compare your solution to the other solutions. You will learn from your mistakes and learn to evaluate your source code against the implementation by others.

Learn React by Doing

Learn React by doing is one of the most common recommendations you will get to hear from other developers, it applies to almost everything in life, because every task will present you with a new challenge to grow as a React developer. Also it is too easy to passively consume video courses, books, and tutorials. Get your hands dirty and implement something yourself. Get stuck with a problem, become desperate, ponder over the problem by taking time off the screen, solve the problem eventually, and reap the fruits of it. That's the basic framework of learning anything.

Learn React every Day

Confront yourself to stick to coding every day. You can participate in the 100 Days of Code challenge, write about it on Twitter or on your website. Make it a daily habit. Only when sticking to it for a while, by having the persistence and by coding every day, you will eventually become a React developer. It's a marathon and not a sprint.

Learn React in Public

The best way to learn something new is to learn it in public. Confront yourself with the feedback from other React developers by showing off your achievements. If you have reached a milestone with your React project, show it to others and ask for their feedback. These are a couple of commonly known platforms to share ideas around React:

- Twitter
- Reddit
- Reactiflux
- Spectrum

You will not only get feedback from people there, but also dive more into React by joining discussions. Be part of the community. Another great way to learn something new yourself is to explain it to someone else. Maybe you are one step ahead of a fellow React beginner and can help them out. Checkout the *React Beginner Thread* in the *React Subreddit* where you can help fellow React developers to solve their problems.

Learn to use React

After you went through the beginner tutorial in React's docs and started to implement a React project yourself, learn to use React with all its facets. React has only a slim API surface area, so try to get proficient with it.

- Learn about React's JSX syntax
- Learn about React Components
- Learn about React Component Composition
- Learn about React Props
- Learn about React State

Learn and understand React. In the end, it doesn't need much to create a React component that returns JSX, to use a React component in another React component while arranging them in a component hierarchy, to pass data from component to component with props, and to make components interactive with state. That's the gist of React. Learn React in depth by applying your learned skills in your project and not by consuming endless of content.

REACT SETUP FOR BEGINNERS

How to get started with actual coding a React project? If you don't have any editor/terminal or IDE installed and just want to checkout how React feels like in a code environment, checkout the go-to online code editor called CodeSandbox. Play around with React and checkout whether you like it. However, if you want to learn how to develop React applications on your local machine, checkout these MacOS and Windows Setup Guides to make it work on your machine:

- Set up React on MacOS

- Set up React on Windows

Both guides come with a minimal set of recommended tools to get you started with React:

- **Execute JavaScript with Node and NPM:** While Node.js makes it possible to run JavaScript outside of the browser, NPM (node package manager) allows you to install libraries like React to your project. You will need both tools for almost any other modern JavaScript project as well.
- **Develop with Visual Studio Code:** A popular lightweight IDE that is the go-to solution for React developers. Visual Studio Code (VS Code) comes with an editor to learn React development and a terminal (command line) to start your React project and install libraries for it.
- **Learn with create-react-app:** The go-to solution from Facebook and the React team to get you started with a lightweight and zero-configuration React starter project. 99% of the React tutorials out there build up on top of this starter project to get you started with React. You can focus only on learning React while all the toolings around it are taken care of by create-react-app.
- **Format with Prettier:** Opinionated code formatter that automatically formats your code in your development environment. It comes with a straightforward setup and integrations for popular editors (VS Code). While you learn React web development, it shows you how to format your React code.

LEARN ADVANCED REACT

After you have learned the basics of React with React components, props/state, and React's JSX syntax to return valid HTML in JavaScript, you can dive deeper with advanced React concepts and patterns. These are a few essential of them that will not go away that quickly:

- Learn React Hooks
- Learn React Higher-Order Components
- Learn React Render Prop Components
- Learn React Context

Being equipped with these advanced React techniques, you should be comfortable to dive into larger React applications where you will find these concepts and techniques more often. Due to React Hooks, the other Higher-Order Components and Render Prop Components may be less used in the future, but you should still see them fairly often throughout the next year.

LEARN REACT ONLINE

After you went through React's documentation to learn about the basics and a few advanced concepts and patterns, and after you have started to build a React application yourself, definitely checkout other React resources online. Note that I mention this very late in this guide, because often React beginners are stuck in a never ending story of just consuming content to learn React. The thing I want to point out: Start to produce by developing React applications yourself or by writing about your learnings. Learning by doing is the key here. Every problem along the way can be looked up online. And then if you want to learn more about React, checkout all the other resources to learn React online. My recommendation is to not only consume one kind of learning material, but to diversify your learning resources:

- React Tutorials
- React Books
- React Videos (YouTube, Twitch, Online Courses)
- Web Development or React Podcasts
- Interactive Courses

However, not every kind of learning material may suit you. The same applies to the teaching styles of the people behind the larning material. Everyone has a different way to produce content and to offer it to their students, so checkout which teaching style is best for you.

> *"Initially, when first approaching React, I had purchased a Udemy course from another teacher. However, I found myself getting quite lost after just a few videos. The teacher in that highly rated series kept it easy at first but then blindsides you with hard to grasp concepts and code and failed to inform well what was going on.* (Source)

If you are into reading, checkout online tutorials and React books. Print books are a good resources to learn React offline as well. Even though many people are afraid that books get outdated too quickly, I can only argue that most books are self-published and it takes the author only 1 day to have a new version of it online (even on Amazon). I believe it takes longer to update a video than written content nowadays.

If you are into watching, checkout online courses by individual developers on YouTube, Udemy and their own course platforms. However, again don't get stuck in only watching the content. Find content where you are forced to apply your learnings.

If you just want to keep up with recent React news, Podcasts are a great way to stay ahead of the curve. You will listen to developers being on the bleeding edge of the technology who always release something new to learn. Also Podcasts are a great way to keep learning React while commuting or while exercising in the gym. That's how I do it at least.

There are plenty of free React learning resources online that have a high quality and are accessible to everyone. If anyone asks me about becoming a React developer, I often just send them straightaway to FreeCodeCamp. Regardless of which learning resource you pick, try to stay pragmatic by applying the learnings yourself. Don't just passively consume the content. Be active and challenge yourself to hone your skills.

LEARN REACT ROADMAP

After you have learned only React and feel comfortable with its basic and advanced implementation details, there are plenty of learning paths you can take to advance your React skills. The React ecosystem is not opinionated but innovative and extensive. A couple of recommendations:

- React + React Router: Enable Navigation in your React application with a Router
- React + Firebase: Connect React to a Database and implement User Authentication/Authorization
- React + GraphQL: Connect React with a GraphQL interface to a Node.js backend application
- React + Redux: Explore predictable and scalable state management in React with Redux

. . .

That's it from my side. Stay curious, commit to it every day by getting your hands dirty, be public about it, and run a marathon and not a sprint. Challenge yourself to advance your skills as a developer and become a React developer this year!

Show Comments

KEEP READING ABOUT REACT ›

HOW TO USE REACT TESTING LIBRARY TUTORIAL

React Testing Library (RTL) by Kent C. Dodds got released as alternative to Airbnb's Enzyme . While Enzyme gives React developers utilities to test internals of React components, React Testing...

HOW TO USECONTEXT IN REACT?

React's Function Components come with React Hooks these days. Not only can React Hooks be used for State in React but also for using React's Context in a more convenient way. This tutorial shows...

THE ROAD TO REACT

Learn React by building real world applications. No setup configuration. No tooling. Plain React in 200+ pages of learning material. Learn React like **50.000+ readers**.

GET THE BOOK ›

GET THE BOOK ›

Get it on Amazon.

TAKE PART

NEVER MISS AN ARTICLE ABOUT WEB DEVELOPMENT AND JAVASCRIPT.

✓ Join 50.000+ Developers

✓ Learn Web Development with JavaScript

✓ Tips and Tricks

✓ Access Tutorials, eBooks and Courses

✓ Personal Development as a Software Engineer

Your email address SUBSCRIBE

View our Privacy Policy.

PORTFOLIO

Online Courses

Open Source

ABOUT

About me

What I use