**How to Learn ReactJS in 2021**

* Last Updated : 16 Feb, 2021

*npx create-react-app myapp*

*npm start*

*npm run build*

*npm install*

Aren’t all the above commands familiar to you?

If yes then you might be working on React or you might have started working on this amazing JavaScript Library. If you’re a passionate developer, then surely you might be aware of the popularity of this JavaScript framework and you might have also tried to learn this library.

Now let’s talk about the developers who are already in the industry and working on some frontend part of the project. How do you feel when you are asked to write the same line of code at multiple places? How do you feel when you’re asked to add the same button on multiple pages?

We know that in software development doing repetitive tasks is frustrating and developers always look for some framework or method to save their precious time. Here React comes into the picture.

This popular amazing JavaScript library not just helps the developers to build an impressive application but also helps them in building the frontend page in less time. React is component-based and that’s the coolest feature of this library. You can break down your complex code into individual pieces i.e components. These components are helpful in organizing the code in a better way.

Learning this library is not easy if you don’t take the right step towards it. There are many concepts in React and it can be overwhelming to learn this library. It is important to know the right things in this library to become an in-demand React Developer. Today in this blog we are going to list down the key skills that are needed to become a proficient React Developer.

**1. Learn the Fundamentals of**[HTML](https://www.geeksforgeeks.org/html-tutorials/)**,**[CSS](https://www.geeksforgeeks.org/css-tutorials/)**, and**[JavaScript](https://www.geeksforgeeks.org/javascript-tutorial/)

Your core concepts should be strong before you jump to this library. If you’re an experienced developer, then you just need to revise the concepts.

If we talk about beginners then most of the time they make mistakes, and they directly jump to this library. Later they don’t understand the concept and keep struggling with understanding this library. Also, your interviewer will firstly check the basic concepts before jumping to React.

Understand the basic building blocks of the website. How HTML, CSS, and JavaScript work. React syntax and most of the concepts of this library are based on mainly these three building blocks. You won’t face any problems while building an application if these fundamentals are clear to you. Create something on your own using these three building blocks.

React inherits most of the features from JavaScript. Both are connected so it’s good to have a solid foundation in JavaScript concepts. Let’s discuss some basic concepts of JavaScript that are needed for React.

* Variables, objects, arrays, functions, and classes.
* Common array methods such as .map(), .filter(), .reduce()
* ES6 features such as arrow function, let, const. Learn about ‘this’ keyword, call, apply, and bind method.
* Concepts of Asynchronous JavaScript such as promises, callback, Async/Await.
* Learn how DOM (document object model) works. How to create, select, and update the HTML elements and their attributes.

**2. Fundamentals of**[NodeJS](https://www.geeksforgeeks.org/introduction-to-nodejs/)**, Import, and Export Keyword**

Being a React developer you will run a lot of NPM scripts/commands, so once the fundamentals (HTML, CSS, and JavaScript) are clear to you, understand the fundamentals of NodeJS.

* **NPM (Node Package Manager):**You should have knowledge about NPM (Node package manage). NPM is used to install the Node modules and packages. Learn how to use the basic commands/scripts to install the packages and modules.
* **Import Keyword:**You will have to use the import keyword in your React project to use the installed Node modules.
* **Export Keyword:**Use this keyword when you are creating a module/component and you have to return only a part, not all the methods and variables.

**3. Learn**[React Fundamentals](https://www.geeksforgeeks.org/reactjs/)

Once the above concepts are clear and you have become confident with JavaScript, jump to the React Fundamentals. Learn the core concepts of React. React is based on component-based architecture. You will have to divide the complete UI or JSX into reusable components.

You will be creating different components for different tasks and these components will be fitted into the parent component. This parent component will be rendered to the user.

Now the question is…what are the core concepts you should learn in ReactJS. Let’s discuss that…

* You will be using JSX syntax in ReactJS. Syntax of JSX is somewhere similar to HTML but keep in mind that it is different from plain HTML.
* Learn to render the JSX elements. Show and hide elements depending on the conditions.
* Learn the concept of State in ReactJS? How does it work? Basically, the state holds the synchronous variable, and changing the value of the state will be reflected in the whole application wherever it is used. Learn how to store and update the data in State.
* In React, you will be using props (*Hey! What it is actually?* ). We know that this might be a new word for you. Basically, props are just like the arguments passed to the function or method. Know when to use it and how to use it. You will be using props as an input argument to the HTML tags.
* Functional component, class component, and the difference between both of these.
* Just like in JavaScript, you will handle events in React. Learn how to handle some common events such as onClick, onChange, and onSubmit (You will be using these events in buttons, inputs, and forms).

Tutorial [ReactJS GeeksforGeeks](https://www.geeksforgeeks.org/reactjs/) and [React Official Tutorial](https://reactjs.org/tutorial/tutorial.html) will help you to learn everything about ReactJS. After the understanding of ReactJS build some basic projects to understand the concepts in a much better way.

* Simple todo-app
* Simple calculator app
* Build a shopping cart
* Display GitHub’s user stats using GitHub API.

**4.**[React Router](https://www.geeksforgeeks.org/reactjs-router/)

Most of the framework works on the concept of routing**(*Hey! What it is actually?*).**Whether it’s an e-commerce site, educational site, or any kind of website, you will find different pages for different purposes. Now there is a question for you…

How would you redirect from the ‘home’ page to the ‘blog’ page? You need to set routing in your application to serve this purpose. React developers use a third-party library called **react-router-dom** to create links for different pages, to load the content of different pages, and to redirect to other pages.

Below are some common features of react-router-dom you should know about to work with your application.

* How to use <Route />, <Switch /> and <BrowserRouter /> components.
* How to do navigation using the <Link />component and using the useHistory() hook.
* Creating dynamic routes using the path prop i.e. <Route path=”/posts/:post-slug” component={Post} /> and getting the path values using the useParams() hook.
* How <Redirect /> works in React Router.

**5.**[React Hooks](https://www.geeksforgeeks.org/introduction-to-react-hooks/)

So you’re done with the basic building blocks and React fundamentals. Now it’s time to learn React Hooks. This new feature was added in React version 16.8. With the help of React Hooks, you can write the React features without using the classes. React hooks allow you to write less code and that makes your app efficient, simpler, and powerful.

You will be working on mainly 5 core concepts of React Hooks. These five core concepts are given below:

* **useState:**Most of the time in your React app you will be using this concept to store and manage the data.
* **useEffect:**You will be using this concept in actions like HTTP requests andworkingwith the Browser API.
* **useRef:**This will be used to reference [JSX](https://www.geeksforgeeks.org/reactjs-introduction-jsx/)elements.
* **useContext:**This property of React Hooks will be usedto access data from React Context to share data among components easily (instead of passing props)
* **useReducer:**This concept in React hooks will be used to store and manage data across multiple components.

We have discussed only 5 hooks that are important to learn. There are more than 5 hooks but that’s not needed initially in your learning phase. As you will progress and you will encounter problems in your application, you will be learning more hooks slowly.

**6. Learn to Work With the APIs in React Apps**

We know that React is a front-end library to build the user interfaces. A user interacts with the frontend part of the application, access the data from there, or navigate from one page to another page.

React interacts with the backend code of the application. We get the data from the backend part of the application, we perform actions such as authentication, display the item to the user, searching, registration of the user, etc.

Mainly there are two ways to work with the data from the backend. The most common and standard way is using the **REST APIs**. The other and the newer one is **GraphQL API**. In your work, you will encounter problems where you will have to work with both of the APIs. Learn how to work with both kinds of APIs using React.

**7. Styling in React**

In development most of the time, we use plain CSS to give styling to our application. But in React mostly developers use the Component library for easier styling. There are many component libraries available to use but the most popular one in React is **Material UI**.

You can also use the **utility class library** to style your element. The popular utility class library is **Tailwind CSS**. Being a developer you will encounter both of these. So it’s good to be familiar with both types of libraries.

**8. Webpack**

Webpack is a module bundler in JavaScript that helps you to maintain dependencies as static files for your project. Webpack also comes with loaders. Loaders help you to run specific tasks around your project.

**9.**[Redux](https://www.geeksforgeeks.org/introduction-to-react-redux/)

In your React application, you will have to manage the states. Once your application will start growing it will become a complex application and you will have to manage the states across components. Redux which a powerful JavaScript library solves this problem and helps in maintaining the state of the application. In Redux, you store all your states in a single source.

**Final Thought**

We discussed the main concept that you should learn in React. Once you will start working on some React project, you will encounter many problems, and depending on the issues you will have to search for an efficient solution in React.

You will be introduced to different libraries or packages in React depending on the problems. Slowly you will go in-depth with this application, and you will become a master in React. Hope this was helpful.