**React Tutorial A Step-by-Step Guide(1) –**

[Kasun Dissanayake](https://kasunprageethdissanayake.medium.com/?source=post_page-----21cb088da39e--------------------------------) Apr 10, 2022

In this React tutorial, I will explain What is React, Why should you learn React, and the technical aspects of React. In the end, I will show how to start a simple React application.

**What is React?**

React is an **open-source JavaScript library** for building **user interfaces**. React is not a framework. It's a library and focuses on building User interfaces (UIs). React does not focus on routing and HTTP requests.

Then how you would handle routing, HTTP, and other functionalities. Don't worry. React has a **rich ecosystem** and is placed really well with other libraries and is more than capable of building full web applications.

**Why do you need React?**

**React is maintained by Meta (formerly Facebook) and a community of individual developers and companies.** Facebook uses react in its own products and invests money and resources to keep the project alive it's not going to die down anytime soon. It has a huge community behind it. It means you can find thousands of articles and stack overflow solutions to most of the problems that you will encounter when getting started with react. React is increasingly popular among developers nowadays.

**Technical aspects of the React**

React has **component-based Architecture**. This lets you break down your application into **small encapsulated parts** which can then be composed to make more complex UIs. For example, a traditional website can be broken down into,

* **Header**
* **SideNav**
* **Main Content**
* **Footer**

A picture containing table

Description automatically generated

Each section represents a **component** that is composed the right way to make up the entire website. The component also makes it possible to write **reusable code**. You could have a component called “Article” on react. The same component can be reused for articles on Angular or Vue by simply passing the right data into the article component. In Enterprise applications, there is an ability to reuse the code and that is a huge plus point.

**What is a Component?**

You can think of a component as a small feature that makes up a piece of the user interface. If I were to describe a component within the scope of Facebook’s UI, A chat window would be a component, a comment feed would be another component, and a constantly updating friend list would represent yet another component.

Each of these components exists within the same space, yet interacts independently from one another. Components have their own structure, their own methods, and their own APIs. Components are also reusable and can be “pasted” into interfaces at will. The independent nature of components allows for developers to create a UI with many different moving parts.

**React is Declarative**

What does declarative means? Basically, we just need to tell React what we want, and React has **React DOM Library**. React Dom Library will build the actual UI. React is declarative **because we write the code that we want** and React is in charge of taking our declared code and performing all of the JavaScript/DOM steps to get us to our desired result.

**[ReactDOM - React](https://reactjs.org/docs/react-dom.html" \t "_blank)**

[The react-dom package provides DOM-specific methods that can be used at the top level of your app and as an escape…](https://reactjs.org/docs/react-dom.html" \t "_blank)

[reactjs.org](https://reactjs.org/docs/react-dom.html" \t "_blank)

Let's say you ask an architect to draw a plan for your house. You don't tell them how to draw it, it's up to them. This is declarative. You tell what has to be done and architects will get it done for you.

Now the same task you from another junior architect and this time you tell them to draw roads first then the gardens, then the rooms and so on. You explicitly specify each step and you control the flow of the plan. That is an imperative paradigm. React is declarative. We have to tell React what the UI looks like and React will create the actual UI. React creates complex UIs by abstracting away difficult parts. React will handle efficiently updating and rendering just the right components in your application when your data changes. DOM updates which are one of the most expensive operations are handled gracefully by React.

You can **seamlessly integrate react into any of your applications**. You can integrate react portion of your page or a complete page or even the application itself. You can create Mobile applications using **React native**.

**[React Native · Learn once, write anywhere](https://reactnative.dev/" \t "_blank)**

[React Native combines the best parts of native development with React, a best-in-class JavaScript library for building…](https://reactnative.dev/" \t "_blank)

[reactnative.dev](https://reactnative.dev/" \t "_blank)

**What are the prerequisites to get started with React?**

* HTML, CSS, and JavaScript fundamentals are necessary.
* Knowledge more than JavaScript is also needed.
* The basic idea of this, filter, map, reduce keywords, and methods.
* let and const declarations, arrow functions, template literals, default parameters, object literals, rest and spread operators, and destructuring assignments.

I will discuss in the next tutorials,

* Fundamentals of React.
* React and HTTP.
* Routing with React.
* State management with Redux.
* Utilities and Libraries in React.

**Fundamentals of React**

**Setting up the development environment**

To create a React application we need two things installed.

* Node installed
* Text Editor (IDE)

Refer to these links and install Node.js stable release.

**How To Install Node**.**js** on Windows:

**[How to Install Node.js and NPM on Your Windows System](https://phoenixnap.com/kb/install-node-js-npm-on-windows" \t "_blank)**

[Introduction Node.js is a run-time environment which includes everything you need to execute a program written in…](https://phoenixnap.com/kb/install-node-js-npm-on-windows" \t "_blank)

[phoenixnap.com](https://phoenixnap.com/kb/install-node-js-npm-on-windows" \t "_blank)

If you already installed make sure to update it. For the Text Editor, you can use any IDE.

**Create React App**

To create a new React application we will be making use of **Create React App**. Create React App is a command-line interface that allows you quickly create and run react applications with no configurations. You simply run the command and the entire react project is created for you. Open your terminal in WebStorm and type this command to start a new React application.

[Create React App](https://github.com/facebookincubator/create-react-app) is a comfortable environment for **learning React** and is the best way to start building **a new**[**single-page**](https://reactjs.org/docs/glossary.html#single-page-application)**application** in React.

It sets up your development environment so that you can use the latest JavaScript features, provides a nice developer experience, and optimizes your app for production. You’ll need to have [Node >= 14.0.0 and npm >= 5.6](https://nodejs.org/en/) on your machine. To create a project, run:

npx create-react-app my-app

Once the command is completed you should have a new folder called my-app. To run this application first navigate inside the project folder and run the command “**npm start**”.

cd my-app  
npm start

Now this will open your browser on localhost port 3000 with your application up and running.

Graphical user interface

Description automatically generated

Default react Application homepage

Alternative approaches for creating React Applications using the Create React App package.

* Using npx method — NPX is an npm package runner which gets installed when you install Node. That is how we are able to run Create React App without having to install it.

npx create-react-app<project\_name>

* Using npm — Here we are going to install Create React App package globally and then use the package to generate the projects.

npm install create-react-app -g   
create-react-app<project\_name>

I hope you got a basic understanding of these theories about What is React, Why should you learn React and the technical aspects of React. In the next tutorial, we will discuss the most important topic which is the folder structure of a React Project.