

# ReactJs Roadmap 🌐 for beginners - 2021

DEV [dev.to/suhailzone/reactjs-roadmap-for-beginners-2021-14en](https://dev.to/suhailzone/reactjs-roadmap-for-beginners-2021-14en)



[#react](#) [#javascript](#) [#webdev](#) [#beginners](#)

## Pre-requisites 📝:

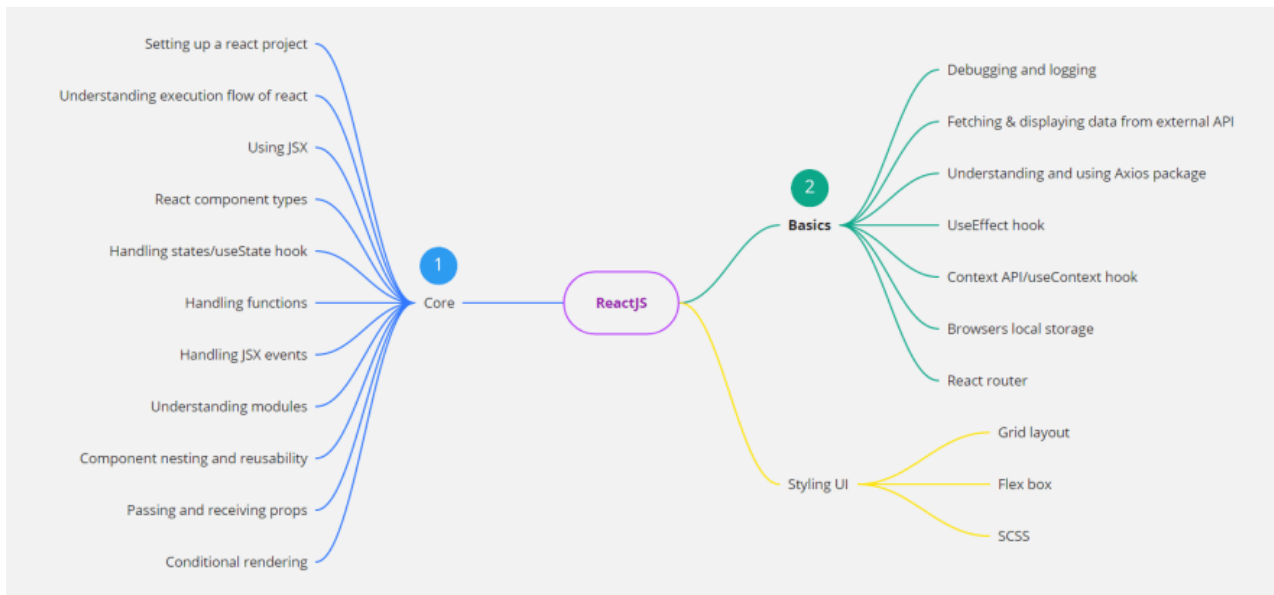
I assume that you have some basic understanding of the following:

1. Basics of HTML(such as HTML elements, form, table, div and list tags with their attributes).
2. Basics of CSS(such as styling the html elements, CSS selectors and box model).
3. Basics of JavaScript(such as variables, conditional statements, loops, data types, DOM manipulation and event triggering).

## So Before jumping into the roadmap let's see first, what is React?

React is one of the most popular JavaScript library for designing rich and interactive User Interfaces. React's component-based functionality lets you make reusable components with which you can scale your application effectively. Using React the views of your app makes your code more predictable and easier to debug.

Below is the mind-map of learning React. I have divided the learning structure into modules and then further into concepts for easy understating the path.

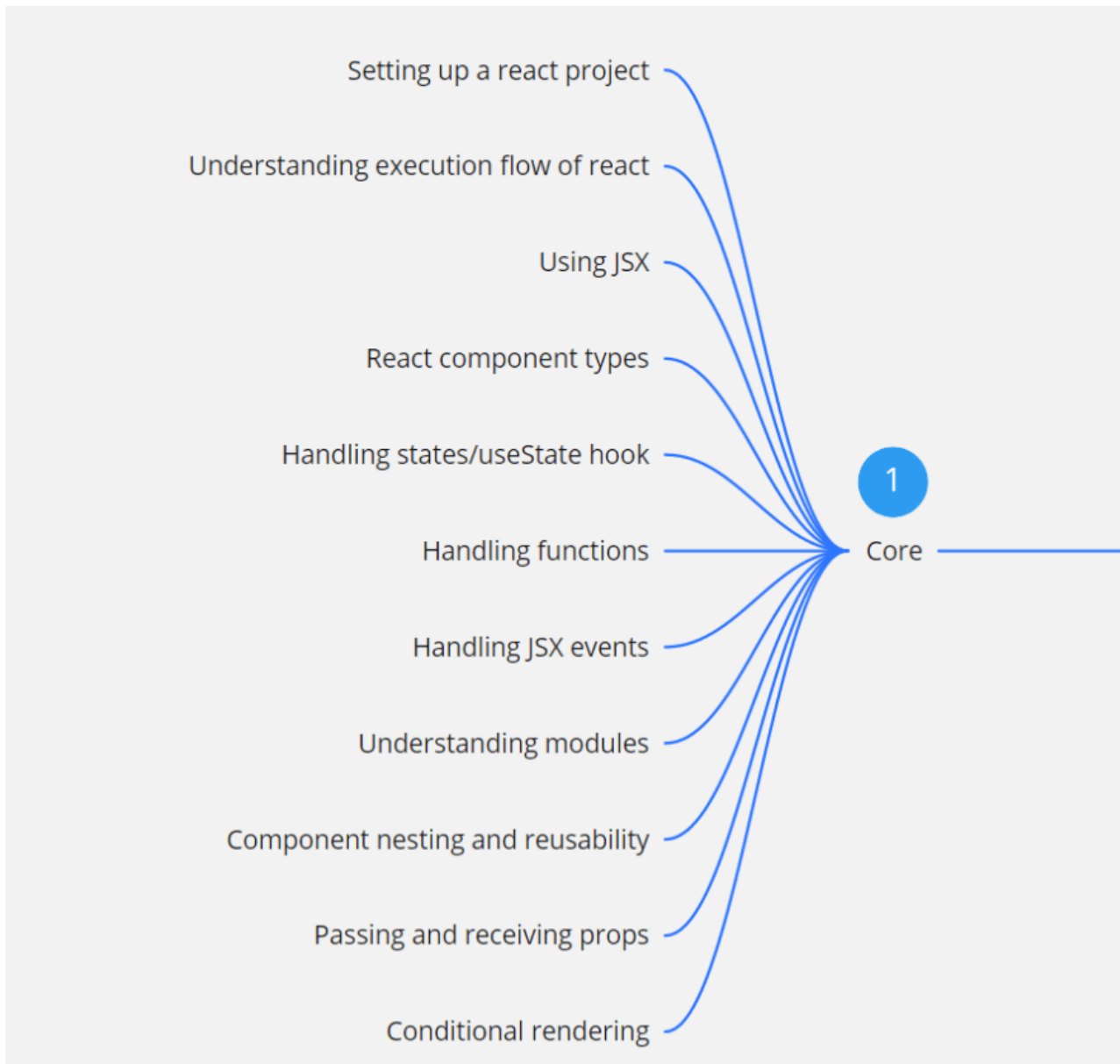


The roadmap is divided into three modules. Core, Basics and Styling the UI. I've added the numbers against each module for better understanding. I prefer not to skip any module as each module is mandatory for understanding React. I've designed this according to my understanding for easy learning, this differs from person to person.

So let's begin! 🚀

## 1. Core

---



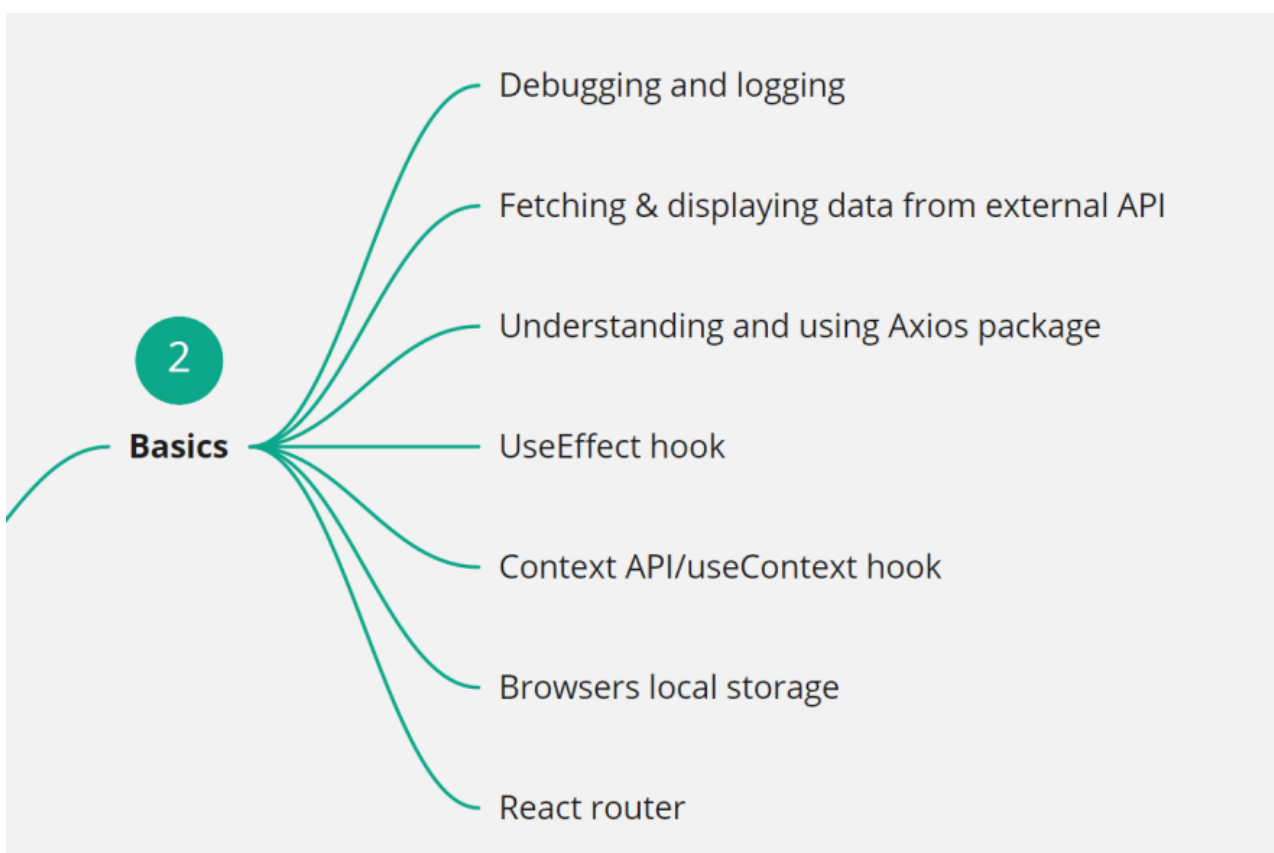
- **Setting up react project** - You can set up a react project by executing some commands, on completing this it gives a React app boilerplate, you can then run you react app to see the initial UI of React. By completing this step you should be familiar with the concept of creating a new react project locally using CLI(Command-line Interface).
- **Execution flow of React** - It is important for a React developer to understand the code flow of a react app. Also understanding the project structure. By completing this step you should be familiar with React app folder structure, Also how react app runs in the browser, what is virtual DOM and how DOM is manipulated in React.

- **Using JSX** - JSX looks like HTML, It is based on XML, JSX transformed into HTML tags during runtime. By learning JSX concept you should be having a good understanding of JSX, Similarities between HTML and JSX and Advantages of JSX over HTML element tags.
- **Component types** - It is important to have knowledge of Component, their types and component life-cycle. By learning the Component concept you should have a good understanding of different component types and which to use when.
- **Handling states/useState hook** - State is an object within which we can store, read and update data in a component. In the functional component we can use 'useState' hook to manage state locally. By learning the React state you should have a good understanding of how a state works, using the useState hook, how to show state value in JSX and how to update the state.
- **Handling functions** - Function in react is same as javascript functions, we can create our own functions to perform specific tasks. By learning about functions you should be able to create functions in react component and calling the function.
- **Handling JSX events** - JSX events allows us to handle events which are react's element(JSX) specific. By learning about JSX events you should have a good understanding of different JSX events(such as onClick, onChange, etc), and their use.
- **Modules** - Modules lets you write sharable code so that you can reuse it by importing. By learning about modules you should be able to understand the concept of modules such as importing and exporting modules, public and private properties and function.
- **Components nesting and reusability** - Components are an independent and reusable block of code which returns JSX and can also perform some specific tasks. By learning component reusability you should be able to understand component nesting, also root, parent and child component, and component tree. you should also be having a good knowledge of creating component in such a way that they can be used in multiple places.

- **Props** - Using Props we can pass data from one component to another. By learning Props you must be familiar with passing the props from the parent component to the child component and then receiving and using them in the child component.
- **Conditional rendering** - Conditional rendering lets you render JSX conditionally just like we use if else condition in javascript. By learning about conditional rendering you should be able to display UI conditionally, and understanding the ternary operator used in React JSX.

## 2. Basics

---



- **Debugging and logging** - React dev tools extension helps you monitor react state and components within the browsers window. By learning this step you should be able to debug react app.
- **Fetching & displaying data from external API** - This lets you perform REST API methods using the built-in method of javascript. By learning this step you should be having a good understanding of javascript's built-in fetch function to fetch the data from an endpoint, storing the data and displaying accordingly on the UI.

- **Understanding and using Axios package** - Axios is one of the popular library for making HTTP requests to external API endpoints effectively. By learning axios you should be familiar with fetching async data from an endpoint using axios and should also be familiar with other REST API methods using axios.
- **UseEffect hook** - UseEffect hook lets you decide what to perform after rendering a component, Also affecting the component based on the dependency provided to it. By learning the UseEffect hook you should be familiar with the side effect of the component rendering with dependency.
- **Context API/useContext hook** - Context API is React's built-in functionality to performs state management without using any external library. By learning about Context API you should be having a good understanding of global state and state management, consumer and provider, useContext and UserReducer hook.
- **Browser's local storage** - Local storage lets you store the data locally within the browser's storage. By learning this step you should have a good understanding of browser local storage, reading and writing to and from local storage.
- **React router** - React router DOM is one of the popular external library for navigating your react app, It also lets your UI in sync with the URL. By learning about React router DOM library you should be familiar with navigating and routing your react application and switching the UI as per the route.

### 3. Styling UI

---



- **Grid layout** - CSS Grid Layout concept offers a grid-based layout system for designing the UI on the basis of rows and columns. By learning Grid layout you should be having a good understanding of placing UI elements as a cell in the grid layout.
- **Flexbox** - Flexbox Layout lets you build flexible and responsive UI without using CSS's positioning and float. By learning flexbox you should be familiar with concept of container, and aligning and justifying elements.
- **SCSS** - To use scss you have to install node-sass package using npm. It's a CSS preprocessor. By learning SCSS you should be familiar with importing other scss files, declaring and using variables, nesting and a lot more things.

Thank you for reading! 👍 I hope you understood each and everything. If you have any doubt or you think I missed something let me know in the comments section. 📄