**REACT JS -** [Shavinda W.A.P\_IT20140298](https://medium.com/@pasindu.shavinda?source=post_page-----59abc96fd2d2--------------------------------) May 12, 2022

**What is React JS ?**

React JS is an open source JavaScript library developed by Facebook which is used to develop interactive web and mobile user interfaces. React JS concerned with the components that utilizes the expressiveness of JavaScript with a HTML like template syntax. React is the component based library which is use to develop interactive UI and it s currently was the most popular JavaScript frontend libraries which are strong foundations and a large community foundations. React JS was basically view the MVC (Model-View-Controller). The major aspects of React are the virtual DOM, One way binding and server side rendering. Virtual DOM is the node tree that lists the elements and there attributes and content as objects and there properties.

**Why React JS ?**

The main reason for use React JS was the other frameworks as we can see the diagram also the data receive from varies resources like initial data, real time data, user input data which is pass to the dispatcher. The dispatcher then forward the data to the store from where it ultimately comes to the view. View is the part that the user interact with the application. What ever we see that the browser as a web page is a view itself. Each time new data is added or updated the backend the browser reload the web page and repeats the whole process again. But the major drawback of the traditional data flow was the Document Object Model.

DOM is an object that was created by the browser. Each time the web page is loaded which can dynamically add or remove the data at the backend. But each time any modifications done the new DOM was created in the same page. The repeated creation of the DOM resulting unnecessary memory wastage and decrease the application performance.

Moreover Manipulation of the DOM is very expensive. Therefore there was a new technology which we save from this problem. This was where “React JS” comes to our rescue. With React JS we can divide our entire application into varies independent components. React JS applications still use the same traditional data flow but something changed at the backend. Each time the backend data was added or updated from the backend the React JS use the new technique to deliver them. Instead of reloading the entire page what React does is it was destroy the old view. After it renders the view with updates of new data and places the new view in place of the old one. For the memory wastage due to DOM React introduce the virtual DOM. The virtual DOM works as three simple steps. Starting with the first step whenever any underlying data changes the entire UI was re rendered virtual DOM representation.

**Advantages Of React JS ?**

Diagram

Description automatically generated

1. Application’s performance was increased.

2. Used on client as well as server side.

3. Readability was increased.

4. Easily can use with other frameworks.

**Features Of React JS ?**

Diagram

Description automatically generated

***1. Learning curve***

React has a shallow learning curve and it was suitable for beginners. ES6 syntax is easier to manage specially for smaller to do apps. In React we coding the JavaScript way it gives us the freedom of choose your tool depending upon our need.

***2. Performance***

React has superior rendering speed. As the name suggests React has instant react to change with the minimum delay.

***3. Size***

React is not a framework and the features added according to the users needs. This is the principle behind the light weight applications built on the react.

***4. Debugging***

React uses compile time debugging and detects errors in the early stage. This ensures the errors don’t silently turn up at the run time. Facebook unidirectional data flow allows clean and smooth debugging.

**Key Terminology**

***JSX — JavaScript Extension***

This JSX allows us to include HTML in the same file along with JavaScript. Each component in React generates some HTML which is rendered by the virtual DOM.

***ES6***

The sixth version of JavaScript was standardize by the ECME international in 2015. This ES6 is not completely supported by the modern web browsers.

***ES5***

The fifth JavaScript framework and this widely expected by all modern web browsers. It was based on the 2009 ECME specifications standards. And tools are use to convert ES6 to ES5 during run time.

***Webpack***

This is a module bundle that generates the build file with all the dependencies required.

***Babel***

This is the tool to convert the ES6 to ES5.

**Components**

We can divide our UI with independent reusable pieces. There are two types of components. Those are class components and functional components.

***Class Components***

The stateful/container component class is a standard ES6 class that extends the React library’s component class. It’s termed a stateful component because it manages how the state changes and how the component logic is implemented. Apart from that, they have access to all phases of the React lifecycle function.

Prior to the introduction of React Hooks, the class component was the only way to create a dynamic and reusable component because it provided access to all React features and lifecycle methods.

Text

Description automatically generated

***Functional Components***

JavaScript functions are used to create functional components. Because they simply accepted and returned data to be rendered to the DOM until the introduction of hooks in React 16.8, they were typically known to as stateless or presentational components.

Previously, the only way to access more React capabilities like state and React lifecycle functions was to use the class component. Hooks, on the other hand, allow you to incorporate state and other React capabilities, as well as write your complete UI with functional components.

Text

Description automatically generated

**State in React**

Your component’s private data is called state. States are not shared between components. The “state” of your component, which you can use to render and alter data.

Text

Description automatically generated

**Props in React**

You’ll need the components to interact with each other in real-world programming. States are private to a component, but data must be passed between them. This is where Props come into play. It’s worth noting that props are read-only.

Diagram

Description automatically generated

**List and Keys in React**

In a React component, we use list to render a list of elements. The task of listing users, TODO tasks, and other objects is fairly prevalent. To iterate over the list and render the results, we utilize the map() function. To tell React to re-render, keys help identify which item from the list has changed. If you neglect to mention the keys for a list, ReactJS will notify you.

Graphical user interface

Description automatically generated

**Conclusion**  
React JS has arrived just in time to assist developers in quickly creating highly engaging online apps and user interfaces. It enables you to deconstruct the components and create a single-page application with less coding.