**React -** [Ravindu Wijerathne](https://medium.com/@ravinduwijerathne629?source=post_page-----410042dbf29b--------------------------------) May 15, 2022

A JavaScript library for building user interfaces. In React we have a whole bunch of different components represented each as if they were an HTML element. So it is effectively like we are creating our own custom HTML elements defining the style and functionality of each of them. And then we can arrange them like HTML elements inside an overall structure keeping the code for our web app incredibly clean and reusable.

React actually combines a small amount of each of these types of files (HTML, CSS, JS) into a single component. So that each component has its own styling and functionality that it controls. And you can create components to each perform a different function and have a different appearance.

Old days when we had Facebook and we have to refresh the page in order to see the updates. But these days it’s all handled through notifications and we have each component listening for changes in the server and updating itself and its own appearance and data and being able to talk to the server independently. So imagine if every single part of the Facebook website was its own component. So newsfeed is its own component, and each of the posts is its own component. So if there is a new comment on a post then that post will be able to update itself without affecting any of the rest of the website.

React is able to re-render these changes really efficiently. And it does this by comparing changes. So whenever a change happens for example new piece of data comes in or the user does something then it’s going to compare the new version of the DOM tree that you want to be rendered with the old version that is already showing up and it will do this kind of spot the difference thing.

React works by creating JSX files. JSX files are files where we have got HTML right in the body of a JavaScript file. Behind the scene, our HTML is picked up by a compiler and it gets converted or compiled down to actual JavaScript. And the compiler comes from React module. Inside the React module, there is something called Babel. And Babel is a JavaScript compiler so its able to take next-generation JavaScript like es6, es7 and compile it down to a version of JavaScript that every browser can understand. And this includes compiling JSX down to plain old JavaScript.

**Create first react app**

**npx create-react-app my-app**

**start the development server**

**npm start**

**React props**

<input **id**=”fName” **placeholder**=”enter first name” **value**=”xxx” >

We use id, name, placeholder, value like attributes from HTML elements to send custom pieces of data to that particular element.

If you think about your react component almost as a custom HTML element then we can define these attributes. In the react component world those attributes are called properties (props).

**Pass props**

A screenshot of a computer

Description automatically generated with medium confidence

**Apply received props**

Text

Description automatically generated

**Class components and Functional components**

In React we can build components in two different ways. They are class components and functional components. A functional component is just a plain JavaScript function that accepts props as an argument and returns a React element. A class component requires you to extend from React. Component and create a render function which returns a React element. There is no render method used in functional components.

**Example for Functional component**

Text

Description automatically generated

**Example for Class component**

Text

Description automatically generated

**React Hooks**

Hooks are a new addition in React 16.8. They let you use state and other React features without writing a class. Functions that allow us to hook into the state of our app and read or modify it. First, we have to **import { useState } from “react”**. You must use a hook inside a functional component.

A screenshot of a computer

Description automatically generated