# Module\_47\_(Conceptual)\_How\_React\_Work

#### **47-1 Module Introduction Create React App under the hood**

* **NPX**: The npx stands for **Node Package Execute,** and it comes with the **npm**, when you installed npm above 5.2.0 version then automatically npx will installed. It is an npm package runner that can execute any package that you want from the npm registry without even installing that package (From [GeeksForGeeks](https://www.geeksforgeeks.org/what-are-the-differences-between-npm-and-npx/)).
* **npm** is a package manager for the JavaScript programming language maintained by npm, Inc. npm is the default package manager for the JavaScript runtime environment Node.js. It consists of a command line client, also called npm, and an online database of public and paid-for private packages, called the npm registry.
* **Create-react-app -** Create a complete development environment for us. It uses webpack, babel, etc behind the scene.

#### **47-2 What is JSX, React Component, babel transpiler**

* What we write in JSX as HTML is not actual HTML. JSX converts it into JavaScript using **Babel,** and finally, React converts that to pure HTML for rendering in Browser.
* **Babel** is a JavaScript compiler that includes the ability to compile JSX into regular JavaScript. You can view babel as an intermediate step between your code and "executable" code. React also uses ES6, which is not supported by most of the browsers. Babel converts the ES6 code to a code which is compatible with the browsers.

#### **47-3 Components and how props works, unidirectional data flow**

* **Props** in React are **uni-directional** (Only can pass from parent -> child -> grandchild, but not child -> parent) and **Read Only** (data passed into props can not be changed).

#### **47-4 (advanced) How states works, asynchronous, console state**

* **useState**() - Takes a parameter which is the default state and returns it first (we get it as **getter),** and 2nd one is **setter** (a function that can be used to change the default state and update the getter value). i.e  
  ***const [steps, setSteps] = useState(0);*** // Here **‘0’** is the default value (state) of **steps**. If we change the value (state) using **setSteps** (one kind of function), will update the latest value (state) to **steps**.  
  **useState** works **asynchronously** and renders the **UI** after every call.  
  The **getter** and **setter** is not the formal name.
* **useEffect()** -> Takes two-parameter. The first one is an **anonymous function** and the second one is generally an e**mpty array** (it is called **Dependency Injection** or **Dependency List**). If we don’t pass the empty array, **useEffect** call after every change of state. But if we pass the empty array, useEffect calls only one time while loading.

#### **47-5 How React Hook works, send state via props, props vs state**

* **useState**, **useEffect** is called Hook in React.
* If the **state** is used in any component is called Stateful Component and doesn’t contain the **state** is called **Stateless** / **Presentational** component.
* Difference between **state and props** - Interview Important - Main difference is: **props** is **Read-only,** but the **state** can be changed. *Google it and read one or two articles on this topic.*

#### **47-6 What is React, when to use it, react vs angular vs vue**

* React is a component-based JavaScript Library used for front-end development.
* Open the React [official site](https://reactjs.org/) and read the documentation top to bottom repeatedly, whether understood or not. The instructor highly recommends it. Also, read this article [Thinking in React](https://reactjs.org/docs/thinking-in-react.html#gatsby-focus-wrapper)
* The reason for learning *React is that it is easy to learn, fast, and has a massive job market* compared to **angular** or **vue**.

#### **47-7 How react works, Render, virtual dom, diff algorithm, fiber**

* When we made a react app, React creates a **virtual dom** internally by combining all the components and send it to the browser. The browser renders it and displays it. If the user does something in UI and needs to update the **Browser dom**, React compares only the changed part with its old **virtual dom** and updates it significantly faster-using **diff algorithm** and sends the browser only the changed part for the update. The browser immediately updates it in **browser dom** and displays it.  
  To know more, Google ‘*How React Work’ and read some articles and watch videos*. - Must do it.

#### **47-8 React component lifecycle and rerender**

* React component has a lifecycle. It has three phases. **Mounting** (componentDidMount -> **Updating** (componentDidUpdate)-> **Unmounting** (componentWillUnmount).
* Google for **‘React component lifecycle’** for study future. - Highly Recommended

#### **47-9 Module Summary, recap, Properties vs attributes**

* Recapped all the topics.
* *Reserve a day, come back to this module, do more research for every topic of this module for better understanding* ***React****.* - Recommended
* Google for ‘*properties vs. attributes in react*’ and study it.