Module : 3 ReactJs

1) **What is React Js?**

-> React is a free and open-source front-end JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and companies.

React is a JavaScript library for building user interfaces. React is used to build single-page applications. React allows us to create reusable UI components.

2) **What is NPM in React Js?**

-> NPM is short for node package manager, an online directory that contains the various already registered open-source packages. NPM modules consume the various functions as a third-party package when installed into an app using the NPM command npm install .

3) **What Is the Role Of Node Js in React Js?**

-> Node. js enables the creation of scalable and quick back-end RESTful APIs. On the other hand, React is a front-end library that creates interactive user interfaces. With both tools, you can quickly build complex and scalable web apps.

4) **What is CLI Command in React Js?**

-> CLI : Command Line Interface

ReactCLI is a ReactJS component that provides a simple way to present a command line interface to your user in your web app .

5) **What are Components in React Js?**

-> Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML. Components come in two types, Class components and Function components, In this tutorial we will concentrate on Function components.

6) **What are Header and Content Components in React Js?**

-> A header component defines a reusable custom header that can be sent in an API request or returned in an API response .
Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and return HTML.
Components come in two types, Class components and Function components, In this tutorial we will concentrate on Function components.

7) **How to install React Js on windows, linux Operating System?**

-> **Step 1**:

Install Node.js installer for windows. Click on this link. Here install the LTS version (the one present on the left). Once downloaded, open NodeJS without disturbing other settings, click on the **Next** button until it's completely installed.

Step 2:

Open command prompt to check whether it is completely installed or not type the command →

Step 3: Now in the terminal run the below command:

```
npm install -g create-react-app
```

It will globally install react apps for you. To check everything went well run the command

```
create-react-app --version
```

Step 4: Now Create a new folder where you want to make your react app using the below command:

```
mkdir newfolder
```

Move inside the same folder using the below command:

```
cd newfolder (your folder name)
```

Step 5: Now inside this folder run the command →

```
create-react-app reactfirst YOUR_APP_NAME
```

Step 6: Now open the IDE of your choice for eg. Visual studio code and open the folder where you have installed the react app newolder (in the above example) inside the folder you will see your app's name reactapp (In our example). Use the terminal and move inside your app name folder.Use command cd reactapp (your app name).

Step 7: To start your app run the below command :

```
npm start
```

8) How to install NPM and How to Check version of NPM?

-> **To see all the installed packages locally or globally, use these commands:**

1. npm list for local packages or npm list -g for globally installed packages.
2. npm list --depth=0.
3. npm list | sls <package name>
4. node -v.

9) How to check the version of React Js?

-> **You can use the below step to identify the "react" and "react-dom".**

1. Open DeveloperTool in your browser.
2. Go to Source Tab.
3. Check your appName .js file.
4. Search for "react" or "react-dom" You will find something like below. That will be the version your react-app is using.

10) How to change the components of React Js?

-> **You can convert a function component like Clock to a class in five steps:**

1. Create an ES6 class, with the same name, that extends React.Component
2. Add a single empty method to it called render() .
3. Move the body of the function into the render() method.
4. Replace props with this.props in the render() body.

11) How to create a List View in React Js?

-> import React from 'react'

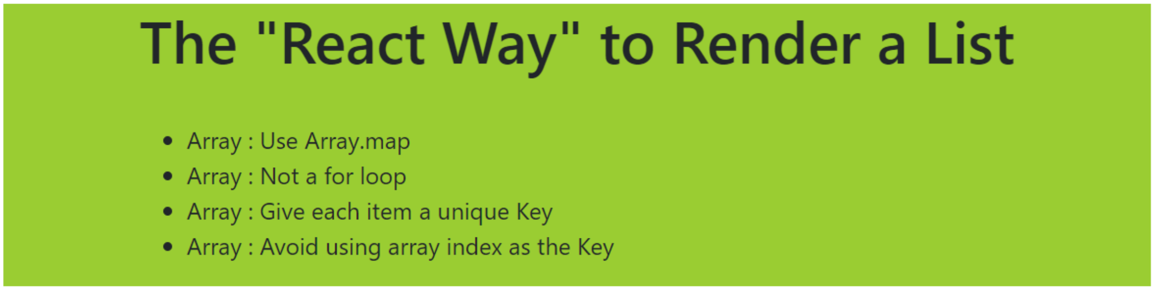
```
import style from "./style.css";

function ListComp(props) {
   return (
      <li>Array : {props.array}</li>
   )
}
export default function List1() {
   const array = ["Use Array.map", "Not a for loop", "Give each item a unique Key",
"Avoid using array index as the Key"];
```

```
    return (
        <>
            <center>
                <form className={`head ${style.head}`}>
                    <table>
                        <div>
                            <h1 className={`heading ${style.heading}`}>The "React Way" to Render a List</h1>
                            <br></br>
                            <ul className={`list ${style.list}`}>
                                {
                                    array.map((c) =>
                                        <ListComp array={c} />
                                    )
                                }
                            </ul>
                        </div>
                    </table>
                </form>



            </center>
        </>
    )
}
```

Style.css :

```
          .heading{
    text-align: left;
}
.head{
    background-color: yellowgreen;
    margin: 50px 250px 50px 250px;
}
```

Output :



The "React Way" to Render a List

- Array : Use Array.map
- Array : Not a for loop
- Array : Give each item a unique Key
- Array : Avoid using array index as the Key

12) Create increment decrement state change by button click?

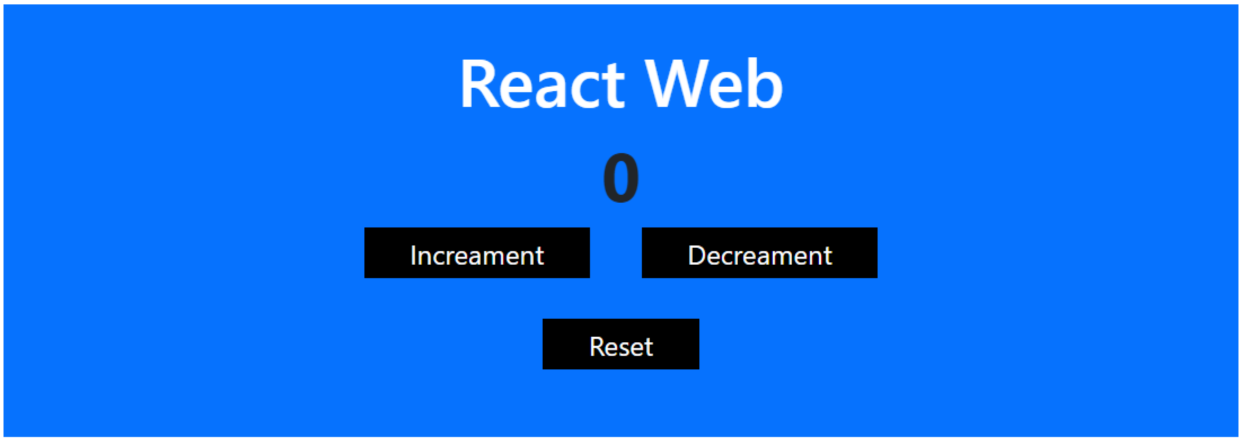
```
->     import React, { Component } from 'react'
import style from "./style.css";

export default class StatMng extends Component {
    constructor(){
        super()
        this.state={
            count : 0
        }
    }
    Handlestate=()=>{
        this.setState({count: this.state.count +1})
    }
    Decrement=()=>{
        this.setState({count: this.state.count -1})
    }
  render() {
    return (
        <>
         <center>
            <div className={`form ${style.form}`}>
            <h1 className={`head1 ${style.head1}`}>React Web</h1>
            <h1 className={`head2 ${style.head2}`}>{this.state.count}</h1>
            <button onClick={this.Handlestate} className={`btn1
${style.btn1}`}>Increament</button>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
            <button onClick={this.Decrement} className={`btn2
${style.btn2}`}>Decreament</button><br/><br/>
            <button className={`btn3 ${style.btn3}`}>Reset</button>
            </div>
            </center>
        </>

    )
  }
}
```

Style.css :

```
       .form{
  background-color: rgb(6, 114, 254);
  padding-bottom: 40px;
  padding-top: 20px;
  margin: 50px 250px 50px 300px ;
}
.head1{
  color: white;
}
.head2{
  font-weight: bold;
}
.btn1{
  background-color: black;
  color: white;
```

```
    padding-left: 25px;
    padding-right: 25px;
}
.btn2{
    background-color: black;
    color: white;
    padding-left: 25px;
    padding-right: 25px;
}
.btn3{
    background-color: black;
    color: white;
    padding-left: 25px;
    padding-right: 25px;
}
```

Output :


React Web
0
Increament
Decreament
Reset