**Lecture-6: React components and props**

**Process to create and how into browser:**

React element => react.createElement() => react object creating using createElement() method => reactDOM.render() call passing element

**Why immutable:**

Suppose we a h2 tag inside ‘masud rana’. We change the name each time using interval of 1min. h2 is immutable, we can’t change. React each time h2 create. And compare before h2 and recent h2. And show the change only. It means element will not be changed. Each time it will be create and react will compare the recent and previous element and change the content of the element.

Note: one react component only return single react element. Mainly what we return from a react component is a react element.

**React component return react element**

**One react component returns one single react element**

function App() {

  return (

    <div className="App">

      <header className="App-header">

        <h2>masud rana inside div and header is mainly a react element</h2>

      </header>

    </div>

  );

}

Here **App is component** and **div inside return is an element**. Now we can reuse this component

We can use this component like App() function all or <App /> like this. We can pass parameter to this function and the app function will receive these parameter as object.

function App({ local }) {

  return (

    <div className="App">

      <header className="App-header">

        <h2>masud rana inside div and header is mainly a react element</h2>

      </header>

    </div>

  );

}

<App local="bd-880" />;

**Above is functional components**

We can create component using class component

**Below is class component:**

in generally react will not know class component. To inform this class into react we need to extend React.Component supper class. Now all the properties of this supper class will be available into the Clock class

// class component

class Clock extends React.Component{

  print() {

    return (

      <h1 className="heading">

        <span> Hellow {new Date().toLocaleDateString()}</span>

      </h1>

    );

  }

}

Now we can use <Clock /> like this. When react get <Clock /> then it search this class then inside class it search a render function and render the code inside render function. So above print directly will not execute.

Render is defined in React.Component class that we inherited

Search clock name then search it extends react.component then search render method.

When we pass parameter to this class. **React.Component** class have a props object. This class push the passed parameter into props object. Now Clock class have props object with the passed data. Now inside clock class we can use this.passData . since we can use variable inside class using this keyword.

// class component

class Clock extends React.Component{

  render() {

    return (

      <h1 className="heading">

        <span> Hellow {new Date().toLocaleDateString(this.props.local)}</span>

      </h1>

    );

  }

}

<Clock local='bn-BD' />

Class component is statefull. Means react can change the content itself.

We can pass a component as parameter of another component. we can catch it inside the component using **this.props.children**

// class component

class Clock extends React.Component {

  render() {

    return (

      <h1 className="heading">

// here I have catched the children of this component

        <span>{this.props.children}</span>

        <span> Hellow {new Date().toLocaleDateString(this.props.local)}</span>

      </h1>

    );

  }

}

<Clock local="bn-BD">this a string inside a component a children</Clock>;

**Note: Never change the props inside component.** Because when it will be change the component will be re rendered again**.** And will change the content of the component

// class component

class Clock extends React.Component {

  render() {

// Never do this

  this.props.local='bn-EN';

// this will create infinite loop

    return (

      <h1 className="heading">

      </h1>

    );

  }

}

<Clock local="bn-BD">this a string </Clock>;

Props change means react will change the content mainly re call the render method.

**Note:** props always will be change outside of the component

React class component or functional component is encapsulated means each component is private from each others.

**Presentational component or dump component:** this type of component have not any state or dynamic to change of this content. It just static presentation.