**Lecture-5: React JSX element, DOM and browser element and DOM**

**Browser DOM**

Browser mainly create DOM tree using the html element in browser as Node. We can create html element using DOM manipulation. Const ele = document.createElement(“a”);

Ele.textContent=”amar sonar bangla”;

Document.body.appendChild(ele);

**React DOM**

React have same way to create react element. React.createElement(‘type’, ‘properties’, ‘content’);

**Type** means tag type. **Properties** means mainly attribute or properties. **Content** means content of the created element.

Console.log(element); but this is valid javascript object. Because \_prototype exists. React is janila javascript object so easy to work. It has many property. React doest not work with outside of react element. Browser use html markup language but react can’t use html.

In react project babel is included.

React have JSX element. It ‘s similar to the html element but they are not equal. Because it is JavaScript XML. Mainly this xml code we can write inside our javascript file. we can create Virtual DOM or JSX dom. We can write any JavaScript code into this JSX element. We can use expression. JSX itself a valid javascript expression. We can return any JSX as return

When we write JSX element, mainly babel transpile this code into react DOM and react true.

// use valid javascript expression inside react element because react element valid javascript code. it has the similarity fo the

const ele = <h2>amar sonar bangla {summation} {new Date.date()} {</h2>;

**multiline JSX or wrap JSX element(using first braces):** we can return JSX block of code using first braces (). Since JSX mainly valid Javascript object so we can return jsx element from javascript code.

function getGreeting(user){

  if(user){

    return (

<p>hello world!</p>

<h3> amar sonar bangla ami tomay valobashi </h3>

)

  }else return <p>you failled</p>

}

function getGreeting(user){

  if(user){

    return <p>hello world!</p>

  }else return <p>you failled</p>

}

// interpolation in react element

const ele = <h2>amar sonar bangla { getGreeting('masud')} </h2>;

**pass attribute to the JSX element and how they looks:**

**JSX element and pass parameter to the JSX:**

const index = 0;

const element = (

  <h2 className="heading" tabIndex={index}>hello masud rana</h2>

  <p className="description">hello shamoli rahman soton</p>

)

**behind the scene:**

element = {

  type: 'h1',

  props: {

    className: 'heading',

    tabIndex: 0,

    children: 'hello masud rana',

  }

}

Always bear in mind: we write JSX and this is javascript object. In javascript property is camel case. Like className, tabIndex

**Nexted JSX and Equivalent JavaScript Object:**

// JSX code and below equivalent REACT DOM Tree

const index = 0;

const element = (

  <h2 className="heading" tabIndex={index}>hello masud rana

  <p>i am inside h2 tag</p>

  <small>this is small</small>

  </h2>

)

// behind the scene in REACT:

element = {

  type: 'h2',

  props: {

    className: 'heading',

    tabIndex: 0,

    children: [ //since children more than one element, so we need array of children objects

     {

      type: 'p',

      props: null,

     },

    {

      type: 'small',

      props: null,

    }

    ],

  }

}

**React protect malicious from code:**

JSX is valid javascript object. Suppose we called an api inside JSX. If there exits viruses then React ‘escape’ these api called as string then it will not execute. But in javascript it should be execute. But react protect it.

<h2>this is api call inside JSX object {api}</h2>

**React is platform independent.** React code we can run in android. Because react element showed in android with help of the REACT NATIVE. In browser we use REACT DOM.

**React element is immutable.** We can’t change after creating. If we want to update mainly react create new element but in UI it just show the change only.

**React element** is the small building block of react. Not react component. React do only create react element. And to show it into browser we need to use REACT DOM.

**Show react element into browser using REACT DOM.**

ReactDOM.render(element, document.getElementById(‘root’));

**Element is the core building block of react**