Source : leela web Dev

<https://www.youtube.com/watch?v=LMFAaNWmKII&list=PL_euSNU_eLbfuahgrm63xLCWfpl2MaTAr>

**Class -1**

React js is a javascript library for building the user interface

React is small and its not complete for solutions, you need to depend on other libraries to solve the solution.

ReactDOM for DOM

Axios for Http requests

Created by facebook in 2011 for their website facebook, in 2013 its open source and then very popular in very large company, in 2015 react native is relased its mobile applicate for both android and IOS

Why React.?

Flexibility

Big Corporate Support & Investment

Community Support

Performance

Testability

Where we can use react .?

Used to develop webapps

Static website using Gatsby

Mobile app use by react native

Desktop app used by Electron

Server side rendering apps using by Next.js

Vitrual Reality apps also developed using React VR

React Renderes

For webapps we use react-dom to convert the components to html

For mobile apps we use react-native to convert into native code

React Performance

We know that js is fast bec of single thread but updating the dom is slow and expansive.

React use virtual DOM that update the DOM in efficient way.

Whenever the state changes in component the react take care updating of DOM, updating happing in memory very fast

React Components

Components are just like vanilla js functions.function what it does, it takes input and gives us some out put, we can reuse this function.

In the same scenario components also takes inputs as props and UI as output, they are reusable.

React Altranatives

Angular

Vue

Angular and Vue separates the HTML and the logic

For looping we need to learn separate syntax

Angular 🡪 <div \*ngFor=’statement’>

Vue 🡪 <div v-for=’statement’>

React uses the JSX syntax instead of having the html template

That means that html written javascript itself

So therefore there is no need to implement separate concept for looping,instead we can use directly the javascript map or for looping dom elements.also we can use javascript if conditions

**Class -2 : How React code looks and executes**

**Index.html Code**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <div id="app">

    </div>

<!--Download / copy  below code

   https://babeljs.io/setup#installation

   <script type="text/babel"></script>

-->

<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

<!--Download / copy  below code

    https://legacy.reactjs.org/docs/cdn-links.html

-->

<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>

<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

<script type="text/babel" src="main.js"></script>

</body>

</html>

**Main.js Code :**

/\*

    Code :

    const element=<h2>Hello Asif</h2>

    ReactDOM.render(element,document.getElementById("app"));

    Output:

    Error:

    Uncaught SyntaxError: Unexpected token '<' (

\*/

/\*below code for creating one html element into javascript

    Code :

    const element1=React.createElement('div',{className:'asif'},'Hello World',React.createElement('h2',{className:'h2class'},'Shaik Asif'));

    ReactDOM.render(element,document.getElementById("app"));

    Output:

    No Error output will come clear

    Conclusion :

    this is to lenghty to create single html element so to overcome this

    JSX introduced

\*/

/\* below is jsx code \*/

const element2=(

    <div>

        Hello World

        <h2>Shaik Asif</h2>

    </div>

);

ReactDOM.render(element2,document.getElementById("app"));

/\*

    output:

    Uncaught SyntaxError: Unexpected token '<' (

    here babel compiler come to solve the error,

    it will conver all html which is written in jsx code into the

    React.createElement

        copy  below code from below link and put in index.html

         https://babeljs.io/setup#installation

        <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

         <script type="text/babel" src="main.js"></script>

\*/

**Class :3 : Why We need used to JSX**

JSX stands for JavaScript XML.

JSX allows us to write HTML in React.

JSX makes it easier to write and add HTML in React.

JSX allows us to write HTML elements in JavaScript and place them in the DOM without any createElement()  and/or appendChild() methods.

You are not required to use JSX, but JSX makes it easier to write React applications.

Using jsx

const myElement = <h1>I Love JSX!</h1>;

const root = ReactDOM.createRoot(document.getElementById('root'));

root.r ender(myElement);

Without Jsx

const myElement = React.createElement('h1', {}, 'I do not use JSX!');

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(myElement);

We can embaded the expression using jsx

const name="Shaik Asif"

const myElement=(

    <div>

        <div>hello {name} <div>

    </div>

);

const name="Shaik Asif"

const myElement=(

    <div>

        <div>hello {name} <div>

// embedded outside variable value

        <h2> {2+2} </h2>

// embedded the expression

    </div>

);

const channelName="Asif"

function getChennalName(){

    return <div>{channelName}</div>

}

const myElement=(

    <div>

        <div>hello {name} <div>

               // embadded the variable name

        <h2> {2+2} </h2>

            // embadded the expression

        <h3>{getChennalName} </h3>

            // embadded the function

    </div>

);

Using if condition

const channelName="Asif"

function getChennalName(){

     if(channelName==='Asif')

     return <div>{channelName}</div>

     else <div>Else part</div>

}

const myElement=(

    <div>

        <div>hello {name} <div>

               // embadded the variable name

        <h2> {2+2} </h2>

            // embadded the expression

        <h3>{getChennalName} </h3>

            // embadded the function

    </div>

);

Jsx uses camelCase

Eg: tabindex 🡪 in html

tabIndex 🡪in jsx

class🡪 className

REACT UNDERSTODABLE CODE

BABEL

JSX CODE

Class :4 : Understanding Virtual DOM

React Virtual Dom will work when what component will update that only update remaining not update

Virtualdom.js

// using old version javascript

function render(){

const element=`<div>

     <div> Hello world </div>

     <div> <input type='text'/>

     ${new Date().toLocaleTimeString()}

     </div>

</div>`;

document.getElementById('app1').innerHTML=element;

// using react virtual dom

/\*

by seeing this react virtual dom will work which component is updating that

componet only will update

\*/

const element2=React.createElement('div',null,

     React.createElement(

    'div',null,'Hello this is React.createElement')

    ,React.createElement('div',null,

    React.createElement('input',{type:'text'}),

    React.createElement('h2',{},new Date().toLocaleTimeString())

    )

    );

ReactDOM.render(element2,document.getElementById('app2'));

     }

setInterval(render,1000);

index.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <div id="app1">

    </div>

    <br/>

    <br/>

    <br/>

    <br/>

    <div id="app2">

    </div>

<!--Download / copy  below code

   https://babeljs.io/setup#installation

   <script type="text/babel"></script>

-->

<script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>

<!--Download / copy  below code

    https://legacy.reactjs.org/docs/cdn-links.html

-->

<script crossorigin src="https://unpkg.com/react@18/umd/react.development.js"></script>

<script crossorigin src="https://unpkg.com/react-dom@18/umd/react-dom.development.js"></script>

<script type="text/babel" src="virtualDom.js"></script>

</body>

</html>

