**Jsx attributes**

**Create class .bgClass in css file**

.bgClass

{

  background-color:yellow;

  color: red;

  font-size:x-large;

}

**In jxsDemo.js file**

import React from 'react';

const e1 = <h1 className="bgClass"> {10+20} </h1>

export default e1;

**in index.js display it**

**ReactDOM.render(e1, document.getElementById(“root”));**

**By using javascript(Without using jsx)**

var reactNodeLi = React.createElement('li', {id:'li1'}, 'one');

ReactDOM.render(reactNodeLi, document.getElementById("root"));

// create React element <li>'s

var rElmLi1 = React.createElement('li', {id:'li1'}, 'one');

var rElmLi2 = React.createElement('li', {id:'li2'}, 'two');

var rElmLi3 = React.createElement('li', {id:'li3'}, 'three');

//create <ul> React element and add child React <li> elements to it

var reactElementUl = React.createElement('ul', {className:'myList'}, rElmLi1,rElmLi2,rElmLi3);

ReactDOM.render(reactElementUl, document.getElementById("root"));

var reactElementUl = React.createElement(

  'ul', {

      className: 'myList'

  },

      React.createElement('li', {id: 'li1'},'one'),

      React.createElement('li', {id: 'li2'},'two'),

      React.createElement('li', {id: 'li3'},'three')

);

**Props Demo**

ParentComponent

import {React , Component} from 'react';

import ChildComponent from './ChildComponent';

class ParentComponent extends Component {

  render() {

    return (

      <h1>

        I'm the parent component.

        <ChildComponent />

        <ChildComponent />

        <ChildComponent />

      </h1>

    );

  }

}

export default ParentComponent;

CHildCOmponent

const ChildComponent = () => {

    return <p>I'm the 1st child!</p>;

  };

  export default ChildComponent;

Comp with props

const ChildComponent = (props) => {

    return <p>I'm the 1st child!

        My text is {props.text}

    </p>;

  };

import {React , Component} from 'react';

import ChildComponent from './ChildComponent';

class ParentComponent extends Component {

  render() {

    return (

      <h1>

        I'm the parent component.

        <ChildComponent text="I am no 1" />

        <ChildComponent text="I am no 2"/>

        <ChildComponent text="I am no 3"/>

      </h1>

    );

  }

}

export default ParentComponent;

**State Demo**

**State outside Constructor**

**Student.js Component**

import React, { Component } from 'react'

class Student extends React.Component

{

  state = {

  name :"Ajay",

  address :"Delhi"

}

  render()

  {

    return (<h3> Hello {this.state.name +' ' +  this.state.address} </h3>);

  }

}

export default Student;

in index.js

import this component

Call it

ReactDOM.render(

  <div>

  <App/>

    {/\* <SecondComp/>

    <StateDemoComp/> \*/}

    <EventComp/>

    <AddNumbers/>

**<Student/>**

    </div>,

  document.getElementById('root')

);

**render()**

**{**

**return (<h3> Hello {this.state.name} , Your Address is {this.state.address} </h3>);**

We can pass address through props

 <Student address="New Delh"/>

In Student Component

class Student extends React.Component

{

  state = {

  name :"Ajay",

  address : this.props.address

}

We have assigned prop to state

**State within Constructor**

import React, { Component } from 'react'

class Student1 extends React.Component

{

**constructor(props)**

**{**

**super(props);**

**this.state = {**

**name :"Ajay",**

**address : this.props.address**

**}**

}

  render()

  {

    return (<h3> Hello {this.state.name} , Your Address is {this.state.address} </h3>);

  }

}

export default Student1;

**Update State**

import React, { Component } from 'react'

class StudentChange extends React.Component

{

  constructor(props)

  {

    super(props);

this.state = {

  name :"Ajay",

  address : this.props.address

  }

}

 handleClick = () =>{

   console.log("Button clicked", this);

 }

 render()

  {

    return (

      <div>

    <h3> Hello {this.state.name} , Your Address is {this.state.address} </h3>

    <button type="button" onClick={this.handleClick}> Click here </button>

    </div>

    );

      }

}

export default StudentChange;

Change name

 handleClick = () =>{

   console.log("Button clicked", this);

   this.setState({name:"Deepak"});

 }

 handleClick = () =>{

   console.log("Button clicked", this);

   this.setState({name:"Deepak", address:"Bombay"});

 }

To retrieve values

handleClick = () =>{

  this.setState((state, props)=> {

    console.log(state.name);

    console.log(props.address);

  });

};

Event Handling Demos

import React from 'react';

import ReactDOM from 'react-dom';

class EventComp extends React.Component {

  shoot() {

    alert("Great Shot!");

  }

  render() {

    return (

      <button onClick={this.shoot}>Take the shot!</button>

    );

  }

}

export default EventComp;

import React, { Component } from 'react';

class App2 extends React.Component {

    constructor(props) {

        super(props);

        this.state = {

            companyName: ''

        };

    }

    changeText(event) {

        this.setState({

            companyName: event.target.value

        });

    }

    render() {

        return (

            <div>

                <h2>Simple Event Example</h2>

                <label htmlFor="name">Enter company name: </label>

                <input type="text" id="companyName" onChange={this.changeText.bind(this)}/>

                <h4>You entered: { this.state.companyName }</h4>

            </div>

        );

    }

}

export default App2;