**App.js**

import React from 'react';

import ReactDOM from react-dom;

const Display = React.createElement('div', nul, 'Hello world')

ReactDOM.render(

Display,

document.getElementById('root')

)

export default Display

**index.js**

import React from 'react';

import ReactDOM from 'react-dom';

import "./Test.css"

const names = ["aamir", "irfan","muneeb" ]

const Display = React.createElement('ul', {

className: "redColor"

},

React.createElement("li", null, names[0]),

React.createElement("li", null, names[1])

)

console.log(Display)

ReactDOM.render(Display, document.getElementById('root'));

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<link rel="shortcut icon" href="%PUBLIC\_URL%/favicon.ico" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<meta name="theme-color" content="#000000" />

<!--

manifest.json provides metadata used when your web app is installed on a

user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/

-->

<link rel="manifest" href="%PUBLIC\_URL%/manifest.json" />

<!--

Notice the use of %PUBLIC\_URL% in the tags above.

It will be replaced with the URL of the `public` folder during the build.

Only files inside the `public` folder can be referenced from the HTML.

Unlike "/favicon.ico" or "favicon.ico", "%PUBLIC\_URL%/favicon.ico" will

work correctly both with client-side routing and a non-root public URL.

Learn how to configure a non-root public URL by running `npm run build`.

-->

<title>React App</title>

</head>

<body>

<noscript>You need to enable JavaScript to run this app.</noscript>

<div id="root"></div>

<!--

This HTML file is a template.

If you open it directly in the browser, you will see an empty page.

You can add webfonts, meta tags, or analytics to this file.

The build step will place the bundled scripts into the <body> tag.

To begin the development, run `npm start` or `yarn start`.

To create a production bundle, use `npm run build` or `yarn build`.

-->

</body>

</html>

**Test.css**

body {

margin: 0;

font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", "Roboto", "Oxygen",

"Ubuntu", "Cantarell", "Fira Sans", "Droid Sans", "Helvetica Neue",

sans-serif;

-webkit-font-smoothing: antialiased;

-moz-osx-font-smoothing: grayscale;

}

code {

font-family: source-code-pro, Menlo, Monaco, Consolas, "Courier New",

monospace;

}

.redColor{

color: red

}

**Index.js When we use array and arrow function**

import React from 'react';

import ReactDOM from 'react-dom';

import "./Test.css"

const names = ["aamir", "irfan","muneeb" ]

const Display = React.createElement('ul', {

className: "redColor"

},

names.map(name => React.createElement("li", null, name))

)

console.log(Display)

ReactDOM.render(Display, document.getElementById('root'));

**Index.js when value/pair will use**

import React from 'react';

import ReactDOM from 'react-dom';

import "./Test.css"

const names = ["aamir", "irfan","muneeb" ]

const Display = React.createElement('ul', {

className: "redColor"

},

names.map(name => React.createElement("h3", {

key: name

},name))

)

console.log(Display)

ReactDOM.render(Display, document.getElementById('root'));

**Index.js** When we use object

import React from 'react';

import ReactDOM from 'react-dom';

import "./Test.css"

const names = ["aamir", "irfan", "osama sir", "muneeb" ]

const Display = <ul>

{names.map(

name => <li key={name}>

{name}

</li>

)}

</ul>

console.log(Display)

ReactDOM.render(Display, document.getElementById('root'));

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz

**app.js**

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

class App extends Component {

render(){

const names = ["Karachi", "Lahore", "Quetta", "Peshawar"]

return(

<ul>

{

names.map((n) =>

<li key={n}>{n}</li>)

}

</ul>

)

}

}

export default App

**App.js** When we use index

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

class App extends Component {

render(){

const names = ["Karachi", "Lahore", "Quetta", "Peshawar"]

return(

<ul>

{

names.map((n, index) =>

<li key={index}>{n}</li>)

}

</ul>

)

}

}

export default App

**app.js** When parent and child both use

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

class Cities extends Component {

render(){

return(

<ul>

{

this.props.anyname.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

}

class App extends Component {

render(){

console.log('working')

const names = ["Karachi", "Lahore", "Quetta", "Peshawar"]

return (

<div>

<Cities anyname={names}/>

</div>

)

}

}

export default App

**app.js** When this approach, const {anyname} = this.props will apply

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

class Cities extends Component {

render(){

const {anyname} = this.props

return(

<ul>

{

anyname.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

}

class App extends Component {

render(){

console.log('working')

const names = ["Karachi", "Lahore", "Quetta", "Peshawar"]

return (

<div>

<Cities anyname={names}/>

</div>

)

}

}

**App.js** = When write two verbs but variable differ from other

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

class ListItems extends Component {

render(){

const {anyname} = this.props

return(

<ul>

{

anyname.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

}

class App extends Component {

render(){

console.log('working')

const names = ["Karachi", "Lahore", "Quetta", "Peshawar"]

const names11 = ["aamir", "Irfan", "Osama", "muneeb"]

return (

<div>

<ListItems anyname={names}/>

<ListItems anyname={names11}/>

</div>

)

}

}

export default App

**App.js** = only working Display

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

function DisplayNames(){

const names11 = ["aamir", "Irfan", "Osama", "muneeb"]

return(

<ul>

{

names11.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

render(){

const {anyname} = this.props

return(

<ul>

{

anyname.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

}

class App extends Component {

render(){

console.log('working')

const names = ["Karachi", "Lahore", "Quetta", "Peshawar"]

return (

<div>

<ListItems anyname={names}/>

<DisplayNames />

</div>

)

}

}

export default App

**App.js**

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

function DisplayNames(props){

return(

<ul>

{

props.name.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

render(){

const {anyname} = this.props

return(

<ul>

{

anyname.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

}

class App extends Component {

render(){

const names = ["Karachi", "Lahore", "Quetta", "Peshawar"]

const names11 = ["aamir", "Irfan", "Osama", "muneeb"]

return (

<div>

<ListItems anyname={names}/>

<DisplayNames name = {names}/>

</div>

)

}

}

export default App

**App.js** : without construction uses

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

function DisplayNames(props){

return(

<ul>

{

props.name.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

state ={

cityNames: ["Karachi", "Lahore", "Quetta", "Peshawar"]

}

render(){

return(

<ul>

{

this.state.cityNames.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

}

class App extends Component {

render(){

return (

<div>

<ListItems />

</div>

)

}

}

export default App

//const names = ["Karachi", "Lahore", "Quetta", "Peshawar"]

//const names11 = ["aamir", "Irfan", "Osama", "muneeb"]

//<DisplayNames name = {names}/>

**App.js**

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

function DisplayNames(props){

return(

<ul>

{

props.name.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

state ={

cityNames: this.props.names

}

render(){

return(

<ul>

{

this.state.cityNames.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

}

class App extends Component {

render(){

return (

<div>

<ListItems names ={["Karachi", "Lahore", "Quetta", "Peshawar"]}/>

</div>

)

}

}

export default App

//const names = ["Karachi", "Lahore", "Quetta", "Peshawar"]

//const names11 = ["aamir", "Irfan", "Osama", "muneeb"]

//<DisplayNames name = {names}/>

**App.js** = Working through construction

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

function DisplayNames(props){

return(

<ul>

{

props.name.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

constructor (props)

{

super(props)

this.state = {

cityNames: this.props.names

}

}

render(){

return(

<ul>

{

this.state.cityNames.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

}

class App extends Component {

render(){

return (

<div>

<ListItems names ={["Karachi", "Lahore", "Quetta", "Peshawar"]}/>

</div>

)

}

}

export default App

//const names = ["Karachi", "Lahore", "Quetta", "Peshawar"]

//const names11 = ["aamir", "Irfan", "Osama", "muneeb"]

//<DisplayNames name = {names}/>

**App.js** Button create

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

function DisplayNames(props){

return(

<ul>

{

props.name.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

render(){

return(

<div>

<ul>

{

this.props.cityNames.map((n, index) =>

<li key={index}>{n}</li>

)

}

</ul>

<button>

Press this button to increase the number

</button>

</div>

)

}

}

class App extends Component {

state = {

cityNames : ["Karachi", "Lahore", "Quetta", "Peshawar"]

}

render(){

return (

<div>

<ListItems cityNames = {this.state.cityNames}/>

</div>

)

}

}

export default App

**App.js =** For result the following:

Karachi

Lahore

Quetta

Peshawar

Press this button to increase the number

**Above code of App.js**

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

function DisplayNames(props){

return(

<ul>

{

props.name.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

render(){

return(

<div>

<ul>

{

this.props.cityNames.map((n, index) =>

<li key={index}>{n}</li>

)

}

</ul>

<button onClick={this.props.plCchangeTheState}>

Press this button to increase the number

</button>

</div>

)

}

}

class App extends Component {

state = {

cityNames : ["Karachi", "Lahore", "Quetta", "Peshawar"]

}

changeTheState = this.setState(

(ps) => (

{

cityNames: ["Irfan", "Aamir", "Muneeb"]

}

)

)

render(){

const { changeTheState } = this.state;

return (

<div>

<ListItems cityNames = {this.state.cityNames} plsChangeTheState={changeTheState}/>

</div>

)

}

}

export default App

**App.js** = That is very important because when we click the button, the value can change

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

function DisplayNames(props){

return(

<ul>

{

props.name.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

render(){

return(

<div>

<ul>

{

this.props.cityNames.map((n, index) =>

<li key={index}>{n}</li>

)

}

</ul>

<button onClick={this.props.plzChangeTheState}>

Press this button to increase the number

</button>

</div>

)

}

}

class App extends Component {

state = {

cityNames : ["Karachi", "Lahore", "Quetta", "Peshawar"],

address: "axiom"

}

changeTheStateNow = () => { this.setState(

{ cityNames: ["asdas", "asdasdsad"]}

)

}

render() {

return (

<div>

<ListItems cityNames = {this.state.cityNames} plzChangeTheState = {this.changeTheStateNow}/>

</div>

)

}

}

export default App

**App.js**

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

import PropTypes from 'prop-types'

function DisplayNames(props){

return(

<ul>

{

props.name.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

static propTypes = {

guruNames: PropTypes.array

}

render(){

return(

<div>

<ul>

{

this.props.guruNames.map((n, index) =>

<li key={index}>{n}</li>

)

}

</ul>

<button onClick={this.props.plzChangeTheState}>

Press this button to increase the number

</button>

</div>

)

}

}

ListItems.propTypes = {

guruNames: PropTypes.array

}

class App extends Component {

state = {

cityNames : ["Karachi", "Lahore", "Quetta", "Peshawar"],

}

changeTheStateNow = () => {

this.setState(

(ps) => {

console.log(ps)

return (

{ cityNames: ["asdas", "asdasdsad"]}

)

}

)

}

render() {

const guruNames = ["irfan", "osama sir", "muneeb"]

return (

<div>

<ListItems guruNames = {guruNames} cityNames = {this.state.cityNames} plzChangeTheState = {this.changeTheStateNow}/>

</div>

)

}

}

export default App

**App2.js**

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

import PropTypes from 'prop-types'

function DisplayNames(props) {

return (

<ul>

{

props.name.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

static propTypes = {

guruNames: PropTypes.oneOfType([PropTypes.string, PropTypes.array])

}

render() {

return (

<div>

<ul>

{

this.props.guruNames.map((n, index) =>

<li key={index}>{n}</li>

)

}

</ul>

<button onClick={this.props.plzChangeTheState}>

Press this button to increase the number

</button>

</div>

)

}

}

ListItems.propTypes = {

guruNames: PropTypes.array

}

class App2 extends Component {

state = {

cityNames: ["Karachi", "Lahore", "Quetta", "Peshawar"],

text: ""

}

render() {

const guruNames = ["irfan", "osama sir", "muneeb"]

return (

<div>

<ListItems guruNames={guruNames} cityNames={this.state.cityNames} plzChangeTheState={this.changeTheStateNow} />

<input type = "text" name = "text" value = {this.state.text} onChange = {(event) => this.setState({[event.target.name]: event.target.value}) }/>

{this.state.text}

</div>

)

}

}

export default App2

**index.js**

import React from 'react';

import ReactDOM from 'react-dom';

import "./Test.css"

import App from './App'

import App2 from "./App2"

ReactDOM.render(<App2/>, document.getElementById('root'));

**App2.js** One approach to solve the problem

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

import PropTypes from 'prop-types'

function DisplayNames(props) {

return (

<ul>

{

props.name.map((n, index) => (

<li key={index}>{n}</li>

))

}

</ul>

)

}

class ListItems extends Component {

static propTypes = {

guruNames: PropTypes.oneOfType([PropTypes.string, PropTypes.array])

}

render() {

return (

<div>

<ul>

{

this.props.guruNames.map((n, index) =>

<li key={index}>{n}</li>

)

}

</ul>

{/\* When you want to give comment for any HTML matter by ctrl + /\*---\*/ }

{/\* <button onClick={this.props.plzChangeTheState}>

Press this button to increase the number

</button> \*/}

</div>

)

}

}

ListItems.propTypes = {

guruNames: PropTypes.array

}

class App2 extends Component {

state = {

cityNames: ["Karachi", "Lahore", "Quetta", "Peshawar"],

text: ""

}

render() {

const guruNames = ["irfan", "osama sir", "muneeb"]

return (

<div>

<ListItems guruNames={guruNames} cityNames={this.state.cityNames} plzChangeTheState={this.changeTheStateNow} />

<input type = "text" name = "text" value = {this.state.text} onChange = {(event) => this.setState({[event.target.name]: event.target.value}) }/>

{this.state.text}

</div>

)

}

}

export default App2

**App.s** =Another approach to solve the problem

import React, { Component } from 'react';

class App2 extends Component{

state={text: ""}

whenChange = (event) => {

this.setState ({text:event.target.value})

}

render(){

return(

<div>

<input type = "text" onChange={this.whenChange}/>

<p>{this.state.text}</p>

</div>

)

}

}

export default App2

**App.js** = Output are coming out of box ,Why?

import React, { Component } from 'react';

class App2 extends Component{

state={text: "zxxxz"}

whenChange = (event) => {

this.setState ({text:event.target.value})

}

render(){

return(

<div>

<input type = "text" onChange={this.whenChange}/>

<p>{this.state.text}</p>

</div>

)

}

}

export default App2

**App.js** When write value of text the value will come in box and also come below line of the box

import React, { Component } from 'react';

class App2 extends Component{

state={text: "zxxxz"}

whenChange = (event) => {

this.setState ({text:event.target.value})

}

render(){

return(

<div>

<input type = "text"

value={this.state.text}

onChange={this.whenChange}/>

<p>{this.state.text}</p>

</div>

)

}

}

export default App2

**App.js** When to write both box then come both

import React, { Component } from 'react';

class App2 extends Component{

state={name: "" ,

fname: ""}

whenChange = (event) => {

const {name, value} = event.target

this.setState ({[name] : value})

}

render(){

return(

<div>

<input type = "text"

value={this.state.name} name= "name"

onChange={this.whenChange}/>

<input type = "text"

value={this.state.fname} name ="fname"

onChange = {this.whenChange}/>

<p>{this.state.name}</p>

<p>{this.state.fname}</p>

</div>

)

}

}

export default App2

zzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzzz

**App.js**

import React, { Component } from 'react';

class App extends Component {

render(){

return(

<div>

<h1>Hello World</h1>

</div>

);

}

}

export default App

**index.js**

import React from 'react';

import ReactDOM from 'react-dom';

import "./Test.css"

import App from './App'

import App2 from "./App2"

ReactDOM.render(<App/>, document.getElementById('root'));

**App.js** Calculating Render number

import React, { Component } from 'react';

class App extends Component {

state = {

number: 0

}

componentDidMount(){

this.setState(() => ({

number: (Math.random(1000))

}))

}

render(){

return(

<div>

<h1>The state number is {this.state.number}</h1>

</div>

);

}

}

export default App

The following portion code to change,You change the code and get values

}

componentDidMount(){

this.setState(() => ({

number: Math.ceil(Math.random() \* 1000)

}))

**App.js** When we try to get value after 5000 mille second

import React, { Component } from 'react';

class App extends Component {

state = {

number: 0

}

componentDidMount(){

setTimeout(() => {

this.setState(()=>({

number : Math.ceil(Math.random() \* 1000)

}))

}, 5000);

}

render(){

return(

<div>

<h1>The state number is {this.state.number} </h1>

</div>

);

}

}

export default App

**App.js** 2nd App to run After 5 seconds

import ReactDOM from 'react-dom';

class App2 extends Component {

render(){

return(

<h1>2nd Component</h1>

)

}

}

class App extends Component {

state = {

number: 0

}

componentDidMount(){

setTimeout(() => {

/\*this.setState(()=>({

number : Math.ceil(Math.random() \* 1000)

}))\*/

ReactDOM.render(<App2 />, document.getElementById('root'));

}, 5000);

}

render(){

return(

<div>

<h1>The state number is {this.state.number} </h1>

</div>

);

}

}

export default App

**App.js** through props we will try to get result In mille second

import React, { Component } from 'react';

import ReactDOM from 'react-dom';

class App2 extends Component {

render(){

return(

<div>

<h1>2nd Component</h1>

<h1>The state number is {this.props.num}</h1>

</div>

)

}

}

class App extends Component {

state = {

number: 0

}

componentWillUnmount(){

console.log("bye bye")

}

componentDidMount(){

setTimeout(() => {

this.setState(()=>({

number : Math.ceil(Math.random() \* 1000)

}))

ReactDOM.render(<App2 />, document.getElementById('root'));

}, 5000);

}

render(){

return(

<div>

<h1>The state number is {this.state.number} </h1>

<App2 num={this.state.number}/>

</div>

);

}

}

export default App

**App.s**

import React, { Component } from 'react';

import ReactDOM from 'react-dom';

class App2 extends Component {

state = {

text: this.props.num

}

static getDerivedStateFromProps(props){

return({

text: props.num

})

}

render(){

return(

<div>

<h1>2nd Component</h1>

<h1>The state number is {this.state.text}</h1>

</div>

)

}

}

class App extends Component {

state = {

number: 0

}

componentWillUnmount(){

console.log("bye bye")

}

componentDidMount(){

setTimeout(() => {

this.setState(()=>({

number : Math.ceil(Math.random() \* 1000)

}))

ReactDOM.render(<App2 />, document.getElementById('root'));

}, 5000);

}

render(){

console.log(this.state.number)

return(

<div>

<h1>The state number is {this.state.number} </h1>

<App2 num={this.state.number}/>

</div>

);

}

}

export default App

**App.js** When we use interval() instead of setTime() for continue printing/show

import React, { Component } from 'react';

import ReactDOM from 'react-dom';

class App2 extends Component {

state = {

text: this.props.num

}

static getDerivedStateFromProps(props){

return({

text: props.num

})

}

render(){

return(

<div>

<h1>2nd Component</h1>

<h1>The state number is {this.state.text}</h1>

</div>

)

}

}

class App extends Component {

state = {

number: 0

}

componentWillUnmount(){

console.log("bye bye")

}

componentDidMount(){

setInterval(() => {

this.setState(()=>({

number : Math.ceil(Math.random() \* 1000)

}))

// ReactDOM.render(<App2 />, document.getElementById('root'));

}, 5000);

}

render(){

console.log(this.state.number)

return(

<div>

<h1>The state number is {this.state.number} </h1>

<App2 num={this.state.number}/>

</div>

);

}

}

export default App