REACT PRACTICE PROGRAMS::

Functional components::

FUNCTIONAL COMPONENTS::

Greet.js::;

import React from 'react'

function Greet(){

    return <h1>Hello, Prathibha</h1>

}

export default Greet

->App.js::

import logo from './logo.svg';

import './App.css';

import Greet from './components/Greet'

function App() {

  return (

    <div className="App">

      <Greet/>

    </div>

  );

}

export default App;

output ::Hello, Prathibha

->with arrow functions::

App.js::

import logo from './logo.svg';

import './App.css';

import {Greet} from './components/Greet'

function App() {

  return (

    <div className="App">

      <Greet/>

    </div>

  );

}

export default App;

Greet.js::

import React from 'react'

/\*function Greet(){

    return <h1>Hello, Prathibha</h1>

}\*/

//name export so give extact name in app.js

export const Greet=() => <h1>Hello prathibha</h1>

//export default Greet

CLASS COMPONENTS::

App.js::

import logo from './logo.svg';

import './App.css';

import {Greet} from './components/Greet'

import Welcome from './components/welcome';

function App() {

  return (

    <div className="App">

      <Greet/>

      <Welcome/>

    </div>

  );

}

export default App;

welcome.js::

import React,{ Component } from 'react'

class Welcome extends Component{

    render(){

        return<h1>Class component</h1>

    }

}

export default Welcome

output::

**Hello prathibha**

**Class component**

Using jsx and without jsx::

Hello.js::

// eslint-disable-next-line no-unused-vars

import React from 'react'

const Hello = () =>{

    //JSX version of the component

    //return(

      //  <div className='dummyClass'>

      //      <h1>Hello</h1>

       // </div>

   // )

   //without using Jsx

   return React.createElement('div',

   {id:'hello',class:'dummy class'},

   React.createElement('h1',null,'hello without jsx'))

}

export default Hello

App.js::

import logo from './logo.svg';

import './App.css';

import {Greet} from './components/Greet'

import Welcome from './components/welcome';

import Hello from './components/Hello';

function App() {

  return (

    <div className="App">

      <Greet/>

      <Welcome/>

      <Hello/>

    </div>

  );

}

export default App;

output::

# Hello prathibha

# Class component

# hello without jsx

working with props::

Greet.js::

import React from 'react'

/\*function Greet(){

    return <h1>Hello, Prathibha</h1>

}\*/

//name export so give extact name in app.js

export const Greet=props => {

    console.log(props)

    return <h1>Hello {props.name}</h1>

}

//export default Greet

App.js::

import logo from './logo.svg';

import './App.css';

import {Greet} from './components/Greet'

import Welcome from './components/welcome';

import Hello from './components/Hello';

function App() {

  return (

    <div className="App">

      <Greet name="pranathi"/>

      <Greet name="parth"/>

      <Greet name="praps"/>

      <Welcome/>

      <Hello/>

    </div>

  );

}

export default App;

import logo from './logo.svg';

import './App.css';

import {Greet} from './components/Greet'

import Welcome from './components/welcome';

import Hello from './components/Hello';

function App() {

  return (

    <div className="App">

      <Greet name="pranathi" heroname="batman">

        <p>this is children props</p>

      </Greet>

      <Greet name="parth" heroname="kaichi">

        <button>Action</button>

      </Greet>

      <Greet name="praps" heroname="sonu"/>

      <Welcome name="pranathi" heroname="batman"/>

      <Welcome name="parth" heroname="kaichi"/>

      <Hello/>

    </div>

  );

}

export default App;

import React,{ Component } from 'react'

class Welcome extends Component{

    render(){

        return<h1>welcome {this.props.name} a.k.a {this.props.heroname}</h1>

    }

}

export default Welcome

import React from 'react'

/\*function Greet(){

    return <h1>Hello, Prathibha</h1>

}\*/

//name export so give extact name in app.js

export const Greet=props => {

    console.log(props)

    return (

    <div>

      <h1>Hello {props.name} a.k.a {props.heroname}</h1>

      {props.children}

      </div>

    )

}

//export default Greet

Output::

# Hello pranathi a.k.a batman

this is children props

# Hello parth a.k.a kaichi

Action

# Hello praps a.k.a sonu

# welcome pranathi a.k.a batman

# welcome parth a.k.a kaichi

# hello without jsx

state example::

App.js:

import logo from './logo.svg';

import './App.css';

import {Greet} from './components/Greet'

import Welcome from './components/welcome';

import Hello from './components/Hello';

import Message from './components/Message';

function App() {

  return (

    <div className="App">

      <Message/>

     {/\* <Greet name="pranathi" heroname="batman">

        <p>this is children props</p>

      </Greet>

      <Greet name="parth" heroname="kaichi">

        <button>Action</button>

      </Greet>

      <Greet name="praps" heroname="sonu"/>

      <Welcome name="pranathi" heroname="batman"/>

      <Welcome name="parth" heroname="kaichi"/>

  <Hello/>\*/}

    </div>

  );

}

export default App;

Message.js::

import React,{ Component } from 'react'

class Message extends Component{

    constructor(){

        super()

        this.state={

            message:'welcome visitor'

        }

    }

    changeMessage(){

        this.setState({

            message:'thank you for subscribing'

        })

    }

    render(){

        return(

        <div>

            <h1>{this.state.message}</h1>

            <button onClick={()=>this.changeMessage()}>subscribe</button>

            </div>

        )

    }

}

export default Message

output::

# thank you for subscribing

subscribe

with setState::

Counter.js:

import React,{ Component } from 'react'

class Counter extends Component {

    constructor(props){

        super(props)

        this.state= {

            count:0

        }

    }

    increment(){

        this.setState({

            count:this.state.count+1

        },

        ()=>{

            console.log('callback value',this.state.count)

        })

        console.log(this.state.count)

    }

    render() {

        return (

            <div>

            <div>

               count - {this.state.count}

            </div>

            <button onClick={()=>this.increment()}>Increment</button>

            </div>

        )

    }

}

export default Counter

output::

count - 2

Increment

EVENT HANDLING::

Functional component using::

import React from 'react'

function FunctionClick() {

    function clickHandler(){

        console.log('Button clicked')

    }

    return (

        <div>

            <button onClick={clickHandler}>Click</button>

        </div>

    )

}

export default FunctionClick

output::

3FunctionClick.js:5 Button clicked

For class component::

import React, { Component } from 'react'

class ClassClick extends Component {

    clickHandler(){

        console.log('clicked me button')

    }

    render() {

        return (

            <div>

                <button onClick={this.clickHandler}>click me</button>

            </div>

        )

    }

}

export default ClassClick

binding in event handlers::

approach 1::binding in render method

import React, { Component } from 'react'

class EventBind extends Component {

    constructor(props){

        super(props)

        this.state={

            message:'Hello'

        }

    }

    ClickHandler(){

        this.setState({

            message:'goodbye'

        })

    }

    render() {

        return (

            <div>

                <div>{this.state.message}</div>

               <button onClick={this.ClickHandler.bind(this)}>Click</button>

            </div>

        )

    }

}

export default EventBind

output:

Hello

Click

After click::

goodbye

Click

By arrow functions approach 2:: in render method

<button onClick={()=>this.ClickHandler()}>Click</button>

Approach 3:binding in constructor::

import React, { Component } from 'react'

class EventBind extends Component {

    constructor(props){

        super(props)

        this.state={

            message:'Hello'

        }

//in constructor binding

        this.ClickHandler=this.ClickHandler.bind(this)

    }

    ClickHandler(){

        this.setState({

            message:'goodbye'

        })

    }

    render() {

        return (

            <div>

                <div>{this.state.message}</div>

              {/\* <button onClick={this.ClickHandler.bind(this)}>Click</button> \*/}

              {/\*<button onClick={()=>this.ClickHandler()}>Click</button>\*/}

              <button onClick={this.ClickHandler}>Click</button>

            </div>

        )

    }

}

export default EventBind

Approach 4 binding by classproperties as arrowfunctions::

import React, { Component } from 'react'

class EventBind extends Component {

    constructor(props){

        super(props)

        this.state={

            message:'Hello'

        }

        this.ClickHandler=this.ClickHandler.bind(this)

    }

  // ClickHandler(){

    //    this.setState({

     //       message:'goodbye'

       // })

    //}

//binding in the class properties

    ClickHandler=()=>{

        this.setState({

            message: 'goodbye'

        })

    }

    render() {

        return (

            <div>

                <div>{this.state.message}</div>

              {/\* <button onClick={this.ClickHandler.bind(this)}>Click</button> \*/}

              {/\*<button onClick={()=>this.ClickHandler()}>Click</button>\*/}

              <button onClick={this.ClickHandler}>Click</button>

            </div>

        )

    }

}

export default EventBind

CONDITIONAL RENDERING::

IF/ELSE::

UserGreeting::

import React, { Component } from 'react'

class UserGreeting extends Component {

    constructor(props) {

        super(props)

        this.state = {

             isLoggedIn:true

        }

    }

    render() {

        if(this.state.isLoggedIn){

            return(

                <div>Welcome prathibha</div>

            )

        }else{

            return(

                <div>welcome guest</div>

            )

        }

        //return (

          //  <div>

          //  <div>Welcome guest</div>

            //<div> Welcome prathibha </div>

            //<///div>

       // )

    }

}

export default UserGreeting

by element variables::

import React, { Component } from 'react'

class UserGreeting extends Component {

    constructor(props) {

        super(props)

        this.state = {

             isLoggedIn:true

        }

    }

    render() {

        let message

        if(this.state.isLoggedIn){

            message=<div>welcome prathibha</div>

        }else{

            message=<div>welcome guest</div>

        }

        return<div>{message}</div>

        //if(this.state.isLoggedIn){

          //  return(

            //    <div>Welcome prathibha</div>

            //)

        //}else{

          //  return(

            //    <div>welcome guest</div>

           // )

        //}

        //return (

          //  <div>

          //  <div>Welcome guest</div>

            //<div> Welcome prathibha </div>

            //<///div>

       // )

    }

}

export default UserGreeting

by ternary operator::

import React, { Component } from 'react'

class UserGreeting extends Component {

    constructor(props) {

        super(props)

        this.state = {

             isLoggedIn:false

        }

    }

    render() {

        return(

            this.state.isLoggedIn?

            <div>welcome prathibha</div>:

            <div>welcome guest</div>

        )

        //let message

        //if(this.state.isLoggedIn){

         //   message=<div>welcome prathibha</div>

        //}else{

        //    message=<div>welcome guest</div>

        //}

        //return<div>{message}</div>

        //if(this.state.isLoggedIn){

          //  return(

            //    <div>Welcome prathibha</div>

            //)

        //}else{

          //  return(

            //    <div>welcome guest</div>

           // )

        //}

        //return (

          //  <div>

          //  <div>Welcome guest</div>

            //<div> Welcome prathibha </div>

            //<///div>

       // )

    }

}

export default UserGreeting

short circuit operator::

import React, { Component } from 'react'

class UserGreeting extends Component {

    constructor(props) {

        super(props)

        this.state = {

             isLoggedIn:true

        }

    }

    render() {

      return  this.state.isLoggedIn && <div>welcome prathibha</div>

       // return(

         //   this.state.isLoggedIn?

           // <div>welcome prathibha</div>:

            //<div>welcome guest</div>

      //  )

        //let message

        //if(this.state.isLoggedIn){

         //   message=<div>welcome prathibha</div>

        //}else{

        //    message=<div>welcome guest</div>

        //}

        //return<div>{message}</div>

        //if(this.state.isLoggedIn){

          //  return(

            //    <div>Welcome prathibha</div>

            //)

        //}else{

          //  return(

            //    <div>welcome guest</div>

           // )

        //}

        //return (

          //  <div>

          //  <div>Welcome guest</div>

            //<div> Welcome prathibha </div>

            //<///div>

       // )

    }

}

export default UserGreeting

LIST of names without map::

import React from 'react'

function NameList() {

    const name=['path', 'dimple','daya']

    return (

        <div>

            <h2>{name[0]}</h2>

            <h2>{name[1]}</h2>

            <h2>{name[2]}</h2>

        </div>

    )

}

export default NameList

list of names with map:

import React from 'react'

function NameList() {

    const name=['path', 'dimple','daya']

    const nameList= name.map(name =><h2>{name}</h2>)

    return (

        <div>

            {nameList}

        </div>

    )

}

export default NameList

by using map with list of names

import React from 'react'

function NameList() {

    const persons=[

        {

            id:1,

            name:'parth',

            age:30,

            skill:'React'

        },

        {

            id:2,

            name:'priya',

            age:25,

            skill:'Angular'

        },

        {

            id:3,

            name:'prass',

            age:20,

            skill:'Vue'

        }

    ]

    const personList= persons.map(person =>(

    <h2>

        I am{person.name}.I am {person.age} years old.I know {person.skill}</h2>))

    return (

        <div>

            {personList}

        </div>

    )

}

export default NameList

output::

## I amparth.I am 30 years old.I know React

## I ampriya.I am 25 years old.I know Angular

## I amprass.I am 20 years old.I know Vue

React redux::

CAKE SHOP EXAMPLE::

App.js:

import logo from './logo.svg';

import React from 'react'

import { Provider } from 'react-redux';

import store from './redux/store'

import './App.css';

import CakeContainer from './components/CakeContainer';

function App() {

  return (

    <Provider store={store}>

    <div className="App">

      <CakeContainer/>

    </div>

    </Provider>

  );

}

export default App;

CakeContainer.js:

import React from 'react'

import { connect } from 'react-redux'

import {buyCake} from 'redux'

function CakeContainer(props) {

    return (

        <div>

            <h2>number of cakes-{props.numOfCakes}</h2>

            <button onClick={props.buyCake}>Buy cake</button>

        </div>

    )

}

const mapStateProps = state =>{

    return{

        numOfCakes:state.numOfCakes

    }

}

const mapDispatchToProps= dispatch =>{

    return{

        buyCake: () =>dispatch(buyCake())

    }

}

export default connect(

    mapStateProps,mapDispatchToProps)

(CakeContainer)

cakeActions.js:

import { BUY\_CAKE } from "./cakesType"

export const buyCake=(number = 1) => {

    return{

        type: BUY\_CAKE,

        payload: number

    }

}

cakeReducer.js:

import {BUY\_CAKE} from './cakesType'

const initialState={

    numOfCakes:10

}

const cakeReducer=(state=initialState,action)=>{

    switch(action.type){

        case BUY\_CAKE: return{

            ...state,

            numOfCakes:state.numOfCakes-action.payload

        }

        default:return state

    }

}

export default cakeReducer

store.js:

import {createStore} from 'redux'

import cakeReducer from './cake/cakeReducer'

const store = createStore(cakeReducer)

export default store

index.js:

export { buyCake } from './cake/cakeActions'

output:

number of cakes 10

buycake(button)