// Props and states example without destructuting

import React, { Component } from 'react';

function Propos(props) {

    return (

        <div>

            <h1>{props.name}</h1>

        </div>

    )

}

export default class PropsAndStates1 extends Component {

    constructor(){

        super();

        this.state={

            company:"TY"

        }

    }

    render() {

        return (

            <div>

                <h1>{this.props.name}</h1>

                <Propos name="shashank"/>

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

            </div>

        )

    }

}

// Props and state Example with Destructuring

import React, { Component } from 'react'

function Propos({name}) {

    return (

        <div>

            <h1>{name}</h1>

        </div>

    )

}

function Propss(props) {

     var {place}=props;

    return (

        <div>

            <Propos name="shashank"/>

            <h1>{place}</h1>

        </div>

    )

}

export default class PropsAndStates2 extends Component {

    render() {

        return (

            <div>

                <Propss name="sudesh" place="mysore"/>

                <h1>hii hello{this.props.name}</h1>

            </div>

        )

    }

}

//functional component

1

import React from 'react'

export default function Navbar(props) {

    return (

        <div>

            <h1>i m navbar component</h1>

            <h1>given input props is {props.name}</h1>

        </div>

    )

}

2

import React from 'react'

export default function Sidebar({name}) {

    return (

        <div>

            <h1>i m sidebar component</h1>

            <h1>destructured props is {name}</h1>

        </div>

    )

}

3

import React from 'react'

export default function Footer() {

    return (

        <div>

            <h1>i m footer component</h1>

        </div>

    )

}

//class component

1

import React, { Component } from 'react'

export default class FirstComponent extends Component {

    render() {

        return (

            <div>

                <h1>Entered prop is {this.props.name}</h1>

            </div>

        )

    }

}

2

import React, { Component } from 'react'

export default class SecondComponent extends Component {

    constructor(){

        super();

        this.state = {

            name:"shashank"

        }

    }

    render() {

        return (

            <div>

                <h1>The state value is {this.state.name}</h1>

            </div>

        )

    }

}

3

import React, { Component } from 'react'

export default class ThirdComponent extends Component {

    constructor(){

        super();

        this.state = {

            name : "shashank"

        }

    }

    changeStateValue = () =>{

        this.setState({

            name: "SHASHANK"

        })

    }

    render() {

        return (

            <div>

                <h1>this is {this.state.name}</h1>

                <button onClick={this.changeStateValue}>Click to change</button>

            </div>

        )

    }

}

// conditional rendering

import React from 'react'

function Choclate(props) {

    return (

        <div>

            <h1>The Requested Food is {props.name}</h1>

        </div>

    )

}

function Biscuit(props) {

    return (

        <div>

            <h1>The Requested Food is {props.name}</h1>

        </div>

    )

}

function ConditionalRendering(props) {

   if(props.food === "choclate"){

       return <Choclate name="Dairymilk"/>

   } else if (props.food === "Biscuit"){

       return <Biscuit name="HappyHappy"/>

   }

}

export default ConditionalRendering;

//event handling

import React, { Component } from "react";

export default class EventHandle extends Component {

  constructor() {

    super();

    this.state = {

      name: "shashank"

    };

  }

  change = () => {

    this.setState({

      name: "santhosh"

    });

  };

  render() {

    return (

      <div>

        <h1>hello i m evevn handler</h1>

        <h>{this.state.name}</h>

        <button onClick={this.change}>click</button>

      </div>

    );

  }

}

//life cycle

//mounting

1

import React from "react";

class Food extends React.Component {

  constructor(props) {

    super(props);

    this.state = { food: "dosa" };

  }

  componentDidMount() {

    setTimeout(() => {

      this.setState({ food: "idli" });

    }, 2000);

  }

  render() {

    return <h1>My Favorite Food is {this.state.food}</h1>;

  }

}

export default Food;

2

import React , {Component} from "react"

 class Greeing extends Component {

    constructor(props) {

      super(props);

      this.state = {greeting: "Hi"};

    }

    greet = () => {

      this.setState({greeting: "Hello"});

    }

    render() {

      return (

        <div>

        <h1>Greet them  {this.state.greeting}</h1>

        <button type="button" onClick={this.greet}>change</button>

        </div>

      );

    }

  }

  export default Greeing

//updating

1

import React , {Component} from "react"

 class Greeing extends Component {

    constructor(props) {

      super(props);

      this.state = {greeting: "Hi"};

    }

    shouldComponentUpdate() {

      return true;

    }

    greet = () => {

      this.setState({greeting: "Hello"});

    }

    render() {

      return (

        <div>

        <h1>Greet them  {this.state.greeting}</h1>

        <button type="button" onClick={this.greet}>change</button>

        </div>

      );

    }

  }

  export default Greeing

2

import React, { Component } from 'react'

export default class Subject extends Component {

    constructor(){

        super();

        this.state = {

            language : "kannada"

        }

    }

    componentDidMount(){

        setTimeout(() => {

            this.setState({

                language : "english"

            })

        },2000)

    }

    getSnapshotBeforeUpdate(prevState){

        document.getElementById('line1').innerHTML="before update the state value was " + prevState.language

    }

    componentDidUpdate(){

        document.getElementById('line2').innerHTML="After update the state value is " + this.state.language

    }

    render() {

        return (

            <div>

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

                <div id="line1"></div>

                <div id="line2"></div>

            </div>

        )

    }

}

// unmounting

1

import React, { Component } from 'react'

export default class Box extends Component {

    constructor(){

        super();

        this.state = {

            condition: true

        }

    }

    DeleteContent=()=>{

        this.setState({

            condition:false

        })

    }

    RetriveContent=()=>{

        this.setState({

            condition:true

        })

    }

    render() {

        let body;

        if(this.state.condition){

            body=<Content/>

        }

        return (

            <div>

                {body}

                <button onClick={this.DeleteContent}>Delete</button>

                <button onClick={this.RetriveContent}>Retrive</button>

            </div>

        )

    }

}

class Content extends Component{

    componentWillUnmount(){

        alert("i m unmounting")

    }

    render(){

        return(

            <div>

                <h1>Hii.. Hello </h1>

            </div>

        )

    }

}

2

import React, { Component } from "react";

export default class Box extends Component {

  constructor() {

    super();

    this.state = {

      condition: true,

    };

  }

  DeleteContent = () => {

    this.setState({

      condition: false,

    });

  };

  render() {

    let body;

    if (this.state.condition) {

      body = <Content />;

    }

    return (

      <div>

        {body}

        <button onClick={this.DeleteContent}>Delete</button>

      </div>

    );

  }

}

class Content extends Component {

  componentWillUnmount() {

    alert("i m unmounting");

  }

  render() {

    return (

      <div>

        <h1>Hii.. Hello </h1>

      </div>

    );

  }

}