



import React, { Component } from 'react'

export default class Product extends Component {

    constructor(props)

    {

        super(props)

        console.log("constructor of Product called")

     this.state=

     {

    productId:"", price:0

     }

    }

    render() {

    return (

      <div>Product

        <Cart/>

      </div>

    )

  }

}

export class Cart extends React.Component

{

    constructor(props)

    {

        super(props)

        console.log("constructor of Cart called")

     this.state=

     {

    count:0

     }

    }

    componentDidMount()

    {

        console.log("componentDidMount of Cart called")

    }

    componentDidUpdate(prevProp, prevState)

    {

        console.log("componentDidUpdate of Cart called")

    }

     Update=()=>

     {

         this.setState({qty: this.state.qty+10})

     }

     render()

     {

        console.log("render called")

        return(

            <>

            <h1> Cart called </h1>

            <button onClick={this.Update}> Update Qty </button>

            </>

        )

     }

}

import React, { Component } from 'react'

export default class Product extends Component {

    constructor(props)

    {

        super(props)

        console.log("constructor of Product called")

     this.state=

     {

    productId:"",

    price:0

     }

    }

 addToCart = (pid)=>

 {

    this.setState({productId: pid, price : this.state.price+1}

    )

 }

    render() {

    return (

      <div>Product

        <button onClick={()=> this.addToCart(1)}> Add to Cart </button>

        <button onClick={()=> this.addToCart(2)}> Add to Cart </button>

        <button onClick={()=> this.addToCart(3)}> Add to Cart </button>

        <Cart productId={this.state.productId} price={this.state.price}/>

      </div>

    )

  }

}

export class Cart extends React.Component

{

    constructor(props)

    {

        super(props)

        console.log("constructor of Cart called")

     this.state=

     {

    price:this.props.price

     }

    }

    componentDidMount()

    {

        console.log("componentDidMount of Cart called")

    }

    static getDerivedStateFromProps (props, state)

    {

        console.log("getDerivedStateFromProps called")

        if(props.price != state.price)

        return{price: props.price};

     return null;

    }

    componentDidUpdate(prevProp, prevState)

    {

        if(this.props.productId != prevProp.productId)

        {

        console.log("Prev Prop" + prevProp.price)

        console.log("Prev State" + prevState.price)

        console.log("componentDidUpdate of Cart called")

        }

    }

     Update=()=>

     {

         this.setState({price: this.state.price+10})

     }

     shouldComponentUpdate(nextProps, nextState)

     {

        if(this.props.price !== nextProps.price)

        {

        console.log("shouldComponentUpdate called")

     return true;

    }

    else

    return false;

     }

     render()

     {

        console.log("render called")

        return(

            <>

            <h1> Cart called  {this.state.price} </h1>

            <button onClick={this.Update}> Update Qty </button>

            </>

        )

     }

}

import { isClickableInput } from '@testing-library/user-event/dist/utils'

import React, { Component } from 'react'

export default class Product extends Component {

    constructor(props)

    {

        super(props)

        console.log("constructor of Product called")

     this.state=

     {

    productId:"",

    price:0,

    isCart:true

     }

    }

 addToCart = (pid)=>

 {

    this.setState({productId: pid, price : this.state.price+1}

    )

 }

 removeCart=()=>

 {

    alert("Remove ")

    this.setState({isCart:false})

 }

    render() {

        alert(this.state.isCart)

    return (

      <div>Product

        <button onClick={()=> this.addToCart(1)}> Add to Cart </button>

        <button onClick={()=> this.addToCart(2)}> Add to Cart </button>

        <button onClick={()=> this.addToCart(3)}> Add to Cart </button>

     <button onClick={this.removeCart}> Remove Cart </button>

       {!this.state.isCart && <h2> Cart has been removed </h2>}

       {this.state.isCart && <Cart productId={this.state.productId} price={this.state.price}/>}

      </div>

    )

  }

}

export class Cart extends React.Component

{

    constructor(props)

    {

        super(props)

        console.log("constructor of Cart called")

     this.state=

     {

    price:this.props.price

     }

    }

    componentDidMount()

    {

        console.log("componentDidMount of Cart called")

    }

    static getDerivedStateFromProps (props, state)

    {

        console.log("getDerivedStateFromProps called")

        if(props.price != state.price)

        return{price: props.price};

     return null;

    }

    componentDidUpdate(prevProp, prevState)

    {

        if(this.props.productId != prevProp.productId)

        {

        console.log("Prev Prop" + prevProp.price)

        console.log("Prev State" + prevState.price)

        console.log("componentDidUpdate of Cart called")

        }

    }

     Update=()=>

     {

         this.setState({price: this.state.price+10})

     }

     shouldComponentUpdate(nextProps, nextState)

     {

        if(this.props.price !== nextProps.price)

        {

        console.log("shouldComponentUpdate called")

     return true;

    }

    else

    return false;

     }

     render()

     {

        console.log("render called")

        return(

            <>

            <h1> Cart called  {this.state.price} </h1>

            <button onClick={this.Update}> Update Qty </button>

            </>

        )

     }

     componentWillUnmount()

     {

        console.log("Cart has been removed")

     }

}