 class Car extends React.Component{

render(){

 return <h2>Hi!I am a blue car</h2>

}

 }

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

 function Car(){

 return <h2>Hi!I am a red car</h2>

}

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

 class Car extends React.Component{

   constructor(){

      super();

        this.state= {color:"red"};

      }

      render(){

      return <h1>I have a {this.state.color} Car.</h1>;

      }

   }

class Car extends React.Component{

  render(){

    return <h1>I am a {this.props.color} Car.</h1>

  }

}

   ReactDOM.render(<Car color="blue-pink"/>,document.getElementById('root'));

class Car extends React.Component{

  render(){

  return <h1>I am a {this.props.brand}</h1>

  }

}

const myelement=<Car brand="Audi"/>

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

class Car extends React.Component{

  render(){

    return <h1>I am a {this.props.brand} Car!</h1>

  }

}

class Garage extends React.Component{

  render(){

    return(

      <div>

        <h2>Who lives in my garage?</h2>

        <Car brand="Ford"/>

      </div>

    );

  }

}

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

class Car extends React.Component{

  render(){

    return <h1>I am a car</h1>

  }

}

class Garage extends React.Component{

  render(){

    return(

      <div>

        <h3>Who lives in my garage</h3>

        <Car/>

      </div>

    );

  }

}

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

import React, { Component } from 'react';

import ReactDOM from 'react-dom';

import Car from './Car.js';

// import './index.css';

// import App from './App';

// import \* as serviceWorker from './serviceWorker';

// ReactDOM.render(

//   <React.StrictMode>

//     <App />

//   </React.StrictMode>,

  //  document.getElementById('root');

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

// If you want your app to work offline and load faster, you can change

// unregister() to register() below. Note this comes with some pitfalls.

// Learn more about service workers: https://bit.ly/CRA-PWA

// serviceWorker.unregister();

class Car extends React.Component{

  render(){

  return <h1>This is my {this.props.brand} Car!!!</h1>

  }

}

class Garage extends React.Component{

  render(){

    const carname="BMW";

    return(

      <div>;

        <h2>Who stays in our garage</h2>

        <Car brand={carname}/>

      </div>

    );

  }

}

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

class Car extends React.Component{

  render(){

    return <h1>I am a {this.props.brand.model} Car!!</h1>

  }

}

class Garage extends React.Component{

  render(){

    const carinfo={name:"Ford",model:"Mustang"};

    return(

      <div>

        <h2>Who lives in the garage</h2>

        <Car brand={carinfo}/>

      </div>

    );

  }

}

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

  class Car extends React.Component {

    constructor(props) {

      super(props);

    }

    render() {

      return <h2>I am a Car!</h2>;

    }

  }

  ReactDOM.render(<Car model="Mustang"/>, document.getElementById('root'));

 class Car extends React.Component{

    constructor(props){

      super(props);

      this.state={

        model:"Mustang",

        brand:"Ford",

        color:"blue",

        tyre:"ceat"

      };

    }

    render(){

      return(

        <div>

          <h1>My{this.state.brand}</h1>

          <h2>My Car Tyre {this.state.tyre}</h2>

          <p>

            It is {this.state.model} and color is  {this.state.color }

          </p>

        </div>

      );

    }

  }

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

 class Person extends React.Component{

   constructor(props){

     super(props);

     this.state={

      firstName:"Swetha",

      lastName:"Janardhan",

      city:"Mumbai",

      gender:"Female"

     };

   }

   changeName=()=>{

     this.setState({lastName:"Arvind"});

   }

   render(){

     return (

       <div>

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

         <p>

           My lastName is {this.state.lastName} and my gender is {this.state.gender} and my city is {this.state.city}

         </p>

         <button type="button" onClick={this.changeName}>Button</button>

       </div>

     );

   }

 }

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

class Header extends React.Component{

  constructor(props){

    super(props);

    this.state={favoritecolor:"Yellow"};

  }

  render(){

    return (

    <h1>My favourite color is {this.state.favoritecolor} .</h1>

    );

  }

}

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

class Header extends React.Component{

  constructor(props){

    super(props);

    this.state={favouritecolor:"Red"};

  }

  static getDerivedStateFromProps(props,state){

    return {favouritecolor:props.favcolor}

  }

  render(){

    return (

      <h1>My favourite color is {this.state.favouritecolor} .</h1>

    );

  }

}

ReactDOM.render(<Header favcolor="green"/>,document.getElementById('root'));

class Header extends React.Component{

  constructor(props){

    super(props);

    this.state={favouritecolor:"Brown"};

  }

  static getDerivedStateFromProps(props,state){

    return {favouritecolor:props.favcolor};

  }

changeColor=() => {

  this.setState({favouritecolor:"Black"});

}

render(){

  return(

    <div>

  <h1>My favourite color is {this.state.favouritecolor}</h1>

  <button type="button" onClick={this.changeColor}>Click me</button>

  </div>

  );

}

}

ReactDOM.render(<Header favcolor="Green"/>,document.getElementById('root'));

class Header extends React.Component{

  constructor(props){

    super(props);

    this.state={favouritecolor:"Blue"};

  }

  shouldComponentUpdate() {

    return true;

  }

  changeColor= () => {

    this.setState({favouritecolor:"Red"});

  }

  render(){

    return (

      <div>

        <h1>My favourite color is {this.state.favouritecolor}.</h1>

        <button type="button" onClick={this.changeColor}>Click me</button>

      </div>

    );

  }

}

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

class Header extends React.Component{

  constructor(props){

    super(props);

    this.state={favouritecolor:"Blue"}

  }

  changeColor=()=>{

    this.setState({favouritecolor:"Yellow"});

  }

  render(){

    return (

      <div>

        <h1>My favourite color is {this.state.favouritecolor}.</h1>

        <button type="button" onClick={this.changeColor}>Click me </button>

      </div>

    );

  }

}

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

class Header extends React.Component{

  constructor(props){

    super(props);

    this.state={favouritecolor:"Red"};

  }

  componentDidMount(){

    setTimeout(() => {

      this.setState({favouritecolor:"Pink"})

    }, 2000);

  }

  getSnapshotBeforeUpdate(prevprop,prevstate){

    document.getElementById('div1').innerHTML=

    " Before update the favourite color was " + prevstate.favouritecolor;

  }

componentDidUpdate(){

document.getElementById('div2').innerHTML=

"After update the value is " + this.state.favouritecolor;

}

render(){

  return (

    <div>

      <h1>My favourite color is {this.state.favouritecolor}.</h1>

      <div id="div1"></div>

      <div id="div2"></div>

    </div>

  );

}

}

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

class Header extends React.Component{

  constructor(props){

    super(props);

    this.state={favouritecolor:"Orange"};

  }

  componentDidMount(){

    setTimeout(() => {

      this.setState({favouritecolor:"Blue"})

    }, 2000)

  }

  componentDidUpdate(){

    document.getElementById('mydiv').innerHTML=

    "The updated favourite is " + this.state.favouritecolor;

  }

  render(){

    return (

      <div>

        <h1>My favourite color is {this.state.favouritecolor} .</h1>

        <div id="mydiv"></div>

      </div>

    );

  }

}

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

function shoot() {

  alert("Great shot");

}

const myelement=(

  <button type="button" onClick={shoot}>Click me!</button>

)

ReactDOM.render(myelement,document.getElementById('root'))

-----------------------------------------------------------------------------

class Football extends React.Component{

  constructor(props){

    super(props);

    this.shoot=this.shoot.bind(this)

  }

  shoot() {

alert(this);

  }

  render(){

    return(

      <button  onClick={this.shoot}>Click me</button>

    )}}

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

class Football extends React.Component{

  // shoot=(a)=>

  shoot(a){

    alert(a);

  }

  render(){

    return (

      // <button onClick={()=>this.shoot("Goal")}>Shoot it</button>

      <button onClick={this.shoot.bind(this,"goal")}>Shoot it </button>

    );

  }

}

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

class Football extends React.Component{

  shoot=(a,b) =>{

    alert(b.type);

  }

  render(){

    return (

      // <button onClick={(ev) => this.shoot("Goal",ev)}>Click me!!</button>

      <button onClick={this.shoot.bind(this,"Goal")}>Click me</button>

    );

  }

}

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

Add a submit button and an event handler in the onSubmit attribute:

class MyForm extends React.Component {

constructor(props) {

super(props);

this.state = { username: '' };

}

mySubmitHandler = (event) => {

event.preventDefault();

alert("You are submitting " + this.state.username);

}

myChangeHandler = (event) => {

this.setState({username: event.target.value});

}

render() {

return (

<form onSubmit={this.mySubmitHandler}>

<h1>Hello {this.state.username}</h1>

<p>Enter your name, and submit:</p>

<input

type='text'

onChange={this.myChangeHandler}

/>

<input

type='submit'

/>

</form>

);

}

}

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

class Container extends React.Component{

  constructor(props){

    super(props);

    this.state={show:true};

  }

  delHeader=() =>{

    this.setState({show:false});

  }

  render(){

    let myheader;

    if(this.state.show){

      myheader=<Child/>;

  };

  return(

    <div>

      {myheader}

      <button type="button" onClick={this.delHeader}>Click me!!</button>

    </div>

  );

}

}

class Child extends React.Component{

  componentWillUnmount(){

    alert("The component has unmounted");

  }

  render() {

    return (

      <h1>Hello World!!!</h1>

    );

  }

}

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

 Myform extends React.Component {

    constructor(props){

      super(props);

      this.state={username:'',

     age:null,

  };

    }

    myChangeHandler=(event) => {

      let name=event.target.name;

      let val=event.target.value;

      this.setState({[name]:val});

    }

    render(){

      return(

        <form>

          <h1>Hello { this.state.username} {this.state.age} </h1>

          <p>Enter your name:</p>

          <input type="text" name="username" onChange={this.myChangeHandler}/>

          <p>Enter your age:</p>

          <input type="text" name="age" onChange={this.myChangeHandler}/>

        </form>

      );

    }

  }

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

 class Myform extends React.Component {

    constructor(props){

      super(props);

      this.state={username:'',

     age:null,

  };

    }

    myChangeHandler=(event) => {

      let name=event.target.name;

      let val=event.target.value;

      if(name ==="age"){

        if(!Number (val)){

          alert("Your age is a number");

        }

      }

      this.setState({[name]:val});

    }

    render(){

      return(

        <form>

          <h1>Hello { this.state.username} {this.state.age} </h1>

          <p>Enter your name:</p>

          <input type="text" name="username" onChange={this.myChangeHandler}/>

          <p>Enter your age:</p>

          <input type="text" name="age" onChange={this.myChangeHandler}/>

        </form>

      );

    }

  }

class MyForm extends React.Component{

   constructor(props){

     super(props);

     this.state={

       username:'',

       age:'',

       errormessage:'',

     };

   }

myChangeHandler=(event) => {

let nam=event.target.name;

let val=event.target.value;

let err='';

if(nam ==="age"){

if(val !="" && !Number(val)){

  err=<strong>Your age must be a number</strong>

}

}

this.setState({errormessage:err})

this.setState({[nam]:val});

}

render() {

  return (

<form>

  <h1>Hello {this.state.username} {this.state.age}</h1>

  <p>Enter your name:</p>

  <input type="text" name="username" onChange={this.myChangeHandler}/>

  <p>Enter your age:</p>

  <input type="text" name="age" onChange={this.myChangeHandler}/>

  {this.state.errormessage}

</form>

  );

}

 }

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

class Myform extends React.Component{

   constructor(props){

     super(props);

     this.state={ username: '',

     age:null,

   };

 }

 mySubmitHandler=(event)=>{

   event.preventDefault();

   let age=this.state.age;

   if(!Number(age)){

     alert(" Your age must be a number ");

   }

 }

 myChangeHandler=(event)=>{

   let nam=event.target.name;

   let val=event.target.value;

   this.setState({[nam]:val});

 }

 render() {

   return (

     <form onSubmit={this.mySubmitHandler}>

      <h1>Hello {this.state.username} {this.state.age}</h1>

      <p>Enter your name:</p>

      <input type="text" name="username" onChange={this.myChangeHandler}/>

      <p>Enter your age:</p>

      <input type="text" name="age" onChange={this.myChangeHandler}/>

      <br/>

      <br/>

      <input type="submit"/>

      </form>

   );

 }

}

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

 class MyForm extends React.Component{

   constructor(props){

     super(props);

     this.state={

       description:'The content of a textarea goes in the value attribute'

     };

   }

   render(){

     return(

       <form>

         <textarea value={this.state.description}/>

       </form>

     );

   }

 }

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

  class MyForm extends React.Component {

    constructor(props) {

      super(props);

      this.state = {

        mycar: 'Volvo'

      };

    }

    myChangehandler=(event)=>{

      let nam=event.target.name;

      let val=event.target.value;

      this.setState({[nam]:val});

    }

    render() {

      return (

        <form>

        <select value={this.state.mycar} onChange={this.myChangehandler}>

          <option value="Ford">Ford</option>

          <option value="Volvo">Volvo</option>

          <option value="Fiat">Fiat</option>

        </select>

        </form>

      );

    }

  }

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

class Header extends React.Component{

  render(){

    return (

      <div>

      <h1 style={{color:"yellow"}}>Hello world</h1>

      <p>This is a header</p>

      </div>

    );

  }

}

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

class Header extends React.Component{

  render(){

    return (

      <div>

        <h1 style={{backgroundColor:"Pink"}}>Hello world!!!</h1>

        <p>This is a header</p>

      </div>

    );

  }

}

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

class Header extends React.Component{

  render(){

    const myelement={

      color:"Red",

      backgroundColor:"Yellow",

      padding:"10px",

      fontFamily:"Verdana"

    };

    return(

      <div>

        <h1 style={myelement}>HELLO WORLD</h1>

        <p>This ia s header</p>

      </div>

    );

  }

}

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

  class Header extends React.Component {

     render() {

       var mystyle={

         color:"Red",

         fontStyle:"italic"

       }

        return (

           <div>

              <h1 style={mystyle}>Header</h1>

              <h2 style={mystyle}>Content</h2>

              <p data-myattribute = "somevalue">This is the content!!!</p>

           </div>

        );

     }

  }