import React, {Component} from 'react'

class **Topic** extends Component {  
 render{  
 return(  
 <div>  
 **{this.props.name}**  
 </div>  
 )  
 }  
}

import React, {Component} from 'react'

class **Welcome** extends Component {  
 render{  
 return(  
 <div>  
 <p> Welcome to React, today you will learn: </p>  
 **<Topic name="Props"/>  
 <Topic name="State"/>**  
 </div>  
 )  
 }  
}

**State**

class **Counter** extends Component{  
   
 constructor(props){  
 super(props);  
 **this.state** = {counter: 0}  
 }

render(){  
 return(  
 <p>{**this.state.counter**}</p>  
 )  
}

class Counter extends Component{  
   
 constructor(props){  
 super(props);  
 **this.state** = {counter: 0}  
 this.increment = this.increment.bind(this);  
 }  
 increment(){  
 this.**setState**({counter: **this.state.counter** + 1})  
 }

render(){  
 return(  
 <button onClick={this.increment}>Like</button>  
 <p>**{this.state.counter}**</p>  
 )  
}

https://www.robinwieruch.de/react-pass-props-to-component/

## **What are Props in React?**

import React, { Component } from 'react';

class App extends Component {

render() {

const greeting = 'Welcome to React';

return (

<div>

<h1>{greeting}</h1>

</div>

);

}

}

export default App;

2]

import React, { Component } from 'react';

class App extends Component {

render() {

return (

<div>

<Greeting />

</div>

);

}

}

class Greeting extends Component {

render() {

const greeting = 'Welcome to React';

return <h1>{greeting}</h1>;

}

}

export default App;

3]

import React, { Component } from 'react';

class App extends Component {

render() {

const greeting = 'Welcome to React';

return (

<div>

<Greeting greeting={greeting} />

</div>

);

}

}

class Greeting extends Component {

render() {

return <h1>{this.props.greeting}</h1>;

}

}

export default App;

4]

import React, { Component } from 'react';

class App extends Component {

render() {

const greeting = 'Welcome to React';

return (

<div>

<Greeting greeting={greeting} />

</div>

);

}

}

const Greeting = props => <h1>{props.greeting}</h1>;

export default App;

5]

import React, { Component } from 'react';

class App extends Component {

render() {

return (

<div>

<Greeting greeting="Welcome to React" />

</div>

);

}

}

const Greeting = ({ greeting }) => <h1>{greeting}</h1>;

export default App;

6]

import React, { Component } from 'react';

class App extends Component {

render() {

return (

<div>

<Greeting greeting={{ text: 'Welcome to React' }} />

</div>

);

}

}

const Greeting = ({ greeting }) => <h1>{greeting.text}</h1>;

export default App;

## **React Props vs. State**

import React, { Component } from 'react';

class App extends Component {

constructor(props) {

super(props);

this.state = {

isShow: true,

};

}

toggleShow = () => {

this.setState(state => ({ isShow: !state.isShow }));

};

render() {

return (

<div>

{this.state.isShow ? <Greeting greeting={greeting} /> : null}

<button onClick={this.toggleShow} type="button">

Toggle Show

</button>

</div>

);

}

}

const Greeting = ({ greeting }) => <h1>{greeting}</h1>;

export default App;

2]

import React, { Component } from 'react';

class App extends Component {

constructor(props) {

super(props);

this.state = {

isShow: true,

};

}

toggleShow = () => {

this.setState(state => ({ isShow: !state.isShow }));

};

render() {

const greeting = 'Welcome to React';

return (

<div>

<Greeting greeting={greeting} isShow={this.state.isShow} />

<button onClick={this.toggleShow} type="button">

Toggle Show

</button>

</div>

);

}

}

const Greeting = ({ greeting, isShow }) =>

isShow ? <h1>{greeting}</h1> : null;

export default App;

## **How to pass Props from child to parent Component?**

import React, { Component } from 'react';

class App extends Component {

render() {

const greeting = 'Welcome to React';

return (

<div>

{isShow ? <Greeting greeting={greeting} /> : null}

<Button />

</div>

);

}

}

class Button extends Component {

constructor(props) {

super(props);

this.state = {

isShow: true,

};

}

toggleShow = () => {

this.setState(state => ({ isShow: !state.isShow }));

};

render() {

return (

<button onClick={this.toggleShow} type="button">

Toggle Show

</button>

);

}

}

const Greeting = ({ greeting }) => <h1>{greeting}</h1>;

export default App;

2]

import React, { Component } from 'react';

class App extends Component {

constructor(props) {

super(props);

this.state = {

isShow: true,

};

}

toggleShow = () => {

this.setState(state => ({ isShow: !state.isShow }));

};

render() {

const greeting = 'Welcome to React';

return (

<div>

{this.state.isShow ? <Greeting greeting={greeting} /> : null}

<Button onClick={this.toggleShow} />

</div>

);

}

}

const Button = ({ onClick }) => (

<button onClick={onClick} type="button">

Toggle Show

</button>

);

const Greeting = ({ greeting }) => <h1>{greeting}</h1>;

export default App;

3]

import React, { Component } from 'react';

class App extends Component {

constructor(props) {

super(props);

this.state = {

isShow: true,

};

}

toggleShow = () => {

this.setState(state => ({ isShow: !state.isShow }));

};

render() {

const greeting = 'Welcome to React';

return (

<div>

<Greeting greeting={greeting} isShow={this.state.isShow} />

<Button onClick={this.toggleShow} />

</div>

);

}

}

const Button = ({ onClick }) => (

<button onClick={onClick} type="button">

Toggle Show

</button>

);

const Greeting = ({ greeting, isShow }) =>

isShow ? <h1>{greeting}</h1> : null;

export default App;

## **Props can be state, props, or derived properties**

import React, { Component } from 'react';

class App extends Component {

render() {

const greeting = {

subject: 'React',

description: 'Your component library for ...',

};

return (

<div>

<Greeting greeting={greeting} />

</div>

);

}

}

const Greeting = ({ greeting }) =>

<div>

<Title title={`Welcome to ${greeting.subject}`} />

<Description description={greeting.description} />

</div>

const Title = ({ title }) =>

<h1>{title}</h1>;

const Description = ({ description }) =>

<p>{description}</p>;

export default App;

## **React Props and Code Style**

import React, { Component } from 'react';

import logo from './logo.svg'

class App extends Component {

render() {

const greeting = {

subject: 'React',

description: 'Your component library for ...',

};

return (

<div>

<Greeting subject={greeting.subject} description={greeting.description} logo={logo} />

</div>

);

}

}

const Greeting = ({ subject, description, logo }) =>

...

So how to overcome this bad code style which is hard to read and maintain? One way would be passing the props with multiple indented lines to a component. The destructuring could follow the same rules:

import React, { Component } from 'react';

import logo from './logo.svg'

class App extends Component {

render() {

const greeting = {

subject: 'React',

description: 'Your component library for ...',

};

return (

<div>

<Greeting

subject={greeting.subject}

description={greeting.description}

logo={logo}

/>

</div>

);

}

}

const Greeting = ({

subject,

description,

logo,

}) =>

...

## **React ...props syntax**

class App extends Component {

render() {

const greeting = {

subject: 'React',

description: 'Your component library for ...',

};

return (

<div>

<Greeting {...greeting} />

</div>

);

}

}

const Greeting = ({ subject, description }) => (

<div>

<Title title={`Welcome to ${subject}`} />

<Description description={description} />

</div>

);

const Title = ({ title }) => <h1>{title}</h1>;

const Description = ({ description }) => <p>{description}</p>;

class App extends Component {

render() {

const greeting = {

subject: 'React',

description: 'Your component library for ...',

};

return (

<div>

<Greeting {...greeting} />

</div>

);

}

}

const Greeting = ({ subject, ...other }) => (

<div>

<Title title={`Welcome to ${subject}`} />

<Description {...other} />

</div>

);

const Title = ({ title }) => <h1>{title}</h1>;

const Description = ({ description }) => <p>{description}</p>;

## **How to pass Components as Props?**

const User = ({ user }) => (

<Profile user={user}>

<AvatarRound user={user} />

</Profile>

);

const Profile = ({ user, children }) => (

<div className="profile">

<div>{children}</div>

<div>

<p>{user.name}</p>

</div>

</div>

);

const AvatarRound = ({ user }) => (

<img className="round" alt="avatar" src={user.avatarUrl} />

);

const User = ({ user }) => (

<Profile

user={user}

avatar={<AvatarRound user={user} />}

biography={<BiographyFat user={user} />}

/>

);

const Profile = ({ user, avatar, biography }) => (

<div className="profile">

<div>{avatar}</div>

<div>

<p>{user.name}</p>

{biography}

</div>

</div>

);

const AvatarRound = ({ user }) => (

<img className="round" alt="avatar" src={user.avatarUrl} />

);

const BiographyFat = ({ user }) => (

<p className="fat">{user.biography}</p>

);

const AvatarSquare = ({ user }) => (

<img className="square" alt="avatar" src={user.avatarUrl} />

);

const BiographyItalic = ({ user }) => (

<p className="italic">{user.biography}</p>

);

## **Children as a Function**

const App = () => (

<div>

<h1>US Dollar to Euro:</h1>

<Amount render={amount => <Euro amount={amount} />} />

<h1>US Dollar to Pound:</h1>

<Amount render={amount => <Pound amount={amount} />} />

</div>

);

class Amount extends Component {

constructor(props) {

super(props);

this.state = {

amount: 0,

};

}

onIncrement = () => {

this.setState(state => ({ amount: state.amount + 1 }));

};

onDecrement = () => {

this.setState(state => ({ amount: state.amount - 1 }));

};

render() {

return (

<div>

<button type="button" onClick={this.onIncrement}>

+

</button>

<button type="button" onClick={this.onDecrement}>

-

</button>

<p>US Dollar: {this.state.amount}</p>

{this.props.render(this.state.amount)}

</div>

);

}

}

const Euro = ({ amount }) => <p>Euro: {amount \* 0.86}</p>;

const Pound = ({ amount }) => <p>Pound: {amount \* 0.76}</p>;

const App = () => (

<div>

<h1>US Dollar to Euro:</h1>

<Amount>{amount => <Euro amount={amount} />}</Amount>

<h1>US Dollar to Pound:</h1>

<Amount>{amount => <Pound amount={amount} />}</Amount>

</div>

);

class Amount extends Component {

constructor(props) {

super(props);

this.state = {

amount: 0,

};

}

onIncrement = () => {

this.setState(state => ({ amount: state.amount + 1 }));

};

onDecrement = () => {

this.setState(state => ({ amount: state.amount - 1 }));

};

render() {

return (

<div>

<button type="button" onClick={this.onIncrement}>

+

</button>

<button type="button" onClick={this.onDecrement}>

-

</button>

<p>US Dollar: {this.state.amount}</p>

{this.props.children(this.state.amount)}

</div>

);

}

}

const Euro = ({ amount }) => <p>Euro: {amount \* 0.86}</p>;

const Pound = ({ amount }) => <p>Pound: {amount \* 0.76}</p>;