import React from 'react';

import { StyleSheet, Text, View, TextInput } from 'react-native';

// เพื่อให้ง่ายเราจะสร้างคลาส MyTextInput อยู่ในไฟล์ App.js

class MyTextInput extends React.Component {

// ทำงานเมื่อ component ถูกสร้างขึ้น

// ใช้สำหรับกำหนดค่าเริ่มต้นให้ state

constructor(props) {

super(props);

this.state = {

value: '',

};

}

// ทำงานครั้งเดียวหลังจากสร้าง component ครั้งแรก

// ทำงานหลังจากเมธอท render

// เหมาะสำหรับใช้เรียก Backend API

componentDidMount() {

console.log('componentDidMount');

}

// ถูกเรียกใช้งานก่อนเมธอท render ทุกครั้ง

// ใช้ในกรณีที่ state ต้องเปลี่ยนตาม props ที่ได้รับเข้ามา

// เป็น static เมธอท

static getDerivedStateFromProps(nextProps, prevState) {

console.log('getDerivedStateFromProps');

// Return null หมายถึงไม่มีการอัพเดท state

return null;

}

// ถูกเรียกใช้งานเมื่อค่า props หรือ state เปลี่ยนแปลง

// โดยปกติเมธอท render ถูกเรียกใช้ทุกครั้งที่ props หรือ state เปลี่ยน

// ถ้า shouldComponentUpdate return false เมธอท render จะไม่ถูกเรียกใช้

shouldComponentUpdate(nextProps, nextState) {

console.log('shouldComponentUpdate');

console.log('nextProps : ' + JSON.stringify(nextProps));

console.log('nextState : ' + JSON.stringify(nextState));

return true;

}

// ถูกเรียกใช้งานเมื่อค่า props หรือ state เปลี่ยนแปลง

// ทำงานหลังเมธอท render

// เรามามารถเรียก setState() ใน componentDidUpdate ได้

// ใช้เรียก Backend API

componentDidUpdate(prevProps, prevState) {

console.log('componentDidUpdate');

console.log('prevProps : ' + JSON.stringify(prevProps));

}

// ทำงานก่อนที่คอมโพแนนท์ถูกทำลาย

componentWillUnmount() {

console.log('componentDidUpdate');

}

// ฟังก์ชั่น callback สำหรับจัดการอีเวนท์ onChangeText

handleChangeText = text => {

this.setState({

value: text,

});

};

render() {

console.log('render');

return (

<View>

{/\* อ่านค่า label จาก props และแสดงเป็น text \*/}

<Text>{this.props.label}</Text>

<TextInput

onChangeText={this.handleChangeText}

style={styles.textInput}

/>

{/\* แสดงค่าจาก state ที่ได้รับจาก TextInput \*/}

<Text>Input Value : {this.state.value}</Text>

</View>

);

}

}

class App extends React.Component {

render() {

return (

<View style={styles.container}>

{/\* คอมโพแนนท์ MyTextInput มี props ชื่อว่า label \*/}

<MyTextInput label="My Text Input" />

</View>

);

}

}

const styles = StyleSheet.create({

container: {

flex: 1,

flexDirection: 'column',

justifyContent: 'center',

alignItems: 'center',

},

textInput: {

borderStyle: 'solid',

borderWidth: 1,

borderRadius: 3,

width: 200,

height: 40,

padding: 5,

marginTop: 7,

marginBottom: 7,

},

});

export default App;