Props và State

## **Props**

props là viết tắt của Properties. Một điều mà bạn cần phải nhớ khi sử dụng props đó là không bao giờ nên thay đổi giá trị của nó, hay nói cách khác, đây là một dữ liệu ****immutable****.

Các component nhận props từ component cha. Bạn không được thay đổi giá trị của props trong các component này mà chỉ được phép đọc giá trị ra thôi. Trong React thì dữ liệu sẽ đi theo một chiều, có nghĩa là từ component cha => các component con.

Bạn có thể tạo ra component sử dụng props. Ý tưởng của props đó là việc trừu tượng hoá các component để có thể sử dụng được ở nhiều chỗ khác nhau trong app. Hãy xem một ví dụ đơn giản sau:

// Parentexport default class ScreenOne extends React.Component {

render () {

return (

<View>

<Heading message={'Custom Heading for Screen One'}/>

</View>

)

}}

// Child componentexport default class Heading extends React.Component {

render () {

return (

<View>

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

</View>

)

}}

Heading.propTypes = {

message: PropTypes.string}

Heading.defaultProps = {

message: 'Heading One'}

## **State**

state thì hoạt động khác với props. state là dữ liệu nội bộ của một Component, trong khi props là dữ liệu được truyền cho Component. Chính vì vậy chúng ta hoàn toàn có thể thay đổi state, và coi nó là một kiểu dữ liệu ****mutable****.

Tuy vậy, hãy nhớ rằng đừng bao giờ thay đổi trực tiếp biến this.state. Thay vào đó hãy dùng hàm setState để cập nhật giá trị. Sở dĩ chúng ta cần dùng hàm này là do nó sẽ kích hoạt việc render lại component và tất cả component con nằm trong nó, còn thay đổi this.state thì không. Còn một vấn đề nữa, đó là setState chạy bất đồng bộ, vậy nên nếu bạn tiến hành đọc ra giá trị state ngay sau khi setState thì chưa chắc giá trị sẽ được update lên mới nhất đâu.

Ví dụ nếu bạn cần phải điền một cái form với nhiều text input, thì mỗi text input sẽ giữ đoạn text mà bạn gõ vào trong state của nó. Nếu bạn gõ một đoạn text khác vào, state của text input đó sẽ thay đổi và kích hoạt việc render lại component, và text mới của bạn sẽ được vẽ lên.

class Form extends React.Component {

constructor (props) {

super(props)

this.state = {

input: ''

}

}

handleChangeInput = (text) => {

this.setState({ input: text })

}

render () {

const { input } = this.state

return (

<View>

<TextInput style={{height: 40, borderColor: 'gray', borderWidth: 1}}

onChangeText={this.handleChangeInput}

value={input}

/>

</View>

)

}

}