**Tên Nhóm: NCP (Nhóm 1)**

Thành Viên:

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ Và Tên | Email | SDT |
| 1 | Lê Đình Cường | [ledinhcuongse@gmail.com](mailto:ledinhcuongse@gmail.com) | 0359752007 |
| 2 | Đoàn Thị Yến Nhi | [Doanthiyennhi1001@gmail.com](mailto:Doanthiyennhi1001@gmail.com) | 0388102525 |
| 3 | Trần Đức Phương | [phuongtranitnl@gmail.com](mailto:phuongtranitnl@gmail.com) | 0937320311 |
| 4 | Bùi Đức Cơ | [Rongtrang137@gmail.com](mailto:Rongtrang137@gmail.com) | 01889361179 |

Email nhóm: ncpdidong3@gmail.com

Pass: Aa359752007

Github: <https://github.com/phuongtranitnl/didong3>

Đề tài: Xây dựng ứng dụng ghi chú

Tuần 3.1

Ngày 12/11/2018

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

**1 Tìm hiểu về Button**

## Button

Tùy vào hệ điều hành dùng button này sẽ hiển thị với giao diện khác nhau.

VD: Sử dụng button

import { Button } from 'react-native';

...

<Button

onPress={onPressLearnMore}

title="Learn More"

color="#841584"

accessibilityLabel="Learn more about this purple button"

/>

**Các props (Các thuộc tính đó)**

- onPress

// Gọi phương thức khi người dùng chạm vào button (Yêu cầu)

- title

// Hiển thị tên nút (Yêu cầu)

- accessibilityLabel

// Văn bản hiện thị các trợ năng (Không yêu cầu)

- color

// Màu sắc của chử trong ios nhưng đây là màu nền của phím trong android

- disabled

// Tắt tương tác cho nút

- testID

// Được sử dụng để truy cập view này

- hasTVPreferredFocus

// Chỉ dành cho nền tảng ios

**2 Tìm hiểu về Text**

Là một thành phần để hiện thị text. Nó cũng xử lý lồng nhau, kiểu sự kiện touch handling.

VD

import React, { Component } from 'react';

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

export default class **TextInANest** extends **Component** {

constructor(props) {

super(props);

this.state = {

titleText: "Bird's Nest",

bodyText: 'This is not really a bird nest.'

};

}

**render**() {

return (

<**Text** *style*={styles.baseText}>

<**Text** *style*={styles.titleText} *onPress*={this.onPressTitle}>

{this.state.titleText}{'\n'}{'\n'}

</**Text**>

<**Text** *numberOfLines*={5}>

{this.state.bodyText}

</**Text**>

</**Text**>

);

}

}

const styles = **StyleSheet**.**create**({

baseText: {

fontFamily: 'Cochin',

},

titleText: {

fontSize: 20,

fontWeight: 'bold',

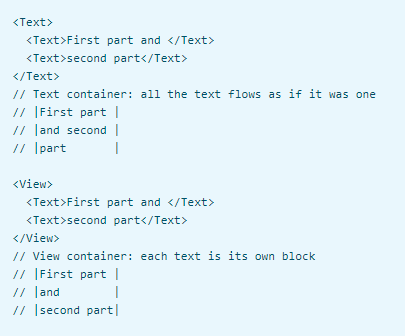
},

});

*// skip this line if using Create React Native App*

AppRegistry.**registerComponent**('TextInANest', () => TextInANest);

**Các thành phần trong Text**



**Các Props (Thuộc tính đó)**

- selectable

// Để cho phép người dùng chọn văn bản, để sử dụng chức năng sao chép và dán. (Không yêu cầu) giá trị là kiểu bool

- accessibilityHint (String)

// Gợi ý trợ năng giúp người dùng hiểu điều gỉ xảy ra khi họ thực hiện hành động trên phần tử (Không yêu cầu)

- accessibilityLabel (node)

// Ghi dè văn bản được đọc bởi trình đọc màn hình khi người dùng tương tác với phần tử. (Không yêu cầu)

- accessible (Bool)

Khi được đặt thành true cho biết chế độ xem là yếu tố trợ năng. Giá trị mặc định cho phần tử văn bản là đúng (Không yêu cầu)

- ellipsizeMode

- TYPE

- nativeID (String)

Được sử dụng để định vị chế độ xem này từ mã gốc (Không yeu cầu)

- numberOfLines (Số)

- onLayout

{nativeEvent: {layout: {x, y, width, height}}}

- onLongPress

// Gọi sự kiện khi nhấn giữ gọi một phương thức trong hàm hoặc một arrow function

- onPress

// Gọi sự kiện khi nhấn

VD

onPress={() => console.log('1st')}

- pressRetentionOffset

- allowFontScaling (bool)

- style (style)

// Định dạng các css trong đây đó

+ color

+fontStyle  enum('normal', 'italic')

+ fontWeight enum('normal', 'bold', '100', '200', '300', '400', '500', '600', '700', '800', '900')

+ lineHeight

+ textAlign enum('auto', 'left', 'right', 'center', 'justify')

+ textDecorationLine enum('none', 'underline', 'line-through', 'underline line-through')

+ fontFamily

+ letterSpacing (Số)

- testID (String)

// Sử dụng cho cục bộ muốn gọi view này

- disabled (Bool)

- selectionColor (color)

// Màu highlight của text

**3 Tìm hiểu về TextInput**

Dùng để nhập dữ liệu

VD

import React, { Component } from 'react';

import { AppRegistry, TextInput } from 'react-native';

export default class **UselessTextInput** extends **Component** {

constructor(props) {

super(props);

this.state = { text: 'Useless Placeholder' };

}

**render**() {

return (

<**TextInput**

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

*onChangeText*={(text) => this.**setState**({text})}

*value*={this.state.text}

/>

);

}

}

*// skip this line if using Create React Native App*

AppRegistry.**registerComponent**('AwesomeProject', () => UselessTextInput);

**Các Props**

- allowFontScaling (bool)

// Chỉ định xem font chữ có nên mở rộng hỗ trợ cài đặt trợ năng hay không. mặc định là true

- autoCapitalize

// Có thể yêu cầu text input tự động viết hoa các ký tự nhất định.

- autoCorrect (bool)

// Mặc định đã được bật

- autoFocus (bool)

- blurOnSubmit (bool)

- clearTextOnFocus (bool) (ios)

- TYPE (IOS)

- defautlValue (String)

- disableFullscreenUI (bool)

- keyboardType

// Quy định kiểu hiện thỉ của bàn phím

+ default

+ nuber-pad

+ decimal-pad

+ numeric

+ email-address

+ phone-pad

- maxLength (số)

- multiline (bool)

- numberOfLines

- onBlur (function)

- onChange (function)

// Gọi phương thức khi text bắt đầu thay đổi

- onChangeText (Function)

// Gọi lại khi văn bản đầu thay đổi

- onEndEditing (function)

// Dược gọi khi nhập xong

- onFocus (function)

// Gọi sự kiện chọn input

- onKeyPress (function)

- onScroll (Function)

- onSubmitEditing (Function)

- placeholder (String)

- placeholderTextColor (color)

- value (String)

- inlineImageLeft

Dùng để hiển thị hình ảnh bên trái text box

<TextInput

inlineImageLeft='search\_icon'

/>

**Các methods**

- clear();

- isFocused();

**4 Tìm hiểu về checkbok**

Đây là thành phần không có sẵn trong thư viện mặc định khi tạo ứng dụng mà ta phải import từ bên ngoài

import { CheckBox } from 'react-native-elements'

<CheckBox

title='Click Here'

checked={this.state.checked}

/>

<CheckBox

center

title='Click Here'

checked={this.state.checked}

/>

<CheckBox

center

title='Click Here'

checkedIcon='dot-circle-o'

uncheckedIcon='circle-o'

checked={this.state.checked}

/>

<CheckBox

center

title='Click Here to Remove This Item'

iconRight

iconType='material'

checkedIcon='clear'

uncheckedIcon='add'

checkedColor='red'

checked={this.state.checked}

/>

<CheckBox

checkedIcon={<Image source={**require**('../checked.png') />}

uncheckedIcon={<Image source={**require**('../unchecked.png') />}

checked={this.state.checked}

onPress={() => this.**setState**({checked: !this.state.checked})}

/>

**Cac props (Thuộc tính của thẻ)**

- iconType (String)

// Có thể thêm biểu tượng

- component

// Chỉ định thành phần react cho nút chính.

- checked (bool)

// Kiểm tra thẻ đã được check chưa

- size (Số)

// Kích thước của checkbox mặc định là 24

- iconRight (bool)

// Duy chuyển icon check sang bên phải của check mặc định giá trị này là false

- right (bool)

Căn chình checkbox sang bên phải

- center

// Căn chình checkbox sang giữa

- title (string)

// Đây là tiêu đề của textbox đây là thuộc tính yêu cầu

- containerStyle (object)

// Kiểu của thành phần chính

- textStyle

// Kiểu của chữ

- onLongPress (function)

Khi nhấn giữ vào check box

- onLongIconPress (function)

Khi nhấn giữ lâu vào check box

- onPress (function)

Khi nhấn vào check box

- onIconPress (function)

khi nhấn vào check box đây là thuộc tính yêu cầu

- checkedIcon

Icon mặc định khi check box chưa được nhấn chúng ta có thể dùng icon từ font awesome icon để thay thê và icon mặc định của giá trị này là check-square-o

- uncheckedIcon

Icon khi ta check vào nó mặc định là square-o

- checkedColor

Màu mặc định khi được check. Mặc định là màu green

- fontFamily

// Chỉ định font chữ

**5 Tìm hiểu về radiobutton**

Đây cũng là thành phần mà ta phải cài đặt từ bên ngoài

**Để thêm radio button ta phải chạy câu lệnh sau**

npm i radio-button-react-native --save

Sau đó import nó vào đồ án

**import** RadioButton **from** 'radio-button-react-native';

Cấu trúc của radio button sẽ giống như sau

render(){

**return**(

**<**View**>**

**<**RadioButton currentValue**=**{this.state.value} value**=**{0} onPress**=**{this.handleOnPress.bind(this)}**>**

**<**Text**>**Button1**</**Text**>**

**</**RadioButton**>**

**<**RadioButton currentValue**=**{this.state.value} value**=**{1} onPress**=**{this.handleOnPress.bind(this)}**>**

**<**Text**>**Button2**</**Text**>**

**</**RadioButton**>**

**<**RadioButton currentValue**=**{this.state.value} value**=**{2} onPress**=**{this.handleOnPress.bind(this)}**>**

**<**Text**>**Button3**</**Text**>**

**</**RadioButton**>**

**</**View**>**

    );

**Cơ bản của một Radio Button**

**<**RadioForm

  radio\_props**=**{radio\_props}

  initial**=**{0}

  onPress**=**{(value) => {this.setState({value**:**value})}}

**/>**

**Đầy đủ hơn thì**

**<**RadioForm

  radio\_props**=**{this.state.types}

  initial**=**{0}

  formHorizontal**=**{false}

  labelHorizontal**=**{true}

  buttonColor**=**{'#2196f3'}

  animation**=**{true}

  onPress**=**{(value) => {this.setState({value**:**value})}}

**/>**

**Cac Props của RadioForm**

- radio\_props

// Giá trị của radio button này giá trị truyền dùng làm nhãn là một mãng.

- onPress (function)

Gọi phương thức này khi radio button được check.

- buttonColor

Màu sắc của radion button

**<**RadioForm

  radio\_props**=**{radio\_props}

  initial**=**{0}

  buttonColor**=**{'#50C900'}

**/>**



- labelColor

Màu sắc của nhãn của radio button

**<**RadioForm

  radio\_props**=**{radio\_props}

  initial**=**{0}

  labelColor**=**{'#50C900'}

**/>**

- formHorizontal (bool)

Giúp xác định chiều hiển thị của radio button mặc định giá trị này là false nên nó sẽ hiển thị ngang còn khi giá trị là true thì nó sẽ hiển thị dọc

**<**RadioForm

  radio\_props**=**{radio\_props}

  initial**=**{0}

  labelHorizontal**=**{false}

**/>**





- animation

Đương nhiên đây là hiệu ứng khi ta click vào radion button rồi mặc định giá trị này là true

- accessible

Cho biết khả năng truy cập riêng lẽ cho từng thành phần nút radio và nút radio riêng lẽ

- accessibilityLabel

- testID

Được sử dụng để thiết lập testId cho đầu vào nút radio và thành nhấn của nó

= [testID]Label[index].

**Props đối với RadioButton Component và RadioButtonIntput, RadioButtonLabel**

<https://www.npmjs.com/package/react-native-simple-radio-button>