

DATABINDING -
TOUCHABLEOPACITY -
STATE - TEXTINPUT

DataBinding

- Ta có thể chèn biến hoặc hàm trực tiếp vào các component bằng cách đặt trong cặp {}

```
import { View, Text } from 'react-native'
import React from 'react'

export default function DataBinding() {
  let name="John Walker"
  let a=5, b=9
  return (
    <View>
      <Text>DataBinding</Text>
      <Text>Hello {name}</Text>
      <Text>5+9={a+b}</Text>
    </View>
  )
}
```

TouchableOpacity

- TouchableOpacity dùng để lắng nghe sự kiện onPress
- Để tạo button: dùng Text bên trong TouchableOpacity

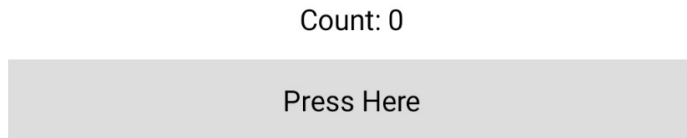
```
<TouchableOpacity>  
  <Text>Press Here</Text>  
</TouchableOpacity>
```

- Cú pháp khai báo sự kiện onPress trong TouchableOpacity:

```
<TouchableOpacity onPress={hàm xử lý sự kiện}>  
  <Text>Press Here</Text>  
</TouchableOpacity>
```

TouchableOpacity

- Giả sử muốn tạo giao diện đếm số lần click button:



- Khai báo biến `count=0`
- Hàm xử lý sự kiện `onPress`: tăng `count` lên 1

```
const onPress={()=>{  
    count=count+1  
    console.log(count)  
}}
```
- Kết quả: trên giao diện `Count` vẫn bằng 0, nhưng trong Console `count` có giá trị 1, 2, ...

```
import { View, Text, TouchableOpacity, StyleSheet } from 'react-native'  
import React from 'react'  
  
export default function DemoTouchableOpacity() {  
    let count=0  
    const onPress={()=>{  
        count=count+1  
        console.log(count)  
    }}  
    return (  
        <View style={styles.container}>  
            <View style={styles.countContainer}>  
                <Text>Count: {count}</Text>  
            </View>  
            <TouchableOpacity  
                style={styles.button}  
                onPress={onPress}  
            >  
                <Text>Press Here</Text>  
            </TouchableOpacity>  
        </View>  
    );  
};
```

State

- State là dữ liệu của component
- Mỗi khi state thay đổi -> giao diện sẽ được re-render (vẽ lại)
- Đối với function component: muốn sử dụng state phải import hook `useState` của React

```
import React, { useState } from "react";
```

- Cú pháp khai báo state:

```
const [count, setCount] = useState(0);
```

- `count`: tên biến (còn gọi là tên state)
 - `setCount`: tên phương thức để thay đổi giá trị của biến (state)
 - `0`: giá trị khởi tạo của biến
- Nếu có nhiều biến thì mỗi biến phải được khai báo bằng 1 mệnh đề như trên

State

```
import { View, Text, TouchableOpacity, StyleSheet } from 'react-native'
import React, {useState} from 'react'

export default function DemoState() {
  const[count, setCount]=useState(0)
  const onPress = () => setCount(count => count + 1);
  return (
    <View style={styles.container}>
      <View style={styles.countContainer}>
        <Text>Count: {count}</Text>
      </View>
      <TouchableOpacity
        style={styles.button}
        onPress={onPress}
      >
        <Text>Press Here</Text>
      </TouchableOpacity>
    </View>
  )
};
```

TextInput

- TextInput là component cho phép nhập dữ liệu
- Các thuộc tính:
 - placeholder: chuỗi gợi ý nội dung nhập vào
 - placeholderTextColor: màu của chuỗi gợi ý nội dung nhập vào
 - textAlign: canh lề nội dung bên trong TextInput
 - value: giá trị được gán vào TextInput
 - defaultValue: giá trị mặc định trong TextInput
 - securityTextEntry: nếu gán bằng true -> nhập dạng mật khẩu

TextInput

- Lấy dữ liệu trong TextInput: sử dụng state và sự kiện onChangeText
 - Khai báo state để lưu giá trị đang có trong TextInput:
`const[text, setText] = useState('')`
 - Sự kiện onChangeText: cho phép truyền vào 1 hàm với 1 tham số kiểu string đại diện cho chuỗi đang có trong TextInput. Hàm này sẽ gọi phương thức `setState` để gán: state lưu giá trị của TextInput = tham số

- Ví dụ:

```
const _onChangeText = (d)=>{setText(d)}
```

```
<TextInput
```

```
  placeholder='Enter any string'
```

```
  defaultValue='abc'
```

```
  onChangeText={_onChangeText} //hoặc onChangeText={(d)=>{setText(d)}}
```

```
></TextInput>
```


Excercise 1 & 2

