06016323 Mobile Device Programming

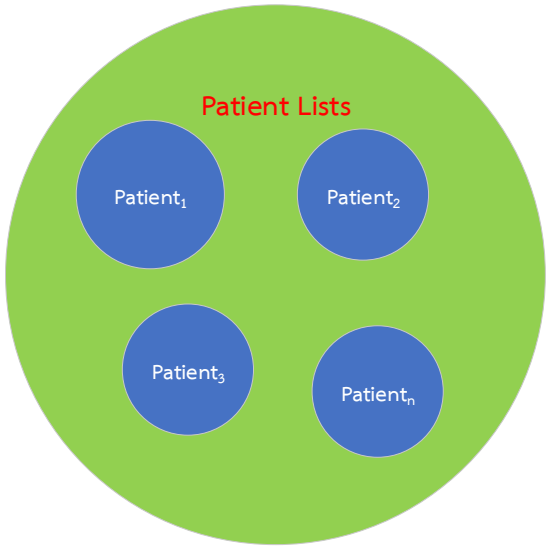
CHAPTER 3 : PROPERTIES AND STATE

การทำงานของ props

- ***props***

Props หรือ properties คือ Component's **configuration,** ที่เราจะสามารถกำหนดหรือไม่กำหนดก็ได้ ภายในตัวโปรแกรม หากมีการกำหนดค่าใน props ตัวแปรที่มีลักษณะ props สามารถส่งค่าของข้อมูลจาก ParentComponent -> ChildComponent ได้ ซึ่งการส่งผ่านค่าจะตามลักษณะของข้อมูล เช่น Numeric String Function หรือ Objects ได้

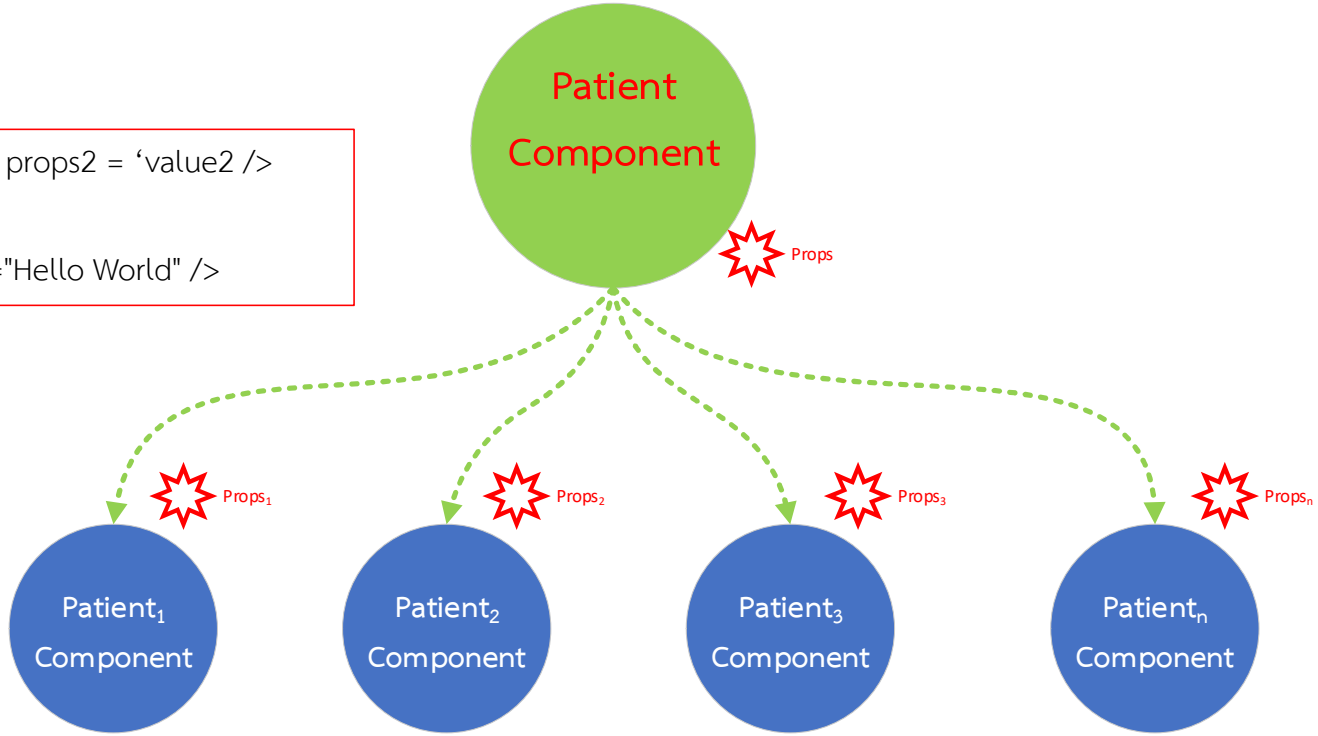
- หากตัวแปรถูกกำหนดแบบ props ในส่วนของ Component ***จะไม่สามารถทำการเปลี่ยนแปลงค่าได้*** นอกจากไปเปลี่ยนที่ตอนกำหนดค่าตัวแปรนั้นครั้งแรก เปรียบเสือนการกำหนด ตัวแปร แบบ Global หรือ Public



Pass by Value in Props

- <ChildComponent props1 = ‘value1’ props2 = ‘value2 />
  - ตัวอย่าง

    <ChildComponent message="Hello World" />



```
class ParentComponent extends React.Component {
    render() {
        return <ChildComponent message="Hello World" />
    }
}
class ChildComponent extends React.Component {
    render() {
        return <p>And then I said, “{this.props.message}”</p>
    }
}
```

```
import React from 'react';

function App() {
  const greeting = 'Hello Function Component.';

  return <Headline value={greeting} />;
}

function Headline(props) {
  return <h1>{props.value}</h1>;
}

export default App;
```

State

- การเปลี่ยน สถานะของ ตัวแปร เราสามารถทำการเปลี่ยนแปลงได้จากการให้ค่าหรือ มีกำหนดค่าจากฟังก์ชันอื่น ๆ ที่เข้ามาใช้ หรือ เรียกใช้งาน Component ที่ถูกสร้างขึ้น
- การเปลี่ยนสถานะ ของ Component ของ React มี 2 รูปแบบ
  - State with Function Components
  - State with Class Components

การทำงานของ state

❖ *state*

ค่าของ State มักจะมีค่าเริ่มต้น เป็น true, 0 เป็นต้น เมื่อ component เรียกใช้งานส่งผลทำให้ ค่าของตัวแปรที่เป็น state ถูกแทนค่าด้วยค่า default และเมื่อค่าของตัวแปรที่เป็น state มีการเปลี่ยนแปลงค่าจะผู้ใช้ หรือ เหตุการณ์ต่าง ๆ ค่าที่อยู่ภายในจะถูกเปลี่ยนแปลงตามทันที (It's a serializable* representation of one point in time—a snapshot.) ไม่ว่าค่าของ state จะอยู่ในฟังก์ชันใด ๆ ใน component เดียวกัน จึงกล่าวได้ว่า ตัวแปรที่เป็น state เปรียบเสมือน private variable (Local variable) .

```
class MyComponent extends React.Component {
        handleClick = e => {
                this.setState({ clicked: true })
        }
render() {
         return ( <a href="#" onClick={this.handleClick}> Click me </a>
                    )
                }
}
```

Identified State

- กำหนด ภายใต้ ฟังก์ชัน หรือ Component

```
import React, {Component} from 'react';

class App extends React.Component{

  state = { variable1: value , variable2: value, ..... , variablen: value }
```

ตัวอย่าง : state = { todo: “บรรทัดแรกของการบันทึก”}

state = { text: “บรรทัดแรกของการบันทึก”}

```
render() {  component }
```

- สามารถปรับปรุงค่า การแสดงผล และประมวลผลใหม่ เมื่อมีการเปลี่ยนแปลง ที่เกิดขึ้นภายใน Component ที่กำหนด
  - onChangeText = {(text)=> this.setState({todo : text})}
  - onChangeText = {(text)=> this.setState({text})}

การกำหนด State แบบมีโครงสร้าง

```
state = {
        PatientName: ‘Jirayu',
        description: {
                Age: 30,
                Hight: 170,
                Weight: 70,
                Nationality: ‘Thai'
            },
       habit:[‘driving ', 'sleep ','Good in eating']
   }
```

การแก้ไขค่าของ State

- Class Component
  - setState() เป็นเมธอดสำหรับการแก้ไข ปรับปรุงค่าของตัวแปรใน State ที่ประกาศใน Class Component เมื่อ State มีการเปลี่ยนแปลง Component จะทำการ re-rendering.
- Function Component
  - useState() เป็นเมธอดสำหรับการกำหนดและแก้ไขค่าของ State ที่ถูกประกาศไว้ใน Function Component เมื่อ State มีการเปลี่ยนแปลง Component จะทำการ re-rendering.

```
class CounterComponent extends React.Component {
    constructor() {
          super()
          this.state = {
             count: 0,
          }
     }

    handleClick = e => {
          this.setState({
               count: this.state.count + 1,
          })
    }

render() {
     return (
          <div>
               <h1>Current Count: {this.state.count}</h1>
               <a href="#" onClick={this.handleClick}>
                 Increment
               </a>;
          </div>
     )
   }
}
```

Display the state

- สำหรับแสดงผลทันทีเมื่อมีการป้อนข้อมูลลง TextInput เราต้องใช้ *onChangeText event* เพื่อให้การแสดงผลนั้นเป็นปัจจุบัน เมื่อผู้ใช้มีการป้อนข้อมูลหรือค่าเข้ามา ดังนี้

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

- หากว่า ต้องการให้การป้อนข้อมูลทุกครั้ง ต้องแสดงผลหลังจากที่มี event ใด ๆ เข้ามากระทบ เช่น onClick ถึงจะให้แสดงผล ควรต้องสร้างฟังก์ชันขึ้นมาภายใน Componnet นั้น ๆ เพื่อจัดการเหตุการณ์ ภายใน TextInput เช่น

```
addText =( ) => {
        this.setState({todo: this.state.text});

}
```

ส่วนใน <Button> ทำการเพิ่ม onPress event เพื่อทำการรับค่าจากการคลิ๊กของเมาท์

useState()

- ถ้าต้องการใช้ หรือ มีการกำหนด state ใน function components ทำได้โดยใช้ useState
  - useState(initValue) ต้องมีการกำหนดค่าเริ่มต้น
  - useState() จะ return เป็น array 2 elements คือ ชื่อตัวแปร และฟังก์ชันสำหรับการเปลี่ยนแปลงค่าตามลำดับ (ตั้งชื่อแบบใดก็ได้)

```
import React, { useState, useEffect } from 'react';

function FormApplication() {
  const [ firstName, setFirstName ] = useState('Steve')
  const [ lastName, setLastName ] = useState('Jobs')

   function handleChangeLastName(e) {
             setLastName(e.target.value)
   }

   return <div>
     <input type="text" value={ firstName } onChange={ e => setFirstName(e.target.value) } />
     <input type="text" value={ lastName } onChange={ handleChangeLastName } />

</div> }
```

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

const App = () => {
  return <Headline />;
};

const Headline = () => {
  const [greeting, setGreeting] = useState(
    'Hello Function Component.'
  );

  const [isShow, setShow] = useState(true)

  return (
    <div>
      <h1 style={{ display: isShow ? 'block': 'none'}}>{greeting}</h1>

      <input
        type="text"
        value={greeting}
        onChange={event => setGreeting(event.target.value)}
      />
      <br />
      <button onClick={event => setShow(!isShow)}>Show or Hide</button>
    </div>
  );
};

export default App;
```

State with Function Components

```
import React, { useState } from "react";
import { Button, Text, View } from "react-native";

const Koala = (props) => {
  const [isHungry, setIsHungry] = useState(true);

  return (
    <View>
      <Text>
        I am {props.name}, and I am {isHungry ? "hungry" : "full"}!
      </Text>
      <Button
        onPress={() => {
          setIsHungry(false);
        }}
        disabled={!isHungry}
        title={isHungry ? "Pour me some milk, please!" : "Thank you!"}
      />
    </View>
  );
}
```

```
const Cafe = () => {
  return (
    <>
      <Koala name="Munkustrap" />
      <Koala name="Spot" />
    </>
  );
}

export default Cafe;
```

คุณสมบัติร่วมของ props และ state

- ทั้ง *props* และ *state* สามารถเขียนโดย **JS objects**
- ทั้ง *props* และ *state* จะทำการเปลี่ยนแปลงหรือปรับปรุงค่าของตัวแปร ด้วย trigger ที่อยู่ในฟังก์ชัน **render update**
- ทั้ง *props* และ *state* มีคุณสมบัติในการกำหนดการแสดงผลหรือค่าของตัวแปร (**deterministic**) ถ้า component มีการแสดงค่าภายในแตกต่างกัน อาจจะแสดงสะท้อนได้ว่า เรากำหนดค่าบางอย่างผิดไปในส่วนของ *props* and *state*

ความแตกต่างระหว่าง State กับ Props

หัวข้อ	Properties	state
การเขียนโปรแกรม	JavaScript	JavaScript
การส่งผ่านค่า	ส่งผ่านโดย Component (เหมือนกับการส่งผ่านแบบ Function Parameter)	ถูกจัดการโดย Component (เหมือนกับการส่งค่าตัวแปรจากการกำหนดค่าด้วยฟังก์ชัน)

การเปลี่ยนจาก props และ state

	Props	***state***
Can get initial value from parent Component?	Yes	Yes
Can be changed by parent Component?	Yes	No
Can set default values inside Component?*	Yes	Yes
Can change inside Component?	No	Yes
Can set initial value for child Components?	Yes	Yes
Can change in child Components?	Yes	No

Array

- การเก็บค่าของตัวแปร หลายค่า ในตัวแปรเดียวกัน ดังนั้น หากต้องการให้ทำการเพิ่มหรือลบ จำเป็นต้องใช้หลักการการจัดการอาเรย์ เข้ามาประยุกต์ใช้งาน อีกทั้ง การเพิ่มเติมหรือลบค่าของข้อมูล บางครั้งจำเป็นต้องใช้อัลกอริทึม List , Push, Pop หรือ อื่น ๆ เป็นต้น

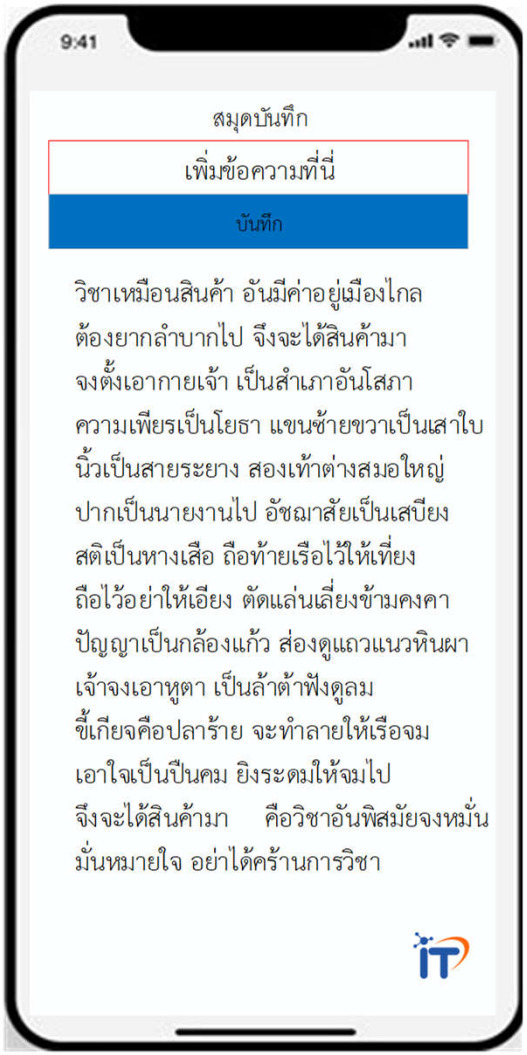
- การประกาศค่าตัวแปรที่เป็นอาเรย์

  <variable name> : [ ]

- การเรียกค่าจากอาเรย์ ใช้ props map เช่น

```
class App extends React.Component{
    state = {
        text: "",
        todo: [ ]
}
```

```
renderTodos = () =>{
    return this.state.todo.map(t=>{
      return (
        <TouchableOpacity key={t}>
            <Text
                style={styles.todo}
                onPress={()=>{this.deleteTodo(t)}}
                >{t}</Text>
        </TouchableOpacity>
      )
 })
}
```


9:41
สมุดบันทึก
เพิ่มข้อความที่นี่
บันทึก
วิชาเหมือนสินค้า อันมีค่าอยู่เมืองไกล
ต้องยากลำบากไป จึงจะได้สินค้ามา
จงตั้งเอากายเจ้า เป็นสำเภาอันโสภา
ความเพียรเป็นโยธา แขนซ้ายขวาเป็นเสาใบ
นิ้วเป็นสายระยาง สองเท้าต่างสมอใหญ่
ปากเป็นนายงานไป อัชฌาสัยเป็นเสบียง
สติเป็นหางเสือ ถือท้ายเรือไว้ให้เที่ยง
ถือไว้อย่าให้เอียง ตัดแล่นเลี่ยงข้ามคงคา
ปัญญาเป็นกล้องแก้ว ส่องดูแถวแนวหินผา
เจ้าจงเอาหูตา เป็นล้าต้าฟังดูลม
ขี้เกียจคือปลาร้าย จะทำลายให้เรือจม
เอาใจเป็นปืนคม ยิงระดมให้จมไป
จึงจะได้สินค้ามา คือวิชาอันพิสมัยจงหมั่น
มั่นหมายใจ อย่าได้คร้านการวิชา