

ทบทวน Components, props,
state ใน React

Component

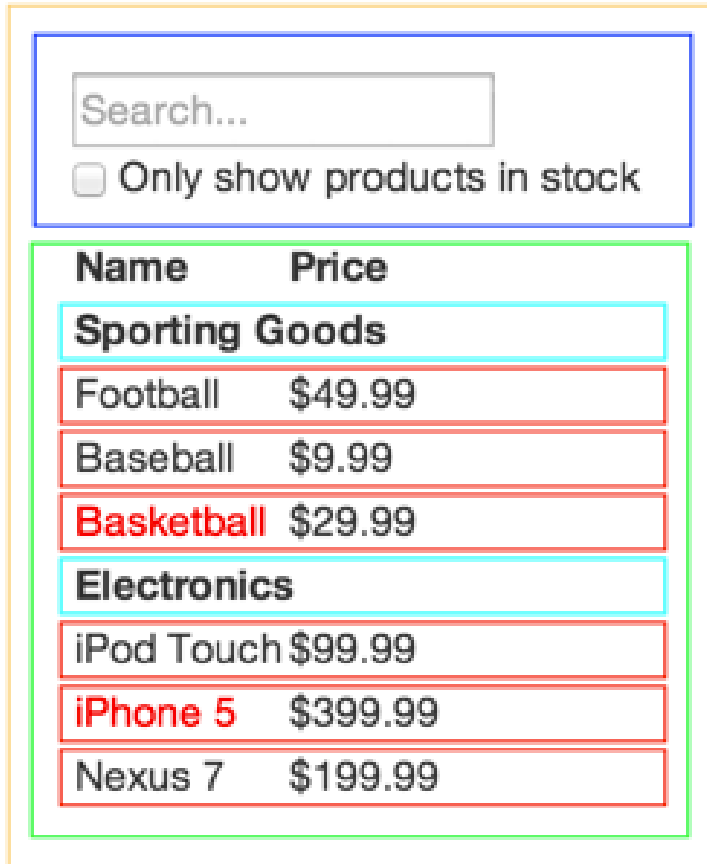
- แนวคิดของ **Component** คือการเขียนองค์ประกอบของ **View** ให้ย่อยที่สุดเท่าที่จะย่อยได้ และเน้นให้ **Reusable** ได้เยอะ
- ตัวอย่างเช่น รูปด้านล่าง ตัวอย่าง **classic** จาก **Facebook**

Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

ที่มา: <https://microbenz.in.th/>

ที่มา: <https://facebook.github.io/react/img/blog/thinking-in-react-components.png>

Component



Search...

☐ Only show products in stock

Name	Price
Sporting Goods	
Football	\$49.99
Baseball	\$9.99
Basketball	\$29.99
Electronics	
iPod Touch	\$99.99
iPhone 5	\$399.99
Nexus 7	\$199.99

เราสามารถแบ่งหน้าจอนี้ออกมาได้ 5 Component ด้วยกัน คือ

- ProductRow (สีแดง): แสดงชื่อสินค้า และราคา
- ProductCategoryRow (สีฟ้าอ่อน): แสดงชื่อประเภทสินค้า
- ProductTable (สีเขียว): แสดงรายการสินค้า (ซึ่งจะประกอบด้วย ProductCategoryRow + ProductRow)
- SearchBar (สีน้ำเงิน): แสดง Search Box ที่ filter ข้อมูลตาราง พร้อมปุ่ม Option ที่จะให้แสดงเฉพาะสินค้าที่มีใน stock
- FilterableProductTable (สีส้ม): แสดงข้อมูลสินค้า ที่เราสามารถ filter ข้อมูลในตารางได้ ด้วยการพิมพ์ใน search box (เป็นคอนโทรลทุกอันทั้งหมด)

Component

- ซึ่งการแบ่ง **Component** จะมีข้อดีที่ว่า
 - มีการแบ่งส่วนกันเขียน แยกส่วนกันชัดเจน
 - สามารถ **reuse component** ไปใช้ที่อื่นได้ หากต้องการอะไรที่คล้ายๆ กัน
 - แก้ไขง่าย เช่น ถ้าเราอยากให้แสดงจำนวนสินค้าเพิ่ม เราก็แก้ไขแค่ **ProductRow** อันเดียวพอ
- โดย **Component** ใน **React** นั้นจะเขียนได้หลายแบบมาก แต่ที่ใช้กันเยอะมากและเป็นมาตรฐานสำหรับมือใหม่ คือ **extends** คลาส **Component** ของ **React** ดังรูป

```
class HelloComponent extends React.Component {  
  render() {  
    return <p>Hello World!<p>;  
  }  
}
```

โครงสร้างของคอมโพเนนต์แบบคลาส (Class Component)

- คอมโพเนนต์ที่มีอินพุต คือ **props**
- คอมโพเนนต์แบบคลาส สามารถเก็บ **state** และเราสามารถเปลี่ยนค่า **state** ด้วย เมธอด **this.setState()**
- คอมโพเนนต์แบบคลาส มีเมธอด **render()** ที่ **return JSX** อิลิเมนต์
- เราสามารถ **initial state** ได้ใน **constructor**

- ตัวอย่างคอมโพเนนท์ **MyTextInput** มีค่าอินพุต **props** เป็น **label** และมี **state** สำหรับเก็บค่าอินพุตที่กรอกเข้ามาทาง **TextInput**

```
class MyTextInput extends React.Component {  
  
  // ทำงานเมื่อ component ถูกสร้างขึ้น  
  // ใช้สำหรับกำหนดค่าเริ่มต้นให้ state  
  constructor(props) {  
    super(props);  
    this.state = {  
      value: '',  
    };  
  }  
}
```

Props คืออะไร ?

- **props** ย่อมาจาก **properties** มันเป็นตัวที่ทำให้ **React Component** หนึ่งสามารถส่งข้อมูลต่างๆ ไปอีก **component** หนึ่งได้ อีกทั้งยังสามารถส่งค่าเป็น "Integer, Float, String, Object, Array หรือแม้แต่ Function" ก็ได้

การส่งค่าผ่าน props ใน React

- ส่งค่า **String**

```
<MyComponent myPropsString={"Send data"} />
```

- ส่งค่า **Float**

```
<MyComponent myPropsFloat={3.42} />
```

- ส่งค่า **Integer**

```
<MyComponent myPropsInteger={342} />
```

- ส่งค่า **Object**

```
let obj = {id:1, name: 'i am object'};  
<MyComponent myPropsObject={obj} />
```


การส่งค่าผ่าน props ใน React

- ส่งค่า **Array**

```
let arr = [{id:1, name: 'i am array'}];  
<MyComponent myPropsObject={arr} />
```

- ส่งค่า **Function**

```
iAmFunction() {  
  // todo something ...  
}  
  
<MyComponent myPropsFunction={this.iAmFunction.bind(this)} />
```

การรับค่าของ props

- ใช้ "this.props.ชื่อของ props ที่เราส่งผ่าน component"
- เช่น เราถ้าเราส่งค่าผ่าน props ชื่อ "iAmProps"
- เราจะส่งค่า "Hello props" ผ่าน props ชื่อ iAmProps

```
<MyComponent iAmProps={"Hello props"} />
```

- ตอนรับค่าเราสามารถเขียนคำสั่งได้ดังนี้

```
let message = this.props.iAmProps;
```

State :

- **state** คือ การเก็บข้อมูลใน **Component** ของ **React** ซึ่งจะเก็บข้อมูลในรูปแบบของ **JavaScript object**
- วิธีการใช้ **state** ให้เราไปประกาศค่าเริ่มต้นของ **state** ใน **constructor** ก่อน ตามนี้

```
constructor(props) {  
  super(props)  
  
  this.state = {  
    iAmState: "Hello",  
    // กำหนดค่าเริ่มต้นให้กับ state ที่ชื่อว่า iAmState เท่ากับ "Hello"  
    helloState: "State"  
  }  
}
```

State :

- วิธีดึงค่าจาก **state** เราก็แค่เรียกใช้ “**this.state**” ชื่อ **state** ที่เราต้องการ” เช่น

```
let iAmState = this.state.iAmState;
```

- วิธีการเปลี่ยนแปลงข้อมูลใน **state** เราจะต้องใช้ฟังก์ชัน “**setState**” โดยเราต้องระบุ **key** ของ **object** ที่เป็นชื่อ **state** และข้อมูลที่ต้องการเปลี่ยนแปลง เช่น

```
this.setState({iAmState: "Hello State"})
```

- ถ้าต้องการเปลี่ยนแปลงข้อมูล **state** หลายตัว เราสามารถใช้แบบนี้ได้

```
this.setState({iAmState: "Hello State", helloState: "Update State"})
```