



React

Nuttachai Kulthammanit



- 
- ภาพรวมของคอร์สนี้สิ่งที่นักเรียนจะได้เรียนรู้อะไรบ้าง

- คอร์สนี้เหมาะกับใคร

- ทำไมถึงควรรู้จักและใช้งาน States and Events handling

- สิ่งที่ต้องมีก่อนเริ่มเรียน

- พื้นฐาน HTML, CSS, JavaScript เรียนคอร์ส React มาก่อน

Chapter 1

Event

คืออะไร?

Event คืออะไร

- . . . จากเว็บที่เราเขียนในบทที่ผ่านมาจะเห็นได้ว่าเป็นแค่หน้าเว็บที่มีแค่ข้อมูลกดไม่ได้
- . . . หรือเรียกว่า Static Website ในบทนี้เราจะทำให้เว็บมีความ Reactive สามารถกดได้หรือ
- . . . เปลี่ยนแปลงการแสดงผลบนหน้าเว็บได้ ซึ่งคือการใช้ State และ Event ซึ่งเราจะมาพูดถึง
- . . . Event กันก่อนว่า Event ใน React คืออะไร



Event คืออะไร

Event คือ Action การกระทำที่เกิดขึ้นที่โปรแกรมสามารถตรวจจับได้ (อ่านเพิ่มเติม: [What is event? | Definition from TechTarget](#))

อ่านนิยามอาจจะงง มาดูจากตัวอย่างดีกว่า เช่น

- User กดปุ่ม

- User กดคีย์บอร์ด

การกระทำเหล่านี้คือ Event



Event คืออะไร

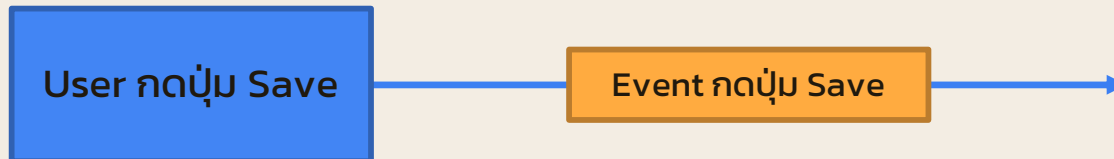
. . . แล้ว Event สำคัญยังไงใน React มันจะช่วยให้เราสามารถรู้ได้ว่า User ต้องการ
จะทำอะไร เช่น User ต้องการ Save ก็ต้องกด ปุ่ม Save

User กดปุ่ม Save



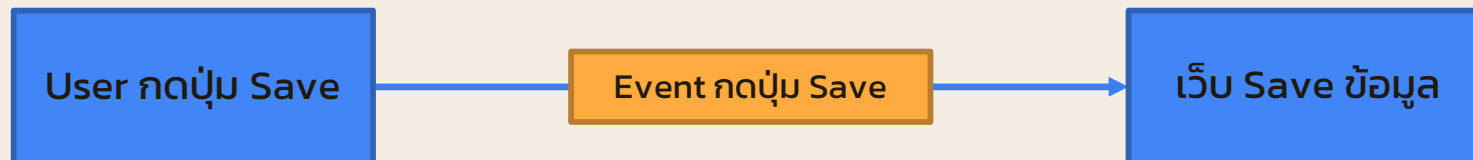
Event คืออะไร

- ...
- ... แล้ว Event สำคัญยังไงใน React มันจะช่วยให้เราสามารถรู้ได้ว่า User ต้องการ
- ... จะทำอะไร เช่น หลังจากนั้น Event ก็จะถูกส่งไปให้เว็บไซต์
- ...
- ...
- ...



Event คืออะไร

- ... แล้ว Event สำคัญยังไงใน React มันจะช่วยให้เราสามารถรู้ได้ว่า User
- ... ต้องการจะทำอะไร เช่น พอเว็บเห็นว่าเป็น Event กดปุ่ม Save ตัวเว็บเซิร์ฟก็จะ
- ... ทำการ Save ข้อมูล
- ...
- ...
- ...





การใส่ Event ใน React

ตัวอย่างของ Event ใน React เช่น
ใน HTML Element

```
<button onclick="changeStatus()">Click me</button>
```

ใน React

```
<button onClick="changeStatus()">Click me</button>
```



การใส่ Event ใน React

ตัวอย่างของ Event ใน React เช่น

ใน HTML Element

```
<button onclick="changeStatus()">Click me</button>
```

ใน React

จะเห็นได้ว่าเวลาใช้ใน React ต้องเป็น camelCase คือหน้าเป็นตัวเล็กและตัวขึ้นต้นคำถัดไปเป็นตัวใหญ่

```
<button onClick={changeStatus()}>Click me</button>
```



การใส่ Function ใน Event

เราสามารถใส่ได้หลายแบบ โดย

ใส่เข้าไปที่ตัว JSX เลย เราเรียกว่า Inline function (ตัวที่ไฮไลท์สีเหลืองคือ Function)

แบบ Arrow-Function

```
<button onClick={()=>{console.log("Clicked")}}>Click me</button>
```

แบบใช้ Function Keyword

```
<button onClick={function(){console.log("Clicked")}}>Click me</button>
```



การใส่ Function ใน Event

แต่ปกติแล้วเราจะประกาศฟังก์ชันไว้ข้างบนก่อน return และมาใช้ด้านล่าง

```
const clickEventHandler = () => {  
  console.log("Clicked")  
}
```

```
return (  
  <div className="App">  
    <button onClick={clickEventHandler}>Click me</button>  
  </div>  
);
```



การใส่ Function ใน Event

แต่ปกติแล้วเราจะประกาศฟังก์ชันไว้ข้างบนก่อน return และมาใช้ด้านล่าง

```
const clickEventHandler = () => {  
  console.log("Clicked")  
}  
  
return (  
  <div className="App">  
    <button onClick={clickEventHandler}>Click me</button>  
  </div>  
)
```



การใส่ Function ใน Event

หรือเราจะประกาศโดยใช้ Function keyword ก็ได้ ไม่จำเป็นต้องเป็น Arrow-function เท่านั้น

```
function clickEventHandler() {  
  console.log("Clicked")  
}  
  
return (  
  <div className="App">  
    <button onClick={clickEventHandler}>Click me</button>  
  </div>  
);
```



คำเตือน: การใส่ Function ใน Event

คำเตือน คนฝึกกันบ่อยๆ เราไม่ต้องใส่ () ตอนที่เราใส่เข้าไปใน onClick นะครับ

```
const clickEventHandler = () => {  
  console.log("Clicked")  
}  
  
return (  
  <div className="App">  
    <button onClick={clickEventHandler}>Click me</button>  
  </div>  
);
```


คำเตือน: การใส่ Function ใน Event

คำเตือน คนฝึกกันบ่อย เราไม่ต้องใส่ () ตอนที่เราใส่เข้าไปใน onClick นะครับ

```

const clickEventHandler = () => {
  console.log("Clicked")
}

return (
  <div className="App">
    <button onClick={clickEventHandler}>Click me</button>
  </div>
);
  
```





คำเตือน: การใส่ Function ใน Event

เพราะว่าเราต้องการส่งฟังก์ชัน clickEventHandler (หรือเรียกว่า pointer) ให้ React และ React ไปรันตอนถูก Click เท่านั้น เราจึงไม่ต้องใส่ () หลังฟังก์ชัน ถ้าเราใส่ไปมันจะถูกรันทันทีที่บรรทัด onClick และส่ง undefined ไปให้ React

ตรงนี้ถ้ายังไม่เป็นไรนะครับ ขอแค่ให้จำไว้ว่าเราไม่ต้องใส่ () ตอนใส่ function เข้าไปให้ตัว Event

```
const clickEventHandler = () => {  
  console.log("Clicked")  
}  
  
return (  
  <div className="App">  
    <button onClick={clickEventHandler}>Click me</button>  
  </div>  
);
```



Code เริ่มต้น

<https://github.com/soncomqiq/ep3-initial-code>



Events

Mini Workshop: Events

Note: Lab-r2-l1



Chapter 2

State

คืออะไร



State คืออะไร

State ใน React คือ สิ่งที่เราเก็บข้อมูลที่จะถูกใช้ใน React Component มันจะคล้ายๆกับตัวแปรตัวหนึ่ง แต่เมื่อมีการเปลี่ยนแปลง React จะมีการ Re-render เพื่อ update ข้อมูลในหน้าเว็บ (ใน DOM)


...





State คืออะไร

วิธีการใช้ State

- การที่เราจะใช้ State เราต้องสร้าง State ผ่านฟังก์ชัน useState
 - เราจะมาดูวิธีใช้ผ่านตัวอย่างกัน
- 

...

State คืออะไร

วิธีการใช้ State – 1. Import

เราสามารถใช้ `import { useState } from "react";` ได้เลย

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

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```

State คืออะไร

วิธีการใช้ State – 2. เรียกใช้งาน useState

การเรียกใช้งานจะมีรูปแบบดังนี้ `const [state, setState] = useState(initialValue);`

โดยที่ `state` คือ state สำหรับการใช้งาน

`setState` คือ ฟังก์ชันสำหรับการใช้เปลี่ยนค่าของ State

`initialValue` คือ ค่าเริ่มต้นของ State นั้น

```
function Example() {  
  // Declare a new state variable, which we'll call "count"  
  const [count, setCount] = useState(0);  
  ...  
}
```


State คืออะไร

วิธีการใช้ State – 2. เรียกใช้งาน

การเรียกใช้งานจะมีรูปแบบดังนี้ `const [state, setState] = useState(initialValue);`

(หน้านี้ถ้า งง ไม่เป็นไรให้ข้ามไปหน้าถัดไปเลย เป็นรายละเอียดเชิงลึก)

ที่จริง useState จะ return ออกมาเป็น array โดยตัวแรกเป็น state ตัวที่สองเป็น function ที่ใช้สำหรับเปลี่ยนค่าของ state นั้น ซึ่งปกติจะนิยมเขียนโดยใช้ Destructuring ออกมาเป็นสองตัวเลยแบบนี้ `const [state, setState] = useState(initialValue);` โดยที่ตัวแรก `state` จะเป็นค่าของ state ส่วน `setState` จะเป็นฟังก์ชันที่ใช้ในการเปลี่ยนค่า (ทั้งสองฝั่งมีค่าเท่ากัน)

```
const [state, setState] = useState(initialValue);
```

```
const arr = useState(0);  
const state = arr[0];  
const setState = arr[1];
```

State คืออะไร

วิธีการใช้ State – 2. เรียกใช้งาน useState

การเรียกใช้งานจะมีรูปแบบดังนี้ `const [state, setState] = useState(initialValue);`

โดยที่ `state` คือ state สำหรับการใช้งาน

`setState` คือ ฟังก์ชันสำหรับการใช้เปลี่ยนค่าของ State

`initialValue` คือ ค่าเริ่มต้นของ State นั้น

```
function Example() {  
  // Declare a new state variable, which we'll call "count"  
  const [count, setCount] = useState(0);  
  ...  
}
```

สร้าง state และให้มีชื่อว่า count

State คืออะไร

วิธีการใช้ State – 2. เรียกใช้งาน

การเรียกใช้งานจะมีรูปแบบดังนี้ `const [state, setState] = useState(initialValue);`

โดยที่ `state` คือ state สำหรับการใช้งาน

`setState` คือ ฟังก์ชันสำหรับการใช้เปลี่ยนค่าของ State

`initialValue` คือ ค่าเริ่มต้นของ State นั้น

```
function Example() {  
  // Declare a new state variable, which we'll call "count"  
  const [count, setCount] = useState(0);  
  ...  
}
```

ฟังก์ชันในการเปลี่ยนค่า State มีชื่อว่า `setCount`

State คืออะไร

วิธีการใช้ State – 2. เรียกใช้งาน useState

การเรียกใช้งานจะมีรูปแบบดังนี้ `const [state, setState] = useState(initialValue);`

โดยที่ `state` คือ state สำหรับการใช้งาน

`setState` คือ ฟังก์ชันสำหรับการใช้เปลี่ยนค่าของ State

`initialValue` คือ ค่าเริ่มต้นของ State นั้น

```
function Example() {  
  // Declare a new state variable, which we'll call "count"  
  const [count, setCount] = useState(0);  
  ...  
}
```

และตั้งให้ค่าเริ่มต้นเป็น 0

State คืออะไร

วิธีการใช้ State – ข้อควรระวังของ useState

useState เป็นหนึ่งใน Hook function ของ React ซึ่งต้องใส่ Component เท่านั้น และไม่อยู่ในฟังก์ชันใด ๆ ของ Components

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

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  const countHandler = () => {
    setCount(count + 1);
  };

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={countHandler}>Click me</button>
    </div>
  );
}
```

แบบนี้เขียนนี้ได้ เพราะว่าอยู่ใน
Component และไม่อยู่ในฟังก์ชันใด

State คืออะไร

วิธีการใช้ State – ข้อควรระวังของ useState

useState เป็นหนึ่งใน Hook function ของ React ซึ่งต้องใส่ Component เท่านั้น และไม่อยู่ในฟังก์ชันใด ๆ ของ Components

```
import React, { useState } from "react";  
  
const [count, setCount] = useState(0); แบบนี้ไม่ได้เพราะอยู่นอก Component  
function Example() {  
  const countHandler = () => {  
    setCount(count + 1);  
  };  
  
  return (  
    <div>  
      <p>You clicked {count} times</p>  
      <button onClick={countHandler}>Click me</button>  
    </div>  
  );  
}
```

State คืออะไร

วิธีการใช้ State – ข้อควรระวังของ useState

useState เป็นหนึ่งใน Hook function ของ React ซึ่งต้องใส่ Component เท่านั้น และไม่ควรอยู่ในฟังก์ชันใด ๆ ของ Components

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

```
function Example() {
```

```
  const countHandler = () => {
```

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

```
    setCount(count + 1);
```

```
  };
```

```
  return (
```

```
    <div>
```

```
      <p>You clicked {count} times</p>
```

```
      <button onClick={countHandler}>Click me</button>
```

```
    </div>
```

```
  );
```

```
}
```

แบบนี้ไม่ได้เพราะอยู่ใน function ของ Components

State คืออะไร

วิธีการใช้ State – ข้อควรระวังของ useState

useState เป็นหนึ่งใน Hook function ของ React ซึ่งต้องใส่ Function Component เท่านั้น และ
ไม่อยู่ในฟังก์ชันใด ๆ ของ Components

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

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```


State คืออะไร

วิธีการใช้ State – 3. นำ State ไปใช้งาน

จะเห็นได้ว่าเราสามารถนำ state ไปใช้งานได้เลย การใช้งานเหมือนกับตัวแปรตัวหนึ่งคือต้องครอบด้วย { }

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

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```

State คืออะไร

วิธีการใช้ State – 4. การเปลี่ยนค่า State โดยใช้ setState

การเปลี่ยนค่าของ state เราจะเรียกฟังก์ชัน setState และใส่ค่าที่ต้องการลงไป

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

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```



เปลี่ยนค่า State





State คืออะไร

setState()

- ใช้เปลี่ยนค่า state ใน component
- การทำงานคือการนำค่าตัวใหม่ไปแทนที่ state ตัวเก่า เช่น setState (ค่าใหม่)
- หลังจาก state มีการเปลี่ยนแปลง setState ตัว React จะ re-render หน้าเว็บใหม่
- setState() ทำงานแบบ async (ค่าจะยังไม่เปลี่ยนแปลงทันที)

...



State คืออะไร

useState() - ตัวอย่าง

ใส่เป็น Arrow Function ให้กับ onClick โดยใน Arrow Function จะมีฟังก์ชัน setCount อยู่

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

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```

State คืออะไร

setState() - ตัวอย่าง

เมื่อมีการ Click จะทำให้สิ่งที่อยู่ข้างใน Arrow Function นั่นก็คือ setCount ทำงาน

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

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```

State คืออะไร

setState() - ตัวอย่าง

การ setState นี้คือนำค่า state เก่านั้นก็คือค่า count เก่ามาบวกเพิ่มไป 1

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

function Example() {
  // Declare a new state variable, which we'll call "count"
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```



State คืออะไร

Mini Workshop: State

Note: Lab-r2-l2



...

...

...

...

...

...



State คืออะไร

คอนเซปต์ของ State ที่ต้องรู้

- เวลาเราสร้าง state โดยใช้ useState ตัว state ที่สร้างขึ้นมาจะเป็นของ Component นั้นเท่านั้นเดี่ยวเราจะมาดูผ่านตัวอย่างกัน

...





State concept

Mini Workshop: State Per Component

Note: Lab-r2-l3

...



State concept

Mini Workshop: State Per Component

This is React Application

Let's get started!

Samuel	Jackson	73	Retired	1950	Click
Keanu	Reeves	58	Working	1965	Click
Tom	Cruise	60	Retired	1963	Click
Johnny	Depp	59	Working	1964	Click

Status: Available

Click me

แต่ละ กล่อง ก็คือ StudentItem 1 อัน

```
return (  
  <div className="App">  
    <h2>This is React Application</h2>  
    <p>Let's get started!</p>  
    <StudentItem name={studentList[0].name}>  
      <StudentItem name={studentList[1].name}>  
        <StudentItem name={studentList[2].name}>  
          <StudentItem name={studentList[3].name}>  
            <h3>Status: {status}</h3>  
            <button onClick={clickEventHandler}>Click</button>  
          </StudentItem>  
        </StudentItem>  
      </StudentItem>  
    </div>  
  )
```

State concept

Mini Workshop: State Per Component

This is React Application

Let's get started!

Samuel	Jackson	73	Retired	1950	Click
Keanu	Reeves	58	Working	1965	Click
Tom	Cruise	60	Retired	1963	Click
Johnny	Depp	59	Working	1964	Click

Status: Available

Click me

```
return (  
  <div className="App">  
    <h2>This is React Application</h2>  
    <p>Let's get started!</p>  
    <StudentItem name={studentList[0].name}>  
    <StudentItem name={studentList[1].name}>  
    <StudentItem name={studentList[2].name}>  
    <StudentItem name={studentList[3].name}>  
    <h3>Status: {status}</h3>  
    <button onClick={clickEventHandler}>Click  
  </div>  
)
```

State concept

Mini Workshop: State Per Component

This is React Application

Let's get started!

Samuel	Jackson	73	Retired	1950	Click
Keanu	Reeves	58	Working	1965	Click
Tom	Cruise	60	Retired	1963	Click
Johnny	Depp	59	Working	1964	Click

Status: Available

Click me

```
return (  
  <div className="App">  
    <h2>This is React Application</h2>  
    <p>Let's get started!</p>  
    <StudentItem name={studentList[0].name}>  
    <StudentItem name={studentList[1].name}>  
    <StudentItem name={studentList[2].name}>  
    <StudentItem name={studentList[3].name}>  
    <h3>Status: {status}</h3>  
    <button onClick={clickEventHandler}>Click  
  </div>  
)
```

State concept

Mini Workshop: State Per Component

This is React Application

Let's get started!

Samuel	Jackson	73	Retired	1950	Click
Keanu	Reeves	58	Working	1965	Click
Tom	Cruise	60	Retired	1963	Click
Johnny	Depp	59	Working	1964	Click

Status: Available

Click me

```
return (  
  <div className="App">  
    <h2>This is React Application</h2>  
    <p>Let's get started!</p>  
    <StudentItem name={studentList[0].name}>  
    <StudentItem name={studentList[1].name}>  
    <StudentItem name={studentList[2].name}>  
    <StudentItem name={studentList[3].name}>  
    <h3>Status: {status}</h3>  
    <button onClick={clickEventHandler}>Click  
  </div>  
)
```

State concept

Mini Workshop: State Per Component

This is React Application

Let's get started!

Samuel	Jackson	73	Retired	1950	Click
Keanu	Reeves	58	Working	1965	Click
Tom	Cruise	60	Retired	1963	Click
Johnny	Depp	59	Working	1964	Click

Status: Available

Click me

```
return (  
  <div className="App">  
    <h2>This is React Application</h2>  
    <p>Let's get started!</p>  
    <StudentItem name={studentList[0].name}>  
    <StudentItem name={studentList[1].name}>  
    <StudentItem name={studentList[2].name}>  
    <StudentItem name={studentList[3].name}>  
    <h3>Status: {status}</h3>  
    <button onClick={clickEventHandler}>Click  
  </div>  
)
```


State concept

Mini Workshop: State Per Component

This is React Application

Let's get started!

Samuel	Jackson	73	Retired	1950	Click
Keanu	Reeves	58	Working	1965	Click
Tom	Cruise	60	Retired	1963	Click
Johnny	Depp	59	Working	1964	Click

Status: Available

Click me

```
return (  
  <div className="App">  
    <h2>This is React Application</h2>  
    <p>Let's get started!</p>  
    <StudentItem name={studentList[0].name}>  
    <StudentItem name={studentList[1].name}>  
    <StudentItem name={studentList[2].name}>  
    <StudentItem name={studentList[3].name}>  
    <h3>Status: {status}</h3>  
    <button onClick={clickEventHandler}>Click  
  </div>  
)
```


State concept

Mini Workshop: State Per Component

This is React Application

Let's get started!

Samuel	Jackson	73	Retired	1950	Click
Keanu	Reeves	58	Working	1965	Click
Tom	Cruise	60	Retired	1963	Click
Johnny	Depp	59	Working	1964	Click

Status: Available

Click me

```
return (  
  <div className="App">  
    <h2>This is React Application</h2>  
    <p>Let's get started!</p>  
    <StudentItem name={studentList[0].name}>  
    <StudentItem name={studentList[1].name}>  
    <StudentItem name={studentList[2].name}>  
    <StudentItem name={studentList[3].name}>  
    <h3>Status: {status}</h3>  
    <button onClick={clickEventHandler}>Click  
  </div>  
)
```

State concept

Mini Workshop: State Per Component

This is React Application

Let's get started!

Samuel	Jackson	73	Retired	1950	Click
Keanu	Reeves	58	Working	1965	Click
Tom	Cruise	60	Retired	1963	Click
Johnny	Depp	59	Working	1964	Click

Status: Available

Click me

```
return (  
  <div className="App">  
    <h2>This is React Application</h2>  
    <p>Let's get started!</p>  
    <StudentItem name={studentList[0].name}>  
    <StudentItem name={studentList[1].name}>  
    <StudentItem name={studentList[2].name}>  
    <StudentItem name={studentList[3].name}>  
    <h3>Status: {status}</h3>  
    <button onClick={clickEventHandler}>Click  
  </div>  
)
```



State concept

Mini Workshop: State Per Component

แต่ละ กล่อง ก็คือ StudentItem 1 อัน
เพราะฉะนั้นแต่ละกล่องนั้นก็จะมี state
ชื่อ name เป็นของตัวเอง

name => "Samuel"

This is React Application

Let's get started!

Samuel	Jackson	73	Retired	1950	Click
Keanu	Reeves	58	Working	1965	Click
Tom	Cruise	60	Retired	1963	Click
Johnny	Depp	59	Working	1964	Click

Status: Available

Click me





State concept

Mini Workshop: State Per Component

แต่ละ กล่อง ก็คือ StudentItem 1 อัน
เพราะฉะนั้นแต่ละกล่องนั้นก็จะมี state ชื่อ
name เป็นของตัวเอง

name => "Samuel"

name => "Keanu"

name => "Tom"

name => "Johnny"

This is React Application

Let's get started!

Samuel	Jackson	73	Retired	1950	Click
Keanu	Reeves	58	Working	1965	Click
Tom	Cruise	60	Retired	1963	Click
Johnny	Depp	59	Working	1964	Click

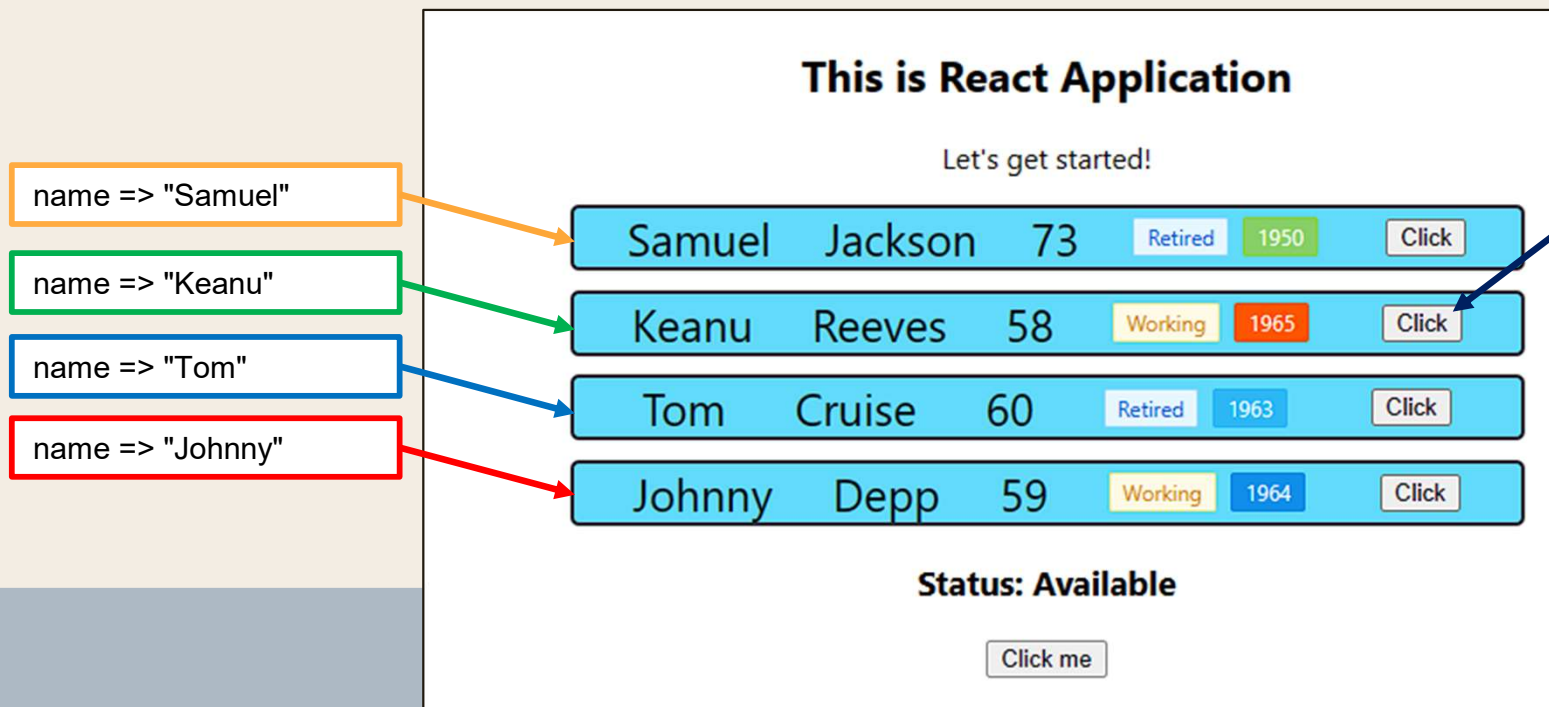
Status: Available

Click me



State concept

Mini Workshop: State Per Component



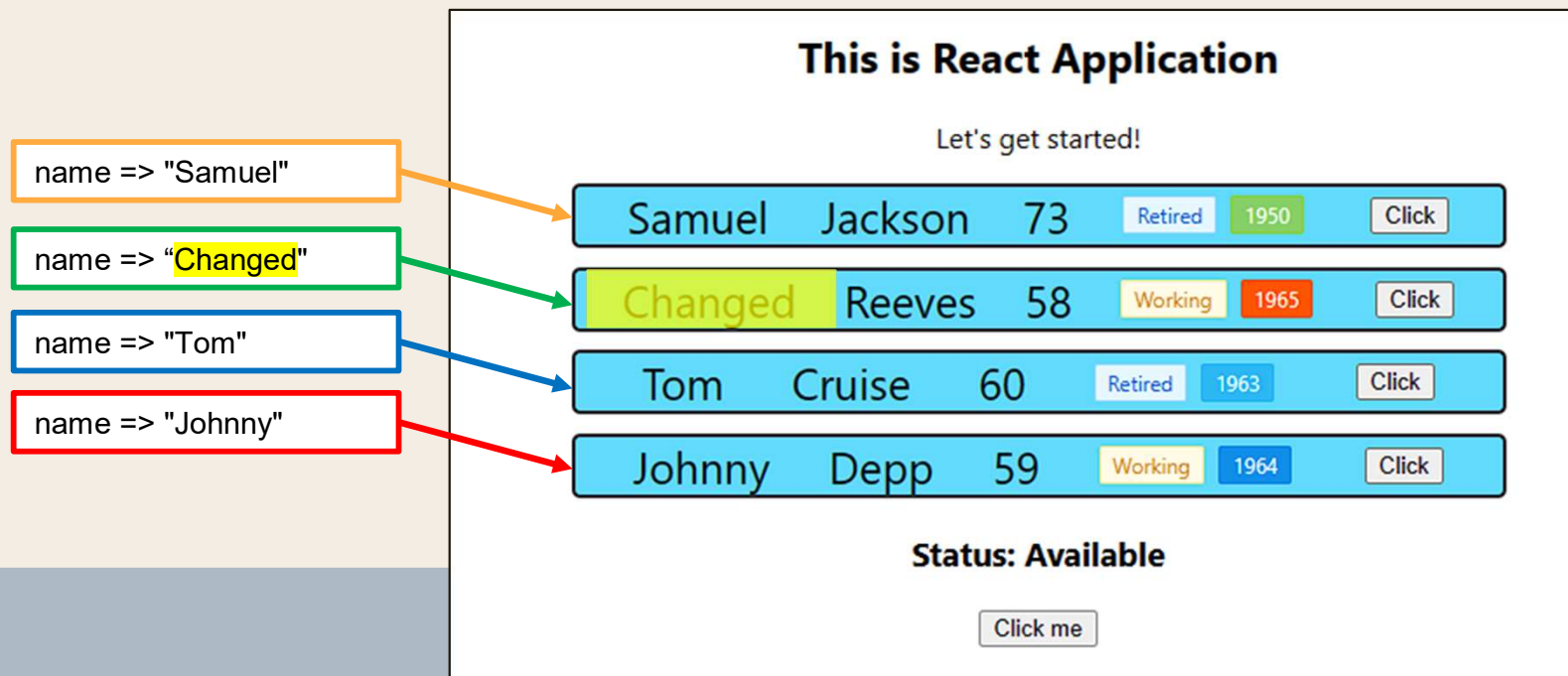
เมื่อเรา Click ที่ปุ่ม
ของอันที่สอง



State concept

Mini Workshop: State Per Component

มันก็จะเปลี่ยนแค่ state ของ
StudentItem อันที่สองเท่านั้น และ
อันอื่นก็ยังคง State เดิมไว้อยู่



Chapter 3

Form Input

Event រួមជាមួយ State



Form Input

Form เป็น HTML element ที่ใช้ในการสร้าง Form สำหรับกรอกข้อมูลเพื่อรับข้อมูลจาก User

First name:

Last name:

https://www.w3schools.com/html/html_forms.asp





Form Input

HTML Element ที่ใช้กับ Form ก็จะมีหลากหลายตัว เช่น

- <input>
- <select>
- <textarea>

แต่จะไม่ขอลงรายละเอียด ดูเพิ่มเติมในคอร์ส HTML หรือ https://www.w3schools.com/html/html_forms.asp





Form Input

Mini Workshop: Form Input

Note: Lab-r2-l4

CSS Link: <https://raw.githubusercontent.com/soncomqiq/react-app-1/main/src/components/NewStudentItem.css>





Form Input

Mini Workshop: Listening to User Input and Handling values

Note: Lab-r2-l5





Form Input

Mini Workshop: Submit Form

Note: Lab-r2-l6





Form Input

Mini Workshop: Component Communication

Note: Lab-r2-l8





Form Input

Mini Workshop: Clear Form

```
const [currentName, setCurrentName] = useState('');
```

ตอนที่เรสร้าง State ก็จะมี state เกิดขึ้นมา



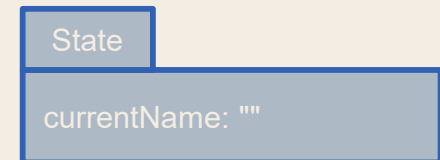


Form Input

Mini Workshop: Clear Form

```
const [currentName, setCurrentName] = useState('');
```

ตอนที่เราร่าง State ก็จะมี state เกิดขึ้นมา





Form Input

Mini Workshop: Clear Form

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler}/>
</div>
```

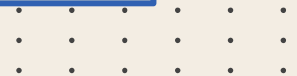
```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

```
const [currentName, setCurrentName] = useState('');
```

และมี SetCurrentName ให้เราไว้ใช้สำหรับ update
ค่าของ currentName

State

currentName: ""



Form Input

Mini Workshop: Clear Form

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler}/>
</div>
```

```
const [currentName, setCurrentName] = useState('');
```

และมี SetCurrentName ให้เราไว้ใช้สำหรับ
update ค่าของ currentName

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

State

currentName: ""



Form Input

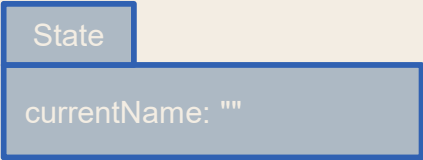
Mini Workshop: Clear Form

เวลา user มีการพิมพ์แต่ละครั้งใน input

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```





Form Input

Mini Workshop: Clear Form

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

React ก็จะไปเรียกฟังก์ชันที่ถูกส่งมาใน
onChange

State

currentName: ""



Form Input

Mini Workshop: Clear Form

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

ซึ่งก็คือฟังก์ชันที่เราเขียนเพื่อ update ค่าของ `currentName` เพราะฉะนั้น `currentName` ก็จะเปลี่ยนตาม Input

State

currentName: "T"

...

Form Input

Mini Workshop: Clear Form

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

ที่นี่ไม่ว่าเราจะพิมพ์อะไรมันก็จะอัปเดตให้
ตามที่เรพิมพ์

State

currentName: "Test"

• • • • •
• • • • •
• • • • •

Form Input

Mini Workshop: Clear Form

แต่ในทางกลับกัน ตอนนี้เรา
setCurrentName ให้เป็นค่าว่าง ตอนกด
Add Student

Add Student

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

State

currentName: "Test"



Form Input

Mini Workshop: Clear Form

ซึ่งมันก็จะไปเรียกฟังก์ชันที่ส่งมาใน
onSubmit ซึ่งเป็นฟังก์ชันที่เรามีการใส่
setCurrentName ให้เป็นค่าว่าง

Add Student

setCurrentName(value: "")

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

State

currentName: "Test"

• • • • •
• • • • •

Form Input

Mini Workshop: Clear Form

มันก็จะ update ไปที่ state currentName

Add Student

```
setCurrentName( value: "")
```

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

State

currentName: "Test"

• • • • •
• • • • •
• • • • •

Form Input

Mini Workshop: Clear Form

ตอนนี้ State เป็นค่าว่างแล้ว แต่ input
ไม่ได้รับรู้ว่า state ถูกเปลี่ยนไปแล้ว

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

Add Student

setCurrentName(value: "")

State

currentName: ""

...

Form Input

Mini Workshop: Clear Form

วิธีการที่ทำให้ input เปลี่ยนแปลงตาม state คือการใส่ value properties

Add Student

setCurrentName(value: "")

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler} value={currentName}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

State

currentName: ""

.....

Form Input

Mini Workshop: Clear Form

การใส่แบบนี้จะทำให้ค่าของ state ส่งไปยัง input ด้วย

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler} value={currentName}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

Add Student

```
setCurrentName( value: "" )
```

State

currentName: ""

...

Form Input

Mini Workshop: Clear Form

สมมติตอนนี้ CurrentName ยังเป็นมีค่า Test อยู่

Add Student

setCurrentName(value: "")

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler} value={currentName}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

State

currentName: "Test"

• • • • •
• • • • •

Form Input

Mini Workshop: Clear Form

เมื่อเรากด Add Student ก็จะมีการ
setCurrentName ให้เป็นค่าว่าง

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler} value={currentName}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

Add Student

```
setCurrentName( value: "")
```

State

currentName: "Test"



Form Input

Mini Workshop: Clear Form

เมื่อเรากด Add Student ก็จะมีการ
setCurrentName ให้เป็นค่าว่าง

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler} value={currentName}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

Add Student

setCurrentName(value: "")

State

currentName: ""

• • • • •
• • • • •

Form Input

Mini Workshop: Clear Form

เนื่องจากเราใส่ properties value เข้าไป
เมื่อ state เปลี่ยนค่าของ input ก็จะเปลี่ยน
ตาม state ด้วย

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler} value={currentName}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

Add Student

```
setCurrentName( value: "" )
```

State

currentName: ""

• • • • •
• • • • •

Form Input

Mini Workshop: Clear Form

แบบนี้เราเรียกว่า Two-way binding

Add Student

Name

```
<div className="StudentInput">
  <label>Name</label>
  <input type="text" onChange={nameChangeHandler} value={currentName}/>
</div>
```

```
const nameChangeHandler = (event) => {
  setCurrentName(event.target.value)
}
```

setCurrentName(value: "")

State

currentName: ""

• • • • •
• • • • •



Form Input

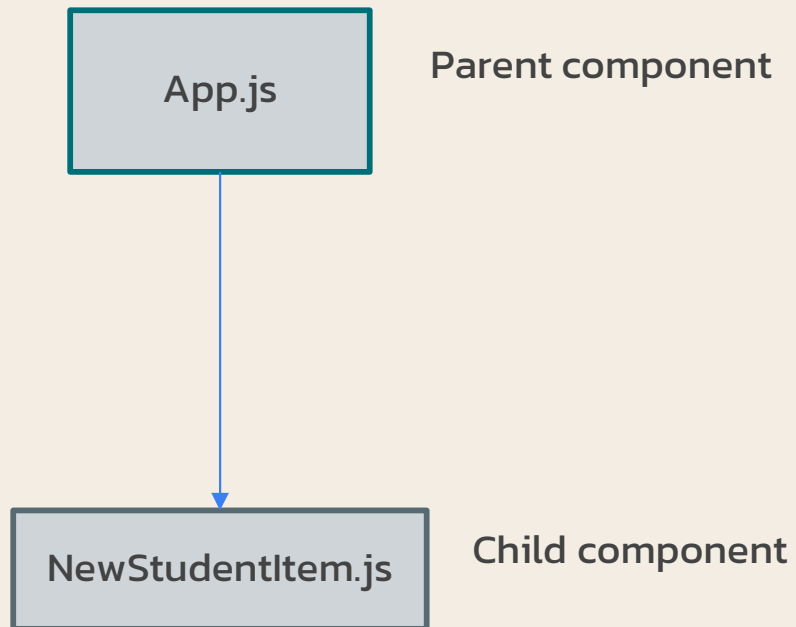
Mini Workshop: Component Communication

Note: Lab-r2-l8



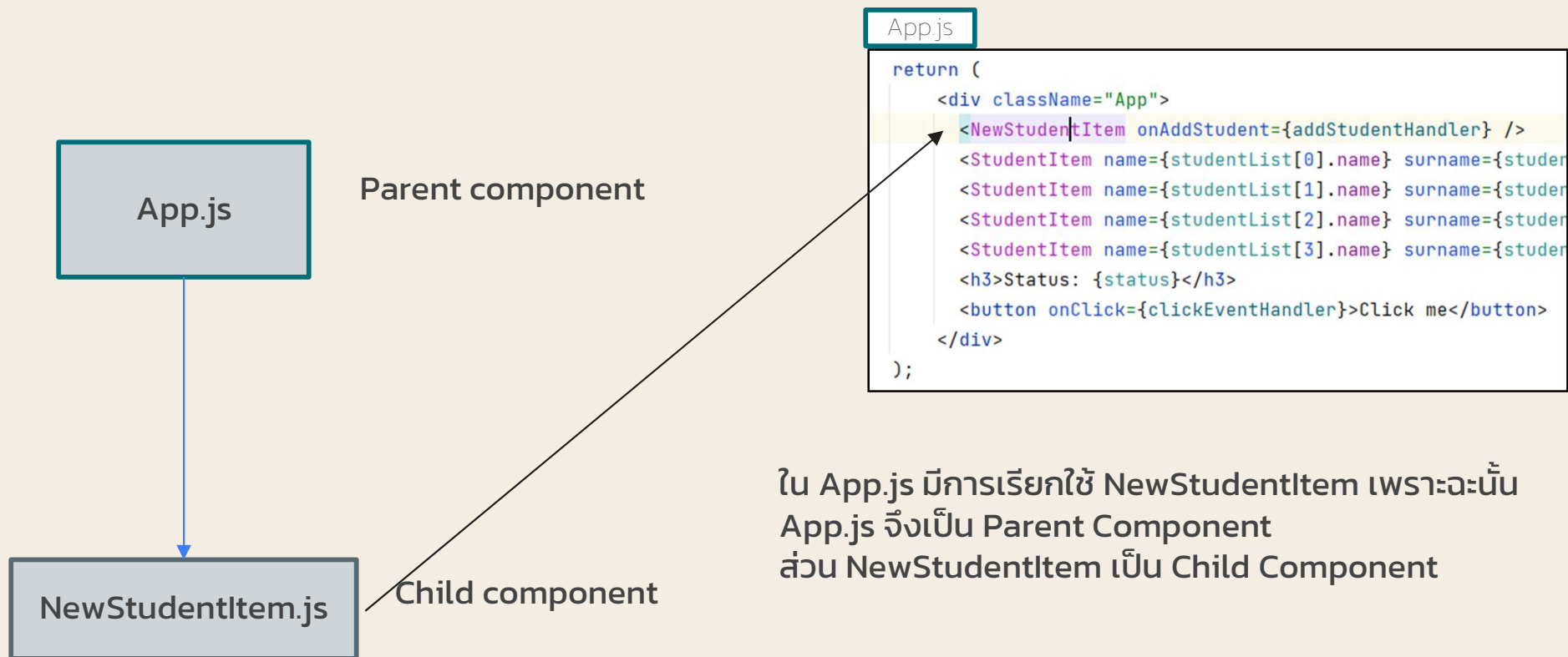
Form Input

Mini Workshop: Component Communication



Form Input

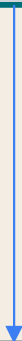
Mini Workshop: Component Communication



Form Input

Mini Workshop: Component Communication

App.js



NewStudentItem.js

```
const addStudentHandler = (newStudentData) => {  
  const newStudent = {  
    ...newStudentData,  
    id: Math.random().toString()  
  }  
  console.log(newStudent)  
}
```

```
return (  
  <div className="App">  
    <NewStudentItem onAddStudent={addStudentHandler} />  
    <StudentItem name={studentList[0].name} surname={stu
```

ใน App.js ส่งฟังก์ชัน addStudentHandler ไปให้ NewStudentItem เพื่อให้เรียกใช้เวลา add student เสร็จแล้ว

Form Input

Mini Workshop: Component Communication

App.js

NewStudentItem.js

```
const addStudentHandler = (newStudentData) => {  
  const newStudent = {  
    ...newStudentData,  
    id: Math.random().toString()  
  }  
  console.log(newStudent)  
}
```

```
return (  
  <div className="App">  
    <NewStudentItem onAddStudent={addStudentHandler} />  
    <StudentItem name={studentList[0].name} surname={stu
```

```
const submitHandler = (event) => {  
  event.preventDefault();  
  const newStudent = {  
    name: currentName,  
    surname: currentLastName,  
    age: Number(currentAge)  
  }  
  
  props.onAddStudent(newStudent)
```

ซึ่งเวลาส่ง addStudentHandler มา
มันถูกส่งมาในชื่อ onAddStudent

Form Input

Mini Workshop: Component Communication

App.js

NewStudentItem.js

```
const addStudentHandler = (newStudentData) => {  
  const newStudent = {  
    ...newStudentData,  
    id: Math.random().toString()  
  }  
  console.log(newStudent)  
}
```

```
return (  
  <div className="App">  
    <NewStudentItem onAddStudent={addStudentHandler} />  
    <StudentItem name={studentList[0].name} surname={stu
```

```
const submitHandler = (event) => {  
  event.preventDefault();  
  const newStudent = {  
    name: currentName,  
    surname: currentLastName,  
    age: Number(currentAge)  
  }  
  
  props.onAddStudent(newStudent)
```

เพราะฉะนั้นเวลาเรียกมันก็
ต้องเรียกโดยใช้ชื่อ onAddStudent

Form Input

Mini Workshop: Component Communication

App.js

NewStudentItem.js

```
const addStudentHandler = (newStudentData) => {  
  const newStudent = {  
    ...newStudentData,  
    id: Math.random().toString()  
  }  
  console.log(newStudent)  
}
```

```
return (  
  <div className="App">  
    <NewStudentItem onAddStudent={addStudentHandler} />  
    <StudentItem name={studentList[0].name} surname={stu
```

```
const submitHandler = (event) => {  
  event.preventDefault();  
  const newStudent = {  
    name: currentName,  
    surname: currentLastName,  
    age: Number(currentAge)  
  }  
  
  props.onAddStudent(newStudent)
```

เนื่องจาก onAddStudent มันก็ฟังก์ชัน
addStudentHandler นั้นแหละ

Form Input

Mini Workshop: Component Communication

App.js

NewStudentItem.js

```
const addStudentHandler = (newStudentData) => {  
  const newStudent = {  
    ...newStudentData,  
    id: Math.random().toString()  
  }  
  console.log(newStudent)  
}
```

```
return (  
  <div className="App">  
    <NewStudentItem onAddStudent={addStudentHandler} />  
    <StudentItem name={studentList[0].name} surname={stu
```

```
const submitHandler = (event) => {  
  event.preventDefault();  
  const newStudent = {  
    name: currentName,  
    surname: currentLastName,  
    age: Number(currentAge)  
  }  
  
  props.onAddStudent(newStudent)
```

เพราะฉะนั้นเวลา NewStudentItem เรียกฟังก์ชัน **onAddStudent** มันก็เปรียบเสมือนการเรียก **addStudentHandler** ของ App.js นั้นเอง

Form Input

Mini Workshop: Component Communication

App.js

NewStudentItem.js

```
const addStudentHandler = (newStudentData) => {  
  const newStudent = {  
    ...newStudentData,  
    id: Math.random().toString()  
  }  
  console.log(newStudent)  
}
```

```
return (  
  <div className="App">  
    <NewStudentItem onAddStudent={addStudentHandler} />  
    <StudentItem name={studentList[0].name} surname={stu
```

```
const submitHandler = (event) => {  
  event.preventDefault();  
  const NewStudent = {  
    name: currentName,  
    surname: currentLastName,  
    age: Number(currentAge)  
  }  
  
  props.onAddStudent(newStudent)
```

ซึ่ง submitHandler ใน NewStudentItem.js
ก็จะส่งข้อมูล `NewStudent` ผ่านฟังก์ชัน
`props.onAddStudent`
มันก็เปรียบเสมือนการเรียก
`addStudentHandler` ใน App.js แต่ข้อมูลที่
ส่งมาจะเป็นชื่อใหม่คือ `newStudentData`
ตามที่เรากำหนดไว้ แล้วเราก็เอาไปใช้ต่อ

Form Input

Mini Workshop: Component Communication

App.js

NewStudentItem.js

```
const addStudentHandler = (newStudentData) => {  
  const newStudent = {  
    ...newStudentData,  
    id: Math.random().toString()  
  }  
  console.log(newStudent)  
}
```

```
return (  
  <div className="App">  
    <NewStudentItem onAddStudent={addStudentHandler} />  
    <StudentItem name={studentList[0].name} surname={stu
```

```
const submitHandler = (event) => {  
  event.preventDefault();  
  const newStudent = {  
    name: currentName,  
    surname: currentLastName,  
    age: Number(currentAge)  
  }  
  
  props.onAddStudent(NewStudent
```

ซึ่ง submitHandler ใน NewStudentItem.js
ก็จะส่งข้อมูล `NewStudent` ผ่านฟังก์ชัน
`props.onAddStudent`
มันก็เปรียบเสมือนการเรียก
`addStudentHandler` ใน App.js แต่ข้อมูลที่
ส่งมาจะเป็นชื่อใหม่คือ `newStudentData`
ตามที่เรากำหนดไว้ แล้วเราก็เอาไปใช้ต่อ

Form Input

Mini Workshop: Component Communication

App.js

NewStudentItem.js

```
const addStudentHandler = NewStudentData => {  
  const newStudent = {  
    ...newStudentData,  
    id: Math.random().toString()  
  }  
  console.log(newStudent)  
}
```

```
return (  
  <div className="App">  
    <NewStudentItem onAddStudent={addStudentHandler} />  
    <StudentItem name={studentList[0].name} surname={stu
```

```
const submitHandler = (event) => {  
  event.preventDefault();  
  const newStudent = {  
    name: currentName,  
    surname: currentLastName,  
    age: Number(currentAge)  
  }  
  
  props.onAddStudent(newStudent)
```

ซึ่ง submitHandler ใน NewStudentItem.js
ก็จะส่งข้อมูล `NewStudent` ผ่านฟังก์ชัน
`props.onAddStudent`
มันก็เปรียบเสมือนการเรียก
`addStudentHandler` ใน App.js แต่ข้อมูลที่
ส่งมาจะเป็นชื่อใหม่คือ `newStudentData`
ตามที่เรากำหนดไว้ แล้วเราก็เอาไปใช้ต่อ

Form Input

Mini Workshop: Component Communication

App.js

NewStudentItem.js

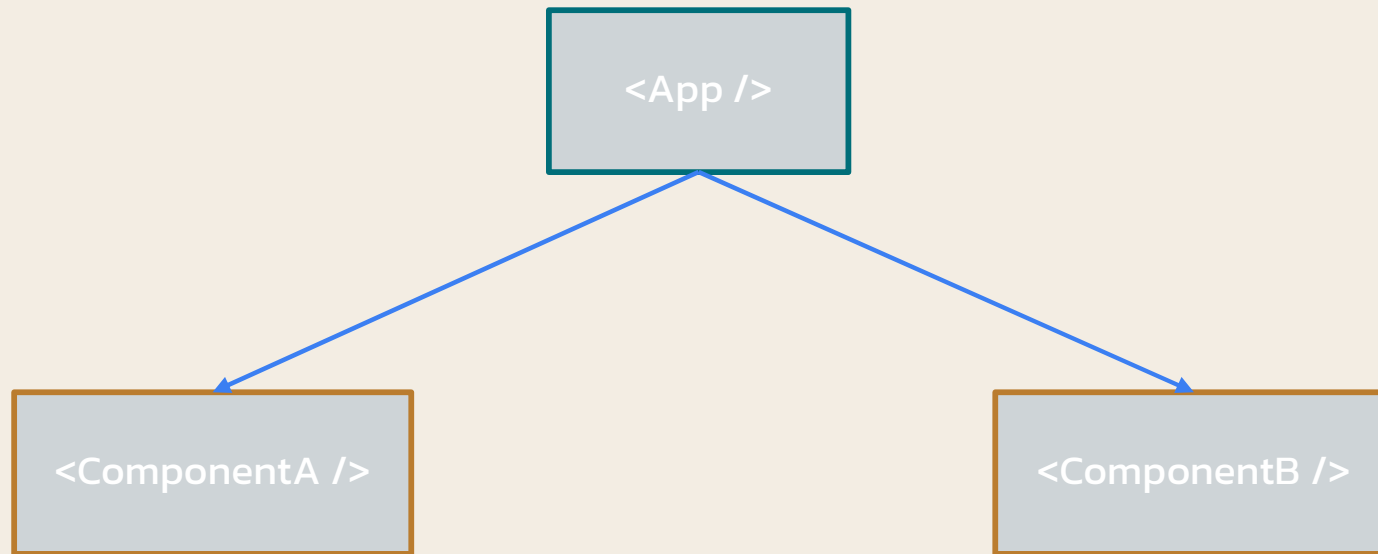
```
const addStudentHandler = (newStudentData) => {  
  const newStudent = {  
    .. NewStudentData  
    id: Math.random().toString()  
  }  
  console.log(newStudent)  
}
```

```
return (  
  <div className="App">  
    <NewStudentItem onAddStudent={addStudentHandler} />  
    <StudentItem name={studentList[0].name} surname={stu
```

```
const submitHandler = (event) => {  
  event.preventDefault();  
  const newStudent = {  
    name: currentName,  
    surname: currentLastName,  
    age: Number(currentAge)  
  }  
  
  props.onAddStudent(newStudent)
```

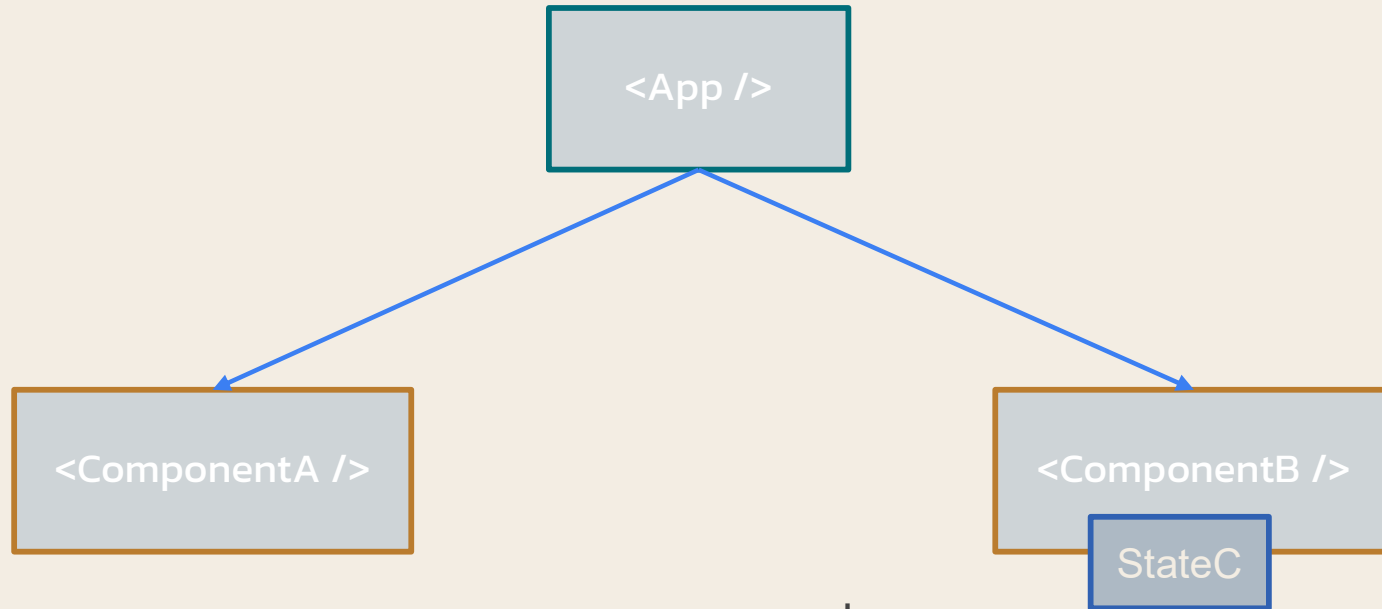
ซึ่ง submitHandler ใน NewStudentItem.js
ก็จะส่งข้อมูล `NewStudent` ผ่านฟังก์ชัน
`props.onAddStudent`
มันก็เปรียบเสมือนการเรียก
`addStudentHandler` ใน App.js แต่ข้อมูลที่
ส่งมาจะเป็นชื่อใหม่คือ `newStudentData`
ตามที่เรากำหนดไว้ แล้วเราก็เอาไปใช้ต่อ

คอนเซปต์ Lifting State Up



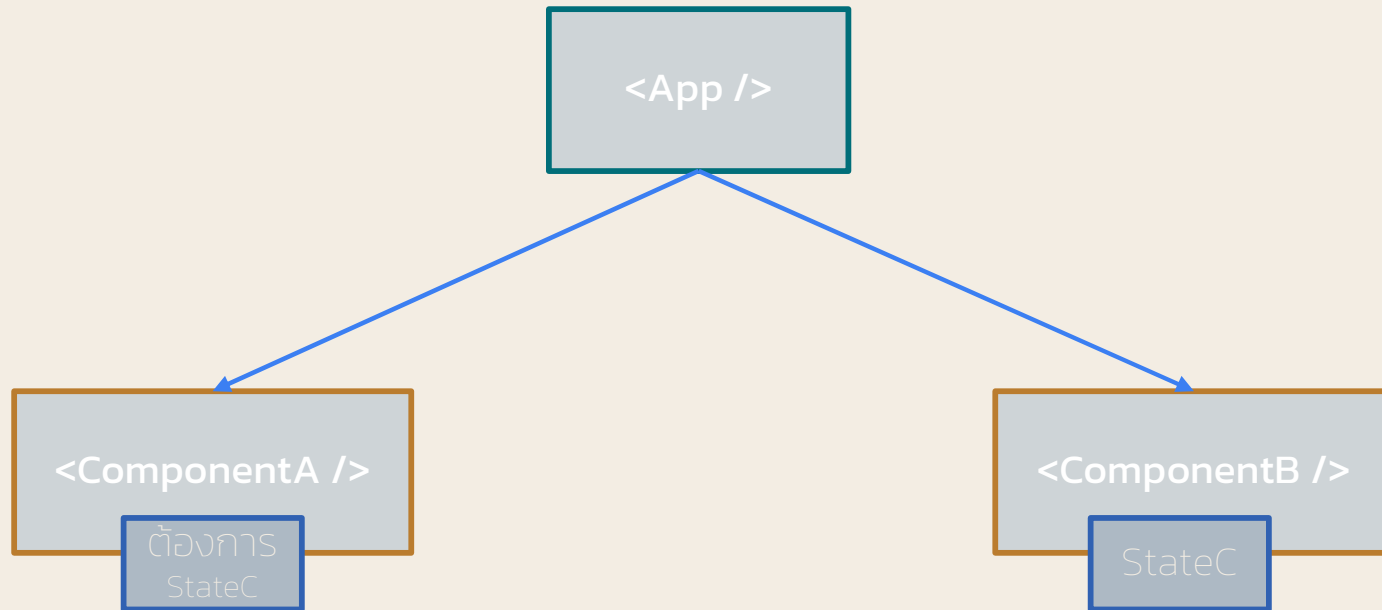
เราใช้ ComponentA และ ComponentB

คอนเซปต์ Lifting State Up



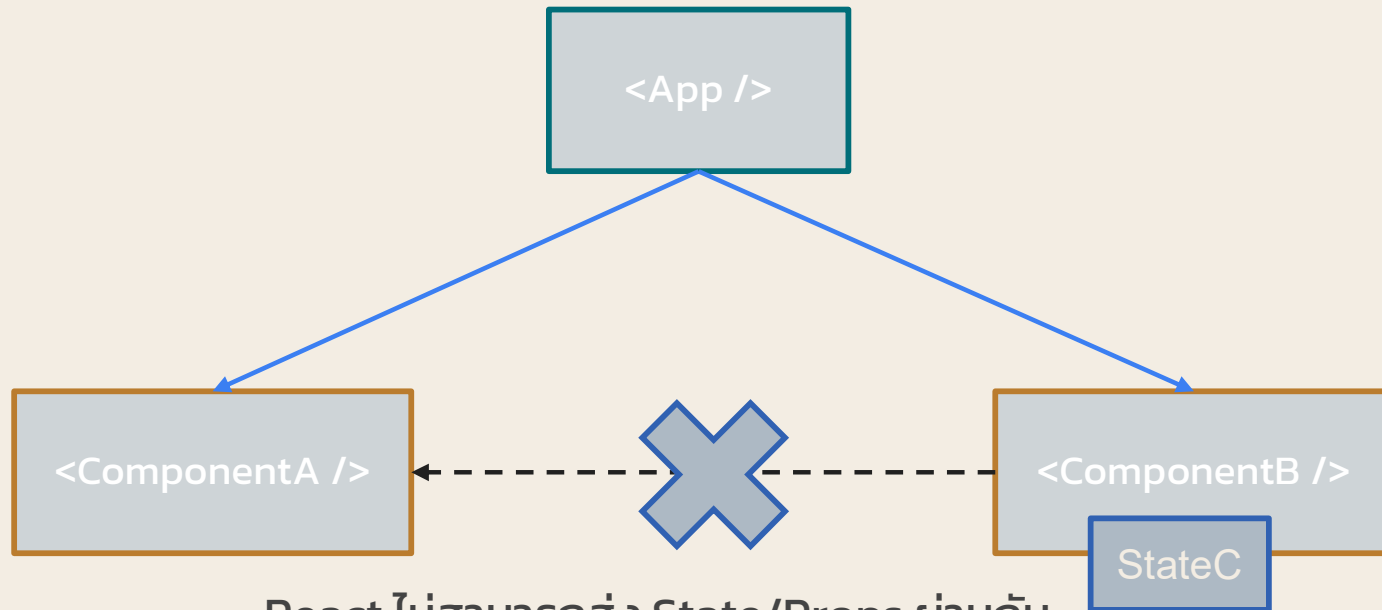
ใน ComponentB มี State ชื่อ StateC
อยู่ และ StateC จะมีปุ่มที่ใช้เปลี่ยน
StateC อยู่ใน ComponentB

คอนเซปต์ Lifting State Up



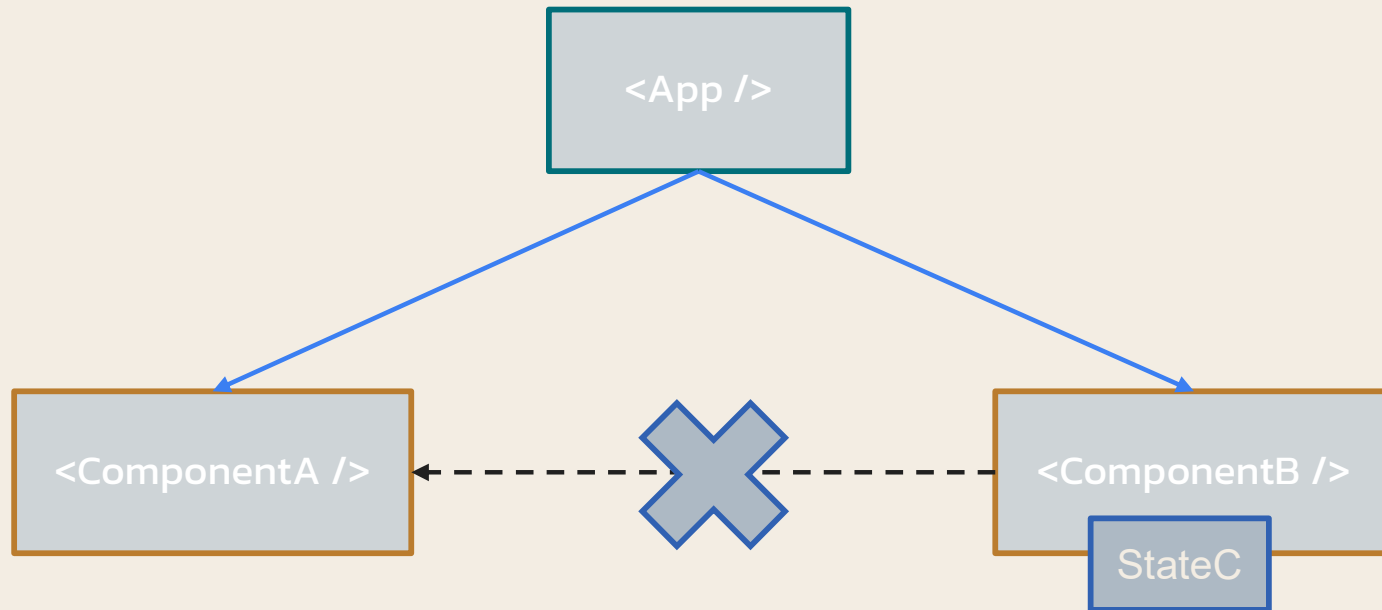
ในกรณีที่ ComponentA มีการแสดงผลที่ขึ้นอยู่กับ StateC เช่น เมื่อ StateC เป็น True ให้ ComponentA เป็นสีแดง

คอนเซปต์ Lifting State Up



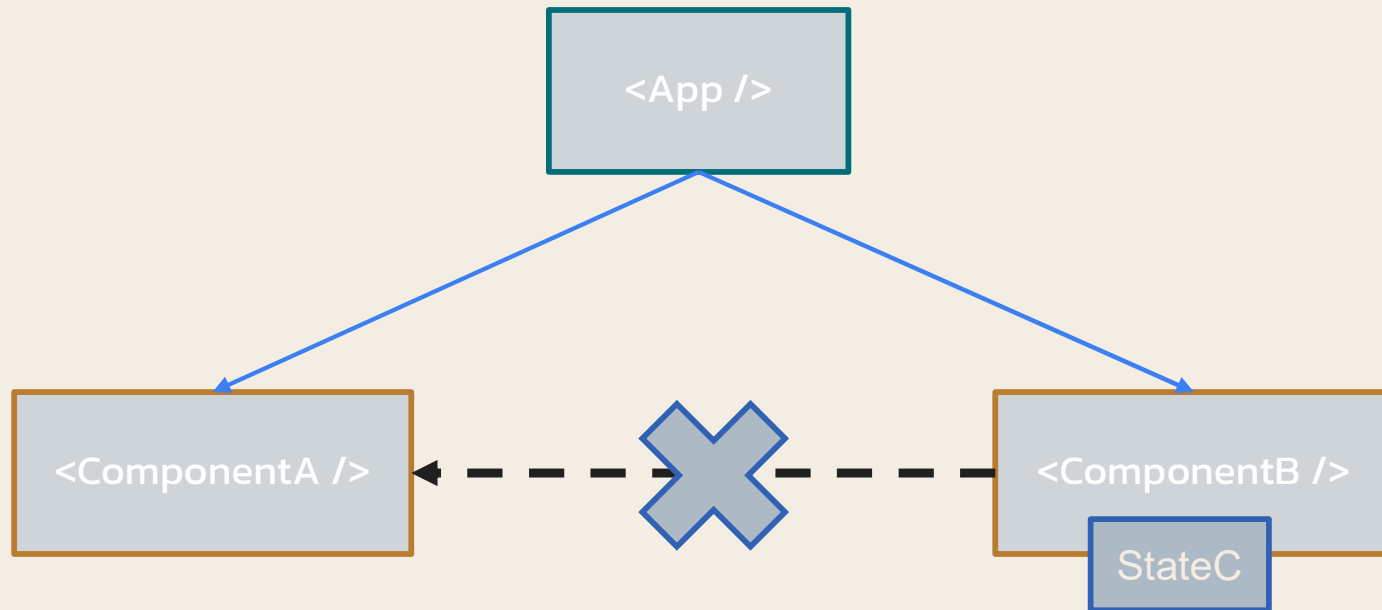
React ไม่สามารถส่ง State/Props ผ่านกัน
โดยตรงแบบนี้ได้ เพราะ ComponentA และ
ComponentB ไม่ได้เป็น Child/Parent กัน แต่เป็น
Sibling

คอนเซปต์ Lifting State Up



เพราะว่าจากที่เราเรียนกันมา React จะ
ส่งได้เฉพาะ Child ไป Parent หรือ
Parent ไป Child เท่านั้น

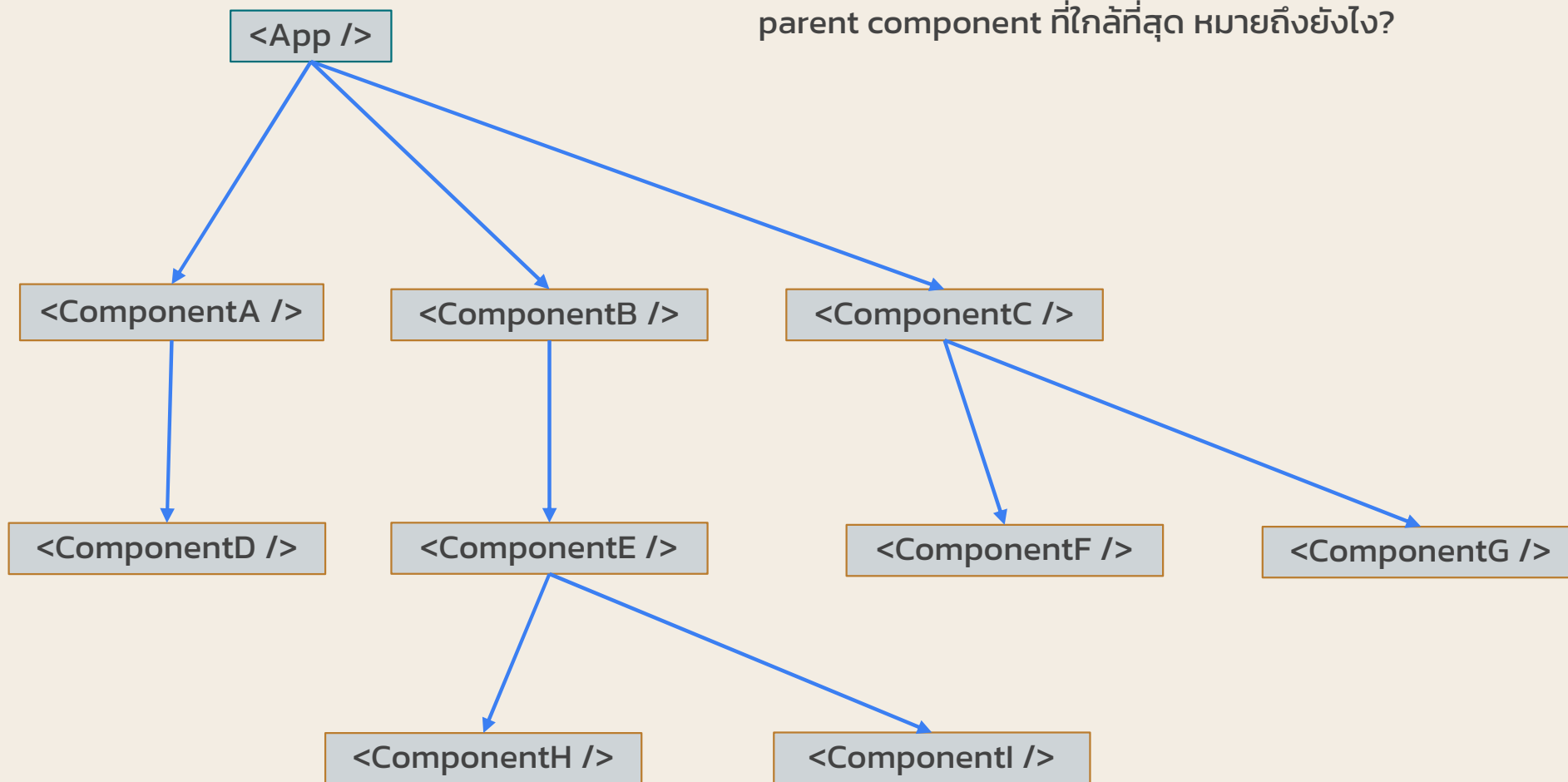
คอนเซปต์ Lifting State Up



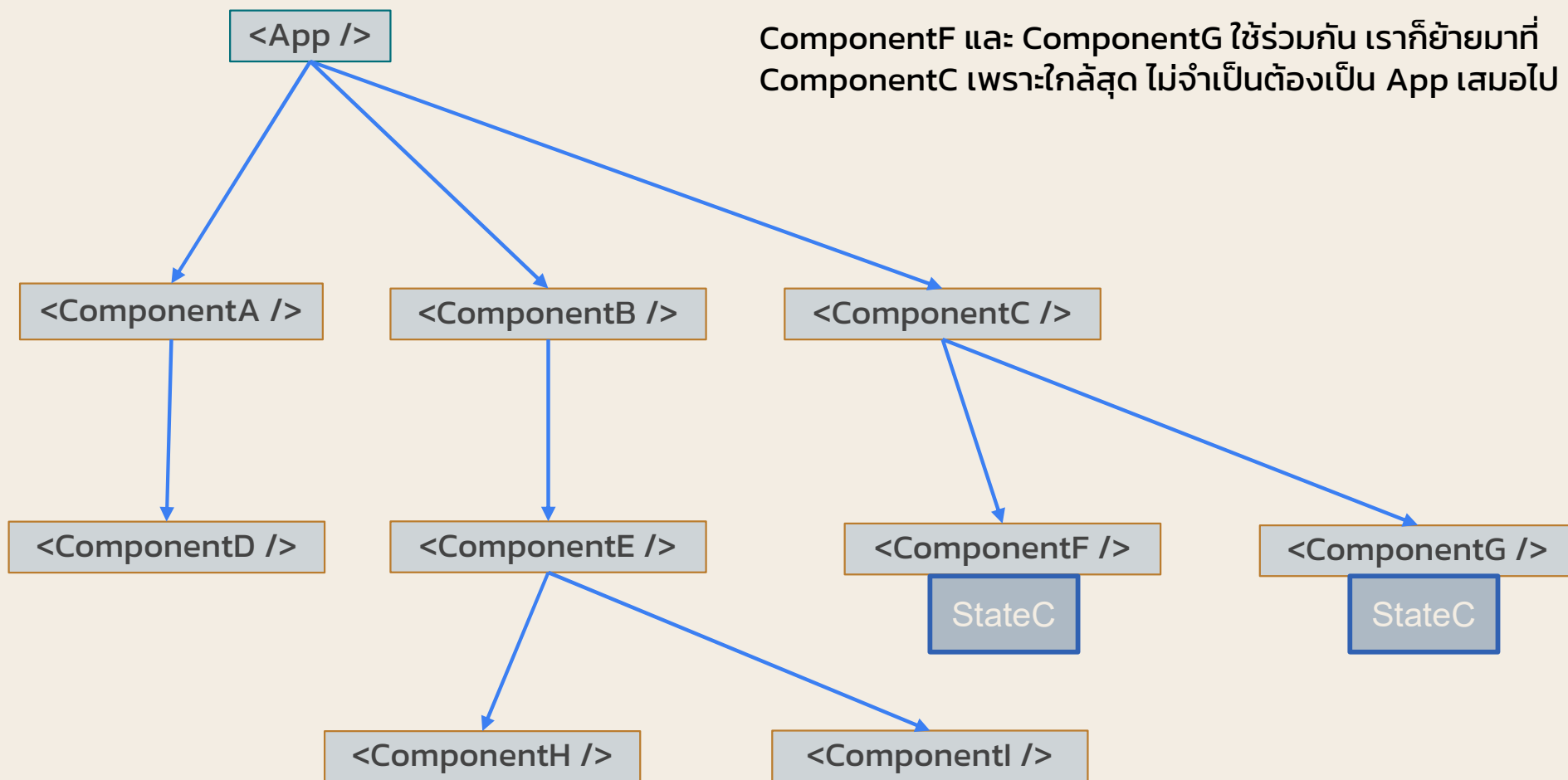
วิธีก็คือให้เราเก็บ state นั้นไว้ที่ parent component ที่ใกล้ที่สุด ที่เข้าถึงสองตัวนั้น ซึ่งในที่นี้ก็คือ App Component

คอนเซปต์ Lifting State Up

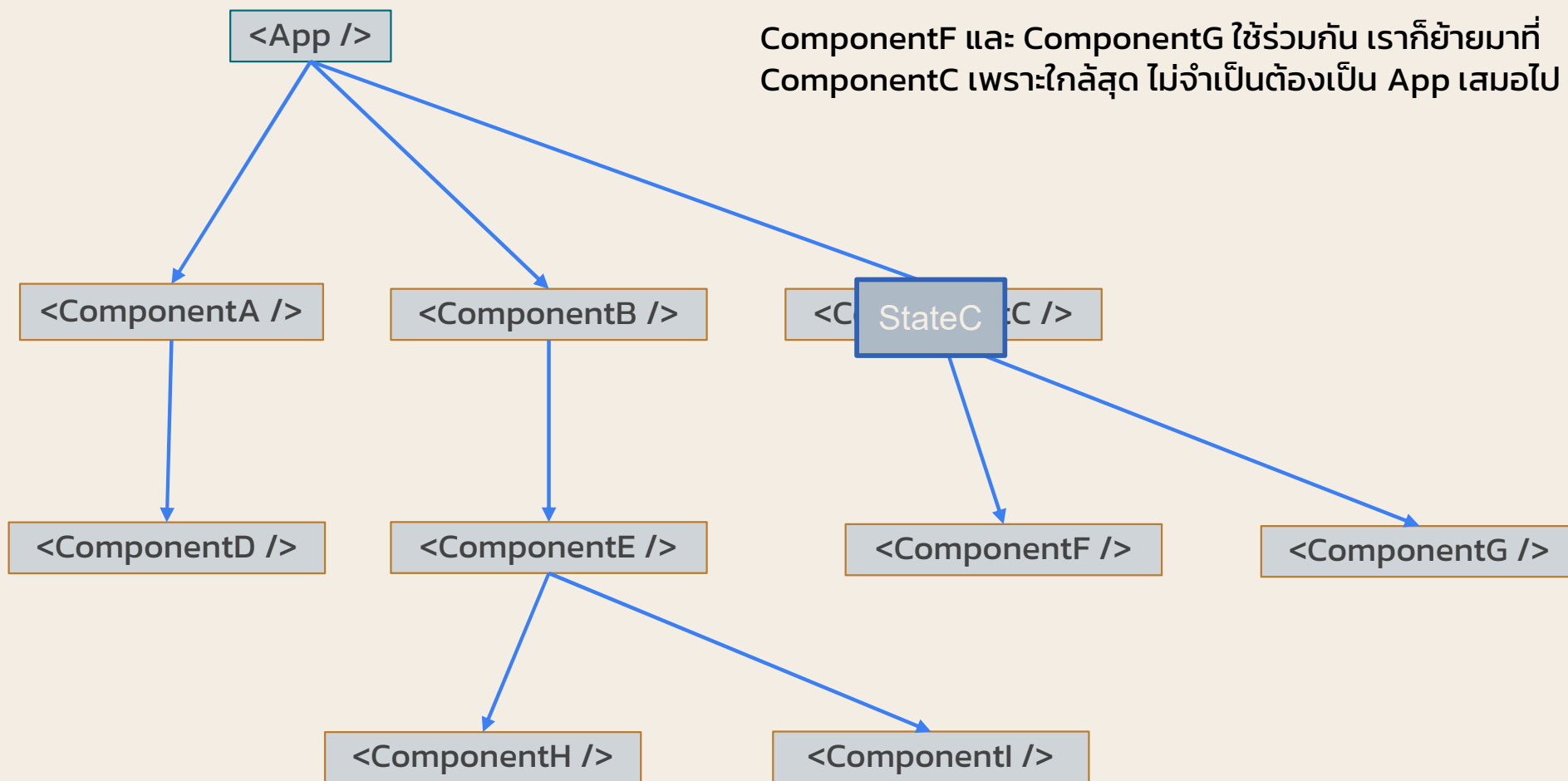
parent component ที่ใกล้ที่สุด หมายถึงยังงั้?



คอนเซปต์ Lifting State Up

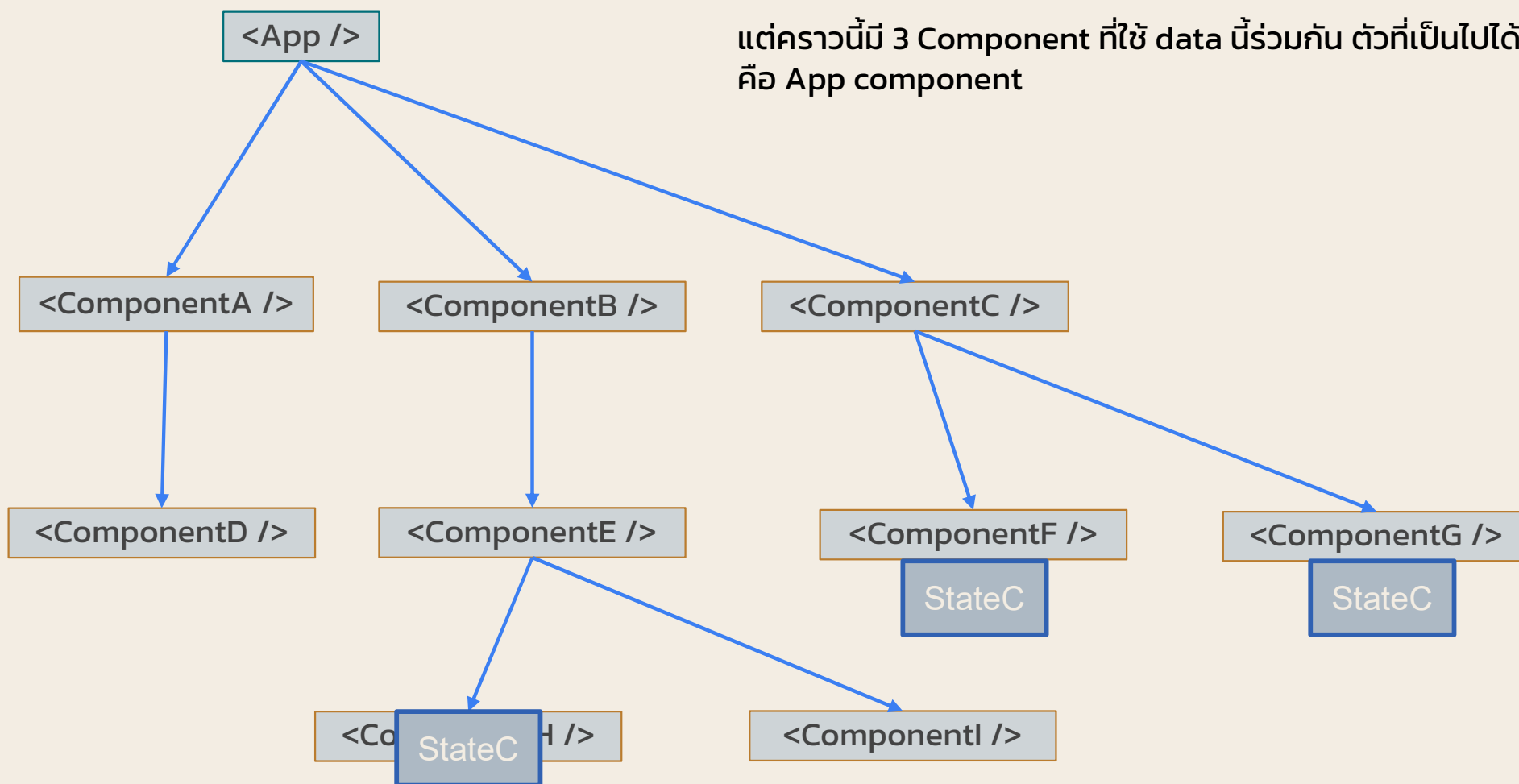


คอนเซปต์ Lifting State Up



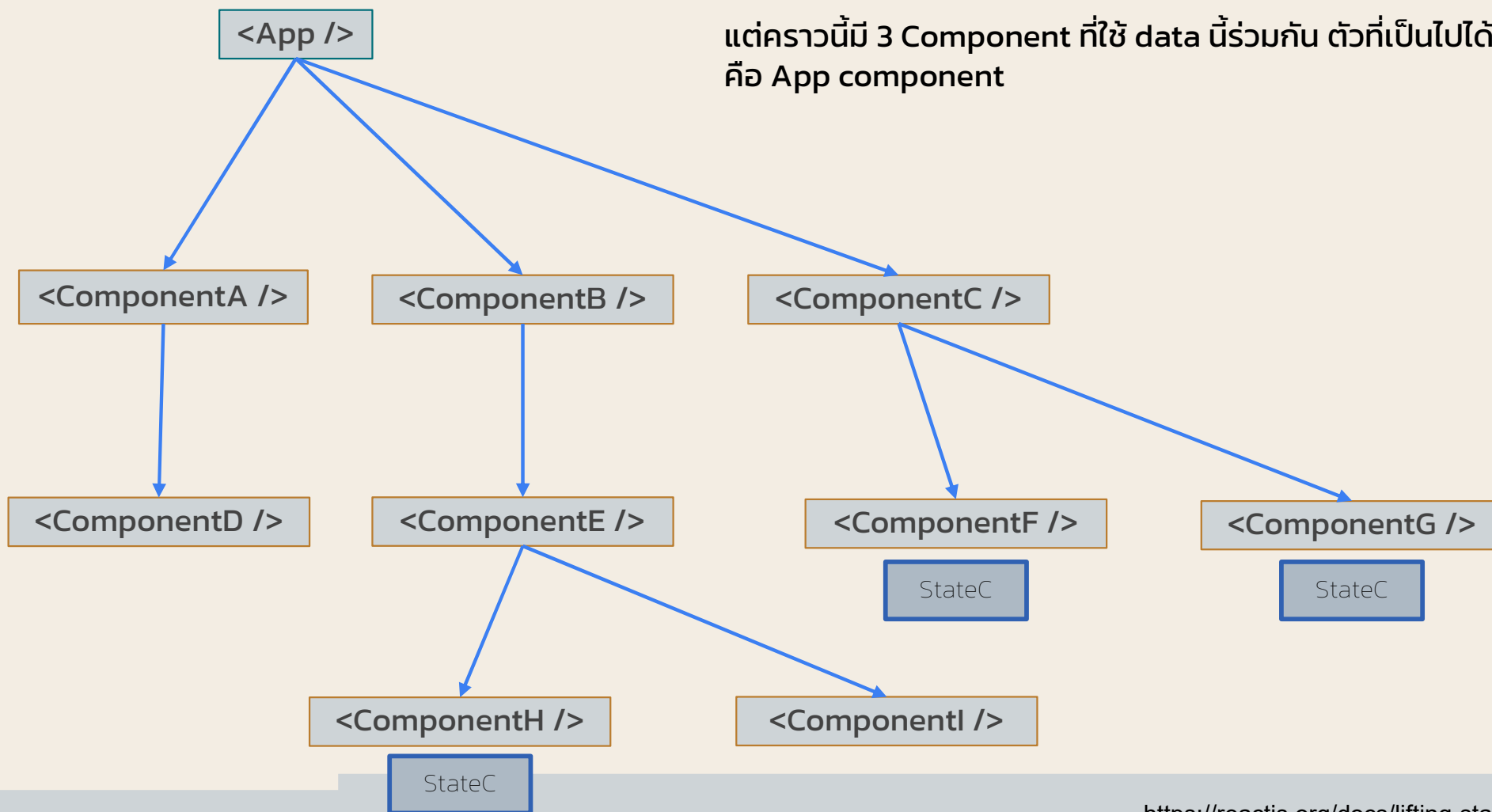
คอนเซปต์ Lifting State Up

แต่คราวนี้มี 3 Component ที่ใช้ data นี้ร่วมกัน ตัวที่เป็นไปได้ก็คือ App component



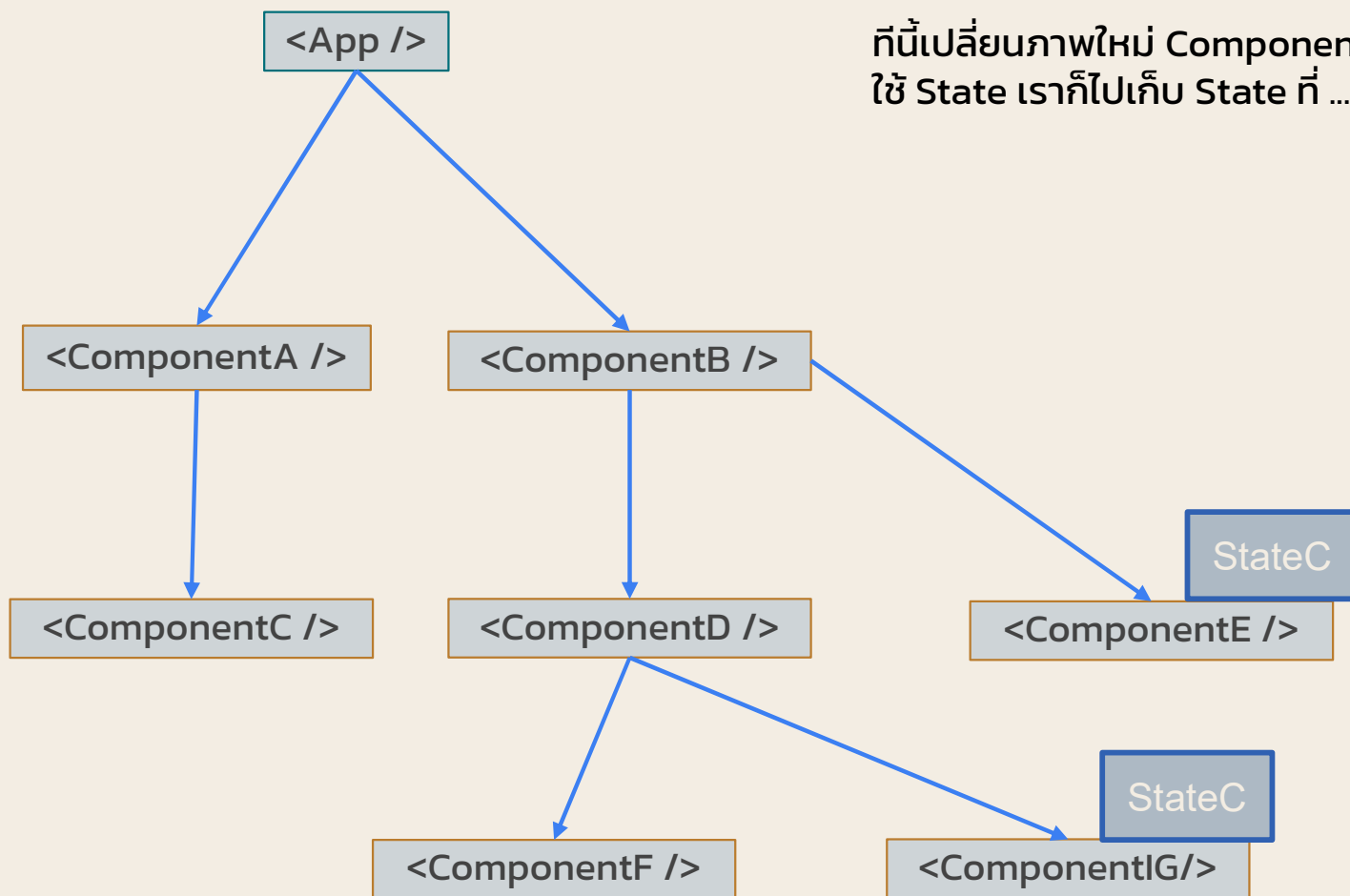
คอนเซปต์ Lifting State Up

แต่คราวนี้มี 3 Component ที่ใช้ data นี้ร่วมกัน ตัวที่เป็นไปได้ก็คือ App component



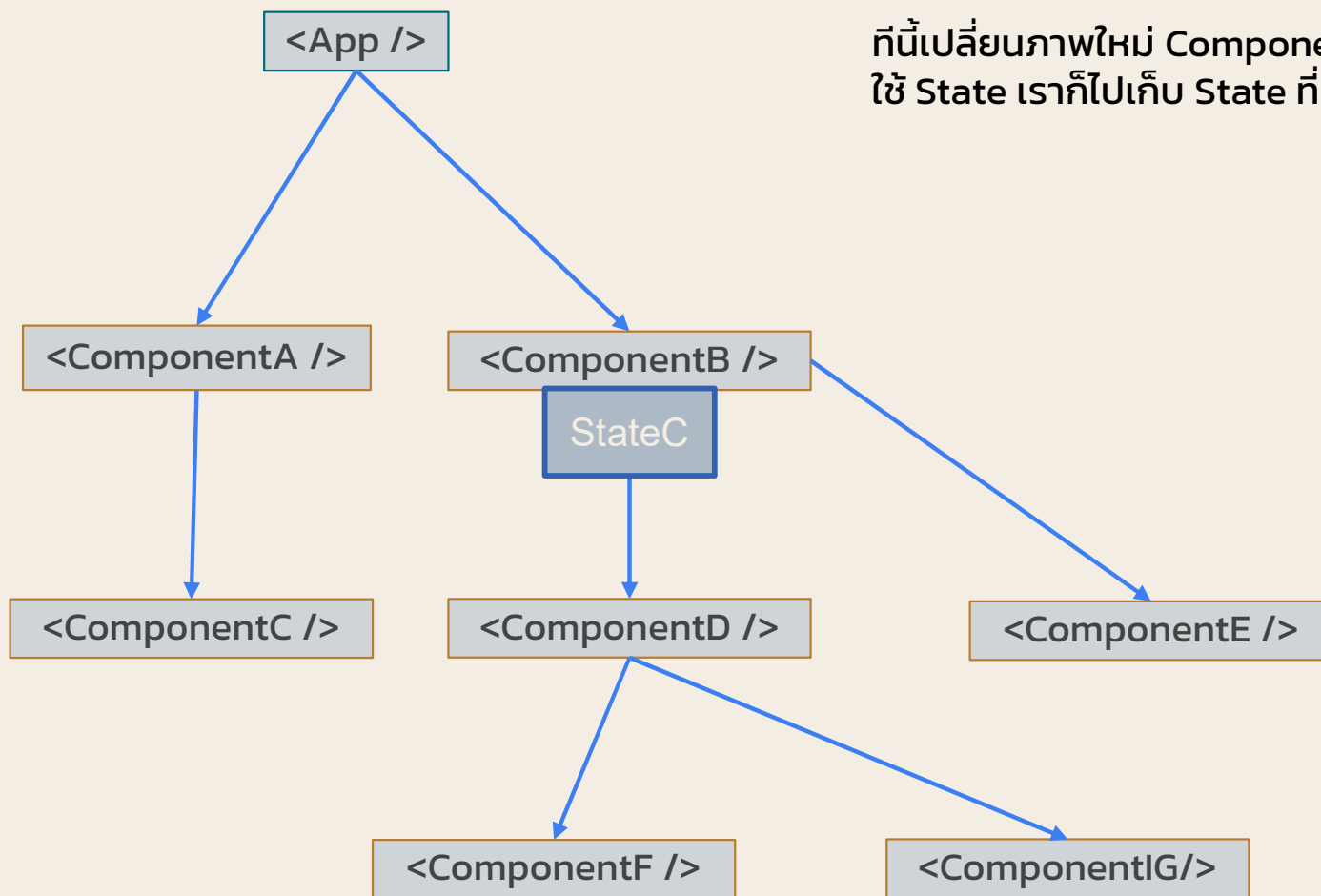
คอนเซปต์ Lifting State Up

ที่นี่เปลี่ยนภาพใหม่ ComponentE กับ ComponentG ต้องการ
ใช้ State เรา ก็ไปเก็บ State ที่ ...

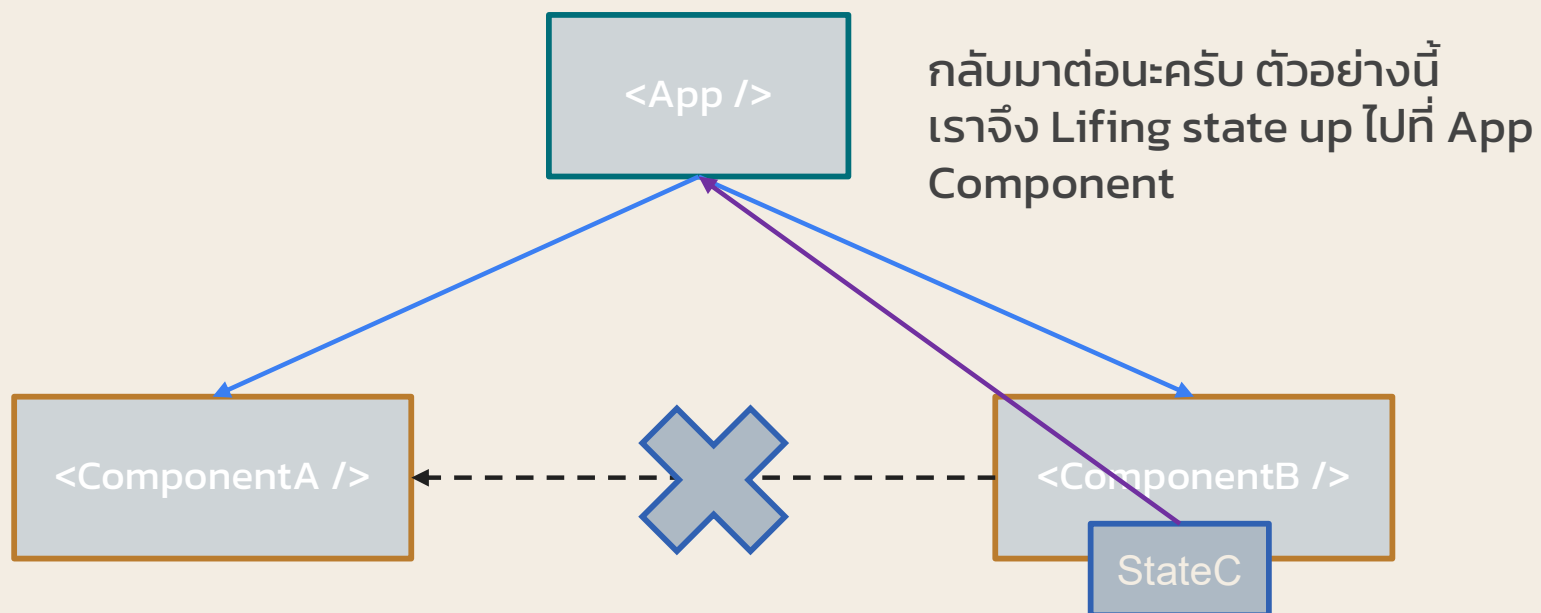


คอนเซปต์ Lifting State Up

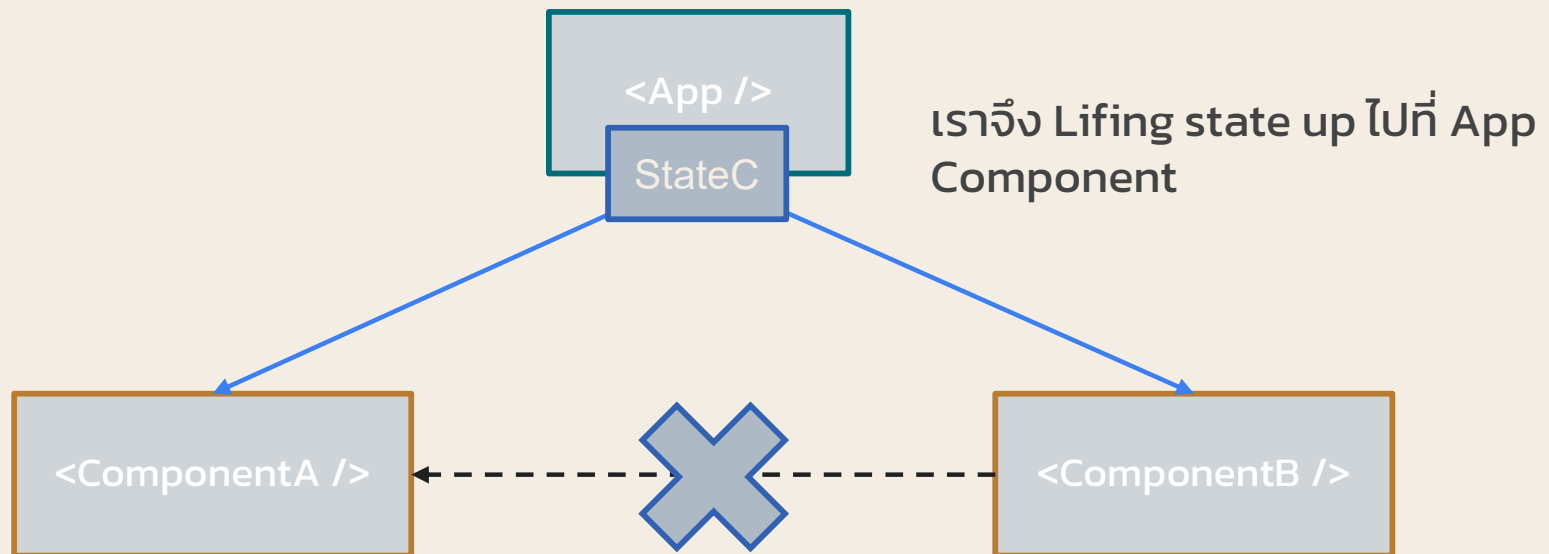
ที่นี่เปลี่ยนภาพใหม่ ComponentE กับ ComponentG ต้องการ
ใช้ State เรา ก็ไปเก็บ State ที่ ...



คอนเซปต์ Lifting State Up

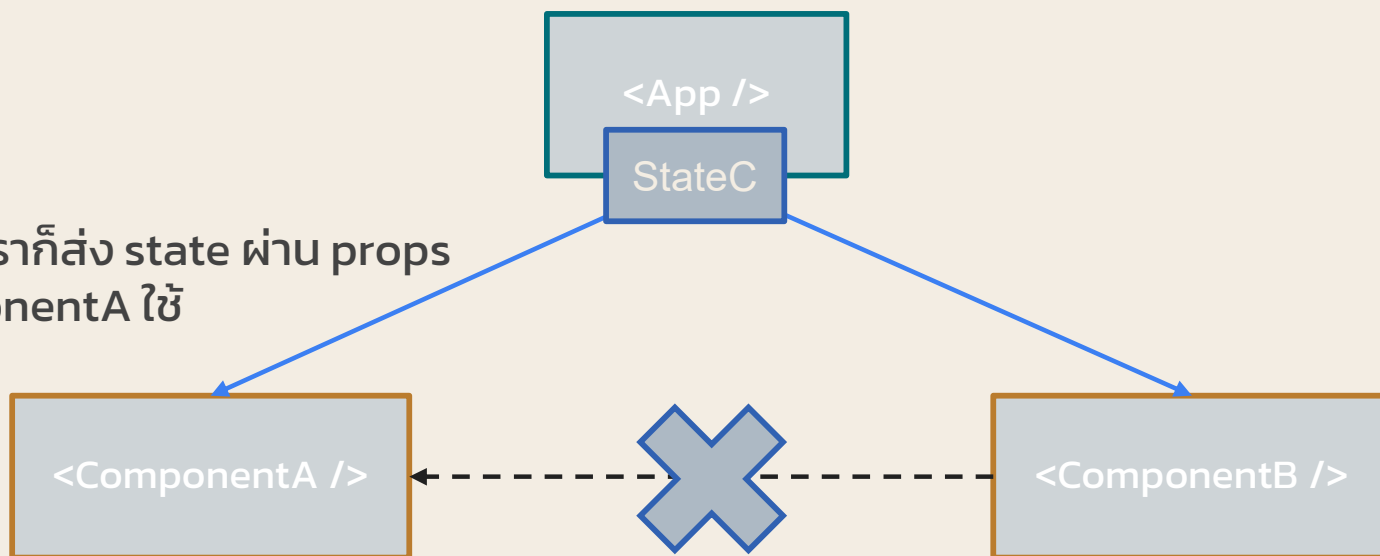


คอนเซปต์ Lifting State Up

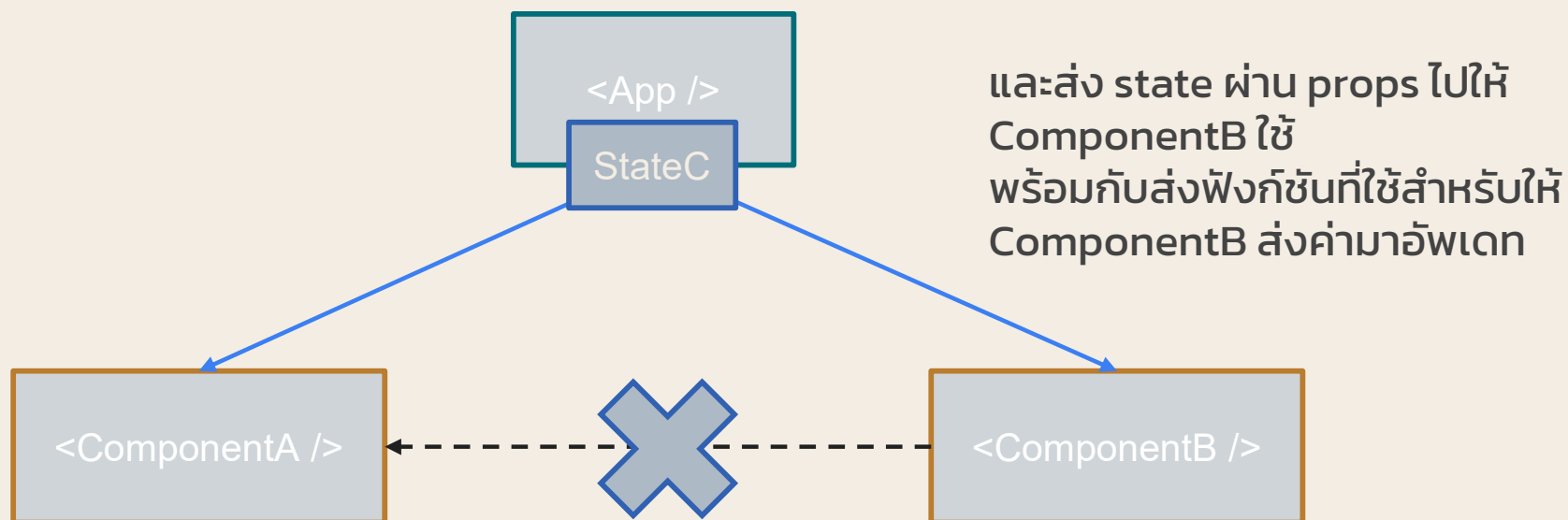


คอนเซปต์ Lifting State Up

หลังจากนั้นเราก็ส่ง state ผ่าน props
ไปให้ ComponentA ใช้



คอนเซปต์ Lifting State Up





Full workshop

Link: <https://github.com/soncomqiq/workshop-todolist-2023/tree/start-ep-2>

