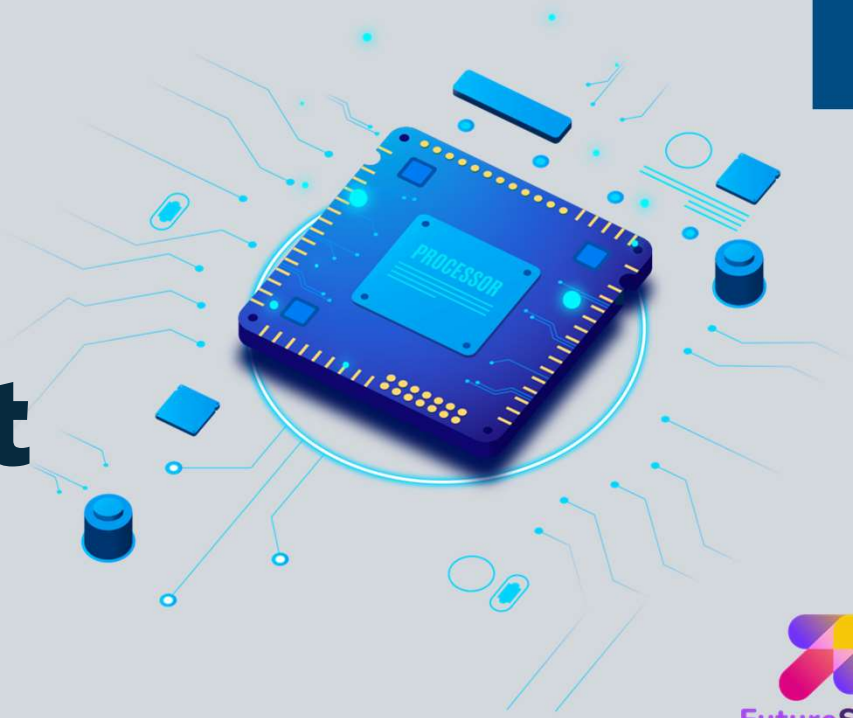
# การใช้ useEffect




FutureSkill

# Side Effect คืออะไร?

**การที่เราจะใช้ useEffect เราต้องมาเข้าใจก่อนว่า "Effect" หรือ "Side Effect" คืออะไร**

โดยปกติแล้ว React จะทำหน้าที่ในการ Render หน้าเว็บและจัดการ State ต่าง ๆ ซึ่งการกระทำอะไรก็ตามที่ไม่อยู่ในขอบเขตของ React จะเรียกว่า Side Effect หรือเรียกว่า Effect ก็ได้ เช่น

- การส่ง Request ไปขอข้อมูลจาก Server
- การใช้ API ของ Browser เช่น การเรียกออฟเจ็กต์ document และ window โดยตรง
- การใช้ฟังก์ชัน setTimeout หรือ setInterval

FutureSkill

# Side Effect คืออะไร?

**แล้วทำไมต้องแยกการกระทำบางอย่างออกมาเป็น side effect ให้มันยุ่งยาก?**

เพราะว่าการกระทำที่เป็น side effect จะไม่สามารถคาดการณ์ผลลัพธ์ได้ (unpredictable) เช่น

- การส่ง Request ไปขอข้อมูลจาก Server อาจจะสำเร็จ รอนาน (ในกรณีที่ Server ช้า) หรือ ล้มเหลว (ในกรณีที่ Server ล่ม) ก็ได้

ซึ่ง side effect ถ้ามารวมกับ code ของ react อาจจะทำให้เกิด bug หรือไปขัดขวางการ Render ของ React

React จึงมีฟังก์ชัน useEffect ให้เราใช้ เพื่อใช้สำหรับแยก code ที่เป็น Side Effect ออกมา

อ่านเพิ่มเติม: https://www.freecodecamp.org/news/react-useeffect-absolute-beginners/

FutureSkill

# การใช้ useEffect

ฟังก์ชัน useEffect

```
useEffect(() => {
    // code here
  }, [dependencies]);
```

ฟังก์ชันสำหรับใส่ Code โดยที่ ถ้า dependencies มีการเปลี่ยนแปลง ฟังก์ชันนี้จะถูกรัน (Executed) หลังจาก Component Render เสร็จ

FutureSkill

# การใช้ useEffect

ฟังก์ชัน useEffect - ตัวอย่างที่ 1 ไม่ใส่ denpendencies

โค๊ดนี้จะถูกรันเมื่อ

- ครั้งแรกที่ Component render (หรือเรียกว่า Mount)
- ทุกครั้งหลังที่ re-render เสร็จ (Component evaluated)

```
useEffect(() => {
  console.log('first render and every after rendering')
});
```

FutureSkill

# การใช้ useEffect

ฟังก์ชัน useEffect – ตัวอย่างที่ 2 ใส่เป็นอาเรย์ว่าง

โค๊ดนี้จะถูกรันเมื่อ

- ครั้งแรกที่ Component render ครั้งเดียวเท่านั้น

```
useEffect(() => {
  console.log('first render only once')
}, []);
```

FutureSkill

# การใช้ useEffect

ฟังก์ชัน useEffect – ตัวอย่างที่ 3 ใส่ denpendencies

โค๊ดนี้จะถูกรันเมื่อ

- ครั้งแรกที่ Component render
- จะถูกรันเมื่อ name หรือ age มีการเปลี่ยนแปลง

```
useEffect(() => {
  console.log(`first render and when ${state} or ${props} are updated`)
}, [name, age])
```

อ่านเพิ่มเติม : https://ihatetomatoes.net/react-hooks-tutorial-for-beginners/

FutureSkill

# การใช้ useEffect

ฟังก์ชัน useEffect - ตัวอย่างที่ 4 นำ useEffect ทั้ง 3 ตัวอย่างแรกมาใส่พร้อมกัน ทันทีที่ Component mounted (render รอบแรก) ทั้ง 3 ตัวจะถูกรันทั้งหมด เพราะว่า useEffect จะรัน 1 ครั้งตอนที่ component mounted เสมอ

```
useEffect(() => {
  console.log('first render and every after rendering')
});
```

```
useEffect(() => {
  console.log('first render only once')
}, []);
```

```
useEffect(() => {
  console.log(`first render and when ${state} or ${props} are updated`)
}, [name, age])
```

FutureSkill

# การใช้ useEffect

ฟังก์ชัน useEffect - ตัวอย่างที่ 4 นำ useEffect ทั้ง 3 ตัวอย่างแรกมาใส่พร้อมกัน

หลังจากนั้น

```
useEffect(() => {
  console.log('first render and every after rendering')
});
```

ตัวนี้จะรันทุก ๆ รอบ
ที่มีการ re-render

```
useEffect(() => {
  console.log('first render only once')
}, []);
```

```
useEffect(() => {
  console.log(`first render and when ${state} or ${props} are updated`)
}, [name, age])
```

FutureSkill

# การใช้ useEffect

ฟังก์ชัน useEffect - ตัวอย่างที่ 4 นำ useEffect ทั้ง 3 ตัวอย่างแรกมาใส่พร้อมกัน

หลังจากนั้น

```
useEffect(() => {
  console.log('first render and every after rendering')
});
```

ตัวนี้จะรันทุก ๆ รอบ
ที่มีการ re-render

```
useEffect(() => {
  console.log('first render only once')
}, []);
```

ตัวนี้จะไม่ถูกรันแล้วเพราะ
รันแค่รอบแรกรอบเดียว

```
useEffect(() => {
  console.log(`first render and when ${state} or ${props} are updated`)
}, [name, age])
```

FutureSkill

# การใช้ useEffect

ฟังก์ชัน useEffect - ตัวอย่างที่ 4 นำ useEffect ทั้ง 3 ตัวอย่างแรกมาใส่พร้อมกัน

หลังจากนั้น

```
useEffect(() => {
  console.log('first render and every after rendering')
});
```

ตัวนี้จะรันทุก ๆ รอบ
ที่มีการ re-render

```
useEffect(() => {
  console.log('first render only once')
}, []);
```

ตัวนี้จะไม่ถูกรันแล้วเพราะ
รันแค่รอบแรกรอบเดียว

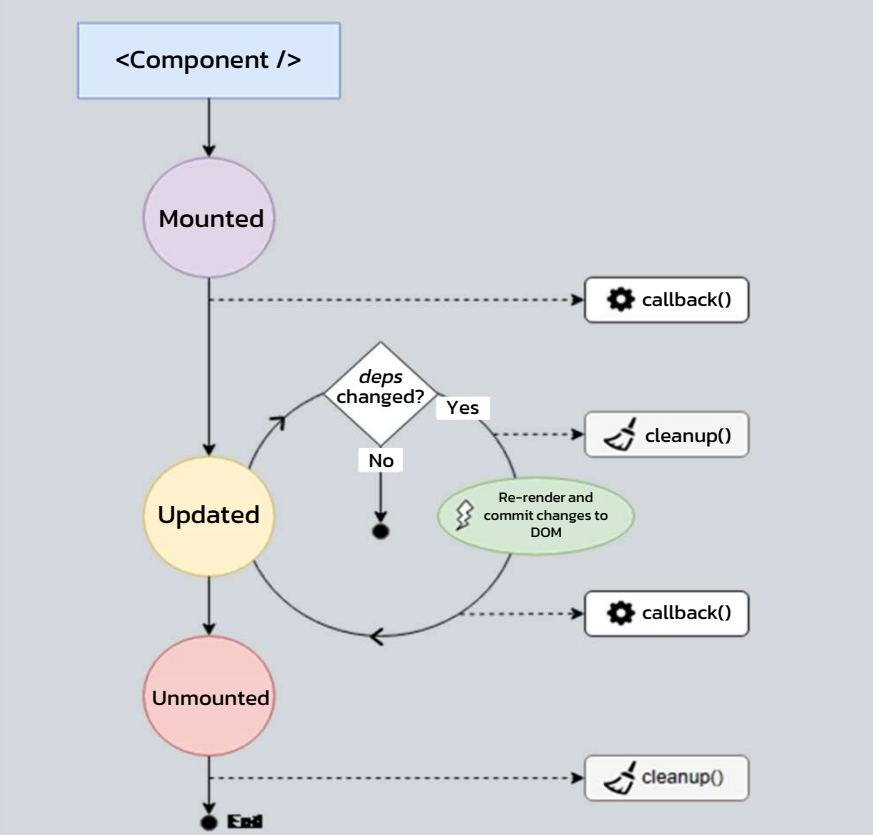
```
useEffect(() => {
  console.log(`first render and when ${state} or ${props} are updated`)
}, [name, age])
```

ตัวนี้จะถูกรันก็ต่อเมื่อ name
หรือ age มีการเปลี่ยนแปลง

FutureSkill

# Flow ของ useEffect

## ฟังก์ชัน useEffect และ Clean up




FutureSkill

# การใช้ useEffect

Mini workshop: การใช้งาน useEffect และ flow ของ useEffect

Initial code: https://github.com/soncomqiq/video-player-project

video link: https://interactive-examples.mdn.mozilla.net/media/cc0-videos/flower.mp4

Note: Lab-r4-l1

FutureSkill

# การใช้ useEffect

## ฟังก์ชัน cleanup ใน useEffect

```
useEffect(() => {
  // code here
  return () => {
    // cleanup here
  }
}, [dependencies]);
```

Cleanup function จะรันก่อน useEffect ครั้งใหม่เสมอ

FutureSkill

# การใช้ useEffect

Mini workshop: การใช้งาน Cleanup function

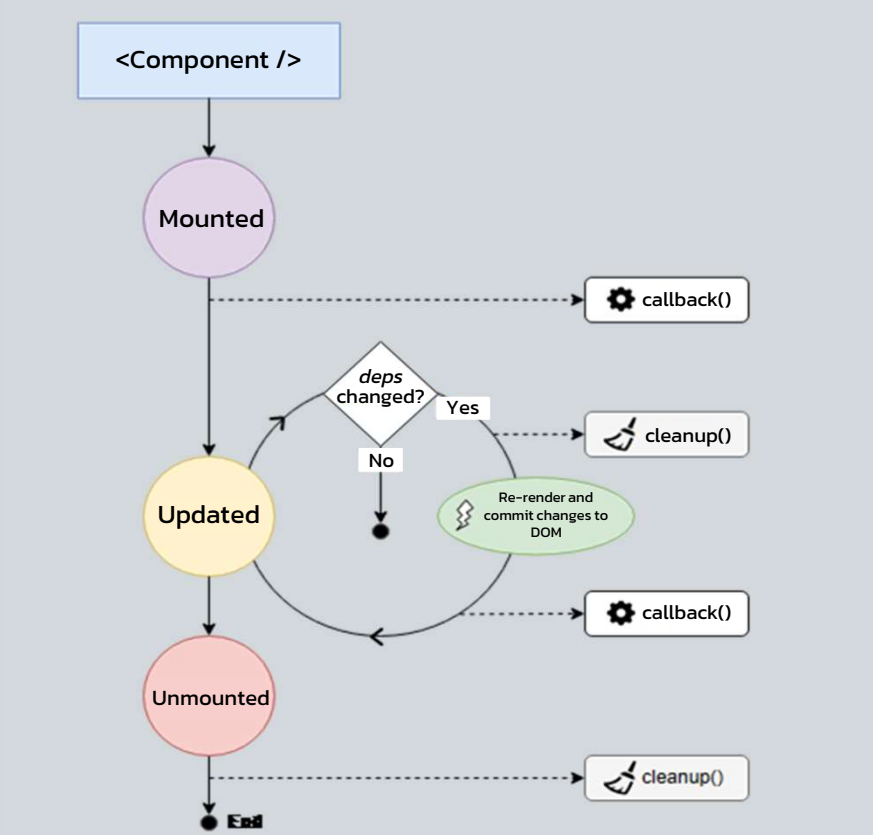
Initial code: https://github.com/soncomqiq/video-player-project/tree/start-lab2

Note: Lab-r4-l2

FutureSkill

# Flow ของ useEffect

## ฟังก์ชัน useEffect และ Clean up




FutureSkill