

Chapter 12. State 끌어올리기

교재: 처음만난 리액트 (저자: 이인제, 한빛출판사)



Contents

- CHAPTER 12: State 끌어올리기

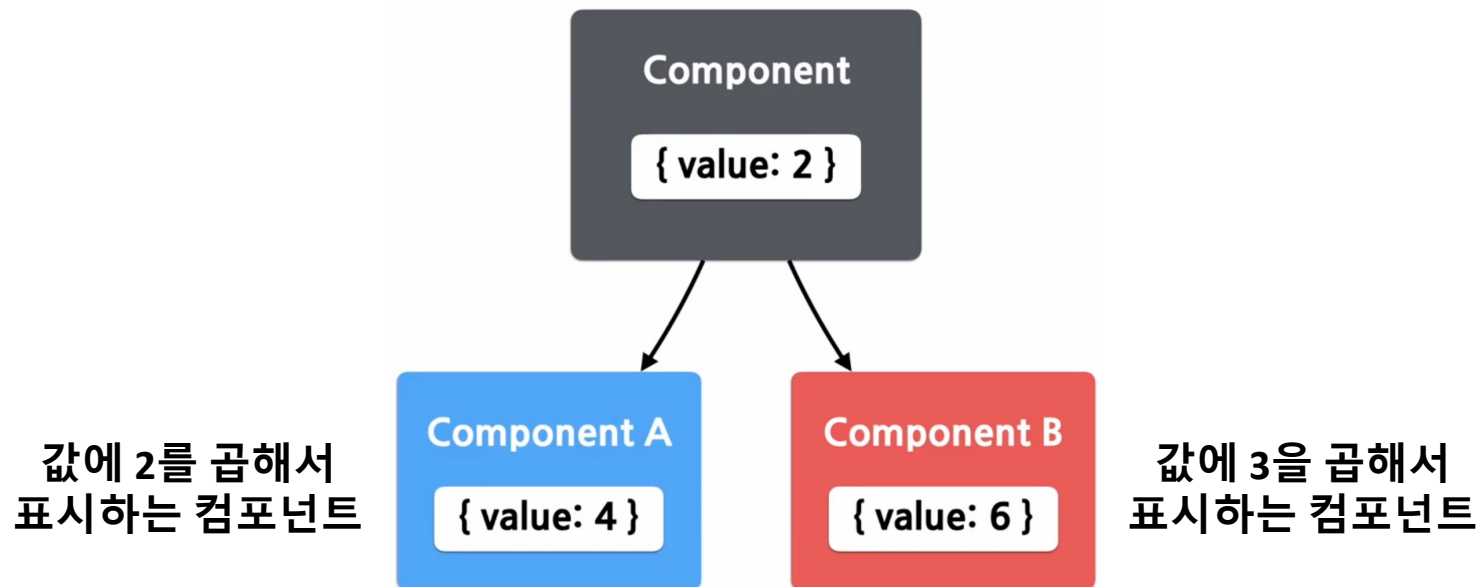
12.1 Shared State

12.2 하위 컴포넌트에서 State 공유하기

SECTION 12.1 Shared State

- shared state

- 하나의 데이터를 여러 개의 컴포넌트에서 표현해야 하는 경우
 - 각 컴포넌트의 state에서 데이터를 각각 보관하지 않고,
공통된 부모 컴포넌트의 state를 공유해서 사용하는 것이 효율적

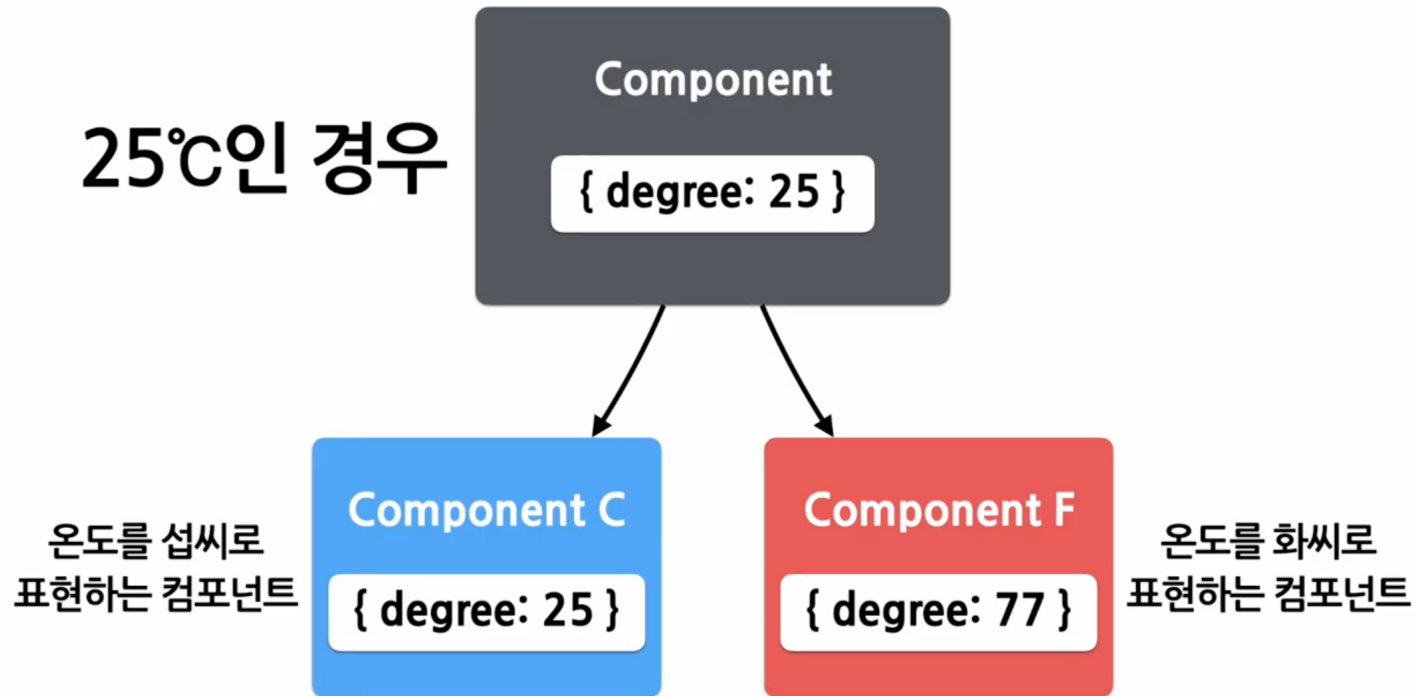


- 어떤 컴포넌트의 state에 있는 데이터를 여러 개의 하위 컴포넌트에서 공통적으로 사용

SECTION 12.1 Shared State

- **shared state**


- 하위 컴포넌트가 공통된 부모 컴포넌트의 state를 공유하여 사용



SECTION 12.2 하위 컴포넌트에서 State 공유하기

- state를 공유하는 컴포넌트 만들어 보기

1. 물의 끓음 여부를 알려주는 컴포넌트



```
function BoilingVerdict(props) {  
  if (props.celsius >= 100) {  
    return <p>물이 끓습니다.</p>  
  }  
  return <p>물이 끓지 않습니다.</p>  
}
```

SECTION 12.2 하위 컴포넌트에서 State 공유하기

- 1-1. 사용자의 입력값을 받아서 물이 끓는지 판별하는 컴포넌트

```
function Calculator(props) {  
  const [temperature, setTemperature] = useState('');  
  
  const handleChange = (event) => {  
    setTemperature(event.target.value);  
  }  
  
  return (  
    <fieldset>  
      <legend>섭씨 온도를 입력하세요.</legend>  
      <input  
        value={temperature}  
        onChange={handleChange} />  
      <BoilingVerdict  
        celsius={temperature} />  
    </fieldset>  
  )  
}
```

fieldset - HTML 양식을 그룹으로 묶을 때 사용
legend - 그룹의 설명 제공

그룹으로 묶기(legend)

이름: 제출

SECTION 12.2 하위 컴포넌트에서 State 공유하기

2. 입력 컴포넌트 추출하기

- 섭씨/화씨온도를 입력받기 위해 재사용 가능한 컴포넌트로 추출

```
const scaleNames = {
  c: '섭씨',
  f: '화씨'
};

function TemperatureInput(props) {
  const [temperature, setTemperature] = useState('');

  const handleChange = (event) => {
    setTemperature(event.target.value);
  }

  return (
    <fieldset>
      <legend>온도를 입력해 주세요(단위:{scaleNames[props.scale]})</legend>
      <input value={temperature} onChange={handleChange} />
    </fieldset>
  )
}
```

SECTION 12.2 하위 컴포넌트에서 State 공유하기

2-1. 추출한 입력 컴포넌트 적용

```
import TemperatureInput from "../TemperatureInput";


function Calculator(props) {
  return (
    <div>
      <TemperatureInput scale="c" />
      <TemperatureInput scale="f" />
    </div>
  )
}
```

- 사용자가 입력하는 온도값이 TemperatureInput의 state에 저장되기 때문에 두 개의 값이 다름

SECTION 12.2 하위 컴포넌트에서 State 공유하기

3. 온도 변환 함수 작성하기

- 섭씨온도와 화씨온도 값을 동기화시키기 위한 변환 함수 작성 (Calculator 컴포넌트)



```
function toCelsius(fahrenheit) {  
  return (fahrenheit - 32) * 5 / 9;  
}
```

```
function toFahrenheit(celsius) {  
  return (celsius * 9 / 5) + 32;  
}
```

```
function tryConvert(temperature, convert) {  
  const input = parseFloat(temperature);  
  if (Number.isNaN(input)) {  
    return '';  
  }  
  const output = convert(input);  
  const rounded = Math.round(output * 1000) / 1000;  
  return rounded.toString();  
}
```

`tryConvert('10.2', toFahrenheit);`

SECTION 12.2 하위 컴포넌트에서 State 공유하기

4. Shared State 적용하기

- 하위 컴포넌트의 state를 공통된 부모 컴포넌트로 올려서 shared state를 적용
- State 끌어올리기(Lifting State Up)

```
function TemperatureInput(props) {  
  // const [temperature, setTemperature] = useState('');  
  
  const handleChange = (event) => {  
    // 변경 전: setTemperature(event.target.value);  
    props.onTemperatureChange(event.target.value);  
  }  
  
  return (  
    <fieldset>  
      <legend>온도를 입력해 주세요(단위:{scaleNames[props.scale]})</legend>  
      { /* 변경 전: <input value={temperature} onChange={handleChange} /> */ }  
      <input value={props.temperature} onChange={handleChange} />  
    </fieldset>  
  )  
}
```

SECTION 12.2 하위 컴포넌트에서 State 공유하기

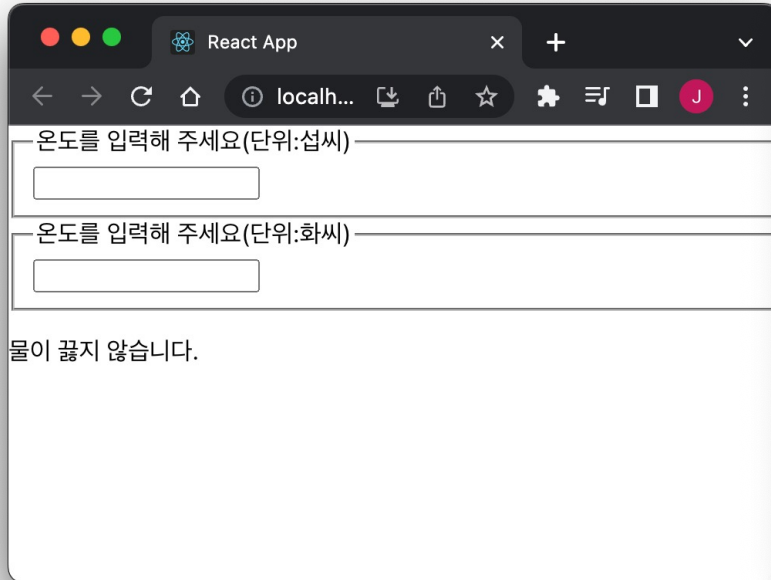
5. Calculator 컴포넌트 변경하기

```
function Calculator(props) {  
  const [temperature, setTemperature] = useState('');  
  const [scale, setScale] = useState('c');  
  
  const handleCelsiusChange = (temperature) => {  
    setTemperature(temperature);  
    setScale('c');  
  }  
  
  const handleFahrenheitChange = (temperature) => {  
    setTemperature(temperature);  
    setScale('f');  
  }  
  
  const celsius = scale === 'f' ?  
    tryConvert(temperature, toCelsius) : temperature;  
  const fahrenheit = scale === 'c' ?  
    tryConvert(temperature, toFahrenheit) : temperature;
```

```
  return (  
    <div>  
      <TemperatureInput  
        scale="c"  
        temperature={celsius}  
        onTemperatureChange={handleCelsiusChange} />  
      <TemperatureInput  
        scale="f"  
        temperature={fahrenheit}  
        onTemperatureChange={handleFahrenheitChange} />  
      <BoilingVerdict  
        celsius={parseFloat(celsius)} />  
    </div>  
  )  
}
```

SECTION 12.2 하위 컴포넌트에서 State 공유하기

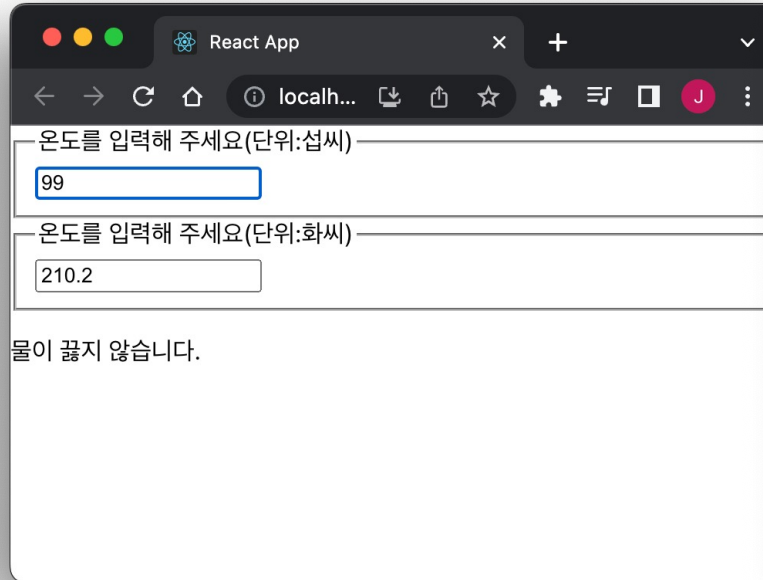
◦ 실행 결과



온도를 입력해 주세요(단위:섭씨) —

온도를 입력해 주세요(단위:화씨) —

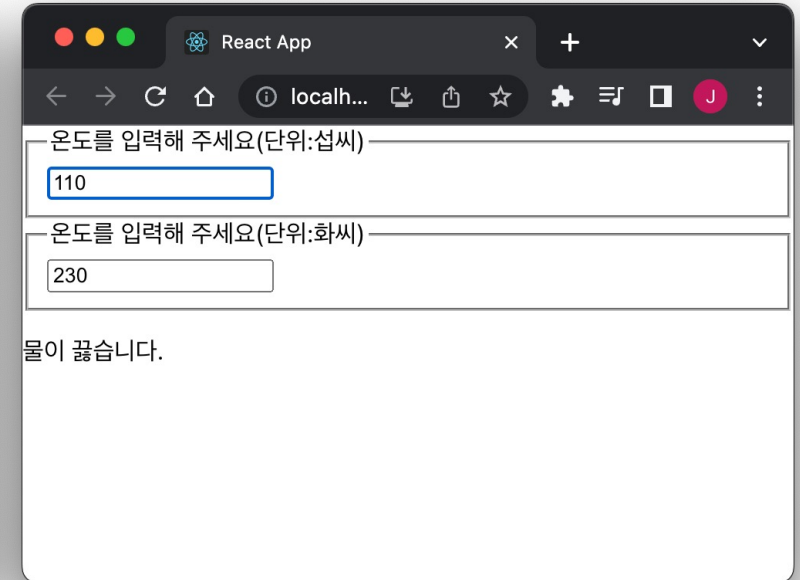
물이 끓지 않습니다.



온도를 입력해 주세요(단위:섭씨) —

온도를 입력해 주세요(단위:화씨) —

물이 끓지 않습니다.



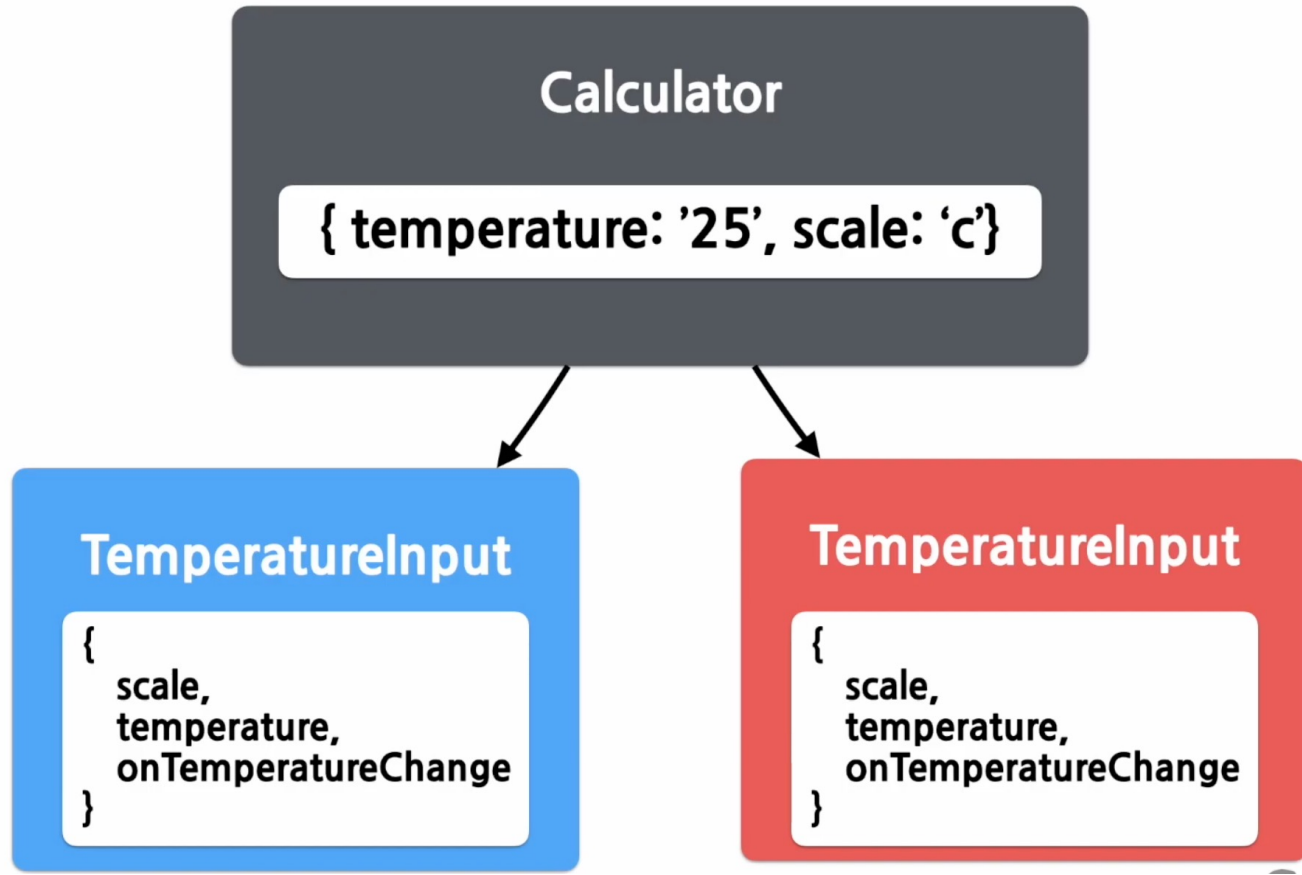
온도를 입력해 주세요(단위:섭씨) —

온도를 입력해 주세요(단위:화씨) —

물이 끓습니다.

SECTION 12.2 하위 컴포넌트에서 State 공유하기

- Lifting state up



상위 컴포넌트 state
- 온도, 단위

하위 컴포넌트 props
- 단위, 변환된 온도, 온도를 업데이트 하기 위한 함수

- 각 컴포넌트가 state 값을 갖고 있지 않고, 공통된 상위 컴포넌트로 올려서 공유하는 방법



[요약]

- Shared state
 - 하위 컴포넌트가 공통된 부모 컴포넌트의 state를 공유하여 사용하는 것
- state 끌어올리기
 - 하위 컴포넌트의 state를 공통된 부모 컴포넌트로 끌어올려서 공유하는 방식