



## #3 STATE

### 3.0 Understanding State

어플리케이션이 시작할때 렌더링을 1번만 하기때문에  
counter++; 할때마다 다시 렌더링 해주어야 함!!!

하지만 우리가 원하는 최고의 방법은 아님!

React JS는 바뀐부분만 업데이트 해줌!!!!

counter만 바뀌고 Total click: 같은 부분은 안 바뀜!

```
const root = document.getElementById("root");
let counter = 0;
function countUp() {
  counter++;
  render();
}
const Container = () => ( // arrow function
  <div>
    <h3>Total click: {counter}</h3>
    <button onClick={countUp}>Click me</button>
  </div>
);
function render(){
  ReactDOM.render(<Container/>, root);
}
render();
```

### 3.1 setState part One

```
const animal = ["cat", "dog", "cow"];
```

```
const [Cat, Dog, Cow] = animal;
```

하면 자동으로 연성됨

React.useState(0); 을 사용할거고 자세한 건 다음 강좌에서

## 3.2 setState part Two

let [counter, setCounter] = React.useState(0);      중요중요

setCounter(counter + 1); 호출함으로써 자동으로 재렌더링 가능하게 함!

```
const root = document.getElementById("root");
function App() {
  let [counter, setCounter] = React.useState(0);
  const onClick = () => {
    setCounter(counter + 1);
  };
  return (
    <div>
      <h3>Total click: {counter}</h3>
      <button onClick={onClick}>Click me</button>
    </div>
  );
}
ReactDOM.render(<App />, root);
```

## 3.3 Recap

전에 한 것 들 요약 (recap 뜻 : 요약)

## 3.4 State Functions

1. setCounter(counter + 1);
2. setCounter((current) => current + 1);

전에는 첫번째 방식 처럼 사용했지만 두번째 방식이 좀 더 안전함  
현재의 변수를 기반으로해서 다음 변수를 생성해주기 때문

## 3.5 Inputs and State

입력폼을 **label** 태그와 함께 사용하면 원하는 input 태그 등의 입력폼을 체크하거나 언체크 가능

event 라는 가짜 함수를 실행하면 event.target.value안에 우리가 원하는 값이 들어있음

react, reactdom을 import하는 script tag에서

production은 배포 모드, development는 개발 모드를 의미합니다.

개발모드는 버그로 이어질 수 있는 요소들을 미리 경고하는 검증 코드가 포함되어 있습니다.

class나 for 같은 문법은 javascript문법이라서 JSX에서는 className과 htmlFor로 사용해야 함

```
const root = document.getElementById("root");
function App() {
  const [minutes, setMinutes] = React.useState();
  const onChange = (event) => {
    setMinutes(event.target.value);
  };
  return (
    <div>
      <h1 className="hi">Super Converter</h1>
      <label htmlFor="minutes">Minutes </label>
      <input
        id="minutes"
        value={minutes}
        placeholder="Minutes"
        type="number"
        onChange={onChange}
      />
      <h4>You want to conver {minutes}</h4>
      <label htmlFor="hours">Hours </label>
      <input id="hours" placeholder="Hours" type="number"/>
    </div>
  );
}
ReactDOM.render(<App />, root);
```

## 3.6 State Practice part One

Math.round()를 이용해 반올림 할 수 있음.

```
const root = document.getElementById("root");
function App() {
  const [minutes, setMinutes] = React.useState();
  const onChange = (event) => setMinutes(event.target.value);
  const reset = () => setMinutes(0);
  return (
    <div>
      <h1>Super Converter</h1>
      <label htmlFor="minutes">Minutes </label>
      <input
        id="minutes"
        value={minutes}
        placeholder="Minutes"
        type="number"
      />
    </div>
  );
}
```

```

        onChange={onChange}
      />
    <br />
    <label htmlFor="hours">Hours </label>
    <input
      id="hours"
      value={Math.round(minutes / 60)}
      placeholder="Hours"
      type="number"
      disabled
    />
    <br />
    <button onClick={reset}>Reset</button>
  </div>
);
}
ReactDOM.render(<App />, root);

```

## 3.7 State Practice part Two

flip하는 과정에서 (A ? True : False) 삼항연산자 배움

```

function App() {
  const [amount, setAmount] = React.useState();
  const [flipped, setFlipped] = React.useState(false);
  const onChange = (event) => setAmount(event.target.value);
  const reset = () => setAmount(0);
  const onFlip = () => {
    reset();
    setFlipped((current) => !current)
  };
  return (
    <div>
      <h1>Super Converter</h1>
      <label htmlFor="minutes">Minutes </label>
      <input
        id="minutes"
        value={flipped ? amount * 60 : amount}
        placeholder="Minutes"
        type="number"
        onChange={onChange}
        disabled={flipped === true}
      />
      <br />
      <label htmlFor="hours">Hours </label>
      <input
        id="hours"
        value={flipped ? amount : Math.round(amount / 60)}
        placeholder="Hours"
        type="number"
        onChange={onChange}
        disabled={flipped === false}
      />
    </div>
  );
}

```

```

    <br />
    <button onClick={reset}>Reset</button>
    <button onClick={onFlip}>Flip</button>
  </div>
);
}

```

## 3.8 Recap

이때까지 한 것 들 요약 (recap 뜻 : 요약)

## 3.9 Final Practice and Recap

이번시간에는 함수를 작은 컴포넌트로 여기고 요소로 추가해보는 시간을 가졌다!

```

<!DOCTYPE html>
<html>
  <body id="root">
  </body>
  <script src="https://unpkg.com/react@17.0.2/umd/react.production.min.js"></script>
  <script src="https://unpkg.com/react-dom@17.0.2/umd/react-dom.production.min.js"></script>
  <script src="https://unpkg.com/@babel/standalone/babel.min.js"></script>
  <script type="text/babel">
    function MinutesToHours() {
      const [amount, setAmount] = React.useState();
      const [inverted, setInverted] = React.useState(false);
      const onChange = (event) => setAmount(event.target.value);
      const reset = () => setAmount(0);
      const onFlip = () => {
        reset();
        setInverted((current) => !current)
      };
      return (
        <div>
          <label htmlFor="minutes">Minutes </label>
          <input
            id="minutes"
            value={inverted ? amount * 60 : amount}
            placeholder="Minutes"
            type="number"
            onChange={onChange}
            disabled={inverted}
          />
          <br />
          <label htmlFor="hours">Hours </label>
          <input
            id="hours"
            value={inverted ? amount : Math.round(amount / 60)}

```

```

        placeholder="Hours"
        type="number"
        onChange={onChange}
        disabled={!inverted}
      />
      <br />
      <button onClick={reset}>Reset</button>
      <button onClick={onFlip}>Flip</button>
    </div>
  );
}
function KmToMiles(){
  return <h3>KM 2 M</h3>;
}
function App() {
  const [index, setIndex] = React.useState("xx");
  const onSelect = (event) => {
    setIndex(event.target.value);
  }
  console.log("render w/", index);
  return (
    <div>
      <h1>Super Converter</h1>
      <select value={index} onChange={onSelect}>
        <option value="xx">Select your units</option>
        <option value="0">Minutes & Hours</option>
        <option value="1">Km & Miles</option>
      </select>
      <hr />
      {index === "xx" ? "Please select your units" : null}
      {index === "0" ? <MinutesToHours /> : null}
      {index === "1" ? <KmToMiles /> : null}
    </div>
  );
}
const root = document.getElementById("root");
ReactDOM.render(<App />, root);
</script>
</html>

```