3. functional 컴포넌트에서 state 사용법_Hook <중요!> (March 7th, 21)

functional 컴포넌트에서 기본적인 Hook 사용법

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

function Example() {
  // "count"라는 새로운 상태 값을 정의합니다.
  const [count, setCount] = useState(0);

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

functional 컴포넌트에서 state를 사용하고 싶을 때, hook을 이용하고, hook을 이용할 땐 useState를 사용한다.

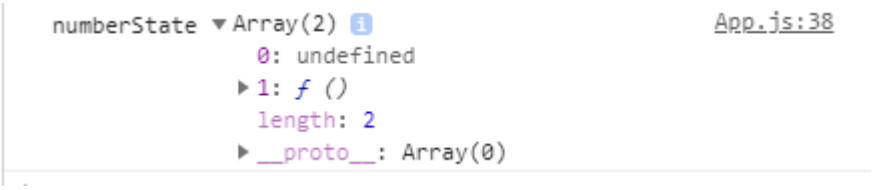
useState를 사용해서 **가져온 변수를 새로 선언**해줘야한다.

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

function FuncComp(props){
  var numberState = useState();  # 가져온 변수를 새로 선언
  return(
    <div className="container">
      <h2>function style component</h2>
      <p>Number : {props.initNumber}</p>
    </div>
```

```
  );
}
```

useState()의 return값은 **두 개의 원소로 이루어진 리스트**로 주어진다.



초기에 따로 설정하지 않았을 때 주어진 리스트의 0번 원소는 undefined이고,

리스트의 0번 원소를 useState()로 호출하는 변수로 선언해줄 때, **functional 컴포넌트가 hook기법을 이용하여 state를 사용할 수 있게 된다**.

```
function FuncComp(props){
  var numberState = useState(props.initNumber); #0번 리스트를 호출할 변수로 선언
  var number = numberState[0];
  return(
    <div className="container">
      <h2>function style component</h2>
      <p>Number : {props.initNumber}</p>
    </div>
  );
}
```

class형 컴포넌트에서 랜덤 버튼을 눌러서 number라는 state를 바꾸고, state가 바뀔 때마다 render()를 실행시킨 것을 확인한 것과 동일한 과정을 진행해보겠다.

useState()가 return하는 두 개의 원소로 이루어진 리스트의 1번 원소가 state를 바꾸는 함수를 담고 있다.

```
function FuncComp(props){
  var numberState = useState(props.initNumber);
  var number = numberState[0];
  var setNumber = numberState[1]; # state를 바꾸는 리스트의 1번 원소를 이용하기 위해
  return(                         # 새로운 이름의 변수로 초기화한다.
    <div className="container">
      <h2>function style component</h2>
      <p>Number : {number}</p>
    </div>
  );
}
```

이제 랜덤 버튼을 만들고 변수를 설정한다.

```
function FuncComp(props){
  var numberState = useState(props.initNumber);
  var number = numberState[0];
  var setNumber = numberState[1];
  return(
    <div className="container">
      <h2>function style component</h2>
      <p>Number : {number}</p>
      <input type="button" value="random" onClick={
          function(){
            this.setState({number:Math.random()})
          }.bind(this)
    </div>
  );
}
```

'this'문법은 클래스 안에서 사용하기 때문에 없애준다.

```
function FuncComp(props){
  var numberState = useState(props.initNumber);
  var number = numberState[0];
  var setNumber = numberState[1];
  return(
    <div className="container">
      <h2>function style component</h2>
      <p>Number : {number}</p>
      <input type="button" value="random" onClick={
          function(){
            setNumber(Math.random()); # class형 컴포넌트에서 this.setState(number)로 쓰던 것이
          }                           # functional에서 useState[1].(useState[0])으로 달라짐
        }></input>
    </div>
```

```
    );
}
```

마지막으로 **날짜(시간)를 출력**하는 요소를 class형과 functional 컴포넌트에서 만들어보자.

+클릭할 때 마다 **날짜(시간)가 갱신되는 date 버튼을 추가**한다.

A. Class형 컴포넌트

```
class ClassComp extends React.Component{
  state = {
    number:this.props.initNumber,
    date:(new Date()).toString()                          # 날짜 표시를 위해 'date' 정의
  }
  render(){
    return (
      <div className="container">
        <h2>class style component</h2>
        <p>Number : {this.state.number}</p>

        <p>Date : {this.state.date}</p>                   # 날짜 표시

        <input type="button" value="random" onClick={
          function(){
            this.setState({number:Math.random()})
          }.bind(this)
        }></input>

        <input type="button" value="date" onClick={
          function(){
            this.setState({date:(new Date()).toString()}) # Date을 업데이트하는
          }.bind(this)                                    # 버튼 추가
        }></input>

      </div>
    )
  }
}
```

B. functional

```
function FuncComp(props){
  var numberState = useState(props.initNumber);
```

```
  var number = numberState[0];
  var setNumber = numberState[1];

  var dateState = useState((new Date()).toString());  #날짜 state에 hook기법을 이용하기
  var _date = dateState[0];                           #위해 필요한 변수 선언
  var setDate = dateState[1];

  return(
    <div className="container">
      <h2>function style component</h2>
      <p>Number : {number}</p>
      <p>Date : {_date}</p>
      <input type="button" value="random" onClick={
          function(){
            setNumber(Math.random());
          }
        }></input>

      <input type="button" value="date" onClick={    #date 버튼 생성
          function(){
            setNumber((new Date()).toString());
          }
        }></input>
    </div>
  );
}
```

```
  // var dateState = useState((new Date()).toString());  # 이 코드를
  // var _date = dateState[0];                           # 아래와 같이 줄여쓸 수 있다
  // var setDate = dateState[1];

  var [_date, setDate] = useState((new Date()).toString());
```