

2. Class형 컴포넌트에서 state사용법<중요!> (March 7th, 21)

컴포넌트는, 컴포넌트를 사용하는 입장에서는 컴포넌트가 제공하는 props를 통해서 컴포넌트를 이요할 수 있고,

컴포넌트를 만드는 쪽에서는 내부적으로 state라고하는, props와 구분되는 data를 통해서 내부의 여러가지 작업들을 할 수 있다.

props는 class형식, function형식 양쪽에서 사용할 수 있다.

하지만, state는 class 형식에서만 사용할 수 있었다.

Step1. function컴포넌트와 class컴포넌트에 initNumber=2 라는 변수를 선언했을 때 각각 어떻게 이변수를 가져오는지 보자

App.js

```
function App() {  
  return (  
    <div className="App">  
      Hello World  
      <FuncComp initNumber={2}></FuncComp>  
      <ClassComp initNumber={2}></ClassComp>  
    </div>  
  );  
}
```

App.js

```
class ClassComp extends React.Component{  
  render(){  
    return (  
      <div className="container">  
        <h2>class style component</h2>  
        <p>Number : {this.props.initNumber}</p>  
      </div>  
    )  
  }  
}
```

```
}  
}
```

class 컴포넌트는 **this.props.initNumber**를 통해 숫자 2를 가져오고,

>>>

Hello World

function style component

class style component

Number : 2

App.js

```
function FuncComp(props){  
  return(  
    <div className="container">  
      <h2>function style component</h2>  
      <p>Number : {props.initNumber}</p>  
    </div>  
  );  
}
```

functional 컴포넌트는 **parameter**를 이용하여 위에서 선언된 변수를 가져온다.

(parameter이름은 자유롭게 선언가능)

>>>

Hello World

function style component

Number : 2

class style component

Number : 2

Step2. state를 class형식의 컴포넌트와 functional 컴포넌트 각각에서 어떻게 사용하는지 보자.

A. Class

⇒ 변수의 값을 props로 바로 받아오지 않고, state로 받아와서 state의 값이 바뀔 때 마다 render가 호출되도록 한다.

```
class ClassComp extends React.Component{
  state = {
    number: this.props.initNumber
  }
  render(){
    return (
      <div className="container">
        <h2>class style component</h2>
        <p>Number : {this.props.number}</p>
      </div>
    )
  }
}
```

class컴포넌트에서 **state**를 사용하기 위해서, 우선 **state**의 초기값을 선언해줘야한다.

```
state = {                                #state의 초기값 선언
  number: this.props.initNumber         #number라는 state의 초기값을 initNumber로 초기화
}
```

이제, 받아온 **number**라는 **state**를 **class** 컴포넌트 안에서 사용할 수 있다.

```
class ClassComp extends React.Component{
  state = {
    number: this.props.initNumber
  }
  render(){
    return (
      <div className="container">
        <h2>class style component</h2>
        <p>Number : {this.state.number}</p>
      </div>
    )
  }
}
```

이렇게 했을 때,

변수의 값이 바뀔 때 마다 (결과적으로 state의 값이 바뀔 때 마다)

render가 진행되는지 확인하기 위해서,

버튼을 클릭할 때 마다 number의 값이 랜덤하게 변하는 장치를 하나 만들어 눈으로 확인해보자.

```
class ClassComp extends React.Component{
  state = {
    number: this.props.initNumber
  }
  render(){
    return (
      <div className="container">
        <h2>class style component</h2>
        <p>Number : {this.state.number}</p>

        <input type="button" value="random" onClick={ # 클릭할 때 마다 function이 실행되는
          function(){                                # "random"글씨가 써있는 "버튼" 생성
            this.setState({number: Math.random()})  # function이 실행되면, number 변수를
          }.bind(this)                               # 난수로 바꾸고 state을 업데이트한다.
        }>
      </div>
    )
  }
}
```

```
    }></input>

    </div>
  )
}
}
```

버튼을 클릭해서 number 변수값이 랜덤하게 바뀌면서 state가 업데이트되면,
state값이 바뀌었으므로 render()가 실행된다.

>>

Hello World

function style component

Number : 2

class style component

Number : 0.6613802642562927

random

결론 : Class형 컴포넌트는 변수의 값을 props로 바로 받아오지 않고, state로 받아와서 state의 값이 바뀔 때 마다 render가 호출되도록 한다.