

# 5. React\_LifeCycle\_Functional

## <중요!> (March 8th, 21)

### B. Functional 컴포넌트의 LifeCycle

어떻게 하면 **FuncComp**의 **render**가 실행된 후에, 추가적으로 필요한 **후처리 작업**을 할 수 있을까?

▼ 이 때 사용하는 것이 react의 **"useEffect"**라고 하는 새로운 **hook** 이다.

▼ side effect라고도 부른다.

```
import React, {useState, useEffect} from 'react'; #App.js에 새로운 Hook을 import

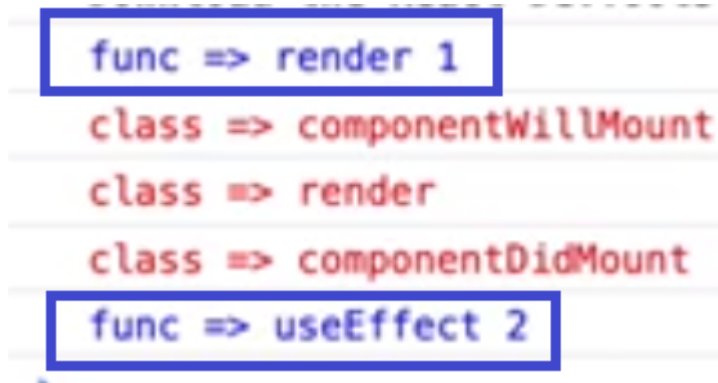
function FuncComp(props){
  var numberState = useState(props.initNumber);
  var number = numberState[0];
  var setNumber = numberState[1];
  // var dateState = useState((new Date()).toString());
  // var _date = dateState[0];
  // var setDate = dateState[1];
  var [_date, setDate] = useState((new Date()).toString());

  useEffect(function(){          # useEffect는 함수를 인자로 갖는다.
    console.log('%cfunc => useEffect'+(++funcId), funcStyle);
  });

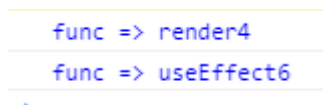
  console.log('%cfunc => render'+(++funcId), funcStyle);

  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={
        function(){
          setDate((new Date()).toString());
        }
      }></input>
    </div>
  );
}
```



**useEffect hook**을 사용하면, **functional** 컴포넌트의 **render**가 끝난 후에, **useEffect**가 인자로 갖는 함수를 호출한다.



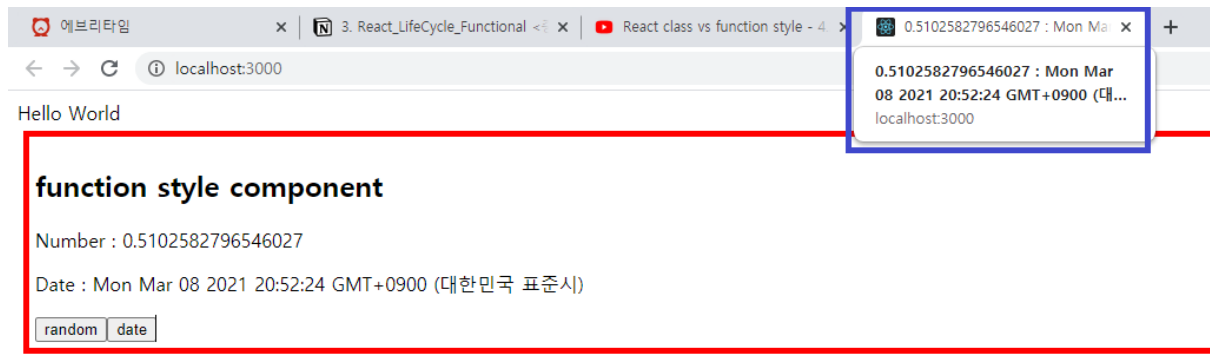
**useEffect hook**은 **functional** 컴포넌트의 **render**가 처음 끝난 후에, 그리고 **functional** 컴포넌트가 **render**될 때 마다 인자로 갖는 함수를 호출한다.

결론적으로, **functional** 컴포넌트에서 **useEffect**는 **class**형 컴포넌트에서의 **componentDidMount**와 **componentDidUpdate**와 같은 역할을 수행한다.

- **useEffect hook**을 사용하는 대표적인 예시 : 문서의 **title**값을 바꾸는 것

ex.

```
useEffect(function(){
  console.log('%cfunc => useEffect'+(++funcId), funcStyle);
  document.title = number + ' : ' + _date;
});
```



컴포넌트가 render될 때 마다, useEffect의 인자 함수 document.title이 실행되어 페이지의 제목이 바뀐다.

- useEffect는 하나의 render 뒤에 여러 개를 설치할 수 있다.

```
func => render 4  
func => useEffect (componentDidMount & componentDidUpdate) A 5  
func => useEffect (componentDidMount & componentDidUpdate) B 6
```