

## 쇼폴의 처음 만나 리액트



## 제 7장 후



## Q & A

- 언제라도 질문하세요  
모르면 외우면 되지만 코딩 안되면 꼭 질문
  1. 대면수업에서 질문
  2. 한성e-class 질의응답게시판
- 강사의 1번 선생님은 여러분의 질문





# 목차

1. 훅이란 무엇인가?
2. useState
3. useEffect
4. useMemo
5. useCallback
6. useRef
7. 훅의 규칙
8. 나만의 훅 만들기
9. 실습: 훅을 사용한 컴포넌트 개발
10. 마치며



## 7.1 훅 (Hook)이란 무엇인가?

- Hook은 갈고리라는 뜻이며 **function**임 (변수도 아니고 **class**자료형도 아님)
- 웹기초9장 이벤트 리스너와 유사하게 동작  
개발자가 **call**하는 것이 아니라 생명주기 조건에 따라 호출됨
- 훅의 이름은 모두 **use**로 시작
- React가 제공하는 훅이 있고 개발자가 만든 훅이 있음 ( Custom hook )
- Hook은 리액트 v16부터 생김. 2024년 현재 v18
- Hook 함수 종류는  
<https://17.reactjs.org/docs/hooks-reference.html>
- 교재는 7.2 useState 7.3 useEffect 7.4 useMemo 7.5 useCallback 7.6 useRef를 설명하지만 7.9 실습에서 다루는 useState와 useEffect만 강의하고 나머지 hook은 향후 실습에서 나오면 강의 예정



## 7.2 useState

- Hook이 없을때 재렌더링 되지 않음  
코드위치: 추가코드 215\_noHook.jsx
- 재렌더링 방법
  1. 4장 Clock.jsx : setInterval ( root.render() )
  2. 6장 class component NotificationList.jsx에서 this.setState()
- Function component에서는 어떻게 해야 하나? useState 혹은 사용  
syntax : const [ 변수명, set함수명 ] = useState(초기값)  
코드위치: 216\_useState.jsx



## 7.2 useState : 216\_useState.jsx

학생> 코드 어려워요. 한줄씩 설명해주세요

학생> 1번줄에서 { useState } 는 무엇인가요?

교수> 코드위치: CDN/chapter\_07\_vanillaJS/index\_export.html

```
export default message;  import message;  
export {name, age};      import {name, age};
```

설명: [https://www.w3schools.com/js/js\\_modules.asp](https://www.w3schools.com/js/js_modules.asp)

학생> 4번줄에서 왜 배열이 대입문 왼쪽에 있나요?

교수> destructuring assignment : 구글링 > MDN

학생> 4번줄은 무슨 뜻인가요?

교수> useState()설명 <https://17.reactjs.org/docs/hooks-reference.html#usestate>

다음 페이지에 자세히



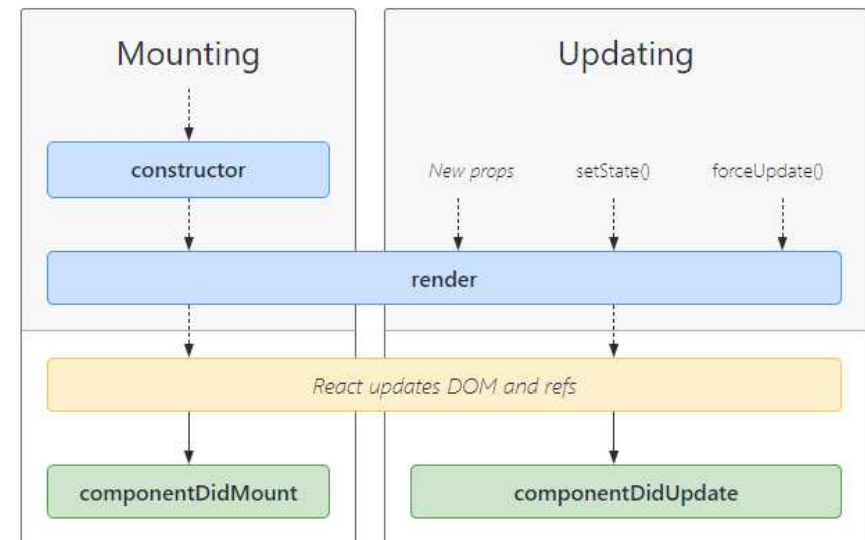
## 7.2 useState : 216\_useState.jsx

학생> 코드 어려워요. 한줄씩 설명해주세요.

- 4번줄 `const [count, setCount] = useState(0);`의 의미는?  
변수명, set함수명
- 6장 클래스컴포넌트 NotificationList.jsx 25줄 `this.state`처럼 `count`를 `state` 프로퍼티로 정한다고 `react`환경에 등록함 (이벤트 리스너 등록과 유사)
- 6장 클래스컴포넌트 NotificationList.jsx 36줄 `this.setState()` 함수처럼 `set`함수를 `setCount()`로 정한다고 `react`환경에 등록함
- `setCount()`로 `state`변수인 `count`가 변하면 NotificationList.jsx 54줄 `render()`가 `react`환경에서 호출되는 것처럼 `function component`인 `Counter(props)`가 호출되어 재렌더링됨  
(F12에서 console 출력 참고)

학생> 6장 class component의 생명주기와 비교하니 재렌더링(Updating)이 어떻게 되는지 이해가 됩니다.

```
>> Counter(count = 0) called
>> Counter(count = 0) called
>> Counter(count = 1) called
>> Counter(count = 1) called
```

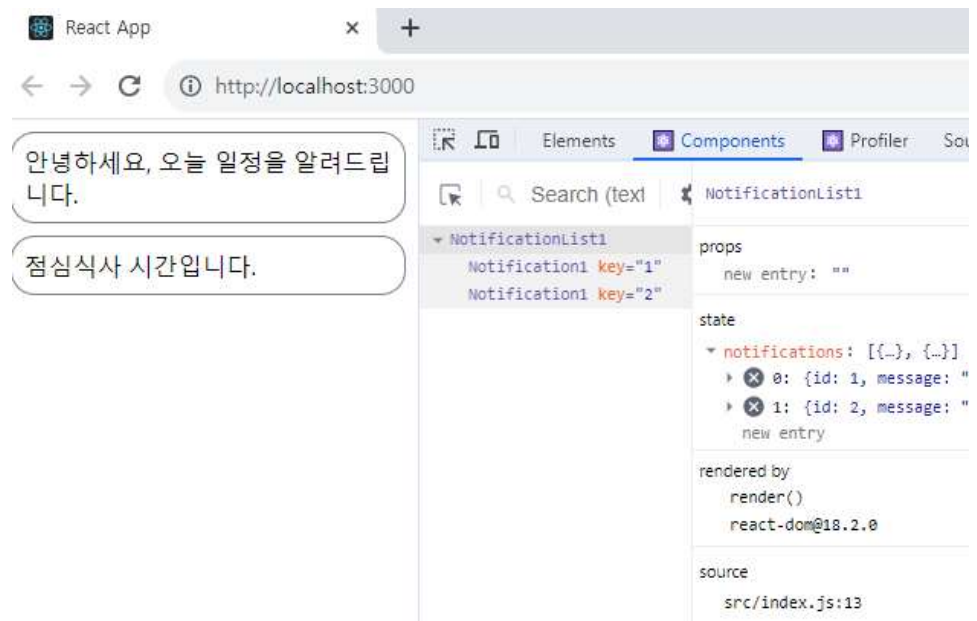
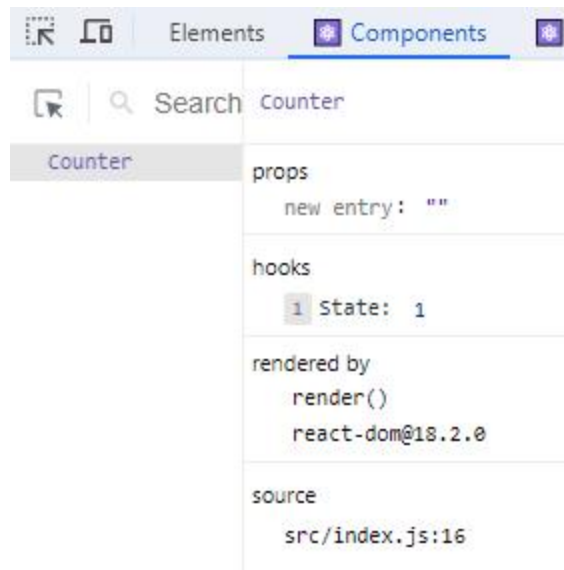




## 7.2 useState : 216\_useState.jsx

학생> 코드 어려워요. 한줄씩 설명해주세요.

학생> F12 리액트 Components창에서 class component는 state변수를 보여주고  
function component는 hook에서 state변수 보여주네요

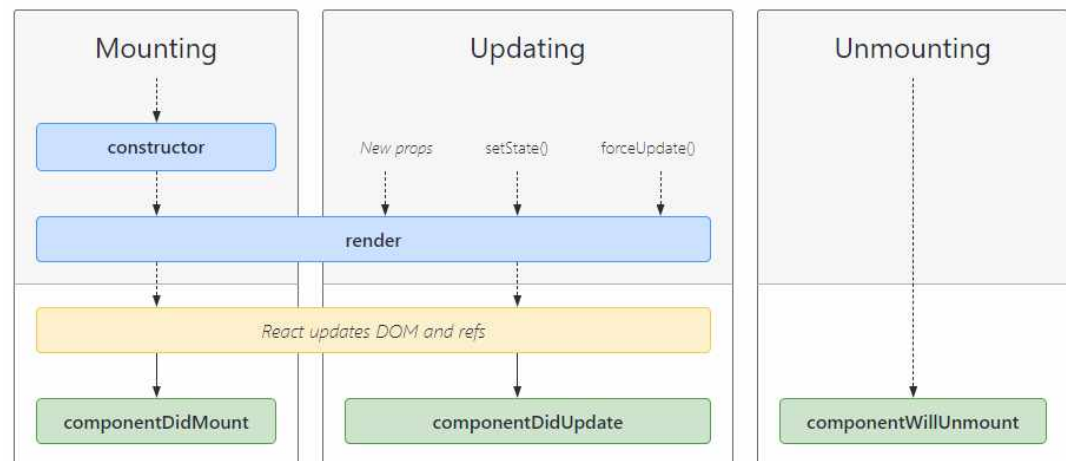






## 7.3 useEffect

- **useEffect** 함수는 클래스컴포넌트에서 제공하는 생명주기함수인 **componentDidMount()**, **componentDidUpdate()**, **componentWillUnmount()** 기능을 하나로 통합해서 함수컴포넌트에 제공함
- **Syntax** : **useEffect**(이펙트 함수, 의존성 배열);  
코드위치: 219\_useEffect.jsx
- 의존성배열이 있는 경우는 7.9 Accommodate.jsx에서 설명함
- 의존성배열이 없는 경우는 컴포넌트가 렌더링 또는 재렌더링 될때 리액트환경이 이펙트함수 호출
- 7장코드에서는 **useEffect**함수가 **componentDidMount()**, **componentDidUpdate()**기능을 보여주지만 6장의 **componentWillUnmount()** 기능을 보여주는 코드 예제가 없음





## 7.3 useEffect : 219\_useEffect.jsx

학생> 코드 어려워요. 한줄씩 설명해주세요.

학생> 1번줄에서 { useState, useEffect } 는 무엇인가요?

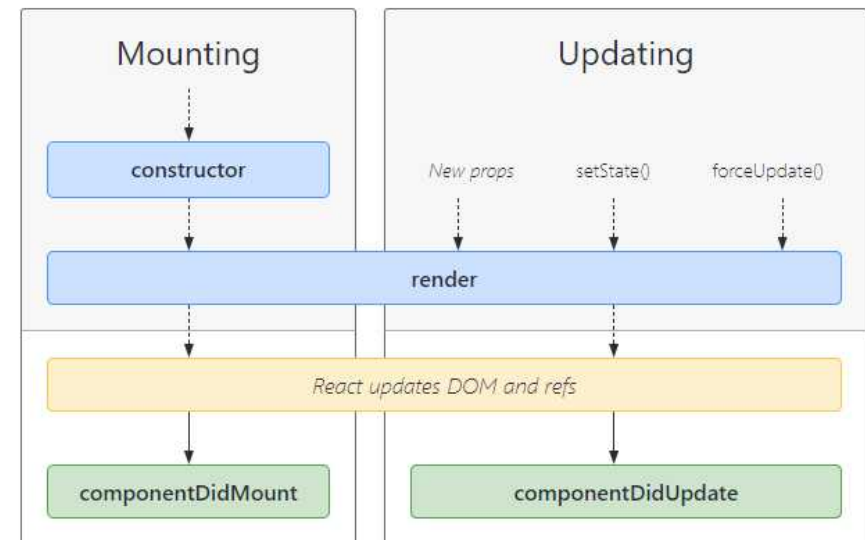
교수> 7.2 useState에서 설명함

학생> 7번줄은 무슨 뜻인가요?

교수> useEffect()설명 <https://17.reactjs.org/docs/hooks-reference.html#useeffect>

- 6장 클래스컴포넌트 Notification.jsx 25줄 componentDidMount(), 29번줄 componentDidUpdate() 기능 처럼 react환경에 등록함 (이벤트 리스너 등록과 유사)
- 의존성배열이 없는 경우는 컴포넌트가 렌더링 또는 재렌더링 될때 리액트환경이 useEffect함수 호출 (F12에서 console 출력 참고)

```
Console What's new Issu
>> Component(count = 0) called
>> Component(count = 0) called
>> useEffect called
>> useEffect called
>> Component(count = 1) called
>> Component(count = 1) called
>> useEffect called
```



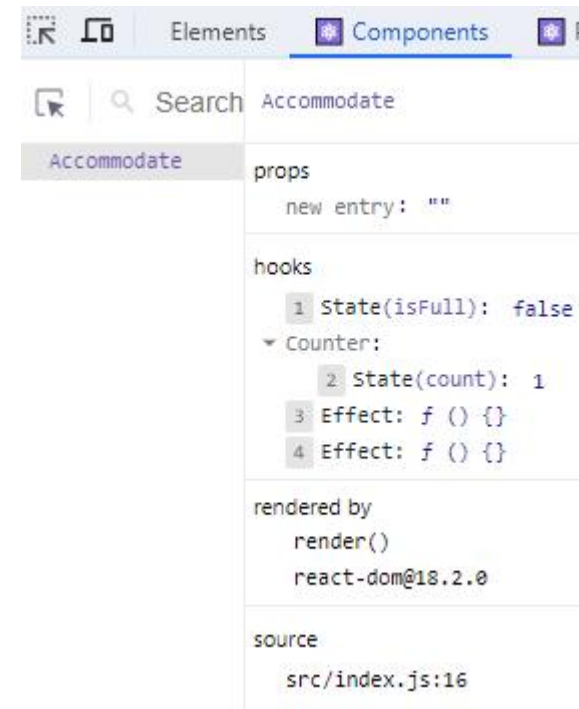
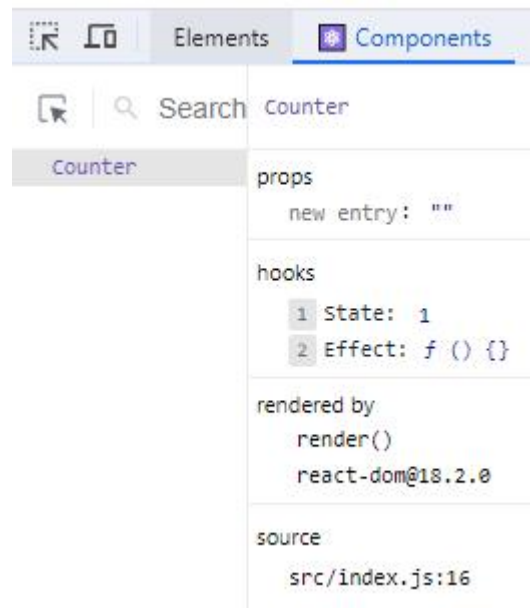


## 7.3 useEffect : 219\_useEffect.jsx

학생> 코드 어려워요. 한줄씩 설명해주세요.

학생> F12 리액트 Components창에서 class component는 state변수를 보여주고 function component는 hook에서 State변수와 Effect함수 보여주세요.

교수> 7.9 Accommodate.jsx에서는 더 많은 state변수와 Effect함수 보여줌  
( 이 코드에서는 Effect함수 이름은 정하지 않았음. State 변수 이름은 count로 정했음)





## 7.4 useMemo 7.5 useCallback 7.6 useRef

- 교재는 7.2 useState 7.3 useEffect 7.4 useMemo 7.5 useCallback 7.6 useRef를 설명하지만 7.9 실습에서 다루는 useState와 useEffect만 강의하고 나머지 hook은 향후 실습에서 나오면 강의 예정  
<https://17.reactjs.org/docs/hooks-reference.html>



## 7.7 훅의 규칙

1. 훅은 함수컴포넌트안의 최상위 레벨에서만 호출(등록)해야 함. 조건문,반복문,중첩된 함수 안에서 호출하면 안됨
2. 훅은 리액트 함수 컴포넌트에서와 **Custom Hook**에서만 훅을 호출(등록)해야 함. 일반적인 JS에서 훅을 호출(등록)하면 안됨

```
01 function MyComponent(props) {  
02     const [name, setName] = useState('Inje');  
03  
04     if (name !== '') {  
05         useEffect(() => {  
06             ...  
07         });  
08     }  
09  
10     ...  
11 }
```



## 7.8 나만의 훅 만들기

- Custom Hook : 리엑트는 기본으로 제공하는 훅 ( <https://17.reactjs.org/docs/hooks-reference.html> )들 이 외에 추가적으로 필요한 기능이 있다면 내가 직접 만들어서 사용할 수 있음
- Custom Hook 사례: 코드위치 : `useCounter.jsx` 훅이름 `useCounter`
- Custom Hook의 이름은 `use`로 시작
- Custom Hook 내부에서 호출(등록)하는 Hook은 리엑트제공 Hook 또는 Custom Hook
- 코드 한줄 한줄 설명은 7.9 실습에서
- 3<sup>rd</sup> party에서도 Custom Hook을 만들어서 사용할 수 있음  
16장 미니블로그 `PostViewPage.jsx`  
`import { useNavigate, useParams } from 'react-router-dom';`



# 목차

1. 훅이란 무엇인가?
2. useState
3. useEffect
4. useMemo (필요시)
5. useCallback (필요시)
6. useRef (필요시)
7. 훅의 규칙
8. 나만의 훅 만들기
9. 실습: 훅을 사용한 컴포넌트 개발
10. 마치며



## 7.9.0(실습) CDN(reactJS)

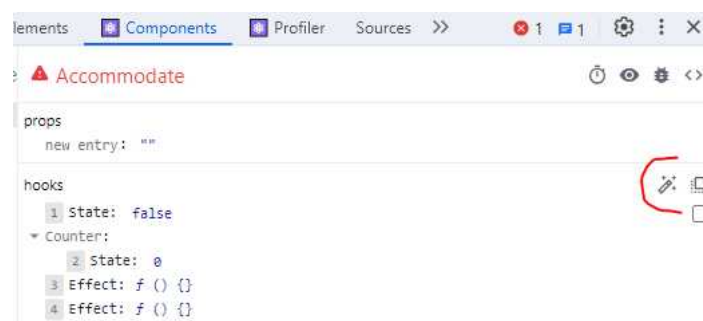
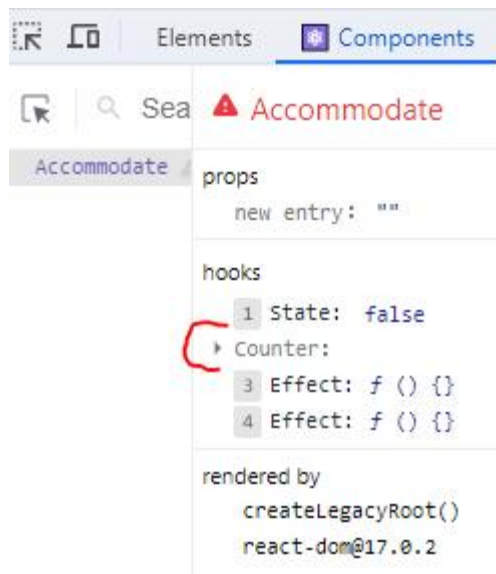
- 7장 Accomodate.jsx 를 babel 에서 변환하여 js코드 생성하여 수행해봄
- Npm start와 LiveServer 수행 동일
- 코드위치 : 추가 코드 CDN/ chapter\_07\_reactJS 에 업로드

학생> F12 컴포넌트창에서 State가 2개가 아니라 1개만 보이는 데요.

교수> Counter 클릭

학생> state변수 name이 안보이는데요

교수> parse버튼 클릭







## 7.9.0(실습) CDN(실습) (local\_reactJS)

- 코드위치 : 7장 코드 CDN/ chapter\_07\_local\_reactJS 에 업로드

학생> reactJS와 local\_reactJS와의 차이점은 무엇인가요?

교수> CDN에서 react.js와 react-dom.js를 다운로드하여 local에서 import하여 사용

학생> 왜 local에서 import하여 사용하나요?

교수> react.js와 react-dom.js의 내용을 보고자 함

학생> react.js는 3K줄 react-dom.js는 29K줄이나 되네요.

코드를 vscode의 outline기능으로 보아도 코드 이해는 어렵네요.

교수> 오픈소스인 리눅스는 1,000K줄이 넘습니다. 이제까지 react 라이브러리는 Blackbox이었는데

코드를 볼 수 있는 Whitebox로 변환되었습니다.

하지만 코드 이해는 어렵기에 리액트 doc으로 이해합니다.

교수> index.js에서 typeof 자료형출력 연산자를 통해 React와 ReactDOM의 자료형을 보았습니다.



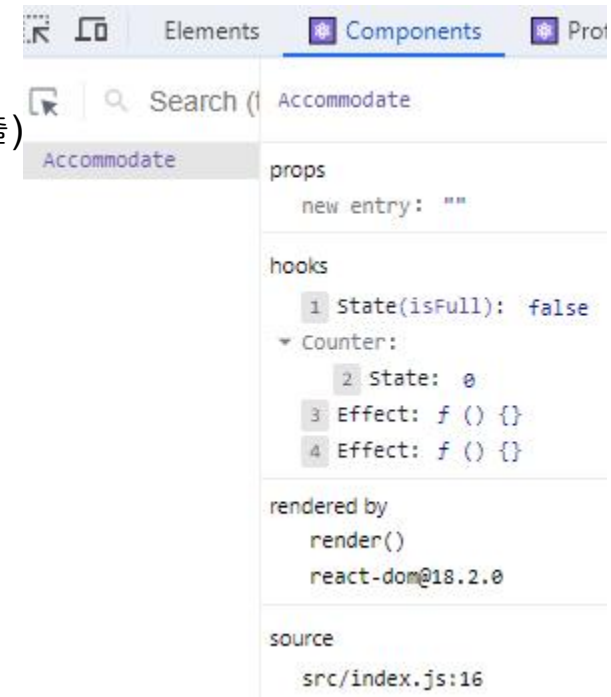
## 7.9.1 (실습) useCounter() 커스텀 훅 만들기

- 코드위치 : useCounter.jsx
- useCounter() 훅을 만들고 내부에서 useState() 훅을 사용함
- useCounter() 훅은 Accommodate.jsx의 8번줄에서 사용함  
syntax : const [ 변수명, set함수명 ] = useState(초기값)  
const [isFull, setIsFull] = useState(false); // 리액트 훅  
const [count, increaseCount, decreaseCount] = useCounter(0); // Custom Hook
- useCounter 훅의 state 변수명은 count, set함수명은 increaseCount와 decreaseCount



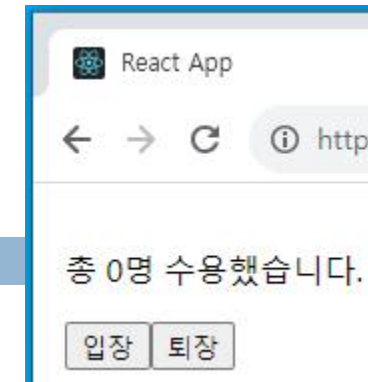
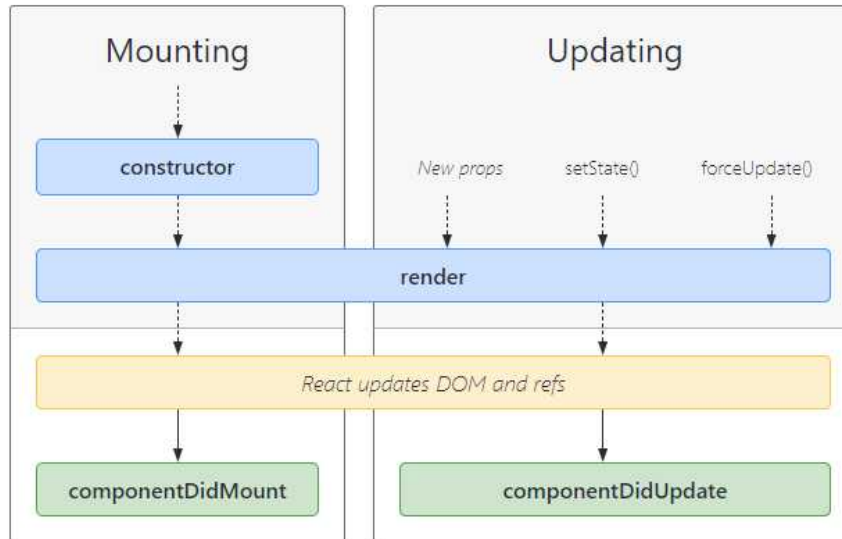
## 7.9.2(실습) Accommodate 컴포넌트 만들기

- Accommodate.jsx 를 Accommodate1.jsx 로 변경. 생명주기를 쉽게 보기 위해. 즉 rendering과 재렌더링
- 변경점은 Vscode에서 2개 파일 비교하기 기능사용하여 비교 (6장에서 배움)
- 코드위치 : 추가코드에 업로드
- 정원초과를 빨리 보기위해 4번줄 변경
- 실행하여 F12 리액트창  
state변수 2개 useEffect혹 2개  
state변수 2개 parse버튼으로 name 표시함
- Syntax : useEffect(이펙트 함수, 의존성 배열);  
1번 이펙트함수는 의존성 배열 없음 ( 컴포넌트가 업데이트될 때마다 호출 )  
2번 이펙트함수는 의존성 배열 있음 ( 배열값이 변경되었을 때 호출 )





## 7.9.3(실습) 초기상태 (Mounting 단계)



- ① >> Accomodate(count = 0, isFull = false) called  
>> Accomodate(count = 0, isFull = false) called
- ② =====  
useEffect() is called.  
isFull: false
- ③ Current count value: 0

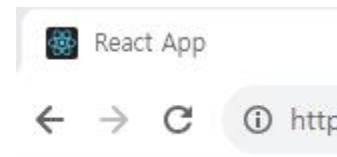
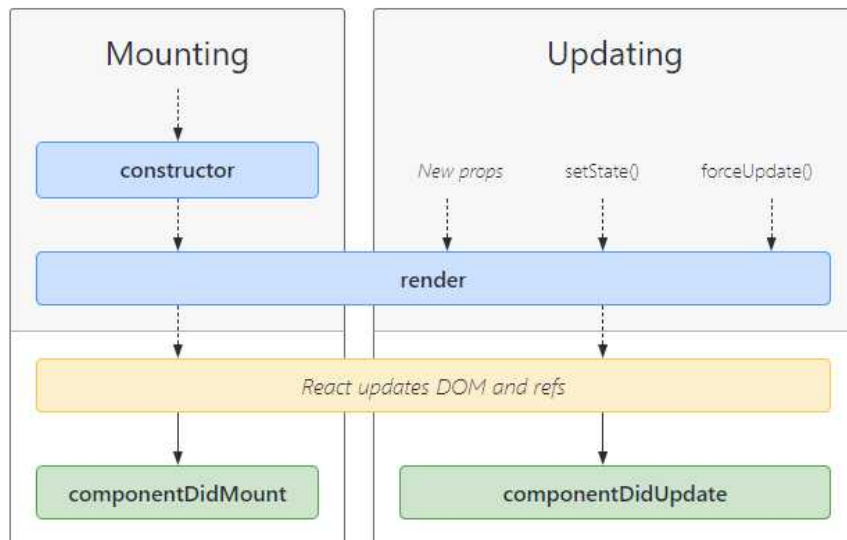
- 1번줄: index.js의 <Accomodate1>으로 호출하여 rendering됨
- 2번줄: Mounting과정에서 componentDidMount()에 해당하는 1번 이펙트함수를 React환경이 호출
- 3번줄: Mounting과정에서 componentDidMount()에 해당하는 2번 이펙트함수를 React환경이 호출

학생> 20번줄이 수행되고 10번줄 이펙트함수가 실행되네요?

교수> 생명주기에서 Accomodate1이 rendering되고 10번줄 이펙트함수 (함수이름은 없음)가 React환경이 호출함  
() => {  
 console.log("=====");  
 console.log("useEffect() is called.");  
 console.log(`isFull: \${isFull}`);  
}



## 7.9.3(실습) Updating 단계



총 1명 수용했습니다.

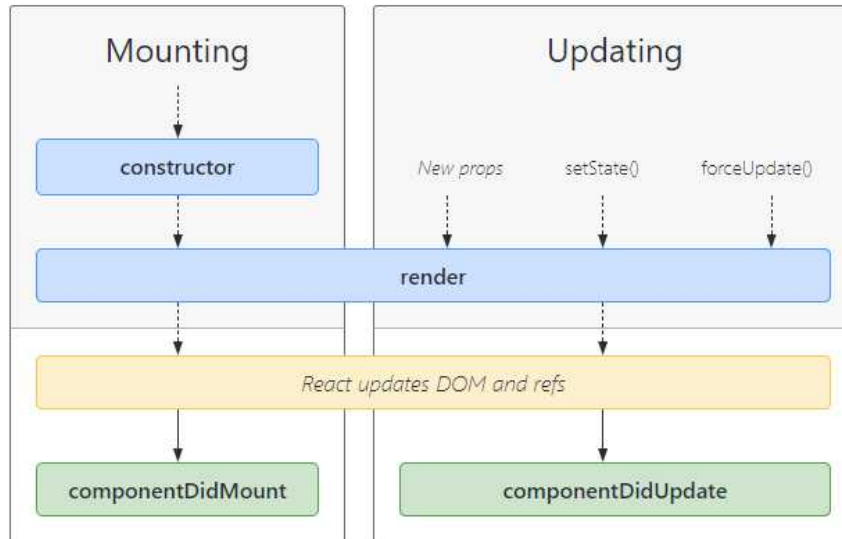
입장 퇴장

```
① >> Accomodate1(count = 1, isFull = false) called
  >> Accomodate1(count = 1, isFull = false) called
② =====
  useEffect() is called.
  isFull: false
③ Current count value: 1
④ >> Accomodate1(count = 1, isFull = false) called
  >> Accomodate1(count = 1, isFull = false) called
```

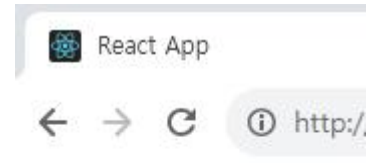
- 1번줄 Accomodate1의 26번줄에서 set함수인 increaseCount로 state변수인 count가 변했기에 Updating단계에서 function component인 Accomodate1호출
- 2번줄 Update과정에서 DidUpdate()에 해당하는 1번 이펙트함수를 React환경이 호출
- 3번줄 Update과정에서 DidUpdate()에 해당하는 2번 이펙트함수를 React환경이 호출
- 4번줄 Update과정에서 Accomodate1이 왜 호출되었는지는 모르겠음



## 7.9.3(실습) Updating 단계



- 1번줄 Accomodate1의 26번줄에서 set함수인 increaseCount로 state변수인 count가 변했기에 Updating단계에서 function component인 Accomodate1호출
- 2번줄 Update과정에서 DidUpdate()에 해당하는 1번 이펙트함수를 React 환경이 호출
- 3번줄 Update과정에서 DidUpdate()에 해당하는 2번 이펙트함수를 React 환경이 호출하고 set함수인 setIsFull로 state변수 isFull이 true로 변경됨
- 4번줄 state변수 isFull이 변경되었기에 Updating단계에서 function component인 Accomodate1호출
- 5번줄 Update단계에서 1번 이펙트함수는 호출되었으나 2번 이펙트함수는 19번줄 의존성배열 count가 변하지 않아 호출되지 않음  
Syntax : useEffect(이펙트 함수, 의존성 배열);



총 2명 수용했습니다.

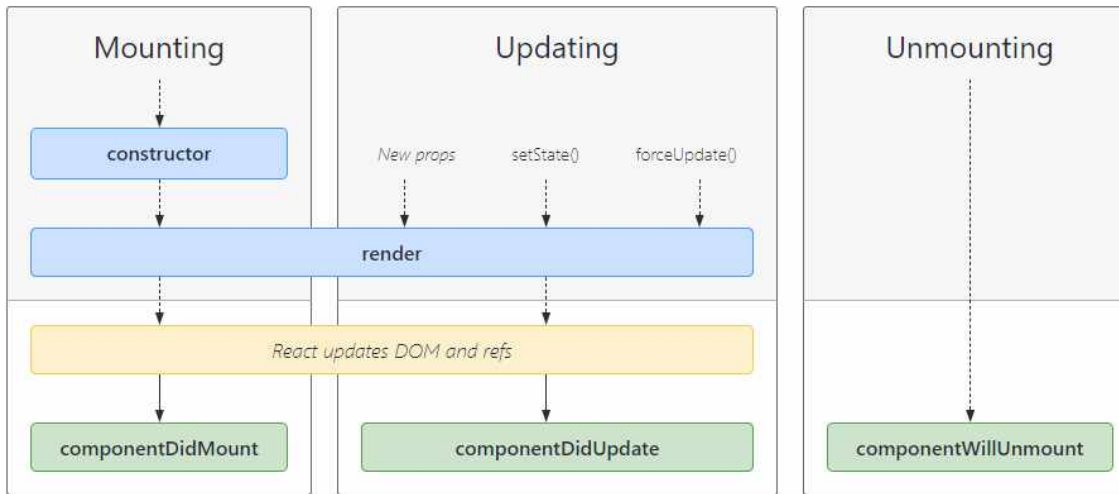
입장 퇴장

정원이 가득차 있습니다.

- 1 >> Accomodate1(count = 2, isFull = false) called  
>> Accomodate1(count = 2, isFull = false) called
- 2 =====  
useEffect() is called.  
isFull: false
- 3 Current count value: 2
- 4 >> Accomodate1(count = 2, isFull = true) called  
>> Accomodate1(count = 2, isFull = true) called
- 5 =====  
useEffect() is called.  
isFull: true

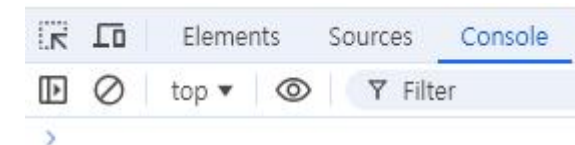


## 7.9.3(실습) Updating 단계



총 0명 수용했습니다.

입장 퇴장



- 0명에서 퇴장버튼을 클릭하면 useCounter.jsx 7번줄에서 state변수 count가 변하지 않음  
`const decreaseCount = () => setCount((count) => Math.max(count - 1, 0));`
- State변수인 count와 isFull이 변하지 않으므로 Updating단계 없음
- 7장 Accommodate1.jsx에서는 6장 NotificationList.jsx 처럼 빈화면이 되는 경우가 없어서 Unmounting 단계 없음



# 이제 나는 할 수 있다

1. 나는 `useState` 훅과 `useEffect` 훅을 사용할 수 있다
  2. 나는 나만의 커스텀훅을 만들 수 있다 [ `useCounter()` ]
  3. 나는 F12 리액트tool에서 `state`변수와 `useEffect` 훅 내용을 볼 수 있다
  4. 나는 교재 코드 `Accommodate.jsx`를 변형하여 6장의 생명주기 함수와 유사하게 `useEffect`훅이 언제 `call`되고 언제 내가 `call` (`useState`의 `set`함수) 할 수 있는지 알았다
  5. 나는 훅을 사용하여 나의 `function component`를 재렌더링 하는 방법을 알았다.
- 유튜브 강의 동영상 : <https://www.youtube.com/c/소플TV>
  - Q&A 사이트 : <https://www.frontoverflow.com>

