

08 / 09 월요일 이론 강의



왜 컴포넌트 내에서 변수를 안 쓰고 state를 사용해야할까?

예시

```
function Test() {  
  let clickCount = 0;  
  
  function handleClick() {  
    clickCount += 1;  
  }  
  return (  
    <div>  
      <input  
        type="button  
        value="버튼"  
        onClick={handleClick}  
      / >  
      <p>{clickCount}</p>  
    </div>  
  )  
}
```

렌더링 시 해당 컴포넌트가 계속 실행 됨

<Test />

==

Test()

함수가 실행될 때마다 내부 변수가 초기화 됨

**즉 컴포넌트가 렌더링 될 때 마다
변수가 초기화 돼 사용 불가**

하지만 State는 렌더링이 돼도 값을 유지

렌더링이 되는 경우

- 부모 컴포넌트가 렌더링 됐을 때
- props에 변경됐을 때
- state가 변경됐을 때

LifeCycle이란?

- 컴포넌트의 생명주기
- 컴포넌트가 생성되거나 제거되거나 업데이트 될 때 발생하는 이벤트

마운트란?

- 컴포넌트가 생성된 시점에 실행되는 이벤트

언마운트란?

- 컴포넌트가 제거된 시점에 실행되는 이벤트

업데이트란?

- 컴포넌트 내부의 변수가 변경됐을 때 실행되는 이벤트

예시

```
function Test() {  
  useEffect(() => {  
    console.log("컴포넌트 마운트");  
  }, []);  
  
  return (  
    <div>마운트 테스트</div>  
  )  
}
```

useEffect

함수로 마운트 구현 가능

마운트란?

- 컴포넌트가 생성된 시점에 실행되는 이벤트

마운트 구현 조건

- 첫번째 인자에 실행될 함수
- 두번째 인자에 빈 배열

예시

```
function Test() {  
  useEffect(() => {  
    console.log("컴포넌트 마운트");  
  
    return () => {  
      console.log("컴포넌트 언마운트");  
    }  
  }, []);  
  
  return (  
    <div>마운트 테스트</div>  
  )  
}
```

useEffect

함수로 언마운트 구현 가능

언마운트란?

- 컴포넌트가 제거된 시점에 실행되는 이벤트

언마운트 구현 조건

- 마운트를 구현한 함수에서
return에 실행될 함수 정의

예시

```
function Test() {
  const [text, setText] = useState('');
  const [newText, setNewText] = useState('');

  useEffect(() => {
    setNewText(text + "입니다.");
  }, [text]); // text가 변할 때마다 실행

  return (
    <div>
      <p>{newText}</p>
      <input
        value={text}
        onChange={
          (e) => setText(e.target.text)
        }
      />
    </div>
  )
}
```

useEffect

함수로 업데이트 구현 가능

업데이트란?

- 컴포넌트 내부의 변수가 변경됐을 때 실행되는 이벤트

업데이트 구현 조건

- 첫번째 인자에 실행될 함수
- 두번째 인자에 업데이트를 감지할 변수 배열

axios란?

- React에서 많이 쓰이는 HTTP Request 라이브러리
 - ajax 통신과 같은 역할 수행
 - 간편한 문법

설치방법

```
> yarn add axios  
# Or  
> npm install axios
```


예시

```
// url에 해당하는 서버로 get method 요청
axios.get('request url')
  .then(
    (response) => {
      console.log("받은 데이터", response.data);
    });
```

예시

```
// url에 해당하는 서버로 post method 요청
axios.post('request url',{/* post 요청에 보낼 객체*/})
  .then(
    (response) => {
      console.log("받은 데이터", response.data);
    });
```

