

웹프로그래밍응용 리액트

Stateful & Stateless

2020904053 윤금비

깃허브 주소 : <https://github.com/geumbi4>



2024/5/21

목차

01

개념

02

실습

03

참고자료



x



x



x



1. Stateful , Stateless란?

- 클래스형 컴포넌트는 로직과 상태를 컴포넌트 내에서 구현하기 때문에 **stateful**이라고 불리며 **상대적으로 복잡한 UI 로직**을 가진다.
- 반면 **함수형 컴포넌트**는 **state를 사용하지 않고 단순히 props를 통해** 데이터를 받아 UI에 뿌려주기 때문에 **stateless**라고 불린다.
- 함수형 컴포넌트에서 Hook을 필요한 곳에 사용, 로직의 재사용이라는 장점이 있어 많은 개발자들이 클래스형 컴포넌트보다 함수형 컴포넌트 + Hook을 사용한다.
- 과거 클래스 컴포넌트 사용한 프로젝트가 있기에 유지보수를 위해서 알아둘 필요가 있다.
- Hook : 클래스형 컴포넌트의 유용한 기능을 함수형 컴포넌트에서 사용할 수 있도록 해주는 기술



1. Stateful , Stateless란?

구분	함수형 컴포넌트	클래스형 컴포넌트
상태(state)	상태가 없는 경우에 적합, 간략한 화면 구성으로 성능 향상 가능	상태가 있는 경우에 적합, state의 생명주기 API 사용 가능
속성(props)	부모 컨테이너에서 전달받은 값 사용 가능	부모 컨테이너에서 전달받은 값 사용 가능 , 추가로 default 값이나 propTypes 등 지정 가능

- API(Application Programming Interface) : 두 소프트웨어 구성 요소가 서로 통신할 수 있게 하는 것
- 컴포넌트 목적에 따라 프레젠테이션 컴포넌트와 컨테이너 컴포넌트로 나누었을 때
기능 부분을 Container, UI 부분을 Presenter로 구분한다. 그러면 기능이 작동을 안한다면 컨테이너 부분을 보면 되고,
UI가 이상하다 싶으면 프레젠테이션 부분을 보면 된다.
- PropTypes : 부모로부터 전달받은 prop의 데이터 type을 검사한다. 자식 컴포넌트에서 명시해 놓은 데이터 타입과 부모로부터 넘겨받은
데이터 타입이 일치하지 않으면 콘솔에 에러 경고문이 띄워진다.



2. 라이프 사이클(컴포넌트 생명주기)이란? (컴포넌트 개념 중 나오는 단어)

- **라이프사이클** : 각 컴포넌트에는 수명이 존재하는데, 이 수명은 준비 과정부터 시작해 페이지에서 사라질 때 끝이 난다.
- 라이프 사이클이 있는 이유는 메모리 비우기가 가장 큰 이유이며, stateless 방식의 lifecycle은 권장되지 않는다.
- 너무 복잡하고, 함수로 비동기처리를 구현하고 상태관리하는 것 자체가 어렵기 때문이다.
- 그래서 **상태를 관리하는 Redux (라이브러리)**가 생기게 되었다.



3. 비동기처리란?

- 특정 로직의 실행이 끝날 때까지 기다려주지 않고 나머지 코드를 먼저 실행하는것

<예제코드>

```
//1
console.log('Hello');
//2
setTimeout(function() {
  console.log('Bye');}, 3000);
//3
console.log('Hello Again');
```

‘Hello’ 출력
3초 있다가 ‘Bye’ 출력
‘Hello Again’ 출력이 될 줄 알았지만

<실행결과>

Hello	App.js:4
Hello Again	App.js:7
2 Bye	App.js:6
>	

setTimeout()가 비동기방식으로 처리됨
비동기처리방식의 문제점은 콜백함수로 해결하면 됨



4. pure component(순수 컴포넌트)란? (함수형 컴포넌트 개념 중 나오는 단어)

- 순수함수 : 동일한 입력에 대해 항상 같은 결과를 반환하며, 외부에서 전달받은 값을 변경하면 안 된다.
또한, 예측 가능하고 테스트하기 쉽다.
- 함수형 컴포넌트를 순수함수로 구성하는 이유 :
 - 동일한 props와 state를 입력으로 받으면 항상 동일한 엘리먼트를 반환한다.
 - 그래서 상태 변화를 추적하기 쉽고, 코드의 복잡성을 줄일 수 있다.

<순수함수 예시>

```
// 두 숫자를 더하는 함수
function add(a, b) {
  return a + b;
}
```

<컴포넌트>

Props

리액트 컴포넌트

리액트 엘리먼트

▶ 리액트 컴포넌트



5. state 설명 (stateful 컴포넌트)

- state란? : 컴포넌트 내부에서 바뀔 수 있는 값을 의미한다.
- let이 아니라 const로 선언하는 이유는? : 변수의 재할당을 막기 위해서
- setState는 왜 비동기적으로 동작할까? : state는 값이 변경되면 리렌더링이 발생하는데, 리렌더링이 많을수록 속도도 저하 등 성능적으로 문제가 많아지게 된다.
그래서 16ms 동안 변경된 상태 값들을 모아서 한 번에 리렌더링을 진행하고, 이를 batch(일괄) update라고 한다.
- state 특징 :
 - 생성자 (constructor)에서 초기화
 - 값을 변경하려면 setState 이용
 - 자동으로 뷰가 갱신됨
 - 상태를 변경하기 위해서는 클래스 컴포넌트이어야 한다.



6. state가 없는 (stateless 컴포넌트)

- stateless 컴포넌트 특징 :
 - 자기자신이 데이터를 갖고 있지 않음
 - 외부에서 데이터(props)를 받아서 처리
 - 입력값 = 출력값



7. stateful 과 stateless 정리

- **클래스 컴포넌트**에서 state를 사용하지 않아도 되지만, state를 써야한다면 반드시 클래스 컴포넌트로 선언해야 함
- 그래서 클래스 컴포넌트는 stateful 컴포넌트 라고도 부름
- 반면 state를 사용하지 않는 **함수형 컴포넌트**를 stateless 컴포넌트 라고도 부름
- 함수형 컴포넌트는 state사용x, 리랜더링 및 state라이프 사이클 관리를 하지 않아서 stateful 컴포넌트보다 성능이 기대가 됨.



7. stateful 컴포넌트와 stateless 컴포넌트 실습

<'현재 시간 나타내기' 실행화면 예시>

Hello, world!

It is 오전 5:33:12.



참고자료

참고자료1

참고자료2

참고자료3

참고자료4

참고자료5

참고자료6

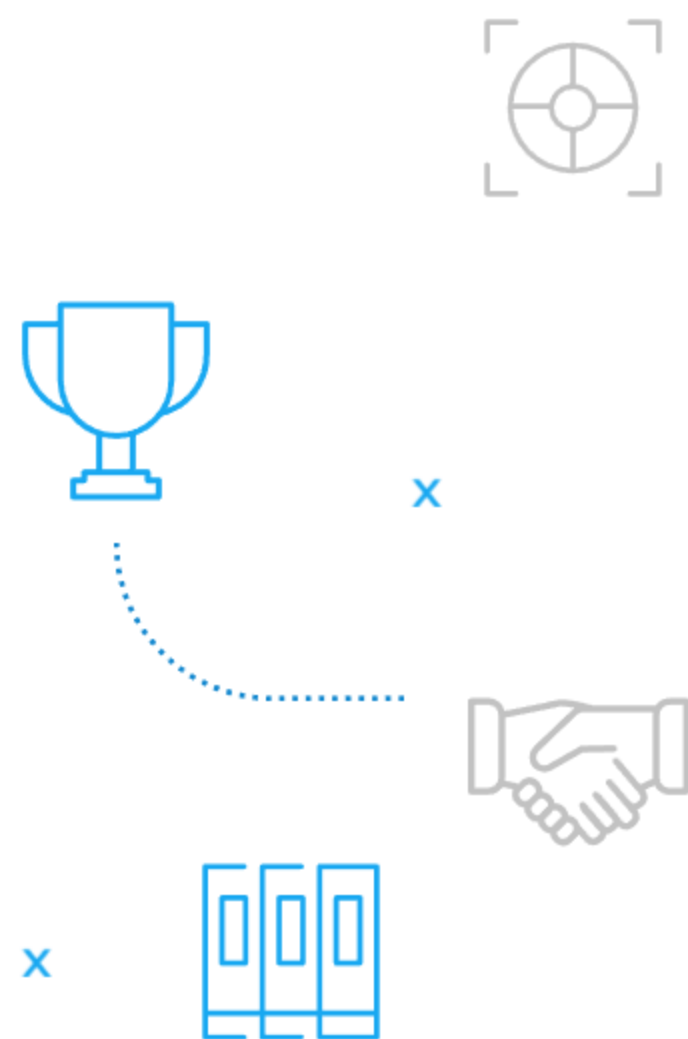
참고자료7

참고자료8

참고자료9

참고자료10

참고자료11



이상입니다.

x

