

<Week1 Report>

이태훈

주제 : Class Component VS Functional Component

1. Component란?

컴포넌트는 특정 부분이 어떻게 생길지 정하는 선언체이다. 재사용이 가능한 API로 수많은 기능들을 내장하고 있다.

2. Class Component

클래스형 컴포넌트의 경우 배열 state 기능 및 라이프 사이클 기능을 사용할 수 있고 임의 메서드를 정의할 수 있다. 클래스형 컴포넌트에는 render함수가 꼭 있어야 한다. render 함수 안에는 JSX를 반환하는 내용이 있어야 한다.

```
import { Component } from 'react';

class Counter extends Component {
  state = {
    number: 0,
    fixedNumber: 0
  };
  render() {
    const { number, fixedNumber } = this.state; // state를 조회할 때는 this.state로 조회합니다.
    return (...);
  }
}

export default Counter;
```

3. Functional Component

함수형 컴포넌트의 경우 클래스형 컴포넌트보다 선언하기가 편리하다. 메모리 자원도 덜 사용하고, 배포 시에도 클래스형 컴포넌트보다 결과물의 파일 크기가 더 작다. 하지만 함수 컴포넌트의 단점이 있다. 바로 State와 라이프사이클 API의 사용이 불가능한 것이다. 하지만 리액트 v16.8 이후 hooks라는 기능이 도입되면서 이 단점들은 해결되었다. 리액트 공식 매뉴얼에서도 함수 컴포넌트와 Hooks를 사용하도록 권장되고 있다. 또한 리액트 개발 팀에서 함수 컴포넌트와 Hooks를 사용하는 것이 주요 컴포넌트 개발 방식이 될 것이라고 공지하였다. State 사용법은 배열 비구조화 할당을 통해서 한다. 클래스 컴포넌트는 setState를 사용해서 상태 변경을 하지만 함수 컴포넌트는 useState를 통해 상태를 변경한다. 함수 컴포넌트는 useState를 한 컴포넌트에서 여러 번 쓸 수 있다.

```
import { useState } from 'react';

const Say = () => {
  const [message, setMessage] = useState('');
  const onClickEnter = () => setMessage('안녕하세요!');
  const onClickLeave = () => setMessage('안녕히 가세요!');

  const [color, setColor] = useState('black');

  return (
    <div>
      <button onClick={onClickEnter}>입장</button>
      <button onClick={onClickLeave}>퇴장</button>
      <h1 style={{ color }}>{message}</h1>
      <button style={{ color: 'red' }} onClick={() => setColor('red')}>
        빨간색
      </button>
      <button style={{ color: 'green' }} onClick={() => setColor('green')}>
        초록색
      </button>
      <button style={{ color: 'blue' }} onClick={() => setColor('blue')}>
        파란색
      </button>
    </div>
  );
};

export default Say;
```