# css 상속 문제

CSS modules 사용 시 클래스가 사용 component 내에 격리되기 때문에 다른 component에선 참조할 수 없게 되어 있습니다.

따라서 아래 코드로는 기존처럼 생각되는 결과가 나오지 않습니다.

**ParentComponent**

import ChildComponent from './ChildComponent';  
import style from 'ParentComponent.scss';  
// ...  
  
class ParentComponent extends React.Component {  
 render() {  
 return (  
 <div className={style.parent}>  
 <ChildComponent />  
 </div>  
 );  
 }  
}

**ChildComponent**

import style from 'ChildComponent.scss';  
// ...  
  
class ChildComponent extends React.Component {  
 render() {  
 return (  
 <div className={style.child}>  
 ...  
 </div>  
 );  
 }  
}

/\* ChildComponent.scss \*/  
child {  
 // 컴포넌트의 기본 스타일  
}  
  
parent {  
 child {   
 // 부모 클래스를 상속 받아 커스텀하고자 하는 스타일  
 }  
}

기존 스타일링 방식에 따르면 .child 와 .parent .child 양쪽이 다 적용 되었습니다.

하지만 css modules가 적용될 경우 .parent 클래스에 hash 코드가 붙어 변환되어 childComponent에선 알수가 없기 때문에 .child 스타일만 적용되게 됩니다.

이를 해결하기 위한 방법을 살펴보도록 하겠습니다.

## classnames

먼저 className을 조건에 따라 사용하기 쉽도록 “classnames” 모듈을 설치하도록 하겠습니다.

$ npm install --save classnames

“classnames”를 import 하여 사용할 수 있습니다.

import classnames from 'classnames';  
  
// ...  
  
const DEMO\_PROPS = {  
 active: true  
}  
  
const testClassName = classnames(style.test1, {  
 [style.is\_active]: DEMO\_PROPS.active  
});  
  
// "test1 is\_active"  
<div className={ testClassName }></div>

## TodoItem 컴포넌트 분리

이어서 TodoItem 컴포넌트를 만들어 보도록 하겠습니다.

TodoList 컴포넌트의 li 부분을 가져다가 TodoItem 컴포넌트를 생성합니다.

**TodoItem.jsx**

import React from 'react';  
import style from './TodoItem.scss';  
  
const DEMO\_PROPS = {  
 text: 'React 가이드 업데이트하기'  
}  
  
class TodoItem extends React.Component {  
 render() {  
 const { text } = DEMO\_PROPS;  
  
 return (  
 <li className={style.item\_todo}>  
 <span className={style.text}>{text}</span>  
 <button type="button" className={style.button\_delete} aria-label="삭제"><i className={style.icon\_delete} /></button>  
 </li>  
 )  
 }  
}  
  
export default TodoItem;

TodoList의 li 부분을 TodoItem으로 변경합니다.

import React from 'react';  
import TodoItem from './TodoItem' // TodoItem 컴포넌트 호출  
import style from './TodoList.scss';  
  
const DEMO\_PROPS = {  
 itemList: [  
 { id: 1 },  
 { id: 2 },  
 { id: 3 }  
 ]  
}  
class TodoList extends React.Component {  
 render() {  
 const { itemList } = DEMO\_PROPS;  
  
 return (  
 <ul className={style.list\_todo}>  
 {itemList.map((item) => (  
 {/\* TodoItem 컴포넌트 삽입 \*/}  
 <TodoItem key={item.id} />  
 ))}  
 </ul>  
 )  
 }  
}  
  
export default TodoList;

이어서 css 상속 관련 실습을 하겠습니다.

## props 전달을 통한 타입별 클래스 적용

첫번째 방법으론 props로 타입 구별 값을 전달하여 적용하는 방법입니다.

**TodoList.jsx**

import React from 'react';  
import TodoItem from './TodoItem'  
import style from './TodoList.scss';  
  
const DEMO\_PROPS = {  
 itemList: [  
 { id: 1 },  
 { id: 2 },  
 { id: 3 }  
 ]  
}  
class TodoList extends React.Component {  
 render() {  
 const { itemList } = DEMO\_PROPS;  
  
 return (  
 <ul className={style.list\_todo}>  
 {itemList.map((item) => (  
 <TodoItem key={item.id} type="a" />  
 ))}  
 </ul>  
 )  
 }  
}  
  
export default TodoList;

**TodoItem.jsx**

import React from 'react';  
import classnames from 'classnames'; // classnames 호출  
import style from './TodoItem.scss';  
  
const DEMO\_PROPS = {  
 text: 'React 가이드 업데이트하기'  
}  
  
class TodoItem extends React.Component {  
 render() {  
 const { text } = DEMO\_PROPS;  
 const { type } = this.props; // 전달받은 props 값 가져오기  
 const childClassName = classnames(style.item\_todo, style['type\_'+type]); // type용 클래스 추가  
  
 return (  
 <li className={childClassName}>  
 <span className={style.text}>{text}</span>  
 <button type="button" className={style.button\_delete} aria-label="삭제"><i className={style.icon\_delete} /></button>  
 </li>  
 )  
 }  
}  
  
export default TodoItem;

위의 코드와 같이 부모 component에서 자식 component로 타입 구분 값을 전달하여 자식 component에서 해당 타입에 따라 특정 클래스를 사용하는 형태입니다.

props 값을 직접 사용해야 하는 부분이 있지만 component 기반 스타일링 규칙을 벗어나지 않고 스타일이 가능하기 때문에 추천하는 방법입니다.

## global 클래스를 이용한 방법

global 클래스를 이용한 방법으로써 상속받아야 할 부모 클래스명을 global로 선언합니다.

**TodoList.jsx**

import React from 'react';  
import TodoItem from './TodoItem'  
import style from './TodoList.scss';  
  
const DEMO\_PROPS = {  
 itemList: [  
 { id: 1 },  
 { id: 2 },  
 { id: 3 }  
 ]  
}  
class TodoList extends React.Component {  
 render() {  
 const { itemList } = DEMO\_PROPS;  
  
 return (  
 <ul className="list\_todo">  
 {itemList.map((item) => (  
 <TodoItem key={item.id} />  
 ))}  
 </ul>  
 )  
 }  
}  
  
export default TodoList;

**TodoItem.scss**

.item\_todo {  
 // 컴포넌트의 기본 스타일  
}  
  
:global(.list\_todo) {  
 .item\_todo {   
 // 부모 클래스를 상속 받아 커스텀하고자 하는 스타일  
 }  
}

props를 이용한 타입 전달 방법을 사용함에 있어 프로젝트 특성 상 사용이 힘들거나 복잡도가 너무 늘어날 경우 사용할 수 있는 방법입니다만 추천하지 않습니다.