

React JS

리액트를 활용한 TODOLIST 만들기
교육과정 5개월차 (5월 13일~14일)

React JS



신연재



Contents

1. 프로젝트 개요

2. 주요 기술

3. 실행 화면

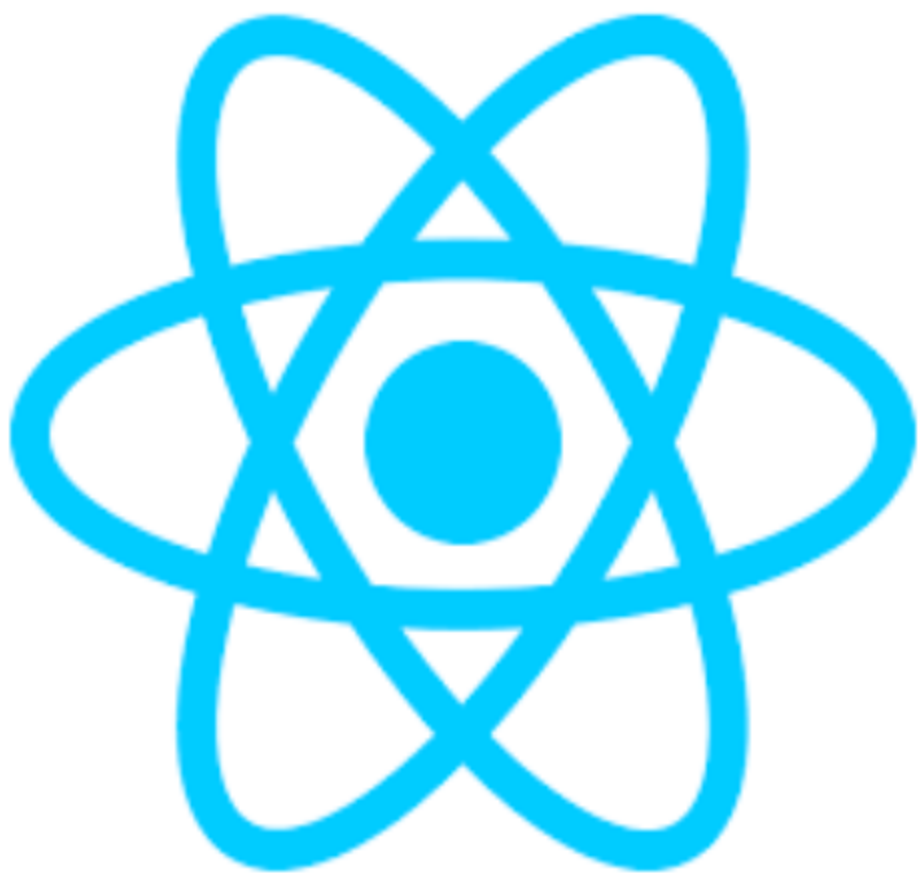
4. 프로젝트 후기

Check Point

- ☒ 아침 운동 가기
- ☐ 마트 가기
- ☐ 프로젝트 준비
- ☐ 강아지 산책하기
- ☒ 영화관 가기

프로젝트 개요

본 프로그램은 REACT교육과정을 수료 후 진행한 간단한 TODOLIST를 기록 할 수 있는 프로그램 입니다. 사용자는 원하는 할일을 작성하고 할일을 추가 또는 삭제를 할 수 있으며 완료된 할일에 한하여 체크를 통해서 완료표시를 할 수 있습니다.



주요 기술

코드 구현 - React



1. 코드의 재사용성을 위해 component로 나누어 구현
2. 항목의 check상태를 관리 하기위해 useState사용
3. 항목을 추가 할때의 상태를 관리 하기위해 useState 사용

화면 구현 - HTML & CSS



간단한 화면 구성을 위해서
순수 HTML과 CSS를 사용 했습니다.

일정 관리

할 일을 입력하세요

+

☐ 해야할 일 : 1

-

☐ 해야할 일 : 2

-

☐ 해야할 일 : 3

-

☐ 해야할 일 : 4

-

☐ 해야할 일 : 5

-

☐ 해야할 일 : 6

-

☐ 해야할 일 : 7

-

☐ 해야할 일 : 8

-

☐ 해야할 일 : 9

-

☐ 해야할 일 : 10

-

☐ 해야할 일 : 11

-

☐ 해야할 일 : 12

-

☐ 해야할 일 : 13

-

☐ 해야할 일 : 14

-

☐ 해야할 일 : 15

-

☐ 해야할 일 : 16

-

실행 화면

```
// 테스트를 위한 대량의 더미데이터 생성 및 반환을 위한 임시 함수 정의
const createDummyTodoList = function () {
  const dummyArray = [];
  for (let i = 1; i <= 20; i++) {
    dummyArray.push({ id: i, title: `해야할 일 : ${i}`, complete: false });
  }
  return dummyArray;
}
```

웹 화면이 나오면 기존에 더미데이터로 만들어 둔 20개의 항목이 나옵니다.
왼쪽 박스에는 항목을 체크 할 수 있는 체크박스가 있고 오른쪽엔 항목을
삭제 할 수 있는 버튼이 있습니다.

실행 화면

일정 관리

강아지 산책 하기 +

☐ 마트 가기 -

일정 관리

할 일을 입력하세요 +

☐ 마트 가기 -

☐ 강아지 산책 하기 -

```
// 새로운 Todo 항목을 입력하고 추가할 수 있는 컴포넌트
// App.js에서 일정 목록(배열)에 일정을 추가하는 함수를 props를 통해 전달받는다.
const TodoInsert = function ({eventHandler}) {
  const [inputValue, setInputValue] = useState('');

  // input 요소 값 변경에 대한 이벤트 처리
  const handleChangeInput = function(event){
    setInputValue(event.target.value);
  }

  // 일정 등록 버튼 이벤트 처리
  const handleAddTodoBtn = function(event){
    event.preventDefault();
    // props를 통해 전달은 함수를 호출하여 일정을 배열에 등록한다.
    eventHandler(inputValue);
    setInputValue('');
  }

  return (
    <form className='TodoInsert' onSubmit={handleAddTodoBtn}>
      <input placeholder='할 일을 입력하세요' value={inputValue} onChange={handleChangeInput} />
      <button type='submit'>
        <MdAdd />
      </button>
    </form>
  );
};
```


```
// 일정 등록
const createTodoFire = function(title) {
  // 새로운 일정 객체 생성 후 배열에 추가
  const todo = { id: App.todoId++, title: title, checked: false };
  // state에 설정
  setTodoList([...todoList, todo]);
}
```


1. App.js에서 새로운 일정 객체를 생성한 후 배열에 추가해서 state에 설정
2. App.js에서 일정을 등록하는 함수를 props를 통해 전달 받습니다.
3. 사용자가 입력을 하고 누른 + 버튼에 이벤트 처리를 합니다.
4. 추가한 항목을 다시 배열에 넣고 화면에 출력을 합니다.

실행 화면

일정 관리

할 일을 입력하세요 +

☒ 마트가기 

☐ 강아지 산책 하기 

일정 관리

할 일을 입력하세요 +

☐ 강아지 산책 하기 

// 일정 삭제

```
const removeTodo = function (id) {  
  const newTodo = todoList.filter(todo => todo.id !== id);  
  setTodoList(newTodo);  
}
```

```
export default React.memo(TodoItem);
```

React.memo를 사용하여 컴포넌트를 최적화합니다. 전달 받은 props가 변경되지 않으면 컴포넌트는 다시 렌더링되지 않습니다.

```
// 각 Todo 항목에 대한 정보를 보여주는 컴포넌트  
const TodoItem = function ( {todo,removeTodo} ) {  
  
  const {id, title, checked} = todo;  
  
  const [inChecked, setChecked] = useState(false);  
  
  const onCheck = () =>{  
    setChecked((e)=>!e);  
  };  
  
  return (  
    <div className='TodoItem'>  
      <div className={`checkbox ${inChecked ? "checked" : " "} }` onClick={onCheck} >  
        {inChecked ? (<MdCheckBox />) : (<MdCheckBoxOutlineBlank />)}  
  
        <div className='text'>{title}</div>  
      </div>  
      <div className='remove' onClick={()=> removeTodo(id)}>  
        <MdRemoveCircleOutline />  
      </div>  
    </div>  
  );  
}
```

1. todo와 removeTodo를 프로퍼티로 받습니다:
2. inChecked와 setChecked라는 상태를 정의하여 체크 여부를 관리합니다:
3. onCheck 함수는 체크박스를 클릭할 때 호출되며, inChecked 상태를 반전시킵니다
4. 체크박스를 클릭하면 할일에 줄이 그어집니다.

프로젝트 후기

리액트를 짧은시간 수강하고 난 후 진행한 프로젝트 였습니다.
기존에 알고 있었던 JavaScript방식의 화면 구성이 아니라서
많이 어려움이 있었고 함수를 구현함에 있어서 어느곳에 함수를
위치 시켜야 할지 몰라서 많이 공부를 하고 진행했습니다.
특히 props를 전달하고 전달받는 것에 어려움이 있었으나
규모는 작지만 간단한 프로젝트를 진행 해보니 개념을 이해하는데
도움이 많이 되었습니다. 나중에 대규모 프로젝트를 한다면
꼭 리액트를 활용하여 화면을 구성해보고 싶습니다. 선생님이 알려
주신 활용 예로는 관리자 모드에 쓰면 좋다고 하셔서 파이널 프로
젝트를 진행하면 꼭 제가 맡아서 해보고 싶단 생각을 하였습니다.



TO DO LIST

1.	_____	<input type="checkbox"/>
2.	_____	<input type="checkbox"/>
3.	_____	<input type="checkbox"/>
4.	_____	<input type="checkbox"/>
5.	_____	<input type="checkbox"/>