React

DAY 3

Rendering

입력상자에 글자를 칠때마다 전체페이지가 Rendering되는 상황이 발생한다.

지금은 페이지가 간단해서 상관이 없겠지만 복잡해진다면 시스템에 부하가 올 것이다.

`React.memo()`

```
import React from 'react'

const index = () => {
  return (
    <div>index</div>
  )
}

export default index
```

Arrow Funtion

Form.js

```
import React from "react";

export default function Form({ todoData, setTodoData, value, setValue }) {
  console.log("Form Component");
  const handleChange = (e) => {
    setValue(e.target.value);
  };
  const handleSubmit = (e) => {
    // Submit event 발생해서 처리하고 있다.
    // 이런 default event처리를 안할래요!
    e.preventDefault();

    let newTodo = {
      id: Date.now(), // Unique한 값을 표현하기 위해서 현재 시간을 찍는다.
      title: value,
      completed: false,
      edited: false,
    };

    setTodoData([...todoData, newTodo]);
    setValue("");
  };
  return (
    <form onSubmit={handleSubmit} className="flex pt-2 ">
      <input
        type="text"
        name="todoItem"
        className=" w-full px-3 py-2 mr-4 text-gray-500 border rounded shadow font-sans  "
        placeholder="Please enter a new to do"
        value={value}
        onChange={handleChange}
      />
      <input
        type="submit"
        className="p-2 text-blue-400 border-blue-400 rounded hover:text-white hover:bg-blue-200 font-sans"
        value="Input"
      />
```

```
    </form>
  );
}
```

App.js

```
import React, { useCallback, useState } from "react";
import "./App.css";
import Form from "./components/Form";
import Lists from "./components/Lists";

// render() method가 존재하지 않는다.
// 우리 함수의 return 값이 JSX이다.
export default function App() {
  console.log("App Component");
  const [todoData, setTodoData] = useState([
    {
      id: "1",
      title: "Training",
      completed: false,
      edited: false,
    },
  ]);
  const [value, setValue] = useState("");
  const editClick = (id) => {
    console.log(id + "edit Click!");
    var ed = document.getElementById("edit");
    ed.type = "text";
    let x = document.getElementsByName("editBtn")[0];
    x.innerText = "Save";
    let Y = document.getElementsByName("titleSpan")[0];
    Y.innerText = "";
  };
  const deleteClick = useCallback(
    (id) => {
      const newTodoData = todoData.filter((data) => data.id !== id);
      setTodoData(newTodoData);
    },
    [todoData]
  );

  return (
    <div className="flex items-center justify-center w-screen h-screen bg-blue-100 font-mono ">
      <div className="w-full p-6 m-4 bg-white rounded shadow md:w-3/4 md:max-w-lg lg:w-3/4 lg: max-w-lg font-mono ">
        <div className="flex justify-between mb-3 font-mono">
          <h2>What is your main focus for today?</h2>
        </div>

        <Lists
          editClick={editClick}
          deleteClick={deleteClick}
          todoData={todoData}
          setTodoData={setTodoData}
        />

        <Form
          todoData={todoData}
          setTodoData={setTodoData}
          value={value}
          setValue={setValue}
        />
      </div>
    </div>
  );
}
```

Lists.js

```
import React from "react";
import { DragDropContext, Draggable, Droppable } from "react-beautiful-dnd";
import List from "./List";

const Lists = React.memo(
  ({ editClick, deleteClick, todoData, setTodoData }) => {
    console.log("Lists Component");
    const handleDrop = (e) => {
      // e: event 객체, event에 대한 세부정보를 가지고 있다.
      // e.source : drag한 객체, e.destination : drop한 객체
      if (!e.destination) return;

      const newTodoData = todoData;

      // drag되는것을 삭제시키는 Code
      const [reorder] = newTodoData.splice(e.source.index, 1);

      // drop되는 위치에 삽입시키는 Code
      newTodoData.splice(e.destination.index, 0, reorder);

      setTodoData(newTodoData);
    };

    return (
      <div>
        <DragDropContext onDragEnd={handleDrop}>
          <Droppable droppableId="to-do">
            {(provided) => (
              <div {...provided.droppableProps} ref={provided.innerRef}>
                {todoData.map((data, index) => (
                  <Draggable
                    key={data.id}
                    draggableId={data.id.toString()}
                    index={index}
                  >
                    {(provided, snapshot) => (
                      // component
                      <List
                        editClick={editClick}
                        deleteClick={deleteClick}
                        id={data.id}
                        title={data.title}
                        completed={data.completed}
                        todoData={todoData}
                        setTodoData={setTodoData}
                        provided={provided}
                        snapshot={snapshot}
                      />
                    )}
                  </Draggable>
                ))}
                {provided.placeholder}
              </div>
            )}
          </Droppable>
        </DragDropContext>
      </div>
    );
  }
);

export default Lists;
```

List.js

```
import React from "react";

const List = React.memo(
  ({
    editClick,
    deleteClick,
```

```
    id,
    title,
    completed,
    edited,
    todoData,
    setTodoData,
    provided,
    snapshot,
  }) => {
    console.log("List Component");

    const handleEditChange = (id, data) => {
      let newTodoData = todoData.map((data) => {
        if (data.id === id) {
          data.edited = !data.edited;
        }
        return data;
      });
      setTodoData(newTodoData);
    };

    const handleCompleteChange = (id) => {
      // id에 대한 todoData의 Completed값을 변경시켜야한다.
      let newTodoData = todoData.map((data) => {
        if (data.id === id) {
          data.completed = !data.completed;
        }
        return data;
      });
      setTodoData(newTodoData);
    };
    return (
      <div
        key={id}
        {...provided.draggableProps}
        ref={provided.innerRef}
        {...provided.dragHandleProps}
      >
        <div
          className={` ${
            snapshot.isDragging ? "bg-violet-200" : "bg-pink-100"
          } flex items-center justify-between w-full px-4 py-1 my-2 text-gray-600  border rounded font-mono`}
        >
          <div className="items-center">
            <input
              type="checkbox"
              defaultChecked={false}
              onChange={() => handleCompleteChange(id)}
            />{" "}
            <input
              type="hidden"
              id="edit"
              onChange={() => handleEditChange(id)}
            />{" "}
            <span
              name="titleSpan"
              className={completed ? "line-through" : undefined}
            >
              {title}
            </span>
          </div>

          <div className="items-center font-mono ">
            <button class="px-3" name="editBtn" onClick={() => editClick(id)}>
              edit
            </button>
            <button class="px-1" onClick={() => deleteClick(id)}>
              delete
            </button>
          </div>
        </div>
      </div>
    );
  }
);
```

```
export default List;
```