



실습 워크북: To-Do List 만들기



학습 목표

- `useState` 를 이용해 상태 관리하기
- `useCallback` 으로 이벤트 핸들러 최적화하기
- `useMemo` 로 계산된 값 캐싱하기
- 스타일링(Tailwind / DaisyUI / CSS)을 직접 적용해보기



실습 요구사항

1. 할 일 추가 기능

- 입력창에 텍스트 입력 후 "추가" 버튼 클릭 시 새로운 할 일이 목록에 나타나야 한다.
- 입력창은 비워져야 한다.

2. 할 일 완료/취소 기능

- 각 항목 옆의 버튼 클릭 시 완료 여부를 토글할 수 있어야 한다.
- 완료된 항목은 스타일로 구분되도록 (예: 줄 긋기, 색상 변경) 표현한다.

3. 완료된 개수 표시

- 화면 하단에 "완료된 일: n개" 형태로 보여주기.
- `useMemo` 를 이용해 계산.

4. 화면 디자인은 직접 구성

- TailwindCSS, DaisyUI 또는 자유롭게 CSS 사용.
- 입력창, 버튼, 목록 디자인은 학생 개성에 맞게 구성.

5. (선택 과제)

- 항목 삭제 기능 추가
- 완료된 항목만 보기 / 전체 보기 토글 버튼 추가
- "Enter" 키 입력 시 할 일 추가



코드 뼈대 (학생들이 채워 넣을 부분 포함)

```
import { useState, useMemo, useCallback } from "react";

type Todo = {
  id: number;
  text: string;
  done: boolean;
};

export default function TodoApp() {
  // 상태 정의
  const [todos, setTodos] = useState<Todo[]>([]);
  const [text, setText] = useState("");

  // 할 일 추가 함수
  const addTodo = useCallback(() => {
    // TODO: text가 비어있지 않으면 새로운 todo 추가
    // setTodos(...) 사용
    // setText("") 로 입력창 비우기
  }, [text]);

  // 완료 토글 함수
  const toggleTodo = useCallback((id: number) => {
    // TODO: 해당 id의 todo.done 값을 반전시켜 업데이트
  }, []);

  // 완료된 개수 (useMemo 활용)
  const doneCount = useMemo(() => {
    // TODO: 완료된 todo 개수를 리턴
    return 0;
  }, [todos]);

  return (
    <div>
      <h1>✨ To-Do List</h1>

      {/* 입력창 + 버튼 */}
    </div>
  );
}
```

```

<div>
  { /* TODO: 입력창과 버튼을 배치하고, 디자인은 학생들이 구성 */ }
  <input value={text}
    onChange={(e) => setText(e.target.value)}
    placeholder="할 일을 입력하세요"
  />
  <button onClick={addTodo}>추가</button>
</div>

{ /* 목록 */ }
<ul>
  { todos.map((todo) => (
    <li key={todo.id}>
      { /* TODO: 완료된 항목은 스타일로 구분되도록 처리 */ }
      <span>{todo.text}</span>
      <button onClick={() => toggleTodo(todo.id)}>
        {todo.done ? "취소" : "완료"}
      </button>
    </li>
  )) }
</ul>

{ /* 완료 개수 */ }
<p>완료된 일: {doneCount}개</p>
</div>
);
}

```

힌트

- 새 할 일 추가 시 `Date.now()` 를 id로 사용 가능
- 배열 업데이트는 `map`, `filter`, 스프레드 연산자(`[...]`)를 활용
- 완료 스타일은 `className` 을 조건부(`todo.done ? "line-through" : ""`)로 적용