# 4주차 과제 #1 TodoList 기능 명세

|  |  |  |
| --- | --- | --- |
| 기능 번호 | 세부 기능 설명 | 추가기능여부 |
| FR1 | 일정을 추가할 수 있어야 합니다. | N |
| FR2 | 일정을 삭제할 수 있어야 합니다. | N |
| FR3 | 일정을 수정할 수 있어야 합니다. | Y |
| FR4 | 일정을 완료 표시 할 수 있어야 합니다 | Y |

# 구현

## 구현 사항

|  |  |  |
| --- | --- | --- |
| 기능 번호 | 완료여부  (O,X) | 세부사항 |
| FR1 | O | <진행사항>  내용을 작성한 뒤 ‘Add’ 버튼을 누르면 박스 안에 일정이 추가됩니다 |
| FR2 | O | <진행사항>  입력된 일정의 옆에 위치한 ‘Delete’ 버튼을 누르면 그 일정이 삭제됩니다. |
| FR3 | O | <진행사항>  ‘Edit’ 버튼을 누르면 prompt 창이 뜨고 수정할 내용을 입력하고 확인을 누르면 수정된 내용이 목록에 렌더링 됩니다. |
| FR4 | O | <진행사항>  목록에 있는 체크 박스를 누르면 글자가 회색으로 바뀌고 선이 그어져 완료된 목록인 것을 확인할 수 있습니다. |

## 웹 화면 구성

제목 : TodoList 화면 (메인)

|  |  |
| --- | --- |
| 화면 | |
| ② | |
| 설명 | 1. 첫 화면에 일정 목록이 들어갈 수 있는 박스와 입력 칸 등등이 보인다. 2. text area에 내용을 입력할 수 있다. |

제목 : 일정 추가

|  |  |
| --- | --- |
| 화면 | |
|  | |
| 설명 | 1. ‘Add’ 버튼을 누르면 박스 안에 일정 내용이 추가된다 2. 새로운 일정을 적고 또 ‘Add’ 버튼을 누르면 기존 일정 밑에 새로운 일정이 추가된다. |

제목 : 일정 수정

|  |  |
| --- | --- |
| 화면 | |
|  | |
| 설명 | 1. ‘Edit’ 버튼을 누르면 prompt 창이 뜨고 Edit Task 부분에 수정할 내용을 작성한다 2. 확인을 누르면 수정한 내용이 적용되고 렌더링되어 보여진다 |

제목 : 일정 완료 체크

|  |  |
| --- | --- |
| 화면 | |
|  | |
| 설명 | 체크박스를 누르면 일정 내용의 글씨 색이 회색으로 변하고 밑줄이 그어져 완료된 것을 확인할 수 있다. |

제목 : 일정 목록 길이 유지

|  |  |
| --- | --- |
| 화면 | |
|  | |
| 설명 | 특정 길이 이상의 내용을 적고 일정을 추가하면 줄바꿈이 적용되어 전체 목록의 크기를 일정하게 유지. |

제목 : 일정 삭제

|  |  |
| --- | --- |
| 화면 | |
|  | |
| 설명 | 특정 일정 목록의 ‘Delete’ 버튼을 누르면 그 일정이 삭제되고, ‘Clear All’ 버튼을 누르면 전체 일정 목록이 사라진다. |

## 구현 내용

### HTML

|  |  |
| --- | --- |
| 파일명 | 구현 내용 |
| 4week\_hw\_1.html | TodoList의 기본 화면을 구성하고 css와 js를 적용 |

### CSS

|  |  |
| --- | --- |
| 파일명 | 구현 내용 |
| 4week\_hw\_1.css | 가장 심플한 스타일로 구현. |

### JavaScript

파일 명 : 4week\_hw\_1.js

|  |  |
| --- | --- |
| 함수명 | 구현 내용 |
| renderTasks() | 함수 인자 설명 : 없음  리턴값 : 없음  동작 설명  - `tasks` 배열에 있는 각 항목에 대해 할일 목록의 각 요소를 생성하고, 체크박스, 텍스트, 수정 버튼, 삭제 버튼이 포함된 HTML 요소를 생성합니다.  - 각 버튼에는 클릭 이벤트 핸들러가 할당되어 해당 작업을 수행합니다.  - 생성된 HTML 요소를 `tasksContainer`에 추가하여 화면에 표시합니다. |
| toggleTaskStatus(index) | 함수 인자 설명  `index`: 체크박스가 속한 할일의 인덱스  리턴값 : 없음  동작 설명  - 체크박스가 체크되어 있으면 해당 할일의 텍스트에 'completed' 클래스를 추가하여 완료된 표시를 합니다.  - 체크박스가 해제되어 있으면 'completed' 클래스를 제거합니다. |
| addTask() | 함수 인자 설명 : 없음  리턴값 : 없음  동작 설명  - `newTaskInput` 엘리먼트에서 입력된 값을 가져와 공백을 제거한 후, 배열 `tasks`에 추가합니다.  - 입력란을 비워주고, 할일 목록을 다시 렌더링합니다. |
| deleteTask(index) | 함수 인자 설명 :  `index`: 삭제할 할일의 인덱스  리턴값 : 없음  동작 설명  - `tasks` 배열에서 해당 인덱스의 항목을 제거하고, 할일 목록을 다시 렌더링합니다. |
| editTask(index) | 함수 인자 설명  `index`: 수정할 할일의 인덱스  리턴값 : 없음  동작 설명  - `prompt`를 사용하여 사용자에게 수정할 내용을 입력받고, 해당 내용으로 선택된 할일을 수정한 후, 할일 목록을 다시 렌더링합니다. |
| clearTasks() | 함수 인자 설명 : 없음  리턴값 : 없음  동작 설명  - `tasks` 배열을 비우고, 할일 목록을 다시 렌더링합니다. |
| Initial rendering | 함수 인자 설명 : 없음  리턴값 : 없음  동작 설명  - `renderTasks` 함수를 호출하여 초기 할일 목록을 화면에 표시합니다. |

# 테스트 결과

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | 테스트 제목 | 테스트 절차 | 관련 FR | 테스트 결과(O,X) |
| 1 | 일정 추가 테스트 | 1. 내용 입력 칸에 내용을 입력한다.  2. ‘Add’버튼을 누른다.  3. 일정 목록에 추가가 되는지 확인한다. | FR1 | O |
| 2 | 일정 삭제 테스트 | 1. Delete 버튼을 누른다  2. 특정 일정만 삭제된다.  3. Clear All을 누르면 모든 일정이 삭제된다. | FR2 | O |
| 3 | 일정 수정 테스트 | 1. Edit 버튼을 누른다.  2. prompt가 뜨면 수정할 내용을 작성한다.  3. 확인을 누르면 일정 목록의 내용이 수정한 내용으로 변하는지 확인한다. | FR3 | O |
| 4 | 일정 완료 테스트 | 1체크박스를 누른다.  2. 체크박스를 누른 일정이 글자가 회색으로 변하고 선이 그어지는지 확인한다.  4. 체크박스를 한번 더 누르면 원래대로 돌아오는지 확인한다. | FR4 | O |
| 5 | 일정 내용 줄 바꿈 테스트 | 1. 10글자 정도 내용을 적은 뒤 일정을 추가한다.  2. 30글자 정도 적은 뒤 일정을 또 추가한다.  3. 많은 양의 글자는 자동으로 줄 바꿈이 되고 전체 화면이 일정 길이를 유지하는지 확인한다. | FR1 | O |

# 집중 피드백 받고 싶은 사항

|  |  |
| --- | --- |
| 번호 | 내용 |
| 1 | 생각나는게 없음 |
| 2 |  |
| 3 |  |

참고

https://