

2025년 상반기 K-디지털 트레이닝

단일 파일 컴포넌트를 이용한 Vue 애플리케이션 개발(심화1)

[KB] IT's Your Life

- ✓ 다음과 같이 프로젝트를 생성하고 초기화 하세요.
 - 프로젝트명 `todolist-app`
 - 프로젝트 템플릿 생성 질문에 모두 No
 - 기본 패키지 설치 및 `bootstrap@5` 패키지 설치
 - 기존 기본 컴포넌트와 스타일 파일들 모두 삭제

✓ 프로젝트 만들기

```
npm init vue todolist-app
```

--> 모두 No 선택

```
cd todolist-app
```

```
npm install
```

```
npm install bootstrap@5
```

✓ assets/main.css를 추가하세요.

✓ 다음 컴포넌트들을 추가하세요.

○ 컴포넌트 목록

- InputTodo.vue
- TodoList.vue
- TodoListItem.vue

✓ 각 컴포넌트의 내용은 다음과 같이 작성하세요.

```
<template>
```

```
</template>
```

```
<script>
```

```
  export default {
```

```
    name : "실제컴포넌트명", // 각각 InputTodo, TodoList, TodoListItem
```

```
  }
```

```
</script>
```

src/InputTodo.vue

```
<template>

</template>

<script>
  export default {
    name : "InputTodo",
  }
</script>
```

src/ToDoListItem.vue

```
<template>

</template>

<script>
  export default {
    name : "ToDoListItem",
  }
</script>
```

src/ToDoList.vue

```
<template>

</template>

<script>
  export default {
    name : "ToDoListI",
  }
</script>
```

✓ src/assets/main.css을 다음과 같이 작성하세요

```
body {  
  margin: 0;  
  padding: 0;  
  font-family: sans-serif;  
}  
.title {  
  text-align: center;  
  font-weight: bold;  
  font-size: 20pt;  
}  
.todo-done {  
  text-decoration: line-through;  
}  
.container {  
  padding: 10px 10px 10px 10px;  
}  
.panel-borderless {  
  border: 0;  
  box-shadow: none;  
}  
.pointer {  
  cursor: pointer;  
}
```

✓ main.js를 다음과 같이 초기화한다.

```
import { createApp } from 'vue'
import App from './App.vue'
import 'bootstrap/dist/css/bootstrap.css'
import './assets/main.css'

createApp(App).mount('#app')
```


✓ App.vue를 다음과 같이 작성하세요.

```
<template>
  <div id="app" class="container">
    <div class="card card-body bg-light">
      <div classe="title">:: Todolist App</div>
    </div>
    <div class="card card-default card-borderless">
      <div class="card-body">
        <InputTodo />
        <TodoList />
      </div>
    </div>
  </div>
</template>
```

✓ App.vue를 다음과 같이 작성하세요.

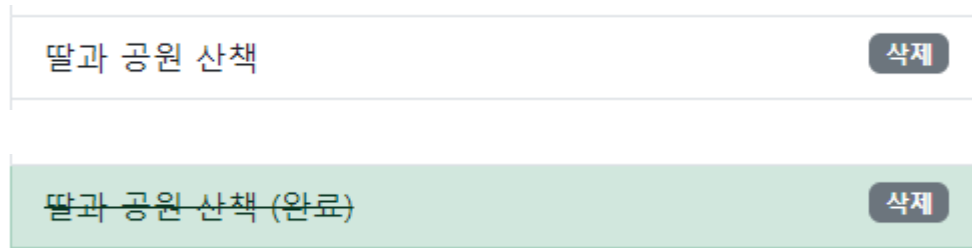
```
<script>
  import TodoList from './components/TodoList.vue'
  import InputTodo from './components/InputTodo.vue'

  let ts = new Date().getTime()

  export default {
    name: "App",
    components : { InputTodo, TodoList },
    data() {
      return {
        todoList : [
          { id: ts, todo:"자전거 타기", completed: false },
          { id: ts+1, todo:"딸과 공원 산책", completed: true },
          { id: ts+2, todo:"일요일 애견 카페", completed: false },
          { id: ts+3, todo:"Vue 원고 집필", completed: false },
        ]
      }
    },
  }
</script>
```

✓ TodoList 컴포넌트에 다음 속성을 추가하세요.

- 속성명: todos
 - 할 일 객체 목록을 가지는 배열
- 해당 속성으로 전달된 데이터를 화면에 출력함
- 이에 맞게 TodoListItem.vue를 수정하세요.



✓ App.vue

```
<template>
  <div id="app" class="container">
    <div class="card card-body bg-light">
      <div class="title">:: Todolist App</div>
    </div>
    <div class="card card-default card-borderless">
      <div class="card-body">
        <InputTodo />
        <TodoList :todoList="todoList"/>
      </div>
    </div>
  </div>
</template>
```

✓ components/TodoListItem.vue

```
<template>
  <li
    class="list-group-item"
    :class="{ 'list-group-item-success': todoItem.completed }">
    <span class="pointer" :class="{ 'todo-done': todoItem.completed }">
      {{ todoItem.todo }} {{ todoItem.completed ? '(완료)' : '' }}
    </span>
    <span
      class="float-end badge bg-secondary pointer"
      >삭제</span>
    </li>
</template>

<script>
export default {
  name: 'TodoListItem',
  props: {
    todoItem: { type: Object, required: true },
  },
};
</script>
```

✓ components/TodoList.vue

```
<template>
  <div class="row">
    <div class="col">
      <ul class="list-group">
        <TodoListItem
          v-for="todoItem in todoList"
          :key="todoItem.id"
          :todoItem="todoItem"/>
      </ul>
    </div>
  </div>
</template>

<script>
import TodoListItem from './TodoListItem.vue';
export default {
  name: 'TodoList',
  components: { TodoListItem },
  props: {
    todoList: { type: Array, required: true },
  },
};
</script>
```

2025년 상반기 K-디지털 트레이닝

단일 파일 컴포넌트를 이용한 Vue 애플리케이션 개발(심화3)

[KB] IT's Your Life

- ✔ 다음과 같은 이벤트를 추가하여 App.vue를 수정하세요.

이벤트명	이벤트 핸들러	처리 내용	이벤트 발생 객체
add-todo	addTodo	새 할 일 추가	InputTodo
toggle-completed	toggleCompleted	완료 여부 토글	TodoListItem, TodoList
delete-todo	deleteTodo	해당 할 일 삭제	TodoListItem, TodoList

- 모든 이벤트 처리는 App.vue가 담당함

✓ App.vue

```
<template>
  <div id="app" class="container">
    <div class="card card-body bg-light">
      <div classe="title">:: Todolist App</div>
    </div>
    <div class="card card-default card-borderless">
      <div class="card-body">
        <InputTodo @add-todo="addTodo" />
        <TodoList :todoList="todoList" @delete-todo="deleteTodo" @toggle-completed="toggleCompleted" />
      </div>
    </div>
  </div>
</template>
```

✓ App.vue

```
<script>
import TodoList from './components/TodoList.vue'
import InputTodo from './components/InputTodo.vue'

let ts = new Date().getTime()

export default {
  name: "App",
  components : { InputTodo, TodoList },
  data() {
    return {
      todoList : [
        { id: ts, todo:"자전거 타기", completed: false },
        { id: ts+1, todo:"딸과 공원 산책", completed: true },
        { id: ts+2, todo:"일요일 애견 카페", completed: false },
        { id: ts+3, todo:"Vue 원고 집필", completed: false },
      ]
    }
  },
}
```

✓ App.vue

```
methods: {  
  addTodo(todo) {  
    if (todo.length >= 2) {  
      this.todoList.push({ id: new Date().getTime(), todo: todo, completed: false });  
    }  
  },  
  deleteTodo(id) {  
    let index = this.todoList.findIndex((item) => id === item.id);  
    this.todoList.splice(index, 1);  
  },  
  toggleCompleted(id) {  
    let index = this.todoList.findIndex((item) => id === item.id);  
    this.todoList[index].completed = !this.todoList[index].completed;  
  }  
}  
</script>
```

✓ **InputTodo.vue** 컴포넌트를 다음과 같이 정의하세요.

- 화면 구성은 다음과 같음

추가

- 엔터를 치거나 추가 버튼을 눌렀을 때 새 할 일 등록되게 함

✓ components/InputTodo.vue

```
<template>
  <div class="row mb-3">
    <div class="col">
      <div class="input-group">
        <input id="msg" type="text" class="form-control" name="msg"
          placeholder="할일을 여기에 입력!" v-model.trim="todo"
          @keyup.enter="addTodoHandler" />
        <span class="btn btn-primary input-group-addon"
          @click="addTodoHandler">추가</span>
      </div>
    </div>
  </div>
</template>
```

✓ components/InputTodo.vue

```
<script>
  export default {
    name : "InputTodo",
    data() {
      return { todo : "" }
    },
    emits : ["add-todo"],
    methods : {
      addTodoHandler() {
        if (this.todo.length >= 3) {
          this.$emit('add-todo', this.todo);
          this.todo = "";
        }
      },
    },
  }
</script>
```

- ✓ **TodoListItem**을 다음과 같이 작성하세요.
 - 제목을 클릭 시 상태를 반전 및 toggle-completed 이벤트 발생
 - 삭제 버튼 클릭 시 delete-todo 이벤트 발생

✓ components/TodoListItem.vue

```
<template>
  <div class="row mb-3">
    <div class="col">
      <div class="input-group">
        <input id="msg" type="text" class="form-control" name="msg"
          placeholder="할일을 여기에 입력!" v-model.trim="todo"
          @keyup.enter="addTodoHandler" />
        <span class="btn btn-primary input-group-addon"
          @click="addTodoHandler">추가</span>
      </div>
    </div>
  </div>
</template>
```


✓ components/TodoListItem.vue

```
<script>
  export default {
    name : "InputTodo",
    data() {
      return { todo : "" }
    },
    emits : ["add-todo"],
    methods : {
      addTodoHandler() {
        if (this.todo.length >= 3) {
          this.$emit('add-todo', this.todo);
          this.todo = "";
        }
      },
    },
  }
</script>
```

✓ TodoList 컴포넌트를 다음과 같이 작성하세요.

○ 화면 구성

자전거 타기	삭제
딸과 공원 산책 (완료)	삭제
일요일 애견 카페	삭제
Vue 원고 집필	삭제

○ TodoListItem에서 발생하는 이벤트를 App.vue로 전달

✓ components/TodoList.vue

```
<template>
  <div class="row">
    <div class="col">
      <ul class="list-group">
        <TodoListItem
          v-for="todoItem in todoList"
          :key="todoItem.id"
          :todoItem="todoItem"
          @delete-todo="$emit('delete-todo', $event)"
          @toggle-completed="$emit('toggle-completed', $event)"
        />
      </ul>
    </div>
  </div>
</template>
```

✓ components/TodoList.vue

```
<script>
import TodoListItem from './TodoListItem.vue';

export default {
  name: 'TodoList',
  components: { TodoListItem },
  props: {
    todoList: { type: Array, required: true },
  },
  emits: ['delete-todo', 'toggle-completed'],
};
</script>
```