KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

Composition API(심화1, 2)

[KB] IT's Your Life

KB국민은행

Composition API

- ✓ 기존 ToDo 앱을 Composition API를 사용하는 것으로 다시 작성하세요.

Composition API

src/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>
```

1 Composition API

src/components/InputTodo.vue

```
</template>

<script setup>
import { ref } from 'vue';

const emit = defineEmits(['add-todo']);
const todo = ref('');
const addTodoHandler = () => {
  if (todo.value.length >= 3) {
    emit('add-todo', todo.value);
    todo.value = '';
  }
};
</script>
```

Composition API

src/components/TodoListItem.vue

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

<script setup>
const props = defineProps({
  todoItem: { type: Object, required: true },
});
const emit = defineEmits(['delete-todo', 'toggle-completed']);
</script>
```

Composition API

src/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', todoItem.id)"
          @toggle-completed="emit('toggle-completed', todoItem.id)"
        />
      </ul>
    </div>
  </div>
</template>

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

const props = defineProps({
  todoList: { type: Array, required: true },
});
const emit = defineEmits(['delete-todo', 'toggle-completed']);
</script>
```

Composition API

src/App.vue

```
<template>
  <div 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"></InputTodo>
        <TodoList
          :todoList="state.todoList"
          @delete-todo="deleteTodo"
          @toggle-completed="toggleCompleted"
        ></TodoList>
      </div>
    </div>
  </div>
</template>
```

Composition API

src/App.vue

```
<script setup>
import { reactive, onMounted } from 'vue';
import InputTodo from './components/InputTodo.vue';
import TodoList from './components/TodoList.vue';

const ts = new Date().getTime();
const state = reactive({ todoList: [] });

onMounted(() => {
  state.todoList.push({ id: ts, todo: '자전거 타기', completed: false });
  state.todoList.push({ id: ts + 1, todo: '딸과 공원 산책', completed: true });
  state.todoList.push({
    id: ts + 2,
    todo: '일요일 애견 카페',
    completed: false,
  });
  state.todoList.push({ id: ts + 3, todo: 'Vue 원고 집필', completed: false });
});
```

Composition API

src/App.vue

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