

Do it! Vue.js 입문 - 06장

실전 프로젝트 제작

- 배운 내용을 종합하는 실전 애플리케이션 제작
- 컴포넌트 구조화, 컴포넌트 통신, 뷰 템플릿, CLI를 모두 사용

완성된 앱 확인하기

프로젝트 생성 및 구조 확인

- 5장에서 배운 뷰 CLI를 이용하여 프로젝트 구성

```
vue init webpack-simple  
npm install
```

프로젝트 초기 설정

- 반응형 웹 디자인 태그 설정
- 폰트 및 파비콘 설정

컴포넌트 생성하고 등록하기

- TodoHeader, TodoInput, TodoList, TodoFooter 컴포넌트 각각 생성 후 App에 등록

```
<div id="app">  
  <TodoHeader></TodoHeader>  
  <TodoInput></TodoInput>  
  <TodoList></TodoList>  
  <TodoFooter></TodoFooter>  
</div>
```

```
import TodoHeader from './components/TodoHeader'
import TodoInput from './components/TodoInput'
import TodoList from './components/TodoList'
import TodoFooter from './components/TodoFooter'

components: {
  'TodoHeader': TodoHeader,
  'TodoInput': TodoInput,
  'TodoList': TodoList,
  'TodoFooter': TodoFooter
}
```

컴포넌트 내용 구현하기

- TodoHeader : 앱 제목을 나타내는 코드 작성
- TodoInput : 할 일을 입력하는 코드 작성
- TodoList : 저장된 할 일 목록을 표시하는 코드 작성
- TodoFooter : 저장된 목록을 모두 삭제하는 코드 작성

기존 애플리케이션의 문제점 확인 및 수정하기

- 컴포넌트 분할로 인해 컴포넌트 간에 통신이 안되는 점을 props와 event emit으로 해결
- TodoInput, TodoList, TodoFooter 컴포넌트 수정

사용자 경험 개선을 위한 기능 추가

- 뷰 애니메이션을 이용하여 추가, 삭제에 부드러운 느낌 추가하기
- 뷰 모달을 이용해 아무 것도 입력하지 않을 때 경고 창 띄우기

끝