

2018-2  
웹 시스템 설계  
실습

*Week 8*

Vue.js (basic)

Composed by:  
WISE Research Lab Ajou University



AJOU UNIVERSITY



# Preliminary

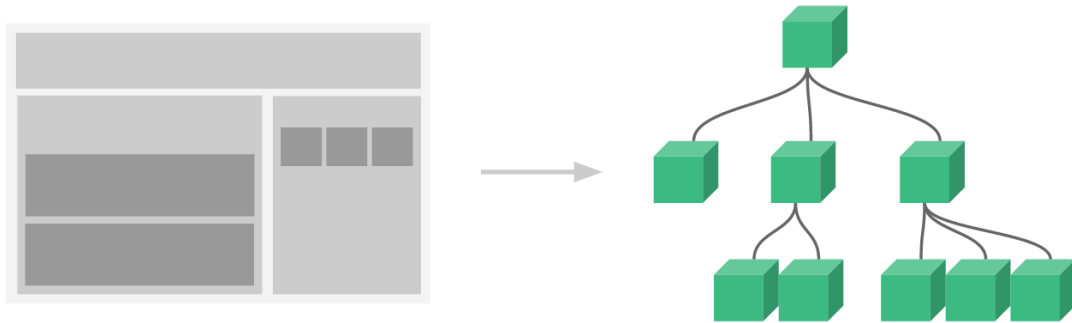
---

## ■ 실습 목표

- vue-cli 를 설치하고 Vue.js 를 활용한 연락처를 개발한다.

## ■ Vue.js

- Vue.js 는 SPA 앱을 개발하기 위한 Web front-end framework 이다.
- Vue.js 는 component 기반의 시스템 설계를 통해, 구현된 기능의 재사용성을 높이고 대규모 프로젝트 개발에 도움을 준다.



## ■ Vue.js 기초

### ■ 인스턴스

- 인스턴스는 Vue 앱을 개발하기 위해서 가장 기본적으로 선언 해야할 요소이다.
- 인스턴스 선언과 함께 앱에 필요한 여러 옵션을 정의한다.
  - ✓ 데이터: 해당 vue 앱에서 사용될 데이터를 정의한다.
  - ✓ 템플릿: 해당 vue 앱에서 사용될 가상 DOM 을 HTML 엘리먼트를 이용하여 정의한다.
  - ✓ 마운트할 엘리먼트: 특정 HTML 엘리먼트에 해당 vue 앱의 가상 DOM 을 렌더링한다.
  - ✓ 옵션에 대한 설명은, component 와 vue-cli 에서 다시 다룬다.
  - ✓ 더 자세한 옵션은 <https://kr.vuejs.org/v2/api/> 참조

### ■ 템플릿

- Vue.js 는 인스턴스 내부의 데이터를 DOM 에 바인딩할 수 있는 템플릿 기능을 제공한다.

## ■ 보간법

- 문자열
  - ✓ Mustache (이중 중괄호)를 사용한 텍스트 보간이 가능하다.

```
<span>{{ data_in_instance }}</span>
```

- ✓ v-once 디렉티브를 통해, 업데이트하지 않는 일회성 보간이 가능하다.

```
<span v-once>다시는 변경하지 않습니다: {{ msg }}</span>
```

## ■ 디렉티브

- v- 접두사가 있는 특수 속성.
- 속성값에 단일 Javascript 표현 식을 작성한다.
- 표현 식의 값이 변경될 때, 사이드 이펙트를 반응적으로 DOM 에 적용한다.
- v-if
  - ✓ 디렉티브의 속성 값에 따라, 엘리먼트를 표현하거나 없앨 수 있다.

```
<h1 v-if="ok">Yes</h1>
```

- ✓ If 외에도 else else-if 도 지원한다.
- ✓ 엘리먼트 뿐만아니라, 템플릿에도 적용가능하다.

- v-bind
  - ✓ 엘리먼트의 클래스 목록 혹은 인라인 스타일을 조작하기 위해 일반적으로 사용된다.

```
1. <div class="static"
2.   v-bind:class="{ active: isActive, 'text-danger': hasError }">
3. </div>
4.   // data in instance
5. data: {
6.   isActive: true,
7.   hasError: false
8. }
```

- ✓ 위 코드를 통해 다음과 같이 class 를 bind 할 수 있다.

```
<div class="static active"></div>
```

- ✓ 클래스뿐만 아니라, component template 에 데이터를 전달할 때도 사용된다.

```
1. // App.vue
2. // v-bind 를 통해 데이터를 전달할 수 있다.
3. <template>
4. <some-components v-bind:dataB="dataA" ></some-components>
5. </template>
6. <script>
7. export default {
8. ...
9. data() {
10.   return {
11.     dataA: "hello"
12.   }
13. }
14. </script>
15.
16.
17. // some-components.vue
18. // 데이터를 받은 component 에서는 props 를 통해 해당 데이터를 정의할 수 있다
19. .
19. <template>
20. <h2>{{dataB}}</h2>
21. </template>
22. <script>
23. export default {
24. ...
25. props: ['dataB']
26. }
27. </script>
```

## ● v-on

- ✓ v-on 디렉티브를 통해 DOM 이벤트를 듣고 트리거 될 때, Javascript 를 실행할 수 있다.

```
1. <div id="example-1">
2.   <button v-on:click="counter += 1">Add 1</button>
3.   <p>위 버튼을 클릭한 횟수는 {{ counter }} 번 입니다.</p>
4. </div>
5. var example1 = new Vue({
6.   el: '#example-1',
7.   data: {
8.     counter: 0
9.   }
10. })
```

- ✓ Form 제출시 페이지 로드 없이 이벤트 처리도 쉽게 가능하다.

1. `<!-- 제출 이벤트가 페이지를 다시 로드 하지 않습니다 -->`
2. `<form v-on:submit.prevent="onSubmit"></form>`

- v-for

- ✓ v-for 을 통해 배열을 기반으로 리스트를 렌더링 할 수 있다.
  - HTML 엘리먼트 뿐만 아니라, 템플릿도 v-for 을 이용하여 렌더링할 수 있다.
- ✓ v-for 의 속성 값에는 item in items 문법이 사용된다.
  - item 은 배열의 엘리먼트, items 는 배열 자체이다.

```
1. <ul id="example-1">
2.   <li v-for="item in items">
3.     {{ item.message }}
4.   </li>
5. </ul>
6. var example1 = new Vue({
7.   el: '#example-1',
8.   data: {
9.     items: [
10.      { message: 'Foo' },
11.      { message: 'Bar' }
12.    ]
13.  }
14. })
```

- ✓ Object 배열의 경우 iteration 을 index 할 수 있는 key 를 v-bind:key=""를 통해 지정해야 한다.

```
1. <div v-for="item in items" :key="item.id">
2.   <!-- content -->
3. </div>
```

- v-model

- ✓ v-model 을 통해 HTML form input 과 textarea 엘리먼트의 양방향 데이터 바인딩을 생성할 수 있다.
- ✓ input[type=text]

```
1. <input v-model="message" placeholder="여기를 수정해보세요">
2. <p>메시지: {{ message }}</p>
```

- ✓ textarea

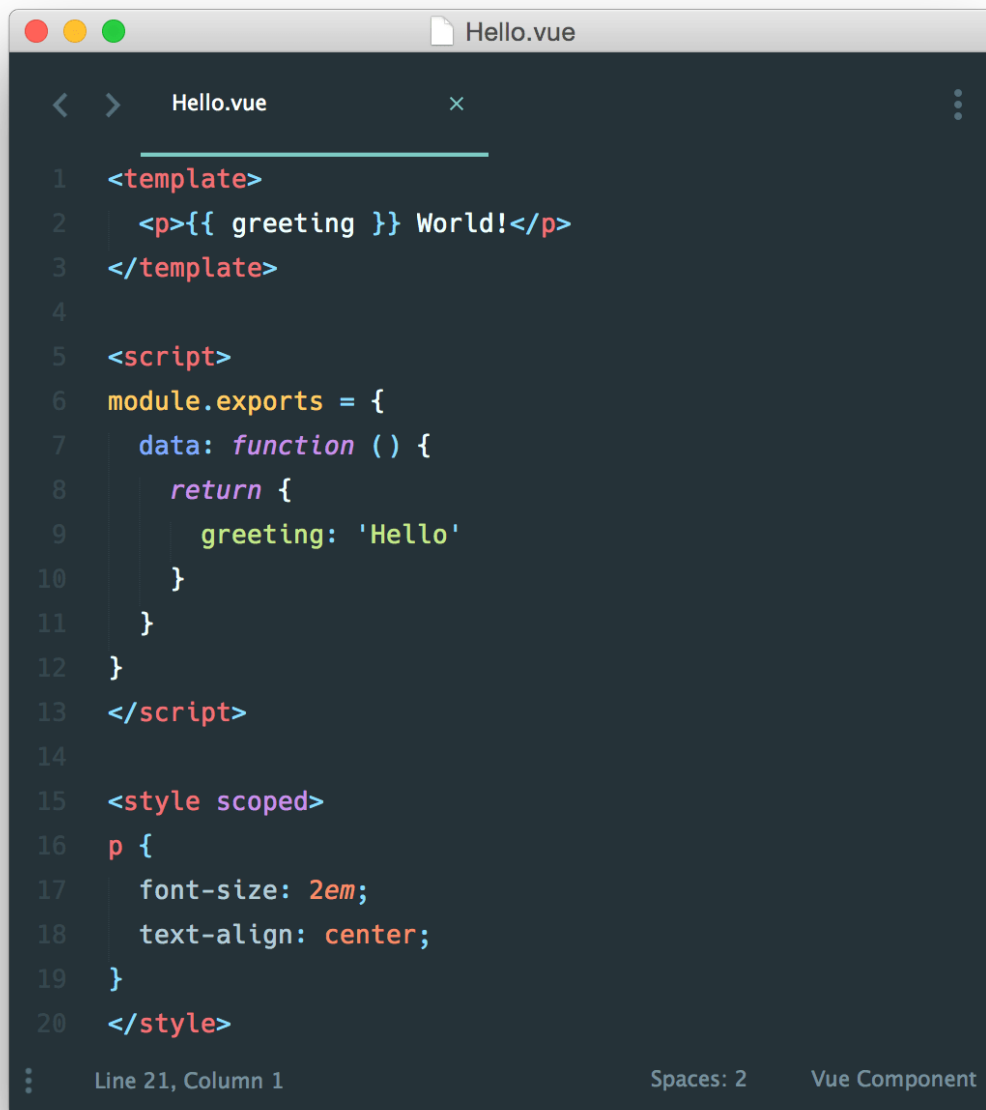
```
1. <span>여러 줄을 가지는 메시지:</span>
```

```
2. <p style="white-space: pre-line">{{ message }}</p>
3. <br>
4. <textarea v-model="message" placeholder="여러줄을 입력해보세요">
```

## ■ component

- component 를 통해 기본 HTML 엘리먼트를 확장하여 재사용 가능한 코드를 캡슐화한다.
- component 는 인스턴스처럼 기존의 인스턴스 옵션들을 사용할 수 있다.
- .vue file 로 single file component 를 생성할 수 있다.
  - ✓ 기존 component 의 좁고 한정적인 정의 방식을 벗어나 새로운 방식으로 개발한다.
    - 기존 문제: CSS 지원 없음, 문자열 템플릿, Pug, Babel 과 같은 전처리 없음.
  - ✓ Webpack, Browserify 와 같은 빌드 도구를 통해 .vue 확장자를 가진 single file component 를 제작한다.
  - ✓ 아래 그림은 .vue file 로 만든 component 의 예시이다.
  - ✓ Template(템플릿)은 component 에 쓰일 HTML 엘리먼트를 정의한다.
  - ✓ Script 는 component 에 쓰일 vue 인스턴스 옵션 혹은 외부 component 사용을 정의한다. 외부에서 사용할 수 있도록 component 객체는 module.exports 로 선언한다.
    - Component 객체에 정의하는 vue 인스턴스 옵션은 다음과 같다. (name,
      - name: component 의 이름을 정의한다. 외부에서 component 를 DOM 으로 불러올 때 사용되는 이름이다.
      - methods: component 에서 사용될 함수들이다.
    - Component 의 값을 조작하거나, 참조하여 사용자의 의도에 맞게 결과값을 도출한다.
    - Components: component 에서 사용되는 외부 component 들을 선언한다. 외부 component 를 사용하기 위해서는 script 에서 import 해야 한다.

- Props: 외부 component 에서 v-bind 를 통해 데이터를 전달할 때, props 를 통해 전달받을 수 있다.
  - data() : componenet 에 쓰일 데이터를 정의한다. Data()의 경우는 함수로 정의된다.
- ✓ Style 은 해당 component 템플릿에서 사용될 스타일을 정의한다.



```
1 <template>
2   <p>{{ greeting }} World!</p>
3 </template>
4
5 <script>
6   module.exports = {
7     data: function () {
8       return {
9         greeting: 'Hello'
10      }
11    }
12  }
13 </script>
14
15 <style scoped>
16   p {
17     font-size: 2em;
18     text-align: center;
19   }
20 </style>
```

Line 21, Column 1      Spaces: 2      Vue Component

## ■ Router

- 사용자 요청 URI 에 따라 DOM 을 변경하는 기술이다.
- vue-router 를 사용하여 쉽게 개발할 수 있다.
- Router 를 추가하고 사용하는 방법은 vue-cli 에서 다시 다룬다.

## ■ vue-cli

- 프로젝트 초기 구성이 복잡하여 보일러플레이트에 의존하는 일이 빈번하다. 또한 이러한 의존성에 의해 생산성이 저하된다.
- vue.js 를 사용하여 앱 개발을 할 때, component 와 플러그인 등, 파일 구조와 코드 위치를 개발자가 직접 고려하여 개발 오버헤드가 발생한다.
- 또한 Single file component 를 구성할 때, 직접 빌드 도구를 설정하는 오버헤드 발생한다.
- 이러한 문제점을 해결하고자 vue 공식 제공 개발 툴인, vue-cli 를 도입한다. 이를 통해 개발자는 잡다한 configuration 에 신경 쓰지 않고 앱 로직 개발에만 집중할 수 있다.

## ■ vue-cli 설치 방법

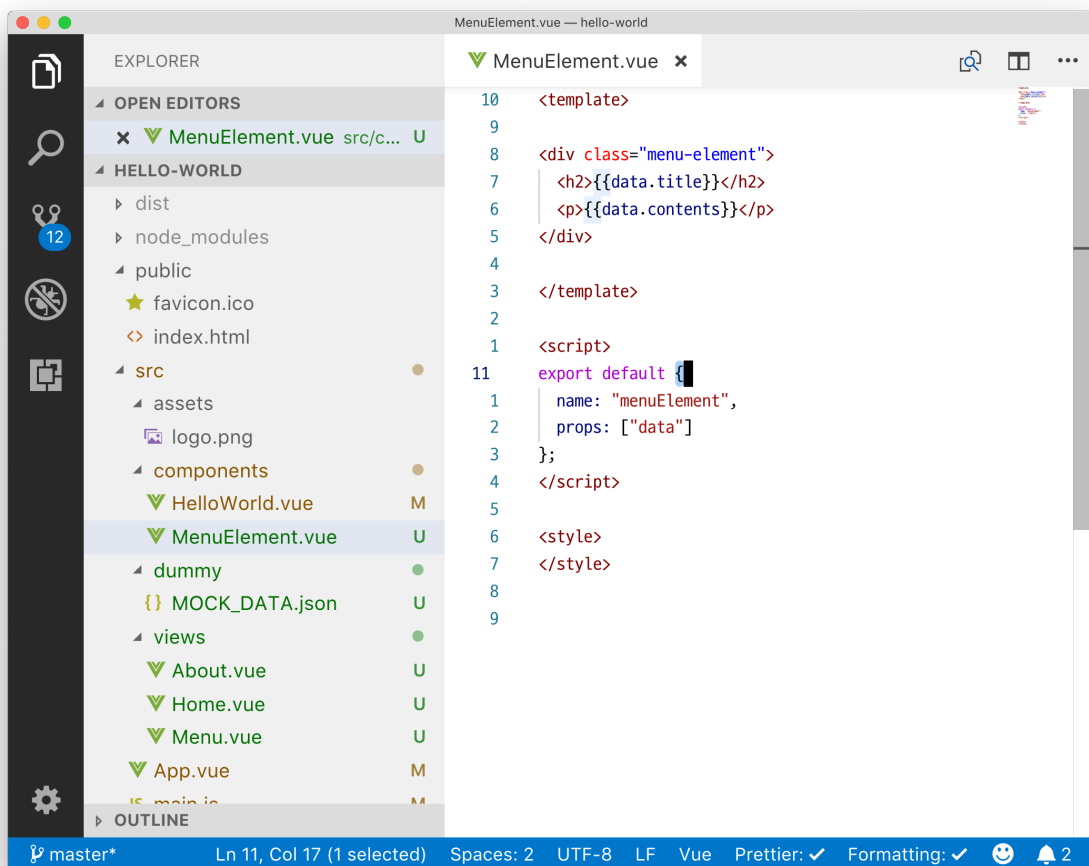
- npm 을 이용하여 vue-cli 를 설치할 수 있다.
- 터미널: **npm install -g @vue/cli**
- vue-cli version 3 이상이 설치되어야 한다.

## ■ vue-cli 사용법

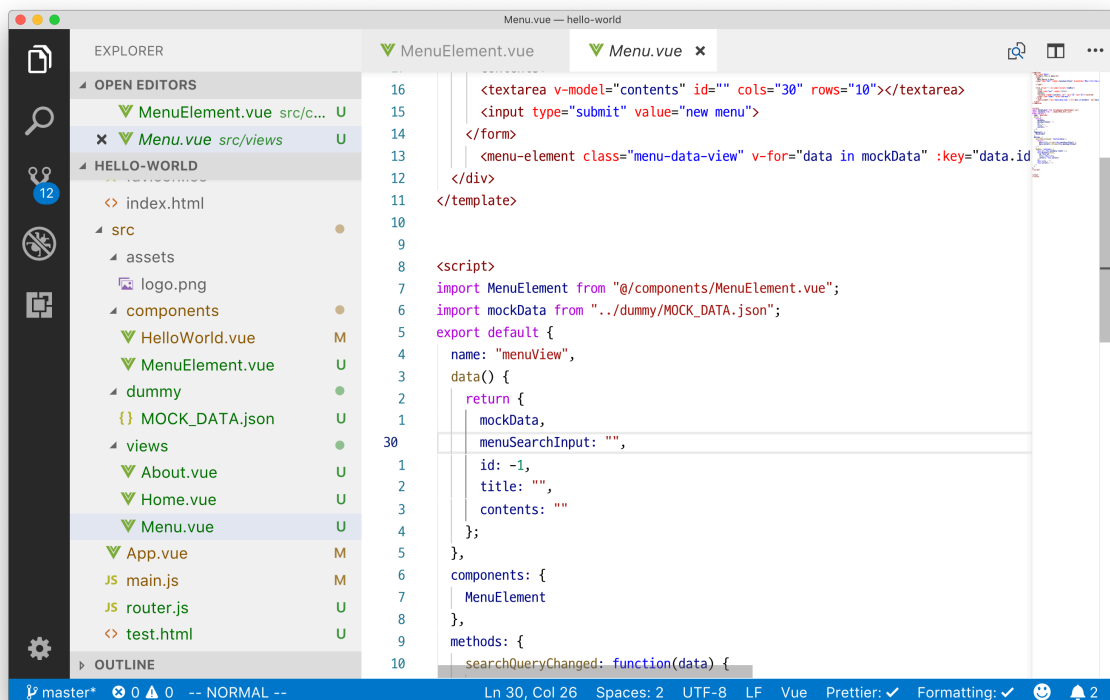
- vue-cli 를 사용하여 프로젝트를 생성하고 vue.js 에 필요한 plugin 들을 받을 수 있다.
- 새로운 프로젝트를 생성하는 방법은 다음과 같다.
  - ✓ 터미널: **vue create project\_name**
  - ✓ 이후 빌드 툴이나, linting 툴 옵션을 선택하는데, 기본 옵션을 사용한다.
  - ✓ Vue.js 개발을 위해 필요한 모든 module 들을 받는 과정이라 오래 걸린다.
  - ✓ 프로젝트 생성이 완료되면, 기본적인 component 들이 정의되어있어 바로 실행할 수 있다.
- 프로젝트를 실행하는 방법은 다음과 같다.



- ✓ 터미널: **npm run serve**
- ✓ 명령을 입력하면, 빌드 툴이 빌드 이후 localhost:8080 에 vue 서버를 돌린다.
  - Node+express 의 HTTP 서버와는 다른 서버이다. Vue 가 돌아가는 모습을 보기 위해서 임시로 돌리는 서버이다.
  - 추후 build 를 통해 사용자가 직접 만든 HTTP 서버에 올릴 수 있다.
- 플러그인 추가
  - ✓ vue-router 와 같은 플러그인을 쉽게 받을 수 있다.
  - ✓ 터미널: **vue add router**
  - ✓ 명령을 입력하면, 프로젝트에 vue-router 가 추가되고, router 를 적용하기 위한 코드가 일부 추가된다.
- component 추가 및 사용
  - ✓ 프로젝트 폴더에 있는 src/components 에 새로운 .vue 파일을 추가하고 코드를 작성하면 된다.



- Template 을 정의하고, script 를 정의한 모습이다.
  - ✓ Template 을 정의할 때, <template>의 자식 node 로 다른 HTML 엘리먼트를 선언해야 한다. (주로 div)
    - 실제로 component 가 렌더링 될 때, template tag 는 사라지고 그 안에 있는 자식 node 만 나오기 때문이다. 이러한 이유로, template 을 제외한 root node 가 필요하다
  - ✓ Script 에서 export 하는 객체의 name property 는 외부 template 에서 해당 component 를 사용할 때 쓰이는 tag 이름이다.
    - camelCase 로 선언하고 사용할 때는 -으로 구분한다.
    - menuElement → <menu-element>



- 외부 component 에서 다른 component 를 사용하는 경우이다.
  - ✓ Script 에 기존에 정의한 component 를 import 하고 export 할 객체의 components property 에 추가한다.

# Lab Assignment

---

## ■ 실습 과제 목표

- Vue.js 를 활용하여 간단한 연락처를 만들어본다.
- 연락처의 기능은 아래와 같다
  - ✓ 메인페이지 – 연락처 리스트를 볼 수 있고, 연락처를 추가 및 검색할 수 있다.

## ■ 프로젝트 파일 구성

- 프로젝트는 vue-cli 를 통하여 생성한다.
- 아래 두 파일만 수정 및 추가한다. 그 외의 수정은 허용하지 않는다.
- 아래 두 파일만 제출한다. (제출 시 제출 폴더 및 압축 양식을 지켜주시길 바랍니다.)
- src/App.vue (수정)
  - ✓ Vue 프로젝트 가장 최상단에 위치하는 component
  - ✓ 메인 페이지를 구성하는 component 로, 연락처 리스트를 볼 수 있고, 연락처를 추가 및 검색할 수 있다.
- src/components/contact.vue (추가)
  - ✓ App.vue 에서 연락처 리스트를 구현할 때 사용되는 component.
  - ✓ 연락처 리스트 전체가 아닌, 리스트에서 하나의 연락처를 렌더링한다.

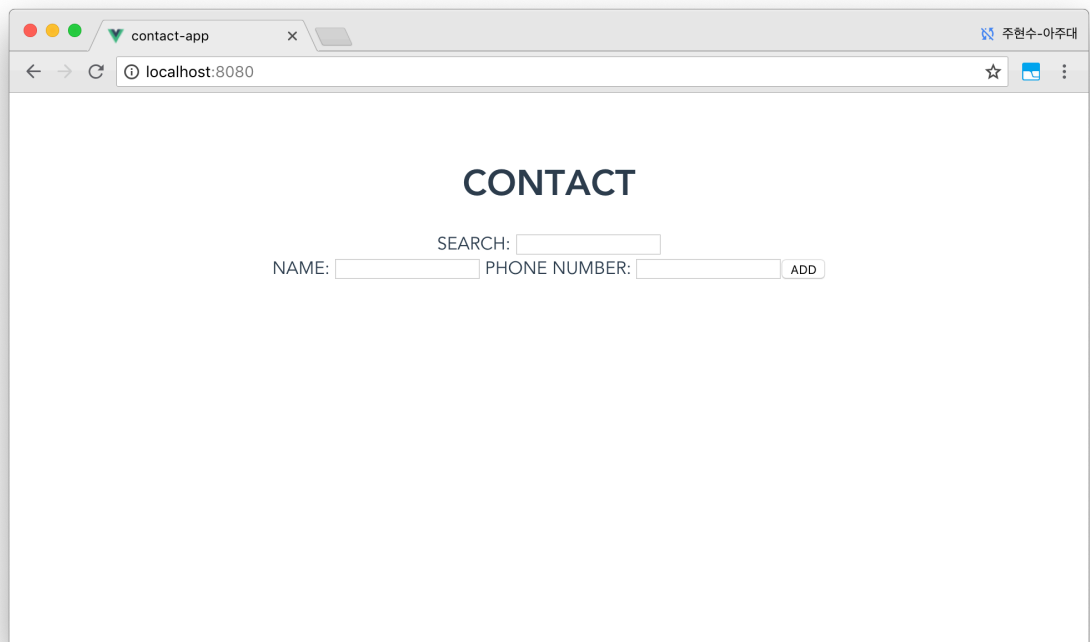
## ■ component 구현

- App
  - ✓ Template
    - 연락처를 검색할 수 있는 input[type=text]. 하나의 input 으로 이름과 전화번호 둘 다 검색 가능.
      - 검색 시, 따로 확인 버튼을 누르지 않고, input 안의 값이 변하면 동적으로 연락처 리스트가 바뀐다.
      - 연락처 리스트가 바뀌는 것은 검색어가 포함되는 모든 연락처만 렌더링한다는 뜻이다.
    - 연락처를 추가할 수 있는 form

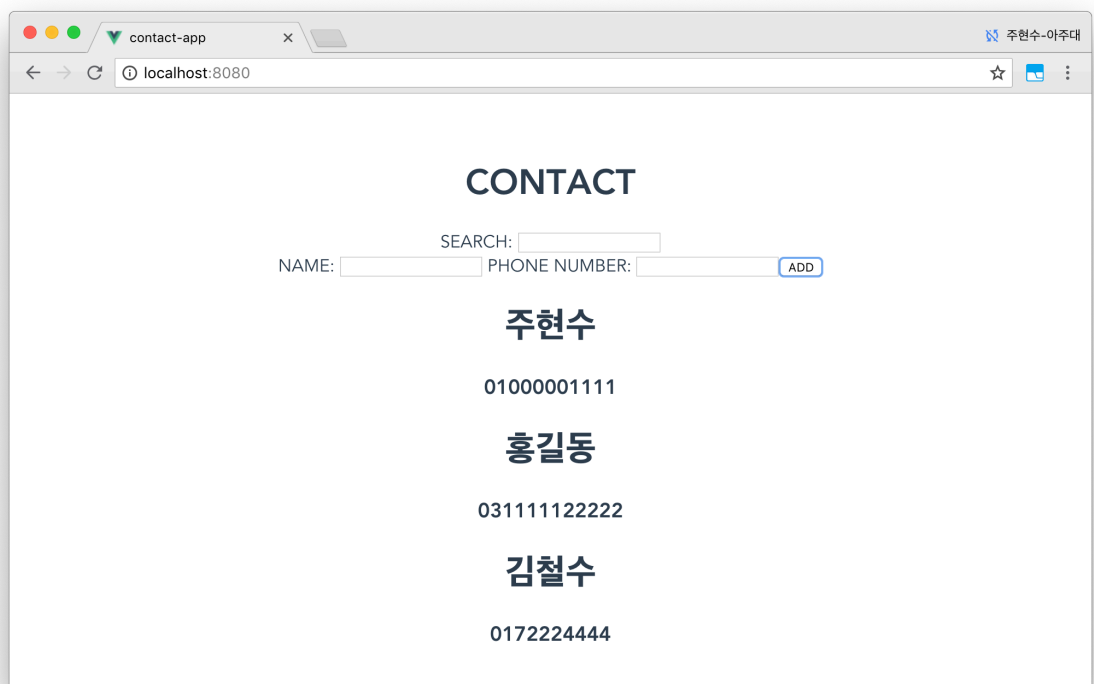
- 이름을 담는 input[type=text], 전화번호를 담는 input[type=text], 제출을 위한 input[type=submit]
- 연락처 리스트를 구현할 contact component
  - contact component 에 하나의 연락처 정보를 전달한다.
  - 연락처의 개수만큼 렌더링 되어야 한다.
  - 검색 input 의 값이 바뀌면 해당 값을 포함하는 연락처만 렌더링한다.
- ✓ Script
  - data 에 연락처 리스트를 저장할 배열 contacts 변수를 추가한다.
    - 배열의 원소는 {id: Number, name: String, phoneNumber: String}으로 이루어져 있다.
    - id 는 0 부터 시작하며, 연락처가 추가될 때마다, 하나씩 증가한다.
  - component 에는 Contact component 를 선언한다.
  - 나머지 App component 를 위해 필요한 함수나 변수는 자유롭게 구성 가능하다.
- Contact
  - ✓ Template
    - 이름을 보여줄 h1
    - 전화번호를 보여줄 h2

## ■ 실행 예시

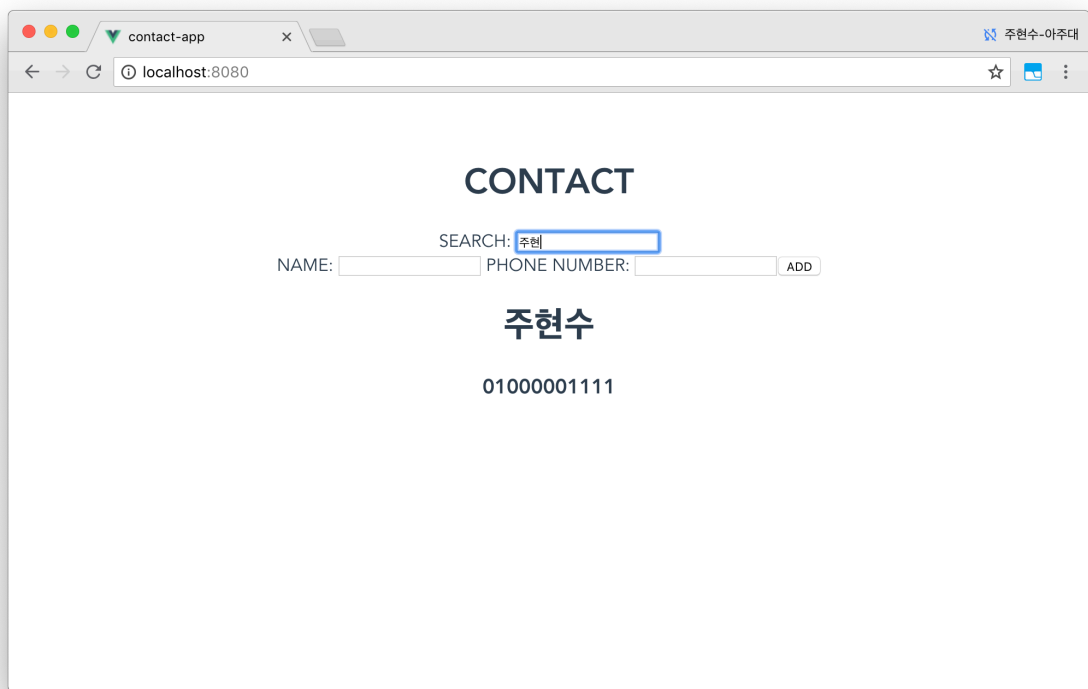
- 처음 실행 시 보이는 화면



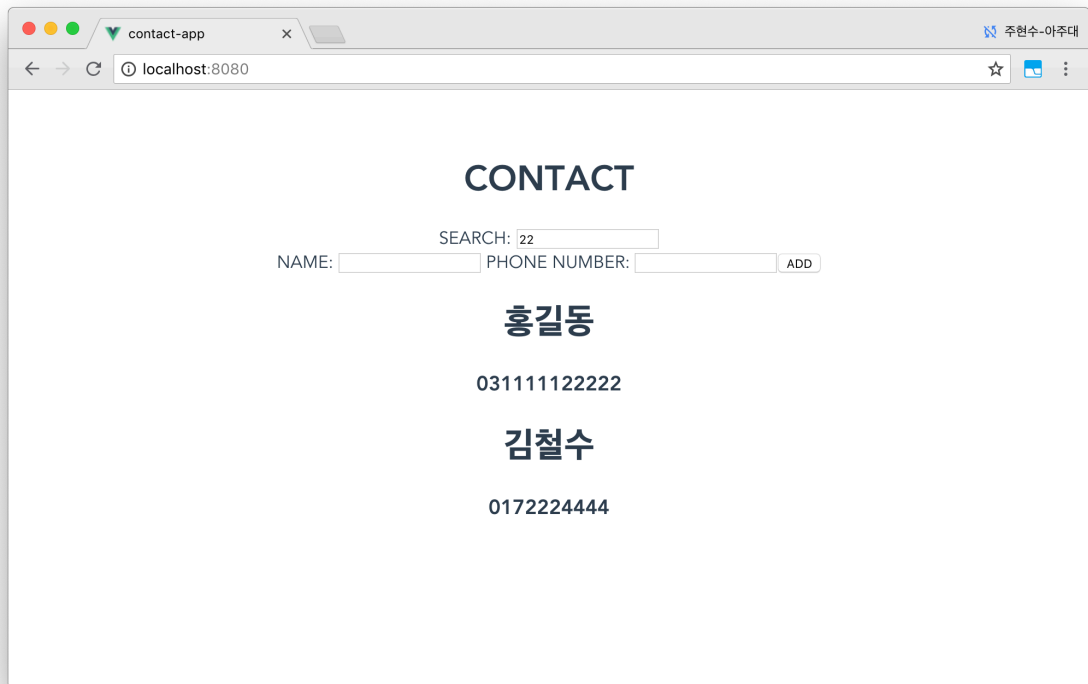
- 연락처가 하나 이상 추가된 화면



- 검색 input 에 "주현"을 입력한 경우



- 검색 input 에 "22"를 입력한 경우



## ■ 제출양식

- 프로젝트 폴더명은 **Lab8-{학번}**으로 한다  
✓ Ex) Lab8-201820000
- 제출하는 압축파일의 이름은 **Lab8-{학번}.zip** 으로 한다  
✓ Ex) Lab8-2018200000.zip
- 명시된 파일명, 폴더명을 반드시 지키기 바랍니다. (감점요인)

## ■ 제출기한

- 금요일 5:59 am 까지 제출: 100% 점수인정
- 금요일 5:59 pm 까지 제출: 70% 점수인정
- 그 이후 제출 시 0%

## ■ 주의사항

- 모든 과제는 본인이 수행한 결과물만 제출
- 처음 Copy 적발 시 해당 과제 0점 처리
- Copy 재적발 시 해당 과제 전체점수 0점 처리