Vue.js 기초 #1

**학습목표**

태그 보이기/보이지 않기 상태 변경

메소드 구현

버튼 비활성화

버튼 클릭 구현

**목차**

[1. v-show v-if 지시자 2](#_Toc68803632)

[1) checkbox3.html 2](#_Toc68803633)

[2) checkbox4.html 4](#_Toc68803634)

[3) v-show v-if 지시자 요약 5](#_Toc68803635)

[2. v-bind 지시자 6](#_Toc68803636)

[1) radio2.html - 버그 6](#_Toc68803637)

[2) radio3.html - 버그 해결 8](#_Toc68803638)

[3) v-bind 지시자 요약 9](#_Toc68803639)

[3. 메소드와 v-on:click 10](#_Toc68803640)

[1) button1.html 10](#_Toc68803641)

[2) button2.html 12](#_Toc68803642)

[4. v-bind:disabled 14](#_Toc68803643)

[1) button3.html 14](#_Toc68803644)

[5. 요약 16](#_Toc68803645)

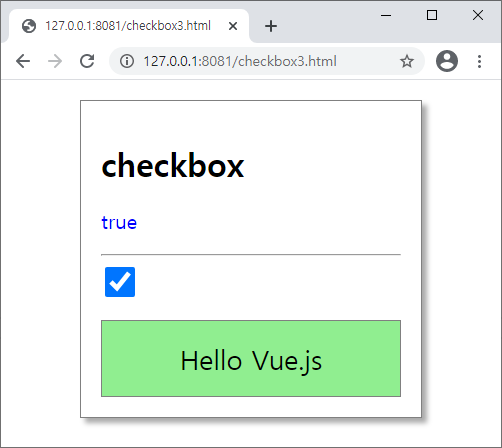
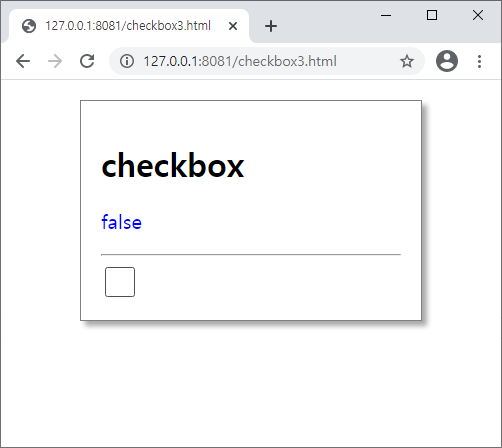
[6. 연습문제 18](#_Toc68803646)

[1) empty1.html 18](#_Toc68803647)

# v-show v-if 지시자

## checkbox3.html

|  |
| --- |
| <html>  <head>    <meta charset="utf-8" />    <script src="https://unpkg.com/vue@next"></script>    <style>      div#app { margin: 20px auto; padding: 20px; width: 300px;            border: 1px solid gray; box-shadow: 5px 5px 5px #aaa; }      p { font-size: 15pt; color: blue; }      input { width: 30px; height: 30px;  }      div#hello { margin-top: 20px; padding: 20px;        background-color: lightGreen; border: 1px solid gray;        text-align: center; font-size: 20pt; }    </style>  </head>  <body>    <div id="app">        <h1>checkbox</h1>        <p>{{ value }}</p>        <hr />        <input type="checkbox" v-model="value" />        <div id="hello" v-show="value">Hello Vue.js</div>    </div>    <script type="text/javascript">      let app = {        data() {          return {            value: false          };        }      };      Vue.createApp(app).mount("#app");     </script>  </body>  </html> |

<div id="hello" v-show="value">

이 div 태그는 앱의 value 속성 값이 true 이면 보이는 상태가 되고, false 이면 보이지 않는 상태가 된다.

체크 박스를 끄면, 앱의 value 속성 값이 false 가 되고, 이 div 태그는 보이지 않는 상태가 된다.

체크 박스를 켜면, 앱의 value 속성 값이 true 가 되고, 이 div 태그는 보이는 상태가 된다.

### CSS 서식

background-color: lightGreen;

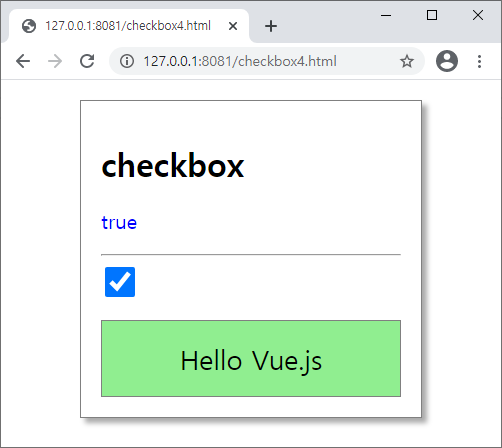
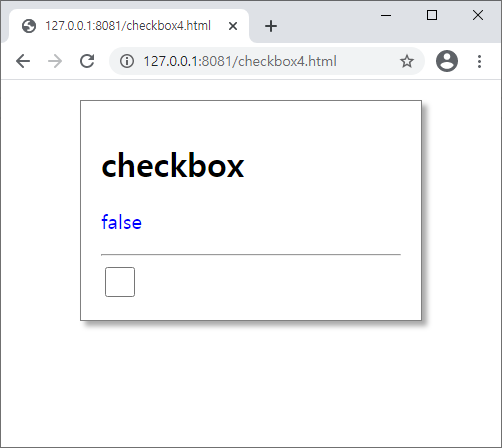
배경색 밝은 녹색

text-align: center;

텍스트 가로방향으로 가운데 정렬

## checkbox4.html

|  |
| --- |
| <html>  <head>    <meta charset="utf-8" />    <script src="https://unpkg.com/vue@next"></script>    <style>      div#app { margin: 20px auto; padding: 20px; width: 300px;            border: 1px solid gray; box-shadow: 5px 5px 5px #aaa; }      p { font-size: 15pt; color: blue; }      input { width: 30px; height: 30px;  }      div#hello { margin-top: 20px; padding: 20px;        background-color: lightGreen; border: 1px solid gray;        text-align: center; font-size: 20pt; }    </style>  </head>  <body>    <div id="app">        <h1>checkbox</h1>        <p>{{ value }}</p>        <hr />        <input type="checkbox" v-model="value" />        <div id="hello" v-if="value">Hello Vue.js</div>    </div>    <script type="text/javascript">      let app = {        data() {          return {          value: false          };        }      };      Vue.createApp(app).mount("#app");     </script>  </body>  </html> |

<div id="hello" v-if="value">

이 div 태그는 앱의 value 속성 값이 false 이면 제거되고, true 이면 다시 생성된다.

v-show 지시자는 태그의 보이는 상태만 변경한다.

v-if 지시자는 태그를 제거하고 다시 생성한다.

따라서 보이고 보이지 않는 상태 변경이 빈번하다면, v-show 지시자를 사용하는 것이 좋다.

반대로 보이고 보이지 않는 상태 변경이 아주 드물다면,

화면에 보이지 않을 때 태그가 제거되어 메모리를 차지하지 않는 v-if 지시자가 좋다.

## v-show v-if 지시자 요약

### v-show 지시자

v-show 지시자의 값이

true 에 해당하는 값이면 그 태그는 보이는 상태가 되고,

false 에 해당하는 값이면 그 태그는 보이지 않는 상태가 된다.

보이고 보이지 않는 상태 변경이 빈번하다면, v-show 지시자를 사용하는 것이 좋다.

### v-if 지시자

v-if 지시자의 값이

true 에 해당하는 값이면 그 태그는 다시 생성되어 보이는 상태가 되고,

false 에 해당하는 값이면 그 태그는 메모리에서 제거되어 보이지 않는 상태가 된다.

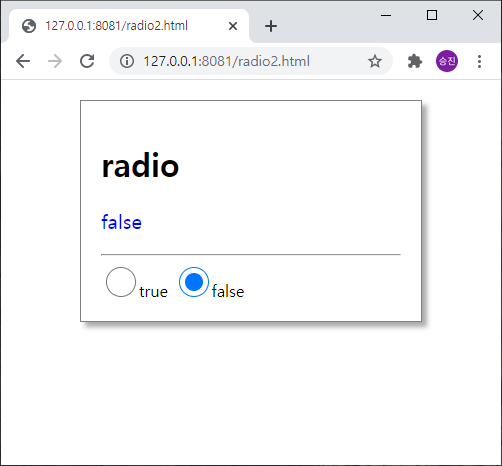
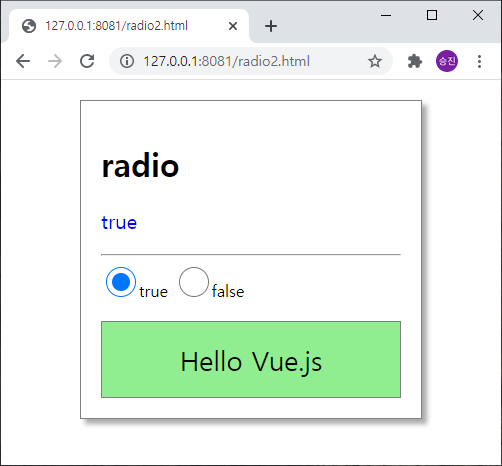
보이고 보이지 않는 상태 변경이 아주 드물다면,

화면에 보이지 않을 때 태그가 제거되어 메모리를 차지하지 않는 v-if 지시자가 좋다.

# v-bind 지시자

## radio2.html - 버그

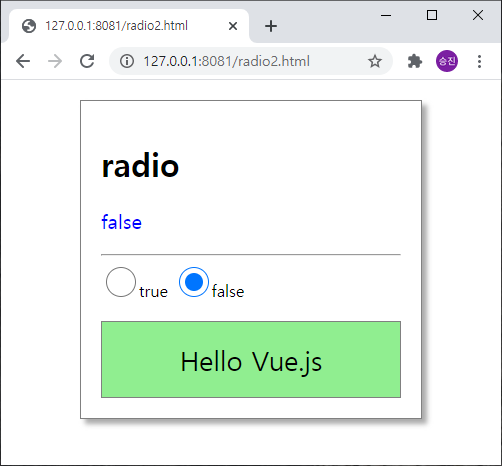
|  |
| --- |
| <html>  <head>    <meta charset="utf-8" />    <script src="https://unpkg.com/vue@next"></script>    <style>      div#app { margin: 20px auto; padding: 20px; width: 300px;            border: 1px solid gray; box-shadow: 5px 5px 5px #aaa; }      p { font-size: 15pt; color: blue; }      input { width: 30px; height: 30px;  }      div#hello { margin-top: 20px; padding: 20px;        background-color: lightGreen; border: 1px solid gray;        text-align: center; font-size: 20pt; }    </style>    </style>  </head>  <body>    <div id="app">        <h1>radio</h1>        <p>{{ visible }}</p>        <hr />        <label>          <input type="radio" value="true" v-model="visible" />true        </label>        <label>          <input type="radio" value="false" v-model="visible" />false        </label>        <div id="hello" v-if="visible">Hello Vue.js</div>    </div>    <script type="text/javascript">      let app = {        data() {          return {          visible: false          };        }      };      Vue.createApp(app).mount("#app");     </script>  </body>  </html> |

처음 실행하면 false 라디오 버튼이 선택되어있고 녹색 div 태그는 보이지 않는다.

true 라디오 버튼을 선택하면 녹색 div 태그가 보인다.

정상 작동하는 것 같은데...



다시 false 라디오 버튼을 선택해도, 녹색 div 버튼의 상태가 변경되지 않는다.

data: {

  visible: false

}

<div id="hello" v-if="visible">

처음에 녹색 div 태그가 보이지 않았던 이유는, visible 앱 속성의 값이 false 이기 때문이다.

<input type="radio" value="true" v-model="visible" />

이 라디오 버튼을 클릭했을 때, 앱의 visible 속성에 대입되는 값은 "true" 문자열이다.

value="true" 여기서 value 애트리뷰트는, Vue 지시자가 아니고 HTML 애트리뷰트이다.

HTML tag 애트리뷰트의 값은 javascript 표현식으로 해석되지 않고, 문자열로 해석된다.

Vue 지시자의 이름은 v- 으로 시작한다.

앱의 visible 속성의 값이 "true" 문자열이면, v-if="visible" 이 부분의 조건식은 true로 해석된다.

javascript 조건식에서 null, undefined, NaN, 0 이 아닌 값은 true로 해석되기 때문이다.

그래서 녹색 div 태그가 보이는 상태가 된다.

<input type="radio" value="false" v-model="visible" />

이 라디오 버튼을 클릭했을 때, 앱의 visible 속성에 대입되는 값은 "false" 문자열이다.

앱의 visible 속성의 값이 "false" 문자열이면, v-if="visible" 이 부분의 조건식은 true로 해석된다.

javascript 조건식에서 null, undefined, NaN, 0, “” 이 아닌 값은 true로 해석되기 때문이다.

그래서 녹색 div 태그가 보이는 상태가 된다.

## radio3.html - 버그 해결

|  |
| --- |
| <html>  <head>    <meta charset="utf-8" />    <script src="https://unpkg.com/vue@next"></script>    <style>      div#app { margin: 20px auto; padding: 20px; width: 300px;            border: 1px solid gray; box-shadow: 5px 5px 5px #aaa; }      p { font-size: 15pt; color: blue; }      input { width: 30px; height: 30px;  }      div#hello { margin-top: 20px; padding: 20px;        background-color: lightGreen; border: 1px solid gray;        text-align: center; font-size: 20pt; }    </style>  </head>  <body>    <div id="app">        <h1>radio</h1>        <p>{{ visible }}</p>        <hr />        <label>          <input type="radio" v-bind:value="true" v-model="visible" />true        </label>        <label>          <input type="radio" v-bind:value="false" v-model="visible" />false        </label>        <div id="hello" v-if="visible">Hello Vue.js</div>    </div>    <script type="text/javascript">      let app = {        data() {          return {          visible: false          };        }      };      Vue.createApp(app).mount("#app");     </script>  </body>  </html> |

HTML 애트리뷰트의 값은 문자열로 해석된다.

그런데, HTML 애트리뷰트 앞에 v-bind: 지시자를 붙이면, 그 애트리뷰트의 값은

javascript 표현식으로 해석된다.

(따옴표 내부만 javascript 표현식으로 해석된다)

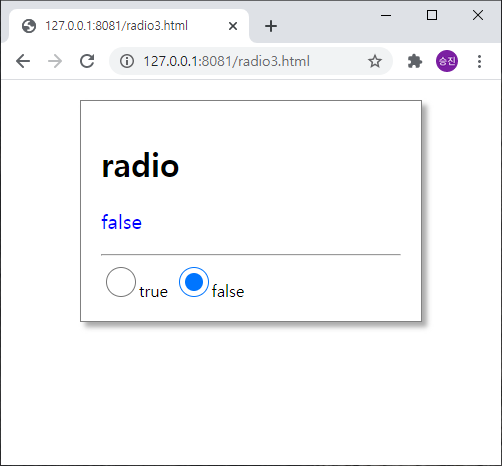
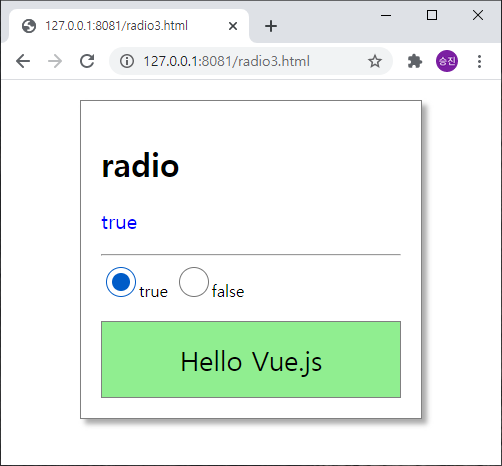
value="true"

value 는 HTML attribute 이므로 이 애트리뷰트의 값은 "true" 문자열로 해석된다.

v-bind:value="true"

v-bind: 지시자가 붙어있으므로, 이 애트리뷰트의 값은 javascript 코드에서 true 로 핵석된다.

즉 이 애트리뷰트의 값은 boolean 타입의 true 이다.

## v-bind 지시자 요약

Vue 지시자의 이름은 v- 으로 시작한다.

HTML tag 애트리뷰트는 v- 으로 시작하지 않는다.

HTML 애트리뷰트의 값은 javascript 표현식으로 해석되지 않고, 문자열로 해석된다.

예:

value="true" value 값은 문자열 타입 "true" 이다.

value='true' value 값은 문자열 타입 'true' 이다.

HTML 애트리뷰트 이름에 v-bind: 지시자를 붙여주면, 그 애트리뷰트 값은 javascript 표현식으로 해석된다.

예:

v-bind:value="true" value 값은 boolean 타입 true 이다.

v-bind:value='true' value 값은 boolean 타입 true 이다.

v-bind:value="'true'" value 값은 문자열 타입 'true' 이다.

v-bind: 대신 : 만 붙여도 된다.

v-bind:value="true"

:value="true"

# 메소드와 v-on:click

## button1.html

|  |
| --- |
| <html>  <head>    <meta charset="utf-8" />    <script src="https://unpkg.com/vue@next"></script>    <style>      div#app { margin: 20px auto; padding: 20px; width: 300px;            border: 1px solid gray; box-shadow: 5px 5px 5px #aaa; }      p { font-size: 15pt; color: blue; }      button { padding: 0.4em 3em; margin-right: 5px; }      div#hello { margin-top: 20px; padding: 20px;        background-color: lightGreen; border: 1px solid gray;        text-align: center; font-size: 20pt; }    </style>  </head>  <body>    <div id="app">        <h1>button</h1>        <p>{{ visible }}</p>        <hr />        <button type="button" v-on:click="show">보이기</button>        <button type="button" v-on:click="hide">감추기</button>        <div id="hello" v-show="visible">Hello Vue.js</div>    </div>    <script type="text/javascript">      let app = {        data() {          return {          visible: false         };  },        methods: {          show() {            this.visible = true;          },          hide() {            this.visible = false;          }        }      };      Vue.createApp(app).mount("#app");     </script>  </body>  </html> |

methods: {

  show() {

    this.visible = true;

  },

  hide() {

    this.visible = false;

  }

}

Vue 앱 객체의 메소드를 구현함.

show 메소드와 hide 메소드를 구현함.

this.visible = true;

Vue 앱 객체의 visible 속성에 true를 대입한다.

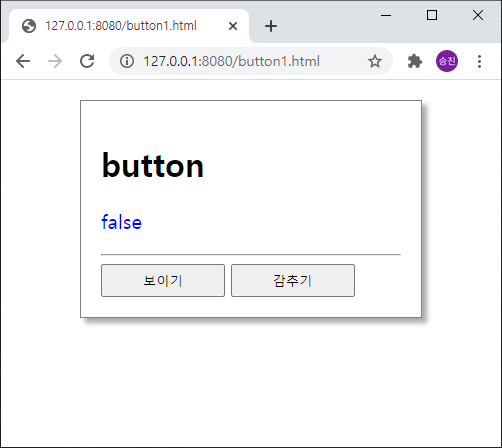
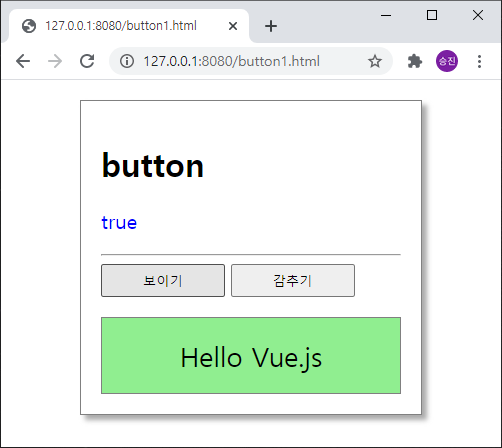
javascript 메소드 본문에서 this를 생략할 수 없음에 주의하자.

<button type="button" v-on:click="show">보이기</button>

이 버튼이 클릭되면 show 메소드가 호출된다.

<button type="button" v-on:click="hide">감추기</button>

이 버튼이 클릭되면 hide 메소드가 호출된다.

v-on:click="show"

@click="show"

v-on: 대신 @ 문자를 사용해도 된다.

## button2.html

|  |
| --- |
| <html>  <head>    <meta charset="utf-8" />    <script src="https://unpkg.com/vue@next"></script>    <style>      div#app { margin: 20px auto; padding: 20px; width: 300px;            border: 1px solid gray; box-shadow: 5px 5px 5px #aaa; }      p { font-size: 15pt; color: blue; }      button { padding: 0.4em 3em; margin-right: 5px; }      div#hello { margin-top: 20px; padding: 20px;        background-color: lightGreen; border: 1px solid gray;        text-align: center; font-size: 20pt; }    </style>  </head>  <body>    <div id="app">        <h1>button</h1>        <p>{{ visible }}</p>        <hr />        <button type="button" v-on:click="show" v-show="!visible">보이기</button>        <button type="button" v-on:click="hide" v-show="visible">감추기</button>        <div id="hello" v-show="visible">Hello Vue.js</div>    </div>    <script type="text/javascript">      let app = {        data() {          return {          visible: false         };  },  methods: {          show() {            this.visible = true;          },          hide() {            this.visible = false;          }        }      };      Vue.createApp(app).mount("#app");     </script>  </body>  </html> |

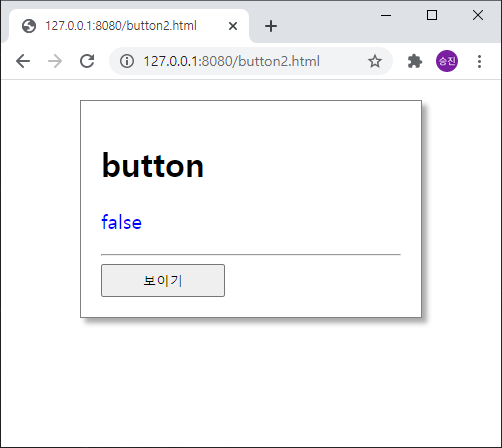
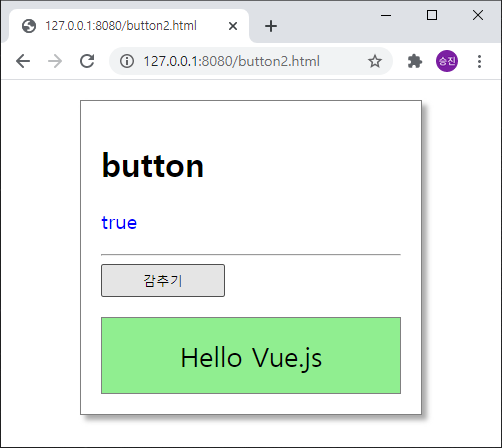
<button type="button" v-on:click="hide" v-show="visible">감추기</button>

앱의 visible 속성 값이 true 일 때만 이 버튼이 화면에 보인다.

<button type="button" v-on:click="show" v-show="!visible">보이기</button>

앱의 visible 속성 값이 false 일 때만 이 버튼이 화면에 보인다.

느낌표 문자는 javascript의 not 연산자이다.

보이지 않는 상태에서 감추기 버튼은 불필요하고, 보이는 상태에서 보이기 버튼은 불필요하다.

불필요한 버튼이 화면에 보이지 않는 것이 좋다.

# v-bind:disabled

## button3.html

|  |
| --- |
| <html>  <head>    <meta charset="utf-8" />    <script src="https://unpkg.com/vue@next"></script>    <style>      div#app { margin: 20px auto; padding: 20px; width: 300px;            border: 1px solid gray; box-shadow: 5px 5px 5px #aaa; }      p { font-size: 15pt; color: blue; }      button { padding: 0.4em 3em; margin-right: 5px; }      div#hello { margin-top: 20px; padding: 20px;        background-color: lightGreen; border: 1px solid gray;        text-align: center; font-size: 20pt; }    </style>  </head>  <body>    <div id="app">        <h1>button</h1>        <p>{{ visible }}</p>        <hr />        <button type="button" v-on:click="show" v-bind:disabled="visible">보이기</button>        <button type="button" v-on:click="hide" v-bind:disabled="!visible">감추기</button>        <div id="hello" v-show="visible">Hello Vue.js</div>    </div>    <script type="text/javascript">      let app = {        data() {          return {          visible: false  };        },        methods: {          show() {            this.visible = true;          },          hide() {            this.visible = false;          }        }      };      Vue.createApp(app).mount("#app");     </script>  </body>  </html> |

<button type="button" v-on:click="show" v-bind:disabled="visible">보이기</button>

앱의 visible 속성 값이 true 일 때 이 버튼은 비활성화 상태가 된다.

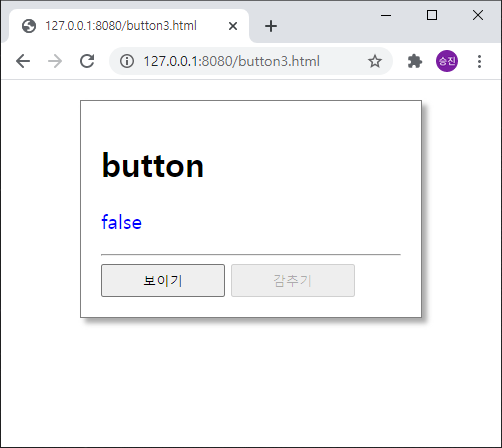
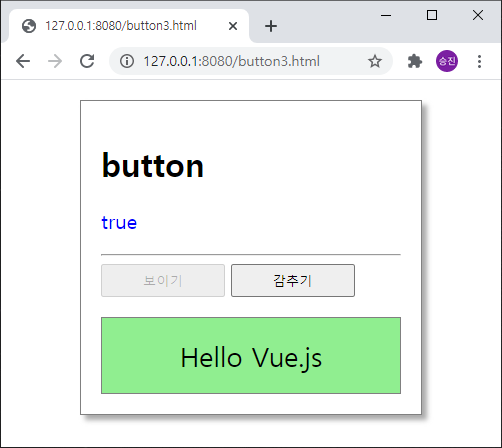
<button type="button" v-on:click="hide" v-bind:disabled="!visible">감추기</button>

앱의 visible 속성 값이 false 일 때 이 버튼은 비활성화 상태가 된다.

disabled 는 표준 HTML 애트리뷰트이다.

표준 HTML 애트리뷰트 앞에 v-bind: 접두어를 붙여야, 그 값이 javascript 표현식으로 인식된다.

v-bind: 접두어를 붙이지 않으면, 그 값은 평범한 문자열로 인식된다.

# 요약

### v-show="조건식"

조건식이 true 이면 이 태그가 보이고, false 이면 보이지 않는다.

보이고 보이지 않는 상태 변경이 빈번하다면, v-show 지시자를 사용하는 것이 좋다.

v-show="i >= 3"

i 값이 3 이상일 때만 이 태그가 보인다.

### v-if="조건식"

조건식이 false 이면 이 태그가 메모리에서 제거되고, true 이면 다시 생성된다.

보이고 보이지 않는 상태 변경이 아주 드물다면,

화면에 보이지 않을 때 태그가 제거되어 메모리를 차지하지 않는 v-if 지시자가 좋다.

v-if="role == 'admin'"

role 속성 값이 'admin' 문자열 일 때만 이 태그가 생성된다.

### v-bind:애트리뷰트="값"

HTML 애트리뷰트의 값은 문자열로 해석된다.

그런데, HTML 애트리뷰트 앞에 v-bind: 접두어를 붙이면, 그 값은 javascript 표현식으로 해석된다.

v-bind: 대신 : 만 붙여도 된다.

value="3"

value 값은 "3" 문자열이다.

v-bind:value="3"

value 값은 숫자 3 이다.

:value="3"

value 값은 숫자 3 이다.

value="3 + 2"

value 값은 "3 + 2" 문자열이다.

v-bind:value="3 + 2"

value 값은 숫자 5 이다.

value="title"

value 값은 "title" 문자열이다.

v-bind:value="title"

value 값은 title 속성의 값 이다.

disabled="visible"

disabled 값은 "visible" 문자열이다.

v-bind:disabled="visible"

disabled 값은 visible 속성의 값 이다.

### v-bind:disabled="조건식"

조건식의 값이 true 이면, 이 태그는 비활성화된 상태가 되고,

false 이면, 이 태그는 활성화된 상태가 된다.

### 메소드 정의

methods: {

메소드이름(파라미터목록) {

메소드 본문

},

메소드이름(파라미터목록) {

메소드 본문

}

}

### v-on:click="메소드이름"

이 태그가 클릭되면 메소드가 호출된다.

v-on:click="메소드이름(파라미터값)"

이 태그가 클릭되면 메소드가 호출되고, 이때 파라미터 값이 전달된다.

V-on:click="메소드이름" 대신 @click="메소드이름" 해도 된다.

# 연습문제

## empty1.html

|  |
| --- |
| <html>  <head>    <meta charset="utf-8" />    <script src="https://unpkg.com/vue@next"></script>    <style>      div#app { padding: 30px; margin: 30px auto; width: 350px;        border: 1px solid #ccc; box-shadow: 3px 3px 3px #aaa; }      div#app > div { margin-top: 30px; }      label:nth-child(1) { display: inline-block; width: 70px; text-align: right;        margin-right: 5px; }      label:nth-child(2) { margin-right: 20px; }      input[type=text] { padding: 6px; width: 200px; }      div.error { margin-top: 5px; margin-left: 70px; color: red; }    </style>  </head>  <body>    <div id="app">      <div>        <label>이메일:</label>        <input type="text" v-model="email" />      </div>      <div>        <label>전화번호:</label>        <input type="text" v-model="phone" />      </div>      <div>        <label>성별:</label>        <label><input type="radio" v-model="sex" value="남" />남자</label>        <label><input type="radio" v-model="sex" value="여" />여자</label>      </div>    <script type="text/javascript">      let app = {        data() {          return {            email: '',            phone: '',            sex: ''          };        }      };      Vue.createApp(app).mount("#app");     </script>  </body>  </html> |

위 소스코드에 다음과 같은 기능을 구현하시오.

아무 것도 입력하지 않으면 에러 메시지가 보이지만,

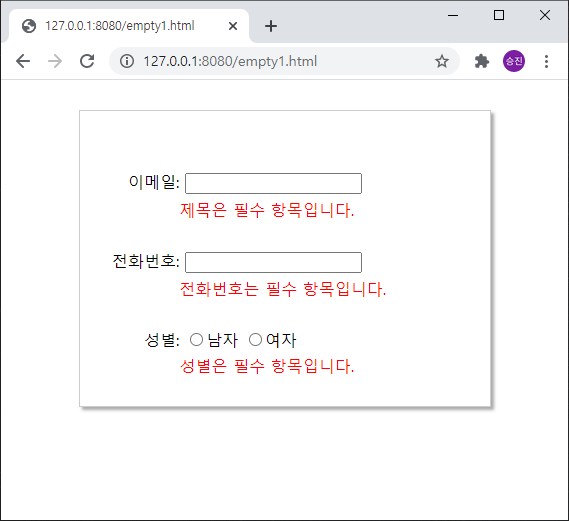
내용을 입력하자 마자 에러 메시자가 보이지 않게 된다.

### 힌트

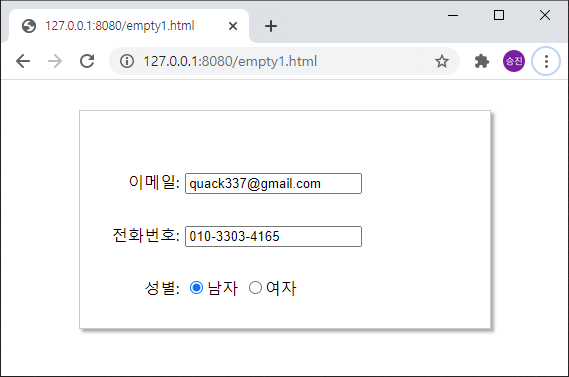
javascript 조건식에서 null 이나 빈 문자열은 false 로 해석되고, 내용이 있는 문자열은 true 로 해석된다.

앱의 속성 값이 빈 문자열일 때만 에러 메시지가 보이고,

빈 문자열이 아닐 때는 에러 메시지가 보이지 않도록 구현하라.



아무 것도 입력하지 않으면 에러 메시지가 보인다.



내용을 입력하자 마자 에러 메시자가 보이지 않게 된다.