원쌤의 Vue.js 퀵스타트

5. 이벤트 처리

1. 이벤트 개요

- Vue 이벤트
  - 동적인 UI 구성을 위해 필수
    - 계산된 속성, 메서드, 관찰 속성만으로는 부족함
  - HTML, 자바스크립트 이벤트 준용
    - HTML, 자바스크립트 이벤트를 미리 익혔다면 쉽게 학습할 수 있음
  - HTML, 자바스크립트 이벤트 학습
    - https://developer.mozilla.org/ko/docs/Web/Reference/Events
    - https://www.w3schools.com/tags/ref_eventattributes.asp

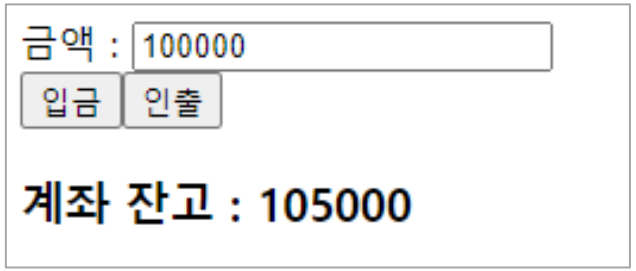
2. 인라인 이벤트 처리

❖Vue의 이벤트 처리 방법

- v-on 디렉티브

```
v-on:[이벤트이름]="표현식"
```

- 예제 05-01 : 인라인 이벤트 사용
  - 인라인 이벤트
    - 표현식에 코드 작성




```
<div id="app">
  금액 : <input type="text" v-model.number="amount" /><br />
  <button v-on:click="balance += parseInt(amount)">입금</button>
  <button v-on:click="balance -= parseInt(amount)">인출</button>
  <br />
  <h3>계좌 잔고 : {{balance}}</h3>
</div>
<script type="text/javascript" src="https://unpkg.com/vue"></script>
<script type="text/javascript">
  var vm = Vue.createApp({
    name: "App",
    data() {
      return { amount: 0, balance: 0 };
    },
  }).mount("#app");
</script>
```

2. 인라인 이벤트 처리

❖v-on 디렉티브 축약형

- v-on:click ----〉 @click

❖인라인 이벤트 표현식에서 이벤트 아규먼트 전달

- $event

```
@click="test($event)"
```

❖ 인라인 이벤트의 단점

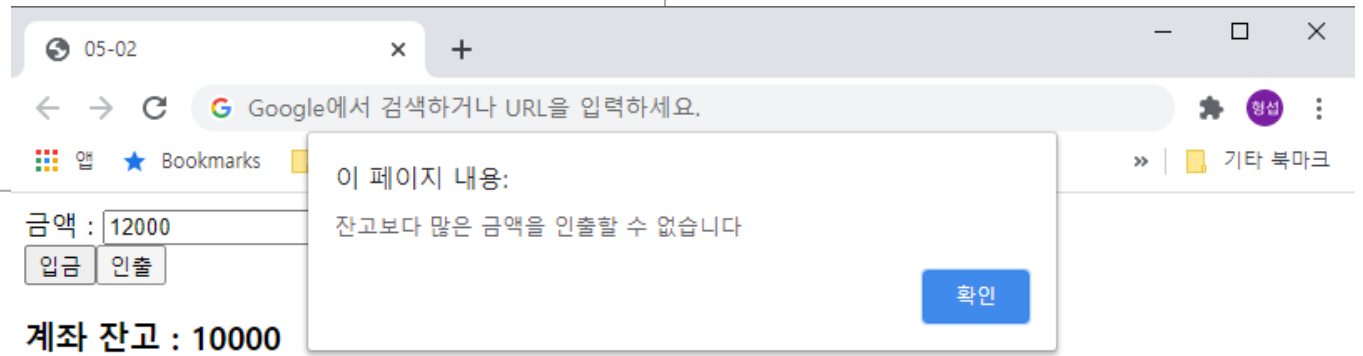
- 복잡한 코드를 작성하기 힘들다.
- 해결책 --〉 이벤트 핸들러 메서드를 지정

3. 이벤트 핸들러 메서드

❖이벤트 핸들러 메서드 지정

- 템플릿에서
  - @click="deposit"
- 스크립트 코드에서

```
methods: {
   deposit() {
      let amt = parseInt(this.amount);
      if (amt <= 0) {
         alert("0보다 큰 값을 예금해야 합니다");
      } else {
         this.balance += amt;
      }
   },
   ...
}
```



3. 이벤트 핸들러 메서드

❖ 다음과 같은 간단한 기능은 이벤트 핸들러 메서드로 만들 필요가 없음

```
methods: {
  changeName(e) {
    this.name = e.target.value;
  },
},
```

- 이와 같은 경우는 인라인 이벤트로

```
<input id="a" type="text" :value="name" @input="this.name = $event.target.value">
```

```
<input id="a" type="text" :value="name"
    @input="(e) => this.name = e.target.value ">
```

- 화살표 함수 대신에 전통적인 함수를 사용해서는 안됨

```
<input id="a" type="text" :value="name"
   @input="function(e) { this.name = e.target.value }">
```

이 경우는 this가 전역객체

- 하나의 이벤트에 여러개 이벤트 핸들러 메서드 연결하기

```
<button @click="change1($event), change2($event)">두 개의 핸들러 실행</button>
```

4. 이벤트 객체

❖이벤트 객체

- 이벤트 핸들러 메서드의 첫번째 인자값 e
- 이벤트의 표현식의 $event
- Vue의 이벤트 객체
  - 표준 HTML DOM 이벤트 모델을 그대로 따르면서 추가적인 속성을 제공
  - 따라서 HTML, 자바스크립트 이벤트 객체의 정보를 대부분 사용할 수 있음

4. 이벤트 객체

❖ 이벤트 객체의 속성

▪ 주요 공통 속성

속성명	설명
target	이벤트가 발생한 HTML 요소를 리턴함.
currentTarget	이벤트 리스너가 이벤트를 발생시키는 HTML 요소를 리턴함.
path	배열값. 이벤트 발생 HTML 요소로부터 document, window 객체로까지 거슬러 올라가는 경로를 나타냄.
bubbles	현재의 이벤트가 버블링을 일으키는 이벤트인지 여부를 리턴함.
cancelable	기본 이벤트의 실행 취소할 수 있는지 여부를 리턴함
defaultPrevented	기본 이벤트의 실행이 금지되었는지 여부를 나타냄.
eventPhase	이벤트 흐름의 단계를 나타냄. 1 : 포착(CAPTURING_PHASE) 2 : 이벤트 발생(AT_TARGET) 3 : 버블링(BUBBLING_PHASE)
srcElement	IE에서 사용되던 속성으로 target과 동일한 속성.

표 05-01 이벤트 객체의 주요 공통 속성

4. 이벤트 객체

- 키보드 이벤트 관련 속성

속성명	설명
altKey	ALT 키가 눌러졌는지 여부를 나타냄(true/false).
shiftKey	SHIFT 키가 눌러졌는지 여부를 나타냄(true/false).
ctrlKey	CTRL 키가 눌러졌는지 여부를 나타냄(true/false).
metakey	메타키가 눌러졌는지 여부를 나타냄. 윈도우에서는 Window Key, macOS에서는 Command Key이다.
key	이벤트에 의해 나타나는 키의 값을 리턴함. 대소문자 구분함
code	이벤트를 발생시킨 키의 코드값을 리턴함. ex) a를 눌렀을 때 "KeyA"를 리턴함. ex) Shift 키를 눌렀을 때 "Shift"를 리턴함.
keyCode	이벤트를 발생시킨 키보드의 고유 키코드 ex) a, A는 65를 리턴함(대소문자 구분하지 않음)
charCode	keypress 이벤트가 발생될 때 Unicode 캐릭터 코드를 리턴함.

표 05-02 키보드 이벤트 관련 속성

4. 이벤트 객체

▪ 마우스 이벤트 관련 속성

속성명	설명
altkey, shiftKey, ctrlKey, metaKey	키보드 이벤트 관련 속성 참조
button	이벤트를 발생시킨 마우스 버튼 0 : 마우스 왼쪽 버튼 1 : 마우스 휠 2 : 마우스 오른쪽 버튼
buttons	마우스 이벤트가 발생한 후에 눌러져 있는 마우스 버튼의 값을 리턴함. 아래 값의 조합으로 이루어짐 1 : 마우스 왼쪽 버튼 2 : 마우스 오른쪽 버튼 4 : 마우스 휠 8 : 4번째 마우스 버튼 16 : 5번째 마우스 버튼 ex) 마우스의 오른쪽 버튼, 휠을 누르고 있는 상태에서 왼쪽 버튼을 클릭할 경우 이 값은 6을 리턴함
clientX, clientY	마우스 이벤트가 일어났을 때의 뷰포트(ViewPort) 영역상의 좌표. 이 좌표는 스크롤바를 내리더라도 좌푯값에 영향을 받지 않음
layerX, layerY	마우스 이벤트가 발생한 HTML 요소 영역상에서의 좌표(IE이외의 브라우저 사용)
offsetX, offsetY	마우스 이벤트가 발생한 HTML 요소 영역상에서의 좌표(IE 브라우저 사용)
pageX, pageY	마우스 이벤트가 일어났을 때의 HTML 문서(Document) 영역상의 좌표
screenX, screenY	마우스 이벤트가 일어났을 때의 모니터 화면(Screen) 영역상의 좌표

표 05-03 마우스 이벤트 관련 속성

4. 이벤트 객체

- 주요 이벤트 객체의 메서드

속성명	설명
preventDefault()	기본 이벤트의 자동 실행을 금지시킴
stopPropagation()	이벤트의 전파를 중단시킴

❖ e.target.value 의 의미

- 표 05-01에서 target 속성
  - 이벤트가 발생한 요소를 가리킴
- "이벤트가 발생한 요소의 value 속성"

5. 기본 이벤트

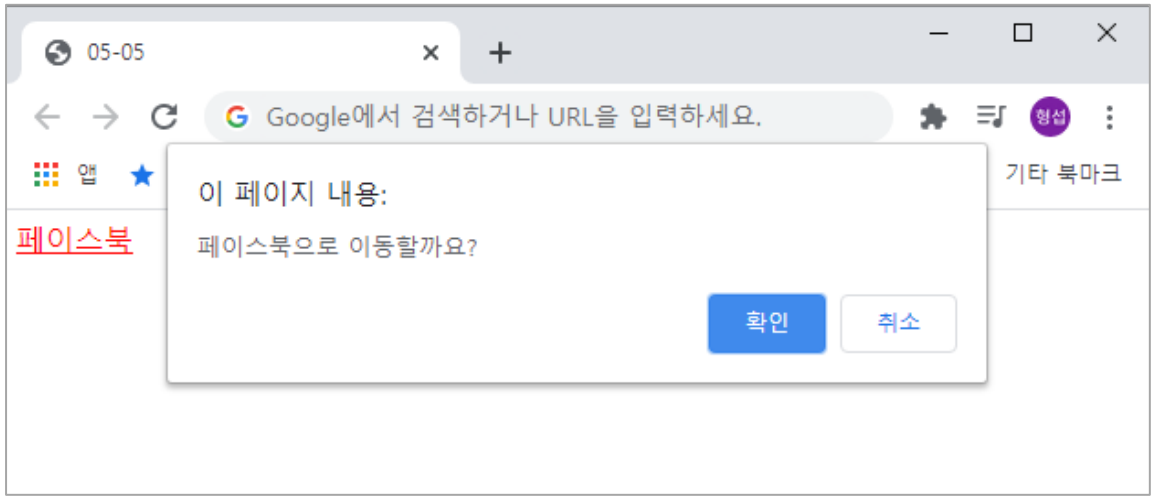
❖기본 이벤트(Default Event)

- HTML 문서나 요소에 어떤 기능을 실행하도록 이미 정의되어 있는 이벤트
- 대표적인 예
  - 〈a〉 요소를 클릭했을 때 href 특성의 경로로 페이지를 이동시킴.
  - 브라우저 화면을 마우스 오른쪽 클릭했을 때 내장 컨텍스트 메뉴(ContextMenu : 과거에는 팝업 메뉴라고 불렀지만 공식적인 명칭은 컨텍스트 메뉴입니다)가 나타남.
  - 〈form〉 요소 내부의 submit 버튼을 클릭했을 때 〈form〉 요소의 action 특성에 지정된 경로로 method 특성에 지정된 방식으로 전송함.
  - 〈input type="text" ... /〉 요소에 키보드를 누르면 입력한 문자가 텍스트 박스에 나타남.
- HTML 마크업 만으로 미리 정의된 기능을 실행하므로 편리함
  - 하지만 때로는 걸림돌 --〉 실행을 중지시킬 수 있어야 함
  - 이벤트 객체의 preventDefault() 메서드

5. 기본 이벤트

❖ 예제 05-05

```
<div id="app">
  <div @contextmenu="ctxStop" style="position: absolute; top: 5px; right: 5px; bottom: 5px; left: 5px">
    <a href="https://facebook.com" @click="confirmFB">페이스북</a>
  </div>
</div>
<script src="https://unpkg.com/vue"></script>
<script type="text/javascript">
  var vm = Vue.createApp({
    name: "App",
    methods: {
      ctxStop(e) {
        e.preventDefault();
      },
      confirmFB(e) {
        if (!confirm("페이스북으로 이동할까요?")) {
          e.preventDefault();
        }
      },
    },
  }).mount("#app");
</script>
```



5. 기본 이벤트

❖ 기본 이벤트와 관련된 이벤트 수식어

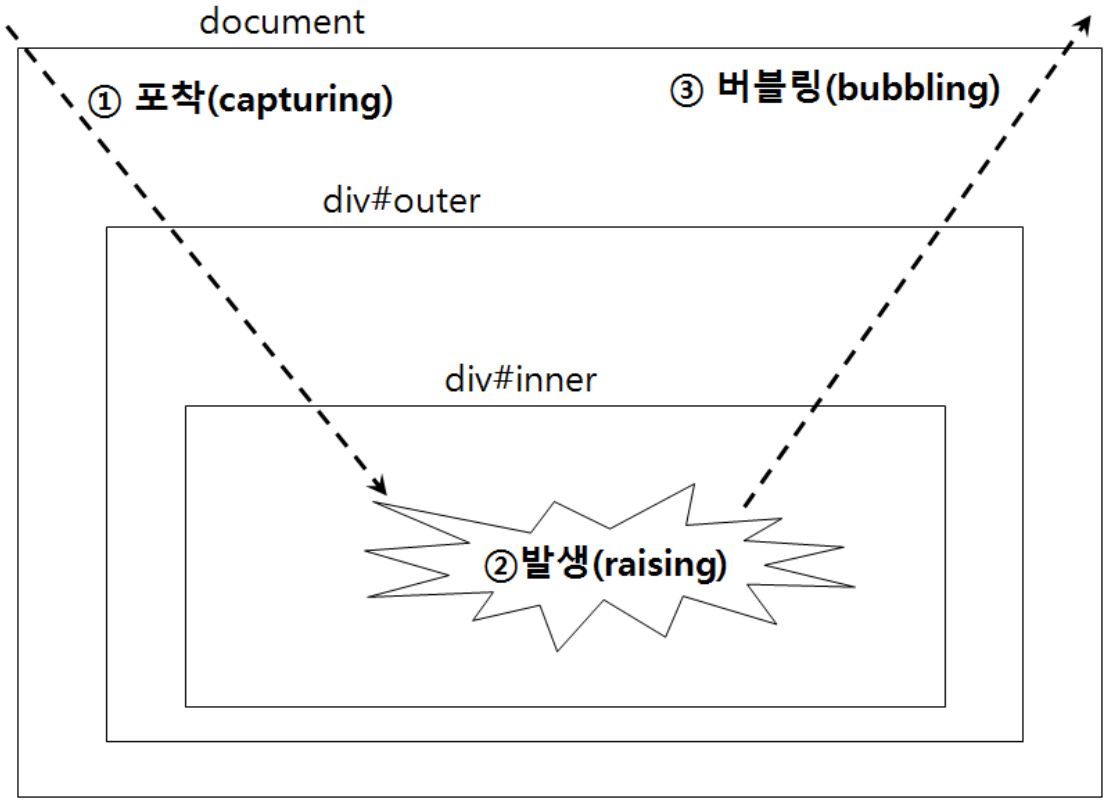
- .prevent
- 예제 05-06

```
<div id="app">
  <div @contextmenu.prevent style="position: absolute; top: 5px; right: 5px; bottom: 5px; left: 5px">
    <a href="https://facebook.com" @click="confirmFB">페이스북</a>
  </div>
</div>
<script src="https://unpkg.com/vue"></script>
<script type="text/javascript">
  var vm = Vue.createApp({
    name: "App",
    methods: {
      //ctxStop 메서드 삭제
      confirmFB(e) {
        if (!confirm("페이스북으로 이동할까요?")) {
          e.preventDefault();
        }
      },
    },
  }).mount("#app");
</script>
```

6. 이벤트 전파와 버블링

❖ HTML 문서에서의 이벤트 처리 3단계

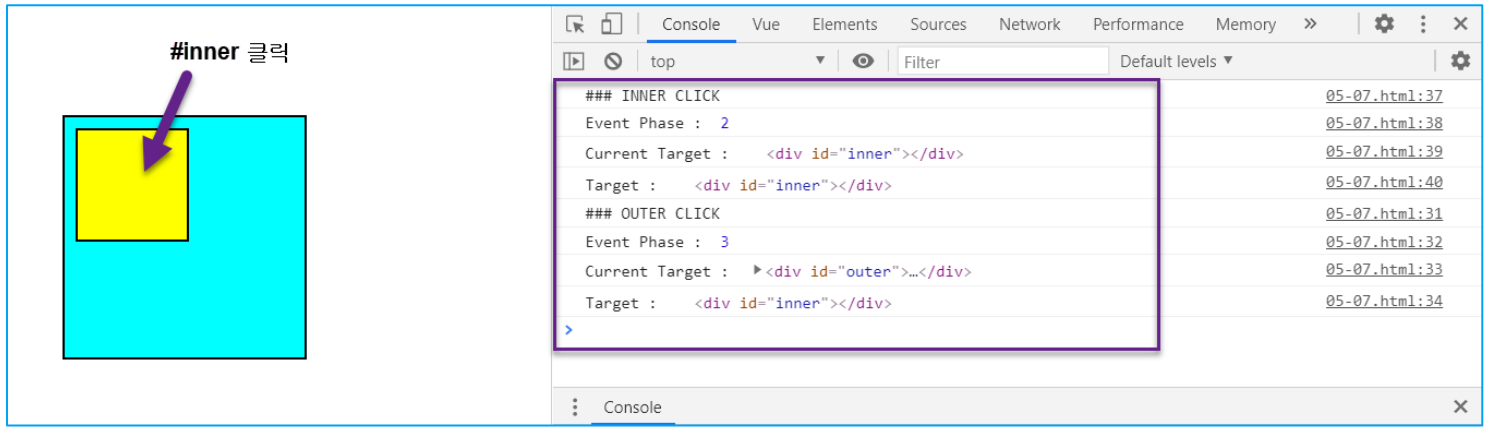
- 포착(CAPTURING_PHASE)
  - HTML 문서의 밖에서부터 이벤트를 발생시킨 HTML 요소까지 포착해 들어가는 단계
- 이벤트 발생(RAISING_PHASE : AT_TARGET)
  - 이벤트를 발생시킨 요소에 다다르면 요소의 이벤트에 연결된 함수를 직접 호출
- 버블링(BUBBLING_PHASE)
  - 이벤트가 발생한 요소로부터 상위 요소로 거슬러 올라가면서 동일한 이벤트를 호출



6. 이벤트 전파와 버블링

❖이벤트 버블링 확인 : 예제 05-07

```
<div id="app">
  <div id="outer" @click="outerClick">
    <div id="inner" @click="innerClick"></div>
  </div>
</div>
<script src="https://unpkg.com/vue"></script>
<script type="text/javascript">
  var vm = Vue.createApp({
    name: "App",
    methods: {
      outerClick(e) {
        console.log("### OUTER CLICK");
        console.log("Event Phase : ", e.eventPhase);
        console.log("Current Target : ", e.currentTarget);
        console.log("Target : ", e.target);
      },
      innerClick(e) {
        console.log("### INNER CLICK");
        console.log("Event Phase : ", e.eventPhase);
        console.log("Current Target : ", e.currentTarget);
        console.log("Target : ", e.target);
      },
    },
  }).mount("#app");
</script>
```



	#inner click	#outer click
eventPhase	2 (AT_TARGET)	3 (BUBBLING)
currentTarget	#inner	#outer
target	#inner	#inner

6. 이벤트 전파와 버블링

❖ 이벤트 버블링을 막으려면?

- 이벤트 객체의 stopPropagation() 메서드

예제 05-08

```
<script type="text/javascript">
var vm = Vue.createApp({
    name : "App",
    methods : {
        outerClick(e) {
            ......
            e.stopPropagation();
        },
        innerClick(e) {
            ......
            e.stopPropagation();
        }
    }
}).mount("#app")
</script>
```

6. 이벤트 전파와 버블링

❖이벤트 전파 관련 이벤트 수식어

- .stop: 이벤트 전파를 중단시킵니다.
- .capture: CAPTURING_PHASE 단계에서만 이벤트가 발생합니다.
- .self: RAISING_PHASE 단계일 때만 이벤트가 발생합니다.

예제 05-09

```
<div id="app">
    <div id="outer" @click.stop="outerClick">
        <div id="inner" @click.stop="innerClick"></div>
    </div>
</div>
```

```
<div id="example">
    <div id="outer" @click.capture.stop="outerClick">
       <div id="inner" @click.stop="innerClick"></div>
    </div>
</div>
```

7. 이벤트 수식어

❖ once 수식어

- 한번만 이벤트를 발생시키고 이벤트 연결을 해제함

❖키코드 관련 수식어

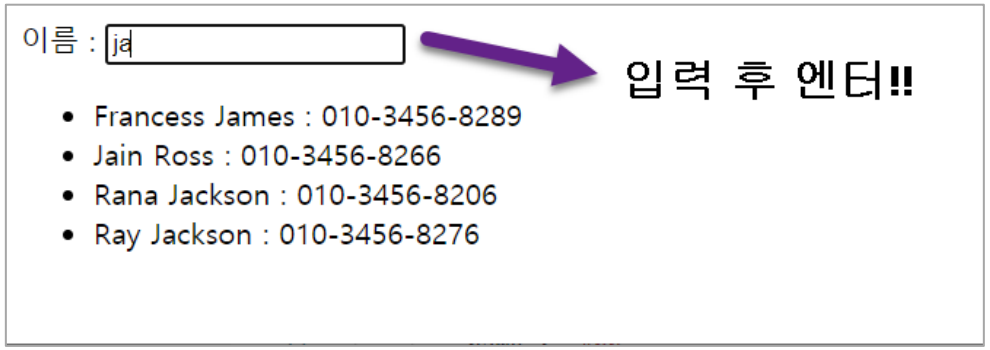
- 키보드 관련 이벤트를 처리할 때 사용할 수 있는 수식어
- 예제 05-11
  - 이벤트 객체의 keyCode 속성 사용
  - 이 예제에 키코드 수식어를 적용하면?

```
<div id="app">
  이름 : <input type="text" v-model.trim="name" @keyup="search" placeholder="영문 두글자 이상을 입력하세요" /><br />
  <ul>
    <li v-for="c in contacts">{{c.name}} : {{c.tel}}</li>
  </ul>
  <div v-show="isLoading">검색중</div>
</div>
<script type="text/javascript" src="https://unpkg.com/vue"></script>
<script type="text/javascript" src="https://unpkg.com/axios"></script>
<script type="text/javascript" src="https://unpkg.com/lodash"></script>
<script type="text/javascript">
  const BASEURL = "https://contactsvc.bmaster.kro.kr";
  var vm = Vue.createApp({
    name: "App",
    data() {
      return { name: "", contacts: [], isLoading: false };
    },
    methods: {
      search(e) {
        if (e.keyCode === 13) {
          if (this.name.length >= 2) {
            this.fetchContacts();
          } else {
            this.contacts = [];
          }
        }
      },
      fetchContacts() {
        this.isLoading = true;
        axios.get(BASEURL + `/contacts_long/search/${this.name}`).then((response) => {
          this.isLoading = false;
          this.contacts = response.data;
        });
      },
    },
  }).mount("#app");
</script>
```

7. 이벤트 수식어

❖예제 05-12

- 3행 : @keyup.enter="...."
- 14, 20행 주석
  - keyCode 값으로 조건 처리할 필요 없음



예제 05-12

```
01: <body>
02:     <div id="app">
03:         이름 : <input type='text' v-model.trim="name" @keyup.enter="search"
04:                     placeholder="영문 두글자 이상을 입력하세요" /><br />
05:         ......
06:     </div>
07:     ......
08:     <script type="text/javascript">
09:     const BASEURL = "https://contactsvc.herokuapp.com";
10:     var vm = Vue.createApp({
11:         ......
12:         methods : {
13:             search() {
14:                 // if (e.keyCode === 13) {
15:                     if (this.name.length >=2) {
16:                         this.fetchContacts();
17:                     } else {
18:                         this.contacts = [];
19:                     }
20:                 // }
21:             },
22:             ......
23:         }
24:     }).mount('#app')
25:     </script>
26: </body>
```

7. 이벤트 수식어

❖다양한 키코드 수식어

.up	.down	.left	.right
.enter	.tab	.delete	.esc
.space	.ctrl	.alt	.shift
.meta			

표 05-06 키 관련 수식어

- 조합하여 사용할 수 있음
  - @keyup.ctrl.enter="..."
  - @click.alt.shift="..."
  - @keyup.ctrl.c="..."

7. 이벤트 수식어

❖마우스 관련 수식어

.left	.right	.middle

- 예제 05-13
  - 마우스 버튼 수식어를 이용해 마우스 왼쪽, 오른쪽 버튼을 클릭해서 목록에서 선택된 아이템을 이동

7. 이벤트 수식어

❖exact 수식어

- 여러 수식어를 함께 사용할 때 정확하게 일치하는 조합으로 이벤트가 일어나야 핸들러가 실행
- 예제 05-14
  - @click, @click.ctrl, @click.ctrl.alt 이벤트를 한 요소에 적용
  - exact 수식어를 지정하지 않고, click.ctrl.alt 한다면?
    --〉 @click, @click.ctrl, @click.ctrl.alt 에 연결된 기능이 모두 실행됨

```
<button @click="num=num+1" @click.ctrl="num=num+10"
    @click.ctrl.alt="num=num+100">
```

- 예제 05-15

```
<button @click.exact="num=num+1" @click.ctrl.exact="num=num+10"
    @click.ctrl.alt.exact="num=num+100">
```

8. 마무리

지금까지 v-on 디렉티브를 이용해 이벤트를 처리하는 방법과 v-on 디렉티브는 @로 간략하게 줄여서 사용할 수 있다는 것을 알아보았습니다. 더불어 기본 이벤트, 이벤트 전파의 개념과 처리 방법과 .prevent, .stop과 같은 관련된 이벤트 수식어도 함께 살펴보았습니다. 이벤트 처리는 브라우저에서 애플리케이션과 사용자 사이의 상호작용을 일으키는 중요한 기법입니다. 사용법을 잘 익혀두어야 합니다.