

인라인 이벤트



■ 인라인 이벤트 처리

- v-on 디렉티브 이용
 - v-on:click="balance += parseInt(amount) "
 - @click="balance += parseInt(amount) "
- 인라인 이벤트는 복잡한 처리 기능을 포함시켜 작성하기 힘들
- 예제 04-01 실행 결과



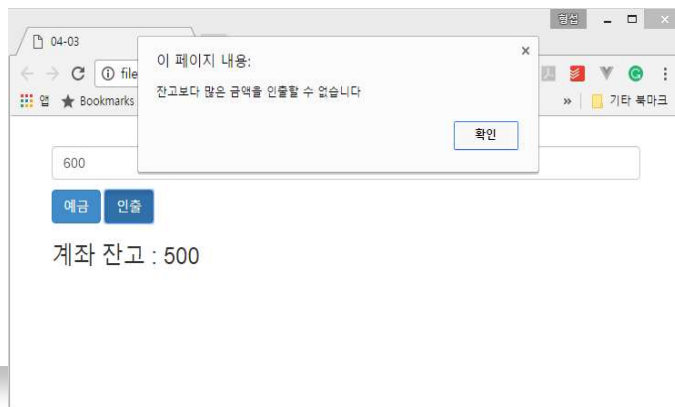
이벤트 핸들러 메서드



■ Vue 인스턴스에 등록한 메서드를 호출할 수 있음.

■ 예제 04-02

```
<div id="example" class="container layout1">
  <p><input type="text" v-model="amount"
    class="form-control" /></p>
  <p>
    <button id="deposit" @click="deposit($event)"
      class="btn btn-primary">예금</button>
    <button id="withdraw" @click="withdraw"
      class="btn btn-primary">인출</button>
  </p>
  <h3>계좌 잔고 : {{balance}}</h3>
</div>
```



```
var vm = new Vue({
  .....
  methods : {
    deposit : function(e) {
      var amt = parseInt(this.amount);
      if (amt <= 0) {
        alert("0보다 큰 값을 예금해야 합니다");
      } else {
        this.balance += amt;
      }
    },
    withdraw : function(e) {
      var amt = parseInt(this.amount);
      if (amt <= 0) {
        alert("0보다 큰 값을 인출할 수 있습니다");
      } else if (amt > this.balance) {
        alert("잔고보다 많은 금액을 인출할 수 없습니다");
      } else {
        this.balance -= amt;
      }
    }
  }
})
```

이벤트 객체



이벤트 객체

- 이벤트 발생시의 상황 정보
- 이벤트 처리 메서드의 첫번째 파라미터로 전달
- 이벤트 객체의 주요 정보

표 04-01 이벤트 객체의 주요 공통 속성

속성명	설명
target	이벤트가 발생한 HTML 요소를 리턴함
currentTarget	이벤트리스너가 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 이벤트를 방지할 수 있는지 여부를 리턴함
defaultPrevented	기본 이벤트가 방지되었는지 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

이벤트 객체



■ 이벤트 객체의 주요 정보

표 04-02 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 발생할 때 Unicode 캐릭터 코드를 리턴함
location	디바이스에서의 키 위치값. 일반 키보드는 이 값이 모두 0이므로 이용할 수 없음

이벤트 객체



■ 이벤트 객체의 주요 정보

표 04-03 마우스 이벤트 관련 속성

속성	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 좌표값에 영향을 받지 않음.
layerX, layerY	마우스 이벤트가 발생한 HTML 요소 영역상에서의 좌표(IE와외의 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

이벤트 객체



- 이벤트 객체의 주요 정보

표 04-04 이벤트 객체의 주요 메서드

메서드명	설명
preventDefault()	기본 이벤트의 자동 실행을 중지시킴
stopPropagation()	이벤트의 전파를 막음

기본 이벤트



■ 기본 이벤트(Default Event)

- 개발자가 이벤트를 직접 연결하지 않아도 실행되는 기능
 - <a> 요소는 클릭이벤트를 작성하지 않아도 href 특성에 정의된 경로로 이동시킴
- 대표적인 기본 이벤트
 - <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
- 기본 이벤트는 유용하지만 때로는 실행을 중지시켜야 할 경우가 있음
 - preventDefault() 메서드!!

기본 이벤트



예제 04-04

```
<div id="example" v-on:contextmenu.prevent="ctxStop">
  <a href="https://facebook.com" @click="confirmFB">페이스북</a>
</div>
```

```
<script type="text/javascript">
var vm = new Vue({
  el: "#example",
  methods: {
    ctxStop: function(e) {
      e.preventDefault();
    },
    confirmFB: function(e) {
      if (!confirm("페이스북으로 이동할까요?")) {
        e.preventDefault();
      }
    }
  }
})
</script>
```



기본 이벤트



■ 매번 preventDefault() 실행을 작성하도록 신경쓰기 어려움

- .prevent 수식어
 - v-on:contextmenu.**prevent**="ctxStop"
 - e.preventDefault() 메서드를 호출하지 않아도 됨.
- 이벤트 수식어가 편리하지만 조건 논리식의 결과에 따라 실행해야 하는 경우에는 preventDefault() 메서드를 직접 호출해야 하는 경우도 있음

이벤트 전파와 버블링



이벤트 처리의 3단계

- 1단계 : CAPTURING_PHASE
- 2단계 : RAISING_PHASE(AT_TARGET)
- 3단계 : BUBBLING_PHASE

일반적인 경우는 2,3단계에서 이벤트 핸들러 메서드가 호출

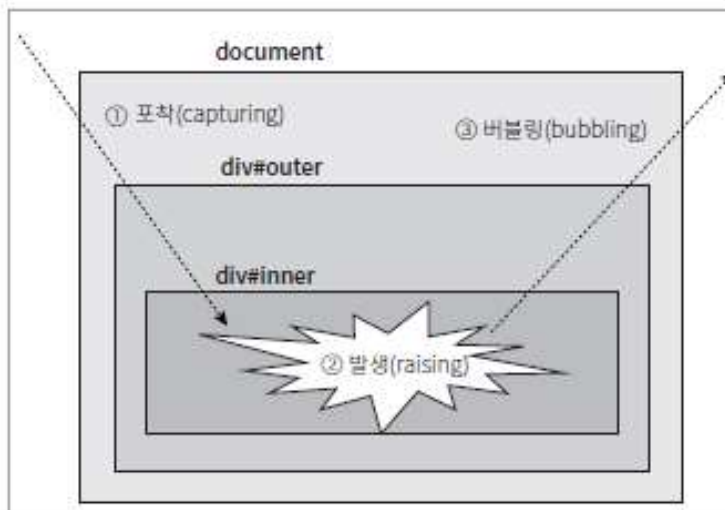


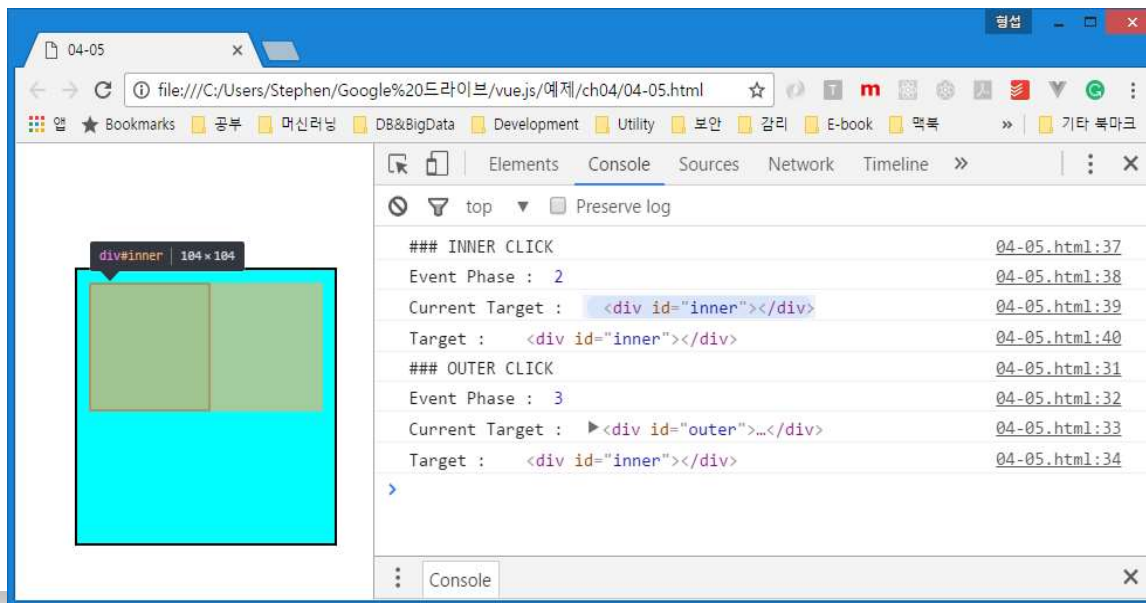
그림 04-04 이벤트 전파의 3단계

이벤트 전파와 버블링



예제 04-05 실행 결과

- #inner 요소를 클릭했음에도 3단계 버블링이 일어나면서 #outer의 요소에 바인딩된 이벤트 핸들러가 호출
- currentTarget과 target
 - currentTarget : 현재 이벤트 핸들러를 실행시킨 요소
 - target : 이벤트가 일어난 원본 요소

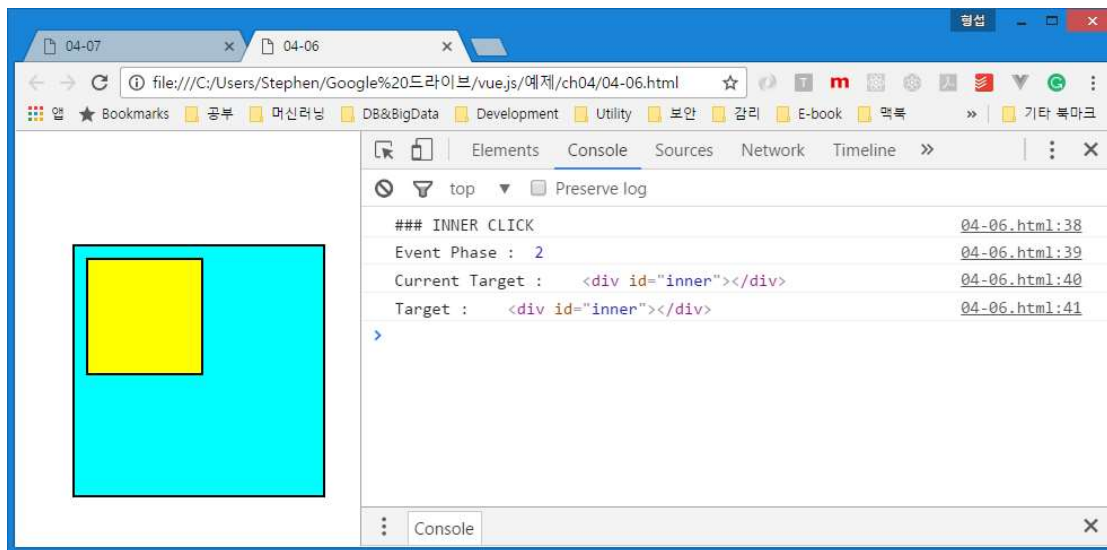


이벤트 전파와 버블링



■ stopPropagation() 메서드

- 이벤트의 전파(propagation)를 막는 메서드
 - e.stopPropagation()
- 예제 04-06 실행 결과



이벤트 전파와 버블링



■ 이벤트 전파 관련 수식어

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

■ 예제 04-07

- stopPropagation() 메서드 대신에 수식어로 코드 변경

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

이벤트 수식어



■ once 수식어

- 이벤트를 한번만 발생시킴.

```
<button id="create" v-on:click.once="specialEvent"  
  class="btn btn-primary">계좌 개설 10000원 이벤트</button>
```

```
var vm = new Vue({  
  .....  
  methods : {  
    specialEvent : function(e) {  
      this.balance += 10000;  
    },  
    .....  
  }  
})
```



이벤트 수식어



❖ 키코드 수식어

- 키보드 관련 이벤트를 처리할 때 유용함
- 관찰 속성을 사용한 예제 03-10을 예제 04-09와 같이 엔터키를 누르면 조회되도록 변경

이름 : `<input type="text" v-model="name" v-on:keyup="search" placeholder="두글자 이상을 입력하세요" />`

```
var vm = new Vue({
  .....
  methods : {
    search : function(e) {
      if (e.keyCode === 13) {
        var val = e.target.value;
        if (val.length >= 2) { this.fetchContacts(); }
        else { this.contactlist = []; }
      }
    },
  },
  .....
})
```

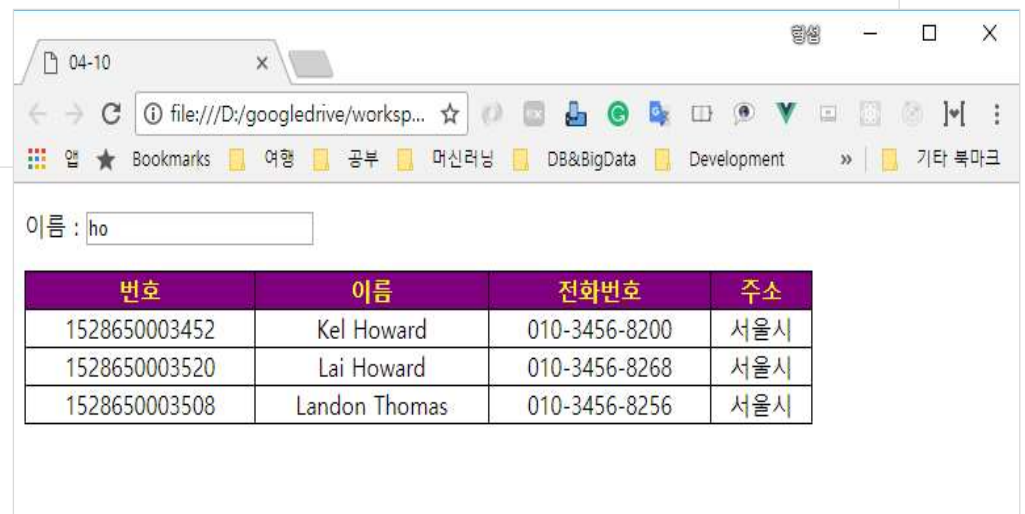
이벤트 수식어



❖ 키코드 수식어(이어서)

이름 : `<input type="text" v-model="name" v-on:keyup.13="search" placeholder="두글자 이상을 입력하세요" />`

```
var vm = new Vue({
  .....
  methods : {
    search : function(e) {
      var val = e.target.value;
      if (val.length >= 2) { this.fetchContacts(); }
      else { this.contactlist = []; }
    },
  },
  .....
})
```



이벤트 수식어



■ 키코드 수식어의 별칭

- v-on:keyup.13
- v-on:keyup.enter

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

- 이벤트 수식어는 동시에 여러개를 적용할 수 있음
 - v-on:keyup.ctrl.67="copy"
 - CTRL + C를 눌렀다가 떼때의 이벤트 처리

이벤트 수식어



■ 마우스 버튼 수식어

left right middle

■ 예제 04-11 실행 결과

