

2025년 상반기 K-디지털 트레이닝

이벤트 처리

사용자 입력에 반응 해야지 그럼

[KB] IT's Your Life



☑ 인라인 이벤트

o v-on 디렉티브로 설정

- o 형식
 - v-on:[이벤트 이름]="표현식"
- 단축 표현
 - @[이벤트 이름]="표현식"
- 이벤트 객체가 필요한 경우
 - @click="test(\$event)"

 대체로 메소드 호출문

```
method : {
    test() {...},
}
```

```
이벤트 이름 목록

click
change
dbclick
mouseover
submit form
reset form
select select값이 선택되었을 때
focus
keyup
keydown
keybress
```

인라인 이벤트 처리

```
2 05-01.html
                                                항상 타입을 신경써라
<div id="app">
                                                                            여기서 계산 표현문.
따로 함수를 정의하여 호출하지 않고
 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 />
                                                                       입력
 <h3>계좌 잔고: [{{balance}}]</h3>
</div>
                                                        금액: 100000
                                                         입금 인출
                                                                   이벤트 입력
<script src="https://unpkg.com/vue"></script>
<script>
                                                        계좌 잔고: 105000 출력
  let vm = Vue.createApp({
   name: "App",
   data() {
     return {
       amount: 0,
       balance: 0,
                                                                  템플릿 파트 (html파트)는
     };
                                                                   [순수하게 출력을 위한 표현만 하는 것이 권장됨
 }).mount("#app");
                                                                  로직 처리는 스크립트 파트에서 해라.
이 예시는 좋은 예시 아님
</script>
```

번외 .number 안붙이면 문자열로 받을 수 있으니

2 이벤트 핸들러 메서드

- ♡ 이벤트 핸들러 메서드
 - Vue 인스턴스의 methods 속성에 정의한 함수를 이벤트 처리 함수로 연결

6 05-02.html

```
<div id="app">
 금액: <input type="text" v-model.number="amount" /> <br />
 <button @click="deposit">입금</button>
 <button @click="withdraw">인출</button>
 <br />
 <h3>계좌 잔고: [{{balance}}]</h3>
</div>
<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
   name: "App",
   data() {
     return {
       amount: 0,
       balance: 0,0
     };
   },
```

☑ 05-02.html

```
methods: {
                                      .number했는데 왜?
소숫점을 쓰는 변태가 있을 수 있잖아
   __deposit() {
       let amt = parseInt(this.amount);
       if (amt <= 0) {
         alert("0보다 큰 값을 예금하세요.");
       } else {
         this.balance += amt;
     Twithdraw() {
                                                            반환 없어도 되는 로직을 처리하는 함수
       let amt = parseInt(this.amount);
       if (amt <= 0) {
         alert("0보다 큰 값을 인출하세요.");
                                                             타입을 주의하라고 했는데
       } else if (amt > this.balance) {
         alert("잔고보다 많은 금액을 인출할 수 없습니다.");
                                                             typeof 연산자를 활용하는 것도 좋은 방법이다.
       } else {
         this.balance -= amt;
                                                           console.log( typeof this.amount );
     },
 }).mount("#app");
</script>
```

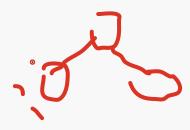
이벤트 핸들러 메서드

6 05-03.html (03-08.html 예제)

```
이벤트
                               v-bind
    <div id="app">
      <input id="a" type="text" :value="name" @input="changeName" />
      <br />
      이름 : <span>{{name}}</span>
    </div>
    <script type="text/javascript" src="https://unpkg.com/vue"></script>
    <script type="text/javascript">
      var vm = Vue.createApp({
        name: "App",
        data() {
          return { name: "" };
        methods: {
                      자동으로 이벤트 객체 매개변수 쏘옥
         -changeName(e) {
가상돔에
적용됨
            this.name = e.target.value;
      }).mount("#app");
    </script>
```

함수명만 제시, 매개변수 명시X 하면 이벤트 객체를 매개변수로 쏘옥 한다. 이벤트 객체말고 다른 정보를 주고 싶다면??

가상 dom



스크립트 파트에서 this니까 가상 dom에서의 this다.

해당 되는 vue인스턴스가 this다. 임의의 매개변수를 지정하면 지정한 정보가 매개변수로 넘어간다

호출문을 적으라는 것.

흔하게 이벤트 객체와 지정한 정보를 둘다 넘기고 싶다면 @input="함수명(\$event, 매개변수들)"

\$변수는 vue의 특정 변수죠?

☑ 05-04.html 인라인 이벤트 핸들러 지정

```
<div id="app">
 <input type="text" :valu="name" @input="(e)=>this.name=e.target.value" />
 <br />
 이름: <span>{{name}}</span>
</div>
                                                          vue인스턴스 this다.
<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
   name: "App",
   data() {
      return {
       name: "",
     };
 }).mount("#app");
</script>
```

🧿 이벤트 객체

○ 주요 공통 속성

| 속성명 | 설명 |
|------------------------|---|
| target | 이벤트가 발생한 HTML 요소 |
| - currentTarget | 이벤트 리스너가 이벤트를 발생시키는 HTML 요소 |
| path | 배열값, 이벤트 발생 HTML 요소로부터 document, window 객체까지 거슬러 올라가는 경로 |
| bubbles | 현재의 이벤트가 버블링을 일으키는 이벤트인지 여부 |
| cancelable | 기본 이벤트의 실행 취소를 할 수 있는지 여부 |
| defaultPrevented | 기본 이벤트의 실행이 금지되어 있는지 여부 |
| eventPhase | 이벤트 흐름의 단계 1: 포착 CAPTURE_PHASE 2: 이벤트 발생 AT_TARGET 3: 버블링 BUBLLING_PHASE |
| srcElement | IE에서 사용되던 속성, target과 동일한 속성 |

☑ 이벤트 객체

○ 키코드 이벤트 관련 속성

| 속성명 | 설명 |
|----------|--|
| altKey | ALT 키가 눌러졌는지 여부 |
| shiftKey | SHIFT 키가 눌러졌는지 여부 |
| ctrlKey | CTRL 키가 눌러졌는지 여부 |
| metaKey | 메타키(Window Key, Command Key)가 눌러졌는지 여부 |
| key | 이벤트에 의해 나타나는 키의 값(대소문자 구분함) |
| code | 이벤트를 발생시킨 키의 코드값 ex) a를 눌렀을 때 "KeyA" Shift키를 눌렀을 때 "Shift" |
| keyCode | 이벤트를 발생시킨 키보드의 고유 키코드 ex) a, A는 65(대소문자 구분하지 않음) |
| charCode | keypress 이벤트가 발생될 때 Unicode |

이벤트 객체

o <u>마우스</u> 이벤트 관련 속성

| 속성명 | 설명 |
|---------------------------------------|--|
| altKey, shiftKey, ctrlKey, metaKey | 키보드 이벤트 속성과 동일 |
| button | 이벤트를 발생시킨 마우스 버튼 0: 마우스 왼쪽버튼 1: 마우스 휠 2: 마우스 오른쪽 버튼 |
| buttons | 마우스 이벤트가 발생한 후 눌러져 있는 마우스 버튼의 값. 아래 값의 조합 1: 마우스 왼쪽 버튼, 2: 마우스 오른쪽 버튼, 4: 마우스 휠 |
| clientX, clientY | 뷰포트(ViewPort) 영역상의 좌표. 스크롤되도 좌표값은 영향받지 않음 |
| layerX, layerY | HTML 요소 영역상에서의 좌표(IE 이외의 브라우저) |
| offsetX, offsetY | HTML 요소 영역상에서의 좌표(IE 브라우저) |
| pageX, pageY | HTML 문서(Document) 영역상의 좌표 |
| screenX, screenY | 모니터 화면(Screen) 영역상의 좌표 |

💟 이벤트 객체

이벤트 객체의 주요 메서드

form처리할 때 신경써야하는 이벤트 버블링과 캡처링'

| 속성명 | 설명 |
|-------------------|----------------------|
| preventDefault() | 기본 이벤트의 자동 실행을 금지 시킴 |
| stopPropagation() | 이벤트의 전파를 중단시킴 |

4 기본 이벤트

🧿 기본 이벤트(Default Event)

- HTML 문서나 요소에 어떤 기능을 실행하도록 이미 정의되어 있는 이벤트
- 대표적인 기본 이벤트
 - <a>>
 - 클릭했을 때 href 특성의 경로로 페이지를 이동
 - 브라우저 화면에서 마우스 오른쪽 클릭 시 → 컨텍스트 메뉴 출력
 - <form> 요소 내부의 submit 버튼 클릭 → 서버로 전송
 - <input type="text"... />에서 키보드 입력 시 → 문자가 텍스트 박스에 나타남

○ 기본 이벤트의 실행 막가

event.preventDefault()

single page application

나중에

배포판을 보면

html 하나로 통합되어있다.

a와 form 의 특징은 또다른 html로 이동한다는 특징을가지고 있따. vue spa정책에 위반되는 특징이다! 해당 default특징을 막아야겠죠?

6 05-05.html

```
<div id="app"> 컨텍스트 메뉴 이벤트명
 <div @contextmenu="ctxStop"</pre>
   style="position: absolute; left: 5px; top: 5px; right: 5px; bottom: 5px">
    <a href="https://facebook.com" @click="confirmFB">페이스북</a>
 </div>
</div>
<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
    name: "App",
                                                    YouTub
                                                            127.0.0.1:5500 내용:
    data() {
                                                   페이스북
                                                            페이스북으로 이동할까요?
      return {};
    },
                                                                                               취소
    methods: {
      ctxStop(e) {
        e.preventDefault();
      confirmFB(e) {
        if (!confirm("페이스북으로 이동할까요?")) {
          e.preventDefault();
                                                      매번 preventDefault함수를 호출해야하냐?
귀찮은데
 }).mount("#app");
</script>
```

4 기본 이벤트

- 수식어를 통한 기본 이벤트 차단
 - o @이벤트.prevent
 - → 해당 이벤트를 차단. event.preventDefault() 호출 효과

6 05-06.html

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

이벤트 전파와 버블링

propagation

💟 이벤트 전파의 3단계

태그가 중첩된 상황 부모자식관계

click이벤트 발생된 상황

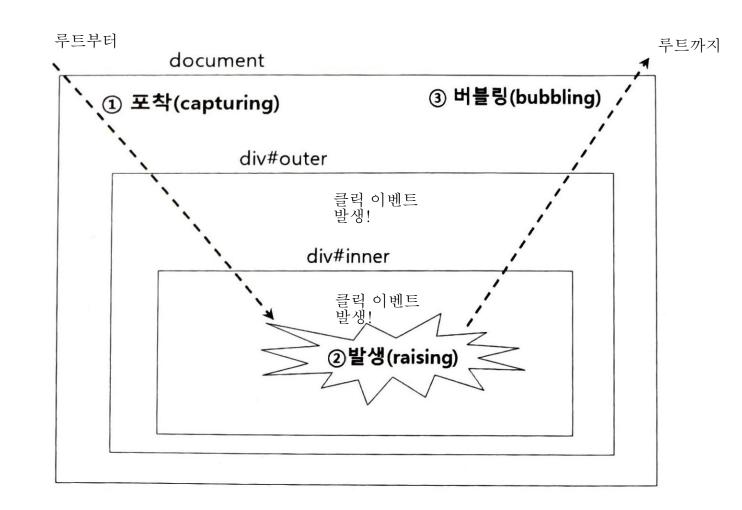
이 클릭을 누가 처리할 것인가??

여러 이벤트 발생할 때 누가 먼저 처리할 것인가??

2가지가 있다. 부모에서 자식 - 캡처링 자식에서 부모 - 버블링

택일 해야한다.

브라우저에서는 일반적으로 버블링전파. 일반적으로 해당요소에서부터 이벤트가 발생하기 원하기에



중간에 전파를 멈추는 STOP PROPAGATION 기능이 있다

2 05-07.html

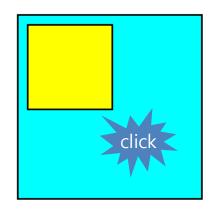
```
<style>
 #outer {
   width: 200px;
    height: 200px;
    border: solid 2px black;
    background-color: aqua;
    position: absolute;
    top: 100px;
    left: 50px;
    padding: 10px;
 #inner {
    width: 100px;
    height: 100px;
    border: solid 2px black;
    background-color: yellow;
</style>
<div id="app">
  <div id="outer" @click="outerClick">
    <div id="inner" @click="innerClick"></div>
  </div>
</div>
```

6 05-07.html

```
<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
    name: "App",
    data() {
      return {};
    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>
```

이벤트 전파와 버블링

OUTER 클릭 시



OUTER CLICK

Event Phase: 2

Current Target: ▶ <div id="outer"> • </div>

Target: ▶<div id="outer">....</div>

outer click Di

INNER 클릭시

default가 버블링 방식



Event Phase: 2

INNER CLICK

Current Target: <div id="inner"></div>

√Target: <div id="inner"></div>

OUTER CLICK

Event Phase: 3

Current Target: ▶<div id="outer">....</div>

➤ Target:

<div id="inner"></div>



wher Cher 对对

target 이벤트가 발생한 최초 요소

☑ 버블링의 차단

o event.stopPropagation() 호출

부모까지도 반응을 원하면 사용하면 안되겠죠?

☑ 관련 수식어

o .stop : 이벤트 전파 중단, event.stopPropagation() 호출과 동일

.capture: CAPTURING_PHASE 단계에서만 이벤트 발생

o .self: RASING_PHASE 단계일 때만 이벤트 발생

거의 안슴


```
<style>
 #outer {
   width: 200px;
    height: 200px;
    border: solid 2px black;
    background-color: aqua;
    position: absolute;
    top: 100px;
    left: 50px;
    padding: 10px;
 #inner {
    width: 100px;
    height: 100px;
    border: solid 2px black;
    background-color: yellow;
</style>
<div id="app">
  <div id="outer" @click.stop = "outerClick">
    <div id="inner" @click.stop = "innerClick"></div>
  </div>
</div>
```

6 05-08.html

```
<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
    name: "App",
    data() {
      return {};
    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>
```

g once 수식어

- 모든 이벤트에 연결가능
- 한 번만 이벤트를 발생시키고 이벤트 연결을 해제

가끔 유용하게 써먹음

<u>키코드 관련 이벤트</u> 수식어

- 키 관련 수식어
 - up, down, left, right
 - enter, esc
 - space, tab, delete
 - ctrl, alt, shift, meta
- @이벤트.enter="..."
 - 엔터를 쳤을 때 이벤트 핸들러 호출
- @keyup.ctrl.enter="..."
 - CTRL+ENTER 조합 시 이벤트 핸들러 호출
- @keyup.ctrl.c="..."
 - CTRL+C 조합

vanila js로는 addEventHandler해야죠? 귀찮죠?

마우스 관련 수식어

- ㅇ 수식어
 - left, right, middle: 마우스 버튼
 - 키 관련 수식어 사용 가능

click.left

키코드 + 마우스 수식어 조합도 가능하려나 click.control

수식어 만세

6 이벤트 수식어

exact 수식어

○ 다른 수식어와 조합해 이벤트를 등록할 때
 → 정확하게 일치하는 조합으로 이벤트가 일어나야 핸들러가 실행되도록 설정

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

다른 조합이 포함되면 안돼

뷰에만 제공되는 기능 ㅎ