2주차 템플릿 문법

작성자 : 배소희

목차

1. 보간법

2. 디렉티브

3. 약어

1. 보간법(Interpolation)

- 랜더링 된 DOM을 기본 Vue 인스턴스의 데이터에 선언적으로 바인딩 할 수 있는 HTML 기반 템플릿 구문을 사용
- 내부적으로는 Vue는 템플릿을 가상 DOM 렌더링 함수로 컴파일
- 템플릿 대신 렌더링 함수를 직접 작성 가능 (선택사항으로 JSX 지원)

* {{msg}}의 데이터는 인스턴스의 data 안의 msg 값을 가져옴

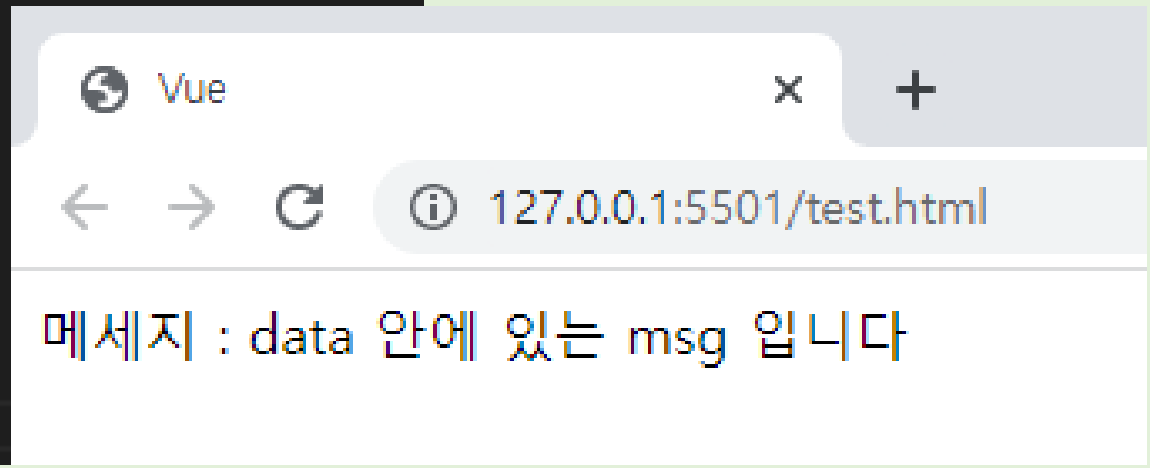
1. 보간법(Interpolation)

문자열

- Mustache(이중 중괄호) 구문을 사용한 텍스트 보간

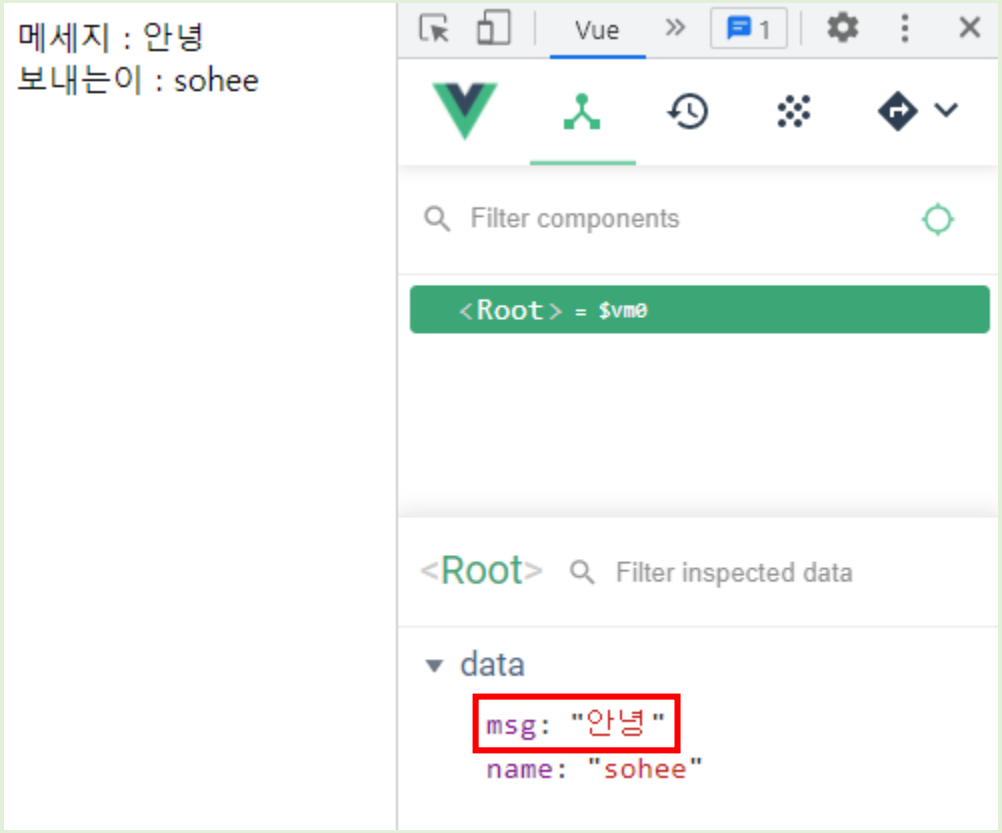
```
<body>
    <div id="app">
        <span>메세지 : {{msg}}</span>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            msg : 'data 안에 있는 msg 입니다'
        }
    });
</script>
```

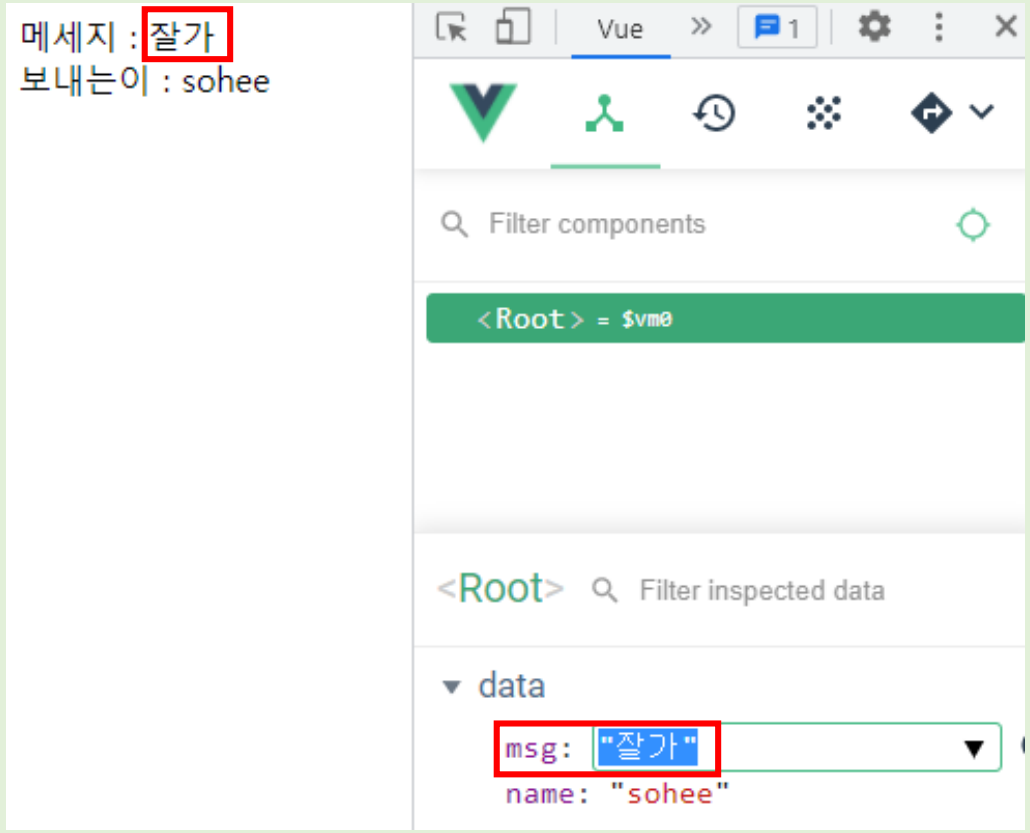
* {{msg}}의 데이터는 인스턴스의 data 안의 msg 값을 가져옴



1. 보간법(Interpolation)

- Mustache 태그는 msg 속성이 변경 될 때마다 갱신됨
- v-once 디렉티브를 사용하게 될 경우 데이터 변경시 업데이트 되지 않는 일회성 보간 (같은 노드의 바인딩에도 영향을 미침)





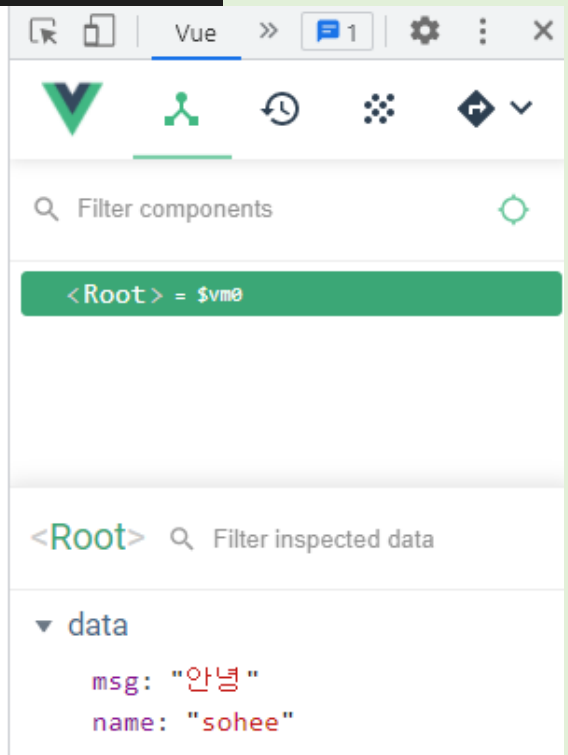
* msg 내용을 수정하면 변경된 데이터로 재랜더링 됨

1. 보간법(Interpolation)

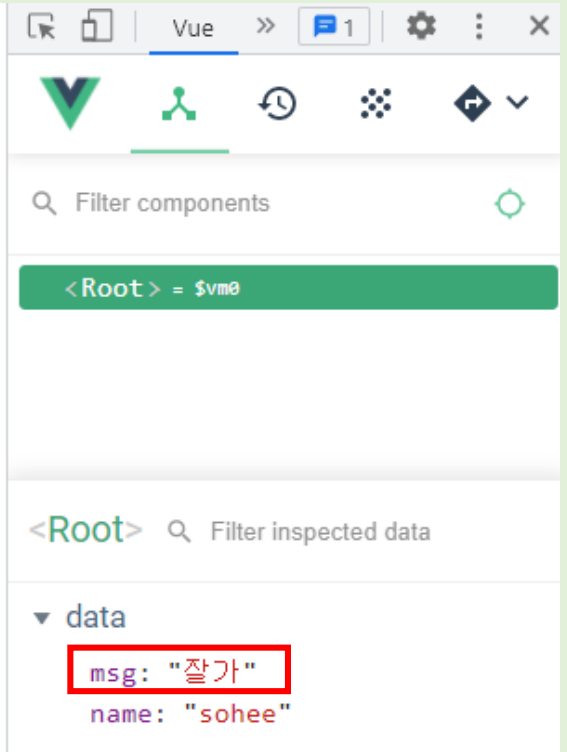
```
<body>
    <div id="app">
        <span v-once>
            <span>메세지 : {{msg}}</span></br>
            <span>보내는이 : {{name}} </span>
        </span>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            msg : '안녕',
            name : 'sohee'
        }

    });
</script>
```

* msg 내용을 수정해도 처음 바인딩 된 데이터 내용이 유지됨(재랜더링 되지 않음)

메세지 : 안녕
보내는이 : sohee



메세지 : 안녕
보내는이 : sohee

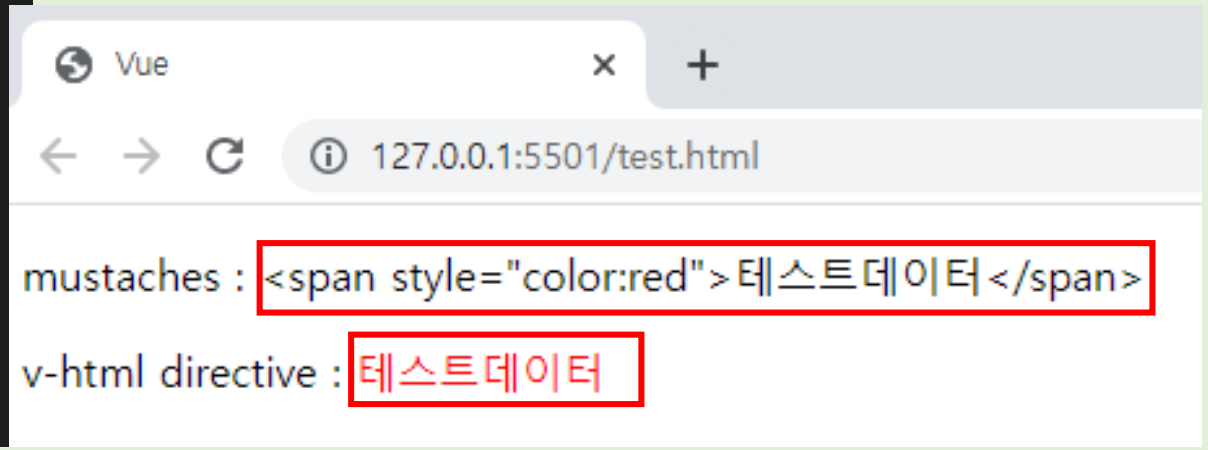


1. 보간법(Interpolation)

원시 HTML

- Mustache(이중 중괄호)는 HTML이 아닌 일반 텍스트로 데이터를 해석
- 실제 HTML을 출력하려면 v-html 디렉티브를 사용

```
<body>
    <div id="app">
        <p>mustaches : {{rawHTML}}</p>
        <p>v-html directive : <span v-html="rawHTML"></span></p>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            rawHTML : '<span style="color:red">테스트데이터</span>'
        }

    });
</script>
```



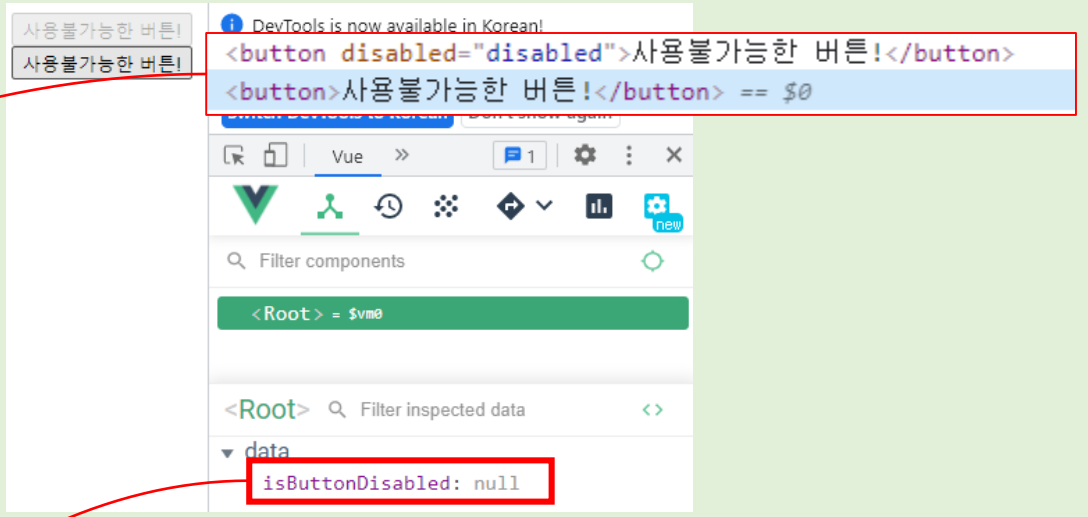
* html을 동적으로 렌더링 하면 XSS 취약점으로 쉽게 이어질 수 있음. 신뢰할 수 있는 콘텐츠에서만 HTML 보간을 사용.

1. 보간법(Interpolation)

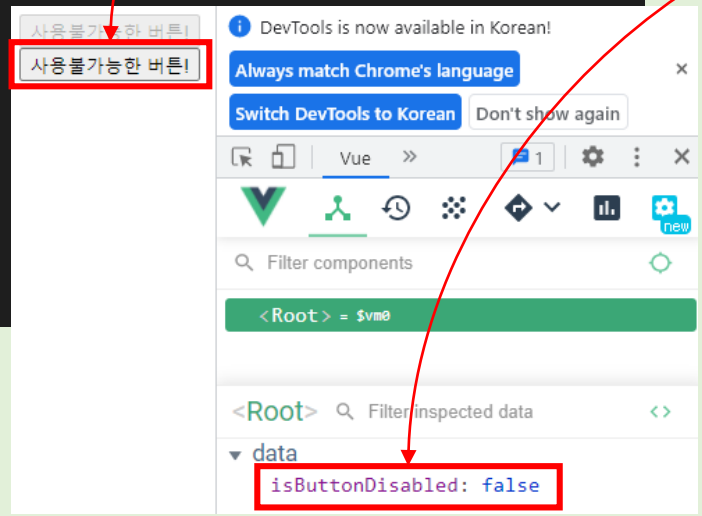
속성

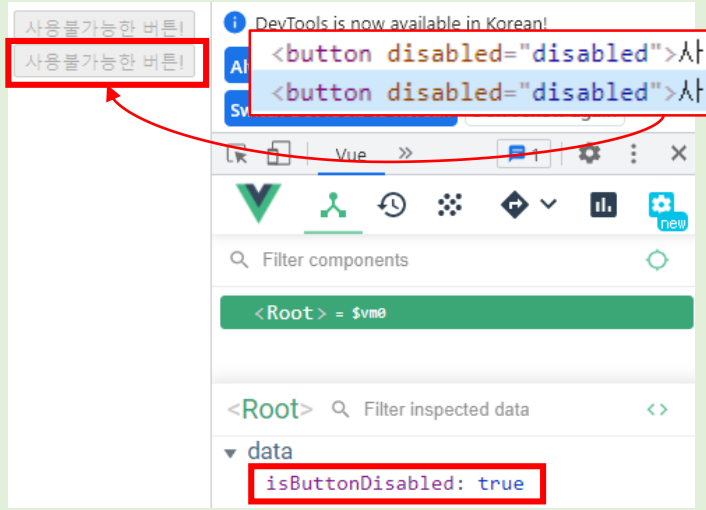
- Mustache는 HTML 속성에서 사용할 수 없음.
- v-bind 디렉티브로 사용 가능

```
<body>
    <div id="app">
        <button disabled>사용불가능한 버튼!</button>
        <button v-bind:disabled="isButtonDisabled">사용불가능한 버튼!</button>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            isButtonDisabled : false
        }

    });
</script>
```



* false, null, undefined 의 경우 disabled 속성은 해당 엘리먼트에 포함되지 않음





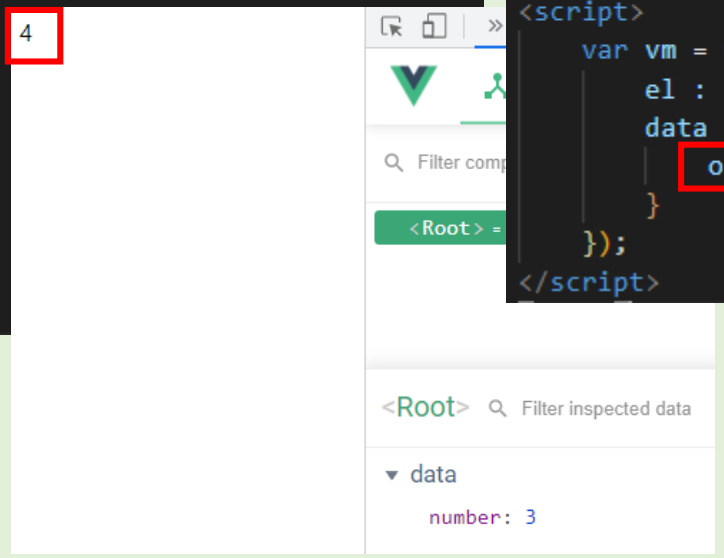
1. 보간법(Interpolation)

Javascript 표현식 사용

- Vue.js 는 모든 데이터 바인딩 내에서 Javascript 표현식의 모든 기능을 지원
- 각 바인딩에 하나의 단일 표현식만 표현 가능

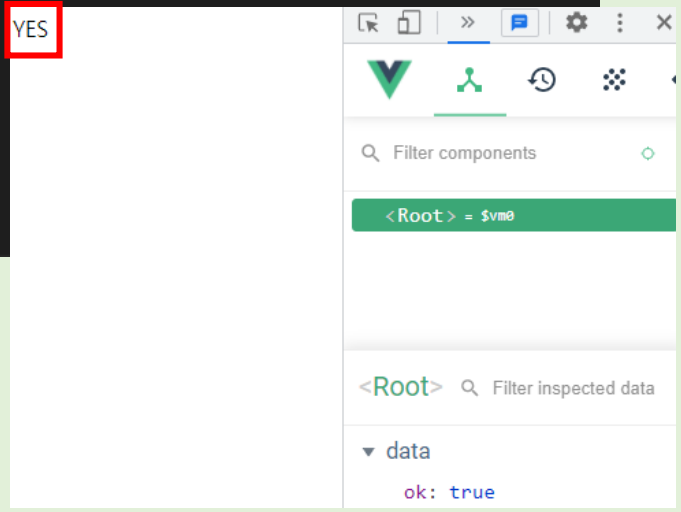
```
<body>
    <div id="app">
        <div>{{ number + 1 }}</div>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            number : 3
        }

    });
</script>
```

4




```
<body>
    <div id="app">
        <div>{{ ok ? 'YES' : 'NO'}}</div>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            ok : true
        }
    });
</script>
```

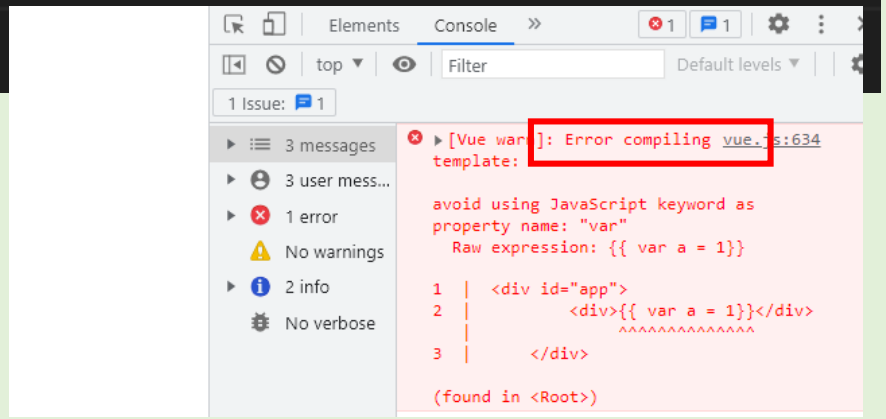
YES



1. 보간법(Interpolation)

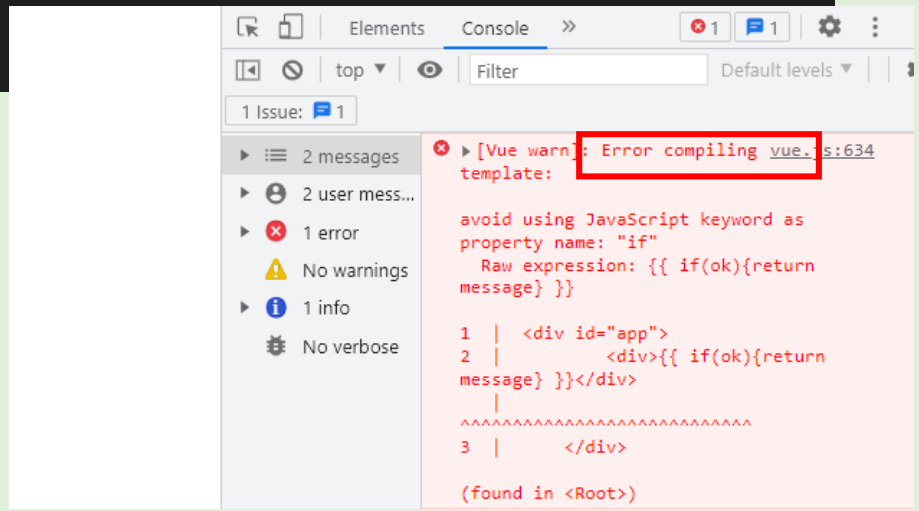
- 제한사항 예시

```
<body>
    <div id="app">
        <div>{{ var a = 1}}</div>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            a : 2
        }
    });
</script>
```



* 해당 내용은 구문이라 불가능, 표현식일 경우 정상적으로 표시
ex) {{a = 1}}
html 에 1이 표시됨

```
<body>
    <div id="app">
        <div>{{ if(ok){return message} }}</div>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            ok : true,
            message : 'hello'
        }
    });
</script>
```

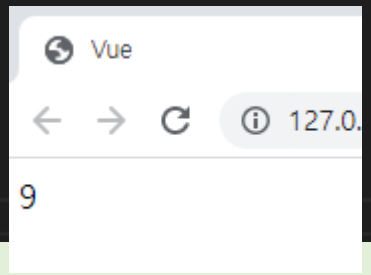


* 조건문은 작동하지 않음. 삼항연사자 사용

1. 보간법(Interpolation)

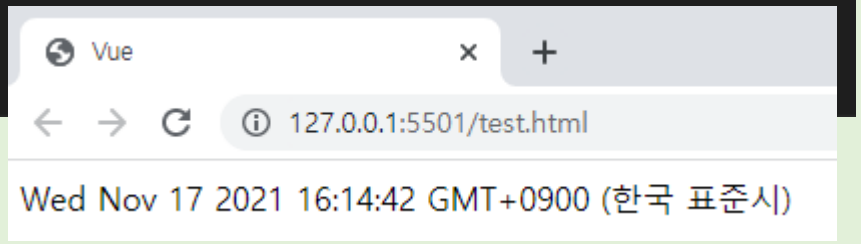
- Math와 Date 같은 전역으로 사용 가능한 것에 접근 가능
  (사용자 정의 전역에는 엑세스 하지 말것)

```
<body>
    <div id="app">
        <div>{{ Math.ceil(b) }}</div>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            b : 8.5
        }
    });
</script>
```



* 8.5 를 올림한다.

```
<body>
    <div id="app">
        <div>{{ new Date() }}</div>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
        }
    });
</script>
```



2. 디렉티브

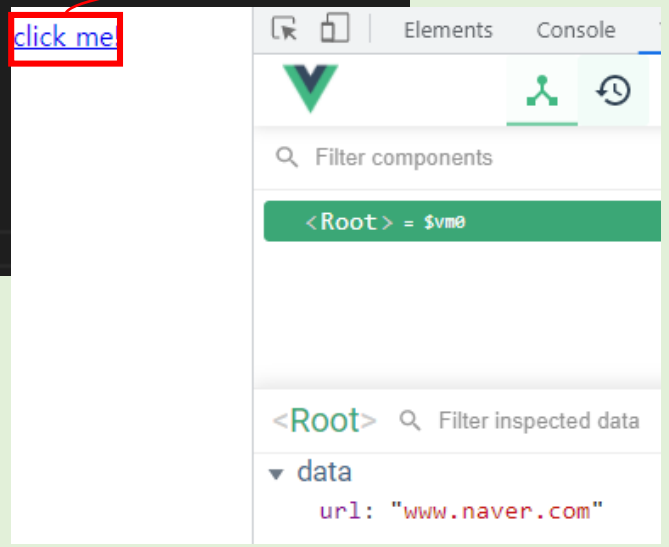
- 디렉티브는 v- 접두사가 있는 특수속성
- 디렉티브의 속성 값은 단일 Javascript 의 표현식이 됨 (예외 : v-for)
- 정의 : 특정한 행위의 기능을 가진 DOM element
- 역할 : 표현식의 값이 변경될 때 사이드 이펙트를 반응적으로 DOM에 적용하는 것
- 종류 : v-text, v-html, v-show, v-if, v-else, v-else-if, v-for, v-on, v-bind, v-model, v-slot, v-pre, v-cloak, v-once

2. 디렉티브

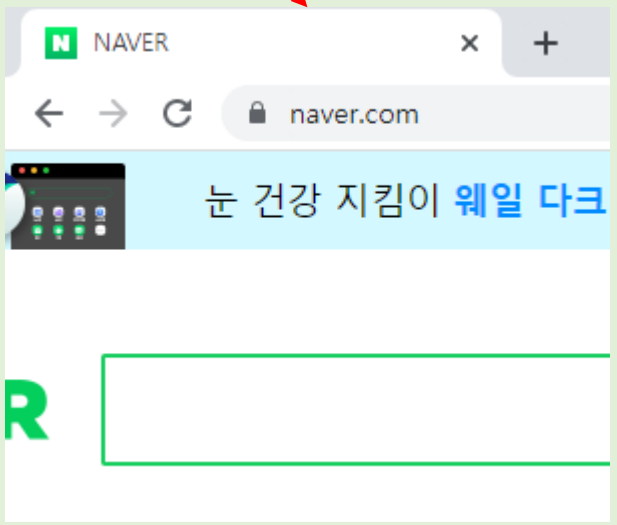
전달인자

- 콜론으로 표시되는 “전달인자” 사용가능

```
<body>
    <div id="app">
        <a v-bind:href="url">click me!</a></br>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            url : 'http://www.naver.com',
        }
    });
</script>
```



* 클릭시 해당 url 데이터에 있는 주소로 이동

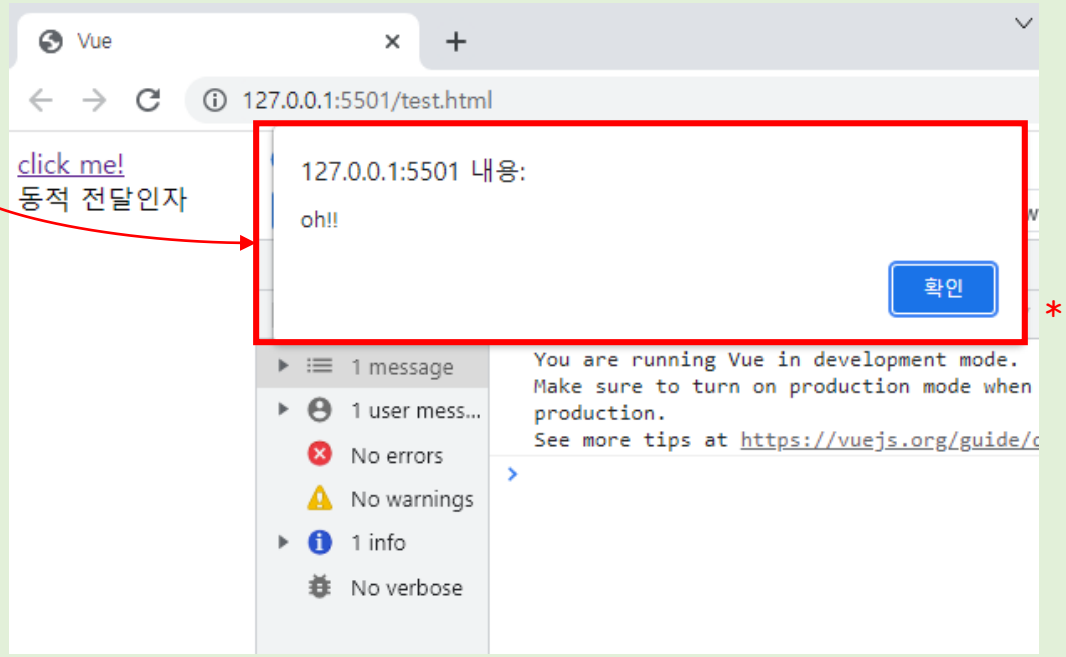


2. 디렉티브

동적 전달인자

- Javascript 표현식을 대괄호로 묶어 디렉티브의 argument 로 사용하는 것도 가능
- 동적 전달인자는 null 을 제외하고 String 으로 변환 (null 은 명시적으로 바인딩을 제거)
- 스페이스와 따옴표 같은 몇몇 문자들은 HTML 속성명으로 적합하지 않기 때문에 제한
- Vue.js 에서는 브라우저가 모든 속성명을 소문자로 만드는 관계로 대문자 사용 금지

```
<body>
    <div id="app">
        <a v-bind:href="url">click me!</a></br>
        <a v-on:[eventname]="doSomething">동적 전달인자</a>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            url : 'http://www.naver.com',
            eventname : 'click'
        },
        methods : {
            doSomething : function(event){
                alert('oh!!');
            }
        }

    });
</script>
```

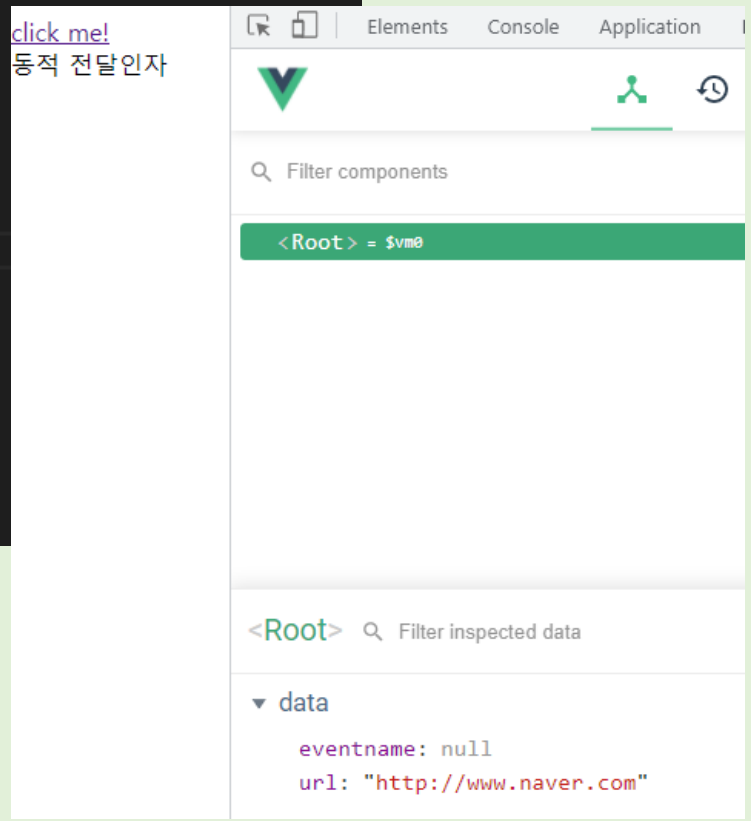


* 클릭시 해당 event 실행

2. 디렉티브

```
<body>
    <div id="app">
        <a v-bind:href="url">click me!</a></br>
        <a v-on:[eventname]="doSomething">동적 전달인자</a>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            url : 'http://www.naver.com',
            eventname : null
        },
        methods : {
            doSomething : function(event){
                alert('oh!!');
            }
        }

    });
</script>
```

* null 은 명시적으로 바인딩 제거

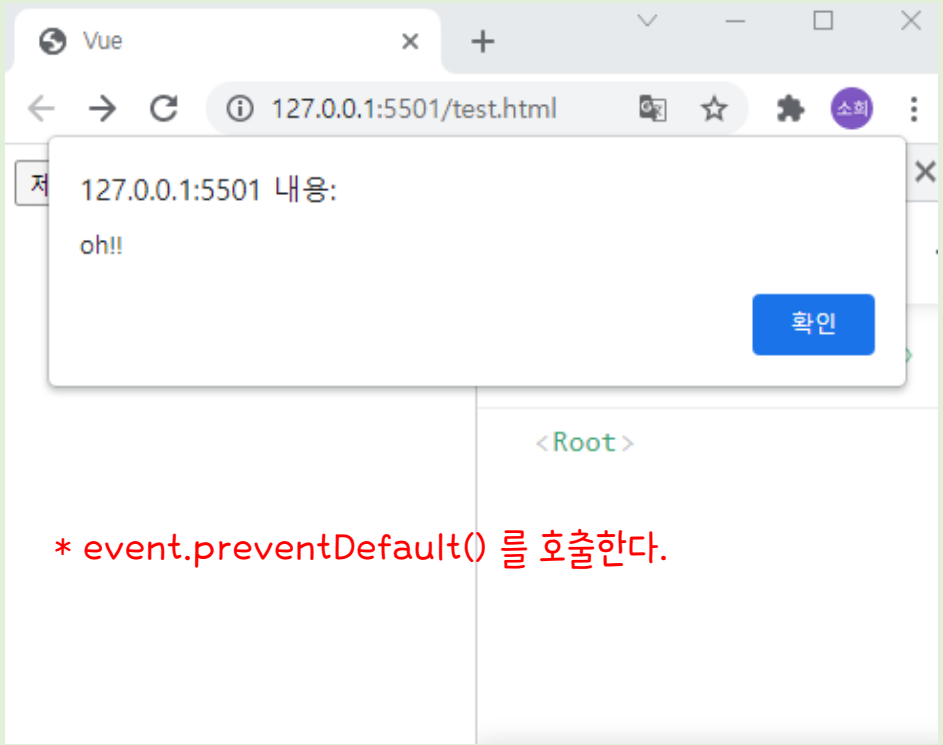


2. 디렉티브

수식어

- 점으로 표시되는 특수 접미사.
- 디렉티브를 특별한 방법으로 바인딩

```
<body>
    <div id="app">
        <form v-on:submit.prevent="doSomething" action="http://www.naver.com">
            <input type="submit">
        </form>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
        },
        methods : {
            doSomething : function(event){
                alert('oh!!');
            }
        }

    });
</script>
</html>
```



* event.preventDefault() 를 호출한다.

3. 약어

* Vue.js가 SPA(Single Page Application)를 만들 때 v- 접두어의 필요성이 떨어져서 가장 자주 사용되는 두개의 디렉티브에 대해 약어를 제공.

v-bind == :

```
<div id="app">
    <a v-bind:href="url">표준</a></br>
    <a :href="url">약어</a>
</div>
```

v-on == @

```
<body>
    <div id="app">
        <a v-on:click="doSomething">표준</a><br>
        <a @click="doSomething">약어</a><br>
        <a @[event]="doSomething">동적약어</a>
    </div>
</body>
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script>
    var vm = new Vue({
        el : '#app',
        data : {
            url : "http://www.naver.com",
            event : 'click'
        },
        methods : {
            doSomething : function(event){
                alert('oh!!');
            }
        }

    });
</script>
```

감사합니다.