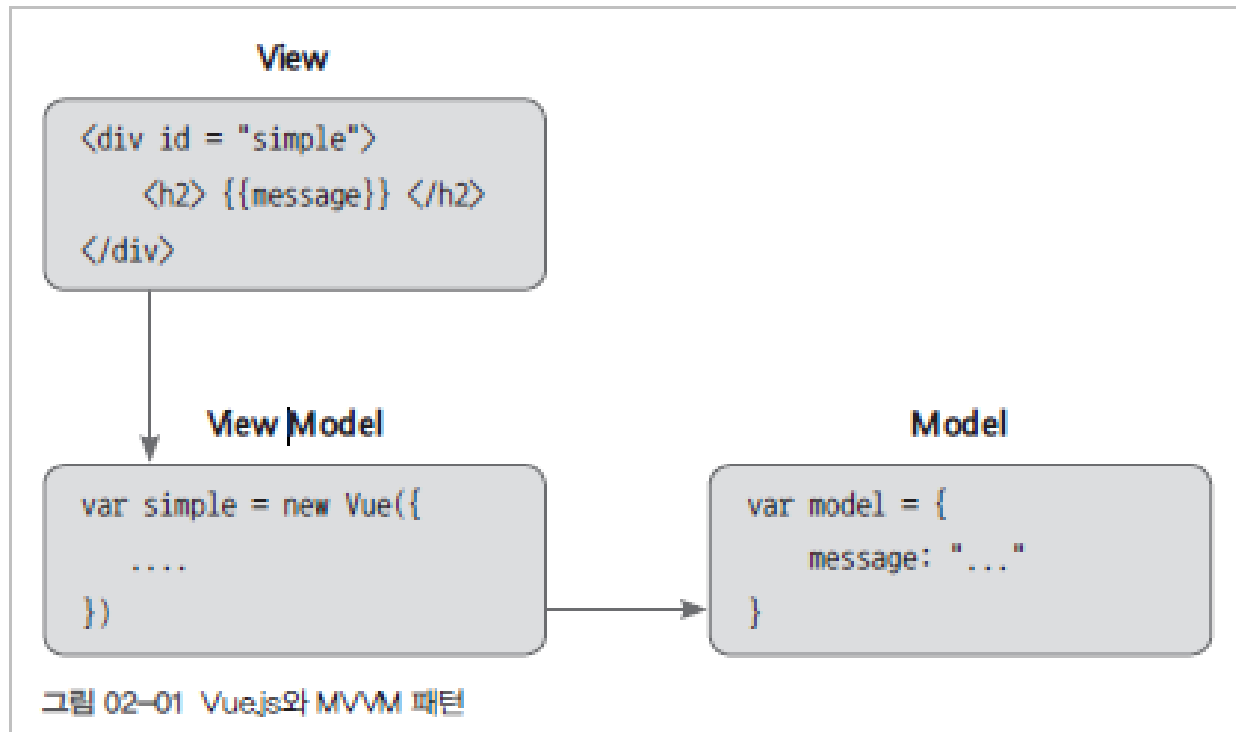


# helloworlds 분석



## ■ MVVM

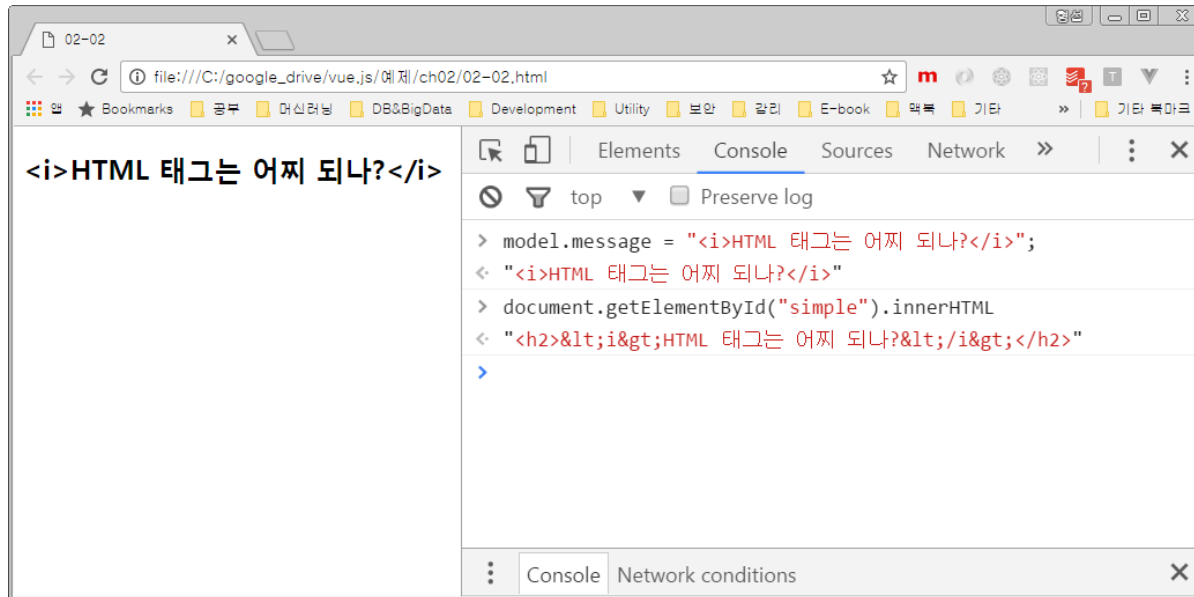


# 기본 디렉티브



## ■ v-text, v-html 디렉티브

- `<h2>{{message}}</h2>` === `<h2 v-text="message"></h2>`



**v-text, {{ }}**

innerHTML 속성에 연결된 태그 문자열을 HTML 인코딩하여 나타내기 때문에 화면에는 태그 문자열이 그대로 나타남.

**v-html**

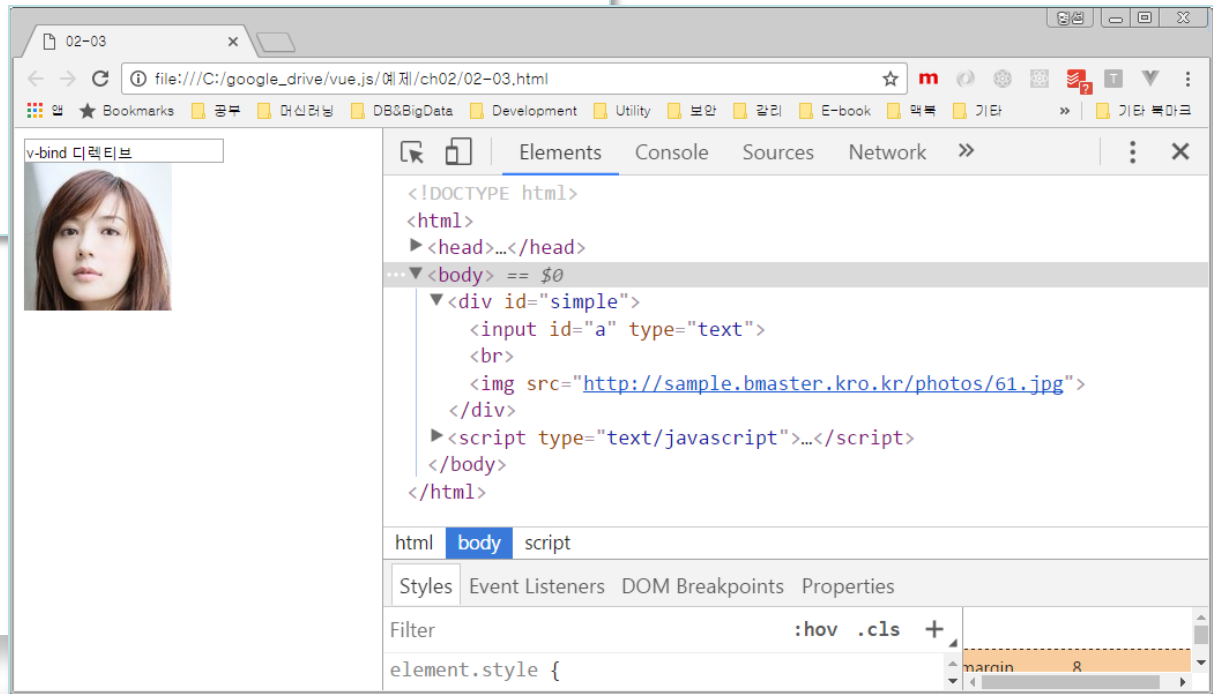
innerHTML 속성에 연결된 태그 문자열을 파싱하여 화면에 나타냄

# 기본 디렉티브



## ❖ v-bind 디렉티브

```
01: <div id="simple">
02:   <input id="a" type="text" v-bind:value="message">
03:   <br />
04:   <image v-bind:src="imagePath" />
05: </div>
06: <script type="text/javascript">
07:   var model = {
08:     message : 'v-bind 디렉티브',
09:     imagePath : "http://sample.bmaster.kro.kr/photos/61.jpg"
10:   };
11:
12:   var simple = new Vue({
13:     el : '#simple',
14:     data : model
15:   })
16: </script>
```



# 기본 디렉티브

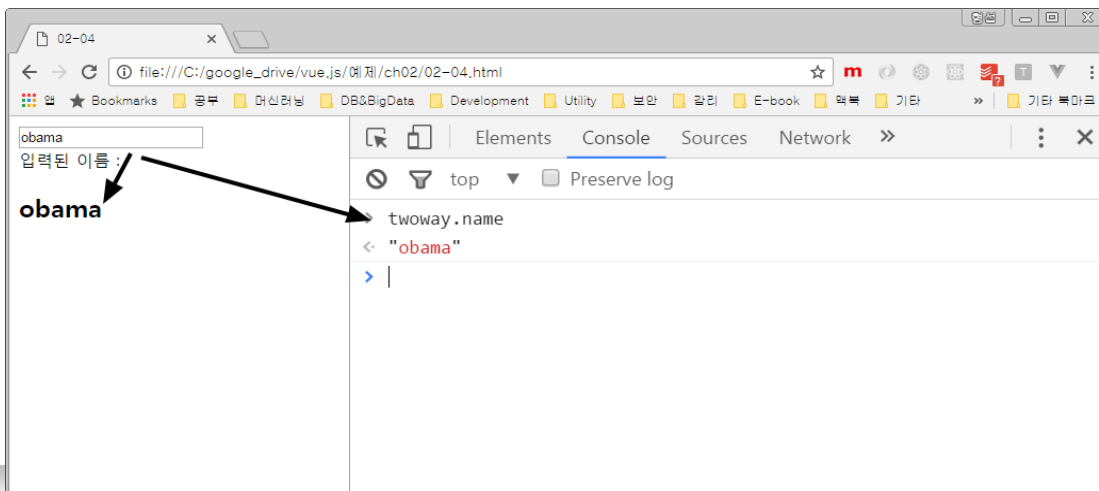


## ■ v-model 디렉티브

- 예제 02-03 활용
- v-bind 디렉티브는 단방향

```
01: <div id="simple">
02:   <input type="text" v-model="name" placeholder="이름을 입력하세요" />
03:   <br />
04:   입력된 이름 : <h2 v-html="name"> </h2>
05: </div>
```

- v-model 디렉티브는 양방향 바인딩

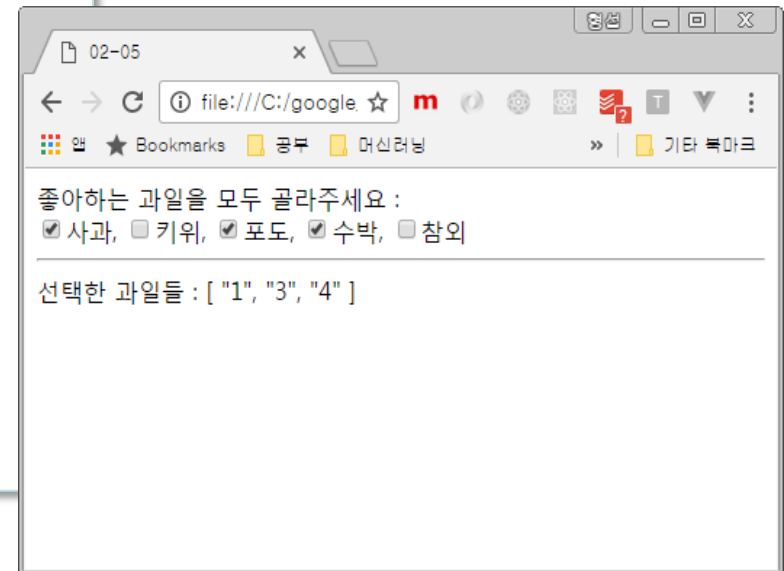
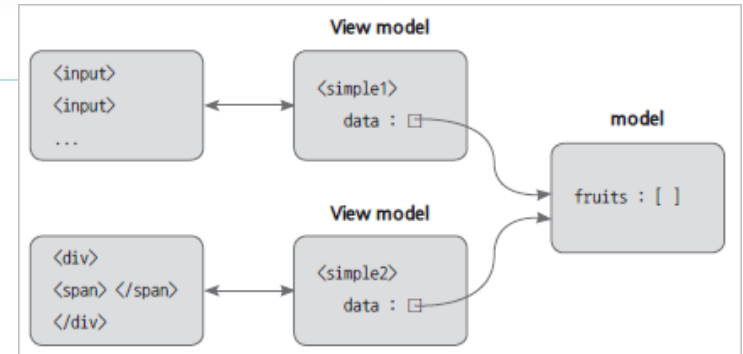


# 기본 디렉티브



## ■ v-model 디렉티브(이어서)

```
01: <div id="simple1">
02:   <div>좋아하는 과일을 모두 골라주세요 : </div>
03:   <input type="checkbox" value="1" v-model="fruits">사과,
04:   <input type="checkbox" value="2" v-model="fruits">키위,
05:   <input type="checkbox" value="3" v-model="fruits">포도,
06:   <input type="checkbox" value="4" v-model="fruits">수박,
07:   <input type="checkbox" value="5" v-model="fruits">참외
08: </div>
09: <hr />
10: <div id="simple2">
11:   선택한 과일들 : <span v-html="fruits"> </span>
12: </div>
13: <script type="text/javascript">
14:   var model = {
15:     fruits : []
16:   };
17:
18:   var simple1 = new Vue({
19:     el : '#simple1', data : model
20:   })
21:
22:   var simple2 = new Vue({
23:     el : '#simple2', data : model
24:   });
25: </script>
```



# 기본 디렉티브



## ■ v-show, v-if, v-else-if 디렉티브

- v-if : 바인딩한 값에 따라 렌더링 여부를 결정함
- v-show : 렌더링은 항상 수행함. display CSS 속성을 이용해 보여줄지만을 결정함.

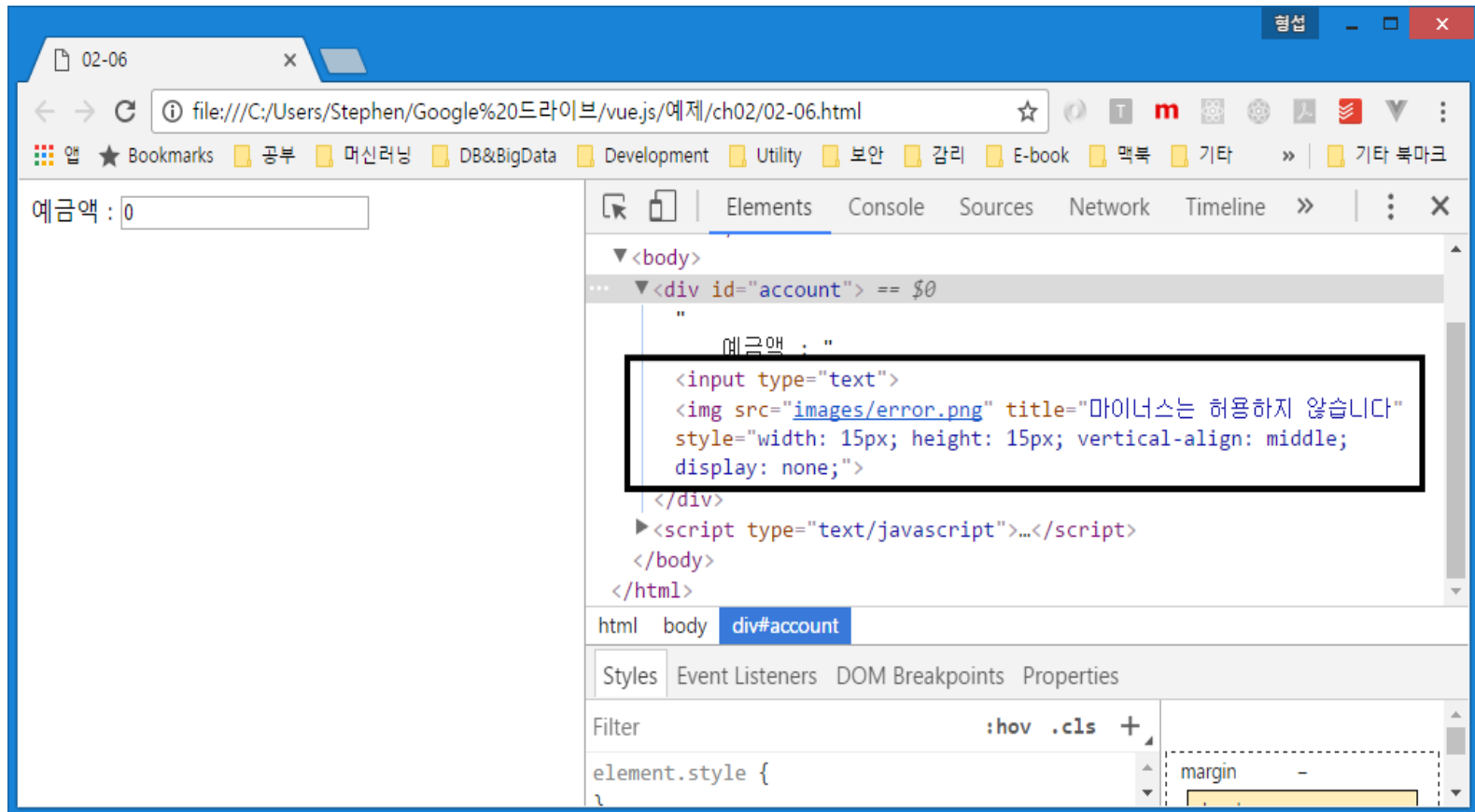
```
01: <div id="account">
02:   예금액 : <input type="text" v-model="amount" />
03:   
06: </div>
07: <script type="text/javascript">
08: var simple1 = new Vue({
09:   el : '#account',
10:   data : {
11:     amount : 0
12:   }
13: })
14: </script>
```

# 기본 디렉티브



## ❖ v-show, v-if, v-else-if 디렉티브(이어서)

- v-show로 변경했을 때

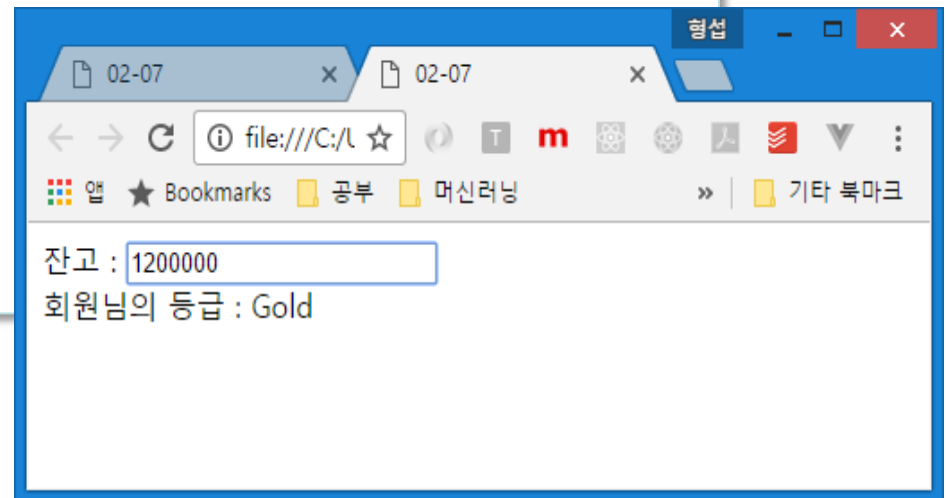


# 기본 디렉티브



## ❖ v-show, v-if, v-else-if 디렉티브(이어서)

```
01: <div id="account">
02:   잔고 : <input type="text" v-model="balance" />
03:   <br />
04:   회원님의 등급 :
05:   <span v-if="balance >= 1000000">Gold</span>
06:   <span v-else-if="balance >= 500000">Silver</span>
07:   <span v-else-if="balance >= 200000">Bronze</span>
08:   <span v-else>Basic</span>
09: </div>
10: <script type="text/javascript">
11:   var simple1 = new Vue({
12:     el : '#account',
13:     data : {
14:       balance : 0
15:     }
16:   })
17: </script>
```





# 반복 렌더링 디렉티브



## ■ v-for 디렉티브

```
<tbody id="contacts">
  <tr v-for="(contact, index) in contacts">
    <td>{{contact.no}}</td>
    <td>{{contact.name}}</td>
    <td>{{contact.tel}}</td>
    <td>{{contact.address}}</td>
  </tr>
</tbody>
```

```
var model = {
  "pageno": 1,
  "pagesize": 10,
  "totalcount": 100,
  "contacts": [
    { "no": 100, "name": "설현", "tel": "010-3456-8299", "address": "서울" },
    .....
  ]
}

var list = new Vue({
  el: "#exmaple",
  data: model
});
```

02-08

file:///C:/User

Bookmarks | 공부 | 머신러닝 | DB&BigData | 기타 북마크

번호	이름	전화번호	주소
100	설현	010-3456-8299	서울시
99	혜리	010-3456-8298	서울시
98	하니	010-3456-8297	서울시
97	성소	010-3456-8296	서울시
96	지아	010-3456-8295	서울시
95	정연	010-3456-8294	서울시
94	쯔위	010-3456-8293	서울시
93	사나	010-3456-8292	서울시
92	모모	010-3456-8291	서울시
91	소진	010-3456-8290	서울시

# 반복 렌더링 디렉티브



- 객체를 이용하는 v-for 디렉티브
  - `<option v-for="(val, key) in regions" v-bind:value="key">{{val}}</option>`
- 인덱스를 사용하는 v-for 디렉티브
  - 배열 데이터인 경우
    - `<tr v-for="(contact, index) in contacts"> ... </tr>`
  - 객체 데이터인 경우
    - `<option v-for="(val, key, index) in regions" ... >... </option>`
- v-for 와 다른 디렉티브 함께 사용
  - v-for가 먼저 적용된 후 다른 디렉티브가 적용됨
    - `<tr v-for="(contact, index) in contacts"`  
`v-if="contact.address.indexOf('서울')> -1">`

번호	이름	전화번호	주소
1	설현	010-3456-8299	서울
2	혜리	010-3456-8298	서울
5	지아	010-3456-8295	서울
8	사나	010-3456-8292	서울
10	소진	010-3456-8290	서울

# 반복 렌더링 디렉티브



## ▪ <template>으로 여러개 요소 묶기

```
01: <tbody id="contacts">
02:   <template v-for="(contact, index) in contacts">
03:     <tr>
04:       <td>{{contact.no}}</td>
05:       <td>{{contact.name}}</td>
06:       <td>{{contact.tel}}</td>
07:       <td>{{contact.address}}</td>
08:     </tr>
09:     <tr class="divider" v-if="index % 5 === 4">
10:       <td colspan="4"></td>
11:     </tr>
12:   </template>
13: </tbody>
```

```
.divider { height:2px; background-color: gray; }
```

번호	이름	전화번호	주소
100	설현	010-3456-8299	서울
99	혜리	010-3456-8298	서울
98	하니	010-3456-8297	경기
97	성소	010-3456-8296	제주
96	지아	010-3456-8295	서울
95	정연	010-3456-8294	강원
94	쯔위	010-3456-8293	경기
93	사나	010-3456-8292	서울
92	모모	010-3456-8291	충남
91	소진	010-3456-8290	서울

# 반복 렌더링 디렉티브



## ❖ v-for 디렉티브 적용시 key 특성

- 가상 DOM은 빠른 렌더링을 위해 변경된 부분만을 업데이트함.
- 렌더링된 배열 데이터의 순서가 변경되면 DOM 요소를 이동시키지 않고 기존 DOM 요소의 데이터를 변경함
- 만일 DOM 요소를 추적하여 DOM 요소의 위치를 직접 변경하고자 한다면 DOM 요소에 key 특성을 바인딩할 수 있음
  - 반드시 고유한 값으로...

## ❖ 배열 데이터 변경에 관한 문제점

- 배열 데이터를 인덱스 번호를 이용해 직접 변경하는 경우

# 기타 디렉티브



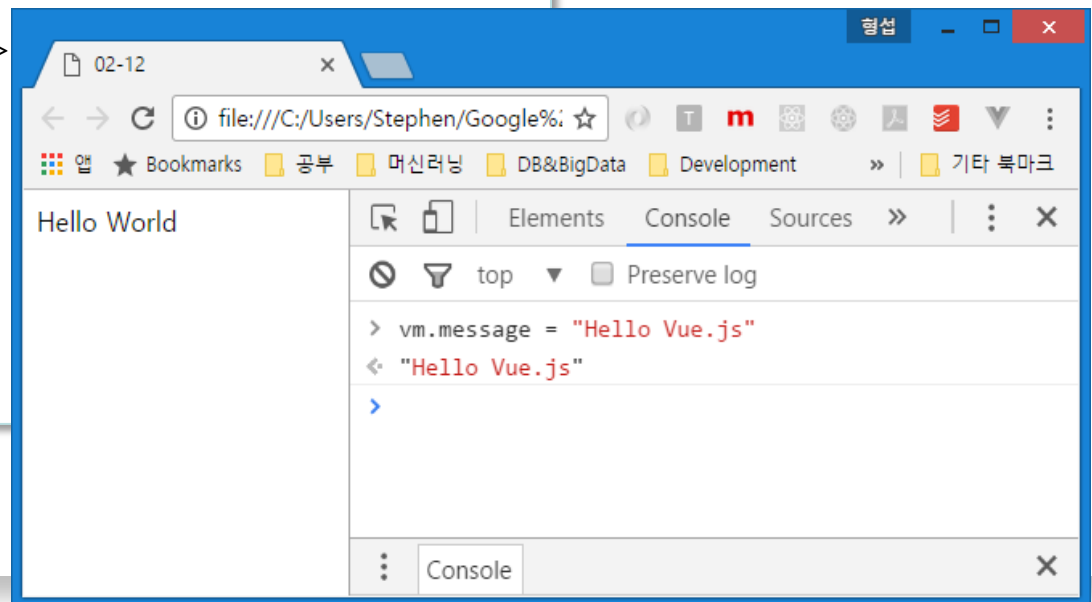
## ■ v-pre 디렉티브

- HTML 요소에 대한 컴파일을 수행하지 않음
- {{}} 표현식이 적용되지 않음

## ■ v-once 디렉티브

- 단 한번만 렌더링하도록 설정

```
01: <div id="example">
02:   <span v-once>{{message}}</span>
03: </div>
04: <script type="text/javascript">
05: var vm = new Vue({
06:   el : "#example",
07:   data : {
08:     message : "Hello World"
09:   }
10: })
11: </script>
```



# 기타 디렉티브



## ■ v-cloak 디렉티브

- el 옵션의 템플릿을 컴파일할 때 소요되는 시간 동안 {}이 일시적으로 화면에 나타날 수 있음
- 이러한 현상을 막기 위해 v-cloak 디렉티브 사용
- 초기에 컴파일되지 않은 템플릿은 나타나지 않도록...

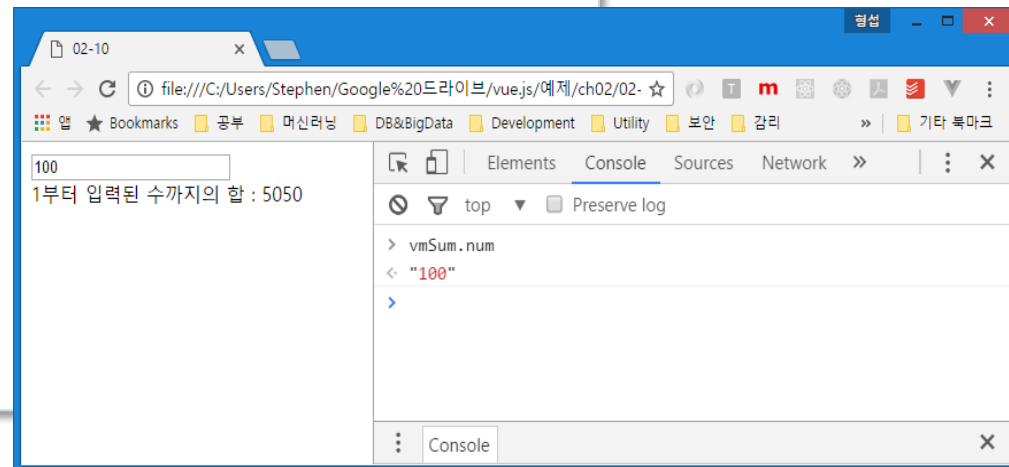
# 계산형 속성



## ■ Computed Property

- Vue 인스턴스를 만들때 computed 옵션을 설정하고 함수를 등록하면 마치 속성처럼 이용할 수 있음

```
01: <div id="example">
02:   <input type="text" v-model="num" /> <br />
03:   1부터 입력된 수까지의 합 : <span>{{sum}}</span>
04: </div>
05: <script type="text/javascript">
06: //1부터 입력된 수까지의 합구하기
07: var vmSum = new Vue({
08:   el : "#example",
09:   data : { num : 0 },
10:   computed : {
11:     sum : function() {
12:       var n = Number(this.num);
13:       if (Number.isNaN(n) || n < 1) return 0;
14:       return ((1+n) * n) / 2;
15:     }
16:   }
17: });
18: </script>
```



# 계산형 속성



## 주의 사항

- 계산형 속성의 함수 내부에서의 this는 객체 자신을 참조하지만 함수내부에서 다른 콜백함수를 실행하거나 했을 경우 this는 다른 값이 될 수 있음
- HTML 요소 내부에서는 모두 문자열로 다루어짐.  
--> 명시적으로 형변환해주어야 함.
- .number와 같은 수식어를 사용할 수도 있음
  - v-model.number="num"



# 계산형 속성



## 예제 02-14~16

```
24: var clist = new Vue({
25:   el : "#exmaple",
26:   data : model,
27:   computed : {
28:     filtered : function() {
29:       var cname = this.countryname.trim();
30:       return this.countries.filter(function(item,index) {
31:         if (item.name.indexOf(cname) > -1) {
32:           return true;
33:         }
34:       });
35:     }
36:   }
37: });
```

```
01: 국가명 : <input type="text" v-model="countryname" placeholder="국가명" />
02: .....
03: <tbody id="contacts">
04:   <tr v-for="c in filtered">
05:     <td>{{c.no}}</td>
06:     <td>{{c.name}}</td>
07:     <td>{{c.capital}}</td>
08:     <td>{{c.region}}</td>
09:   </tr>
10: </tbody>
```

