

2025년 상반기 K-디지털 트레이닝

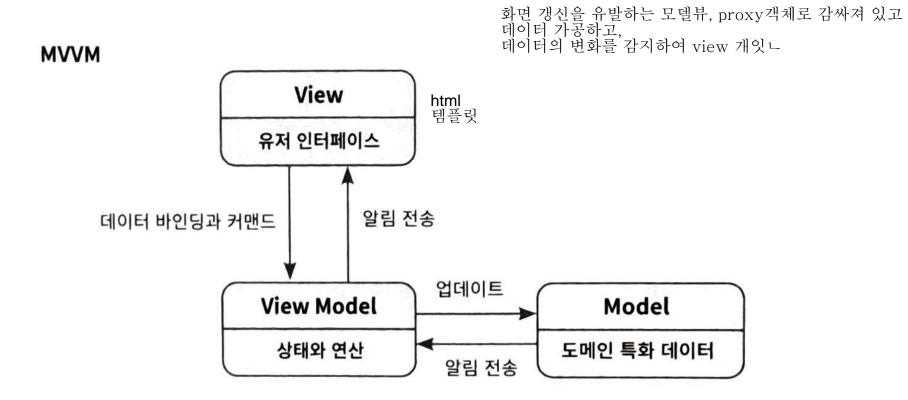
Vue.js 기초와 Template

[KB] IT's Your Life

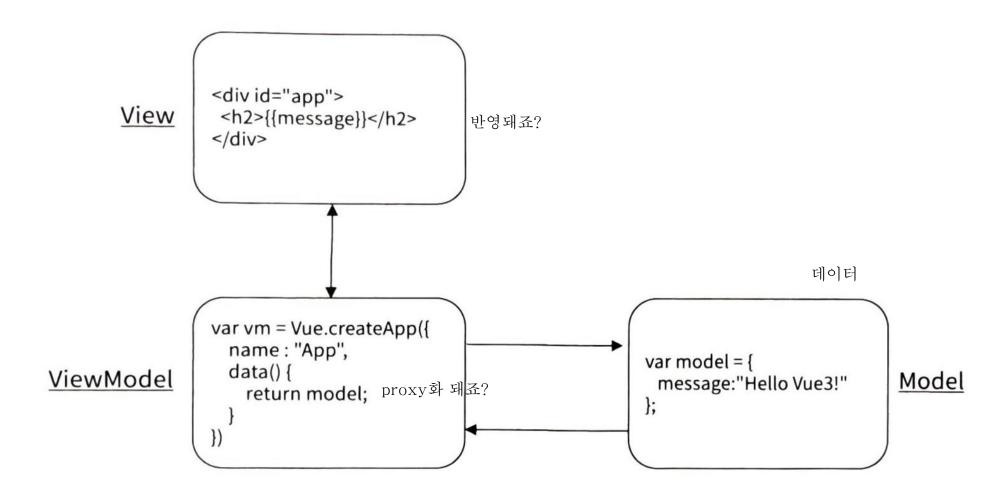


MVVM 패턴

Model-View-ViewModel



✓ Vue.js와 MVVM 패턴



Vue 디렉티브

directive : 지시자՝

- V-로 시작하는 속성
- o HTML 요소와 관련된 작업을 지정 html의 요소의 속성으로 표현한다

👂 v-text, v-html 디렉티브

.innerHTML과 .textContent 라는 기본 속성이 기억나는가?

vue에서는 v-html와 v-text 이 가각의 역할드을 한다

ㅇ 텍스트 출력

- v-text, {{}}: 내용을 text로 출력, html 태그 문자가 있는 경우 이스케이프됨
- v-html: 내용을 html로 출력, html 태그 문자가 있는 경우 html 태그로 해석

표준 속성의 값은 문자열이다.

하지만 vue가 제공하는 v-text, t-html 디렉티브 속성의 값은 js의 표현식이다! js의 문법이 적용된다. 속성의 값으로!

실제로 부 디렉티브 속성의 값을 문자열로 사용하고 싶다면 따옴표로 한번더 감싸야한다

```
<div id="app">
                     <h2 v-text="message"></h2>
                   </div>
                v-text속성 사용
<div id="app">
  <div v-text="message"></div>
</div>
<script src="https://unpkg.com/vue"></script>
<script>
  let model = { message: '<h2>hello vue3</h2>' };
                                                   결과는
  let vm = Vue.createApp({
                                                   <h2>hello</h2>
문자열 그대로
출력됨.
    name: 'App',
    data: function () {
      return model;
  }).mount('#app');
//comints
```

```
<h2>{{message}}</h2> 랑 같다
                                v-html 속성 사용
                <div id="app">
                  <div v-html="message"></div>
                </div>
                <script src="https://unpkg.com/vue"></script>
                <script>
                  let model = { message: '<h2>hello vue3</h2>' };
                  let vm = Vue.createApp({
                    name: 'App',
                    data: function () {
                      return model;
                  }).mount('#app');
```

2 03-01.html

```
<div id="app">
 <h2>{{message}}</h2>
</div>
<script type="text/javascript" src="https://unpkg.com/vue"></script>
<script type="text/javascript">
 var model = { message: "Hello Vue3!" };
 var vm = Vue.createApp({
   name: "App",
   data() {
     return model;
   },
 }).mount("#app");
</script>
```

v-bind 디렉티브

예) 해당 src속성값에 js변수를 사용하고 싶다면? => v-bind 디렉티브 사용해라

- 요소의 속성을 바인딩
- 속성값을 문자열이 아닌 javascript 표현식으로 처리됨
- 형식
 - **<태그명 v-bind:속성="표현식">** 표현식은 js 표현식을 사용하면 된다.
 - <태그명 :속성="표현식">

○ v-bind의 특징

- 단방향
 - Vue 인스턴스의 데이터나 속성이 바뀌면 UI를 갱신
 - 화면의 바인딩된 요소에서 값을 변경하더라도 데이터는 변경되지 않음

뷰 모델에서의 변화는 ui에 적용되지만

ui에서의 변화가 뷰모델에는 적용되지 않음.

정리:

요소의 속성 값에 문자열이 아닌 js의 변수 정보를 입력시키고 싶다면 v-bind

요소의 textContent, 텍스트를 넣고 싶은데 js의 변수 정보를 입력시키고 싶다면 v-text

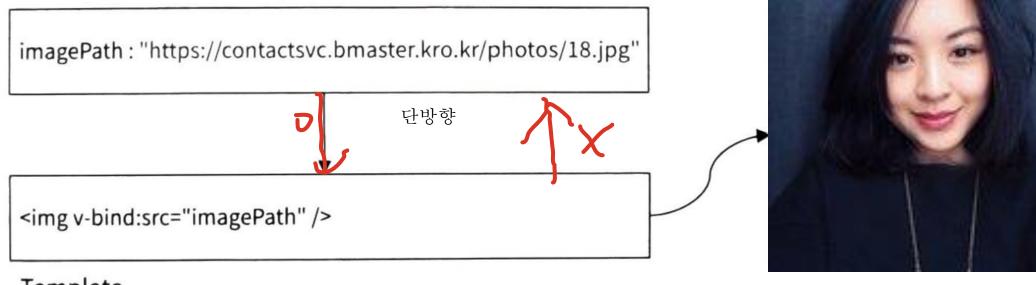
inner HTML 은 v-html로.

사용될 변수들은 객체로 감싸소 반환

☑ 03-03.html v-bind 디렉티브

```
value속성에다가 값으로 message라는 js변수 message변수를 연결하고 싶음.
<div id="app">
   <input type="text" v-bind:value="message">
   <br/>
   <img v-bind:src="imagePath"/> src속성에다가 값으로 js변수 imagPath변수를 연결
</div>
<script src="https://unpkg.com/vue"></script>
<script>
   let vm = Vue.createApp({
       name: "App",
       data() {
           return {
                                                                          변수들은 객체에 묶어서 보내기
               message: "v-bind 디렉티브",
               imagePath: "https://contactsvc.bmaster.kro.kr/photos/18.jpg"
           };
   }).mount('#app');
</script>
```

Vue 인스턴스의 데이터



Template

축약 표현

축약 표현이 몇개 나옴. 햇갈리지 말자

o v-bind:src → :src

FORM처리

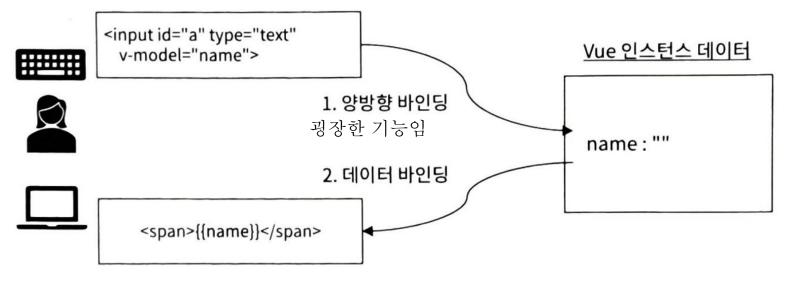
🗸 v-model 기본 사용법

- 양방향 데이터 바인딩 ✔
 - 속성값 변경시 → UI 변경됨
 - UI 입력값 변경시 → 속성값 변경됨
- 주로 form 요소에 바인딩
 - input, select, textarea 등
- ㅇ 형식
 - <태그명 v-model="속성"> 축약 없음!

2 03-04.html v-model 디렉티브

```
v-model은 해당 ipnut태그의 value속성과 관련있음.
value속성의 값을 name js변수와 연결.____
<div id="app">
   <input type="text" v-model="name">
   <br/>
   입력한 값: <span>{{name}}</span>
                v-text="name">랑 같음
</div>
<script src="https://unpkg.com/vue"></script>
<script>
   let vm = Vue.createApp({
       name: "App",
       data() {
   </script>
```

홍길동 입력한 값: 홍길동



2 03-05.html

v-model 디렉티브

```
<div id="app">
                                                   그냥
입력이
                               여기서는
   <div>
                              v-model은
value말고 어떤 걸까
                                                   v-model에 해당
된다고 생각하자
        <h2>취미생황</h2>
        <label>
            <input type="checkbox" value="A" v-model="hobby">
            운동
        </label>
        <label>
           <input type="checkbox" value="B" v-model="hobby">
            독서
        </label>
        <label>
            <input type="checkbox" value="C" v-model="hobby">
            음악
        </label>
        <label>
           <input type="checkbox" value="D" v-model="hobby">
            댄스
        </label>
        <label>
           <input type="checkbox" value="E" v-model="hobby">
            역사
        </label>
   </div>
```

다중 선택이 가능하므로 hobby는 배열이어야 함

☑ 03-05.html v-model 디렉티브

```
<div>
       <h2>상품 분류 선택</h2>
       <select v-model="category">
          <option value="">----상품 분류를 선택하세요.</option>
          <option value="C01">레저</option>
          <option value="C02">가전</option>
                                                               select태그에 multiple 속성이 없으므로
          <option value="C03">음식</option>
                                                               단일값.
          <option value="C04">도서</option>
          <option value="C05">주방</option>
       </select>
   </div>
   <hr/>
   <div>
       선택한 취미: {{hobby.join(',')}} <br/>
       선택한 상품 분류: {{category}}
   </div>
</div>
```

2 03-05.html v-model 디렉티브



checkbox의 v-model 바인딩

- 다중 선택 시 → 배열 속성에 바인딩
- 단일 선택 시 → 단일 속성에 바인딩
 - 디폴트 값: true/false
 - true-value, false-value로 값 지정 가능

2 03-06.html

```
<div id="app">
   위의 내용에 동의하십니까?
   <input type="checkbox" v-model="agree" true-value="yes" false-value="no">
   <hr/>
   <span>선택된 값 : {{agree}}</span>
</div>
                                                 위의 내용에 동의하십니까? ☑
<script src="https://unpkg.com/vue"></script>
                                                 선택된 값 : yes
<script>
   let vm = Vue.createApp({
       name: "App",
       data() {
           return {
              agree: 'no' 초기값
           };
   }).mount('#app');
</script>
```

수식어(modifier)

o v-model 디렉티브에 특별한 기능을 추가

o lazy

- input에서 엔터를 치거나 <u>포커스 이동했을 때</u> 입력값과 속성을 동기화
- <input type="text" v-model.lazy="name" />

o **number**

- 문자열을 숫자로 자동 형변환시켜 속성에 반영
- <input type="text" v-model.number="num" />

변환하고 속성에 반영

o trim

- 문자열의 앞뒤 공백을 자동으로 제거
- <input type="text" v-model.trim="message" />

2 03-07.html

```
<div id="app">
   <input type="text" v-model.number="num"/><br/>>
    입려된 수: <span>{{num}}</span>
</div>
<script src="https://unpkg.com/vue"></script>
<script>
    let vm = Vue.createApp({
        name: "App",
        data() {
            return {
                num: 0
            };
   }).mount('#app');
</script>
```

입력에는 제한이 없죠? 문자써도 돼죠? 하지만 js 변수에 저장되기전에 형변환하죠? .number니까?

그러니까 넘어가는 값은 12. 12가 js변수에 저장된다

12abc

입려된 수: 12

v-model의 한글 처리 문제

- 값을 받아내는 시점이 한글 <u>한 글자가 입력</u>이 완료될 때 처리됨
- 정확히는 다음 글자로 넘어갈때 js변수로 넘어감. 이건 다른 프레임웤도 그럼

- 해결방법
 - v-model을 사용하지 않음
 - v-bind를 이용해서 바인딩
 - 입력 이벤트(input)가 발생했을 때 이벤트 핸들러에서 속성 데이터에 반영

자세히는 다음 자료에

정리2

3 v-model 디렉티브

2 03-08.html

```
v-text, {{ }} textContent
                                       입력이 있을 때 이벤트 핸들러 지정'
                                                                         v-html innerHtml 같은
                        v-bind로
<div id="app">
                                                                         v-bind 단뱡향
                                                                         v-model 양방향 보통 form요소에서
   <input type="text" :value="name" @input="changeName"/>
   <br/>
                                                                         보통 일반 태그의 속성은 v-bind
   이름: <span>{{name}}</span>
                                                                         form택의 속성에서는 v-model
</div>
<script src="https://unpkg.com/vue"></script>
<script>
   let vm = Vue.createApp({
       name: "App",
       data() {
           return { name: '' };
       },
                                        메써드 필드 라는 곳에 이벤트 함수
       methods: {
                                                                   화살표 함수
           changeName(e)
                                                                   안돼 this가 전역객체가 되어버림
              this.name = e.target.value;
                                                   this는 누고? addEvent로 할 때는 발생시킨 태그요소객체인데.
                                                   뷰에서는 vm 이다.
   }).mount('#app');
</script>
```

잊지말자 v-xxx="js표현식"

∨−show

- <u>화면에 보여줄지 말지를 결정하는 디렉티브</u>
- HTML 요소는 생성하지만 화면에 보여줄지 않을 수 있음

dom트리에 변경X

2 03-09.html

v-show 디렉티브

```
v-show의 값으로 true인 표현식이면
보여주고 아니명 안보여주고
<div id="app">
    예금액: <input type="text" v-model="amount"/>
    <img v-show="amount < 0" src="https://contactsvc.bmaster.kro.kr/img/error.png"</pre>
         title="마이너스는 허용하지 않습니다."
         style="width:15px; height:15px; vertical-align:middle"/>
</div>
<script src="https://unpkg.com/vue"></script>
<script>
    let vm = Vue.createApp({
                                                                                  K [0
                                                                                                  Console Sources >>
                                          예금액: 100
                                                                                          Elements
         name: "App",
                                                                                   <!DOCTYPE html>
         data() {
                                                                                   <html lang="ko">
                                                                                   <head> 
              return { amount: '' };
                                                                                   ▼ <body>
                              문자열로 저장하게금 설정댔네
                                                                                    ▼ <div id="app" data-v-app>
                                                                                       " 예금액: "
    }).mount('#app');
                                                                                       <input type="text">
</script>
                                                                                       <img src="https://contactsvc.bmaster.kro.kr/img/error.png"</pre>
                                                                                       title="마이너스는 허용하지 않습니다." style="color: red; width:
                                                              문자열 < 0
는 자동변환되어
비교연산 잘돼
                                                                                       15px; height: 15px; vertical-align: middle; display: none;
                                                                                       "> == $0
    예금액: -100
                                                                                      </div>
                                                                                      <script src="https://unpkg.com/vue"></script>
                                                                                     ▶<script> ····
```

- ∨-if
 - 조건에 부합되지 않을 경우 요소를 생성하지 않음(화면 출력 없음)

아예 생성하지 않아서 DOM TRee에 들어가지 않게함

dom 트리에 변경을 가함

☑ 03-10.html v-if 디렉티브

```
<div id="app">
    예금액: <input type="text" v-model="amount"/>
    <img v-if="amount < 0" src="https://contactsvc.bmaster.kro.kr/img/error.png"</pre>
        title="마이너스는 허용하지 않습니다."
        style="width:15px; height:15px; vertical-align:middle"/>
</div>
<script src="https://unpkg.com/vue"></script>
<script>
    let vm = Vue.createApp({
        name: "App",
                                                                                   K [0
                                                                                           Elements Console Sources >>
                                           예금액:
        data() {
                                                                                    <!DOCTYPE html>
            return { amount: '' };
                                                                                    <html lang="ko">
                                                                                    ▶ <head> ···· </head>
    }).mount('#app');
                                                                                    ▼<body>
                                                                                   " ▼ <div id="app" data-v-app> == $0
</script>
                                                                                        " 예금액: "
                                                                                        <input type="text">
```

<!--v-if-->

▶ <script> · · · · · /script>

<script src="https://unpkg.com/vue"></script>

</div>

- v-else, v-else-if
 - o v-if와 함께 사용
 - o if문의 else, else if 에 해당

✓ 03-11.html v-else, v-else-if 디렉티브

```
<div id="app">
    잔고: <input v-model="balance"/>
    <br/>
    회원님의 등급:
   <span v-if="balance > 1000000">Gold</span>
    <span v-else-if="balance >= 500000">Silver</span>
    <span v-else-if="balance >= 200000">Bronze</span>
   \span v-else="balance">Basic</span>
</div>
                                                        잔고: 500000
                                                        회원님의 등급: Silver
<script src="https://unpkg.com/vue"></script>
<script>
    let vm = Vue.createApp({
        name: "App",
        data() {
            return { balance: 0 };
    }).mount('#app');
</script>
```

☑ v-for 디렉티브

- 반복적인 데이터를 렌더링
- 형식(배열 사용시)
 - <태그명 v-for="변수 in 배열" :key="id값">

v-bind

배열의 요소

key라는 속성과

변수로

☑ 03-12.html v-for 디렉티브

```
<style>
  #list {width: 600px; border:1px solid black; border-collapse: collapse;}
  #list td, #list th { border:1px solid black; text-align: center;}
  #list > thead > tr { color: yellow; background-color: purple;}
</style>
<div id="app">
  <thead>
       번호이름전화번호
       </thead>
     {{contact.no}}
                                           해당 태그 묶음을
          {{contact.name}}
                                           배열의 요소들 순회하며 반복 생성
          {{contact.tel}}
       </div>
```

☑ 03-12-2.html v-for 디렉티브

```
<script src="https://unpkg.com/vue"></script>
<script>
   let vm = Vue.createApp({
       name: "App",
       data() {
           return {
               contacts: [
                   {no: 1011, name: 'RM', tel: '010-3456-8299'},
                   {no: 1012, name: '정국', tel: '010-3456-8298'},
                   {no: 1013, name: '제이홉', tel: '010-3456-8297'},
                   {no: 1014, name: '슈가', tel: '010-3456-8296'},
           };
   }).mount('#app');
</script>
```

번호	이름	전화번호
1011	RM	010-3456-8299
1012	정국	010-3456-8298
1013	제이홉	010-3456-8297
1014	슈가	010-3456-8296

v-for 디렉티브

- 형식(객체 사용시)
 - < <태그명 v-for="(val, key) in 객체":key="key">

내용 키 프로퍼티명 프로퍼니 내용

☑ 03-13.html 객체에 대한 v-for

```
<div id="app">
   <select id="regions">
        <option disabled="disabled" selected>지역 선택</option>
        <option v-for="(val, key) in regions" _:value="key" :key="key">{{val}}</option>
    </select>
                          키 내용.
</div>
                          이름을 햇갈리게 지정했네
<script src="https://unpkg.com/vue"></script>
<script>
    let vm = Vue.createApp({
        name: "App",
                                                                               K [0
                                                                                              Console Sources >>
                                                                                                                  (£)
                                                                                       Elements
                                                  지역 선택 🗸
        data() {
                                                                               <!DOCTYPE html>
           return {
                                                  Asia
                                                                               <html lang="ko">
                                                  America
                                                                                <head> ···· </head>
                regions: {
                                                  Europe
                                                                                ▼ <body>
                    A : 'Asia',
                                                  Africa
                                                                                 ▼ <div id="app" data-v-app>
                    B: 'America',
                                                  Oceania
                                                                                  ▼<select id="regions"> == $0
                    C: 'Europe',
                                                                                     <option disabled>지역 선택</option> (문 slot)
                                                                                     D: 'Africa',
                                                                                     <option value="B">America</option> ( slot )
                    E: 'Oceania'
                                                                                     value로
                         옵션태그의 내용으로
                                                                                     <option value="E">Oceania</option> ( slot )
           };
                                                                                   </select>
                                                                                  </div>
   }).mount('#app');
</script>
```

🗸 v-for 디렉티브

○ 인덱스 번호가 필요한 경우

🧑 형식(배열 사용시)

• <태그명 v-for="(contact, index) in contacts" ...>

- 형식(객체 사용시)
 - <태그명 v-for="(val, key, index) in regions" ...>


```
<style>
  #list {width: 600px; border:1px solid black; border-collapse: collapse;}
  #list td, #list th { border:1px solid black; text-align: center;}
  #list > thead > tr { color: yellow; background-color: purple;}
</style>
<div id="app">
  <thead>
        번호이름전화번호
       </thead>
     {{index+1}}
          {{contact.name}}
          {{contact.tel}}
                                     번호
                                              이름
                                                               전화번호
       RM
                                                             010-3456-8299
     정국
                                      2
                                                             010-3456-8298
  제이홉
                                                             010-3456-8297
</div>
                                              슈가
                                                             010-3456-8296
```

☑ 03-14.html v-for에서 index 얻기

```
<script src="https://unpkg.com/vue"></script>
<script>
 let vm = Vue.createApp({
   name: 'App',
   data() {
     return {
        contacts: [
         { no: 1011, name: 'RM', tel: '010-3456-8299' },
         { no: 1012, name: '정국', tel: '010-3456-8298' },
          { no: 1013, name: '제이홉', tel: '010-3456-8297' },
          { no: 1014, name: '슈가', tel: '010-3456-8296' },
        ],
     };
 }).mount('#app');
</script>
```

- 여러 요소를 묶어서 반복 렌더링하기
 - <template> 요소에 v-for 디렉티브 사용
 - <template v-for="...">

✓ 03-15.html < template > 태그

```
<style>
  #list {width: 600px; border:1px solid black; border-collapse: collapse;}
  #list td, #list th { border:1px solid black; text-align: center;}
  #list > thead > tr { color: yellow; background-color: purple;}
  .divider { height: 2px; background-color: gray;}
</style>
<div id="app">
  <thead>
        번호이름전화번호
        </thead>
     <template v-for="(contact, index) in contacts" :key="contact.no">
           ≺tr>
              {{contact.no}}
              {{contact.name}}
              {{contact.tel}}
           4줄당 한번 비어잇는 라인 하나 추가(경계선) 추가
              </template>
     </div>
```



```
<script src="https://unpkg.com/vue"></script>
<script>
   let vm = Vue.createApp({
       name: "App",
       data() {
           return {
               pageno: 1,
               pageSize: 4,
               totalcount: 100,
               contacts: [
                   {no: 1011, name: 'RM', tel: '010-3456-8299'},
                   {no: 1012, name: '정국', tel: '010-3456-8298'},
                   {no: 1013, name: '제이홉', tel: '010-3456-8297'},
                   {no: 1014, name: '슈가', tel: '010-3456-8296'},
                   {no: 1015, name: '진', tel: '010-3456-8295'},
                   {no: 1016, name: '뷔', tel: '010-3456-8294'},
                    {no: 1017, name: '지민', tel: '010-3456-8293'},
           };
   }).mount('#app');
</script>
```

번호	이름	전화번호
1011	RM	010-3456-8299
1012	정국	010-3456-8298
1013	제이홉	010-3456-8297
1014	슈가	010-3456-8296
1015	진	010-3456-8295
1016	뷔	010-3456-8294
1017	지민	010-3456-8293

v-for 디렉티브와 key 특성

- 배열을 렌더링할 때 데이터 변경없이 위치가 바뀌는 경우
 - ▶ key 속성이 없으면 → 전부 다시 렌더링
 - key 속성이 있으면 → 위치만 변경
- → key 특성에는 인덱스 번호를 부여하지 않음. 반드시 고유한 변경되지 않는 값을 부여

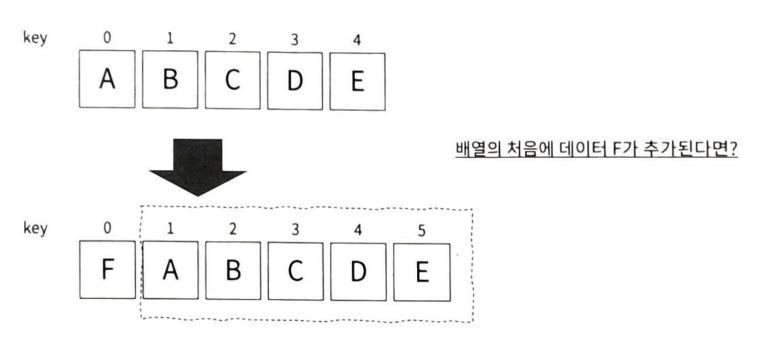
수정 전과 후를 인식못해서 그럼..

키가 있으면 변화를 인식할 수 있으니 수정된 녀석만 업데이트를 함 위치변경도 포함됨

♡ v-for 디렉티브와 key 특성

o key 특성에 인덱스 번호를 부여한 경우





인덱스가 바뀌면서 key가 모두 변경되므로 모두 렌더링(reflow, repaint)을 시도함.

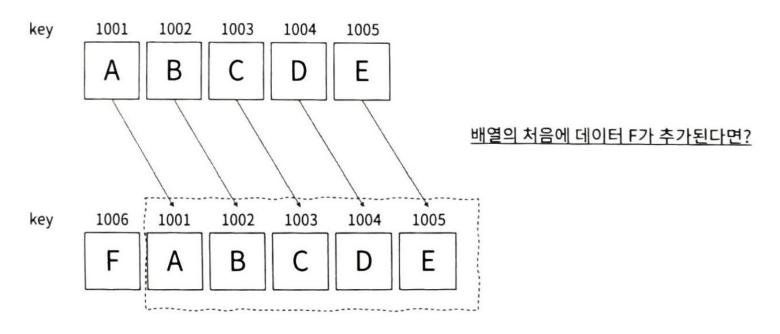
🗸 v-for 디렉티브와 key 특성

o key 특성에 고유 값을 부여한 경우

리액트는 참조값 변경됐을때 감지하지만 부는 배열의 요소의값이 변경되면 감지된다.

Key에 고유키를 부여하는 경우

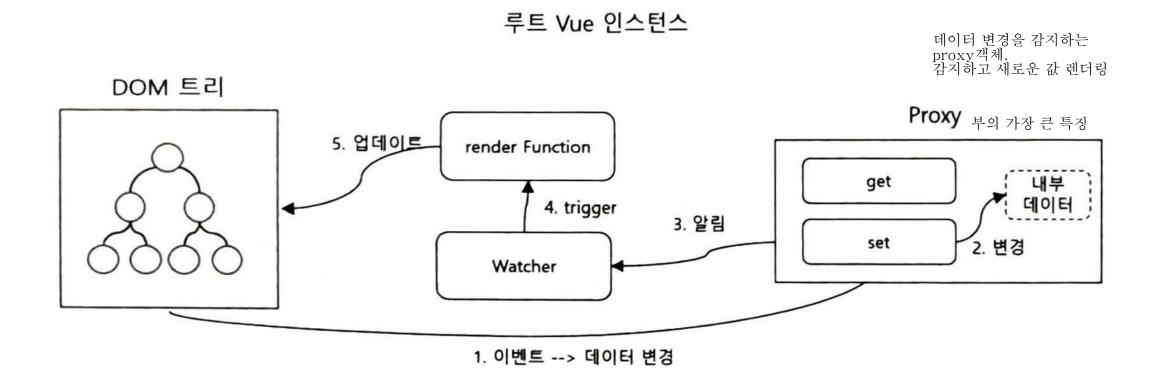
v-for 사용시에는 웬만하면 고유값을 붙이는게 좋음



A,B,C,D,E는 key가 변경되지 않았기 때문에 데이터를 비교해 동일하다면 요소의 위치가 변경되는 작업(reflow)만 수행함.

Proxy 객체

- 데이터의 변경사항을 감시하며, 값 변경 시 렌더링을 다시하도록 유도
- Vue 인스턴스의 data 옵션으로 지정한 객체
- o data로 지정된 객체는 모두 proxy로 래핑됨



☑ 03-16.html

```
<body>
<div id="app">
 <thead>
   >
   번호
   이름
   전화번호
   </thead>
  \t  {\{index + 1\}} 
   {{contact.name}}
   {{contact.tel}}
   </div>
```

✓ 03-16.html

```
<script src="https://unpkg.com/vue"></script>
 <script>
   let = model = {
     contacts: [
       { no: 1011, name: "RM", tel: "010-3456-8299" },
       { no: 1012, name: "정국", tel: "010-3456-8298" },
       { no: 1013, name: "제이홉", tel: "010-3456-8297" },
       { no: 1014, name: "슈가", tel: "010-3456-8296" },
    let vm = Vue.createApp({
     name: "App",
     data() {
       return model;
   }).mount("#app");
 </script>
</body>
```

🗸 Proxy로 래핑된 data

```
> vm.contacts
vm (루트 컴포넌트 인스턴스)
    ▶ [[Handler]]: MutableReactiveHandler
                                                                                               model
    ▶ [[Target]]: Array(4)
     [[IsRevoked]]: false
                                                                   contacts
                                                                                              contacts
> model.contacts

⟨ ▶ (4) [{...}, {...}, {...}, {...}]

                                                                                                객체
                                                                     Proxy
> vm.contacts[0]

⟨ ▶ Proxy(Object) {no: 1011, name: 'RM', tel: '010-3456-8299'}

> model.contacts[0]
                                                                                           model.contacts
                                                                 vm.contacts
⟨ ▶ {no: 1011, name: 'RM', tel: '010-3456-8299'}
    ∨ VIII,CUITUCUS TOTI TTI ETIO □
```

○ 일반 객체 model.contacts는 수정해도 다시 렌더링 되지 않음

Proxy의 배열의 메서드

- 배열의 데이터를 변경하는 메서드들도 래핑되어 있음
- o psuh(), splice(), sort() 등
- → 배열의 내용 변경시 watcher에게 변경을 알려서 다시 렌더링을 일으킴

```
model.contacts[0].tel = "010-9999-9999"
model.contacts.push({no:1201, name: "진", tel: "010-7777-7777"});
→ 다시 렌더링되지 않음
```

```
vm.contacts[1].tel = "010-8888-8888"
vm.contacts.push({no:1202, name: "뷔", tel: "010-6666-6666"});

아 다시 렌더링됨
```

뷰에서는 배열을 많이 사용되고 배열의 메소드도 적용되기 때문에 배열의 메소드가 많이 사용됨'