

2025년 상반기 K-디지털 트레이닝

# Vue 인스턴스

[KB] IT's Your Life



### 1 Vue 인스턴스 개요

### ☑ 애플리케이션 인스턴스

- o Vue.createApp()로 만든 객체
- o 전체 컴포넌트 트리는 애<u>플</u>리케이션 인스턴스를 마운트 할 때 렌더링하게 됨

렌더링: 화면에 출력 vite라는 패킹 툴로 컴포넌트들을 모듈화.

나중에 사용법 ㄱㄱ 그전까지는 하나의 컴포넌트로 실습

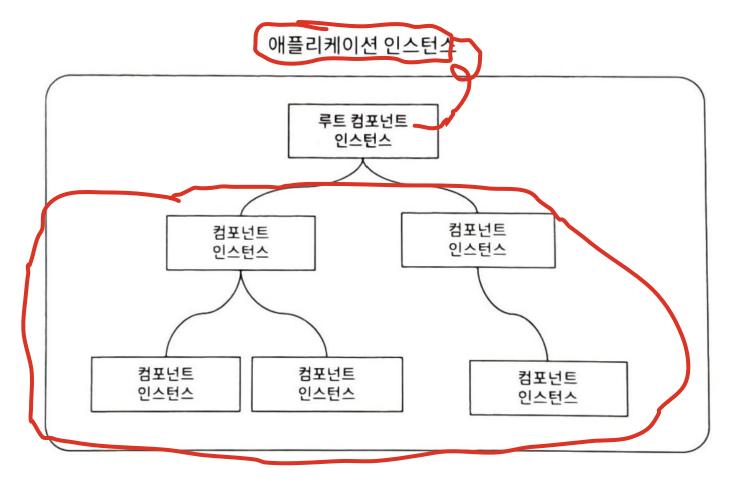
```
let vm = Vue.createApp({
  name: "App",
  data() {
    return { name: ""};
  },
}).mount("#app");
```

가상돔이 만들어지는데

만들어진 서브트리를 기존 트리에 연결시키는 행위를

마운트 한다라고 표현한다

나머지 자식들은 트리 계층으로 컴포넌트 노드



### data 옵션

### data 옵션

- 컴포넌트가 관리하고 추정해야 할 데이터를 등록할 때 사용
  - 관리/추적 → 변경을 탐지하고 추적하여 화면을 갱신함
- <u>반드시 객체를 리턴하는 함수로</u> 정의
- 리턴 객체가 Proxy로 래핑됨

vm.\$data → data 옵션으로 구성된 proxy 객체

그냥 data: 객체 형식으로 지정하면 해당 객체는 전역객체가 되고

vue3버전부터는 위에껀 안스고 data : 객체를반환하는함수 형식이면 함수를 호출할 때마다 객체가 생성됨으로 고유한 객체가 될 수 있다.

- \$, \_로 시작하는 이름은 Vue에서 특수한 용도로 사용하고 있음 vue 자체에서 내부적으로 쓰는
- → 일반 식별자의 시작 문자로 사용하면 안됨

\_는 굉장히 유명한 써드파티 lodash'가 사용하는 \_ 이름이기에 쓰지마

## 계산된 속성(Computed Property)

### ☑ 계산된 속성

리턴된

- 올바른값이 예측 하기 힘들수있따
- data나 다른 속성이 변경될 때 함수가 실행되어 저장된 캐싱된 값
- 함수의 실행은 의존하고 있는 속성 또는 data가 변경될 때 한 번 호출됨
- computed 옵션에 함수를 등록 ✓
- o 함수명이 계산된 속성명이 됨

다른 점. 캐시되어있는 값을 재활용하기에 실제 호출이 명시된것만틈은 하지 않는다.

computed안에 있는 함수안에 잇는 proxy객체에 새로운 값이 적용되면 캐시를 찾지 않고 실제로 호출하여 연산한다

html 출력용 속성 {{}}, v-text, v-html

함수의 리턴값이 속성의 값이 되는 것. 실체는 함수

```
data : function() { ... },
computed :{
function () {...}
```

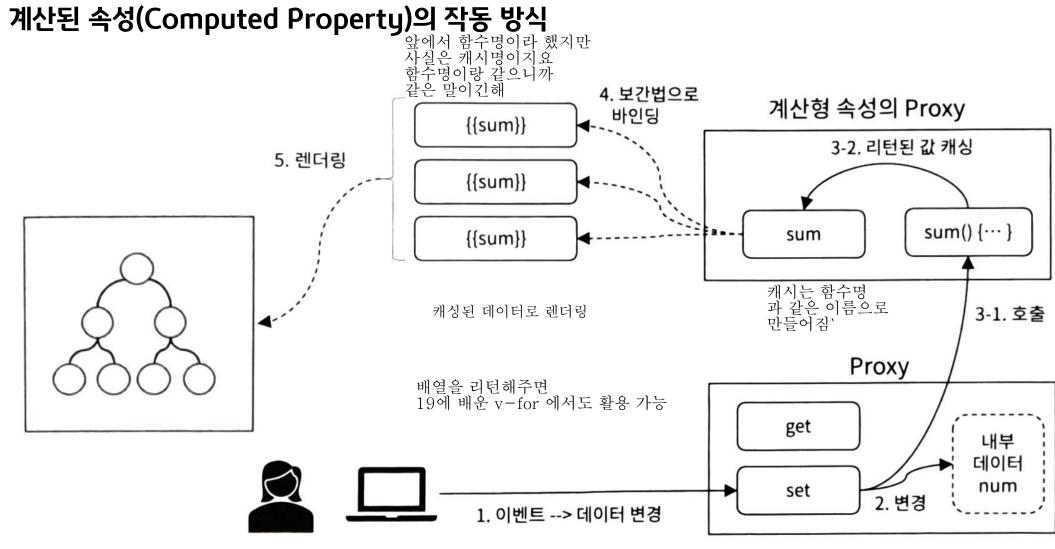
## 계산된 속성(Computed Property)

### **2** 04-03.html

```
3/19에 배운 입력 양방향 바인딩
<div id="app">
  1보다 큰 수: <input type="text" v-model.number="num"
  <br />
  1부터 입력한 값까지의 합: <span>{{sum}}</span><br /> v-text 역할을 하는 {{}} 출력
  1부터 입력한 값까지의 합: <span>{{sum}}</span><br />
  1부터 입력한 값까지의 합: <span>{{sum}}</span><br />
                                                                           다른건 알겠는데 여기서 말하는 this는 뭔데 computed객체 안에 있으니까 computed객체가 this아님? 아님.
가공되어 현재의 컴포넌트가 this가 된다.
</div>
<script src="https://unpkg.com/vue"></script>
                                                                           즉 vm이 된다.
즉 this가 vm이므로
<script>
  let vm = Vue.createApp({
                                                                           this.num은 vm의 data함수의 반환객체의 num변수다.
    name: "App",
    data() { num속성을 갖는 객체를 반환하는 data함수
      return { num: 0 };
        computed: {
                                                                                                          Sources >> = 1
                                                   보다 큰 수: 100
      sum() {
                                                                                                                           (2)
                                                   부터 입력한 값까지의 합: 5050
                                                   부터 입력한 값까지의 합: 5050
                                                                                      Default levels ▼ 1 Issue: ■ 1
                                                   부터 입력한 값까지의 합: 5050
         let n = parseInt(this.num);
                                                                                        You are running a development build of Vue. vue: 10685
        if (Number.isNaN(n)) return 0;
                                                                                        Make sure to use the production build (*.prod.js) when
                                                                                        deploying for production.
        return (n * (n + 1)) / 2;
                                                                                        ## num: 0
                                                                                                                   04-03.html:25
      },
                                                                                        ## num: 1
                                                                                                                   04-03.html:25
                                                                                        ## num: 10
                                                                                                                   04-03.html:25
  }).mount("#app");
                         sum함수의 반환 값이
                                                                                        ## num: 100
                                                                                                                   04-03.html:25
</script>
                         html에서 sum속성을 갖는 태그의 속성값으로 지정되어 렌더링됨
```

## 3 계산된 속성(Computed Property)

캐싱



computed내 함수 내에 변수에 붙은 proxy객체

### 메서드 옵션

스크립트 파트에서 호출하는 방법

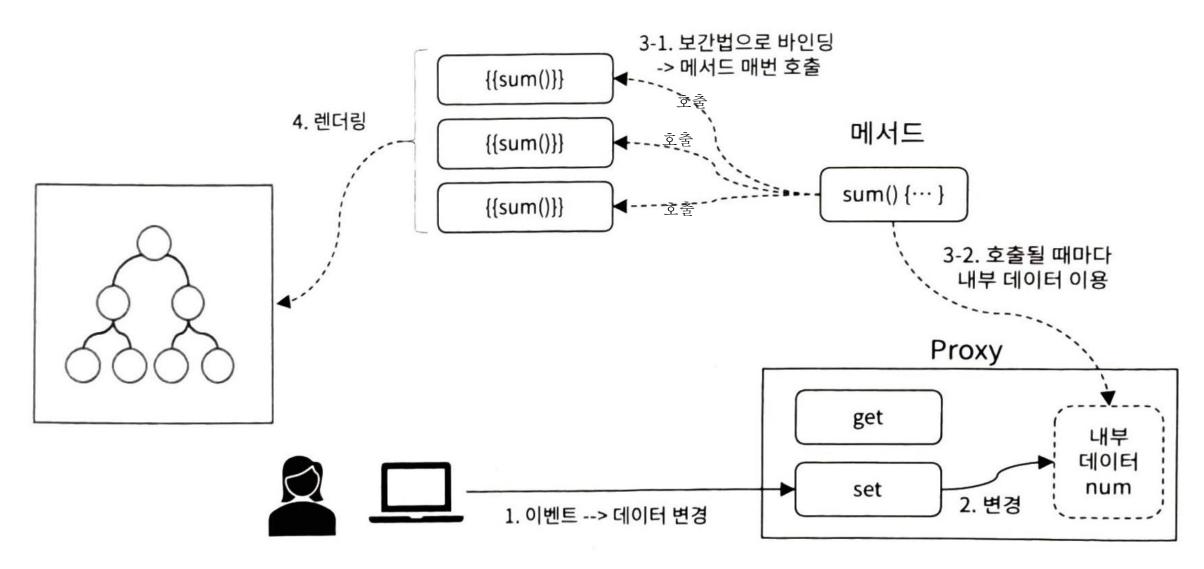
- Vue 인스턴스에서 사용할 메서드를 등록하는 옵션
- 인스턴스에서 직접 호출하거나, 디렉티브의 표현식 ({}}의 표현식에서 호출 가능
- 이벤트 핸들러로 이용가능

```
제일
많이 쓰이는 형태
```

### **2** 04-05.html

```
아까는 계산된 속성으로 사용햇는데
<div id="app">
  1보다 큰 수: <input type="text" v-model.number="num" />
                                                                                      이번엔 함수 호출문 사용.
그래서 함수라는 것도 명시적으로 표시됨,
  <br />
  1부터 입력한 값까지의 합: <span>{{sum()}}}</span><br />
                                                                                      아까와 차이점은 캐시를 사용X. 캐시명을 사용하지 않고
함수를 직접 호출했으니까!
  1부터 입력한 값까지의 합: <span>{<mark>{sum()</mark>}}</span><br />
  1부터 입력한 값까지의 합: <span>{ sum()} } </span><br />
                                                                                      조금 비효율적이죠?
</div>
<script src="https://unpkg.com/vue"></script>
<script>
                                                                                                               Console
                                                                                                                     Sources >> = 1
                                                                                                       Elements
                                                        1보다 큰 수: 100
  let vm = Vue.createApp({
                                                                                                                                        (e)3
                                                        1부터 입력한 값까지의 합: 5050
                                                                                                      top ▼ ◎
                                                                                                               Filter
                                                        1부터 입력한 값까지의 합: 5050
    name: "App",
                                                                                               Default levels ▼ 1 Issue: ■ 1
                                                        1부터 입력한 값까지의 합: 5050
    data() {
                                                                                                  You are running a development build of Vue. vue:10685
                                                                                                  Make sure to use the production build (*.prod.js) when
       return { num: 0 };
                                                                                                  deploying for production.
                                                                                                ## num: 0
                                                                                                                              04-05v.html:25
    methods: {
                                                                                                  Live reload enabled.
                                                                                                                              04-05y.html:61
       sum() {
                                                                                                  ## num: 1
                                                                                                                              04-05y.html:25
         console.log("## num: " + this.num);
                                                                                                  ## num: 1
                                                                                                                              04-05y.html:25
         let n = parseInt(this.num);
                                                                                                  ## num: 1
                                                                                                                              04-05y.html:25
         if (Number.isNaN(n)) return 0;
                                                                                                  ## num: 10
                                                                                                                              04-05y.html:25
                                                                                                  ## num: 10
                                                                                                                              04-05y.html:25
         return (n * (n + 1)) / 2;
                                                                                                  ## num: 10
                                                                                                                              04-05v.html:25
       },
                                                                                                  ## num: 100
                                                                                                                              04-05y.html:25
                                                                                                  ## num: 100
                                                                                                                              04-05y.html:25
  }).mount("#app");
                                                                                                  ## num: 100
                                                                                                                              04-05y.html:25
</script>
```

### ☑ methods 옵션의 실행 방식



### 관찰 속성 Watched Property

- JS 코드에서 데이터가 변경된 경우
  - 이를 감지하여 다른 JS 코드를 실행하고자 하는 경우
  - 주로 긴 처리 시간이 필요한 비동기 처리에 적합

### watch 옵션 객체로 정의

- 함수로 정의
- data 속성명과 동일하게 함수명 정의
- 매개변수
  - 첫 번째 인자: 변경 후의 값
  - 두 번째 인자: 변경 전의 값

```
watch : {
함수( ) {
}
```

프록시라는 것에 의해서 html파트와 스크립트 파트가 서로 영향을 준다

스크립트 파트에 수정이 일어났을 때 그것에 의해서 다른 코드가 실행되기를 원할 때 해바뀌는 부분이 바뀌는 지 모니터링할 때 사용하는 기법.

흔하게 사용되지 않음. 아주 가끔 필요함. 자주쓰지 않음.

computed 내 함수와 규칙이 같다

```
watch내 함수

data 또는 속성명 변경후 x값 변경전 x값

이자 함수명

x는
속성으로도
다른 코드의
변수로도 가용
가능

if (isNaN(result)) this.sum = 0;
else this.sum = result;
```

### 

### 

```
watch: {
    x(current, old) {
       console.log(`## x : ${old} --> ${current}`);
       var result = Number(current) + Number(this.y);
       if (isNaN(result)) this.sum = 0;
       else this.sum = result;
                                                                         watched는 언제 사용되느냐
                                                                         입력할 때마다 실시간으로 작동해야할 대
     y(current, old) {
       console.log(`## y : ${old} --> ${current}`);
                                                                         근데 렌더링말고
       var result = Number(this.x) + Number(current);
                                                                         로직이 작동해야할때
       if (isNaN(result)) this.sum = 0;
       else this.sum = result;
     },
 }).mount("#app");
</script>
```

### ☑ 04-07.html 계산된 속성과의 비교

```
<div id="app">
 x : <input type="text" v-model.number="x" /><br />
 y : <input type="text" v-model.number="y" /><br />
 덧셈 결과 : {{sum}}
</div>
<script type="text/javascript" src="https://unpkg.com/vue"></script>
<script type="text/javascript">
 var vm = Vue.createApp({
    name: "App",
    data() {
      return { x: 0, y: 0 };
                                                              computed 가 더 편하죠?
    computed: {
      sum() {
        let result = Number(this.x) + Number(this.y);
        if (isNaN(result)) return 0;
       else return result;
     },
 }).mount("#app");
</script>
```

### **☑ 04-08.html** 비동기 처리 관찰 넘어가 사전지식 부족

```
<div id="app">
 이름 : <input type="text" v-model.trim="name" placeholder="영문 두글자 이상을 입력하세요" /><br />
 <l
   {{c.name}} : {{c.tel}}
 <div v-show="isLoading">검색중</div>
</div>
<script type="text/javascript" src="https://unpkg.com/vue"></script>
<script type="text/javascript" src="https://unpkg.com/axios"></script>
<script type="text/javascript" src="https://unpkg.com/lodash"></script>
<script type="text/javascript">
 const BASEURL = "https://contactsvc.bmaster.kro.kr";
 var vm = Vue.createApp({
   name: "App",
   data() {
     return { name: "", contacts: [], isLoading: false };
   },
```

### ☑ 04-08.html 비동기 처리 관찰

```
watch: {
      name(current) {
        if (current.length >= 2) {
          this.fetchContacts();
        } else {
          this.contacts = [];
    methods: {
      fetchContacts: _.debounce(function () {
        this.isLoading = true;
        axios.get(BASEURL + \( \) /contacts_long/search/\( \) {this.name}\( \) .then((response) => {
          this.isLoading = false;
          this.contacts = response.data;
        });
      }, 300),
 }).mount("#app");
</script>
```

### **2** 04-09.html

### 비동기 처리 관찰

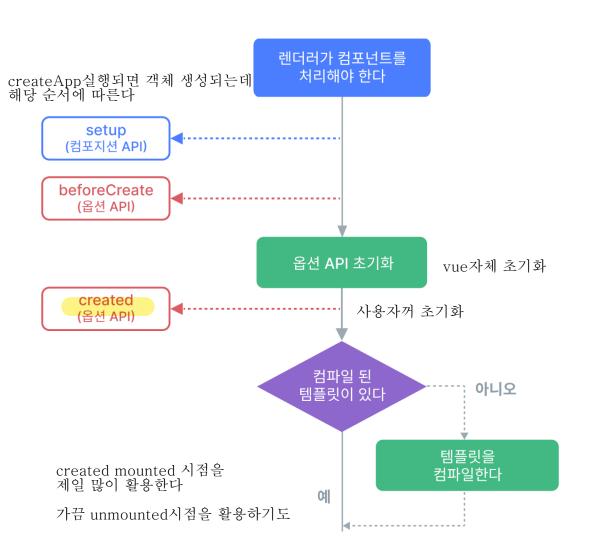
```
<div id="app">
 x : <input type="text" v-model.number="values.x" /><br />
 y : <input type="text" v-model.number="values.y" /><br />
 덧셈 결과 : {{sum}}
</div>
<script type="text/javascript" src="https://unpkg.com/vue"></script>
<script type="text/javascript">
 var vm = Vue.createApp({
    name: "App",
    data() {
      return { values: { x: 0, y: 0 }, sum: 0 };
    },
   watch: {
     values(current) {
       var result = Number(current.x) + Number(current.y);
        if (isNaN(result)) this.sum = 0;
       else this.sum = result;
      },
 }).mount("#app");
</script>
```

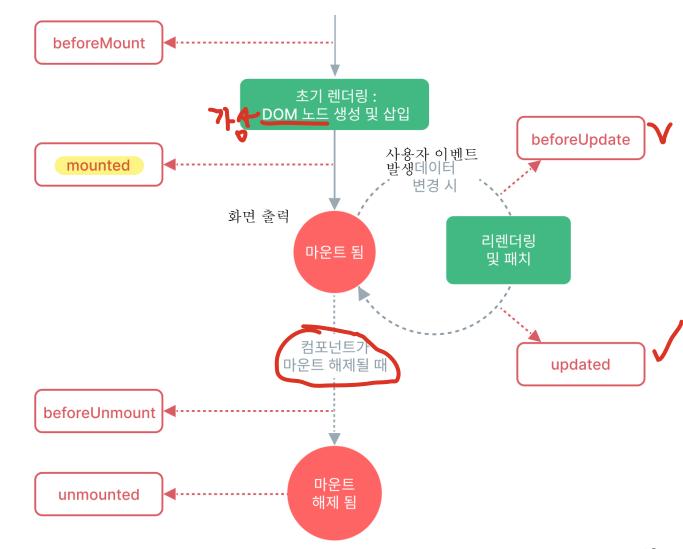
### **2** 04-09.html

### 수정: 비동기 처리 관찰

```
<script type="text/javascript" src="https://unpkg.com/vue"></script>
<script type="text/javascript">
  var vm = Vue.createApp({
    name: "App",
    data() {
      return { values: { x: 0, y: 0 }, sum: 0 };
    },
    watch: {
      values: {
        handler: function (current) {
          var result = Number(current.x) + Number(current.y);
          if (isNaN(result)) this.sum = 0;
          else this.sum = result;
        },
        deep: true,
      },
  }).mount("#app");
</script>
```

### ✓ Vue 생명주기





# ☑ 컴포넌트 생명주기 메서드

	이벤트명이면서	<u>5</u> 43
	이벤트명이면서 어떤 에서드명 에서드명 에서드명 에서드명	설명
<b>∨</b>	beforeCreate	인스턴스가 초기화된 후, 데이터에 대한 관찰 기능, 이벤트가 설정되기 전에 동기적으로 호출
	created	인스턴스 생성이 완료된 후 동기적으로 호출됨. 데이터 관찰 기능, 계산된 속성, 메서드, 이벤트 초기화 등이 모두 완료된 상태임
	beforeMount	마운팅이 시작되기 직전에 호출됨.
	mounted	mount()에 의해 컴포넌트 인스턴스가 HTML Document의 요소에 마운트된 후에 호출
	beforeUpdate	데이터가 변경되고 DOM이 패치되기 전에 호출됨.
	updated	데이터의 변경으로 가상 DOM이 다시 렌더링되고 패치된 후에 호출됨 컴포넌트의 DOM이 업데이트된 상태로 DOM에 종속성이 있는 처리를 이 단계에서 수행
	beforeUnmount	컴포넌트 인스턴스가 언마운트되기 직전에 호출됨
	unmounted	컴포넌트이 인스턴스가 언마운트된 후에 호출됨(자식도 모두 언마운트된 상태임) 컴포넌트의 모든 디렉티브들의 연결이 해제되고, 모든 이벤트 리스너가 제거됨
	errorCaptured	컴포넌트의 자식 컴포넌트 트리에서 오류가 발생하면 컴포넌트 트리를 거슬러 올라와서 이 메서드가 실행됨 컴포넌트 자신에서 발생한 오류는 잡아내지 못함

### **2** 04-10.html

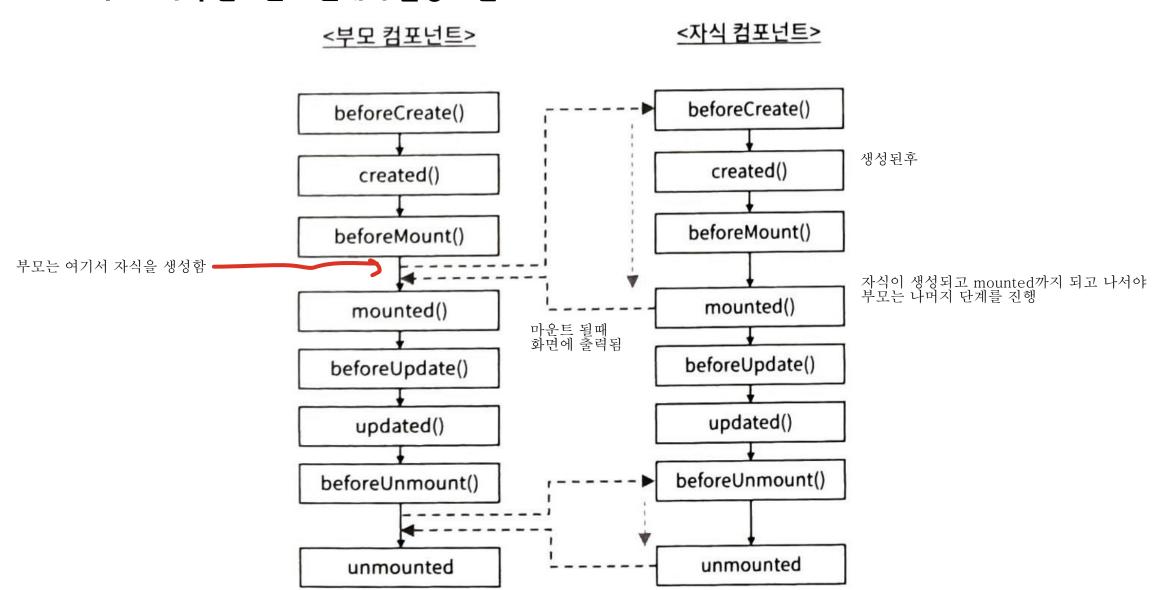
```
<div id="app">
 1보다 큰 수: <input type="text" v-model.number="num" />
 <br />
 1부터 입력한 값까지의 합: <span>{{sum}}</span><br />
</div>
                                      계산된 속성 출력
<script src="https://unpkg.com/vue"></script>
<script>
  let vm = Vue.createApp({
   name: "App",
   data() {
     return { num: 0 };
   }
   computed: {
     sum() {
       let n = parseInt(this.num);
       if (Number.isNaN(n)) return 0;
       return (n * (n + 1)) / 2;
     },
   },
```

### 생명주기 메서드

### **2** 04-10.html

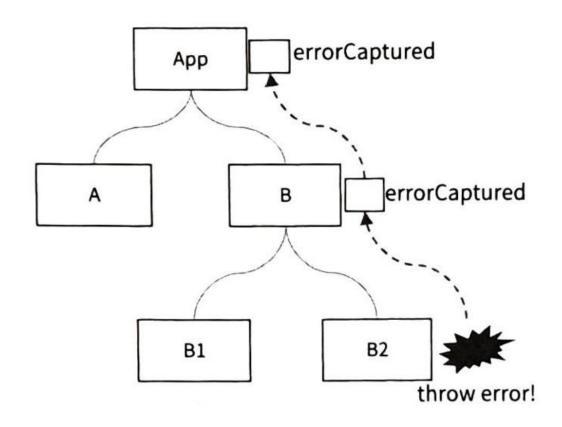
```
created() {벡엔드로부터 데이터 추출코드가 들어잇는 경우가 많다. console.log("## created");
     },
                                                                                                                Elements
                                                                                                                         Console
                                                                                                                                  Sources >> = 1
                                                     1보다 큰 수: 0
                                                                                                                                                         (£)
                                                                                                               top ▼ 🔘
                                                                                                                         Filter
                                                     1부터 입력한 값까지의 합: 0
    mounted() {
                                                                                                       Default levels ▼ 1 Issue: ■ 1
        console.log("## mounted");
                                                                                                         You are running a development build of Vue. vue:10689
                                                                                                         Make sure to use the production build (*.prod.js) when
     },
                                                                                                         deploying for production.
                                                                                                                                             04-10.html:22
                                                                                                         ## created
   Tupdated() {
                                                                                                                                             04-10.html:25
                                                                                                         ## mounted
        console.log("## updated");
     },
                                                                                                                         Console
                                                                                                                                  Sources >> = 1
                                                                                                                Elements
                                                     1보다 큰 수: 100
                                                                                                                                                         (2)
                                                                                                               top ▼ 🔘
                                                                                                                           Filter
                                                     1부터 입력한 값까지의 합: 5050
  }).mount("#app");
                                                                                                      Default levels ▼ 1 Issue: ■ 1
</script>
                                                                                                         You are running a development build of Vue. vue: 10689
                                                                                                         Make sure to use the production build (*.prod.js) when
                                                                                                         deploying for production.
                                                                                                         ## created
                                                                                                                                             04-10.html:22
                                                                                                         ## mounted
                                                                                                                                             04-10.html:25
                                                                                                       ## updated
                                                                                                                                             04-10.html:28
```

### ☑ 부모-자식 컴포넌트 관계의 실행 흐름



### 💟 컴포넌트 트리에서의 에러 처리

○ 프로젝트 수준이나 주요 모듈 단위로 정의해서 중앙 집중화된 에러 처리를 수행함



예외 발생시 부모에게 전달 dom의 부모자식 관계를 타고 계속 에러가 처리될 때까지