

조건부 렌더링

v-if

v-if 디렉티브는 조건부로 블록을 렌더링하는 데 사용됩니다. 블록은 디렉티브 표현식이 truthy 값을 반환하는 경우에만 렌더링됩니다.

```
<h1 v-if="awesome">Vue는 정말 멋지죠!</h1>
```

template

v-else

v-else 디렉티브를 사용하여 v-if 에 대한 "else 블록"을 나타낼 수 있습니다:

```
<button @click="awesome = !awesome">전환</button>
```

template

```
<h1 v-if="awesome">Vue는 정말 멋지죠!</h1>
```

```
<h1 v-else>아닌가요? 🤖</h1>
```

전환

Vue는 정말 멋지죠!

온라인 연습장으로 실행하기

v-else 엘리먼트는 v-if 또는 v-else-if 엘리먼트 바로 다음에 와야 합니다. 그렇지 않으면 인식되지 않습니다.

v-else-if

v-else-if 는 이름에서 알 수 있듯이 v-if 에 대한 "else if 블록" 역할을 합니다. 여러 번 연결될 수도 있습니다:

```
<div v-if="type === 'A'">
  A
</div>
<div v-else-if="type === 'B'">
  B
</div>
<div v-else-if="type === 'C'">
  C
</div>
<div v-else>
  A/B/C 아님
</div>
```

template

v-else 와 마찬가지로 v-else-if 엘리먼트는 v-if 또는 v-else-if 엘리먼트 바로 다음에 와야 합니다.

<template> 에서 v-if

v-if 는 디렉티브이므로 단일 엘리먼트에 연결해야 합니다. 하지만 둘 이상의 엘리먼트를 전환하려면 어떻게 해야 할까요? 이 경우 보이지 않는 래퍼 역할을 하는 <template> 엘리먼트에 v-if 를 사용할 수 있습니다. 최종 렌더링된 결과에는 <template> 엘리먼트가 포함되지 않습니다.

```
<template v-if="ok">
  <h1>제목</h1>
  <p>단락 1</p>
  <p>단락 2</p>
</template>
```

template

v-else 및 v-else-if 는 <template> 에서도 사용할 수 있습니다.

v-show

엘리먼트를 조건부로 표시하는 다른 옵션은 v-show 디렉티브입니다. 사용법은 대체로 동일합니다:

```
<h1 v-show="ok">안녕!</h1>
```

template

차이점은 v-show 가 있는 엘리먼트는 항상 렌더링되고 DOM에 남아 있다는 것입니다. v-show 는 엘리먼트의 display CSS 속성만 전환합니다.

v-show 는 <template> 엘리먼트를 지원하지 않으며 v-else 와 상호작용하지 않습니다.

v-if vs v-show

v-if 는 "실제" 조건부 렌더링입니다. 왜냐하면 조건부 블록이 전환될 경우, 블록 내 이벤트 리스너와 자식 컴포넌트가 제대로 제거되거나 재생성되기 때문입니다.

또한 v-if 는 **게으르므로**(lazy), 초기 조건이 false면 아무 작업도 수행하지 않습니다. 조건부 블록은 조건이 true가 될 때까지 렌더링되지 않습니다.

이에 비해 v-show 는 훨씬 간단합니다. 엘리먼트는 CSS 기반으로 전환 되므로, 초기 조건과 관계없이 항상 렌더링 됩니다.

일반적으로 v-if 는 전환 비용이 더 높고, v-show 는 초기 렌더링 비용이 더 높습니다. 따라서 매우 자주 전환해야 하는 경우 v-show 를, 실행 중에 조건이 변경되지 않는 경우 v-if 를 사용하는 것이 좋습니다.

v-if with v-for

참고

v-if 와 v-for 를 함께 사용하는 것은 권장되지 않습니다.

자세한 내용은 스타일 가이드 를 참조하세요.

엘리먼트에 v-if 와 v-for 를 함께 사용하면 v-if 가 먼저 평가됩니다. 자세한 내용은 리스트 렌더링 가이드를 참조하세요.