



# 조건부 렌더링

Vue School의 무료 동영상 강의 보기

## v-if

디렉티브 `v-if` 는 블록을 조건부로 렌더링하는 데 사용됩니다. 디렉티브의 표현식이 참(truthy) 값을 반환할 때만 블록이 렌더링됩니다.

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

template

## v-else

`v-if` 에 대한 "else 블록"을 나타내기 위해 `v-else` 디렉티브를 사용할 수 있습니다:

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

template

```
<h1 v-if="awesome">Vue는 멋져요!</h1>
<h1 v-else>오 안돼 😠</h1>
```

토글

Vue는 멋져요!



v-else 요소는 반드시 v-if 또는 v-else-if 요소 바로 뒤에 위치해야 하며, 그렇지 않으면 인식되지 않습니다.

---

## v-else-if

v-else-if 는 이름에서 알 수 있듯이 v-if 에 대한 "else if 블록" 역할을 합니다. 또한 여러 번 연속으로 사용할 수 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/C가 아님  
</div>
```

template

v-else 와 마찬가지로, v-else-if 요소도 반드시 v-if 또는 v-else-if 요소 바로 뒤에 위치해야 합니다.

---

## <template>에서의 v-if

v-if 는 디렉티브이기 때문에 반드시 하나의 요소에만 부착되어야 합니다. 하지만 여러 요소를 토글하고 싶다면 어떻게 해야 할까요? 이 경우 <template> 요소에 v-if 를 사용할 수 있습니다. <template> 은 보이지 않는 래퍼 역할을 하며, 최종 렌더링 결과에는 <template> 요소가 포함되지 않습니다.

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

template



## v-show

요소를 조건부로 표시하는 또 다른 방법은 `v-show` 디렉티브를 사용하는 것입니다. 사용법은 거의 동일합니다:

```
<h1 v-show="ok">안녕하세요!</h1>
```

template

차이점은 `v-show` 가 적용된 요소는 항상 렌더링되어 DOM에 남아 있다는 점입니다. `v-show` 는 요소의 `display` CSS 속성만 토글합니다.

`v-show` 는 `<template>` 요소를 지원하지 않으며, `v-else` 와도 함께 사용할 수 없습니다.

## v-if vs. v-show

`v-if` 는 "진짜" 조건부 렌더링입니다. 조건부 블록 내의 이벤트 리스너와 자식 컴포넌트가 토글 시 적절하게 파괴되고 다시 생성되도록 보장합니다.

`v-if` 는 또한 **지연(lazy)** 동작을 합니다. 초기 렌더 시 조건이 거짓이면 아무 작업도 하지 않으며, 조건이 처음으로 참이 될 때까지 조건부 블록이 렌더링되지 않습니다.

반면, `v-show` 는 훨씬 단순합니다. 초기 조건과 상관없이 요소가 항상 렌더링되며, CSS 기반 토글만 수행합니다.

일반적으로 `v-if` 는 토글 비용이 더 높고, `v-show` 는 초기 렌더 비용이 더 높습니다. 매우 자주 토글해야 한다면 `v-show` 를, 런타임에 조건이 거의 바뀌지 않는다면 `v-if` 를 사용하는 것이 좋습니다.

## v-for 와 함께 사용하는 v-if

`v-if` 와 `v-for` 가 같은 요소에 함께 사용될 때, `v-if` 가 먼저 평가됩니다. 자세한 내용은 리스트 렌더링 가이드를 참고하세요.



## ⚠️ 심고

암묵적인 우선순위 때문에 같은 요소에 `v-if` 와 `v-for` 를 함께 사용하는 것은 **권장되지 않습니다.** 자세한 내용은 [리스트 렌더링 가이드](#)를 참고하세요.

[GitHub](#)에서 이 페이지 편집

< Previous

[클래스 및 스타일 바인딩](#)

Next >

[리스트 렌더링](#)