Conditional Rendering

Watch a free video lesson on Vue School

`v-if`

The directive `v-if` is used to conditionally render a block. The block will only be rendered if the directive's expression returns a truthy value.

```template
<h1 v-if="awesome">Vue is awesome!</h1>
```

`v-else`

You can use the `v-else` directive to indicate an "else block" for `v-if`:

```template
<button @click="awesome = !awesome">Toggle</button>

<h1 v-if="awesome">Vue is awesome!</h1>
<h1 v-else>Oh no 😢</h1>
```

Toggle

Vue is awesome!

Try it in the Playground

`v-else-if`

The `v-else-if`, as the name suggests, serves as an "else if block" for `v-if`. It can also be chained multiple times:

```template
<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>
  Not A/B/C
</div>
```

Similar to `v-else`, a `v-else-if` element must immediately follow a `v-if` or a `v-else-if` element.

`v-if` on `<template>`

Because `v-if` is a directive, it has to be attached to a single element. But what if we want to toggle more than one element? In this case we can use `v-if` on a `<template>` element, which serves as an invisible wrapper. The final rendered result will not include the `<template>` element.

```template
<template v-if="ok">
  <h1>Title</h1>
  <p>Paragraph 1</p>
  <p>Paragraph 2</p>
</template>
```

`v-else` and `v-else-if` can also be used on `<template>`.

`v-show`

Another option for conditionally displaying an element is the `v-show` directive. The usage is largely the same:

```template
<h1 v-show="ok">Hello!</h1>
```

The difference is that an element with `v-show` will always be rendered and remain in the DOM; `v-show` only toggles the `display` CSS property of the element.

`v-show` doesn't support the `<template>` element, nor does it work with `v-else`.

`v-if` vs. `v-show`

`v-if` is "real" conditional rendering because it ensures that event listeners and child components inside the conditional block are properly destroyed and re-created during toggles.

`v-if` is also **lazy**: if the condition is false on initial render, it will not do anything - the conditional block won't be rendered until the condition becomes true for the first time.

In comparison, `v-show` is much simpler - the element is always rendered regardless of initial condition, with CSS-based toggling.

Generally speaking, `v-if` has higher toggle costs while `v-show` has higher initial render costs. So prefer `v-show` if you need to toggle something very often, and prefer `v-if` if the condition is unlikely to change at runtime.

`v-if` with `v-for`

When `v-if` and `v-for` are both used on the same element, `v-if` will be evaluated first. See the list rendering guide for details.

⚠ Note

It's **not** recommended to use `v-if` and `v-for` on the same element due to implicit precedence. Refer to list rendering guide for details.

‹ Previous
Class and Style Bindings

Next ›
List Rendering